

Research Article

Azimuth and Elevation Dynamic Tracking of UAVs via 3-Axial ULA and Particle Filtering

Andrea Papaiz¹ and Andrea M. Tonello²

¹WiPLi Lab, University of Udine, Via delle Scienze 206, 33100 Udine, Italy

²EcoSys Lab, Alpen-Adria-Universität Klagenfurt, Universitätsstraße 65-67, 9020 Klagenfurt, Austria

Correspondence should be addressed to Andrea Papaiz; andrea.papaiz@uniud.it

Received 1 April 2016; Revised 23 May 2016; Accepted 29 May 2016

Academic Editor: Yanbo Zhu

Copyright © 2016 A. Papaiz and A. M. Tonello. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Unmanned Aerial Vehicles (UAVs) localization has become crucial in recent years, mainly for navigation or self-positioning and for UAV based security monitoring and surveillance. In this paper, azimuth and elevation radio positioning of UAVs are considered. The localization is based on multiple differential phase-of-arrival measures exploiting a 3-Axial Uniform Linear Array of antennas. An ad hoc particle filtering algorithm is applied to improve the positioning performance using a dynamic motion model. A novel adaptive algorithm, namely, Particles Swarm Adaptive Scattering (PSAS), is proposed to increment the algorithm stability and precision. To assess performance a Confined Area Random Aerial Trajectory Emulator (CARATE) algorithm has been developed to generate actual paths of flying UAVs. The algorithm performance is compared with the baseline method and with the average trajectory Cramér Rao lower bound to show the effectiveness of the proposed algorithm.

1. Introduction

Unmanned Aerial Vehicles (UAVs) are attracting considerable attention since they can be used for a number of consumer, industrial, and military applications ranging, for instance, from sport video making to environmental monitoring and parcel delivery [1, 2]. A key component is the technology that allows to locate and navigate the UAVs. Presently, global navigation satellite systems (GNSS) and inertial sensors are used to provide information on position, speed, and direction of movement. Reliable localization is very important also in view of new regulations that aim at better controlling the use and the status of the UAVs for higher safety and security [3]. Furthermore, recently, UAVs technology has started to be under the spotlight of police audits because of possible security threats [4].

In this paper, a radio localization approach is considered and it is based on azimuth and elevation positioning using a transmitting source as a reference. Azimuth and elevation are determined by processing with particle filtering (PF) the signals that impinge on a 3-Axial Uniform Linear Array (3A-ULA). The 3A-ULA can be mounted either on a ground base

station or on the UAVs. In the first case, namely, Ground-Localization Scenario (GL-S), the base station passively eavesdrops the signals emitted by the UAVs to determine their angular coordinates. In the latter case, namely, Self-Localization Scenario (SL-S), the ground node acts as a radio anchor allowing UAVs self-localization. The system can be used in a standalone way or to complement existing GNSS/inertial or inertial sensors employing data fusion techniques [5].

In Section 2, the analytical model for the signals received by the 3A-ULA is described. The baseline method (BM) coordinates estimates obtained from the 3A-ULA in Section 3 are iteratively processed with an ad hoc particle filtering algorithm (PF) described in Section 4. The PF technique uses a novel dynamic particles swarm management routine, namely, Particles Swarm Adaptive Scattering (PSAS), introduced in Section 5 to improve both convergence and precision performance.

In order to quantify the performance of the proposed tracking algorithm an emulator for the UAV behaviours is specifically developed in Section 6. This ad hoc numerical model, namely, Confined Area Random Aerial Trajectory

Emulator (CARATE), randomly generates trajectories in the 3-dimensional space following a well defined set of rules in order to emulate real UAV behaviours.

Finally, in Section 8, numerical results are reported to assess performance as a function of the UAVs motion model, the PF algorithm parameters, and external detrimental impairments, such as phase noise (PN) and carrier frequency Doppler shift (DS). A comparison with the average trajectory Cramér Rao lower bound (AT-CRLB) for the angular estimation of the elevation θ and azimuth ϕ is also provided.

2. Positioning System Description

A system that determines the position of flying UAVs through the analysis of a signal impinging on the 3 branches of a 3A-ULA is considered.

The GL-S is built as follows. UAVs send narrow band radio signals that are captured by a ground base station equipped with a 3A-ULA. The radio signals are properly frequency/time duplexed to allow multiple UAVs tracking. The ground node can then compute locally the UAVs azimuth and elevation coordinates [6]. By sharing the data collected from more ground nodes, it is possible to achieve a full 3D positioning for the UAVs trajectories.

The SL-S is analogously arranged. Each 3A-ULA is mounted on the UAVs so that the ground device takes the role of a radio beacon anchor [7], allowing the UAVs to be self-aware of their position. Using more anchors transmitting at different frequency or in a time division multiplexing fashion, the UAVs can accomplish a full 3D positioning. Clearly, the implementation of the SL-S is dependent on the UAV and array sizes.

The 3A-ULA is constituted, for each one of the 3 branches, by N_a antennas. Each branch is labelled with $a \in [x, y, z]$ to indicate along which one of the axes it is displaced. An antenna of the same branch a is indexed with $i \in [1, 2, \dots, N_a]$ to indicate its position along the branch. Thus, the coordinates of the antennas can be written as

$$\begin{aligned} x^{(a,i)} &= (i-1)D\delta_{xa}, \\ y^{(a,i)} &= (i-1)D\delta_{ya}, \\ z^{(a,i)} &= (i-1)D\delta_{za}, \end{aligned} \quad (1)$$

where δ_{ij} is the Kronecker delta, that is, 1 for $i = j$ and 0 otherwise, and D is the constant distance between antennas.

The signal model considered in the following describes for simplicity the scenario with only one UAV; this assumption can be easily extended to a multi-UAVs application. Both GL-S and SL-S scenarios, due to symmetry, can be modelled in the same way. Hence, in general, the downconverted signal received by the i th antenna of the 3A-ULA branch a can be written as

$$s_n^{(a,i)} = A_n e^{j[\psi_n^{(a,i)} + \gamma_n^{(a,i)} + \varphi_n + \zeta]} + w_n^{(a,i)}, \quad (2)$$

where n indexes the time, sampled with period T . The quantity $\psi_n^{(a,i)}$ is the phase of the demodulated signal impinging

on the i th sensor of the branch a , namely, the phase of arrival (PoA), in the ideal case without impairing effects. $\gamma_n^{(a,i)}$ represents the phase Doppler shift (DS) caused by the nonzero speed of the radio source with respect to each receiving array element. The parameter φ_n is the phase noise (PN) of the oscillators. It is considered white for this application [8]. The component ζ is the phase offset (PO) between modulating and demodulating oscillators. The phase components of (2) are described in more detail in the following:

$$\begin{aligned} \psi_n^{(a,i)} &= 2\pi f_0 \tau_n^{(a,i)}, \\ \gamma_n^{(a,i)} &= 2\pi f_0 C_{DS} (nT - \tau_n^{(a,i)}) \tau_n^{(a,i)}, \end{aligned} \quad (3)$$

where f_0 is the carrier frequency of the received modulated signal and where $\tau_n^{(a,i)}$ is the time of arrival (ToA) of the plane wave from the source to the sensor. $C_{DS}(t) = v_r(t)/c_0$ is the DS coefficient due to the movement of the UAV. The parameter $v_r(t)$ is the UAV speed in the direction of the center of the 3A-ULA and c_0 is the speed of light. Finally, $w_n^{(a,i)}$ is circular additive white Gaussian noise (C-AWGN) with standard deviation σ_w . The receiving architecture comprises N_a downconverting branches for each axis of the 3A-ULA all fed by the same oscillator. For this reason PN and PO components are identical for all antennas.

Considering the scenario where the transmitted signals carry also information, the component A_n of (2) is representative of the unknown informative part of the signal, specifically the modulated complex data symbols. For the application considered in this paper A_n is assumed to be an equiprobable symbol of an unknown order PSK modulation scheme. For this reason the symbol is modelled as complex uniform ring shaped noise (RSN) $A_n = |A|e^{j\xi_n}$ where $\xi_n \sim U(0, 2\pi)$ is a random uniform variable between 0 and 2π and $|A|$ is the constant amplitude of the modulated signal.

In this model, the presence of multipath propagation is not considered; a line of sight environment is assumed. This can hold true when the ground and UAV antennas have a wide vertical lobe, respectively, directed upward and downward.

3. Azimuth and Elevation Estimation with 3A-ULA

A radio source moving in the 3D Cartesian space is considered. Its angular spherical coordinates θ_n and ϕ_n are determined using the same coordinate system of the 3A-ULA defined in (1). A trajectory example is visible in Figure 2. The positioning technique used to locate the radio source is based on the estimation of the phase difference of the received signals between different antenna couples belonging to the same 3A-ULA branch a , namely, the differential phase of arrival (D-PoA) $\hat{\psi}_n^{(a)}$. The D-PoA is defined as $\hat{\psi}_n^{(a)} = \psi_n^{(a,2i-1)} - \psi_n^{(a,2i)} \forall i \in [1, 2, \dots, N_a/2]$.

In [8] a technique to estimate $\hat{\psi}_n^{(a)}$ in a scenario with impaired signals similar to (2) is described. Applying this technique for the model considered in this paper, it allows exploiting the D-PoAs from the signal model introduced in

(2). $\widehat{u}_n^{(a)}$ are estimated as $\angle u_n^{(a)}$, where \angle stands for the phase operator, where

$$u_n^{(a)} = \frac{2}{N_a} \sum_{i=1}^{N_a/2} s_n^{(a,2i-1)} s_n^{(a,2i)*}, \quad (4)$$

where an average over $N_a/2$ distinct antenna pairs is performed to mitigate noise and compensate other zero mean impairments. Using $u_n^{(a)}$, the angular spherical coordinates θ_n and ϕ_n are estimated, respectively, with $\tilde{\theta}_n$ and $\tilde{\phi}_n$ as follows:

$$\begin{aligned} \tilde{\theta}_n &= \arctan \left(\sqrt{\left(\angle u_n^{(x)}\right)^2 + \left(\angle u_n^{(y)}\right)^2}, \angle u_n^{(z)} \right), \\ \tilde{\phi}_n &= \arctan \left(\angle u_n^{(y)}, \angle u_n^{(x)} \right), \end{aligned} \quad (5)$$

where $\arctan(\cdot, \cdot)$ is the arctangent function extended to the $[0, 2\pi)$ domain.

The estimations computed in (5) will be referred to as baseline method (BM) estimations to differ them from the particle filter (PF) estimations that will be introduced in Section 4.

4. PF Algorithm Estimation

In this section, the application of the PF algorithms [9] to the localization problem introduced in Section 3 is briefly described. PF algorithm techniques have been chosen because of their simple, scalable, and flexible numerical implementation with respect to other methods like Extended Kalman Filter [10] that approach the problem linearising the system and approximating the probability density functions as Gaussians. PF is specifically designed to fit complex system models as well as nonlinear and non-Gaussian configurations. The target of PF methods is to estimate the dynamic evolution of the hidden states in a system that are the coordinates θ_n and ϕ_n of the radio source, using the sequential discrete noisy measures $u_n^{(a)}$ introduced in (4). In PF, it is necessary to hypothesize a model for the observations and for the states temporal evolution [9]. The evolution model for the hidden states θ_n and ϕ_n is

$$\begin{aligned} \theta_n &= g_\theta \left(\theta_{[n-N_{mm} \div n-1]}, p_{\theta,n} \right), \\ \phi_n &= g_\phi \left(\phi_{[n-N_{mm} \div n-1]}, p_{\phi,n} \right), \end{aligned} \quad (6)$$

while the model for the observations related to the hidden states introduced in (4) is

$$u_n^{(a)} = h \left([\theta_n, \phi_n], q_n \right). \quad (7)$$

The relations (6) show how the current hidden state $[\theta_n, \phi_n]$ evolves thanks to the previous N_{mm} states and the random uncertain variables $p_{\theta,n}$ and $p_{\phi,n}$ through the motion model function g . Relation (7) shows how the measurements $u_n^{(a)}$ depend only on the current state $[\theta_n, \phi_n]$ and on a random uncertain variable q_n through the measure model function h .

For notational simplicity, both θ_n and ϕ_n will be denoted with the generic angle α_n . When needed this generalization will be removed.

PF has been chosen because of its simplicity and versatility. PF algorithms are a numeric implementation of Bayesian estimation [10], a statistical approach to iteratively increase the knowledge of the states to be estimated by narrowing their pdf conditioned by the sequential acquisition of new measures during time. The pdf of α_n , defined as $f_{\alpha_n}[\alpha]$, is approximated in a discrete way [11] with a set of M particles $\{\alpha_n^{(m)}\}_{m=1,\dots,M}$ and a set of weights $\{w_{\alpha,n}^{(m)}\}_{m=1,\dots,M}$. Particles and weights are generated from the partial knowledge given from the measure model h and the state evolution model g_α . This knowledge is more and more incremented by the continuous time acquisition of the measures $u_n^{(a)}$. Clearly, the UAV moves so that its coordinates change over time n .

As shown in (6), in the PF algorithm an assumption about the model that controls the evolution of the hidden state of the system g_α is made. For this purpose the motion model used is the Drift Motion Model (DMM) proposed in [12]. The DMM generates the new particles cloud $\{\alpha_n^{(m)}\}$ at step n shifting the previous step resampled cloud [9] $\{\alpha_{n-1,\text{RES}}^{(m)}\}$ using the regression constant \mathbf{m}_{n-1} :

$$\alpha_n^{(m)} = \alpha_{n-1,\text{RES}}^{(m)} + \mathbf{m}_{n-1}T + \eta, \quad (8)$$

where $\alpha_{n-1,\text{RES}}^{(m)}$ is the m th particle of $n-1$ step cloud after the resampling process [11]. $\eta \sim \mathcal{N}(\sigma_\eta, m_\eta = 0)$ is a Gaussian random variable that represents the fundamental statistical scattering part of the motion model. The parameter \mathbf{m}_{n-1} is the slope of the linear regression of N_{mm} previous values of $\tilde{\alpha}_n$ that are the PF estimates of α_n . It is important to emphasize that in this regression model, due to the periodic nature of α_n , it is necessary to use the previously saved N_{mm} unwrapped values of $\tilde{\alpha}_n$. This is necessary to correctly track the azimuthal coordinate ϕ_n during 2π angular shifts from $-\pi$ to $+\pi$ going from the third to the second Cartesian quadrant.

Particles weights $w_{\alpha,n}^{(m)}$ are calculated by the PF algorithm using in each cycle the temporary estimates $\tilde{\alpha}_n$ obtained from the real acquired measures in (5). This temporary estimations are compared with the cloud of generated particles $\alpha_n^{(m)}$ through an exponential weighting function to obtain the set of weights $w_{\alpha,n}^{(m)}$:

$$w_{\alpha,n}^{(m)} = \exp \left(- \left| \alpha_n^{(m)} - \tilde{\alpha}_n \right| \right), \quad (9)$$

where, to correct unnecessary 2π offsets of $|\alpha_n^{(m)} - \tilde{\alpha}_n|$ due to the periodicity of α_n , the minimum absolute value of the distance of the two angles $\alpha_n^{(m)}$ and $\tilde{\alpha}_n$ in 2π -modulo space is computed. A reshaping technique [12] is then applied, before estimation, to regularize the particle cloud weights.

The PF estimation $\hat{\alpha}_n$ of the hidden state α_n is then computed through a weighted average of the reshaped particles swarm:

$$\hat{\alpha}_n = \sum_{m=1}^M w_{\alpha,n,\text{RESH}}^{(m)} \alpha_n^{(m)}, \quad (10)$$

where $w_{\alpha,n,\text{RESH}}^{(m)}$ is the weight after the reshaping process. Finally, after the estimation stage, the particles cloud is resampled, generating the set $\alpha_{n,\text{RES}}^{(m)}$, to eliminate spurious particles and avoid fragmentation.

5. Novel Particles Swarm Adaptive Scattering Algorithm (PSAS)

The parameter σ_s is the standard deviation of the random variable η introduced in (8). In the previous section σ_s , it was considered constant over time. However, depending on the situation of the tracking algorithm, it will be shown that it is better to have a dynamic $\sigma_{s,n}$. It is possible to define two parameters that describe the particle cloud: Ω_n , namely, the Particles Cloud Width (PC-W) that describes how much wide the set of particles is and, ε_n , the Particles Cloud Granularity (PC-G) that indicates the interparticle distance:

$$\begin{aligned}\Omega_n &= \max_m \{\alpha_n^{(m)}\} - \min_m \{\alpha_n^{(m)}\} \leq 2\pi, \\ \varepsilon_n &= \frac{\Omega_n}{(M-1)}.\end{aligned}\quad (11)$$

The value of Ω_n influences the convergence speed of the algorithm. In fact, if the particles cloud is positioned over the value to be estimated the convergence is fast because the exponential weighting function introduced in (9) gives high weights to the nearest particles and negligible weights to the further ones. Instead, if the particles cloud is located far from the value to be estimated, convergence is slower because the exponential weighting function algorithm gives almost the same weights to all particles. Thus, it will need more iterations to move the particles swarm, cycle by cycle, near the convergence point. For these reasons a bigger value of Ω_n will lead to wider clouds, encouraging the convergence. On the other hand, smaller values of Ω_n will make the algorithm slower and more unstable to fast state changes.

The value of ε_n influences the precision of convergence. In fact, hypothesising that it has reached the convergence point, the estimation precision is upper bounded by the average interparticle distance, that is, the PC-G.

Considering the spreading variable η introduced in (8) as a Gaussian variable with zero mean and standard deviation σ_s , it is possible to estimate the PC-W using the Tchebysheff's inequality as $\Omega_n \approx 6\sigma_s \propto \sigma_s$. Likewise the PC-G can be estimated as $\varepsilon_n \approx 6\sigma_s/(M-1) \propto \sigma_s$. These relations show that σ_s is directly proportional to PC-W and PC-G. Because of the previous observation it is possible to conclude that, given a swarm of M particles, during the convergence phase a bigger value of σ_s would be more appropriate because it would lead to faster results. On the other hand, after the convergence phase, smaller values of σ_s would be more appropriate to increase the estimation precision.

For these reasons in this paper a novel algorithm that manages the dynamic evolution of σ_s across the iterations, namely, Particle Swarm Adaptive Scattering (PSAS), is proposed.

The distance metric $d_n^{(m)}$ is defined as a baseline error cloud for the particles and is used by the PSAS to select the convergence status of the PF algorithm. The error cloud $d_n^{(m)}$ is calculated between the baseline estimation $\tilde{\alpha}_n$ introduced

in (5) and the resampled particle swarm $\alpha_{n,\text{RES}}^{(m)}$ as follows:

$$\begin{aligned}d_n^+ &= \text{avg}(d_n^{(m)}) + \text{std}(d_n^{(m)}), \\ d_n^- &= \text{avg}(d_n^{(m)}) - \text{std}(d_n^{(m)}), \\ d_n^{(m)} &= |\alpha_{n,\text{RES}}^{(m)} - \tilde{\alpha}_n|,\end{aligned}\quad (12)$$

where the operators avg and std denote, respectively, the numeric operations of average and standard deviation and the absolute value operates in the periodic $[-\pi, \pi]$ domain. The parameters d_n^- and d_n^+ represent, respectively, the lower and higher bounds of the error cloud $d_n^{(m)}$ and are used as thresholds by the PSAS algorithm to trigger the increment or the decrement of $\sigma_{s,n}$ as follows:

$$\sigma_{s,n} = \sigma_{s,n-1} (1 \pm \delta\sigma_s), \quad (13)$$

where $\delta\sigma_s > 0$ is the proportional feedback parameter of the PSAS. The value of $\delta\sigma_s$, depending on the two feedback parameters d_n^+ and d_n^- and the threshold value γ_{th} , is set positive to iterative increment $\sigma_{s,n}$ and negative to decrement $\sigma_{s,n}$ and is set to zero to keep it constant.

The value of $\sigma_{s,n}$ is incremented with respect to its previous value when the condition of nonconvergence is detected. This condition holds true when the error cloud $d_n^{(m)}$ is reasonably out of $[-\gamma_{\text{th}}, \gamma_{\text{th}}]$ convergence interval, that is, when $d_n^- > \gamma_{\text{th}}$ or $d_n^+ < -\gamma_{\text{th}}$. Then, the PC-W is incremented to encourage convergence. If the previous condition is not satisfied, the value of $\sigma_{s,n}$ is decremented w.r.t. its previous value when the convergence state is detected. This condition holds true when the error cloud $d_n^{(m)}$ is on the edges of $[-\gamma_{\text{th}}, \gamma_{\text{th}}]$ convergence interval, that is, when $d_n^+ > \gamma_{\text{th}}$ or $d_n^- < -\gamma_{\text{th}}$. Then, the PC-G is reduced to increment the precision. For the remaining values of d_n^+ and d_n^- the condition of tight convergence is reached. In this state the value of $\sigma_{s,n}$ is kept the same w.r.t. its previous value. This PSAS case keeps the cloud at the same size to avoid instability conditions due to rapid changes of the convergence point. The values of the feedback parameter $\delta\sigma_s$ and the threshold parameter γ_{th} are related to the dynamic behaviour of the convergence point that is related to the UAV motion model. Optimal values for $\delta\sigma_s$ and γ_{th} were numerically calculated for the trajectories generated by CARATE in this paper.

6. UAVs Trajectory Emulator

To assess performance of BM and PF algorithms estimations and to evaluate the effects of different parameters on the estimation process, it is necessary to emulate UAVs trajectories using a proper coordinates evolution algorithm. Herein, a possible trajectory generation method is proposed.

The proposed algorithm, named Confined Area Random Aerial Trajectory Emulator (CARATE), generates iteratively a 3D path obtained from a variable length previous history of the trajectory and a tunable set of random variables. CARATE is specifically designed to emulate UAVs trajectories inside a limited flight area, in the GL-S, around the receiving array.

Thus, the generated trajectories allow testing the tracking algorithms for a broad range of arrival angles. An UAV located far from the receiving array instead would be localized using a limited range of angles. The Cartesian position $[x_n, y_n, z_n]$ at time n of the UAV evolves as

$$\begin{aligned} x_n &= x_{n-1} + T v_n \cos(\phi'_n) \sin(\theta'_n), \\ y_n &= y_{n-1} + T v_n \sin(\phi'_n) \sin(\theta'_n), \\ z_n &= z_{n-1} + T v_n \cos(\theta'_n). \end{aligned} \quad (14)$$

The angle ϕ'_n , depicted in Figure 1, is the local azimuthal angle at time n of the actual trajectory coordinates $[x_n, y_n, z_n]$ seen from the previous path point $[x_{n-1}, y_{n-1}, z_{n-1}]$. The local elevation angle θ'_n can be defined analogously. v_n is the absolute speed of the UAV. At each step, the trajectory increments generated by θ'_n , ϕ'_n , and v_n are related to the “seed” components $\theta_{\text{seed},n}$, $\phi_{\text{seed},n}$, and $v_{\text{seed},n}$ and to the random components $\delta\theta_n$, $\delta\phi_n$, and δv_n as follows:

$$\begin{aligned} \theta'_n &= \theta_{\text{seed},n} + \delta\theta_n, \\ \phi'_n &= \phi_{\text{seed},n} + \delta\phi_n, \\ v_n &= v_{\text{seed},n} + \delta v_n. \end{aligned} \quad (15)$$

The random components are generated following, respectively, the normal distributions $\delta\theta \sim N(m_{\delta\theta}, \sigma_{\delta\theta})$, $\delta\phi \sim N(m_{\delta\phi}, \sigma_{\delta\phi})$, $\delta v \sim N(m_{\delta v}, \sigma_{\delta v})$. The parameters of the random component distributions affect the UAV path behaviour. For example, for $m_{\delta v} > 0$, the UAV will have an accelerating trend and for $m_{\delta\phi} > 0$ the UAV will tend to turn anticlockwise. The seed values introduced in (15) are the history parameters that express how much the evolution of the path is related to its past and are given by

$$\begin{aligned} \theta_{\text{seed},n} &= \arccos \left[\Delta z_n(N_{\text{ang}}), \Delta r_n(N_{\text{ang}}) \right], \\ \phi_{\text{seed},n} &= \arctan \left[\Delta y_n(N_{\text{ang}}), \Delta x_n(N_{\text{ang}}) \right], \\ v_{\text{seed},n} &= \sum_{k=1}^{N_{\text{sp}}} \frac{v_{n-k}}{N_{\text{sp}}}, \end{aligned} \quad (16)$$

where $\arctan(\cdot, \cdot)$ is the arctangent function defined within $[0, 2\pi)$ and where

$$\begin{aligned} \Delta x_n(N_{\text{ang}}) &= x_{n-1} - x_{n-N_{\text{ang}}}, \\ \Delta y_n(N_{\text{ang}}) &= y_{n-1} - y_{n-N_{\text{ang}}}, \\ \Delta z_n(N_{\text{ang}}) &= z_{n-1} - z_{n-N_{\text{ang}}}, \\ \Delta r_n(N_{\text{ang}}) &= \sqrt{\Delta x_n(N_{\text{ang}})^2 + \Delta y_n(N_{\text{ang}})^2 + \Delta z_n(N_{\text{ang}})^2}. \end{aligned} \quad (17)$$

In Figure 1, the relations among the CARATE components for the variable ϕ_n are represented. As shown in (16), the smaller

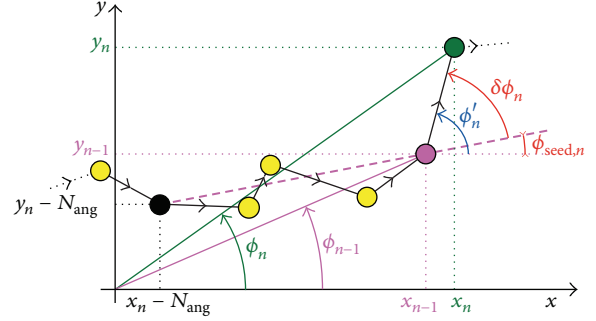


FIGURE 1: Diagram of CARATE components for variable ϕ ($N_{\text{ang}} = 5$).

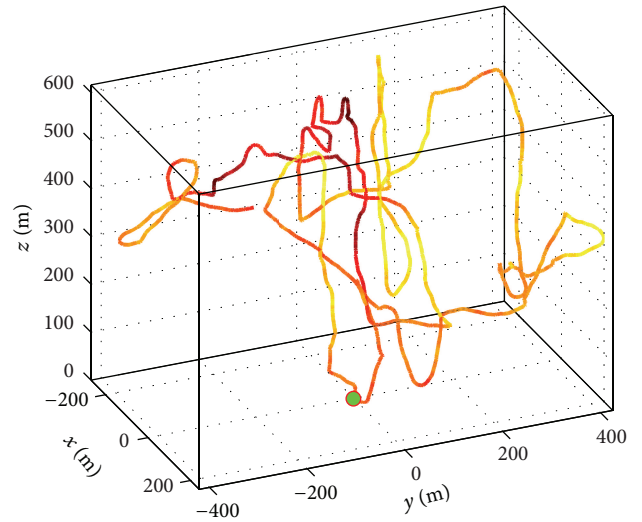


FIGURE 2: Example of UAV 3D trajectory generated with CARATE.

the window width of the seed parameters N_{ang} and N_{sp} , the higher the influence of the path randomization with respect to the past history of the path itself. The same effect will occur for higher values of the standard deviation $\sigma_{\delta\theta}$, $\sigma_{\delta\phi}$, and $\sigma_{\delta v}$ of the random components $\delta\theta$, $\delta\phi$, and δv introduced in (15).

The CARATE algorithm, as visible in the sample path in Figure 2, generates a 3D trajectory inside a limited flight area, formed by the horizontal bound region S_{xy} and the vertical bound region S_z . S_{xy} is defined as $\{|x| < L_{xy} \cap |y| < L_{xy}\}$ while S_z is defined as $\{z > L_z^{\text{min}} \cap z < L_z^{\text{max}}\}$. A specific routine of the algorithm has been developed to keep the trajectory inside the limited flight area, preserving it smooth and without sharp edges on region bounds. If x_n or y_n exceed S_{xy} the value of $\phi_{\text{seed},n}$ introduced in (16) is forced for the subsequent N_{bend} steps to follow a set of equispaced values between $\phi_{\text{seed},n}$ and $\phi_{\text{seed},\text{END}}$. The value of $\phi_{\text{seed},\text{END}}$ is appropriately defined depending on the Cartesian quadrant where the trajectory crosses the surface of S_{xy} . Analogously, if z_n exceeds S_z the value of $\theta_{\text{seed},n}$ introduced in (16) is forced for the next N_{bend} steps to a set of equally spaced values between $\theta_{\text{seed},n}$ and

$\theta_{\text{seed,END}}$. The value of $\theta_{\text{seed,END}}$ is also defined depending on the Cartesian quadrant where the surface of S_z has been crossed.

7. Cramér-Rao Lower Bound

In statistics the Cramér-Rao Lower Bound (CRLB) expresses the lowest value of the Root Mean Square Error (RMSE) of an optimal estimator given a certain set of data [13]. In this paper the estimations of the coordinates $[\theta_n, \phi_n]$ are carried out from the information given by the measurements $u_n^{(x)}$, $u_n^{(y)}$, and $u_n^{(z)}$.

Now, we define $\mathbf{b}_n = [b_n^{(a)}]_{n \in [x,y,z]}$ as the vector of ideal measured values, namely, $u_n^{(a)}$, in (4) but without the detrimental effects due to the impairments introduced in (2) such as PN, DS, and C-AWGN:

$$b_n^{(a)} = |A|^2 e^{j\tilde{\psi}_n^{(a)}}, \quad (18)$$

where, as introduced in Section 2, $\tilde{\psi}_n^{(a)}$ is the D-PoA for a th 3A-ULA antenna branch and $|A|$ is the constant amplitude of the complex downconverted constellation. The vector of real measurements $\mathbf{u}_n = [u_n^{(x)}, u_n^{(y)}, u_n^{(z)}]$, defined in (4), can be written as

$$u_n^{(a)} = \frac{2}{N_a} \sum_{i=1}^{N_a/2} \left(|A|^2 e^{j\tilde{\psi}_n^{(a)}} + W_n^{(a,i)} \right) = \widehat{A} e^{j\tilde{\psi}_n^{(a)}} + \widehat{W}_n^{(a)}, \quad (19)$$

where the intermediate signal $|A|^2 e^{j\tilde{\psi}_n^{(a)}} + W_n^{(a,i)}$ is averaged through antenna pairs and $\widehat{A} = (2/N_a) \sum_{i=1}^{N_a/2} |A|^2 = |A|^2$. The noise components in (19) are

$$\begin{aligned} W_n^{(a,i)} &= |A| e^{j\psi_n^{(a,2i-1)}} w_n^{(a,2i)*} + |A|^* e^{-j\psi_n^{(a,2i)}} w_n^{(a,2i-1)} \\ &\quad + w_n^{(a,2i-1)} w_n^{(a,2i)*}, \quad (20) \\ \widehat{W}_n^{(a)} &= \frac{2}{N_a} \sum_{i=1}^{N_a/2} W_n^{(a,i)}, \end{aligned}$$

where the notation $(\cdot)^*$ denotes the complex conjugate operator. $w_n^{(a,i)}$ is the C-AWGN with zero mean and variance σ_w^2 . $\psi_n^{(a,i)}$ is the PoA as defined in (3). The expectation of the differential noise $W_n^{(a,i)}$ among the antennas of the same branch is $m_W = 0$, while its variance $\sigma_W^2 = 2|A|^2 \sigma_w^2 + \sigma_w^4$. Analogously, the average differential noise $\widehat{W}_n^{(a)}$ is described by its mean $m_{\widehat{W}} = 0$ and its variance is $\sigma_{\widehat{W}}^2 = (2/N_a) \sigma_W^2$.

To calculate the CRLB it is necessary to evaluate the log-likelihood $L(\phi, \theta) = \ln P[\mathbf{u}_n | \mathbf{b}_n]$ of the acquired data given the parameters to be estimated. $\ln(\cdot)$ represents the natural logarithm. In this context, for the calculation of CRLB, the measurements vector \mathbf{u}_n is considered affected only by C-AWGN noise without the influence of the other impairments introduced in (2). This assumption yields to the likelihood Gaussianity and is verified via numerical fitting similarly to [14], so that

$$L(\phi, \theta) = -3 \ln(\pi \sigma_{\widehat{W}}^2) - \frac{1}{\sigma_{\widehat{W}}^2} \|\mathbf{b} - \mathbf{u}\|^2, \quad (21)$$

where the time index n is omitted until the end of the CRLB derivation for notational simplicity. The CRLB of the variable α_r , defined as r th element of the unknowns vector $\boldsymbol{\alpha} = [\phi, \theta]$, is

$$\text{CRLB}_{\alpha_r}(\phi, \theta) = \{F(\phi, \theta)^{-1}\}_{(r,r)}, \quad (22)$$

where $F(\phi, \theta)$ is the Fischer information matrix:

$$F(\phi, \theta) = -E \left\{ \left[\frac{\partial^2}{\partial \alpha_n \partial \alpha_k} L(\phi, \theta) \right]_{(h,k) \in [1,2]} \right\}. \quad (23)$$

For the log-likelihood function described in (21), we obtain

$$\begin{aligned} F(\phi, \theta) &= \begin{bmatrix} \left\| \frac{\partial \boldsymbol{\beta}}{\partial \phi} \right\|^2 & \frac{\partial \boldsymbol{\beta}}{\partial \phi} \frac{\partial \boldsymbol{\beta}^T}{\partial \theta} \\ \frac{\partial \boldsymbol{\beta}}{\partial \theta} \frac{\partial \boldsymbol{\beta}^T}{\partial \phi} & \left\| \frac{\partial \boldsymbol{\beta}}{\partial \theta} \right\|^2 \end{bmatrix} \\ &= 2 \left(\frac{2\pi |A|^2 D_{\text{norm}}}{\sigma_{\widehat{W}}} \right)^2 \begin{bmatrix} (\sin \theta)^2 & 0 \\ 0 & 1 \end{bmatrix}, \quad (24) \end{aligned}$$

where $\boldsymbol{\beta} = [\cos \phi \sin \theta, \sin \phi \sin \theta, \cos \theta]$ is the unit direction vector and $D_{\text{norm}} = D/\lambda_0$ is the normalized interantenna distance (Figure 4). The notation $(\cdot)^T$ denotes the transpose operator. Finally, the CRLB expressions for ϕ and θ are

$$\begin{aligned} \text{CRLB}_{\phi}(\phi, \theta) &= \frac{1}{2} \left(\frac{\sigma_{\widehat{W}}}{2\pi |A|^2 D_{\text{norm}}} \right)^2 \left(\frac{1}{\sin \theta} \right)^2, \\ \text{CRLB}_{\theta}(\phi, \theta) &= \frac{1}{2} \left(\frac{\sigma_{\widehat{W}}}{2\pi |A|^2 D_{\text{norm}}} \right)^2. \quad (25) \end{aligned}$$

The RMSE of every estimator of the unknown α_r from the measurements \mathbf{u} is lower bounded by $\sqrt{\text{CRLB}_{\alpha_r}}$. It is important to state again that this C-AWGN-only CRLB calculated in (25) takes into account C-AWGN but not the other previously introduced impairments, in particular DS and PN. Given the previous assumptions, the AWGN-only CRLB is lower or equal to the fully impaired CRLB. Another relevant point is that the computed CRLB does not take into account the motion model and the trajectory history differently from the PF algorithm. The CRLB is simply computed from the 3 observations $u_n^{(a)}$ introduced in (4) at time nT , as the BM does. Instead, the proposed PF algorithm exploits the motion model jointly with PSAS, using information coming from the trajectory history. For this reason, it may occur that the PF estimation offers better performance than the CRLB.

8. Performance Analysis

The performance of the PF algorithm is now assessed. A comparison with the BM estimation introduced in Section 3 and the CRLB calculated in Section 7 is also made. Different UAV trajectories are generated with the CARATE to assess performance. Each UAV trajectory is N time samples long. In order to have a common performance evaluation metric, in

the following we define the average of the Root Mean Square Error (RMSE) calculated through every trajectory sample, namely, the Average Trajectory RMSE (AT-RMSE).

n_{traj} th generated trajectory is described with the set $\{\theta_n^{(n_{\text{traj}})}, \phi_n^{(n_{\text{traj}})}\}$, its BM estimation is $\{\tilde{\theta}_n^{(n_{\text{traj}})}, \tilde{\phi}_n^{(n_{\text{traj}})}\}$, and its PF estimation $\{\hat{\theta}_n^{(n_{\text{traj}})}, \hat{\phi}_n^{(n_{\text{traj}})}\}$. Their AT-RMSE are, respectively, $\overline{\text{RMSE}}_{\alpha, \text{BM}} = (1/N_{\text{traj}}) \sum_{n_{\text{traj}}=1}^{N_{\text{traj}}} \text{RMSE}_{\alpha, \text{BM}}^{(n_{\text{traj}})}$ and $\overline{\text{RMSE}}_{\alpha, \text{PF}} = (1/N_{\text{traj}}) \sum_{n_{\text{traj}}=1}^{N_{\text{traj}}} \text{RMSE}_{\alpha, \text{PF}}^{(n_{\text{traj}})}$ for $\alpha \in [\theta, \phi]$ where

$$\text{RMSE}_{\alpha, \text{BM}}^{(n_{\text{traj}})} = \sqrt{\frac{\sum_{n=1}^N \left(\alpha_n^{(n_{\text{traj}})} - \tilde{\alpha}_n^{(n_{\text{traj}})} \right)^2}{N}}, \quad (26)$$

$$\text{RMSE}_{\alpha, \text{PF}}^{(n_{\text{traj}})} = \sqrt{\frac{\sum_{n=1}^N \left(\alpha_n^{(n_{\text{traj}})} - \hat{\alpha}_n^{(n_{\text{traj}})} \right)^2}{N}}$$

are the RMSEs calculated for each trajectory n_{traj} . N_{traj} is the number of generated trajectories.

Analogously, with the purpose to allow a performance comparison with the CRLB, we define the average trajectory CRLB (AT-CRLB), as $\overline{\text{CRLB}}_{\alpha} = (1/N_{\text{traj}}) \sum_{n_{\text{traj}}=1}^{N_{\text{traj}}} \text{CRLB}_{\alpha}^{(n_{\text{traj}})}$ for $\alpha \in [\theta, \phi]$ and where

$$\text{CRLB}_{\alpha}^{(n_{\text{traj}})} = \frac{1}{N} \sum_{n=1}^N \text{CRLB}_{\alpha} \left(\phi_n^{(n_{\text{traj}})}, \theta_n^{(n_{\text{traj}})} \right). \quad (27)$$

The default parameters for the performance evaluation are (a) angles $\sigma_{\delta\theta} = 7^\circ$, $\sigma_{\delta\phi} = 7^\circ$, and $N_{\text{ang}} = 20$; (b) speeds $v_{\text{min}} = 0.5$ km/h, $v_{\text{max}} = 10$ km/h, and $N_{\text{sp}} = 2$. The array parameters are $N_a = 6$ and $D_{\text{norm}} = 1/3$. The PF algorithm basic settings are $M = 20$ and $N_{\text{mm}} = 20$. PSAS parameters $\delta\sigma_s = 25\%$ and $\gamma_{\text{th}} = 10^\circ$ were numerically optimized for the CARATE algorithm. In the signal model introduced in (2), the PN is considered white with standard deviation $\sigma_{\text{PN}} = 1^\circ$, the PO is considered uniform in $[0, 2\pi)$ for each array element, and the signal-to-noise ratio is $\text{SNR} = [|A|^2/\sigma_w^2]$ dB = 10 dB. The number of simulated trajectories N_{traj} and their length N have been properly dimensioned to guarantee the statistical confidence of the results.

In Figure 2 an example of 3D-trajectory generated for performance evaluation is shown. The 3A-UULA is positioned in the center of the axis.

8.1. Performance with Static and Dynamic σ_s . We now consider the performance as a function of the signal-to-noise ratio $\text{SNR} = |A|^2/\sigma_w^2$. In Figure 3, the AT-RMSE of the azimuthal coordinate ϕ and the elevation coordinate θ for both the PF algorithm and the BM are reported. Furthermore the impact on performance of the PSAS algorithm introduced in Section 5 is analysed. It is possible to observe how, due to the geometrical asymmetry, the performance in terms of AT-RMSE is significantly different between θ and ϕ . Greater SNR values lead to lower AT-RMSE for both PF and BM estimations. It is important to state that, for a static value of σ_s , PF performs better than BM in a broad range of SNR values. For example, at 2 dB of SNR, it outperforms the BM

algorithm by about 17° for ϕ and 10° for θ . The analysis shows that the introduction of the PSAS algorithm leads to a further improvement of performance. For example, at 2 dB of SNR the PSAS leads to a performance improvement of 5° for ϕ and 3° for θ with respect to the use of a static σ_s . It is visible in Figure 3 how the introduction of the PF algorithm allows overcoming the AT-CRLB. This is possible thanks to the a priori knowledge of the UAV path extracted by the motion model from the previous steps of the trajectory. For higher SNR the performance improvement of PF with respect to the BM algorithm is less prominent than for lower SNR.

8.2. Performance for Different Interantenna Distances D_{norm} . Now, the behaviour of the AT-RMSE for the BM and PF algorithms is analysed varying the normalized interantenna distance D_{norm} . To unambiguously extract the phase from $u_n^{(a)}$ the interelement distance must be less than half of the impinging signal wavelength [15]: $D_{\text{norm}} \leq 0.5$. Furthermore in (25) it is visible that for higher values of D_{norm} the CRLBs increase. This leads to $D_{\text{norm}} = 0.5$ as the optimal value for the estimation. However, in lower SNR scenarios and for D_{norm} values near 0.5 the CRLB is found to be overly optimistic. In fact, high C-AWGN and the other impairments in (2) severely impair performance corrupting the value of $\angle u_n^{(x)}$ in (4) and making it exceeding 180° . The PF algorithm, thanks to its hypothesised a priori knowledge of the UAV path given by the motion model, attenuates this impairing effect. In Figure 3, we can see that the proposed PF algorithm is less dependent from $D_{\text{norm}} = Df_0/c_0$ than the BM. Thus, the same 3A-UULA is usable for a wider set of carrier frequency f_0 with respect to the BM that exhibits its optimal performance for a narrow interval of D_{norm} around 0.4. For this reason, the PF algorithm can offer good performance also in a multiple UAVs scenario where multiplexing is implemented in a frequency division fashion.

9. Conclusions

We have discussed the application of an appropriately designed PF algorithm for self-localization (SL-S) and ground localization (GL-S) of UAVs using a 3A-UULA of antennas, showing an overall increase of performance with respect to the BM. A novel algorithm to manage the amplitude of the particle swarm, namely, Particles Swarm Adaptive Scattering (PSAS), has been developed and tested, showing a further increase of precision. A complete, fully adjustable, and effective 3D UAV's trajectory emulator, namely, CARATE, has been proposed and used to assess performance. Strong impairing effects like Doppler spread, phase offset, and phase noise have been considered in the performance evaluation. The effects on the proposed localization algorithm of the PF model parameters as well as the SNR and the 3A-UULA characteristics have been studied. Numerical results show that the proposed PF algorithm is able of dynamically track the UAVs angular position better than the BM. A critical point of this approach is that, similarly to PSAS behaviour, a dynamic calibration procedure has to be implemented to optimize the PF parameters. Future work will also investigate

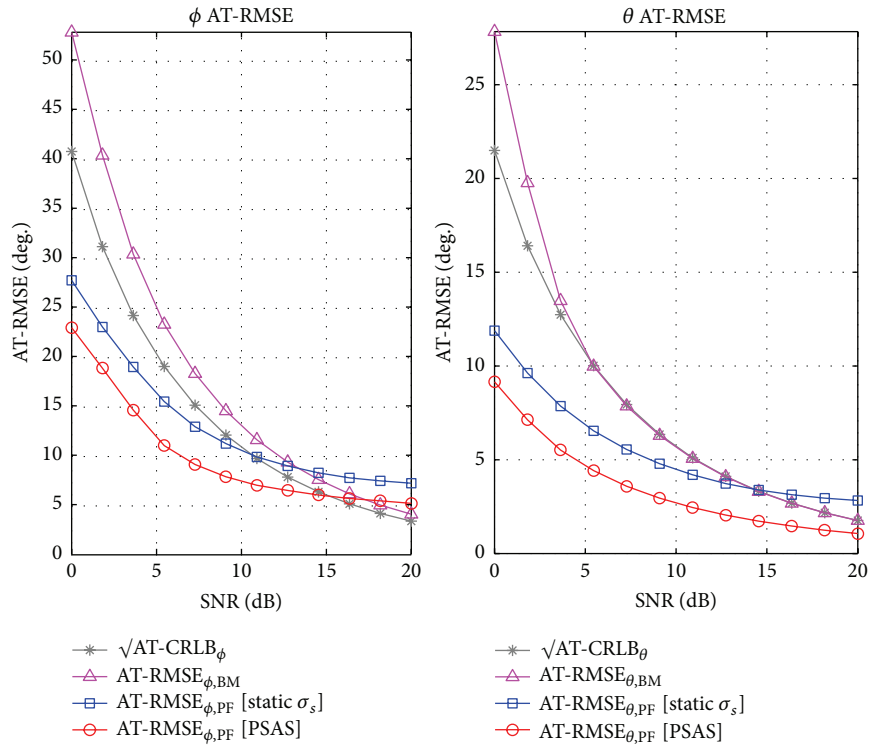


FIGURE 3: AT-RMSE for different SNR_{dB} conditions with dynamic and static σ_s .

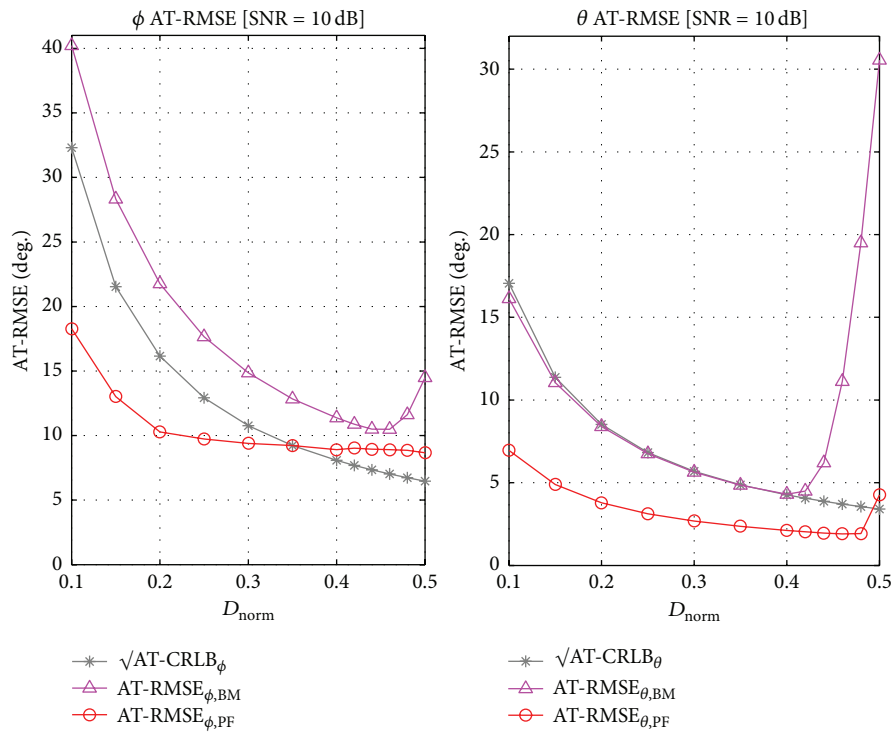


FIGURE 4: AT-RMSE for different normalized 3A-ULA interantenna distance $D_{\text{norm}} = D/\lambda_0$.

the integration of ranging algorithms like Received Signal Strength Estimation (RSSE) and ToA to PF to provide full 3D localization.

Competing Interests

The authors declare that they have no competing interests.

References

- [1] P. B. Quater, F. Grimaccia, S. Leva, M. Mussetta, and M. Aghaei, "Light Unmanned Aerial Vehicles (UAVs) for cooperative inspection of PV plants," *IEEE Journal of Photovoltaics*, vol. 4, no. 4, pp. 1107–1113, 2014.
- [2] M. R. Haque, M. Muhammad, D. Swarnaker, and M. Arifuz-zaman, "Autonomous Quadcopter for product home delivery," in *Proceedings of the 1st International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT '14)*, pp. 1–5, Dhaka, Bangladesh, April 2014.
- [3] E. Ackerman, "FAA unveils drone rules: autonomy is in, drone delivery is out," *IEEE Spectrum*, February 2015, <http://spectrum.ieee.org/automaton/robotics/drones/faa-proposed-commercial-drone-rules/>.
- [4] M. Peacock and M. N. Johnstone, "Towards detection and control of civilian unmanned aerial vehicles," in *Proceedings of the Australian Information Warfare Conference (WCICA '13)*, SRI Security Research Institute, Edith Cowan University, Perth, Western Australia, 2013.
- [5] S. Panzneri, F. Pascucci, and G. Ulivi, "An outdoor navigation system using GPS and inertial platform," *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 2, pp. 134–142, 2002.
- [6] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.
- [7] K. Dogancay, "UAV path planning for passive emitter localization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 2, pp. 1150–1166, 2012.
- [8] D. Inserra and A. M. Tonello, "Characterization of hardware impairments in multiple antenna systems for doa estimation," *Journal of Electrical and Computer Engineering*, vol. 2011, Article ID 908234, 10 pages, 2011.
- [9] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.
- [10] S. Maskell and N. Gordon, "A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking," in *Target Tracking: Algorithms and Applications*, vol. 2, Ref. no. 2001/174, pp. 2/1–2/15, IEE, 2001.
- [11] Y. Liu, B. Wang, W. He, J. Zhao, and Z. Ding, "Fundamental principles and applications of particle filters," in *Proceedings of the 6th World Congress on Intelligent Control and Automation (WCICA '06)*, vol. 2, pp. 5327–5331, June 2006.
- [12] A. Papaiz and A. M. Tonello, "Particle filtering with weight reshaping for opportunistic angle of arrival estimation in a vehicular scenario," in *Proceedings of the IEEE 5th International Conference on Consumer Electronics (ICCE '15)*, pp. 145–149, Berlin, Germany, September 2015.
- [13] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall, 1993.
- [14] D. Inserra and A. M. Tonello, "DoA estimation with compensation of hardware impairments," in *Proceedings of the IEEE 72nd Vehicular Technology Conference Fall (VTC '10)*, pp. 1–5, Ottawa, Canada, September 2010.
- [15] Z.-M. Liu and F.-C. Guo, "Azimuth and elevation estimation with rotating long-baseline interferometers," *IEEE Transactions on Signal Processing*, vol. 63, no. 9, pp. 2405–2419, 2015.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

