Towards Lower Precision Adaptive Filters: Facts from Backward Error Analysis of RLS

Lee, J., & Vandierendonck, H. (2021). Towards Lower Precision Adaptive Filters: Facts from Backward Error Analysis of RLS. *IEEE Transactions on Signal Processing*. Advance online publication. https://doi.org/10.1109/TSP.2021.3086355

Published in:
IEEE Transactions on Signal Processing

Document Version:
Peer reviewed version

Towards Lower Precision Adaptive Filters: Facts from Backward Error Analysis of RLS

JunKyu Lee, Hans Vandierendonck
{junkyu.lee, h.vandierendonck}@qub.ac.uk

***Abstract*—Lower precision arithmetic can improve the throughput of adaptive filters, while requiring less hardware resources and less power. Such benefits are crucial for adaptive filters, especially for IoT and wearable applications. In order to apply lower precision arithmetic to adaptive filters, a clear rounding error analysis framework is required, since lower precision arithmetic can degrade the filter performance. Previously, rounding error analyses of adaptive filters were based on forward error analysis. This limited the descriptiveness of rounding error impact on adaptive filter performance in relation to other external variables such as measurement noise, regularisation, and numerical stability of an algorithm. To overcome such limitations, we first present a new backward error analysis framework for adaptive Recursive Least Squares (RLS) filters. Our framework transforms finite precision arithmetic adaptive filters into exact arithmetic adaptive filters with the input data corrupted by rounding error noise that is additive to measurement noise. Findings throughout our backward error analysis framework can provide a guide on how to apply lower precision arithmetic to adaptive filters: (i) the magnitudes of the rounding error noise depend on the numerical stability of the implementation algorithm, arithmetic precision, and regularisation, (ii) the rounding error noise is independently additive to measurement noise, (iii) a higher regularisation is recommended for lower precision arithmetic adaptive filters, and (iv) adaptive filters using lower precision arithmetic have equivalent filter performance to those using higher precision if the magnitudes of rounding error noise are lower than measurement noise.**

***Index Terms*—Lower Precision Adaptive Filter; Rounding Error; Lower Precision RLS; Backward Error Analysis**

I. INTRODUCTION

Linear regression using least squares is a very common estimation structure in a wide range of adaptive filter applications [1], [2]. The Recursive Least Squares (RLS) algorithm, in particular, has been widely used in adaptive filter applications such as system identification, noise cancellation, prediction, and inverse modelling, thanks to its fast convergence and tracking ability [3], [4]. Adaptive filters adapt the weights per input sample arrival in real time. As a consequence, adaptive filters often require the improved throughput using a reduced memory footprint due to resource and throughput constraints. For example, the improved throughput is required to deal with a flood of real-time data and the implementation of an adaptive filter should fit into the memory budget given by an embedded system especially designed for IoT applications or a wearable device. Therefore, the modest computational cost and storage are required for adaptive filters along with other characteristics such as good convergence, good generalisation ability, numerical robustness, etc. [3].

Recently, many research attempts have explored minimising the computational cost of RLS adaptive filters in order to improve their throughput capability [5]–[11]. Most of this work has explored low dimensional approximation techniques in order to reduce computational complexity of RLS. Applying lower precision arithmetic (i.e., a lower precision of arithmetic than is used by a baseline implementation) to RLS adaptive filters is another promising approach towards high throughput filter capabilities, since the reduced precision arithmetic computation reduces the memory footprint and the time spent transferring data across buses and interconnects [12], [13]. While arithmetic precision is an important lever to accelerate signal processing applications, many applications may not tolerate lower precision arithmetic as the lack of sufficient precision would result in inaccurate predictions. For example, the rounding errors magnified by reduced precision arithmetic can break an essential mathematical property of an RLS algorithm, resulting in inaccurate predictions [14], [15]. Therefore, it is problematic to apply lower precision arithmetic to RLS without first understanding how the increased rounding error caused by lower precision arithmetic affects RLS performance. A simple, clear analysis model that describes the rounding error effects to RLS performance is required to provide a guide on how to apply safely lower precision arithmetic to RLS adaptive filters.

The numerical linear algebra community has utilised two types of rounding error analysis: forward error analysis and backward error analysis [16]. Forward error analysis investigates the error in the computed solution by tracking the propagation of the rounding error; the error analysis follows the forward direction, analysing the effect of rounding error to the computed solution accuracy. In contrast, backward error analysis investigates the minimum perturbation of the system that makes the computed solution (including rounding error) the exact solution for the perturbed system; the error analysis follows the backward direction, analysing the effect of rounding error to the perturbation of the system. Previous research attempts discovered several important properties of rounding error effects to the performance of RLS filters [4], [14], [15], [17]. All of these works, to the best of our knowledge, explored forward error analysis, generating a large number of intermediate variables. In contrast, backward error analysis supports a simple, clear rounding error analysis model for least squares algorithms, providing a better insight into

J. Lee and H. Vandierendonck are with the Institute of Electronics, Communications and Information Technology (ECIT), Queens University Belfast, Northern Ireland, UK.

Manuscript received XXX, 2020; revised .

the numerical stability of the algorithms over forward error analysis [16].

Backward error analysis interprets a finite precision arithmetic system as an exact arithmetic system with perturbations [18]; Wilkinson invented the backward error analysis technique, and it was named as "backward type of error analysis" in [19]. Backward error analysis has been recognised as a significant error analysis tool particularly for the numerical linear algebra community [16], [18], [20], [21]. However, it was unacquainted in other communities, including the signal processing community. Although the work of [22] used the concept of "backward consistency" in its rounding error propagation analysis, it did not investigate the backward errors (i.e., the perturbation of the original system such as the perturbation on the targets in RLS).

To the best of our knowledge, this paper is the first to present an exact arithmetic model of finite precision arithmetic adaptive filter, generated by backward error, in which rounding error components sits on an equal footing with the measurement noise; finite precision arithmetic RLS filters are converted to exact arithmetic filters with input data corrupted by noise generated by rounding errors that is additive to measurement noise. We will denote the noise generated by rounding error as *rounding error noise* from this point forward in order to distinguish it from measurement noise. The main contributions in this paper are three fold:

- We present a new rounding error analysis model for an adaptive filter that includes rounding error explicitly on an equal footing with measurement noise.
- Based on our model, we evaluate the impact of lower precision on the rounding error noise with respect to different types of RLS (e.g., different numerical stability of the algorithm). We also analyse the mutual impacts between regularisation and rounding error noise theoretically and empirically - such evaluation was not feasible with traditional adaptive filter rounding error modelling, since rounding error components were not explicitly included in traditional modelling.
- Our findings from the analysis model can provide useful information on application of lower precision arithmetic to RLS adaptive filters. The findings are as follows:
  - The magnitude of rounding error noise depends on arithmetic precision, the numerical stability of the implementation algorithm, and the feature size.
  - Applying lower precision arithmetic to adaptive filters requires a higher regularisation to abate the rounding error noise effects on performance. Therefore, higher regularisation can be considered for lower precision adaptive filters for IoT or wearable applications having the constraints of hardware budgets, computational throughput, and power.
  - Rounding error noise is independently additive to measurement noise. Therefore, we can apply lower precision arithmetic without losing the filter performance unless the magnitude of rounding error noise exceeds the magnitude of measurement noise.
  - Rounding error noise is not white noise. Therefore, rounding error noise dominantly affects adaptive filter performance if measurement noise follows white Gaussian noise, since an exact arithmetic RLS is an unbiased estimator against white Gaussian noise [3].

We describe two RLS algorithms, Matrix Inversion Lemma RLS (MIL-RLS) and QR-RLS used in [23], and general backward error analysis in Section II, our theoretical framework in Section III (with traditional rounding error analysis models in Section III-C and our new model in Section III-D), experimental framework in Section IV, related work in Section V, discussion in Section VI, and conclusion in Section VII. Since it is well known that when the forgetting factor is less than '1', the numerical stability against rounding errors has been improved [14], [15], [17], [24], we focus on the regularised RLS algorithms when the forgetting factor equals '1' to consider the worst case for lower precision RLS filter performance.

II. Regularised RLS and Backward Error Analysis

We discuss two regularised RLS algorithms, the IEEE 754 standard [25], and backward error analysis in this section.

A. Regularised RLS Algorithms

An RLS algorithm for linear regression seeks the weights $\hat{\mathbf{w}}_t$ that minimise [3]

$$\min_{\hat{\mathbf{w}}_t} \|\mathbf{X}_t\hat{\mathbf{w}}_t - \dot{\mathbf{y}}_t\|_2^2, \tag{1}$$

where $\mathbf{X}_t = [\mathbf{x}_1, ..., \mathbf{x}_t]^T \in \mathbb{R}^{t\times N}$ where $\mathbf{x}_i$ is an input vector at time i having N features, $\dot{\mathbf{y}}_t = [\dot{y}_1, ..., \dot{y}_t]^T \in \mathbb{R}^{t\times 1}$, $\dot{y}_i$ is a target at time i, and $\|\cdot\|_2$ represents the 2 norm of a vector [26]. Eq. (1) is equivalent to seek $\hat{\mathbf{w}}_t$ in

$$(\mathbf{X}_t^T\mathbf{X}_t)\hat{\mathbf{w}}_t = \mathbf{X}_t^T\dot{\mathbf{y}}_t \tag{2}$$

Regularisation plays an essential role in adaptive filters to make them converge to the optimal Wiener solution smoothly and continuously in the presence of measurement noise [27], [28]. Also, regularisation is essential in RLS due to the ill-posed nature of least-squares estimation [3]. Therefore, we explore the impact of rounding errors on the optimal regularisation parameter in regularised RLS algorithms. A regularised RLS seeks the weights, $\hat{\mathbf{w}}_t$, that solves the minimisation problem with a regularisation parameter λ:

$$\min_{\hat{\mathbf{w}}_t} \|\mathbf{X}_t\hat{\mathbf{w}}_t - \dot{\mathbf{y}}_t\|_2^2 + \lambda\|\hat{\mathbf{w}}_t\|_2, \tag{3}$$

where λ is a Tikhonov regularisation [29] parameter that we consider in this paper. The $\hat{\mathbf{w}}_t$ in Eq. (3) can be sought with

$$\hat{\mathbf{w}}_t = (\mathbf{X}_t^T\mathbf{X}_t + \lambda\mathbf{I})^{-1}\mathbf{X}_t^T\dot{\mathbf{y}}_t, \tag{4}$$

where $\mathbf{I} \in \mathbb{R}^{N\times N}$ is an identical matrix. The RLS algorithm for adaptive filters updates the weights:

$$\hat{\mathbf{w}}_t = \hat{\mathbf{w}}_{t-1} + (\dot{y}_t - \hat{\mathbf{w}}_{t-1}^T\mathbf{x}_t)\mathbf{P}_t\mathbf{x}_t, \tag{5}$$

where $\mathbf{P}_t$ (=$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}$) can be found by recursive operations and the regularisation λ should be used when $\mathbf{P}_t$ is initialised as Eq. (7) for a MIL-RLS and Eq. (11) for a QR-RLS. Depending on how $\mathbf{P}_t$ is updated, RLS can be implemented using various types of numerical schemes such as a MIL, a QR decomposition using Givens rotation [23], etc.

1) Numerically Unstable MIL-RLS: MIL-RLS utilises Eq. (6) to update the matrix $\mathbf{P}_t$ and Eq. (5) to update $\hat{\mathbf{w}}_t$ at time t:

$$\mathbf{P}_t = \mathbf{P}_{t-1} - \frac{\mathbf{P}_{t-1}\mathbf{x}_t\mathbf{x}_t^T\mathbf{P}_{t-1}}{1+\mathbf{x}_t^T\mathbf{P}_{t-1}\mathbf{x}_t}. \tag{6}$$

MIL-RLS is numerically unstable due to so called "divergence phenomenon [14]"; the positive definiteness of $\mathbf{P}_t$ can be broken due to the accumulation of rounding errors. For example, if $\mathbf{x}_t^T\mathbf{P}_{t-1}\mathbf{x}_t \approx -1$, some of the updated matrix elements will blow up. The $\mathbf{P}_1$ is initialised with:

$$\mathbf{P}_1 = \frac{1}{\lambda}\mathbf{I}. \tag{7}$$

2) Numerically Stable QR-RLS: Using a QR decomposition, the matrix $\mathbf{P}_t$ is represented as [23]

$$\mathbf{P}_t = (\mathbf{X}_t^T\mathbf{X}_t)^{-1} = ((\mathbf{Q}_t\mathbf{R}_t)^T(\mathbf{Q}_t\mathbf{R}_t))^{-1} = \mathbf{R}_t^{-1}\mathbf{R}_t^{-T}. \tag{8}$$

Hence, Eq. (5) becomes

$$\hat{\mathbf{w}}_t = \hat{\mathbf{w}}_{t-1} + \frac{(\dot{y}_t - \hat{\mathbf{w}}_{t-1}^T\mathbf{x}_t)}{1+\|\mathbf{R}_{t-1}^{-T}\mathbf{x}_t\|_2^2}\mathbf{R}_{t-1}^{-1}\mathbf{R}_{t-1}^{-T}\mathbf{x}_t. \tag{9}$$

By denoting $\mathbf{u}_t = \mathbf{R}_{t-1}^{-T}\mathbf{x}_t$,

$$\hat{\mathbf{w}}_t = \hat{\mathbf{w}}_{t-1} + \frac{(\dot{y}_t - \hat{\mathbf{w}}_{t-1}^T\mathbf{x}_t)}{1+\|\mathbf{u}_t\|_2^2}\mathbf{R}_{t-1}^{-1}\mathbf{u}_t. \tag{10}$$

QR-RLS is more widely used than MIL-RLS due to its superior numerical stability [3]. No divergence phenomenon occurs by replacing $\mathbf{x}_t^T\mathbf{P}_{t-1}\mathbf{x}_t$ to $\|\mathbf{u}_t\|^2$. $\mathbf{R}_{t-1}^{-1}$ is initialised with:

$$\mathbf{R}_{t-1}^{-1} = \frac{1}{\sqrt{\lambda}}\mathbf{I} \tag{11}$$

and updated:

$$\begin{bmatrix}\mathbf{R}_t^{-1} & \mathbf{r}_t\end{bmatrix} = \begin{bmatrix}\mathbf{R}_{t-1}^{-1} & \mathbf{0}\end{bmatrix}\mathbf{Q}_t^T, \tag{12}$$

where $\mathbf{r}_t = -\mathbf{R}_{t-1}^{-1}\mathbf{u}_t/\sqrt{1+\|\mathbf{u}_t\|_2^2}$. QR-RLS updates $\mathbf{R}_t^{-1}$ using (12) and $\hat{\mathbf{w}}_t$ using (10) at time t.

B. IEEE 754 Standard

The IEEE 754 floating point data format [25] consists of sign, exponent, and significand as shown in Eq. (13). For example, a floating point number has a $(p+1)$-bit significand (including the hidden one), an e-bit exponent, and an 1 sign bit. The value represented by the radix 2 floating point number format is as follows:

$$sign \times (1\times 2^0 + d_1\times 2^{-1} + ... + d_p\times 2^{-p})\times 2^{exponent-bias}, \tag{13}$$

where d_1, ... , d_p represent binary digits, the '1' associated with the coefficient 2^0 is referred to as the hidden '1', the $exponent$ is stored in offset notation, and the $bias$ is a positive constant. For double precision format, $p = 52$, $e = 11$, and $bias = 1023$, and for single precision format, $p = 23$, $e = 8$, and $bias = 127$. The machine epsilon ϵ_{mach} is defined as $2^{-(p+1)}$. For example, single precision machine epsilon is 2^{-24}, and double precision machine epsilon is 2^{-53}.

IEEE 754 standard requires exact rounding for addition, subtraction, multiplication, and division; the floating point arithmetic result should be identical to the one obtained from the final rounding after exact calculation. Based on the IEEE 754 rounding to nearest mode, floating point arithmetic should follow Eq. (14) [26]:

$$fl(x_1 \odot x_2) = (x_1 \odot x_2)(1+\epsilon_r), \tag{14}$$

where $|\epsilon_r| \leq \epsilon_{mach}$, $\odot$ is one of the four arithmetic operations (addition, subtraction, multiplication, and division) and $fl(\cdot)$ represents the result from the floating point arithmetic. Eq. (14) is used for *forward error analysis* to obtain the error bound of the computed quantity [20]. In other words, forward error analysis is used to analyse the propagation of the error from the input to the output.

C. Backward Error Analysis

While forward error analysis is used to seek the errors from intermediate computing components, backward error analysis is used to seek which perturbed system works with the exact arithmetic due to finite precision arithmetic [16]. For example, Eq. (14) can be represented in another way:

$$fl(x_1 \odot x_2) = x_1(1+\epsilon_{r_1}) \odot x_2(1+\epsilon_{r_2}). \tag{15}$$

The interpretation of Eq. (15) is different to Eq. (14). In Eq. (15), the floating point arithmetic can be interpreted as the exact arithmetic on the two perturbed (noisy) data, $x_1(1+\epsilon_{r_1})$ and $x_2(1+\epsilon_{r_2})$. The ϵ_{r_1} and ϵ_{r_2} are not unique in general [20]. One of the possible values is $\epsilon_{r_1} = \epsilon_{r_2} = \epsilon_r$ if $\odot$ is a floating point addition. Backward error analysis seeks the possible smallest size of ϵ_{r_1} and ϵ_{r_2} (e.g., the minimum of $\sqrt{\epsilon_{r_1}^2 + \epsilon_{r_2}^2}$) that satisfies Eq. (15). Backward error also can be defined for a linear system solver as follows:

Definition 1. *Backward Error for Linear Solver:*$A\mathbf{x} = \mathbf{b}$ *[18] The norm-wise backward error,* $\eta_{ALG}(\tilde{\mathbf{x}})$*, for an algorithm ALG that solves a linear system is*

$$\eta_{ALG}(\tilde{\mathbf{x}}) = min\{\varepsilon : (A+\Delta A)\tilde{\mathbf{x}} = \mathbf{b}+\Delta\mathbf{b}\}, \tag{16}$$

satisfying both $\|\Delta A\|_2 \leq \varepsilon\|E\|_2$ *and* $\|\Delta\mathbf{b}\|_2 \leq \varepsilon\|\mathbf{f}\|_2$*, where the matrix* E *and the vector* $\mathbf{f}$ *are arbitrary.*

For example, the computed weights $\tilde{\mathbf{x}}$ is the exact solution for a perturbed system, $(A+\Delta A)\tilde{\mathbf{x}} = \mathbf{b}+\Delta\mathbf{b}$ due to rounding errors from finite precision arithmetic. Then, the backward error, $\eta_{ALG}(\tilde{\mathbf{x}})$, is [18]

$$\eta_{ALG}(\tilde{\mathbf{x}}) = \frac{\|\mathbf{b}-A\tilde{\mathbf{x}}\|_2}{\|E\|_2\|\tilde{\mathbf{x}}\|_2 + \|\mathbf{f}\|_2}. \tag{17}$$

An algorithm is called *backward stable* if it produces a small backward error for *any input data*, A and $\mathbf{b}$. The definition of "small" depends on circumstances of the problem that the algorithm solves. Adaptive filters implemented with a numerically stable algorithm such as QR-RLS will limit the backward error within finite bounds, while adaptive filters implemented with a numerically unstable algorithm such as MIL-RLS may not limit the backward error.

In backward error analysis, the accuracy of the weights in RLS depends on two factors including the backward error and the condition number of the system (i.e., the condition number of $\mathbf{X}^T\mathbf{X}$) [26]. The backward error depends on how to implement algorithm and the regularisation parameter in RLS (e.g., lower backward error, more stable), and the accuracy of the weights improves in proportion to *((backward error)* $\times$ *(condition number))*$^{-1}$. In other words, the forward error (i.e., the error in the computed weights) is bounded by the condition number if the algorithm is backward stable.

III. Forward vs Backward Error Analysis of Adaptive Filters

A. Application of Linear Regression to Adaptive Filters

Our rounding error analysis model is constructed based on the linear regression model:

$$\dot{y}_t = \tilde{\mathbf{w}}_{t-1}^T \mathbf{x}_t + \tilde{e}_t, \tag{18}$$

where $\dot{y}_t$ is a target at time t, $\mathbf{x}_t$ is the input sample at time t, $\tilde{\mathbf{w}}_{t-1}$ is the updated weights at time $(t-1)$, and $\tilde{e}_t$ is the error between the target and the estimation at time t. We apply the linear regression model to an adaptive filter application (i.e., system identification) by mapping the $\dot{y}_t$ to noisy targets such that $\dot{y}_t = y_t + \sigma_t$ where y_t is a true target at time t and σ_t is the measurement noise.

B. Exact Arithmetic Adaptive Filter

Fig. 1 shows an adaptive filter used for system identification [4]. The adaptive filter consists of the two types of computations: the estimated system output, $\hat{y}_t$, and the weight updates. The $\hat{y}_t$ is estimated by the inner product between $\hat{\mathbf{w}}_{t-1}$ and $\mathbf{x}_t$ at time t. The weights are updated using the information of the estimation error, e_t, every time step. The measurement from the system output $\dot{y}_t$ is subject to inevitable measurement imperfection (i.e., measurement noise σ_t) at time t.

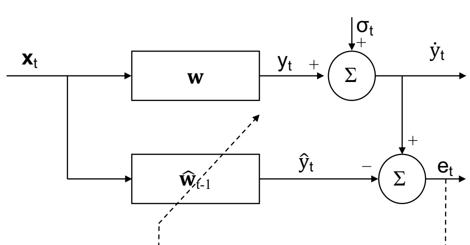

Fig. 1. Exact Arithmetic Adaptive Filter Model [4]

C. Forward Error Analysis of Adaptive Filters

Fig. 2 represents the exact arithmetic model for an adaptive filter implemented with finite precision arithmetic, interpreted by forward error analysis. There are the three rounding error terms in the figure: $\sigma_t^{(f1)}$, $\sigma_t^{(f2)}$, and $\sigma_t^{(f3)}$. These terms represent the rounding error from the estimation computation of $\tilde{\mathbf{w}}_{(t-1)}^T\mathbf{x}_t$, the rounding error from the subtraction, and the rounding error from the weight updates respectively. Notice that in Fig. 2, the weights, $\tilde{\mathbf{w}}_t$, have been updated at time t with implicit rounding errors (i.e., all rounding errors are located between computational components.). Therefore, such forward error analysis was used in signal processing community to identify the numerical stability of individual computing components in [14], [15], [17].

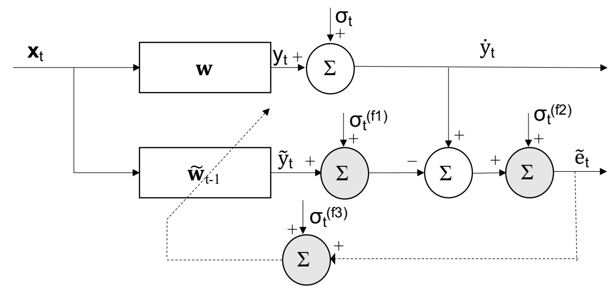

Fig. 2. Finite Precision Adaptive Filter Model by Forward Error Analysis

Such forward error analysis models are limited in understanding the mutual impacts between the rounding errors and the external variables of the system. In order to understand those mutual impacts, the rounding error component should be explicitly described on an RLS adaptive filter system.

D. Backward Error Analysis of Adaptive Filters

The application of backward error analysis to adaptive filters can make the rounding error analysis model simpler and more useful, providing a way to interpret internal rounding errors as external input noise. Our backward analysis framework of adaptive filters requires zero matrices for E and $\mathbf{X}_t^T\dot{\mathbf{y}}_t$ for $\mathbf{f}$ in Definition 1. Then, backward error analysis can describe the weights, $\tilde{\mathbf{w}}_t$, of an adaptive filter implemented with finite precision arithmetic as

$$(\mathbf{X}_t^T\mathbf{X}_t)\tilde{\mathbf{w}}_t = \mathbf{X}_t^T\dot{\mathbf{y}}_t + \Delta_{\mathbf{X}_t^T\dot{\mathbf{y}}_t}, \tag{19}$$

where $\Delta_{\mathbf{X}_t^T\dot{\mathbf{y}}_t}$ represents a vector generated by rounding errors, and $\tilde{\mathbf{w}}_t$ represents the perturbed weights to make Eq. (19) hold. We assume that

- $\mathbf{X}_t^T\mathbf{X}_t$ and $\mathbf{X}_t^T\dot{\mathbf{y}}_t$ are explicitly known,
- $\mathbf{X}_t^T\mathbf{X}_t$ is not a singular matrix.

We can simply find $\Delta_{\mathbf{X}_t^T\dot{\mathbf{y}}_t} = \mathbf{X}_t^T(\mathbf{X}_t\tilde{\mathbf{w}}_t - \dot{\mathbf{y}}_t) = -\mathbf{X}_t^T\mathbf{r}_t$, where $\mathbf{r}_t = \dot{\mathbf{y}}_t - \mathbf{X}_t\tilde{\mathbf{w}}_t$. Notice that we employ the two notations for the weights: $\hat{\mathbf{w}}_t$ for the exact weights with data corrupted only by measurement noise and $\tilde{\mathbf{w}}_t$ for the exact weights with data corrupted by both measurement noise and rounding error noise.

Fig. 3 represents the exact arithmetic model used for the adaptive filter implemented with finite precision arithmetic, interpreted by backward error analysis. There exists only one rounding error term in the figure: $\delta\dot{y}_t$. More importantly, the rounding error's effects on the filter performance are indicated outside computational components so that they can be interpreted as another external noise in the system; we name $\delta\dot{y}_t$ as *rounding error noise*. In this rounding analysis model, the weights are updated using exact arithmetic from the input

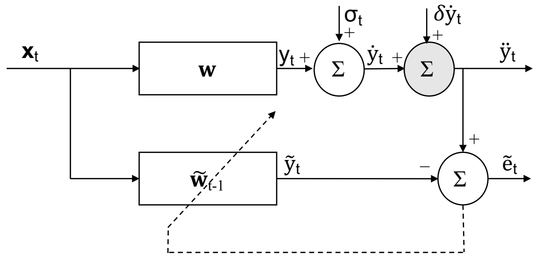

Fig. 3. Finite Precision Adaptive Filter Model by Backward Error Analysis

samples $\mathbf{X}_t = [\mathbf{x}_1, ..., \mathbf{x}_t]^T$ with the perturbed observations, $\ddot{\mathbf{y}}_t = \dot{\mathbf{y}}_t + \delta\dot{\mathbf{y}}_t = [\dot{y}_1 + \delta\dot{y}_1, ..., \dot{y}_t + \delta\dot{y}_t]^T$. By linearity, we can set $\delta\dot{\mathbf{y}}_t = [\delta\dot{y}_1, ..., \delta\dot{y}_t]^T$. Replacing $\mathbf{X}_t^T\dot{\mathbf{y}}_t$ with $\mathbf{X}_t^T\ddot{\mathbf{y}}_t$ in Eq. (2) yields

$$(\mathbf{X}_t^T\mathbf{X}_t)\tilde{\mathbf{w}}_t = \mathbf{X}_t^T\dot{\mathbf{y}}_t + \mathbf{X}_t^T\delta\dot{\mathbf{y}}_t. \tag{20}$$

To make Eq. (20) identical to Eq. (19) for the existence of the rounding error noise, Eq. (21) should be hold.

$$\Delta_{\mathbf{X}_t^T\dot{\mathbf{y}}_t} = \mathbf{X}_t^T\delta\dot{\mathbf{y}}_t. \tag{21}$$

Such rounding error noise exists as shown in Numerical Property 1.

Numerical Property 1. *Existence of Rounding Error Noise The rounding error noise, $\delta\dot{y}_t$ making Eq. (21) hold exists, and $\mathbf{X}_t^T\delta\dot{\mathbf{y}}_t = -\mathbf{X}_t^T\mathbf{r}_t$ if $\|\mathbf{r}_t\|_2$ is bounded.*

Proof. From Eq. (2), using the properties of $\mathbf{X}_{t-1}^T\dot{\mathbf{y}}_{t-1} + \dot{y}_t\mathbf{x}_t = \mathbf{X}_t^T\dot{\mathbf{y}}_t$ and $\hat{\mathbf{w}}_{t-1} = (\mathbf{X}_{t-1}^T\mathbf{X}_{t-1})^{-1}\mathbf{X}_{t-1}^T\dot{\mathbf{y}}_{t-1}$ can find $\hat{\mathbf{w}}_t$ with a recursive relation:

$$\begin{aligned}\hat{\mathbf{w}}_t &= (\mathbf{X}_t^T\mathbf{X}_t)^{-1}(\mathbf{X}_{t-1}^T\dot{\mathbf{y}}_{t-1} + \dot{y}_t\mathbf{x}_t)\\ &= (\mathbf{X}_t^T\mathbf{X}_t)^{-1}((\mathbf{X}_{t-1}^T\mathbf{X}_{t-1})\hat{\mathbf{w}}_{t-1} + \dot{y}_t\mathbf{x}_t)\\ &= (\mathbf{X}_t^T\mathbf{X}_t)^{-1}((\mathbf{X}_t^T\mathbf{X}_t - \mathbf{x}_t\mathbf{x}_t^T)\hat{\mathbf{w}}_{t-1} + \dot{y}_t\mathbf{x}_t)\\ &= \hat{\mathbf{w}}_{t-1} + (\dot{y}_t - \hat{\mathbf{w}}_{t-1}^T\mathbf{x}_t)\mathbf{P}_t\mathbf{x}_t.\end{aligned} \tag{22}$$

The application of backward error analysis to an RLS updates the weights, $\tilde{\mathbf{w}}_t$, in a system perturbed by the rounding error noise, $\delta\dot{y}_t$:

$$(\mathbf{X}_t^T\mathbf{X}_t)\tilde{\mathbf{w}}_t = \mathbf{X}_t^T\dot{\mathbf{y}}_t - \mathbf{X}^T\mathbf{r}_t. \tag{23}$$

From Eq. (23), the computed weights are represented:

$$\begin{aligned}\tilde{\mathbf{w}}_t &= \mathbf{P}_t(\mathbf{X}_{t-1}^T\dot{\mathbf{y}}_{t-1} + \dot{y}_t\mathbf{x}_t - \mathbf{X}^T\mathbf{r}_t)\\ &= \mathbf{P}_t((\mathbf{X}_{t-1}^T\mathbf{X}_{t-1})\tilde{\mathbf{w}}_{t-1} + \dot{y}_t\mathbf{x}_t) - \mathbf{P}_t\mathbf{X}^T\mathbf{r}_t\\ &= \tilde{\mathbf{w}}_{t-1} + (\dot{y}_t - \tilde{\mathbf{w}}_{t-1}^T\mathbf{x}_t)\mathbf{P}_t\mathbf{x}_t - \mathbf{P}_t\mathbf{X}^T\mathbf{r}_t.\end{aligned} \tag{24}$$

Now, let us derive the weights, $\tilde{\mathbf{w}}_t$, from Fig. 3:

$$\begin{aligned}\tilde{\mathbf{w}}_t &= \mathbf{P}_t(\mathbf{X}_{t-1}^T\ddot{\mathbf{y}}_{t-1} + \ddot{y}_t\mathbf{x}_t)\\ &= \mathbf{P}_t(\mathbf{X}_{t-2}^T\ddot{\mathbf{y}}_{t-2} + \ddot{y}_{t-1}\mathbf{x}_{t-1} + \ddot{y}_t\mathbf{x}_t)\\ &\ \ \vdots\\ &= \mathbf{P}_t(\Sigma_{i=1}^t\ddot{y}_i\mathbf{x}_i)\\ &= \mathbf{P}_t(\Sigma_{i=1}^t(\dot{y}_i\mathbf{x}_i) + \Sigma_{i=1}^t(\delta\dot{y}_i\mathbf{x}_i))\\ &= \mathbf{P}_t(\mathbf{X}_t^T\dot{\mathbf{y}}_t + \mathbf{X}_t^T\delta\dot{\mathbf{y}}_t)\\ &= \tilde{\mathbf{w}}_{t-1} + (\dot{y}_t - \tilde{\mathbf{w}}_{t-1}^T\mathbf{x}_t)\mathbf{P}_t\mathbf{x}_t + \mathbf{P}_t\mathbf{X}_t^T\delta\dot{\mathbf{y}}_t.\end{aligned} \tag{25}$$

Therefore, the application of backward error analysis to the adaptive filter is valid under the numerical condition:

$$\mathbf{X}_t^T\delta\dot{\mathbf{y}}_t = -\mathbf{X}_t^T\mathbf{r}_t. \tag{26}$$

Numerous vectors satisfy $\delta\dot{\mathbf{y}}_t$ to make Eq. (26) hold. Out of the vectors, the rounding error noise corresponds to a vector that has the minimum 2 norm. □

Numerical Property 2. *Magnitude of Backward Error Backward error depends on the applied arithmetic precision, the numerical stability of an algorithm, and the feature size [26]. For example, $\|\mathbf{X}_t^T\mathbf{r}_t\|_2 = \|\mathbf{X}_t^T\delta\dot{\mathbf{y}}_t\|_2 \leq c(N)c(\mathcal{A})\epsilon_{mach}\|\mathbf{X}_t^T\dot{\mathbf{y}}_t\|_2$, where $c(N)$ is a positive value depending on a feature size, N, and $c(\mathcal{A})$ is a positive value depending on the numerical stability of the implementation algorithm, $\mathcal{A}$. For the proof, please refer to [18], [30].*

We denote $\|\mathbf{X}_t^T\delta\dot{\mathbf{y}}_t\|_2$ as the magnitude of rounding error noise from this point forward. Similarly, we denote $\|\mathbf{X}_t^T\boldsymbol{\sigma}_t\|_2$ as the magnitude of measurement error noise. More precisely, those magnitudes are error quantities projected onto input feature space, $\mathbf{X}_t$ [26].

Numerical Property 3. *Independence of Backward Error from Measurement Noise The backward error, $\eta_{ALG}(\tilde{\mathbf{w}}_t)$, is independent from measurement noise for adaptive filters.*

Proof. In our backward analyses for RLS, the backward error for a least squares is converted to the backward error for a linear system of Eq. (2) in Definition 1 as the targets containing the measurement noise, $\dot{\mathbf{y}}_t$, are projected onto the reduced dimensional subspace (i.e., $\dot{\mathbf{y}}_t \in \mathbb{R}^{t\times 1} \rightarrow \mathbf{X}_t^T\dot{\mathbf{y}}_t \in \mathbb{R}^{N\times 1}$, where $t \geq N$.). As consequence, the backward error, with our setting of $E = \mathbf{0}$ and $\mathbf{f} = \mathbf{X}_t^T\dot{\mathbf{y}}_t$ in Definition 1, can be expressed as

$$\begin{aligned}\eta_{ALG}(\tilde{\mathbf{w}}_t) &= \frac{\|\mathbf{X}_t^T\dot{\mathbf{y}}_t - \mathbf{X}_t^T\mathbf{X}_t\tilde{\mathbf{w}}_t\|_2}{\|\mathbf{X}_t^T\dot{\mathbf{y}}_t\|_2}\\ &= \frac{\|\mathbf{X}_t^T\dot{\mathbf{y}}_t - \mathbf{X}_t^T\mathbf{X}_t(\mathbf{w}_t + \delta\mathbf{w}_t)\|_2}{\|\mathbf{X}_t^T\dot{\mathbf{y}}_t\|_2}\\ &= \frac{\|\mathbf{X}_t^T\mathbf{X}_t(\delta\mathbf{w}_t)\|_2}{\|\mathbf{X}_t^T\dot{\mathbf{y}}_t\|_2},\end{aligned} \tag{27}$$

where $\delta\mathbf{w}_t$ is the error in the weights caused by rounding error in solving the system: $\mathbf{X}_t^T\mathbf{X}_t\mathbf{w}_t = \mathbf{X}_t^T\dot{\mathbf{y}}_t$. Therefore, regardless of which system used in Eq. (27), the backward error mainly depends on the relative rounding error in solving the system, implying that the backward error is independent of measurement noise, since the backward error is generated in solving the system already corrupted by the measurement noise due to finite precision arithmetic. If exact arithmetic is applied, no backward error exists in Eq. (27) regardless of the measurement noise. Therefore, the backward error is independent of the measurement noise. □

Using the property that the magnitudes of rounding error noise are equal to the magnitudes of backward error multiplied by $\|\mathbf{X}_t^T\dot{\mathbf{y}}_t\|_2$, along with Numerical Properties 1, 2, and 3, we are led to the following corollaries.

Corollary 1. *Rounding error noise is independently additive to measurement noise in the adaptive filter.*

Corollary 2. *If the magnitude of the rounding error noise is relatively lower than the measurement noise, employing lower precision arithmetic does not affect the performance of the adaptive filter.*

Corollary 3. *Applying lower precision arithmetic to the adaptive filter requires a higher regularisation if the magnitude of rounding error noise is higher than the magnitude of measurement noise.*

Corollary 4. *Rounding error noise is not white noise. It makes the method of least-squares a biased estimator even though measurement noise is white noise.*

E. Role of Regularisation on Backward Error Analysis

Applying a regularisation to adaptive filters validates the assumptions we made in III-D for the application of backward error analysis to adaptive filters. First, regularisation can prevent the correlation matrix, $(\mathbf{X}_t^T\mathbf{X}_t)$, from being a singular matrix by increasing the values of the diagonal elements. Second, regularisation makes $\|\mathbf{r}_t\|_2$ bounded in practice [31] so that the computed value of $\tilde{\mathbf{w}}_t$ can be assessed by the backward error analysis framework. Notice that our backward error analysis of RLS adaptive filters deviates from that in common usage by the numerical linear algebra community. For example, the residual measurement from our backward error analysis after applying a regularisation, λ, is: $(\mathbf{X}_t^T\mathbf{X}_t)\hat{\mathbf{w}}_t - \mathbf{X}_t^T\dot{\mathbf{y}}_t$ instead of $(\mathbf{X}_t^T\mathbf{X}_t + \lambda\mathbf{I})\hat{\mathbf{w}}_t - \mathbf{X}_t^T\dot{\mathbf{y}}_t$, even though an adaptive filter employs a non-zero λ. Such deviation makes it possible to interpret λ as a regularisation parameter to suppress both the rounding error noise and the measurement noise effects to filter performance. Therefore, employing regularisation generates the bias that makes the system deviate from the original system. For example, a higher λ deviates the system further from the original system, resulting in a higher bias. In this case, the rounding error noise magnitude in Numerical Property 2 should consider the effects from employing a regularisation parameter. This leads to:

$$\|\mathbf{X}_t^T\delta\dot{\mathbf{y}}_t\|_2 \leq c(N)c(\mathcal{A},\lambda)(1+\alpha|bias(\lambda)|)\epsilon_{mach}\|\mathbf{X}_t^T\dot{\mathbf{y}}_t\|_2, \tag{28}$$

where $\alpha \geq 0$. With our backward error analysis framework, a trade-off exists in the magnitude of rounding error noise according to the magnitude of a regularisation. For example, employing a lower regularisation can either decrease the magnitude of rounding error noise by minimising the bias (i.e., $|bias(\lambda)| \propto \lambda$.) or increase the magnitude of rounding error noise by making an algorithm less numerically stable (i.e., $c(\mathcal{A},\lambda) \propto \lambda^{-1}$.). Therefore, the optimal value of λ minimises the rounding error noise in Eq. (28), and the approximated optimal λ can be sought using a statistic approach such as cross-validation [32]. Our work constructs a mathematical model that formally explains how the rounding error interacts with other types of errors caused by measurement noise, the algorithm used for the implementation, and the regularisation employed for the system. The optimal λ can experimentally be determined using cross-validation, but it is possible to evaluate the impact of rounding errors on the optimal λ within our model, which was not previously feasible.

F. Bias-Variance Decomposition with Rounding Error Noise

To explore the effects of rounding error noise on the bias-variance decomposition of [33], we consider a simple linear regression that the targets, $\dot{y}_t$, which are generated based on a linear model with white noise: $\dot{y}_t = \mathbf{w}^T\mathbf{x}_t + w_t$, where $\mathbf{w}$ are true system weights and w_t is white noise. Including rounding error noise, Eq. (18) should be modified as follows:

$$\dot{y}_t + \delta\dot{y}_t = \tilde{\mathbf{w}}_{t-1}^T\mathbf{x}_t + \tilde{e}_t. \tag{29}$$

Our task, essentially, is making a guess at $\mathbf{w}$ with $\tilde{\mathbf{w}}_t$ in Eq. (29) using the dataset D having n samples: $D = \{(\mathbf{x}_1, \dot{y}_1), ..., (\mathbf{x}_n, \dot{y}_n)\}$. When the magnitude of measurement noise is lower than rounding error noise with lower precision arithmetic, a regularisation chosen by cross-validation, λ_{CV}, is increased with lower precision arithmetic. In such case, lower precision arithmetic increases the bias due to the increased regularisation. However, when the magnitude of rounding error noise is lower than measurement noise, the optimal regularisation parameter is determined mainly by measurement noise. Since lower precision arithmetic increases the rounding error noise, the regularisation needs to consider both measurement noise and rounding error noise for lower precision adaptive filters. For example, Eq. (30) employs a regularisation, λ, derived from both measurement noise, $\boldsymbol{\sigma}_t$, and rounding error noise, $\delta\dot{\mathbf{y}}_t$:

$$(\mathbf{X}_t^T\mathbf{X}_t + \lambda\mathbf{I})\tilde{\mathbf{w}}_t = \mathbf{X}_t^T\mathbf{y}_t + \mathbf{X}_t^T\boldsymbol{\sigma}_t + \mathbf{X}_t^T\delta\dot{\mathbf{y}}_t. \tag{30}$$

Based on Eq. (30), the λ regulates the effects of both measurement noise and rounding error noise on filter performance, resulting in the decreased variance in the bias-variance decomposition. In other words, the optimal λ minimises the overfitting occurred when the weights tend to be fitted to the noise rather than the original data; without noise (i.e., $\boldsymbol{\sigma}_t = \mathbf{0}$, $\delta\dot{\mathbf{y}}_t = \mathbf{0}$), the optimal value for λ should be zero since there is no need to deal with overfitting.

IV. Experimental Framework

We experiment on the impact of lower precision on adaptive RLS filters in terms of the regularisation and the filter performance, and compare experimental results with the theoretical numerical properties and corollaries. We aim to evaluate the impact of lower precision arithmetic on MIL-RLS and QR-RLS adaptive filter performances.

A. Experimental Setup

The experimental setting is as follows:

- Source code: MATLAB
- Algorithms: MIL-RLS and Givens rotation QR-RLS (e.g., the implementation of [23]). In our setting, the forgetting factor equals ‘1’ as discussed in Section I and the regularisation parameters, λs, used for the experiments range from 2^{-50} to 1.

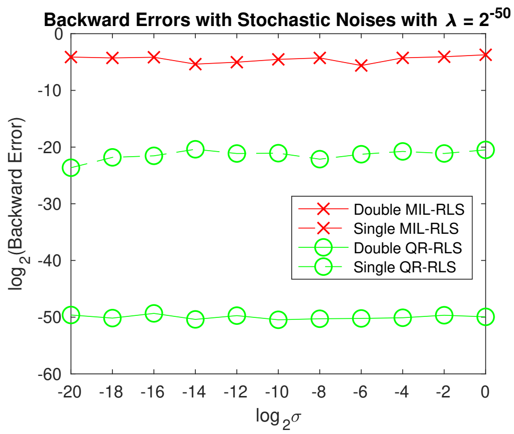

Fig. 4. Backward Errors according to σs when $\lambda = 2^{-50}$

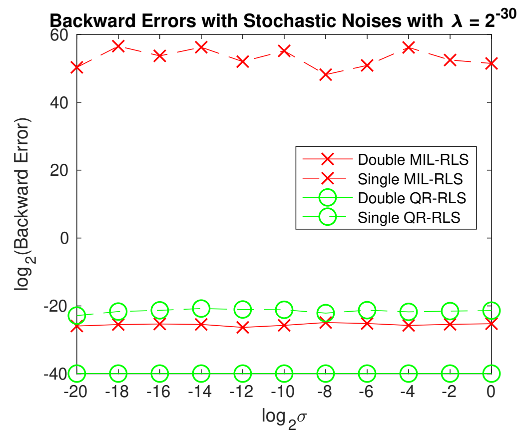

Fig. 5. Backward Errors according to σs when $\lambda = 2^{-30}$

- Problem: System identification of $\mathbf{w}$ in Eq. (31) with data $\{\mathbf{x}_i, y_i\}$ with the feature size, $N = 10$, where $i = 1, 2, ..., n$.

$$y_i = \mathbf{w}^T \mathbf{x}_i. \tag{31}$$

The target weights $\mathbf{w}$ are generated with standard normal distribution. The $11,000$ data samples with $N = 10$ are generated with the standard normal distribution for training. The early $1,000$ samples are used for 10-fold cross-validation [32] to choose an adequate regularisation parameter according to arithmetic precision, numerical stability of an algorithm, and measurement noise (i.e., for each fold, 900 data samples are used for training and 100 data samples are used for validation.). For testing, another $1,000$ samples are generated. White Gaussian noise with various standard deviations, σs, as a measurement noise has been used to generate the noisy targets.

B. Independence of Backward Errors from Measurement Noise

We will show empirically that the magnitudes of backward errors are independent with respect to measurement noise, supporting Numerical Property 3. The backward error magnitudes are measured based on Eq. (32) using double precision arithmetic.

$$\eta_{ALG}(\tilde{\mathbf{w}}_t) = \frac{\|\mathbf{X}_t^T(\dot{\mathbf{y}}_t - \mathbf{X}_t\tilde{\mathbf{w}}_t)\|_2}{\|\mathbf{X}_t^T\dot{\mathbf{y}}_t\|_2}. \tag{32}$$

The magnitudes of the backward errors are described in Fig. 4, 5, 6, and 7 according to the magnitudes of measurement noise (i.e., stochastic noise), with the $\lambda = 2^{-50}$, 2^{-30}, 2^{-15}, and 2^{-7} respectively. We used 1000 training samples for the experiments. In the figures, the magnitudes of backward errors are independent of measurement noise that corresponds to Numerical Property 3.

When the magnitude of the regularisation is minuscule in Fig. 4, the magnitudes of the backward errors are 2^{-50} for double precision QR-RLS and 2^{-22} for single precision QR-RLS, while 2^{-4} for double precision MIL-RLS and ∞ (unbounded) for single precision MIL-RLS. For the numerically stable QR-RLS, the rounding error noise magnitude

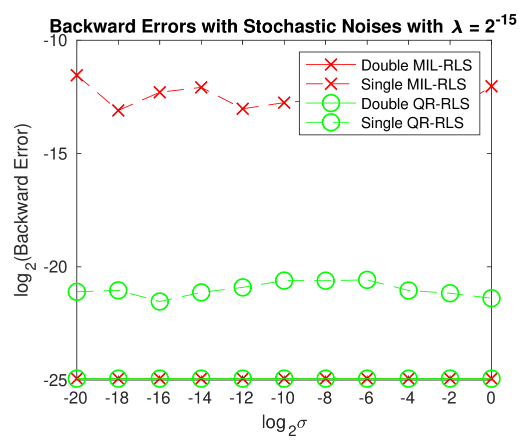

Fig. 6. Backward Errors according to σs when $\lambda = 2^{-15}$

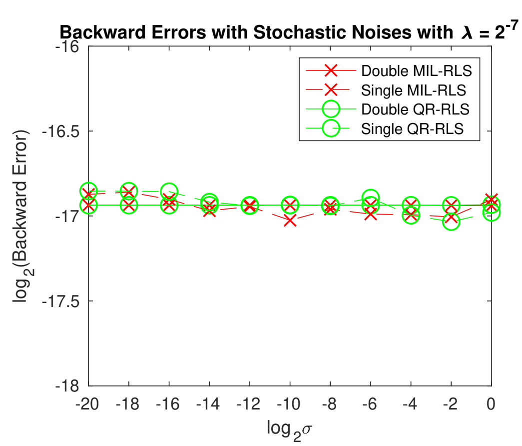

Fig. 7. Backward Errors according to σs when $\lambda = 2^{-7}$

gap between single and double precision is approximately the machine epsilon gap between single and double precision (i.e., $log_2\epsilon_{double} - log_2\epsilon_{single} = -29$). In Fig. 5, increasing the value of λ to 2^{-30} makes the backward errors decrease for MIL-RLS, while it increases for double precision QR-RLS. Now, the backward error for single precision MIL-RLS is bounded, and the magnitudes of backward errors for single precision QR-RLS are equivalent to double precision MIL-RLS. Compared to Fig. 4, it is the bias increment by the increased λ that lets the magnitudes of the backward errors for double precision QR-RLS increase. In contrast, the backward error of MIL-RLS is decreased by improving the numerical stability of the algorithm with higher regularisation. For example, higher regularisation makes $c(\mathcal{A})$ in Numerical Property 2 lower for MIL-RLS but does not affect $c(\mathcal{A})$ for QR-RLS in practice. In Fig. 6, employing a higher $\lambda = 2^{-15}$ lets the backward errors of double precision MIL-RLS go down further, and become equivalent to double precision QR-RLS. The magnitudes of the backward errors have increased for double precision QR-RLS compared to $\lambda = 2^{-30}$, while remained equivalent for single precision QR-RLS. Finally, the backward errors are equivalent for all cases when $\lambda = 2^{-7}$ in Fig. 7. When we increase the value of the λ further, the magnitudes of the backward errors increase, keeping the magnitudes of the backward errors equivalent among all the four cases.

C. Impact of Lower Precision on Regularisation

We use 10-fold cross-validation [32] to seek λ_{CV}s (i.e., regularisation parameters chosen by cross-validation) using $1,000$ data samples. In our 10-fold cross-validation setup, the $1,000$ data samples are divided evenly into 10 data groups, $G_1, ..., G_{10}$, where each group has 100 data samples. The filter performance with a particular λ is evaluated at each run in the cross-validation using the data in each group (e.g., G_1) in terms of the Mean Squared Error (MSE) after the filter is trained using the data from the other 9 groups (e.g., $G_2, ..., G_{10}$). Therefore, the filter performance with a particular value of λ is evaluated 10 times using the data from G_1 to G_{10}. The MSEs from 10 different runs are then averaged out to provide one representative MSE for the filter performance with the λ. The λ having the minimum average MSE is finally chosen for the λ_{CV} after the filter performance is evaluated with all λs in the hyperparameter set. We first seek λ_{CV} variation according to the 11 different types of measurement noise having standard deviation from 2^{-20} to 2^0 with a multiplicative scale of 2^2. We perform 30 times cross-validations with 30 times measurement noise generation per each standard deviation in order to seek the average regularisation parameter chosen by cross-validation. The regularisation parameter set for cross-validation also ranges from 2^{-20} to 2^0 with a multiplicative scale of 2^2.

1) Relation between Rounding Error Noise and Optimal Regularisation Parameter: Fig. 8 shows the average optimal regularisation parameters chosen by the cross-validation according to measurement noise with standard error bars. As expected, the regularisation parameter chosen by cross-validation increases in proportion to the measurement noise for all algorithms using double precision arithmetic. The relation between the optimal regularisation parameter and the rounding error noise can be observed with fixing the measurement noise. For example, when the measurement noise is fixed to 2^{-12}, single precision arithmetic having larger rounding error noise requires a higher regularisation parameter over double precision arithmetic for both MIL-RLS and QR-RLS. This supports Corollary 3 empirically; while fixing the measurement noise, higher rounding error noise requires a higher regularisation. The regularisation parameters are identical for the two double precision RLS algorithms; double precision arithmetic is excessively high for the two algorithms compared to measurement noise. Single precision MIL-RLS keeps the regularisation parameter as 2^{-8} until the standard deviation of measurement noise reaches 2^{-6} since the regularisation of $\lambda \approx 2^{-7}$ makes the backward error minimised for single precision MIL-RLS in our cross-validation hyperparameter set (refer to Fig. 7). When the standard deviation of measurement noise is higher than 2^{-6}, the regularisation parameters are equivalent for all the four cases (with slight deviation from single precision MIL-RLS).

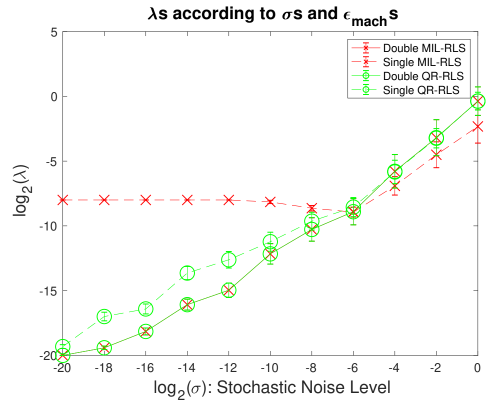

Fig. 8. The λs from Cross-Validation

D. Impact of Lower Precision on Learning Curves

This section will support Corollary 2 empirically. We explore the learning curves for data with and without measurement noise using a standard deviation between 2^{-20} and 2^{-14} inclusively. We employ regularisation parameters chosen from the cross-validation. Fig. 9 shows the learning curves from the two different types of algorithms employing $\lambda = 2^{-30}$ with the data without measurement noise. Here, the rounding error noise is only noise source for adaptive filter accuracy. The Mean Squared Errors (MSEs) should be zeros for linear regressions using exact arithmetic in our experimental setting when the number of training samples is higher than 10. The selected regularisation parameters minimise the effect of rounding error noise on adaptive filter performance. Double precision QR-RLS shows the best adaptive filter performance while single precision MIL-RLS degrades the performance significantly due to large rounding error noise. The magnitudes of the MSEs increase with single precision arithmetic for both QR-

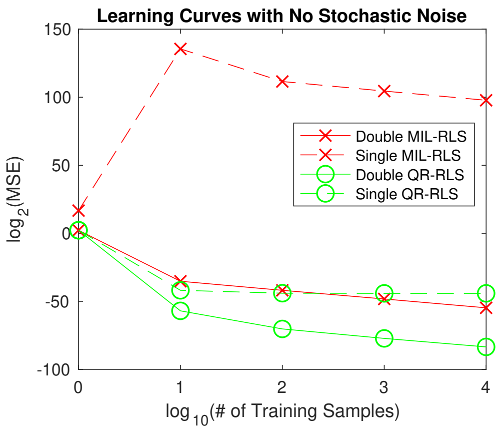

Fig. 9. Learning Curves without Measurement Noise

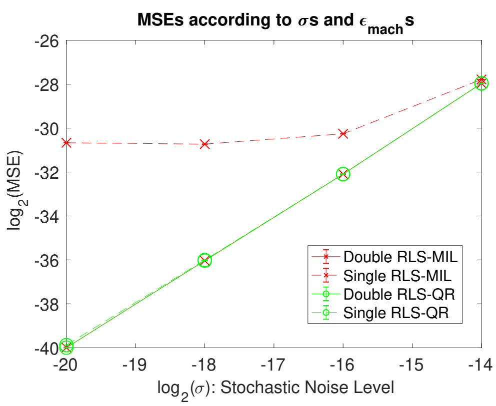

Fig. 10. MSEs from Cross-Validation

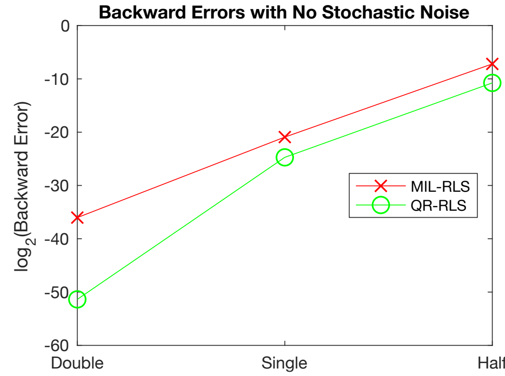

Fig. 11. Rounding Error Noises according to Precision Variation

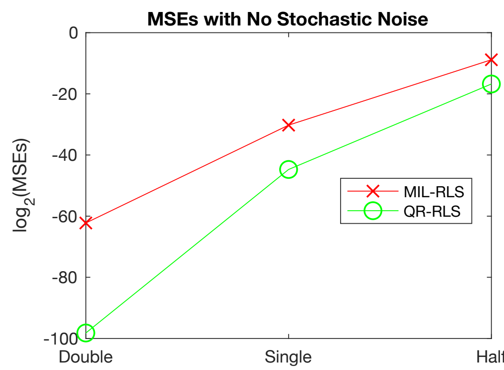

Fig. 12. Prediction MSEs according to Precision Variation

RLS and MIL-RLS algorithms due to the increased magnitude of rounding error noise, because rounding error noise is in proportion to a ϵ_{mach}. The MSEs follow the backward error magnitudes (i.e., rounding error noise magnitudes) in this case. Fig. 10 shows the MSEs from MIL-RLS and QR-RLS with the regularisation parameters selected from the cross-validation according to the level of measurement noise using validation data set. The MSEs measured with both double precision MIL-RLS and QR-RLS follow the variance of the measurement noise, σ^2s, since the magnitude of the measurement noise is higher than the rounding error noise. However, when the rounding error noise larger than the measurement noise in single precision MIL-RLS, rounding error noise affects the filter performance. When rounding error noise magnitude is lower than measurement noise for single precision MIL-RLS (i.e., $\sigma = 2^{-14}$ in Fig. 10), the MSEs become equivalent in all four cases. Therefore, the adaptive filter performance is equivalent regardless of arithmetic precision when measurement noise is higher than rounding error noise, supporting Corollary 2.

E. Impact of Lower Precision on Adaptive Filter Performance

Fig. 11 represents the magnitudes of backward errors for half, single and double precision arithmetic MIL-RLS and QR-RLS without measurement noise and with the regularisation parameters selected from cross-validation. As expected, the magnitudes of backward errors are in proportion to the machine epsilons. The backward errors of MIL-RLS are larger than QR-RLS since $c(\mathcal{A})$ is higher for MIL-RLS.

Fig. 12 represents adaptive filter performance (i.e., MSEs) variation for half, single and double precision arithmetic MIL-RLS and QR-RLS without measurement noise and with the regularisation parameters selected from cross-validation. As expected, the magnitudes of MSEs approximately follow the squares of the magnitudes of the backward error for QR-RLS. For a numerically unstable MIL-RLS, such tendency becomes weaker from a higher bias in the bias-variance decomposition due to a higher regularisation for MIL-RLS compared to QR-RLS.

V. Related Work

A. Application of Linear Algebra to Signal Processing

Backward error analysis of least squares was discussed for general cases in [16], [34]. In [35], linear algebra techniques were applied to signal processing applications and the backward error was discussed in terms of numerical stability of the algorithm.

B. Rounding Error Analysis of RLS

The work of [4] performed forward error analysis of the three individual computing components in MIL-RLS including inner product, weight correct term computation, and weight update computation, and showed that the weights of MIL-RLS were no longer unbiased weights under finite precision arithmetic even with a forgetting factor equal to '1' (i.e., rounding error noise is not white noise if this finding is interpreted with our backward error analysis.). The analysis in [4] also found that the weights update computation is numerically unstable when the forgetting factor approaches '1'. The numerical stability of RLS algorithms has been intensively investigated in [14], [15], [17]. Unlike other previous related work, this work explores the impact of lower precision on adaptive filter performance by introducing rounding error noise. The rounding error noise in this paper is different to the noise introduced in [4] and [17]. Both the "*floating point noise*" in [4] and the "*round-off noise*" in [17] were derived from forward error analysis, therefore such noise cannot be interpreted as explicit noise in the input data.

C. Regularisation for RLS

In [28], effective regularisation was studied in terms of the energy of the input data, the window length, and the echo to noise ratio. In order to improve numerical stability of RLS algorithms, various regularisation methods were proposed for linear regression such as ridge regression employing square 2-norm regularisation parameters [31], LASSO employing 1-norm regularisation parameters [36], and elastic net that combines the ridge regression and the LASSO [37].

D. Low Complexity/Precision for RLS

Recently, there have been many research attempts to improve computational efficiency for RLS [5]–[11], [38]. In [5], [6], a computationally efficient RLS algorithm was developed by extending the idea of Kronecker product decomposition proposed in [39]. The work of [5] performed the impulse response decomposition based on Kronecker product to transform a high dimension system identification problem to a low dimension problem, resulting in lower computational complexity of RLS compared to a conventional RLS algorithm. Such idea was particularly fitted to echo cancellation, and further extended to multiple input single output system identification in [7]. In [8], the idea of the impulse response decomposition for RLS was extended to a Kalman filter. In [9], a low complexity RLS algorithm was proposed by exploiting a dichotomous coordinate descent algorithm. The work of [9] developed a transversal RLS adaptive filter requiring only $3N$ multiplications rather than $O(N^2)$ multiplications required by a conventional RLS algorithm. In [10], the algorithm of [9] was implemented on FPGA, and the accuracy of the algorithm was accessed empirically according to variable arithmetic precision. In [11], a variation of RLS algorithm having computational complexity from $N^2/2$ to $5N^2/6$ was presented, but the algorithm was not suitable for lower precision fixed point arithmetic due to the accumulation of rounding errors. In [38], the clipped LMS and RLS algorithms, that quantise the input signals to $\{-1, 0, 1\}$, were evaluated in terms of accuracy and computational complexity, compared to other low-complexity RLS algorithm counterparts such as the signed regressor RLS, the M Max tap-selection RLS, and the original RLS. Based on the evaluations, the optimal step sizes and forgetting factors for the clipped LMS and RLS that minimise the weight misalignment were derived.

E. Comparison of Our Work to [22]

The work of [22] performed the rounding error propagation analysis with various adaptive filter algorithms by utilising the backward consistency concept. In [22], a conceptual backward consistency manifold was created to analyse the propagation of rounding error (i.e., forward error). An algorithm was referred to as a *backward consistent* algorithm "if the computed state remains on the manifold" (i.e., "if it always leads to a computed solution that can be interpreted as a perturbed problem with the required structure") in [22]. For example, if an algorithm is backward consistent, the rounding errors (e.g., the perturbation on the computed quantities) should not break the required structure of the problem (e.g., the symmetry and the positive definiteness of a $\mathbf{P}_t$ in an RLS). Even though the backward consistency concept was considered in [22], the backward errors (i.e., the perturbation on the external system rather than the computed quantities) were not explored. For example, in [22] "We focus here specifically on the propagation of the numerical errors; We assume that from a certain time instant onward no more round-off errors are made and we observe how the effect of the accumulated errors evolves in time from that point forward." Therefore, [22] explored the forward error rather than the backward error.

Fig. 13 describes the adaptive filter model used in [22]. The initial perturbation $\delta\mathbf{w}_{t-1}^{(0)}$ and $\Delta\mathbf{P}_{t-1}^{(0)}$ were generated due to the rounding errors until time $(t-1)$. The [22] evaluated how $\delta\mathbf{w}_{t-1}^{(0)}$ and $\Delta\mathbf{P}_{t-1}^{(0)}$ could propagate under exact arithmetic by using a conceptual backward consistency manifold. Notice that the rounding error propagation analysis in [22] focused on the errors in the computed quantities (e.g., the propagation of $(\tilde{\mathbf{w}}_{t-1}+\delta\mathbf{w}_{t-1}^{(0)})$ and $(\mathbf{P}_{t-1}+\Delta\mathbf{P}_{t-1}^{(0)})$ under exact arithmetic in Fig. 13) rather than the deviation from the external environment of the system.

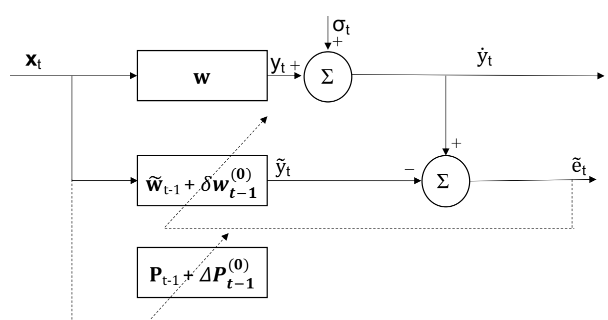

Fig. 13. Adaptive Filter Model in [22]

Several adaptive filter algorithms were examined using the backward consistency manifold in [22]. For example, the backward consistency was examined with a conventional Kalman filter employing Riccati equation for a covariance matrix update and a square-root Kalman filter employing the square-root algorithm for a covariance matrix update. Under the backward consistency requirement, the covariance matrix in a Kalman filter should be symmetric. The work of [22] decomposed the rounding error in the covariance matrix (e.g., $\Delta\mathbf{P}_{t-1}^{(0)}$ for an RLS) into symmetric and antisymmetric components. The symmetric part in the error does not affect the symmetry when it propagates under exact arithmetic, while the antisymmetric part propagates in both symmetric and antisymmetric components. In other words, the propagation of the symmetric part always travels within the backward consistency manifold, while the antisymmetric part could propagate both within the manifold and outside the manifold. Therefore, maintaining the backward consistency requires the antisymmetric component not to be magnified through propagation.

If we apply the work of [22] to MIL-RLS, the rounding error propagation using Eq. (6) can break the backward consistency of the symmetry of $\mathbf{P}_t$; the antisymmetric component can be magnified through the propagation of $\Delta\mathbf{P}_{t-1}^{(0)}$ when $\mathbf{P}_{t-1}\mathbf{x}_t \rightarrow \mathbf{0}$ and $\|\Delta\mathbf{P}_{t-1}^{(0)}\mathbf{x}_t\|_2 > \|\mathbf{P}_{t-1}\mathbf{x}_t\|_2$ in Eq. (6). For QR-RLS, the backward consistency of the symmetry will be kept, since QR-RLS stores only a triangular matrix factorised by QR decomposition (e.g., the antisymmetric component after the rounding error propagation will be removed by keeping a triangular matrix only.).

As mentioned by the example above, [22] used the backward consistency concept (i.e., the backward consistency manifold) to explain the effects of the rounding error propagation on the backward consistency, improving the explainability of the rounding error propagation analysis. In contrast, our work sets up a new diagram in which rounding errors can be interpreted as the explicit noise that sits on equal footing with the measurement noise as shown in Fig. 3. To the best of our knowledge, the aim of [22] was to enhance the explainability of the rounding error propagation in terms of the backward consistency, while our aim is to provide a guide on how to apply lower precision arithmetic to adaptive filters. Since the research aims are different, the approach and the findings of our work are different to [22].

The summary of the comparison of our work to [22] is as follows:

1) Research Method: Our aim is to provide a guide on how to apply lower precision arithmetic to RLS filters. Therefore, we derive the rounding error noise explicitly that sits on equal footing with measurement noise from our backward error analysis first, and then seek the mutual influence of the explicit rounding error noise on another filter explicit variables such as the measurement noise, the regularisation, the arithmetic precision, and numerical stability of the implementation algorithm later. In contrast, since [22] seeks to enhance the explainability of rounding error propagation in terms of the backward consistency, it derives the backward consistency manifold from the concept of backward consistency first and then explains the backward consistency conditions of rounding error propagation using the backward consistency manifold later.

2) Backward Error: Our work seeks the backward error of the RLS adaptive filters. We derived the rounding error noise from the backward errors. In contrast, the [22] did not seek backward errors; the concept of backward error consistency was employed to derive the backward consistency manifold.

3) Derived Object (DO): Our work derives the rounding error noise: the rounding error noise is the noise imposed on the target in the system, incurred by the rounding errors. In contrast, [22] derived the backward consistency manifold. This backward consistency manifold is a conceptual manifold that consists of the set of the computed quantities satisfying the backward consistency.

4) Exploitation of DO: Our work uses the rounding error noise to explore its dependence on the measurement noise, the arithmetic precision, the numerical stability of the implementation algorithm, the regularisation, and the feature size. In contrast, the backward consistency conditions were explained with the backward consistency manifold for various adaptive filters in [22].

5) Findings from DO: The findings from our work are as follows:

- Applying lower precision to adaptive filters requires a higher regularisation to abate the rounding error noise effects on performance; a higher regularisation is recommended for lower precision adaptive filters.
- The rounding error noise is not white Gaussian noise; the rounding error noise dominantly affects adaptive filter performance if the measurement noise follows white Gaussian noise.
- The rounding error noise is independent of measurement noise; lower precision can be applied to RLS without losing performance unless the magnitude of rounding error noise exceeds the magnitude of measurement noise.

In contrast, the findings from [22] are as follows:

- The errors in the computed quantities can be decomposed into the tangential and the normal component of the backward consistency manifold.
- The error in the tangential component of the manifold propagates only within the manifold under exact arithmetic, while the error in the normal component of the manifold can propagate in both tangential and normal direction of the manifold.
- If the error in the normal component of the manifold is magnified through the propagation of the error, the algorithm can break the backward consistency condition.

6) Main Contribution: Our work provides a guide on how to apply a lower precision arithmetic to adaptive filters. In contrast, [22] enhanced the interpretation of the propagation of the rounding errors.

Therefore, our work is not a simple application of [22] and is instead orthogonal in terms of research aims, methods, and findings. For example, we explore the backward error (i.e., the deviation of the external system) while [22] did not seek the backward error. Therefore, our work can evaluate the mutual impacts between the explicit rounding error noise and the external system variables such as the measurement noise and the regularisation. The evaluation of such mutual impacts was not a straightforward consequence of the techniques described in [22].

VI. Discussion

The motivation of this work comes from the desire to provide the signal processing community with (i) a simpler rounding error analysis framework for adaptive filters and (ii) a guide on how to apply lower precision to adaptive filters in order to improve battery life, hardware resource utilisation, and computational throughput at the same time. This paper can be an initiative work exploring the answers for the question: "What if we apply considerably low precision to adaptive filters?" More specifically,

- How could the filter performance be affected with lower precision arithmetic?
- Can the degraded filter performance due to lower precision arithmetic improve with a higher regularisation or a lower regularisation?

This paper is the first to present the adaptive filter system model including rounding error components that sits on equal footing with the measurement noise, and by representing rounding errors *explicitly*, it is possible to describe the mutual impacts between the explicit rounding error components (i.e., rounding error noise) and other external variables (e.g., the measurement noise and the optimal regularisation), and the impact of the stability of the algorithm used for the implementation on the rounding error noise. It was not feasible to describe such impacts from previous rounding error analysis models, since the rounding error components in their models were located between computing components. We believe that our model and both the theoretical and empirical evaluation of the model in a number of RLS can help to answer such questions; our findings are described by Numerical Properties with Corollaries that can be used as a guide on how to apply lower precision arithmetic to adaptive filters without degrading filter performance significantly compared to higher precision arithmetic. For example, when a new optimal regularisation parameter needs to be found for a lower precision arithmetic RLS, only higher regularisation parameters can be considered for the hyperparameter set in cross-validation, since lower precision arithmetic generates larger rounding error noise.

Noise is represented as uncertainty of data due to various circumstances. For example, measurement noise comes from measurement, under-modelling noise comes from the limitation of model complexity over data complexity, and rounding error noise comes from finite precision arithmetic. We claim that the three types of noise including measurement, under-modelling, and rounding error noise should be considered for low precision adaptive filters since rounding error noise can be the main factor affecting performance, and the three types of noise are actual noise after implementing an adaptive filter on a finite precision computing machine.

Ljung presented his opinion on future system identification in the next 40 years in an interview [40]: "We will use particle filters and other techniques like that, Monte Carlo simulations, to estimate nonlinear functions in a non-parametric way, so you can approach the problem without any prejudice about the structures you are going to see." Our findings in numerical properties and corollaries are currently limited to RLS filters but can play the role of a cornerstone assisting future system identification development in low precision arithmetic. For example, kernel-based adaptive filters have drawn attention of the signal processing community for non-linear system identifications [41]. A kernel-based RLS described in [42] is equivalent to the least squares algorithm (i.e., orthogonal projection of data to a reduced dimensional space), but with infinite dimension feature mapping. This kernel RLS maps finite dimensional data to infinite dimensional data and formulates a kernel matrix instead of the correlation matrix. Therefore, our backward error analysis can be applied to nonlinear system identification using kernel RLS, but with a kernel matrix instead of $(\mathbf{X}_t^T\mathbf{X}_t)$ in Eq. (19). Our future work includes extending this work for non-linear system identification.

VII. Conclusion

The aim of this paper is to provide the signal processing community with (i) a simpler rounding error analysis framework for adaptive filters and (ii) a guide on how to apply lower precision to adaptive filters, given constraints such as hardware budget, power cap, computational throughput, and filter performance. To the best of our knowledge, this paper is the first that formulates rounding errors as external noise by applying the backward error analysis to adaptive filters. This means we can explore the impact of lower precision on the adaptive filter performance with other external variables such as regularisation, numerical stability of algorithm, and measurement noise. As one of our representative findings, lower precision arithmetic does not affect adaptive filter performance compared to exact arithmetic when measurement noise is relatively larger than rounding error noise since the measurement noise and the rounding error noise are independently additive with respect to each other. In other words, arithmetic precision can be scaled down in proportion to measurement noise without losing the filter performance.

Future work includes the enhancement of this mathematical framework and the extension of this work to nonlinear adaptive filter algorithms.

Acknowledgement

This project has received funding by the European Commission Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 798209 (Entrans) and the grant agreement No. 732631 (OPRECOMP). We also would like to thank Michael Davis for his proofreading.

References

[1] A. Wiesel, Y. C. Eldar, and A. Yeredor, "Linear regression with gaussian model uncertainty: Algorithms and bounds," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2194–2205, June 2008.

[2] L. Ljung, T. Chen, and B. Mu, "A shift in paradigm for system identification," *International Journal of Control*, vol. 93, no. 2, pp. 173–180, 2020. [Online]. Available: https://doi.org/10.1080/00207179.2019.1578407

[3] S. Haykin, *Adaptive filter theory (3rd ed.)*. Prentice-Hall, Inc., 1996.

[4] S. Ardalan, "Floating-point error analysis of recursive least-squares and least-mean-squares adaptive filters," *IEEE Transactions on Circuits and Systems*, vol. 33, no. 12, pp. 1192–1208, December 1986.

[5] C. Elisei-Iliescu, C. Paleologu, J. Benesty, and S. Ciochină, "A recursive least-squares algorithm based on the nearest kronecker product decomposition," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 4843–4847.

[6] C. Elisei-Iliescu, C. Paleologu, J. Benesty, C. Stanciu, C. Anghel, and S. Ciochină, “Recursive least-squares algorithms for the identification of low-rank systems,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 5, pp. 903–918, 2019.

[7] ——, “A multichannel recursive least-squares algorithm based on a kronecker product decomposition,” in *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, 2020, pp. 14–18.

[8] L.-M. Dogariu, C. Paleologu, J. Benesty, and S. C. a, “An efficient kalman filter for the identification of low-rank systems,” *Signal Processing*, vol. 166, p. 107239, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168419302853

[9] Y. V. Zakharov, G. P. White, and J. Liu, “Low-complexity rls algorithms using dichotomous coordinate descent iterations,” *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3150–3161, 2008.

[10] J. Liu and Y. Zakharov, “FPGA implementation of RLS adaptive filter using dichotomous coordinate descent iterations,” in *Proceedings of the 2009 IEEE International Conference on Communications*, ser. ICC’09. IEEE Press, 2009, pp. 2861–2865.

[11] P. Monsurrò and A. Trifiletti, “The recursive batch least squares filter: An efficient rls filter for floating-point hardware,” in *2017 European Conference on Circuit Theory and Design (ECCTD)*, 2017, pp. 1–4.

[12] V. Sze, Y.-H. Chen, J. Emer, A. Suleiman, and Z. Zhang, “Hardware for machine learning: Challenges and opportunities,” in *Custom Integrated Circuits Conference (CICC), 2017 IEEE*. IEEE, 2017, pp. 1–8.

[13] G. L. Steele, Jr. and J.-B. Tristan, “Adding approximate counters,” in *Proceedings of the 21st ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, ser. PPoPP ’16. New York, NY, USA: ACM, 2016, pp. 15:1–15:12. [Online]. Available: http://doi.acm.org/10.1145/2851141.2851147

[14] J. M. Cioffi, “Limited-precision effects in adaptive filtering,” *Circuits and Systems, IEEE Transactions on*, vol. 34, no. 7, pp. 821–833, 1987.

[15] G. E. Bottomley and S. T. Alexander, “A novel approach for stabilizing recursive least squares filters,” *Signal Processing, IEEE Transactions on*, vol. 39, no. 8, pp. 1770–1779, 1991.

[16] Å. Björck, “Error analysis of least squares algorithms,” in *Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms*, G. H. Golub and P. Van Dooren, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 41–73.

[17] S. Ljung and L. Ljung, “Error propagation properties of recursive least-squares adaptation algorithms,” *Automatica*, vol. 21, no. 2, pp. 157–167, 1985.

[18] D. Higham and N. Higham, “Backward error and condition of structured linear systems,” *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 1, pp. 162–175, 1992. [Online]. Available: https://doi.org/10.1137/0613014

[19] J. H. Wilkinson, “Error analysis of floating-point computation,” *Numer. Math.*, vol. 2, no. 1, pp. 319–340, Dec. 1960. [Online]. Available: https://doi.org/10.1007/BF01386233

[20] J. Wilkinson, *Rounding errors in algebraic processes*. Prentice Hall, 1963.

[21] C. Moler, “Iterative refinement in floating point,” *J. ACM*, vol. 14, no. 2, pp. 316–321, 1967.

[22] D. T. M. Slock, “Backward consistency concept and round-off error propagation dynamics in recursive least-squares algorithms,” *Optical Engineering*, vol. 31, no. 6, pp. 1153–1169, 1992, 10.1117/12.57673.

[23] M. Moonen and J. G. McWhirter, “Systolic array for recursive least squares by inverse updating,” *Electronics Letters*, vol. 29, no. 13, pp. 1217–1218, June 1993.

[24] L. Ljung, “Issues in system identification,” *IEEE Control Systems Magazine*, vol. 11, no. 1, pp. 25–29, Jan 1991.

[25] “IEEE standard for floating-point arithmetic,” *IEEE Std 754-2008*, pp. 1–70, 2008.

[26] L. N. Trefethen, *Numerical Linear Algebra*. SIAM, 1998.

[27] J. Benesty, C. Paleologu, and S. Ciochină, “On regularization in adaptive filtering,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1734–1742, Aug 2011.

[28] ——, “Regularization of the RLS algorithm,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E94.A, no. 8, pp. 1628–1629, 2011.

[29] A. Tikhonov, “Solution of incorrectly formulated problems and the regularization method,” *Soviet Mathematics Doklady*, vol. 4, pp. 1035 – 1038, 1963.

[30] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2002.

[31] A. E. Hoerl and R. W. Kennard, “Ridge regression: Biased estimation for nonorthogonal problems,” *Technometrics*, vol. 42, no. 1, pp. 80–86, Feb. 2000. [Online]. Available: http://dx.doi.org/10.2307/1271436

[32] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI’95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 1137–1143. [Online]. Available: http://dl.acm.org/citation.cfm?id=1643031.1643047

[33] S. Geman, E. Bienenstock, and R. Doursat, “Neural networks and the bias/variance dilemma,” *Neural Computation*, vol. 4, no. 1, pp. 1–58, 1992. [Online]. Available: https://doi.org/10.1162/neco.1992.4.1.1

[34] J. R. Bunch, R. C. Le Borne, and I. K. Proudler, “A conceptual framework for consistency, conditioning, and stability issues in signal processing,” *IEEE Transactions on Signal Processing*, vol. 49, no. 9, pp. 1971–1981, Sep. 2001.

[35] J. Bunch, R. L. Borne, and I. Proudler, “Applying numerical linear algebra techniques to analyzing algorithms in signal processing,” *Linear Algebra and its Applications*, vol. 361, pp. 133 – 146, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002437950200318X

[36] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. [Online]. Available: http://www.jstor.org/stable/2346178

[37] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005. [Online]. Available: http://www.jstor.org/stable/3647580

[38] M. Bekrani and M. Lotfizad, “Clipped LMS/RLS adaptive algorithms: Analytical evaluation and performance comparison with low-complexity counterparts,” *Circuits, Systems, and Signal Processing*, vol. 34, no. 5, pp. 1665–1682, 2015. [Online]. Available: https://doi.org/10.1007/s00034-014-9923-1

[39] C. Paleologu, J. Benesty, and S. Ciochină, “Linear system identification based on a kronecker product decomposition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1793–1808, 2018.

[40] “Lennart Ljung on the past, present, and future of system identification,” https://www.mathworks.com/videos/lennart-ljung-on-the-past-present-and-future-of-system-identification-96990.html, Accessed: 2020-02-22.

[41] G. Pillonetto, F. Dinuzzo, T. Chen, G. D. Nicolao, and L. Ljung, “Kernel methods in system identification, machine learning and function estimation: A survey,” *Automatica*, vol. 50, no. 3, pp. 657 – 682, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S000510981400020X

[42] Y. Engel, S. Mannor, and R. Meir, “The kernel recursive least-squares algorithm,” *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2275–2285, Aug 2004.