Signal Processing 104 (2014) 212–224

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Stochastic filtering of a random Fibonacci sequence: Theory and applications

A. Farina [a], C. Fantacci [b,*], M. Frasca [c,**]

[a] Selex ES, Via Tiburtina km. 12,400, 00131 Rome, Italy
[b] Dipartimento di Ingegneria dell'Informazione, Università di Firenze, Via di Santa Marta 3, 50139 Florence, Italy
[c] MBDA Italia S.p.A., Seeker Division, Via Carciano 4-50, 60-70, 00131 Rome, Italy

ARTICLE INFO

Article history:
Received 3 August 2013
Received in revised form
2 January 2014
Accepted 31 March 2014
Available online 18 April 2014

Keywords:
Random Fibonacci sequence
Viswanath constant
Stochastic filtering
Kalman Filter
Random process
Malware diffusion in computer network

ABSTRACT

This paper conceives a stochastic filtering problem to estimate, from noisy measurements, the numbers of the random Fibonacci sequence. The dynamical system is amenable to an exact solution, being the convolution of Bernoullian and Gaussian variables, yielding a closed form for the equations of the filter. The derived optimal filter has exponential computational complexity, thus a suboptimal filter with affordable computational load is conceived. The stochastic filter performance is then evaluated with two applications: one of theoretical value and one for a more practical application. More precisely the first case study estimates the Viswanath constant from noisy measurements of the random Fibonacci sequence. It is shown how the filter performs well in estimating the Viswanath constant even if the noise is significantly increased. The second case study refers to a model of malware propagation in a computer network. In this case, it is assumed that there is a random rate for the infection, assuming that a finite time is needed before a computer is infected. A random generalized Fibonacci sequence fits well in this case. Additional applications are possible in view of the fact that several systems both in biology and economy are well represented by Fibonacci binary random trees.

1. Introduction

The Fibonacci numbers (which will be precisely defined in Section 2.2) were introduced in 1202 by Leonardo of Pisa [1] and their properties have been studied for centuries. Such numbers were used by biologists to describe patterns in nature such as branching in trees, arrangement of leaves on a stem, seashells and so forth; by computer scientists for efficient searching algorithms [2], heap data structures [3] and graphs [4,5]; by Matiyasevich, in an essential role, to solve the puzzling *Hilbert's tenth problem* [6,7]. In estimation theory, in [8–10], connections between the Kalman Filter, the Fibonacci sequence and the golden ratio have been studied and developed. The Fibonacci numbers were also studied in control theory by modeling the sequence of such numbers as a dynamical system and applying a Linear-Quadratic Regulator (LQR) [8] and a deadbeat regulator [11].

One of the most important properties of the Fibonacci numbers is that the ratio of successive numbers forms a new series that converges to the irrational value 1.61803398875..., that is the well known *golden ratio* φ [12,13]. Such irrational number has a lot of interesting properties, one of which is that powers of φ provide a close approximation to the Fibonacci number. This property has led Jacques P. M. Binet to formulate an exact solution for the Fibonacci numbers given the sole golden

* Principal corresponding author.
** Corresponding author.
E-mail addresses: alfonso.farina@selex-es.com (A. Farina), claudio.fantacci@unifi.it (C. Fantacci), marco.frasca@mbda.it (M. Frasca).

http://dx.doi.org/10.1016/j.sigpro.2014.03.052
0165-1684/

ratio, i.e.

$$f_k = \frac{\varphi^k - (-\varphi)^{-k}}{\sqrt{5}} \tag{1}$$

where f_k is the k-th Fibonacci number. As k goes to infinity, the growth rate of the Fibonacci sequence approaches the power of the golden ratio φ, i.e.

$$f_k \approx \frac{\varphi^k}{\sqrt{5}}, \text{ as } k \to \infty \tag{2}$$

Despite no one would expect much novelty on the Fibonacci sequence, during the last decade a whole new branch of such numbers has spread out when Divakar Viswanath proposed a new variation of the series [14], called *random Fibonacci*. Instead of adding the last two terms to produce a new one, you either add or subtract them depending on a random process described by a coin tossing [14,15]. Recalling the growth rate (2) for the Fibonacci sequence, an important result provided in [14] is the determination of the growth rate c of such a random sequence, called after his discovery as the *Viswanath constant*. In 1960, Hillel Furstenberg and Harry Kesten showed in [16] that for a general class of random matrices the norm of the product of n factors converges, almost surely, to the n-th power of a constant value. As a consequence, the n-th root of such n-th power converges almost surely to the aforementioned constant value. Random Fibonacci belongs to this broad class of product of random matrices as it has been demonstrated by Viswanath in [14] discovering an explicit expression for c. The Viswanath constant on a random Fibonacci sequence is given by [14]

$$c \triangleq e^{\gamma_f} = \lim_{k \to \infty} |f_k|^{1/k} = 1.13198824\ldots \tag{3}$$

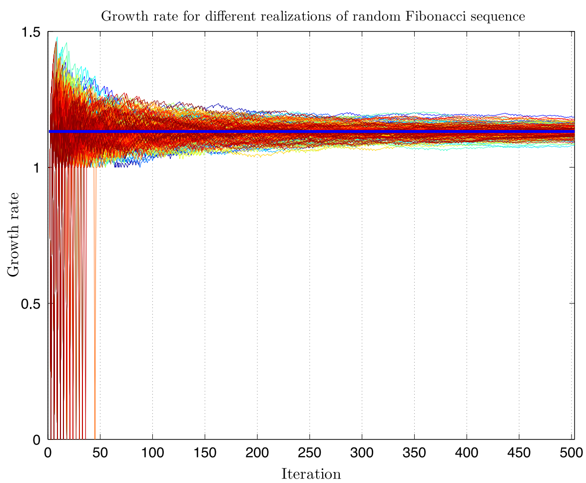

Fig. 1. Pictorial view of the growth rate of random Fibonacci numbers for different realizations of the random sequence. The blue solid line represents the Viswanath constant c. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

Fig. 1 displays the growth rate of different realizations of the random Fibonacci sequence.

Notice that in Fig. 1 the growth rate of each realization tends toward the Viswanath constant c (depicted as a blue solid line).

Real-life applications exploiting random Fibonacci as random generator are the distribution of memory for parallel computers [17] and the online gambling [18].

From a mathematical point of view, the random Fibonacci can be seen as a *random walk* [19], in particular on a graph. A random walk is a succession of random steps generating a path, e.g. the path traced by a molecule as it travels in a liquid or a gas or the financial status of a gambler. A random walk on a graph is a very special case of a Markov chain and has been widely used in economics, physics, electrical networks, Brownian motion mathematics, Laplace's equation, sampling problems and so forth. There is a direct correspondence between paths traced out through the branches of the tree by a random walk and the random Fibonacci. Any such a path can be written as a sequence of *left* and *right* commands, giving directions for how to get from the root of the tree to a specific interior node. Such correspondence has been used by Viswanath in [14], exploiting the Stern–Brocot tree [20,21], to calculate the value of the Viswanath constant, resulting in $c = 1.13198824\ldots$.

Motivated by the great interest on the growth rate of random processes, as well as estimating the state of a random walk on a graph, the authors propose the Bayesian derivation of a recursive filter designed to sequentially *predict* and *estimate* the state of a stochastic dynamical system [22] in which the transition matrix will switch depending on a coin tossing process, i.e. the random Fibonacci, given noisy measurements and the probabilities associated to the coin tossing. It will then be shown how the proposed filter is also capable of estimating the growth rate of the stochastic dynamical system. In particular, considering the random Fibonacci sequence, the resulting growth rate is the Viswanath constant c. It will be presented a brand new application of random Fibonacci filtering to a cyber-security study case. In particular it will be tackled the problem of estimating the infection of a malware in a network of computers.

Further related work on filters for switching discrete time systems can be found in [23] which might shed further light on the stochastic filtering of the random Fibonacci sequence. This might be a future research track to exploit possible connections between the filter developed in this paper and the theory presented in the above-mentioned reference.

The paper is organized as follows. Section 2 introduces the Fibonacci sequence and the Bayesian derivation of a filtering problem. Section 3 introduces the random Fibonacci sequence and tackles the stochastic filtering problem for a randomly switching dynamical system. Section 4 proposes a suboptimal solution of the stochastic filtering problem described in the previous section to cope with the exponential complexity of the filtering algorithm. Section 5 presents simulation results supporting the proposed approach for recursively estimating the state of a randomly switching dynamical system. Section 6 presents simulation results for a specific application scenario in which the aim is to estimate the infection of a malware with random propagation behavior on a network of computers. Finally Section 7 ends the paper with concluding remarks and perspectives for future work and possible applications.

2. Background and problem formulation

2.1. Dynamical state and measurement equations

Consider a nonlinear dynamical state equation with forcing noises v_k

$$x_{k+1} = f_k(x_k, v_k) \tag{4}$$

A particular case of (4) is where noise is modeled as an additive white variable with Gaussian *Probability Density Function* (PDF)

$$x_{k+1} = f_k(x_k) + v_k \tag{5}$$

Moreover, by considering $f_k(\cdot)$ in (5) linear and without forcing noise (i.e. $v_k = 0$, $\forall k$), it results in

$$x_{k+1} = A_k x_k \tag{6}$$

where A_k is an $n_x \times n_x$ matrix, $n_x = \dim(x)$.

The nonlinear measurement equation is

$$z_k = h_k(x_k) + w_k \tag{7}$$

where w_k is the measurement noise, modeled as white process with Gaussian PDF. As a particular case of (7), $h_k(\cdot)$ can be linear, i.e.

$$z_k = C_k x_k + w_k \tag{8}$$

where C_k is an $n_z \times n_x$ matrix, $n_z = \dim(z)$. From the measurement equation (7), the likelihood function of x_k w.r.t. z_k is defined as

$$g_k(z_k | x_k) = p_w(z_k - h_k(x_k)) \tag{9}$$

2.2. Fibonacci series and related stochastic filtering

Consider the case in which we have the classical Fibonacci series

$$\begin{cases} f_{k+2} = f_{k+1} + f_k, \quad n \geq 2 \\ f_0 = 0 \\ f_1 = 1 \end{cases} \tag{10}$$

where the k-th term is the sum of the previous two terms. Furthermore, consider the related stochastic filtering problem associated to the stationary ($A_k = A$ and $C_k = C$, $\forall k$) Fibonacci sequence (10), i.e. rewritten in terms similar to (6) and (8)

$$\begin{cases} x_{k+1} = A_+ x_k + v_k \\ z_k = C x_k + w_k \end{cases} \tag{11}$$

where $x_k = [f_k, f_{k-1}]^T$ are, respectively, the $k+1$-th and the k-th Fibonacci numbers:

$$A_+ = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad C = [1 \ 0], \tag{12}$$

v_k and w_k are, respectively, the process noise and measurement noise with covariances, respectively, Q and R. Note that Q is a matrix while R is a scalar. A pictorial view of the Fibonacci series (10) written as a dynamical system (cfr. (11)) is depicted in Fig. 2.

Considering the first of the three equation of system (10), i.e. $f_{k+1} = f_k + f_{k-1}$, it is possible to derive the state evolution equation exploiting the Z-transform $F(z) \triangleq \sum_{k=1}^{\infty} f_k z^{-k}$ as follows:

$$zF(z) - zf_0 = F(z) + z^{-1}F(z) \tag{13}$$

$$zF(z) - f_1 = F(z) + z^{-1}F(z) \tag{14}$$

$$(z - 1 - z^{-1})F(z) = f_1 \tag{15}$$

$$\frac{z^2 - z - 1}{z} F(z) = f_1 \tag{16}$$

$$F(z) = \frac{z}{z^2 - z - 1} f_1 \tag{17}$$

where z^i represents a discrete time shift for $i > 0$ and a discrete time reversal for $i < 0$.

The state evolution equation is

$$\begin{cases} F(z) = h(z) f_1 \\ h(z) = \dfrac{z}{z^2 - z - 1} = \dfrac{1}{\sqrt{5}} \left(\dfrac{z}{z - \varphi} - \dfrac{z}{z + \frac{1}{\varphi}} \right) \\ \varphi = \dfrac{1 + \sqrt{5}}{2} = 1.61803399\ldots \\ \dfrac{1}{\varphi} = \dfrac{-1 + \sqrt{5}}{2} = \varphi - 1 = 0.618033989\ldots \end{cases} \tag{18}$$

where φ is the well known *golden ratio* [8]. Z inverse-transforming $h(z)$ in (18) provides

$$h_k = \frac{1}{\sqrt{5}} \left[\varphi^k - \left(-\frac{1}{\varphi} \right)^k \right] \approx \frac{1}{\sqrt{5}} \varphi^k \tag{19}$$

which is the Binet's formula. Note that the growth rate of the Fibonacci sequence is φ.

Consider now another sequence where the Fibonacci sequence (10) is replaced by the subtraction of the k-th

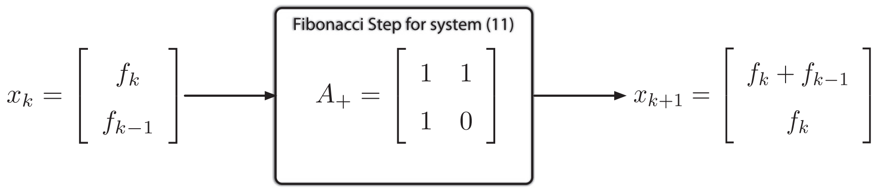

Fig. 2. Pictorial view of the Fibonacci sequence generation (10).

and the $k+1$-th numbers, i.e.

$$\begin{cases} f_{k+2} = -f_{k+1} + f_k, & k \geq 2 \\ f_0 = 0 \\ f_1 = 1 \end{cases} \tag{20}$$

Rewriting the recurrence (20) in state space form yields

$$\begin{cases} x_{k+1} = A_- x_k + v_k \\ z_k = C x_k + w_k \end{cases} \tag{21}$$

where

$$A_- = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{22}$$

that is a stationary stochastic dynamical system.

Considering the first of the three equations of system (20), i.e. $f_{k+1} = -f_k + f_{k-1}$, it is possible to derive, as it has been done for system (10), the state evolution equation using the Z-transform $F(z) \triangleq \sum_{k=1}^{\infty} f_k z^{-k}$. Thus, the state evolution equation is

$$\begin{cases} F(z) = h(z) f_1 \\ h(z) = \dfrac{z}{z^2 + z - 1} = \dfrac{1}{\sqrt{5}} \left(\dfrac{z}{z - \frac{1}{\varphi}} - \dfrac{z}{z + \varphi} \right) \end{cases} \tag{23}$$

Z inverse-transforming $h(z)$ in (23) provides

$$h_k = \frac{1}{\sqrt{5}} \left[\left(-\frac{1}{\varphi} \right)^k - \varphi^k \right] \approx -\frac{1}{\sqrt{5}} \varphi^k \tag{24}$$

which is the Binet's-like formula for the system (20). Note that, even with a change of the sign for f_{k+1} in (20), the growth rate of the sequence is still φ.

2.3. Bayesian derivation of the optimal stochastic filter with application to Fibonacci sequences

The problem of sequentially estimating the state of a dynamic system from a sequence of noisy measurements made on the system can be tackled considering (1) the *transition probability equation* associated to the dynamic system, (2) the *Chapman–Kolmogorov equation* and (3) the *Bayes rule*. This classical approach can be adopted in both *linear* and *nonlinear* systems.

The transition probability associated to the Fibonacci dynamical system (11) is

$$\pi_{k|k-1}(x_k | x_{k-1}) \tag{25}$$

The Chapman–Kolmogorov equation for the PDF of the prediction of state x at time $k+1$, for the dynamical state equations (6), when measurement z_k is available, see (7), is

$$p(x_{k+1}|z_{1:k}) = \int \pi_{k+1|k}(x_{k+1}|x_k) \cdot p(x_k|z_{1:k}) \, dx_k \tag{26}$$

where $z_{1:k} = \{z_1, z_2, \ldots, z_k\}$ is the set of measurements up to time k.

The Bayes rule for the PDF update given measurement z_k

$$p(x_k|z_{1:k}) = \frac{g_k(z_k|x_k) p(x_k|z_{1:k-1})}{\int g_k(z_k|x_k) p(x_k|z_{1:k-1}) \, dx_k} \tag{27}$$

Assume now for system (11) linear and Gaussian hypotheses as follows: (1) the process is Markovian, i.e. transition probabilities depend only on the current state; (2) the prior distribution x_0, is Gaussian; (3) the transition probability (25) is Gaussian; and (4) the likelihood function (9) is Gaussian; that is

$$\pi_{k|k-1}(x_k|x_{k-1}) = \mathcal{N}(x_k; A_+ x_{k-1}, Q) \tag{28}$$

$$x_0 \sim \mathcal{N}(m_0, P_0) \tag{29}$$

$$g_k(z_k|x_k) = \mathcal{N}(z_k; C x_k, R) \tag{30}$$

where Q and R are, respectively, process and measurement covariance matrices. $\mathcal{N}(m, P)$ refers to a Gaussian random variable with mean m and covariance P, while $\mathcal{N}(x; m, P)$ is the relative PDF.

From (26) it follows:

$$\begin{aligned} p(x_{k+1}|z_{1:k}) &= \int \pi_{k+1|k}(x_{k+1}|x_k) \cdot p(x_k|z_{1:k}) \, dx_k \\ &= \int \mathcal{N}(x_{k+1}; A_+ x_k, Q) \cdot \mathcal{N}(x_k; m_k, P_k) \, dx_k \\ &= \{\text{See Lemmas 1 and 2, Appendices A and B}\} \\ &= \mathcal{N}(x_{k+1}; A_+ x_k, A_+ P_k A_+^T + Q) \end{aligned} \tag{31}$$

The same steps can be repeated for the PDF $p(x_k|z_{1:k})$ of the state x_k given the whole set of measurements $z_{1:k}$ (Eq. 27)

$$p(x_{k+1}|z_{1:k+1}) = \frac{g_k(z_{k+1}|x_{k+1}) p(x_{k+1}|z_{1:k})}{\int g_k(z_{k+1}|x_{k+1}) p(x_{k+1}|z_{1:k}) \, dx_{k+1}} \tag{32}$$

$$p(x_{k+1}|z_{1:k+1}) \propto g_k(z_{k+1}|x_{k+1}) p(x_{k+1}|z_{1:k}) \tag{33}$$

$$\begin{aligned} p(x_{k+1}|z_{1:k+1}) &\propto \mathcal{N}(z_{k+1}; C x_{k+1}, R) \\ &\cdot \mathcal{N}(x_{k+1}; A_+ x_k, A_+ P_k A_+^T + Q) \end{aligned} \tag{34}$$

Using now Lemma 1 in Appendix A one can find

$$\begin{aligned} p(x_{k+1}|z_{1:k+1}) = \mathcal{N}(&x_{k+1}; A_+ x_k + [A_+ P_k A_+^T + Q] \\ &\cdot [C(A_+ P_k A_+^T + Q) C^T + R]^{-1} (z_{k+1} - C x_{k+1}), \\ &(A_+ P_k A_+^T + Q) - [(A_+ P_k A_+^T + Q) C^T] \cdot \\ &\cdot [C(A_+ P_k A_+^T + Q) C^T + R]^{-1} [C(A_+ P_k A_+^T + Q)]) \end{aligned} \tag{35}$$

The recursive state estimator, i.e. the classical Kalman Filter, from the Bayesian recursions (31) and (35), is summarized in Table 1.

The procedure described above to obtain the Kalman Filter, starting from (26) and (27) for system (11), can also be done for system (21), but it will not be shown here for the sake of brevity.

In the next section a particular system, namely the *random Fibonacci* system, will be formulated starting from systems (10) and (20).

Table 1
Kalman Filter for the filtering problem (11).

1. <u>PREDICTION</u>
$\hat{x}_{k+1\|k} = A_+ \hat{x}_{k\|k}$
$P_{k+1\|k} = A_+ P_{k\|k} A_+^T + Q$
2. <u>CORRECTION</u>
$S_{k+1} = C P_{k+1\|k} C^T + R$
$L_{k+1} = P_{k+1\|k} C^T S_{k+1}^{-1}$
$e_{k+1} = z_{k+1} - C \hat{x}_{k+1\|k}$
$\hat{x}_{k+1\|k+1} = \hat{x}_{k+1\|k} + L_{k+1} e_{k+1}$
$P_{k+1\|k+1} = P_{k+1\|k} - L_{k+1} S_{k+1} L_{k+1}^T$

3. Stochastic filtering problem for the random Fibonacci sequence

Define a random switching stochastic system in which the Fibonacci recurrence (10) is modified as follows:

$$\begin{cases} f_{k+2} = \pm f_{k+1} + f_k, & k \geq 2 \\ f_0 = 0 \\ f_1 = 1 \end{cases} \tag{36}$$

The system (36) can be modeled as a combination of the two stationary dynamical systems (10) and (20) in which the matrices (12) and (22) are switched, at each time instant k, with a fixed probability. In this work the authors will consider a *coin tossing* process for the randomness of system (36), i.e. a parameter, namely p, will describe the probability of having one of the two sides of the coin. This process is completely and exhaustively described by the *Bernoulli process*. Therefore, it is worth pointing out that the recurrence is now random and non-stationary. This process will be referred to as the *random Fibonacci* system [14].

Fig. 3 displays the spread of the numbers obtained by different realizations of the random sequence.

It is worth pointing out that the numbers are “compressed” using

$$\mathrm{sign}(f_k) \cdot \log_{10}(|f_k|) \tag{37}$$

to cope with the high order of magnitude. Note that Fig. 3 presents a bifurcation. Each number is a realization of the random variable F that is obtained from the summation of a Bernoulli B and a Gaussian N random variable, i.e. $F = B + N$. Since B and N have known, respectively, *Probability Mass Function* (PMF) and PDF, i.e.

$$B \sim \mathcal{B}(p) = \begin{cases} p, & \text{if } x = x_1 \\ 1 - p \triangleq q, & \text{if } x = x_2 \end{cases} = p \cdot \delta(x - x_1) + q \cdot \delta(x - x_2)$$

$$x_1, x_2 \in \mathbb{R}^{n_x}$$

$$p \in [0, 1]$$

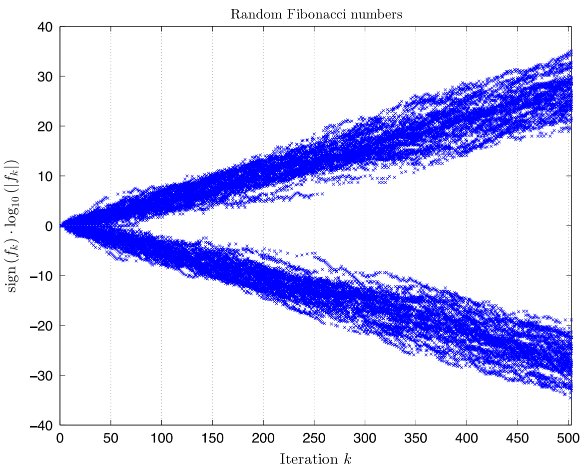

Fig. 3. Pictorial view of the random Fibonacci numbers for different realization of the random sequence.

and

$$N \sim \mathcal{N}(m, P) = \frac{1}{\sqrt{\det(2\pi P)}} e^{-\frac{1}{2}(x-m)^T P^{-1}(x-m)}, \tag{38}$$

and are mutually independent, it follows that the PDF $\mathcal{F}$ of F is the convolution of the Bernoulli and Gaussian probability functions, i.e.

$$\begin{aligned} \mathcal{F} &= \mathcal{B} \otimes \mathcal{N} \\ &= [p \cdot \delta(x - x_1) + q \cdot \delta(x - x_2)] \\ &\quad \otimes \left[\frac{1}{\sqrt{\det(2\pi P)}} e^{-\frac{1}{2}(x-m)^t P^{-1}(x-m)} \right] \\ &= p \cdot \frac{1}{\sqrt{\det(2\pi P)}} e^{-\frac{1}{2}(x - x_1 - m)^t P^{-1}(x-m)} \\ &\quad + q \cdot \frac{1}{\sqrt{\det(2\pi P)}} e^{-\frac{1}{2}(x - x_2 - m)^t P^{-1}(x - x_2 - m)} \\ &= p \cdot \mathcal{N}(x; m - x_1, P) + q \cdot \mathcal{N}(x; m - x_2, P) \end{aligned} \tag{39}$$

Eq. (39) justifies the aforementioned bifurcation phenomenon.

It is worth noticing from Fig. 3 that the absolute value of the random Fibonacci numbers, i.e. $|f_k|$, grows to high order of magnitude. At a first glance, it would be natural to think that the effect of switching between systems (11) and (21) could lead to an almost zero growth of $|f_k|$. Conversely, since (11) and (21) are two unstable systems with the same equilibrium point $[0\ 0]^T$ and the initial state differs from it, it is reasonable to think that also (36) has a similar unstable behavior. Consequently, it could be expected that the growth rate of the random Fibonacci numbers is slower than the one of system (11) and system (21) because of the switch. In fact, what it has been speculated is confirmed by the Viswanath constant c (representing the growth rate of the random Fibonacci sequence (3)) since its value is lower than the golden ratio φ (representing the growth rate of the systems (11) and (21)).

3.1. Stochastic filtering problem of random Fibonacci system

A new stochastic filtering problem is conceived where the system is described by two equations which are selected randomly using a *coin tossing* procedure with probability p.

Definition (*Random Fibonacci system*). The Random Fibonacci system is defined as follows:

$$\begin{cases} x_{k+1} = A_+ x_k + v_k, & \text{with probability } p \\ x_{k+1} = A_- x_k + v_k, & \text{with probability } 1 - p \\ z_k = C x_k + w_k \end{cases} \tag{40}$$

having $p \in [0, 1]$.

A pictorial view of system (40) is in Fig. 4, where z_k is the noisy measurement of the random Fibonacci sequence.

The aim of this paper is to have a recursion like (31) and (35) to estimate the random series described by system (40). To tackle this problem let us now consider (26) and (27) and assume, for system (40), that

1. the process is not Markov because it depends on a Bernoulli random variable $s \sim \mathcal{B}(p)$;

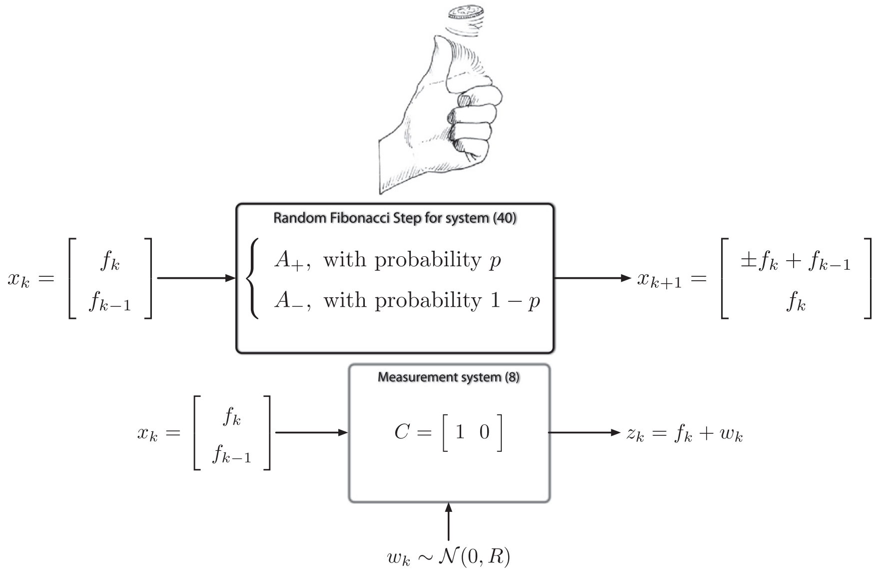

Fig. 4. Pictorial view of the random Fibonacci dynamic system (40). A_+ and A_- occur in a long time in repeated tosses of a biased coin (when $p \neq 0.5$).

2. the prior distribution x_0 is Gaussian;
3. the transition probability $\pi_{k|k-1}(x_k|x_{k-1})$ is a Gaussian mixture with weights p and $q=1-p$, as it has been described in (39); and
4. the likelihood function $g_k(z_k|x_k)$ is Gaussian.

That is

$$\pi_{k|k-1}(x_k|x_{1:k-1}) = p\cdot\mathcal{N}(x_k; A_+x_{k-1}, Q) + q\cdot\mathcal{N}(x_k; A_-x_{k-1}, Q) \tag{41}$$

$$x_0 \sim \mathcal{N}(m_0, P_0) \tag{42}$$

$$g_k(z_k|x_k) = \mathcal{N}(z_k; Cx_k, R) \tag{43}$$

It follows from (26)

$$\begin{aligned} p(x_{k+1}|z_{1:k}) &= \int \pi_{k+1|k}(x_{k+1}|x_k)\cdot p(x_k|z_{1:k})\,dx_k \\ &= \int [p\cdot\mathcal{N}(x_{k+1}; A_+x_k, Q) + q\cdot\mathcal{N}(x_{k+1}; A_-x_k, Q)] \\ &\quad \cdot \alpha_k\mathcal{N}(x_k; m_k, P_k)\,dx_k \\ &= \int \alpha_k p\cdot\mathcal{N}(x_{k+1}; A_+x_k, Q)\cdot\mathcal{N}(x_k; m_k, P_k)\,dx_k \\ &\quad + \int \alpha_k q\cdot\mathcal{N}(x_{k+1}; A_-x_k, Q)\cdot\mathcal{N}(x_k; m_k, P_k)\,dx_k \\ &= \{\text{See Lemma 2, Appendix B and (31)}\} \\ &= \alpha_k p\cdot\mathcal{N}(x_{k+1}; A_+x_k, A_+P_kA_+^T + Q) \\ &\quad + \alpha_k q\cdot\mathcal{N}(x_{k+1}; A_-x_k, A_-P_kA_-^T + Q) \end{aligned} \tag{44}$$

The same steps of (35) are then repeated, i.e.

$$p(x_{k+1}|z_{1:k+1}) = \frac{g_k(z_{k+1}|x_{k+1})p(x_{k+1}|z_{1:k})}{\int g_k(z_{k+1}|x_{k+1})p(x_{k+1}|z_{1:k})\,dx_k} \tag{45}$$

$$p(x_{k+1}|z_{1:k+1}) \propto g_k(z_{k+1}|x_{k+1})p(x_{k+1}|z_{1:k}) \tag{46}$$

$$\begin{aligned} p(x_{k+1}|z_{1:k+1}) &\propto \mathcal{N}(z_{k+1}; Cx_{k+1}, R) \\ &\quad \cdot[\alpha_k p\cdot\mathcal{N}(x_{k+1}; A_+x_k, A_+P_kA_+^T + Q) \\ &\quad + \alpha_k q\cdot\mathcal{N}(x_{k+1}; A_-x_k, A_-P_kA_-^T + Q)] \end{aligned} \tag{47}$$

Using now Lemma 1 in Appendix A one can find

$$\begin{aligned} p(x_{k+1}|z_{1:k+1}) &= \frac{\alpha_{+;k+1}}{\kappa}\cdot\mathcal{N}(x_{k+1}; A_+x_k + [A_+P_kA_+^T + Q] \\ &\quad \cdot[C(A_+P_kA_+^T + Q)C^T + R]^{-1}(z_{k+1} - Cx_{k+1}), \\ &\quad (A_+P_kA_+^T + Q) - [(A_+P_kA_+^T + Q)C^T] \\ &\quad \cdot[C(A_+P_kA_+^T + Q)C^T + R]^{-1}[C(A_+P_kA_+^T + Q)]) \\ &\quad + \frac{\alpha_{-;k+1}}{\kappa}\cdot\mathcal{N}(x_{k+1}; A_-x_k + [A_-P_kA_-^T + Q] \\ &\quad \cdot[C(A_-P_kA_-^T + Q)C^T + R]^{-1}(z_{k+1} - Cx_{k+1}), \\ &\quad (A_-P_kA_-^T + Q) - [(A_-P_kA_-^T + Q)C^T] \\ &\quad \cdot[C(A_-P_kA_-^T + Q)C^T + R]^{-1}[C(A_-P_kA_-^T + Q)]) \end{aligned} \tag{48}$$

where the weighting parameters $\alpha_{+;k+1}$ and $\alpha_{-;k+1}$ are updated as follows:

$$\alpha_{+;k+1} = \alpha_k p\cdot\mathcal{N}(z_{k+1}; A_+m, C(A_+P_kA_+^T + Q)C^T + R) \tag{49}$$

$$\alpha_{-;k+1} = \alpha_k q\cdot\mathcal{N}(z_{k+1}; A_-m, C(A_-P_kA_-^T + Q)C^T + R) \tag{50}$$

and the normalization constant c is

$$\kappa = \int g_k(z_{k+1}|x_{k+1})p(x_{k+1}|z_{1:k})dx_k = \alpha_{+;k+1} + \alpha_{-;k+1} \tag{51}$$

Result (*Stochastic filtering for the random Fibonacci dynamic problem*). Eqs. (44)–(51) are the closed form solution for the stochastic filtering problem associated to the random Fibonacci system.

It is worth pointing out that in this scenario the weighting parameters (49) and (50) are updated exploiting Lemma 1 in Appendix A using the measurement z_{k+1}. It is straightforward that, at each step k, the most probable hypothesis has higher weight than the others.

In (35) the weighting and normalization constants are omitted because, having just one hypothesis, such constants will always simplify to 1. For example, taking into account (48), and by considering $p=1$ and $q=1-p=0$, one has $\alpha_{-;k+1}=0$, hence $c=\alpha_{+;k+1}$ and

$$
\begin{aligned}
p(x_{k+1}|z_{1:k+1}) = {} & \frac{\alpha_{+;k+1}}{\alpha_{+;k+1}} \cdot \mathcal{N}(x_{k+1}; A_+ x_k + [A_+ P_k A_+^T + Q] \\
& \cdot [C(A_+ P_k A_+^T + Q)C^T + R]^{-1}(z_{k+1} - Cx_{k+1}), \\
& (A_+ P_k A_+^T + Q) - [(A_+ P_k A_+^T + Q)C^T] \\
& \cdot [C(A_+ P_k A_+^T + Q)C^T + R]^{-1}[C(A_+ P_k A_+^T + Q)]) \\
& + 0 \cdot \mathcal{N}(x_{k+1}; A_- x_k + [A_- P_k A_-^T + Q] \\
& \cdot [C(A_- P_k A_-^T + Q)C^T + R]^{-1}(z_{k+1} - Cx_{k+1}), \\
& (A_- P_k A_-^T + Q) - [(A_- P_k A_-^T + Q)C^T] \\
& \cdot [C(A_- P_k A_-^T + Q)C^T + R]^{-1}[C(A_- P_k A_-^T + Q)])
\end{aligned} \tag{52}
$$

that simplifies to

$$
\begin{aligned}
p(x_{k+1}|z_{1:k+1}) = {} & \mathcal{N}(x_{k+1}; A_+ x_k + [A_+ P_k A_+^T] \\
& \cdot [C(A_+ P_k A_+^T)C^T + R]^{-1}(z_{k+1} - Cx_{k+1}), \\
& (A_+ P_k A_+^T + Q) - [(A_+ P_k A_+^T + Q)C^T] \\
& \cdot [C(A_+ P_k A_+^T + Q)C^T + R]^{-1}[C(A_+ P_k A_+^T + Q)])
\end{aligned} \tag{53}
$$

that turns out to be Eq. (35). The same steps could be repeated by considering $p=0$ and $q=1-p=1$.

The recursive optimal filter for the random Fibonacci dynamic system (40) described by (44)–(51) is displayed in Fig. 5 and described in pseudo-code in Table 2. Note that, at each recursion step, each hypothesis of $\hat{x}_{k|k}$ will generate two new hypotheses, i.e. $\hat{x}^i_{+;k+1|k+1}$ and $\hat{x}^i_{-;k+1|k+1}$. Therefore, having N_h initial hypotheses at time instant k, one has $2 \cdot N_h$ hypotheses after the *Correction* step.

This exponential growth of the number of hypotheses leads to a mathematically intractable approach. Thus, suboptimal solutions will be adopted as described in the next section.

4. Suboptimal solution for random Fibonacci stochastic filtering problem

As said, at each time step k, the *Prediction* and *Correction* steps generate two weighted Gaussian PDF, respectively with weights $\alpha_{+;k+1}\kappa^{-1}$ and $\alpha_{-;k+1}\kappa^{-1}$. This leads to an exponential growth of Gaussian components at each step k by 2^k. To tackle this problem, one could adopt a suboptimal solution considering, at each step, the *Most Probable Hypothesis* (MPH). The suboptimal algorithm MPH is portrayed in Fig. 6 and described in pseudo-code in Table 3.

The authors would like to refer to this algorithm with the nickname "4F" (i.e.: Fibonacci, Farina, Fantacci, Frasca).

5. Simulation case studies

In this section, simulations are provided for the random Fibonacci filtering problem exploiting the MPH suboptimal solution described in Table 3, for different values of p and Q.

5.1. Random Fibonacci numbers estimation – Case $Q=0$

The parameters of the simulations are as follows:

1. $p \in \{0, 0.2, 0.5, 0.8, 1\} \Rightarrow 1-p=q \in \{1, 0.8, 0.5, 0.2, 0\}$.
2. Process noise covariance matrix $Q=0$.

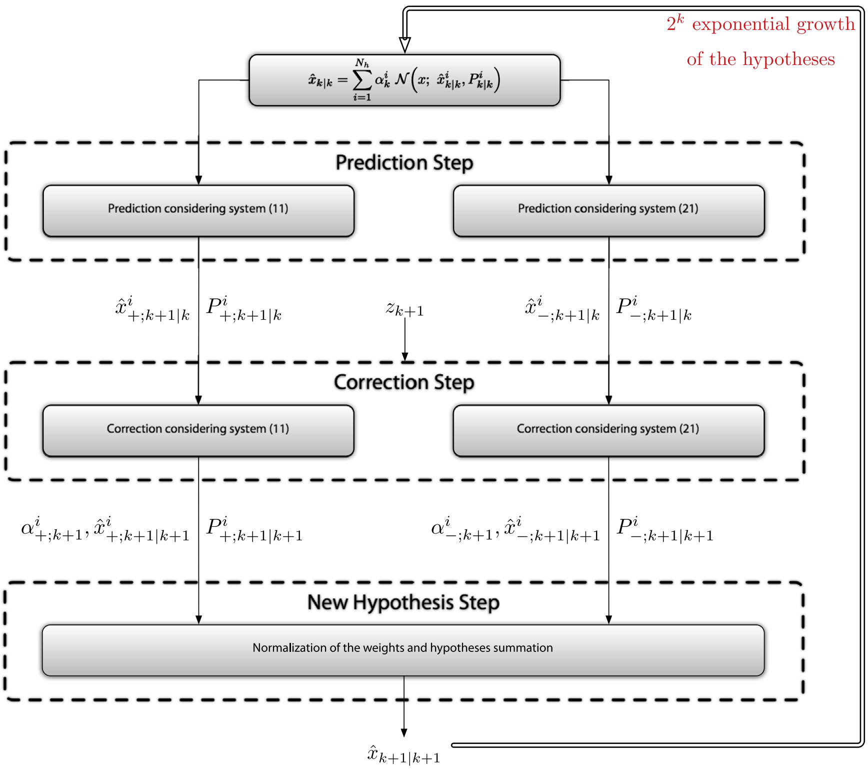

Fig. 5. Flow chart of the optimal stochastic filtering algorithm for the random Fibonacci dynamic problem.

Table 2
Stochastic optimal filtering algorithm for the random Fibonacci dynamic problem at time instant k.

INPUT: $\hat{x}_{k|k} = \sum_{i=1}^{N_h} \alpha_k^i \mathcal{N}(x; \hat{x}_{k|k}^i, P_{k|k}^i)$

1. PREDICTION

for all hypotheses $i = 1, \dots, N_h$ **do**

Prediction considering system (11)

$\hat{x}^i_{+;k+1|k} = A_+ \hat{x}^i_{k|k}$

$P^i_{+;k+1|k} = A_+ P^i_{k|k} A_+^T + Q$

Prediction considering system (21)

$\hat{x}^i_{-;k+1|k} = A_- \hat{x}^i_{k|k}$

$P^i_{-;k+1|k} = A_- P^i_{k|k} A_-^T + Q$

end for

2. CORRECTION

for all hypotheses $i = 1, \dots, N_h$ **do**

Correction considering system (11)

$S^i_{+;k+1} = CP^i_{+;k+1|k} C^T + R$

$L^i_{+;k+1} = P^i_{+;k+1|k} C^T (S^i_{+;k+1})^{-1} \qquad e^i_{+;k+1} = z_{k+1} - C\hat{x}^i_{+;k+1|k}$

$\hat{x}^i_{+;k+1|k+1} = \hat{x}^i_{+;k+1|k} + L^i_{+;k+1} e^i_{+;k+1}$

$P^i_{+;k+1|k+1} = P^i_{+;k+1|k} - L^i_{+;k+1} S^i_{+;k+1} (L^i_{+;k+1})^T$

$\alpha^i_{+;k+1} = \alpha^i_k p \cdot \mathcal{N}(z_{k+1}; \hat{x}^i_{+;k+1|k}, S^i_{+;k+1})$

Correction considering system (21)

$S^i_{-;k+1} = CP^i_{-;k+1|k} C^T + R$

$L^i_{-;k+1} = P^i_{-;k+1|k} C^T (S^i_{-;k+1})^{-1}$

$e^i_{-;k+1} = z_{k+1} - C\hat{x}^i_{-;k+1|k}$

$\hat{x}^i_{-;k+1|k+1} = \hat{x}^i_{-;k+1|k} + L^i_{-;k+1} e^i_{-;k+1}$

$P^i_{-;k+1|k+1} = P^i_{-;k+1|k} - L^i_{-;k+1} S^i_{-;k+1} (L^i_{-;k+1})^T$

$\alpha^i_{-;k+1} = \alpha^i_k q \cdot \mathcal{N}(z_{k+1}; \hat{x}^i_{-;k+1|k}, S^i_{-;k+1})$

end for

3. NEW HYPOTHESES

$$\kappa = \sum_{i=1}^{N_h} (\alpha^i_{+;k+1} + \alpha^i_{-;k+1})$$

$$\hat{x}_{k+1|k+1} = \sum_{i=1}^{N_h} \left[\frac{\alpha^i_{+;k+1}}{\kappa} \cdot \mathcal{N}\left(x_{k+1}; \hat{x}^i_{+;k+1|k+1}, P^i_{+;k+1|k+1}\right) + \frac{\alpha^i_{-;k+1}}{\kappa} \cdot \mathcal{N}\left(x_{k+1}; \hat{x}^i_{-;k+1|k+1}, P^i_{-;k+1|k+1}\right) \right]$$

OUTPUT: $\hat{x}_{k+1|k+1}$

3. Measurement noise covariance matrix $R = 1$.
4. Number of Fibonacci recursion steps $N_{fs} = 500$.
5. Number of independent Monte Carlo trials $N_{mc} = 300$.
6. The initial conditions, i.e. a priori estimated Fibonacci numbers, are chosen as follows:

$$\hat{x}_{0|-1} = \begin{bmatrix} \text{uniform random}(0, 1000) \\ \text{uniform random}(0, 1000) \end{bmatrix},$$

$$P_{0|-1} = \begin{bmatrix} \hat{x}_{0|-1}(1)^2 & 0 \\ 0 & \hat{x}_{0|-1}(2)^2 \end{bmatrix}.$$

The performance of MPH is measured by the *Root Mean Square Error* (RMSE) of the first component of the state, i.e. f_k. Averaging is performed over N_{mc} independent Monte Carlo trials for different Fibonacci recursions (except for $p=1$ and $p=0$) and different, independently generated, measurement noise realizations. Simulation results for $p \in \{0, 0.2, 0.5, 0.8, 1\}$ are depicted in Fig. 7, where the solid lines are the estimation errors averaged over all the independent Monte Carlo trials.

As it can be noticed from Fig. 7, the filtering algorithm works extremely well for all values of p. Results show that the proposed approach, adopting a fairly intuitive suboptimal algorithm, leads to acceptable estimation performance for each value of p. Table 4 shows the estimated covariance matrix $P_{500|500}$, i.e., for each value of p, the estimated covariance matrix of the last step averaged over N_{mc} independent Monte Carlo trials.

Note that the element $(1, 1)$ of $P_{500|500}$ for $p=0$ and $p=1$ is the inverse of the golden ratio, i.e. $1/\varphi \approx 0.6180340$ where $\varphi \approx 1.61803399$. This result has been also analytically found in [8], for the case $p=1$, i.e. the a posteriori state variance, for $p \in \{0, 1\}$, converges to the golden ratio φ.

5.2. Estimating the Viswanath constant, case Q=0, R=1 and R=100

The filter proposed in Table 3 is an estimator of the Viswanath constant when $p=0.5$. Indeed, we are estimating Fibonacci numbers. Fig. 8 displays the estimate $\hat{e}^{\gamma_f} = |\hat{f}_k|^{1/k}$ of (3), averaged over N_{mc} independent Monte Carlo trials,

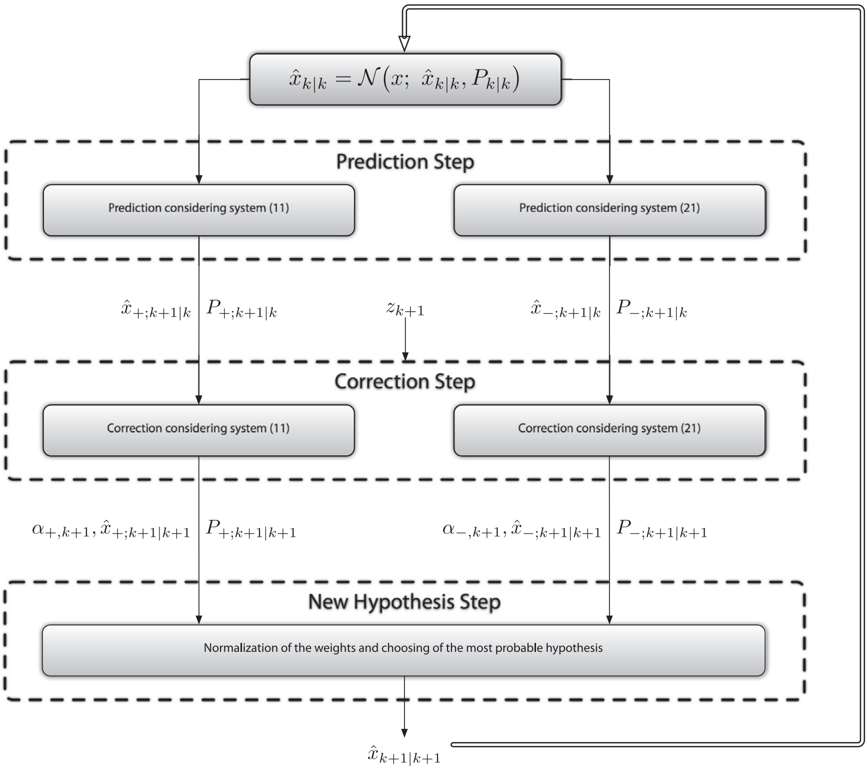

Fig. 6. Flow chart of the suboptimal stochastic filtering algorithm, i.e. MPH, for the random Fibonacci dynamic problem.

exploiting the estimates $\hat{f}_k$ of the random Fibonacci numbers, having the hypotheses (1)–(6) of Section 5.1. The green solid line in Fig. 8 gives a remarkable feeling of the filter performance and shows how fast it converges to the right value. The red solid line is the value of the Viswanath constant. It is thus really interesting to evaluate such an estimation when the noise variance R changes. Increasing the noise variance by two orders of magnitude, i.e. $R=100$, one gets the blue solid line in Fig. 8. Thus the filter proves again to be a good estimator for the Viswanath constant.

5.3. *Random Fibonacci numbers estimation – Case Q=1*

The parameters of the simulations are as follows:

1. $p \in \{0, 0.2, 0.5, 0.8, 1\} \Rightarrow 1-p=q \in \{1, 0.8, 0.5, 0.2, 0\}$.
2. Process noise covariance matrix $Q=1$.
3. Measurement noise covariance matrix $R=1$.
4. Number of Fibonacci recursion steps $N_{fs}=500$.
5. Number of independent Monte Carlo trials $N_{mc}=300$.
6. The initial conditions, i.e. a priori estimated Fibonacci numbers, are chosen as follows:

$$\hat{x}_{0|-1} = \begin{bmatrix} \text{uniform random}(0, 1000) \\ \text{uniform random}(0, 1000) \end{bmatrix},$$

$$P_{0|-1} = \begin{bmatrix} \hat{x}_{0|-1}(1)^2 & 0 \\ 0 & \hat{x}_{0|-1}(2)^2 \end{bmatrix}.$$

The performance of MPH is measured by the *Root Mean Square Error* (RMSE) of the first component of the state, i.e. f_{nf_k}. Averaging is performed over N_{mc} independent Monte Carlo trials for different Fibonacci recursions (except for $p=1$ and $p=0$) and different, independently generated, measurement noise realizations. Simulation results for $p \in \{0, 0.2, 0.5, 0.8, 1\}$ are depicted in Fig. 9, where the solid lines are the estimation errors averaged over all the independent Monte Carlo trials.

As it can be noticed from Fig. 9, the filtering algorithm still works extremely well w.r.t. the estimation error, for each value of p.

5.4. *Estimating the Viswanath constant, case Q=1, Q=100, R=1 and R=100*

As it has be done in Section 5.1, it is of interest to verify if, by having $Q>0$, the MPH is still capable of estimating the Viswanath constant. Fig. 10 depicts $\hat{e}^{\gamma_f}$ with different values of both Q and R. The red solid line is the value of the Viswanath constant. Thus the filter still proves to be capable of estimating the Viswanath constant and to be robust w.r.t. higher values of R and Q.

6. Estimation of a malware propagation in a computer network

An application scenario concerning the estimation of a malware propagation in a computer network will be presented. By adding a randomly changing behavior to the innovative malware propagation model proposed in [24], our approach turns out to be more realistically defined.

Table 3
Most Probable Hypothesis (MPH) stochastic filtering algorithm for the random Fibonacci dynamic problem at time instant k.

INPUT: $\hat{x}_{k|k} \sim \mathcal{N}(\hat{x}_{k|k}, P_{k|k})$

1. PREDICTION

Prediction considering system (11)

$\hat{x}_{+;k+1|k} = A_+ \hat{x}_{k|k}$

$P_{+;k+1|k} = A_+ P_{k|k} A_+^T$

Prediction considering system (21)

$\hat{x}_{-;k+1|k} = A_- \hat{x}_{k|k}$

$P_{-;k+1|k} = A_- P_{k|k} A_-^T$

2. CORRECTION

Correction considering system (11)

$S_{+;k+1} = CP_{+;k+1|k} C^T + R$

$L_{+;k+1} = P_{+;k+1|k} C^T (S_{+;k+1})^{-1}$

$e_{+;k+1} = z_{k+1} - C\hat{x}_{+;k+1|k}$

$\hat{x}_{+;k+1|k+1} = \hat{x}_{+;k+1|k} + L_{+;k+1} e_{+;k+1}$

$P_{+;k+1|k+1} = P_{+;k+1|k} - L_{+;k+1} S_{+;k+1} (L_{+;k+1})^T$

$\alpha_{+;k+1} = \mathcal{N}(z_{k+1}; \hat{x}_{+;k+1|k}, S_{+;k+1})$

Correction considering system (21)

$S_{-;k+1} = CP_{-;k+1|k} C^T + R$

$L_{-;k+1} = P_{-;k+1|k} C^T (S_{-;k+1})^{-1}$

$e_{-;k+1} = z_{k+1} - C\hat{x}_{-;k+1|k}$

$\hat{x}_{-;k+1|k+1} = \hat{x}_{-;k+1|k} + L_{-;k+1} e_{-;k+1}$

$P_{-;k+1|k+1} = P_{-;k+1|k} - L_{-;k+1} S_{-;k+1} (L_{-;k+1})^T$

$\alpha_{-;k+1} = \mathcal{N}(z_{k+1}; \hat{x}_{-;k+1|k}, S_{-;k+1})$

3. NEW HYPOTHESIS

$\kappa = \alpha_{+;k+1} + \alpha_{-;k+1}$

If $\frac{\alpha_{+;k+1}}{\kappa} > \frac{\alpha_{-;k+1}}{\kappa}$ **then**

$\hat{x}_{k+1|k+1} \sim \mathcal{N}(\hat{x}_{+;k+1|k+1}, P_{+;k+1|k+1})$

else

$\hat{x}_{k+1|k+1} \sim \mathcal{N}(\hat{x}_{-;k+1|k+1}, P_{-;k+1|k+1})$

end if

OUTPUT: $\hat{x}_{k+1|k+1}$

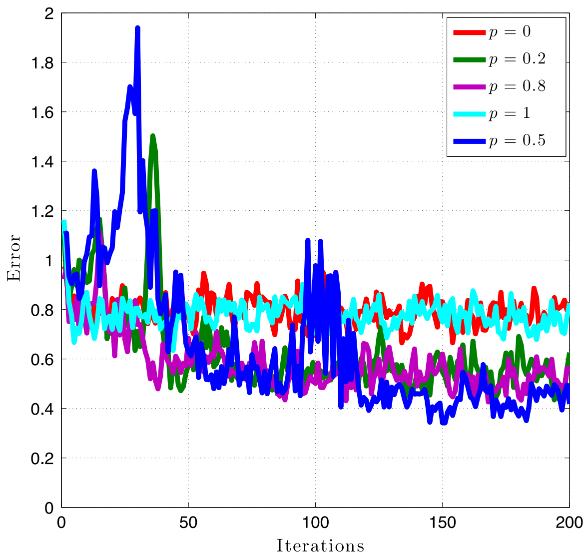

Fig. 7. State estimation error graph for random Fibonacci filtering for different values of p. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

Table 4
Estimate covariance matrix $P_{500|500}$.

$$p=0 \Rightarrow q=1:\; P_{500|500} = \begin{bmatrix} \mathbf{0.61803} & -0.38197 \\ -0.38197 & 0.23607 \end{bmatrix}$$

$$p=0.2 \Rightarrow q=0.8:\; P_{500|500} = \begin{bmatrix} 0.30972 & -0.13157 \\ -0.13157 & 0.16919 \end{bmatrix}$$

$$p=0.5 \Rightarrow q=0.5:\; P_{500|500} = \begin{bmatrix} 0.18617 & -0.008062 \\ -0.008062 & 0.1387 \end{bmatrix}$$

$$p=0.2 \Rightarrow q=0.8:\; P_{500|500} = \begin{bmatrix} 0.28103 & 0.1129 \\ 0.1129 & 0.1725 \end{bmatrix}$$

$$p=1 \Rightarrow q=0:\; P_{500|500} = \begin{bmatrix} \mathbf{0.61803} & 0.38197 \\ 0.38197 & 0.23607 \end{bmatrix}$$

In the recently published work [24] the authors propose a general model for a malware propagation in a computer network. Specifically, the model takes into account the following different factors that have a direct impact on the propagation:

1. the number of Internet Protocol (IP) addresses that the malware scans in the network looking for vulnerable hosts;
2. the number of IP ports[1] at which the malware tries to exploit vulnerabilities;
3. the number of threads[2] in the malware;
4. the number of destructed hosts (formatted hard drive) over the number of infected hosts;
5. the rate at which the vulnerable machines are patched;
6. the rate at which new vulnerable hosts join the network;
7. the number of contagious hosts that can infect other hosts; and
8. the number of vulnerable host/port combinations.

The authors, inspired by the Fibonacci sequences, propose a *Generalized Fibonacci Malware Propagation* (GFMP) in which a malware cannot scan or infect other hosts until it has gained control of infected hosts. The propagation time of the malware is thus used to model the delay between the time instant when the host gets attacked and the time instant when the host starts to attack other hosts. This is exactly the same logic used in the Fibonacci recursion with the rabbit paradigm. New-born rabbits cannot give birth to baby rabbits immediately. Instead, they need some time to mature, which is reminiscent of the infection/propagation time problem discussed above: a captured host cannot scan and infect other hosts until its infection matures, i.e. until it is completely infected. To our

[1] A port number is a 16-bit unsigned integer, thus in the range [1; 65535].

[2] A thread is a light-weight process that can be managed independently by an operating system. Multiple threads can exist within the same program and share resources, e.g. the memory.

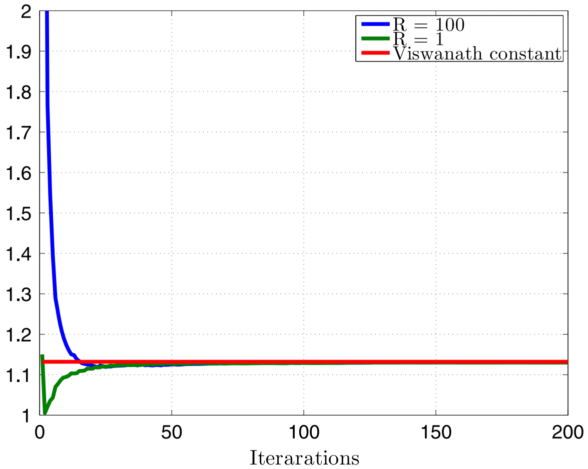

Fig. 8. Estimation of the Viswanath constant having $R=1$ and $R=100$. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

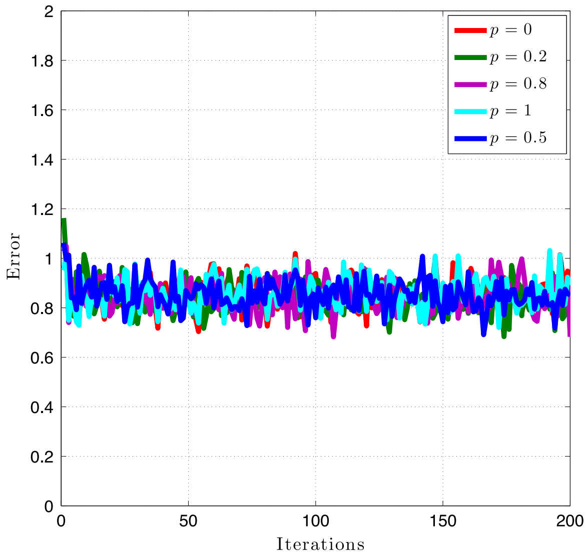

Fig. 9. State estimation error graph for random Fibonacci filtering for different values of p. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

purposes, consider the simplified model of Eq. (15) of [24]

$$I_k = \begin{cases} 0, & \text{if } k=0 \\ 1, & \text{if } k=1 \\ I_{k-1} + \beta \cdot I_{k-2}, & \text{if } k>1 \end{cases} \tag{54}$$

where I_k is the number of infected hosts in the network and

$$\beta = \frac{t \cdot V}{o \cdot (2^{32} - d)} \tag{55}$$

is the infection coefficient with t the number of threads in the malware, V the number of vulnerable host/port combinations, o the number of ports at which the malware tries to exploit vulnerabilities and d the number of IPv4[3] (IP version 4) addresses that the malware avoids scanning in the network, i.e. 2^{32} - d is the total number of IPv4 addresses that the malware scans in the network looking for vulnerable hosts. Note that system (54) recalls the Fibonacci sequence (10).

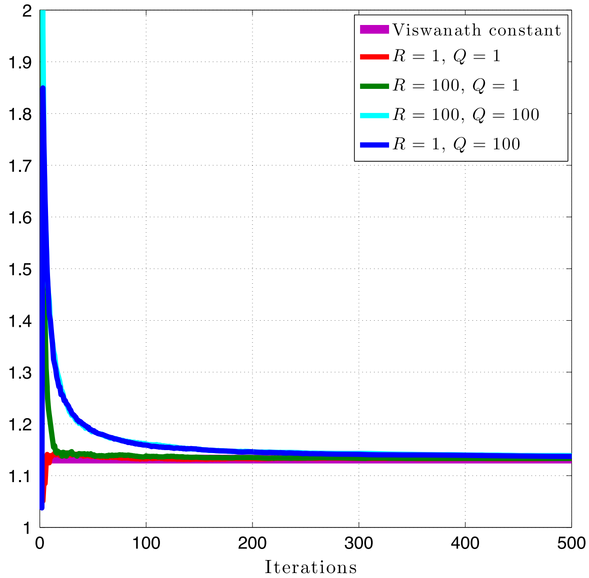

Fig. 10. Estimation of the Viswanath constant with different values of R and Q. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

Consider now a more realistic extension of the malware model (54) in which the infection randomly propagates with different rates β given a probability p. In particular, the specific stochastic model assumed herewith is as follows:

$$I_k = \begin{cases} 0, & \text{if } k=0 \\ 1, & \text{if } k=1 \\ I_{k-1} + \beta_1 \cdot I_{k-2}, & \text{if } k>1 \text{ with probability } p \\ I_{k-1} + \beta_2 \cdot I_{k-2}, & \text{if } k>1 \text{ with probability } 1-p \end{cases} \tag{56}$$

where $p=0.25$, $V=10{,}000{,}000$, $o=30$, $d=2^{25}$ and t is chosen as 0 and 1024 respectively for β_1 and β_2. By adopting as number of threads $t=0$ the behavior of the malware has a one step random delay with probability $p=0.25$.

By having system (56) it is straightforward to apply the proposed MPH described in Table 3. Assume a scenario in which an administrator of a given network needs to estimate the number of infected computers so that he can adopt proper countermeasures. The administrator is assumed to be capable of remotely querying the computers of the network to recognize the infected ones. This type of information can be considered as a noisy measurement, i.e. the administrator will receive a partial information of the infection w.r.t. actual true infected

[3] IPv4 uses 32-bit (four-byte) addresses, which limits the address space to 2^{32} addresses.

computers. The overall scenario can be modeled assuming the following parameters:

1. Process noise covariance matrix $Q=0$.
2. Measurement noise covariance matrix $R=10{,}000$.
3. Number of malware propagation steps $N_{fs}=400$.
4. Number of independent Monte Carlo trials $N_{mc}=100,000$.
5. The initial conditions, i.e. a priori estimated Fibonacci numbers, are chosen as follows:

$$\hat{x}_{0|-1}=\begin{bmatrix}\text{uniform random}(0,10,000)\\ \text{uniform random}(0,10,000)\end{bmatrix},$$

$$P_{0|-1}=\begin{bmatrix}\hat{x}_{0|-1}(1)^2 & 0\\ 0 & \hat{x}_{0|-1}(2)^2\end{bmatrix}.$$

It is also assumed that the estimation of the malware propagation does not start at time $k=0$, instead it is delayed at time $k=200$. The main reason of this choice is because the administrator of a network is unaware of the infection until a considerable number of computers are infected. To model this peculiarity, the initial time of the estimation process is thus delayed.

Simulation results for the aforementioned case study is depicted in Fig. 11, where the blue solid line is the estimation error averaged over all the independent Monte Carlo trials, while the red dashed line is the ± 1 standard deviation.

As it can be seen from Fig. 11, the performance of the proposed MPH in the aforementioned scenario is satisfactory w.r.t. the high uncertainty of the measurements ($R=10{,}000$). In particular, the steady state averaged estimation error settles approximately to 3000, with a standard deviation of the same order of magnitude.

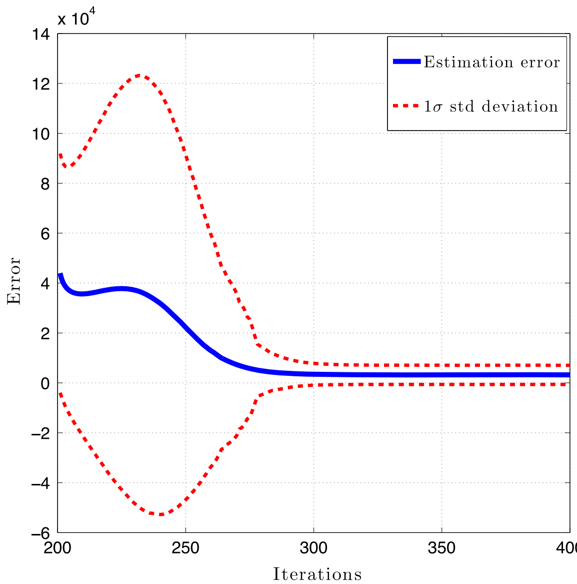

Fig. 11. Estimation error on the total number of infected computers of a given network. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

In the following we give comments of the practical application described in this section. The stochastic filter “4*F*”, which has been conceived in this paper, is capable of estimating a random Fibonacci sequence and can find a relevant application in cyberattacks due to malware propagating in a computer network. A modeling of this has been recently proposed in [24] wherein the authors propose to model the dynamics of an infection over a network of computers as the evolution of a generalized Fibonacci sequence over a tree. This matching is done considering that the payload of the malware is destructive and no patch can be applied. This implies that an infected computer is removed from the network. Besides, they consider a propagation time for the infection and a maturation time on the infected computer. This makes the application of a Fibonacci sequence quite straightforward and indeed remarkable. However, this model can be upgraded by just removing the condition of destructive payload by using a generalized random Fibonacci sequence. So, we could expect that, in some cases, the network administrator will be able to apply a patch and to put a computer back at work. As this process is typically random, our generalized Fibonacci random model is the one that could best fit the bill. So, using our stochastic filter “4*F*”, it is possible to design a tool to forecast a cyberattack in a computer network and, possibly, apply promptly proper countermeasures.

7. Conclusions and future work

Future work will consider that the measurement noise for the estimation of a malware propagation in a computer network should not be assumed as constant. In particular such noise could be modeled as a percentage of the real number of infected nodes at a given time instant k. This kind of assumption would violate the Gaussian assumption of the likelihood measurement equation (9), thus specific analysis and considerations should be carried out. An alternative approach could be the one based on the so-called “interval analysis” [25] where, rather than specifying a probability density function of the noise, one specifies the interval to which the measurements error might belong.

A further research topic could be the following. As it has been shown in [26], Black–Scholes equation, describing options pricing, can be recovered through a discrete model known as Binomial Options Pricing Model (BOPM). BOPM uses, at each discrete time step, a tree with an estimation of the option price depending on a given probability p. It is part of our future activity to extend the application area of the “4*F*” filter to the case of the BOPM.

Acknowledgments

Professor L. Chisci and Professor G. Battistelli (University of Florence) are warmly acknowledged for careful reading of the manuscript and providing useful suggestions. We are also very grateful to the Associate Editor and the referees for the excellent work during the review phase of this manuscript.

Appendix A. Lemma 1 – product of two Gaussian densities

Consider two random variables X_1 and X_2 which have Gaussian PDF:

$$X_1|X_2 \sim \mathcal{N}(x_1; Ax_2, Q) \tag{A.1}$$

$$X_2 \sim \mathcal{N}(x_2; m, P) \tag{A.2}$$

then the product of the densities is

$$\mathcal{N}(x_1; Ax_2, Q) \cdot \mathcal{N}(x_2; m, P) = \alpha(x_1)\mathcal{N}(x_2; m_u, P_u) \tag{A.3}$$

$$\alpha(x_1) = \mathcal{N}(x_1; Am, APA^T + Q) \tag{A.4}$$

$$m_u = m + PA^T(APA^T + Q)^{-1}(x_1 - Am) \tag{A.5}$$

$$P_u = P - PA^T(APA^T + Q)^{-1}AP \tag{A.6}$$

Proof. Proof of Lemma 1 can be found in [27]. □

Appendix B. Lemma 2 – marginalization of Gaussian densities

Consider two random variables X_1 and X_2 which have Gaussian PDF:

$$X_1|X_2 \sim \mathcal{N}(x_1; Ax_2, Q) \tag{B.1}$$

$$X_2 \sim \mathcal{N}(x_2; m, P) \tag{B.2}$$

with A, d, Q, m and P known and Q and P semidefinite positive. Then, the marginalization of the product of the two distributions w.r.t x_1 is

$$\int \mathcal{N}(x_1; Ax_2, Q) \cdot \mathcal{N}(x_2; m, P)\, dx_2 = \mathcal{N}(x_1; Am, APA^T + Q) \tag{B.3}$$

Proof. The proof of Lemma 2 is easily derived using Lemma 1 and the integral w.r.t. x_2 as follows:

$$\begin{aligned}
&\int \mathcal{N}(x_1; Ax_2, Q) \cdot \mathcal{N}(x_2; m, P)\, dx_2 \\
&\quad = \{\text{See Lemma 1, Appendix A}\} \\
&\quad = \int \mathcal{N}(x_1; Am, APA^T + Q) \cdot \mathcal{N}(x_2; m_u, P_u)\, dx_2 \\
&\quad = \mathcal{N}(x_1; Am, APA^T + Q) \qquad \square
\end{aligned} \tag{B.4}$$

References

[1] L. Sigler, Fibonacci's Liber Abaci, Springer, New York, 2002.

[2] J. Kiefer, Sequential minimax search for a maximum, Proc. Am. Math. Soc. 4 (3) (1953) 502–506.

[3] M.L. Fredman, R.E. Tarjan, Fibonacci heaps and their uses in improved network optimization algorithms, J. Assoc. Comput. Mach. 34 (3) (1987) 596–615.

[4] W.-J. Hsu, Fibonacci cubes – a new interconnection topology, IEEE Trans. Parallel Distrib. Syst. 4 (1) (1993) 3–12.

[5] W.-J. Hsu, M.J. Chung, Generalized Fibonacci cubes, in: International Conference on Parallel Processing – ICPP 1993, Syracuse, NY, USA, 1993, pp. 299–302.

[6] Y. Matiyasevich, Hilbert's 10th Problem, The MIT Press, Cambridge, MA, 1993 ISBN: 0-262-13295-8.

[7] Y. Matiyasevich, J. Robinson, Reduction of an arbitrary Diophantine equation to one in 13 unknowns, Acta Arith. XXVII (1975) 521–549.

[8] A. Benavoli, L. Chisci, A. Farina, Fibonacci sequence, golden section, Kalman filter and optimal control, Signal Process. 89 (8) (2009) 1483–1488.

[9] A. Capponi, A. Farina, C. Pilotto, Expressing stochastic filters via number sequences, Signal Process. 90 (7) (2010) 2124–2132.

[10] J. Donoghue, The Kalman lter for complex Fibonacci systems, ISRN Signal Processing 2012 (2012). Article ID 631873.

[11] J. Donoghue, State estimation and control of the Fibonacci system, Signal Process. 91 (5) (2011) 1190–1193.

[12] M. Livio, The Golden Ratio, Broadway Books, New York, NY, 2002.

[13] A.S. Posamentier, I. Lehmann, The Fabulous Fibonacci Number, Prometheus Press, Amherst, NY, 2007.

[14] D. Viswanath, Random Fibonacci sequences and the number 1.13198824, Math. Comput. 69 (231) (2000) 1131–1156.

[15] M. Embree, L.N. Trefethen, Growth and decay of random Fibonacci sequences, in: Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, vol. 455, 1999, pp. 2471–2485.

[16] H. Furstenberg, H. Kesten, Products of random matrices, Ann. Math. Stat. (1960) 457–469.

[17] S. Aluru, Lagged Fibonacci random number generators for distributed memory parallel computers, J. Parallel Distrib. Comput. 45 (1) (1997) 1–12.

[18] A. Biello, C. J. Cacciatore, L. Thomas, Random Fibonacci-type sequences in online gambling, Department of Mathematics, University of Massachusetts Dartmouth, North Dartmouth, MA, Final Report, December 19, 2011.

[19] K. Pearson, The Problem of the Random Walk, Nature 72 (1905) 294.

[20] M.A. Stern, Über eine zahlentheoretische Funktion, J. Reine Angew. Math. 55 (1858) 193–220.

[21] A. Brocot, Calcul des rouages par approximation nouvelle méthode, Revue Chon. 3 (1861) 186–194.

[22] A.H. Jazwinski, Stochastic processes and filtering theory, Academic Press, San Diego, CA, 1970 (Dover Publications, Mineola, NY, 2007).

[23] A. Alessandri, M. Baglietto, G. Battistelli, A maximum-likelihood Kalman filter for switching discrete-time linear systems, Automatica 46 (11) (2010) 1870–1876.

[24] Y. Zhang, B. Bhargava, P. Hurni, The effects of threading, infection time, and multiple-attacker collaboration on malware propagation, in: 28th IEEE International Symposium on Reliable Distributed Systems (SRDS), Niagara Falls, New York, USA, 2009, pp. 73–82.

[25] R.E. Moore, R.B. Kearfott, M.J. Cloud, Introduction to Interval Analysis, SIAM, Philadelphia, PA, 2009.

[26] J.C. Cox, S.A. Ross, M. Rubinstein, Option pricing: a simplified approach, J. Financ. Econ. 7 (3) (1979) 229–263.

[27] Y.C. Ho, R.C.K. Lee, A Bayesian approach to problems in stochastic estimation and control, IEEE Trans. Autom. Control 9 (4) (1964) 333–339.