

A Tracking System for Wireless Embedded Nodes using Time-of-Flight Ranging

Evangelos B. Mazomenos, Jeff S. Reeve, *Senior Member, IEEE*, Neil M. White, *Senior Member, IEEE* and Andrew D. Brown *Senior Member, IEEE*,

Abstract—In this work we present the design, development and evaluation of a real-time target tracking system for wireless embedded nodes, capable of effectively tracking manoeuvring targets. The proposed tracking system is designed to operate solely on range measurements obtained with the use of a two-way Time-of-Flight method without the need for additional hardware being incorporated in the nodes. To address the challenge of coping with manoeuvring targets, the tracking problem is formulated as a dynamical estimation problem where an adaptive multiple-model approach is employed to represent the motion pattern of manoeuvring targets. The ranging observations are produced in real-time and used as inputs to a Particle Filter algorithm which produces the estimates of the target's kinematic variables. Simulations are provided to assess the effect of several factors on the system's performance. Ultimately, the entire system is implemented on commercially available hardware and tested in an outdoor deployment. A total of 25 experiments demonstrate an average rms accuracy of 2.6m for position and 1.9m/s for velocity, in a 15m x 15m area. Such performance, which is additionally confirmed from simulation results reveals the potential of the proposed range-only system in application scenarios where real-time tracking of mobile targets is needed.

Index Terms—Wireless sensor networks, Real-time and embedded Systems, Target Tracking

1 INTRODUCTION

AFTER a decade of continuous evolution and development, Wireless Sensor Networks (WSNs), have earned a prominent spot among pervasive computing technologies. The flexibility they offer, by encompassing various sensor modalities, low-power wireless communication and processing ability, in a limited-sized hardware platform, initiated research in various interdisciplinary directions for exploiting this novel technology in a number of application domains [1]. Examples include environmental monitoring, smart structures, habitat monitoring, military defense applications, surveillance and security, mobile robotics, health care and medical applications, agriculture and asset management [2], [3].

Locationing and tracking objects of interest is considered to be a pivotal functionality for a number of application domains. WSNs are considered to be a technology able to provide innovative solutions for locationing and tracking applications. They offer the possibility of employing a large number of observers, tasked with monitoring the same phenomena, an approach that enables decentralised sensing, distributed computing and collaborative signal processing [4]. An abundant amount of information is accumulated from the network with high spatial and temporal resolution. For locationing and tracking, this is of particular

interest, facilitating the development of more robust, flexible and cost-effective tracking systems [5], [6]. The basic concept for target tracking with WSNs is to deploy a number of cooperative embedded nodes to monitor a specific region of interest. Whenever a target is present, the nodes interact with the target and collect useful information for the tracking operation. Generally in tracking systems, the target's dynamics are inferred by processing specific information, associated with the target's kinematic variables (position, velocity, direction of movement). For example, from various sensor readings (e.g. acoustic energy) the relative distance between the source (target) and the sensor (anchor) can be derived. The collected data, are then imported into the "tracking algorithm", which produces an estimate of the target's kinematic variables.

Under this context, the work presented in this paper attempts to exploit the capabilities of distributed WSNs in developing a real-time, range-only target tracking system. The range-only characterization of the system pertains to the type of data that the system utilizes in order to infer the target's kinematics. The proposed system operates exclusively on range observations acquired with the use of a two-way Time-of-Flight (ToF) ranging method. Our choice of employing ToF ranging, differentiates the proposed system from a number of approaches that employ additional types of observations (bearings, velocity) which require costly and energy demanding additional hardware (micro RADARS, directional antennas) to be installed on the wireless nodes.

The tracking problem is theoretically formulated

• Evangelos B. Mazomenos, Jeff S. Reeve, Neil M. White and Andrew D. Brown are with the School of Electronics and Computer Science, University of Southampton, SO17 1BJ, Southampton, U.K. E-mail: {ebm07r, jsr, nmw, adb}@ecs.soton.ac.uk

as a dynamical system with the objective being, the real-time estimation of the target's kinematic variables based on range observations. The proposed system is intended to effectively track manoeuvring targets, which is the case in the majority of real-world tracking scenarios. For this, a multiple-model approach is used to represent the dynamics of manoeuvring targets. Such an approach diversifies the proposed system from other approaches in the area that only consider a constant velocity (CV) model for describing the target's dynamics, thus providing limited support for manoeuvring targets [7], [8], [9]. The resulting system is non-linear, due to the non-linearity between the range observations and the kinematic variables. To solve this system, an adaptive multiple-model Particle Filter (PF) tracking algorithm is proposed. By applying multiple models to represent the evolution of the target's dynamics in time, we demonstrate that our design tracks manoeuvring targets efficiently. Simulation results of applying the proposed framework on manoeuvring targets can be found in our previous work [10]. As an expansion of this work, the multiple-model approach is verified through a simulation comparison against the CV model for a manoeuvring scenario. Moreover, we study and quantify the effect of two important parameters on the performance of the tracking system, namely the number of particles and the sampling interval. We also calculate the Cramer-Rao lower bound (CRLB) and utilise it as a benchmark to assess the performance of our system. This analysis is based on the accuracy obtained by the ToF ranging method which is also analysed here and incorporated in the tracking system.

As a final extension to our previous work, in this paper we present a working prototype of our system, implemented on Commercially-off-the-Self (COTS) hardware. The T.I. EZ430-RF2500 platform was used for this. In our prototype, three types of node are designated; the anchor nodes, the target node and the central node. For each different class of nodes a separate piece of software was developed. In addition, the tracking algorithms were implemented as MATLAB routines, which collect the ranging data and produce an estimate of the target's trajectory in real-time. The system was deployed in 15m x 15m outdoors area and multiple experiments were executed with the target moving in a variety of trajectories. The results obtained are compared to simulations which verify the achieved performance. These prominent outcomes justify the choice of a range-only tracking system for embedded nodes and also reveal that the proposed system satisfies the three main objectives of accuracy ($\sim 1\%$ of the area), real-time operation and tracking of manoeuvring targets.

The rest of the paper is structured as follows. Section 2 summarizes related work in the literature. The two-way ToF ranging method is discussed in Section 3. In Section 4 we provide the mathematical

formulation of the tracking problem as a nonlinear estimation problem and introduce the models that account for the motion dynamics as well as the observations. In addition this section reviews the PF tracking algorithms. Simulation results are presented in Section 5 alongside the calculation of the CRLB which is used as a benchmark to compare the system's performance. Section 6 presents the implementation on COTS hardware of the tracking system. Results obtained from the outdoors experimentation of the full system are provided in Section 7 alongside a comparison to simulation results. In the final section, concluding remarks and future directions are discussed.

2 RELATED WORK

Many ideas regarding tracking and locationing with WSNs are present in the relevant literature. We will restrain ourselves to the tracking systems that were implemented and demonstrated at full scale. Coates *et al.* consider a clustered WSN for tracking, comprising of class-B sensor nodes that measure, either the range or the bearing of the target and class-A cluster heads that aggregate the data gathered from the class-B nodes. Each of the class-A cluster heads runs its own local PF, based on the data acquired from the class-B nodes in its neighborhood (cluster). The weights for each particle are then calculated based on information from all cluster heads. Each cluster head represents a particle with a certain weight associated to it and a global estimation can then be extracted. The drawback of such an approach is that there is a need for a large number of cluster heads-particles (> 200) and subsequently even larger for class-A nodes, to achieve accurate performance, resulting in a network that involves an excessive number of nodes [11], [12].

The *CRICKET* indoor locationing system [13] developed at MIT, consists of beacons that are attached to the ceiling of a building, and receivers, called listeners, that require locationing. The beacons periodically transmit their location information in an RF message. At the same time, the beacons also transmit an ultrasonic pulse. The listeners listen to beacon transmissions and compute their own locations by calculating the TDoA of the two signals emitted from nearby beacons. The user's location is determined in relation to the already known location of the mounted nodes [14]. The *CRICKET* locationing method is used in a centralised localization and tracking algorithm named *LaSLAT*. *LaSLAT* algorithm reported a few centimeters of error in an 7m x 7m indoor area and approximately 0.5m error in an 27m x 32m dense outdoor deployment [15].

RADAR is another indoor locationing system which is based on low-power WSNs. A number of infrastructure nodes, positioned in known locations is used to generate RSSI values for different positions in the

coverage area and build a signal strength database. Whenever a blind node requires positioning, its RSSI value is measured by the closest infrastructure nodes. The observations are fused to a central server, which examines the signal strength map to obtain the best fit for the current transmitter position. The achieved accuracy is between 2 to 3 meters [16]. Ahmed *et al.* address the combined problem of target detection and tracking. Target presence or absence is modeled by a probability function. The tracking algorithm that is used estimates, apart from the target's state vector, an extra binary variable which indicates the presence of the target [17]. In the prototype system, a dense network of MicaZ nodes provided range readings inferred by measuring the acoustic intensity and a PF algorithm, which employs a large number of particles (5000) produces both the target's dynamics as well as decides on the presence of the target. The reported results are in the area of 0.1 - 0.25m in an indoors 1 x 3m area with the use of at least 8 anchor nodes [18].

Radio interferometry was presented as a ranging method for embedded nodes in [19]. Kusy *et al.* employ this method in the *inTrack* tracking system [20]. In *inTrack*, the target node is programmed to transmit an unmodulated high frequency sine wave. One stationary infrastructure node, positioned in known coordinates, transmits a similar signal simultaneously with the target. The resulting composite signal demonstrates a phase offset, which when measured at other infrastructure nodes depends only on the in-node distance between the participating nodes. The target's position is inferred, by employing multiple anchors to measure the phase offset. An extension of *inTrack*, is presented, where the Doppler shift of the transmitted sine-wave is also measured at the infrastructure nodes and is used to estimate the target's velocity. A tracking algorithm combining the extended Kalman Filter and a constrained non-linear least squares optimisation method is used to infer the target's position and velocity. The reported accuracy of the system increases with increasing number of participating anchor nodes [21]. A deployment of 8 infrastructure nodes in a 50m x 30m area reported results of 1.3 - 2.2m for position and 0.13 - 0.35 m/s for velocity [22], [23].

Previous research considered a linear model in order to represent the target's motion dynamics. However this approach can not effectively cope with alterations in the position and velocity vectors of a manoeuvring target. To achieve tracking of manoeuvring targets, adaptive estimation algorithms and a multiple-model approach to describe the development of the target's dynamics in time are investigated in this work. The ToF method to be used in the proposed tracking system yields several advantages over acoustic ranging methods. The systems that are based on acoustic ranging not only require a dense deployment of anchor nodes even in small areas, but also have the need for additional hardware like ultrasound

transceivers to be attached on the WSNs nodes. If audible acoustic signals are used, the target must itself produce these acoustic signals. Different to the interferometric ranging method which required the target node and another node to transmit sine waves simultaneously, the proposed ToF ranging scheme only has a calibration requirement.

3 TWO-WAY TIME-OF-FLIGHT RANGING

In this section we highlight the major aspects of the ToF ranging technique that we employ in the proposed range-only tracking system. For a detailed analysis the reader is directed to our previous work which was focused entirely on this method [24]. In principle, ToF methods attempt to estimate the transit time of a signal. The a-priori knowledge of the signal's velocity allows the approximation of the distance between transmitter and receiver. The developed method intends to quantify the distance between a pair of unsynchronised wireless nodes and is considered to be ideal for the range-only tracking system that we consider. The fundamental idea of our ToF ranging method is to achieve an estimation of the distance between the two nodes by conducting multiple two-way message exchanges and calculating the mean ToF value.

The objective is to estimate the distance between node A and node B (Fig. 1). A local timer on node A is employed to provide the ToF timing values. Initially node A sends the first ranging signal and captures the time of its timer (t_{tAB}). Node B receives the signal and after a period of time, that corresponds to node B swapping its state, from receiver to transmitter (as well as a number of other delays) node B sends a ranging signal back to node A. Following, node A receives the reply signal and stores the time of its reception (t_{rBA}). The timer in node A measures $t_A = t_{rBA} - t_{tAB}$ multiple times.

Within t_A , the delay related to the ranging message being processed at node B is included. In order to measure the amount of time that corresponds to delays that occur during the two-way message exchange process and remove it, we introduce a calibration stage. This is accomplished by placing the transceivers at a very close distance ($< 0.2m$), so the ToF period is minimal and executing multiple transactions that are averaged to produce the minimum time (t_{min}) that is required in order to complete a message exchange. This amount of time corresponds to a minimal ToF period and reveals all the hardware and software delays that occur during a two-way ranging transaction. We make the assumption that these delays remain constant and are independent of the distance between the nodes. Subsequently only the propagation delay will increase the two-way time transfer value as the nodes are placed at greater distance.

Fig. 2 illustrates a timing diagram of a message exchange between the two nodes. Send and receive

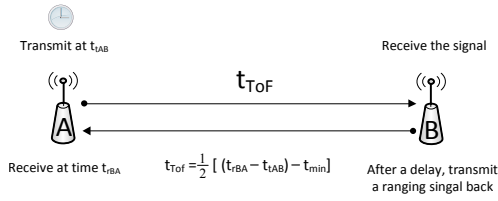


Fig. 1. Proposed Two-way ToF Ranging

occurs on the rising edge of the nodes clocks. Assuming that for a given distance, the t_{ToF} will be the same and the delay T_{B_proc} that node B requires to process the ranging signal and submit the reply is constant, the only ambiguity will be inserted by the delays associated to the clocks phase shift and frequency drift. Given that the two clocks are unsynchronized and have a small difference in frequency the phase offset between the devices will oscillate, thus the delays T_{d1} and T_{d2} will follow a similar varying pattern. By oversampling, we capture a normally distributed set of multiple timing transactions centered around the mean ToF value. Therefore, capturing a sufficiently large number of timing values will allow us to extract the mean ToF value which can be, linearly associated to the distance between the nodes.

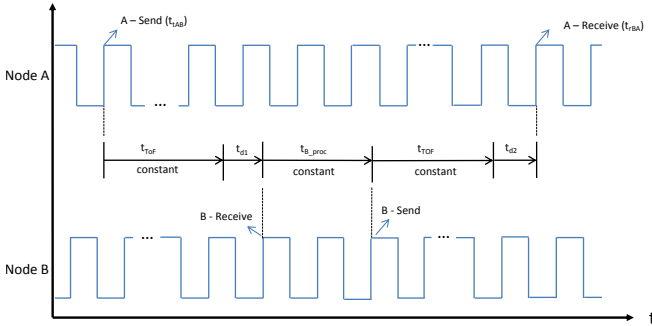


Fig. 2. Timing Diagram of a two-way message exchange

Following the completion of the required number of two-way transactions, node A enters the calculation phase. The calculation phase involves the extraction of the ToF out of the multiple stored timer values. In the event that one, or in general a small fraction of these n transactions has produced erroneous timing, including them in the average calculation will result in distorting the correct mean value. To avoid this, the following procedure is followed. Let us assume that we obtain n two-way ToF values t_n . Initially the mean \hat{t}_{ToF} and standard deviation of the n values are calculated. In the following step, we calculate the absolute difference of each one of the n values from \hat{t}_{ToF} . Ultimately, out of the n collected ToF values we exclude the ones that fall outside the one deviant limit. The final \hat{t}_{ToF} value is calculated by averaging the remaining m values. The mean \hat{t}_{ToF} two-way ToF value is then converted to distance by executing the

following.

- 1) Calibrate the \hat{t}_{ToF} value by subtracting it from the minimum two-way ToF (t_{min}).
- 2) Divide the calibrated value by two, to get a single-way ToF time. $t_{ToF_final} = \frac{1}{2} (\hat{t}_{ToF} - t_{min})$.
- 3) Multiply the above by the speed of light in air ($c/1.0003$) to convert time to distance

3.1 Implementation and Evaluation

The ranging system was implemented on the T.I EZ430-RF2500 platform and two types of nodes were designated and programmed with different pieces of software. A *requester* node, which is in essence node A, that initiates the entire tracking operation, logs the two-way timing values and executes the calculation phase and a *responder* that acts as the relay node B. The clock on node A was set to the maximum possible frequency of 16 MHz. We carried out experiments in various environments outdoors and indoors with 1000 ranging transactions at data rate settings 250kbps and 500kbps. The achieved accuracy was in the area of 1-3m RMS.

To verify the distribution of the measurements that the proposed ToF ranging system yields, an experiment is designed where two nodes are placed in short distance ($\sim 2m$) indoors and a vast number of ToF estimates is logged over a period of time. Approximately 10000 ToF estimations were logged. Fig. 3 depicts the histogram of the obtained values and it is clear that they can be considered as normally distributed.

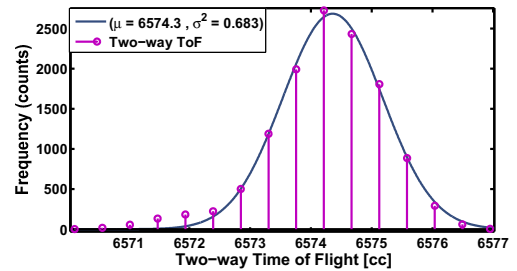


Fig. 3. Timing Histogram of 10000 two-way values at 500 kbps

In our implementation the node performed the calculation phase whenever 100 two-way transactions were completed. Part of the process is the calculation of the standard deviation for these 100 transactions in order to exclude the timing values that fall outside the single deviant boundary. This procedure is repeated 10 times to reach the required 1000 transactions. From all the experiments carried out the standard deviation of the timing values was initially in the range of $1.4cc - 1.8cc$ (cc:clock cycles) for the 500kbps setting and $2.4cc - 3cc$ at 250kbps. After averaging the values (excluding the ones outside the one deviant limit) the deviation was reduced to $0.3cc - 0.9cc$ at both

500kbps and 250kbps. The deviation in the outdoors experiments was found to be smaller than the one indoors. Particularly for the 500kbps datarate setting, which is used in the tracking system, the standard deviation that the two-way ranging values exhibit was approximately $0.4cc$ on average. Dividing this by two we get $\sigma_{ToF} = 0.2cc$. This value is expressed in clock cycles and a single clock cycle of the 16MHz timer is $(1/16\text{MHz}) = 62.5\text{ns}$. Thus the standard deviation of the proposed system can be approximated as $\sigma_{ToF} = 12.5\text{ns}$. This translates to a standard deviation of approximately $3.7m$ calculated from $\sigma_{ranging} = c * \sigma_{tof}$.

4 TRACKING SYSTEM OVERVIEW

In this section, we provide the theoretical foundation of the proposed range-only tracking system. We consider a scenario where a dedicated WSN is deployed in an area, in order to track a target of interest. A number of anchor nodes is considered to be deployed in known positions. Fig. 4 illustrates the envisioned setup. The proposed tracking system is formulated as a dynamic state estimation problem in the discrete-time state-space domain. Here we focus on the mathematical formulation of the tracking problem assuming that the range observations becomedd available from the ToF scheme analysed in the previous section. In this section we also analyse the developed tracking algorithms which are used to solve the dynamic problem and infer the target's kinematic variables.

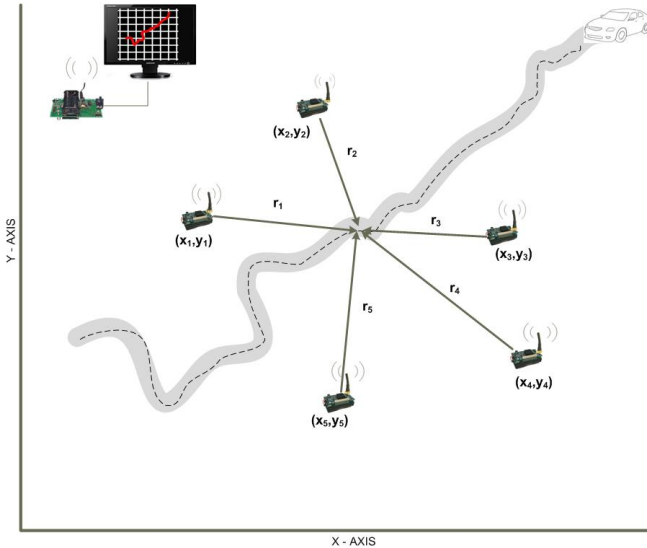


Fig. 4. Tracking System Overview

To begin with, the state vector is populated from the target's kinematic variables that the system attempts to estimate. Since we consider ground targets, we choose to estimate the target's planar coordinates and two-axis velocity. Thus the state vector is given as,

$$\mathbf{x} = [x \ y \ v_x \ v_y]^T \quad (1)$$

Next, we consider that the target's motion pattern can be described with the following dynamic model.

$$\mathbf{x}(t) = \mathbf{F}\mathbf{x}(t-1) + \mathbf{\Gamma}\mathbf{w}(t-1) \quad (2)$$

- where,

$$\mathbf{\Gamma} = \begin{bmatrix} T_s^2/2 & 0 \\ 0 & T_s^2/2 \\ T_s & 0 \\ 0 & T_s \end{bmatrix} \quad (3)$$

- T_s : is the sampling period,
- $\mathbf{w}(t-1)$: is a 2×1 i.i.d process noise vector with dimension of acceleration m/s^2 , sampled from a known distribution which represents any mis-modeling effects or disturbances in the motion model
- and $\mathbf{x}(t)$: is the state vector, defined in Eq.1

For the purposes of the proposed tracking system we follow two different approaches in populating the motion matrix \mathbf{F} in Eq.2.

In the first approach, \mathbf{F} is formed according to the constant velocity (CV) model under which, the target is assumed to be constantly moving with velocity around a certain value. In this case matrix \mathbf{F} is given as:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

One of the main objectives of the proposed tracking system is to provide enhanced support in tracking manoeuvring targets. Therefore, to provide the required support for manoeuvring targets, we adopt an approach where the state-update equation is modeled with the use of multiple switching dynamic models. The multiple-model approach intends to provide additional support for when a manoeuvring target is the object of interest and effectively capture the sudden changes in the velocity vector that manoeuvring targets exhibit. In this case our system is modeled using three switching dynamic models. The models we consider are the CV model described previously and two coordinated turn models.

An integer parameter, termed as regime variable, is introduced in the multiple-model case. The regime variable $r(t)$ dictates which of the three state models (regimes) is in use during the time interval $(t-1, t]$. The regime variable $\mathbf{r}(t)$ is modeled as a time homogeneous, three-state, first-order Markov chain with transitional probability matrix given by the following relationship:

$$\pi_{mn} \triangleq \text{Prob}\{\mathbf{r}(t) = m \mid \mathbf{r}(t-1) = n\} \quad (5)$$

The probability matrix indicates the probability of a regime transition occurring, between consecutive sampling steps, as well as the probability of the system remaining on the same regime.

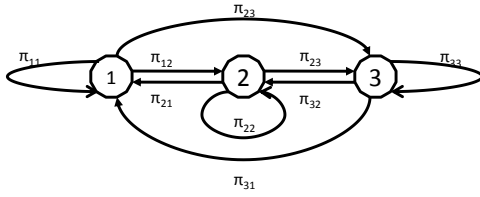


Fig. 5. Regime probability transitions

In the multiple-model case, the state update equation is formulated as follows:

$$\mathbf{x}(t) = \mathbf{F}(r(t))\mathbf{x}(t-1) + \mathbf{\Gamma}\mathbf{w}(t-1) \quad (6)$$

The state transition matrix \mathbf{F} , at time t is defined according to the value of the regime variable $r(t)$, ($\mathbf{F}(r(t))$) and can be one of the following:

- 1) The Constant Velocity motion model provided in Eq.4
- 2) The First Coordinated Turn model, defined as:

$$\mathbf{F}(2) = \begin{bmatrix} 1 & 0 & \sin(\omega T_s)/\omega & (\cos(\omega T_s) - 1)/\omega \\ 0 & 1 & (1 - \cos(\omega T_s))/\omega & \sin(\omega T_s)/\omega \\ 0 & 0 & \cos(\omega T_s) & -\sin(\omega T_s) \\ 0 & 0 & \sin(\omega T_s) & \cos(\omega T_s) \end{bmatrix} \quad (7)$$

- 3) The Second Coordinated Turn model given as:

$$\mathbf{F}(3) = \begin{bmatrix} 1 & 0 & \sin(\omega T_s)/\omega & (\cos(\omega T_s) - 1)/\omega \\ 0 & 1 & (1 - \cos(\omega T_s))/\omega & \sin(\omega T_s)/\omega \\ 0 & 0 & \cos(\omega T_s) & -\sin(\omega T_s) \\ 0 & 0 & -\sin(\omega T_s) & \cos(\omega T_s) \end{bmatrix} \quad (8)$$

T_s : denotes the sampling interval and,

w : is the turning rate, expressed in rad/s and considered to be constant.

The two coordinated turn models are used to model turning manoeuvres in the anticlockwise and the clockwise direction respectively. These type of motion modeling has been used previously in scenarios involving bearings only tracking as well as in aircraft navigation [25], [26].

4.1 Observations Model

The measurements vector obtained at each time step, contains an estimation of the distance between the target's position and the position of each one of the N_s anchor nodes. These range estimations are produced with the use of the two-way ToF ranging method. Subsequently at each time step the complete measurements vector is given as,

$$\mathbf{z}_i(t) = [z_1, z_2, z_3 \dots z_{N_s}], i = 1, 2 \dots, N_s \quad (9)$$

The completion of the dynamical system requires the definition of the measurements equation, which mathematically relates the observations $\mathbf{z}(t)$ and the

state-vector $\mathbf{x}(t)$. Since we consider ranging measurements the equation is formed with the use of the Euclidean norm:

$$\mathbf{z}_i(t) = \sqrt{(y(t) - y_i)^2 + (x(t) - x_i)^2} + \mathbf{v}(t), i = 1 \dots N_s \quad (10)$$

$x(t), y(t)$: are the target's x-y coordinates at time t and x_{N_s}, y_{N_s} are the coordinates of the anchor nodes and, $\mathbf{v}(t)$: is a $N_s \times 1$ noise vector sampled from a known distribution that represents the observations noise. From our ranging experiments we see that the observations error in position follows a Gaussian distribution with zero mean and 3.7m standard deviation.

4.2 Tracking Algorithms

Based on the state model (single, multiple) that is used, we develop an algorithmic framework for the recursive solution of the formulated tracking system. Considering the non-linear nature of the system, we choose to employ PF as the basis of our tracking algorithms. PF are a class of recursive Bayesian Estimation methods inspired by the techniques of Importance Sampling and Monte Carlo Integration [27], [28]. In the Bayesian Estimation framework an unknown state (in this case the state vector $\mathbf{x}(t)$) is estimated in a two-stage procedure given the incoming measurements (observations) and a mathematical process model [29].

To calculate a state estimate at time t given the sequence of measurements $\mathbf{z}(t)$ up to that time, the posterior probability density function (pdf) $p(\mathbf{x}(t) | \mathbf{z}(t))$ of the state at time t should be estimated. After obtaining the posterior pdf $p(\mathbf{x}(t) | \mathbf{z}(t))$, an estimation of the state vector can then be produced with the use of a certain criterion like the Minimum Mean Square Error (MMSE). In PF based algorithms, to estimate the pdf $p(\mathbf{x}(t) | \mathbf{z}(t))$ at time t , N particles are generated from a proposal distribution $q(\mathbf{x}(t) | \mathbf{z}(t))$. Let's denote the generated particles at time t as, $\mathbf{x}(t)^i$ and their corresponding weights as $w(t)^i$. The weights are calculated from the following:

$$w(t)^i \propto \frac{p(\mathbf{x}(t)^i | \mathbf{z}(t))}{q(\mathbf{x}(t)^i | \mathbf{z}(t))} \quad (11)$$

After obtaining the weight for each particle an estimation of the desired pdf at time t can be produced from:

$$p(\mathbf{x}(t) | \mathbf{z}(t)) = \sum_{i=1}^N w(t)^i \delta(\mathbf{x}(t) - \mathbf{x}(t)^i) \quad (12)$$

where δ is Dirac's delta function.

A well-known issue of concern in PF is the degeneracy problem. In practical terms, after a number of iterations all but one particles have negligible weights. Thus, a substantial amount of computation is devoted in updating particles with minimal contribution to the approximation of the pdf. To avoid this, a measure,

called effective sample size N_{eff} , is introduced and defined as follows:

$$N_{eff} = \frac{1}{\sum_{i=1}^N (w(t)^i)^2} \quad (13)$$

A resampling step is carried out whenever N_{eff} is found to be smaller than a pre-defined threshold N_{thr} . Resampling eliminates samples with low importance weights while multiplies samples with high importance weights [27].

4.3 Range-Only Tracking Particle Filter Algorithm (ROT-PF)

The algorithm described in this section is intended for a tracking scenario where the target's motion is represented by the CV model. To begin with, we considered that both the state and measurements noise follow known distributions that can be sampled. The transitional prior $p(\mathbf{x}(t)|\mathbf{x}(t-1))$ is chosen as the importance density function to sample particles from. Initial particles (at time $t = 0$) are drawn from a distribution $p(x_0)$ which represents the system's prior knowledge regarding the target's initial state condition.

To produce a sample from the transitional prior, a noise sample $\mathbf{w}(t-1)^i$ is initially generated and used in Eq.2 to produce a sample $\mathbf{x}(t)^i$ distributed accordingly to the transitional prior. Upon receiving a new measurement the weight for each particle is computed. Because the transitional prior is chosen as the importance density function, Eq.11, which calculates the weight for each particle, simplifies to $\tilde{w}(t)^i \propto p(\mathbf{z}(t)|\mathbf{x}(t)^i)$ which is the likelihood of the measurement vector (real observation) $\mathbf{z}(t) = [z_1, z_2 \dots z_j]$, given the predicted observation $\mathbf{z}(t)^i$, calculated from Eq.9, using the sampled particle $\mathbf{x}(t)^i$.

The measurements $\mathbf{z}(t)$ follow a Gaussian distribution $N(\mu_v, \sigma_v)$, the weight $\tilde{w}(t)^i$ for particle $\mathbf{x}(t)^i$ is calculated from:

$$\tilde{w}(t)^i = \prod_{j=1}^{N_s} \frac{1}{\sqrt{2\pi\sigma_v^2}} \exp\left(-\frac{(\mathbf{z}(t) - \mathbf{z}(t)^i)^2}{2\sigma_v^2}\right) \quad (14)$$

The final step in the ROT-PF algorithm involves resampling, whenever N_{eff} is found to be smaller than N_{thr} .

4.4 Range only Tracking Multiple Model Particle Filter Algorithm ROT-MMPF

To recursively estimate the state vector in the multiple-model case, a multiple model PF algorithm is employed. The state vector in the multiple-model case is the augmented state vector which contains both the state $\mathbf{x}(t)$ and the regime variable $\mathbf{r}(t)$. The augmented state vector is denoted as, $\mathbf{y}(t) = [\mathbf{x}(t) \ r(t)]$.

In this case, initial particles are drawn from two distributions $p(\mathbf{r}_0)$ and $p(\mathbf{x}_0)$. Particles for the state $\mathbf{x}(t)$ are sampled from the transitional prior similar to the ROT-PF algorithm, while particles for the regime variable are sampled according to the transitional probability matrix $\Pi = [\pi_{mn}]$. As with the ROT-PF algorithm, whenever a new measurement vector becomes available the weight for each particle is computed by using the likelihood function $p(\mathbf{z}(t)|\mathbf{y}(t)^i)$, which in this case depends on the augmented state vector. Similar, to the ROT-PF algorithm the predicted observation $\mathbf{z}(t)^i$ is calculated based on the sampled particles of the state vector $\mathbf{x}(t)^i$, using Eq.9. The final step of the ROT-MMPF algorithm includes the resampling step whenever it is necessary.

An iteration of the ROT-PF and the ROT-MMPF algorithms is given in Fig. 6.

```

Initialize
- Draw Initial Particles
for  $i = 1$  to  $N$  do
   $\mathbf{x}_0^i \sim p(\mathbf{x}_0)$ , ( $\sim$ : denotes sampling from)
   $\mathbf{r}_0^i \sim p(\mathbf{r}_0)$  (for the ROT-MMPF case only )
end for
Sequential Importance Sampling Step
- Sample Particles and Calculate Weights
for  $i = 1$  to  $N$  do
   $\mathbf{r}(t)^i \sim \pi_{ij}$  (for the ROT-MMPF case only )
   $\mathbf{x}(t)^i \sim p(\mathbf{x}(t)|\mathbf{x}(t-1))$ 
   $\tilde{w}(t)^i = p(\mathbf{z}(t)|\mathbf{x}(t)^i)$  (for ROT-PF)
   $\tilde{w}(t)^i = p(\mathbf{z}(t)|\mathbf{x}(t)^i, \mathbf{r}(t)^i)$  (for ROT-MMPF)
end for
- Calculate total weight
 $t = \sum_{i=1}^N \tilde{w}(t)^i$ 
- Normalize weights
for  $i = 1$  to  $N$  do
   $w(t)^i = t^{-1} \tilde{w}(t)^i$ 
end for
Resampling Step
if  $N_{eff} < N_{thr}$  then
  - Resample with replacement to obtain N new particles distributed according to  $p(\mathbf{x}(t)|Z_{0:t})$ 
end if

```

Fig. 6. Pseudocode of the ROT-PF and ROT-MMPF Algorithms

5 SIMULATION EVALUATION

This section provides results from simulating the proposed tracking system under various two-dimensional scenarios where a single target is considered. We aim to compare the multiple-model approach against the single-model one and investigate the effect on the achieved accuracy that the sampling interval and number of particles have. Moreover in this section we derive the theoretical Cramer-Rao lower bound of the proposed system and compare it with results obtained from simulations.

To quantify the accuracy achieved in estimating the target's coordinates the Root Mean Square Error

(RMSE) is used. The RMSE for position is defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{T} \sum_{t=1}^T (x(t) - x(t)_{est})^2 + (y(t) - y(t)_{est})^2} \quad (15)$$

5.1 Multiple-Model vs Single-Model

Here a comparison of the two approaches that are employed to form the state equation, takes place. A scenario where the target performs a predefined trajectory which involves a quick two-turn manoeuvre is simulated with the single and multiple-model state formulation. From the Fig. 7 it is clear that the MMPF-ROT algorithm can successfully track the target's manoeuvres. On the other hand the PF-ROT algorithm keeps track of the target during the first turn but loses focus in the second turn and requires some time until the algorithm's output converges back to the target's trajectory.

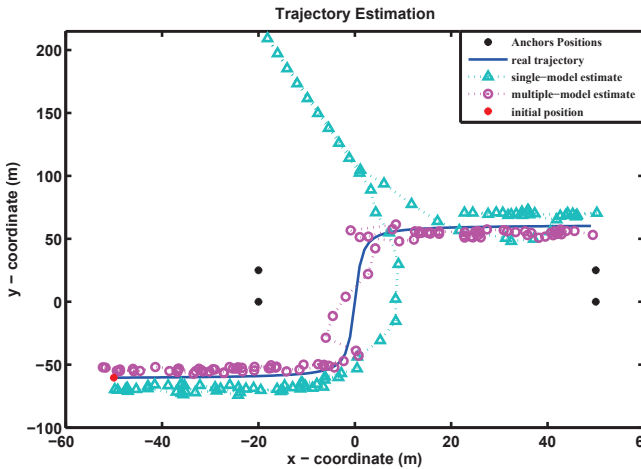


Fig. 7. Comparison of the two models under a manoeuvring scenario

5.2 Sampling Interval - Number of Particles

Two of the system parameters that affect the accuracy of the proposed system in a real-world scenario, is the sampling period T_s and the number of particles the PF algorithms employ to approximate the state vector. Increased T_s translates to smaller number of observations becoming available to the system within a specific amount of time. The Sampling Period is heavily dependent on the amount of time required by the anchor nodes to collect and fuse the ranging estimates. Moreover the Sampling Period is also affected by the time the system requires to run the tracking algorithm and produce an estimate. Conversely increasing the number of generated particles results in

increased accuracy since the posterior pdf is approximated with higher precision. To evaluate the effect of these two parameters, we conduct simulations where initially we simulate the system for constant number of particles $N = 500$ but with increasingly sampling period $T_s = 2, 3, 4, 5, 6, 7 \text{ sec}$ and following we maintain the sampling period at $T_s = 2 \text{ s}$ and use a varying particle size $N = 500, 1000, \dots, 4000$. For each set of values, 100 Monte Carlo trials are conducted. At each execution the RMSE for position was calculated and finally the average RMSE was calculated for the total of 100 runs. In Fig. 8 the average RMSE is illustrated against both of these parameters for the ROT-PF (a,b) and ROT-MMPF (c,d) algorithms respectively.

5.3 Posterior Cramer - Rao lower bounds

The Cramer-Rao lower bound (CRLB) is a theoretically derived lower bound of the second-order error of an unbiased estimator. The CRLB is utilised as a benchmark for evaluating the performance of dynamic estimation algorithms [30], [31]. In the majority of the situations the CRLB is calculated recursively with the use of the Fisher information matrix. This bound is called "posterior" since it is applicable in systems modeled with nonzero process noise [32]. We compute the posterior CRLB for the two approaches, used to formulate the state dynamics of the proposed range-only tracking system. Moreover simulations are presented to assess the performance of the proposed system against the theoretically derived lower bound. For a dynamical estimation problem, the covariance matrix $\mathbf{P}(t)$ of an unbiased estimator $\hat{\mathbf{x}}(t)$ of the state vector at time k has a lower bound (CRLB) which is expressed as :

$$\mathbf{P}(t) \triangleq \mathbb{E}\{(\hat{\mathbf{x}}(t) - \mathbf{x}(t))(\hat{\mathbf{x}}(t) - \mathbf{x}(t))^T\} \geq \mathbf{J}(t)^{-1} \quad (16)$$

where $\mathbf{J}(t)$ is the Fisher's information matrix.

5.3.1 CRLB for the ROT-PF algorithm

In this case the state equation is modeled with the use of the CV model. Considering the process noise to be zero ($\mathbf{w}(t) = 0$), which means a purely deterministic trajectory, matrix $\mathbf{J}(t)$ is recursively calculated from the following [33].

$$\mathbf{J}(t) = [\mathbf{F}^{-1}]^T \mathbf{J}(t-1) \mathbf{F}^{-1} + \sum_{i=1}^{N_s} H(t)_i^T \mathbf{R}(t)_i^{-1} \mathbf{H}(t)_i \quad (17)$$

where: N_s is the number of anchors, $H(t)_i$ is the Jacobian of the measurements equation ($\nabla_{\mathbf{x}(t)} z_i(t)$) with respect to the state vector, evaluated at the true value of $\mathbf{x}(t)$ and $\mathbf{R}(t)_i^{-1}$ is the inverse of the observations noise covariance matrix.

Since the initial density to sample particles from, is chosen to be Gaussian ($p(\mathbf{x}_0) = N(\mathbf{x}_0; \mu_o, \mathbf{P}_0)$) the iteration begins with $\mathbf{J}_0 = \mathbf{P}_0^{-1}$

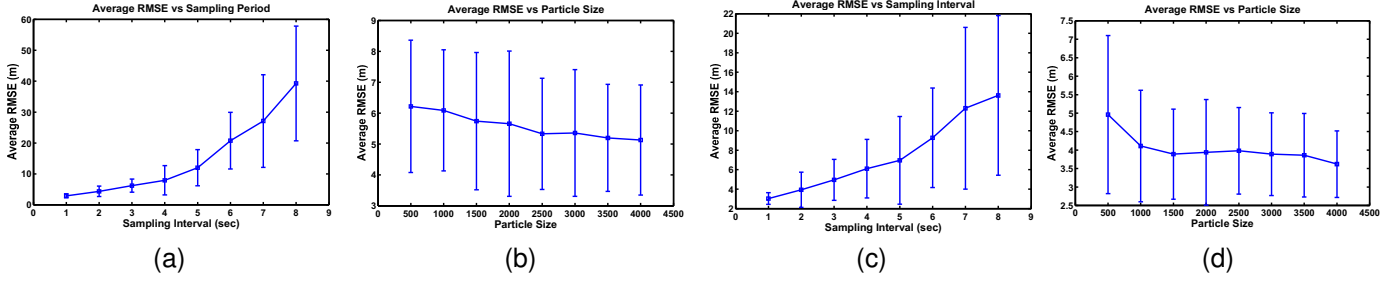


Fig. 8. The effect of the sampling interval and the particle size on the system's accuracy: (a),(b)ROT-PF algorithm. (c),(d) MMPF-ROT algorithm

The Jacobian matrix of the measurements equation $\mathbf{z}(t)$ is a 1×4 matrix given in Eq. 18.

$$\begin{aligned} H(t)_i[1,1] &= \frac{(x(t) - x_i)}{\sqrt{(x(t) - x_i)^2 + (y(t) - y_i)^2}} \\ H(t)_i[1,2] &= \frac{(y(t) - y_i)}{\sqrt{(x(t) - x_i)^2 + (y(t) - y_i)^2}} \\ H(t)_i[1,3] &= 0 \\ H(t)_i[1,4] &= 0 \end{aligned} \quad (18)$$

To compare the ROT-PF algorithm against the CRLB, we simulate the following scenario. Four anchor nodes are considered deployed at coordinates $s_1 = [10 \ 0]$, $s_2 = [50 \ 0]$, $s_3 = [10 \ 25]$, $s_4 = [50 \ 25]$. In line with the theoretical assumptions the process noise is considered zero while the observations noise is considered Gaussian with $\sigma_v = 3.7m$. The target's initial state vector is $x_0 = [10m \ 10m \ 0.1m/s \ 0.1m/s]$ and initial particles are sampled from a Gaussian distribution with $\mu_0 = x_0 + N(0, 1)$ and covariance matrix $\mathbf{S}_0 = \mathbf{J}_0^{-1} = \text{diag}[1 \ 1 \ 1 \ 1]$. This scenario was simulated for 400 time steps for a total of 500 Monte Carlo runs and the CRLB for position was calculated as:

$$\text{CRLB}_{pos} = \sqrt{\mathbf{J}(t)^{-1}[1,1] + \mathbf{J}(t)^{-1}[2,2]} \quad (19)$$

where $\mathbf{J}(t)^{-1}[1,1]$ and $\mathbf{J}(t)^{-1}[2,2]$ are the diagonal elements of the information matrix corresponding to the CRLB for x and y coordinates respectively.

Results are illustrated in Fig. 9, from where it is clear that the achieved RMS error follows a similar to the CRLB trend cases and is bounded by it.

5.3.2 CRLB for the ROT-MMPF algorithm

In the multiple-model case the derivation of CRLB is being done using the same approach as for the single-model case presented previously. Consequently, taking into account a zero process noise system, the CRLB is computed recursively for a sequence of regime variables [34].

Considering a specific sequence of regime variables $r(t)^l \triangleq \{r(1)^l, r(2)^l, \dots, r(t)^l\}$, with $l = 1, 2, \dots, s^t$ being the possible regime values up to time t , the covariance of an estimator of the state vector is given by Eq. 16 conditioned on the particular regime sequence.

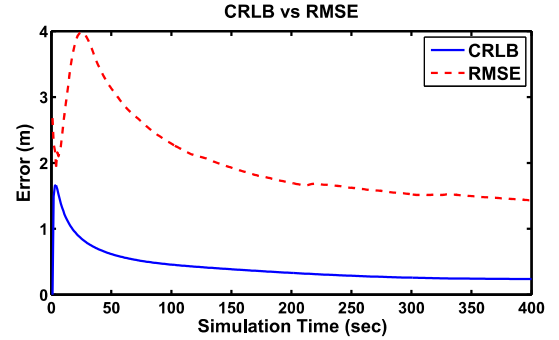


Fig. 9. CRLB for the ROT-PF algorithm

$$\mathbf{P}(t) \triangleq \mathbb{E}\{(\hat{\mathbf{x}}(t) - \mathbf{x}(t))(\hat{\mathbf{x}}(t) - \mathbf{x}(t))^T | r(t)^l\} \geq [\mathbf{J}(t)^l]^{-1} \quad (20)$$

Fisher's information matrix $\mathbf{J}(t)^l$ is computed for the specific regime variable sequence $r(t)^l$ from Eq. 17. The conditional (on the regime sequence $r(t)^l$) CRLB is given as known from the inverse of the information matrix:

$$\text{CRLB}^l(\mathbf{x}(t)) \triangleq [\mathbf{J}(t)^l]^{-1} \quad (21)$$

Considering that at time k the regime variable can be any of the possible s^k different permutations, the unconditional CRLB is calculated as the expectation of the conditional bounds [34],

$$\text{CRLB}(\mathbf{x}(t)) = \sum_{l=1}^{s^k} pr(r(t)^l) [\mathbf{J}(t)^l]^{-1} \quad (22)$$

where $pr(r(t)^l)$ is the forward probability of a particular sequence of regimes, of the first order Markov chain defined by the transition probability matrix $\Pi = [\pi_{ij}]$ (see Eq. 5)

The computational complexity of the CRLB defined in Eq. 22 increases exponentially with time and requires the enumeration of the growing regime sequences. As a result this bound can only be calculated for small numbers of t [35]. For this in a number of works on target tracking [36], [37] an a-priori known regime sequence $R(t)^\epsilon$ is considered. This sets the

probability of that particular sequence to “1” and the probability of any other regime sequence to “0” in Eq. 22. The CRLB in this case is simplified to the following equation (similar to Eq. 17).

$$\begin{aligned} \mathbf{J}(t)^\epsilon &= [[\mathbf{F}(t-1)^\epsilon]^{-1}]^T \mathbf{J}(t-1)^\epsilon [\mathbf{F}(t-1)^\epsilon]^{-1} + \\ &= \sum_{i=1}^{N_s} [H(t, i)^\epsilon]^T \mathbf{R}(t)_i^{-1} \mathbf{H}(t)_i^\epsilon \end{aligned} \quad (23)$$

The evaluation of the CRLB for the MMPF took place both with the enumeration method as well as for a purely deterministic trajectory (absent process noise, a-priori known regime sequence). The simulation setup was similar to the one for the investigation of the CRLB for the ROT-PF algorithm. The target’s initial state vector is $x_0 = [10m \ 10m \ 1m/s \ 1m/s]$. The manoeuvring turning rate was set to $w_r = \pi/3$. Initial particles for the regime variable were sampled with equal probability $P_0 = [1/3 \ 1/3 \ 1/3]$ and the transitional probability is set to $m = 0.95$.

A realisation of a regime sequence for $K = 100$ time steps was produced and it was used to calculate the CRLB in a deterministic way for that particular regime sequence. this scenario was simulated for a total of 500 Monte Carlo runs and the results are shown in Fig. 10b. For the enumeration method the total time steps were set to $K = 12$. Results of 500 Monte Carlo runs are illustrated in Fig. 10a.

From Fig. 10a and Fig. 10b it is clear that the RMSE of the MMPF-ROT system follows a similar trend as the theoretical CRLB for both cases.

6 SYSTEM IMPLEMENTATION AND EXPERIMENTATION

In the prototype implementation of the complete tracking system we designate three different types of wireless embedded nodes. For each different class of nodes a specific piece of software was developed to implement the node’s operation. The T.I. EZ430-RF2500 hardware was used.

6.1 Anchor nodes

The anchor nodes are a number of embedded nodes deployed in known locations and their mission is to interact with the target-node in order for the ranging data to be produced from the two-way ToF technique. Our approach is to designate the anchor nodes as *responder* nodes and have them operate in the exact way that the *responder* node operates in the two-way ToF method.

6.2 Target node

The target node is the mobile object of which the trajectory the system attempts to estimate. For the two-way ToF ranging method we employ, it was deduced that the role of *requester* would be suitable

for the target node. In the resulting system the target initiated the communication between itself and the anchors. Initially, the target engages in a ranging process with the first anchor node; as soon as the nominal number of transactions is achieved and the two-way ToF estimate is calculated, the value is fused to the central node and the target-node carries on and starts ranging with the next anchor node. A data cycle is completed when the target-node has acquired one ranging estimate from every anchor node. The four estimates are sequentially fused to the central node at the moment of their production.

The target-node must exchange ranging transactions on a one-to-one basis with each one of the anchors within a single sampling period. To guarantee this, and prevent message collisions between the target and the anchors, each anchor’s CC2500 radio is programmed to operate on a different communication channel. This approach allows the target-node to complete the ranging process with a specific anchor without the risk of another anchor node intercepting this process which would result in faulty time readings. The target-node is aware of the communication channel that each anchor operates on and loops through these during each sampling interval.

6.3 Central node

The central node is responsible for the collection of ranging estimates and for the execution of the tracking algorithm. Several important challenges were considered in the design of the central-node software. Firstly the issue of time synchronization between the data acquired and the estimates produced. As soon as the central node acquires the required data from the anchor nodes, the execution of the tracking algorithm is initialized and the state estimates are produced. Following, a new set of observations will be available at the central node and the algorithm is executed based on the new set of data to produce the next state estimate. It is imperative to ensure, that both the operations of continuous data accumulation and execution of the algorithm will run in the central node effectively. Under these conditions a choice was made to employ a laptop computer as the platform to execute the PF tracking algorithm. An EZ430-RF2500 node connected to a USB port acts as the bridge between the target-node and the laptop. The central EZ430-RF2500 node forwards each ranging estimate to its UART port which is connected to the laptop and then the software running on the laptop takes over for further processing.

The tracking algorithm (either MMPF-ROT or PF-ROT) is implemented as a MATLAB routine. A top-level script initializes the procedure and it is there where all the relevant system parameters are set. These involve the noise levels, the distribution from which the initial particles are sampled, the target’s

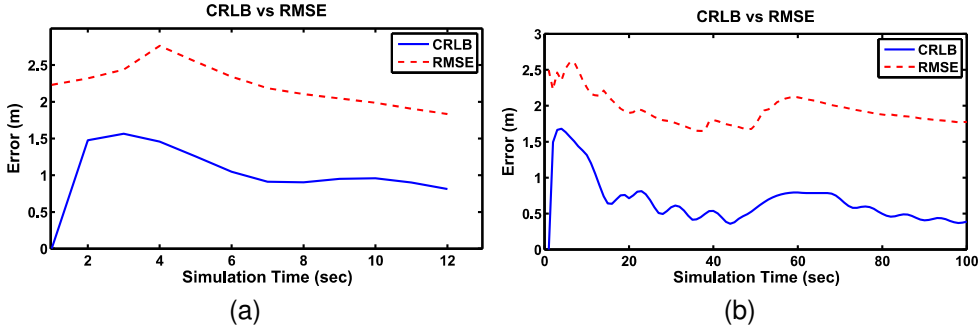


Fig. 10. CRLB for the MMPF-ROT algorithm: (a) using the enumeration method; (b) using a predefined trajectory

initial location, the anchor's positions and the connection parameters with the EZ430-RF2500 central node are all defined before the initialisation of the tracking operation. The front-end script monitors the serial port where the central node is connected and is programmed to signal an interrupt whenever the required amount of bytes (i.e. four range estimates) is accumulated. The tracking algorithm routine is scheduled to run whenever such an interrupt is raised. After the tracking algorithm runs to completion the results are stored and the program waits for another set of ranging data to become available.

Different to our simulation experiments, the sampling interval for the real-world experiments was not set to a constant value. The sampling interval is the time elapsed between two successive executions of the tracking algorithm or in terms of the state-space model, the time elapsed between the current and the previous state vectors. It is a parameter which has a significant effect on the system's performance as it was revealed through the simulations investigation. In the real-world experiments the sampling interval is affected primarily by the amount of time required to collect a ranging estimate from all the anchors. We have observed that on the EZ430-RF2500 the ToF method ideally requires approximately 152ms to execute on the *requester* node with 100 two-way transactions. Subsequently a sampling interval of 1s can be achieved in the presence of four anchors, if no delays are introduced.

Nevertheless, in the event that the connection between the target and the anchors is not ideal, retransmissions of the ranging messages may be required in order to reach the nominal number of 100 two-way transactions, which is required to obtain a ToF range estimate between the target and the respective anchor. Subsequently this will result in an increase of the sampling interval since more time is required to obtain the range estimates. To avoid using an erroneous sampling interval in our model, an adaptive scheme is employed. The sampling interval is calculated in MATLAB as the required amount of time to obtain the four range estimates (time when the interrupt is raised). The value of a real-world clock is captured

whenever an interrupt is raised. By subtracting the previous value of that clock the sampling interval can be calculated. Using this method, we guarantee that the state-update model that is employed takes into account the varying amount of time that has lapsed between two successive executions of the tracking algorithm.

7 EXPERIMENTS EXECUTION AND RESULTS

7.1 Preliminaries

In the experiments carried out the four anchors were placed in known positions in the corners of a 15m x 15m square area with excellent LoS conditions. The central processing node was placed on the top of the square area. In our experiment we restrained in tracking a single mobile node. The target mobile node was carried in the hands of a person which was walking in the designated square area. Finally the nodes were strapped on plastic traffic poles and elevated from the ground (1m) in order to avoid potential deflection of the RF signals from the ground. The target node was carried at a similar height as the anchor nodes.

In all experiments the nominal number of two-way ranging transactions with each anchor node was set to 100. The reason for choosing 100 two-way transactions to estimate the range between the target and each anchor is related to the required real-time system operation. The EZ430-RF2500 does not have enough memory to store more than 100 timing values, which means that in order to utilise more than 100 transactions the ranging routine should be executed multiple times. This was done without a problem in the ranging experiments where real-time operation was not a critical component. However in the tracking system where the ranging estimates must reach the central node in a timely manner with minimum delay such an approach would add significant latency that would hinder the ability for real-time operation. Due to these issues, we chose to proceed with setting the nominal number of ranging transactions to 100.

The rest of the parameters for the tracking algorithms were defined as follows. The state noise

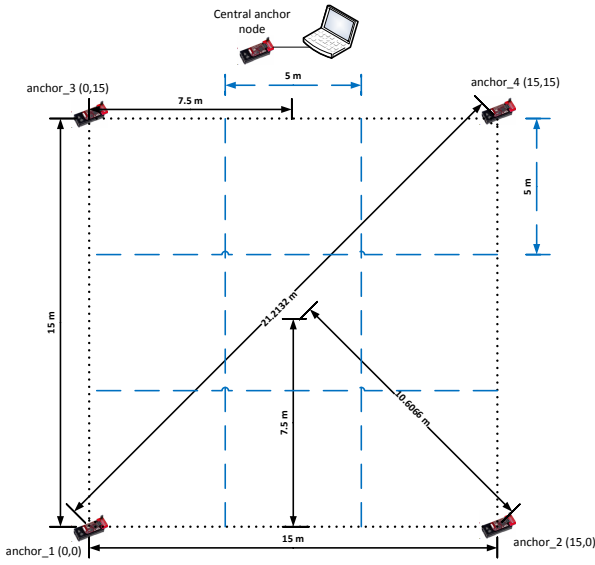


Fig. 11. The deployment of the anchor nodes and the dimensions of the experimental area

was defined as zero-mean white Gaussian noise with $\sigma_w = 0.5$. Similarly the observations noise is also defined as zero-mean white Gaussian with $\sigma_v = 3.7$. The target's initial state was known. The sampling interval was set as described previously (6.3). In the implementation of the PF tracking algorithms the particle size was set to $N = 1500$ to provide improved robustness in case the sampling period increased. The distribution to sample the initial particles from was a Gaussian distribution with zero mean and unity covariance. Finally the transition probability for the regime variable was set at $m = 0.8$ for the MMPF-ROT algorithm. The constant turning rate was set at $\pi/4 \text{ rad/s}$. The datarate was set to 500kbps.

To calculate the proposed system's tracking accuracy the output obtained from the system is compared with respect to the ground truth of the target. To effectively measure the ground truth, the target's trajectory was predefined before the experiments and divided into individual segments at which the target moved at a straight line with approximately constant speed. During the execution of the experiment we recorded the times when the ranging data from all nodes were collected (reached the central node). Additionally, the total time of each individual straight line segment was recorded. This facilitated the computation of the target's true velocity in each segment of it's trajectory. In addition by knowing the estimation times, we were able to interpolate and calculate the target's ground truth position and velocity at the time of the execution of the algorithm. Following this approach, the estimation produced by the system could be compared against an accurate enough approximation of the true target variables.

7.2 Experimental Results

In this section we present results from a number of experiments that were carried out with the proposed tracking system. The experiments are categorized in three groups based on the target's trajectory. Straight Line Trajectories, Trajectories involving one or two manoeuvres, Trajectories involving more than two manoeuvres. Examples of tracking results are illustrated in Fig. 12. A total of 25 tracking experiments are investigated in this sections. The position and velocity errors are calculated with respect to the ground truth. The collective performance results are illustrated in Table. 1.

	Position RMSE (m)	x-velocity RMSE (m/s)	y-velocity RMSE (m/s)
Average	2.62	1.57	1.22
Best Case	1.4612	0.1883	0.7864
Worst Case	4.0774	3.4838	2.5435

TABLE 1
Accuracy results from 25 outdoor experimental executions

7.2.1 Comparison to Simulation Results

Here we present a comparison between the results obtained from the full-scale experiments with simulations results obtained after simulating multiple times a tracking scenario similar to the one we experimented with in the full-scale experiments. Four anchors were considered placed in coordinates $((0\text{m},0\text{m}), (15\text{m},0\text{m}), (0\text{m},15\text{m}), (15\text{m},15\text{m}))$ exactly as in the outdoors deployment. The system parameters (target's initial state, distribution to sample initial particles), the PF algorithm parameters (particle size) and the noise levels, where the same as in the full-scale experiments.

To approximate the behaviour the system demonstrated in the full-scale experiments, we used a varied sampling interval. Based on observations from the real-world experiments, the minimum observed value of the sampling interval during the experiments was around 0.8s and the maximum one around 2.2s. To approach this in our simulations we randomized the sampling interval variable between 0.8 and 2.2s. The simulations were run for 30 time steps and with each step having varying sampling interval. This approach resulted in simulation executions that run for a total time similar to the one that the real-world experiments lasted for. We simulated this scenario for 100 runs and included both random and deterministic trajectories. The deterministic trajectories were the same as the ones used in the full-scale experiments and included straight line trajectories as well trajectories with predefined manoeuvres. The average RMSE obtained from the simulation analysis is 2.5m, a result which is very close to the one (2.6m) observed in our

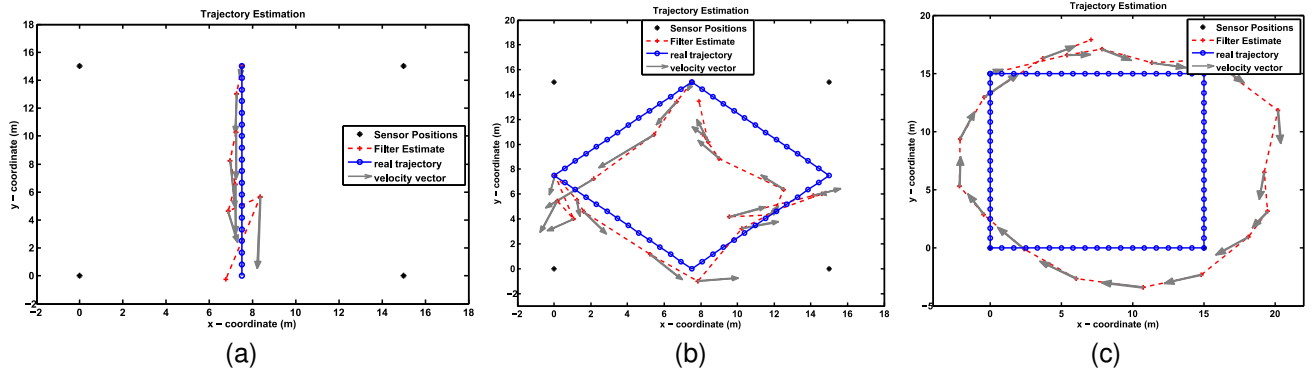


Fig. 12. Tracking results on various trajectories (a) Straight line trajectory (b),(c) Trajectories involving turning manoeuvres

real-world experiments. Additionally the accuracy in the velocity estimation obtained from simulations is 1.89m/s which again is similar to the one obtained in the full-scale experiments (1.9m/s). The RMSE for position, from 100 simulations of this simulation set-up are illustrated in Figure 13.

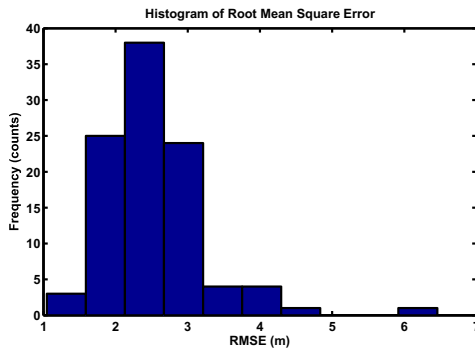


Fig. 13. Histogram of RMSE evaluated from 100 simulations under similar conditions with our real-world experiments

8 CONCLUSIONS

In this paper we have presented the design, implementation and evaluation of a real-time, range-only target tracking system for wireless embedded nodes. The motivating idea of this work considers a small number of wireless embedded nodes (anchors) to be deployed in known coordinates tasked with acquiring ranging information and a central node which receives the accumulated data and executes the tracking algorithm in real-time, to estimate the target's position and velocity. The proposed system aims to achieve accurate performance, real-time operation as well as provide support for manoeuvring targets. The proposed system is based on range observations produced by utilising a two-way ToF ranging method. To provide enhanced support for manoeuvring targets, the target's pattern is modeled with the use of multiple switching motion models. Two

Particle Filters based algorithms were designed for the tracking system. The performance of the proposed system is evaluated in a simulation environment and we quantify the effect that two important system parameters, the number of particles and the sampling interval have on the system's accuracy. Additionally we deduce the CRLB and compare the performance of the system against the theoretical bound. Finally we implement the entire tracking system on T.I. EZ430-RF2500 hardware and conduct experiments in an outdoor area of $225m^2$. From a total of 25 experiments, an average accuracy of 2.6m for position and 1.9m/s for velocity was observed. Additionally, the simulation investigation attests and verifies the accuracy levels and the system's performance, as it was demonstrated in the real-world experiments.

REFERENCES

- [1] D. Culler, D. Estrin, and M. Srivastava, "Guest editors' introduction: Overview of sensor networks," *IEEE Computer*, vol. 37, no. 8, pp. 41–49, Aug. 2004.
- [2] T. Arampatzis, J. Lygeros, and S. Manesis, "A survey of applications of wireless sensors and wireless sensor networks," in *Proc. 13th IEEE Mediterranean Conf. on Control and Automation*, 2005, pp. 719–724.
- [3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Comput. Netw.*, vol. 38, no. 4, Mar. 2002.
- [4] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 61–72, Mar 2002.
- [5] N. Patwari, J. Ash, S. Kyperountas, I. Hero, A.O., R. Moses, and N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54 – 69, jul. 2005.
- [6] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *IEEE Computer*, vol. 34, no. 8, pp. 57–66, Aug. 2001.
- [7] X. Sheng and Y. H. Hu, "Sequential acoustic energy based source localization using particle filter in a distributed sensor network," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Montreal, Quebec, Canada, 2004.
- [8] X. Sheng, Y.-H. Hu, and P. Ramanathan, "Distributed particle filter with gmm approximation for multiple targets localization and tracking in wireless sensor network," in *4th ACM/IEEE Int. Symp. Information Processing in Sensor Networks*, Los Angeles, CA, USA, 2005, pp. 181–188.

- [9] H. Ma and B.-H. Ng, "Distributive target tracking in wireless sensor networks under measurement origin uncertainty," in *Proc. 3rd Int. Conf. on Intelligent Sensors, Sensor Networks and Information (ISSNIP)*, Melbourne, Australia, 2007, pp. 299–304.
- [10] E. Mazomenos, J. Reeve, and N. White, "Tracking with range-only measurements using a network of wireless sensors," in *6th Intl Conf. on Broadband Communications, Networks, and Systems*, 2009, pp. 80–88.
- [11] M. Coates, "Distributed particle filters for sensor networks," in *Proc. 3rd Int. Symp. Information Processing in Sensor Networks*, 2004, pp. 99–107.
- [12] M. Coates and G. Ing, "Sensor network particle filters: motes as particles," in *IEEE/SP 13th Workshop on Statistical Signal Processing*, 2005, pp. 1152–1157.
- [13] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proc. 6th Ann Intl. Conf. on Mobile computing and networking*, 2000, pp. 32–43.
- [14] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Mobile-Assisted Localization in Wireless Sensor Networks," in *Proc. 24th IEEE Intl. Conf. on Computer Communications*, 2005.
- [15] C. Taylor, A. Rahimi, J. Bachrach, H. Shrobe, and A. Grue, "Simultaneous localization, calibration, and tracking in an ad hoc sensor network," in *Proc. 5th Intl. Conf. on Information Processing in Sensor Networks (IPSN)*, 2006, pp. 27–33.
- [16] P. Bahl and V. N. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," in *Proc. IEEE 19th Ann. Joint Conf. of the IEEE CS and Comm. Societies*, vol. 2, 2000, pp. 775–784.
- [17] N. Ahmed, Y. Dong, T. Bokareva, S. Kanhere, S. Jha, T. Bessell, M. Rutten, B. Ristic, and N. Gordon, "Detection and tracking using wireless sensor networks," in *Proc. of the 5th Intl. Conf. on Embedded Networked Sensor Systems (SenSys)*, 2007, pp. 425–426.
- [18] N. Ahmed, M. Rutten, T. Bessell, S. Kanhere, N. Gordon, and S. Jha, "Detection and tracking using particle-filter-based wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 9, pp. 1332–1345, Sep 2010.
- [19] M. Maróti *et al.*, "Radio interferometric geolocation," in *Proc. of the 3rd Intl. Conf. on Embedded networked sensor systems*, 2005, pp. 1–12.
- [20] B. Kusý, G. Balogh, J. Sallai, Ákos Lédeczi, and M. Maróti, "intrack: High precision tracking of mobile sensor nodes," in *Proc. European Conf. in Wireless Sensor Networks (EWSN)*, vol. LNCS 4373, 2007, pp. 51–66.
- [21] B. Kusý, A. Ledeczi, and X. Koutsoukos, "Tracking mobile nodes using rf doppler shifts," in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys '07)*, 2007, pp. 29–42.
- [22] B. Kusý, J. Sallai, G. Balogh, A. Ledeczi, V. Protopopescu, J. Tolliver, F. DeNap, and M. Parang, "Radio interferometric tracking of mobile wireless nodes," in *Proc. 5th Intl. Conf. on Mobile systems, Applications and Services (MobiSys)*, 2007, pp. 139–151.
- [23] I. Amundson, X. Koutsoukos, and J. Sallai, "Mobile sensor localization and navigation using rf doppler shifts," in *Proc. 1st ACM Intl. workshop on Mobile entity localization and tracking in GPS-less environments (MELT)*, 2008, pp. 97–102.
- [24] E. B. Mazomenos, D. D. Jager, J. S. Reeve, and N. M. White, "A two-way time of flight ranging scheme for wireless sensor networks," in *8th European Conf. on Wireless Sensor Networks (EWSN)*, 2011, pp. 163–178.
- [25] M. S. Arulampalam, B. Ristic, N. Gordon, and T. Mansell, "Bearings-only tracking of manoeuvring targets using particle filters," *EURASIP J. Appl. Signal Process.*, vol. 2004, pp. 2351–2365, January 2004.
- [26] K. Radhakrishnan, A. Unnikrishnan, and K. Balakrishnan, "Bearing only tracking of maneuvering targets using a single coordinated turn model," *International Journal of Computer Applications*, vol. 1, no. 1, pp. 25–33, February 2010.
- [27] R. Srinivasan, *Importance Sampling: Applications in Communications and Detection*. New York: Springer-Verlag, 2002.
- [28] A. Doucet, N. De Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*. New York: Springer - Verlag, 2001.
- [29] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Trans. on Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb 2002.
- [30] B. Bobrovsky and M. Zakai, "A lower bound on the estimation error for markov processes," *IEEE Trans. Aut. Control*, vol. 20, no. 6, pp. 785 – 788, dec 1975.
- [31] J. H. Taylor, "The cramer-rao estimation error lower bound computation for deterministic nonlinear systems," in *Proc 1978 IEEE Conf. on Decision and Control (CDC)*, 1978, pp. 1178 –1181.
- [32] P. Tichavsky, C. Muravchik, and A. Nehorai, "Posterior cramer-rao bounds for discrete-time nonlinear filtering," *IEEE Trans. Signal. Process.*, vol. 46, no. 5, pp. 1386 –1396, may 1998.
- [33] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter - Particle Filters For Tracking Applications*. Artech House Publishers, 2004.
- [34] A. Bessell, B. Ristic, A. Farina, X. Wang, and M. Arulampalam, "Error performance bounds for tracking a manoeuvring target," in *Proc 6th Intl. Conf. on Information Fusion*, 2003., 2003, pp. 903 – 910.
- [35] A. Farina, B. Ristic, and L. Timmoneri, "Cramer-rao bound for nonlinear filtering with pd lt;1 and its application to target tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 8, pp. 1916 –1924, aug 2002.
- [36] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li, *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: John Wiley & Sons, Inc., 2002.
- [37] W. Blair, G. Watson, T. Kirubarajan, and Y. Bar-Shalom, "Benchmark for radar allocation and tracking in ecm," *IEEE Trans. on Aerosp. Electron. Syst.*, vol. 34, no. 4, pp. 1097 –1114, oct 1998.