

## Article

# Augmented Physics-Based Models for High-Order Markov Filtering

Shuo Tang <sup>1,\*</sup>, Tales Imbiriba <sup>1</sup>, Jindřich Duník <sup>2</sup>, Ondřej Straka <sup>2</sup> and Pau Closas <sup>1</sup><sup>1</sup> Electrical and Computer Engineering Department, Northeastern University, Boston, MA 02115, USA; closas@ece.neu.edu (P.C.)<sup>2</sup> Department of Cybernetics, University of West Bohemia, Univerzitní 8, 30100 Plzeň, Czech Republic

\* Correspondence: tang.shu@northeastern.edu

**Abstract:** Hybrid physics-based data-driven models, namely, augmented physics-based models (APBMs), are capable of learning complex state dynamics while maintaining some level of model interpretability that can be controlled through appropriate regularizations of the data-driven component. In this article, we extend the APBM formulation for high-order Markov models, where the state space is further augmented with past states (AG-APBM). Typically, state augmentation is a powerful method for state estimation for a high-order Markov model, but it requires the exact knowledge of the system dynamics. The proposed approach, however, does not require full knowledge of dynamics, especially the Markovity order. To mitigate the extra computational burden of such augmentation we propose an approximated-state APBM (AP-APBM) implementation leveraging summaries from past time steps. We demonstrate the performance of AG- and AP-APBMs in an autoregressive model and a target-tracking scenario based on the trajectory of a controlled aircraft with delay-feedback control. The experiments showed that both proposed strategies outperformed the standard APBM approach in terms of estimation error and that the AP-APBM only degraded slightly when compared to AG-APBM. For example, the autoregressive (AR) model simulation in our settings showed that AG-APBM and AP-APBM reduced the estimate error by 31.1% and 26.7%. The time cost and memory usage were reduced by 37.5% and 20% by AP-APBM compared to AG-APBM.

**Keywords:** nonlinear filtering; high-order Markov; hybrid neural network

**Citation:** Tang, S.; Imbiriba, T.; Duník, J.; Straka, O.; Closas, P. Augmented Physics-Based Models for High-Order Markov Filtering. *Sensors* **2024**, *24*, 6132. <https://doi.org/10.3390/s24186132>

Academic Editor: Wei Yi

Received: 8 August 2024

Revised: 13 September 2024

Accepted: 20 September 2024

Published: 23 September 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

State estimation and filtering with noisy measurements is an essential component in numerous information processing engineering applications [1]. The filtering process usually involves two steps: the prediction through the transition model and a correction based on the observed measurements and associated model. Based on the complexity of the models, the filters can be classified into two categories: (1) linear filters, e.g., Kalman filter (KF) for the linear models [2], and (2) nonlinear filters, e.g., extended Kalman filters (EKF) [3], sigma-point Kalman filters [4,5], and particle filters [6].  $H_\infty$  filter is an alternative filtering technique other than Kalman filters, which works especially for unknown noise statistics and the worst-case estimation error [7].  $H_\infty$  filters are also applied to various applications, e.g., network systems [8], target tracking [9], battery charging [10], etc. In this article, we focus on Kalman filter techniques with given noise statistics.

When considering nonlinear dynamics, machine learning (ML) strategies are appealing when dealing with complex models due to their flexibility and effectiveness in constructing mappings and capturing intricate patterns. However, purely data-driven ML solutions, which do not leverage the knowledge brought by physics-based models (PBM), lack interpretability of the physical meaning of estimated quantities, which is especially relevant when one aims at recovering latent states [11,12]. To incorporate the information from physics knowledge, hybrid ML algorithms are usually used to provide

corrections to the estimation [13,14] through neural networks (NNs) or estimate the state directly [15]. In parallel, it has been shown that the NN parameter learning process can be interpreted as a state estimation filtering problem [16,17], which can be leveraged to design sequential training schemes that jointly estimate the state process and NN parameters. This training approach is exploited by augmented physics-based models (APBMs) [18] where physics-based models are augmented by a data-driven component that complements or learns the behaviors that physics-based models cannot represent. A survey of augmented physics-based models for navigation systems is presented in [19].

High-order Markov chains are widely applied and are able to achieve better performance in many time-series data processing applications, especially when data latency or communication delay occurs, e.g., biological sequence analysis [20], speech recognition [21], classification and detection problem [22], and autoregressive process estimation [23,24]. Current solutions to this problem usually involve a more complex system model. In those situations the APBM has great potential, where the complexity that can be captured by the NN part while the physics-based component of the model is kept simple.

In this paper, we focus on adapting the APBM approach to high-order Markov models when the system dynamics and the order of the Markovity are not given. More precisely, we augment the state of the high-order Markov model to fit the APBM and train the parameters of the NN with longer-memory data. An approximation-based method is also put forward to reduce the computational cost brought by the higher-dimensional augmented state. Section 2 introduces these two proposed APBM-based methods. Finally, a tracking experiment based on a delayed-feedback control application is discussed in Section 3 to validate the two proposed approaches. The remainder of this section quickly reviews APBMs and how high-order Markovianity is typically handled in filtering contexts.

### 1.1. Augmented Physics-Based Model

Consider the Markovian dynamics and measurement models

$$\begin{aligned} \mathbf{x}_k &= f(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}^x \\ \mathbf{y}_k &= h(\mathbf{x}_k) + \mathbf{v}_k^y, \end{aligned} \quad (1)$$

where  $\mathbf{x}_k \in \mathbb{R}^{d_x \times 1}$  is the state vector at time  $k$  and  $\mathbf{y}_k \in \mathbb{R}^{d_y \times 1}$  is the measurement vector at time  $k$ .  $f(\cdot)$  and  $h(\cdot)$  are, possibly nonlinear, vector functions describing the state dynamics and measurement equations, respectively.  $\mathbf{w}_{k-1}^x \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^x)$  is the zero-mean Gaussian noise of dynamics model and  $\mathbf{v}_k^y \sim \mathcal{N}(\mathbf{0}, \mathbf{R}^y)$  is the zero-mean Gaussian noise of measurement model. The proposed hybrid NN framework is able to learn the dynamics (1) from both sampling data and physics knowledge. The state dynamics, as described by an APBM, can be expressed as

$$\mathbf{x}_k = \check{g}(\bar{f}(\mathbf{x}_{k-1}), \mathbf{x}_{k-1}; \boldsymbol{\theta}) + \mathbf{w}_{k-1}^x, \quad (2)$$

where  $\bar{f}(\cdot) : \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_x}$  is the PBM, which can be obtained from the simplification, approximation, or partial knowledge of the true dynamics model  $f(\cdot)$ .  $\check{g}(\cdot) : \mathbb{R}^{d_x} \times \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_x}$  is a vector-valued function, including a NN parameterized by  $\boldsymbol{\theta} \in \mathbb{R}^{d_\theta}$  to compensate the mismatch of the PBM  $\bar{f}$  compared with true dynamics function  $f$ .

Furthermore, the regularization method involving a parameter value  $\bar{\boldsymbol{\theta}}$  is introduced in the APBM framework [18] to prevent the NN augmentation from completely taking over the model dynamics and neglecting the physics. The value  $\bar{\boldsymbol{\theta}} \in \mathbb{R}^{d_\theta}$  is defined such that the augmented model is equivalent to the PBM when  $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}$ :  $\check{g}(\bar{f}(\mathbf{x}_{k-1}), \mathbf{x}_{k-1}; \boldsymbol{\theta} = \bar{\boldsymbol{\theta}}) = \bar{f}(\mathbf{x}_{k-1})$ .

The learning process infers the state  $\mathbf{x}_k$  and the parameter vector  $\boldsymbol{\theta}$  estimates. Considering the Bayesian estimation training approach [16,17], based on augmentation of the state

$\mathbf{x}_k$  by the parameter vector  $\boldsymbol{\theta}_k$ , the state-space dynamics and measurement model can be rewritten as

$$\begin{aligned} \begin{bmatrix} \boldsymbol{\theta}_k \\ \mathbf{x}_k \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\theta}_{k-1} \\ \check{g}(\bar{f}(\mathbf{x}_{k-1}), \mathbf{x}_{k-1}; \boldsymbol{\theta}_{k-1}) \end{bmatrix} + \begin{bmatrix} \mathbf{w}_{k-1}^\theta \\ \mathbf{w}_{k-1}^x \end{bmatrix} \\ \begin{bmatrix} \mathbf{y}_k \\ \bar{\boldsymbol{\theta}} \end{bmatrix} &= \begin{bmatrix} h(\mathbf{x}_k) \\ \boldsymbol{\theta}_k \end{bmatrix} + \begin{bmatrix} \mathbf{v}_k^y \\ \mathbf{v}_k^\theta \end{bmatrix}, \end{aligned} \quad (3)$$

where  $\mathbf{w}_k^\theta \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^\theta)$  denotes the zero-mean Gaussian noise for NN parameter dynamics. The equation  $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta}_k + \mathbf{v}_k^\theta$  in (3) serves for regularization purposes and can be perceived as a soft constraint of the APBM with respect to the PBM. It has a form of the pseudo-measurement with noise  $\mathbf{v}_k^\theta \sim \mathcal{N}(\mathbf{0}, \frac{1}{\lambda} \mathbf{I})$ , where  $\lambda$  is the user-defined parameter that controls penalization of distance between  $\boldsymbol{\theta}_k$  and  $\bar{\boldsymbol{\theta}}$ .

### 1.2. Augmented State for High-Order Markov Models

The transition model shown in (1) actually follows a first-order Markov chain, which means the current state is independent of all previous states except the most recent one [25] (Ch. 13). In a probabilistic framework, the joint distribution for a sequence of  $K$  states  $[\mathbf{x}_1, \dots, \mathbf{x}_K]$  under such a model (ignoring the measurements) is given by

$$p(\mathbf{x}_1, \dots, \mathbf{x}_K) = p(\mathbf{x}_1) \prod_{k=2}^K p(\mathbf{x}_k | \mathbf{x}_{k-1}). \quad (4)$$

The first-order Markov model is a general and useful assumption in many applications, but it is restrictive in others. In reality, the sequential observations usually indicate a trend, which means the past data could provide nontrivial information to the next prediction. In this case, we model a high-order Markov chain [25] (Ch. 13) and [26], which allows the prediction to depend on a sequence of previous states. For instance, the joint distribution of the states in a second-order Markov chain is given by

$$p(\mathbf{x}_1, \dots, \mathbf{x}_K) = p(\mathbf{x}_1) p(\mathbf{x}_2 | \mathbf{x}_1) \prod_{k=3}^K p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{x}_{k-2}). \quad (5)$$

A common approach to deal with high-order Markov is to augment the state vector with the previous states [23,27,28] so that the model becomes first-order Markovian. The augmented-state example of a second-order linear Markov model is shown in [29] (Ch. 6). Considering a general  $p$ th-order Markov model for the dynamics, Equation (1) can be expressed as

$$\begin{aligned} \mathbf{x}_k &= f(\mathbf{x}_{k-1}, \dots, \mathbf{x}_{k-p}) + \mathbf{w}_{k-1}^x \\ \mathbf{y}_k &= h(\mathbf{x}_k) + \mathbf{v}_k^y, \end{aligned} \quad (6)$$

where  $\mathbf{x}_k \in \mathbb{R}^{d_x \times 1}$  is the state vector at time  $k$ ,  $\mathbf{y}_k \in \mathbb{R}^{d_y \times 1}$  is the measurement vector at the time  $k$ . To filter this model, the conventional technique is to augment the state as  $\tilde{\mathbf{x}}_k = [\mathbf{x}_k^\top, \dots, \mathbf{x}_{k-p+1}^\top]^\top$  and the transition model becomes

$$\tilde{\mathbf{x}}_k = \begin{bmatrix} f(\tilde{\mathbf{x}}_{k-1}) \\ [\mathbf{I}_{d_x p \times d_x p} \quad \mathbf{0}_{d_x p \times 1}] \tilde{\mathbf{x}}_{k-1} \end{bmatrix} + \begin{bmatrix} \mathbf{w}_{k-1}^x \\ \mathbf{0}_{d_x p \times 1} \end{bmatrix}, \quad (7)$$

where  $\mathbf{I}$  and  $\mathbf{0}$  denote the identity and zero matrices with the dimensions indicated in the sub-indices. This augmented state (AGS) transforms the high-order Markovity into a first-order Markov model. After the augmentation, standard Bayesian filtering techniques can be used on the transformed first-order Markov model. In this article, due to the nonlinear nature of the APBM and measurement models, we employ a Cubature Kalman filter (CKF) in the experiments, which propagates the cubature points through the transition

and measurement model [5], although other filtering solutions could be considered without loss of generality.

### 1.3. Contributions

State estimation in high-order Markov models presents a significant challenge due to the dependence of system dynamics on previous states. A widely-used method to address this issue is the aforementioned AGS approach. However, AGS is not applicable when (1) the full knowledge of the complex system dynamics (associated with the high-order Markov model) is unavailable or (2) the order of Markovity is unknown. In this paper, we introduce the augmented-state APBM, which leverages the APBM technique to learn system dynamics via the AGS method, allowing for state estimation even with partially known system dynamics or an unknown Markovian order. To reduce the computational burden brought by the state augmentation (based on the order of Markovity), we proposed the approximated-state APBM by using the point estimate of the posterior distribution of the previous states as the training input, while this approximation introduces a slight degradation in estimation accuracy, it significantly reduces computational costs by lowering the dimension of the state space.

## 2. Augmented-State APBM for High-Order Markov Models

In this section, we extend APBMs to cope with high-order Markov models, aiming at learning the dynamics from data while constraining the augmented model around the PBM. Since high-order Markov models considered in this paper do not lead to changes in the measurement model (1), we focus next on the transition model and associated prediction process.

### 2.1. Augmented-State APBM

APBMs are appealing when the nonlinear dynamics  $f(\cdot)$  of the  $p$ th-order Markov model in (6) are not accurately known, either due to their parametric representation and/or the knowledge of the Markov process order. In this case, we employ APBM to learn the transition model

$$\mathbf{x}_k = g(\tilde{f}(\mathbf{x}_{k-1}), \mathbf{x}_{k-1}, \dots, \mathbf{x}_{k-l}; \boldsymbol{\theta}) + \mathbf{w}_{k-1}, \quad (8)$$

where  $g(\cdot) : \underbrace{\mathbb{R}^{d_x} \times \dots \times \mathbb{R}^{d_x}}_l \mapsto \mathbb{R}^{d_x}$  includes the PBM and a NN parameterized by  $\boldsymbol{\theta} \in \mathbb{R}^{d_\theta}$ .

We use  $g$  instead of  $\tilde{g}$  to denote the APBM function that accounts for longer state memory.  $l$  is a design parameter that is intuitively assumed to be chosen such that  $l \geq p$ , although simulations show that reasonable results can be obtained otherwise.

For the  $p$ th-order Markov chain, the predictive distribution of the state can be computed as

$$p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) = \int \dots \int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \dots, \mathbf{x}_{k-p}, \mathbf{y}_{1:k-1}) \cdot p(\mathbf{x}_{k-1:k-p} | \mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1} d\mathbf{x}_{k-2} \dots d\mathbf{x}_{k-p}. \quad (9)$$

However, the joint posterior distribution  $p(\mathbf{x}_{k-1:k-p} | \mathbf{y}_{1:k-1})$  is usually not accessible. As mentioned before, the typical approach to tackling this problem is through state augmentation. Considering a  $p$ th-order Markovian model (6) and the APBM (8), augmentation of the state vector with the  $l-1$  previous states results in  $\tilde{\mathbf{x}}_k = [\mathbf{x}_k^\top, \dots, \mathbf{x}_{k-l+1}^\top]^\top$ . The augmented-state APBM (AG-APBM) is defined by

$$\begin{aligned}
\tilde{\mathbf{x}}_k &= \begin{bmatrix} g(\tilde{f}(\mathbf{x}_{k-1}), \mathbf{x}_{k-1}, \dots, \mathbf{x}_{k-l}; \boldsymbol{\theta}_{k-1}) \\ [\mathbf{I}_{d_x l \times d_x l} \quad \mathbf{0}_{d_x l \times 1}] \tilde{\mathbf{x}}_{k-1} \end{bmatrix} + \begin{bmatrix} \mathbf{w}_{k-1}^x \\ \mathbf{0}_{d_x l \times 1} \end{bmatrix} \\
&\equiv \tilde{g}(\tilde{\mathbf{x}}_{k-1}; \boldsymbol{\theta}_{k-1}) + \tilde{\mathbf{w}}_{k-1}^x \\
\boldsymbol{\theta}_k &= \boldsymbol{\theta}_{k-1} + \mathbf{w}_{k-1}^\theta,
\end{aligned} \tag{10}$$

where  $\tilde{\mathbf{w}}_k^x \sim \mathcal{N}(\mathbf{0}, \text{diag}[\mathbf{Q}^x, \mathbf{0}_{(p-1) \times (p-1)}])$  denotes the augmented state processing noise. The first row in  $\tilde{\mathbf{x}}_k$  is the original  $p$ th-order Markov transition model and the other rows represent the dynamics of the augmentation. For the joint state-parameter estimation we assume the same augmented measurement model as in (3) since it regularizes the NN contribution to the dynamics, i.e.,  $\tilde{g}(\tilde{\mathbf{x}}_{k-1}; \boldsymbol{\theta} = \hat{\boldsymbol{\theta}}) = \tilde{f}(\mathbf{x}_{k-1})$  [18]. For this augmented state, the predictive distribution is given by

$$p(\tilde{\mathbf{x}}_k | \mathbf{y}_{1:k-1}) = \int p(\tilde{\mathbf{x}}_k | \tilde{\mathbf{x}}_{k-1}, \mathbf{y}_{1:k-1}) \cdot p(\tilde{\mathbf{x}}_{k-1} | \mathbf{y}_{1:k-1}) d\tilde{\mathbf{x}}_{k-1}. \tag{11}$$

The above Equation (11) has the same quantities with Equation (9), but different meaning and feasibility. The posterior  $p(\mathbf{x}_{k-1:k-p} | \mathbf{y}_{1:k-1})$  in Equation (9) is not easy to compute because it is in the joint form. However, notice that by augmenting the vector, the posterior  $p(\tilde{\mathbf{x}}_{k-1} | \mathbf{y}_{1:k-1})$  is not the joint distribution anymore, but a high-dimensional simpler posterior distribution, as it only involves a single variable  $\tilde{\mathbf{x}}_{k-1}$ . This posterior distribution is accessible during the filtering process. For example, in a Kalman filter, we recursively compute the posterior for state estimation at each time step.

## 2.2. Approximated-State APBM

The augmentation approach discussed above can lead to very high-dimensional state spaces if  $l$  is too large; therefore, it increases the computational cost of such a solution. This cost is mostly due to the need to solve multi-dimensional integrals in the Bayesian filtering Equation (11). To circumvent this issue, using the point estimate from the previous steps—rather than using the entire distribution—reduces the computational complexity in evaluating the aforementioned integrals. This leads to the approximated-state APBM (AP-APBM):

$$\begin{aligned}
\mathbf{x}_k &= g(\tilde{f}(\mathbf{x}_{k-1}), \mathbf{x}_{k-1}, \hat{\mathbf{x}}_{k-2}, \dots, \hat{\mathbf{x}}_{k-l}; \boldsymbol{\theta}_{k-1}) + \mathbf{w}_{k-1} \\
\boldsymbol{\theta}_k &= \boldsymbol{\theta}_{k-1} + \mathbf{w}_{k-1}^\theta,
\end{aligned} \tag{12}$$

where  $\hat{\mathbf{x}}_{k-j} = \mathbb{E}\{\mathbf{x}_{k-j} | \mathbf{y}_{1:k-j}\}$ ,  $j = 2, \dots, l$ . It is worth noting that approximations have to be made to obtain the above model. Equation (13) provides a mathematical explanation of this intuition by approximating the posterior distribution previous to its use in the integral.

$$\begin{aligned}
p(\mathbf{x}_{k-2:k-l} | \mathbf{y}_{1:k-1}) &\approx \prod_{i=2}^l p(\mathbf{x}_{k-i} | \mathbf{y}_{1:k-1}) \approx \prod_{i=2}^l p(\mathbf{x}_{k-i} | \mathbf{y}_{1:k-i}) \\
&\approx \prod_{i=2}^l \delta(\mathbf{x}_{k-i} - \mathbb{E}\{\mathbf{x}_{k-i} | \mathbf{y}_{1:k-i}\}),
\end{aligned} \tag{13}$$

where  $\delta(\cdot)$  is the Dirac delta function. Note that in the above approximation, we first assume that the posterior distributions at each time step are independent, and then we neglect the dependency between states and future measurements. In addition, we finally replace the posterior distribution of the states with their point estimates, which is equivalent to the model in (12).

Finally, we present the AP-APBM

$$\begin{bmatrix} \theta_k \\ x_k \end{bmatrix} = \begin{bmatrix} \theta_{k-1} \\ g(\bar{f}(x_{k-1}), x_{k-1}, \hat{x}_{k-l:k-2}; \theta_{k-1}) \end{bmatrix} + \begin{bmatrix} w_{k-1}^\theta \\ w_{k-1}^x \end{bmatrix}$$

$$\begin{bmatrix} y_k \\ \hat{\theta} \end{bmatrix} = \begin{bmatrix} h(x_k) \\ \theta_k \end{bmatrix} + \begin{bmatrix} v_k^y \\ v_k^\theta \end{bmatrix}. \quad (14)$$

where it can be observed that although the AP-APBM needs memory from past state estimates  $[\hat{x}_{k-1}, \dots, \hat{x}_{k-l}]$ , its computational cost is much lower than the cost of the AG-APBM since the dimension of the state is reduced significantly.

### 3. Numerical Simulations

In this section, we implement the proposed AG- and AP-APBM approaches in the context of the high-order Markov models, particularly with the aim of performing state estimation in two common applications: the autoregressive (AR) model problem and the time-delayed control problem.

#### 3.1. AR Model

The AR model is one of the most common high-order Markov models. Here, we implemented the proposed approaches to filter the states generated by an AR(3) model based on noisy measurements. Considering the 2-dimensional system state  $x = [x_1, x_2]^\top$  and the 2-dimensional measurement  $y = [y_1, y_2]^\top$ , the transition and measurement models are given by

$$x_k = F_1 x_{k-1} + F_2 x_{k-2} + F_3 x_{k-3} + w_{k-1}^x \quad (15)$$

$$y_k = H x_k + v_k^y, \quad (16)$$

where  $w_{k-1}^x \sim \mathcal{N}(0, Q^x)$  and  $v_k^y \sim \mathcal{N}(0, R^y)$  are additive Gaussian noise with  $Q^x = \sigma_x^2 I_{2 \times 2}$ ,  $R^y = \sigma_y^2 I_{2 \times 2}$  and  $\sigma_x = 0.1$ ,  $\sigma_y = 0.1$ .  $H = I_{2 \times 2}$  and

$$F_1 = \begin{bmatrix} 0.5 & -0.3 \\ 0.4 & -0.2 \end{bmatrix} \quad F_2 = \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0.2 \end{bmatrix} \quad F_3 = \begin{bmatrix} -0.1 & 0.05 \\ -0.05 & 0.1 \end{bmatrix}. \quad (17)$$

We then compared the performance of the estimation of the filtering process when using the true model above, the PBM-only model, AG- and AP-APBM in the 1st order and 3rd order. It is noted that the true model is constructed based on the AGS method as shown in Equation (7). In the experiments, the PBM refers to the below AR(1) model

$$x_k = F_1 x_{k-1} + w_{k-1}^x. \quad (18)$$

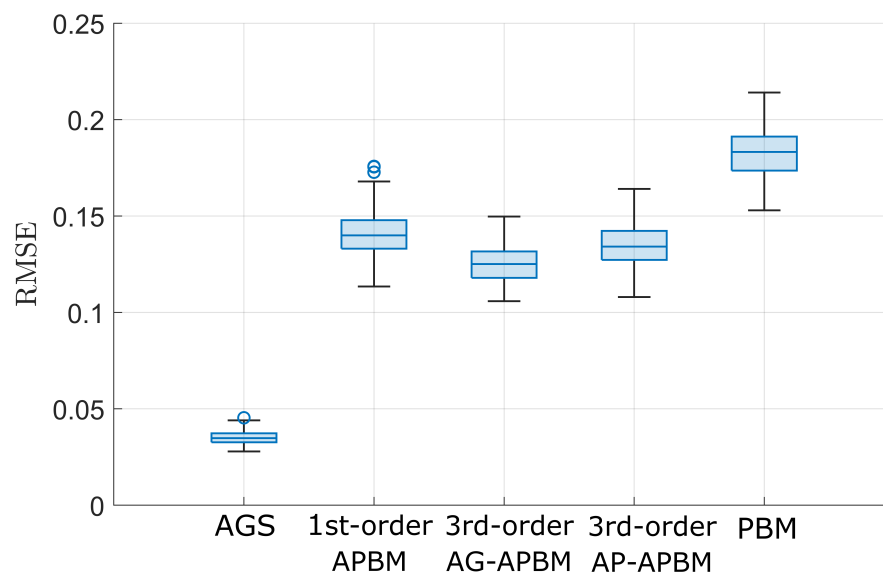
The APBM is a linear combination between the PBM and multilayer perceptrons (MLP) as shown below

$$g_\theta(x_{k-1}, \cdot) = w_0 F_1 x_{k-1} + w_1 \gamma_\phi(x_{k-1}, \cdot), \quad (19)$$

where  $\theta = [w_0, w_1, \phi]$  and the MLP consists of 1 hidden layer with 5 units and ReLu activation function and output layer with two output units and linear activation function. The second argument in the function (19) can be null for 1st-order APBM,  $[x_{k-2}, x_{k-3}]$  for 3rd-order AG-APBM, and  $[\hat{x}_{k-2}, \hat{x}_{k-3}]$  for 3rd-order AP-APBM. Figure 1 shows the root mean square errors (RMSEs) of the estimation using the measurements from the AR(3) model, which are computed based on 200 Monte Carlo (MC) simulations. The PBM gives the largest error because it employs limited knowledge with an AR(1) model, while the true model always achieves the best precision since it has full knowledge of the dynamics model. All three APBMs have better estimation precision compared to the PBM. In terms of the estimate error median, the 1st-order APBM reduces the error from PBM by 22.2%,

the 3rd-order AG-APBM reduces the error by 31.1%, and the 3rd-order AP-APBM reduces the error by 26.7%. Among the hybrid learning approaches, the 3rd-order AG-APBM has the best performance since it learns the dynamics through the augmented state with the last two steps. Instead of incorporating the last two steps into the state, the 3rd-order AP-APBM leverages the point estimate of the last two steps as the input of the NN. The median of the RMSE increases about 6.5% compared to 3rd-order AG-APBM, but still lower than the 1st-order APBM. More importantly, the computational cost is significantly reduced by the approximation. Figure 2 shows that the time cost of the 3rd-order AP-APBM decreases by about 37.5% compared with the 3rd-order AG-APBM and the memory usage decreases by about 20% over 200 MC simulations. The simulations are implemented based on the following:

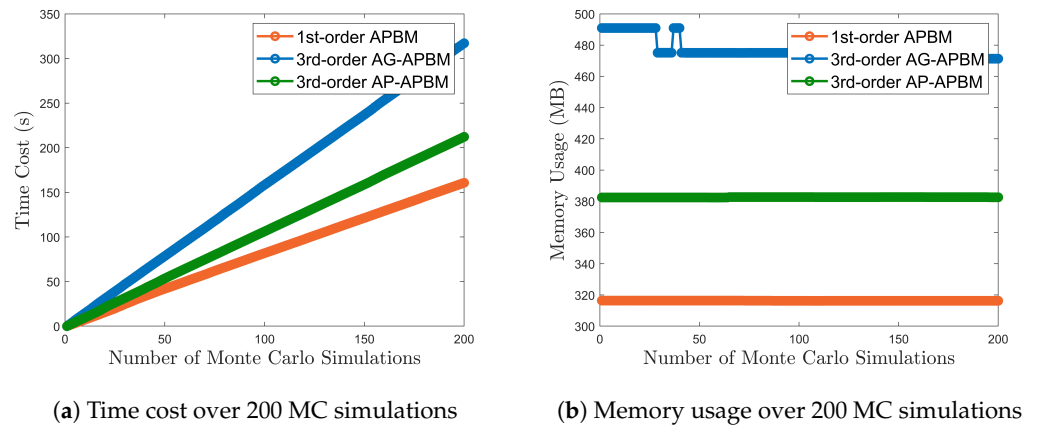
- **Processor (CPU):** Intel Core i7-10700KF, 8 cores, 3.80 GHz. The multi-core CPU allowed for efficient parallel processing during data preprocessing.
- **Memory (RAM):** 32 GB DDR4.
- **Storage:** 1 TB NVMe SSD.
- **Operating System:** Windows 11 Pro.
- **Software Platform:** MATLAB R2023a.



**Figure 1.** Box plots for the root mean square error (RMSE) of state estimation for the autoregressive-3 (AR(3)) model computed over 200 Monte-Carlo (MC) simulations. The central line in the box indicates the median. The bottom and top edges of the box denote the 25th and 75th percentiles, and the circles denotes the outliers. The AGS and PBM represent the estimation based on the true model and AR(1) model.

It is shown that for AP-APBM, the time cost and memory usage are larger than the cost of the 1st-order APBM, since the extra memory is used to store the approximation  $\{\hat{x}_{k-2}, \hat{x}_{k-3}\}$  when estimating  $\hat{x}_k$ . However, the computational cost of the AP-APBM has been greatly reduced compared to the AG-APBM excluding the extra augmented state.





**Figure 2.** Computational cost over 200 MC simulations. Each MC repetition includes a filtering process of 600 time steps. The simulations were implemented on MATLAB. The cost of AGS and PBM model are not plotted here because the deficiency of training process makes their computational cost incomparable to the hybrid NN models.

### 3.2. A Delayed-Feedback Control Nonlinear Model

We test the different approaches based on a target tracking problem, where the target is moving on a two-dimensional horizontal plane. The target is operates with an internal feedback control loop, unobserved to the tracker, where communication delays usually occur inside [30,31]. The corresponding dynamics and measurement models are also common in target tracking problems [32]. Considering the state  $[x^\top, \Omega]^\top$ , where  $x = [p_x, v_x, p_y, v_y]^\top$  is the two-dimensional position and velocity,  $\Omega$  is the turning rate of the target and the feedback control input  $u$ , the dynamics model is given by

$$x_k = Fx_{k-1} + Bu_{k-1} + w_{k-1}^x, \quad (20)$$

$$\Omega_k = \Omega_{k-1} + w_{k-1}^\Omega, \quad (21)$$

where  $w_{k-1}^x \sim \mathcal{N}(0, \text{diag}(0.1, 0.1, 0.1, 0.1))$ ,  $w_{k-1}^\Omega \sim \mathcal{N}(0, 10^{-4})$ ,  $F$  defines the constant-velocity model

$$F = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (22)$$

$T_s = 1s$  denotes the sampling period,  $B = I_{4 \times 4}$  is the identity matrix, and  $u_{k-1} = s(k_c G_{k-1}(x_{k-3} - \bar{x}_{k-1}))$  is the input controlled by the difference between the previous state  $x_{k-3}$  and reference  $\bar{x}_{k-1}$ .  $s(A) : \mathbb{R}^{4 \times 4} \rightarrow \mathbb{R}^{4 \times 4}$  denotes a saturation function which restricts all elements in the matrix  $A$  within  $[-5, 5]$  due to the possible limitation of the actuator and  $k_c = 1$  is the controller gain. Specifically, we introduce  $G_{k-1}$ , controlling the state only by the difference between the true velocity and the reference.

$$G_{k-1} = \begin{bmatrix} 0 & \frac{\sin \Omega_{k-1} T_s}{\Omega_{k-1}} & 0 & -\frac{1 - \cos \Omega_{k-1} T_s}{\Omega_{k-1}} \\ 0 & \cos \Omega_{k-1} T_s & 0 & -\sin \Omega_{k-1} T_s \\ 0 & \frac{1 - \cos \Omega_{k-1} T_s}{\Omega_{k-1}} & 0 & \frac{\sin \Omega_{k-1} T_s}{\Omega_{k-1}} \\ 0 & \sin \Omega_{k-1} T_s & 0 & \cos \Omega_{k-1} T_s \end{bmatrix}. \quad (23)$$



The error is usually defined between the current state  $\mathbf{x}_{k-1}$  and the reference state  $\bar{\mathbf{x}}_{k-1}$  at the same moment  $k - 1$ . However, in this case, it is computed based on the state before two steps  $\mathbf{x}_{k-3}$ , due to communication delay, which means the aircraft can access its state until two steps from the current time. (Note that the true state is known to the aircraft but it is unknown to the tracker and is thus estimated by the tracker.) Notice that  $\mathbf{G}_{k-1}$  introduces the high-order Markovian property into the dynamics. The measurement model is given by the received signal strength and bearings from two collocated sensors

$$\mathbf{y}_k = \begin{pmatrix} 10 \log_{10} \left( \frac{\Psi_0}{\|\mathbf{p}_0 - \mathbf{p}_k\|^q} \right) \\ \angle(\mathbf{p}_0, \mathbf{p}_k) \end{pmatrix} + \mathbf{v}_k^y, \quad (24)$$

with  $\mathbf{p}_0 = (0, 0)^\top$  being the position of the sensors,  $\mathbf{p}_k = (p_{x,k}, p_{y,k})^\top$  the unknown position of the target,  $10 \log_{10}(\Psi_0) = 30$  dBm,  $q = 2.2$  the path loss exponent,  $\angle(\mathbf{p}_0, \mathbf{p}_k)$  denoting the angle between locations  $\mathbf{p}_0$  and  $\mathbf{p}_k$  in radians, and  $\mathbf{v}_k^y \sim \mathcal{N}(\mathbf{0}, \text{diag}(1, 0.1))$  the measurement noise. The above model can be rewritten as a third-order Markov form

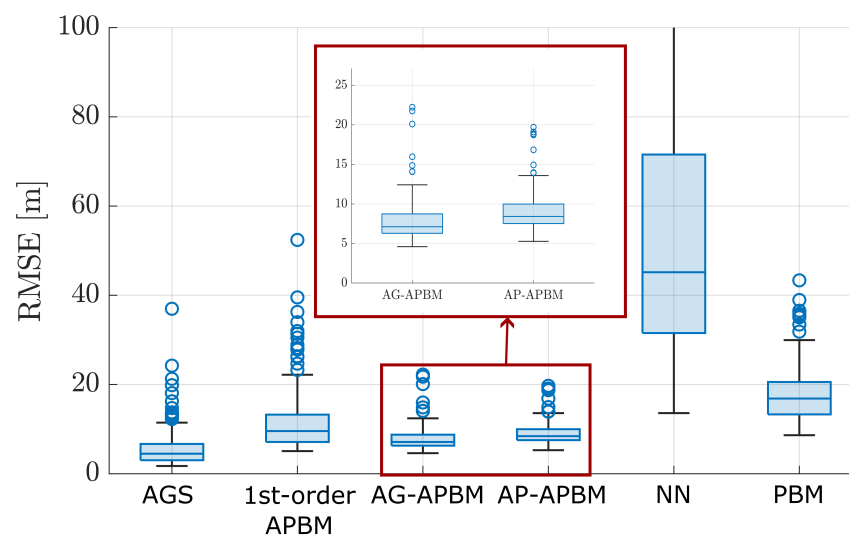
$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \dots, \mathbf{x}_{k-3}) + \mathbf{w}_{k-1} \quad (25)$$

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{v}_k. \quad (26)$$

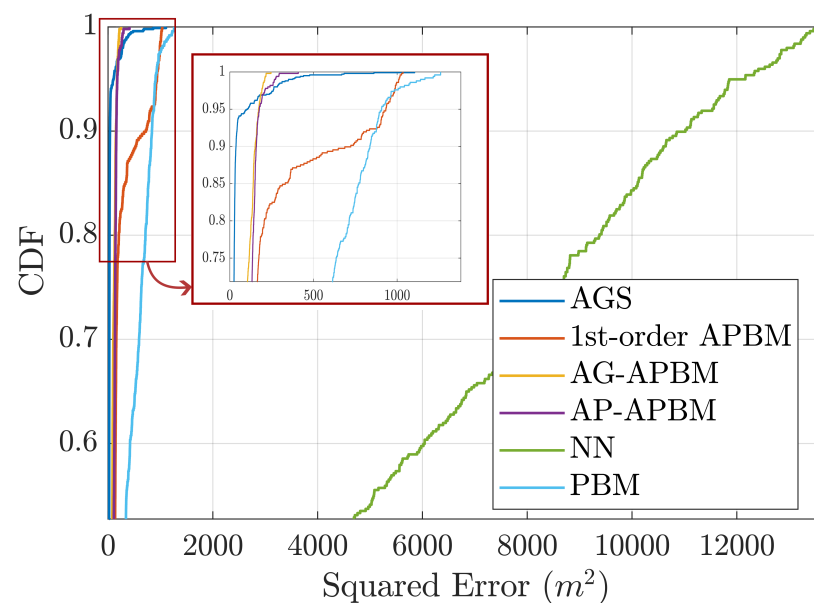
### 3.3. AG-APBM and AP-APBM Performance

We implement CKF to estimate the state of interest  $\mathbf{x} = [p_x, v_x, p_y, v_y]^\top$  in 200 MC experiments. We will compare the filtering performance of the AGS approach as described in (7), 1st-order Markov APBM in (3), AG-APBM in (10), AP-APBM in (14), pure NN, and pure PBM. It is noted that the AGS approach is considered as the benchmark in this tracking problem, since it has the full knowledge of the dynamics, including the Markovity order and the control input, while the other approaches are not aware of these facts. The constant-velocity model  $\tilde{f}(\mathbf{x}_{k-1}) = \mathbf{F}\mathbf{x}_{k-1}$  is used as the PBM. Here, the APBM is similar to Equation (19), which consists of the PBM and MLP, denoted  $\gamma_\phi(\cdot)$  and parameterized by  $\phi$ , with appropriate inputs such that  $g_\theta(\mathbf{x}_{k-1}, \cdot) = w_0 \mathbf{F}\mathbf{x}_{k-1} + w_1 \gamma_\phi(\mathbf{x}_{k-1}, \cdot)$ , where  $\theta = [w_0, w_1, \phi]$ . All NNs have one hidden layer with five hidden units and ReLU activation functions, and output layer with dimension  $d_x = 4$  and linear activation functions. It is noted that the six approaches leverage different-level information of the transition model. The AGS has full knowledge of the system, while 1st-order Markov APBM is only given the assumed PBM. The AG-APBM and AP-APBM are aware of the existence of the high-order Markov process, but not the exact order, where they use  $l = 5$  instead. Moreover, our reference is designed for the control problem, but the APBM approaches do not have the knowledge of the existence of the reference.

Figure 3 shows the box plot of the RMSE of each approach. Figure 4 shows the averaged experimental cumulative distribution function (CDF) over the 200 experiments. Both plots show statistically significant improvements of APBM approaches when compared with both PBM and NNs. In terms of the median, the error of 1st-order APBM is reduced by 41.2%, the error of AG-APBM is reduced by 58.8%, and the error of AP-APBM is reduced by 52.9% compared to PBM. The performances of APBMs are only surpassed by the true model, AGS. When comparing AG- and AP-APBM results the difference is minimal with AP-APBM leading to slightly higher errors. We also highlight that the two proposed approaches led to RMSE values that are very close to the ones obtained with the AGS, despite the perfect knowledge of the order of the Markov process and the transition model assumed in the AGS implementation.



**Figure 3.** Box plots for the RMSE of position estimations computed over 200 MC simulations. The central line in the box indicates the median. The bottom and top edges of the box denote the 25th and 75th percentiles, and the circles denotes the outliers. The AGS and PBM represents the true model and the constant-velocity model.



**Figure 4.** Each curve represents the average empirical cumulative distribution function (CDF) of squared error of position estimation. The average was computed over 200 MC simulations.

#### 4. Conclusions and Future Work

In this study, we extended the APBM framework to deal with high-order nonlinear Markov processes. To this end, we proposed two different implementations with different levels of computational complexity. In the first, we proposed the AG-APBM, where we augmented the state space with past states, thus considerably increasing the number of computations required for the estimation process. In the second approach, AP-APBM, we mitigated the additional complexity by approximating past states by their point estimates, eliminating the state augmentation requirements. Simulated experiments demonstrated the performance of the proposed models in an AR(3) model state estimation and a nonlinear third-order Markov target tracking scenario. The estimation results from the AR model indicated that the high-order AG- and AP-APBM reduced the error by about 30% compared

the PBM. Both of them outperformed the legacy 1st-order APBM since they leveraged more state information from the previous steps. Furthermore, the performance of AP-APBM only degrades slightly from the performance of AG-APBM, while significantly reducing the computational cost with 37.5% reduction in time cost and 20% reduction in memory usage based on our simulation environment. The tracking results showed that the AG-APBM and AP-APBM reduced the estimate error by 58.8% and 52.9% compared with the error of PBM even without precise knowledge of the order of the underlying Markov process. Both of them result in a lower RMSE than the standard 1st-order APBM does.

As we mentioned earlier, this article focuses on the state estimation under unknown dynamics (especially the unknown order of Markovity) by learning the system dynamics with PBM. In relation to this, future works could focus on: (1) dealing with unknown noise parameters. The noise statistics is also an important factor in the dynamics system. The noise can be estimated by the maximum likelihood method [29] or the correlation measurement difference method [33]. It is also shown that the data-driven algorithm is effective when learning the noise statistics, e.g., Long short-term memory Kalman filters [34], EKFNet [35] and KalmanNet [36]; (2) considering more general state-space distributions beyond the widely used Gaussian assumption. That is, while APBMs focused on Gaussian models, the general concept can be applied to non-Gaussian setups; and (3) using a different NN structure to learn the high-Markov dynamics. We implemented an MLP to learn the system dynamics in the simulation. The recurrent neural network (RNN), long-short term memory (LSTM) [37], and Transformer [38] may be used due to their mechanisms of forgetting and memorizing the past information especially when the order of the Markovity is not given.

**Author Contributions:** Conceptualization, T.I.; Methodology, S.T., J.D. and O.S.; Software, S.T. and T.I.; Validation, S.T.; Writing—original draft, S.T.; Writing—review & editing, J.D., O.S. and P.C.; Supervision, P.C.; Funding acquisition, P.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has been partially supported by the NSF under Awards ECCS-1845833 and CCF-2326559.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The article provides a definition for all the symbols and variables that are used in the derivations. For the convenience of the reader, a list of these notations is provided here.

$x_k$	State vector at time instance $k$
$y_k$	Measurement vector at time instance $k$
$w_k^x$	Noise of the dynamics model
$v_k^y$	Noise of the measurement model
$h(\cdot)$	Possibly nonlinear measurement model
$f(\cdot)$	Possibly nonlinear true dynamics model
$\bar{f}(\cdot)$	Physics-based model (PBM)
$\tilde{g}(\cdot)$	Augmented physics-based model (APBM)
$\theta$	Neural network (NN) parameters
$w_k^\theta$	Noise of NN parameter dynamics model
$\bar{\theta}$	Pseudo-measurement for NN parameter regularization
$v_k^\theta$	Noise of NN parameter pseudo-measurement model
$\tilde{x}_k$	Augmented state vector at time instance $k$

$\tilde{g}(\cdot)$	Augmented-state APBM (AG-APBM)
$\tilde{w}_k^x$	Noise of AG-APBM
$\hat{x}_k$	Estimated state at time instance $k$
$\delta(\cdot)$	Dirac delta function

## References

- Duník, J.; Biswas, S.K.; Dempster, A.G.; Pany, T.; Closas, P. State Estimation Methods in Navigation: Overview and Application. *IEEE Aerosp. Electron. Syst. Mag.* **2020**, *35*, 16–31. [\[CrossRef\]](#)
- Särkkä, S.; Svensson, L. *Bayesian Filtering and Smoothing*; Cambridge University Press: Cambridge, UK, 2023; Volume 17.
- Kalman, R.E. Contributions to the theory of optimal control. *Bol. Soc. Mat. Mex.* **1960**, *5*, 102–119.
- Wan, E.A.; Van Der Merwe, R. The unscented Kalman filter. In *Kalman Filtering and Neural Networks*; Wiley: Hoboken, NJ, USA, 2001; pp. 221–280.
- Arasaratnam, I.; Haykin, S. Cubature Kalman filters. *IEEE Trans. Autom. Control* **2009**, *54*, 1254–1269. [\[CrossRef\]](#)
- Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188. [\[CrossRef\]](#)
- Simon, D. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*; John Wiley & Sons: Hoboken, NJ, USA, 2006.
- Yan, S.; Gu, Z.; Park, J.H.; Shen, M. Fusion-Based Event-Triggered H-infinity State Estimation of Networked Autonomous Surface Vehicles With Measurement Outliers and Cyber-Attacks. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 7541–7551. [\[CrossRef\]](#)
- Xia, J.; Gao, S.; Qi, X.; Zhang, J.; Li, G. Distributed cubature H-infinity information filtering for target tracking against uncertain noise statistics. *Signal Process.* **2020**, *177*, 107725. [\[CrossRef\]](#)
- Chen, Z.; Zhou, J.; Zhou, F.; Xu, S. State-of-charge estimation of lithium-ion batteries based on improved H infinity filter algorithm and its novel equalization method. *J. Clean. Prod.* **2021**, *290*, 125180. [\[CrossRef\]](#)
- Haoqing, L.; Borsoi, R.A.; Imbiriba, T.; Closas, P.; Bermudez, J.C.M.; Erdoğan, D. Model-Based Deep Autoencoder Networks for Nonlinear Hyperspectral Unmixing. *IEEE Geosci. Remote. Sens. Lett.* **2022**, *19*, 1–5. [\[CrossRef\]](#)
- Imbiriba, T.; Wu, P.; LaMountain, G.; Erdoğan, D.; Closas, P. Recursive Gaussian Processes and Fingerprinting for Indoor Navigation. In Proceedings of the 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS), Portland, ON, USA, 20–23 April 2020; pp. 933–940.
- Chin, L. Application of neural networks in target tracking data fusion. *IEEE Trans. Aerosp. Electron. Syst.* **1994**, *30*, 281–287. [\[CrossRef\]](#)
- Vaidehi, V.; Chitra, N.; Krishnan, C.; Chokkalingam, M. Neural network aided Kalman filtering for multitarget tracking applications. *Comput. Electr. Eng.* **2001**, *27*, 217–228.
- Gao, C.; Yan, J.; Zhou, S.; Varshney, P.; Liu, H. Long short-term memory-based deep recurrent neural networks for target tracking. *Inf. Sci.* **2019**, *502*, 279–296. [\[CrossRef\]](#)
- Singhal, S.; Wu, L. Training multilayer perceptrons with the extended Kalman algorithm. *Adv. Neural Inf. Process. Syst.* **1988**, *1*. Available online: <https://proceedings.neurips.cc/paper/1988/file/38b3eff8baf56627478ec76a704e9b52-Paper.pdf> (accessed on 7 August 2024).
- Haykin, S. *Kalman Filtering and Neural Networks*; John Wiley & Sons: Hoboken, NJ, USA, 2004.
- Imbiriba, T.; Demirkaya, A.; Duník, J.; Straka, O.; Erdoğan, D.; Closas, P. Hybrid neural network augmented physics-based models for nonlinear filtering. In Proceedings of the 2022 25th International Conference on Information Fusion (FUSION), Linköping, Sweden, 4–7 July 2022; pp. 1–6.
- Imbiriba, T.; Straka, O.; Duník, J.; Closas, P. Augmented physics-based machine learning for navigation and tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 2692–2704. [\[CrossRef\]](#)
- Krogh, A. An introduction to hidden Markov models for biological sequences. In *New Comprehensive Biochemistry*; Elsevier: Amsterdam, The Netherlands, 1998; Volume 32, pp. 45–63.
- Lee, L.M.; Lee, J.C. A study on high-order hidden Markov models and applications to speech recognition. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 682–690.
- Salnikov, V.; Schaub, M.T.; Lambiotte, R. Using higher-order Markov models to reveal flow-based communities in networks. *Sci. Rep.* **2016**, *6*, 23194. [\[CrossRef\]](#)
- Urteaga, I.; Djurić, P.M. Sequential estimation of hidden ARMA processes by particle filtering—Part one. *IEEE Trans. Signal Process.* **2016**, *65*, 482–493. [\[CrossRef\]](#)
- Djuric, P.M.; Kay, S.M. Order selection of autoregressive models. *IEEE Trans. Signal Process.* **1992**, *40*, 2829–2833. [\[CrossRef\]](#)
- Bishop, C.M.; Nasrabadi, N.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4.
- Raftery, A.E. A model for high-order Markov chains. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1985**, *47*, 528–539. [\[CrossRef\]](#)
- Terwijn, B.; Porta, J.M.; Kröse, B.J. A particle filter to estimate non-Markovian states. In Proceedings of the International Conference on Intelligent Autonomous Systems, IAS, Singapore, 22–25 June 2004; Volume 4, pp. 1062–1069.
- Jin, B.; Guo, J.; He, D.; Guo, W. Adaptive Kalman filtering based on optimal autoregressive predictive model. *GPS Solut.* **2017**, *21*, 307–317. [\[CrossRef\]](#)

29. Shumway, R.H.; Stoffer, D.S.; Stoffer, D.S. *Time Series Analysis and Its Applications*; Springer: Berlin/Heidelberg, Germany, 2000; Volume 3.
30. Kerns, A.J.; Shepard, D.P.; Bhatti, J.A.; Humphreys, T.E. Unmanned aircraft capture and control via GPS spoofing. *J. Field Robot.* **2014**, *31*, 617–636. [[CrossRef](#)]
31. Kling, M.T.; Lau, D.; Witham, K.L.; Closas, P.; LaMountain, G.M. System for Closed-Loop GNSS Simulation US Patent Application No. 17/662,822, 10 May 2022.
32. Bar-Shalom, Y.; Li, X.R.; Kirubarajan, T. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*; John Wiley & Sons: Hoboken, NJ, USA, 2001.
33. Kost, O.; Duník, J.; Straka, O. Measurement difference method: A universal tool for noise identification. *IEEE Trans. Autom. Control* **2022**, *68*, 1792–1799. [[CrossRef](#)]
34. Coskun, H.; Achilles, F.; DiPietro, R.; Navab, N.; Tombari, F. Long short-term memory Kalman filters: Recurrent neural estimators for pose regularization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5524–5532.
35. Xu, L.; Niu, R. EKFNet: Learning system noise statistics from measurement data. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 4560–4564.
36. Revach, G.; Shlezinger, N.; Ni, X.; Escoriza, A.L.; Van Sloun, R.J.; Eldar, Y.C. KalmanNet: Neural network aided Kalman filtering for partially known dynamics. *IEEE Trans. Signal Process.* **2022**, *70*, 1532–1547. [[CrossRef](#)]
37. Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [[CrossRef](#)]
38. Vaswani, A. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; NIPS 2017.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.