IMPROVING THE ENERGY-EFFICIENCY OF A KALMAN FILTER USING UNRELIABLE MEMORIES

Jonathan Kern [1,2], Elsa Dupraz[1], Abdeldjalil Aïssa-El-Bey[1], François Leduc-Primeau[2]

[1] IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-29238, France
[2] Department of Electrical Engineering, École Polytechnique de Montréal, Montreal (QC), Canada

ABSTRACT

Kalman filters are widely used for real-time estimation of dynamic systems, and they sometimes need to be implemented on energy-constrained devices. A Kalman filter implementation from unreliable memories is considered, where the flipping probability of a bit in a memory cell directly depends on its energy consumption. The degradation in estimation performance caused by the noise in the memory is theoretically investigated. Updated equations are then developed for the Kalman filter, taking into account the new source of noise from the unreliable memory. Finally, a method is proposed to optimize the bit energy allocation in the memory, and it is shown from numerical simulations that this method allows for important energy gains.

Index Terms— Kalman filter, Energy-efficiency, Unreliable memory

1. INTRODUCTION

Kalman filtering is a linear quadratic method which uses a set of recursive equations to estimate the state of a variable from noisy measurements. The Kalman filter has been applied in a variety of areas such as tracking and navigation of vehicles [1], estimation of the state of charge of batteries [2], or 3D scene reconstruction in computer vision [3], and a variety of Gaussian [1, 2, 3] and non-Gaussian [4] models have been considered. Kalman filters are sometimes implemented on battery-constrained systems [5, 6]. Therefore, designing a more energy-efficient Kalman filter could improve the estimation performance when dealing with a constrained energy budget.

Kalman filtering can be quite computationally intensive and most existing work focus on reducing its complexity and making it more computationally efficient, see e.g. [7, 8] for FPGA implementations, and [9] for a recent ASIC implementation. Generally speaking, an approach to decrease the energy consumption of a hardware implementation consists of reducing its power supply [10]. But this can create unreliability in the system due to a high level of faults occurring in memories or during computations. Some algorithms such as neural networks [11] or hypothesis testing [12] can inherently tolerate a certain level of faults. But for others these faults cause a performance degradation that needs to be compensated. It was proposed to overcome this loss of performance, either using hardware modifications [13] or algorithmic modifications such as adding redundancy in the system with algorithmic noise tolerance [14]. These strategies have been applied into different applications such as digital signal processing systems [15] and neural networks [16]. To date, Kalman filtering implemented on unreliable hardware has not been studied in the literature.

In this work, we study Kalman filters implemented with an unreliable memory. The focus on memory is motivated by the fact that fetches from memory can consume more than a hundred times more energy than integer operations [17]. Furthermore, unreliable memories were shown to improve the energy efficiency of algorithms such as binary recursive estimation [18] and deep neural networks [19, 20]. These works all rely on the statistical model of [10], which relates the level of noise in the memory cells to the energy consumption of the memory. This statistical model was also used in [21] to optimize the energy consumption of an unreliable memory while satisfying a performance criterion on the Mean-Squared Error (MSE) on a word stored in the memory.

Using the model from [10] and the MSE calculation from [21], we aim to relate the energy consumption of the memory with the performance of the Kalman filter, measured here by the variance of its estimation error. After describing the Kalman filter implemented from an unreliable memory (Section 2), we derive updated recursive equations for the Kalman filter, so as to take into account the memory noise. This also provides us a relation between the estimation error of the Kalman filter and its energy consumption (Section 3). We then use this relation in order to minimize the energy consumption of the filter under the constraint of a desired level of performance. We propose a method to compute the optimal allocation of energy supply across the memory cells so as to minimize the total energy needed by the memory (Section 4). Simulation results show that this method can provide large gains in energy efficiency (Section 5).

This work was supported by grant ANR-17-CE40-0020 and by the Samuel de Champlain program.

2. FAULTY KALMAN FILTER

We first review how to use the Kalman filter in order to estimate dynamic state variables from noisy measurements. We then describe the considered quantization model of the filter and its implementation using an unreliable memory.

2.1. The Kalman filter

Consider a linear dynamic variable $\boldsymbol{x} \in \mathbb{R}^c$ described by a process

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}\boldsymbol{x}_k + \boldsymbol{u}_k \, , \tag{1}$$

where $\boldsymbol{x}_k$ is the state vector of the process at step k, $\boldsymbol{F}$ is the $c \times c$ state transition matrix, and $\boldsymbol{u}_k \in \mathbb{R}^c$ is an additive white noise vector. The state of $\boldsymbol{x}$ is observed through the measurement vector $\boldsymbol{y} \in \mathbb{R}^d$:

$$\boldsymbol{y}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k \, , \tag{2}$$

where $\boldsymbol{H}$ is the $d \times c$ measurement model and $\boldsymbol{v}_k \in \mathbb{R}^d$ is an additive white noise on the measurements, independent from the model noise $\boldsymbol{u}_k$. The covariance matrices of the noise vectors $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$ are known and denoted $\boldsymbol{Q}$ and $\boldsymbol{R}$, respectively.

The Kalman filter recursively estimates the successive states $\boldsymbol{x}_k$ from the measurement vectors $\boldsymbol{y}_k$ and from the knowledge of the model, by minimizing the mean squared error $\mathrm{MSE}(\boldsymbol{x}) = \mathbb{E}[\|\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k\|^2]$ between $\boldsymbol{x}_k$ and its estimate $\hat{\boldsymbol{x}}_k$ at each step k. The filter can be decomposed in two phases: the a priori estimation uses only the knowledge of the model, and the a posteriori estimation takes into account the measurements. Each phases computes both estimates $\hat{\boldsymbol{x}}_{k+1|k}$ (a priori phase) and $\hat{\boldsymbol{x}}_{k+1|k+1}$ (a posteriori phase) of the state vector $\boldsymbol{x}_{k+1}$ and the covariance matrices of the estimations errors $\boldsymbol{P}_{k+1|k} = \mathrm{Cov}[\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k}]$ and $\boldsymbol{P}_{k+1|k+1} = \mathrm{Cov}[\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k+1}]$. The recursive equations of the a priori estimation step are as follows:

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{F}\hat{\boldsymbol{x}}_{k|k} \, , \tag{3}$$

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{F}\boldsymbol{P}_{k|k}\boldsymbol{F}^t + \boldsymbol{Q} \, , \tag{4}$$

and the equations of the a posteriori estimation step are

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1|k}\boldsymbol{H}^t(\boldsymbol{H}\boldsymbol{P}_{k+1|k}\boldsymbol{H}^t + \boldsymbol{R})^{-1} \, , \tag{5}$$

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{K}_{k+1}(\boldsymbol{y}_{k+1} - \boldsymbol{H}\hat{\boldsymbol{x}}_{k+1|k}) \, , \tag{6}$$

$$\boldsymbol{P}_{k+1|k+1} = (\boldsymbol{I} - \boldsymbol{K}_{k+1}\boldsymbol{H})\boldsymbol{P}_{k+1|k} \, . \tag{7}$$

In these equations, the covariance matrices $\boldsymbol{P}$ of size $c \times c$ and the Kalman gain $\boldsymbol{K}$ of size $c \times d$ can be computed in an offline manner. On the opposite, the terms $\hat{\boldsymbol{x}}_{k+1|k}$ and $\hat{\boldsymbol{x}}_{k+1|k+1}$ depend on the measurements $\boldsymbol{y}_k$ and therefore need to be computed online.

2.2. Quantized Kalman filter

In the rest of the article, the implementation of a Kalman filter is studied using a fixed-point quantization model [22]. In this model, each number can be represented as a signed integer coded on $(1+n+m)$ bits, where one bit is used for the sign, n bits are used for the integral part of the number, and m bits are used for its fractional part. A given number z can then be written as $z = (-1)^{z_n} \sum_{b=-m}^{n-1} 2^b z_b$, where $z_b \in \{0, 1\}$ are the bits stored in memory to represent z. In our modeling of the Kalman filter, all the variables (including matrices components) involved in equations (3)-(7) are stored using this quantization model, all with the same value of n and m.

The value of n is chosen so as to be able to represent the biggest possible value in the system. In addition, the value of m sets the resolution of the quantization, meaning that the smallest difference between two quantized numbers is 2^{-m} [22]. Generally speaking, if the resolution of the quantization model is too low, the quantization error may become too large, which could lead to an instability and divergence of the filter [23]. Consequently, it is important to choose a value of m that is large enough to avoid these issues. In our simulations, we choose this value empirically to ensure the convergence of the filter and also to ensure that the quantization error is negligible compared to the total estimation error.

2.3. Unreliable implementation of the filter

In order to reduce its energy consumption, the Kalman filter can be implemented using unreliable hardware. Here, we assume, as in [18], that only the memory is faulty. In this case, each memory cell has a bit flipping probability p. We use the model of [10] to express p with respect to the memory cell energy consumption e as

$$p = \exp(-ea) \, , \tag{8}$$

where a is a parameter that depends on the technology. We assume that the bit errors from each memory cells are independent.

Here we consider that only the estimates $\boldsymbol{x}_{k+1|k}$ and $\boldsymbol{x}_{k+1|k+1}$ are stored in unreliable memory cells. We make this assumption since the other terms of the filter can be precomputed offline and stored on a reliable memory separately in the system. Therefore, in the Kalman filter, instead of having an estimate component $\hat{x}$ we have a possibly incorrect estimate component $\tilde{x}$. Using the binary representation given in Section 2.2, we define an energy per memory cell vector:

$$\boldsymbol{e} = \left[e_{-m}, e_{-m+1}, \cdots, e_{n-1}\right] \, . \tag{9}$$

A bit at position b stored in the unreliable memory can then be expressed as $\tilde{x}_b = \hat{x}_b \oplus \gamma_b$ where $p_b = \Pr(\gamma_b = 1) = \exp(-e_b a)$ and $\oplus$ denotes modulo-2 addition. As the filter would be particularly sensitive to a fault on the sign bit,we consider a sign-preserving model, as in [18]. This can be implemented by storing the sign bits in a reliable memory.

Using the above noise model defined at the bit-level $\tilde{x}_b$, we can define a noise model at the symbol level as $\tilde{x} = \hat{x} + \gamma$, where γ is the noise introduced by the unreliable memory. For the subsequent theoretical analysis, we assume to be in a case where the mean of this memory noise is negligible with respect to its variance $\mathrm{Var}[\gamma] = \sigma_\gamma^2$. This was verified in the simulations. The covariance matrix $\boldsymbol{\Gamma}$ of a memory noise vector $\boldsymbol{\gamma}$ of length c is defined as $\boldsymbol{\Gamma} = \mathrm{Cov}[\boldsymbol{\gamma}] = \boldsymbol{I}_c\sigma_\gamma^2$, and has size $c \times c$. The matrix $\boldsymbol{\Gamma}$ is diagonal since the memory noise variables within each component of quantized vectors are considered independent.

With the error model defined in this section, we see that the amount of errors introduced in the memory depends on the energy supply of the system. In order to optimize the energy consumption of the Kalman filter on an unreliable system, the first step is to analyze how the noise in memory affects the total estimation error of the filter.

3. THEORETICAL ANALYSIS

To analyze how the unreliable memory degrades the estimation performance of the filter, we first compute the variance of the memory noise σ_γ^2. For this, we rely on previous work [21] on the MSE of a faulty computation on unreliable memory to compute the covariance matrix $\boldsymbol{P}^*_{k+1,k+1} = \mathrm{Cov}[\boldsymbol{x}_{k+1} - \tilde{\boldsymbol{x}}_{k+1|k}]$ of the total estimation error. This allows to evaluate how the memory noise γ degrades the performance of the filter and to derive new Kalman filter equations that correct the memory noise.

3.1. Variance of the memory noise

Using the assumption from Section 2.3 where we consider that the bias $\mathbb{E}[\gamma]$ of the unreliable estimation is negligible in front of the variance $\mathrm{Var}[\gamma]$ of the memory noise, this variance can be approximated by the MSE: $\mathbb{E}[(\tilde{x} - \hat{x})^2] \approx \sigma_\gamma^2$. Normally, the value of $\mathbb{E}[(\tilde{x} - \hat{x})^2]$ would depend on two factors: the errors probabilities p_b across the bits and the probability distributions of the variables x which are stored in memory. However, the latter has actually no impact on the final result since, from [21, Claim 17], if $p_{n-1} \ll \frac{1}{2}$ or $\Pr(\hat{x}_b = \hat{x}_{b'}) \simeq \Pr(\hat{x}_b \neq \hat{x}_{b'})$ for any $b \neq b'$, then the MSE $\mathbb{E}[(\tilde{x} - \hat{x})^2]$ can be approximated as

$$\sigma_\gamma^2 = \mathbb{E}[(\tilde{x} - \hat{x})^2] \approx \sum_{b=-m}^{n-1} 4^b p_b = \sum_{b=-m}^{n-1} 4^b e^{-e_b a} , \tag{10}$$

where the last equality is obtained from the noise-vs-energy model (8). Equation (10) gives us a relation between the noise variance σ_γ^2 and the vector $\boldsymbol{e}$ of energy levels defined in (9).

3.2. Variance of the total estimation error

We now show how to recompute the covariance matrix $\boldsymbol{P}_{k+1,k}$ of the estimation error $\tilde{\boldsymbol{e}}_{k+1|k}$ taking into account the noise in the memory. We denote $\tilde{\boldsymbol{e}}_{k+1|k}$ the estimation error on the faulty a priori estimate $\tilde{\boldsymbol{x}}_{k+1|k}$, and we denote $\hat{\boldsymbol{e}}_{k+1|k}$ the estimation on the non faulty estimate $\hat{\boldsymbol{x}}_{k+1|k}$. The a-priori estimation error $\tilde{\boldsymbol{e}}_{k+1|k}$ can then be written as $\tilde{\boldsymbol{e}}_{k+1|k} = \boldsymbol{x}_{k+1} - \tilde{\boldsymbol{x}}_{k+1|k}$ and

$$\begin{aligned}\tilde{\boldsymbol{e}}_{k+1|k} &= \boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{\gamma}_{k+1|k} \\ &= \hat{\boldsymbol{e}}_{k+1|k} + \boldsymbol{\gamma}_{k+1|k} \,. \end{aligned} \tag{11}$$

To simplify the derivation we consider as an approximation that the estimation error inherent from the filter $\hat{\boldsymbol{e}}_{k+1|k}$ and the memory noise $\boldsymbol{\gamma}_{k+1|k}$ are independent. Using this hypothesis, we recompute the new covariance matrix $\boldsymbol{P}^*_{k+1,k} = \mathrm{Cov}[\tilde{\boldsymbol{e}}_{k+1,k}]$ of the estimation error taking into account the memory noise:

$$\begin{aligned}\boldsymbol{P}^*_{k+1,k} &= \mathrm{Cov}[\hat{\boldsymbol{e}}_{k+1,k}] + \mathrm{Cov}[\boldsymbol{\gamma}] \\ &= \boldsymbol{F}\boldsymbol{P}^*_{k,k}\boldsymbol{F}^t + \boldsymbol{Q} + \boldsymbol{\Gamma} \,. \end{aligned} \tag{12}$$

Using the same notations and hypotheses, we apply the same derivation to the a posteriori estimation phase:

$$\tilde{\boldsymbol{e}}_{k+1|k+1} = \boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k+1} + \boldsymbol{\gamma}_{k+1|k+1} \,, \tag{13}$$

and the covariance matrix $\boldsymbol{P}_{k+1|k+1}$ of the a posteriori estimation error $\tilde{\boldsymbol{e}}_{k+1|k+1}$ becomes

$$\boldsymbol{P}^*_{k+1|k+1} = (\boldsymbol{I} - \boldsymbol{K}_{k+1}\boldsymbol{H})\boldsymbol{P}^*_{k+1|k} + \boldsymbol{\Gamma} \,. \tag{14}$$

Finally, we propose to use equations (12) and (14) instead of (4) and (7) for the covariance matrices $\boldsymbol{P}_{k+1|k}$ and $\boldsymbol{P}_{k+1|k+1}$, which allow to compensate for a part of the memory noise. Moreover, due to the noise versus energy model defined in equation (8), the covariance matrix $\boldsymbol{P}^*_{k+1|k+1}$ of the total estimation error can be computed based on the energy levels in the memory. This allows to optimize the energy levels in the memory, as we now describe.

4. OPTIMAL ENERGY ALLOCATION

In this section, we propose to optimize the energy consumption of the memory while satisfying a performance constraint defined on the total estimation error of the filter. More precisely, we consider $\boldsymbol{P}^*_{N|N}$ that is the covariance matrix of the total estimation error at step N, where N is chosen to be large enough so that the filter can converge. We then use this matrix as the performance criterion and build a matrix $\boldsymbol{\mathcal{V}}$ of the same size as $\boldsymbol{P}^*_{N|N}$. The matrix $\boldsymbol{\mathcal{V}}$ defines the desired performance constraint and contains the maximum desired values for the variances and covariances of estimation error on each component. In our optimization, we consider that the parameters m, n, used in the quantization are fixed. The optimization of these parameters is left for future works.

We then introduce the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{e}} \quad e_{\text{tot}} = \sum_{b=-m}^{n-1} e_b = \mathbb{1}^T \boldsymbol{e} \,, \\ &\text{s.t.} \quad \boldsymbol{P}^*_{N|N} \prec \boldsymbol{\mathcal{V}} \text{ and } e_b \geq e_{\text{thres}} \; \forall b \in [\![-m, n-1]\!] \,, \end{aligned} \tag{15}$$

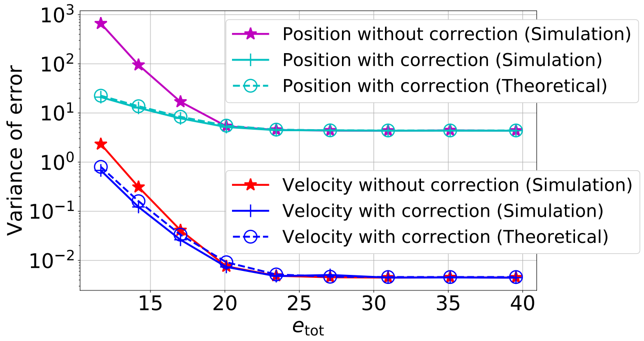

Fig. 1. Comparison between the simulated and theoretical variance of the estimation error at step $N = 250$ using the updated filter equations from Section 3.2 and the conventional equations both on a unreliable system.

where $\prec$ is a component-wise inequality between the two matrices, and where the minimum is taken over all the energy vectors $\boldsymbol{e}$ as defined in (9). The value e_{thres} is the minimum level of energy on each memory cell, so as to avoid undesired effects such as circuit delays and energy leakage [10].

We derive the optimal solution to this problem by using the Karush–Kuhn–Tucker conditions. From these conditions, we show that the optimal energy level e_b^* for bit b has expression:

$$e_b^* = \begin{cases} e_{\text{thres}}, & \text{if } \nu < \frac{1}{4^b a}, \\ \frac{1}{a}\log(4^b a\nu), & \text{otherwise}, \end{cases} \tag{16}$$

where ν is a dual variable that balances the trade-off between reducing the energy consumption and preserving the performance of the system. The optimal vector $\boldsymbol{e}^*$ can then be computed using a water filling algorithm [21] for a fixed desired performance $\boldsymbol{\mathcal{V}}$ of the filter. With this optimal solution, we see that the least significant bits have their energy adjusted to the minimum possible energy level e_{thres}. The energy levels then increase logarithmically for each bit as their importance increase.

5. NUMERICAL RESULTS

In this section, we want to verify the accuracy of the theoretical analysis and also to study the potential gains in energy when using an unreliable memory. In our simulations, we considered a simple tracking problem where the state vector x was composed of two variables representing the position and velocity of an object. Measurements y only consisted of noisy observations of the position of the object. We then measured the variance of the error on the estimation at a step $N = 250$, thus giving enough time for the filter to converge in normal conditions. All variables were quantized using the format presented in Section 2.2 with $n = 11$ bits and $m = 20$ bits. The factor a in (8) was taken as $a = 12.8$ as in [19].

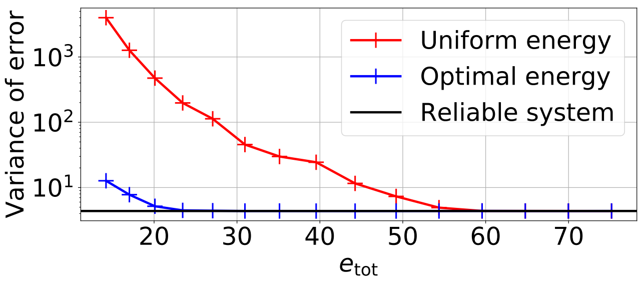

Fig. 2. Comparison of the variance of the total estimation error of the position at step $N = 250$ between using a uniform allocation of the energy across the bits and using the optimal allocation (16).

Figure 1 compares the variance of the estimation error on the position when using the conventional equations of the filter and when using the updated equations. It also verifies the analytical results of Section 3 by comparing the theoretical variance of estimation error with the one measured from simulations. Both curves were obtained with respect to different total energy values e_{tot}. We see that at a low level of energy, using the updated equations allows to better correct the effect of the noise in memory. Moreover, the theoretical curves computed using the updated covariance matrices illustrate that the theoretical derivation of Section 3.2 predicts accurately the variance of the estimation error. To give an order of magnitude, $\mathbb{E}[x_{250}] \approx 250$.

We now study the effect of the optimal energy allocation method proposed in Section 4. We compare the estimation error $\tilde{\boldsymbol{e}}_{250|250}$ after convergence of the filter when using a uniform energy allocation and when using the optimal energy allocation. The results are shown in Figure 2 for different value of energy e_{tot}. In this figure, we can see that in the case of an optimal energy allocation, the system requires less than half the energy of a memory with uniform allocation to achieve the same level of error as a conventional reliable system, which confirms the interest of the proposed approach.

6. CONCLUSION

In this paper we studied the problem of Kalman filtering implemented by using an unreliable memory. We showed how to take the memory noise into account in the filter's equations so as to lower the performance degradation of the filter. We then provided an optimal energy allocation for the bit cells of the memory under a performance constraint. Simulations have shown that using this optimal energy allocation, the filter can achieve the same accuracy as a reliable filter with less than half the energy needed when using a uniform energy allocation.

7. REFERENCES

[1] A. Nadzilah, D. M. Gandana, J. Muliadi, and Y. Daryanto, "Application of Kalman filter to track ship Maneuver," in *5th International Conference on CITSM*, Aug. 2017, pp. 1–5.

[2] Z. Chen, Y. Fu, and C. C. Mi, "State of Charge Estimation of Lithium-Ion Batteries in Electric Drive Vehicles Using Extended Kalman Filtering," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 3, pp. 1020–1030, Mar. 2013.

[3] S. Y. Chen, "Kalman Filter for Robot Vision: A Survey," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4409–4420, Nov. 2012.

[4] Y. Huang, Y. Zhang, N. Li, Z. Wu, and J. A. Chambers, "A Novel Robust Student's t-Based Kalman Filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 3, pp. 1545–1554, June 2017.

[5] X. Lai, T. Yang, Z. Wang, and P. Chen, "IoT Implementation of Kalman Filter to Improve Accuracy of Air Quality Monitoring and Prediction," *Applied Sciences*, vol. 9, no. 9, pp. 1831, Jan. 2019.

[6] G. Anania, A. Tognetti, N. Carbonaro, M. Tesconi, F. Cutolo, G. Zupone, and D. De Rossi, "Development of a novel algorithm for human fall detection using wearable sensors," in *IEEE SENSORS*, Oct. 2008, pp. 1336–1339, ISSN: 1930-0395.

[7] Nurul H. Noordin, Z. Ibrahim, M. H. J. Xie, R. Samad, and N. Hasan, "FPGA Implementation of Simulated Kalman Filter Optimization Algorithm," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 10, no. 1-3, pp. 21–24, Jan. 2018.

[8] A. Jarrah, "Optimized Parallel Architecture of Kalman Filter for Radar Tracking Applications," *Jordan Journal of Electrical Engineering*, vol. 2, no. 3, pp. 215–230, May 2016.

[9] P. T. L. Pereira, G. Paim, P. Ücker, E. Costa, S. Almeida, and S. Bampi, "Exploring Architectural Solutions for an Energy-Efficient Kalman Filter Gain Realization," in *26th IEEE ICECS*, Nov. 2019, pp. 650–653.

[10] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 253–266, Feb. 2010.

[11] J. Vialatte and F. Leduc-Primeau, "A Study of Deep Learning Robustness Against Computation Failures," in *COGNITIVE, The Ninth International Conference on Advanced Cognitive Technologies and Applications*, Feb. 2017, pp. 65–68.

[12] H. Chen, L. R. Varshney, and P. K. Varshney, "Noise-Enhanced Information Systems," *Proceedings of the IEEE*, vol. 102, no. 10, pp. 1607–1621, Oct. 2014.

[13] S. Liu, Y. Wang, M. Fardad, and P. K. Varshney, "A Memristor-Based Optimization Framework for Artificial Intelligence Applications," *IEEE Circuits and Systems Magazine*, vol. 18, no. 1, pp. 29–44, 2018.

[14] R. Hegde and N. R. Shanbhag, "Energy-efficient signal processing via algorithmic noise-tolerance," in *Proceedings. 1999 International Symposium on Low Power Electronics and Design*, Aug. 1999, pp. 30–35.

[15] S. Zhang and N. R. Shanbhag, "Embedded error compensation for energy efficient DSP systems," in *IEEE GlobalSIP*, Dec. 2014, pp. 30–34.

[16] Y. Lin and J. R. Cavallaro, "Energy-efficient Convolutional Neural Networks via Statistical Error Compensated Near Threshold Computing," in *IEEE ISCAS*, May 2018, pp. 1–5, ISSN: 2379-447X.

[17] M. Horowitz, "1.1 Computing's energy problem (and what we can do about it)," in *IEEE ISSCC Digest of Technical Papers*, Feb. 2014, pp. 10–14, ISSN: 2376-8606.

[18] E. Dupraz and L. R. Varshney, "Binary Recursive Estimation on Noisy Hardware," in *IEEE ISIT*, July 2019, pp. 877–881, ISSN: 2157-8117.

[19] G. B. Hacene, F. Leduc-Primeau, A. B. Soussia, V. Gripon, and F. Gagnon, "Training Modern Deep Neural Networks for Memory-Fault Robustness," in *IEEE ISCAS*, May 2019, pp. 1–5.

[20] S. Henwood, F. Leduc-Primeau, and Y. Savaria, "Layerwise Noise Maximisation to Train Low-Energy Deep Neural Networks," in *2nd IEEE International Conference on AICAS*, Aug. 2020, pp. 271–275.

[21] Y. Kim, M. Kang, L. R. Varshney, and N. R. Shanbhag, "Generalized Water-Filling for Source-Aware Energy-Efficient SRAMs," *IEEE Transactions on Communications*, vol. 66, no. 10, pp. 4826–4841, Oct. 2018.

[22] William J. Dally, R. Curtis Harting, and Tor M. Aamodt, *Digital Design Using VHDL: A Systems Approach*, Cambridge University Press, 2015.

[23] M. Verhaegen and P. Van Dooren, "Numerical aspects of different Kalman filter implementations," *IEEE Transactions on Automatic Control*, vol. 31, no. 10, pp. 907–917, Oct. 1986.