# Low Power Detection using Stochastic Resonance

Mohammed Nafie and Ahmed H. Tewfik
Dept. of Electrical Engineering
University of Minnesota, Minneapolis, MN 55455
mnafie,tewfik@ece.umn.edu

## Abstract

*Mobile Communications dictates the use of low power detection and estimation algorithms to prolong the battery life. In this paper, we present a very low power detection scheme and evaluate its performance. This scheme can be used instead of the optimal quantized detectors when the noise variance, or other problem parameters, are unknown or change online. Our detector uses an adaptive Discrete Time Stochastic Resonator that consists of a simple Schmitt trigger. Theoretical results evaluating the detection capability of such a device are presented, and simulations show improvement in detection probability over other low power schemes.*

## 1. Introduction

Wireless applications are driving the development of low power algorithms, for the sake of saving the battery power. Here, we propose using a very low power detection algorithm that relies on one-bit quantization of the observations using stochastic resonance.

Stochastic Resonance (SR) was initially used to describe certain physical phenomena, such as the earth's climatic change [1]. Recently, it has been gaining increasing interest as a potential signal processing tool. Despite claims that it outperforms the *optimal* matched filter [2], and results to the contrary showing it *decreases* processing gain [3], there is still a lack in applications that can utilize some appealing characteristics of SR. SR is essentially a nonlinear device which encounters increase in the output SNR with increasing noise [4], and the output SNR reaches a peak, *resonates* at a certain input noise level.

Fig. 1 shows a very simple stochastic resonator, a Schmitt trigger [5]. This device can be modeled as a Markov chain with 2 states, and the probabilities of transition can be easily calculated [6]. In this paper we show how to use a Discrete Time Stochastic Resonator (DTSR) to achieve low power detection. In the next section we introduce the DTSR using a simple Schmitt trigger. We show how to employ such a device in detection problems, and how to calculate its error probability. We also present modified adaptive architectures for the DTSR to enhance its detection capabilities. Next, we show how to employ that DTSR in low power detection schemes. An optimum detection receiver, in the case of Gaussian white noise is the matched filter. The matched filter can be implemented as a single filter, or can implemented as two filters, a decimation filter followed by a downsampler and then a shorter matched filter working at a lower rate. We discuss an approach for replacing the full complexity matched filter by a properly designed adaptive DTSR. To make use of the $1/f$ roll off in the noise spectrum at the output of the DTSR, we also propose a detector that uses ana adaptive DTSR as the decimation filter, following that with a shorter matched filter. We then present the conclusion.

## 2 Discrete Time Stochastic Resonance

### 2.1 Theory

Fig. 1 shows a very simple stochastic resonator, a Schmitt trigger [5]. In this figure, $s(n)$ is the input signal to the DTSR, $g(n)$ is the feedback factor of the Schmitt trigger, and $x(n)$ is the output of the DTSR, all at time instance $n$. This device can be modeled as a 2-state Markov chain, with transition matrix given by:

$$\mathbf{A} = \begin{bmatrix} 1-\alpha(n) & \alpha(n) \\ \beta(n) & 1-\beta(n) \end{bmatrix}$$

where $\alpha(n)$=Prob.$(g(n) + s(n) < 0)$, $\beta(n)$=Prob.$(-g(n) + s(n) > 0)$. In the case of time invariant probability distribution of $s(n)$ and constant feedback factor $g(n) = g$ the state probabilities converge to their steady states which can be easily

0-7803-5148-7/98/$10.00©1998 IEEE

derived [6].

Consider the binary hypotheses:
**H0** :$r(n) = s(n) + w(n)$
**H1** :$r * n) = -s(n) + w(n)$
where $r(n)$ is the received sequence, $s(n)$ is a signal sequence, and $w(n)$ is a white Gaussian noise sequence. Assume that the apriori probabilities of $H0$ and $H1$ are equal, and use a minimum probability of error criterion. Minimizing this probability entails choosing the one of the two hypotheses with the higher likelihood function, that is with the higher Prob.($\mathbf{R}|\mathbf{H}$) where $\mathbf{R}$ is the output sequence from the device.

The output of this device is discrete, and, if the number of observations is $N$, the possible outputs are finite in number and equal to $2^N$. Therefore we can calculate the theoretical error performance of our DTSR by calculating the likelihood of all the possible output sequences of $\pm 1$'s under both hypothesis, and calculating the probabilities of a miss and that of a false alarm.

Using a likelihood function for our decision here is fairly expensive computationally but it presents an upper bound on the performance of our device. Fig. 2 shows the detection probability of a discrete time Schmitt trigger driven using 1 to 18 observations of one of the two previous hypotheses where $s(n) = s = .8$ and $w(n)$ has a standard deviation of 0.5. The feedback factor $g(n)$ is chosen to be a constant and equal to 1. The same figure shows the detection probability of a matched filter. It is clear that the matched filter outperforms the SR using any number of observations, excluding the trivial case of one observation. An advantage of the DTSR is that it uses less power than the matched filter, and hence, in this paper we exploit that characteristic.

### 2.2. Adaptive DTSR

Adapting the parameters of a SR device can enhance the performance by causing the SR to resonate at the operating noise level [7]. Therefore we might expect better results if there was an algorithm to optimally adapt $g(n)$. Such algorithms exist in the theory of Finite Memory detection and decentralized detection[8]. These algorithms require knowledge of the noise variance, and are fairly complex to optimize. Thus, if the noise variance is unknown, or is changing, we cannot apply such algorithms. Here we present a heuristic approach for such adaptation. Fig. 3 shows the detection probability when

$$g(n) = g(0) + \mu \mid \sum_{i=1}^{n-1} y(i) \mid \qquad (1)$$

beginning with $g(0) = 0$. A motivation for this is that if $\mu \mid \sum_{i=1}^{n-1} y(i) \mid$ is high, the probability of one of the two hypothesis is higher than the other, thus we can increase the feedback factor given we started from a low feedback factor. This adaptive scheme conforms with the increase and decrease in the optimal thresholds, when these are specified based on all the previous observations. In cases of known signal amplitude, noise variance and number of observations, we will present techniques for obtaining the optimum $\mu$ later in the paper.

## 3. Optimum One Bit Detection

The results in Fig. 2 were derived assuming a constant feedback factor $g(n) = 1$. If the signal amplitude and the noise variance are known, we can obtain optimal feedback gains. If we keep the feedback gain constant, we get a time invariant one-bit detector. In that case, assume that the optimum $g(n) = M$. Therefore, we, optimally, compare with $-M$ if the previous output is 1 and with $M$ if the previous output is -1, where $M$ can be shown to be greater than zero. $M$ is a function of the signal amplitude, the noise variance and the number of observations.

In Finite Memory detection, the detection decision is based solely on the final quantized output, while in decentralized detection, or general quantized detection, the final decision is allowed to be any function of all the quantized observations. If the number of observation is large, the optimization techniques for obtaining the optimal quantization thresholds at each observation become computationally prohibitive, as these techniques depended on enumerating all the possible outputs. In many of the papers on decentralized detection, simulation results are provided for 2 or 3 observations only. Here, we will present new techniques for obtaining the optimal feedback in several cases of interest. We will relax the condition that the feedback is a multiplicative function of the output bit, and allow it to be a general function. We will also allow the feedback to depend on a specified number of previous outputs instead of just the latest output.

### 3.1. Case 1: DC Signals

In this section, we discuss adapting the feedback value based on the sum of the previous outputs. This is motivated by the fact that, for the sake of lower complexity, our detection decision will be a function of the sum of those outputs. Otherwise, we might need a large look up table, or have to perform some multiplications operations. Notice that if $s(n)$ is not constant, summing the outputs may not yield good results. Therefore, here we limit our attention to only DC signals. We will later show how this can be extended to time varying signals such as sinusoidal signals.

Let our feedback $g(n)$ be $g(n) = \mu \mid \sum_{i=1}^{n-1} y(i) \mid$. We want to obtain the optimum $\mu$ that gives the minimum probability of error. We have the same hypothesis defined before, and we decide **H0** if the final sum of the outputs is positive, and decide **H1** otherwise. We can obtain the optimal $\mu$ by enumerating all the possible outputs for our $N$ observations. We can then calculate the probability of each one of them, given **H0** for instance, using our Markov chain model. We then sum these probabilities for those sequences that will make us decide **H1**. This sum will be the probability of error given **H0**. We repeat for **H1**. Notice that in the case of even $N$, we will have to randomly decide on one of the hypothesis with probability 0.5 when the sum of the outputs is zero. In cases of large $N$, enumerating all the possible $2^N$ outputs might be computationally prohibitive, and so we propose another technique for optimization. We redefine our Markov state to be the current output of the DTSR and the current sum of the all the previous outputs. Since the sum can be any where from $-N$ to $N$, we have $2 * (2N + 1)$ possible states. We can calculate the probability of transition from one state to another, noting that only few transitions are possible: a transition is allowed only from a state to another state if the difference in the value of the sum in the two states is $\pm 1$ *and* the output in the next state corresponds to that difference. We can then sum the probability of the final states that correspond to negative sums to calculate the probability of error given **H0**, and those corresponding to positive sums for the probability of error given **H1**. Fig. 4 shows the optimum $\mu$ with the number of observations at $s = .8$ and $\sigma = .5$. The optimal $\mu$ was evaluated numerically. An interesting observation is that the optimum $\mu$ is zero when the number of observations is odd. We tried several other $s$ and $\sigma$, and this observation still holds true. Unfortunately, a closed form expression for the derivative of the probability of error is difficult to obtain, and hence a rigorous proof of that observation is currently unavailable.

### 3.2. Case 2: General Input Signals

We allow the feedback to be a function of previous $k$ outputs. Using different feedbacks is equivalent to using quantizers with different thresholds. Therefore, our quantizer is

$Q(x) = 1$ if $x > g_i$
$Q(x) = -1$ if $x < g_i$
where $g_i$ is the feedback in state $i, 1 < i < 2^k$, and $x$ is $s + w$.

We base our detection decision either on the final output, or allow it to be a function of all the outputs. If our decision is only based on the final output, this is equivalent to a 1-bit Finite memory detection, and we can calculate the probability of error as the probability of the final state of a 2-state Markov chain being a $-1$ given **H0** added to its probability being a 1 given **H1**. This can be done for both DC and arbitrary signal sequences, with the difference that the transition matrix in the DC case is time invariant, while in the arbitrary signal case, it varies with time. If the detection decision is based on all the outputs, we have 2 cases to consider. In the DC case, we can make this function the sum of all the outputs as we have done before. Here we model our system as a $2^k * (2N + 1)$ state Markov chain. Notice that in the case of Finite memory detection, it can be proven that if you have $k$ bits, the optimal detector is achieved by giving them to the previous observation, i.e. have a $2^k$ state quantizer. But since our objective is low power detection, giving a bit to each of the previous $k$ observations achieves lower power, as you only compare the current observation with a single threshold as opposed to $k$ thresholds in the Finite memory case. Fig. 5 compares between basing our decision on the final output or basing it on all the outputs instead, when $k = 1$.

For arbitrary signal sequences, we can base our decision on the result of a matched filtering operation on the outputs of the DTSR. In that case, to obtain the optimal feedback gain, we have to enumerate all the possible $2^N$ outputs. Again, this might be computationally prohibitive for large $N$. If we want to achieve even lower power, and at the same time be able to optimize, we can decide to sum the outputs. In order to do so, we have to use different quantizers when the signal sequence $s(n)$ changes signs. Therefore, at observation $n$, we will use the quantizer $Q_n$ such that if $s(n) > 0$, $Q_n(x) = 1$ if $x > g_i$, and if if $s(n) < 0$, $Q_n(x) = 1$ if $x < g_i$. Another choice is to use the quantizer $Q_n$ such that if $s(n) > 0$, $Q_n(x) = 1$ if $x > g_i$, and if if $s(n) < 0$, $Q_n(x) = 1$ if $x < -g_i$. Fig. 6 compares between the probability of error obtained using these two techniques, and matched filtering the output of the DTSR. Here $s(n) = .8 * cos(w * n)$, where $n = 1, 2, ..10$, $w = 2 * \pi/10$.

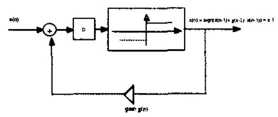

**Figure 1. Schmitt Trigger: A Discrete Time Stochastic Resonator**

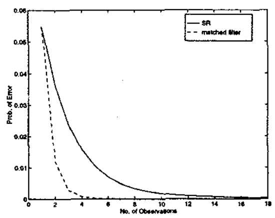

**Figure 2. Error Probability**

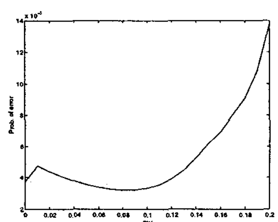

**Figure 3. Probability of error for 10 outputs of an adaptive discrete time Schmitt trigger driven by a constant signal of 0.8 Volts corrupted by a white Gaussian noise sequence of standard deviation 0.5 volt**

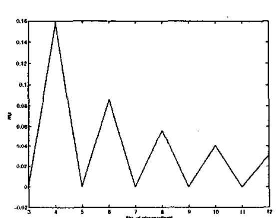

**Figure 4. Optimum Feedback for DC in Gaussian Noise**

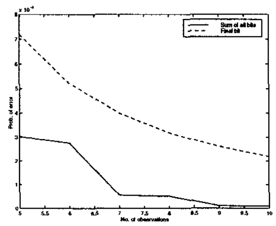

**Figure 5. Probability of Error when decision is based on the final bit or on the sum of the bits**

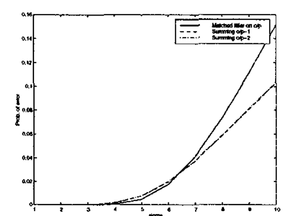

**Figure 6. Probability of Error for a sine wave in noise**

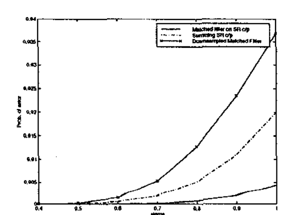

**Figure 7. Probability of Error for a sine wave in noise when downsampling is used**

### 3.3. Low Complexity Detection by Downsampling DTSR Output

Let us now discuss one approach to low complexity detection that relies on the spectral noise shaping that can be achieved with a DTSR. In a normal detection scheme involving signals in white Gaussian noise, assuming we have discrete observations, the complexity of the detection algorithm is essentially that of the matched filter. The number for multiplications and additions of the matched filter is directly proportional to the length of the input sequence, or the number of observations. We may chose to downsample the input sequence to reduce the computational complexity. Saving in multiplications and additions will be proportional to the downsampling rate. To maintain almost the same detection performance level as with no downsampling, we have to use two filters. The first filter is a decimation filter, which limits the bandwidth of the noise, and the second filter is a matched filter with length equal to the length of the data sequence divided by the decimation ratio. Otherwise the *wide band* noise would fold into the band of the signal and hence we get lower SNR and consequently lower detection probability. This decimation filter involves a number of multiplications and additions proportional to its length, and hence we have a trade off between the length of our low pass filter and the performance of our low power detector. An interesting feature of the SR output is that its noise part has a $1/f$ spectrum [5]. Hence, downsampling its output would have a lower effect of folding, or aliasing than the original input which has a white spectrum. Therefore we can save the first filter, the decimation filter and replace it by a DTSR. The DTSR has a very low computational complexity, as for every sample we need only to compute a 1-bit comparison and a 1-bit addition. This leads to a low complexity detection algorithm: preprocess the input signal using the DTSR and then downsample it. We now have a shorter matched filter to be applied to our data. The complexity added by the DTSR preprocessor is low, since for every observation we only need to perform one comparison. An added advantage is that the matched filter now will have inputs of only $\pm 1$, which alleviates the need for multiplication in the matched filter.

We now find the optimal feedback factor $g$, when the detection decision is based on downsampling the output of the DTSR by a factor of $R$. The detection decision will be based on the sum of the observations $mR$, where $m$ is an integer, $1 < m < \frac{N}{R}$. The feedback factor, $g$, will depend on the previous $k$ outputs. Therefore, again, we have a $2^k(2N+1)$ state Markov chain, but we have 2 kinds of the transition matrix: those at observations $mR$, and those at other observations. At observations $mR$, state transitions are allowed to states that differ in the value of the sum, while at other observations state transitions are allowed to only states with the same sum value. Fig. 7 shows the probability of error when $s(n) = .8 * cos(w * n)$, where $n = 1, 2, ..100$, $w = 2 * \pi/100$ and $R = 10$, for three different receivers: matched filtering the downsampled version of $s(n)$, matched filtering the downsampled version of the output of the DTSR, and summing the downsampled version of the output of the DTSR.

## 4. Conclusion

In this paper we presented a low power detection scheme that can be applied to DC and general signals. This technique relies on quantizing and/or downsampling the observations. We also presented computationally feasible techniques to optimize the structures we propose using in low power detection algorithms.

## References


[1] Ronald Fox, "Stochastic Resonance in a double well," Physical review A, Vol. 39, No. 8, pp. 4148-4153, April 1989.

[2] Hu Gnag, "Stochastic Resonance in a nonlinear system driven by an aperiodic force," Physical review A, Vol. 46, No. 6, pp. 3250-3254, Sept. 1992.

[3] T. Albert, A. Bulsara, G. Schmera, and M. Inchiosa, "An evaluation of the Stochastic Resonance Phenomenon as a Potential Tool for Signal Processing," Asilomar Conf., pp. 583-587, 1993.

[4] P. Jung and P. Hanggi, "Amplification of small signals via stochastic Resonance," Physical review A, Vol. 44, No. 12, pp. 8032-8042, Dec. 1991.

[5] B. McNamara and K. Wiesenfeld, "Theory of Stochastic Resonance," Physical review A, Vol. 39, No. 9, pp. 4854-44869, May 1989.

[6] H. Papadopoulos and G. Wornell, "A class of Stochastic Resonance Systems for Signal Processing Applications," Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Atlanta, GA, 1996.

[7] A. Asdi and A. Tewfik, "Detection of weak signals using adaptive Stochastic Resonance," Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Detroit, MI, 1995.

[8] M. Nafie and A. Tewfik, "Low Power Detection," Proc. of the European Signal Processing Conference, Rhodes Island, Greece, Sept. 1998.