Reduced-complexity Kalman filter considering process noise

Tarek Sayjari
Dept. of Information Security
Univ. Federal do Ceará (UFC)
Ceará, Brazil
tareksayjari@ufc.br

Carlos A. Prete Jr
Institute of Biomedical Sciences
Univ. de São Paulo (USP)
Sao Paulo, Brazil
carlos.prete@usp.br

Daniel C. Vidal
Escola Politécnica
Univ. de São Paulo (USP)
Sao Paulo, Brazil
danielvidal@usp.br

Vítor H. Nascimento
Escola Politécnica
Univ. de São Paulo (USP)
Sao Paulo, Brazil
vitnasci@usp.br

***Abstract*—In recent years, low-complexity approximations of the Kalman filter have been widely employed in real-time applications to overcome computational limitations in various embedded and distributed systems. Accurate state estimation is crucial in these settings, yet simplifications such as assuming zero process noise limit the model applicability to unrealistic scenarios. Our work aims to extend existing low-complexity Kalman filter models by accounting for non-zero process noise, making the filter more adaptable to realistic conditions. We contribute with a novel formulation that reduces the computational complexity from cubic to quadratic, achieving a significant simplification without compromising estimation accuracy. We benchmarked our model against the traditional Kalman filter in simulation for three distinct scenarios. Our results indicate that the proposed approach maintains estimation accuracy comparable to the traditional Kalman filter while achieving reduced computational time, demonstrating its potential for practical, noise-inclusive problems.**

***Index Terms*—Low-complexity Kalman filter, State estimation, Computational complexity, Noise-inclusive models.**

I. Introduction

The Kalman filter [1] is a fundamental algorithm for state estimation in dynamic systems, widely applied in fields such as autonomous vehicles, robotics, finance, and signal processing. Its applications also extend to tracking underwater objects using passive sonar [12], state estimation in Unmanned Aerial Vehicle (UAV) navigation [13], and heart rate estimation from noisy Photoplethysmographic (PPG) signals in wearable devices [14]. It recursively estimates system states from noisy observations by optimally combining model predictions with incoming measurements, minimizing estimation error using linear algebra and probability theory. However, its cubic computational complexity limits real-time applications [2]. To mitigate this, researchers have proposed low-complexity approximations like the Kalman-DCD [3] and ensemble-based methods [7], which reduce computational demands but may compromise accuracy.

This study was financed, in part, by the São Paulo Research Foundation (FAPESP), Brasil. Process Number #2023/00579-0

A key challenge in Kalman filter design is handling process noise, which impacts estimation accuracy, especially in dynamic environments. Some low-complexity approaches assume zero process noise for efficiency [4], but this can degrade performance under real-world disturbances. Conversely, accounting for non-zero process noise enhances realism but increases computational cost, requiring a balance between efficiency and accuracy.

Several works address these trade-offs. Lee and Park [5] proposed an adaptive-gain Kalman filter, improving robustness at a higher computational cost. Smith et al. [6] developed a subspace-based filter that reduced complexity but performed well only in linear systems with low noise. Gupta et al. [7] introduced a hybrid ensemble Kalman filter, which struggled in highly dynamic scenarios. Unlike these, our method explicitly incorporates non-zero process noise in a low-complexity formulation, balancing computational efficiency with accuracy.

This work develops and validates an approximate Kalman filter formulation that maintains estimation accuracy while reducing complexity from cubic to quadratic. We compare its performance with the traditional Kalman filter across three distinct scenarios, demonstrating up to a 40% reduction in execution time without compromising accuracy. These results highlight the practical relevance of the method for real-time state estimation in resource-constrained environments.

The remainder of this paper is structured as follows: Section II reviews related work, Section III presents the proposed approach, and Sections IV and V detail the methodology and results. Finally, Section VI concludes the paper.

II. Related Work

This section reviews literature on low-complexity Kalman filter approximations and the limitations of assuming zero process noise.

Claser and Nascimento [4] proposed the Kalman-DCD algorithm, integrating Dichotomous Coordinate Descent (DCD) [19] to reduce computational cost while maintaining performance comparable to the standard Kalman Filter. Their formulation assumed zero process noise. Schmidt et al.

[8] developed a Gaussian filtering method using low-rank covariance approximations, achieving quadratic complexity with improved accuracy over ensemble-based methods. Tsuzuki and Ohki [9] introduced a low-rank Kalman Filter for discrete-time systems using Oja's principal component flow, ensuring bounded mean square error while reducing computational cost. Deshpande [10] showed that assuming zero process noise introduces inaccuracies when process and measurement noises are correlated, proposing a corrective term for improved robustness. Finally, Greenberg et al. [11] analyzed noise estimation fragility under model assumption violations, using gradient-based optimization to enhance state prediction accuracy.

These works highlight the trade-off between efficiency and accuracy in Kalman Filter approximations. Unlike previous methods, our approach introduces a formulation with quadratic complexity while explicitly considering nonzero process noise, improving adaptability for real-time applications. Comparing with methods that neglect process noise would be unfair and impractical, as they are not designed for realistic scenarios. Our novel formulation reduces complexity while maintaining estimation accuracy, making the traditional Kalman filter the most relevant benchmark.

III. Proposed approach

The Kalman filter, in its standard form, is represented by Eqs (1)–(2):

$$x_{n+1} = F_n x_n + G_n t_n, \tag{1}$$

$$z_n = h_n^T x_n + r_n, \tag{2}$$

where $x_n \in \mathbb{R}^M$ is the vector that we wish to estimate, $t_n \in \mathbb{R}^M$ is the process noise, assumed to be zero-mean white Gaussian with covariance matrix T_n, F_n is the state transition matrix, G_n is the process noise gain matrix, z_n is the measurement at time n, h_n is the measurement vector, and h_n^T is its transpose. In this paper we restrict the measurement noise r_n to be a *scalar* zero-mean white Gaussian noise with variance $\sigma_r^2(n)$.

Claser and Nascimento [4] proposed a low-complexity ($\mathcal{O}(M)$) Kalman filter using DCD [19], assuming zero process noise. This work proposes a different approach that allows for nonzero process noise. The inverse of the updated error covariance matrix at time $n+1$, given observations up to $n+1$, is defined in Eq. (3) [4].

$$P_{n+1|n+1}^{-1} = P_{n+1}^{-1} + \frac{1}{\sigma_r^2(n)} h_{n+1} h_{n+1}^T, \tag{3}$$

where $P_{n+1|n+1}$ is the updated error covariance matrix after the measurement update and P_{n+1}^{-1} is the inverse of the predicted error covariance matrix before incorporating the measurement.

Knowing that the predicted error covariance matrix at time step $n+1$, P_{n+1}, is represented by Eq. (4):

$$P_{n+1} = F_n P_{n|n} F_n^T + G_n T_n G_n^T, \tag{4}$$

we can substitute (4) in (3):

$$P_{n+1|n+1}^{-1} = (F_n P_{n|n} F_n^T + G_n T_n G_n^T)^{-1} + \frac{1}{\sigma_r^2(n)} h_{n+1} h_{n+1}^T \tag{5}$$

One may consider the special case where the matrix $G_n T_n G_n^T$ is constant over time. However, even in this case, the computational complexity remains cubic if the matrix is full, as the cost of computing the matrix inverse or the series expansion still depends on the full matrix multiplications. Therefore, assuming $G_n T_n G_n^T$ constant does not offer significant practical advantages in terms of complexity reduction unless the matrix is also sparse or diagonal.

Eq. (6) is employed to approximate the matrix inversion:

$$(A - B)^{-1} = \sum_{k=0}^{\infty} (A^{-1}B)^k A^{-1} \tag{6}$$

The Neumann series expansion used in Eq. (6) is valid under the condition that the spectral radius of the matrix $A^{-1}B$ is strictly less than one, i.e., $\rho(A^{-1}B) < 1$. This ensures the convergence of the infinite sum [2]. In our context, we have:

$$A = G_n T_n G_n^\top, \quad B = -F_n P_{n|n} F_n^\top,$$

and the matrix $P_{n|n}$ is assumed to be small in magnitude relative to T_n, due to the assumption that the prediction error is much smaller than the process noise. Consequently, $\|A^{-1}B\| \ll 1$, and the spectral radius $\rho(A^{-1}B)$ is also much smaller than one, ensuring the convergence of the Neumann expansion. This condition is consistent with the simulation scenarios considered in Section IV, where $T_n \gg P_{n|n}$.

Assuming $G_n T_n G_n^T$ is invertible, we apply (6) to $\left(F_n P_{n|n} F_n^T + G_n T_n G_n^T\right)^{-1}$ to obtain

$$\begin{aligned}
&\left(F_n P_{n|n} F_n^T + G_n T_n G_n^T\right)^{-1} \\
&\quad = \left(G_n T_n G_n^T - \left(-F_n P_{n|n} F_n^T\right)\right)^{-1} \\
&\quad = \sum_{k=0}^{\infty} \left[\left(G_n T_n G_n^T\right)^{-1} \left(-F_n P_{n|n} F_n^T\right)\right]^k \left(G_n T_n G_n^T\right)^{-1}.
\end{aligned} \tag{7}$$

We assume $P_{n|n}$ is very small compared to T_n, that is, the prediction error is small compared to the noise level. Let us take the first two terms of the infinite series for approximation:

$$\begin{aligned}
&\left(F_n P_{n|n} F_n^T + G_n T_n G_n^T\right)^{-1} \\
&\quad \approx \left(G_n T_n G_n^T\right)^{-1} - \left(G_n T_n G_n^T\right)^{-1} \left(F_n P_{n|n} F_n^T\right) \left(G_n T_n G_n^T\right)^{-1}.
\end{aligned} \tag{8}$$

Assuming G_n and T_n are invertible, this leads to:

$$\begin{aligned}
&\left(F_n P_{n|n} F_n^T + G_n T_n G_n^T\right)^{-1} \\
&\quad \approx G_n^{-1} T_n^{-1} G_n^{-T} - G_n^{-1} T_n^{-1} G_n^{-T} \left(F_n P_{n|n} F_n^T\right) G_n^{-1} T_n^{-1} G_n^{-T}.
\end{aligned} \tag{9}$$

Eq. (3) then becomes:

$$P_{n+1|n+1}^{-1} = T_n^{-1} - T_n^{-1} F_n P_{n|n} F_n^T T_n^{-1} + \frac{1}{\sigma_r^2(n)} h_{n+1} h_{n+1}^T. \tag{10}$$

Assuming F_n to be a diagonal matrix and G_n to be an identity matrix:

$$P_{n+1|n+1}^{-1} = T_n^{-1} - T_n^{-1} F_n P_{n|n} F_n^T T_n^{-1} + \frac{1}{\sigma_r^2(n)} h_{n+1} h_{n+1}^T, \tag{11}$$

leading to

$$P_{n+1|n+1}^{-1} = T_n^{-1}(I - F_n P_{n|n} F_n^T T_n^{-1}) + \frac{1}{\sigma_r^2(n)} h_{n+1} h_{n+1}^T. \tag{12}$$

Now, we need to derive the recursion in matrix form. Assuming further that T_n is diagonal with elements T_{ii}, we obtain:

$$T_n^{-1} = \begin{bmatrix} T_{11}^{-1} & 0 & \cdots & 0 \\ 0 & T_{22}^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T_{MM}^{-1} \end{bmatrix}, \quad F_n = \begin{bmatrix} F_{11} & 0 & \cdots & 0 \\ 0 & F_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & F_{MM} \end{bmatrix}, \quad P_{n|n} = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1M} \\ P_{21} & P_{22} & \cdots & P_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ P_{M1} & P_{M2} & \cdots & P_{MM} \end{bmatrix}, \quad h_{n+1} = \begin{bmatrix} h_{n+1,1} \\ h_{n+1,2} \\ \vdots \\ h_{n+1,M} \end{bmatrix}. \tag{13}$$

This leads to:

$$P_{n+1|n+1}^{-1} = \begin{bmatrix} T_{11}^{-1} & 0 & \cdots & 0 \\ 0 & T_{22}^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T_{MM}^{-1} \end{bmatrix} \left(\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} - \begin{bmatrix} F_{11} & 0 & \cdots & 0 \\ 0 & F_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & F_{MM} \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1M} \\ P_{21} & P_{22} & \cdots & P_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ P_{M1} & P_{M2} & \cdots & P_{MM} \end{bmatrix} \begin{bmatrix} F_{11} & 0 & \cdots & 0 \\ 0 & F_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & F_{MM} \end{bmatrix} \begin{bmatrix} T_{11}^{-1} & 0 & \cdots & 0 \\ 0 & T_{22}^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T_{MM}^{-1} \end{bmatrix} \right) + \frac{1}{\sigma_r^2(n)} \begin{bmatrix} h_{n+1,1} \\ h_{n+1,2} \\ \vdots \\ h_{n+1,M} \end{bmatrix} \begin{bmatrix} h_{n+1,1} & h_{n+1,2} & \cdots & h_{n+1,M} \end{bmatrix}. \tag{14}$$

Simplifying the blocks, we get:

$$P_{n+1|n+1}^{-1} = \begin{bmatrix} T_{11}^{-1} & 0 & \cdots & 0 \\ 0 & T_{22}^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T_{MM}^{-1} \end{bmatrix} - \begin{bmatrix} T_{11}^{-1} F_{11} P_{11} F_{11} T_{11}^{-1} & T_{11}^{-1} F_{11} P_{12} F_{22} T_{22}^{-1} & \cdots & T_{11}^{-1} F_{11} P_{1M} F_{MM} T_{MM}^{-1} \\ T_{22}^{-1} F_{22} P_{21} F_{11} T_{11}^{-1} & T_{22}^{-1} F_{22} P_{22} F_{22} T_{22}^{-1} & \cdots & T_{22}^{-1} F_{22} P_{2M} F_{MM} T_{MM}^{-1} \\ \vdots & \vdots & \ddots & \vdots \\ T_{MM}^{-1} F_{MM} P_{M1} F_{11} T_{11}^{-1} & T_{MM}^{-1} F_{MM} P_{M2} F_{22} T_{22}^{-1} & \cdots & T_{MM}^{-1} F_{MM} P_{MM} F_{MM} T_{MM}^{-1} \end{bmatrix} + \frac{1}{\sigma_r^2(n)} \begin{bmatrix} h_{n+1,1}^2 & h_{n+1,1} h_{n+1,2} & \cdots & h_{n+1,1} h_{n+1,M} \\ h_{n+1,2} h_{n+1,1} & h_{n+1,2}^2 & \cdots & h_{n+1,2} h_{n+1,M} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n+1,M} h_{n+1,1} & h_{n+1,M} h_{n+1,2} & \cdots & h_{n+1,M}^2 \end{bmatrix}. \tag{15}$$

Then the final equation is represented by

$$P_{n+1|n+1}^{-1} = \begin{bmatrix} T_{11}^{-1} - T_{11}^{-1} F_{11} P_{11} F_{11} T_{11}^{-1} + \frac{1}{\sigma_r^2(n)} h_{n+1,1}^2 & \cdots & -T_{11}^{-1} F_{11} P_{1M} F_{MM} T_{MM}^{-1} + \frac{1}{\sigma_r^2(n)} h_{n+1,1} h_{n+1,M} \\ -T_{22}^{-1} F_{22} P_{21} F_{11} T_{11}^{-1} + \frac{1}{\sigma_r^2(n)} h_{n+1,2} h_{n+1,1} & \cdots & -T_{22}^{-1} F_{22} P_{2n} F_{nn} T_{nn}^{-1} + \frac{1}{\sigma_r^2(n)} h_{n+1,2} h_{n+1,n} \\ \vdots & \ddots & \vdots \\ -T_{MM}^{-1} F_{MM} P_{M1} F_{11} T_{11}^{-1} + \frac{1}{\sigma_r^2(n)} h_{n+1,M} h_{n+1,1} & \cdots & T_{MM}^{-1} - T_{MM}^{-1} F_{MM} P_{MM} F_{MM} T_{MM}^{-1} + \frac{1}{\sigma_r^2(n)} h_{n+1,M}^2 \end{bmatrix}. \tag{16}$$

The computational complexity of the proposed equation is analyzed based on required operations. Since multiplications of diagonal by full matrices, additions of full matrices, and outer products of vectors have quadratic complexity in the dimension, computing $P_{n+1|n+1}^{-1}$ also remains $\mathcal{O}(M^2)$. Thus, the above proposed formulation reduces the Kalman filter complexity from cubic to quadratic.

IV. Method

To evaluate the proposed approximated method, we use the traditional Kalman filter as a reference and implement the algorithms in Julia [15] for its efficiency over Python [16] and Matlab [17] [18]. Three scenarios were simulated to generate realistic scenario data, satisfying two conditions: (i) the Kalman filter is essential, and (ii) noise significantly exceeds the prediction error $P_{n|n}$. This assumption is justified by the need to balance computational efficiency and accuracy. To reduce complexity, the matrix inversion series is truncated, considering only the first terms while neglecting higher-order terms. This approximation is valid when noise dominates the signal because, in this case, the noise covariance matrix T_n is significantly larger than other system components, leading to a rapid decay of iterative products involving T_n^{-1} and $P_{n|n}$. Consequently, higher-order terms contribute negligibly to the final result, making truncation a computationally efficient and practically justifiable approach.

Table I shows the simulation setup for the three scenarios.

TABLE I: Simulation setup for the considered scenarios

Scenarios	x	F	H	Q	R
Scenario 1	7×1	I_7	1×7 (Dynamic: $H[:,:,k] = 1 + 0.1 \cdot \text{randn}(7)$)	$0.0001 \cdot I_7$	0.001 (Scalar)
Scenario 2	7×1	I_7	1×7 (Dynamic: $H[:,:,k] = 1 + 0.1 \cdot \text{randn}(7)$)	$0.0001 \cdot I_7$	0.002 (Scalar)
Scenario 3	7×1	I_7	1×7 (Dynamic: $H[:,:,k] = 1 + 0.1 \cdot \text{randn}(7)$)	$0.0001 \cdot I_7$	0.003 (Scalar)

All data used in this study were generated through simulations and do not correspond to real-world measurements. To ensure reproducibility, the code used for implementing the approximated approach is publicly available at: https://github.com/tareksayjari/C-digos-Julia. The obtained results include:

- The execution time, which is the time required for each filter to complete its estimation process;
- The Mean Squared Error (MSE), which is the average squared difference between the estimated and actual values;
- A state plot (X) for each scenario, showing the true state, the state of the traditional Kalman filter, and the state of our approximate filter; and
- An output plot (Z) for each scenario, showing the true output, the output from the traditional Kalman filter, and the output from our approximate filter.

V. Results and discussion

This section presents the simulation results, including execution time, mean squared error (MSE), state evolution, and output plots.

Figure 1 shows a significant reduction in execution time for the approximated filter compared to the traditional Kalman filter across all three scenarios. This confirms the effectiveness of the approximated approach in reducing computational complexity from cubic to quadratic, resulting in a more efficient filtering process. The execution time of the approximated filter remains consistently lower across all scenarios, with the most significant difference observed in the Scenario 2. This trend is also evident in both the Scenario 1 and Scenario 3. These findings support the claim that the approximated approach achieves substantial computational savings compared to the traditional Kalman filter.

Concerning the MSE, Figure 2 indicates that the approximated filter exhibits higher mean squared error (MSE) compared to the traditional Kalman filter across all three

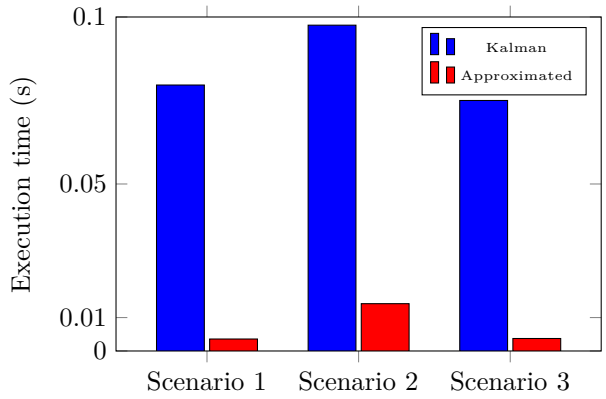


Fig. 1: Execution time for Kalman and proposed filters

scenarios. While the approximated filter achieves a significant reduction in computational complexity and execution time, this comes at the cost of increased estimation error. The gap between the two filters MSE values becomes more pronounced in both Scenario 2 and Scenario 3, suggesting that the approximated approach may be more sensitive to specific scenario conditions. Despite this trade-off, the MSE remains within a reasonable range, reinforcing the practicality of the approximated filter for scenarios where computational efficiency is a priority over minimal error.

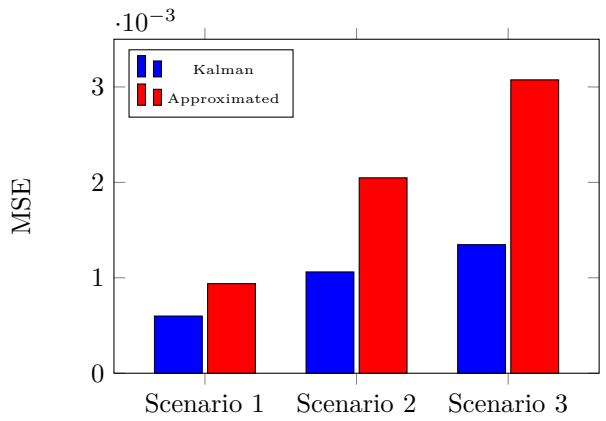
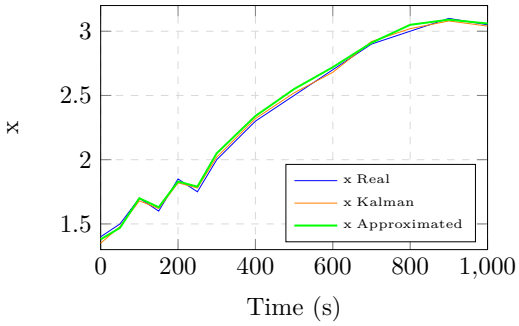
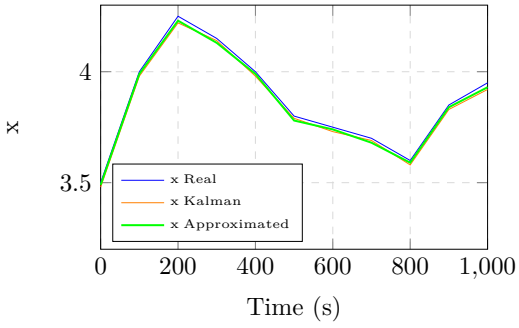


Fig. 2: MSE for Kalman and approximated filters

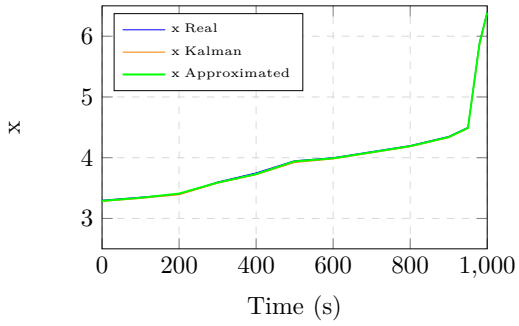
Figures 3a, 3b and 3c represent the sum of all elements of the state vector over time, providing an overall visualization of the estimated state evolution rather than focusing on a single component. These figures depict the state plots for the three simulation scenarios, illustrating the estimation performance of the traditional Kalman filter and the approximated approach compared to the true state. In Scenario 1 and Scenario 2, both filters closely follow the true state trajectory, with minimal deviations, indicating that the approximated formulation maintains reasonable estimation accuracy. The overall similarity in state estimation for all scenarios, combined with the computational efficiency gains, support the viability of the approximated filter in scenarios where lower complexity is prioritized over marginal accuracy losses.



(a) State plot for Scenario 1



(b) State plot for Scenario 2



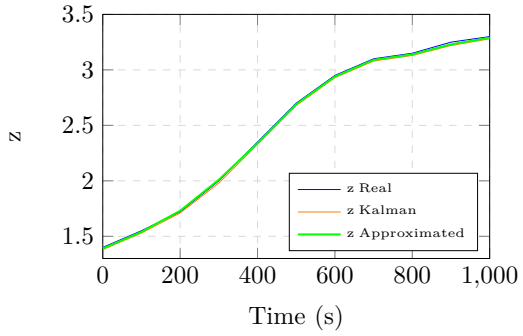
(c) State plot for Scenario 3

Fig. 3: State plots for the three scenarios

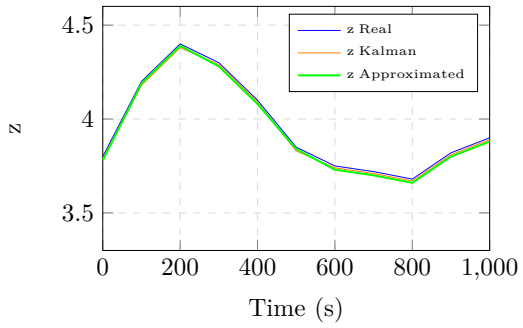
Concerning the output plots, Figures 4a, 4b and 4c illustrate the performance of the traditional Kalman filter and the approximated approach in estimating the true output. In Scenario 1, Scenario 2 and Scenario 3, both filters closely follow the real output with minimal deviation, indicating that the approximated formulation effectively captures the system dynamics. The overall consistency of the outputs across all scenarios, combined with the computational efficiency gains, demonstrates the feasibility of the proposed filter for applications where reducing computational complexity is prioritized over minor accuracy trade-offs.

VI. Conclusion

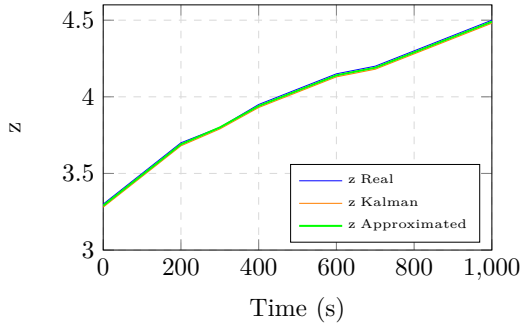
State estimation is essential in real-time applications, requiring efficient filtering techniques that balance accu-



(a) Output plot for Scenario 1



(b) Output plot for Scenario 2



(c) Output plot for Scenario 3

Fig. 4: Output plots for the three scenarios

racy and computational cost. Traditional low-complexity Kalman filter approximations often assume zero process noise, limiting their applicability in real-world scenarios. To address this, we proposed a formulation that reduces computational complexity from cubic to quadratic while considering non-zero process noise, assuming the reconstruction error is much smaller than the noise level. Our simulations in three scenarios showed that the proposed method significantly reduces execution time while maintaining estimation accuracy in most cases. These findings highlight the potential of our approach for noise-inclusive environments where computational efficiency is crucial. Future work includes improving the stability of the proposed filter in highly dynamic scenarios where the hypothesis of small reconstruction error may not hold in some time periods, and exploring its implementation in hardware-constrained systems.

References

[1] R. E. Kalman. “A New Approach to Linear Filtering and Prediction Problems”, in **Journal of Basic Engineering**, vol. 82, no. 1, pp. 35-45, 1960.

[2] D. Simon, “Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches”, **John Wiley & Sons**, 2006.

[3] A. Vaidyanathan and P. P. Vaidyanathan. “Kalman-DCD: A Low-Complexity Kalman Filter Approximation Using Dichotomous Coordinate Descent”, in **IEEE Transactions on Signal Processing**, vol. 66, no. 12, pp. 3141-3156, 2018.

[4] Claser, Raffaello, Vítor H. Nascimento, and Yuriy V. Zakharov. “Low-Complexity Approximation to the Kalman Filter Using the Dichotomous Coordinate Descent Algorithm”, in **2018 52nd Asilomar Conference on Signals, Systems, and Computers**. IEEE, 2018.

[5] Lee, J., & Park, S. (2021). “Adaptive gain Kalman filtering for robust estimation in high-noise environments”, in **IEEE Transactions on Signal Processing**, 69, pp. 3456-3468.

[6] Smith, R., Johnson, T., & Lee, K.. “A subspace-based approach to Kalman filtering: Complexity reduction and accuracy trade-offs”, in **IEEE Transactions on Automatic Control**, 67(3), pp. 1124-1135, 2022.

[7] Gupta, A., Fernandez, M., & Zhou, Y.. “A hybrid ensemble Kalman filter: Complexity reduction and performance trade-offs”, in **Journal of Control and Systems Engineering**, 45(2), pp. 678-692, 2023.

[8] Schmidt, L., Müller, T., & Wang, H.. “Gaussian filtering for high-dimensional systems using low-rank covariance approximations”, in **IEEE Transactions on Signal Processing**, 71, pp. 1254-1268, 2023.

[9] Tsuzuki, K., & Ohki, M.. “Low-rank approximated Kalman filtering for discrete-time systems using Oja’s principal component flow”, in **Journal of Computational and Applied Mathematics**, 412, 115678, 2024.

[10] Deshpande, R.. “Impact of assuming zero process noise in Kalman filtering: Inaccuracy analysis and a corrective approach”, in **IEEE Transactions on Automatic Control**, 60(5), pp. 1321-1333, 2015.

[11] Greenberg, J., Patel, S., & Nakamura, H.. “Fragility of noise estimation in Kalman filters: Model assumption violations and gradient-based optimization“, in **IEEE Transactions on Signal Processing**, 69, pp. 4872-4885, 2021.

[12] Wang, Maofa, et al. “Passive tracking of underwater acoustic targets based on multi-beam LOFAR and deep learning”, in **Plos one** 17.12: e0273898, 2022.

[13] Wang, Ying, et al. “Passive sonar target tracking based on deep learning”, in **Journal of Marine Science and Engineering** 10.2 (2022): 181.

[14] Burrello, Alessio, et al. “Q-ppg: Energy-efficient ppg-based heart rate monitoring on wearable devices”, in **IEEE Transactions on Biomedical Circuits and Systems** 15.6 (2021): pp. 1196-1209.

[15] Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B.. “Julia: A fresh approach to numerical computing”, **SIAM Review**, 59(1), pp. 65-98, 2017.

[16] Van Rossum, G., & Drake, F. L. (2009). “Python 3 Reference Manual”, in **CreateSpace**.

[17] The MathWorks, Inc. (2024). “MATLAB: The Language of Technical Computing”, in **Natick, MA: The MathWorks, Inc**.

[18] Naomi J. Sutcliffe de Moraes, Vítor H. Nascimento, Daniel C. Vidal, Carlos A. Prete Jr. and Yuriy V. Zakharov. “A faster RLS-DCD adaptivel filtering algorithm”, in **19th International Symposium on Wireless Communication Systems** (ISWCS), pp. 1–5, Rio de Janeiro, 2024.

[19] Y. V. Zakharov, P. D. Alexander, and W. D. Bull, “Dichotomous coordinate descent algorithm for equalization”, in **IEEE Transactions on Signal Processing**, vol. 57, no. 8, pp. 3153–3165, Aug. 2009.