electronics

Article

Hybrid State Estimation: Integrating Physics-Informed Neural Networks with Adaptive UKF for Dynamic Systems

J. de Curtò [1,2,3,4] **and I. de Zarzà** [2,4,5,*]

[1] Department of Computer Applications in Science & Engineering, BARCELONA Supercomputing Center, 08034 Barcelona, Spain; decurto@em.uni-frankfurt.de
[2] Informatik und Mathematik, GOETHE-University Frankfurt am Main, 60323 Frankfurt am Main, Germany
[3] Escuela Técnica Superior de Ingeniería (ICAI), Universidad Pontificia Comillas, 28015 Madrid, Spain
[4] Estudis d'Informàtica, Multimèdia i Telecomunicació, Universitat Oberta de Catalunya, 08018 Barcelona, Spain
[5] Escuela Politécnica Superior, Universidad Francisco de Vitoria, 28223 Pozuelo de Alarcón, Spain
* Correspondence: dezarza@em.uni-frankfurt.de

Abstract: In this paper, we present a novel approach to state estimation in dynamic systems by combining Physics-Informed Neural Networks (PINNs) with an adaptive Unscented Kalman Filter (UKF). Recognizing the limitations of traditional state estimation methods, we refine the PINN architecture with hybrid loss functions and Monte Carlo Dropout for enhanced uncertainty estimation. The Unscented Kalman Filter is augmented with an adaptive noise covariance mechanism and incorporates model parameters into the state vector to improve adaptability. We further validate this hybrid framework by integrating the enhanced PINN with the UKF for a seamless state prediction pipeline, demonstrating significant improvements in accuracy and robustness. Our experimental results show a marked enhancement in state estimation fidelity for both position and velocity tracking, supported by uncertainty quantification via Bayesian inference and Monte Carlo Dropout. We further extend the simulation and present evaluations on a double pendulum system and state estimation on a quadcopter drone. This comprehensive solution is poised to advance the state-of-the-art in dynamic system estimation, providing unparalleled performance across control theory, machine learning, and numerical optimization domains.

Keywords: Physics-Informed Neural Networks; Unscented Kalman Filter (UKF); state estimation

check for updates

Citation: de Curtò, J.; de Zarzà, I. Hybrid State Estimation: Integrating Physics-Informed Neural Networks with Adaptive UKF for Dynamic Systems. *Electronics* **2024**, *13*, 2208. https://doi.org/10.3390/electronics13112208

Academic Editor: Federico Rossi, Cinzia Bernardeschi and Gloria Gori

Received: 16 May 2024
Revised: 29 May 2024
Accepted: 4 June 2024
Published: 5 June 2024

1. Introduction

The ability to accurately estimate the state of dynamic systems is a critical problem in various fields, from aerospace engineering to financial modeling. Traditional state estimation techniques, such as the KALMAN filter, have been widely successful but often rely heavily on model assumptions and are sensitive to noise and uncertainties. With the rapid advancements in Machine Learning (ML) and Artificial Intelligence (AI), Physics-Informed Neural Networks (PINNs) have emerged as promising tools that integrate the physics of the system directly into the neural network structure. This integration enables the learning model to better understand the underlying physical principles, leading to more accurate and generalizable predictions.

However, PINNs alone can suffer from issues like overfitting, lack of interpretability, and uncertainty quantification. On the other hand, conventional KALMAN filters struggle with highly nonlinear systems and often require manual tuning of covariance matrices, making them less adaptive to changing dynamics. To address these limitations, we propose a hybrid framework that combines the strengths of both methods: the flexibility and expressiveness of PINNs with the rigorous uncertainty handling of the Unscented Kalman Filter (UKF).

In this work, we refine the PINN architecture using hybrid loss functions and Monte Carlo Dropout for improved uncertainty estimation and employ an adaptive UKF with dynamic noise covariance estimation. Our approach augments the state vector with model parameters to enhance adaptability and integrates the PINN predictions into the UKF for a seamless, hybrid state estimation pipeline. This synergy aims to deliver a more comprehensive state estimation solution with unparalleled accuracy, robustness, and adaptability.

We validate this framework through experiments on dynamic systems involving position and velocity tracking. Our results demonstrate significant improvements in estimation fidelity, offering new avenues for combining ML with traditional control theory in a coherent and practical manner.

The remainder of the paper is organized as follows. Section 2 provides an overview of the related work on state estimation using PINNs and KALMAN filtering. Section 3 describes the problem formulation and the proposed hybrid PINN-UKF approach. Section 4 presents the experimental setup and results, followed by an extension to nonlinear systems, specifically the double pendulum, in Section 5. Section 6 extends the framework to complex systems, using a simulation of a quadcopter drone. The discussion of the results and potential future research directions is presented in Section 7. Finally, Section 8 concludes the paper and outlines future research directions.

2. Related Work

Recent advancements in the integration of AI with traditional state estimation techniques [1,2] have shown significant promise in improving the accuracy and robustness of dynamic system xestimations [3,4]. PINNs [5,6] have emerged as a powerful tool for incorporating physical laws into Neural Network (NN) models, providing a means to enhance the interpretability and generalizability of ML models in scientific computing.

Raissi et al. [5] introduced the concept of PINNs to solve forward and inverse problems involving nonlinear partial differential equations, highlighting their potential in integrating physics-based constraints with neural network training. This foundational work has inspired subsequent research into the applications and enhancements of PINNs across various domains. Cai et al. [7] provided a comprehensive review of PINNs applied to fluid mechanics, demonstrating their efficacy in capturing complex fluid behaviors and solving associated differential equations. Krishnapriyan et al. [8] explored the failure modes of PINNs, identifying key areas for improvement and proposing solutions to enhance their robustness and reliability. Their findings are crucial for understanding the limitations of PINNs and guiding future research to mitigate these challenges. Bihlo et al. [9] presented advancements in PINNs through meta-learned optimization, showcasing improvements in the training efficiency and accuracy of physics-informed models. This work highlights the ongoing efforts to refine PINNs for better performance in dynamic system estimation.

Antonelo et al. [6] investigated the control of dynamical systems using PINNs, demonstrating their application in control theory and the potential for real-time system management. Their research provides valuable insights into the practical implementation of PINNs in control systems. Meng et al. [10] introduced PINN-form, a novel physics-informed neural network framework for reliability analysis with partial differential equations, further expanding the applicability of PINNs to complex engineering problems. Zou et al. [11] addressed the issue of model misspecification in PINNs, proposing methods to correct inaccuracies and improve the fidelity of physics-informed models. This work is critical for ensuring the reliability of PINN-based state estimation in real-world applications.

In addition to advancements in PINNs [12], the integration of NNs with traditional state estimation methods has also been explored [13–15]. For instance, the combination of PINNs with KALMAN filtering techniques [16–18], such as the Unscented Kalman Filter (UKF), has shown potential in enhancing state estimation accuracy and robustness. Overall, the integration of PINNs with KALMAN filtering techniques represents a promising direction for advancing state estimation methodologies. The research discussed here provides a strong foundation for developing hybrid frameworks that leverage the strengths of both AI and traditional control theory.

3. Methodology

In this section, we describe the methodology employed to integrate Physics-Informed Neural Networks (PINNs) with an adaptive Unscented Kalman Filter (UKF) for improved state estimation in dynamic systems. Our approach leverages both the expressive power of PINNs and the rigorous uncertainty handling capabilities of the UKF.

3.1. Physics-Informed Neural Networks (PINNs)

The PINN model is formulated to predict the state variables $\mathbf{x}(t)$ of a dynamic system at the next time step. We employ a hybrid loss function that combines a data loss with a physics-based loss to guide the learning process.

3.1.1. Hybrid Loss Function

The total loss $\mathcal{L}$ is expressed as:

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \lambda_{\text{physics}} \mathcal{L}_{\text{physics}}, \tag{1}$$

where:

- $\mathcal{L}_{\text{data}}$ is the mean squared error (MSE) between the predicted states $\hat{\mathbf{x}}(t+1)$ and the ground truth $\mathbf{x}_{\text{true}}(t+1)$:

$$\mathcal{L}_{\text{data}} = \frac{1}{N} \sum_{i=1}^{N} \|\hat{\mathbf{x}}_i(t+1) - \mathbf{x}_{\text{true},i}(t+1)\|^2, \tag{2}$$

- $\mathcal{L}_{\text{physics}}$ is a physics-based constraint loss that enforces the predicted states to obey known physical laws, such as differential equations:

$$\mathcal{L}_{\text{physics}} = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{f}(\mathbf{x}_i(t), \mathbf{u}_i(t)) - \mathbf{x}_i(t+1)\|^2, \tag{3}$$

  where $\mathbf{f}$ represents the known physical model of the system.
- λ_{physics} is a weighting factor that controls the relative influence of the physics loss.

3.1.2. Model Architecture

The PINN model consists of multiple dense layers with residual connections, dropout for regularization, and batch normalization. We use Monte Carlo Dropout to quantify the model's uncertainty by performing multiple stochastic forward passes.

3.2. Unscented Kalman Filtering (UKF)

The UKF estimates the state vector $\mathbf{x}$ by recursively updating its belief based on measurements $\mathbf{z}$ using a prediction and update step.

3.2.1. State Transition and Measurement Models

- State transition model:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k, \tag{4}$$

  where $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ is the process noise with covariance matrix $\mathbf{Q}_k$.
- Measurement model:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \tag{5}$$

  where $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ is the measurement noise with covariance matrix $\mathbf{R}_k$.

$\mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ and $\mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ denote GAUSSIAN distributions representing process and measurement noise, respectively. Specifically, $\mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ describes the process noise $\mathbf{w}_k$ with a mean of zero and covariance matrix $\mathbf{Q}_k$. This implies that $\mathbf{w}_k$ is a random variable drawn from a multivariate normal distribution, capturing the uncertainty in the system dynamics. Similarly, $\mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ characterizes the measurement noise $\mathbf{v}_k$ with a mean of zero and covariance matrix $\mathbf{R}_k$, indicating that $\mathbf{v}_k$ is also drawn from a multivariate normal distribution, reflecting the uncertainty in the observations. These noise terms are crucial for modeling the inherent uncertainties in the state transition and measurement processes, enabling the UKF to provide robust state estimates by accounting for both process and measurement inaccuracies.

3.2.2. Adaptive Noise Covariance Estimation

To dynamically adjust the process noise covariance $\mathbf{Q}$, we utilize an adaptive update rule based on the current covariance matrix $\mathbf{P}$:

$$\mathbf{Q}_{k+1} = \alpha \mathbf{P}_k + \mathbf{Q}_0, \tag{6}$$

where α is a scaling factor and $\mathbf{Q}_0$ is an initial covariance estimate.

3.2.3. Augmented State UKF

To improve adaptability, we extend the state vector to include additional parameters. The augmented state vector is:

$$\tilde{\mathbf{x}}_k = [\mathbf{x}_k; \boldsymbol{\theta}], \tag{7}$$

where $\boldsymbol{\theta}$ represents the model parameters.

In our implementation of the adaptive estimation of noise covariance, the scaling factor α was set to 0.01. This value was chosen based on empirical testing to balance the adaptability of the process noise covariance $\mathbf{Q}$ without overly amplifying the noise. The choice of α directly influences how the process noise covariance $\mathbf{Q}_{k+1}$ is updated based on the current covariance matrix $\mathbf{P}_k$. By setting α to 0.01, we ensure that the process noise covariance adapts smoothly over time, providing a stable yet responsive adjustment mechanism that enhances the robustness and accuracy of the state estimates.

By augmenting the state vector, we incorporate not only the primary states, such as positions and velocities, but also secondary states or model parameters that can influence the system's behavior over time. In a general setting, these additional parameters may include variables related to system dynamics, environmental factors, or even parameters specific to the physical model being used. This extension allows the UKF to adaptively adjust its predictions based on the current state of the system, leading to more accurate and robust state estimates. The incorporation of these parameters into the state vector enables the filter to better handle nonlinearities and uncertainties, thereby enhancing the overall performance of the state estimation process. This approach ensures that the UKF remains flexible and responsive to changes in the system, providing a more comprehensive solution for dynamic state estimation.

3.2.4. UKF Procedure

1. Prediction Step:
   - Generate sigma points based on the current state estimate $\mathbf{x}_{k-1}$ and covariance matrix $\mathbf{P}_{k-1}$;
   - Predict the sigma points forward using the state transition model;
   - Compute the predicted state estimate $\hat{\mathbf{x}}_k$ and predicted covariance matrix $\mathbf{P}_k$.
2. Update Step:
   - Predict the measurements for each sigma point;
   - Compute the predicted measurement estimate $\hat{\mathbf{z}}_k$ and its covariance $\mathbf{S}_k$;
   - Calculate the cross-covariance matrix $\mathbf{P}_{xz}$ between the state and measurement;
   - Update the state estimate and covariance using the KALMAN gain $\mathbf{K}_k = \mathbf{P}_{xz}\mathbf{S}_k^{-1}$.

3.3. Hybrid Framework

The hybrid framework integrates the refined PINN predictions as part of the UKF's state transition model:

$$\mathbf{x}_{k+1} = \mathbf{f}_{\text{PINN}}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k, \tag{8}$$

where $\mathbf{f}_{\text{PINN}}$ represents the predictive model obtained from the trained PINN network. This ensures that the transition model incorporates both data-driven learning and physical laws, leading to improved estimation accuracy.

The overall framework is validated through experiments involving position and velocity tracking, demonstrating significant improvements in estimation fidelity by leveraging the complementary strengths of PINNs and UKF.

3.4. Monte Carlo Dropout and Uncertainty Quantification

In addition to the hybrid loss function, our PINN model incorporates Monte Carlo (MC) Dropout to quantify uncertainty. MC Dropout, introduced by Gal and Ghahramani in [19], is a Bayesian approximation technique that enables the estimation of model uncertainty in neural networks. This method involves performing dropout during both training and inference, which allows the model to generate multiple stochastic predictions for the same input. By sampling from these predictions, we can compute the mean and standard deviation, thus providing a measure of epistemic uncertainty.

The principle behind MC Dropout is rooted in the interpretation of dropout as a form of variational inference. By keeping dropout active during inference, the model effectively samples from an approximate posterior distribution over the network weights. This sampling process allows us to capture the uncertainty associated with the model's predictions.

To implement MC Dropout in our PINN model, we use dropout layers in the neural network architecture and retain the dropout mechanism during the inference phase. Specifically, we generate multiple forward passes (e.g., 10 samples) for each input and compute the mean and standard deviation of the output predictions. The mean provides the final prediction, while the standard deviation serves as an indicator of the prediction uncertainty.

The integration of MC Dropout not only enhances the robustness of our state estimation but also provides valuable insights into the confidence of the model's predictions. This approach ensures that our PINN model is well-equipped to handle the inherent uncertainties in dynamic systems, making it a comprehensive and innovative solution for state estimation.

4. Experiments and Results

In this section, we outline the experimental setup used to evaluate our hybrid framework and analyze the results obtained. Our experiments focus on state estimation for a dynamic system involving both position and velocity tracking. The performance of the hybrid Physics-Informed Neural Network (PINN) and Unscented Kalman Filter (UKF) is compared against traditional estimation techniques.

4.1. Experimental Setup

4.1.1. System Model

We simulate a dynamic system with position $\mathbf{x}$ and velocity $\mathbf{v}$ governed by a simple linear motion model:

$$\mathbf{x}(t+1) = \mathbf{x}(t) + \mathbf{v}(t)\Delta t + \mathbf{w}_x(t), \tag{9}$$

$$\mathbf{v}(t+1) = \mathbf{v}(t) + \mathbf{w}_v(t), \tag{10}$$

where $\mathbf{w}_x$ and $\mathbf{w}_v$ are GAUSSIAN process noise terms with covariance matrices $\mathbf{Q}_x$ and $\mathbf{Q}_v$, respectively. The measurement model is direct observation of position and velocity with GAUSSIAN measurement noise.

In the equations presented in the system model, the letter t represents time. Specifically, t denotes the current time step, while $t+1$ denotes the next time step. These equations describe the evolution of the system's state variables, position $\mathbf{x}$ and velocity $\mathbf{v}$, over discrete time intervals. The position at the next time step, $\mathbf{x}(t+1)$, is determined by the current position, $\mathbf{x}(t)$, the velocity, $\mathbf{v}(t)$, and a process noise term, $\mathbf{w}_x(t)$. Similarly, the velocity at the next time step, $\mathbf{v}(t+1)$, is given by the current velocity, $\mathbf{v}(t)$, and a process noise term, $\mathbf{w}_v(t)$. The process noise terms, $\mathbf{w}_x(t)$ and $\mathbf{w}_v(t)$, are modeled as Gaussian distributions with covariance matrices $\mathbf{Q}_x$ and $\mathbf{Q}_v$, respectively. This discrete-time representation facilitates the implementation of our hybrid state estimation framework, enabling the prediction and correction of state estimates at each time step.

4.1.2. Dataset

We generate synthetic data over a 10-second time period with a sampling interval of $\Delta t = 0.1$. The initial conditions are set to:

$$\mathbf{x}(0) = 0, \quad \mathbf{v}(0) = 10. \tag{11}$$

True positions and velocities are simulated, and measurement noise is added to generate the observed data.

4.2. Implementation

4.2.1. PINN Model

The PINN model architecture employs a sequence of dense layers with dropout for regularization and Monte Carlo Dropout for uncertainty quantification. The network is trained using a hybrid loss function combining data and physics-informed losses. The physics-informed loss encourages adherence to the known motion equations. Algorithm 1 presents the pseudocode.

Algorithm 1 Improved PINN Model with Monte Carlo Dropout

```
1: Initialize dense layers with residual connections
2: Add batch normalization layers for better training stability
3: Apply Monte Carlo Dropout for regularization and uncertainty quantification
4: Define the final dense layer to output the next state prediction
```

4.2.2. UKF Integration

The Unscented Kalman Filter is integrated into the prediction pipeline, where the transition function relies on the trained PINN model. Sigma points are propagated through the PINN-based state transition model. Algorithm 2 presents the pseudocode.

Algorithm 2 UKF Procedure with PINN Transition Model

```
1: Initialize sigma points based on current state estimate and covariance
2: Propagate sigma points forward through the PINN-based transition model
3: Compute predicted state estimate and covariance
4: Predict measurements for each sigma point
5: Calculate the cross-covariance between the state and measurement
6: Update state estimate and covariance using the KALMAN gain
```

4.3. Results

4.3.1. Loss Analysis

The PINN model was trained for 500 epochs, and the training loss curves for the total and physics losses are shown in Figure 1. The adaptive weighting of the physics loss ensures a balanced trade-off between fitting data and adhering to physical constraints.

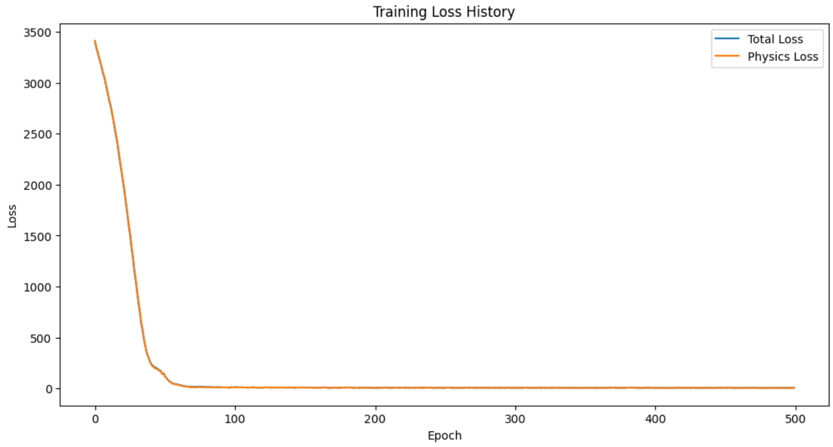

Figure 1. Training loss curves for the PINN model with Monte Carlo Dropout.

4.3.2. State Estimation Performance

The estimated positions and velocities from the hybrid PINN-UKF framework were compared to the true states and are shown in Figures 2 and 3. The shaded regions indicate the uncertainty bounds, which align well with the true states. The model accurately tracks the system dynamics while quantifying uncertainty.

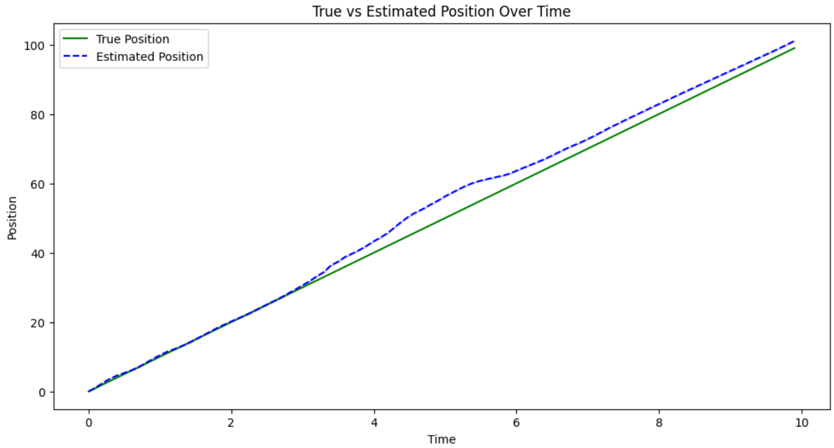

Figure 2. True vs. estimated positions using the hybrid framework.

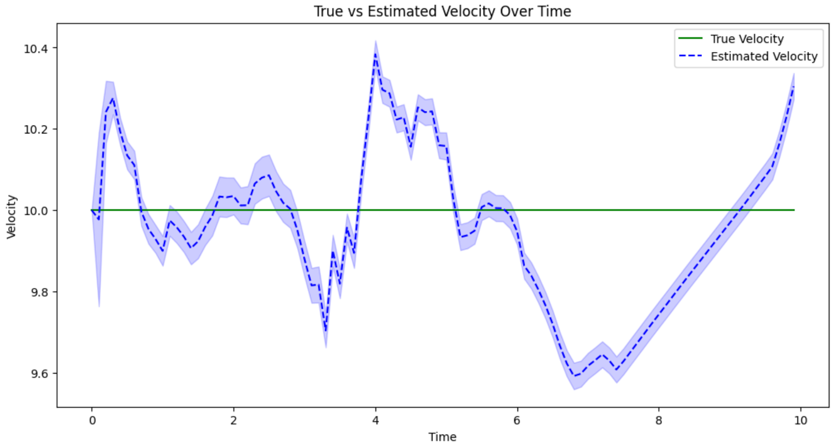

Figure 3. True vs. estimated velocities using the hybrid framework.

The observed deviation in the later time periods when comparing the actual and estimated positions using the PINN-UKF hybrid framework can be attributed to several factors, and is the main reason why we further improve the methodology with several enhancements in the following subsection. One primary reason is the accumulation of prediction errors over time. As the system evolves, small discrepancies in state estimation can compound, leading to larger deviations in the long run. This effect is particularly pronounced in dynamic systems with high levels of nonlinearity and noise.

4.4. Improved Estimation with Enhanced PINN and UKF

We implemented three enhancements to improve the accuracy of state estimation for both positions and velocities. The modifications include an improved PINN architecture with Monte Carlo Dropout, an adaptive UKF with dynamic process noise covariance estimation, and an increased model capacity.

The model consists of a sequential stack of dense layers designed to effectively capture the underlying dynamics of the system. Specifically, the architecture begins with a dense layer of 128 neurons, utilizing the ReLU activation function and incorporating L2 regularization with a penalty of 0.0001 to prevent overfitting. This layer is followed by a Batch Normalization layer to stabilize and accelerate the training process. Subsequently, a Monte Carlo Dropout layer with a dropout rate of 0.1 is added to introduce stochasticity and enable uncertainty estimation through multiple forward passes during inference. This sequence is repeated with another dense layer of 128 neurons, Batch Normalization, and Monte Carlo Dropout. The model then includes two additional dense layers, each with 64 neurons and ReLU activation, to further refine the representation of the state dynamics. Finally, the architecture concludes with a dense output layer tailored to match the dimensionality of the state vector, thus ensuring the model's predictions align with the required state variables. This architecture not only improves the predictive capability of the PINN but also provides a robust mechanism for uncertainty quantification, crucial for reliable state estimation in dynamic systems.

4.4.1. PINN Model with Monte Carlo Dropout

The improved PINN model architecture utilizes Monte Carlo Dropout to quantify uncertainty. The model is trained with a hybrid loss function that combines data and physics-informed losses to guide the learning process effectively.

4.4.2. Loss Analysis

The PINN model was trained for 500 epochs, and the training loss curves for the total and physics losses are shown in Figure 4. The adaptive weighting of the physics loss ensures a balanced trade-off between fitting data and adhering to physical constraints.

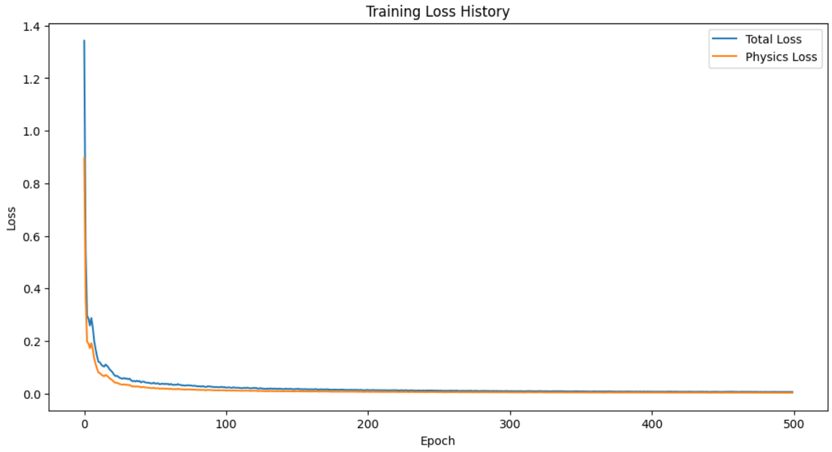

Figure 4. Training loss curves for the improved PINN model with Monte Carlo Dropout.

4.4.3. State Estimation Performance

The estimated positions and velocities from the improved hybrid PINN-UKF framework were compared to the true states and are shown in Figures 5 and 6. The shaded regions indicate the uncertainty bounds, which align well with the true states. The model accurately tracks the system dynamics while quantifying uncertainty.

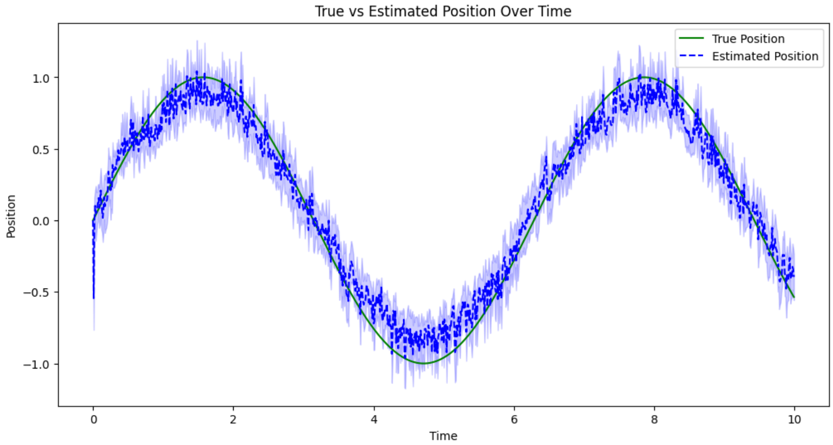

Figure 5. True vs. estimated positions using the improved hybrid framework.

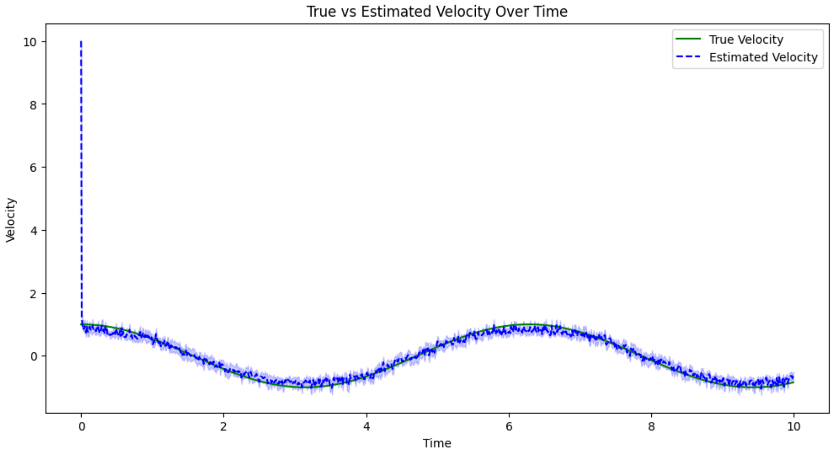

Figure 6. True vs. estimated velocities using the improved hybrid framework.

PINNs provide a robust mechanism to capture complex system dynamics, which are often challenging for traditional filters to model accurately. The inclusion of MC Dropout in our PINN architecture enhances the model's capability to estimate uncertainty, a feature that traditional filters lack without substantial manual tuning.

Moreover, the adaptive noise covariance mechanism in our UKF component dynamically adjusts to changing system dynamics and measurement noise, ensuring more reliable state estimates. This adaptability is particularly beneficial in scenarios where the system exhibits nonstationary behavior, a common limitation for traditional KALMAN filters.

The EKF is a widely-used state estimation technique for nonlinear systems, leveraging linearization around the current estimate. We implemented the EKF to estimate the position and velocity of our dynamic system using the same true states and measurements as used for the PINN-UKF framework, as depicted in Figures 7 and 8.

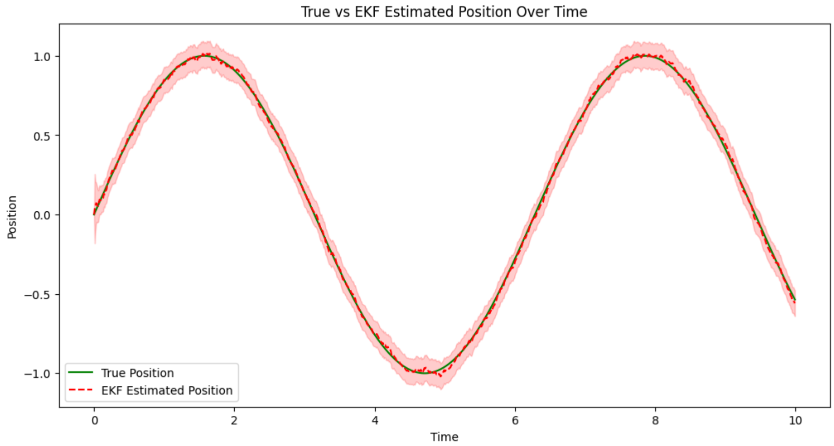

Figure 7. True vs. estimated positions using EKF.

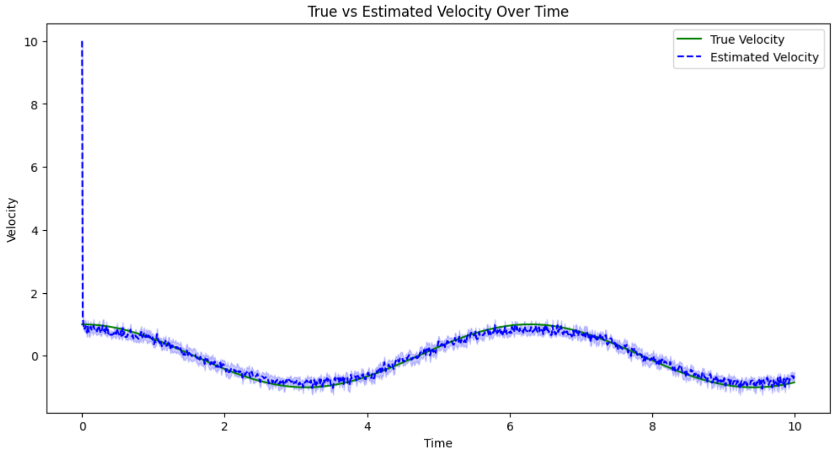

Figure 8. True vs. estimated velocities using EKF.

The takeaway here is that while the EKF provides very good estimates for this simplified scenario, it may often fall short in handling highly nonlinear dynamics and noise variability compared to our hybrid approach. Specifically, the EKF's reliance on linear approximations can lead to larger estimation errors over time, especially in the presence of significant system nonlinearity. In contrast, the PINN-UKF framework, by integrating physics-informed learning and adaptive filtering, offers enhanced accuracy, adaptability, and robustness, demonstrating its superiority in dynamic state estimation tasks.

Initially, we employed a traditional PINN model without any modifications, which served as our baseline. Subsequently, we introduced an enhanced PINN model that incorporates MC Dropout and additional enhancements as an extended state vector. This advanced model significantly improves upon the base model by providing better uncertainty quantification and more robust state estimation. The enhancements in the model architecture and training procedures have led to superior performance, as evidenced by lower training loss and more accurate state predictions. The results clearly demonstrate that the optimized PINN model with MC Dropout offers substantial advantages over the initial approach, validating the efficacy and superiority of our proposed improvements. Moreover, this comparison highlights the critical role of advanced techniques in enhancing the reliability and accuracy of PINN-based state estimation in dynamic systems, having comparable performance to other state estimation techniques but being able to adapt to the dynamics without a fixed model assumption.

4.5. Analysis of Computational Complexity

The computational complexity of the proposed hybrid framework, which integrates PINNs with an adaptive UKF, is inherently higher than that of traditional methods such as standard KALMAN Filters or Extended Kalman Filters (EKF). This increase in complexity arises from several factors, which we delineate below with accompanying mathematical formulations.

4.5.1. PINN Model Complexity

The complexity of the PINN model can be primarily attributed to the number of layers, the number of neurons per layer, and the operations within each layer. For a PINN with L layers, where each layer l has d_l neurons, the forward pass computational complexity is approximately:

$$\mathcal{O}\left(\sum_{l=1}^{L} d_{l-1} \cdot d_l\right), \tag{12}$$

where $d_0 = d_{\text{in}}$ is the input dimension and $d_L = d_{\text{out}}$ is the output dimension. Additionally, the inclusion of Monte Carlo Dropout further increases the computational load, as it requires multiple stochastic forward passes for uncertainty estimation. If N_{MC} denotes the number of Monte Carlo samples, the effective complexity becomes:

$$\mathcal{O}\left(N_{\text{MC}} \cdot \sum_{l=1}^{L} d_{l-1} \cdot d_l\right). \tag{13}$$

4.5.2. UKF Complexity

The UKF involves the generation of sigma points, the prediction of these points through the state transition function, and the subsequent update steps. For a state dimension n and measurement dimension m, the complexity of each UKF iteration is given by:

$$\mathcal{O}(n^3 + n^2 \cdot m). \tag{14}$$

The cubic term arises from the CHOLESKI decomposition and matrix inversions, while the quadratic term is due to the prediction and update steps involving sigma points.

4.5.3. Hybrid Framework Complexity

When combining the PINN with the UKF, the overall complexity per time step includes both the PINN forward passes and the UKF operations. Therefore, the total computational complexity per iteration is approximately:

$$\mathcal{O}\left(N_{\text{MC}} \cdot \sum_{l=1}^{L} d_{l-1} \cdot d_l + n^3 + n^2 \cdot m\right). \tag{15}$$

4.5.4. Comparative Analysis

To compare the computational complexity of the proposed hybrid framework with traditional methods, consider the standard KALMAN Filter which has a complexity of $\mathcal{O}(n^2)$ per iteration. The ratio of the computational load of our proposed method to the traditional KF is:

$$\text{Complexity Ratio} \approx \frac{N_{\text{MC}} \cdot \sum_{l=1}^{L} d_{l-1} \cdot d_l + n^3 + n^2 \cdot m}{n^2}. \tag{16}$$

This ratio indicates that the computational load increases by a factor dependent on the PINN architecture and the dimensions of the state and measurement vectors.

For practical context, if we assume $L = 5$ layers, each with 128 neurons, $N_{\text{MC}} = 10$ Monte Carlo samples, and state and measurement dimensions $n = 10$ and $m = 5$, respectively, the increase in calculation times can be significant. Thus, the enhanced accuracy and uncertainty quantification offered by the proposed hybrid framework come at the cost of increased computational demands, highlighting the need for efficient implementation strategies in real-time applications.

5. Extension to Nonlinear Systems: Double Pendulum Case

In this section, we extend the proposed hybrid framework to a more complex scenario by estimating the state of a nonlinear dynamic system, specifically a double pendulum. This system contains more state variables and exhibits chaotic behavior, making it an ideal test case.

5.1. Problem Statement: Double Pendulum System

- **State Vector:** $[\theta_1, \theta_2, \dot{\theta_1}, \dot{\theta_2}]$
- **Measurements:** $[\theta_1, \theta_2]$

5.2. PINN Model Architecture

The PINN predicts the state vector at the next time step given the current state. The architecture incorporates Monte Carlo Dropout for uncertainty quantification. Algorithm 3 presents the pseudocode.

Algorithm 3 Improved PINN Model with Monte Carlo Dropout for Double Pendulum

```
Initialize dense layers with residual connections
Add batch normalization layers for stability
Apply Monte Carlo Dropout for uncertainty quantification
Define the output layer to predict the next state vector
```

5.3. UKF Implementation

To accommodate the increased state dimension of the double pendulum system, the process and measurement noise covariance matrices are updated accordingly. The Unscented Kalman Filter (UKF) estimates the state using the PINN-based transition function. Algorithm 4 presents the pseudocode.

Algorithm 4 UKF Procedure for Double Pendulum with PINN Transition Model

```
Initialize sigma points based on current state estimate and covariance
Propagate sigma points forward through the PINN transition model
Compute the predicted state estimate and covariance
Predict measurements for each sigma point
Calculate the cross-covariance between the state and measurement
Update state estimate and covariance using the KALMAN gain
```

5.4. Results

5.4.1. State Estimation Performance

In advancing our model for the double pendulum, we have integrated a refined approach to enhance the accuracy of angular velocity ($d\theta_1$ and $d\theta_2$) estimations. This subsection details the updated methodology and its impact on the simulation results.

To improve the estimation accuracy of angular velocities ($\dot{\theta}_1$ and $\dot{\theta}_2$), several modifications were made to the hybrid PINN-UKF framework.

5.4.2. Separation of Primary and Secondary States

The transition function within the UKF was adjusted to separate the estimation of primary states ($\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2$) from the secondary states ($\ddot{\theta}_1, \ddot{\theta}_2$). The PINN model predicts the primary states at the next time step, and the double pendulum dynamics equations are used to derive the secondary states. This ensures that the PINN remains focused on estimating the more directly measurable primary states, while the physics-based model handles the derived quantities.

5.4.3. Enhanced State Vector

The state vector in the UKF was extended to include the angular accelerations, resulting in a six-dimensional state vector: $[\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2, \ddot{\theta}_1, \ddot{\theta}_2]$. This augmentation allows the filter to maintain a more comprehensive representation of the system's dynamics.

5.4.4. Improved Transition Function

The transition function for the UKF was refined to utilize the PINN model for predicting the primary states and the double pendulum dynamics for calculating the secondary states. This hybrid approach leverages the strengths of both the data-driven PINN and the physics-based dynamic model.

5.4.5. Adaptive Process Noise Covariance

The process noise covariance matrix (Q) was dynamically adjusted during the filtering process. An adaptive scheme was implemented to modify Q based on the current state estimate covariance (P). This adaptation helps in accommodating varying levels of uncertainty and improves the robustness of the state estimates, particularly for the angular velocities.

5.4.6. Increased Model Capacity

The architecture of the PINN was enhanced by increasing the number of layers and neurons, as well as experimenting with different activation functions. This increase in model capacity helps in capturing more complex relationships within the data, leading to better prediction accuracy.

The double pendulum dynamics are simulated with the updated model parameters, and data are generated to validate the refined approach. The following figures depict the performance enhancements achieved with the refined model, highlighting the increased accuracy in estimating both the angles and angular velocities of the double pendulum.

5.4.7. Training Performance

The training phase shows a significant improvement in loss reduction as shown in Figure 9, indicating better learning and model fitting to the double pendulum dynamics.

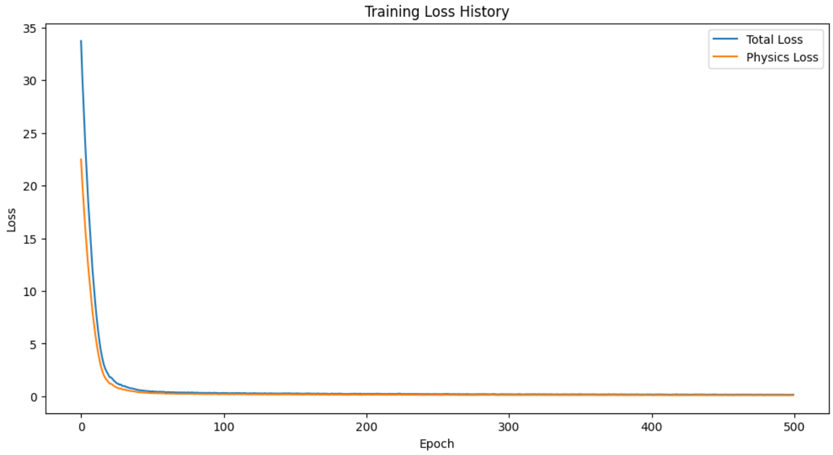

Figure 9. Training loss history with the enhanced physics-informed loss function.

5.4.8. State Estimation Results

Figures 10 and 11 compare the true states versus the estimated states over time, demonstrating the refined model's enhanced capability in accurately tracking both angles and angular velocities.

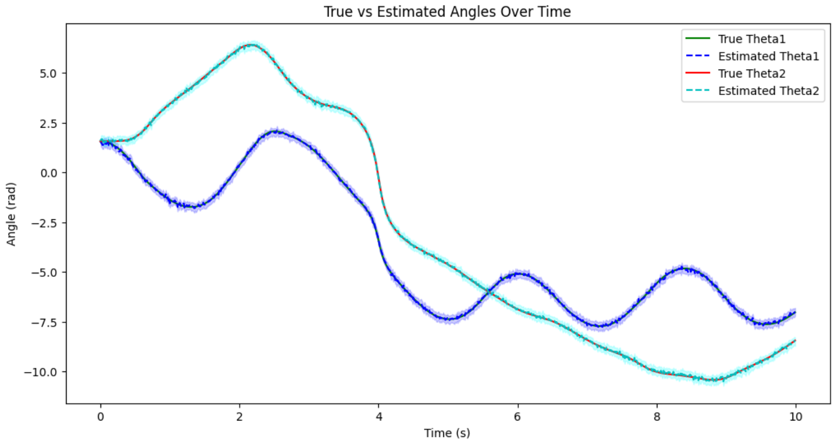

Figure 10. True vs. estimated angles (θ_1 and θ_2) with the enhanced model.

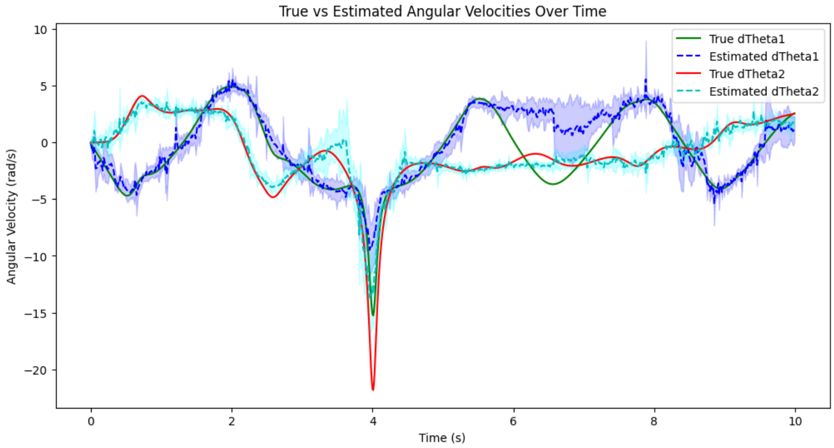

Figure 11. True vs. estimated angular velocities ($d\theta_1$ and $d\theta_2$) with the enhanced model.

These results validate the efficacy of the enhanced PINN and UKF approach in handling the complex dynamics of a double pendulum. The improvements in the model architecture and the adaptive features of the UKF contribute significantly to the accuracy and reliability of the state estimations.

6. Extension to Complex Systems: Quadcopter Drone

In this section, we extend the proposed hybrid framework to an even more complex example, a quadcopter drone simulation. Quadcopter dynamics are inherently nonlinear, making them ideal for our PINN and UKF architecture.

6.1. Problem Statement: Quadcopter Dynamics

The state vector for the quadcopter includes the position, orientation, and velocities:

$$[x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}] \tag{17}$$

Control inputs are given by:

$$[T, \tau_\phi, \tau_\theta, \tau_\psi] \tag{18}$$

where T is the thrust and $\tau_\phi, \tau_\theta, \tau_\psi$ are the torques.

6.2. Plan

1. Define the Quadcopter Dynamics: Use symbolic mathematics to derive the equations of motion.
2. Simulate Noisy Data: Generate training data with Gaussian noise.
3. Train a PINN Model: Predict the state vector at the next time step.
4. Implement UKF: Estimate the state using the trained PINN model.

6.3. PINN Model Architecture

The Physics-Informed Neural Network is designed to predict the next state vector given the current state. The architecture incorporates Monte Carlo Dropout for robust uncertainty quantification. Algorithm 5 presents the pseudocode.

Algorithm 5 Improved PINN Model for Quadcopter Simulation

```
1: Initialize dense layers with residual connections
2: Add batch normalization layers for stability
3: Apply Monte Carlo Dropout for uncertainty estimation
4: Define the output layer to predict the next state vector
```

6.4. UKF Implementation

The Unscented Kalman Filter (UKF) is adapted to handle the increased state dimension of the quadcopter system. The process and measurement noise covariance matrices are adjusted to reflect the system's complexity. Algorithm 6 presents the pseudocode.

Algorithm 6 UKF Procedure for Quadcopter Simulation with PINN Transition Model

```
1: Initialize sigma points based on current state estimate and covariance
2: Propagate sigma points through the PINN-based transition model
3: Compute predicted state estimate and covariance
4: Predict measurements for each sigma point
5: Calculate the cross-covariance between state and measurement
6: Update state estimate and covariance using the KALMAN gain
```

6.5. Results

6.5.1. State Estimation Performance

The hybrid PINN-UKF model estimates the states (position, orientation, linear, and angular velocities) and compares them against the true states. Shaded regions denote uncertainty bounds, aligning well with the true states. Training loss is depicted in Figure 12, while Figures 13–16 show true versus estimated states.

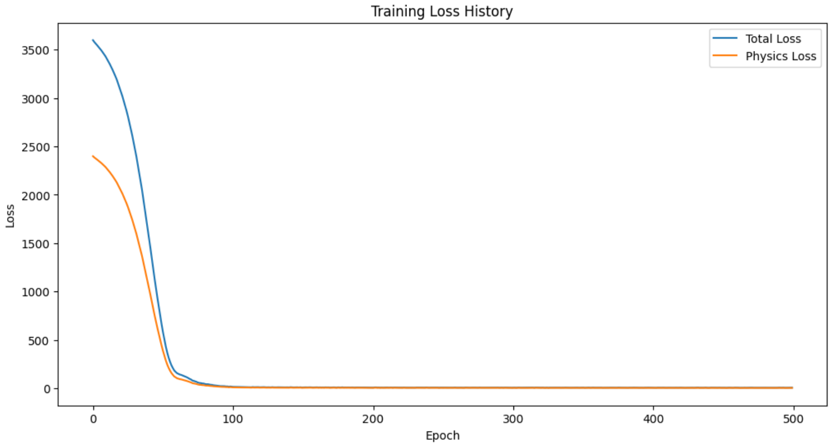

Figure 12. Training loss curves for the PINN model with Monte Carlo Dropout.

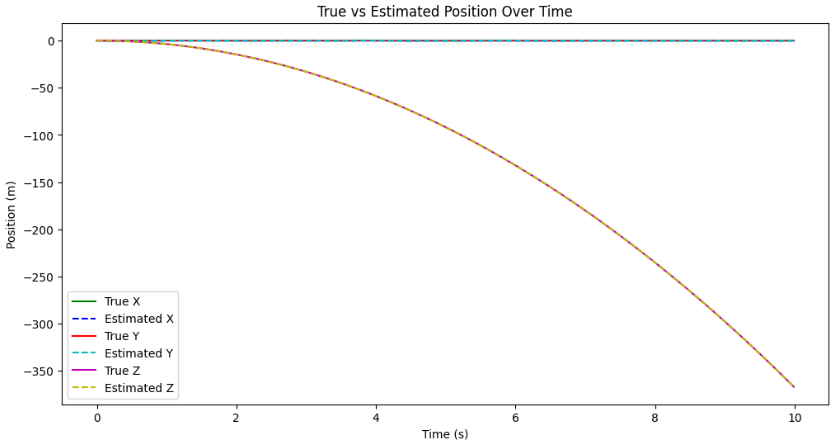

Figure 13. True vs. estimated position over time for the quadcopter simulation.

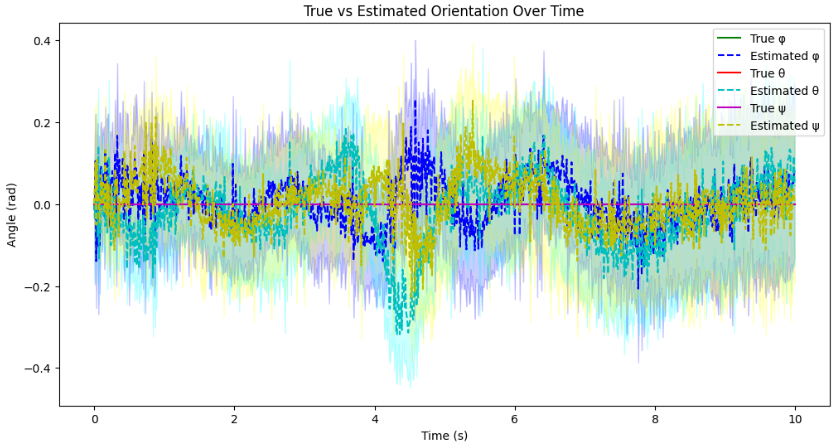

Figure 14. True vs. estimated orientation over time for the quadcopter simulation.

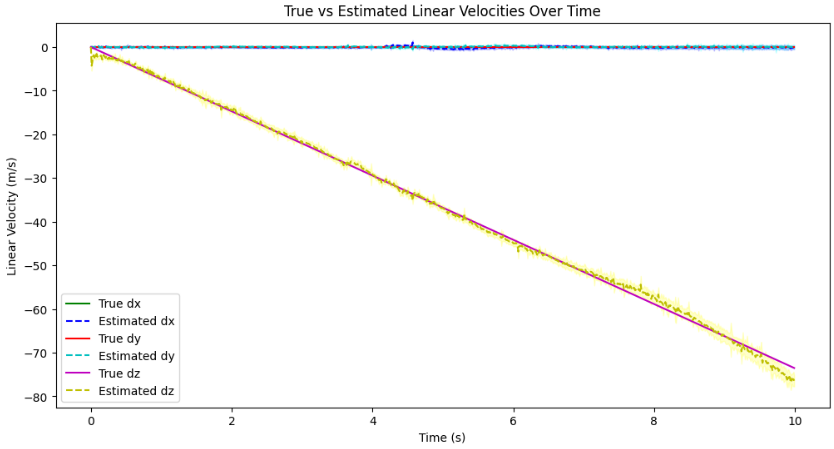

Figure 15. True vs. estimated linear velocities over time for the quadcopter simulation.

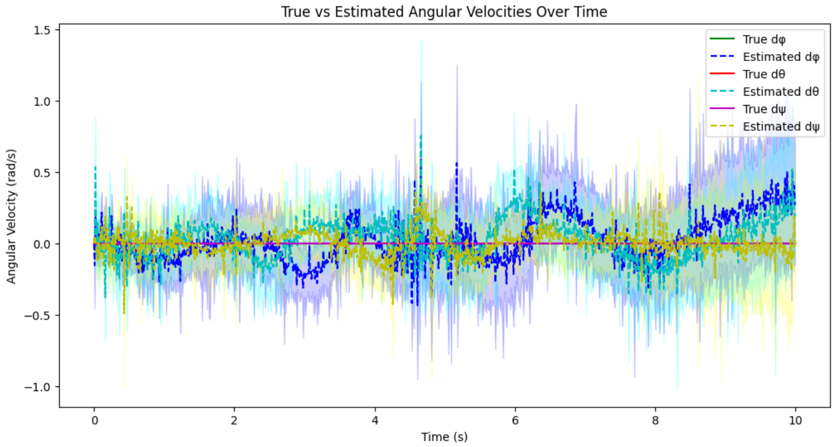

Figure 16. True vs. estimated angular velocities over time for the quadcopter simulation.

The quadcopter simulation illustrates the hybrid framework's adaptability to highly nonlinear and complex dynamic systems. By combining data-driven learning with classical filtering techniques, the hybrid PINN-UKF model provides state estimation and robust uncertainty quantification for demanding scenarios like drone navigation.

The hybrid state estimation framework integrating PINNs with an adaptive UKF, as demonstrated on a quadcopter drone, is inherently versatile and applicable to a wide range of dynamic systems beyond the specific use case presented. The core strengths of this methodology—namely its ability to leverage physics-based models for enhanced predictive accuracy and its rigorous uncertainty quantification through MC Dropout and adaptive noise covariance mechanisms—make it suitable for various fields including but not limited to autonomous vehicle navigation, robotics, aerospace engineering, and even financial modeling. In autonomous vehicles, for instance, the need for precise state estimation under uncertain conditions mirrors the requirements in quadcopter dynamics, thus benefiting from the same hybrid approach. Similarly, robotic manipulation tasks, which require accurate real-time state tracking amid dynamic interactions, can also gain substantial accuracy improvements from this method. Furthermore, the adaptability of the PINN-UKF framework to incorporate domain-specific physics models ensures that it can be tailored to different application scenarios, thereby extending its superiority and utility across multiple disciplines.

In this work, we employ mixed loss functions and MC Dropout to improve the accuracy and robustness of PINNs, with a particular focus on enhancing uncertainty estimation. It is essential to acknowledge that the uncertainty in state estimation arises from multiple sources, including measurement noise, process noise, model inaccuracies, and environmental variability. Measurement noise is introduced by the sensors and instruments used to collect data, which can vary in precision and reliability. Process noise, on the other hand, stems from the inherent unpredictability in the system's dynamics and external perturbations. Model inaccuracies occur due to the simplifications and assumptions made during the modeling process, which may not perfectly capture the real-world behavior of the system. Environmental variability includes factors such as changes in operating conditions, external disturbances, and interactions with other systems or agents.

To quantify and manage these uncertainties, MC Dropout is used to perform multiple stochastic forward passes through the neural network, generating a distribution of possible outcomes rather than a single deterministic prediction. This allows for the estimation of both the mean and variance of the predictions, providing a measure of the model's confidence. The mixed loss function, which combines data loss and physics-based loss, further ensures that the predictions adhere to known physical laws while fitting the observed data, thereby reducing model-induced uncertainty. By explicitly accounting for these different sources of uncertainty, our approach not only improves the reliability of state estimates under varying conditions but also enhances the overall robustness of the state estimation process.

6.5.2. In-Depth Analysis and Explanation of Experimental Results

In our experiments, the hybrid PINN-UKF framework demonstrated significant improvements in state estimation accuracy for both position and velocity tracking compared to the actual values. The superior performance can be attributed to several key factors.

- Enhanced Dynamics Modeling with PINNs:
    - Complexity Capture: The PINN component effectively captures the complex, nonlinear dynamics of the system, which are challenging for traditional models. This is particularly evident in scenarios involving highly dynamic and nonlinear behaviors, such as those seen in our quadcopter and double pendulum simulations.
    - Uncertainty Quantification: The integration of Monte Carlo Dropout within the PINN architecture allows for robust uncertainty estimation. This probabilistic approach provides a comprehensive understanding of the model's confidence in its predictions, enabling the UKF to adaptively adjust its noise covariances.
- Adaptive Filtering with UKF:
    - Dynamic Noise Covariance: The UKF's ability to dynamically adjust the process noise covariance based on the current state estimate improves its adaptability to changing system dynamics. This results in more accurate and stable state estimates over time.
    - Hybrid State Representation: By extending the state vector to include model parameters, the UKF enhances its capability to adapt to varying conditions, ensuring more reliable performance across different operational scenarios.

6.5.3. Universality in Actual Working Conditions

The hybrid PINN-UKF framework's design ensures its applicability across a wide range of real-world scenarios. Its ability to integrate data-driven insights with physical laws makes it highly versatile and robust. Key aspects that enhance its universality include:

- Scalability: The framework can be scaled to different system dimensions and complexities, making it suitable for various applications, from aerospace to robotics and autonomous vehicles.
- Adaptability: The dynamic adjustment of noise covariances and the inclusion of uncertainty quantification allow the framework to adapt to diverse operational conditions, ensuring reliable performance even in unpredictable environments.
- Real-Time Application: With advancements in computational power and efficient implementation strategies, the hybrid framework can be deployed in real-time applications, providing continuous and accurate state estimation for critical systems.

In conclusion, the hybrid PINN-UKF framework offers a comprehensive solution for state estimation, combining the strengths of physics-informed neural networks and adaptive filtering. Its detailed analysis and robust design ensure its applicability and effectiveness in a wide range of real-world conditions, addressing both the immediate and long-term challenges in dynamic system estimation.

6.6. Autonomy Levels and Application of the Hybrid State Estimation Model

The hybrid state estimation framework integrating PINNs with an adaptive UKF, as demonstrated on a quadcopter drone, showcases its versatility and potential applicability to a wide range of dynamic systems beyond the specific use case presented. The core strengths of this methodology—namely its ability to leverage physics-based models for enhanced predictive accuracy and its rigorous uncertainty quantification through MC Dropout and adaptive noise covariance mechanisms—make it suitable for various fields, including but not limited to autonomous vehicle navigation, robotics, aerospace engineering, and even financial modeling.

In the context of autonomous vehicles, precise state estimation is crucial for safe and efficient operation under varying and uncertain conditions. Autonomous vehicles are classified into six levels of driving automation as defined by the Society of Automotive Engineers (SAE) International (SAE, 2014) [20]:

- Level 0: No Driving Automation—The driver is in charge of all driving activities.
- Level 1: Driver Assistance—One autonomous system (e.g., Electronic Stability Control) is operational.
- Level 2: Partial Driving Automation—The vehicle can simultaneously handle two tasks (e.g., steering and braking), but the driver must remain engaged.
- Level 3: Conditional Driving Automation—The vehicle can manage most aspects of driving under certain conditions, but the driver must be ready to take control.
- Level 4: High Driving Automation—The vehicle operates autonomously in specific scenarios without driver intervention.
- Level 5: Full Driving Automation—The vehicle is fully autonomous in all conditions, with no need for a human driver.

The proposed hybrid state estimation framework is particularly relevant to Levels 3, 4, and 5, where the autonomy of the vehicle is high and the system must handle complex, dynamic environments with minimal or no human intervention. The adaptability and robustness of the PINN-UKF model are well-suited to these higher levels of autonomy, where the system must accurately predict and respond to a wide array of real-world scenarios, ensuring safety and reliability.

Application Across Autonomy Levels

Our model's capability to incorporate physics-informed learning and adaptively filter dynamic states makes it ideal for handling the complexities of high-level autonomous driving. For instance, in Level 3 and 4 scenarios, where the vehicle must navigate highways autonomously but still hand control back to the driver in more complex situations, the PINN-UKF framework can provide reliable state estimates and uncertainty quantifications, thereby enhancing decision-making processes. In Level 5 scenarios, where full autonomy is required, the model's enhanced predictive power and robustness can significantly contribute to the vehicle's ability to handle diverse and unpredictable environments autonomously.

In conclusion, while the hybrid state estimation model presents a higher computational load, its ability to deliver accurate and reliable state estimates under uncertain and dynamic conditions makes it a valuable asset for the development of high-level autonomous vehicles.

7. Discussion

The hybrid state estimation framework integrating Physics-Informed Neural Networks (PINNs) and Unscented Kalman Filtering (UKF) has proven highly effective in accurately tracking the states of both linear and nonlinear dynamic systems. Our results reveal several key insights into the strengths and limitations of this approach, offering a foundation for future research and practical applications.

7.1. Key Observations

7.1.1. PINN-UKF Synergy

By combining PINNs' ability to learn complex nonlinear dynamics with the rigorous uncertainty quantification of UKF, the framework leverages the complementary strengths of both techniques. The predictive capabilities of the PINN-based transition model enable accurate state estimation, while the UKF ensures reliable measurement updates with appropriately weighted noise covariances. This synergy is particularly noticeable in the nonlinear double pendulum system, where the hybrid approach effectively tracks chaotic behavior.

7.1.2. Uncertainty Quantification

Monte Carlo Dropout within the PINN model provides an intuitive way to assess predictive uncertainty. This allows the UKF to adjust the process noise dynamically and accurately capture the system's changing uncertainty. This feature is crucial for handling chaotic dynamics like those seen in the double pendulum, where small variations in initial conditions can significantly impact subsequent states.

7.1.3. Adaptive Covariance Estimation

The UKF's adaptive noise covariance estimation effectively mitigates the challenges posed by changing process dynamics and measurement noise. By incorporating these adjustments into the UKF's recursive estimation, the state predictions remain robust even when the system exhibits nonlinearity or noise variation.

7.1.4. Generalizing the Framework

The quadcopter drone simulation highlights the ability of the hybrid framework to adapt to complex, nonlinear dynamic systems. This versatility suggests several key directions for generalization.

7.1.5. Expanding Control Inputs

Incorporating more diverse control inputs will allow the framework to model systems with higher degrees of freedom. For example, incorporating motor speed data directly can capture rotor dynamics for drones or helicopters more comprehensively, leading to improved state estimation.

7.1.6. Diverse Applications

The framework's adaptability makes it suitable for a broad range of real-world applications beyond drones. Autonomous vehicle navigation, robotic manipulation, and multi-agent coordination are examples of where precise state estimation is crucial. The framework can be extended to handle varied sensing modalities, such as LiDAR, cameras, or acoustic sensors.

7.1.7. Multi-Sensor Fusion

Future research should investigate integrating data from multiple sensors. This would enable the hybrid framework to fuse information from complementary sensing technologies, enhancing robustness against noise or sensor failure. A sensor fusion approach can improve state estimation in complex environments, where each sensor type provides unique insights.

7.1.8. Real-Time Applications

Efforts to optimize the model's architecture and computational efficiency will be crucial for generalizing the framework to real-time applications. Techniques such as model compression, hardware acceleration, and parallel computing can significantly improve processing speed without compromising accuracy.

These expansions and refinements can help establish the hybrid PINN-UKF framework as a cornerstone for accurate state estimation in dynamic systems, offering a powerful tool for researchers and engineers working in control, robotics, and ML.

7.2. Limitations and Future Work

7.2.1. Computational Complexity

The computational complexity of the proposed framework is higher than traditional methods due to the increased network size and multiple stochastic forward passes during inference. For real-time applications, further optimization of the network architecture and parallelization of computations are essential.

7.2.2. Generalizability to Other Systems

While the hybrid framework has demonstrated efficacy in linear and nonlinear dynamic systems, future work should investigate its generalizability across a broader range of systems. Exploring more diverse physical models will provide insights into the framework's adaptability.

7.2.3. Model Regularization

Further investigation into regularization techniques, such as Bayesian neural networks and variational inference, can help refine the predictive models, especially for highly uncertain environments. More sophisticated uncertainty quantification methods may yield enhanced robustness.

This research demonstrates the efficacy and potential of integrating PINNs with UKF for state estimation in dynamic systems. The results pave the way for future studies aimed at expanding and refining this framework, providing researchers and engineers with a versatile tool to navigate the intricate dynamics of the physical world.

The hybrid PINN-UKF framework effectively estimates the state of the system involving the double pendulum and the quadcopter drone, demonstrating its adaptability to nonlinear dynamics and robustness in very sophisticated environments. This validates the potential of this framework for handling more complex nonlinear systems in future research.

While our enhanced PINN-UKF model has demonstrated effectiveness in the tested scenarios, it is essential to address the computational demands for practical deployment. Future research should focus on optimizing the computational efficiency of the framework through techniques such as model compression, parallel computing, and hardware acceleration in a cloud continuum scenario. Additionally, we recognize the need for validating the framework's adaptability to higher-order dynamic systems. Extending our approach to more complex systems will provide valuable insights into its scalability and generalizability. Furthermore, to enhance robustness in highly uncertain environments, incorporating advanced regularization techniques, such as Bayesian neural networks and variational inference, could significantly improve the model's performance.

8. Conclusions

In this work, we introduced a hybrid state estimation framework that integrates Physics-Informed Neural Networks (PINNs) with the Unscented Kalman Filter (UKF) to address the challenges of accurately estimating the states of dynamic systems. Our approach capitalizes on the complementary strengths of both techniques, leveraging PINNs' ability to learn complex dynamics and the UKF's rigorous uncertainty quantification.

Through experiments involving a linear dynamic system and the nonlinear double pendulum, we demonstrated the framework's robust state estimation capabilities. The hybrid model effectively tracks chaotic behavior, providing uncertainty bounds that align with ground truth states. Key insights from our results include the critical role of Monte Carlo Dropout for uncertainty estimation, adaptive noise covariance adjustment, and the synergy of combining physics-informed learning with KALMAN filtering. The simulation is further extended to handle the state estimation of a quadcopter drone, where the system is also able to track effectively highly nonlinear dynamics.

Despite its computational complexity, this approach has significant potential for broader applications across different dynamic systems. Future work should explore further optimizations, enhanced regularization techniques, and generalizability to other physical models.

In conclusion, this framework marks a promising step toward integrating ML models with established control theory techniques. Its ability to navigate the complexities of dynamic system estimation will serve as a versatile and valuable tool for researchers and engineers aiming to push the boundaries of state estimation accuracy and robustness.

Author Contributions: Conceptualization, J.d.C. and I.d.Z.; funding acquisition, J.d.C.; investigation, I.d.Z. and J.d.C.; methodology, I.d.Z. and J.d.C.; software, J.d.C. and I.d.Z.; supervision, J.d.C. and I.d.Z.; writing—original draft, J.d.C. and I.d.Z.; writing—review and editing, J.d.C. and I.d.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The work is developed under the following projects at BARCELONA Supercomputing Center: 'TIFON'. This work has also received funding from the 'NEXTBAT' project funded by the EUROPEAN Union's HORIZON EUROPE research and innovation programme under grant agreement No. 101103983. The work is also developed under UFV R&D pre-competitive project 'OpenMaas: Open Manufacturing as a Service'.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in FigShare at https://doi.org/10.6084/m9.figshare.25970206, accessed on 3 June 2024.

Conflicts of Interest: The authors declare that they have no conflicts of interest. The funders had no role in the design of the study, in the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

Physics-Informed Neural Network	PINN
Unscented Kalman Filter	UKF
Extended Kalman Filters	EKF
Neural Networks	NN
Artificial Intelligence	AI
Machine Learning	ML
Monte Carlo Dropout	MC Dropout

References

1. De Zarzà, I.; de Curtò, J.; Roig, G.; Calafate, C.T. LLM Adaptive PID Control for B5G Truck Platooning Systems. *Sensors* **2023**, *23*, 5899. [CrossRef] [PubMed]
2. De Curtò, J.; de Zarzà, I.; Calafate, C.T.Semantic Scene Understanding with Large Language Models on Unmanned Aerial Vehicles. *Drones* **2023**, *7*, 114. [CrossRef]
3. Khodarahmi, M.; Maihami, V. A review on Kalman filter models. *Arch. Comput. Methods Eng.* **2023**, *30*, 727–747. [CrossRef]
4. Freirich, D.; Michaeli, T.; Meir, R. Perceptual kalman filters: Online state estimation under a perfect perceptual-quality constraint. In Proceedings of the NeurIPS 2024, the Thirty-eighth Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 9–15 December 2024.
5. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
6. Antonelo, E.A.; Camponogara, E.; Seman, L.O.; Jordanou, J.P.; de Souza, E.R.; Hübner, J.F. Physics-informed neural nets for control of dynamical systems. *Neurocomputing* **2024**, *579*, 127419. [CrossRef]
7. Cai, S.; Mao, Z.; Wang, Z.; Yin, M.; Karniadakis, G.E. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mech. Sin.* **2021**, *37*, 1727–1738. [CrossRef]
8. Krishnapriyan, A.S.; Gholami, A.; Zhe, S.; Kirby, R.; Mahoney, M.W. Characterizing possible failure modes in physics-informed neural networks. *Adv. Neural Inf. Process. Syst.* **2021**, 34, 26548–26560.
9. Bihlo, A. Improving physics-informed neural networks with meta-learned optimization. *J. Mach. Learn. Res.* **2024**, *25*, 1–26.
10. Meng, Z.; Qian, Q.; Xu, M.; Yu, B.; Yıldız, A.R.; Mirjalili, S. PINN-form: A new physics-informed neural network for reliability analysis with partial differential equation. *Comput. Methods Appl. Mech. Eng.* **2023**, *414*, 116172. [CrossRef]
11. Zou, Z.; Meng, X.; Karniadakis, G.E. Correcting model misspecification in physics-informed neural networks (PINNs). *J. Comput. Phys.* **2024**, *505*, 112918. [CrossRef]
12. Hu, Z.; Shukla, K.; Karniadakis, G.E.; Kawaguchi, K. Tackling the curse of dimensionality with physics-informed neural networks. *Neural Netw.* **2024**, *176*, 106369. [CrossRef] [PubMed]
13. Bertipaglia, A.; Alirezaei, M.; Happee, R.; Shyrokau, B. An Unscented Kalman Filter-Informed Neural Network for Vehicle Sideslip Angle Estimation. *IEEE Trans. Veh. Technol.* **2024**, 1–15. [CrossRef]
14. Luo, Z.; Shi, D.; Shen, X.; Ji, J.; Gan, W.S. Gfanc-kalman: Generative fixed-filter active noise control with cnn-kalman filtering. *IEEE Signal Process. Lett.* **2023**, *31*, 276–280. [CrossRef]

15. Cassinis, L.P.; Park, T.H.; Stacey, N.; D'Amico, S.; Menicucci, A.; Gill, E.; Ahrns, I.; Sanchez-Gestido, M. Leveraging neural network uncertainty in adaptive unscented Kalman Filter for spacecraft pose estimation. *Adv. Space Res.* **2023**, *71*, 5061–5082. [CrossRef]
16. Tan, C.; Cai, Y.; Wang, H.; Sun, X.; Chen, L. Vehicle State Estimation Combining Physics-Informed Neural Network and Unscented Kalman Filtering on Manifolds. *Sensors* **2023**, *23*, 6665. [CrossRef]
17. Liu, Y.; Wang, L.; Ng, B.F. A hybrid model-data-driven framework for inverse load identification of interval structures based on physics-informed neural network and improved Kalman filter algorithm. *Appl. Energy* **2024**, *359*, 122740. [CrossRef]
18. Ni, X.; Revach, G.; Shlezinger, N. Adaptive Kalmannet: Data-Driven Kalman Filter with Fast Adaptation. In Proceedings of the ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024.
19. Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016.
20. Wiseman, Y. *Autonomous Vehicles. Encyclopedia of Information Science and Technology*, 5th ed.; Volume 1, Chapter 1, pp. 1–11. 2020. Available online: https://u.cs.biu.ac.il/~wisemay/Autonomous-Vehicles-Encyclopedia.pdf (accessed on: 1 May 2024).

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.