Parallel Concatenation of Bayesian Filters: Turbo Filtering

June 14, 2018

Abstract

In this manuscript a method for developing novel filtering algorithms through the parallel concatenation of two Bayesian filters is illustrated. Our description of this method, called turbo filtering, is based on a new graphical model; this allows us to efficiently describe both the processing accomplished inside each of the constituent filter and the interactions between them. This model is exploited to develop two new filtering algorithms for conditionally linear Gaussian systems. Numerical results for a specific dynamic system evidence that such filters can achieve a better complexity-accuracy tradeoff than marginalized particle filtering.

Giorgio M. Vitetta, Pasquale Di Viesti, Emilio Sirignano and Francesco Montorsi

University of Modena and Reggio Emilia
Department of Engineering "Enzo Ferrari"
Via P. Vivarelli 10/1, 41125 Modena - Italy
email: giorgio.vitetta@unimore.it, 190010@studenti.unimore.it, emilio.sirignano@unimore.it, francesco.montorsi@gmail.com

Keywords: Hidden Markov Model, Particle Filter, Sum-Product Algorithm, Marginalized Particle Filter, Turbo Decoding, Concatenated Channel Coding.
1 Introduction

The nonlinear filtering problem consists of inferring the posterior distribution of the hidden state of a nonlinear dynamic system from a set of past and present measurements [1]. A general recursive solution to this problem, known as Bayesian filters (e.g., see [1] Sect. II, eqs. (3)-(5)), is available, but, unfortunately, can be put in closed form in few cases [4]. In the past, various filtering methods generating a functional approximation of the desired posterior pdf have been developed; these can be divided into local and global methods on the basis of the way the posterior pdf is approximated [2], [3]. On the one hand, local techniques, like extended Kalman filtering (EKF) [4], are computationally efficient, but may suffer from error accumulation over time; on the other hand, global techniques, like sequential Monte Carlo (SMC) algorithms [5], [6] (also known as particle filtering, PF [7], [8]) may achieve high accuracy at the price, however, of unacceptable complexity and numerical problems. These considerations have motivated the investigation of other methods able to achieve high accuracy under given computational constraints. Some of such solutions are based on the idea of combining (i.e., concatenating) local and global methods; relevant examples of this approach are represented by a) marginalized particle filtering (MPF) [9] and other techniques related to it (e.g., see [3] and [10]) and b) cascaded architectures based on the joint use of EKF and PF (e.g., see [11] and [12]). Note that, in all these cases, two heterogeneous methods are combined in a way that the resulting filtering algorithm is forward only and, within its recursion, each of such methods is executed only once; for this reason, if the jargon of coding theory is adopted in this context, such filtering algorithms can be seen as specific instances of the general concept of serial concatenation [13], [14] of two (constituent) filtering methods.

In this manuscript, we focus on the novel concept of parallel concatenation (PC) of Bayesian filterings, i.e. on the idea of combining two (constituent) filters in a way that, within each recursion of the resulting concatenated algorithm, they can iteratively refine their statistical information through the mutual exchange of probabilistic (i.e., soft) information; this concept is dubbed turbo filtering (TF) for its resemblance to the iterative (i.e., turbo) decoding of concatenated channel codes [15]. More specifically, we first develop a general graphical model that allows us to: a) represent the PC of two Bayesian filters as the interconnection of two soft-in soft-out (SISO) modules, b) represent the iterative processing accomplished by these modules as a message passing technique and c) to derive the expressions of the passed messages by applying the
sum-product algorithm (SPA) [16], [17], together with a specific scheduling procedure, to the graphical model itself. Then, the usefulness of this approach is exemplified by developing two TF algorithms for the class of conditionally linear Gaussian (CLG) SSMs [9]. Our computer simulations for a specific CLG SSM evidence that, in the considered case, these algorithms perform very closely to MPF, but are substantially faster.

It is worth mentioning that the TF principle has been formulated for the first time in [18], where it has also been successfully applied to inertial navigation. However, all the theoretical results illustrated in this manuscript have been obtained later and have been inspired by various results available in the literature about: a) the representation of filtering methods as message passing procedures on factor graphs (e.g., see [16], [17] and [19]); b) the use of graphical models in the derivation and interpretation of turbo decoding and turbo equalization [16], [17], [20].

The remaining part of this manuscript is organized as follows. A description of the considered SSM is illustrated in Section 2. In Section 3 a new graphical model describing the TF principle is devised; then, a specific case of that model, referring to the use of an extended Kalman filter and particle filter as constituent filters, and a CLG SSM is analysed. The derivation of two TF algorithms based on the last model is illustrated in Section 4, whereas their interpretation from a coding theory perspective is discussed in Section 5. Such algorithms are compared with EKF and MPF, in terms of accuracy and execution time, in Section 6. Finally, some conclusions are offered in Section 7.

2 Model Description

In the following we focus on a discrete-time CLG SSM [9], whose $D$-dimensional hidden state $x_l \triangleq [x_{0,l}, x_{1,l}, ..., x_{D-1,l}]^T$ in the $l$-th interval is partitioned as $x_l = [(x^{(L)}_l)^T, (x^{(N)}_l)^T]^T$; here, $x^{(L)}_l \triangleq [x^{(L)}_{0,l}, x^{(L)}_{1,l}, ..., x^{(L)}_{D_L-1,l}]^T$ $(x^{(N)}_l \triangleq [x^{(N)}_{0,l}, x^{(N)}_{1,l}, ..., x^{(N)}_{D_N-1,l}]^T$ is the so called linear (nonlinear) component of $x_l$, with $D_L < D$ ($D_N = D - D_L$). Following [9] and [10], the models

$$x^{(Z)}_{l+1} = A^{(Z)}_l (x^{(N)}_{l}) x^{(L)}_{l} + f^{(Z)}_l (x^{(N)}_{l}) + w^{(Z)}_l$$

(1)

and

$$y_l \triangleq [y_{0,l}, y_{1,l}, ..., y_{P-1,l}]^T = g_l (x^{(N)}_l) + B_l (x^{(N)}_l) x^{(L)}_l + e_l$$

(2)
are adopted for the update of the linear \((Z = L)\) and nonlinear \((Z = N)\) components, and for the \(P\)-dimensional vector of noisy measurements available in the \(l\)-th interval, respectively. In the state update model \(f_l(x) = A_l(N) x_l^{(N)} + B_l(x_l^{(N)}) x_l^{(L)} + w_l^{(N)}\), \(l\)-th element of the process noise sequence \(\{w_k^{(N)}\}\); this sequence consists of \(D_Z\)-dimensional independent and identically distributed (iid) Gaussian noise vectors, each characterized by a zero mean and a covariance matrix \(C_{w}^{(Z)}\) (independence between \(\{w_k^{(L)}\}\) and \(\{w_k^{(N)}\}\) is also assumed for simplicity). Moreover, in the measurement model \(g_l(x_l^{(N)}) = B_l(x_l^{(N)}) x_l^{(L)} + e_l\), \(l\)-th element of the measurement noise sequence \(\{e_k\}\); this sequence consists of \(P\)-dimensional iid Gaussian noise vectors (each characterized by a zero mean and a covariance matrix \(C_{e}\), and is independent of both \(\{w_k^{(N)}\}\) and \(\{w_k^{(L)}\}\).

In the following we take into consideration not only the detailed models \(f_l(x)\) and \(g_l(x_l^{(N)})\), but also their more compact counterparts

\[
x_{l+1} = f_l(x_l) + w_l
\]

and

\[
y_l = h_l(x_l) + e_l
\]

respectively, which refer to the whole state; here, \(f_l(x_l)\) \((w_l)\) is a \(D\)-dimensional function (Gaussian noise vector) deriving from the ordered concatenation of the vectors \(A_l^{(L)} x_l^{(N)} + A_l^{(N)} x_l^{(N)}\) and \(f_l^{(L)} x_l^{(N)} + f_l^{(N)} x_l^{(N)}\) \((w_l^{(L)} + w_l^{(N)})\); see \(\text{I}\), and \(h_l(x_l) = g_l(x_l^{(N)}) + B_l(x_l^{(N)}) x_l^{(L)}\). Moreover, since EKF is employed in the TF algorithms developed in the following, the linearized versions of \(f_l\) and \(g_l\) are also considered; these can be expressed as (e.g., see \(\text{II}\) pp. 194-195)

\[
x_{l+1} = F_l x_l + u_l + w_l
\]

and

\[
y_l = H_l f_l x_l + v_l + e_l,
\]

respectively; here, \(F_l \triangleq \left[\partial f_l(x)/\partial x\right]_{x = x_{f_{l-1}}, t}|_{l}, x_{f_{l-1}}\) is the (forward) estimate of \(x_l\) evaluated by EKF in its \(l\)-th recursion, \(u_l \triangleq f_l(x_{f_{l-1}}) - F_l x_{f_{l-1}}, H_l f_l \triangleq \left[\partial h_l(x)/\partial x\right]_{x = x_{f_{l-1}}, t}|_{l}\) is the (forward) prediction \(x_l\) computed by EKF in its \((l-1)\)-th recursion and \(v_l \triangleq h_l(x_{f_{l-1}}) - H_l f_l x_{f_{l-1}}\).

In the following Section we focus on the so-called filtering problem, which concerns the evaluation of the posterior pdf \(f(x_l|y_{1:l})\) at an instant \(t \geq 1\),

\footnote{The covariance matrix \(C_{w}\) of \(w_l\) can be easily computed on the basis of the matrices \(C_{w}^{(L)}\) and \(C_{w}^{(N)}\).}
given a) the initial pdf \( f(x_1) \) and b) the \( t \cdot P \)-dimensional measurement vector \( y_{1:t} = [y_1^T, y_2^T, \ldots, y_t^T]^T \).

### 3 Graphical Modelling for Turbo Filtering

Let us consider first a SSM described by the Markov model \( f(x_{l+1}|x_l) \) and the observation model \( f(y_l|x_l) \) for any \( l \). In this case, the computation of the posterior pdf \( f(x_t|y_{1:t}) \) for \( t \geq 1 \) can be accomplished by means of an exact Bayesian recursive procedure, consisting of a measurement update (MU) step followed by a time update (TU) step. Following [16, Sec. II, p. 1297], the equations describing the \( l \)-th recursion of this procedure (with \( l = 1, 2, \ldots, t \)) can be easily obtained by applying the SPA to the Forney-style FG shown in Fig. 1, if the joint pdf \( f(x_t, y_{1:t}) \) is considered in place of the associated a posteriori pdf \( f(x_t|y_{1:t}) \).

In fact, given the measurement message \( \tilde{m}_{ms}(x_l) = f(y_l|x_l) \), if the input message \( \tilde{m}_{fp}(x_l) = f(x_l, y_{1:(l-1)}) \) enters this FG, the message going out of the equality node is given by

\[
\tilde{m}_{fe}(x_l) = \tilde{m}_{fp}(x_l) \tilde{m}_{ms}(x_l) = f(x_l, y_{1:(l-1)}) f(y_l|x_l) = f(x_l, y_{1:l}) \tag{7}
\]

and, consequently, the message emerging from the function node referring to the pdf \( f(x_{l+1}|x_l) \) is expressed by

\[
\int f(x_{l+1}|x_l) \tilde{m}_{fe}(x_l) \, dx_l = f(x_{l+1}, y_{1:l}) = \tilde{m}_{fp}(x_{l+1}) . \tag{8}
\]

Eqs. (7) and (8) express the MU and the TU, respectively, that need to be accomplished in the \( l \)-th recursion of Bayesian filtering.

Let us see now how the FG illustrated in Fig. 1 can be exploited to devise a graphical model efficiently representing the TF concept. As already stated in the Introduction, any TF scheme results from the parallel concatenation of two constituent Bayesian filters (denoted \( F_1 \) and \( F_2 \) in the following), that can iteratively improve their accuracy through the exchange of their statistical information. In practice, in developing TF techniques, the following general rules are followed:

**R1** the constituent filters operate on partially overlapped portions of system state;

**R2** the filter \( F_1 \) (\( F_2 \)) is the core of a processing module (called soft-in soft-out, SISO, module in the following) receiving statistical information from \( F_2 \) (\( F_1 \)) and generating new statistical information useful to \( F_2 \)

---

\[\text{In the following the acronyms } \text{fp and fe are employed in the subscripts of various messages, so that readers can easily understand their meaning; in fact, the messages these acronyms refer to represent a form of one-step forward prediction and of forward estimation, respectively.}\]
Figure 1: Factor graph representing the $l$-th recursion of Bayesian filtering for a SSM described by the Markov model $f(x_{l+1}|x_l)$ and the observation model $f(y_l|x_l)$; the SPA message flow is indicated by green arrows.

(F$_1$); R3) each constituent filter relies on exact Markov/observation models or approximate (e.g., linearized) versions of them. These rules can be motivated and implemented as follows. The first rule (i.e., R1) ensures that any TF filtering algorithm contains a form of redundancy, that represents the first of the two fundamental properties characterizing each error correction method employed in digital communications [13]. In our general description of a TF scheme, it is assumed that (see Fig. 2(a)): 1) filter $F_1$ ($F_2$) estimates the state vector $\tilde{x}_l$ ($\hat{x}_l$) of size $\tilde{D}$ ($\hat{D}$), with $\tilde{D} \leq D$ ($\hat{D} \leq D$); 2) the portion $\tilde{x}_l$ ($\hat{x}_l$) of $x_l$ not included in $\tilde{x}_l$ ($\hat{x}_l$) is contained in (or at most coincides with) $\tilde{x}_l$ ($\hat{x}_l$). This entails that: a) an overall estimate of the system state $x_l$ can be generated on the basis of the posterior pdfs of $\tilde{x}_l$ and $\hat{x}_l$ evaluated by $F_1$ and $F_2$, respectively; b) the portion $[x_{D-\tilde{D}+1,l}, x_{D-\hat{D}+1,l}, ..., x_{D-1,l}]^T$ of $x_l$, consisting of

$$N_d \triangleq \tilde{D} + \hat{D} - D$$

elements, is estimated by both $F_1$ and $F_2$. Consequently, rule R1 requires the parameter $N_d$ (9), that represents the degree of redundancy of the overall filtering algorithm, to be strictly positive.

The second rule (i.e., R2) has been inspired by the fact that, generally speaking, iterative decoders of concatenated channel codes are made of multiple SISO modules, one for each constituent code. The implementation of this rule in TF requires accurately defining the nature of the statistical information to be passed from each constituent filter to the other one. Actually, this problem has been already tackled in the development of MPF, where the information passed from a particle filter to a bank of Kalman filters takes the form of pseudo-measurements (PMs) evaluated on the basis of the mathematical constraints established by state update equations [9]. The use of PMs allows us to exploit the memory characterizing the time evolution of dynamic models (and
representing the second fundamental property of each error correction method employed in digital communications). Moreover, PMs can be processed as they were real measurements \[9\]; for this reason, their use can be incorporated in the FG shown in Fig. 1 by including a new MU, i.e. by adding a new equality node through which the message emerging from the first MU (i.e., from the MU based on real measurements) is merged with a message conveying PM information. This idea is implemented in the graphical model shown in Fig. 2(b) and providing a detailed description of the overall processing accomplished by a SISO module based on \(F_1\) (a similar model can be easily drawn for \(F_2\) by interchanging the couple \((\hat{x}_i, \bar{x}_i)\) with \((\bar{x}_i, \hat{x}_i)\) in that figure). In fact, this model represents the \(F_1\) filtering algorithm (\(F_1\) block), the conversion of the statistical information provided from \(F_2\) into a form useful to \(F_1\) (\(F_1\)-IN block) and the generation of the statistical information made available by \(F_1\) to \(F_2\) (\(F_1\)-OUT block). Its structure can be explained as follows:

1. The algorithm employed by \(F_1\) is based on the Markov model \(\hat{f}(\bar{x}_{t+1}|\bar{x}_t, \hat{x}_t)\) and on the observation model \(\hat{f}(y_t|\bar{x}_t, \hat{x}_t)\), that represent the exact models \(f(\bar{x}_{t+1}|\bar{x}_t, \hat{x}_t)\) and \(f(y_t|\bar{x}_t, \hat{x}_t)\), respectively, or approximations of one or both of them (as required by the third rule, i.e. by R3). The pdf of the state component \(\bar{x}_t\) (unknown to \(F_1\)) is provided by \(F_2\) through the message \(\bar{m}_{fe}l(\bar{x}_t)\). Moreover, as already stated above, the forward estimate of \(\bar{x}_t\) is computed by \(F_1\) in two distinct MU steps, the first one involving the message \(\bar{m}_{ms}(\hat{x}_t)\) (based on the measurement \(y_t\)), the second one involving the message \(\bar{m}_{pm}(\hat{x}_t)\) (conveying the PM information computed by \(F_2\)); these steps generate the messages \(\bar{m}_{fe1}(\bar{x}_t)\) and \(\bar{m}_{fe2}(\bar{x}_t)\), respectively.

2. The forward estimate \(\bar{m}_{fe2}(\hat{x}_t)\) computed by \(F_1\) is passed to \(F_2\) together with the PM message \(\bar{m}_{pm}(\hat{x}_t)\). The last message is evaluated on the basis of the messages \(\bar{m}_{fe1}(\bar{x}_t)\) and \(\bar{m}_{fe2}(\hat{x}_t)\), i.e. on the basis of the forward estimates available before and after the second MU of \(F_1\). Note also that the computation of \(\bar{m}_{pm}(\hat{x}_t)\) is carried out in the block called PM generation (PMG) inside the \(F_1\)-OUT block.

3. The statistical information made available by \(F_2\) to \(F_1\) is condensed in the messages \(\bar{m}_{fe2}(\hat{x}_t)\) and \(\bar{m}_{pm}(\hat{x}_t)\). The message \(\bar{m}_{fe2}(\hat{x}_t)\) acquired by \(F_1\) can be computed by marginalizing the message \(\bar{m}_{fe2}(\hat{x}_t)\), since, generally speaking, \(\hat{x}_t\) is a portion of \(\bar{x}_t\) (marginalization is accomplished in block labelled with the letter M in Fig. 2(b)); moreover, \(\bar{m}_{fe2}(\hat{x}_t)\) is processed jointly with \(\bar{m}_{pm}(\hat{x}_t)\) to generate the PM message \(\bar{m}_{pm}(\hat{x}_t)\) (this is accomplished in the block called

---

Note that oriented edges are used in our graphical models wherever message passing along such edges can be accomplished along a single direction only.
PM conversion, PMC, inside the F1-IN block).

Merging the graphical model shown in Fig. 2-(b) with its counterpart referring to F2 results in the PC architecture shown in Fig. 3. This model, unlike the one illustrated in Fig. 1, is not cycle free. For this reason, generally speaking, the application of the SPA to it leads to iterative algorithms with no natural termination and whose accuracy can be substantially influenced by the adopted message scheduling [16], [17]. This consideration and the possibility of choosing different options for F1 and F2 lead easily to the conclusion that the graphical models shown in Figs. 2-(b) and 3 can be employed to develop an entire family of filtering algorithms, called turbo filters.

In the remaining part of this manuscript we focus on a specific instance of the proposed PC architecture, since we make specific choices for both the SSM and the two filters. In particular, we focus on the CLG SSM described in Section 2 and assume that F1 is an extended Kalman filter operating over the whole system state (so that \( \hat{x}_l = x_l \) and \( \tilde{x}_l = \emptyset \)), whereas F2 is a particle filter (in particular, a sequential importance resampling, SIR, filter [1]) operating on the nonlinear state component only (so that \( \hat{x}_l = x^{(N)}_l \) and \( \tilde{x}_l = x^{(L)}_l \)); note that, in this case, the degree of redundancy is \( N_d = D_N \) (see [9]). Our choices aim at developing a new concatenated filtering algorithm in which an extended Kalman filter is aided by a particle filter in its most difficult task, i.e. in the estimation of the nonlinear state component. Moreover, the proposed TF scheme can be easily related to MPF, since the last technique can be considered as a form of serial concatenation of PF with Kalman filtering. However, our TF instance employs, unlike MPF, a single (extended) Kalman filter in place of a bank of Kalman filters; moreover, such a filter estimates the whole system state, instead of its nonlinear component only. Based on the general models shown in Figs. 2-(b) and 3, the specific graphical model illustrated in Fig. 4 can be drawn for the considered case. This model deserves the following comments:

1. The upper (lower) rectangle delimited by a grey line allow to easily identify the message passing accomplished by EKF (PF).
2. Filter F1 is based on the approximate models \( \hat{f}(x_{l+1}|x_l) \) and \( \hat{f}(y_l|x_l) \), that can be easily derived from the linearised eqs. (5) and (6), respectively. Moreover, the (Gaussian) messages processed by it are \( \tilde{m}_{fp}(x_l), \tilde{m}_{ms}(x_l), \tilde{m}_{fe1}(x_l), \tilde{m}_{pm}(x_l), \tilde{m}_{fe2}(x_l) \) and \( \tilde{m}_{fp}(x_{l+1}) \), and are denoted FP, MS, FE1, PM, FE2 and FP’, respectively, to ease reading.
3. Filter F2 is based on the exact models \( f(x^{(N)}_{l+1}|x^{(N)}_l, x^{(L)}_l) \) and \( f(y_l|x^{(N)}_l, x^{(L)}_l) \), that can be easily derived from the eqs. (1) (with \( Z = N \)) and (2), respectively. Moreover, the messages processed by it and appearing in Fig. 4 refer to the
Figure 2: a) Partitioning adopted for the system state $x_i$ in the PC of two filtering algorithms; b) Graphical model referring to a SISO module based on $F_1$. Black and blue (red) lines are used to identify the edges and the blocks related to filtering and processing of information coming from $F_2$ (to be delivered to $F_2$), respectively.
Figure 3: Parallel concatenation of SISO modules based on filters $F_1$ and $F_2$; the flow of the messages exchanged between them is indicated by green arrows.
j-th particle predicted in the previous (i.e. (l - 1)-th) recursion and denoted $x_{fp,i}^{(N)}$, with $j = 0, 1, \ldots, N_p - 1$ (where $N_p$ represents the overall number of particles); such messages are $\bar{m}_{fp,j}(x_i^{(N)})$, $\bar{m}_{ms,j}(x_i^{(N)})$, $\bar{m}_{fe1,j}(x_i^{(N)})$, $\bar{m}_{pm,j}(x_i^{(N)})$, $\bar{m}_{fe2,j}(x_i^{(N)})$, and $\bar{m}_{fp,j}(x_i^{(N)1})$, and are denoted $FPN_j$, $MSN_j$, $FEN_{1,j}$, $PMN_j$, $FEN_{2,j}$ and $FPN_j$, respectively, to ease reading.

4. The message $\bar{m}_{fe1}(x_i) (\bar{m}_{fe2}(x_i))$ generated by $F_1$ undergoes marginalization in the block labelled with the letter M; this results in the message $\bar{m}_{fe1}(x_i^{(L)}) (\bar{m}_{fe2}(x_i^{(L)}))$, denoted $FEL1$ ($FEL2$). Based on the general model shown in Fig. 2-b), we exploit the messages $\bar{m}_{fe1}(x_i^{(L)})$ and $\bar{m}_{fe2}(x_i^{(L)})$ to compute the PM message $\bar{m}_{pm,j}(x_i^{(N)})$ (denoted $PMN_j$) in the block called PMG\textsubscript{EKF}. Moreover, $\bar{m}_{fe2}(x_i^{(L)})$ is employed for marginalising the PF state update and measurement models (i.e., $f(x_{i+1}^{(N)}|x_i^{(N)})$, $x_i^{(L)}$ and $f(y_i|x_i^{(N)}, x_i^{(L)})$, respectively); this allows us to compute the messages $\bar{m}_{ms,j}(x_i^{(N)})$ and $\bar{m}_{fp,j}(x_i^{(N)})$, respectively.

5. The message $\bar{m}_{fe2,j}(x_i^{(N)})$ produced by PF is processed in the block called PMG\textsubscript{PF} in order to generate the PM message $\bar{m}_{pm,j}(x_i^{(L)})$ (the message $\bar{m}_{fe1,j}(x_i^{(N)})$ is not required in this case; see the next Section). Moreover, the two sets $\{\bar{m}_{pm,j}(x_i^{(L)})\}$ and $\{\bar{m}_{fe2,j}(x_i^{(N)})\}$ (each consisting of $N_p$ messages) are merged in the block called PMC\textsubscript{PF}, where the information they convey are converted into the (single) PM message $\bar{m}_{pm}(x_i)$ feeding $F_1$.

6. At the end of the l-th recursion, a single statistical model is available for $x_i^{(L)}$. On the contrary, two models are available for $x_i^{(N)}$, one particle-based, the other one Gaussian, since this state component is shared by $F_1$ and $F_2$; note that the former model, unlike the second one, is able to represent a multimodal pdf.

Let us now focus on the evaluation of the PMs for the considered TF scheme. On the one hand, the PM messages $\{\bar{m}_{pm,j}(x_i^{(N)})\}$ evaluated for $F_2$ are exploited to improve the estimation accuracy for the nonlinear state component only. Their computation involves the pdf of the random vector

$$z_i^{(N)} \triangleq x_{i+1}^{(L)} - A_i^{(L)} \left(x_i^{(N)} \right) x_i^{(L)},$$

defined on the basis of the state update equation (1) (with $Z = L$). This pdf need to be evaluated for each of the $N_p$ particles representing $x_i^{(N)}$; in the following, its expression associated with the j-th particle (i.e., conditioned on $x_i^{(N)} = x_{fp,j}^{(N)}$) and evaluated on the basis of the joint pdf of $x_i^{(L)}$ and $x_i^{(L)}$ provided by $F_1$ is conveyed by the message $\bar{m}_j(z_i^{(N)})$. Note also that, based on
Figure 4: Parallel concatenation of an extended Kalman filter with a particle filter.
\[ z_i^{(N)}(10) \] is expected to equal the sum
\[ f_i^{(L)}(x_i^{(N)}) + w_i^{(L)}, \]
that depends on \( x_i^{(N)} \) only; the pdf of \( z_i^{(N)} \) evaluated on the basis of (11) is denoted \( f(z_i^{(N)}|x_i^{(N)}) \) in the following.

On the other hand, the PM message \( \vec{m}_{pm}(x_i) \) evaluated for \( F_1 \) is expected to improve the estimation accuracy for the whole state. For this reason, in our TF techniques, its computation involves the two message sets \( \{\vec{m}_{pm,j}(x_i^{(L)})\} \) and \( \{\vec{m}_{fe,j}(x_i^{(N)})\} \), generated by \( F_2 \) and referring to the two distinct components of \( x_i \). The messages \( \{\vec{m}_{fe,j}(x_i^{(N)})\} \) convey a particle-based representation of \( x_i^{(N)} \). The message \( \vec{m}_{pm,j}(x_i^{(L)}) \), instead, represents the pdf of the random vector \( z_i^{(L)} \) conditioned on \( x_i^{(N)} = x_{i+1}^{(N)} \) for any \( j \). This pdf is evaluated on the basis of the joint representation of the couple \( (x_i^{(N)}, x_{i+1}^{(N)}) \) produced by \( F_2 \) and is conveyed by the message \( \vec{m}_j(z_i^{(N)}) \); note also that, based on (1) (with \( Z = N \)), the quantity \( z_i^{(L)}(12) \) is expected to equal the sum
\[ A_i^{(N)}(x_i^{(N)}) x_i^{(L)} + w_i^{(N)}, \]
that depends on \( x_i^{(L)} \) and \( x_i^{(N)} \) only; the pdf of \( z_i^{(N)} \) evaluated on the basis of (13) is denoted \( f(z_i^{(L)}|x_i^{(L)}, x_i^{(N)}) \) in the following.

Two specific message scheduling for the graphical model shown in Fig. 4 are proposed in the following Section, where the computation of all the involved messages is also analysed in detail.

4 Message Passing in Turbo Filtering

In this Section two different options are considered for the scheduling of the messages appearing in Fig. 4. The first option consists in running EKF before PF within each iteration, whereas the second one in doing the opposite; the resulting algorithms are dubbed TF#1 and TF#2, respectively. The message scheduling adopted in TF#1 is represented in Fig. 5 that refers to the \( k \)-th iteration accomplished within the \( l \)-th recursion (with \( k = 1, 2, ..., N_{it} \), where \( N_{it} \) is the overall number of iterations); this explains why the superscripts \( (k) \) and \( (k-1) \) have been added to all the iteration-dependent messages appearing in Fig. 4.
Figure 5: Message scheduling adopted in TF#1.
As far as the evaluation of the messages passed in TF#1 and TF#2 is concerned, this is mainly based on three computational rules (CR) resulting from the application of the SPA to equality nodes and function nodes. More specifically, the first computational rule, denoted CR1, applies to an equality constraint node; if the messages $\vec{m}_1(x)$ and $\vec{m}_2(x)$ denote the messages entering it, the message $\vec{m}_3(x) = \vec{m}_1(x) \vec{m}_2(x)$ emerges from it. In particular, if $\vec{m}_i(x) = \mathcal{N}(x; \eta_i, C_i)$ (with $i = 1$ and 2), then $\vec{m}_3(x) = \mathcal{N}(x; \eta_3, C_3)$; moreover, the precision matrix $W_3$ and the transformed mean vector $w_3$ associated with $C_3$ and $\eta_3$, respectively, are given by (see [16, Table 2, p. 1303, eqs. (II.1) and (II.2)])

$$W_3 \triangleq C^{-1}_3 = W_1 + W_2$$  \hspace{1cm} (14)

and

$$w_3 \triangleq C^{-1}_3 \eta_3 = w_1 + w_2$$  \hspace{1cm} (15)

respectively, where $W_i \triangleq C^{-1}_i$ and $w_i \triangleq C^{-1}_i \eta_i$ for $i = 1, 2$. The second computational rule, denoted CR2, applies to a node representing the function $f(x_1, x_2)$; if the message $\vec{m}_1(x_1)$ denotes the message entering it, the message $\vec{m}_2(x_2)$ emerging from it is given by

$$\vec{m}_2(x_2) = \int \vec{m}_1(x_1) f(x_1, x_2) dx_1.$$  \hspace{1cm} (16)

In particular, if $\vec{m}_1(x_1) = \mathcal{N}(x_1; \eta_1, C_1)$ and $f(x_1, x_2) = \mathcal{N}(x_2; Ax_1 + b, C)$, then

$$\vec{m}_2(x_2) = \mathcal{N}(x_2; \eta_2, C_2),$$  \hspace{1cm} (17)

with $\eta_2 = A \eta_1 + b$ and $C_2 = C + AC_1 (A)^T$ (see [16, Table 2, p. 1303, eqs. (II.7) and (II.9); Table 3, p. 1304, eqs. (III.1) and (III.3)]). Finally, the third computational rule, denoted CR3, applies to a node representing the function $f(x) = \mathcal{N}(x; \eta_2, C_2)$ and fed by the message $\vec{m}_1(x) = \mathcal{N}(x; \eta_1, C_1)$; the output message is the constant message

$$\vec{m}_2 = D \exp \left[ \frac{1}{2} (\eta^T W \eta - \eta_1^T W_1 \eta_1 - \eta_2^T W_2 \eta_2) \right]$$  \hspace{1cm} (18)

where $W_1 \triangleq C^{-1}_1$, $W_2 \triangleq C^{-1}_2$, $W = W_1 + W_2$, $W \eta = W_1 \eta_1 + W_2 \eta_2$, $D = (\det [C_1 + C_2])^{-N/2}$ and $N$ is the size of $x$.

In the following we show how, applying the above mentioned CRs, simple formulas can be derived for all messages passed in the graphical model shown in Fig. 5. However, before doing this, we need to define the input messages for the considered recursion; these are

$$\vec{m}_{fp} (x_i) = \mathcal{N}(x_i; \eta_{fp,i}, C_{fp,i})$$  \hspace{1cm} (19)
for the EKF (upper part of the graphical model) and the set of $N_p$ messages \{$\vec{m}_{fp,j}(x_{l}^{(N)})$\} for the PF (lower part of the graphical model), where

$$m_{fp,j}(x_{l}^{(N)}) = \delta \left( x_{l}^{(N)} - x_{fp,l,j}^{(N)} \right),$$  \hspace{1cm} (20)

with $j = 0, 1, ..., N_p - 1$; in the following we also assume that the $N_p$ available particles are collected in the set $S_l \triangleq \{x_{fp,l,j}^{(N)}\}$. On the other hand, the output messages are $\vec{m}_{fp}(x_{l+1})$ (for EKF) and $\{\vec{m}_{fp,j}(x_{l+1}^{(N)})\}$ (for PF); since, as shown below, the devised TF algorithms preserve the mathematical structure of the filtered densities from recursion to recursion, $\vec{m}_{fp}(x_{l+1})$ and $\vec{m}_{fp,j}(x_{l+1}^{(N)})$ have the same functional form as $\vec{m}_{fp}(x_{l})$ \[19\] and $\vec{m}_{fp,j}(x_{l}^{(N)})$ \[20\] (for any $j$), respectively.

It is also worth mentioning that not all the messages appearing in Fig. 5 depend on the iteration index $k$. More specifically, the following messages are computed only once:

1. The messages $\vec{m}_{fe1}(x_{l})$ and $\vec{m}_{fe1}(x_{l}^{(L)})$ evaluated by EKF in its first MU. In particular, $\vec{m}_{fe1}(x_{l})$ is computed as (see Fig. 5)

$$\vec{m}_{fe1}(x_{l}) = \vec{m}_{fp}(x_{l}) \vec{m}_{ms}(x_{l}),$$  \hspace{1cm} (21)

where $\vec{m}_{ms}(x_{l})$ is the message conveying the information provided by $y_l$, whose statistical representation is expressed by the pdf $\tilde{f}(y_l | x_{l})$ (resulting from the linearised equation \[6\]); therefore, it can be expressed as

$$\vec{m}_{ms}(x_{l}) = \mathcal{N}(y_l; H_l^T x_{l} + v_l, C_e),$$  \hspace{1cm} (22)

or, equivalently, as (see \[16\], Table 3, p. 1304, eqs. (III.5) and (III.6))

$$\vec{m}_{ms}(x_{l}) = \mathcal{N}(x_{l}; \eta_{ms,l}, C_{ms,l});$$  \hspace{1cm} (23)

here, the covariance matrix $C_{ms,l}$ and the mean vector $\eta_{ms,l}$ can be evaluated from the associated precision matrix

$$W_{ms,l} \triangleq (C_{ms,l})^{-1} = H_l W_e H_l^T,$$  \hspace{1cm} (24)

and the transformed mean vector

$$w_{ms,l} \triangleq W_{ms,l} \eta_{ms,l} = H_l W_e (y_l - v_l),$$  \hspace{1cm} (25)

respectively, and $W_e \triangleq C_e^{-1}$. Therefore, $\vec{m}_{fe1}(x_{l})$ \[21\] can be put in the form

$$\vec{m}_{fe1}(x_{l}) = \mathcal{N}(x_{l}; \eta_{fe1,l}, C_{fe1,l}),$$  \hspace{1cm} (26)
where the covariance matrix $C_{fe1,l}$ and the mean vector $\eta_{fe1,l}$ can be evaluated from the associated precision matrix (see CR1, eq. (14))

$$W_{fe1,l} \triangleq (C_{fe1,l})^{-1} = W_{fp,l} + W_{ms,l}$$

and the transformed mean vector (see CR1, eq. (15))

$$w_{fe1,l} \triangleq w_{fe1,l} + w_{ms,l},$$

respectively; here, $W_{fp,l} \triangleq (C_{fp,l})^{-1}$ and $W_{ms,l} \triangleq W_{fp,l} + W_{ms,l}$. The message $\tilde{m}_{fe1}(x_{f1}^{(L)})$, instead, is easily obtained from $\tilde{m}_{fe1}(x_{1})$ (26) by marginalizing the last message with respect to $x^{(N)}$; this produces

$$\tilde{m}_{fe1}(x_{1}^{(L)}) = \int \tilde{m}_{fe1}(x_{1}) dx_{1}^{(N)} = N(x_{1}^{(L)}; \tilde{\eta}_{fe1,l}, \tilde{C}_{fe1,l}),$$

where $\tilde{\eta}_{fe1,l}$ and $\tilde{C}_{fe1,l}$ are extracted from the mean $\eta_{fe1,l}$ and the covariance matrix $C_{fe1,l}$ of $\tilde{m}_{fe1}(x_{1})$, respectively, since $x_{1}^{(L)}$ consists of the first $D_{L}$ elements of $x_{1}$.

2. The output messages $\tilde{m}_{fp}(x_{t+1})$ and $\tilde{m}_{fp,j}(x_{t+1}^{(N)})$ (for any $j$), since they are evaluated on the basis of the forward estimates $\tilde{m}_{fp}(x_{1})$ and $\tilde{m}_{fp,j}(x_{1}^{(N)})$ computed by EKF and PF, respectively, in the last iteration.

In the following, a detailed description of the messages passed in TF#1 is provided. The formulas derived for this algorithm can be easily re-used in the computation the messages passed in TF#2; for this reason, after developing TF#1, we limit to providing a brief description of the scheduling adopted in TF#2.

The scheduling illustrated in Fig. 5 for TF#1 consists in computing the involved (iteration-dependent) messages according to the following order: 1) $\tilde{m}_{fe1}(x_{1})$, $\tilde{m}_{fe2}(x_{1}^{(L)})$; 2) $\{\tilde{m}_{ms,j}(x_{1}^{(N)})\}, \{\tilde{m}_{fe1,j}(x_{1}^{(N)})\}$; 3) $\{\tilde{m}_{pm,j}(x_{1}^{(N)})\}, \{\tilde{m}_{fe2,j}(x_{1}^{(N)})\}$; 4) $\{\tilde{m}_{pm,j}(x_{1}^{(L)})\}, \tilde{m}_{pm}(x_{1})$. Therefore, the evaluation of these messages can be organized according to the four steps described below and to be carried out for $k = 1, 2, ..., N_{lt}$. Note that in our description of TF#1 scheduling, particle-dependent messages always refer to the $j$-th particle (with that $j = 0, 1, ..., N_{p} - 1$) and that, generally speaking, the structure of the particle set changes from iteration to iteration, even if it preserves its cardinality; moreover, the particle set available at the beginning of the $k$-th iteration is $S_{k}^{(k-1)} = \{x_{fp,j}^{(N)}[k-1], j = 0, 1, ..., N_{p} - 1\}$, with $S_{t}^{(0)} = S_{t}$ and $x_{fp,j}^{(N)}[0] = x_{fp,l,j}^{(N)}$.

1. Second MU in EKF - This step aims at updating our statistical knowledge about $x_{1}$ on the basis of the PM information conveyed by the message $\tilde{m}_{pm}^{(k-1)}(x_{1})$. Therefore, the evaluation of these messages can be organized according to the four steps described below and to be carried out for $k = 1, 2, ..., N_{lt}$. Note that in our description of TF#1 scheduling, particle-dependent messages always refer to the $j$-th particle (with that $j = 0, 1, ..., N_{p} - 1$) and that, generally speaking, the structure of the particle set changes from iteration to iteration, even if it preserves its cardinality; moreover, the particle set available at the beginning of the $k$-th iteration is $S_{k}^{(k-1)} = \{x_{fp,j}^{(N)}[k-1], j = 0, 1, ..., N_{p} - 1\}$, with $S_{t}^{(0)} = S_{t}$ and $x_{fp,j}^{(N)}[0] = x_{fp,l,j}^{(N)}$. 

17
respectively; here, $W$ and (see CR1, eq. (15)) $eq. (14))$

variance matrix $\tilde{\eta}$ where $C$ covariance matrix $\eta$

This is carried out by computing the new message (see Fig. 5)

\[ \vec{m}_{fe2}(x_l) = \vec{m}_{pm}^{(k-1)}(x_l) \vec{m}_{fe1}(x_l), \]  

(30)

where $\vec{m}_{fe1}(x_l)$ is expressed by (26), and $\vec{m}_{pm}^{(k-1)}(x_l)$ is equal to unity for $k = 1$

(because of the adopted scheduling) and is given by (51) for $k > 1$. Consequently,

\[ \vec{m}_{fe2}(x_l) = N(x_l; \eta_{fe2,l}^{(k)}; C_{fe2,l}^{(k)}) , \]  

(31)

where $\eta_{fe2,l}^{(k)} = \eta_{fe1,l}$ and $C_{fe2,l}^{(k)} = C_{fe1,l}$ for $k = 1$, whereas, for $k > 1$, the covariance matrix $C_{fe2}$ and the mean vector $\eta_{fe2}$ are evaluated as (see CR1, eq. (14))

\[ C_{fe2,l}^{(k)} = W_{l}^{(k)} C_{pm,l}^{(k-1)} \]  

(32)

and (see CR1, eq. (15))

\[ \eta_{fe2,l}^{(k)} = W_{l}^{(k-1)} [C_{pm,l}^{(k-1)} W_{fe1,l} + \eta_{pm,l}^{(k-1)}] , \]  

(33)

respectively; here, $W_{l}^{(k-1)} \triangleq \left[ C_{pm,l}^{(k-1)} W_{fe1,l} + I_D \right]^{-1}$. Marginalizing the message $\vec{m}_{fe2}(x_l)$ (31) with respect to $x_l^{(N)}$ results in the message

\[ \vec{m}_{fe2}^{(k)}(x_l) \triangleq \int \vec{m}_{fe2}(x_l) dx_l^{(N)} = N(x_l^{(L)}; \tilde{\eta}_{fe2,l}^{(k)}, \tilde{C}_{fe2,l}^{(k)}), \]  

(34)

where $\tilde{\eta}_{fe2,l}^{(k)}$ and $\tilde{C}_{fe2,l}^{(k)}$ are easily extracted from the mean $\eta_{fe2,l}^{(k)}$ and the covariance matrix $C_{fe2,l}^{(k)}$ of $\vec{m}_{fe2}(x_l)$ (31), respectively, since $x_l^{(L)}$ consists of the first $D_L$ elements of $x_l$.

2. First MU in PF - This step aims at updating the weight of the $j$-th particle $x_{fp,l,j}^{(N)}[k - 1]$, conveyed by the message (see (20))

\[ \vec{m}_{fp,j}(x_l^{(N)}) = \delta(x_l^{(N)} - x_{fp,l,j}^{(N)}[k - 1]), \]  

(35)

on the basis of the new measurements $y_l$. It involves the computation of the messages $\vec{m}_{ms,j}^{(k)}(x_l^{(N)})$ and (see Fig. 5)

\[ \vec{m}_{fe1,j}^{(k)}(x_l^{(N)}) = \vec{m}_{ms,j}^{(k)}(x_l^{(N)}) \vec{m}_{fp,j}^{(k)}(x_l^{(N)}). \]  

(36)

The evaluation of the message $\vec{m}_{ms,j}^{(k)}(x_l^{(N)})$ requires marginalizing the measurement model $f_l(y_l|x_l^{(N)}, x_l^{(L)})$ with respect to $x_l^{(N)}$ (see Fig. 5), whose pdf is provided by the message $\vec{m}_{fe2}(x_l^{(L)})$ (34). Therefore, the message $\vec{m}_{ms,j}^{(k)}(x_l^{(N)})$
emerging from the function node representing \( f(y_i|x_i^N) \), \( x_i^L \) is given by

\[
\mathcal{m}_{ms}(x_i^N) = \mathcal{N}(y_i; \bar{m}_{ms}(x_i^N), \bar{C}_{ms}(x_i^N)) \tag{37}
\]

Based on CR2, it is easy to show that

\[
\mathcal{m}_{ms}(x_i^N) = \mathcal{N}(y_i; \eta_{ms,l}(x_i^N), \bar{C}_{ms,l}(x_i^N)) \tag{38}
\]

where \( \eta_{ms,l}(x_i^N) \equiv B_l(x_i^N) \eta_{fe2,l} + g_l(x_i^N) \) and \( \bar{C}_{ms,l}(x_i^N) \equiv B_l(x_i^N) \bar{C}_{fe2,l} B_l^T(x_i^N) + C_e \). Then, substituting (35) and (38) in (36) yields

\[
\mathcal{m}_{fe1,l,j}(x_i^N) = w_{fe1,l,j}(k) \delta \left( x_i^N - x_{fp,l,j}[k-1] \right) \tag{39}
\]

where \( w_{fe1,l,j}(k) \) is the new particle weight combining the a priori information about \( x_i^N \) with the information provided by the new measurement; here,

\[
\eta_{ms,l,j}(k) \equiv \eta_{ms,l}(x_{fp,l,j}[k-1]) = B_{l,j}[k] \eta_{fe2,l} + g_{l,j} \tag{41}
\]

and

\[
\bar{C}_{ms,l,j}(k) \equiv \bar{C}_{ms,l}(x_{fp,l,j}[k-1]) = B_{l,j}[k] \bar{C}_{fe2,l} (B_{l,j}[k])^T + C_e \tag{42}
\]

with \( g_{l,j}[k] \equiv g_l(x_{fp,l,j}[k-1]) \) and \( B_{l,j}[k] \equiv B_l(x_{fp,l,j}[k-1]) \).

3. Computation of the PMs for PF and second MU in PF - This step aims at updating the weight of the \( j \)-th particle \( x_{fp,l,j}[k-1] \) (provided by the message \( \mathcal{m}^{(k)}_{fe1,l,j}(x_i^N) \) \( \tag{39} \)) on the basis of the PM \( z_i^N \) \( \tag{10} \). It involves the computation of the PM message \( \mathcal{m}^{(k)}_{pm,j}(x_i^N) \) and of the message (see Fig. 5)

\[
\mathcal{m}^{(k)}_{fe2,l,j}(x_i^N) = \mathcal{m}_{fe1,l,j}(x_i^N) \mathcal{m}^{(k)}_{pm,j}(x_i^N) \tag{43}
\]

The algorithm for computing \( \mathcal{m}^{(k)}_{pm,j}(x_i^N) \) is executed in the PMG\(_{EKF} \) block shown in Figs. 4-5 and is described in detail in Appendix A where it is shown

\( ^4 \)In evaluating the weight \( w_{fe1,l,j}(k) \) \( \tag{40} \), the factor \( [\det(\bar{C}_{ms,l,j}(k))]^{-1/2} \) appearing in the expression of the involved Gaussian pdf is neglected in our simulations, since this entails a negligible loss in estimation accuracy. Similar comments apply to the factor \( \bar{D}^{(k)}_{pm,l,j} \) appearing in the weight \( w_{pm,l,j}(k) \) \( \tag{44} \).
that
\[ w_{pm,l,j}^{(k)} = \tilde{w}_{pm,l,j}^{(k)}(x_i^{(N)}) \]
\[ = \tilde{D}_{pm,l,j}(\exp \left[ \frac{1}{2} \left( \eta_{pm,l,j}^{(k)} \right)^T \hat{W}_{pm,l,j}^{(k)} \eta_{pm,l,j}^{(k)} \right] - \left( \eta_{\tilde{z},l,j}^{(k)} \right)^T \hat{W}_{z_{\tilde{z},l,j}}^{(k)} \eta_{\tilde{z},l,j}^{(k)} - \left( f_{l,j}^{(L)}[k] \right)^T W_w^{(L)} f_{l,j}^{(L)}[k] ) \right]; \quad (44) \]

here
\[ \hat{W}_{pm,l,j}^{(k)} \triangleq \left( \hat{C}_{pm,l,j}^{(k)} \right)^{-1} = \hat{W}_{z_{\tilde{z},l,j}}^{(k)} + W_w^{(L)} f_{l,j}^{(L)} \quad (45) \]
\[ \tilde{w}_{pm,l,j}^{(k)} \triangleq \tilde{w}_{pm,l,j}^{(k)} \left( \check{C}_{pm,l,j}^{(k)} \right) = \tilde{w}_{z_{\tilde{z},l,j}}^{(k)} + W_w^{(L)} f_{l,j}^{(L)} \quad (46) \]
\[ \hat{W}_{z_{\tilde{z},l,j}}^{(k)} \triangleq (\hat{C}_{z_{\tilde{z},l,j}}^{(k)})^{-1}, \hat{w}_{z_{\tilde{z},l,j}}^{(k)} \triangleq \hat{W}_{z_{\tilde{z},l,j}}^{(k)} \eta_{z_{\tilde{z},l,j}}^{(k)} \quad \text{and} \quad \check{C}_{z_{\tilde{z},l,j}}^{(k)} \text{are given by} \quad (83) \quad \text{and} \quad (84), \quad \text{respectively}, \quad \text{and} \quad \hat{C}_{z_{\tilde{z},l,j}}^{(k)} = \check{C}_{z_{\tilde{z},l,j}}^{(k)} = \check{W}_{z_{\tilde{z},l,j}}^{(k)} + W_w^{(L)} f_{l,j}^{(L)} \quad \text{Then, substituting} \quad (39) \quad \text{and} \quad (44) \quad \text{in} \quad (43) \quad \text{yields} \]
\[ \tilde{w}_{f_{e_{2,l,j}}}^{(k)}(x_i^{(N)}) = w_{f_{e_{2,l,j}}}^{(k)} \delta \left( x_i^{(N)} - x_{f_{p,l,j}}^{(N)}[k - 1] \right); \quad (47) \]

where
\[ w_{f_{e_{2,l,j}}}^{(k)} \triangleq w_{f_{e_{1,l,j}}}^{(k)} \cdot w_{pm,l,j}^{(k)} \quad (48) \]

represents the overall weight for the \( j \)-th particle of the set \( S_i^{(k-1)} \); such a weight accounts for both the (real) measurement \( y_t \) and the PM \( z_i^{(N)} \) (through the weights \( w_{f_{e_{1,l,j}}}^{(k)} \) and \( w_{pm,l,j}^{(k)} \), respectively). Once all the weights \( \{w_{f_{e_{2,l,j}}}^{(k)}\} \) are available, their normalization is accomplished; this produces the normalised weights
\[ W_{f_{e_{2,l,j}}}^{(k)} \triangleq \frac{w_{f_{e_{2,l,j}}}^{(k)}}{K_{f_{e_{2,l}}^{(k)}}} \quad (49) \]

where \( K_{f_{e_{2,l}}^{(k)}} \triangleq 1/\sum_{l=0}^{N_p-1} w_{f_{e_{2,l,j}}}^{(k)} \). Note that the particles \( \{x_{f_{p,l,j}}^{(N)}[k - 1]\} \) and their new weights \( \{W_{f_{e_{2,l,j}}}^{(k)}\} \) provide a statistical representation of the forward estimate of \( x_i^{(N)} \) computed by PF in the \( k \)-th iteration.

Resampling with replacement is now accomplished for the particle set \( S_i^{(k-1)} \) on the basis of the new weights \( \{W_{f_{e_{2,l,j}}}^{(k)}\} \) (see \( (49) \)). Note that this task does not emerge from the application of SPA to the considered graphical model; however, it ensures that the particles emerging from it are equally likely. Resampling simply entails that the \( N_p \) particles \( \{x_{f_{p,l,j}}^{(N)}[k - 1]\} \) and their associated weights \( \{W_{f_{e_{2,l,j}}}^{(k)}\} \) (49) are replaced by the new particles \( \{x_{f_{p,l,j}}^{(N)}[k]\} \), forming the set \( S_i^{(k)} \) and having identical weights (all equal to \( 1/N_p \)). Consequently, the effect
of resampling can be simply represented as turning the message \(\hat{m}^{(k)}_{f_{c2j}}(x^{(N)}_{i})\) into
\[
\hat{m}^{(k)}_{f_{c2j}}(x^{(N)}_{i}) = \delta \left( x^{(N)}_{i} - x^{(N)}_{f_{p,j}[k]} \right),
\]
with \(j = 0, 1, ..., N_{p} - 1\).

4. Computation of the PMs for EKF - This step aims at computing the Gaussian message
\[
\hat{m}^{(k)}_{pm,j}(x_{i}) = N \left( x_{i}; \eta_{pm,l,j}, C^{(k)}_{pm,l} \right),
\]
providing the PM information exploited by EKF in its second MU of the next iteration. This requires combining the \(N_{p}\) messages \(\{\hat{m}^{(k)}_{f_{c2j}}(x^{(N)}_{i})\}\) (see (50)) with the \(N_{p}\) messages \(\{\hat{m}^{(k)}_{pm,j}(x^{(L)}_{i})\}\), evaluated in the PMGPF block appearing in Figs. 4-5 and conveying the (particle-dependent) statistical information acquired about \(x^{(L)}_{i}\) on the basis of the PM \(z^{(L)}_{i}\) (12). The computation of the message \(\hat{m}^{(k)}_{pm,j}(x^{(L)}_{i})\) is described in detail in Appendix A, where it is shown that
\[
\hat{m}^{(k)}_{pm,j}(x^{(L)}_{i}) = N \left( x^{(L)}_{i}; \eta^{(k)}_{pm,l,j}, \tilde{C}_{pm,l,j}^{(k)} \right);
\]
here, the covariance matrix \(\tilde{C}_{pm,l,j}^{(k)}\) and the mean vector \(\eta^{(k)}_{pm,l,j}\) are computed on the basis of the precision matrix
\[
\tilde{W}^{(k)}_{pm,l,j} \triangleq \left( \tilde{C}_{pm,l,j}^{(k)} \right)^{-1} = \left( A^{(N)}_{l,i} \right)^{T} W^{(N)}_{w} A^{(N)}_{l,i} \left[ k \right]
\]
and the transformed mean vector
\[
\tilde{w}^{(k)}_{pm,l,j} \triangleq \tilde{W}^{(k)}_{pm,l,j} \eta^{(k)}_{pm,l,j} = \left( A^{(N)}_{l,i} \right)^{T} W^{(N)}_{w} z^{(L)}_{i} \left[ k \right],
\]
respectively; moreover, \(A^{(N)}_{l,i} \left[ k \right] \triangleq A^{(N)}_{l,i} \left( x^{(N)}_{f_{p,l,j}[k]} \right), f^{(N)}_{l,i} \left[ k \right] \triangleq f^{(N)}_{l,i} \left( x^{(N)}_{f_{p,l,j}[k]} \right)\) and \(z^{(L)}_{i} \left[ k \right]\) is defined by (90).

The proposed technique for merging the information provided by \(\{\hat{m}^{(k)}_{f_{c2j}}(x^{(N)}_{i})\}\) with those conveyed by \(\{\hat{m}^{(k)}_{pm,j}(x^{(L)}_{i})\}\) is based on the following considerations. The message \(\hat{m}^{(k)}_{f_{c2j}}(x^{(L)}_{i})\) is coupled with \(\hat{m}^{(k)}_{pm,j}(x^{(N)}_{i})\) (for any \(j\)), since the evaluation of the former message relies on the latter one (see Appendix A). Moreover, these two messages provide complementary information, because they refer to the two different components of the overall state \(x_{i}\). This explains why the joint statistical information conveyed by the sets \(\{\hat{m}^{(k)}_{f_{c2j}}(x^{(N)}_{i})\}\) and \(\{\hat{m}^{(k)}_{pm,j}(x^{(L)}_{i})\}\) can be expressed through the joint pdf
\[
f^{(k)}_{p} \left( x^{(L)}_{i}, x^{(N)}_{i} \right) \triangleq \frac{1}{N_{p}} \sum_{i=0}^{N_{p}-1} \hat{m}^{(k)}_{f_{c2j}} \left( x^{(N)}_{i} \right) \hat{m}^{(k)}_{pm,j} \left( x^{(L)}_{i} \right).
\]
Then, the message $\tilde{m}_{\eta_{pm,l}}^{(k)}(x)$ can be computed by projecting the last function onto a single Gaussian pdf (see (51)), since message passing over the EKF portion of our graphical model involves Gaussian messages only; the transformation adopted here to achieve this result ensures that the mean and the covariance of the pdf $f(k)(x_{l}^{(L)}, x_{r}^{(N)})$ are preserved. For this reason, if the mean $\eta_{pm,l}^{(k)}$ and the covariance matrix $C_{\eta_{pm,l}}^{(k)}$ of the message $\tilde{m}_{\eta_{pm,l}}^{(k)}(x)$ (51) are put in the form

$$\eta_{pm,l}^{(k)} = \left[ \begin{array}{c} \eta_{pm,l}^{(k)} \setminus T \eta_{pm,l}^{(k)} \end{array} \right]^{T}$$

(56)

and

$$C_{\eta_{pm,l}}^{(k)} = \left[ \begin{array}{cc} \tilde{C}_{\eta_{pm,l}}^{(k)} & \tilde{C}_{\eta_{pm,l}}^{(k)} \setminus T \tilde{C}_{\eta_{pm,l}}^{(k)} \\ \tilde{C}_{\eta_{pm,l}}^{(k)} \setminus T & \tilde{C}_{\eta_{pm,l}}^{(k)} \setminus T \tilde{C}_{\eta_{pm,l}}^{(k)} \setminus T \end{array} \right]$$

(57)

respectively, the $D_{L}$-dimensional mean vector $\tilde{\eta}_{pm,l}^{(k)}$ and the $D_{N}$-dimensional mean vector $\tilde{\eta}_{pm,l}^{(k)}$ are computed as

$$\tilde{\eta}_{pm,l}^{(k)} = \frac{1}{N_p} \sum_{j=0}^{N_p-1} \tilde{\eta}_{pm,l,j}^{(k)}$$

(58)

and

$$\tilde{\eta}_{pm,l}^{(k)} = \frac{1}{N_p} \sum_{j=0}^{N_p-1} x_{fe,l,j}^{(N)}[k]$$

(59)

respectively, whereas the $D_{L} \times D_{L}$ covariance matrix $\tilde{C}_{\eta_{pm,l}}^{(k)}$, the $D_{N} \times D_{N}$ covariance matrix $\tilde{C}_{\eta_{pm,l}}^{(k)}$, and $D_{L} \times D_{N}$ cross-covariance matrix $\tilde{C}_{\eta_{pm,l}}^{(k)}$ are computed as

$$\tilde{C}_{\eta_{pm,l}}^{(k)} = \frac{1}{N_p} \sum_{j=0}^{N_p-1} r_{pm,l,j}^{(k)} \left( \eta_{pm,l}^{(k)} \right)^{T},$$

(60)

$$\tilde{C}_{\eta_{pm,l}}^{(k)} = \frac{1}{N_p} \sum_{j=0}^{N_p-1} r_{fe,l,j}^{(N)}[k] - \tilde{\eta}_{pm,l}^{(k)} \left( \eta_{pm,l}^{(k)} \right)^{T},$$

(61)

and

$$\tilde{C}_{\eta_{pm,l}}^{(k)} = \frac{1}{N_p} \sum_{j=0}^{N_p-1} \tilde{r}_{pm,l,j}^{(k)} - \tilde{\eta}_{pm,l}^{(k)} \left( \eta_{pm,l}^{(k)} \right)^{T},$$

(62)

respectively; here, $r_{pm,l,j}^{(k)} \triangleq \tilde{C}_{\eta_{pm,l}}^{(k)} + \tilde{C}_{\eta_{pm,l}}^{(k)} \left( \tilde{\eta}_{pm,l,j}^{(k)} \right)^{T}$, $r_{fe,l,j}^{(N)}[k] \triangleq x_{fe,l,j}^{(N)}[k] \left( x_{fe,l,j}^{(N)}[k] \right)^{T}$ and $\tilde{r}_{pm,l,j}^{(k)} \triangleq \tilde{\eta}_{pm,l,j}^{(k)} \left( x_{fe,l,j}^{(N)}[k] \right)^{T}$. The evaluation of the parameters $\eta_{pm,l}^{(k)}$ (56)

---

[Details about the employed method for condensing the $N_p$-component Gaussian mixture (GM) representing $x_{l}^{(L)}$ into a single Gaussian pdf can be found in [21 Sec. IV].]
and \( C_{pm,l}^{(k)} \) for the message \( \bar{m}_{pm}^{(k)}(x_i) \) concludes step 4. (i.e., the last step of the \( k \)-th iteration). This message is stored for the next iteration; then, if the iteration index \( k \) is less than \( N_{it} \), it is increased by one, so that a new iteration can be started by going back to step 1. On the contrary, if \( k = N_{it} \), the message (see (31)-(33) and Fig. 5)

\[
\bar{m}_{fe,2,l}^{(N_{it}+1)}(x_i) = N\left(x_i; \eta_{fe,2,l}^{(N_{it}+1)}, C_{fe,2,l}^{(N_{it}+1)}\right),
\]

(63)
is computed as if a new iteration was started. Finally, if \( l < t \), the output messages \( \{\bar{m}_{fp,j}(x_{l+1}^{(N)})\} \) and \( \bar{m}_{fp}(x_{l+1}) \) (i.e., the new predictions of the two state components) are computed. On the one hand, the message \( \bar{m}_{fp,j}(x_{l+1}^{(N)}) \) is easily generated as (see (87)-(89))

\[
\bar{m}_{fp,j}(x_{l+1}^{(N)}) = \bar{m}_{fp,j}^{(N_{it})}(x_{l+1}^{(N)})
\]

(64)
for \( j = 0, 1, ..., N_p - 1 \). On the other hand, \( \bar{m}_{fp}(x_{l+1}) \) is computed as (see Fig. [5])

\[
\bar{m}_{fp}(x_{l+1}) = \int \tilde{f}(x_{l+1} | x_l) \bar{m}_{fe,2,l}^{(N_{it}+1)}(x_l) \, dx_l.
\]

(65)

Since \( \tilde{f}(x_{l+1} | x_l) = N(x_{l+1}; F_l x_l + u_l, C_{w}) \) (see (5)) and \( \bar{m}_{fe,2,l}^{(N_{it}+1)}(x_l) \) is a Gaussian message (see (63)), applying CR2 to the evaluation of the RHS of (65) produces

\[
\bar{m}_{fp}(x_{l+1}) = N(x_{l+1}; \eta_{fp,l+1}, C_{fp,l+1}),
\]

(66)
where

\[
\eta_{fp,l+1} \triangleq F_l \eta_{fe,2,l}^{(N_{it}+1)} + u_l
\]

(67)
and

\[
C_{fp,l+1} \triangleq C_w + F_l C_{fe,2,l} C_{fe,2,l}^T F_l^T.
\]

(68)
The \( l \)-th recursion is now over.

The algorithm described above needs a proper initialization. In our work, the Gaussian pdf \( f(x_1) = N(x_1; \eta_1, C_1) \) is assumed for \( x_1 \). Consequently, as far as PF is concerned, before starting the first recursion (corresponding to \( l = 1 \)), the set \( S_1 = \{x_{1,fp,1,j}^{(N)}, j = 0, 1, ..., N_p - 1\} \) is generated for \( x_1^{(N)} \) by sampling the pdf \( f(x_1^{(N)}) \) (that results from the marginalization of \( f(x_1) \)) with respect to \( x_1^{(L)} \) \( N_p \) times; then, the same weight is assigned to each particle (i.e., \( w_{fp,1,j} = 1/N_p \) for any \( j \)). Moreover, we set \( \bar{m}_{fp}(x_1) = f(x_1) \) for the EKF portion of the TF#1 algorithm.

All the processing tasks accomplished in the message passing procedure derived above are summarized in Algorithm 1. Note also that, at the end of
the \( l \)-th recursion, estimates of \( \hat{x}_j^{(N)} \) and \( \hat{x}_j^{(L)} \) can be evaluated as: a) \( \hat{x}_j^{(N)} = \sum_{j=0}^{N-1} \hat{x}_j^{(N)} \) (see our previous comments following eq. (49)) or \( \hat{x}_j^{(N)} = \hat{x}_j^{(N+1)} \), where \( \hat{x}_j^{(N+1)} \) consists of the last \( D_N \) elements of \( \eta_j^{(N+1)} \) (see (63); b) \( \hat{x}_j^{(L)} = \hat{x}_j^{(N+1)} \), where \( \eta_j^{(N+1)} \) consists of the first \( D_L \) elements of \( \eta_j^{(N+1)} \).

The scheduling adopted in the \( k \)-th iteration of the \( l \)-the recursion accomplished by TF\#2 consists in computing the involved messages according to the following order: 1) \( \{m_{ms,j}^{(k)}(x_i^{(N)})\} \), \( \{m_{fc1,j}^{(k)}(x_i^{(N)})\} \) (first MU in PF); 2) \( \{m_{fc2,j}^{(k)}(x_i^{(N)})\} \) (second MU in PF; note that \( m_{pm,j}^{(0)}(x_i^{(N)}) = 1 \) for any \( j \)); 3) \( \{m_{pm,j}^{(k)}(x_i^{(L)})\}, m_{pm}(x_i), m_{fc2}(x_i), m_{fc2}(x_i^{(L)}) \) (computation of PMs for EKF and second MU in EKF); 4) \( \{m_{pm,j}^{(k)}(x_i^{(N)})\} \) (computation of PMs for PF). This algorithm can be easily derived following the same line of reasoning as TF\#1 and is summarised in Algorithm 2.

As far as the computational complexity of TF\#1 and TF\#2 is concerned, it can be shown that it is of order \( O(N_{TF}) \), with

\[
N_{TF} = 2DP^2 + PD^2 + (N_{it} + 4)D^3 + N_{it} \cdot N_p(PD_L^2 + P^2D_L + P^3 + 6D_L^2 + 2D_ND_L^2 + 3D_LD_N^2 + D_N^3). \tag{69}
\]

The last expression has been derived keeping into account all the dominant contributions due to matrix inversions, matrix products and Cholesky decompositions, that need to be accomplished for the complete state update and measurement models expressed by \[\{1\} \] and \[\{2\}, \] respectively. However, all the possible contributions originating from the evaluation of the matrices \( A_i^{(Z)}(x_i^{(N)}) \) and the functions \( f_{i}^{(Z)}(x_i^{(N)}) \) (with \( Z = L \) and \( N \) ) over the considered particle sets are not accounted for. A similar approach has been followed for MPF, whose complexity\footnote{An assessment of MPF complexity is also available in \[\{24\}.]} is of order \( O(N_{MPF}) \), with

\[
N_{MPF} = N_p(2PD_L^2 + 3P^2D_L + P^3 + 5D_L^3 + 2D_L^2D_N + 3D_LD_N^2 + D_N^3/3). \tag{70}
\]

Finally, it is worth mentioning that TF\#1 and TF\#2 have substantially smaller memory requirements than MPF; in fact, the former algorithms need to store the state estimates generated by a single extended Kalman filter, whereas the latter one those computed by \( N_p \) Kalman filters running in parallel. This means that, if MPF is employed, a larger number of memory accesses must
Algorithm 1: Turbo Filtering #1

1 Initialisation: For $j = 0$ to $N_p - 1$: sample the pdf $f(x_i^{(N)})$ to generate the particles $x_{fp,l,j}^{(N)}$ (forming $s_i^{(0)}$), and assign the weight $w_{fp,l} = 1/N_p$ to each of them. Set $W_{fp,1} = W_1 = [C_1]^{-1}$, $w_{fp,1} = W_1 \eta_1$.

2 Filtering: For $l = 1$ to $t$:
   a- First MU in EKF: Compute $W_{fe,l+1}$ and $w_{fe,l+1}$
   $C_{fe,l} = [W_{fe,l}]^{-1}$ and $\eta_{fe,l} = C_{fe,l} W_{fe,l}$. Then, extract $\eta_{fe,l}$ and $\bar{C}_{fe,l}$ from $\eta_{fe,l}$, and $C_{fe,l}$, respectively. Set $W_{pm,l} = 0_{D,D}$ and $w_{pm,l}^{(0)} = 0_D$.

   for $k = 1$ to $N_{it}$ do
      b- Second MU in EKF. Compute $C_{fe,l}^{(k)}$ and $\eta_{fe,l}^{(k)}$.
      c- Marginalization: extract $\bar{z}_{fe,l}^{(k)}$ and $\bar{C}_{fe,l}^{(k)}$ from $\eta_{fe,l}^{(k)}$, and $C_{fe,l}^{(k)}$, respectively.
      d- MUs in PF:
         for $j = 0$ to $N_p - 1$ do
            d1- First MU in PF: compute $\tilde{z}_{ms,l,j}^{(k)}$, $\bar{C}_{ms,l,j}^{(k)}$ and $w_{fe,l,j}^{(k)}$
            d2- Computation of PMs for PF: compute $\tilde{z}_{z,l,j}^{(k)}$ and $\bar{C}_{z,l,j}^{(k)}$
               $\tilde{W}_{z,l,j}^{(k)} = [\bar{C}_{z,l,j}]^{-1}$ and $\tilde{w}_{z,l,j}^{(k)} = \tilde{W}_{z,l,j}^{(k)} \eta_{z,l,j}^{(k)}$. Then, compute $\tilde{W}_{pm,l,j}^{(k)}$ and $\tilde{w}_{pm,l,j}^{(k)}$.
               $\eta_{pm,l,j}^{(k)} = C_{pm,l,j} \tilde{w}_{pm,l,j}^{(k)}$. Finally, compute $w_{pm,l,j}^{(k)}$.
            d3- Second MU in PF: compute $w_{fe,l,j}^{(k)}$.
         end
      end
   end

   e- Normalization of particle weights: compute the normalized weights $\{W_{fe,l,j}^{(k)}\}$ according to (49).
   f- Resampling with replacement: generate the new particle set $S_l^{(k)} = \{x_{fp,l,j}^{(N)}\}$ by resampling $S_l^{(k-1)}$ on the basis of the weights $\{W_{fe,l,j}^{(k)}\}$.
   g- Computation of PM for EKF: For $j = 1$ to $N_p$: Compute $\tilde{v}_{fp,l,j}^{(k)}$ and $\bar{C}_{fp,l,j}^{(k)}$, and sample the pdf $N(x_{fp,l+1,j}^{(N)}; \tilde{v}_{fp,l+1,j}^{(k)}, \bar{C}_{fp,l+1,j}^{(k)})$ to generate the new particle $x_{fp,l+1,j}^{(N)}$ and assign the weight $1/N_p$ to it. Then, compute $z_{l,j}^{(k)}$ and $\tilde{w}_{pm,l,j}^{(k)}$ and $\tilde{w}_{pm,l,j}^{(k)}$ by (52), $\bar{C}_{pm,l,j}^{(k)} = [\tilde{W}_{pm,l,j}^{(k)}]^{-1}$ and $\eta_{pm,l,j}^{(k)} = \bar{C}_{pm,l,j}^{(k)} \tilde{w}_{pm,l,j}^{(k)}$. Finally, compute $w_{pm,l,j}^{(k)}$ (according to (57), (62), $W_{pm,l}^{(k)} = [C_{pm,l}]^{-1}$ and $w_{pm,l}^{(k)} = W_{pm,l} \eta_{pm,l}^{(k)}$.
   end

   h- Compute forward prediction (if $l < t$): For $j = 1$ to $N_p$: set $x_{fp,l+1,j}^{(N)} = x_{fp,l+1,j}^{(N)}$ (these particles form the set $S_{l+1}$) and the weight $W_{fp,l+1,j}^{(N)} = W_{fp,l+1,j}^{(N)}$. Compute $C_{fp,l+1,j}^{(N)}$ and $\eta_{fp,l+1,j}^{(N)}$ on the basis of (52) and (53). Then, compute $\tilde{v}_{fp,l+1}^{(k)}$ and $\bar{C}_{fp,l+1}^{(k)}$, $W_{fp,l+1}^{(k)} = [C_{fp,l+1}]^{-1}$ and $w_{fp,l+1}^{(k)} = W_{fp,l+1} \eta_{fp,l+1}$. 


Algorithm 2: Turbo Filtering #2

1 **Initialisation:** Same as Alg. 1.

2 **Filtering:** For $l = 1$ to $t$:
   a- *First MU in EKF:* Same as Alg. 1 task a.

   for $k = 1$ to $N_t$ do
   b- **MUs in PF:**
      for $j = 0$ to $N_p - 1$ do
      b1- *First MU in PF:* Same as Alg. 1 task d1.
      b2- *Second MU in PF:* Same as Alg. 1 task d3.
      end
   c- *Normalization of particle weights:* Same as Alg. 1 task e.
   d- *Resampling with replacement:* Same as Alg. 1 task f.
   e- *Computation of PM for EKF:* Same as Alg. 1 task g.
   f- *Second MU in EKF:* Same as Alg. 1 task b.
   g- *Marginalization:* Same as Alg. 1 task c.
   h- *Computation of PMs for PF:* Same as Alg. 1 task d2.
   end

i- *Compute forward prediction* (if $l < t$):

   for $j = 0$ to $N_p - 1$ do
   Compute $\eta_{ms,l,j}^{(k)}$, $\tilde{C}_{ms,l,j}^{(k)}$ and $w_{fe1,l,j}^{(k)}$, then compute
   $w_{fe2,l,j}^{(k)}$.
   end

Finally, compute $\eta_{fp,l+1}$ and $C_{fp,l+1}$, $\mathbf{W}_{fp,l+1} = [C_{fp,l+1}]^{-1}$
and $\mathbf{w}_{fp,l+1} = \mathbf{W}_{fp,l+1} \eta_{fp,l+1}$. 

26
be accomplished on the hardware platform on which the filtering algorithm is run; as evidenced by our numerical results, this feature can make the overall execution time of MPF much larger than that required by TF, even if \( N_{TF} > N_{MPF} \) for the same value of \( N_p \).

## 5 Interpretation of Turbo Filtering

An interesting interpretation of the processing tasks accomplished by the TF#1 and TF#2 algorithms can be developed as follows. In TF#1, the \( j \)-th particle weight \( w^{(k)}_{f	ext{e}2,l,j} \) available at the end of the second MU of PF expresses the a posteriori statistical information about the particle \( x^{(N)}_{f,p,l,j}[k-1] \) and can be put in the equivalent form

\[
w^{(k)}_{f	ext{e}2,l,j} = w^{(a)}_{l,j} \cdot w^{(k)}_{f	ext{e}1,l,j} \cdot w^{(k)}_{pm,l,j};
\tag{71}
\]

where \( w^{(a)}_{l,j} \) denotes the a priori information available for the particle itself (in our derivation \( w^{(a)}_{l,j} = 1 \) has been assumed, in place of \( w^{(a)}_{l,j} = 1/N_p \), to simplify the notation; see \((20)\)). Taking the natural logarithm of both sides of \((71)\) produces

\[
L_{l,j}[k] = L^{(a)}_{l,j} + L^{(y)}_{l,j}[k] + L^{(z)}_{l,j}[k]
\tag{72}
\]

where \( L_{l,j}[k] \triangleq \ln(w^{(k)}_{f	ext{e}2,l,j}), L^{(a)}_{l,j} \triangleq \ln(w^{(a)}_{l,j}), L^{(y)}_{l,j}[k] \triangleq \ln(w^{(k)}_{f	ext{e}1,l,j}) \) and \( L^{(z)}_{l,j} \triangleq \ln(w^{(k)}_{pm,l,j}) \). The last equation has the same mathematical structure as the well known formula (see [13, Sec. 10.5, p. 450, eq. (10.15)] or [22, Par. II.C, p. 432, eq. (20)])

\[
L(u_j|y) = L(u_j) + L_c(y_j) + L_e(u_j)
\tag{73}
\]

expressing of the log-likelihood ratio (LLR) available for the \( j \)-th information bit \( u_j \) at the output of a SISO channel decoder operating over an additive white Gaussian noise (AWGN) channel and fed by: a) the channel output vector \( y \) (whose \( j \)-th element \( y_j \) is generated by the communications channel in response to a channel symbol conveying \( u_j \) and is processed to produce the so-called channel LLR \( L_c(y_j) \)); b) the a priori LLR \( L(u_j) \) about \( u_j \); c) the extrinsic LLR \( L_e(u_j) \), i.e. a form of soft information available about \( u_j \), but intrinsically not influenced by such a bit (in turbo decoding of concatenated channel codes extrinsic information is generated by another channel decoder with which soft information is exchanged with the aim of progressively refining data estimates).

This correspondence is not only formal, since the term \( L^{(y)}_{l,j}[k] \) (\( L^{(a)}_{l,j} \)) in \((72)\) provides the same kind of information as \( L_c(y_j) \) (\( L(u_j) \)), since these are both related to the noisy data (a priori information) available about the quantities to
be estimated (the system state in one case, an information bit in the other one). These considerations suggest that the term \( L(z)_{l,j}^k \) of (72) should represent the counterpart of the quantity \( L(u_j) \) appearing in (73), i.e. the so called extrinsic information (in other words, that part of the information available about \( x_i^{(N)} \) and not intrinsically influenced by \( x_i^{(N)} \) itself). This interpretation is confirmed by the fact that \( L(z)_{l,j}^k \) is computed on the basis of the statistical knowledge available about \( x_i^{(L)} \) and \( x_{l+1}^{(L)} \) (see Appendix A), which, thanks to (1) (with \( Z = L \)), does provide useful information about \( x_i^{(N)} \).

The reader can easily verify that an interpretation similar to that provided for \( w(k)_{fe2,l,j} \) (48) can be given for \( \vec m(k)_{fe2}(x_l) \) (31) (that conveys our a posteriori information about \( x_l \)). In fact, the last message results from the product of the messages \( \vec m_{fp}(x_l) \) (19), \( \vec m_{fe1}(x_l) \) (26) and \( \vec m_{pm}(x_l) \) (51); these convey prior, measurement and extrinsic information about \( x_l \), respectively. It is worth noting, however, that \( \vec m_{pm}(x_l) \) (51) combines two different contributions, namely the contributions from the message sets \( \{ \vec m_{fe2,l,j}(x_l^{(N)}) \} \) (50) and \( \{ \vec m_{pm,l,j}(x_l^{(L)}) \} \) (52); however, only the message \( \vec m_{pm,l,j}(x_l^{(L)}) \) can be really interpreted as the counterpart of \( w(k)_{pm,l,j} \) (44), since its computation is based on the PM message \( \vec m_{j}(z_{l}^{(L)}) \) (91).

6 Numerical Results

In this Section we compare, in terms of accuracy and execution time, the TF#1 and TF#2 algorithms with EKF and MPF for a specific CLG SSM. The considered SSM refers to an agent moving on a plane and whose state \( x_l \) in the \( l \)-th observation interval is defined as \( x_l \triangleq [p_{x,l}^T, v_{x,l}^T]^T \), where \( v_{x,l} \triangleq [v_{x,l}, v_{y,l}]^T \) and \( p_{l} \triangleq [p_{x,l}, p_{y,l}]^T \) represent the agent velocity and its position, respectively (their components are expressed in m/s and in m, respectively). As far as the state update equations are concerned, we assume that: a) the agent velocity is approximately constant within each sampling interval; b) the model describing its time evolution is obtained by including the contribution of a position- and velocity-dependent force in a first-order autoregressive model (characterized by the forgetting factor \( \rho \), with \( 0 < \rho < 1 \)). Therefore, the dynamic model

\[
\mathbf{v}_{l+1} = \rho \mathbf{v}_l + (1 - \rho) \mathbf{n}_{v,l} + \mathbf{a}_l (\mathbf{p}_l, \mathbf{v}_l) T_s, \tag{74}
\]

is adopted for velocity; here, \( \{ \mathbf{n}_{v,l} \} \) is an additive white Gaussian noise (AWGN) process (whose elements are characterized by the covariance matrix \( \mathbf{I}_2 \)), \( T_s \) is the sampling interval and

\[
\mathbf{a}_l (\mathbf{p}_l, \mathbf{v}_l) = -(a_0/d_0) \mathbf{p}_l - \tilde{a}_0 f_v (\|\mathbf{v}_l\|) \mathbf{n}_{v,l}. \tag{75}
\]
In the RHS of the last formula, \(a_0\) and \(\tilde{a}_0\) are scale factors (both expressed in \(\text{m/s}^2\)), \(d_0\) is a reference distance, \(u_{v,l} \triangleq v/l/\|v_l\|\) is the versor associated with \(v_l\) and \(f_v(x) = (x/v_0)^3\) is a continuous, differentiable and dimensionless function expressing the dependence of the second term on the intensity of \(v_l\) (the parameter \(v_0\) represents a reference velocity). Note that the first term and the second one in the RHS of (75) represent the contribution of position-dependent force pointing towards the origin and proportional to \(\|p_l\|\), and that of velocity-dependent force acting as a resistance to the motion of the agent, respectively.

Given (74), the dynamic model
\[
p_{l+1} = p_l + v_l T_s + \frac{1}{2} a_l (p_l, v_l) T_s^2 + n_{p,l}
\]
can be employed for the position of the considered agent; here, \(\{n_{p,l}\}\) is an AWGN process (whose elements are characterized by the covariance matrix \(\sigma_p^2 \mathbf{I}_2\)), independent if \(\{n_{v,l}\}\) and accounting for model inaccuracy.

In our study the measurement model
\[
y_l = [p_l^T \|v_l\|]^T + e_l,
\]
is also adopted; here, \(\{e_l\}\) is an AWGN process, whose elements are characterized by the covariance matrix \(C_e = \text{diag}(\sigma_p^2, \sigma_e^2, \sigma_v^2)\). Then, if we set \(x_l^{(L)} = p_l\) and \(x_l^{(N)} = v_l\), it is not difficult to show that the state equation (74) and the measurement equation (77) can be considered as instances of (1) with \(Z = L\) and (1) with \(Z = N\) and (2), respectively.

In our computer simulations, the estimation accuracy of the considered filtering techniques has been assessed by evaluating two root mean square errors (RMSEs), one for the linear state component, the other for the nonlinear one, over an observation interval lasting \(T = 300 T_s\); these are denoted \(RMSE_L(\text{alg})\) and \(RMSE_N(\text{alg})\), respectively, where ‘alg’ denotes the algorithm these parameters refer to. Our assessment of computational requirements is based, instead, on assessing the average execution time required over the whole observation interval (this quantity is denoted ET(\text{alg}) in the following). Moreover, the following values have been selected for the parameters of the considered SSM: \(\rho = 0.99\), \(T_s = 0.1 \text{ s}\), \(\sigma_p = 0.01 \text{ m}\), \(\sigma_{e,p} = 5 \cdot 10^{-2} \text{ m}\), \(\sigma_{e,v} = 5 \cdot 10^{-2} \text{ m/s}\), \(a_0 = 1.5 \text{ m/s}^2\), \(d_0 = 0.5 \text{ m}\), \(\tilde{a}_0 = 0.05 \text{ m/s}^2\) and \(v_0 = 1 \text{ m/s}\) (the initial position \(p_0 \triangleq [p_{x,0}, p_{y,0}]^T\) and the initial velocity \(v_0 \triangleq [v_{x,0}, v_{y,0}]^T\) have been set to \([5 \text{ m}, 8 \text{ m}]^T\) and \([4 \text{ m/s}, 4 \text{ m/s}]^T\), respectively).

Some numerical results showing the dependence of \(RMSE_L\) and \(RMSE_N\) on the number of particles \((N_p)\) for MPF, TF#1 and TF#2 are illustrated in
Figure 6: RMSE performance versus $N_p$ for the linear component ($RMSE_L$) and the nonlinear component ($RMSE_N$) of system state; the CLG SSM described by eqs. (74)-(75) and four filtering techniques (EKF, MPF, TF#1 and TF#2) are considered.

Fig. 6 (simulation results are indicated by markers, whereas continuous lines are drawn to fit them, so facilitating the interpretation of the available data); in this case $N_{it} = 1$ has been selected for both TF#1 and TF#2, and the range [10, 150] has been considered for $N_p$. These results show that:

1) The value of $RMSE_L$ is significantly smaller than $RMSE_N$ for all the algorithms; this is mainly due to the fact that the measurement vector $y_l$ (77) provides richer information about $x_l^L$ (i.e., $p_l$) than about $x_l^N$ ($v_l$).

2) The EKF technique is appreciably outperformed by the other three filtering algorithms in terms of both $RMSE_L$ and $RMSE_N$ for any value of $N_p$; for instance, $RMSE_L$(EKF) ($RMSE_N$(EKF)) is about 1.65 (1.80) time larger than $RMSE_L$(TF#1) ($RMSE_N$(TF#1)) for $N_p = 100$.

3) Both TF#1 and TF#2 perform slightly worse than MPF for the same value of $N_p$ (for instance, $RMSE_L$(TF#1) and $RMSE_N$(TF#1) are about 5% larger than the corresponding quantities evaluated for MPF); moreover, there is no visible performance gap between TF#1 and TF#2, in terms of both $RMSE_L$ and $RMSE_N$.

4) No real improvement in terms of $RMSE_L$(alg) and $RMSE_N$(alg) is found for $N_p \gtrsim 100$, if alg = MPF, TF#1 or TF#2.

Despite their similar accuracies, MPF and TF algorithms require different execution times; this is evidenced by the numerical results appearing in Fig. 7 and showing the dependence of the ET parameter on $N_p$ for all the considered
filtering algorithms. These results show that TF#1 and TF#2 require an appreciably shorter execution time than MPF; more precisely, the value of ET for TF1 (TF#2) is approximately 0.61 (0.67) times smaller than that required by MPF for the same value of $N_p$. Moreover, from Fig. 6 it is easily inferred that, in the considered scenario, TF#1 achieves a better RMSE - ET tradeoff than both MPF and TF#2.

Further simulation results (not shown here for space limitations) have also evidenced that, in the considered scenario, no improvement in estimation accuracy is obtained if $N_i > 1$ is selected for TF#1 and TF#2.

7 Conclusions

In this manuscript the concept of parallel concatenation of Bayesian filters has been illustrated and a new graphical model has been developed for it. This model can be exploited to develop a new family of filtering algorithms, called turbo filters. Two turbo filters have been derived for the class of CLG SSMs and have been compared, in terms of both accuracy and execution time, with EKF and MPF for a specific SSM. Simulation results evidence that the devised TF schemes perform closely to MPF, but have limited memory requirements and are appreciably faster.
Appendix A

In this Appendix, the evaluation of the PM messages \( \hat{m}_{pm,j}^{(k)}(x_l^{(N)}) \) (44) and \( \hat{m}_{pm,j}^{(k)}(x_l^{(L)}) \) (51) is analysed in detail. The algorithm for computing \( \hat{m}_{pm,j}^{(k)}(x_l^{(N)}) \) can be represented as a message passing over the FG shown in Fig. 8-a). The expressions of the messages appearing in this graph can be derived as follows. Given \( x_l^{(N)} = x_{fp,l}^{(N)}[k-1] \) (conveyed by \( \hat{m}_{fe2,j}^{(k)}(x_l^{(N)}) \) (47)) and \( \hat{m}_{fe2}^{(k)}(x_l^{(L)}) \) (34), the message\footnote{The scale factor \( w_{fe2,l,j}^{(k)} \) originating from \( \hat{m}_{fe2,j}^{(k)}(x_l^{(N)}) \) (47) can be ignored in the following formula, since the resulting message is Gaussian} (78) producing a statistical representation of the prediction of \( x_l^{(L)} \) is computed first.

\[
\hat{m}_{fp,j}^{(k)}(x_{l+1}^{(L)}) = \int f(x_{l+1}^{(L)} | x_l^{(L)}, x_{fp,l}^{(N)}[k-1]) \cdot \hat{m}_{fe2}^{(k)}(x_l^{(L)}) \, dx_l^{(L)}
\]

Since \( f(x_{l+1}^{(L)} | x_l^{(L)}, x_{fp,l}^{(N)}[k-1]) = \mathcal{N}(x_{l+1}^{(L)}, I_{fp,l+1,j}^{(k)} + A_{l,j}^{(L)}[k] x_l^{(L)}, C_{fp}^{(L)}) \) (with \( A_{l,j}^{(L)}[k] \equiv A_l^{(L)}(x_{fp,l,j}^{(N)}[k-1]) \) and \( I_{fp,l,j}^{(k)} \equiv \hat{f}_{fp,l,j}^{(k)}(x_{fp,l,j}^{(N)}[k-1]) \)), applying CR2 to the evaluation of the integral in the RHS of (78) produces

\[
\hat{m}_{fp,j}^{(k)}(x_{l+1}^{(L)}) = \mathcal{N}(x_{l+1}^{(L)}, \hat{I}_{fp,l+1,j}^{(k)} + \hat{C}_{fp,l+1,j}^{(k)}),
\]

Figure 8: Representation of the processing accomplished by a) the PMG\( \text{EKF} \) block and b) the PMG\( \text{PF} \) block (see Fig. 5) as message passing over a FG.
where
\[ \tilde{\eta}^{(k)}_{fp,l+1,j} \triangleq A_{i,j}^{(L)}[k] \tilde{\eta}^{(k)}_{fc,l+1} + f_i^{(L)}[k] \]  
(80)
and
\[ \tilde{C}^{(k)}_{fp,l+1,j} \triangleq C_w^{(L)} + A_{i,j}^{(L)}[k] \tilde{C}^{(k)}_{fc,l} \left( A_{i,j}^{(L)}[k] \right)^T. \]  
(81)

Then, the message \( \tilde{m}^{(k)}_j(z_i^{(N)}) \) is evaluated (this message is denoted \( ZN_j^{(k)} \) in Fig. 3a)); this expresses the pdf of \( z_i^{(N)} \) (10) under the assumptions that: a) \( x_i^{(N)} = x_{fp,l,j}^{(N)}[k - 1] \); b) \( \tilde{\eta}^{(k)}_{z,i,l,j} \) and \( \tilde{\eta}^{(k)}_{z,i,j} \) are jointly Gaussian vectors; c) the pdfs of \( x_i^{(L)} \) and \( x_{i+1}^{(L)} \) are expressed by \( \tilde{m}^{(k)}_j(x_i^{(L)}) \) (29) and \( \tilde{m}^{(k)}_j(x_{i+1}^{(L)}) \) (79), respectively; d) the pdf of \( x_{i+1}^{(L)} \) conditioned on \( x_i^{(L)} \) and \( x_i^{(N)} = x_{fp,l,j}^{(N)}[k - 1] \) is \( f(x_{i+1}^{(L)}|x_i^{(L)}; x_{fp,l,j}^{(N)}[k - 1]) = N(x_{i+1}^{(L)}; \xi_{i+1}^{(L)}[k] + A_{i,j}^{(L)}[k]x_i^{(L)}, C_w^{(L)} \) (see (11)) with \( Z = L \). Therefore, based on eq. (10), the message \( \tilde{m}^{(k)}_j(z_i^{(N)}) \) can expressed as
\[ \tilde{m}^{(k)}_j(z_i^{(N)}) = N \left( z_i^{(N)}; \tilde{\eta}^{(k)}_{z,i,l,j}, \tilde{C}^{(k)}_{z,i,l,j} \right), \]  
(82)
where
\[ \tilde{\eta}^{(k)}_{z,i,l,j} = \tilde{\eta}^{(k)}_{fp,l+1,j} - A_{i,j}^{(L)} \tilde{\eta}^{(k)}_{fc,l+1,j} \]
\[ = A_{i,j}^{(L)} \left[ \tilde{\eta}^{(k)}_{fc,l+1,j} - \tilde{\eta}^{(k)}_{fc,l+1} \right] + f_i^{(L)}[k], \]  
(83)
and
\[ \tilde{C}^{(k)}_{z,i,l,j} = \tilde{C}^{(k)}_{fp,l+1,j} - A_{i,j}^{(L)}[k] \tilde{C}^{(k)}_{fc,l+1,j} \left( A_{i,j}^{(L)}[k] \right)^T \]
\[ = C_w^{(L)} + A_{i,j}^{(L)}[k] \left[ \tilde{C}^{(k)}_{fc,l+1,j} - \tilde{C}^{(k)}_{fc,l} \right] \left( A_{i,j}^{(L)}[k] \right)^T. \]  
(84)

Finally, \( \tilde{m}^{(k)}_j(z_i^{(N)}) \) (82) is exploited to evaluate
\[ \tilde{m}^{(k)}_{pm,j}(x_i^{(N)}) = \int \tilde{m}^{(k)}_j(z_i^{(N)}) \left( z_i^{(N)} \right) x_{fp,l,j}^{(N)}[k - 1] \right) dz_i^{(N)}. \]  
(85)

Substituting (82) and \( f(z_i^{(N)}|x_{fp,l,j}^{(N)}[k - 1]) = N(z_i^{(N)}; \xi_{i,j}^{(L)}[k], C_w^{(N)} \) (see (11)) in the RHS of the last expression and applying CR3 to the evaluation of the resulting integral yields (44).

Similarly as \( \tilde{m}^{(k)}_{pm,j}(x_i^{(N)}) \), the algorithm for computing the message \( \tilde{m}^{(k)}_{pm,j}(x_i^{(L)}) \) can be represented as a message passing over a graphical model. Such a model

---

8Note that the following message represents the *correlation* between the pdf \( \tilde{m}_j(z_i^{(N)}) \) evaluated on the basis of the definition (10) and the pdf originating from the fact that this quantity is expected to equal the random vector \( \xi_i^{(L)} + w_i^{(L)} \) (see (11)). For this reason, it expresses the *degree of similarity* between these functions.
is shown in Fig. 8(b); moreover, the derivation of the messages passed over it is sketched in the following. Given \( x_i^{(N)} = x_{fp,l,j}^{(N)}[k] \) (conveyed by the message \( \bar{m}_{f,r}^{(k)}(x_i^{(N)}) \) (50) and \( m_{f,r}^{(k)}(x_i^{(L)}) \) (34)), the message

\[
\bar{m}_{f,p}^{(k)}(x_{i+1}^{(N)}) = \int \int f \left( x_{i+1}^{(N)} \bigg| x_i^{(L)}, x_{fp,l,j}^{(N)}[k] \right) \bar{m}_{f,r}^{(k)}(x_i^{(L)}) \, dx_i^{(L)},
\]

(86)

representing a forward prediction of \( x_{i+1}^{(N)} \), is evaluated first. Applying CR2 to the evaluation of the last integral (note that \( f(x_{i+1}^{(N)}|x_i^{(N)}[k], x_i^{(L)}) = N(x_{i+1}^{(N)}; A_{l,j}^{(N)}[k]x_i^{(L)} + f_{l,j}^{(N)}[k], C_{l,j}^{(N)}) \), with \( A_{l,j}^{(N)}[k] \equiv A_{l,j}^{(N)}(x_{fp,l,j}^{(N)}[k]) \) and \( f_{l,j}^{(N)}[k] \equiv f_{l,j}^{(N)}(x_{fp,l,j}^{(N)}[k]) \), and that \( \bar{m}_{f,r}^{(k)}(x_i^{(L)}) \) (34) is Gaussian) yields

\[
\bar{m}_{f,p}^{(k)}(x_{i+1}^{(N)}) = N\left( x_{i+1}^{(N)}; \tilde{\eta}_{f,p,l,j}^{(k)}, \tilde{C}_{f,p,l,j}^{(k)} \right),
\]

(87)

where

\[
\tilde{\eta}_{f,p,l,j}^{(k)} \equiv A_{l,j}^{(N)}[k] \tilde{\eta}_{f,r}^{(k)} + f_{l,j}^{(N)}[k]
\]

(88)

and

\[
\tilde{C}_{f,p,l,j}^{(k)} \equiv C_{l,j}^{(N)} + A_{l,j}^{(N)}[k] \tilde{C}_{f,r}^{(k)} \left( A_{l,j}^{(N)}[k] \right)^T.
\]

(89)

Then, the message \( \bar{m}_{f,p}^{(k)}(x_{i+1}^{(N)}) \) (87) is replaced by its particle-based representation; this result is achieved sampling the Gaussian function \( N(x_{i+1}^{(N)}; \tilde{\eta}_{f,p,l,j}^{(k)}, \tilde{C}_{f,p,l,j}^{(k)}) \) (see (87)), that is drawing the sample \( x_{fp,l+1,j}^{(N)}[k] \) from it and b) assigning the weight \( 1/N_p \) to this sample. The value of the PM \( z_i^{(L)} \) (12) associated with the couple \( (x_i^{(N)}, x_{i+1}^{(N)}) = (x_{fp,l}^{(N)}[k], x_{fp,l+1,j}^{(N)}[k]) \) is

\[
z_{l,j}^{(L)}[k] \equiv x_{fp,l+1,j}^{(N)}[k] - f_{l,j}^{(N)}[k]
\]

(90)

and is conveyed by the message (denoted \( ZL_j^{(k)} \) in Fig. 8(b))

\[
\bar{m}_{j}^{(k)}(z_i^{(L)}) = \delta \left( z_i^{(L)} - z_{l,j}^{(L)}[k] \right).
\]

(91)

Then, the message \( m_{p}^{(k)}(x_{i}^{(L)}) \) is evaluated as (see Fig. 8(b))

\[
m_{p}^{(k)}(x_{i}^{(L)}) = \int \bar{m}_{j}^{(k)}(z_i^{(L)}) f \left( z_i^{(L)} \bigg| x_i^{(L)}, x_{i+1}^{(N)} \right) \, dz_i^{(L)}.
\]

(92)

Substituting (91) and \( f(z_i^{(L)}|x_i^{(L)}, x_{i}^{(N)}) = N(z_i^{(L)}; A_{l,j}^{(N)}[k]x_i^{(L)}, C_{l,j}^{(N)}) \) (see (13)) in the RHS of (92) yields the message \( m_{p}^{(k)}(x_{i}^{(L)}) = N(z_i^{(L)}[k]; A_{l,j}^{(N)}[k]x_i^{(L)}, C_{l,j}^{(N)}) \), that can be easily put in the equivalent Gaussian form (52).
References

[1] M. S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking”, IEEE Trans. Sig. Proc., vol. 50, no. 2, pp. 174-188, Feb. 2002.

[2] S. Mazuelas, Y. Shen and M. Z. Win, “Belief Condensation Filtering”, IEEE Trans. Sig. Proc., vol. 61, no. 18, pp. 4403-4415, Sept. 2013.

[3] V. Smidl and A. Quinn, “Variational Bayesian Filtering”, IEEE Trans. Sig. Proc., vol. 56, no. 10, pp. 5020-5030, Oct. 2008.

[4] B. Anderson and J. Moore, Optimal Filtering, Englewood Cliffs, NJ, Prentice-Hall, 1979.

[5] A. Doucet, J. F. G. de Freitas and N. J. Gordon, “An Introduction to Sequential Monte Carlo methods,” in Sequential Monte Carlo Methods in Practice, A. Doucet, J. F. G. de Freitas, and N. J. Gordon, Eds. New York: Springer-Verlag, 2001.

[6] A. Doucet, S. Godsill and C. Andrieu, “On Sequential Monte Carlo Sampling Methods for Bayesian Filtering”, Statist. Comput., vol. 10, no. 3, pp. 197-208, 2000.

[7] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson and P. Nordlund, “Particle Filters for Positioning, Navigation, and Tracking”, IEEE Trans. Sig. Proc., vol. 50, 425-435, 2002.

[8] C. Andrieu and A. Doucet, “Particle filtering for partially observed Gaussian state space models”, J. Roy. Statist. Soc.: Ser. B, vol. 64, no. 4, pp. 827–836, 2002.

[9] T. Schö n, F. Gustafsson, P.-J. Nordlund, “Marginalized Particle Filters for Mixed Linear/Nonlinear State-Space Models”, IEEE Trans. Sig. Proc., vol. 53, no. 7, pp. 2279-2289, July 2005.

[10] T. Lu, M. F. Bugallo and P. M. Djuric, “Simplified Marginalized Particle Filtering for Tracking Multimodal Posteriors”, Proc. IEEE/SP 14th Workshop on Stat. Sig. Proc. (SSP ’07), pp. 269-273, Madison, WI (USA), 2007.

[11] B. Krach and P. Roberston, “Cascaded estimation architecture for integration of foot-mounted inertial sensors”, Proc. of the 2008 IEEE/ION Position, Location and Navigation Symposium, Monterey (CA), pp. 112-119, 5-8 May, 2008.
[12] G. M. Vitetta and G. Baldini, “Theoretical framework for In-Car Navigation based on Integrated GPS/IMU Technologies”, JRC, Tech. Rep., 2014 (available online at)

[13] G. M. Vitetta, D. P. Taylor, G. Colavolpe, F. Pancaldi and P. A. Martin, *Wireless Communications: Algorithmic Techniques*, John Wiley & Sons, 2013.

[14] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, “Serial Concatenation of Interleaved Codes: Performance Analysis, Design, and Iterative Decoding”, *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 909-926, May 1998.

[15] C. Berrou and A. Glavieux, “Near Optimum Error Correcting Coding and Decoding: Turbo-Codes”, *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261 - 1271, Oct. 1996.

[16] H.-A. Loeliger, J. Dauwels, Junli Hu, S. Krol, Li Ping, F. R. Kschischang, “The Factor Graph Approach to Model-Based Signal Processing”, *IEEE Proc.*, vol. 95, no. 6, pp. 1295-1322, June 2007.

[17] F. R. Kschischang, B. Frey, and H. Loeliger, “Factor Graphs and the Sum-Product Algorithm”, *IEEE Trans. Inf. Theory*, vol. 41, no. 2, pp. 498-519, Feb. 2001.

[18] F. Montorsi, ”Localization and Tracking for Indoor Environments”, PhD Thesis, 2013 (available at https://morethesis.unimore.it/theses/available/etd-01142013-121728/)

[19] J. Dauwels, S. Krol and H.-A. Loeliger, “Particle Methods as Message Passing”, *Proc. 2006 IEEE Int. Symp. on Inf. Theory*, pp. 2052-2056, 9-14 July 2006.

[20] A. P. Worthen and W. E. Stark, “Unified Design of Iterative Receivers using Factor Graphs”, *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 843-849, Feb. 2001.

[21] A. R. Runnalls, “Kullback-Leibler Approach to Gaussian Mixture Reduction”, *IEEE Trans. on Aer. and Elec. Syst.*, vol. 43, no. 3, pp. 989-999, July 2007.

[22] J. Hagenauer, E. Offer and L. Papke, “Iterative decoding of binary block and convolutional codes”, *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 429-445, Mar 1996.
[23] R. Karlsson, T. Schön, F. Gustafsson, “Complexity Analysis of the Marginalized Particle Filter”, *IEEE Trans. Sig. Proc.*, vol. 53, no. 11, pp. 4408-4411, Nov. 2005.