

# Improved Animal Tracking Algorithms Using Distributed Kalman-based Filters

(Invited Paper)

Nelson I. Dopico, Benjamin Bejar, Sergio Valcarcel Macua, Pavle Belanovic and Santiago Zazo

ETSI de Telecomunicacion

Universidad Politecnica de Madrid (UPM)

Ciudad Universitaria, 28040 Madrid, Spain

Email: {nelson,min,sergio,pavle,santiago}@gaps.ssr.upm.es

**Abstract**—Animal tracking has been addressed by different initiatives over the last two decades. Most of them rely on satellite connectivity on every single node and lack of energy-saving strategies. This paper presents several new contributions on the tracking of dynamic heterogeneous asynchronous networks (primary nodes with GPS and secondary nodes with a kinetic generator) motivated by the animal tracking paradigm with random transmissions. A simple approach based on connectivity and coverage intersection is compared with more sophisticated algorithms based on ad-hoc implementations of distributed Kalman-based filters that integrate measurement information using Consensus principles in order to provide enhanced accuracy. Several simulations varying the coverage range, the random behavior of the kinetic generator (modeled as a Poisson Process) and the periodic activation of GPS are included. In addition, this study is enhanced with HW developments and implementations on commercial off-the-shelf equipment which show the feasibility for performing these proposals on real hardware.

## I. INTRODUCTION

Animal localization and tracking is technically feasible for some time now as GPS devices are in use since the 90s. Over the last two decades different systems have been proposed to reach such goal by means exclusively of satellites [1]. They have been widely used in turtle [2], duck [3] or whale [4] tracking. However, its use is extremely expensive and requires all the satellite transmitters on the animals to be updated in a given satellite database. Other approaches make use of storage systems based on solar energy as environmental energy source [5], [6]. Some of these implementations have been used for animal tracking, as in the *zebranet* project [7] or the *Turtlenet* project [8]. Other systems may become unfeasible due to the cost per head of herd or because of the environmental impact of lost or damaged batteries.

This paper deals with new methods for animal tracking in environments where the use of batteries has to be minimized and replaced by energy harvesting procedures. The baseline for the current article was proposed in [9] where basically two kinds of nodes are defined: primary nodes which integrate a GPS module for their own positioning and secondary nodes with a kinetic generator that just broadcasts their identification according to the animal movement. Upon such a broadcast from a secondary, a primary node in the secondary transmission range may receive the beacon and store its payload (secondary ID) along with the time stamp and the

This work has been supported by the Community of Madrid and the European Social Fund, partially by the Spanish Ministry of Science and Innovation under the grant TEC2009-14219-C03-01; the Spanish Ministry of Science and Innovation in the program CONSOLIDER-INGENIO 2010 under the grant CSD2008-00010 COMONSENS; the European Commission under the grant FP7-ICT-2009-4-248894-WHERE-2; the European Commission under the grant FP7-ICT-223994-N4C and the Spanish Ministry of Science and Innovation under the complementary action grant TEC 2008-04644-E; Spanish Ministry of Science and Innovation under the grant TEC2010-21217-C02-02-CR4HFDVL.

last positioning provided by the GPS. Therefore, a heterogeneous asynchronous dynamic network is meant to be the network paradigm across the article.

The aforementioned positioning/tracking procedure is simple, but typically provides poor estimates. In many applications, for instance if animals are close to motorways, accuracy needs to be improved. The first approach is presented in this paper as *Connectivity + Fusion (C-F)* procedure where every primary node collects the information while the Fusion Center calculates the overlapping of the coverage and evaluates the center of mass of this region as the estimate of the secondary-node position.

A second approach, namely *Decentralized Kalman-like (DCL)* procedures, enables much more sophisticated algorithms where every primary node integrates the information related to the distance measurement and subsequently, as a whole, reaches an agreement by exchanging local messages. In this paper, we extend some preliminary results already published by the authors [10] where some new distributed procedures by means of Consensus algorithms [11] are applied to solve specific implementations of Kalman-based filters. Such preliminary contributions are extended to this particular scenario including some specific features:

- 1) Random transmission of secondary nodes. This randomness is modelled as a Poisson process of different occurrence parameter ( $\lambda$ ) that may represent different movements of animals (eating, coordinated movement as a herd, isolated movement...).
- 2) Limited range of connectivity. Only a reduced number of primary nodes are within the range of the secondary node. However, we assume that, for instance using flooding algorithms, all the primary nodes are aware of the secondary positioning estimates and can use this information a priori while they are within its coverage area.
- 3) In order to save energy, the primary-node GPS module is turned on periodically. GPS duty cycle is a very important parameter because an additional error source arises since primary nodes are continuously moving too. In order to make a simple model that takes into account this fact, we have evaluated the distribution of the positioning of an arbitrary animal in a certain interval reaching the conclusion that the distance run in both axes keeps to a zero-mean double exponential distribution with a standard deviation that is proportional to the GPS activation interval ( $\sigma=13T_{GPS}$  meters/minute).

These theoretical contributions are eventually enhanced with a real test bed (with individuals moving) with well defined trajectories in order to evaluate the degradation effect of the real world: real distance measurement errors –quantized RSSI, multipath...–, message collisions and constrained computational resources in the motes to

run complex algorithms.

Hereafter, four more sections make up the paper. Section 2 describes the hardware used, the characteristics of every scenario considered and their parameters. Section 3 comprises both the experiments run –based on the previous taxonomy– and the results obtained. Section 4 regards a test bed from which samples were collected to serve as an input for the presented algorithms. Conclusions make up Section 5 and complete the article.

## II. SCENARIO

Different kind of nodes operate in our network on study. Target nodes are the nodes to be tracked. Anchor nodes know their own location and aim at tracking target nodes. At times –mainly regarding C-F tracking– the term primary node is used instead of anchor node and likewise for secondary node and target node.

Besides the nodes themselves, our scenario consists of two operational modes which have been tested with real equipment both in experimental simulations and test beds. The first operational mode (*Connectivity + Fusion, C-F*) is based on connectivity to estimate the location of a target node given the knowledge gained across the anchor nodes (listeners) which a fusion center uses to compute the joint center of mass from the candidate area of every anchor as an approximation of the real target position.

The second operational mode (*Decentralized Kalman-like, DCL*) relies on regular wireless sensor nodes with extended computational capabilities, namely iMote2 [12]. The equipment used for the C-F mode could not be used in this one as it is customized hardware designed and programmed for a very specific application [13] which could not integrate such extended computational capabilities within the timing of the current study. However, such a difference can be overridden since it is a proof of concept what is aimed rather than introducing a commercial off-the-shelf product (COTS).

In addition to the aforementioned particular procedures used to localize or track a target, the current study focuses on three parameters to express quantitative differences across every scenario: GPS duty cycle, beacon inter-transmission period and transmission range.

- **GPS Duty Cycle.** Every primary node integrates a GPS module, but it is not powered permanently due to energy constraints. Consequently, its operation turns out to be a trade off between tracking accuracy and sensor-node lifetime. Our study considers different duty cycles expressed as the time elapsed between consecutive activations of the GPS module. Such activations just comprise the time necessary for synchronizing with the satellites, obtain a positioning record and deactivate the module. Activations are uniformly distributed across time. The minimum power-saving period is 2 minutes as it is the minimum activation period from a cold start [9]; any other below two is feasible, but means that the GPS module is powered on permanently.
- **Beacon Inter-Transmission Period.** Ideally, a target node would transmit permanently with a high granularity and the anchor nodes would have a number of samples to be processed which would enable them to track it smoothly. However, transmissions can become scarce for a number of reasons –animal behavior, fading, shadowing...– and our tracking system will have to face an irregular broadcasting pattern which may impact on its performance. As a consequence, different transmission patterns were simulated so that the scarcity of beacons on the overall system performance could be assessed. Eight different transmission patterns for the same trajectory were evaluated. The first has an equally distributed inter-transmission period of 0.1 seconds. From that pattern seven more are derived following a Poisson

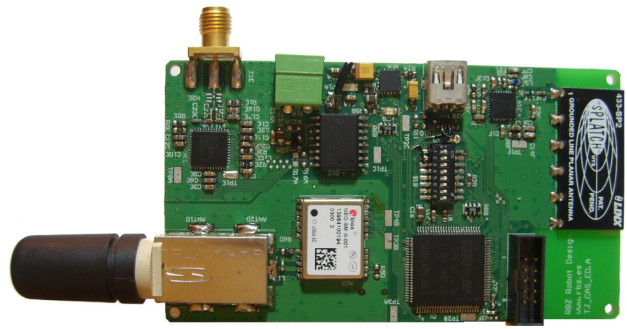


Fig. 1. Primary node



Fig. 2. Collar with both primary and secondary nodes mounted on a reindeer

process, i.e. their inter-transmission times keep to an exponential distribution with  $\lambda$  equal 0.15, 0.2, 0.3, 0.5, 0.7, 1 and 2 seconds. This way, we can simulate different kind of movements such as feeding, foraging, swarm movements, etc.

- **Transmission range.** Saving-power policies, hardware features or animal movements (like in a secondary node) may influence on the transmission power. Although one would like to reach some hundreds of meters, hardware –both custom [9] or COTS [12]– usually constraints it to a few tens. In order to reflect the current state of the art, several transmission ranges have been studied given as the radius of a coverage disk. Its variation will make more or less anchor nodes *hear* the beacons, thus pushing the system to operate with more or less information.

### A. Connectivity-and-Fusion Localization

1) *Hardware:* Two kind of nodes implement the current operational mode. The first kind, namely primary node or anchor node, is battery powered and integrates a GPS module, a low-power ARM processor and two transceivers: 433 MHz (primary-fusion-center link) and 166 MHz (secondary-primary link). The GPS module spends 2 minutes from a cold start and, as explained earlier, such time is the threshold for the power-saving duty cycle. The secondary node (target node) is kinetically powered from animal motion. It does not integrate any microprocessor, but a 166 MHz transceiver which transmits beacons containing the secondary ID. Both were designed to be mounted on a collar around the neck of a reindeer as shown in Fig.2. This application has been motivated by [13], where the project focuses on improving the life quality of a remote threatened community – Sami people.

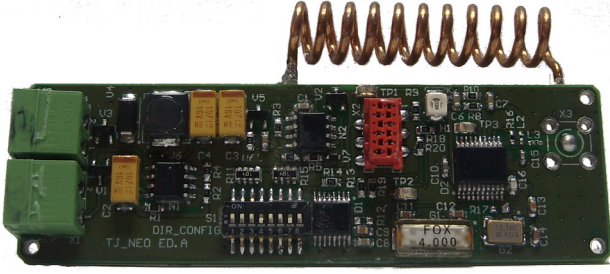


Fig. 3. Secondary node

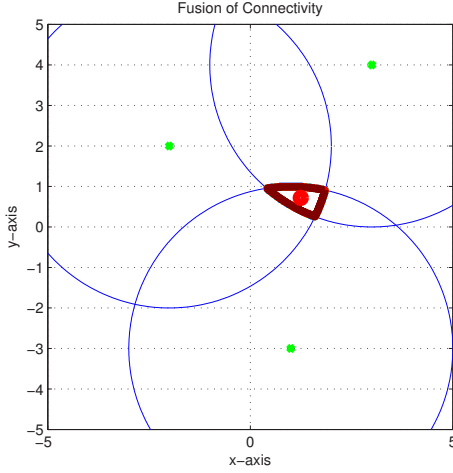


Fig. 4. Example of a C-F result

2) *Operation*: C-F Localization relies on three elements to operate besides the mere existence of the aforementioned hardware:

- 1) Transmission radius. Such a radius can be set from the system characterization and is the outcome of a trade-off between the maximum transmission range –which would imply a high beacon loss– and a (short) radius which may guarantee to be received nearly every beacon broadcasted.
- 2) Primary nodes within the transmission range. Certain amount of primary nodes must be able to *hear* the target node, otherwise the system will have no information regarding the target presence.
- 3) Fusion center. A fusion center collects the information regarding the beacons received by the primary nodes across the network and computes a position estimation.

A target node is expected to broadcast beacons as it moves across a certain area in which our localization network operates. Beacons are packets which may include a sequence number to assist coordination tasks, but here we will just consider the case in which a target node transmits beacons randomly as detailed earlier. Upon their reception by the primary nodes, they subsequently transmit such information to a base station or fusion center – mechanisms to do that follow standard routing procedures and will not be reported here. The fusion center then computes the intersection of the area from which every primary node may have received the beacon –here comes to play the transmission radius– taking into account the last GPS reading of every primary node. Then, the target location estimate is approximated by the center of mass of the intersection.

## B. Decentralized Kalman-like Tracking

In order to obtain estimates of the target position, nodes employ RSSI measurements. We assume that the RSSI follows a linear relationship with the received power  $P_R$ . A common assumption, see [14] and references therein, is that the received power follows a lognormal distribution with a distance-dependent mean as

$$P_R[\text{dB}] = P_0 - 10n_p \log_{10} \left( \frac{d}{d_0} \right) + X, \quad (1)$$

where  $P_0$  is the received power (in dB) at reference distance  $d_0$ ,  $n_p$  is the path-loss exponent and  $X$  is a Gaussian random variable of zero mean and variance  $\sigma_{\text{dB}}^2$ . Let us denote  $P_{R,i}$  as the measured power at the  $i$ -th locating node. The maximum likelihood estimate of the distance to the target is then given by

$$\hat{d}_i = d_0 10^{\left( \frac{P_0 - P_{R,i}}{10n_p} \right)}. \quad (2)$$

In the following each node  $n$  is assumed to be capable of computing an estimate  $\hat{d}_n$  of its distance to the target. It is also assumed that each node is aware of its own position by using an appropriate self-location algorithm [15].

For determining the true target location, the following set of equations should be satisfied

$$\begin{aligned} d_1^2 &= (x_1 - x_0)^2 + (y_1 - y_0)^2 \\ d_2^2 &= (x_2 - x_0)^2 + (y_2 - y_0)^2 \\ &\vdots \\ d_N^2 &= (x_N - x_0)^2 + (y_N - y_0)^2 \end{aligned} \quad (3)$$

where  $d_i$ ,  $i = 1 \dots N$  is the distance between the target and the  $i$ -th node,  $[x_i \ y_i]^T$  are the node coordinates, and  $[x_0 \ y_0]^T$  are the target coordinates.

By further developing (3) and considering measured data (distance estimates) we can express the target localization problem as the following optimization problem [10]:

$$\hat{\mathbf{x}} = \min_{\mathbf{x}} \frac{1}{2} \left\| \begin{pmatrix} \mathbf{x}^T \mathbf{x} \\ \mathbf{1} - 2\mathbf{A}\mathbf{x} - \mathbf{c} \end{pmatrix} \right\|^2, \quad (4)$$

where  $\mathbf{1}$  is a  $N \times 1$  vector of all ones,  $\mathbf{A}$  is a  $N \times 2$  with  $i$ -th row given by  $\mathbf{a}_i^T = [x_i \ y_i]$  and  $\mathbf{c}$  is a  $N \times 1$  row vector whose  $i$ -th entry is given by  $c_i = d_i^2 - x_i^2 - y_i^2$ . The optimization problem (4) represents a nonlinear least-squares problem that can be solved in a distributed way as in [10] by using a distributed consensus-based version of the standard Gauss-Newton algorithm.

In general, the tracking problem can be represented by two equations, one describing the dynamics of the state variable (i.e. variable to be tracked) and another one that relates the state variable to some measurement. The general state-space representation is given by

$$\mathbf{s}^{(k)} = \mathbf{f} \left( \mathbf{s}^{(k-1)}, \mathbf{w}^{(k)} \right) \quad (5)$$

$$\mathbf{z}^{(k)} = \mathbf{g} \left( \mathbf{s}^{(k)}, \mathbf{n}^{(k)} \right) \quad (6)$$

where  $\mathbf{s}^{(k)}$  is the state variable at time instant  $k$  and  $\mathbf{w}^{(k)}$  is the driving noise process. The variable  $\mathbf{z}^{(k)}$  represents the measurement at time instant  $k$  and  $\mathbf{n}^{(k)}$  is the measurement noise process. The functions  $\mathbf{f}(\cdot)$  and  $\mathbf{g}(\cdot)$  may be nonlinear functions of the state and noise processes.

Regarding the target movement it is assumed that it follows a random force movement. The target can be placed at any arbitrary

position in the network area and can move freely through the network as described by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_k \Delta t + \frac{1}{2} \mathbf{a}_{k+1} \Delta t^2 \quad (7)$$

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \mathbf{a}_{k+1} \Delta t, \quad (8)$$

where  $\mathbf{x}_k$  is the target position at time instant  $k$ ,  $\mathbf{v}_k$  is the target speed,  $\mathbf{a}_k$  is the acceleration and  $\Delta t$  is the elapsed time between consecutive samples. It is assumed that the target is initially at some position  $\mathbf{x}_0 = [x_0, y_0]^\top$  with initial speed of  $\mathbf{v}_0 = [v_0^x, v_0^y]^\top$  and that the acceleration follows a Gaussian distribution,  $\mathbf{a} \sim \mathcal{N}(\mathbf{0}, \sigma_a^2 \mathbf{I})$ . The elapsed time between consecutive samples is not constant, but an irregular pattern which keeps to an exponential distribution,  $\Delta t \sim \text{Exp}(\lambda)$ . In the following, we will describe three different contributions in order to propose energy-efficient distributed implementations.

1) *Kalman Filtering on the joint estimate*: In this subsection we consider a distributed approach that uses the Kalman filter for tracking the target. In the considered system model each node (or sensor) measures the received power coming from the target node which is a non-linear function of the state variable. However if we consider a distributed target localization using the distributed Gauss-Newton method for localization [10] we could use this measurement to feed a Kalman tracker on each node separately. As the number of nodes increases and, by the central limit theorem, we can approximate

$$\hat{\mathbf{x}} = \mathbf{x} + \mathbf{e}, \quad (9)$$

where  $\mathbf{e}$  is an error term that follows a Gaussian distribution. It is well known that the Kalman filter provides the optimal tracking filter when the system is linear and Gaussian. A simple introduction to the Kalman filter can be found in [16]. If the Gaussian approximation holds, we could use the jointly estimated quantity (9) as the (noisy) input of a Kalman filter tracker. To express it more formally consider the state variable  $\mathbf{s}^{(k)} = [\mathbf{x}^{(k)\top} \mathbf{v}^{(k)\top}]^\top$  of the position and velocity of the target at time instant  $k$ . Using the target movement model in (7) and considering joint distributed estimation of the target position via consensus Gauss-Newton [10], the following state-space representation results:

$$\mathbf{s}^{(k)} = \mathbf{F} \mathbf{s}^{(k-1)} + \mathbf{W} \mathbf{w}^{(k)} \quad (10)$$

$$\mathbf{z}^{(k)} = \mathbf{G} \mathbf{s}^{(k)} + \mathbf{e}^{(k)}, \quad (11)$$

where

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{W} = \sigma_a \begin{bmatrix} \Delta t^2/2 & 0 \\ 0 & \Delta t^2/2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

and the noise processes are  $\mathbf{w}^{(k)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ , where  $\mathbf{I}$  is the identity matrix and  $\mathbf{e}^{(k)}$  is the error term of (9) at time instant  $k$ .

Using this tricky representation we can construct a Kalman filter tracker as described in Algorithm 1. It is worth to mention that the Gaussian approximation becomes more accurate as the number of locating nodes increases. However, even for a small number of nodes the proposed approach appears to give good performance.

2) *Distributed Unscented Kalman Filter*: Instead of using the joint estimate for linearizing the problem and having an approximate Gaussian state-space representation there exists other approaches of the Kalman filter that try to cope with nonlinearity and/or non-Gaussianity. One of the classical approaches is the Extended Kalman

---

### Algorithm 1 Kalman filter on the joint estimate

---

1:  $\mathbf{s}^{(k)} = [\mathbf{x}_0^\top \mathbf{v}_0^\top]^\top$  and  $\mathbf{P}^{(0)} = \mathbf{I}$  {Initialization}

2: **for**  $i = 1$  to  $N$  **do**

3:  $z_i^{(k)} \leftarrow P_{R,i}^{(k)}$  {Local sensing}

4:  $\hat{d}_i^{(k)} \leftarrow d_0 10^{\left(\frac{P_0 - z_i^{(k)}}{10 n_p}\right)}$  {Distance estimate}

5: **end for**

6: Get distributed joint position estimate

$$\hat{\mathbf{x}}^{(k)} \leftarrow \text{consensusGaussNewton}(\tilde{\mathbf{s}}^{(k)}, \text{nodeCoordinates})$$

7: Kalman filtering step

$$\mathbf{P}^{(k+1)} \leftarrow \mathbf{F} \mathbf{\Sigma}^{(k)} \mathbf{F}^\top + \mathbf{W} \mathbf{W}^\top$$

$$\mathbf{E}^{(k+1)} \leftarrow \mathbf{G} \mathbf{P}^{(k+1)} \mathbf{G}^\top + \sigma_n^2$$

$$\mathbf{s}^{(k+1)} \leftarrow \mathbf{F} \mathbf{s}^{(k)} + \mathbf{P}^{(k+1)} \mathbf{G}^\top \mathbf{E}^{(k+1)-1} (\hat{\mathbf{x}}^{(k)} - \mathbf{G} \mathbf{F} \mathbf{s}^{(k)})$$

$$\mathbf{\Sigma}^{(k+1)} \leftarrow \mathbf{P}^{(k+1)} - \mathbf{P}^{(k+1)} \mathbf{G}^\top \mathbf{E}^{(k+1)-1} \mathbf{G} \mathbf{P}^{(k+1)}$$

8:  $k \leftarrow k + 1$

---

Filter (EKF) which is based on a linear approximation of the nonlinear functions by their first order Taylor expansion. Another approach is the use of the Unscented Kalman Filter (UKF) proposed by Simon Julier et al. [17], [18]. The basic principle behind the unscented Kalman filter is to approximate the nonlinear functions that govern the evolution of the state variable and measurement equation by a statistically linearized version of them using a set of sigma points. Sigma points are drawn deterministically from the covariance matrix of the state variable and propagated through the nonlinear state/measurement functions. There exist a family of the so-called Sigma-point Kalman Filters (SPKF) [19] and the UKF is a particular case of them. In this contribution we only consider the UKF as a special form of a SPKF. We will use a distributed version of the UKF based on consensus in a similar way as the one proposed in [20] with a consensus step on estimates.

3) *Local distance tracking and joint location (DT-GN)*: So far we have considered the approach of tracking the state variable consisting on the position and velocity of the target. Given the particular characteristics of the problem at hand it could also be interesting considering the problem of distance tracking and joint positioning. Nodes can use their received signal in order to track distance variations and then use these smoother distance estimates in order to perform joint localization. This means that we will avoid the interchange of local information for the tracking part (data aggregation) and hence, the only communication exchange will happen at the consensus step for getting the position estimate of the target. Once we have obtained a joint estimate of the target location we will use the joint estimate to perform an additional correction step of the tracking variable.

Our state variable is now going to be the squared distance to the target and the rate of variation of the squared distance. We can then model the state evolution as in [21]

$$\begin{aligned} r^{(k+1)} &= r^{(k)} + 2 \rho^{(k)} \Delta t \\ \rho^{(k)} &= \rho^{(k-1)} + \delta^{(k)} \Delta t \end{aligned} \quad (12)$$

We then have that our new state variable at the  $i$ -th node is given by  $\tilde{\mathbf{s}}^{(k)} = [r_i^{(k)}, \rho_i^{(k)}]^\top$ , where  $r_i^{(k)}$  and  $\rho_i^{(k)}$  are squared distance and the rate of variation of the squared distance, respectively at time instant  $k$  and for the  $i$ -th node. With this formulation of the problem,

---

**Algorithm 2** Distance Tracking and joint localization using consensus Gauss-Newton
 

---

- 1:  $\tilde{\mathbf{s}}_i^{(0)} = [r_i^{(0)}, \rho_i^{(0)}]^\top$   $\mathbf{P}_{\tilde{\mathbf{s}}_i}^{(0)} = \mathbf{P}_{\tilde{\mathbf{s}}_i}^{(0)}$  {Initialization}
  - 2: **while** new data exists **do**
  - 3:   Local UKF on  $\tilde{\mathbf{s}}_i^{(k)}$
  - 4:   Joint localization using Consensus Gauss-Newton  
 $\hat{\mathbf{x}}^{(k)} \leftarrow \text{consensusGaussNewton}(\tilde{\mathbf{s}}^{(k)}, \text{nodeCoordinates})$
  - 5:   Correction step using LMS  
 $d_i^{(k)} \leftarrow (\hat{\mathbf{x}}^{(k)} - \mathbf{x}_i)^\top (\hat{\mathbf{x}}^{(k)} - \mathbf{x}_i)$   
 $e_i^{(k)} \leftarrow r_i^{(k)} - d_i^{(k)}$   
 $r_i^{(k)} \leftarrow r_i^{(k)} + \mu e_i^{(k)} d_i^{(k)}$
  - 6: **end while**
- 

the measured signal is related to the state variable by

$$\tilde{z}_i^{(k)} = P_0 - 10 n_p \log \left( \frac{\sqrt{r_i^{(k)}}}{d_0} \right) + n_i^{(k)}, i = 1, \dots, N \quad (13)$$

Tracking of the state variable can be done independently on each node without the need of any information exchange among nodes. Since the measurement equation follows a nonlinear relationship with the state variable, we can use the unscented Kalman filter for tracking but other approaches are also possible. Once we have the smoothed estimates we perform a nonlinear least-squares estimation of the true target position using the consensus-based Gauss-Newton algorithm. In order to improve the performance of the tracking algorithm we then apply an additional correction step using the joint position estimate. This correction step can be done using the Least Mean Squares (LMS) algorithm. It is important to mention that a proper choice of the noise process statistics will be crucial in the performance of the tracking algorithm. We must allow a generous value for the noise process variance in order to be capable of tracking distance variations.

### III. SIMULATION RESULTS

Simulations allowed for the four aforementioned tracking techniques: C-F, regular Kalman filter (KF), Unscented Kalman filter (UKF) and local Distance Tracking and joint location with Gauss-Newton (DT-GN). Every technique depends on three variables: inter-transmission period, GPS duty cycle and transmission radius.

Kalman-based techniques perform outstandingly compared to the C-F technique. The latter is computationally cheaper, but its localization error can be up to one order of magnitude higher than the worst Kalman-based option. Fig. 5 supports the aforementioned sentence by plotting the error range for a GPS duty cycle of 20% (10 min) across every transmission radius and inter-transmission period.

As a general trend, UKF usually performs better than any of the other three. However, such a difference is low if compared with KF and DT-GN and much greater if compared with C-F. DT-GN may outperform KF if GPS activations do not become very scarce, but still the inter-transmission period can play a relevant role if it becomes longer, then KF performance will be affected more than DT-GN. For such a case, the former will only perform better if GPS activations take place longer than approximately 18 min. As an example, one can look at Fig. 6 which shows all the simulations run for 40 m of transmission radius. Other transmission radii may impact on the tracking performance as Figs. 8 and 7 show. However,

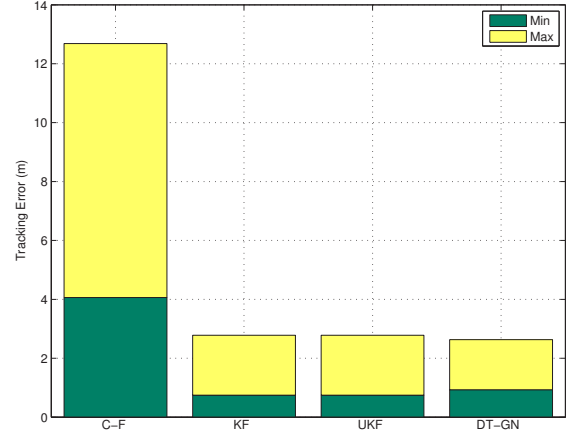


Fig. 5. Tracking Error range for 20% GPS duty cycle

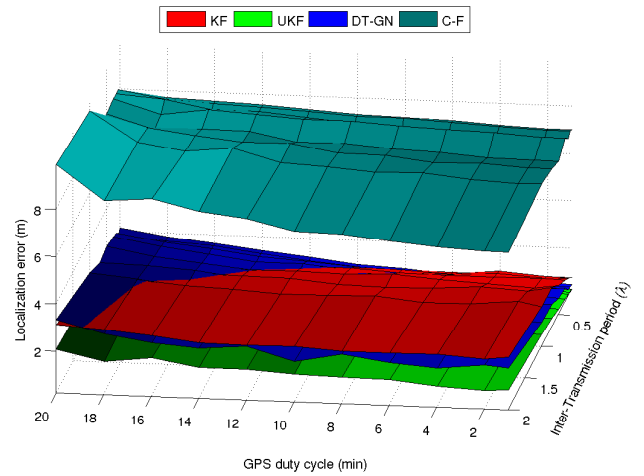


Fig. 6. Tracking Error for 40 m of transmission radius

the radius influence is much more noticeable in the C-F case. The aforementioned figures help us understand how the tracking error depends on the GPS duty cycle and Inter-Transmission period for a given radius. On the other hand, Fig. 9 shows a given inter-transmission period ( $\lambda = 0.7$  s) across the radius set and GPS considered values.

From the techniques on study, two are more feasible to be implemented in a real deployment: C-F due to its low cost and hardware availability and DT-GN since it is distributed and requires a lower amount of message interchange if compared with the other two Kalman-based proposed solutions. Any Kalman-based technique proposed in the current paper is distributed and therefore can operate even if the WSN infrastructure fails – i.e. the fusion center is unreachable due to any reason. However, we find an advantage in DT-GN over the other two due to the aforementioned reason – a lower amount of message interchange. Fig. 10 focuses on C-F and DT-GN and shows the error Cumulative Density Function (CDF) dependency on the transmission radius, for  $\lambda = 1$  sec and 11% GPS duty cycle. As it can be observed, transmission radius impacts dramatically on the error distribution for C-F, but the difference turns out to be negligible

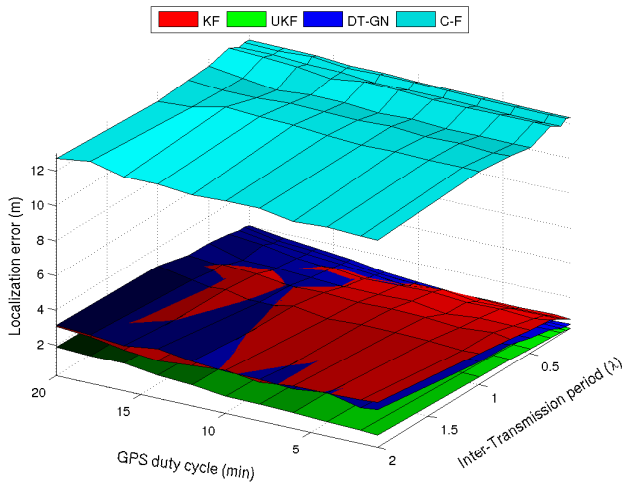


Fig. 7. Tracking Error for 50 m of transmission radius

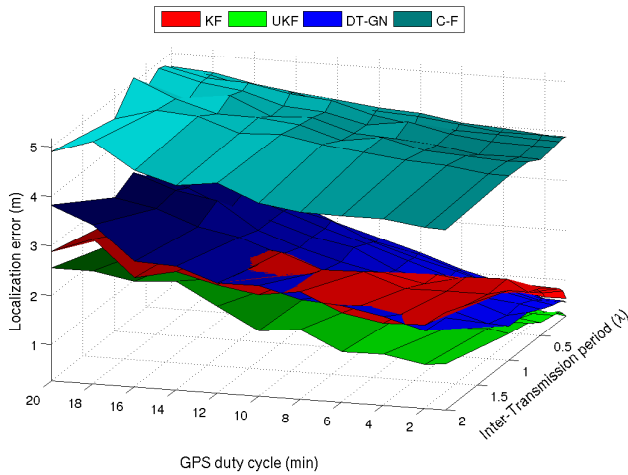


Fig. 8. Tracking Error for 20 m of transmission radius

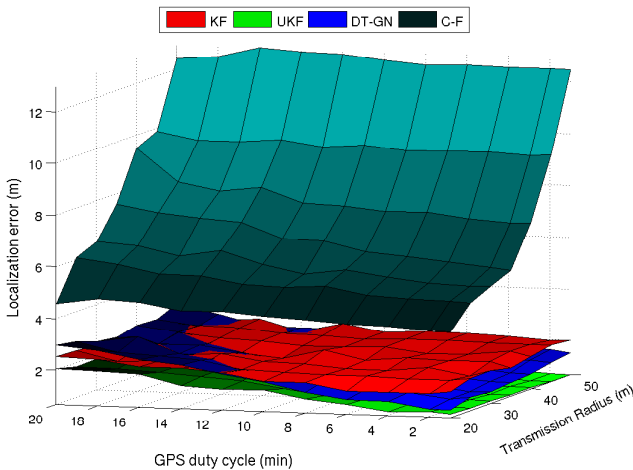


Fig. 9. Tracking Error for  $\lambda = 1$  s

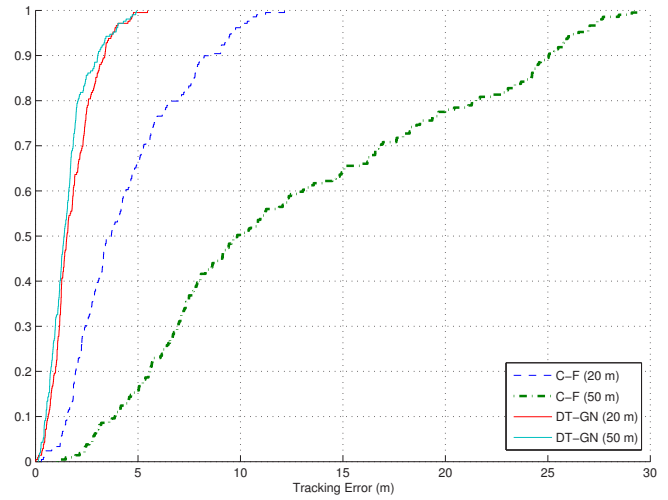


Fig. 10. Error CDF for C-F, DT-GN and radii 20 m and 50 m

for DT-GN.

#### IV. TESTBED MEASUREMENTS

##### A. Hardware description

iMote2 [12] was the hardware platform chosen to perform real tests based on the previous sections. Its core is a Marvell XScale microprocessor PXA271 which has 32MB of SDRAM and 256kB SRAM and its radio chip is a Texas Instruments CC2420 [22]. PXA271 supports a wide range of operating frequencies (13-416 MHz), however we selected 13MHz because it has been widely tested for our operating system. Our application never required the complete processor memory map, as a consequence it was configured just for the SRAM and SDRAM was never activated. Radio transmission power was always set to 0 dBm, though lower values were possible. On the software side, test applications were developed for TinyOS 2.1.1, which was the latest release as of 2010.

In order to perform our test, two changes were done on the overall aforementioned system setup. On one hand, the surface mounted antenna was substituted by an external monopole connected over an SMA connector soldered on the board. At the same time, the former antenna was disconnected by means of switching the capacitor from C20 to C5 as shown in Fig.11. On the other hand, a different XScale compiler from the default in TinyOS was used to obtain better executable code and avoid third-party conflicts as detailed in [23]. The compiler used was GCC v4.3.3 in eabi mode suitable for TinyOS.

1) *Propagation model fit*: An individual propagation model (per mote) from the formula

$$P = P_0 - 10n_p \log_{10} \left( \frac{d}{d_0} \right) \quad (14)$$

was adjusted from an outdoor calibration process where anchor nodes were taken four by four plus a target node which was broadcasting beacons that were received by the anchors. Every anchor collected samples on nine different spots distant from the broadcasting node 1 to 9 meters. For every spot, the broadcasting node transmitted oriented towards four different sides and from every orientation 1000 beacons were broadcasted. This way, a calibration process comprising a group of 4 anchor nodes consisted on 144,000 beacons.

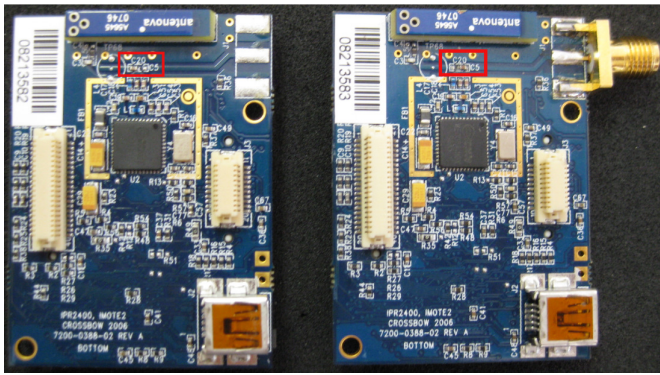


Fig. 11. Changes on iMote2 board

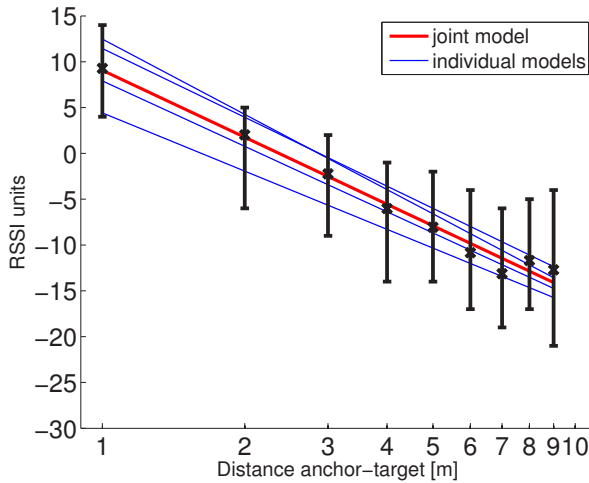


Fig. 12. Fitted Propagation Model

Once all the samples were collected, the least-square-fit method was utilized to calculate the individual propagation model parameters ( $P_0$  and  $n_p$ ) considering  $d_0 = 1\text{m}$  and the fitting variable as:

$$x = -10\log_{10}(d) \quad (15)$$

Fitting the propagation model as described, produced a model as shown in Fig.12. In order to obtain better result the individual propagation model (blue) was chosen rather than the joint propagation model (red).

### B. Real Performance

The test-bed deployment is shown in Fig.13 with 6 anchor nodes on the spots therein referred and a base station (not shown) connected to a laptop. Several movement patterns over the same trajectory (see Fig.14) were tested. The target node broadcasted three beacons per second and 15 control points (marked as squares and triangles) were set to compare the tracking outcomes and the real trajectory.

Results show the tracking obtained from running the UKF method already described on the iMote2 motes. The trajectory plotted is 24.13 m long and speed was variable with a mean of 0.1589 m/s. As one can observe, our results are very promising as the mean error is as less as 0.668 m and the mean square error is 0.812 m<sup>2</sup>.

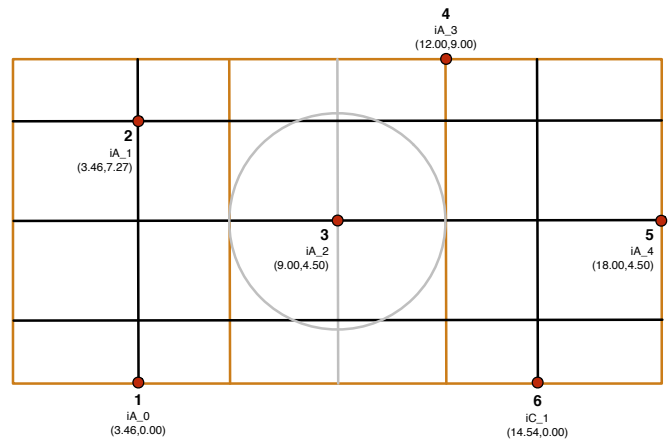


Fig. 13. Deployment Scenario for Testbed Measurements

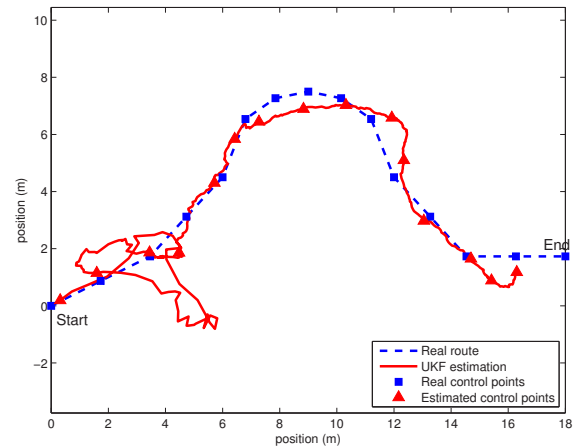


Fig. 14. Tracking in Testbed with UKF

## V. CONCLUSIONS

In this paper we have evaluated different approaches for the animal tracking application where an asynchronous dynamic and heterogeneous network has been analyzed. Several distributed Kalman-based filters have been implemented and their performances are compared according to the GPS activation period, the transmission coverage area and the type of animal movement. We may conclude that local Distance Tracking and joint localization using Consensus which solves a non-linear LS problem using a well known optimization method, namely Gauss-Newton, is the most appropriate technique as a trade-off between performance and complexity in terms of computational operations and amount of interchanged messages for the Consensus procedure. On the other hand, the connectivity-based localization performs with errors up to one order of magnitude over the Kalman-based ones, therefore it is just suitable in environments in which either rough localization is required (rather than tracking) or cheaper terminals must be deployed. Very valuable is also the confirmation that these approaches have been tested in real commercial HW providing many practical considerations on calibration and implementation issues.

## REFERENCES

- [1] M. Taillade, "Trends in satellite-based animal tracking," in *Biotelemetry XII*, P. Macini, S. Fioretti, C. Cristallini, and R. Bendini, Eds. Pisa, Italy: Litografia Felici, 1993, pp. 291–297.
- [2] F. Papi, P. Luschi, R. Crosio, and G. R. Hughes, "Satellite tracking experiments on the navigational ability and migratory behaviour of the loggerhead turtle *Caretta caretta*," *Marine Biology*, vol. 129, no. 2, pp. 215–220, 1997.
- [3] M. C. Perry, D. M. Kidwell, A. M. Wells-Berlin, E. J. R. Lohnes, G. H. Olsen, and P. C. Osenton, "Habitats used by black and surf scoters in eastern north american as determined by satellite radio telemetry," in *Second North American Sea Duck Conference*. Laurel, MA: Patuxent Wildlife Research Center, 2005, pp. 21–35.
- [4] L. D. Rosa, E. R. Secchi, Y. G. Maia, A. N. Zerbini, and M. Heide-Jorgensen, "Movements of satellite-monitored humpback whales on their feeding ground along the antarctic peninsula," *Polar Biology*, vol. 31, no. 7, pp. 771–781, 2008.
- [5] R. Vijay, A. Kansal, J. Friedman, and S. Mani, "Design consideration for solar energy harvesting wireless embedded systems," in *Proceedings of the IEEE International Conference on Information Processing and Sensor Networks*. Piscataway, NJ: IEEE Press, 2005, pp. 457–462.
- [6] H. Ritter, J. Schiller, T. Voigt, A. Dunkels, and J. Alonso, "Experimental evaluation of lifetime bounds for wireless sensor networks," in *Proceedings of the Second European Workshop on Wireless Sensor Networks*. Piscataway, NJ: IEEE Press, 2005, pp. 25–32.
- [7] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi, "Hardware design experiences in zebranet," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems*. New York, NY: ACM Press, 2004, pp. 227–238.
- [8] "Turtlenet project website." [Online]. Available: <http://prisms.cs.umass.edu/dome/turtlenet>
- [9] A. Gutiérrez, C. González, J. Jiménez-Leube, S. Zazo, N. Dopico, and I. Raos, "A heterogeneous wireless identification network for the localization of animals based on stochastic movements," *Sensors*, vol. 9, no. 5, pp. 3942–3957, 2009.
- [10] B. Bejar, P. Belanovic, and S. Zazo, "Distributed gauss-newton method for localization in ad-hoc networks," in *44th Asilomar Conference on Signals, Systems and Computers, Proceedings of*, Nov. 2010.
- [11] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1520 – 1533, 2004.
- [12] L. Nachman, J. Huang, J. Shahabdeen, R. Adler, and R. Kling, "IMOTE2: Serious computation at the edge," in *Wireless Communications and Mobile Computing Conference, 2008. IWCMC '08. International*, 2008, pp. 1118–1123.
- [13] "N4C project. Networking for Communications Challenged Communities - FP7-ICT-223994-N4C." <http://www.n4c.eu/>, Jun. 2010. [Online]. Available: <http://www.n4c.eu/>
- [14] N. Patwari, J. Ash, S. Kyperountas, I. Hero, A.O., R. Moses, and N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 54 – 69, 2005.
- [15] V. Savic, A. Poblacion, S. Zazo, and M. Garcia, "Indoor positioning using nonparametric belief propagation based on spanning trees," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, p. 12, Aug. 2010. [Online]. Available: <http://www.hindawi.com/journals/wcn/2010/963576.cta.html>
- [16] G. B. Greg Welch, "An introduction to the kalman filter," University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175, Tech. Rep. TR 95-041, July 2006.
- [17] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *American Control Conference, 1995. Proceedings of the*, vol. 3, Jun. 1995, pp. 1628 –1632 vol.3.
- [18] S. Julier and J. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," in *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*, 1997.
- [19] R. van der Merwe, "Sigma-Point kalman filters for probabilistic inference in dynamic State-Space models," PhD Thesis, Oregon Health & Science University, Portland, OR, USA, Apr. 2004.
- [20] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Decision and Control, 46th IEEE Conference on*, 2007, pp. 5492 –5498.
- [21] B. Bejar, N. I. Dopico, P. Belanovic, and S. Zazo, "Practical Tracking Strategies in Wireless Sensor Networks," in *Submitted*, 2011.
- [22] Texas Instruments, Inc, "CC2420 data sheet, rev. b," Mar. 2007. [Online]. Available: <http://focus.ti.com/docs/prod/folders/print/cc2420.html>
- [23] N. I. Dopico, "Compiling for an ARM architecture: iMote2-TinyOS," Universidad Politecnica de Madrid (UPM), Madrid, Spain, Technical TR-UPM-GAPS-10-01, Nov. 2010. [Online]. Available: <http://oa.upm.es/3837/>