

Optimizing the Energy Efficiency of Unreliable Memories for Quantized Kalman Filtering

Jonathan Kern, Elsa Dupraz, Abdeldjalil Aissa El Bey, Lav R Varshney, François Leduc-Primeau

▶ **To cite this version:**

Jonathan Kern, Elsa Dupraz, Abdeldjalil Aissa El Bey, Lav R Varshney, François Leduc-Primeau. Optimizing the Energy Efficiency of Unreliable Memories for Quantized Kalman Filtering. Sensors, 2022, Special Issue Machine Learning, Signal, and/or Image Processing Methods to Enhance Environmental Sensors, 22 (3), pp.853. 10.3390/s22030853 . hal-03540228

HAL Id: hal-03540228
https://imt.hal.science/hal-03540228

Submitted on 23 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

sensors

MDPI

Article

Optimizing the Energy Efficiency of Unreliable Memories for Quantized Kalman Filtering

Jonathan Kern [1,2,*], Elsa Dupraz [1], Abdeldjalil Aïssa-El-Bey [1], Lav R. Varshney [3] and François Leduc-Primeau [2]

[1] IMT Atlantique, Lab-STICC, CNRS UMR 6285, 29238 Brest, France; elsa.dupraz@imt-atlantique.fr (E.D.); abdeldjalil.aissaelbey@imt-atlantique.fr (A.A.-E.-B.)
[2] Department of Electrical Engineering, École Polytechnique de Montréal, Montreal, QC H3T 1J4, Canada; francois.leduc-primeau@polymtl.ca
[3] Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA; varshney@illinois.edu
* Correspondence: jonathan.kern@imt-atlantique.fr

Abstract: This paper presents a quantized Kalman filter implemented using unreliable memories. We consider that both the quantization and the unreliable memories introduce errors in the computations, and we develop an error propagation model that takes into account these two sources of errors. In addition to providing updated Kalman filter equations, the proposed error model accurately predicts the covariance of the estimation error and gives a relation between the performance of the filter and its energy consumption, depending on the noise level in the memories. Then, since memories are responsible for a large part of the energy consumption of embedded systems, optimization methods are introduced to minimize the memory energy consumption under the desired estimation performance of the filter. The first method computes the optimal energy levels allocated to each memory bank individually, and the second one optimizes the energy allocation per groups of memory banks. Simulations show a close match between the theoretical analysis and experimental results. Furthermore, they demonstrate an important reduction in energy consumption of more than 50%.

Keywords: quantized Kalman filter; unreliable memories; energy optimization; error propagation; faulty hardware

check for updates

Citation: Kern, J.; Dupraz, E.; Aïssa-El-Bey, A.; Varshney, L.R.; Leduc-Primeau, F. Optimizing the Energy Efficiency of Unreliable Memories for Quantized Kalman Filtering. *Sensors* **2022**, *22*, 853. https://doi.org/10.3390/s22030853

Academic Editors: Matthieu Puigt, George Tzagkarakis, Gilles Delmaire and Gilles Roussel

Received: 28 October 2021
Accepted: 17 January 2022
Published: 23 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Kalman filtering is a very common recursive estimation task in statistical signal processing [1], and it is often implemented on resource-limited hardware. Applications that require an embedded energy-efficient Kalman filter include air quality monitoring [2], biomedical wearable sensors [3], forest fire detection [4] and vehicle positioning [5]. Energy budgets for embedded systems show that memory access consumes about a hundred-times more energy than integer computations [6]. Therefore, in this paper, we focus on optimizing the energy used by memories in Kalman filters.

All memories used in integrated circuits exhibit a fundamental trade-off between data storage reliability and energy consumption that is related to the inability of perfectly controlling the fabrication process. For example, the energy consumption of static random access memories (SRAMs) can be reduced by lowering its supply voltage; however, this increases the probability that some of the stored bits cannot be retrieved correctly [7]. Following this principle, ref. [8] developed an optimization method to lower the energy consumed by SRAM access by reducing the bit-line voltages. This methodology was also used to decrease the write energy of magnetic random access memories (MRAM) in [9]. In both cases, however, this introduces errors in the words stored in memory.

The robustness to unreliability in computation operations and memories has been investigated for several signal processing and machine-learning applications, including binary

recursive estimation [10], binary linear transformation [11], deep neural networks [12,13], multi-agent systems [14] and distributed logistic regression [15]. Moreover, several techniques have been proposed to compensate for faults introduced by unreliable systems. For instance, [16] proposed to add redundancy in the system through algorithmic noise tolerance, and [17] investigated the use of error-correction codes (ECC) for fault correction.

Although Kalman filtering has not previously been investigated under unreliable hardware implementation, some related works considered this filter and other similar models for linear systems under uncertain conditions. These include errors on the filter's gain [18,19], sensors failures, uncertainties on the observations [20,21], or inaccuracies in the filter parameters [22–24]. In these works, new filter equations were derived using the Riccati equations approach to find new bounds or guarantee on the performance of the filter.

Although these models are not relevant for characterizing the effect of unreliable memories, the main lessons they provide are that Kalman filtering is very sensitive to inaccuracies and that one should re-derive the optimal Kalman filter depending on the specifically considered uncertainty model. On a different line of research, other prior works aim at reducing the energy requirements for Kalman filtering by focusing on reduced computational complexity in field-programmable gate arrays (FGPAs) [25,26] and application-specific integrated circuits (ASICs) [27].

Designing a digital hardware implementation requires quantizing all the variables and computational operations. Therefore, to further reduce the memory energy consumption, one option is to properly optimize the quantization to reduce the memory requirements of the implementation. Significant energy gains from optimized quantization have been demonstrated in [28–30] for signal processing and digital communications applications and in [31–33] for neural networks. The effects of quantization on the Kalman filter were first studied in [34,35] to understand the convergence of filters with reduced precision.

More recently, refs. [36–38] considered two distributed quantized Kalman filters, one based on quantized observations and one based on quantized innovations, where sensors process and transmit quantized observations and innovations to a fusion center. Furthermore, ref. [36] proposed to optimize the number of quantization bits at each sensor to minimize the required data transmission energy.

More general linear stochastic systems were also investigated under quantized measurements [39] and quantized innovations [40], where it was shown that the derived quantized filters converged to standard Kalman filters as the number of quantization levels increased. However, none of these theoretical works considered quantized parameters (e.g., quantized Kalman gain matrices, quantized measurement matrices, etc.), in addition to quantized observations/innovations. Therefore, in this paper, we study a fully-quantized Kalman filter and investigate its energy consumption when using unreliable memories.

Here, we aim to optimize the energy consumption of a Kalman filter implemented with fixed-point quantization [41] and with unreliable memories. Fixed-point representations are often preferred in energy-constrained systems as a fixed-point operation can consume 10-times less energy than a floating-point one [6]. We consider the statistical model of [7], which relates the amount of faults introduced in memory for its energy consumption. Then, as a first contribution, we propose a unified framework to analyze the performance of Kalman filters with both quantization errors and faults introduced in the memory.

To develop this framework, we build on the approach of [35], which consists of evaluating the covariance matrix of the estimation error at each filter iteration by considering both error propagation from previous iterations and errors introduced at the current iteration. Our analysis also includes quantized filter parameters and further incorporates the effect of unreliable memories. Determining the covariance matrix of the estimation error has two advantages. First, it allows us to derive the optimal Kalman filter equations under the considered quantization and memory error models. Second, and more specific to our case, it defines a performance criterion that will be used to optimize the memory energy consumption.

As a second contribution, we define two optimization problems to minimize the memory energy consumption while satisfying a target constraint on the estimation performance of the Kalman filter. In the first problem, we optimize the number B of quantization bits and the energy allocated to each bit position to minimize the overall energy consumption of the memory. This optimization problem extends to [8], which was not dedicated to Kalman filtering but derived optimal bitwise energy allocations with a fixed number of quantization bits by considering a generic Mean-Squared Error (MSE) performance criterion when reading a word in memory.

Although a useful baseline, the setting where each bit position can have a different energy allocation is not practical, since each of the B bits should be placed in a different memory bank with its own power supply. This is why we also introduce a second optimization problem in which we fix the number $L < B$ of possible energy levels and optimize the energy value in each level and the mapping of bit positions to an energy level.

At the price of a small energy increase, this optimization problem allows us to build a practical implementation that only requires L memory banks. By using the Karush–Kuhn–Tucker (KKT) conditions, we provide solutions for the two considered optimization problems. Both solutions can be numerically computed using water-filling. Numerical simulations show that, after optimization, the memory energy consumption is reduced by up to 56% compared to uniform allocation.

The main contributions of this paper can be summarized as:

1. We develop an error propagation model of the Kalman filter that takes different sources of errors (quantization, unreliable memories) into account and allows us to derive new filter equations to minimize the estimation error. Moreover, these equations accurately predict the filter's performance, depending on the considered sources of errors and on their parameters.
2. We propose a methodology for minimizing the energy of the unreliable memories used in the Kalman filter, under a given performance constraint. This methodology consists of computing the optimal number of quantization levels and bit energy allocation in two setups. The first setup considers that the B energy levels can be chosen freely, while the second one assumes that only $L < B$ energy levels can be set.

A preliminary version of this paper [42] only considered optimizing the energy allocation for each memory bank individually and for a fixed number of bits without taking into account the quantization noise nor trying to reduce the energy consumption by adjusting the number of bits. Since the number of quantization bits also affects the energy consumption of the memory, in this work, we add this parameter to the theoretical analysis and the optimization problems.

The rest of the paper is organized as follows. Section 2 describes the quantized Kalman filter and introduces the uncertainty model for unreliable memories. Section 3 investigates the theoretical performance of the filter. Section 4 formally defines and solves the two considered optimization problems. Section 5 presents the simulation results.

2. System Model

We first review the Kalman filter for estimating dynamic state variables from noisy measurements. We then present the considered implementation of the filter by first introducing its quantization model and then describing its implementation with an unreliable memory.

2.1. Kalman Filter

The process:

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

describes the linear dynamic variable $\boldsymbol{x} \in \mathbb{R}^c$, where the state vector of the process at step k is noted as $\boldsymbol{x}_k$, $\boldsymbol{F}$ is the state transition matrix of size $c \times c$, and $\boldsymbol{u}_k \in \mathbb{R}^c$ is an additive

white noise vector [43]. Observation of the state of $\boldsymbol{x}$ can be obtained through $\boldsymbol{y} \in \mathbb{R}^d$, the measurement vector defined as

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

Here, $\boldsymbol{H}$ 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 $\boldsymbol{u}_k$. We denote $\boldsymbol{Q}$ and $\boldsymbol{R}$ as the known covariance matrices of the noise vectors $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$, respectively.

Using the knowledge of the model as well as the the measurement vectors $\boldsymbol{y}_k$, the Kalman filter [1] recursively estimates the successive states $\boldsymbol{x}_k$. This is done by minimizing the mean squared error between $\boldsymbol{x}_k$ and its estimate $\hat{\boldsymbol{x}}_k$ at each step k: $\mathrm{MSE}(\boldsymbol{x}) = \mathbb{E}[\|\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k\|^2]$. The filter can be decomposed in two phases: the a priori estimation uses only the known model, and the a posteriori estimation takes into account the measurements.

At each phase, both the estimates $\hat{\boldsymbol{x}}_{k+1|k}$ (for the a priori phase) and $\hat{\boldsymbol{x}}_{k+1|k+1}$ (for the 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 $\boldsymbol{P}_{k+1|k+1} = \mathrm{Cov}[\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k+1}]$ are computed. The recursive equations of the a priori estimation step are [44]:

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

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

and the recursive equations of the a posteriori estimation step are:

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

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

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

where $\boldsymbol{A}^\top$ denotes the transpose of a matrix $\boldsymbol{A}$. In these equations, the Kalman gain $\boldsymbol{K}$ of size $c \times d$ and the covariance matrices $\boldsymbol{P}$ of size $c \times c$ can be computed offline. On the other hand, the terms $\hat{\boldsymbol{x}}_{k+1|k}$ and $\hat{\boldsymbol{x}}_{k+1|k+1}$ must be computed online as they depend on the measurements $\boldsymbol{y}_k$.

2.2. Quantized Implementation of the Filter

In the rest of the paper, we study Kalman filters that are implemented under fixed-point quantization [41]. Under this model, each number is 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. Using this model, we can write a given number z as

$$z = (-1)^{z_n} \sum_{b=-m}^{n-1} 2^b z_b, \tag{8}$$

where $z_b \in \{0, 1\}$ are the bits stored in memory to represent z. In our modeling of the Kalman filter, all variables (including matrix components) involved in Equations (3)–(7) are stored using this quantization model, all with the same values of n and m. The quantization of the variables to this fixed-point model is done using a uniform quantizer. Note that the distribution of the quantized data is not necessarily uniform (the random variables $\hat{\boldsymbol{x}}_{k|k}$ and $\boldsymbol{y}_k$ could follow Gaussian distributions for example). However, in [45] it is shown that a uniform quantizer can be applied independently of the probability distribution of the source with only a small difference to an optimal quantizer.

In the considered quantizer, the value of n is chosen to be able to represent the largest possible value in the system. The value of m sets the resolution of the quantization so that the smallest difference between two quantized numbers is 2^{-m} [41]. The value of m will be a parameter that is optimized for minimizing the energy in later sections.

In the case of fixed values, such as components of the matrices of the filter, the fixed-point quantized value can be written as $\underline{f} = f + \delta_f$ where δ_f is the quantization error. Using the previously described uniform quantizer, $\delta_f < 2^{-m}$. In the case of quantized random variables, such as the components of $\hat{\boldsymbol{x}}_{\boldsymbol{k}|\boldsymbol{k}}$ or $\boldsymbol{y}_{\boldsymbol{k}}$, we let ϵ_x be the quantization error and express $\underline{x} = x + \epsilon_x$. In [46], conditions are given for the quantization error ϵ_x to be independent from the quantized variable depending on the distribution of the quantized data.

For the special case of a Gaussian distribution, the quantization step needs to be significantly smaller than the variance of the quantized data. In this case, it can be shown that the quantization error is a white noise following a uniform distribution of variance $\frac{2^{-2m}}{12}$. This independence assumption will be used in the theoretical derivations of Section 3. Note that most existing works on quantized Kalman filters only consider that random quantities, such as $\hat{\boldsymbol{x}}_{\boldsymbol{k}|\boldsymbol{k}}$ and $\boldsymbol{y}_{\boldsymbol{k}}$, are quantized, whereas here, the components of the matrices, e.g., $\boldsymbol{K}_k$ of the filter, are also quantized. This will require a new theoretical analysis to treat this case.

2.3. Implementation of the Filter by Using an Unreliable Memory

In order to reduce its energy consumption, the quantized Kalman filter can be implemented on unreliable hardware [8,10–12]. Here, we assume, as in [10,12], that only the memory is faulty. In this case, each memory cell of a memory bank has a bit flipping probability p. We then use the model of [7] to express p with respect to the memory bank energy consumption e as

$$p = \exp(-ea)\,, \tag{9}$$

where a is a parameter that depends on the device technology. We assume that bit errors occur independently. This is justified first by the fact that, in many cases of interest, such as the common case of SRAM memories in a CMOS digital circuit, memory failures can be assumed to occur independently for each bit cell [47]. Therefore, we have a spatial independence between each memory cell for one iteration.

However, typically faults are caused by fabrication variations, and therefore this cannot guarantee a temporal independence for successive reads of the same memory cells. To resolve this issue, we can assume that a diversity scheme is implemented at the system level to avoid re-using the same memory location to store the same variable, which can be implemented at very low cost simply by modifying the memory addressing scheme.

Each memory bank has a uniform energy consumption (e.g., single supply voltage) and is used in our case to store the bits at a certain position of all components of matrices that are stored in the unreliable memory. Since the other terms of the filter can be precomputed offline and stored on a reliable memory separately in the system, we assume that only the estimates $\hat{\boldsymbol{x}}_{k+1|k}$ and $\hat{\boldsymbol{x}}_{k+1|k+1}$ are stored in an unreliable memory bank. Therefore, in the Kalman filter, instead of having an estimate component $\hat{x}$, such as the one computed in (3), we have a possibly incorrect estimate component $\tilde{x}$. We can define an energy per memory bank vector using the binary representation given in (8):

$$\boldsymbol{e} = \left[e_{-m}, e_{-m+1}, \ldots, e_{n-1}\right]. \tag{10}$$

We can then express as $\tilde{x}_b = \hat{x}_b \oplus \gamma_b$ a bit at position b stored in the unreliable memory. Here, $p_b = \Pr(\gamma_b = 1) = \exp(-e_b a)$, and $\oplus$ denotes the modulo-2 addition. As the filter would be particularly sensitive to faults on the sign bit, we consider a sign-preserving model, as in [10,48,49]. This sign-preserving model can be implemented by storing the sign bits in a separate reliable memory.

Using this noise model defined at the bit-level $\tilde{x}_b$, we can define a noise model at the symbol level $\tilde{x}$ as

$$\tilde{x} = \hat{x} + \gamma\,, \tag{11}$$

where γ is the noise introduced by the unreliable memory. For the subsequent theoretical analysis, we assume that the mean $\mathbb{E}[\gamma]$ of this memory noise is negligible compared to its variance $\mathrm{Var}[\gamma] = \sigma_\gamma^2$. We verified this condition with Monte Carlo simulations. The covariance matrix $\mathbf{\Gamma}$ of a memory noise vector $\boldsymbol{\gamma}$ of length c is defined as $\mathbf{\Gamma} = \mathrm{Cov}[\boldsymbol{\gamma}] = \boldsymbol{I}_c\sigma_\gamma^2$, and has size $c \times c$. The matrix $\mathbf{\Gamma}$ is diagonal since the memory noise variables are considered independent.

3. Error Analysis

As described in Section 2, we consider two types of errors affecting the filter: the quantization error and the unreliable memory noise. In this section, we first describe a generic model of error propagation in the Kalman filter, before studying both types of errors in more detail. Finally, we compute the new covariance matrix $\boldsymbol{P}^*_{k|k} = \mathrm{Cov}[\tilde{\boldsymbol{x}}_{k|k} - \boldsymbol{x}_k]$ of the total estimation error by taking both sources of noise (quantization and unreliable memories) into account, compared to a standard Kalman filter, which does not include either.

3.1. Error Propagation Model

Our objective is to compute the total error $\boldsymbol{\Delta\hat{x}_{k+1|k+1}}$ on the computation of $\hat{\boldsymbol{x}}_{k+1}$ at step $k+1$ by considering the two types of errors: quantization and unreliable memory. To handle recursion as in [35], we choose to split the error model in two parts: the errors occurring at step k and the errors from the previous steps, which are propagated up to step k.

To compute $\boldsymbol{\Delta\hat{x}_{k+1|k+1}}$, we first need to express the total error $\boldsymbol{\Delta P}_{k+1|k}$ on the a posteriori covariance matrix $\boldsymbol{P}_{k+1|k}$ after step $k+1$. As in [35], we express this total error as

$$\boldsymbol{\Delta P_{k+1|k}} = f_P(\boldsymbol{\Delta P_{k|k-1}}) + \boldsymbol{\delta P_{k+1|k}}\,, \tag{12}$$

where the function f_P models the errors propagated from step k, and $\boldsymbol{\delta P_{k+1|k}}$ represents the errors occurring at step $k+1$. In this case, according to [35]:

$$f_P(\boldsymbol{\Delta P_{k|k-1}}) = \boldsymbol{G_k \Delta P_{k|k-1} G_k^\top} + o(\boldsymbol{\Delta^2})\,, \tag{13}$$

where $\boldsymbol{\Delta} = \left\| \boldsymbol{H\Delta P_{k|k-1}H^\top} \right\|_2 \ll \sigma_{\min}(\boldsymbol{HP}_{k|k-1}\boldsymbol{H}^\top + \boldsymbol{R})$ and $\boldsymbol{HP}_{k|k-1}\boldsymbol{H}^\top + \boldsymbol{R}$ is a square nonsingular matrix with $\sigma_{\min}$ representing the smallest singular value. Therefore, we have the approximation

$$f_P(\boldsymbol{\Delta P_{k|k-1}}) \approx \boldsymbol{G_k \Delta P_{k|k-1} G_k^\top}\,, \tag{14}$$

where $\boldsymbol{G_k} = \boldsymbol{F}(\boldsymbol{I} - \boldsymbol{K_k H})$. We then express the total error $\boldsymbol{\Delta\hat{x}_{k+1|k+1}}$ on $\boldsymbol{\hat{x}_{k+1|k+1}}$ by considering the same separation between propagation errors and errors from the current iteration. This gives

$$\boldsymbol{\Delta\hat{x}_{k+1|k+1}} = f_x(\boldsymbol{\Delta\hat{x}_{k|k}}, \boldsymbol{\Delta P_{k|k-1}}) + \boldsymbol{\delta\hat{x}_{k+1|k+1}}\,, \tag{15}$$

where the error propagation function f_x is provided in [35], using the same assumption as for (13), as

$$\begin{aligned} f_x(\boldsymbol{\Delta\hat{x}_{k|k}}, \boldsymbol{\Delta P_{k|k-1}}) \approx \\ (\boldsymbol{I} - \boldsymbol{K_k H})(\boldsymbol{F\Delta\hat{x}_{k|k}} + \boldsymbol{\Delta P_{k|k-1}}\boldsymbol{H}^\top(\boldsymbol{HP}_{k,k-1}\boldsymbol{H}^\top + \boldsymbol{R})^{-1}(\boldsymbol{y}_{k+1} - \boldsymbol{HF\hat{x}}_{k,k}))\,. \end{aligned} \tag{16}$$

In this expression, we observe the error propagation from the previous computations of $\hat{\boldsymbol{x}}_{k|k}$ $\boldsymbol{P}_{k+1|k}$. In particular, $\hat{\boldsymbol{x}}_{k+1|k+1}$ depends on $\boldsymbol{K}_{k+1}$, which is precomputed from $\boldsymbol{P}_{k+1|k}$ at each iteration.

Using the recursive Equations (12) and (15), we now estimate the covariance matrix of the total estimation error $\boldsymbol{P^*_{k|k}}$. Note that [35] considered only quantization errors, while here we consider two sources of errors: quantization and unreliable memories. To evaluate $\boldsymbol{P^*_{k|k}}$, we must first compute the covariance of each term of $\boldsymbol{\Delta\hat{x}_{k+1|k+1}}$. By assuming

that the two sources of noise (quantization and unreliable memory noise) are statistically independent, we decompose $\delta\hat{\boldsymbol{x}}_{\boldsymbol{k+1|k+1}}$ as

$$\delta\hat{\boldsymbol{x}}_{\boldsymbol{k+1|k+1}} = \delta\hat{\boldsymbol{x}}^{\mathbf{quant}}_{\boldsymbol{k+1|k+1}} + \delta\hat{\boldsymbol{x}}^{\mathbf{mem}}_{\boldsymbol{k+1|k+1}}\,, \tag{17}$$

and study the two terms $\delta\hat{\boldsymbol{x}}^{\mathbf{quant}}_{\boldsymbol{k+1|k+1}}$ and $\delta\hat{\boldsymbol{x}}^{\mathbf{mem}}_{\boldsymbol{k+1|k+1}}$ separately.

3.2. Quantization Error

We now aim for an analytical expression for $\delta\hat{\boldsymbol{x}}^{\mathbf{quant}}_{\boldsymbol{k+1|k+1}}$, defined as the difference between the full precision estimate $\hat{\boldsymbol{x}}_{k+1|k+1}$ and its quantized version $\underline{\hat{\boldsymbol{x}}}_{k+1|k+1}$:

$$\delta\hat{\boldsymbol{x}}^{\mathbf{quant}}_{\boldsymbol{k+1|k+1}} = \hat{\boldsymbol{x}}_{k+1|k+1} - \underline{\hat{\boldsymbol{x}}}_{k+1|k+1}\,. \tag{18}$$

Before expressing $\delta\hat{\boldsymbol{x}}^{\mathbf{quant}}_{\boldsymbol{k+1|k+1}}$, we first review generic quantization errors expressions [35]. For the scalar fixed-point multiplication of a coefficient $\underline{s}$ with a random variable $\underline{t}$ both quantized according to the model presented in Section 2.2, we can show that

$$\begin{aligned}\underline{st} &= (s+\delta_s)(t+\epsilon_t) + \epsilon_{st}\\ &= st + s\epsilon_t + t\delta_s + \delta_s\epsilon_t + \epsilon_{st}\,,\end{aligned} \tag{19}$$

where $\delta_s = \underline{s} - s$ and ϵ_t and ϵ_{st} follow uniform distributions of variance $\frac{2^{-2m}}{12}$. The scalar expression (19) can then be generalized to the case of a product between a matrix of fixed-point coefficients $\underline{\boldsymbol{A}}$ of size $p \times q$ and a matrix of fixed-point random variables $\underline{\boldsymbol{B}}$ of size $q \times r$ as

$$\underline{\boldsymbol{AB}} = \boldsymbol{AB} + \boldsymbol{A\epsilon_B} + \boldsymbol{B\delta_A} + \boldsymbol{\delta_A\epsilon_B} + \boldsymbol{\epsilon_{AB}}\,, \tag{20}$$

where $\boldsymbol{\epsilon_{AB}}$ is of size $p \times r$ with $\epsilon_{ABi,j} = \sum_{k=1}^{q} \epsilon_{ABi,j,k}$. According to Section 2.2, each $\epsilon_{ABi,j,k}$ follows a uniform distribution of variance $\frac{2^{-2m}}{12}$. In (20), the product $\boldsymbol{\delta_A\epsilon_B}$ can be considered as negligible compared to the other error terms. Indeed, all scalar quantization errors ϵ and δ are upper-bounded by 2^{-m-1}, and since $m \geq 1$, their product is bounded by 2^{-2m-2}. Thus, given that, for a value of m large enough, the value of 2^{-m-1} is much less than 1 and $2^{-2m-2} = (2^{-m-1})^2$, we have that 2^{-2m-2} is negligible compared to 2^{-m-1}. Therefore, in the following derivation, we neglect the products of quantization errors.

We now study quantization errors introduced during the computation of $\underline{\hat{\boldsymbol{x}}}_{k+1|k+1}$. While existing works, e.g., [36–38], assume that only the random quantities $\underline{\hat{\boldsymbol{x}}}_{k|k}$ and $\underline{\boldsymbol{y}}_{k+1}$ are quantized, we here also consider that the matrices $\underline{\boldsymbol{D}}_{k+1}$ and $\underline{\boldsymbol{K}}_{k+1}$ are quantized as well. This corresponds to a more practical implementation setup and requires a more complex theoretical analysis. We first note that Equation (6) can be rewritten as

$$\underline{\hat{\boldsymbol{x}}}_{k+1|k+1} = \underline{\boldsymbol{D}}_{k+1}\underline{\hat{\boldsymbol{x}}}_{k|k} + \underline{\boldsymbol{K}}_{k+1}\underline{\boldsymbol{y}}_{k+1}\,, \tag{21}$$

where both $\boldsymbol{D}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H})\boldsymbol{F}$ and the Kalman gains $\boldsymbol{K}_k$ can be computed offline. We thus consider that the matrices $\boldsymbol{K}_k$ and $\boldsymbol{D}_k$ are computed in full precision and then quantized with a fixed point model. Under these conditions, according to (20) and if we consider that the product of quantization errors is negligible , the quantized vector $\underline{\hat{\boldsymbol{x}}}_{k+1|k+1}$ can be approximated as

$$\begin{aligned}\underline{\hat{\boldsymbol{x}}}_{k+1|k+1} = {} & \boldsymbol{D}_{k+1}\hat{\boldsymbol{x}}_{k|k} + \boldsymbol{\delta_{D_{k+1}}}\hat{\boldsymbol{x}}_{k|k} + \boldsymbol{D}_{k+1}\boldsymbol{\epsilon}_{x_{k|k}}\\ & + \boldsymbol{\epsilon}_{D_{k+1}x_{k|k}} + \boldsymbol{K}_{k+1}\boldsymbol{y}_{k+1} + \boldsymbol{\delta_{K_{k+1}}}\boldsymbol{y}_{k+1}\\ & + \boldsymbol{K}_{k+1}\boldsymbol{\epsilon_{y_{k+1}}} + \boldsymbol{\epsilon}_{\underline{\boldsymbol{K}}_{k+1}\underline{\boldsymbol{y}}_{k+1}} + o(2^{-m-1})\,.\end{aligned} \tag{22}$$

We see that the expression of $\underline{\hat{\boldsymbol{x}}}_{k+1|k+1}$ depends on the full precision vectors $\hat{\boldsymbol{x}}_{k|k}$ and $\boldsymbol{y}_{k+1}$ and on the quantization errors and noise. Finally, the quantization error $\delta\hat{\boldsymbol{x}}^{\mathbf{quant}}_{\boldsymbol{k+1|k+1}}$ defined in (18) can be computed by using (22):

$$\begin{aligned}\delta\hat{\boldsymbol{x}}^{\mathbf{quant}}_{\boldsymbol{k+1|k+1}} \approx\ & \delta_{\boldsymbol{D}_{k+1}}\hat{\boldsymbol{x}}_{k|k} + \boldsymbol{D}_{k+1}\boldsymbol{\epsilon}_{x_{k|k}} + \boldsymbol{\epsilon}_{\boldsymbol{D}_{k+1}\boldsymbol{x}_{k|k}} \\ & + \delta_{\boldsymbol{K}_{k+1}}\boldsymbol{y}_{k+1} + \boldsymbol{K}_{k+1}\boldsymbol{\epsilon}_{\boldsymbol{y}_{k+1}} + \boldsymbol{\epsilon}_{\boldsymbol{K}_{k+1}\boldsymbol{y}_{k+1}},\end{aligned} \tag{23}$$

where the covariance matrix $\boldsymbol{\Sigma}_{\times}$ of $\boldsymbol{\epsilon}_{\times} = \boldsymbol{\epsilon}_{\boldsymbol{D}_{k+1}\boldsymbol{x}_{k|k}} + \boldsymbol{\epsilon}_{\boldsymbol{K}_{k+1}\boldsymbol{y}_{k+1}}$ is given by

$$\boldsymbol{\Sigma}_{\times} = \mathrm{Cov}[\boldsymbol{\epsilon}_{\boldsymbol{D}_{k+1}\boldsymbol{x}_{k|k}}] + \mathrm{Cov}[\boldsymbol{\epsilon}_{\underline{\boldsymbol{K}}_{k+1}\underline{\boldsymbol{y}}_{k+1}}] \tag{24}$$

$$= \mathbf{I}_c(c+d)\frac{2^{-2m}}{12}, \tag{25}$$

and $\mathrm{Cov}[\boldsymbol{\epsilon}_{x_{k|k}}] = \mathbf{I}_c\frac{2^{-2m}}{12}$, $\mathrm{Cov}[\boldsymbol{\epsilon}_{y_{k+1}}] = \mathbf{I}_d\frac{2^{-2m}}{12}$. Equation (23) gives us the quantization error on the computation of $\hat{\boldsymbol{x}}_{\boldsymbol{k+1|k+1}}$ based on the unquantized values of $\hat{\boldsymbol{x}}_{\boldsymbol{k|k}}$, the filter parameters and the quantization resolution m.

3.3. Unreliable Memory Error

We now consider the second source of noise from the unreliable memories and derive an expression for the covariance matrix $\boldsymbol{\Gamma} = \mathrm{Cov}[\delta\hat{\boldsymbol{x}}^{\mathbf{mem}}_{\boldsymbol{k+1|k+1}}]$ of the unreliable memory noise $\delta\hat{\boldsymbol{x}}^{\mathbf{mem}}_{\boldsymbol{k+1|k+1}}$ introduced in (15).

Assuming $\mathbb{E}[\gamma] \ll \mathrm{Var}[\gamma]$, as discussed in Section 2.3, the variance $\mathrm{Var}[\gamma] = \sigma_\gamma^2$ of the memory noise γ can be approximated by the MSE as $\sigma_\gamma^2 \approx \mathbb{E}[(\tilde{x}-\hat{x})^2]$. The value of $\mathbb{E}[(\tilde{x}-\hat{x})^2]$ depends on the error probabilities p_b as well as on the probability distributions of the variables x, which are stored in memory. However, from ([8], 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

$$\sigma_\gamma^2 = \mathbb{E}[(\tilde{x}-\hat{x})^2] \approx \sum_{b=-m}^{n-1} 4^b p_b = \sum_{b=-m}^{n-1} 4^b e^{-e_b a}, \tag{26}$$

where the last equality is obtained from the noise-versus-energy model (9). Therefore, the probability distributions of the variables x have no significant impact on the value of the MSE.

Equation (26) gives us a relation between the noise variance σ_γ^2 and the vector $\boldsymbol{e}$ of energy levels defined in (10). Moreover, by using (26), we show that the covariance $\boldsymbol{\Gamma}$ of the memory noise vector $\delta\hat{\boldsymbol{x}}^{\mathbf{mem}}_{\boldsymbol{k+1|k+1}}$ is given by

$$\boldsymbol{\Gamma} = \boldsymbol{I}_c\sigma_\gamma^2. \tag{27}$$

3.4. Total Error

After separately studying the two error terms $\delta\boldsymbol{x}^{\mathbf{quant}}_{\boldsymbol{k+1|k+1}}$ and $\delta\hat{\boldsymbol{x}}^{\mathbf{mem}}_{\boldsymbol{k+1|k+1}}$, we now combine them to get an expression of the total estimation error $\boldsymbol{e}^{*}_{\boldsymbol{k+1|k+1}} = \tilde{\boldsymbol{x}}_{k+1|k+1} - \boldsymbol{x}_{k+1}$. We then provide the covariance matrix $\boldsymbol{P}^{*}_{k+1|k+1}$ of this total error.

By using $\tilde{\boldsymbol{x}}_{k|k}$ to denote the faulty estimate of $\boldsymbol{x}_k$, we can express

$$\tilde{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k} + \boldsymbol{\Delta\tilde{x}}_{\boldsymbol{k|k}}. \tag{28}$$

Considering that only the $\hat{\boldsymbol{x}}_{k|k}$ are stored in the unreliable memories, the error propagation model (15) can be rewritten as

$$\boldsymbol{\Delta\tilde{x}}_{\boldsymbol{k+1|k+1}} = \boldsymbol{D}_{k+1}\boldsymbol{\Delta\tilde{x}}_{\boldsymbol{k|k}} + \delta\hat{\boldsymbol{x}}^{\mathbf{quant}}_{\boldsymbol{k+1|k+1}} + \delta\hat{\boldsymbol{x}}^{\mathbf{mem}}_{\boldsymbol{k+1|k+1}} \tag{29}$$

$$
\begin{aligned}
&= \boldsymbol{D}_{k+1}\boldsymbol{\Delta\tilde{x}_{k|k}} + \boldsymbol{\delta}_{\boldsymbol{D}_{k+1}}\tilde{\boldsymbol{x}}_{k|k} + \boldsymbol{D}_{k+1}\boldsymbol{\epsilon}_{x_{k|k}} \\
&\quad + \boldsymbol{\delta}_{\boldsymbol{K}_{k+1}}\boldsymbol{y}_{k+1} + \boldsymbol{K}_{k+1}\boldsymbol{\epsilon}_{y_{k+1}} \\
&\quad + \boldsymbol{\epsilon}_{\times} + \boldsymbol{\delta\hat{x}^{mem}_{k+1|k+1}} \qquad (30) \\
&= (\boldsymbol{D}_{k+1} + \boldsymbol{\delta}_{\boldsymbol{D}_{k+1}})\boldsymbol{\Delta\tilde{x}_{k|k}} + \boldsymbol{\delta}_{\boldsymbol{D}_{k+1}}\hat{\boldsymbol{x}}_{k|k} \\
&\quad + \boldsymbol{D}_{k+1}\boldsymbol{\epsilon}_{x_{k|k}} + \boldsymbol{\delta}_{\boldsymbol{K}_{k+1}}\boldsymbol{y}_{k+1} + \boldsymbol{K}_{k+1}\boldsymbol{\epsilon}_{y_{k+1}} \\
&\quad + \boldsymbol{\epsilon}_{\times} + \boldsymbol{\delta\hat{x}^{mem}_{k+1|k+1}}, \qquad (31)
\end{aligned}
$$

where (30) is obtained by replacing $\boldsymbol{\delta\hat{x}^{quant}_{k+1|k+1}}$ by its expression (23), and (31) comes from (28), which allows u to write $\tilde{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k} + \boldsymbol{\Delta\tilde{x}_{k|k}}$. Equation (31) provides a recursive form of the error at step k, since $\boldsymbol{\Delta\tilde{x}_{k+1|k+1}}$ depends on $\boldsymbol{\Delta\tilde{x}_{k|k}}$ and $\hat{\boldsymbol{x}}_{k|k}$. All the other terms in (31) come from the current iteration $k+1$.

In certain conditions, such as if $\boldsymbol{H}$ and $\boldsymbol{F}$ are only composed of integer components, the total estimation error $\tilde{\boldsymbol{x}}_{k+1|k+1} - \boldsymbol{x}_{k+1} = \hat{\boldsymbol{x}}_{k+1|k+1} + \boldsymbol{\Delta\tilde{x}_{k+1|k+1}} - \boldsymbol{x}_{k+1}$ can be further developed as (see Appendix A for more details):

$$
\begin{aligned}
\tilde{\boldsymbol{x}}_{k+1|k+1} - \boldsymbol{x}_{k+1} &= (\boldsymbol{D}_{k+1} + \boldsymbol{\delta}_{\boldsymbol{D}_{k+1}})(\tilde{\boldsymbol{x}}_{k|k} - \boldsymbol{x}_k) && (32) \\
&\quad + (\boldsymbol{K}_{k+1} + \boldsymbol{\delta}_{\boldsymbol{K}_{k+1}})\boldsymbol{v}_{k+1} + ((\boldsymbol{K}_{k+1} + \boldsymbol{\delta}_{\boldsymbol{K}_{k+1}})\boldsymbol{H} - \mathbf{I})\boldsymbol{u}_k && (33) \\
&\quad + \boldsymbol{D}_{k+1}\boldsymbol{\epsilon}_{x_{k|k}} + \boldsymbol{K}_{k+1}\boldsymbol{\epsilon}_{y_{k+1}} + \boldsymbol{\epsilon}_{\times} + \boldsymbol{\delta\hat{x}^{mem}_{k+1|k+1}}. && (34)
\end{aligned}
$$

This equation gives us a recursive form of the total estimation error $\tilde{\boldsymbol{x}}_{k+1|k+1} - \boldsymbol{x}_{k+1}$ at step $k+1$, depending on the estimation error $\tilde{\boldsymbol{x}}_{k|k} - \boldsymbol{x}_k$ at step k and on the quantization resolution 2^{-m} and the memory noise $\boldsymbol{\delta\hat{x}^{mem}_{k+1|k+1}}$.

Finally, we can compute the covariance matrix $\boldsymbol{P}^*_{k+1|k+1}$ of this error as

$$
\begin{aligned}
&\boldsymbol{P}^*_{k+1|k+1} = \mathrm{Cov}[\tilde{\boldsymbol{x}}_{k+1|k+1} - \boldsymbol{x}_{k+1}] \\
&= (\boldsymbol{D}_{k+1} + \boldsymbol{\delta}_{\boldsymbol{D}_{k+1}})\boldsymbol{P}^*_{k|k}(\boldsymbol{D}_{k+1} + \boldsymbol{\delta}_{\boldsymbol{D}_{k+1}})^\top \\
&+ (\boldsymbol{K}_{k+1} + \boldsymbol{\delta}_{\boldsymbol{K}_{k+1}})\boldsymbol{R}(\boldsymbol{K}_{k+1} + \boldsymbol{\delta}_{\boldsymbol{K}_{k+1}})^\top \\
&+ ((\boldsymbol{K}_{k+1} + \boldsymbol{\delta}_{\boldsymbol{K}_{k+1}})\boldsymbol{H} - \mathbf{I})\boldsymbol{Q}((\boldsymbol{K}_{k+1} + \boldsymbol{\delta}_{\boldsymbol{K}_{k+1}})\boldsymbol{H} - \mathbf{I})^\top \\
&+ \boldsymbol{D}_{k+1}\,\mathrm{Cov}[\boldsymbol{\epsilon}_{x_{k|k}}]\boldsymbol{D}^\top_{k+1} + \boldsymbol{K}_{k+1}\,\mathrm{Cov}[\boldsymbol{\epsilon}_{y_{k+1}}]\boldsymbol{K}^\top_{k+1} + \boldsymbol{\Sigma}_{\times} + \boldsymbol{\Gamma}, \qquad (35)
\end{aligned}
$$

where all the terms involved, including the covariance matrices, have been explicited in the previous sections. Equation (35) shows that the covariance matrix $\boldsymbol{P}^*_{k+1|k+1}$ can be computed recursively.

Equation (35) provides us a measure of the performance of the filter, depending on the quantization resolution and on the energy supplied to the memory. Equipped with this derivation, we can now use the covariance matrix $\boldsymbol{P}^*_{k+1|k+1}$ as a performance criterion against which to optimize the energy consumed of the unreliable memory.

4. Energy Optimization

In this section, we optimize the energy consumption of the memory while satisfying a performance constraint defined on the total estimation error of the filter. As parameters to optimize, we consider the number of bits m for the quantization, and the energy vector $\boldsymbol{e}$ of the memory banks. We define two optimization problems, which both seek to minimize the energy consumed by the memory. In the first problem, we find the optimal number of bits m and the corresponding $n+m$ levels of energy to allocate to the memory banks. Although solving this problem provides the minimum energy that needs to be supplied to the memory, it is not very practical since each of the $n+m$ bits should be stored in a different memory bank with a specific voltage supply.

Therefore, in the second problem, we consider that the number of bits m is fixed but that the number of possible energy levels is limited to L possibilities. Both the L energy values and the allocation of each bit to one of the L possible values should be optimized. Solving this problem allows to consider only $L < n + m$ different memory banks.

4.1. Optimization across All the Bits

We first find the optimal level of energy e_b of each memory bank and the optimal number of fractional bits m to minimize the total memory energy consumption. As performance criterion, we consider the covariance matrix $\boldsymbol{P}^*_{N|N}$ of the total estimation error at step N, where N is chosen to be large enough so that the filter can converge. We further introduce a matrix $\boldsymbol{\mathcal{V}}$ of the same size as $\boldsymbol{P}^*_{N|N}$ to define the performance constraint for the variances and covariances of estimation error on each component. The optimization problem is then defined as follows:

$$\begin{aligned} \min_{\mathbf{e},m} \quad & e_{\text{tot}} = \sum_{b=-m}^{n-1} e_b = \mathbb{1}^\top \boldsymbol{e}\,, \\ \text{s.t.} \quad & \boldsymbol{P}^*_{N|N} \prec \boldsymbol{\mathcal{V}} \text{ and } e_b \geq e_{\text{thres}}\ \forall b \in [\![-m, n-1]\!]\,, \end{aligned} \tag{36}$$

where $\prec$ is a component-wise inequality between the two matrices and where the minimum is taken over all energy vectors $\boldsymbol{e}$ as defined in (10) and for all the possible values of the number of bits m. We consider that $m \in [\![0, M]\!]$, where M is the maximum number of bits, which could be stored in a memory. The value e_{thres} is the minimum level of energy for each memory bank to avoid undesired effects, such as circuit delays and energy leakage [7].

Problem (36) involves one discrete parameter m and $m + n$ continuous parameters $\boldsymbol{e}$, which makes it difficult to solve at once. As a first step, we assume that the value of m is fixed and solve the following simplified problem:

$$\begin{aligned} \min_{\mathbf{e}} \quad & e_{\text{tot}} = \sum_{b=-m}^{n-1} e_b = \mathbb{1}^\top \boldsymbol{e}\,, \\ \text{s.t.} \quad & \boldsymbol{P}^*_{N|N} \prec \boldsymbol{\mathcal{V}} \quad \text{and} \quad e_b \geq e_{\text{thres}}\ \forall b \in [\![-m, n-1]\!]\,, \end{aligned} \tag{37}$$

by using the Karush–Kuhn–Tucker (KKT) conditions (see Appendix B). From these conditions, we show that the optimal energy level e^*_b for bit b has expression

$$e^*_b = \begin{cases} e_{\text{thres}}, & \text{if } \lambda < \frac{1}{4^b a}\,, \\ \frac{1}{a}\log(4^b a \lambda), & \text{otherwise}\,, \end{cases} \tag{38}$$

where λ is a dual variable. It allows balancing a trade-off between preserving the performance of the system and reducing the energy consumption. A water-filling algorithm [8] can be used to compute the optimal vector $\boldsymbol{e}^*$ for a fixed desired performance $\boldsymbol{\mathcal{V}}$ of the filter. We can observe that, according to this optimal solution, the energy of the least significant bits will be set to the threshold energy level e_{thres}. The energy levels then increase logarithmically for each bit as their significance increases.

Since m is discrete, the optimal solution (38) is computed using the water-filling algorithm for each possible value of m. We then retain the solution $(m^\star, \boldsymbol{e}^\star)$, which gives the lowest total energy $e^\star_{\text{tot}} = \sum_{b=-m^\star}^{n} e^\star_b$. In this method, the influence of the quantization error is taken into account through the performance criterion $\boldsymbol{P}^*_{N|N}$. For a small number of bits m, quantization errors may make it impossible to satisfy the desired performance constraint, and therefore the water-filling algorithm will not be able to find an optimal solution. In this case, if we detect that the algorithm converges toward a performance value that is still higher than the constraint, the algorithm is stopped, and we proceed to the next value of m in the considered range.

The full optimization process is summarized in Algorithm 1. In this algorithm, the parameter β controls the rate at which the energy for each memory bank is increased at each

iteration. The value of β is chosen either using the precision with which energy can be set in a given device technology, or based on the desired rate of convergence for the water-filling algorithm.

The condition $\left\| \boldsymbol{P}_{\text{prev}} - \boldsymbol{P}^*_{N|N} \right\| > \xi$ is used to detect whether the water-filling algorithm has a feasible solution, and thus the value of ξ is set to be low. The computation of $\boldsymbol{P}^*_{N|N}(\boldsymbol{e}, m)$ accounts for most of the computing time of this algorithm. The total run time thus depends on the number of iterations required by the water-filling algorithm (while loop in Algorithm 1). For fixed values of β and $\boldsymbol{\mathcal{V}}$, we expect the number of iterations to increase with m.

Algorithm 1: Computing the optimal values for $\boldsymbol{e}$ and m.

Input: $\boldsymbol{\mathcal{V}}$, a, β, ξ, e_{thres} ;
$\boldsymbol{e}_{\min} \leftarrow +\infty$;
for *each value of* m **do**
  $\boldsymbol{e} \leftarrow e_{\text{thres}}$;
  $\boldsymbol{P}_{\text{prev}} \leftarrow 0$;
  while $\boldsymbol{P}^*_{N|N}(\boldsymbol{e}, m) \succ \boldsymbol{\mathcal{V}}$ *and* $\left\| \boldsymbol{P}_{prev} - \boldsymbol{P}^*_{N|N} \right\| > \xi$ **do**
    $\boldsymbol{P}_{\text{prev}} \leftarrow \boldsymbol{P}^*_{N|N}(\boldsymbol{e}, m)$;
    $b \leftarrow \arg\min_{b} \{\log(\frac{1}{4^b a}) + ea\}$;
    $e_b \leftarrow e_b + \beta$;
  end
  if $\boldsymbol{P}^*_{N|N}(\boldsymbol{e}, m) \prec \boldsymbol{\mathcal{V}}$ *and* $\sum_{b=-m_{opti}}^{n} e_{min_b} > \sum_{b=-m}^{n} e_b$ **then**
    $\boldsymbol{e}_{\min} \leftarrow \boldsymbol{e}$;
    $m_{\text{opti}} \leftarrow m$;
  end
end
Result: Optimal number of bits m_{opti} and optimal energy allocation vector $\boldsymbol{e}_{\min}$

4.2. Optimization with a Limited Number of Energy Levels

In practice, the solution of Problem 1 makes the implementation costly as each bit position should be stored in a separate memory bank. Therefore, we define a second optimization problem with only $L < m + n$ possible levels of energy. For implementation purposes, we only consider small values for L ($L < 10$). The vector $\boldsymbol{f} = [f_0, \ldots, f_{L-1}]$ contains the L levels of energy. We use n_ℓ to denote the number of bits allocated to energy level f_l, so that $\sum_{\ell=0}^{L-1} n_\ell = n + m$. This means that each memory bank of the energy group ℓ has an energy level $e_b = \frac{f_\ell}{n_\ell}$. We write $\boldsymbol{n} = [n_0, \ldots, n_{L-1}]$ for the vector containing the L values n_ℓ.

In the following, for simplicity, we consider that the number of bits n and m are fixed, and we seek to optimize the total energy consumption of the unreliable memory for a fixed number of energy levels L. The objective is to reduce the total energy consumed by the unreliable memory by allocating different levels of energy to the L groups of bits. Two parameters are considered in this optimization: the values of each energy level e_ℓ and the number of bits allocated to each of these energy levels n_ℓ. The optimization problem can be written as

$$\begin{aligned} \min_{\boldsymbol{f},\boldsymbol{n}} \quad & e_{\text{tot}} = \sum_{l=0}^{L-1} f_\ell = \mathbb{1}^\top \boldsymbol{f}\,, \\ \text{s.t.} \quad & \boldsymbol{P}^*_{N|N} \prec \boldsymbol{\mathcal{V}} \quad \text{and} \qquad f_\ell \geq e_{\text{thres}} n_\ell,\ n_\ell \in [\![1, n+m-L]\!]\ \forall \ell \in [\![0, L-1]\!]\,. \end{aligned} \tag{39}$$

First, we solve the optimization problem in the case where we know which bit is allocated to which energy level. This means that the values of n_ℓ are known and that we only want to compute the optimal values of the energy levels f_ℓ. In this case, the optimization problem can be written as

$$\begin{aligned} \min_{\boldsymbol{f}} \quad & e_{\text{tot}} = \sum_{l=0}^{L-1} f_\ell = \mathbb{1}^\top \boldsymbol{f}\,, \\ \text{s.t.} \quad & \boldsymbol{P}^*_{N|N} \prec \boldsymbol{\mathcal{V}} \quad \text{and} \quad f_\ell \geq e_{\text{thres}} n_\ell \quad \forall \ell \in [\![0, L-1]\!]\,. \end{aligned} \tag{40}$$

Problem (40) is quite similar to the one described in Section 4.1 and can be solved using the same method as the one presented in Appendix B, by relying on the KKT conditions. The optimal solution in this case is

$$f_l^* = \begin{cases} e_{\text{thres}} n_\ell, & \text{if } \lambda < \frac{1}{\sum_{b=0}^{n_\ell} 4^b a}\,, \\ \frac{1}{a} \log(\sum_{b=0}^{n_\ell} 4^b a \lambda), & \text{otherwise}\,. \end{cases} \tag{41}$$

This solution allows us to compute the optimal energy levels f_ℓ for a given energy allocation across the bits. The second step consists of computing the best allocation of bits to each energy group. Given that we only consider small values of L, we compute the optimal solution from (41) for each possible energy allocation of the bits. Then, the solution with the smallest total energy $\sum_{\ell=0}^{L-1} f_\ell$ is retained. Although Problem 2 leads to a more practical solution, it is expected that the optimal total energy of the memory is higher for Problem 2 than for Problem 1.

5. Simulation Results

In our simulations, unless explicitly stated, we consider a simple tracking problem where the state vector $\boldsymbol{x}$ is composed of two variables representing the position and velocity of an object. Measurements y only consist of noisy observations of the position of the object. The process matrix $\boldsymbol{F}$ and measurement matrix $\boldsymbol{H}$ are defined as

$$\boldsymbol{F} = \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix}, \boldsymbol{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \tag{42}$$

and the process noise covariance matrix $\boldsymbol{Q}$ and measurement covariance matrix $\boldsymbol{R}$ are given by

$$\boldsymbol{Q} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_x^2 \end{bmatrix}, \boldsymbol{R} = \begin{bmatrix} \sigma_y^2 \end{bmatrix}. \tag{43}$$

where $\delta t = 1$ and $\sigma_x = 0.01$ and $\sigma_y = 10$. The factor a in (9) is taken as $a = 12.8$ as in [13]. This section is divided into two parts. We first evaluate the accuracy of the proposed theoretical analysis, and we then provide solutions to the two considered optimization problems.

5.1. Accuracy of the Theoretical Analysis

First, to evaluate the accuracy of the proposed theoretical analysis, we perform Monte Carlo simulations ($N_{mc} = 10^7$) and measure the covariance matrix of the error on the estimation at step $N = 250$, thus giving enough time for the filter to converge in normal conditions. This covariance matrix is compared with the theoretical expression of the covariance of the estimation error $\boldsymbol{P}^*_{N|N}$ computed in Section 3.4. Figure 1 shows the variance of the estimation errors on the position and the velocity for different values of m in the case of a reliable memory, meaning that we consider only the quantization error and not the memory noise.

We observe that the theoretical predictions of the errors closely match the simulations, which shows the accuracy of our theoretical analysis. From Figure 1, we can observe that, for a small number of bits m, the quantization error is large and dominates the total estimation error. However, starting from $m = 10$ bits, the estimation error reaches a

constant level, which can be interpreted as a minimum bound on the estimation error that one can obtain from a standard Kalman filter for this tracking problem. This shows that, from $m = 10$ bits, the quantization errors become negligible compared to the estimation error achieved by the standard full precision Kalman filter. Thus, at this point, using more bits will not result in minimizing the estimation error, justifying the need for optimizing the parameter m.

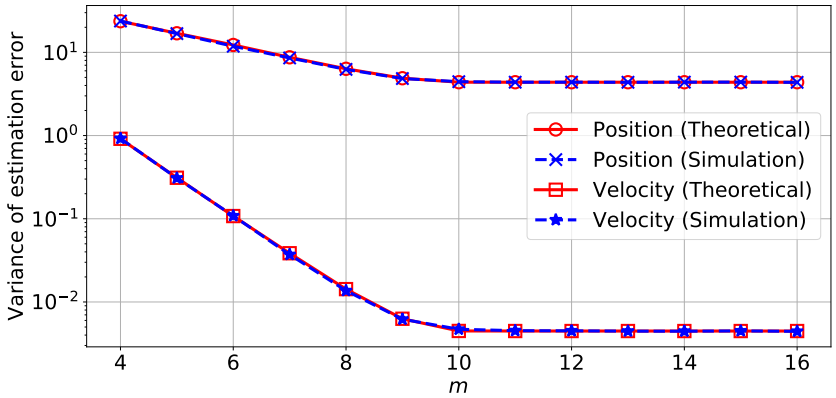


Figure 1. Theoretical and simulated variance of estimation error on the position and velocity depending on the numbers of bits in the representation, using a reliable memory.

In a second step, we introduce the memory noise in addition to the quantization. Figure 2 shows the variance of the estimation error on the position depending on the total energy e_{tot} for different values of m. The variance values were obtained from both Monte Carlo simulations and from the theoretical analysis of Section 3.4. Further, Figure 3 shows the variance of estimation error on the position depending on the number of bits m for different values of the total energy e_{tot}. The comparison between theoretical results and Monte Carlo simulations show the accuracy of the theoretical analysis that predicts the new computed covariance $\boldsymbol{P}^*_{k|k}$.

From Figure 2, we can also see that both the number of bits and the total energy can affect the variance of the estimation error. If the number of bits or the energy supplied is too low, then the quantization error or the memory noise will dominate the total estimation error. However, we can see that there is a minimum number of bits, around $m = 12$, from which, given enough energy, it will be possible to reach the minimum possible variance of estimation error.

Moreover from Figure 3, we can see that for a low value of supplied energy per variable, the variance of the estimation error will increase with the number of bits as there is too little energy. However, for a larger amount of energy $e_{\text{tot}} > 10$, the variance of the estimation error will decrease with the number of bits since the quantization error decreases. Finally for a large enough number of bits, the total estimation error will only depend on the total energy and on the variance of the estimation error of a reliable full precision filter.

As the work presented in this paper was conducted to reduce the energy consumption of the memory of a Kalman filter, it is of a greater utility when the memory is large. For this reason, the results presented before were also tested on a larger Kalman filter with a dimension for the state vector x of $c = 20$. For simulations on this large-size example, we use a state transition model that performs a shifting of the entries of the state to the next state at each iteration, such as the one used in [50]. That is,

$$\boldsymbol{F}_{i,j} = \begin{cases} 1, & \text{if } i - j = 1\,, \\ 0, & \text{otherwise}\,, \end{cases} \tag{44}$$

and $\boldsymbol{F}_{c,1} = 1$. The initial state vector is drawn from a normal distribution.

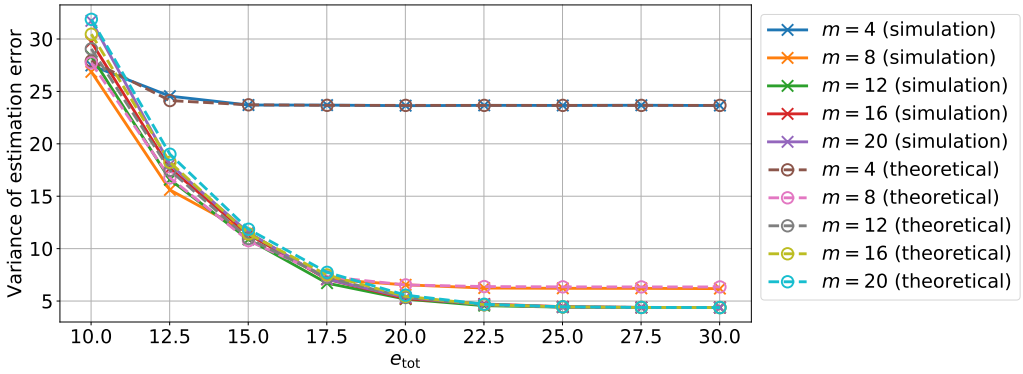


Figure 2. Theoretical and simulated variance of estimation error on the position depending on the energy supplied to each variable using an unreliable memory for different numbers of bits in the representation m.

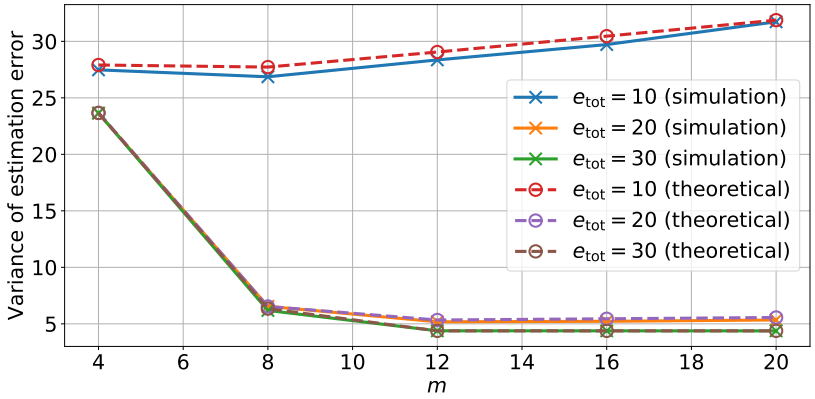


Figure 3. Theoretical and simulated variance of estimation error on the position depending on the number of quantization bits using an unreliable memory for different for different total energy values e_{tot}.

In this case, the performance of the filter is measured by the trace of the covariance matrix $\boldsymbol{P}_{N|N}$. The results in Figures 4 and 5 show that the same conclusion can be taken from the simulations done on the small-size Kalman filter and that the method presented in this paper can therefore be applied to large-size filters.

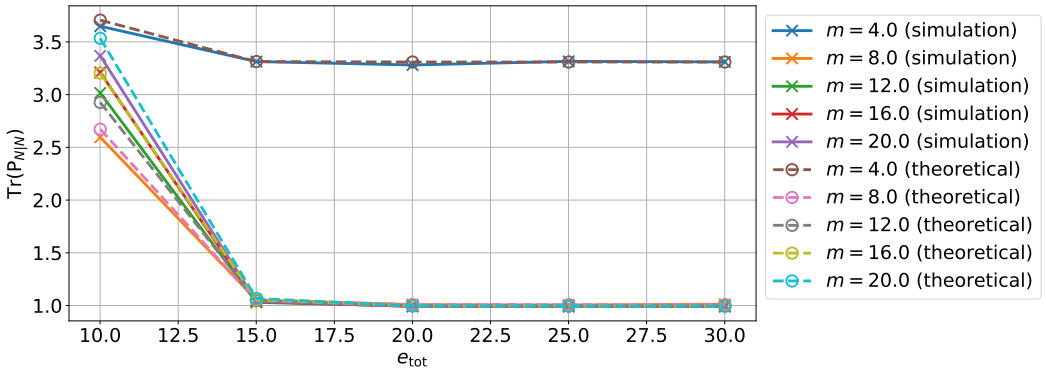


Figure 4. Theoretical and simulated variance of estimation error on the position depending on the energy supplied to each variable using an unreliable memory for different numbers of bits in the representation m in the case of the large-size example.

5.2. Solutions to the Optimization Problems

We now focus on the optimization problems introduced in Section 4, starting with the first one. Figure 6 shows the amount of energy e_{tot} needed to store each number in the

unreliable memory to achieve a fixed variance of estimation error on the position for each value of m.

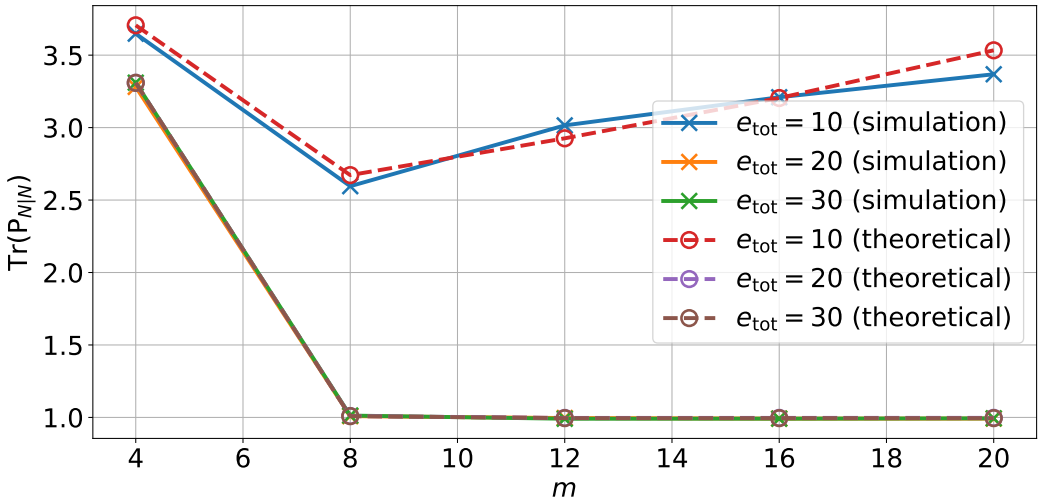


Figure 5. Theoretical and simulated variance of estimation error on the position depending on the number of quantization bits using an unreliable memory for different total energy values e_{tot} in the case of the large-size example.

The total energy e_{tot} was calculated both using the optimal allocation from Algorithm 1 and using a uniform energy allocation. From Figure 7, we can see that the total energy e_{tot} of the memory slightly increases with the number of bits $m + n$. The slight increase in memory consumption is due to the form of the optimal solution (38). Indeed, once the minimum number of bits needed to achieve the performance constraint is reached, then additional bits will be set at the minimum energy threshold e_{thres}.

Figure 7 compares the optimal solution from Figure 6 with a uniform energy allocation. This shows that the optimal energy allocation allows for a significant energy gain compared to the uniform allocation. Here, for the minimum number of bits needed to achieve the performance constraint, the optimal allocation require 56% less energy than the uniform allocation.

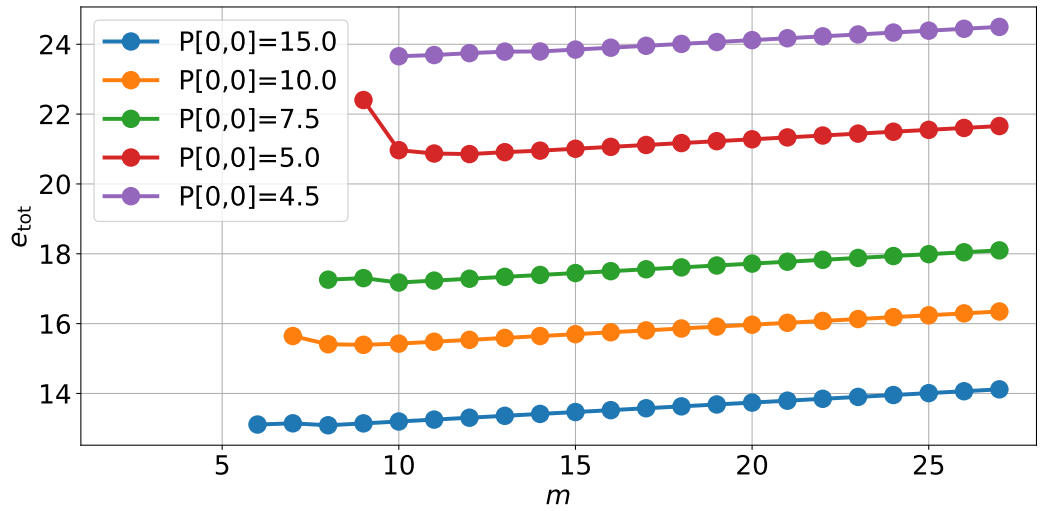


Figure 6. The energy needed to store each variable in an unreliable memory to achieve various desired variances of estimation error on the position depending on the number m of bits in the representation with the optimal energy allocation.

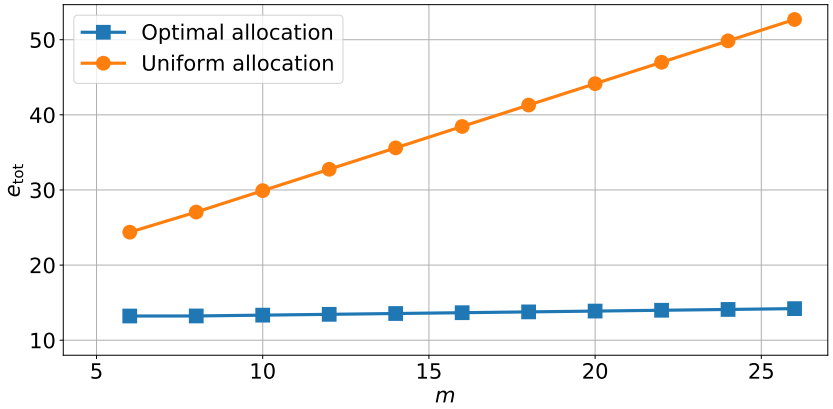


Figure 7. Energy needed to store each variable in an unreliable memory to achieve a variance of estimation error on the position $P_{N|N}[0,0] = 15$, depending on the number m of bits in the representation.

We now focus on the second optimization problem defined in Section 4.2 at equation (40) where only a limited number of energy levels are available. Figure 8 shows the total energy needed for each variable in memory to achieve a fixed level of error depending on the number L of energy level possible. For each considered number of levels $L \in [\![1,7]\!]$, the total energy e_{tot} was computed for all possible energy allocations using the optimal solution (41). The minimum energy possible for each value of L was then kept and is shown in Figure 8. This minimum energy is compared with the minimum energy needed for Problem (36) where there are as many energy levels possible as the number of bits. Here, the total number of quantization bits is $B = 20$.

We observe that even a small number of energy levels L can lead to significant gains in energy. In this case, only seven levels of energy allow achieving 95% of the maximum energy gain that was obtained in the first optimization problem. When looking at the optimal energy allocation for each value of L bit by bit, we notice that, in most cases, the optimal solution seems to be when the energy levels are uniformly shared between the bits. This means that, if there are B bits and L levels available and L is a divisor of B, then each group of bits assigned to each energy level will have a size of $n_\ell = \frac{B}{L}$.

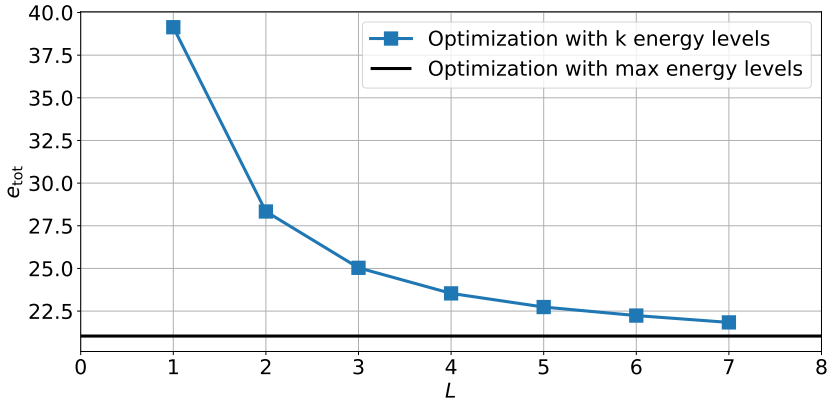


Figure 8. Energy needed to store a variable in memory e_{tot} for different values of energy level available L to achieve a fixed covariance value.

6. Conclusions

In this paper, we studied a quantized Kalman filter implemented with unreliable memories. We provided analytical expressions for the covariance matrix of the estimation error and provided updated filter equations to take into account all considered sources of errors. We proposed and solved two optimization problems that allowed us to find

the best trade-offs between the energy consumption and the performance of the filter. The simulation results showed the accuracy of the theoretical analysis and illustrated the significant energy gains provided by our approach.

Due to the generic nature of the considered error propagation model, these results could be used for various realistic noise-versus-energy models of unreliable components. Furthermore, the methodology presented in this work could also be extended to other algorithms where sources of unreliability can be introduced, such as belief propagation [51], binary recursive estimation [10], and multi-agent systems [52].

Author Contributions: Conceptualization, E.D., A.A.-E.-B., L.R.V. and F.L.-P.; Data curation, J.K.; Formal analysis, J.K.; Funding acquisition, E.D., A.A.-E.-B. and F.L.-P.; Methodology, J.K.; Project administration, E.D., A.A.-E.-B. and F.L.-P.; Software, J.K.; Supervision, E.D., A.A.-E.-B. and L.R.V.; Validation, E.D., A.A.-E.-B., L.R.V. and F.L.-P.; Visualization, J.K. and F.L.-P.; Writing—original draft, J.K.; Writing—review and editing, J.K., E.D., A.A.-E.-B., L.R.V. and F.L.-P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by grant ANR-17-CE40-0020 (EF-FECtive project), by Fonds de recherche du Québec—Nature et technologies (grant 2021-NC-286323) and by the "Make our Planet Great Again" Initiative of the Thomas Jefferson Fund.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Computation of $\tilde{x}_{k+1|k+1} - x_{k+1}$

$$\tilde{x}_{k+1|k+1} - x_{k+1} = \hat{x}_{k+1|k+1} + \Delta\tilde{x}_{k+1|k+1} - x_{k+1} \tag{A1}$$

$$= (D_{k+1} + \delta_{D_{k+1}})\Delta\tilde{x}_{k|k} + (D_{k+1} + \delta_{D_{k+1}})\hat{x}_{k|k} + D_{k+1}\epsilon_{x_{k|k}} + (K_{k+1} + \delta_{K_{k+1}})y_{k+1} + K_{k+1}\epsilon_{y_{k+1}} + \epsilon_{\times} + \delta\hat{x}^{\mathbf{mem}}_{k+1|k+1} - x_{k+1} \tag{A2}$$

$$= (D_{k+1} + \delta_{D_{k+1}})\Delta\tilde{x}_{k|k} + (D_{k+1} + \delta_{D_{k+1}})\hat{x}_{k|k} + D_{k+1}\epsilon_{x_{k|k}} + (K_{k+1} + \delta_{K_{k+1}})(Hx_{k+1} + v_{k+1}) + K_{k+1}\epsilon_{y_{k+1}} + \epsilon_{\times} + \delta\hat{x}^{\mathbf{mem}}_{k+1|k+1} - x_{k+1} \tag{A3}$$

$$= (D_{k+1} + \delta_{D_{k+1}})\Delta\tilde{x}_{k|k} + (D_{k+1} + \delta_{D_{k+1}})\hat{x}_{k|k} + D_{k+1}\epsilon_{x_{k|k}} + (K_{k+1} + \delta_{K_{k+1}})v_{k+1} + K_{k+1}\epsilon_{y_{k+1}} + \epsilon_{\times} + \delta\hat{x}^{\mathbf{mem}}_{k+1|k+1} + ((K_{k+1} + \delta_{K_{k+1}})H - \mathbf{I})x_{k+1} \tag{A4}$$

$$= (D_{k+1} + \delta_{D_{k+1}})\tilde{x}_{k|k} - (\mathbf{I} - (K_{k+1} + \delta_{K_{k+1}})H)Fx_k + (K_{k+1} + \delta_{K_{k+1}})v_{k+1} + K_{k+1}\epsilon_{y_{k+1}} + \epsilon_{\times} + \delta\hat{x}^{\mathbf{mem}}_{k+1|k+1} + D_{k+1}\epsilon_{x_{k|k}} + ((K_{k+1} + \delta_{K_{k+1}})H - \mathbf{I})u_k\,. \tag{A5}$$

If H and F are only composed of integer components, due to how the quantization is done:

$$(D_{k+1} + \delta_{D_{k+1}}) = (\mathbf{I} - (K_{k+1} + \delta_{K_{k+1}})H)F\,. \tag{A6}$$

Appendix B. Computation of the Optimal Solution to Problem 1

From optimization Problem (37), we can define the Lagrangian:

$$L(e, \nu, \lambda) = \sum_{b=0}^{B-1} e_b + \nu\Big(\sum_{b=0}^{B-1} 4^b e^{-e_b a} - \mathcal{V}\Big) - \sum_{b=0}^{B-1} \lambda_b(e_b - e_{\text{thres}})\,. \tag{A7}$$

From the KKT conditions, for the optimal solution e^*:

$$\nu\Big(\sum_{b=0}^{B-1} 4^b e^{-e_b^* a} - \mathcal{V}\Big) = 0\,, \nu \geq 0 \tag{A8}$$

$$\lambda_b(e_b^* - e_{\text{thres}}) = 0\,, \lambda_b \geq 0\ \forall b \in [\![0, B-1]\!] \tag{A9}$$

$$\frac{\partial L}{\partial e_b^*} = 1 - \nu 4^b a e^{-e_b a} - \lambda_b = 0 \tag{A10}$$

From (A9) and (A10):

$$\lambda_b = 1 - \nu 4^b a e^{-e_b a} \geq 0\,. \tag{A11}$$

If $\nu = 0$ then $\lambda_b = 1$ and $e_b = e_{\text{thres}}$. Therefore, we claim that $\nu \neq 0$, and thus $\sum_{b=0}^{B-1} 4^b e^{-e_b^* a} = \mathcal{V}$.

If $\nu \leq \frac{1}{4^b a}$, then it is not possible to have $e_b > e_{\text{thres}}$ since it would mean that $\lambda_b = 0$, and thus $\nu = \frac{1}{4^b a} e^{e_b a} \geq \frac{1}{4^b a}$, which is in contradiction with the hypothesis. Therefore, if $\nu \leq \frac{1}{4^b a}$, then $e_b = e_{\text{thres}}$.

If $\nu > \frac{1}{4^b a}$, then by the same logic as before $e_b > e_{\text{thres}}$. In this case, $\lambda_b = 0$, and thus $e_b = \frac{1}{a} \log(\nu 4^b a)$.

References

1. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* **1960**, *82*, 35–45, doi:10.1115/1.3662552.
2. Lai, X.; Yang, T.; Wang, Z.; Chen, P. IoT Implementation of Kalman Filter to Improve Accuracy of Air Quality Monitoring and Prediction. *Appl. Sci.* **2019**, *9*, 1831, doi:10.3390/app9091831.
3. Anania, G.; Tognetti, A.; Carbonaro, N.; Tesconi, M.; Cutolo, F.; Zupone, G.; Rossi, D.D. Development of a novel algorithm for human fall detection using wearable sensors. In Proceedings of the IEEE SENSORS, Lecce, Italy, 26–29 October 2008; pp. 1336–1339, doi:10.1109/ICSENS.2008.4716692.
4. Wang, T.; Hu, J.; Ma, T.; Song, J. Forest fire detection system based on Fuzzy Kalman filter. In Proceedings of the 2020 International Conference on Urban Engineering and Management Science (ICUEMS), Zhuhai, China, 24–26 April 2020; pp. 630–633, doi:10.1109/ICUEMS50872.2020.00138.
5. Sung, K.; Kim, H. Simplified KF-based energy-efficient vehicle positioning for smartphones. *J. Commun. Netw.* **2020**, *22*, 93–107, doi:10.1109/JCN.2020.000003.
6. Horowitz, M. 1.1 Computing's energy problem (and what we can do about it). In Proceedings of the IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), San Francisco, CA, USA, 9–13 Febuary 2014; pp. 10–14, doi:10.1109/ISSCC.2014.6757323.
7. Dreslinski, R.G.; Wieckowski, M.; Blaauw, D.; Sylvester, D.; Mudge, T. Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits. *Proc. IEEE* **2010**, *98*, 253–266, doi:10.1109/JPROC.2009.2034764.
8. Kim, Y.; Kang, M.; Varshney, L.R.; Shanbhag, N.R. Generalized Water-Filling for Source-Aware Energy-Efficient SRAMs. *IEEE Trans. Commun.* **2018**, *66*, 4826–4841.
9. Kim, Y.; Jeon, Y.; Guyot, C.; Cassuto, Y. Optimizing the Write Fidelity of MRAMs. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), Los Angeles, CA, USA, 21–26 June 2020; pp. 792–797, doi:10.1109/ISIT44484.2020.9173990.
10. Dupraz, E.; Varshney, L.R. Binary Recursive Estimation on Noisy Hardware. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), Paris, France, 7–12 July 2019; pp. 877–881.
11. Yang, Y.; Grover, P.; Kar, S. Computing Linear Transformations With Unreliable Components. *IEEE Trans. Inf. Theory* **2017**, *63*, 3729–3756, doi:10.1109/TIT.2017.2692244.
12. Henwood, S.; Leduc-Primeau, F.; Savaria, Y. Layerwise Noise Maximisation to Train Low-Energy Deep Neural Networks. In Proceedings of the 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), Genova, Italy, 31 August–2 September 2020; pp. 271–275, doi:10.1109/AICAS48895.2020.9073854.
13. Hacene, G.B.; Leduc-Primeau, F.; Soussia, A.B.; Gripon, V.; Gagnon, F. Training Modern Deep Neural Networks for Memory-Fault Robustness. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 26–29 May 2019; pp. 1–5, doi:10.1109/ISCAS.2019.8702382.
14. Shang, Y. Resilient consensus in multi-agent systems with state constraints. *Automatica* **2020**, *122*, 109288.
15. Yang, Y.; Grover, P.; Kar, S. Fault-tolerant distributed logistic regression using unreliable components. In Proceedings of the 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 27–30 September 2016; pp. 940–947, doi:10.1109/ALLERTON.2016.7852335.
16. Hegde, R.; Shanbhag, N.R. Energy-efficient signal processing via algorithmic noise-tolerance. In Proceedings of the International Symposium on Low Power Electronics and Design (Cat. No. 99TH8477), San Diego, CA, USA, 17 August 1999; pp. 30–35, doi:10.1145/313817.313834.
17. Huang, C.; Li, Y.; Dolecek, L. ACOCO: Adaptive Coding for Approximate Computing on Faulty Memories. *IEEE Trans. Commun.* **2015**, *63*, 4615–4628, doi:10.1109/TCOMM.2015.2481898.
18. Yaz, E.E.; Jeong, C.S.; Yaz, Y.I. An LMI approach to discrete-time observer design with stochastic resilience. *J. Comput. Appl. Math.* **2006**, *188*, 246–255, doi:10.1016/j.cam.2005.04.016.
19. Chen, Y.; Chen, C.; Xue, A. Distributed non-fragile l 2 - l ∞ filtering over sensor networks with random gain variations and fading measurements. *Neurocomputing* **2019**, *338*, 154–162, doi:10.1016/j.neucom.2018.12.008.

20. Nahi, N. Optimal recursive estimation with uncertain observation. *IEEE Trans. Inf. Theory* **1969**, *15*, 457–462.
21. Hounkpevi, F.O.; Yaz, E.E. Robust minimum variance linear state estimators for multiple sensors with different failure rates. *Automatica* **2007**, *43*, 1274–1280, doi:10.1016/j.automatica.2006.12.025.
22. Petersen, I.R.; McFarlane, D.C.; Rotea, M.A. Optimal Guaranteed Cost Control of Discrete-time Uncertain Linear Systems. *IFAC Proc. Vol.* **1993**, *26*, 35–38, doi:10.1016/S1474-6670(17)48889-6.
23. Yang, G.H.; Wang, J.L. Robust nonfragile Kalman filtering for uncertain linear systems with estimator gain uncertainty. *IEEE Trans. Autom. Control* **2001**, *46*, 343–348, doi:10.1109/9.905707.
24. Huang, Y.; Zhang, Y.; Wu, Z.; Li, N.; Chambers, J. A Novel Adaptive Kalman Filter With Inaccurate Process and Measurement Noise Covariance Matrices. *IEEE Trans. Autom. Control* **2018**, *63*, 594–601, doi:10.1109/TAC.2017.2730480.
25. Jarrah, A. Optimized parallel architecture of Kalman filter for radar tracking applications. *Jordan J. Electr. Eng.* **2016**, *2*, 215–230.
26. Sunil Kumar, T.; Duraiswamy, P. Optimization of Kalman Filter for Target Tracking Applications. In *Advances in Multidisciplinary Analysis and Optimization*; Salagame, R.R., Ramu, P., Narayanaswamy, I., Saxena, D.K., Eds.; Springer: Singapore, 2020; pp. 203–212, doi:10.1007/978-981-15-5432-2_17.
27. Pereira, P.T.L.; Paim, G.; Ücker, P.; Costa, E.; Almeida, S.; Bampi, S. Exploring Architectural Solutions for an Energy-Efficient Kalman Filter Gain Realization. In Proceedings of the 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Genoa, Italy, 27–29 November 2019; pp. 650–653, doi:10.1109/ICECS46596.2019.8964734.
28. Wang, Z.; Zhang, J.; Verma, N. Reducing quantization error in low-energy FIR filter accelerators. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, Australia, 19–24 April 2015; pp. 1032–1036, doi:10.1109/ICASSP.2015.7178126.
29. Xia, D.; Zhang, Y.; Cai, P.; Huang, L. An Energy-Efficient Signal Detection Scheme for a Radar-Communication System Based on the Generalized Approximate Message-Passing Algorithm and Low-Precision Quantization. *IEEE Access* **2019**, *7*, 29065–29075, doi:10.1109/ACCESS.2019.2899883.
30. Marcastel, A.; Fijalkow, I.; Swindlehurst, L. Energy efficient downlink massive MIMO: Is 1-bit quantization a solution? In Proceedings of the 16th International Symposium on Wireless Communication Systems (ISWCS), Oulu, Finland, 27–30 August 2019; pp. 198–202, doi:10.1109/ISWCS.2019.8877256.
31. Hashemi, S.; Anthony, N.; Tann, H.; Bahar, R.I.; Reda, S. Understanding the impact of precision quantization on the accuracy and energy of neural networks. In Proceedings of the Conference on Design, Automation & Test in Europe, Lausanne, Switzerland, 27–31 March 2017; pp. 1478–1483.
32. Ding, R.; Liu, Z.; Blanton, R.D.S.; Marculescu, D. Quantized deep neural networks for energy efficient hardware-based inference. In Proceedings of the 23rd Asia and South Pacific Design Automation Conference (ASP-DAC), Jeju, Korea, 22–25 Janaury 2018; pp. 1–8, doi:10.1109/ASPDAC.2018.8297274.
33. Jain, S.; Venkataramani, S.; Srinivasan, V.; Choi, J.; Chuang, P.; Chang, L. Compensated-DNN: Energy efficient low-precision deep neural networks by compensating quantization errors. In Proceedings of the 55th Annual Design Automation Conference, San Francisco, CA, USA, 24–28 June 2018; pp. 1–6, doi:10.1145/3195970.3196012.
34. Stripad, A.B. Performance Degradation in Digitally Implemented Kalman Filters. *IEEE Trans. Aerosp. Electron. Syst.* **1981**, *AES-17*, 626–634, doi:10.1109/TAES.1981.309093.
35. Verhaegen, M.; Dooren, P.V. Numerical aspects of different Kalman filter implementations. *IEEE Trans. Autom. Control* **1986**, *31*, 907–917.
36. Sun, S.; Lin, J.; Xie, L.; Xiao, W. Quantized Kalman Filtering. In Proceedings of the IEEE 22nd International Symposium on Intelligent Control, Singapore, 1–3 October 2007; pp. 7–12, doi:10.1109/ISIC.2007.4450852.
37. Li, D.; Kar, S.; Cui, S. Distributed Kalman Filtering with quantized sensing state. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, Australia, 19–24 April 2015; pp. 4040–4044, doi:10.1109/ICASSP.2015.7178730.
38. Hu, X.; Bao, M.; Zhang, X.; Wen, S.; Li, X.; Hu, Y. Quantized Kalman Filter Tracking in Directional Sensor Networks. *IEEE Trans. Mob. Comput.* **2018**, *17*, 871–883, doi:10.1109/TMC.2017.2742948.
39. You, K.; Xie, L.; Sun, S.; Xiao, W. Quantized filtering of linear stochastic systems. *Trans. Inst. Meas. Control* **2011**, *33*, 683–698, doi:10.1177/0142331209342213.
40. You, K.; Zhao, Y.; Xie, L. Recursive quantized state estimation of discrete-time linear stochastic systems. In Proceedings of the 7th Asian Control Conference, Hong Kong, China, 27–29 August 2009; pp. 170–175.
41. Dally, W.J.; Harting, R.C.; Aamodt, T.M. *Digital Design Using VHDL: A Systems Approach*; Cambridge University Press: Cambridge, UK, 2015.
42. Kern, J.; Dupraz, E.; Aïssa-El-Bey, A.; Leduc-Primeau, F. Improving the Energy-Efficiency of a Kalman Filter Using Unreliable Memories. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 5345–5349, doi:10.1109/ICASSP39728.2021.9413430.
43. Bar-Shalom, Y.; Li, X.R.; Kirubarajan, T. State Estimation in Discrete-Time Linear Dynamic Systems. In *Estimation with Applications to Tracking and Navigation*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, pp. 199–266, doi:10.1002/0471221279.ch5.
44. Thacker, N.; Lacey, A. *Tutorial: The Kalman Filter*; Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester: Manchester, UK, 1998; p. 61.
45. Ziv, J. On universal quantization. *IEEE Trans. Inf. Theory* **1985**, *31*, 344–347, doi:10.1109/TIT.1985.1057034.

46. Sripad, A.; Snyder, D. A necessary and sufficient condition for quantization errors to be uniform and white. *IEEE Trans. Acoust. Speech Signal Process.* **1977**, *25*, 442–448, doi:10.1109/TASSP.1977.1162977.
47. Mukhopadhyay, S.; Mahmoodi, H.; Roy, K. Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2005**, *24*, 1859–1880, doi:10.1109/TCAD.2005.852295.
48. Dupraz, E.; Declercq, D.; Vasić, B.; Savin, V. Analysis and Design of Finite Alphabet Iterative Decoders Robust to Faulty Hardware. *IEEE Trans. Commun.* **2015**, *63*, 2797–2809, doi:10.1109/TCOMM.2015.2452259.
49. Kameni Ngassa, C.; Savin, V.; Dupraz, E.; Declercq, D. Density Evolution and Functional Threshold for the Noisy Min-Sum Decoder. *IEEE Trans. Commun.* **2015**, *63*, 1497–1509, doi:10.1109/TCOMM.2015.2388472.
50. Berberidis, D.; Giannakis, G.B. Data Sketching for Large-Scale Kalman Filtering. *IEEE Trans. Signal Process.* **2017**, *65*, 3688–3701, doi:10.1109/TSP.2017.2691662.
51. Huang, C.; Li, Y.; Dolecek, L. Belief Propagation Algorithms on Noisy Hardware. *IEEE Trans. Commun.* **2015**, *63*, 11–24, doi:10.1109/TCOMM.2014.2379663.
52. Shang, Y. Group consensus of multi-agent systems in directed networks with noises and time delays. *Int. J. Syst. Sci.* **2015**, *46*, 2481–2492, doi:10.1080/00207721.2013.862582.