

Tracking A Non-cooperative Mobile Target Using Low-Power Pulsed Doppler Radars

Jong Hyun Lim
Department of Computer Science
Johns Hopkins University
Baltimore, MD
Email: ljh@cs.jhu.edu

I-Jeng Wang
Johns Hopkins University
Applied Physics Laboratory
Laurel, MD
Email: i-jeng.wang@jhuapl.edu

Andreas Terzis
Department of Computer Science
Johns Hopkins University
Baltimore, MD
Email: terzis@cs.jhu.edu

Abstract—Most target tracking applications developed for wireless sensor networks thus far employ passive sensors which detect the target’s emissions. Their performance is thereby dictated by the magnitude of the target’s signal and the receivers’ sensitivity. Instead, we propose using a network of low-power Doppler radars that actively measure the target’s radial velocity. Nodes combine their measurements to solve a system of non-linear equations that estimate the target’s position and velocity. Because these equations have no closed-form solutions we solve them using numerical methods. These methods however can lead to local minima or even diverge in the presence of even small measurement noise. For this reason we couple the proposed numerical method with an Extended Kalman filter that models the target’s movement along a straight line. Clearly the target does not always follow a linear path and so we augment this simple Kalman filter with a method that detects the target’s turns and updates the filter’s dynamical model accordingly. The combination of the two approaches improve tracking accuracy compared to the numerical solution alone. Results from simulations and a prototype implementation suggest that the combined solution can effectively track a mobile target with average localization error as low as 25 cm in a 10×10 m outdoor field.

I. INTRODUCTION

Target localization has been an active research area from the early days of wireless sensor networks (WSNs). Nevertheless, most approaches proposed in the research literature either assume that the intended target has a device that actively transmits beacons (e.g., [9], [20]) or that the target naturally generates signals that can be detected (e.g., [1]). Nevertheless, there exist scenarios (e.g., intruder detection) in which one needs to detect targets that do not cooperate by intentionally or unintentionally emitting signals.

We focus on such non-cooperative targets and propose a tracking system that uses low-power pulsed Doppler radars to actively estimate a target’s location. Specifically, each of the system’s radars independently measures a mobile target’s Doppler velocity by periodically emitting sinusoidal pulses and processing the reflected signals. A PC-class base station then collects the estimated Doppler velocities by at least four radars and estimates the position and velocity of the target by solving a system of non-linear equations. Because these equations do not have a closed form solution, we use numerical methods to solve them. The issue is that these numerical methods can converge to local minima or even worse diverge in the

presence of measurement noise. Instead, we use an Extended Kalman filter to overcome this limitation. The Kalman filter uses a linear dynamical model to predict the target’s position and velocity and uses the raw Doppler velocity measurements to correct the possibly inaccurate prediction.

While the Extended Kalman filter outperforms the numerical method alone, it assumes that the target’s velocity vector is constant. For this reason, the filter’s correction step is insufficient when the target turns. We introduce a *velocity correction* (VC) step to remove this shortcoming. The basestation solves a system of two linear equations during the VC step to update the target’s velocity. The EKF then uses the updated velocity to improve the target’s estimated position. We term this modified EKF, the EKF_{vc} .

Experimentally validating the performance of EKF_{vc} under all possible target trajectories is implausible. Instead, we developed a simulator whose error models are experimentally derived from measurements collected using the Bumblebee Doppler radars [16] and a robotic target. The robot’s constant speed and pre-determined trajectory allowed us to know the target’s ground truth location and thus estimate measurement errors. Using these realistic models we evaluate the accuracy of EKF_{vc} for simulated target trajectories with one and multiple turns. Furthermore, we analyze the error bounds for the target’s position and velocity by using a Cramér-Rao (CR) bound. This bound allows us to find the radar geometry that minimizes position and velocity errors. Finally, we prototyped our approach using TelosB motes connected to the Bumblebee radars and validated our simulation results with outdoor experiments. These results show that EKF_{vc} reduces the position, speed, and orientation errors by 35%, 48%, and 56% respectively, compared to the standard EKF.

Our contributions: (1) We developed a *velocity correction* approach that improves the estimation accuracy of the standard EKF. This method can also be applied to other prediction-correction based filtering schemes, such as Bayesian filtering. (2) We combine modeling, simulation, and prototyping to quantitatively investigate the efficacy of using low-power, pulsed Doppler radars for tracking a non-cooperative target.

The rest of this paper is organized as follows: Section II outlines the differences between our system and previous

target tracking proposals. Section III details the system’s architecture and the algorithms used to estimate the target’s location and velocity, while Section IV describes its implementation. Section V presents quantitative results from simulations and outdoor experiments. After concluding in Section VII, we further analyze the estimation error bounds by using a Cramér-Rao bound in Appendix A.

II. RELATED WORK

Dutta et al. proposed the use of radars in the context of WSNs [4]. They designed a mote platform equipped with a UWB radar and presented mechanisms for target detection and characterization based on the waveform characteristics of different targets. Along the same lines, Ly and Liang use radar sensor networks to detect multiple targets in a surveillance application through a maximum likelihood detection algorithm [12]. Instead, we are interested in estimating the location of the target and tracking its trajectory.

Multi-static radar networks localize a target by using a single transmitter and multiple receivers. The location of the target is determined using Time Difference of Arrival (TDOA) by the intersection of the equi-TOA ellipses whose foci are the transmitter and each of the receivers. If received signal strength is used instead, the target’s location is determined by the intersection of so-called *Cassini ovals*. Chiani et al. considered both approaches and provided bounds regarding the theoretical coverage and localization uncertainty of a multi-static network of UWB radars [14]. Instead, each of the radars in our network operates in a mono-static configuration and we combine their individual Doppler velocity measurements to estimate the target’s location and speed.

The most relevant work using Doppler shift for tracking mobile nodes is by Kusy et al. [11]. However, the mobile targets in that case must transmit sinusoidal signals that the motes receive through their radios before analyzing their Doppler shifts. Instead, we use a dedicated radar which does not require any cooperation from the target and frees the mote’s radio. Furthermore, Kusy et al. improve the accuracy of the original EKF estimate using constrained optimization. This additional step constrains the potential location of the target to be within a circle with known radius, effectively rejecting unreasonable location estimates caused by poor EKF estimates as the target maneuvers. On the other hand, we detect target maneuvering with our velocity correction, and then make use of the velocity change to improve EKF position estimation.

Smith et al. have used outlier rejection to improve EKF estimations [17]. Specifically, they use the EKF’s ability to predict measurements, to compare between the actual measurement and its estimate. If the difference between the two is greater than a predetermined threshold, the system discards the (faulty) measurements to avoid polluting the state estimation process. On the other hand, we focus on detecting the target’s maneuvering rather than filtering the radars’ measurements.

Chang et al. localize a stationary target using spinning radio beacons that generate a Doppler-shifted signal [3]. Both the target and reference nodes receive the signal and send it to a

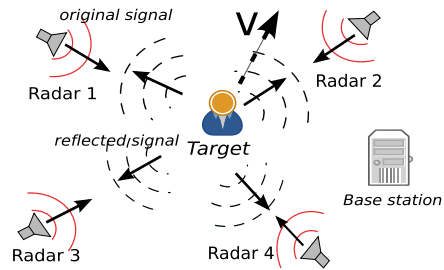


Fig. 1. Each radar independently measures the Doppler velocity of a mobile target and transmits it to a base station.

base station that analyzes the relative angles between a target and a beacon and uses triangulation to localize a target. While this scheme may be applicable to localizing mobile targets, its performance would depend critically on the beacon’s rotation speed relative to the target’s speed.

Finally, camera sensor networks can track non-cooperative targets. Funiak et al. addressed the problem of simultaneously estimating the cameras’ poses and tracking a target [5], while Teixeira et al. used cameras to localize and count people indoors [19]. Both methods however, do not work in the dark and require resource-intensive computer vision algorithms.

III. APPROACH

Our system comprises a network of radar motes and a PC-class base station.

Radar Mote. Each mote uses its locally connected radar to periodically generate sinusoidal pulses and analyzes the reflection of these pulses from the target to estimate the target’s radial velocity based on the Doppler Effect [21]. Each node subsequently delivers its locally computed radial velocity to the base station. The radar literature refers to the radars operating independently as *monostatic*. Thereby, our system comprises multiple monostatic radars (see Fig.1 in a configuration which we term as *multi-monostatic*). The benefit of this approach is that it does not require tight synchronization among the network’s nodes.

Base Station. After collecting Doppler velocities from at least four radar motes, the base station estimates the position and velocity of the target by solving a system of non-linear equations which relate the knowns (i.e., radar positions), the unknowns (target position and true velocity), and the measured Doppler velocities (§III-A). Standard numerical methods, such as Newton-Raphson, do not work well for solving these equations as they tend to converge to local minima or diverge due to measurement noise. Instead, we employ a modified Extended Kalman Filter (EKF) to effectively filter out measurement noise. The EKF predicts the target’s position and velocity by a dynamic model and corrects these possibly incorrect estimates by using Doppler velocity measurements.

A. Doppler Velocity Estimation

Doppler Effect. The Doppler effect describes the frequency change of a periodic signal when it reflects of a moving target.

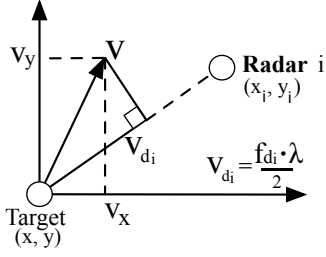


Fig. 2. Two dimensional geometry between the known position of Radar 2, the target's unknown position and velocity, and measured Doppler velocity (\vec{v}_{d_i}). Vector \vec{v}_{d_i} is the projection of the target's true velocity (\vec{v}) onto the axis between Radar i and the target.

The radar literature refers to the magnitude of this change as *Doppler frequency* (f_d). Figure 2 presents the relation between f_d and Doppler velocity (v_d), where λ denotes the wavelength of the transmitted signal.

Measuring Doppler velocity in the time domain. A mote samples the radar's pre-processed output waveform to measure v_d . This output waveform is a sinusoidal signal whose frequency is equal to f_d . Using frequency domain analysis (e.g., FFT) to estimate f_d on the mote is infeasible due to the mote's limited resources [6]. At the same time, transmitting raw samples from the output waveform is also impractical due to the high data volume.

Instead, we use the well-known auto-correlation (AC) method to estimate f_d in the time domain [15]. AC is computationally much simpler than FFT as it only involves summations and products of complex number samples. Although it is simpler, it is equivalent to FFT with respect to the accuracy of estimating f_d [13]. Moreover, AC requires $O(1)$ storage and smooths any possible synchronization errors among the network's nodes as it calculates the target's average speed over a time interval. Readers interested in more information regarding AC should refer to [15].

B. Target Tracking

The base station uses the radars' reported Doppler velocities (v_{d_i}) to estimate the position (x, y) and velocity (v_x, v_y) of a mobile target. To do so, we derive the relationship between the knowns and unknowns based on the model presented in [11]. As Figure 2 illustrates, assuming that the (x_i, y_i) position of a radar i is known, then:

$$v_{d_i} = F_i(\mathbf{x}) = \frac{v_x(x_i - x) + v_y(y_i - y)}{\sqrt{(x_i - x)^2 + (y_i - y)^2}}, \quad (1)$$

where we simply denote the four unknown parameters as $\mathbf{x} = [x, y, v_x, v_y]^T$.

As (1) has four unknowns, we need at least four radars to track a target. The standard method for solving (1) is Newton-Raphson. However, as we will show in Section V-A, this method leads to local minima even when small measurement noise is present. In the next section, we introduce an alternative estimation technique that is robust to measurement noise.

C. Improving Tracking Accuracy by Extended Kalman Filter

The Extended Kalman Filter (EKF) is a tool for non-linear estimation from noisy measurements [8]. It recursively predicts the system's next state based on a state transition model and corrects state estimation errors by incorporating collected measurements. It can be shown that as the number of iterations grows, the estimates eventually converge to the ground truth, assuming that the noise follows a i.i.d zero-mean normal distribution [8].

The EKF consists of two steps for updating the target's state $\mathbf{x} \in \mathcal{R}^4$: (1) *time step* and (2) *measurement step*. Furthermore, the a priori error covariance matrix $P_t^- \in \mathcal{R}^{4 \times 4}$ is updated during the time step, while the posteriori error covariance matrix $P_t \in \mathcal{R}^{4 \times 4}$ is updated during the measurement step. We denote the time step estimates as \mathbf{x}_t^- and P_t^- , whereas the absence of “-” signifies a measurement step. In addition, the subscript t indicates that the operation occurs at time t .

EKF Time Step. Before it uses the radars' measurements, the EKF predicts the target's position and velocity based on a model of its movement. Specifically, the EKF time step uses the state estimates from the previous measurement step (\mathbf{x}_{t-1} , P_{t-1}) and a transitional model (A) to predict the next state (\mathbf{x}_t^- , P_t^-):

$$\mathbf{x}_t^- = A\mathbf{x}_{t-1} + \mathbf{w}_{t-1}, \quad P_t^- = AP_{t-1}A^T + Q_{t-1}, \quad (2)$$

whereas the process error noise vector $\mathbf{w}_{t-1} \sim N(0, Q)$ and $Q \in \mathcal{R}^{4 \times 4}$ is the process error covariance matrix describing the overall uncertainty of the transition model. For the transition model (A) we assume that the target moves with a constant velocity and thus:

$$A = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad Q = \begin{pmatrix} 0.5 & 0 & \Delta t & 0 \\ 0 & 0.5 & 0 & \Delta t \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{pmatrix},$$

where Δt denote the time between time steps $t-1$ and t .

EKF Measurement Step. Once new measurements are collected, the EKF corrects the prediction generated during the time step. This step is necessary because the linear transition model (A) may be insufficient to represent and predict all the dynamics of a mobile target.

$$K_t = P_t^- \mathbf{J}_t^T (\mathbf{J}_t P_t^- \mathbf{J}_t^T + R)^{-1}$$

$$\mathbf{x}_t = \mathbf{x}_t^- + K_t(\mathbf{z}_t - \mathbf{F}(\mathbf{x}_t^-)), \quad P_t = (I - K_t \mathbf{J}_t) P_t^-, \quad (3)$$

where the *Kalman gain* $K_t \in \mathcal{R}^{4 \times 4}$, is a weight factor that decides the contribution of the measurement set $\mathbf{z}_t = [v_{d1}, \dots, v_{d4}]^T$ on the new state at time t . $\mathbf{J}_t \in \mathcal{R}^{4 \times 4}$ is the Jacobian matrix of the non-linear equations $\mathbf{F} = [F_1, \dots, F_4]^T$ evaluated at \mathbf{x}_t^- . The measurement error covariance matrix $R \in \mathcal{R}^{4 \times 4}$ is used for describing errors added to the measurements \mathbf{z}_t —assuming the measurement error follows $N(0, R)$. In our implementation, we set $R = 0.4I$. We refer readers interested in a detailed description of the EKF to [8].

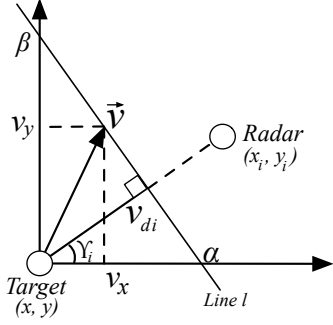


Fig. 3. Estimating the target's velocity $\vec{v} = (v_x, v_y)$ when target position (x, y) is assumed to be known. Given Doppler velocity v_{di} , the possible position of \vec{v} is constrained by line l . We use this knowledge to correct the target's velocity in the EKF model.

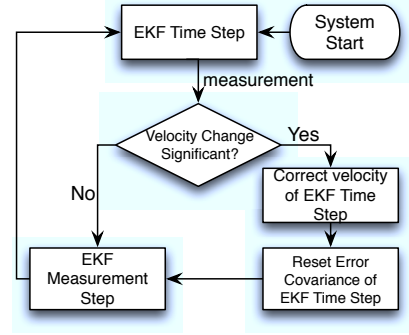


Fig. 4. Flowchart of the EKF process, including velocity correction. The result of the velocity correction is applied only when the correction is larger than a pre-determined threshold (see Section V).

D. Improving EKF Estimation through Velocity Correction

Since our simplified transition model (A) assumes a constant target velocity, the EKF prediction may be quite poor when the target turns. Although EKF corrects the inaccurate prediction with measurements, the correction may not be enough, especially when R is large. In this case, the EKF reduces the measurements' contribution during the measurement step and therefore does not correct the prediction by much. On the other hand, making R smaller could lead to undesirable fluctuations in the EKF estimate driven by the measurement noise.

To compensate for the poor prediction without changing R , we propose a *velocity correction* (VC for short) technique. The VC is a new step introduced between the EKF time and measurement steps. It is evident from (3) that the more accurate the prediction \mathbf{x}_t^- is, the better estimation \mathbf{x}_t becomes. Thus, rather than retroactively trying to compensate for the prediction error during the measurement step, VC tries to improve the velocity prediction of the EKF time step and pass this modified velocity to the EKF measurement step.

Figure 3 presents the scenario in which we improve the velocity prediction of a target. As before, we denote the Doppler velocity measured at radar i as v_{di} , while the angle γ_i ($= \arctan(y_i - y)/(x_i - x)$). We assume here that the target's estimated position (x, y) is known. To do so, we set the target's position (x, y) to be equal to the position estimate from the EKF time step.

We can then express the velocity vector $\vec{v} = (v_x, v_y)$ as a function of v_{di} and γ_i . To do so we consider the line l (see Figure 3) that intersects the x and y axes at points α and β respectively. It is easy to see that $\alpha = \frac{v_{di}}{\cos \gamma_i}$ and $\beta = \frac{v_{di}}{\cos(\pi/2 - \gamma_i)} = \frac{v_{di}}{\sin \gamma_i}$. Line l is then defined by the following equation:

$$y = -\frac{\beta}{\alpha}x + \beta \Rightarrow \quad (4)$$

$$y = -\frac{\cos \gamma_i}{\sin \gamma_i}x + \frac{v_{di}}{\sin \gamma_i} \Rightarrow \quad (5)$$

$$y \sin \gamma_i = -\cos \gamma_i x + v_{di} \Rightarrow \quad (6)$$

$$v_{di} = \cos \gamma_i x + \sin \gamma_i y \quad (7)$$

Setting $(x, y) = (v_x, v_y)$ in (7), we get

$$v_{di} = v_x \cos \gamma_i + v_y \sin \gamma_i + \epsilon_i, \quad (8)$$

where the term ϵ_i models measurement noise. We solve (8) numerically with one or more line equations from other radars.

We, however, do not use the velocity suggested by the VC step all the time. Rather, we do so when the velocity proposed by the VC step shows significant difference from the one computed during the EKF time step. This decision is made based on a static threshold in our system (Section V shows our experimental results to find the threshold). If the difference is above the threshold, we use the velocity calculated by the VC step in the EKF time step. This modified estimation is then passed to the EKF measurement step to improve tracking performance. While the new velocity estimate is not generally equal to the target's true velocity, the Cramér-Rao analysis in Appendix A proves that the error in target velocity is not large if we assume that our estimate of the target's position is reasonably accurate.

Moreover, solving (8) does not require us to know the velocity in advance; thus, the new velocity estimate is only affected by the position prediction from the EKF time step and the Doppler velocity measurements. Last, the target's velocity v_x and v_y have a linear relationship with v_{di} and therefore the impact of the measurement noise ϵ_i is not significant.

The VC mechanism described above can be incorporated to other filtering methods based on the prediction-correction framework, such as Bayesian Filtering. This is so because VC is independent of any particular filtering scheme. In Section V-D, we show the estimation results of the Bayesian filtering method.

Resetting Error Covariance Matrix. We compensate for the consequence of unexpected velocity change by correcting the v_x and v_y elements of the state \mathbf{x}_t^- from the time step. Thus, we also need to revise the covariance matrix (P_t^-) used during the EKF time step. Specifically, whenever we apply the VC step, we reset P_t^- with the new process error covariance (\hat{Q}):

$$\bar{Q} = \begin{pmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.2\Delta t & 0 \\ 0 & 0 & 0 & 0.2\Delta t \end{pmatrix}, \quad P_t^- = AP_{t-1}A^T + \bar{Q},$$

where Δt denotes the time elapsed from the last update. This suggests that the error increases with Δt . Furthermore, \bar{Q} only increases the magnitudes of the velocity components because we do not change our prediction of the target's position.

The introduction of a new \bar{Q} implies that we regard the velocity difference—between the one used during the EKF time step and that of actual target—as an error originating from the naive state transition model (A). This is a natural perspective because our model does not know the target's future velocity. Thus, allowing the simple model to generate the error and subsequently correcting the velocity prediction and error covariance is a simple, yet effective solution.

Figure 4 presents the overall EKF estimation process, including velocity correction. The whole process repeats as new measurements become available.

IV. IMPLEMENTATION

We developed a prototype of the proposed target tracking system using TmoteSky motes running TinyOS 2.x connected to Bumblebee radars [16]. The Bumblebee radar is a battery-operated device specially designed for use in WSNs. It transmits sinusoidal pulses centered at 5.4GHz and is able to scan a 60° conical area. It detects targets with speeds ranging from 2.6 cm/s to 2.6 m/s at a maximum distance of 10 m. Finally, the Bumblebee's maximum current draw is 12 mA. The radar connects to the Tmote's ADC pins. Per the mote's request, it delivers digital samples of the pre-processed reflected sinusoidal signal that only bears the Doppler frequency. The frequency of this output signal ranges from 1 Hz to 100 Hz. Thus, based on the Nyquist sampling theorem, the mote needs to sample the radar's output signal at a rate of 200 Hz.

The basestation in the current proof of concept collects all the motes' measurements. Specifically, the basestation broadcasts a beacon to the radar motes when the system initializes. Assuming that the beacon arrives simultaneously at every mote, each mote starts sampling its radar at 200 Hz for 900 ms thereby generating 180 samples, each 4 bytes long. The node then stops sampling the radar, calculates the average Doppler velocity using the auto-correlation method and sends the estimate to the base station. After receiving all the nodes' estimates, the basestation estimates the location and velocity of the target. The same process repeats once every second.

Rather than taking turns, radars transmit their pulses concurrently and can therefore potentially interfere with each other. Nevertheless, we found that if the distances between the radars are larger than 5 m, the interference becomes negligible to the point where Doppler velocity measurements are as accurate as when a single radar is active.

V. EVALUATION

We evaluate the performance of our system by measuring two errors: position error and velocity error. The position error

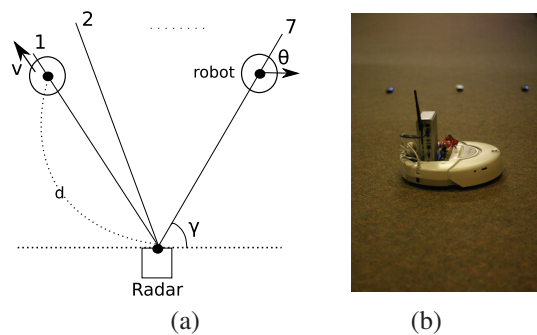


Fig. 5. (a) Configuration and parameters used in the robot experiment. The distance between the center of robot and the radar is d . The robot moves with speed v (20 cm/s, 30 cm/s, 40 cm/s). The angle γ between the x -axis and the line that connects the radar and the robot (Γ line) changes at 10° increments. Angle θ denotes how much the robot deviates from the Γ line (θ ranges from 0° to 90° at 15° increments). (b) IRobot Create used for collecting measurements, with a Linux computer on top.

measures the Euclidean distance between the target's true and estimated positions. We divide the velocity error into two components: orientation (angular degree, $0^\circ \sim 360^\circ$) and speed (cm/s). We measure the orientation and velocity errors by calculating the absolute value of the difference between the ground truth and estimated value. Using these metrics, we compare the performances of three methods: Newton-Raphson, EKF, and EKF_{vc} .

A. Simulation

Testing all velocity and direction changes of a real target is a formidable task. To facilitate the tests, we developed a simulator that incorporates realistic Doppler velocity models. We derive these models from measurements collected in a real environment using the IRobot Create robot [7], shown in Figure 5(b), as a target. Because the robot follows pre-defined paths at pre-determined yet constant speeds, we can estimate the radar's measurement errors. We ran all the experiments on a flat tennis court to minimize deviation in the robot's path.

We use the robot to investigate the parameters that potentially affect the Doppler measurement errors. Figure 5(a) summarizes the parameters' definitions. We then model measurement error as a function $\epsilon()$ of these parameters:

$$\hat{v}_{di} = v_{di} + \epsilon(d, \gamma, v, \theta),$$

where v_{di} is target's true Doppler velocity and \hat{v}_{di} is value the radar measures. When analyzing errors of one parameter, we fix the remaining parameters. For instance, to determine the error at $d = 2m$, we set $v = 30cm/s$, $\gamma = 60^\circ$, and $\theta = 0^\circ$. Then, we collect Doppler velocity measurements when the robot is at distance $d = 2m$. For each point, we collect 100 samples and calculate the average error across all points. We found that errors can be adequately modeled by a Gaussian distribution and therefore we look at the mean (μ) and standard deviation (σ) of Doppler velocity errors.

In the case of parameters d and γ , μ and σ do not change appreciably as d or γ change. This is also the case for v if the robot moves along the Γ line. On the other hand, the σ

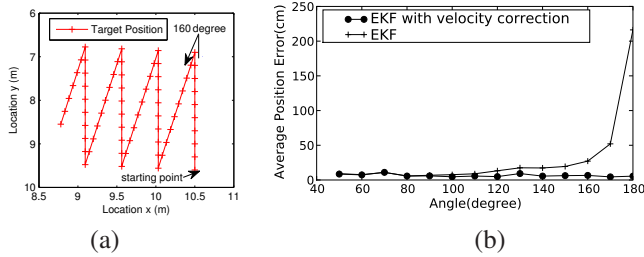


Fig. 6. (a) Multiple turns scenario in which a target turns repeatedly by 160° . (b) The average position errors for EKF_{vc} and EKF. The position errors for EKF grow as the target’s turning angle increases. As the magnitude of the turns increases, it becomes necessary to include the VC to cope with the continuous target turns.

for the different θ values are about eight times larger than the ones for d and γ . The reason is that if a target leaves the Γ line with $\theta > 0^\circ$ (e.g., line #7 in Figure 5(a)), the Doppler velocity of the target will change over time even if the target moves at a constant speed. Thus, the auto-correlation (AC) which averages Doppler velocities over a time interval, inherently generates errors. The error becomes worse as the robot’s speed increases. As a result, among all four parameters, θ is the major contributor to the measurement noise. Hence our simulation models noise as a Gaussian distribution whose mean and variance are the linear function of θ .

We use a simple mechanism to integrate these measurement error models into our simulator. Specifically, given that we only test particular angles and distances, we linearly interpolate the error parameters (μ and σ) between the particular distances and angles we tested.

B. Comparison between Newton-Raphson and EKF

We mentioned in Section III-C that the Newton-Raphson leads to local minima when even small measurement noise is present. We verify this claim with simulations in which a target moves along the same line at 30 cm/s for 60 seconds (Doppler velocities are measured every second). In each experiment, we linearly increase the σ of a zero-mean Gaussian additive noise from 0.3 cm/s up to 2.7 cm/s. The simulations show that when $\sigma=1.2$ cm/s, the average position error for Newton-Raphson is about 1 m (2 m when $\sigma=2.0$ cm/s). When $\sigma \geq 2.7$ cm/s, Newton-Raphson’s results often diverge. In contrast, EKF stays robust to measurement noise (position error ≤ 5 cm). This result suggests that EKF filters noise well when estimating target positions.

In the following two paragraphs, we investigate the impact of target turns on position estimation and the threshold beyond which we apply our VC mechanism. We do so through simulated target trajectories such as the one shown in Figure 6(a). This figure illustrates one example of a target moving at a constant speed of 30 cm/s, taking multiple turns in a $4\text{ m} \times 4\text{ m}$ space. We assume that turns are instantaneous. The red crosses represent the actual positions of the target measured every second. Four radars, not shown in the figure, measure Doppler velocities every second. Thus, we are able to compare

the actual target positions and the estimated positions.

Multiple Turns. Even though EKF outperforms the Newton Raphson method, there are scenarios in which EKF alone is insufficient to accurately estimate the target’s position. For tests such as the one in Figure 6(a), the target turns every 10 s with the same turning angle. We tested turning angles ranging from 0° to 180° at 10° increments. In each case, we compare EKF and EKF_{vc} . The EKF_{vc} applies velocity correction when the target’s estimated orientation changes by more than 20° .

Figure 6(b) shows the average position errors at each turning angle. The results stay consistent when the turning angle is less than 140° . Beyond this angle, however, the errors of EKF start growing. This increase in positioning error occurs as the amount of update by EKF is less than that of the errors occurring by the large angle turns. For example, in the 170° case, the EKF estimates the turn as $\sim 140^\circ$. Considering that EKF uses the erroneous velocity for position estimation, the 30° difference, in turn, results in large position errors. The error becomes worse as the target turns more often or makes larger turns.

On the other hand, the EKF_{vc} performs consistently up to 180° turns. This is because the VC mechanism revises the inaccurate velocity predictions obtained by the simple transition model. By incorporating these updated velocities to the EKF measurement step, EKF_{vc} improves position estimations.

Threshold. The multiple turns section shows the cases when the VC improves the estimation accuracy of our system. There are however situations in which velocity compensation can decrease accuracy. We demonstrate this deterioration through an experiment in which a target moves straight for 15 steps. We add noise ($\sigma = 1.13$ cm/s) to the Doppler velocity measurements and estimates the target’s orientations. The EKF_{vc} method estimated in this case an average of 2.36° orientation change even though the target did not turn. Thus, applying the VC at all times may decrease the system’s performance.

Simply applying VC when orientation changes by more than 2.36° is not a robust solution. This is because the 2.36° constant can vary depending on the noise level. If, however, EKF (without VC) can recover from the position errors caused by the target’s turn up to a certain threshold, one can apply the VC only when the turning angle is larger than this threshold. In order to determine the value of this threshold, we ran simulations in which the target makes a single turn. The target moves a total of 20 steps and turns after the 10^{th} step. We repeat this test using turning angles from 0° to 180° . The experiments show that for turns less than 20° EKF returns to the error level before the turn after only one to two seconds. Thus, by choosing 20° as the threshold above which we apply the VC, the false positives generated by measurement noise can be avoided.

Error analysis on the velocity estimation in (9) suggests that the geometry and the angle between a radar and the target also affect the estimation error of a target velocity. For simplicity, we assume the scenario in which we do not need to consider the geometry—when radar placements are equi-

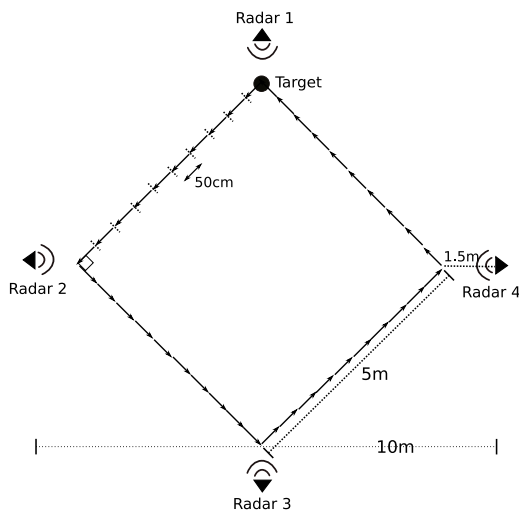


Fig. 7. Four radars facing the center of a diamond-shaped target trajectory are placed at each corner. A human moves at a constant speed of 50 cm/s, stepping on each 50 cm marker. The human, starting from the top of the diamond, walks counter-clockwise for 80 seconds.

angular. Appendix A explains the derivation and details of equi-angular placement.

C. Outdoor Experiments

Next, we show the performance of our system in a real environment. Specifically, we compare the position and velocity errors of Newton-Raphson, EKF, EKF_{vc} when the target follows a closed-loop trajectory. As Figure 7 shows, we installed four radars around a $10\text{m} \times 10\text{m}$ area in a parking lot. A human target starts to move from the top of the diamond-shaped trajectory at an approximate speed of 50 cm/s. We choose a human target because the robot cannot turn around the corners without stopping to change direction as in simulations. In total, the target moves, without any interruption, for 80 seconds.

The results in Table I suggest that EKF_{vc} outperforms Newton-Raphson and EKF. The poor results of Newton-Raphson come from the fact that noise-affected Doppler measurements lead it to local minima. Compared to EKF, EKF_{vc} improves the accuracy of the target velocity by $\sim 50\%$, which in turn improves the position accuracy by 34%.

At the same time, the result for EKF_{vc} significantly differs from the simulation result using the same trajectory. The simulation errors for position, speed, and orientation were 4.5 cm, 0.62 cm/s, and 0.8° respectively. Two main factors contribute to this performance degradation. First, the target moves at a variable speed when turning around the corners, which violates our constant velocity model. Second, as the different parts (e.g., arms and torso) of a human body have different speeds, the Doppler velocities shows larger variance than those of the robot. This observation suggests that a more sophisticated dynamic model describing complex human movements is necessary.

Method	Position Error (cm)	Speed (cm/s)	Orientation (degree)
Newton-Raphson	210.60 (89%)	9.57 (58%)	6.04° (47%)
EKF	33.89 (34%)	7.61 (48%)	7.35° (56%)
EKF_{vc}	22.49	4.031	3.23°

TABLE I
THE AVERAGE POSITION AND VELOCITY ERRORS FOR THE OUTDOOR EXPERIMENT SHOWN IN FIGURE 7. THE NUMBERS IN PARENTHESES DENOTE THE IMPROVEMENTS THAT EKF_{vc} OFFERS OVER EKF AND NEWTON-RAPHSON.

D. Comparison with the Bayesian Filtering Method

We compare the EKF estimation with that of a Bayesian filter. Similar to the EKF time and measurement steps, the Bayesian filter updates its prior distribution based on the target's dynamic model and the previously derived posterior distribution. As more measurements become available, the filter updates the posterior distribution. We implemented such a Bayesian filter using the method proposed by Taylor et al. [18]. It suggests using the Laplace approximation to simplify the calculation of the posterior distribution. Instead, we apply the same dynamic model (i.e., constant velocity) as with the EKF to the prior distribution (whereas [18] assumes the target velocity is unknown and thus uses a large covariance matrix).

The average errors of the Bayesian filtering method for position, speed, and orientation were 33.64 cm, 7.78 cm/s, and 8.01° respectively. These results are similar to those achieved by the EKF and are in contrast to the results presented in [18], where the Bayesian filter performed better. Our conjecture about the difference at this point is that their Bayesian was able to apply three Newton-Raphson iterations for each set of measurements, which contributed improving the estimation results. On the other hand, when we used the Bayesian, we was not able to apply such number of iterations because the significant measurement errors caused local minima (we only applied once).

Our results also show that the VC considerably improves the original Bayesian filter's performance. Specifically, the estimation errors for the Bayesian filter with VC for position, speed, and orientation are 25.75cm, 5.1cm/s, and 4.12° respectively. This improvement suggest that the VC mechanism can be applied to other filtering schemes.

VI. DISCUSSION

The experiments presented thus far were conducted outdoors. We set up an indoor experiment in which a human target moved along an 8 m straight line at a speed of 50 cm/s. The locations of the four radars tracking the target formed a parallelogram enclosing the target's path. The mean speed error was approximately 19 cm/s, which in turn generated ~ 3 m average position estimation error. We believe that the cause behind these large errors is the severe multipath propagation inherent in indoor environments. As the auto-correlation

method averages all signals received over a fixed interval, reflected signals containing multiple Doppler frequency components create inaccurate Doppler velocity estimates.

Furthermore, while not experimentally tested, we believe that the current system is not well suited for tracking multiple targets. The reason is that the auto-correlation method, which estimates Doppler velocities in the time domain, cannot distinguish the Doppler velocities of different targets. Using a frequency domain technique (e.g., FFT) could be a solution, but it is computationally prohibitive for the TelosB platform that we use. Emerging sensor mote platforms, such as the Egs platform [10] based on the Cortex M3 [2], could make this technique feasible.

VII. SUMMARY AND FUTURE WORK

We propose a WSN tracking system for a non-cooperative target using pulsed Doppler radars. Compared to existing approaches, our solution requires no cooperation from a target; instead it uses the signals reflected from the target which bear a Doppler frequency. After collecting a set of Doppler velocities from radar motes, we use an Extended Kalman Filter (EKF) to estimate the target's position and velocity.

We show that combining EKF with a target velocity correction technique (EKF_{vc}) outperforms Newton-Raphson and EKF both in simulations and outdoor experiments. The velocity correction mechanism, independent of any particular filtering scheme, improves estimation performance by more than 30%. The outdoor experiments also prove the feasibility of our system in real-life scenarios. In addition, while collecting Doppler velocity measurements for building a realistic simulator, we derive a simple error model and use it to analyze the error bounds of estimations.

The following describes the direction of our future work. First, as the outdoor experiments suggest, we need a sophisticated model that can capture complex target movements (e.g., variable speeds with turns). Alternatively, an adaptive Kalman filter, which does not require exact knowledge of the process noise covariance matrix Q and the measurement noise covariance matrix R , may improve the current simple model of target dynamics. In this way, our system would become more resilient to complex target behaviors without changing the dynamic model. We are also interested in developing a technique to optimize the velocity correction threshold that balances sensitivity to target turns and robustness to fluctuations in measurement noise. While we currently keep the radars always on to maximize the detection probability, we are interested in developing a scheduling algorithm that attempts to balance energy consumption and detectability.

APPENDIX

The unique estimation problem arising in the proposed system cannot be easily analyzed to yield useful insights as it involves four unknown parameters. Instead, we derive the Cramér-Rao Bounds (CRBs) for target velocity estimation, not only to obtain analytical results directly relevant to our

algorithmic approach, but also to justify the feasibility of using velocity correction mentioned in Section III-D.

Target Velocity Estimation. Considering a scenario, as in Figure 3, in which the positions of the target and radar i are known, we find the error bound for the target's velocity based on (8). If we assume that ϵ_i 's in (8) are i.i.d zero-mean Gaussian and standard deviation σ , then we can derive the Fisher information matrix (FIM) for the estimation problem:

$$J = \frac{1}{2\sigma^2} \begin{bmatrix} \sum_{i=1}^N (1 + \cos 2\gamma_i) & \sum_{i=1}^N \sin 2\gamma_i \\ \sum_{i=1}^N \sin 2\gamma_i & \sum_{i=1}^N (1 - \cos 2\gamma_i) \end{bmatrix} \quad (9)$$

where the CRB is the inverse of J (the Gaussian assumption is also verified in Section V-A). By maximizing the determinant of FIM, we obtained the necessary and sufficient conditions for the theoretically optimal radar placement

$$\sum_{i=1}^N \sin 2\gamma_i = 0, \quad \sum_{i=1}^N \cos 2\gamma_i = 0.$$

This implies an equi-angular placement with $N \geq 3$.

REFERENCES

- [1] Andreas M. Ali, Kung Yao, Travis C. Collier, Charles E. Taylor, Daniel T. Blumstein, and Lewis Girod. An empirical study of collaborative acoustic source localization. In *IPSN*, 2007.
- [2] Atmel Corporation. AT91 ARM Cortex-M3 based Microcontrollers: SAM3U Specifications, 2009.
- [3] Ho-lin Chang, Jr-ben Tian, Tsung-Te Lai, Hao-Hua Chu, and Polly Huang. Spinning beacons for precise indoor localization. In *Sensys*, 2008.
- [4] Prabal K. Dutta, Anish K. Arora, and Steven B. Bibyk. Towards radar-enabled sensor networks. In *IPSN*, 2006.
- [5] Stanislav Funiak, Mark Paskin, Rahul Sukthankar, and Carlos Guestrin. Distributed localization of networked cameras. In *IPSN*, 2006.
- [6] L. Gu, D. Jia, P. Vicaire, T. Yan, L. Luo, A. Tirumala, Q. Cao, T. He, J. Stankovic, T. Abdelzaher, and B. Krogh. Lightweight Detection and Classification for Wireless Sensor Networks in Realistic Environments. In *Sensys*, 2005.
- [7] IRobot. IRobot Create. Available at: <http://www.irobot.com>.
- [8] Thomas Kailath and Ali H Sayed. *Linear Estimation*. Prentice Hall, 2000.
- [9] Lasse Klingbeil and Tim Wark. A wireless sensor network for real-time indoor localisation and motion monitoring. In *IPSN*, 2008.
- [10] Jeonggil Ko, Qiang Wang, Thomas Schmid, Wanja Hofer, Prabal Dutta, and Andreas Terzis. Egs: A cortex m3-based mote platform, 2010.
- [11] Branislav Kusy, Akos Ledeczi, and Xenofon Koutsoukos. Tracking mobile nodes using RF doppler shifts. In *Sensys*, 2007.
- [12] Hung D. Ly and Qilian Liang. Collaborative multi-target detection in radar sensor networks. In *MILCOM*, 2007.
- [13] R. Nitzberg. *Radar signal processing and adaptive systems*. Artech house, 1999.
- [14] Enrico Paolini, Andrea Giorgetti, Marco Chiani, Riccardo Minutolo, and Mauro Montanari. Localization capability of cooperative anti-intruder radar systems. In *EURASIP*, 2008.
- [15] Peyton Z. Peebles. *Radar Principles*. Wiley-Interscience Publication, 1998.
- [16] Samraksh Company. BumbleBee Radar. Available from: <http://www.samraksh.com>, 2007.
- [17] Adam Smith, Hari Balakrishnan, Michel Goraczko, and Nissanka Priyantha. Tracking moving devices with the cricket location system. In *MobiSys*, 2004.
- [18] Christopher Taylor, Ali Rahimi, Jonathan Bachrach, Howard Shrobe, and Anthony Grue. Simultaneous localization, calibration, and tracking in an ad hoc sensor network. In *IPSN*, 2006.
- [19] T. Teixeira and A. Savvides. Lightweight people counting and localizing in indoor spaces using camera sensor nodes. In *ICDSC*, 2007.
- [20] Roy Want, Andy Hopper, Veronica Falco, and Jonathan Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 1992.
- [21] Wikipedia. Doppler Effect. Available at http://en.wikipedia.org/wiki/Doppler_effect.