RAO-BLACKWELLIZED VARIABLE RATE PARTICLE FILTERING FOR HANDSET TRACKING IN COMMUNICATION AND SENSOR NETWORKS

Carles Fernández–Prades, Pau Closas**, Juan A. Fernández–Rubio***

* Centre Tecnològic de Telecomunicacions de Catalunya (CTTC)
Av. Canal Olímpic s/n, 08860 Castelldefels (Barcelona), Spain.
**Universitat Politènica de Catalunya (UPC). Dept. of Signal Theory and Communications
Campus Nord UPC, Edifici D5, 08034 Barcelona, Spain.
e-mail: carles.fernandez@cttc.cat, {closas, juan}@gps.tsc.upc.edu

ABSTRACT

Knowledge about the position of a user's mobile handset device constitutes a valuable information for emerging location–based services and applications. While satellite–based navigation systems are the chosen technology for outdoor, rural and semi–rural environments, location in urban canyon or indoor navigation is still unsolved. In this paper, we exploit the fact that handsets are usually surrounded by different kinds of communication and sensor networks, which can be used to enhance the coverage, accuracy and robustness of satellite–based systems. After outlining models for motion, measurements and positioning, we depict a common scenario and propose a methodology for handset tracking. We show the suitability of Particle Filtering to the problem at hand by the appliance of a modified version of a recently proposed algorithm, the Variable Rate Particle Filtering (VRPF). The novelty relies on the inclusion of a Rao–Blackwellization procedure that significantly reduces the computational load. Details about the implementation and some significant numerical results of computer simulations are also provided.

1. INTRODUCTION

Information about the position of a user's mobile handset device is a potential trigger for a myriad of emerging applications. From fireman operations to interactive, personalized touristic guides, position constitutes a valuable data to be exploited by systems which only imagination can bound. Unfortunately, the problem of user's position remains unsolved, at least with the degree of coverage, reliability and accuracy that applications –and imagination– demand. The outstanding approach to mass-market positioning is usually referred to as Global Navigation Satellite Systems (GNSS), a concept which includes the well-known GPS, GLONASS or the forthcoming Galileo system. These satellite–based systems, often assisted by some kind of aiding system providing local information (WAAS, EGNOS, RTK), are able to determine the user's position with a high degree of accuracy under proper conditions. However, the performance of those systems severely degrades in common scenarios such as the urban canyon or indoor navigation, where there is no line–of–sight between the mobile device and the satellites. In these cases, the weak receiving power, the multipath effect and interferences make a stand–alone GNSS receiver useless.

Our everyday's life is surrounded by a continuous set of radiating sources coming from everywhere. The rapid deployment and pervasiveness of wireless communication systems (GSM, UMTS, WiFi, WiMAX, Bluetooth, UWB, TDT, DAB) and sensor networks, in combination with the availability of flexible, low–cost commercial off–the–shelf components, offers a bundle of electromagnetic waveforms that can be exploited in order to compute user's position. Although these systems are primary designed for other applications, they can provide worthy information for positioning. For instance, the fine time resolution of UWB signals makes them quite promising for indoor, high–resolution ranging [1]. In addition, the mobile device could include a low–cost inertial measurement unit (for instance, accelerometers based on MEMS technology, in the fashion of Nintendo's Wii console) which provide valuable complementary information [2].

In this paper, we address the problem of exploiting available data coming from heterogeneous systems, with the purpose of locating and tracking a mobile target. As shown hereinafter, equations involved in positioning are nonlinear. Sequential Monte Carlo (SMC) methods [3], also known as Particle Filters, constitute a clear trend followed by the signal processing community for dealing with nonlinearities while adding information (in a Bayesian sense) to the set of observation equations, and thus it will be applied to the problem at hand.

2. GENERAL SYSTEM MODEL

2.1 Motion equations

For the target motion, we will follow the approach taken in [4] considering a model which is linear in the state dynamics and nonlinear in the measurements. States can be linearly expressed as

$$\underbrace{\begin{pmatrix}\mathbf{x}_{t+1}^{NL}\\ \mathbf{x}_{t+1}^{LIN}\end{pmatrix}}_{\mathbf{x}_{t+1}} = \underbrace{\begin{pmatrix}\mathbf{I} & \mathbf{A}^{NL}\\ \mathbf{0} & \mathbf{A}^{LIN}\end{pmatrix}}_{\mathbf{A}}\underbrace{\begin{pmatrix}\mathbf{x}_{t}^{NL}\\ \mathbf{x}_{t}^{LIN}\end{pmatrix}}_{\mathbf{x}_{t}} + \underbrace{\begin{pmatrix}\mathbf{B}^{NL}\\ \mathbf{B}^{LIN}\end{pmatrix}}_{\mathbf{B}}\mathbf{u}_t + \underbrace{\begin{pmatrix}\mathbf{C}^{NL}\\ \mathbf{C}^{LIN}\end{pmatrix}}_{\mathbf{C}}\mathbf{f}_t \quad (1)$$

where the subindex t refers to time, $\mathbf{x}_t \in \mathbb{R}^{N_x \times 1}$ is the state vector (containing the position of the target $\mathbf{r}_t$ and possibly other motion parameters, such as velocity, acceleration or heading), $\mathbf{u}_t \in \mathbb{R}^{N_u \times 1}$ is a vector containing the inputs taken by the target (for instance the acceleration $\mathbf{a}_t$ provided by an inertial measurement unit) and $\mathbf{f}_t \in \mathbb{R}^{N_f \times 1}$ stands for unmeasured forces. The transition matrix $\mathbf{A} \in \mathbb{R}^{N_x \times N_x}$ relates the previous state $\mathbf{x}_t$ to the updated state $\mathbf{x}_{t+1}$. In a similar way, matrices $\mathbf{B} \in \mathbb{R}^{N_x \times N_u}$ and $\mathbf{C} \in \mathbb{R}^{N_x \times N_f}$ relate the measured inputs and the unmeasured forces to $\mathbf{x}_{t+1}$, respectively. We have assumed a known probability density $p_{\mathbf{x}_0}$ (not necessarily Gaussian) and a Gaussian distribution $p_{\mathbf{f}_t}$. A general way to express the state evolution density is $\mathbf{x}_{t+1} \sim f(\mathbf{x}_{t+1}|\mathbf{x}_t)$, where $\sim$ denotes that the variable on the left is drawn independently from the probability density function on the right.

Observe that we have splitted these terms into a linear (superscript *LIN*) and a nonlinear (superscript *NL*) part. This is because the optimal solution for linear state space models and Gaussian noise is well–known – the Kalman filter. Indeed, we will solve separately the linear part (details about the Kalman filtering will be given in section 4) and the nonlinear part (by means of a Particle Filtering method). This well-known strategy, which saves an important amount of computational effort, is usually referred to as Rao-Blackwellization in Particle Filtering literature. Its applicability has also been shown for approaches with more relaxed statistical

This work has been partially supported by the Catalan, Spanish and French Governments: 2006ITT-10003, FIT-330503-2006-2, SGR2005-00690, TEC2006-06481/TCM, TEC2004-04526, TIC2003-05482 and 2005SGR-00639.

assumptions such as the Cost–Reference Particle Filtering without significative loss of performance [5].

2.2 Measurement equations

Modern handset devices are equipped with a set of wireless system interfaces. Measurements taken from these systems (mainly time delay or receiving power strength) are nonlinearly related to the device position. A general model could be written as

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{e}_t \tag{2}$$

where $h(\cdot)$ is a possibly nonlinear function relating the state to the measurements $\mathbf{y}_t \in \mathbb{R}^{N_y \times 1}$, and $\mathbf{e}_t \in \mathbb{R}^{N_y \times 1}$ is the measurement error, with known probability density $p_{\mathbf{e}_t}$ and not necessarily Gaussian. When the measurement is time delay, it can be converted to range multiplying by the speed of light c, but taking into account the desynchronization between emitter and receiver (time stamp of measures has been omitted for clarity):

$$y = \underbrace{c\,(t_{Rx} - t_{Tx})}_{d} + c(\Delta t_{Rx} - \Delta t_{Tx}) + e \tag{3}$$

where $d = \sqrt{(r_{Tx\_x} - r_{Rx\_x})^2 + (r_{Tx\_y} - r_{Rx\_y})^2 + (r_{Tx\_z} - r_{Rx\_z})^2}$ is the geometric distance, and Δt refers to the device's clock bias with respect to an agreed time framework. In case of power aware sensors, we will assume an stochastic model for the strength loss:

$$y = P_{Tx} - \bar{P}_L(d_0) - 10n\log_{10}\left(\frac{d}{d_0}\right) + \varepsilon, \tag{4}$$

where P_{Tx} is the transmitted power, d_0 is the reference distance, $\bar{P}_L(d_0)$ is the mean loss at d_0, n is the slope of the loss (depending on the scenario) and ε is a random variable with a log–normal distribution.

2.3 Positioning equations

The equations relating measurements to target's position depend on whether the time of transmission t_{Tx} is known or unknown. When the system consists of several beacons emitting from known locations at known instants (*i.e.* a synchronized beacon network), we speak about spherical positioning. The estimated travel time or receiving power is converted to range. Each range measurement defines a sphere centered at the beacon, on which the receiver must lie. The intersection of several spheres (corresponding to several beacons) defines the position of the receiver.

If t_{Tx} is not known, we can measure differences in travel times and convert them into differences in ranges to the beacons, in order to cancel the unknown. The locus of all points at the same distance of two given points in a 3D space is a hyperboloid, which is defined by a range difference. Again, the intersection of several hyperboloids indicates the receiver's position. A review on the location equations can be found in [6].

Since both sphere and hyperboloid equations are nonlinear, the traditional approach [2] consists on a linearization of the positioning model in the vicinity of a point of interest. Then, we can add *aprioristic* information to the model, extend it to other parameters of interest (velocity, acceleration an so on) and optimize it via the Kalman filter, assuming Gaussianity for the error terms.

The fact that Particle Filtering methods are able to deal with nonlinear optimization and more relaxed statistical assumptions suggests that the optimization can be performed directly over the parameters of interest (that is, the target position) instead of over the parameters of the linearized model (that is, propagation time, signal strength or Doppler shift). This new concept has been shown in [7], obtaining remarkable improvements in the overall performance, and it will be the approach taken in this paper. It also has the benefit of allowing the inclusion of the motion models in a more natural way, since all the equations are referred to position parameters, in contrast to including prior information in the parameters, which can be troublesome in many applications. In communication systems' terminology, this strategy can be regarded as a cross-layer optimization.

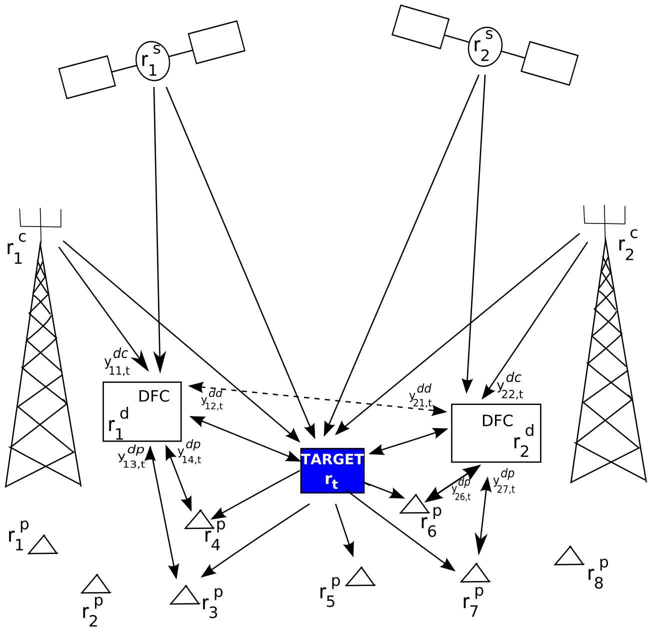

Figure 1: Example of a generic system architecture with $N_s = 2$, $N_c = 2$, $N_d = 2$ and $N_p = 8$. Notation of positions and measurements is also depicted.

2.4 Multiple target tracking

The extension to multiple targets can be achieved by expanding the motion model (properly stacking the new equations in model (1)) and applying some mechanism in order to efficiently cope with an unknown number of tracks, initiate and terminate tracks, switch between two near tracks and to give some robustness against low detection and high false alarm probabilities (typical features of a sensor network). In that sense, see [8, 9] and references herein for some suitable algorithms. Since the focus of this paper is on the methodology for the location and tracking problems, we will restrict ourselves to a single target.

3. PROBLEM STATEMENT

We assume a moving target equipped with a GNSS receiver, some kind of mobile communication system (GSM, 3G), a set of embedded accelerometers and the capability of being detected by a power–aware sensor network.

A total number of N_p power–aware sensors are located at fixed unknown positions $\mathbf{r}^p_{1:N_p} = \{\mathbf{r}^p_1, \ldots, \mathbf{r}^p_{N_p}\}$ with an arbitrary topology. The n–th sensor takes a measure $y_{n,t} = h(d_{n,t}, \varepsilon)$ at time t, where $d_{n,t}$ is the distance between the target and the sensor and ε is a random perturbation with known p_ε. This measure is broadcasted by the sensor if the distance is under a certain threshold and otherwise remains silent. We assume that sensors are not synchronized.

There are also N_c mobile communication base–stations at fixed unknown positions $\mathbf{r}^c_{1:N_c} = \{\mathbf{r}^c_1, \ldots, \mathbf{r}^c_{N_c}\}$ and N_s GNSS satellites (with known positions, since it is broadcasted in their navigation message) at $\mathbf{r}^s_{1:N_s} = \{\mathbf{r}^s_1, \ldots, \mathbf{r}^s_{N_s}\}$.

We will also assume a set of N_d control nodes in the network (called Data Fusion Centers, DFCs), with unknown positions $\mathbf{r}^d_{1:N_d} = \{\mathbf{r}^d_1, \ldots, \mathbf{r}^d_{N_d}\}$, which receive the messages broadcasted by some of the sensors and are able to perform an estimation of the distance to that sensors. In addition, DFCs can also sense measurements of GNSS and communication systems. The main purpose of the paper is to jointly estimate the target track $\mathbf{r}_{0:t} = \{\mathbf{r}_0, ..., \mathbf{r}_t\}$, the sensor locations $\mathbf{r}^p_{1:N_s}$, the communication station locations $\mathbf{r}^c_{1:N_c}$ and the DFC positions $\mathbf{r}^d_{1:N_d}$ given

- Measurements taken by the target device (GNSS, mobile communication systems, inertial measurements), which are sent to one or more DFCs.
- Measurements taken by the DFCs (GNSS, mobile communication systems, distance to some sensors, information gathered by other DFCs).
- Sensor network broadcast to the DFCs.

4. PROPOSED SOLUTION

The proposed solution closely resembles the one proposed in [10] for a beacon–free sensor network. In our scenario, we assume a minimum of one node with known location to avoid the rotation ambiguity of a relative positioning. This can be translated into the assumption that at least one DFC has simultaneous lines–of–sight with at least four GNSS satellites, has decoded the position of such satellites and has computed its own position. Provided that most of the information is firstly gathered by the DFCs, it seems reasonable to perform computation there. In any case, secure communication between DFCs and the mobile handset is mandatory for privacy reasons.

4.1 Initialization: estimation of node locations

The first step in the determination of the target's position and subsequent tracking is the estimation of node locations (DFCs, mobile communications' base stations and sensors). The Maximum A Posteriori estimation of the DFC's positions can be expressed as

$$\hat{\mathbf{r}}^d_{1:N_d} = \arg\max_{\tilde{\mathbf{r}}^d_{1:N_d}} \left\{ \prod_{\substack{i=1 \\ i\neq l}}^{N_d} p\left(\mathbf{y}^{dd}_{il} \middle| \tilde{\mathbf{r}}^d_i, \tilde{\mathbf{r}}^d_l\right) p\left(\tilde{\mathbf{r}}^d_{1:N_d}\right) \right\} \tag{5}$$

where $\mathbf{y}^{dd}_{il}$ is a measurement obtained by the i–th DFC from a signal of the l–th DFC. Assuming a static network, problem (5) can be solved considering a narrow Gaussian distribution for those DFCs positioned by GNSS and a uniform distribution for the other ones. Following the approach taken in [10], we suggest to apply the Accelerated Random Search algorithm (proposed in [11] and detailed in Algorithm 1) due to its simplicity when dealing with high–dimensional problems. Once $\hat{\mathbf{r}}^d_{1:N_d}$ has been computed, the positions of the sensors and the base stations can be selected independently:

$$\hat{\mathbf{r}}^p_n = \arg\max_{\tilde{\mathbf{r}}^s_n} \left\{ \prod_{l=1}^{N_s} p\left(\mathbf{y}^{dp}_{ln} \middle| \tilde{\mathbf{r}}^d_n, \tilde{\mathbf{r}}^p_n\right) p\left(\tilde{\mathbf{r}}^p_n\right) \right\} \tag{6}$$

where $\mathbf{y}^{dp}_{ln}$ is a measurement obtained by the l–th DFC from a signal of the n–th sensor, and

$$\hat{\mathbf{r}}^c_m = \arg\max_{\tilde{\mathbf{r}}^c_m} \left\{ \prod_{l=1}^{N_c} p\left(\mathbf{y}^{dc}_{lm} \middle| \tilde{\mathbf{r}}^d_m, \tilde{\mathbf{r}}^c_m\right) p\left(\tilde{\mathbf{r}}^c_m\right) \right\} \tag{7}$$

where $\mathbf{y}^{dc}_{lm}$ is a measurement obtained by the l–th DFC from a signal of the m–th communication base station. Again, Algorithm 1 has been used for solving problems (6) and (7) considering uniform distributions for $p\left(\tilde{\mathbf{r}}^p_n\right)$ and $p\left(\tilde{\mathbf{r}}^c_m\right)$ and Gaussian distribution for the others. The requirement is that each DFC, sensor and base station should transmit at least one signal burst which can be collected by other DFCs in the network.

Algorithm 1 ACCELERATED RANDOM SEARCH

Require: A function to optimize $f(\mathbf{r}_{1:N})$, d_{min}, d_{max}, contraction factor c, number of iterations N_{iter}. $\mathscr{R}$ is the domain of $\mathbf{r}_{1:N}$ and $\mathscr{B}^{(i)} = \left\{ \tilde{\mathbf{r}}_{1:N} \in \mathscr{R} \,:\, \left\| \tilde{\mathbf{r}}_{1:N} - \mathbf{r}^{(i)}_{1:N} \right\|_2 < d^{(i)} \right\}$

Ensure: $\hat{\mathbf{r}}_{1:N} = \arg\max_{\mathbf{r}_{1:N}\in\mathscr{R}} f(\mathbf{r}_{1:N})$.

Initialize $d^{(1)}$, $\mathbf{r}^{(1)}_{1:N}$
for $i = 1$ to $N_{iter} - 1$ **do**
  Draw $\tilde{\mathbf{r}}_{1:N} \sim U(\mathscr{B}^{(i)})$
  if $f(\tilde{\mathbf{r}}_{1:N}) > f\left(\mathbf{r}^{(i)}_{1:N}\right)$ **then**
    $\mathbf{r}^{(i+1)}_{1:N} = \tilde{\mathbf{r}}_{1:N}$, and $d^{(i+1)} = d_{max}$
  else
    $\mathbf{r}^{(i+1)}_{1:N} = \mathbf{r}^{(i)}_{1:N}$, and $d^{(i+1)} = \frac{d^{(i)}}{c}$
  end if
  if $d^{(i+1)} < d_{min}$ **then**
    $d^{(i+1)} = d_{max}$
  end if
end for
$\hat{\mathbf{r}}_{1:N} = \mathbf{r}^{(N_{iter})}_{1:N}$

4.2 Target tracking

The data fusion algorithm must face a problem in which the observations coming from different systems arrive at different (and probably incommensurate) rates. In addition, it would be desirable to parameterize rapid manoeuvres or nearly–Brownian motions with closely spaced states and smooth, straight trajectories with few state points. The intuitive idea behind this approach is to track more finely the sharp movements (for instance, someone moving inside a building) and perform more spaced updates when the movement is smooth and predictable (someone traveling by bus), by means of a suitable choice for the dynamical model. In order to cope with that, we propose an algorithm inspired in the Variable Rate Particle Filtering (VRPF) presented in [12, 13]. Unlike standard space–state modeling, the VRPF approach does not model the states at the same rate than the observations, and thus it is not necessary to update the target state every time we receive a new observation.

The variable rate state can be modeled by adding a random, continuous–value variable to the state model, expressing the time at which the target changes its state. Thus, the state space is expanded to $\theta_n = \{\mathbf{x}_n, \tau_n\}$, where $\tau_n \in \mathbb{R}^+ > \tau_{n-1}$ denotes the state change time. The observation samples $\mathbf{y}_{t|_{t=t_0+kT_s}}$ are assumed to be generated independently from a density function $g(\cdot)$, conditionally upon a neighborhood of close states $\theta_{\mathscr{N}_t} = \{\theta_n;\, n \in \mathscr{N}_t\}$. Thus, the likelihood model for the observations can be expressed as follows:

$$\mathbf{y}_t \sim g(\mathbf{y}_t|\theta_{0:\infty}) = g\left(\mathbf{y}_t|\{\mathbf{x}_n, \tau_n; n \in \mathscr{N}_t\}\right) = g\left(\mathbf{y}_t|s\left(\theta_{\mathscr{N}_t}\right)\right) \tag{8}$$

being $\mathscr{N}_t$ a local neighborhood of state indices that defines the dependence structure of the observations, and $s\left(\theta_{\mathscr{N}_t}\right)$ is a deterministic function that depends on the dynamical model.

Defining $\alpha_{far} = (t - \tau_{n-2})\,\mathbf{x}_{n-2} + (\tau_{n+1} - t)\,\mathbf{x}_{n+1}$ and $\alpha_{close} = (t - \tau_{n-1})\,\mathbf{x}_{n-1} + (\tau_n - t)\,\mathbf{x}_n$, a possible interpolation function can be put in the form

$$\hat{\mathbf{x}}_t = s(\theta_{\mathscr{N}_t}) = \frac{\beta\alpha_{far} + (1-\beta)\alpha_{close}}{\tau_{n+1} - \tau_{n-2}} \tag{9}$$

From equation (9) it seems clear that the neighborhood is obtained as $\mathscr{N}_t = \{n, n-1, n-2, n+1;\ \tau_{n-2} < \tau_{n-1} \le t < \tau_n < \tau_{n+1}\}$. Note that, as implicitly stated in equation (8), the addition of new state points beyond $\mathscr{N}_t$ does not change the neighborhood of $\theta_{0:t}$. The aim is to estimate recursively the sequence of variable rate state points as new measurements become available, that is, every T_s seconds. All the information concerning this sequence is included in its conditional probability distribution,

$$\frac{p\left(\theta_{0:\mathcal{N}_t^+}|\mathbf{y}_{0:t}\right)}{p\left(\theta_{0:\mathcal{N}_{t-1}^+}|\mathbf{y}_{0:t-1}\right)} = \frac{g\left(\mathbf{y}_t|\theta_{\mathcal{N}_t}\right) f\left(\theta_{\mathcal{N}_{t-1}^+ +1:\mathcal{N}_t^+}|\theta_{\mathcal{N}_{t-1}^+}\right)}{p\left(\mathbf{y}_t|\mathbf{y}_{0:t-1}\right)} \quad (10)$$

where $\mathcal{N}_t^+ = max(\mathcal{N}_t)$ refers to the member of $\mathcal{N}_t$ having the largest time index τ_n. Note that, if the neighborhood does not increase in a time step, $\mathcal{N}_t^+ = \mathcal{N}_{t-1}^+$ and thus $f\left(\theta_{\mathcal{N}_{t-1}^+ +1:\mathcal{N}_t^+}|\theta_{\mathcal{N}_{t-1}^+}\right) = 1$. In general, equation (10) is analytically intractable and we must resort to efficient numerical approximations such as particle filters. As aforementioned, we propose an algorithm with a variable state rate. The general methodology of VRPFs, their applicability to target tracking and implementation details can be found in [14]. Here we introduce a modified algorithm which follows the common structure of Monte Carlo approximations (prediction and update) for the nonlinear states and a Kalman filtering for the linear ones, according to a particular case of Rao–Blackwellization which is known as Mixture Kalman Filter [15]. A description of the algorithm is shown in Algorithm 2.

In brief, the M generated particles are replicated M_i times according to their weight, and then propagated forward with only one likelihood evaluation. Related to the implementation, it is worthwhile to mention that the prediction samples of the nonlinear part of the state vector can be drawn from a Gaussian distribution, $\mathbf{x}_{t+1}^{NL\,(i)(j)} \sim N(\mu^{(i)(j)},\Sigma)$ with $\mu^{(i)(j)} = \mathbf{x}_t^{NL\,(i)(j)} + \mathbf{A}^{NL}\mathbf{x}_{t|t-1}^{LIN\,(i)(j)} + \mathbf{B}^{NL}\mathbf{u}_t$ and $\Sigma = \mathbf{A}^{NL}\mathbf{P}_{t|t-1}^{LIN}\left(\mathbf{A}^{NL}\right)^T + \mathbf{C}^{NL}\mathbf{Q}_t\left(\mathbf{C}^{NL}\right)^T$, where $\mathbf{Q}_t$ is the covariance of $\mathbf{f}_t$ [4]. Then, the linear subset of unknowns can be solved by means of the Kalman filter with a remarkable saving of computational effort:

$$\begin{aligned}\hat{\mathbf{x}}_{t+1|t}^{LIN\,(i)(j)} &= \bar{\mathbf{A}}^{LIN}\left(\hat{\mathbf{x}}_{t|t-1}^{LIN\,(i)(j)} + \mathbf{K}_t\left(\mathbf{z}_t^{(i)(j)} - \mathbf{A}^{NL}\hat{\mathbf{x}}_{t|t-1}^{LIN\,(i)(j)}\right)\right) + \\ &+ \mathbf{B}^{LIN}\mathbf{u}_t + \mathbf{C}^{LIN}\left(\mathbf{C}^{NL}\right)^{\dagger}\mathbf{z}_t^{(i)(j)} \quad (11)\end{aligned}$$

where $\bar{\mathbf{A}}^{LIN} = \mathbf{A}^{LIN} - \mathbf{C}^{LIN}\left(\mathbf{C}^{NL}\right)^{\dagger}\mathbf{A}^{NL}$, $(\cdot)^{\dagger}$ denotes the Moore–Penrose pseudoinverse, $\mathbf{z}_t^{(i)(j)} = \hat{\mathbf{x}}_{t+1}^{NL\,(i)(j)} - \hat{\mathbf{x}}_t^{NL\,(i)(j)}$, and

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}^{LIN}\left(\mathbf{A}^{NL}\right)^T\left(\mathbf{A}^{NL}\mathbf{P}_{t|t-1}^{LIN}\left(\mathbf{A}^{NL}\right)^T + \mathbf{C}^{NL}\mathbf{Q}_t\left(\mathbf{C}^{NL}\right)^T\right)^{-1} \quad (12)$$

$$\mathbf{P}_{t|t-1}^{LIN} = \bar{\mathbf{A}}^{LIN}\left(\mathbf{P}_{t-1|t-2}^{LIN} - \mathbf{K}_{t-1}\mathbf{A}^{NL}\mathbf{P}_{t-1|t-2}^{LIN}\right)\left(\bar{\mathbf{A}}^{LIN}\right)^T. \quad (13)$$

The completedness test of steps 9 and 17 refers to whether the set of states such that $\tau_{n-2} < \tau_{n-1} \le t < \tau_n < \tau_{n+1}$ have been yet generated or not. The update of weights performed in step 27 is formally defined as

$$w_t^{(i)(j)} \propto \frac{\bar{w}_{t-1}^{(i)}}{\breve{w}_{t-1}^{(i)}} \frac{g\left(\mathbf{y}_t|\theta_{\mathcal{N}_t}^{(i)(j)}\right) f\left(\theta_{\mathcal{N}_{t-1}^+ +1:\mathcal{N}_t^+}^{(i)(j)}|\theta_{\mathcal{N}_{t-1}^+}^{(i)}\right)}{q\left(\theta_{\mathcal{N}_{t-1}^+ +1:\mathcal{N}_t^+}^{(i)(j)}|\theta_{0:\mathcal{N}_{t-1}^+}^{(i)},\mathbf{y}_{0:t}\right)} \quad (14)$$

where

$$q\left(\theta_{\mathcal{N}_{t-1}^+ +1:\mathcal{N}_t^+}^{(i)(j)}|\theta_{0:\mathcal{N}_{t-1}^+}^{(i)},\mathbf{y}_{0:t}\right) = \prod_{n=\mathcal{N}_{t-1}^+ +1}^{\mathcal{N}_t^+} q\left(\theta_n^{(i)(j)}|\theta_{0:n-1}^{(i)},\mathbf{y}_{0:t}\right). \quad (15)$$

A simple choice for this proposal function is $q\left(\theta_{\mathcal{N}_{t-1}^+ +1:\mathcal{N}_t^+}^{(i)(j)}|\theta_{0:\mathcal{N}_{t-1}^+}^{(i)},\mathbf{y}_{0:t}\right) = f\left(\theta_{\mathcal{N}_{t-1}^+ +1:\mathcal{N}_t^+}^{(i)(j)}|\theta_{\mathcal{N}_{t-1}^+}^{(i)}\right)$, leading to $w_t^{(i)(j)} \propto \frac{\bar{w}_{t-1}^{(i)}}{\breve{w}_{t-1}^{(i)}} g\left(\mathbf{y}_t|\theta_{\mathcal{N}_t}^{(i)(j)}\right)$.

Algorithm 2 RAO–BLACKWELLIZED VRPF

Require: $\mathbf{y}_{0:K}$, $\mathbf{u}_{0:K}$, $\hat{\mathbf{r}}_{1:N_d}^d$, $\hat{\mathbf{r}}_{1:N_p}^p$, $\hat{\mathbf{r}}_{1:N_c}^c$

Ensure: Target tracking $\hat{\mathbf{x}}_{1:K}$.

1: Initialize $\left\{\theta_0^{(i)}\right\}_{i=1}^M$, $\mathbf{P}_0 = diag\left(\sigma_{\mathbf{x}_0^{LIN}}^2\right)$, $\mathbf{K}_0 = \mathbf{1}_{4\times 2}$ and $\left\{\bar{w}_0^{(i)}\right\}_{i=1}^M = \frac{1}{M}$.
2: **for** $t = 1$ to K **do**
3: Compute the covariance matrix $\mathbf{P}_{t|t-1}$ as in (13).
4: Compute the Kalman gain matrix $\mathbf{K}_t$ as in (12).
5: Assign a selection weight $\breve{w}_{t-1}^{(i)} = \bar{w}_{t-1}^{(i)}$.
6: Compute $M_i = \max\left(1, \lfloor M\breve{w}_{t-1}^{(i)} \rfloor\right)$.
7: Select M_i replicates of particle i according to $\breve{w}_{t-1}^{(i)}$ $\rightarrow \theta_{0:\mathcal{N}_{t-1}^+}^{(i)(j)}$, where $i = 1,\ldots,M$ and $j = 1,\ldots,M_i$.
8: **for** $i = 1$ to M **do**
9: **if** $\mathcal{N}_t^{(i)}$ is complete **then**
10: Update weight: $w_t^{(i)(1)} \propto \frac{M_i \bar{w}_{t-1}^{(i)}}{\breve{w}_{t-1}^{(i)}} g(\mathbf{y}_t|\theta_{\mathcal{N}_t}^{(i)})$.
11: Reset $M_i = 1$.
12: Set $\theta_{0:\mathcal{N}_t^+}^{(i)(1)} = \theta_{0:\mathcal{N}_{t-1}^+}^{(i)}$.
13: **else**
14: **for** $j = 1$ to M_i **do**
15: Set $\theta_{0:\mathcal{N}_{t-1}^+}^{(i)(j)} = \theta_{0:\mathcal{N}_{t-1}^+}^{(i)}$.
16: Set $n = \mathcal{N}_{t-1}^{+\,(i)}$.
17: **while** $\mathcal{N}_t^{(i)(j)}$ is incomplete **do**
18: Set $n = n+1$.
19: $\tau_{n+1}^{(i)(j)} - \tau_n^{(i)(j)} \sim G(\gamma_\tau, \varphi_\tau)$.
20: Draw $\mathbf{x}_n^{NL\,(i)(j)} \sim N(\mu^{(i)(j)},\Sigma)$
21: Compute $\mathbf{x}_n^{LIN\,(i)(j)}$ as in (11).
22: Build $\theta_n^{(i)(j)}$.
23: **end while**
24: Set $\mathcal{N}_t^{+(i)(j)} = n$.
25: Append new states to particle: $\theta_{0:\mathcal{N}_t^+}^{(i)(j)} = \left(\theta_{0:\mathcal{N}_t^+}^{(i)}, \theta_{\mathcal{N}_{t-1}^+ +1:\mathcal{N}_t^+}^{(i)(j)}\right)$.
26: **end for**
27: For each replicate j update particle weights: $w_t^{(i)(j)} \propto \frac{\bar{w}_{t-1}^{(i)}}{\breve{w}_{t-1}^{(i)}} g\left(\mathbf{y}_t|\theta_{\mathcal{N}_t}^{(i)(j)}\right)$.
28: **end if**
29: Restack the particles and weights from the replicates $\rightarrow$ Particle $(i)(j)$ becomes particle $j + \sum_{i'<i} M_i$.
30: Renormalize weights such that $\sum_{i=1}^{\sum_i M_i} w_t^{(i)} = 1$ $\rightarrow \bar{w}_t^{(i)}$.
31: **end for**
32: $\hat{\mathbf{x}}_t = \sum_{i=1}^M \bar{w}_t^{(i)} s\left(\theta_{\mathcal{N}_t}^{(i)}\right)$.
33: **end for**

5. COMPUTER SIMULATIONS

In order to provide illustrative numerical results, we have particularized the system model to the following network deployment. Let us assume a 2D field $\mathscr{R}$, a square centered at $(0,0)$ with sides of 100 m length. On $\mathscr{R}$, there are $N_c = 4$ mobile communications' base stations, distributed as $\mathbf{r}_{1:4}^c \sim [\mathbb{R}\mathrm{e}\{\rho\};\mathbb{I}\mathrm{m}\{\rho\}]$, where ρ is complex-Gaussian distributed as $CN(30\times[0,-1,1+j,1-j]^T, 25\mathbf{I}_4)$. There also are a total number of $N_p = 20$ power-aware sensors, with the same measurement function that the DFCs (equation (4)). The chosen values have been $P_{Tx} = 0$ dBm, $d_0 = 1$ m, $\bar{P}_L(d_0) = 30$ dB,

$n = 3$ (a typical value for urban area cellular radio) and ε is a random variable with a log–normal distribution with zero mean and $\sigma = 3$ m. The n–th sensor only transmits its measurements to the DFCs if both the distance to the target and the distance to the DFC is $d < 30$ m. We have supposed a random, uniformly distributed sensor deployment over $\mathscr{R}$. In addition, there are $N_d = 4$ DFCs, two of them with direct line–of–sight with $N_s = 5$ GNSS satellites. The DFCs are distributed as $\mathbf{r}^d_{1:4} \sim [\mathbb{R}\mathrm{e}\{\rho\}; \mathbb{I}\mathrm{m}\{\rho\}]$, where $\rho \sim CN(25 \times [-1-j, -1+j, 1+j, 1-j]^T, 2\mathbf{I}_4)$. Results are averaged over $L = 100$ independent computer simulations, each one with a different network deployment and target trajectory. Parameters of Algorithm 1 have been set to $d_{min} = 10^{-4}$, $d_{max} = 100$, $c = 2$, $N_{iter} = 5000$ for $\hat{\mathbf{r}}^d_{1:Nd}$ and $N_{iter} = 1000$ for each $\hat{\mathbf{r}}^c$ and $\hat{\mathbf{r}}^p$. Results are shown in table 1, where the measure of performance is

$$RMSE = \sqrt{\frac{1}{LN}\sum_{l=1}^{L}\sum_{n=1}^{N}\left\|\mathbf{r}_n - \hat{\mathbf{r}}^l_n\right\|^2} \qquad (16)$$

	$\mathbf{r}^d$	$\mathbf{r}^c$	$\mathbf{r}^p$
RMSE (m)	1.32	2.46	3.16

Table 1: RMSE of the estimated node positions

Then, we have simulated 100 seconds of target movement following the motion model expressed in equation (1). We assume that the target is equipped with a unit of inertial measurements with some bias, in the form $\mathbf{a}_{true,t} = \mathbf{a}_t + \delta\mathbf{a}_t$. Since the position is extracted by dead–reckoning of $\mathbf{a}_t$, the presence of bias is critical and must be taken into account. This is the reason why the acceleration bias is included in the state vector and the measured acceleration in the input signal. Thus, general model (1) has been particularized to $\mathbf{x}^{NL}_t = \mathbf{r}_t$, $\mathbf{x}^{LIN}_t = \left[\mathbf{v}^T_t \;\; \delta\mathbf{a}^T_t\right]^T$, with $\mathbf{A}^{NL} = \begin{pmatrix} T_s\mathbf{I}_2 & \frac{T_s^2}{2}\mathbf{I}_2 \end{pmatrix}$, $\mathbf{A}^{LIN} = \begin{pmatrix} \mathbf{I}_2 & T_s\mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 \end{pmatrix}$, $\mathbf{u}_t = \mathbf{a}_t$, $\mathbf{B}^{NL} = \frac{T_s^2}{2}\mathbf{I}_2$, $\mathbf{B}^{LIN} = \begin{pmatrix} T_s\mathbf{I}_2 \\ \mathbf{0}_2 \end{pmatrix}$, $\mathbf{C}^{NL} = \frac{T_s^3}{6}\mathbf{I}_2$, and $\mathbf{C}^{LIN} = \begin{pmatrix} \frac{T_s^2}{2}\mathbf{I}_2 \\ T_s\mathbf{I}_2 \end{pmatrix}$. GNSS availability has not been considered for the handset. Other parameters used in the simulations are $T_s = 0.25$ s, $T_{OBS} = 0.5$ s, $K = 400$ samples, $M = 3000$ particles, $\beta = 0.2$, and $G(\gamma_\tau = 0.5, \varphi_\tau = 2)$ for the Gamma distribution. The estimation performance has again been evaluated by means of the RMSE, this time defined as $RMSE = \sqrt{\frac{1}{LK}\sum_{l=1}^{L}\sum_{t=1}^{K}\left\|\mathbf{x}_t - \hat{\mathbf{x}}^l_t\right\|^2}$.

	$\mathbf{r}_t$	$\mathbf{v}_t$	$\delta\mathbf{a}_t$
RMSE	1.83 m	0.58 m/s	0.003 m/s^2

Table 2: RMSE of the target tracking

6. CONCLUSIONS

This paper has proposed a methodology for handset tracking in scenarios where GNSS, mobile communication systems and sensor networks are present. The solution is not intrusive, in the sense that required modifications in the existing systems are minimum: only communication capabilities are needed, which are commonly embedded in modern handsets, and a set of Data Fusion Centers for collecting measures and performing computations. The appropriateness of Particle Filtering methods has been justified in a general framework, and their applicability has been shown for a concrete scenario. In addition, we have proposed a modification of a recently proposed tracking algorithm which considerably reduces the computational load. The complete system has been simulated, and the obtained numerical results are encouraging. Although the work has been focused on a very specific scenario, equations and methodology are quite general and can be easily adapted to other systems and network configurations. Future work should be pointed to accurate statistical modeling for the different sources and fine–tuning of system's parameters.

REFERENCES

[1] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra–wideband radios," *IEEE SP Magazine*, vol. 22, no. 4, pp. 70–84, July 2005.

[2] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*, John Wiley & Sons, 2001.

[3] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo in Practice*, Springer, 2001.

[4] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, February 2002.

[5] P.M. Djurić and M.F. Bugallo, "Cost–reference particle filtering for dynamic systems with nonlinear and conditionally linear states," in *Proceedings of the Nonlinear Statistical Signal Processing Workshop (NSSPW)*, Cambridge, UK, 2006.

[6] A. H. Sayed, A. Tarighat, and N. Khajehnouri, "Network–based wireless location," *IEEE SP Magazine*, vol. 22, no. 4, pp. 24–40, July 2005.

[7] P. Closas, C. Fernández Prades, and J.A. Fernández Rubio, "Maximum likelihood estimation of position in GNSS," *IEEE Signal Processing Letters*, vol. 14, no. 5, pp. 359 – 362, May 2007.

[8] S. Oh, S. Sastry, and L. Schenato, "A hierarchical multiple–target tracking algorithm for sensor networks," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, 2005, pp. 2197 – 2202.

[9] W. Ng, J. Li, S. Godsill, and J. Vermaak, "A hybrid approach for online joint detection and tracking for multiple targets," in *IEEE Aerospace Conference*, Big Sky, MO, March 2005, pp. 2126 – 2141.

[10] J. Míguez and A. Artés-Rodríguez, "A monte carlo method for joint node location and maneuvering target tracking in a sensor network," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, 2006, vol. 4, pp. IV–989 – IV–992.

[11] M.J. Appel, R. Labarre, and D. Radulović, "On accelerated random search," *SIAM Journal of Optimization*, vol. 14, no. 3, pp. 708–730, 2004.

[12] S. Godsill and J. Vermaak, "Models and algorithms for tracking using trans–dimensional sequential monte carlo," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Montreal, Quebec, Canada, May 2004, vol. 3, pp. III–976 – III–979.

[13] S. Godsill and J. Vermaak, "Variable rate particle filters for tracking applications," in *Proceedings of the IEEE/SP 13th workshop on Statistical Signal Processing*, Bordeaux, France, 2005.

[14] S. J. Godsill, J. Vermaak, W. Ng, and J. Li, "Models and algorithms for tracking of manoeuvring objects using variable rate particle filters," *IEEE Transactions on Large Scale Dynamical Systems Workshop*, accepted. To appear in 2007.

[15] R. Chen and J. Liu, "Mixture Kalman filters," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 62, no. 3, pp. 493–508, 2000.