On the other hand, we have

$$\mathbf{E}[Q_N]\mathbf{E}[Q_N]^T = R_{ab}(0)R_{abT}(0) + R_{ab}(0)R_{cdT}(0) + R_{cd}(0)R_{abT}(0) + R_{cd}(0)R_{cdT}(0).$$

Subtracting the aforementioned equations and taking the limit $N \to \infty$ yields

$$\begin{aligned}\lim_{N\to\infty} N\mathbf{Cov}[Q_N] = \sum_{\tau=-\infty}^{\infty} [&R_{aa}(\tau)R_{bbT}(\tau) + R_{ba}(\tau)R_{abT}(\tau) + R_{ac}(\tau)R_{bdT}(\tau) \\ &+ R_{bc}(\tau)R_{adT}(\tau) + R_{ca}(\tau)R_{dbT}(\tau) + R_{da}(\tau)R_{cbT}(\tau) \\ &+ R_{cc}(\tau)R_{ddT}(\tau) + R_{dc}(\tau)R_{cdT}(\tau)].\end{aligned}$$

Applying the formula

$$\sum_{\tau=-\infty}^{+\infty} R_{ab}(\tau)R_{cd}(\tau) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Phi_{ab}(\omega)\bar{\Phi}_{cd}(\omega)\,d\omega$$

componentwise and inserting the expressions for the cross-spectra finally furnishes the claim of Proposition 4.1 with the obvious substitutions applying. □

REFERENCES

[1] J. R. Blum, "Multidimensional stochastic approximation methods," *Ann. Math. Statist.*, vol. 25, pp. 737–744, 1954.

[2] R. Hildebrand, A. Lecchini, G. Solari, and M. Gevers, "A convergence theorem for iterative feedback tuning," CESAME, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, Tech. Rep. 2003/17, 2003.

[3] ——, "Prefiltering in iterative feedback tuning: Optimization of the prefilter for accuracy," *IEEE Trans. Autom. Control*, vol. 49, no. 10, pp. 1801–1805, Oct. 2004.

[4] ——, "Optimal prefiltering in iterative feedback tuning," *IEEE Trans. Autom. Control*, vol. 50, no. 8, pp. 1196–1200, Aug. 2005.

[5] H. Hjalmarsson, "Performance analysis of iterative feedback tuning," Dept. Signals, Sensors, Syst., Royal Inst. Technol., Stockholm, Sweden, Tech. Rep., 1998.

[6] ——, "Iterative feedback tuning—An overview," *Int. J. Adapt. Control Signal Processing*, vol. 16, no. 5, pp. 373–395.

[7] H. Hjalmarsson, M. Gevers, S. Gunnarson, and O. Lequin, "Iterative feedback tuning: Theory and applications," *IEEE Control Syst. Mag.*, vol. 18, no. 4, pp. 26–41, Aug. 1998.

[8] H. Hjalmarsson, S. Gunnarson, and M. Gevers, "A convergent iterative restricted complexity control design scheme," in *Proc. 33rd IEEE Conf. Decision and Control*, Lake Buena Vista, FL, Dec. 1994, pp. 1735–1740.

[9] M. B. Nevelson and R. Z. Khasminskii, *Stochastic Approximation and Recursive Estimation, Volume 47 of Translations of Mathematical Monographs*. Providence, RI: AMS, 1976.

[10] M. Rosenblatt, *Stationary Sequences and Random Filelds*. Boston, MA: Birkhäuser, 1985.

[11] G. Yin, "A stopped stochastic approximation algorithm," *Syst. Control Lett.*, vol. 11, pp. 107–115, 1988.

Multiscale Bayesian Restoration in Pairwise Markov Trees

François Desbouvries and Jean Lecomte

***Abstract*—An important problem in multiresolution analysis of signals and images consists in estimating continuous hidden random variables $\mathbf{x} = \{\mathbf{x}_s\}_{s\in\mathcal{S}}$ from observed ones $\mathbf{y} = \{\mathbf{y}_s\}_{s\in\mathcal{S}}$. This is done classically in the context of hidden Markov trees (HMTs). In this note we deal with the recently introduced pairwise Markov trees (PMTs). We first show that PMTs are more general than HMTs. We then deal with the linear Gaussian case, and we extend from HMTs with independent noise (HMT-IN) to PMT a smoothing Kalman-like recursive estimation algorithm which was proposed by Chou *et al.*, as well as an algorithm for computing the likelihood.**

***Index Terms*—Gaussian processes, hidden Markov trees (HMTs), multiscale algorithms, pairwise Markov trees (PMTs), recursive estimation.**

I. INTRODUCTION

Multiresolution analysis and multiscale algorithms are of interest in a large variety of signal and image processing problems (see, e.g., [1]–[7], as well as the tutorial [8]). Efficient restoration algorithms have been developed in the context of tree-based structures [1]–[3], [7]. These algorithms estimate the hidden random variables $\mathbf{x}$ from the observed ones $\mathbf{y}$, under the assumption that the stochastic interactions of $\mathbf{x}$ and $\mathbf{y}$ are modeled by a hidden Markov tree (HMT).

On the other hand, it is well known that if $(\mathbf{x}, \mathbf{y})$ is a classical hidden Markov model (HMM), then the pair $(\mathbf{x}, \mathbf{y})$ itself is Markovian. Conversely, starting from the sole assumption that $(\mathbf{x}, \mathbf{y})$ is Markovian, i.e., that $(\mathbf{x}, \mathbf{y})$ is a so-called pairwise Markov model (PMM), is a more general point of view which nevertheless enables the development of similar restoration algorithms. More precisely, some of the classical Bayesian restoration algorithms used in hidden Markov fields (HMFs), hidden Markov chains (HMCs), or HMTs, have been generalized recently to the more general frameworks of pairwise Markov fields (PMFs) [9], pairwise Markov chains (PMCs) with discrete [10] or continuous [11], [12] state process, and of PMTs with discrete [13], [14] or continuous [14] hidden variables.

The aim of this note is to extend to PMTs the smoothing Kalman-like algorithm which was developed in [3] in the context of HMTs with independent noise (HMT-IN), as well as an algorithm for computing the likelihood. As we will see, in a PMT the hidden tree $\mathbf{x}$ is not necessarily Markovian, and the observed variables $\mathbf{y}$ are not necessarily related to $\mathbf{x}$ as simply as in the HMT-IN case. Yet the conditional law of $\mathbf{x}$ given $\mathbf{y}$ remains Markovian, which in turn enables us to propose an efficient restoration algorithm.

This note is organized as follows. In Section II, we briefly recall the three embedded HMT-IN, HMT, and PMT models, and we show (in the case where $\mathbf{x}$ is continuous) that PMT are more general than HMT. An extension to the PMT model of the recursive restoration algorithm of [3] is given in Section III, and an extension to the PMT model of an algorithm for computing the likelihood is proposed in Section IV.

II. HMTS VERSUS PMTS

Let $\mathcal{S}$ be a finite set of indices, and let us consider a tree structure with nodes indexed on $\mathcal{S}$. Let us partition $\mathcal{S}$ in terms of its successive

Manuscript received December 23, 2003; revised November 20, 2004. Recommended by Associate Editor F. Vazquez-Abad.

The authors are with the GET/INT/CITI and CNRS UMR 5157, 91011 Evry, France (e-mail: Francois.Desbouvries@int-evry.fr).

Digital Object Identifier 10.1109/TAC.2005.852552

generations $\mathcal{S}_1, \ldots, \mathcal{S}_n$. So, $\mathcal{S}_1$ is made of the root node r, $\mathcal{S}_2$ gathers the children of node r, and so on. Each node s (apart from the root node r) has one father s^-. The set of all descendants of a node s is denoted by s^{++}. We assume for notational simplicity that the tree is dyadic, i.e., that each node s (which is not in the last generation $\mathcal{S}_n$) has exactly two children s_1 and s_2.

Let now $\mathbf{x} = \{\mathbf{x}_s\}_{s\in\mathcal{S}}$ and $\mathbf{y} = \{\mathbf{y}_s\}_{s\in\mathcal{S}}$ be two sets of random variables indexed on $\mathcal{S}$. Each $\mathbf{x}_s$ (respectively, $\mathbf{y}_s$) belongs to $\mathbb{R}^p$ (respectively, to $\mathbb{R}^q$). Let $p(\mathbf{x}_s)$ (respectively, $p(\mathbf{y}_s)$) denote the probability density function (p.d.f.) of $\mathbf{x}_s$ (respectively, of $\mathbf{y}_s$) w.r.t. Lebesgue measure, and for any set of indices Σ, let $p(\mathbf{x}_s \,|\, \{\mathbf{y}_\sigma\}_{\sigma\in\Sigma})$ denote the conditional p.d.f. of $\mathbf{x}_s$ given $\{\mathbf{y}_\sigma\}_{\sigma\in\Sigma}$. Other p.d.f. or conditional p.d.f. of interest are defined similarly.

We now recall the HMT-IN, HMT, and PMT models. The following model is widely used for modeling $p(\mathbf{x}, \mathbf{y})$:

$$p(\mathbf{x},\mathbf{y}) = \underbrace{p(\mathbf{x}_r)\prod_{i=2}^{n}\prod_{s\in\mathcal{S}_i} p(\mathbf{x}_s \,|\, \mathbf{x}_{s^-})}_{p(\mathbf{x})} \times \underbrace{\prod_{s\in\mathcal{S}} p(\mathbf{y}_s \,|\, \mathbf{x}_s)}_{p(\mathbf{y}\,|\,\mathbf{x})}. \tag{1}$$

In this model, $\mathbf{x}$ is a Markov tree (MT), and conditionally on $\mathbf{x}$, the variables $\mathbf{y}_s$ are independent and satisfy $p(\mathbf{y}_s \,|\, \mathbf{x}) = p(\mathbf{y}_s \,|\, \mathbf{x}_s)$. In order to avoid any possible confusion in the sequel, we will refer to model (1) as an HMT-IN.

Now, let us introduce the pair $\mathbf{z}_s = (\mathbf{x}_s, \mathbf{y}_s)$, and let $\mathbf{z} = \{\mathbf{z}_s\}_{s\in\mathcal{S}}$. A PMT model is a model in which we only assume that $\mathbf{z}$ is an MT

$$p(\mathbf{z}) = p(\mathbf{z}_r)\prod_{i=2}^{n}\prod_{s\in\mathcal{S}_i} p(\mathbf{z}_s \,|\, \mathbf{z}_{s^-}). \tag{2}$$

Our interest for PMT comes from a simple observation. A set of variables $(\mathbf{x}, \mathbf{y})$ satisfying (1) also satisfies (2), so any HMT-IN is a PMT. However, the converse is not true, as can be seen at the local level, since in (2) the transition p.d.f. $p(\mathbf{z}_s \,|\, \mathbf{z}_{s^-})$ reads

$$\begin{aligned} p(\mathbf{z}_s \,|\, \mathbf{z}_{s^-}) &= p(\mathbf{x}_s, \mathbf{y}_s \,|\, \mathbf{x}_{s^-}, \mathbf{y}_{s^-}) \\ &= p(\mathbf{x}_s \,|\, \mathbf{x}_{s^-}, \mathbf{y}_{s^-})\, p(\mathbf{y}_s \,|\, \mathbf{x}_s, \mathbf{x}_{s^-}, \mathbf{y}_{s^-}) \end{aligned}$$

so a classical HMT-IN is a PMT in which $p(\mathbf{x}_s \,|\, \mathbf{x}_{s^-}, \mathbf{y}_{s^-})$ reduces to $p(\mathbf{x}_s \,|\, \mathbf{x}_{s^-})$ and $p(\mathbf{y}_s \,|\, \mathbf{x}_s, \mathbf{x}_{s^-}, \mathbf{y}_{s^-})$ to $p(\mathbf{y}_s \,|\, \mathbf{x}_s)$, which is rather rough. In other words, making use of PMT enables to model more complex physical situations, since at node s the state $\mathbf{x}_s$ is allowed to depend not only on the state, but also on the observation at node s^-; and the observation $\mathbf{y}_s$ is allowed to depend not only on the state at node s, but also on the state and on the observation at node s^-. The restoration algorithm of Section III can thus be directly applied to any of the image or signal processing problems in which a tree structure is used (see, e.g., [8] and the references therein), but for which the local stochastic dependencies described by an HMT-IN model are too rough approximations of the actual physical situation.

Let us finally denote an HMT a model in which both $\mathbf{x}$ and $\mathbf{z}$ are MT. As we have just seen, a PMT needs not be an HMT-IN. Now, a PMT needs not even be an HMT, because if (2) holds, $\mathbf{x}$ is not necessarily an MT, as we see from the following result.

Proposition 1: Let $\mathbf{z}$ be a dyadic PMT satisfying (2). Assume that

$$\text{for all } s\in\mathcal{S}\backslash\mathcal{S}_1 \quad p(\mathbf{x}_s \,|\, \mathbf{x}_{s^-}, \mathbf{y}_{s^-}) = p(\mathbf{x}_s \,|\, \mathbf{x}_{s^-}). \tag{3}$$

Then, $\mathbf{x}$ is an MT. Conversely, assume that $\mathbf{x}$ is an MT, and that for all $s \in \mathcal{S}\setminus\mathcal{S}_n$, $p(\mathbf{z}_{s_1} \,|\, \mathbf{z}_s) = p(\mathbf{z}_{s_2} \,|\, \mathbf{z}_s)$, i.e., that conditionally on the father, the laws of the children are equal. Then, (3) holds.

Proof of Proposition 1: A proof of Proposition 1 has been given in [13] for the case where $\{\mathbf{x}_s\}_{s\in\mathcal{S}}$ are discrete random variables. An adaptation of this proof to the continuous case can be found in the Appendix. ■

III. COMPUTATION OF THE POSTERIOR P.D.F. OF A GIVEN NODE

From now on, we will assume that $\mathbf{z}$ is a Gaussian PMT. The aim of this section consists in computing the posterior p.d.f. $p(\mathbf{x}_s \,|\, \mathbf{y})$ for an arbitrary $s \in \mathcal{S}$.

A. Modeling Assumptions and Structure of the Algorithm

Our assumptions are as follows. We assume that (2) holds and, moreover, that

$$\mathbf{z}_{s_i} = \mathbf{F}_{s_i}\mathbf{z}_s + \mathbf{w}_{s_i} \quad E\left(\mathbf{w}_{s_i}\mathbf{w}_{s_i}^T\right) = \mathbf{Q}_{s_i} \tag{4}$$

in which $\mathbf{w} = \{\mathbf{w}_s\}_{s\in\mathcal{S}\backslash\mathcal{S}_1}$ are random vectors which are zero-mean, independent and independent of $\mathbf{z}_r$, and in which $\mathbf{Q}_s$ is positive definite ($\mathbf{Q}_s > \mathbf{0}$) for all $s \in \mathcal{S}$. We also assume that $\mathbf{w}$ is Gaussian, and that $p(\mathbf{z}_r) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_r)$ (the assumption $E(\mathbf{z}_r) = \mathbf{0}$ can easily be released and is made here only for notational simplicity). As a consequence, $\mathbf{z}$ is a zero-mean Gaussian process and we set $p(\mathbf{z}_s) \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_s)$. All conditional p.d.f. related to $\mathbf{z}$ are also Gaussian, so propagating these conditional p.d.f. amounts to propagating the parameters of these Gaussian densities. Let us thus introduce the following notations[1]:

$$p(\underbrace{\mathbf{x}_s, \mathbf{y}_s}_{\mathbf{z}_s} \,|\, \{\mathbf{y}_\sigma\}_{\sigma\in\Sigma}) \sim \mathcal{N}\left(\underbrace{\begin{bmatrix}\hat{\mathbf{x}}_{s\,|\,\Sigma} \\ \hat{\mathbf{y}}_{s\,|\,\Sigma}\end{bmatrix}}_{\hat{\mathbf{z}}_{s\,|\,\Sigma}}, \underbrace{\begin{bmatrix}\mathbf{P}^{x,x}_{s\,|\,\Sigma} & \mathbf{P}^{x,y}_{s\,|\,\Sigma} \\ \mathbf{P}^{y,x}_{s\,|\,\Sigma} & \mathbf{P}^{y,y}_{s\,|\,\Sigma}\end{bmatrix}}_{\mathbf{P}_{s\,|\,\Sigma}}\right). \tag{5}$$

The algorithm we propose is a direct extension of the algorithm [3], which itself was an extension to HMT-IN of the RTS smoothing algorithm [15] derived in the HMC framework.

Following [3], our algorithm is essentially made of two sweeps, one filtering sweep in the backward (fine-to-coarse) direction and then one smoothing sweep in the forward (coarse-to-fine) direction. More precisely, the structure of the algorithm is a follows.

1) From $p(\mathbf{z}_r)$ and (4), we compute recursively $p(\mathbf{z}_s)$ for all $s \in \mathcal{S}$ via

$$\mathbf{P}_{s_i} = \mathbf{Q}_{s_i} + \mathbf{F}_{s_i}\mathbf{P}_s\mathbf{F}_{s_i}^T. \tag{6}$$

2) *Fine-to-coarse sweep*: Starting from $\{p(\mathbf{x}_s \,|\, \mathbf{y}_s)\}_{s\in\mathcal{S}_n}$, we compute recursively, in the fine-to-coarse direction, $\{p(\mathbf{x}_s \,|\, \mathbf{y}_s, \mathbf{y}_{s^{++}})\}_{s\in\mathcal{S}_m}$ for all $m \in \{n-1, \ldots, 1\}$. Since each p.d.f. $p(\mathbf{x}_s \,|\, \mathbf{y}_s, \mathbf{y}_{s^{++}})$ is computed from $\{p(\mathbf{x}_{s_i} \,|\, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}})\}_{i=1}^{2}$ (see Section III-B for details), the computations of the 2^{m-1} p.d.f. $p(\mathbf{x}_s \,|\, \mathbf{y}_s, \mathbf{y}_{s^{++}})$ of a given generation m can be performed in parallel. At the end of this backward sweep, $p(\mathbf{x}_r \,|\, \mathbf{y})$ has been computed.

3) *Coarse-to-fine sweep*: It remains to compute $p(\mathbf{x}_s \,|\, \mathbf{y})$ for an arbitrary s. There is a unique path $\{\sigma_i\}_{i=1}^{m}$ (with $\sigma_1 = r$ and $\sigma_m = s$) connecting node s to the root node r. Along this path, the conditional law of $\{\mathbf{x}_{\sigma_i}\}_{i=1}^{m}$ given $\mathbf{y}$ is Markovian, so $p(\mathbf{x}_s \,|\, \mathbf{y})$ can be computed recursively from $p(\mathbf{x}_{\sigma_1} \,|\, \mathbf{y})$ and $\{p(\mathbf{x}_{\sigma_i} \,|\, \mathbf{x}_{(\sigma_i)^-}, \mathbf{y})\}_{i=2}^{m}$. On the other hand, we will see in Section III-C that each p.d.f. $p(\mathbf{x}_{s_i} \,|\, \mathbf{x}_s, \mathbf{y})$ can be computed from $p(\mathbf{x}_s \,|\, \mathbf{y})$, and from $p(\mathbf{x}_{s_i} \,|\, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}})$ and $p(\mathbf{z}_s \,|\, \mathbf{z}_{s_i})$ which have been computed previously.

[1]Later on, we will also need to partition $\mathbf{F}_s$, $\tilde{\mathbf{F}}_s$, $\mathbf{Q}_s$, and $\tilde{\mathbf{Q}}_s$ conformably with the dimensions of $\mathbf{x}_s$ and $\mathbf{y}_s$ (for instance, $\mathbf{F}_s^{x,x}$ denotes the top-left $p\times p$ submatrix of $\mathbf{F}_s$, which is of dimensions $(p+q)\times(p+q)$).

We now turn to the computational details of the algorithm. The backward sweep is explained in Section III-B and the forward sweep in Section III-C. The derivations rely on two ingredients. First, the algorithm relies on the Gaussian assumption.[2] Second, the PMT assumption plays an important role; in particular, the following two properties of Markov trees will prove useful in the sequel.

- P1) Let $s \in \mathcal{S}_m$ with $1 < m < n$. Conditionally on $\mathbf{z}_s$, $\{\mathbf{z}_s^{++}\}$, and $\{\mathbf{z}_\sigma\}_{\mathcal{S}\setminus\{s,s++\}}$ are independent.
- P2) Let $s \in \mathcal{S}_m$ with $1 \leq m < n$. Conditionally on $\mathbf{z}_s$, $\{\mathbf{z}_{s_1}, \mathbf{z}_{s_1^{++}}\}$ and $\{\mathbf{z}_{s_2}, \mathbf{z}_{s_2^{++}}\}$ are independent.

B. Fine-to-Coarse Sweep

Each elementary step of the backward sweep can be decomposed into three substeps.

1) *Backward prediction step*: For $i = 1, 2$, computation of $p(\mathbf{z}_s \,|\, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}})$ from $p(\mathbf{x}_{s_i} \,|\, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}})$.
2) *Fusion step*: Computation of $p(\mathbf{z}_s \,|\, \mathbf{y}_{s++})$ from $p(\mathbf{z}_s \,|\, \mathbf{y}_{s_1}, \mathbf{y}_{s_1^{++}})$ and $p(\mathbf{z}_s \,|\, \mathbf{y}_{s_2}, \mathbf{y}_{s_2^{++}})$.
3) *Measurement-update step*: Computation of $p(\mathbf{x}_s \,|\, \mathbf{y}_s, \mathbf{y}_{s++})$ from $p(\mathbf{z}_s \,|\, \mathbf{y}_{s++})$.

These three substeps are described, respectively, by the following three propositions.

Proposition 2 (Backward Prediction Step): $p(\mathbf{z}_s \,|\, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}})$ can be computed from $p(\mathbf{x}_{s_i} \,|\, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}})$ via the following recursion:

$$\hat{\mathbf{z}}_{s\,|\,s_i,s_i^{++}} = \tilde{\mathbf{F}}_{s_i} \begin{bmatrix} \hat{\mathbf{x}}_{s_i\,|\,s_i,s_i^{++}} \\ \mathbf{y}_{s_i} \end{bmatrix} \tag{7}$$

$$\mathbf{P}_{s\,|\,s_i,s_i^{++}} = \tilde{\mathbf{Q}}_{s_i} + \begin{bmatrix} \tilde{\mathbf{F}}^{x,x}_{s_i} \\ \tilde{\mathbf{F}}^{y,x}_{s_i} \end{bmatrix} \mathbf{P}^{x,x}_{s_i\,|\,s_i,s_i^{++}} \left[\left(\tilde{\mathbf{F}}^{x,x}_{s_i}\right)^T \left(\tilde{\mathbf{F}}^{y,x}_{s_i}\right)^T \right] \tag{8}$$

in which $\tilde{\mathbf{F}}_{s_i}$ and $\tilde{\mathbf{Q}}_{s_i}$ are given by

$$\tilde{\mathbf{F}}_{s_i} = \mathbf{P}_s \mathbf{F}^T_{s_i} \mathbf{P}^{-1}_{s_i} \tag{9}$$

$$\tilde{\mathbf{Q}}_{s_i} = \mathbf{P}_s - \mathbf{P}_s \mathbf{F}^T_{s_i} \mathbf{P}^{-1}_{s_i} \mathbf{F}_{s_i} \mathbf{P}_s. \tag{10}$$

Proof of Proposition 2: See the Appendix. ■

Proposition 3 (Fusion Step): $p(\mathbf{z}_s \,|\, \mathbf{y}_{s++})$ can be computed from $p(\mathbf{z}_s \,|\, \mathbf{y}_{s_1}, \mathbf{y}_{s_1^{++}})$ and $p(\mathbf{z}_s \,|\, \mathbf{y}_{s_2}, \mathbf{y}_{s_2^{++}})$ via the following recursions:

$$\hat{\mathbf{z}}_{s\,|\,s++} = \mathbf{P}_{s\,|\,s++} \left[\sum_{i=1}^{2} \mathbf{P}^{-1}_{s\,|\,s_i,s_i^{++}} \hat{\mathbf{z}}_{s\,|\,s_i,s_i^{++}} \right] \tag{11}$$

$$\mathbf{P}_{s\,|\,s++} = \left[\sum_{i=1}^{2} \mathbf{P}^{-1}_{s\,|\,s_i,s_i^{++}} - \mathbf{P}^{-1}_s \right]^{-1}. \tag{12}$$

Proof of Proposition 3: See the Appendix. ■

Proposition 4 (Measurement-Update Step): $p(\mathbf{x}_s \,|\, \mathbf{y}_s, \mathbf{y}_{s++})$ can be computed from $p(\mathbf{z}_s \,|\, \mathbf{y}_{s++})$ via the following recursions:

$$\hat{\mathbf{x}}_{s\,|\,s,s++} = \hat{\mathbf{x}}_{s\,|\,s++} + \mathbf{P}^{x,y}_{s\,|\,s++} \left(\mathbf{P}^{y,y}_{s\,|\,s++}\right)^{-1} (\mathbf{y}_s - \hat{\mathbf{y}}_{s\,|\,s++}) \tag{13}$$

$$\mathbf{P}^{x,x}_{s\,|\,s,s++} = \mathbf{P}^{x,x}_{s\,|\,s++} - \mathbf{P}^{x,y}_{s\,|\,s++} \left(\mathbf{P}^{y,y}_{s\,|\,s++}\right)^{-1} \mathbf{P}^{y,x}_{s\,|\,s++}. \tag{14}$$

Proof of Proposition 3: Use Proposition 8. ■

[2]Our algorithm could of course alternately be obtained as a recursive linear minimum mean square error restoration procedure, which indeed was the original approach of [3]. We chose to adopt the Gaussian point of view because the proofs are obtained in a simpler and more direct way; in particular, we will make an extensive use of Propositions 8 and 9 (see the Appendix).

C. Coarse-to-Fine Sweep

Remember from Section III-A that the key point of the coarse-to-fine sweep is the recursive computation of $p(\mathbf{x}_{s_i} \,|\, \mathbf{y})$ from $p(\mathbf{x}_s \,|\, \mathbf{y})$.

Proposition 5: $p(\mathbf{x}_{s_i} \,|\, \mathbf{y})$ can be computed from $p(\mathbf{x}_s \,|\, \mathbf{y})$ via the following recursions:

$$\mathbf{J}_{s_i} = \mathbf{P}^{x,x}_{s_i\,|\,s_i,s_i^{++}} \left[\left(\tilde{\mathbf{F}}^{x,x}_{s_i}\right)^T \left(\tilde{\mathbf{F}}^{y,x}_{s_i}\right)^T \right] \mathbf{P}^{-1}_{s\,|\,s_i,s_i^{++}} \tag{15}$$

$$\hat{\mathbf{x}}_{s_i\,|\,S} = \hat{\mathbf{x}}_{s_i\,|\,s_i,s_i^{++}} + \mathbf{J}_{s_i} \left(\begin{bmatrix} \hat{\mathbf{x}}_{s\,|\,S} \\ \mathbf{y}_s \end{bmatrix} - \hat{\mathbf{z}}_{s\,|\,s_i,s_i^{++}} \right) \tag{16}$$

$$\mathbf{P}^{x,x}_{s_i\,|\,S} = \mathbf{P}^{x,x}_{s_i\,|\,s_i,s_i^{++}} + \mathbf{J}_{s_i} \left(\begin{bmatrix} \mathbf{P}^{x,x}_{s\,|\,S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} - \mathbf{P}_{s\,|\,s_i,s_i^{++}} \right) \mathbf{J}^T_{s_i}. \tag{17}$$

Proof of Proposition III.C: See the Appendix. ■

D. Comments and Remarks

- The assumption $\mathbf{Q}_s > \mathbf{0}$ in Section III-A is a simple sufficient condition ensuring that all computations are valid. For if $\mathbf{Q}_s > \mathbf{0}$ for all s, then from (6) we get $\mathbf{P}_s > \mathbf{0}$, so the covariance matrix of $(\mathbf{z}_s, \mathbf{z}_{s_i})$ in (32) is $>\mathbf{0}$ as well, and $\tilde{\mathbf{Q}}_{s_i}$ in (10) is $>\mathbf{0}$. On the other hand, from (35) we get $\mathbf{\Pi}_s > \mathbf{0}$, so the covariance matrix in (37) is also $>\mathbf{0}$ which, in turn, ensures that $\mathbf{P}_{s\,|\,s_i,s_i^{++}} > \mathbf{0}, \mathbf{P}_{s\,|\,s++} > \mathbf{0}$, and $\mathbf{P}^{x,x}_{s\,|\,s,s++} > \mathbf{0}$. Next, since $\tilde{\mathbf{Q}}_{s_i} > \mathbf{0}$ and $\mathbf{P}^{x,x}_{s\,|\,s,s++} > \mathbf{0}$, the covariance matrix in (33) is $>\mathbf{0}$, so the matrix $\mathbf{C}_{s_i}$ in (44) is $>\mathbf{0}$. Finally, $\mathbf{P}^{x,x}_{r\,|\,r,r++} = \mathbf{P}^{x,x}_{r\,|\,\mathcal{S}} > \mathbf{0}$, and by induction we see that $\mathbf{P}^{x,x}_{s\,|\,\mathcal{S}} > \mathbf{0}$, for all s.
- Equations (7)–(17) still hold if we only assume that $\mathbf{P}_s > \mathbf{0}$ for all s (with $\mathbf{Q}_s$ and $\mathbf{\Pi_s}$ possibly singular). The proof is slightly more technical and is omitted here for simplicity.
- The algorithm derived in [3] can be obtained as a particular case of our algorithm (details are omitted due to lack of space).
- Our algorithm inherits good properties of that of Chou *et al.* In particular, its complexity is linear in the number of nodes, and its regular pyramidal structure (which is consistent with that of the dyadic tree) yields considerable parallelism in the computations.
- The algorithm can easily be adapted to the case where each node s admits an arbitrary number of children ν_s; of course, depending on the specific tree structure, parallelism may no longer be ensured. All equations remain valid, apart from Proposition 3 which needs to be adapted. The sums in (38) and in (40) run from 1 to ν_s, so (11) and (12) become

$$\hat{\mathbf{z}}_{s\,|\,s++} = \mathbf{P}_{s\,|\,s++} \sum_{i=1}^{\nu_s} \mathbf{P}^{-1}_{s\,|\,s_i,s_i^{++}} \hat{\mathbf{z}}_{s\,|\,s_i,s_i^{++}} \tag{18}$$

$$\mathbf{P}_{s\,|\,s++} = \left[\sum_{i=1}^{\nu_s} \mathbf{P}^{-1}_{s\,|\,s_i,s_i^{++}} - (\nu_s - 1)\mathbf{P}^{-1}_s \right]^{-1}. \tag{19}$$

IV. Computation of the Likelihood

In the previous sections, an algorithm for computing $p(\mathbf{x}_s \,|\, \mathbf{y})$ has been derived. On the other hand, in such problems as hypothesis testing or parameter estimation, one needs to compute the likelihood $p(\mathbf{y})$ of the data. In this section we propose an algorithm for computing the likelihood, which is a direct generalization to the PMT case of an efficient solution already proposed in the HMT-IN case [16], [17], [8].

Proposition 6: For all $s \in \mathcal{S}_m, m \in [1,\ldots,n-1]$

$$p(\mathbf{y}_{s++} \mid \mathbf{z}_s) = \prod_{i=1}^{2} p(\mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}} \mid \mathbf{z}_s) \tag{20}$$

and for all $s \in \mathcal{S}_m, m \in [2,\ldots,n-1]$

$$p(\mathbf{y}_s, \mathbf{y}_{s++} \mid \mathbf{z}_{s-}) = \int p(\mathbf{z}_s \mid \mathbf{z}_{s-}) p(\mathbf{y}_{s++} \mid \mathbf{z}_s) d\mathbf{x}_s. \tag{21}$$

These recursions can be used to compute the likelihood recursively: the algorithm is initialized by $\{p(\mathbf{y}_{s_i} \mid \mathbf{z}_s)\}_{s\in\mathcal{S}_{n-1}}$; then $p(\mathbf{y}_{r++} \mid \mathbf{z}_r)$ is computed via a succession of (20) and (21) for all $s \in \mathcal{S}_m, m = n-1,\ldots,2$, and (20) for $m = 1$; finally $p(\mathbf{y})$ is obtained as

$$p(\mathbf{y}) = \int p(\mathbf{z}_r) p(\mathbf{y}_{r++} \mid \mathbf{z}_r) d\mathbf{x}_r. \tag{22}$$

Proof of Proposition 6: Equation (20) holds because of P2). On the other hand, due to P1) we have $p(\mathbf{y}_s, \mathbf{y}_{s++} \mid \mathbf{z}_{s-}) = \int p(\mathbf{x}_s, \mathbf{y}_s, \mathbf{y}_{s++} \mid \mathbf{z}_{s-})\, d\mathbf{x}_s = \int p(\mathbf{z}_s \mid \mathbf{z}_{s-})\, p(\mathbf{y}_{s++} \mid \mathbf{z}_s) d\mathbf{x}_s$. ■

Remarks: The algorithm is an extension of the algorithm already proposed in [16] and [17] in the context of HMT-IN. If $\{\mathbf{z}_s\}_{s\in\mathcal{S}}$ is an HMT-IN, then $p(\mathbf{z}_s \mid \mathbf{z}_{s-}) = p(\mathbf{x}_s \mid \mathbf{x}_{s-}) p(\mathbf{y}_s \mid \mathbf{x}_s)$, and (20), (21) reduce, respectively, to

$$\frac{p(\mathbf{y}_s, \mathbf{y}_{s++} \mid \mathbf{x}_s)}{p(\mathbf{y}_s \mid \mathbf{x}_s)} = \prod_{i=1}^{2} p\left(\mathbf{y}_{s_i, s_i^{++}} \mid \mathbf{x}_s\right) \tag{23}$$

which coincides with [8, eq. (43)] and to

$$\begin{aligned} p(\mathbf{y}_s, \mathbf{y}_{s++} \mid \mathbf{z}_{s-}) &= \int p(\mathbf{x}_s \mid \mathbf{x}_{s-}) p(\mathbf{y}_s \mid \mathbf{x}_s) \frac{p(\mathbf{y}_s, \mathbf{y}_{s++} \mid \mathbf{x}_s)}{p(\mathbf{y}_s \mid \mathbf{x}_s)} d\mathbf{x}_s \\ &= \int p(\mathbf{x}_s \mid \mathbf{x}_{s-}) p(\mathbf{y}_s, \mathbf{y}_{s++} \mid \mathbf{x}_s) d\mathbf{x}_s \\ &= p(\mathbf{y}_s, \mathbf{y}_{s++} \mid \mathbf{x}_{s-}) \end{aligned} \tag{24}$$

which coincides with [8, eq. (42)] (equations (41) and (42) are displayed in [8] in the discrete case, but of course they are also valid in the continuous case). ■

Now, let moreover $\mathbf{z}$ be a zero-mean Gaussian process as before. So, $p(\mathbf{y})$ is a Gaussian p.d.f., the parameters of which can be computed via the following recursions.

Proposition 7: Let

$$\begin{aligned} p(\mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}} \mid \mathbf{z}_s) &\sim \mathcal{N}(\mathbf{U}_{s_i}\mathbf{z}_s, \mathbf{\Pi}_{s_i}) \\ p(\mathbf{y}_{s++} \mid \mathbf{z}_s) &\sim \mathcal{N}(\tilde{\mathbf{U}}_s\mathbf{z}_s, \tilde{\mathbf{\Pi}}_s). \end{aligned} \tag{25}$$

Then, (20) and (21) reduce, respectively, to

$$\tilde{\mathbf{U}}_s = \begin{bmatrix} \mathbf{U}_{s_1} \\ \mathbf{U}_{s_2} \end{bmatrix} \quad \tilde{\mathbf{\Pi}}_s = \begin{bmatrix} \mathbf{\Pi}_{s_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Pi}_{s_2} \end{bmatrix} \tag{26}$$

and to

$$\begin{aligned} \mathbf{U}_s &= \begin{bmatrix} [\mathbf{0}, \mathbf{I}] \\ \tilde{\mathbf{U}}_s \end{bmatrix} \mathbf{F}_s \\ \mathbf{\Pi}_s &= \begin{bmatrix} \mathbf{Q}_s^{y,y} & [\mathbf{Q}_s^{y,x}, \mathbf{Q}_s^{y,y}]\, \tilde{\mathbf{U}}_s^T \\ \tilde{\mathbf{U}}_s \begin{bmatrix} \mathbf{Q}_s^{x,y} \\ \mathbf{Q}_s^{y,y} \end{bmatrix} & \tilde{\mathbf{\Pi}}_s + \tilde{\mathbf{U}}_s \mathbf{Q}_s \tilde{\mathbf{U}}_s^T \end{bmatrix}. \end{aligned} \tag{27}$$

Equations (26)–(27) enable to compute recursively the likelihood

$$p(\mathbf{y}) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Pi}_r) \tag{28}$$

the algorithm is initialized by $\{p(\mathbf{y}_{s_i} \mid \mathbf{z}_s) \sim \mathcal{N}([\mathbf{F}_{s_i}^{y,x} \mathbf{F}_{s_i}^{y,y}]\mathbf{z}_s, \mathbf{Q}_{s_i}^{y,y})\}_{s\in\mathcal{S}_{n-1}}$.

Proof of Proposition 7: Equations (26) are immediate. Using Proposition 9, we get

$$\begin{aligned} p(\mathbf{z}_s, \mathbf{y}_{s++} \mid \mathbf{z}_s^-) &= \underbrace{p(\mathbf{z}_s \mid \mathbf{z}_s^-)}_{\mathcal{N}(\mathbf{F}_s\mathbf{z}_s^-, \mathbf{Q}_s)} \times \underbrace{p(\mathbf{y}_{s++} \mid \mathbf{z}_s)}_{\mathcal{N}(\tilde{\mathbf{U}}_s\mathbf{z}_s, \tilde{\mathbf{\Pi}}_s)} \\ &\sim \mathcal{N}\left(\begin{bmatrix} \mathbf{F}_s\mathbf{z}_s^- \\ \tilde{\mathbf{U}}_s\mathbf{F}_s\mathbf{z}_s^- \end{bmatrix}, \begin{bmatrix} \mathbf{Q}_s & \mathbf{Q}_s\tilde{\mathbf{U}}_s^T \\ \tilde{\mathbf{U}}_s\mathbf{Q}_s & \tilde{\mathbf{\Pi}}_s + \tilde{\mathbf{U}}_s\mathbf{Q}_s\tilde{\mathbf{U}}_s^T \end{bmatrix} \right) \end{aligned}$$

whence (27), and similarly one can show that (22) reduces to (28). ■

Appendix

A. Proof of Proposition 1

Let $\mathbf{z}_{i:j} = \{\mathbf{z}_s\}_{s\in\mathcal{S}_i,\ldots,\mathcal{S}_j}$, and let us define $\mathbf{x}_{i:j}$ and $\mathbf{y}_{i:j}$ similarly. Using (2) and (3), we get

$$\begin{aligned} p(\mathbf{x}_{1:n}) &= \int p(\mathbf{z}_{1:n})\, d\mathbf{y}_{1:n} \\ &= \int p(\mathbf{z}_{1:n-1}) \left[\prod_{s\in\mathcal{S}_n} \int p(\mathbf{z}_s \mid \mathbf{z}_{s-})\, d\mathbf{y}_s \right] d\mathbf{y}_{1:n-1} \\ &= p(\mathbf{x}_{1:n-1}) \prod_{s\in\mathcal{S}_n} p(\mathbf{x}_s \mid \mathbf{x}_{s-}) \\ &= p(\mathbf{x}_r) \prod_{i=2}^{n} \prod_{s\in\mathcal{S}_i} p(\mathbf{x}_s \mid \mathbf{x}_{s-}) \end{aligned}$$

so $\mathbf{x}$ is an MT. Conversely, let $\mathbf{x}$ and $\mathbf{z}$ be both MT. Then, for all $s \in \mathcal{S} \setminus \mathcal{S}_n$

$$\begin{aligned} p\left(\mathbf{z}_s, \mathbf{z}_{s_1}, \mathbf{z}_{s_2}\right) &= \frac{p\left(\mathbf{z}_s, \mathbf{z}_{s_1}\right) p\left(\mathbf{z}_s, \mathbf{z}_{s_2}\right)}{p(\mathbf{z}_s)} \\ &= \frac{p\left(\mathbf{y}_s, \mathbf{y}_{s_1} \mid \mathbf{x}_s, \mathbf{x}_{s_1}\right) p\left(\mathbf{y}_s, \mathbf{y}_{s_2} \mid \mathbf{x}_s, \mathbf{x}_{s_2}\right)}{p(\mathbf{y}_s \mid \mathbf{x}_s)} \\ &\quad \times \underbrace{\frac{p(\mathbf{x}_s, \mathbf{x}_{s_1}) p(\mathbf{x}_s, \mathbf{x}_{s_2})}{p(\mathbf{x}_s)}}_{p(\mathbf{x}_s, \mathbf{x}_{s_1}, \mathbf{x}_{s_2})}. \end{aligned}$$

Integrating w.r.t. $\mathbf{y}_s, \mathbf{y}_{s_1}$, and $\mathbf{y}_{s_2}$, we get

$$\int \frac{p(\mathbf{y}_s \mid \mathbf{x}_s, \mathbf{x}_{s_1}) p(\mathbf{y}_s \mid \mathbf{x}_s, \mathbf{x}_{s_2})}{p(\mathbf{y}_s \mid \mathbf{x}_s)} d\mathbf{y}_s = 1. \tag{29}$$

Let $p_\omega^i(\mathbf{y}_s) = p(\mathbf{y}_s \mid \mathbf{x}_s, \mathbf{x}_{s_i} = \omega)$. By assumption

$$p_\omega^1(\mathbf{y}_s) = p_\omega^2(\mathbf{y}_s). \tag{30}$$

Using (30) and then (29), we get

$$\begin{aligned} &\int (p_\omega^1(\mathbf{y}_s) - p_{\omega'}^1(\mathbf{y}_s))^2 \frac{1}{p(\mathbf{y}_s \mid \mathbf{x}_s)} d\mathbf{y}_s \\ &= \underbrace{\int \frac{p_\omega^1(\mathbf{y}_s) p_\omega^2(\mathbf{y}_s)}{p(\mathbf{y}_s \mid \mathbf{x}_s)} d\mathbf{y}_s}_{1} \\ &\quad + \underbrace{\int \frac{p_{\omega'}^1(\mathbf{y}_s) p_{\omega'}^2(\mathbf{y}_s)}{p(\mathbf{y}_s \mid \mathbf{x}_s)} d\mathbf{y}_s}_{1} - 2 \underbrace{\int \frac{p_\omega^1(\mathbf{y}_s) p_{\omega'}^2(\mathbf{y}_s)}{p(\mathbf{y}_s \mid \mathbf{x}_s)} d\mathbf{y}_s}_{1} = 0. \end{aligned}$$

So, $p_\omega^1(\mathbf{y}_s) = p_{\omega'}^1(\mathbf{y}_s)$ (and, similarly, $p_\omega^2(\mathbf{y}_s) = p_{\omega'}^2(\mathbf{y}_s)$), which proves that conditionally on $\mathbf{x}_s, \mathbf{x}_{s_i}$, and $\mathbf{y}_s$ are independent. ■

B. Proof of Proposition 2

From P1), we have for $i = 1, 2$

$$p\left(\mathbf{x}_{s_i}, \mathbf{z}_s \,\middle|\, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}}\right) = p\left(\mathbf{x}_{s_i} \,\middle|\, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}}\right) p\left(\mathbf{z}_s \,\middle|\, \mathbf{z}_{s_i}, \mathbf{y}_{s_i^{++}}\right) \stackrel{(P1)}{=} p\left(\mathbf{x}_{s_i} \,\middle|\, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}}\right) p\left(\mathbf{z}_s \,|\, \mathbf{z}_{s_i}\right). \quad (31)$$

We first need to compute $p(\mathbf{z}_s \,|\, \mathbf{z}_{s_i})$ from $p(\mathbf{z}_s)$ and $p(\mathbf{z}_{s_i} \,|\, \mathbf{z}_s)$. Using Proposition 9, (6), and Proposition 8, we get

$$p\left(\mathbf{z}_s, \mathbf{z}_{s_i}\right) = \underbrace{p(\mathbf{z}_s)}_{\mathcal{N}(\mathbf{0}, \mathbf{P}_s)} \underbrace{p(\mathbf{z}_{s_i} \,|\, \mathbf{z}_s)}_{\mathcal{N}(\mathbf{F}_{s_i}\mathbf{z}_s, \mathbf{Q}_{s_i})} \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{P}_s & \mathbf{P}_s\mathbf{F}_{s_i}^T\\ \mathbf{F}_{s_i}\mathbf{P}_s & \mathbf{P}_{s_i}\end{bmatrix}\right) \quad (32)$$

and so $p(\mathbf{z}_s \,|\, \mathbf{z}_{s_i}) \sim \mathcal{N}(\tilde{\mathbf{F}}_{s_i}\mathbf{z}_{s_i}, \tilde{\mathbf{Q}}_{s_i})$, in which $\tilde{\mathbf{F}}_{s_i}$ and $\tilde{\mathbf{Q}}_{s_i}$ are given by (9) and (10).

We next turn back to the computation of (31). Using Proposition 9, we have (33), as shown at the bottom of the page, from which we deduce (7) and (8). ■

C. Proof of Proposition 3

We are going to compute $p(\mathbf{z}_s \,|\, \mathbf{y}_{s^{++}})$ from $p(\mathbf{z}_s, \mathbf{y}_{s^{++}}) = p(\mathbf{z}_s)p(\mathbf{y}_{s^{++}} \,|\, \mathbf{z}_s)$. From (P2), we have

$$p(\mathbf{y}_{s^{++}} \,|\, \mathbf{z}_s) = p\left(\mathbf{y}_{s_1}, \mathbf{y}_{s_1^{++}} \,\middle|\, \mathbf{z}_s\right) p\left(\mathbf{y}_{s_2}, \mathbf{y}_{s_2^{++}} \,\middle|\, \mathbf{z}_s\right). \quad (34)$$

On the other hand, due to model (4)

$$\begin{bmatrix}\mathbf{y}_{s_i}\\ \mathbf{y}_{s_i^{++}}\end{bmatrix} = \mathbf{U}_{s_i}\mathbf{z}_s + \mathbf{V}_{s_i}\begin{bmatrix}\mathbf{w}_{s_i}\\ \mathbf{w}_{s_i^{++}}\end{bmatrix} \quad (35)$$

for some matrices $\mathbf{U}_{s_i}$ and $\mathbf{V}_{s_i}$. Let $\mathbf{\Pi}_{s_i} = \mathrm{Cov}(\mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}} \,|\, \mathbf{z}_s)$. Since $\mathbf{z}_s$ and $[\mathbf{w}_{s_i}^T, \mathbf{w}_{s_i^{++}}^T]^T$ are independent, (35) yields

$$p(\mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}} \,|\, \mathbf{z}_s) \sim \mathcal{N}(\mathbf{U}_{s_i}\mathbf{z}_s, \mathbf{\Pi}_{s_i}). \quad (36)$$

Next, using Proposition 9, we get (37), as shown at the bottom of the page. We can now compute $p(\mathbf{z}_s \,|\, \mathbf{y}_{s^{++}})$ with the help of Proposition 8. Using the matrix inversion lemma (45), we get

$$\mathbf{P}_{s\,|\,s^{++}} = \left[\mathbf{P}_s^{-1} + \sum_{i=1}^{2}\mathbf{U}_{s_i}^T\mathbf{\Pi}_{s_i}^{-1}\mathbf{U}_{s_i}\right]^{-1} \quad (38)$$

$$\mathbf{P}_{s\,|\,s_i, s_i^{++}} = \left[\mathbf{P}_s^{-1} + \mathbf{U}_{s_i}^T\mathbf{\Pi}_{s_i}^{-1}\mathbf{U}_{s_i}\right]^{-1} \quad (39)$$

from which we deduce (12). On the other hand, using (46), (38), and (39), we get

$$\hat{\mathbf{z}}_{s\,|\,s^{++}} = \mathbf{P}_{s\,|\,s^{++}}\left[\sum_{i=1}^{2}\mathbf{U}_{s_i}^T\mathbf{\Pi}_{s_i}^{-1}\begin{bmatrix}\mathbf{y}_{s_i}\\ \mathbf{y}_{s_i^{++}}\end{bmatrix}\right] \quad (40)$$

$$\hat{\mathbf{z}}_{s\,|\,s_i, s_i^{++}} = \mathbf{P}_{s\,|\,s_i, s_i^{++}}\mathbf{U}_{s_i}^T\mathbf{\Pi}_{s_i}^{-1}\begin{bmatrix}\mathbf{y}_{s_i}\\ \mathbf{y}_{s_i^{++}}\end{bmatrix} \quad (41)$$

from which we deduce (11). ■

D. Proof of Proposition 5

We are going to compute $p(\mathbf{x}_{s_i} \,|\, \mathbf{y})$ via

$$p(\mathbf{x}_{s_i} \,|\, \mathbf{y}) = \int p(\mathbf{x}_s \,|\, \mathbf{y}) p\left(\mathbf{x}_{s_i} \,|\, \mathbf{x}_s, \mathbf{y}\right) d\mathbf{x}_s. \quad (42)$$

Now, from (P1) and (P2), one can show easily that

$$p\left(\mathbf{x}_{s_i} \,|\, \mathbf{x}_s, \mathbf{y}\right) = p\left(\mathbf{x}_{s_i} \,\middle|\, \mathbf{x}_s, \mathbf{y}_s, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}}\right). \quad (43)$$

On the other hand, $p(\mathbf{x}_{s_i}, \mathbf{z}_s \,|\, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}})$ has already been computed [see (33)]. Using Proposition 8, (7), and (8), we see that

$$p\left(\mathbf{x}_{s_i} \,\middle|\, \mathbf{z}_s, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}}\right) \sim \mathcal{N}\left(\hat{\mathbf{x}}_{s_i\,|\,s_i, s_i^{++}} + \mathbf{J}_{s_i}\left(\mathbf{z}_s - \hat{\mathbf{z}}_{s\,|\,s_i, s_i^{++}}\right), \mathbf{C}_{s_i}\right) \quad (44)$$

in which $\mathbf{J}_{s_i}$ is given by (15), and $\mathbf{C}_{s_i} = \mathbf{P}^{x,x}_{s_i\,|\,s_i, s_i^{++}} - \mathbf{J}_{s_i}\mathbf{P}_{s\,|\,s_i, s_i^{++}}\mathbf{J}_{s_i}^T$. So $p(\mathbf{x}_{s_i} \,|\, \mathbf{x}_s,\ \mathbf{y}_s, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}}) \sim \mathcal{N}(\mathcal{A}_{s_i}\mathbf{x}_s + \mathbf{b}_{s_i}, \mathbf{C}_{s_i})$ for some matrices $\mathcal{A}_{s_i}$ and $\mathbf{b}_{s_i}$. Coming back to (42), we see from Proposition 9 that

$$p(\mathbf{x}_{s_i} \,|\, \mathbf{y}) = \int \underbrace{p(\mathbf{x}_s \,|\, \mathbf{y})}_{\mathcal{N}(\hat{\mathbf{x}}_{s\,|\,S}, \mathbf{P}^{x,x}_{s\,|\,S})} \underbrace{p\left(\mathbf{x}_{s_i} \,\middle|\, \mathbf{x}_s, \mathbf{y}_s, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}}\right)}_{\mathcal{N}(\mathcal{A}_{s_i}\mathbf{x}_s + \mathbf{b}_{s_i}, \mathbf{C}_{s_i})} d\mathbf{x}_s = \mathcal{N}\left(\mathcal{A}_{s_i}\hat{\mathbf{x}}_{s\,|\,S} + \mathbf{b}_{s_i}, \mathbf{C}_{s_i} + \mathcal{A}_{s_i}\mathbf{P}^{x,x}_{s\,|\,S}\mathcal{A}_{s_i}^T\right)$$

from which we deduce (16) and (17). ■

E. Some Useful Identities

The derivations of Sections III and IV rely on the following two properties of Gaussian random variables, which are recalled for convenience of the reader.

Proposition 8: Let

$$p(\mathbf{u}_1, \mathbf{u}_2) \sim \mathcal{N}\left(\begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix}, \begin{bmatrix}\mathbf{\Sigma}_{1,1} & \mathbf{\Sigma}_{1,2}\\ \mathbf{\Sigma}_{2,1} & \mathbf{\Sigma}_{2,2}\end{bmatrix}\right).$$

$$p\left(\mathbf{x}_{s_i}, \mathbf{z}_s \,|\, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}}\right) = \underbrace{p(\mathbf{x}_{s_i} \,|\, \mathbf{y}_{s_i}, \mathbf{y}_{s_i^{++}})}_{\mathcal{N}(\hat{\mathbf{x}}_{s_i\,|\,s_i, s_i^{++}}, \mathbf{P}^{x,x}_{s_i\,|\,s_i, s_i^{++}})} \underbrace{p(\mathbf{z}_s \,|\, \mathbf{x}_{s_i}, \mathbf{y}_{s_i})}_{\mathcal{N}(\tilde{\mathbf{F}}_{s_i}\mathbf{z}_{s_i}, \tilde{\mathbf{Q}}_{s_i})}$$
$$\sim \mathcal{N}\left(\begin{bmatrix}\hat{\mathbf{x}}_{s_i\,|\,s_i, s_i^{++}}\\ \tilde{\mathbf{F}}_{s_i}\begin{bmatrix}\hat{\mathbf{x}}_{s_i\,|\,s_i, s_i^{++}}\\ \mathbf{y}_{s_i}\end{bmatrix}\end{bmatrix}, \begin{bmatrix}\mathbf{P}^{x,x}_{s_i\,|\,s_i, s_i^{++}} & \mathbf{P}^{x,x}_{s_i\,|\,s_i, s_i^{++}}\left[\left(\tilde{\mathbf{F}}^{x,x}_{s_i}\right)^T \left(\tilde{\mathbf{F}}^{y,x}_{s_i}\right)^T\right]\\ \begin{bmatrix}\tilde{\mathbf{F}}^{x,x}_{s_i}\\ \tilde{\mathbf{F}}^{y,x}_{s_i}\end{bmatrix}\mathbf{P}^{x,x}_{s_i\,|\,s_i, s_i^{++}} & \tilde{\mathbf{Q}}_{s_i} + \begin{bmatrix}\tilde{\mathbf{F}}^{x,x}_{s_i}\\ \tilde{\mathbf{F}}^{y,x}_{s_i}\end{bmatrix}\mathbf{P}^{x,x}_{s_i\,|\,s_i, s_i^{++}}\left[\left(\tilde{\mathbf{F}}^{x,x}_{s_i}\right)^T \left(\tilde{\mathbf{F}}^{y,x}_{s_i}\right)^T\right]\end{bmatrix}\right) \quad (33)$$

$$p(\mathbf{z}_s, \mathbf{y}_{s^{++}}) = \underbrace{p(\mathbf{z}_s)}_{\mathcal{N}(\mathbf{0}, \mathbf{P}_s)} \underbrace{p\left(\mathbf{y}_{s_1}, \mathbf{y}_{s_1^{++}}, \mathbf{y}_{s_2}, \mathbf{y}_{s_2^{++}} \,|\, \mathbf{z}_s\right)}_{\mathcal{N}\left(\begin{bmatrix}\mathbf{U}_{s_1}\mathbf{z}_s\\ \mathbf{U}_{s_2}\mathbf{z}_s\end{bmatrix}, \begin{bmatrix}\mathbf{\Pi}_{s_1} & \mathbf{0}\\ \mathbf{0} & \mathbf{\Pi}_{s_2}\end{bmatrix}\right)} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}\mathbf{P}_s & \mathbf{P}_s[\mathbf{U}_{s_1}^T\mathbf{U}_{s_2}^T]\\ \begin{bmatrix}\mathbf{U}_{s_1}\\ \mathbf{U}_{s_2}\end{bmatrix}\mathbf{P}_s & \begin{bmatrix}\mathbf{\Pi}_{s_1} & \mathbf{0}\\ \mathbf{0} & \mathbf{\Pi}_{s_2}\end{bmatrix} + \begin{bmatrix}\mathbf{U}_{s_1}\\ \mathbf{U}_{s_2}\end{bmatrix}\mathbf{P}_s\left[\mathbf{U}_{s_1}^T\mathbf{U}_{s_2}^T\right]\end{bmatrix}\right) \quad (37)$$

Then, $p(\mathbf{u}_1 \mid \mathbf{u}_2) \sim \mathcal{N}(\mu_{1\mid 2}, \boldsymbol{\Sigma}_{1\mid 2})$, with $\mu_{1\mid 2} = \mu_1 + \boldsymbol{\Sigma}_{1,2}\boldsymbol{\Sigma}_{2,2}^{-1}(\mathbf{u}_2 - \mu_2)$, $\boldsymbol{\Sigma}_{1\mid 2} = \boldsymbol{\Sigma}_{1,1} - \boldsymbol{\Sigma}_{1,2}\,\boldsymbol{\Sigma}_{2,2}^{-1}\boldsymbol{\Sigma}_{2,1}$.

Proposition 9: Let $p(\mathbf{u}_1) \sim \mathcal{N}(\mu_1, \boldsymbol{\Sigma}_1)$ and $p(\mathbf{u}_2 \mid \mathbf{u}_1) \sim \mathcal{N}(\mathbf{A}\mathbf{u}_1 + \mathbf{b}, \boldsymbol{\Sigma}_{2\mid 1})$. Then

$$p(\mathbf{u}_1, \mathbf{u}_2) \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mathbf{A}\mu_1 + \mathbf{b} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_1\mathbf{A}^T \\ \mathbf{A}\boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_{2\mid 1} + \mathbf{A}\boldsymbol{\Sigma}_1\mathbf{A}^T \end{bmatrix}\right).$$

The following well-known identities are also useful in this note:

$$(\mathbf{A} + \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} + \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} \tag{45}$$

$$(\mathbf{A} + \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}\mathbf{B}\mathbf{D}^{-1} = \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} + \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \tag{46}$$

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{I} & -\mathbf{A}^{-1}\mathbf{B} \\ -\mathbf{D}^{-1}\mathbf{C} & \mathbf{I} \end{bmatrix} \begin{bmatrix} (\mathbf{M}/\mathbf{D})^{-1} & \mathbf{0} \\ \mathbf{O} & (\mathbf{M}/\mathbf{A})^{-1} \end{bmatrix} \tag{47}$$

$$(\mathbf{M}/\mathbf{D}) = \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C} \tag{48}$$

$$(\mathbf{M}/\mathbf{A}) = \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}. \tag{49}$$

REFERENCES

[1] M. Basseville, A. Benveniste, K. C. Chou, S. A. Golden, R. Nikoukhah, and A. S. Willsky, "Modeling and estimation of multiresolution stochastic processes," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 766–784, Mar. 1992.

[2] C. A. Bouman and M. Shapiro, "A multiscale random field model for Bayesian image segmentation," *IEEE Trans. Image Process.*, vol. 3, no. 3, pp. 162–177, Mar. 1994.

[3] K. C. Chou, A. S. Willsky, and A. Benveniste, "Multiscale recursive estimation, data fusion, and regularization," *IEEE Trans. Autom. Control*, vol. 39, no. 3, pp. 464–478, Mar. 1994.

[4] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using Hidden Markov models," *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 886–902, Apr. 1998.

[5] R. H. Riedi, M. S. Crouse, V. J. Ribeiro, and R. G. Baraniuk, "A multifractal wavelet model with application to network traffic," *IEEE Trans. Inf. Theory*, vol. 45, no. 3, pp. 992–1018, Apr. 1999.

[6] R. Nowak, "Multiscale hidden Markov models for Bayesian image analysis," in *Bayesian Inference in Wavelet Based Models*, B. Vidakovic and P. Muller, Eds. New York: Springer-Verlag, 1999.

[7] J. M. Laferté, P. Perez, and F. Heitz, "Discrete Markov image modeling and inference on the quadtree," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 390–404, Mar. 2000.

[8] A. S. Willsky, "Multiresolution Markov models for signal and image processing," *Proc. IEEE*, vol. 90, no. 8, pp. 1396–1458, Aug. 2002.

[9] W. Pieczynski and A. N. Tebbache, "Pairwise Markov random fields and segmentation of textured images," *Machine Graph. Vision*, vol. 9, no. 3, pp. 705–718, Mar. 2000.

[10] W. Pieczynski, "Pairwise Markov chains," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 5, pp. 634–639, May 2003.

[11] W. Pieczynski and F. Desbouvries, "Kalman filtering using pairwise Gaussian models," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing (ICASSP 2003)*, vol. 6, Hong Kong, Apr. 6–10, 2003, pp. VI-57–VI-60.

[12] F. Desbouvries and W. Pieczynski, "Particle filtering in pairwise and triplet Markov chains," in *Proc. IEEE—EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP 2003)*, Grado-Gorizia, Italy, Jun. 8–11, 2003.

[13] W. Pieczynski, "Arbres de Markov couple—Pairwise Markov trees" (in French), *Comptes Rendus de l'Académie des Sciences—Mathématiques*, ser. I, vol. 335, pp. 79–82, 2002.

[14] E. Monfrini, J. Lecomte, F. Desbouvries, and W. Pieczynski, "Image and signal restoration using pairwise Markov trees," in *Proc. 2003 IEEE Workshop on Statistical Signal Processing*, St. Louis, MI, Sep. 2003.

[15] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA J.*, vol. 3, no. 8, pp. 1445–1450, Aug. 1965.

[16] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.

[17] M. J. Wainwright, E. P. Simoncelli, and A. S. Willsky, "Random cascades on wavelet trees and their use in modeling natural images," *Appl. Comput. Harmon. Anal.*, vol. 11, pp. 89–123, 2001.

Analysis and Estimation of Tracking Errors of Plug-in Type Repetitive Control Systems

Wu-Sung Yao and Mi-Ching Tsai

***Abstract*—Subject to stability constraints, tracking performance of a plug-in type repetitive control system with periodic inputs is explored in this note. An upper bound for the square integral of the tracking error over one time period of periodic input signals is derived based on Fourier analysis. The magnitude and the decay rate of the tracking error can be estimated using the harmonics of the input signal within the designed bandwidth. Both computer simulations and experimental results are presented to illustrate the effectiveness of the proposed method. It is also shown that the tracking error can decay significantly in the first two time periods in the proposed repetitive control system if the required bandwidth conditions are satisfied.**

***Index Terms*—Fourier analysis, plug-in, repetitive control, tracking error.**

I. INTRODUCTION

Repetitive control [1]–[8] is one area in which the objective is mainly to remove the steady-state error due to periodic inputs. Its highly accurate tracking property is due to a periodic signal generator implemented in the repetitive controller. Hara *et al.* [2] derived sufficient conditions for the stability of a repetitive control system and a modified repetitive control system, which sacrifices tracking performance at high frequencies for system stability. Srinivasan and Shaw [9] examined the absolute and relative stability of repetitive control systems using the regeneration spectrum and showed that their results provide improved insights into design tradeoffs. Moon *et al.* [7] designed a repetitive controller using a graphical technique based on the frequency-domain analysis of a linear interval system. Li and Tsao [6] addressed the analysis and synthesis of robust stability and robust performance repetitive control systems. Simple and intuitive design procedures of the proposed repetitive controller were developed in [10]. In this note, interest is in the analysis and estimation of the tracking error of the plug-in type repetitive control system [10]. The proposed method offers better insights into the repetitive control system used to check the magnitude and convergence of the tracking error when considering stability and performance. More details of the tracking errors in a designed example [10] are given in the following.

Consider the standard unity feedback control system shown in Fig. 1(a), where $G_p(s)$ denotes the loop transfer function, r the reference periodic input of period T_d, y the system output, d the disturbance, and e the tracking error. Fig. 1(b) shows a plug-in type repetitive controller with two designed parameters $K_q(s)$ and $K_b(s)$ introduced to improve the overall closed-loop bandwidth. To illustrate the tracking responses resulting from the proposed control scheme, consider a designed-loop transfer function for a linear servo system [10], which is given by

$$G_p(s) = \frac{27\,450\,525}{s^3 + 742.32s^2 + 202\,662.4s}. \tag{1}$$

Then, the unity feedback system of Fig. 1(a) is a stable system, which has been designed for good tracking for nonperiodical inputs. Let the

Manuscript received June 2, 2003; revised May 17, 2004, October 31, 2004, and April 21, 2005. Recommended by Associate Editor Y. Ohta.

The authors are with the Department of Mechanical Engineering, National Cheng Kung University, Tainan 701, Taiwan, R.O.C. (e-mail: mctsai@mail.ncku.edu.tw).

Digital Object Identifier 10.1109/TAC.2005.852553