A Receding Horizon Algorithm to Generate Binary Signals with a Prescribed Autocovariance

Cristian R. Rojas James S. Welsh Graham C. Goodwin

***Abstract*— Optimal test signals are frequently specified in terms of their second order properties, e.g. autocovariance or spectrum. However, to utilize these signals in practice, one needs to be able to produce realizations whose second order properties closely approximate the prescribed properties. Of particular interest are binary waveforms since they have the highest form-factor in the sense that they achieve maximal energy for a given amplitude. In this paper we utilize ideas from model predictive control to generate a binary waveform whose sampled autocovariance is as close as possible to some prescribed autocovariance. Several simulated examples are presented verifying the veracity of the algorithm. Also, a proof of convergence is given for the special case of bandlimited white noise. This proof is based on expressing the system in the form of a switched linear system.**

I. Introduction

In many fields, the problem of generating a waveform having specified second order properties arises, see for example [1], [5], [6], [7], [12], [16], [17]. For instance, in experiment design [3], [8] one typically obtains an optimal test signal specified in terms of its spectral properties. This leads to the problem of implementing a real signal with a specified spectrum. Moreover, it is usual that the input should also be constrained in its amplitude, i.e. the amplitude must lie in an interval $[a, b] \subset \mathbb{R}$. In general, frequency domain techniques do not work properly with this kind of constraint, and as such are translated into an 'equivalent' power constraint under which the input is designed to satisfy the conditions.

In many applications it is important to implement an input signal which, within the constraints of its amplitude, has maximum power. This is the case, for example, in experiment design, where the quality of the estimation typically increases with the signal to noise ratio. The signal to noise ratio is obviously improved by choosing an input with high power. Binary signals have precisely this desirable property: their power is maximum for a given amplitude constraint [13]. This then motivates the question of how to design a binary signal with a given autocovariance.

This question arises in many areas of physics and engineering. For example, the study of two-phase random media [16], [17], [1], [12], [6] (e.g. ferromagnetic materials, composites, porous materials, microemulsions, ceramics or polymer blends) involves the measurement of the statistical properties of a material. The measurements are commonly restricted to the first two moments, and then the simulation of the material properties must be based on these measurements.

Several techniques have been proposed to solve this problem (see e.g. [5], [7], [16], [17], [1], [12], [6] and the references therein). For example, [5] and [7] consider a scheme consisting of a linear system followed by a static nonlinearity. The nonlinear block, in this case, is used to force the output signal to be binary, and the linear system is tuned to produce an output signal with the desired autocovariance. However, it can be shown that this method has severe limitations, e.g. it cannot be used to generate binary signals with a bandlimited spectrum [7], [15]. A similar procedure consisting of a linear system followed by a level crossing block is developed in [6]. A simulated annealing method is proposed in [16] and [17]. The methods outlined above generally involve complex calculations and are computationally intensive.

In this paper, we develop a simple procedure to solve the same problem, based on the use of the Receding Horizon concept commonly employed in Model Predictive Control [2]. Heuristically speaking, the idea is to solve, for each time instant, a finite horizon optimisation problem to find the optimal set of the next, say, T values of the sequence such that the sampled autocovariance sequence so obtained is as close as possible (in a prescribed sense) to the desired autocovariance. One then takes the first term of this optimal set for the sequence, advances time by one step and repeats the procedure. The idea behind this procedure is thus closely related to finite alphabet receding horizon control [10], where receding horizon concepts are employed to control a linear plant whose input is restricted to belong to a finite set.

Notice that in order to find the true optimal binary sequence, we would have to compute the sample autocovariance function of all sequences in $\{0, 1\}^N$ and then choose the sequence whose autocovariance is closest to the desired one according to some prescribed norm. This procedure, however, would be computationally intractable as it involves 2^N comparisons, a truly large number in general.

Several kinds of measures can be used to compare the sampled autocovariance of the generated signal with the desired autocovariance, including the Euclidean or the infinity norm of their difference. However, we have verified via simulations that the Euclidean norm produces very good results when compared to other norms. Furthermore, by Theorem 1 of Section IV, the algorithm is shown to converge for a special case when the Euclidean norm is used.

The algorithm described in this paper is fast and easy to implement when compared with the existing methods. The algorithm can also be run in realtime. This allows

The authors are with the School of Electrical Engineering and Computer Science, *The* University *of* Newcastle, University Drive, Callaghan, NSW 2308, Australia
Cristian.Rojas@studentmail.newcastle.edu.au
James.Welsh@newcastle.edu.au
Graham.Goodwin@newcastle.edu.au

the possibility of implementing adaptive input generation schemes, which can be useful when the signal properties must change with time, as in sequential experiment design procedures [14, pg. 331].

To demonstrate the application of the algorithm, two examples, motivated by experiment design, are provided. A typical input signal used in system identification is bandlimited white noise [8, Section 13.3]. In this paper we show how the proposed algorithm is used to generate this type of signal and also provide the obtained spectrum to highlight how closely it approximates the desired spectrum. The second example is inspired by recent work on experiment design where it was shown that a more robust input for a particular class of systems is in fact one with a bandlimited '$1/f$' spectrum [11], [4]. We again provide the spectrum generated by the receding horizon algorithm as well as that of the prescribed signal, for the purpose of comparison.

The paper is structured as follows. In Section II we present the algorithm and provide a detailed explanation. Section III shows the results of some numerical examples that illustrate the quality of the signals generated by the algorithm. In Section IV we prove convergence for the special case of generating a pseudo white noise sequence (i.e. when the desired autocovariance sequence is a Kronecker delta at 0). We present conclusions in Section V.

II. The algorithm

In this Section we formulate and develop the receding horizon algorithm that generates a binary signal with a prescribed autocovariance. This is done in two parts. Firstly we convert the problem to an equivalent one that allows us to simplify the computation and to force the generated signal to have zero mean. We then develop the algorithm as a series of steps and finally present it as Matlab® code.

Let $\{r_k^d\}_{k=0}^{\infty}$ be a given desired autocovariance sequence. Also, let N be the length of the signal to be generated, n the number of lags of $\{r_k^d\}_{k=0}^{\infty}$ to be compared to the corresponding lags of the sampled autocovariance sequence of the designed signal, and m be the length of the receding horizon over which we apply the optimisation algorithm.

Notice that in order for $\{r_k^d\}_{k=0}^{\infty}$ to be a valid autocovariance sequence, it must be positive definite [9, pg. 329], i.e.

$$\sum_{1\leq i\leq j\leq M} a_i a_j^* r_{i-j} \geq 0 \tag{1}$$

for every $M \in \mathbb{N}$ and $\{a_i\}_{k=1}^{M} \subseteq \mathbb{C}^M$. Here * denotes complex conjugation.

For simplicity, we force the designed signal to have zero mean and restrict its values to $\{-1,1\}$. This implies that r_0^d must be equal to 1.

The proposed algorithm is as follows:

(A) Conversion to an Equivalent Problem

We begin by converting the desired autocovariance sequence $\{r_k^d\}_{k=0}^{\infty}$ into the non-central autocovariance of a $\{0,1\}$ sequence. That is, define

$$\tilde{r}_k^d := \frac{1}{4}(r_k^d + 1), \qquad k = 0, \dots, n. \tag{2}$$

Remark 1: The idea here is that the algorithm will generate a sequence $\{\tilde{y}_i\}_{i=1}^{N}$ taking only the values $\{0,1\}$ such that

$$\frac{1}{N}\sum_{i=k+1}^{N} \tilde{y}_i \tilde{y}_{i-k} \approx \tilde{r}_k^d, \qquad k = 0, \dots, n, \tag{3}$$

where the left side corresponds to the sampled non-central autocovariance of the signal evaluated at lag k. The approximation criterion will be the Euclidean norm, as shown in step 5 below.

Notice that since $\tilde{y}_i \in \{0,1\}$ for every i, we see that equation (3) for $k=0$ is equivalent to

$$\frac{1}{N}\sum_{i=1}^{N} \tilde{y}_i \approx \tilde{r}_0^d = \frac{1}{2}, \qquad k = 0, \dots, n. \tag{4}$$

This implies that equation (3) is actually forcing $\{\tilde{y}_i\}_{i=1}^{N}$ to have sampled mean $1/2$, or equivalently, forcing the designed signal to have zero sampled mean.

(B) The Algorithm

We now provide an outline of the algorithm as a series of steps:

1) Set $t = 1$.
2) Set $(y'_t, \dots y'_{t+m-1}) = \mathbf{0}_{1,m} \in \{0,1\}^m$, where $\mathbf{0}_{1,m}$ denotes a zero matrix of order $1 \times m$.
3) Compute the first n lags of the sampled non-central autocovariance of $(\tilde{y}_1, \dots, \tilde{y}_{t-1}, y'_t, y'_{t+1}, \dots y'_{t+m-1})$ (or of $(y'_1, \dots y'_m)$, if $t = 1$) via
$$r'_k := \frac{1}{t+m-1}\sum_{i=k+1}^{t+m-1} y'_i y'_{i-k}, \qquad k = 0, \dots, n, \tag{5}$$
where we are considering $y'_i = \tilde{y}_i$ for $i = 1, \dots, t-1$.
4) Generate a new m-tuple $(y'_{t+1}, \dots, y'_{t+m}) \in \{0,1\}^m$ and repeat step 3 until all m-tuples have been tested.
5) Let $\tilde{y}_t = y'_t$ for the m-tuple $(y'_t, \dots y'_{t+m-1}) \in \{0,1\}^m$ for which $\|\{r'_i\}_{i=0}^{n} - \{\tilde{r}_i^d\}_{i=0}^{n}\|_2$ is minimum. If this norm is equal for both values of y'_t, take $\tilde{y}_t = 0$.
6) If $t < N$, let $t = t+1$ and go to step 2.
7) Convert the $\{0,1\}$ N-tuple $(\tilde{y}_1, \dots, \tilde{y}_N)$ into a $\{-1,1\}$ N-tuple $(y_1, \dots, y_N)$ via
$$y_t := 2\tilde{y}_t - 1, \qquad t = 1, \dots, N. \tag{6}$$

It is straightforward to extend the method to more general cases. For example, to generate signals with non zero mean $\overline{y}$ and/or taking values in $\{a, b\}$, it is necessary to alter equations (2) and (6), and to let $r_0^d = \overline{y}^2$.

To provide further insight into the implementation of the algorithm we add the following comments: First, the computation of the sampled autocovariance at step 3 can be done in a recursive manner (with respect to t), which reduces the execution time of the algorithm. Second, the execution time of the algorithm depends exponentially on m. However, it can be empirically verified that $m = 1$ gives very good results (in fact, we will show later in Section IV that the algorithm converges successfully for $m = 1$ in a particular case). Thus, for ease of reference, we present below an optimised version of the algorithm for $m = 1$ in Matlab® code:

```
% Initialization
y  = zeros(N+n, 1);
r0 = zeros(n+1, 1);
r1 = zeros(n+1, 1);

% Conversion of the autocovariance
% sequence to the equivalent problem
rd = 0.25*(rd + 1);

for i = 1:N,

  % Calculation of the next
  % autocovariance sequence, if we
  % add "0" or "1" to the output
  % signal, respectively
  r0 = r0 - rd;
  r1 = r0 + [1; y(n+i-1:-1:i)];

  % Comparison of the resulting
  % autocovariance sequences
  if norm(r0) > norm(r1),
    y(n+i)  = 1;
    r0      = r1;
  end

end

% Conversion to the original problem
% to obtain the desired sequence
y = 2*y(n+1:end) - 1;
```

Here, `y` is the generated input signal; `r0` and `r1` are the non-central autocovariances of the signal plus an additional 0 or 1, respectively; and `rd` is the desired autocovariance sequence (to be specified by the user).

As with $\{r_k^d\}_{k=0}^{\infty}$, the user of the algorithm also has to choose three other variables: N, n and m. The choice of N depends on the external circumstances which arise in the particular context where the user needs to generate a binary signal; for example, in experiment design, N usually depends on a number of factors such as the sampling period, the total time the plant will be available for experimentation and the required precision of the parameter estimates. The variable n depends on the characteristics of the prescribed autocovariance sequence $\{r_k^d\}_{k=0}^{\infty}$; in particular, it is convenient to choose n such that $r_k^d \approx 0$ for all $k > n$. The choice of m is a tradeoff between precision and execution time, as will be seen in the examples provided in the next Section; however, for most practical applications, it seems that $m = 1$ provides a very good performance.

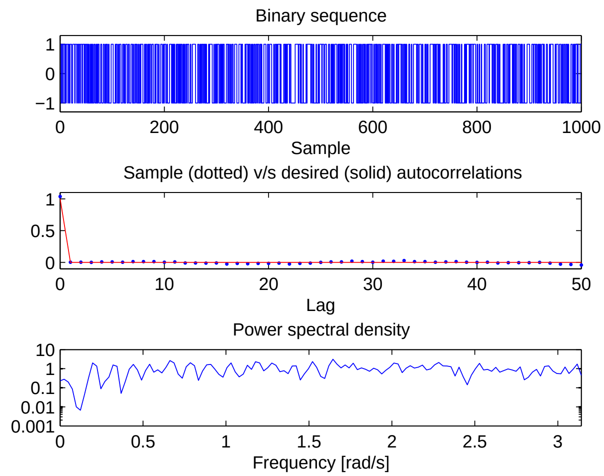

Fig. 1. Characteristics of the generated pseudo white noise signal for $m = 1$, $N = 1000$ and $n = 50$.

III. Input Signal Design Examples

In this section we present two examples. The first example deals with the problem of generating pseudo random signals (i.e. pseudo white noise). The second example relates to the generation of bandlimited '$1/f$' noise, which has recently been shown to possess important robustness properties in experiment design [11].

A. Pseudo white noise

The tuning parameters of the algorithm are N, the number of data points to be generated; n, the number of lags to be considered for the comparison of the desired and sampled autocovariance sequences; and m, the length of the receding horizon.

For $m = 1$, $N = 1000$ and $n = 50$, we obtain the results presented in Figure 1. From this figure, we can see that both the autocovariance and spectrum of the generated signal are very similar to those of white noise. If we increase N to 10^6, we obtain Figure 2, which shows that the algorithm has remarkably good asymptotic properties. With respect to the execution time, we find that the algorithm requires only a small amount of time to run, e.g. on a PC with a Pentium III 871 Mhz CPU and 512 Mb of RAM it takes less than 42 sec to generate 10^6 points! A plot of the dependence of the cost function on N is given in Figure 3. Note that the cost is on a logarithmic scale. From this figure it can be seen that the convergence rate of the algorithm appears to be $O(1/N)$ (although the proof given below in Section IV establishes a convergence rate of $O(1/\sqrt{N})$, as it is based on a conservative upper bound for the cost function $\|\{r_i'\}_{i=0}^n - \{\tilde{r}_i^d\}_{i=0}^n\|_2$).

In Figure 4 the dependence of the cost function on the horizon length m is shown (for $N = 10^4$ and $n = 50$). As expected, it can be seen that the cost function decreases with m. Note however that the computational complexity of the

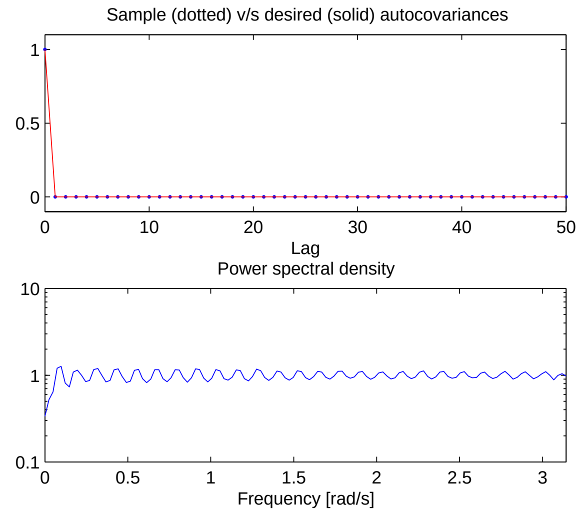

Fig. 2. Characteristics of the generated pseudo white noise signal for $m = 1$, $N = 10^6$ and $n = 50$.

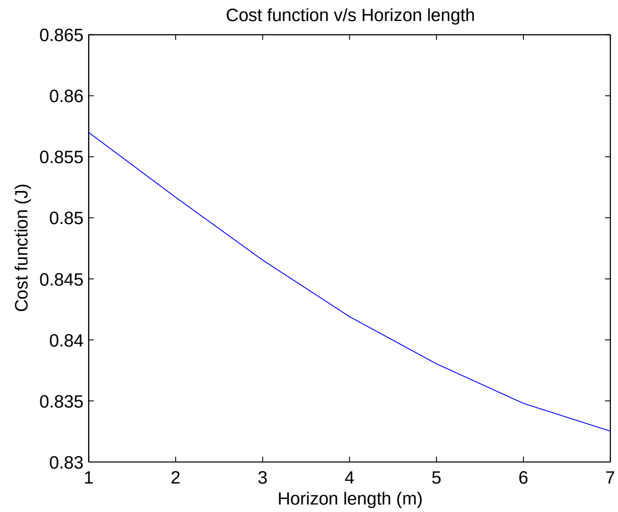

Fig. 4. Dependence of the cost function $J := \|\{r'_i\}_{i=0}^n - \{\tilde{r}_i^d\}_{i=0}^n\|_2$ on the horizon length, for $N = 1000$ and $n = 50$, when generating pseudo white noise.

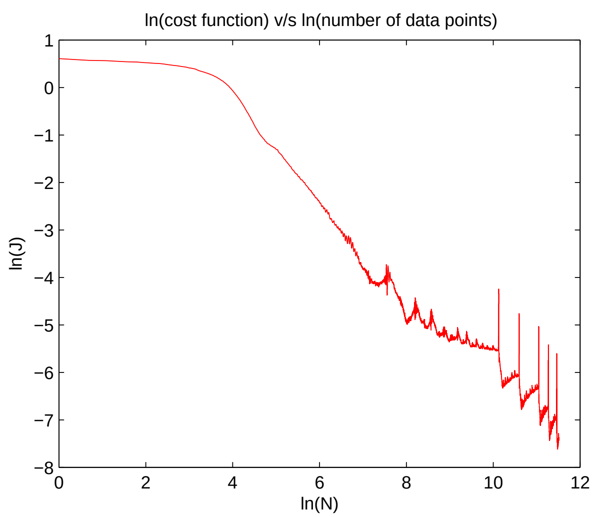

Fig. 3. Dependence of the cost function $J := \|\{r'_i\}_{i=0}^n - \{\tilde{r}_i^d\}_{i=0}^n\|_2$ on N, for $m = 1$ and $n = 50$, when generating pseudo white noise.

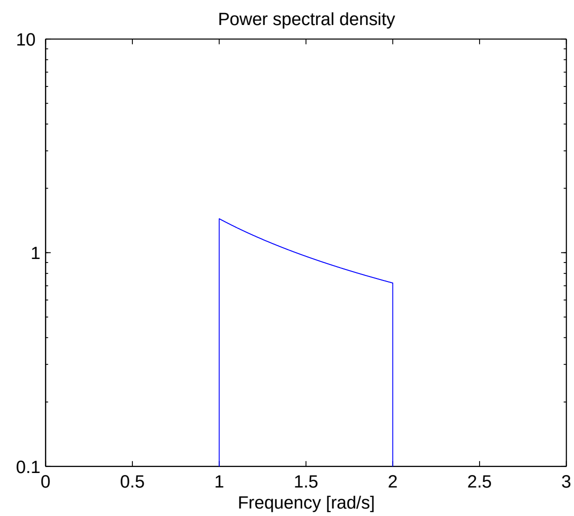

Fig. 5. Power spectral density of bandlimited '$1/f$' noise signal for $\underline{\omega} = 1$ and $\overline{\omega} = 2$.

algorithm depends exponentially on m, thus a tradeoff needs to be considered between accuracy and execution time.

B. Bandlimited '$1/f$' noise

Bandlimited '$1/f$' noise is defined by the following spectrum:

$$\phi^{1/f}(\omega) := \begin{cases} \dfrac{1/\omega}{\ln\overline{\omega} - \ln\underline{\omega}}, & \omega \in [\underline{\omega}, \overline{\omega}], \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where $\underline{\omega}, \overline{\omega} \in \mathbb{R}^+$ ($\underline{\omega} < \overline{\omega}$). The autocovariance sequence of this signal is given by

$$r_k^{1/f} := \frac{1}{\ln\overline{\omega} - \ln\underline{\omega}} \int_{\underline{\omega}}^{\overline{\omega}} \frac{\cos kx}{x} dx; \quad k \in \mathbb{N}_0. \tag{8}$$

Figure 5 shows the ideal spectral density of bandlimited '$1/f$' noise for $\underline{\omega} = 1$, $\overline{\omega} = 2$. In Figure 6 we present the results obtained from the receding horizon algorithm for $\underline{\omega} = 1$, $\overline{\omega} = 2$, $m = 1$, $N = 10^6$ and $n = 50$. This last Figure verifies the ability of the algorithm to generate a binary non-white noise signal. The discrepancies between the desired and the achieved autocovariances seem to be due to the impossibility of generating a binary signal with a true bandlimited '$1/f$' spectrum, as the results do not appear to improve significantly by increasing m and n.

IV. CONVERGENCE

In this section we study the convergence of the receding horizon algorithm for the special case of generating 'pseudo' white noise, i.e., when the desired autocovariance sequence

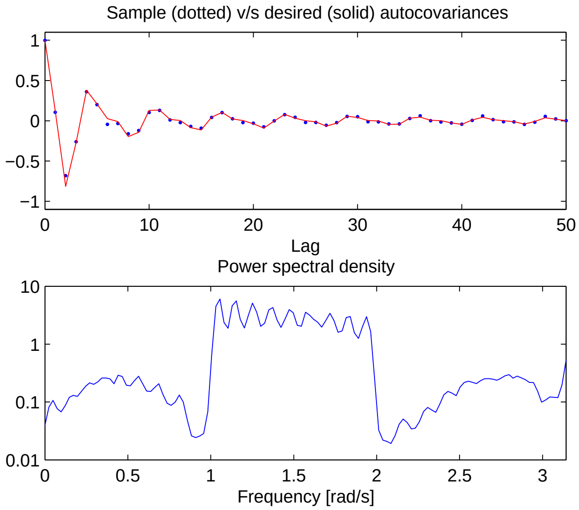

Fig. 6. Characteristics of the generated pseudo bandlimited '$1/f$' noise signal for $m = 1$, $N = 10^6$ and $n = 50$.

is a Kronecker delta ($r_0^d = 1$ and $r_k^d = 0$ for $k \neq 0$). We proceed by describing the algorithm as a switched linear system and then apply a simple geometric inequality to establish its convergence.

A. A switched linear system representation of the algorithm

To aid the development of the switched linear system, let

$$\begin{aligned} \mathbf{x}_t &:= [\mathbf{y}_t^T \quad \mathbf{r}_t^T]^T, \\ \mathbf{y}_t &:= [y_{t-n} \quad \cdots \quad y_{t-1}]^T \in \mathbb{R}^{n\times 1}, \\ \mathbf{r}_t &:= [r_{n,t} \quad \cdots \quad r_{1,t}]^T \in \mathbb{R}^{n\times 1}, \end{aligned} \tag{9}$$

where

$$r_{k,t} := \sum_{i=k+1}^{t-1} (y_i y_{i-k} - r_k^d); \qquad 1 \le k \le \min(t-2, n) \tag{10}$$

and

$$\mathbf{r}^d := [r_n^d \quad \cdots \quad r_1^d]^T \in \mathbb{R}^{n\times 1}. \tag{11}$$

Now,

$$r_{k,t+1} = r_{k,t} + y_t y_{t-k} - r_k^d, \tag{12}$$

allowing the dynamics of $\mathbf{r}_t$ to be given by

$$\mathbf{r}_{t+1} = \mathbf{r}_t + y_t \mathbf{y}_t - \mathbf{r}^d \tag{13}$$

and the initial condition

$$\mathbf{r}_0 = \mathbf{0}_{n,1}. \tag{14}$$

The dynamics of $\mathbf{y}_t$ are given by

$$\mathbf{y}_{t+1} = \begin{bmatrix} \mathbf{0}_{n-1,1} & \mathbf{I}_{n-1} \\ 0 & \mathbf{0}_{1,n-1} \end{bmatrix} \mathbf{y}_t + \begin{bmatrix} \mathbf{0}_{n-1,1} \\ 1 \end{bmatrix} y_t \tag{15}$$

where $\mathbf{I}_{n-1}$ is an identity matrix of order $n-1$, and the initial condition

$$\mathbf{y}_0 = \mathbf{0}_{n,1}. \tag{16}$$

For the generation of pseudo white noise, we have that $\mathbf{r}^d = \mathbf{0}_{n,1}$. This simplifies the expressions, and allows the algorithm to be written as the following switched linear system:

$$\mathbf{x}_{t+1} = \begin{cases} \left[\begin{array}{cc|c} \mathbf{0}_{n-1,1} & \mathbf{I}_{n-1} & \\ 0 & \mathbf{0}_{1,n-1} & \mathbf{0}_{n,n+1} \\ \hline \multicolumn{2}{c|}{-\mathbf{I}_n} & \mathbf{I}_n \end{array}\right] \mathbf{x}_t + \left[\begin{array}{c} \mathbf{0}_{n-1,1} \\ -1 \\ \hline \mathbf{0}_{n,1} \end{array}\right], y_t = -1, \\ \left[\begin{array}{cc|c} \mathbf{0}_{n-1,1} & \mathbf{I}_{n-1} & \\ 0 & \mathbf{0}_{1,n-1} & \mathbf{0}_{n,n+1} \\ \hline \multicolumn{2}{c|}{\mathbf{I}_n} & \mathbf{I}_n \end{array}\right] \mathbf{x}_t + \left[\begin{array}{c} \mathbf{0}_{n-1,1} \\ 1 \\ \hline \mathbf{0}_{n,1} \end{array}\right], \; y_t = 1. \end{cases} \tag{17}$$

Notice that from (17) we have that

$$\mathbf{r}_{t+1} = \mathbf{r}_t \pm \mathbf{y}_t, \tag{18}$$

where the $\pm$ sign is chosen so as to make $\|\mathbf{r}_{t+1}\|_2$ as small as possible.

B. Proof of convergence

The basic idea behind the proof of convergence is to establish a worst case bound for $\|\mathbf{r}_t\|_2$, and to check that according to this bound, $\|\mathbf{r}_t/t\|_2 \to 0$ as $t \to \infty$. Thus, to proceed, we require the following result from functional analysis:

Lemma 1: Let $\mathbf{x}, \mathbf{y}$ be elements of an inner product space $(X, \langle , \rangle)$. Then

$$\min\{\|\mathbf{x}+\mathbf{y}\|^2, \|\mathbf{x}-\mathbf{y}\|^2\} \le \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2, \tag{19}$$

where $\|\mathbf{z}\| := \sqrt{\langle \mathbf{z}, \mathbf{z}\rangle}$ for every $\mathbf{z} \in X$.

Proof: Notice that

$$\|\mathbf{x} \pm \mathbf{y}\|^2 = \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 \pm 2\,\mathrm{Re}\langle \mathbf{x}, \mathbf{y}\rangle, \tag{20}$$

so

$$\begin{aligned} &\min\{\|\mathbf{x}+\mathbf{y}\|^2, \|\mathbf{x}-\mathbf{y}\|^2\} \\ &\qquad = \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - 2|\,\mathrm{Re}\langle \mathbf{x}, \mathbf{y}\rangle| \\ &\qquad \le \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2. \end{aligned} \tag{21}$$

■

The convergence of the algorithm for the special case of generating pseudo white noise is established in the following theorem.

Theorem 1: For the algorithm described by (17), where $\{y_t\}_{t=1}^{\infty} \subseteq \{-1, 1\}^{\mathbb{N}}$ is chosen such that

$$\|\mathbf{r}_t + y_t \mathbf{y}_t\|_2 = \min\{\|\mathbf{r}_t - \mathbf{y}_t\|_2, \|\mathbf{r}_t + \mathbf{y}_t\|_2\}; \qquad t \in \mathbb{N}, \tag{22}$$

it holds that

$$\lim_{t\to\infty} \frac{1}{t} \sum_{i=k+1}^{t} y_i y_{i-k} = r_k^d; \qquad k = 1, \ldots, n. \tag{23}$$

Proof: First notice that

$$\|\mathbf{y}_t\|_2^2 = n; \qquad t > n. \tag{24}$$

Hence, by Lemma 1 and (18) we have that

$$\begin{aligned}\|\mathbf{r}_{t+1}\|_2^2 &= \min\{\|\mathbf{r}_t + \mathbf{y}_t\|_2^2, \|\mathbf{r}_t - \mathbf{y}_t\|_2^2\} \\ &\le \|\mathbf{r}_t\|_2^2 + \|\mathbf{y}_t\|_2^2 \\ &= \|\mathbf{r}_t\|_2^2 + n; \quad t > n.\end{aligned} \tag{25}$$

Since $\mathbf{r}_0 = \mathbf{0}_{n,1}$ (see (14)), we can iterate (25) over $t \in \mathbb{N}$, giving

$$\|\mathbf{r}_t\|_2^2 \le nt + c; \quad t > n, \tag{26}$$

where $c \in \mathbb{R}^+$ is an upper bound on $\sum_{t=1}^n \|\mathbf{r}_t\|_2^2$. Now, by applying the Cauchy-Schwartz inequality to (10) and using the fact that $|y_t| = 1$ and $r_k^d = 0$ we have that $\|\mathbf{r}_t\|_2^2 \le n(t-1)$, so

$$\sum_{t=1}^{n} \|\mathbf{r}_t\|_2^2 \le \sum_{t=1}^{n} n(t-1) = \frac{n^3}{2}, \tag{27}$$

hence we can take $c = n^3/2$. Then, if we divide (26) by t^2 and recall the definition of $\mathbf{r}_t$ (see (10)), we obtain

$$\sum_{k=1}^{n} \left[\sum_{i=k+1}^{t-1} \frac{y_i y_{i-k} - r_k^d}{t} \right]^2 \le \frac{n}{t} + \frac{c}{t^2}; \quad t = n+2, \ldots \tag{28}$$

Therefore,

$$\left| \frac{1}{t} \sum_{i=k+1}^{t-1} (y_i y_{i-k} - r_k^d) \right| \le \sqrt{\frac{n}{t} + \frac{c}{t^2}}; \quad t = n+2, \ldots \tag{29}$$

Since the right side of (29) tends to 0 as $t \to \infty$, we conclude that

$$\begin{aligned}\lim_{t\to\infty} \frac{1}{t} \sum_{i=k+1}^{t} y_i y_{i-k} &= \lim_{t\to\infty} \frac{t+1}{t} \frac{1}{t+1} \sum_{i=k+1}^{t} y_i y_{i-k} \\ &= r_k^d; \quad k = 1, \ldots, n.\end{aligned} \tag{30}$$

■

Theorem 1 establishes that the algorithm generates a binary signal whose sampled autocovariance converges, as t goes to ∞, to the autocovariance of white noise.

V. CONCLUSIONS

In this paper we have presented a novel method for generating binary signals with a specified autocovariance. The algorithm is based on ideas from model predictive control, hence utilizes a receding horizon algorithm. The algorithm is simple and straightforward to implement, and exhibits fast convergence as verified by simulation studies. We have shown empirically that the algorithm has good asymptotic properties, and have been able to establish global convergence for the case of generating pseudo white noise.

REFERENCES

[1] D. Cule and S. Torquato. Generating random media from limited microstructural information via stochastic optimization. *Journal of Applied Physics*, 86(6):3428–3437, September 1999.

[2] G. C. Goodwin, S. F. Graebe, and M. E. Salgado. *Control System Design*. Prentice Hall, Upper Saddle River, New Jersey, 2001.

[3] G. C. Goodwin and R. L. Payne. *Dynamic System Identification: Experiment Design and Data Analysis*. Academic Press, New York, 1977.

[4] G. C. Goodwin, C. R. Rojas, and J. S. Welsh. Good, bad and optimal experiments for identification. In T. Glad, editor, *Forever Ljung in System Identification - Workshop on the occasion of Lennart Ljung's 60th birthday*. September 2006.

[5] U. G. Gujar and R. J. Kavanagh. Generation of random signals with specified probability density functions and power density spectra. *IEEE Transactions on Automatic Control*, 13:716–719, December 1968.

[6] P. S. Koutsourelakis and G. Deodatis. Simulation of binary random fields with applications to two-phase random media. *Journal of Engineering Mechanics*, pages 397–412, April 2005.

[7] B. Liu and D. C. Munson. Generation of a random sequence having a jointly specified marginal distribution and autocovariance. *IEEE Transactions on Accoustics, Speech, and Signal Processing*, 30(6):973–983, December 1982.

[8] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, Upper Saddle River, New Jersey, 2nd edition, 1999.

[9] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 3rd edition, 1991.

[10] D. E. Quevedo, G. C. Goodwin, and J. A. De Doná. Finite constraint set receding horizon quadratic control. *International Journal of Robust and Nonlinear Control*, 13:1–22, 2003.

[11] C. R. Rojas, G. C. Goodwin, J. S. Welsh, and A. Feuer. Robust optimal experiment design for system identification. *Automatica (accepted for publication)*, 2006.

[12] N. Sheehan and S. Torquato. Generating microstructures with specified correlation functions. *Journal of Applied Physics*, 89(1):53–60, January 2001.

[13] A. H. Tan and K. R. Godfrey. The generation of binary and near-binary pseudo-random signals: An overview. In *IEEE Instrumentation and Measurement Technology Conference*, pages 766–771, Budapest, Hungary, May 21-23 2001.

[14] E. Walter and L. Pronzato. *Identification of Parametric Models from Experimental Data*. Springer-Verlag, Berlin, Heidelberg, New York, 1997.

[15] G. L. Wise, A. P. Traganitis, and J. B. Thomas. The effect of a memoryless nonlinearity on the spectrum of a random process. *IEEE Transactions on Information Theory*, 23(1):84–89, January 1977.

[16] C. L. Y. Yeong and S. Torquato. Reconstructing random media. *Physical Review E*, 57(1):495–506, January 1998.

[17] C. L. Y. Yeong and S. Torquato. Reconstructing random media. ii. three-dimensional media from two-dimensional cuts. *Physical Review E*, 58(1):224–233, July 1998.