BAYESIAN DLL FOR MULTIPATH MITIGATION IN NAVIGATION SYSTEMS USING PARTICLE FILTERS

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

Universitat Politècnica de Catalunya
Dept. of Signal Theory and Communications
Jordi Girona 1–3, Edifici D5 Campus Nord UPC, 08034 Barcelona, Spain.
e-mail: {closas,carlos,juan}@gps.tsc.upc.edu
web: http://gps-tsc.upc.es/comm

ABSTRACT

In direct–sequence spread–spectrum (DS–SS) navigation based systems, multipath can degrade seriously synchronization performance causing time delay and code phase estimates, to deviate from the actual value. This bias depends on the relative amplitudes and delays of multipath replicas with respect to the direct signal. The error in the estimated position due to multipath, when using a standard Delay Lock Loop, can be on the order of several tens of meters, which is a critical aspect in high–precision applications. This works presents a Sequential Monte Carlo based algorithm which tries to iteratively estimate complex amplitudes and delays of the direct signal and multipath replicas by characterizing the posterior probability density function of these parameters relying on particle filter theory. Simulations are presented for navigation systems, which are particular applications of DS-SS systems.

1. INTRODUCTION

Global Navigation Satellite Systems (GNSS) is the general concept used to identify those systems that allow user positioning based on a constellation of satellites. Specific GNSS systems are the well-known american GPS or the forthcoming european Galileo. Both based on the same principle: the user computes its position based on the distances between its receiver and the set of in-view satellites. These distances are calculated estimating the propagation time that transmitted signals take from each satellite to the receiver. At least 4 satellites are needed in order to compute user position [1].

Each satellite is uniquely identified by its own DS–SS signal, which are transmitted synchronously by all satellites. GNSS receivers are only interested in estimating delays of signals received directly from the satellites, hereafter referred to as line-of-sight-signal (LOSS), since they are the ones that carry information of direct propagation time. Hence, reflections, distort the received signal in a way that may cause a bias in delay and carrier–phase estimations. Thus, multipath is probably the dominant source of error in high–precision applications since it can introduce a bias up to a hundred of meters when employing a 1-chip wide (standard) delay locked loop (DLL) to track the delay, which is a common synchronization method used in DS–SS receivers [2]. To the aim of reducing this annoying effect, several methods have been proposed such as the Narrow Correlator, the MEDLL, the Pulse Aperture Correlator (PAC) or the Strobe Correlator, which are studied and compared in [2]. Although they achieve much better results than the conventional DLL in terms of multipath-caused timing bias, they do not mitigate this effect completely. Specially troublesome is the case of coherent multipath, replicas with relative delays shorter than the chip period, where timing synchronization may fail.

In this paper, a coherent DLL is proposed to the aim of reducing the effect of multipath on code timing estimation. It takes into account statistical properties of both LOSS and multipath delays to achieve a Bayesian solution, assuming a model that is a function of amplitudes and delays, which are the unknown parameters to estimate. The novelty relies on the use of a Sequential Monte–Carlo (SMC) based algorithm [3] to recursively obtain the *posterior* probability density function (pdf) of the parameters of interest given the data–set received, in an iterative parameter estimation algorithm.

2. SIGNAL MODEL

If there are N satellites in view, each LOSS affected by $M_n - 1$ multipath signals, the received complex baseband DS–SS signal in additive white gaussian noise can be modeled as

$$x(t) = \sum_{j=1}^{N} \sum_{i=0}^{M_j - 1} \alpha_{ji} q_j(t - \tau_{ji}) e^{j\phi_{ji}} + w(t) \qquad (1)$$

This work was financed by the Spanish/Catalan Science and Technology Commissions and FEDER funds from the European Commission: TIC2003-05482, TEC2004-04526 and 2001SGR-00268.

where α_{ji}, τ_{ji} and ϕ_{ji} are the amplitude, delay and carrier phase of the *i–th* signal of the *j–th* satellite, respectively. Notice that the subscript $i = 0$ stands for the line–of–sight–signal (LOSS). Due to physical reasons, it is considered that

$$\begin{aligned} |\alpha_{ji}| &< |\alpha_{j0}| \\ \tau_{ji} &> \tau_{j0} \qquad \forall i = \{1, \ldots, M-1\} \end{aligned} \tag{2}$$

$q_j(t)$ is the DS–SS signal of the *j–th* satellite, formed with the sequence of data symbols, $\{d_j(l)\}$, and its pseudo–noise code sequence $\{c_j(n)\}$ which spreads to a rate equal to the chip period, T_c. Data symbols are transmitted at a lower bit rate, T_b. $g(t)$ is the chip-shaping pulse.

$$q_j(t) = \sum_{l=-\infty}^{\infty} d_j(l) p_j(t - lT_b) \tag{3}$$

$$p_j(t) = \sum_{n=0}^{P-1} c_j(n) g(t - nT_c) \tag{4}$$

Correlating the input signal with a filter matched to the $p_j(t)$ sequence, is commonly known as *despreading*. Hence, for the received signal model in (1), the despreading process for a given satellite results in

$$y(t) = \alpha_0 R(t - \tau_0) e^{j\phi_0} + \sum_{i=1}^{M-1} \alpha_i R(t - \tau_i) e^{j\phi_i} + n(t) \tag{5}$$

being $R(t)$ the code autocorrelation function of the given satellite and $n(t)$ contains white gaussian noise and spread spectrum interferences of the other satellites. If phases are taken into account in the estimation procedure, it is said to be the decision–directed phase–directed case, and a coherent DLL is used. On the other hand, if phases are neglected in the estimation, removed for example by a squaring stage, it is the case of non–data–aided phase–independent criterion, using a non-coherent DLL for code timing estimation.

If K snapshots are assumed to be recorded, the despreading model in (5) can be expressed in matrix form as,

$$\mathbf{y} = \mathbf{a}\mathbf{R}(\boldsymbol{\tau}) + \mathbf{n} \tag{6}$$

where $\mathbf{a}, \boldsymbol{\tau} \in \mathbb{R}^{1\times M}$ are column vectors containing amplitudes and delays of all M signals, $\mathbf{R}(\boldsymbol{\tau}) \in \mathbb{R}^{M\times K}$ is the matrix containing samples of the despreaded M signals which only differences are their delays, denoted by $\boldsymbol{\tau}$. The composite signal and the zero–mean additive white gaussian noise are expressed as $\mathbf{y}, \mathbf{n} \in \mathbb{R}^{1\times K}$ respectively, which covariance matrix is denoted by $\mathbf{C}_m$.

3. BAYESIAN DLL FOR CODE TIMING ESTIMATION

SMC methods adopt the state-space approach for modeling system evolution over time. These methods deal with the non-linear filtering problem, this is to recursively compute estimates of states $\mathbf{x}_k \in \mathbb{R}^{n_x \times 1}$ given measurements $\mathbf{y}_k \in \mathbb{C}^{n_y \times K}$ at time index k. State equation models the evolution of target states as a discrete–time stochastic model, in general

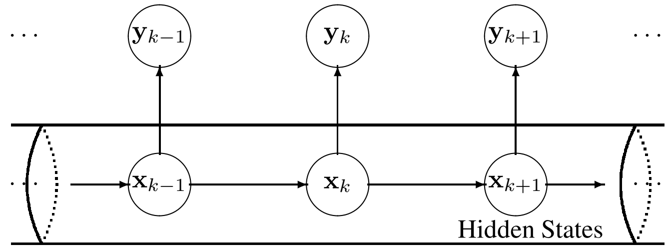

Fig. 1. State–space as a Markov process of order one

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}) \tag{7}$$

where $\mathbf{f}_{k-1}$ is a known, possibly non-linear, function of the previous state $\mathbf{x}_{k-1}$ and $\mathbf{v}_{k-1}$ is the process noise which gathers any mismodeling effect or disturbances in the state characterization, with covariance matrix $\mathbf{C}_{s_k}$. The relation between measurements and states is modeled by

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{w}_k) \tag{8}$$

where $\mathbf{h}_k$ is a known possibly non-linear function and $\mathbf{w}_k$ is referred to as measurement noise, with covariance matrix $\mathbf{C}_{m_k}$. Both process and measurement noise are assumed white, with known statistics and mutually independent. The initial *a priori* pdf of the state vector is assumed known, $p(\mathbf{x}_0)$.

Equations (7) and (8) show that the assumed state–space model describes a Markov process of order one, since states at time instant k only depend on states on the previous instant, $k-1$, and on process noise. On the other hand, states are hidden and the only information available are measurements at every time step k, $\mathbf{y}_k$, which are also independent among them. This idea is graphically shown in Fig. 1.

The problem consists on computing filtered estimates of $\mathbf{x}_k$ taking into account all available measurement up to time k, $\mathbf{Y}_k = \{\mathbf{y}_i, i = 1, \ldots, k\}$. From a Bayesian point of view, the solution resides in recursively obtain the *a posteriori* pdf of states at time k given all available measurements, $p(\mathbf{x}_k|\mathbf{Y}_k)$. Particle Filter methods use the Sequential Importance Sampling (SIS) concept to characterize this density. Basically it involves the approximation of the posterior by a set of N_s random samples taken from an *importance density function*, $\mathbf{x}^i \sim \pi(\mathbf{x})$, with associated importance weights w^i. Where the importance function has the same support as the true posterior. The posterior approximation is,

$$\hat{p}(\mathbf{x}_k|\mathbf{Y}_k) = \sum_{i=1}^{N_s} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{9}$$

being $\delta(\cdot)$ the Dirac's delta function. This approximation converges almost surely to the true posterior as $N_s \to \infty$ [4]

On the sequel, we assume that the received signal is digitized at a frequency rate of f_s and correlated with the desired satellite signature. Then, equation (6) models the measurements, and the state-vector is composed by the $2M$ unknown parameters: amplitudes and delays. Thus, $\mathbf{x}_k = \left[\boldsymbol{\tau}^T, \mathbf{a}^T\right]^T$, where $\boldsymbol{\tau}$ and $\mathbf{a}$ are the delays and amplitudes column vectors respectively. Although no time evolution is assumed in static parameter estimation, it is convenient to add some perturbations to the state-parameter equation to avoid the known effect of *sample impoverishment*,

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \boldsymbol{\xi}_k \tag{10}$$

where $\boldsymbol{\xi}_k$ is the process noise at time instant k with known covariance matrix $\mathbf{C}_{s_k}$.

It is desirable to reduce state vector to a lower dimensional space to accelerate convergence and reduce the minimum N_s needed. To this aim, we notice that, for a given delay $\boldsymbol{\tau}^i$, we can obtain the Least Squares estimation of amplitudes from measurement equation (6) in a straightforward operation,

$$\hat{\mathbf{a}}_{LS}^i = \mathbf{y}\mathbf{R}(\boldsymbol{\tau}^i)^H \left(\mathbf{R}(\boldsymbol{\tau}^i)\mathbf{R}(\boldsymbol{\tau}^i)^H\right)^{-1} \tag{11}$$

which coincides with the Maximum Likelihood ($\hat{\mathbf{a}}_{ML}^i$) solution under the assumption of zero-mean gaussian noise, which holds. With this procedure, the dimension of the state-space is reduced to M parameters.

3.1. Algorithm description

In a GNSS receiver, data from the correlation modules is not sequentially processed but batch processed, following the Software Defined Radio (SDR) philosophy. Hence, we can define N_t in two ways: as the number of correlation outputs needed by the Bayesian DLL to achieve a certain accuracy or as the number of iterations performed for a given $\mathbf{y}_k$, in both cases $k = 1, \ldots, N_t$. The latter is the definition adopted hereafter. Although equivalent to the other one in terms of algorithm design, the underlying idea behind is quite different. Below we describe the operation of the proposed Bayesian DLL:

1. **Initialization**. The degree of uncertainty after the acquisition stage, previous to the DLL tracking, is $\pm T_c/2$ for the LOSS delay. Hence, the initial ambiguity of the Bayesian DLL algorithm is equal to T_c for τ_0, the LOSS, and $2T_c$ for multipath delays $(\tau_1, \cdots, \tau_{M-1})$ with respect to τ_0, since it makes no sense to look for replicas further, as they will not affect the estimation of τ_0. Thus, all N_s particles are initialized with the same value taking into account the a priori state uncertainty, given by $\mathbf{C}_{s_0}$. At time index $k = 0$,

$$\begin{aligned} \left\{\hat{\tau}_0^i\right\}_{i=1}^{N_s} &= \hat{\tau}_0 \sim \mathcal{U}\left(-T_c/2, T_c/2\right) \\ \left\{\hat{\boldsymbol{\tau}}_{mp}^i\right\}_{i=1}^{N_s} &= \hat{\boldsymbol{\tau}}_{mp} \sim \mathcal{U}\left(\hat{\tau}_0, \hat{\tau}_0 + 2T_c\right) \end{aligned} \tag{12}$$

where $\boldsymbol{\tau} = [\tau_0, \boldsymbol{\tau}_{mp}^T]^T$ is the delay parameters vector.

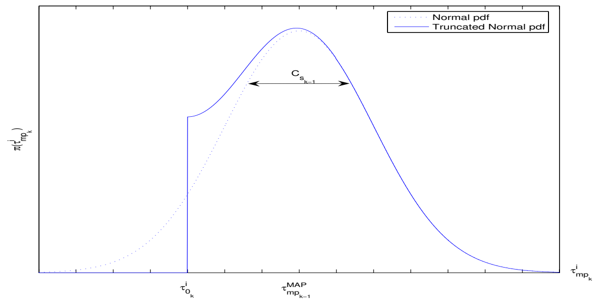

Fig. 2. Importance density function, $\pi\left(\tau_{mp_k}\right)$, for replica ℓ. Being $\sigma_{mp_\ell}^2 = \mathbf{C}_s(\ell+1, \ell+1)$ and $\ell = 1, \ldots, M-1$.

2. **Importance Sampling**. The SIS algorithm computes a set of support points from the importance density function $\pi(\cdot)$. The choosing of $\pi(\cdot)$ is a critical issue in any particle filter design. As seen in equation (11), states defining amplitudes have been analytically solved given a set of delays, $\boldsymbol{\tau}$. Hence, the importance density will only be a function of delay vector, and we can write $\boldsymbol{\pi}\left(\boldsymbol{\tau}\right)$. A common solution is to adopt the prior density as the importance, here we propose a Gaussian importance density function for LOSS delay particle generation with mean a previous estimation and variance the one computed from the posterior $p(\mathbf{x}_{k-1}|\mathbf{y})$

$$\tau_{0_k}^i \sim \pi\left(\hat{\tau}_{0_{k-1}}, \mathbf{C}_{s_{k-1}}\right) = \mathcal{N}\left(\hat{\tau}_{0_{k-1}}, \mathbf{C}_{s_{k-1}}(1,1)\right) \tag{13}$$

On the other hand, taking into account relations in equation (2), a truncated Gaussian importance density function is proposed for multipath delays generation. This is a normal multivariate density function with mean and covariance obtained in time step $k-1$ (as in the LOSS case) but constraining the multipath delays, in the i–th particle, to be always higher than its correspondent LOSS delay particle, $\tau_{0_k}^i$, hence

$$\begin{aligned} \boldsymbol{\tau}_{mp_k}^i &\sim \boldsymbol{\pi}\left(\hat{\boldsymbol{\tau}}_{mp_{k-1}}, \mathbf{C}_{s_{k-1}}\right) = \\ &= \tau_{0_k}^i + \left|\left(\hat{\boldsymbol{\tau}}_{mp_{k-1}} - \tau_{0_k}^i\right) + \mathcal{N}\left(\mathbf{0}, \mathbf{C}_{s_{k-1}}\right)\right| \end{aligned} \tag{14}$$

A representation of the importance density proposed for multipath delay generation can be seen in figure 2.

3. **Weight Update**. Since the Importance density has been chosen to fit the prior and the weights will not be propagated, weights update is given by

$$\tilde{w}_k^i = p(\mathbf{y}|\boldsymbol{\tau}_k^i, \mathbf{a}_k^i) \tag{15}$$

where $p(\mathbf{y}|\boldsymbol{\tau}_k^i, \mathbf{a}_k^i)$ is the likelihood function of measurements $\mathbf{y}$ given the set of support points $\left\{\boldsymbol{\tau}_k^i, \mathbf{a}_k^i\right\}_{i=1}^{N_s}$. After normalization, weights and support points can be used to characterize the posterior density,

$$w_k^i = \frac{\tilde{w}_k^i}{\sum_{j=1}^{N_s} \tilde{w}_k^j} \tag{16}$$

Pseudo-code describing Bayesian DLL algorithm for code timing estimation

$[\hat{\mathbf{x}},\ \hat{\mathbf{a}}]$= Bayesian_DLL($\mathbf{y}, \mathbf{C}_m, \mathbf{C}_{s_0}, N_s, N_t$)

- Initialize: $\left\{ \hat{\boldsymbol{\tau}}^i_{k=0} \right\}_{i=1}^{N_s}$ using eq. (12)
- FOR k=1:N_t
  - FOR i=1:N_s
    - Generate $\boldsymbol{\tau}^i_k \sim \boldsymbol{\pi}(\boldsymbol{\tau}_k | \hat{\boldsymbol{\tau}}^i_{k-1}, \mathbf{C}_{s_{k-1}})$
    - Estimate $\mathbf{a}^i_k$ according to (11) with $\boldsymbol{\tau}^i_k$
    - Calculate $\tilde{w}^i_k = p(\mathbf{y} | \boldsymbol{\tau}^i_k, \mathbf{a}^i_k, \mathbf{C}_m)$
  - END FOR
  - Normalize $w^i_k = \tilde{w}^i_k \left(\sum_{j=1}^{N_s} \tilde{w}^j_k \right)^{-1}$
  - $\left[\hat{\boldsymbol{\tau}}^{\text{MAP}}_k, \hat{\mathbf{a}}^{\text{MAP}}_k \right] = \arg\max p_k(\boldsymbol{\tau}^i_k, \mathbf{a}^i_k | \mathbf{y})$
  - Calculate $\mathbf{C}_{s_k}$ using eq. (18)
  - Initialize next iteration: $\left\{ \hat{\boldsymbol{\tau}}^i_k = \hat{\boldsymbol{\tau}}^{\text{MAP}}_k \right\}_{i=1}^{N_s}$
- END FOR
- Final estimates: $\hat{\boldsymbol{\tau}} = \hat{\boldsymbol{\tau}}^{\text{MAP}}_{N_t}$, $\hat{\mathbf{a}} = \hat{\mathbf{a}}^{\text{MAP}}_{N_t}$

4. **Estimation**. Now we consider the *Maximum a Posteriori* (MAP) estimation of delays and amplitudes. Substituting generated particles $\left\{ \boldsymbol{\tau}^i_k, \mathbf{a}^i_k, w^i_k \right\}_{i=1}^{N_s}$ into equation (9) yield an approximation of the posterior. Applying the MAP estimator:

$$\hat{\mathbf{x}}^{MAP}_k = \left[\hat{\boldsymbol{\tau}}^{\text{MAP}}_k, \hat{\mathbf{a}}^{\text{MAP}}_k \right] = \arg \max_{\boldsymbol{\tau}_k, \mathbf{a}_k} p_k(\boldsymbol{\tau}^i_k, \mathbf{a}^i_k | \mathbf{y}) \tag{17}$$

In addition to this estimation, the covariances of state error estimation must be calculated. A characterization is obtained from the uncertainty region of the posterior,

$$\mathbf{C}_{s_k} \approx \sum_{i=1}^{N_s} w^i_k \left(\mathbf{x}^i_k - \hat{\mathbf{x}}^{MAP}_k \right) \left(\mathbf{x}^i_k - \hat{\mathbf{x}}^{MAP}_k \right)^H \tag{18}$$

where $(\cdot)^H$ stands for the Hermitian operator.

5. **Resampling**. All particles are initialized to $\hat{\mathbf{x}}^{MAP}_k$ at $k+1$. This avoids the resampling step, which is shown to be the bottle neck in particle filter parallelization.

4. SIMULATIONS

We have studied the BPSK modulation of C/A code in GPS system. A scenario composed of a LOSS and a multipath replica is considered ($M = 2$), assuming that the channel parameters are constant during the observation interval, which is N_t ms, since the correlation outputs are taken every code period. We choose a carrier-to-noise density ratio (C/N_0) of 45 dB-Hz for the LOSS, a signal-to-multipath ratio (SMR) of 6 dB and the LOSS and multipath to be in–phase, the worst possible case. Figure 3 shows the Root Mean Square Error (RMSE) of time estimation for the LOSS with respect to the relative delay of the multipath replica, with a chip rate normalization. Several values of N_s are considered and the theoretical Cramér Rao Lower Bound is plotted.

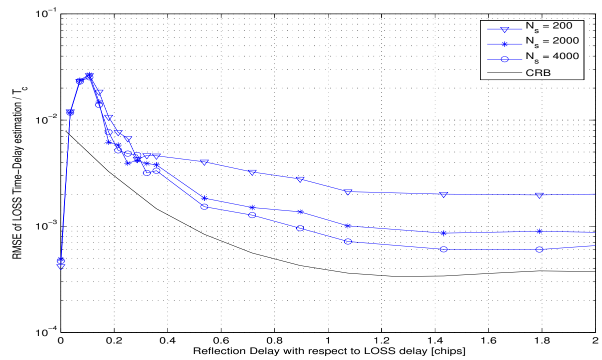

Fig. 3. RMSE of time delay of the LOSS. $f_s = 5.714$ MHz.

5. CONCLUSIONS

A new code–phase tracking algorithm has been presented under the SMC framework, that allows the characterization of the posteriori pdf of delays given a set of measurements. The algorithm overcomes the problem of parameter estimation in particle filters by properly choosing the importance density function. Simulations have been done with BPSK modulations used in navigation systems. The synchronization algorithm proposed obtains accurate estimates of delays under severe multipath conditions in few iterations and, although computationally intensive, it allows high level of parallelization since resampling step is avoided. Notice that, in the absence of multipath ($\boldsymbol{\tau}_{mp} = \mathbf{0}$), the algorithm is operating properly.

6. REFERENCES

[1] B.W. Parkinson and J.J. Spilker, Eds., *Global Positioning System: Theory and Applications, vol. I, II*, Progress in Astronautics and Aeronautics. American Institute of Aeronautics, Inc., Washington DC, 1996.

[2] M. Irsigler and B. Eissfeller, "Comparison of multipath mitigation techniques with consideration of future signal structures," in *Proceedings of the ION GPS/GNSS 2003*, Portland, OR, September 2003.

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

[4] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering for practitioners," *IEEE Trans. Signal Processing*, vol. 50, no. 3, pp. 736–746, March 2002.