

Correspondence

Split-KalmanNet: A Robust Model-Based Deep Learning Approach for State Estimation

Geon Choi ¹, Graduate Student Member, IEEE,
 Jeonghun Park ², Member, IEEE, Nir Shlezinger ³, Member, IEEE,
 Yonina C. Eldar ⁴, Fellow, IEEE,
 and Namyoon Lee ⁵, Senior Member, IEEE

Abstract—Estimation of the state of a discrete-time state-space model from noisy measurements is a crucial aspect of signal processing. The extended Kalman filter (EKF) is widely used as a low-complexity solution based on a state evolution and measurement model of the state-space model. However, obtaining precise information about these models can be difficult in practice, and model mismatch greatly reduces state estimation accuracy. In this paper, we introduce Split-KalmanNet, a robust EKF algorithm that utilizes the power of deep learning for state estimation. Split-KalmanNet is inspired by the recent KalmanNet and calculates the Kalman gain using the Jacobian matrix of a measurement function and two recurrent neural networks (RNNs) with a split structure. The RNNs independently learn the covariance matrices of the prior state estimate and the innovation from data. The proposed split structure in calculating the Kalman gain allows for compensation of both state and measurement model mismatch effects. Numerical simulations show that Split-KalmanNet outperforms traditional EKF and the state-of-the-art KalmanNet algorithm in various scenarios of model mismatch.

Index Terms—Kalman filter, model-based deep learning, state-space model, sequential state estimation.

I. INTRODUCTION

The problem of estimating the state of a discrete-time state-space (SS) model from noisy measurements in real-time is a crucial task in signal processing and finds application in various domains, such as object tracking and localization [1], [2], [3], [4]. To address this problem, low-complexity recursive Bayesian filtering algorithms have been developed. The Kalman filter (KF) is simple optimal sequential estimator that achieve minimum mean-squared error (MMSE) performance in linear state models under additive white Gaussian noise. Several efforts have been made to extend the KF to more general SS models [5], [6], [7], [8], [9].

Manuscript received 10 October 2022; revised 5 February 2023; accepted 17 April 2023. Date of publication 8 May 2023; date of current version 19 September 2023. This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT (MSIT), Korea Government under Grant 2022R1A5A1027646 and in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT), under Grant RS-2023-00208552. The review of this article was coordinated by Prof. Rui Dinis. (*Corresponding author: Namyoon Lee.*)

Geon Choi is with the Department of Electrical Engineering, POSTECH, Pohang 37673, South Korea (e-mail: simon03062@postech.ac.kr).

Jeonghun Park is with the School of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, South Korea (e-mail: jhpark@yonsei.ac.kr).

Nir Shlezinger is with the School of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel (e-mail: nirshl@bgu.ac.il).

Yonina C. Eldar is with the Math and CS Faculty, Weizmann Institute of Science, Rehovot 7610001, Israel (e-mail: yonina.eldar@weizmann.ac.il).

Namyoon Lee is with the School of Electrical Engineering, Korea University, Seoul 02841, South Korea (e-mail: namyoon@korea.ac.kr).

Digital Object Identifier 10.1109/TVT.2023.3270353

The most widely used state estimation algorithm for non-linear SS models is the Extended Kalman Filter (EKF). This method leverages the model knowledge gained from the physical dynamics of the SS model, and alternately performs state updates according to a predefined state evolution model and measurement updates based on a measurement model that relates the current state to the noisy measurements. The EKF is a generalization of the KF for non-linear SS models, and remains mathematically tractable and optimal for linear and Gaussian processes [10], [11]. Despite its versatility, this model-based online estimation method is susceptible to physical and statistical model mismatch effects, which may result in divergence when these effects are pronounced [12].

Recently, KalmanNet, a model-based deep neural network (DNN)-assisted Kalman filter algorithm, was proposed in [13]. KalmanNet learns the Kalman gain matrix with model-less DNNs [14], [15] and integrates it into the KF algorithm's state and measurement updates. The trained Kalman gain matrix is robust to model mismatch, thanks to deep learning. However, KalmanNet underutilize the model-based structure when computing the Kalman gain, leading to performance loss in state estimation when state and measurement model mismatch coexist.

In this paper, we introduce a deep learning algorithm referred to as Split-KalmanNet for state estimation. Our algorithm employs a model-based approach, with the main concept being the independent training of the prior state estimate and innovation covariance matrices using two parallel DNNs. The Kalman gain matrix is then calculated by combining the learned covariance matrices with the knowledge of the measurement function's Jacobian matrix, according to the standard Kalman gain update rule. The split learning structure of Split-KalmanNet makes it robust to both state and measurement model mismatch effects, as it decouples these effects through the independent optimization of the two DNNs. Our simulations demonstrate that the proposed algorithm provides a considerable reduction in mean-squared error (MSE) compared to traditional model-based EKF and KalmanNet algorithms, even under state and measurement model mismatch conditions.

The paper is organized as follows. In Section II, we present a SS model and state estimation problem. Then, we briefly review conventional model-based EKF and KalmanNet algorithms for state estimation in Section III. Section IV introduces the proposed Split-KalmanNet algorithm. Section V shows the effectiveness of the proposed method via simulations.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we present a discrete-time non-linear SS model, consisting of both a state evolution model and a measurement model.

State evolution model: Let $\mathbf{x}_t \in \mathbb{R}^N$ be the hidden state vector at time t . The state evolution model is given by

$$\mathbf{x}_t = \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_t) + \mathbf{w}_t. \quad (1)$$

The underlying model of the system dynamics is described by a non-linear function $\mathbf{f} : \mathbb{R}^N \times \mathbb{R}^L \rightarrow \mathbb{R}^N$, where $\mathbf{u}_t \in \mathbb{R}^L$ represents the control input. The state \mathbf{x}_{t-1} transitions to \mathbf{x}_t with the addition of a Gaussian random vector \mathbf{w}_t , which has zero mean and an unknown covariance matrix \mathbf{Q}_t .

Measurement model: For each time t , we measure the noise observation $\mathbf{y}_t \in \mathbb{R}^M$ corrupted by additive noise. The measurement model is given by

$$\mathbf{y}_t = \mathbf{h}(\mathbf{x}_t) + \mathbf{v}_t, \quad (2)$$

where $\mathbf{h} : \mathbb{R}^N \rightarrow \mathbb{R}^M$ is a non-linear measurement function and \mathbf{v}_t is zero-mean Gaussian random vector with unknown covariance \mathbf{R}_t .

State estimation problem: The state estimation problem aims to recover the hidden state \mathbf{x}_t using the noisy measurements up to time t in terms of MMSE. Mathematically, we need to solve the following optimization problem in each time t :

$$\arg \min_{\hat{\mathbf{x}}_t} \mathbb{E} \left[(\mathbf{x}_t - \hat{\mathbf{x}}_t)^2 \mid \mathbf{y}_1, \dots, \mathbf{y}_t \right]. \quad (3)$$

The EKF algorithm offers a recursive solution to this problem. Its successful implementation, however, relies on accurate knowledge of both process and measurement noise distributions. Under the assumption of zero-mean Gaussian noise, it is essential to have a precise understanding of the covariance matrices \mathbf{Q}_t and \mathbf{R}_t . Unfortunately, obtaining such information perfectly can be challenging in practical scenarios, leading to a significant performance degradation in the EKF algorithm due to model mismatch. Hence, a robust sequential state estimation algorithm that can handle model mismatch is imperative.

Remark 1: The SS model considered in this paper is a generic model, and it can be applicable to diverse applications, including multi-object tracking and simultaneous localization and mapping (SLAM) applications. In these applications, each object and landmark in an environment is modeled as its own SS model and associating sensor measurements with specific SS model (i.e., data association) is a challenging problem. A framework incorporating the random finite set and probability hypothesis density filter can resolve the data association problem by fully integrating data association uncertainty into state estimation problem [16], [17]. However, these approaches require prior knowledge of the model parameters, such as noise covariance matrices, and still faces the challenge of model mismatch.

III. CONVENTIONAL APPROACHES

In this section, we briefly review the EKF and KalmanNet algorithm in [13] to solve the state estimation problem in (3).

A. Model-Based Extended Kalman Filter

The EKF algorithm performs i) state-update and ii) measurement update recursively. To present the algorithm concisely, we first define some notations. Let *a priori* mean and covariance of the state vector at time t be $\hat{\mathbf{x}}_{t|t-1}$ and $\Sigma_{t|t-1} = \mathbb{E}[(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1})(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1})^\top]$. We also denote *a posteriori* mean and its covariance at time t by $\hat{\mathbf{x}}_{t|t}$ and $\Sigma_{t|t} = \mathbb{E}[(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})^\top]$.

State-update: The state-update step predicts *a priori* mean $\hat{\mathbf{x}}_{t|t-1}$ and covariance $\Sigma_{t|t-1}$ of the state vector using *a posteriori* mean $\hat{\mathbf{x}}_{t-1|t-1}$ and covariance matrix $\Sigma_{t-1|t-1}$, which are obtained in the previous measurement-update step:

$$\begin{aligned} \hat{\mathbf{x}}_{t|t-1} &= \mathbf{f}(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_t), \\ \Sigma_{t|t-1} &= \mathbf{F}_{t-1} \Sigma_{t-1|t-1} \mathbf{F}_{t-1}^\top + \mathbf{Q}_t, \end{aligned} \quad (4)$$

where $\mathbf{F}_t = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \big|_{\mathbf{x}=\hat{\mathbf{x}}_{t|t}}$ is the Jacobian of $\mathbf{f}(\cdot)$ evaluated at $\hat{\mathbf{x}}_{t|t}$.

Measurement-update: In the measurement-update step, *a posteriori* mean $\hat{\mathbf{x}}_{t|t}$ and covariance matrix $\Sigma_{t|t}$ are computed using *a priori* mean $\hat{\mathbf{x}}_{t|t-1}$ and covariance matrix $\Sigma_{t|t-1}$ attained in the previous state-update step. In detail, measurement prediction $\hat{\mathbf{y}}_{t|t-1} = \mathbf{h}(\hat{\mathbf{x}}_{t|t-1})$ is computed.

Then, the Kalman gain matrix refines *a priori* mean $\hat{\mathbf{x}}_{t|t-1}$ using the innovation $\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}$ acquired in time t toward minimizing the MSE. The Kalman gain matrix is given by

$$\mathbf{K}_t = \Sigma_{t|t-1} \mathbf{H}_t^\top \mathbf{S}_t^{-1}, \quad (5)$$

where $\mathbf{H}_t = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \big|_{\mathbf{x}=\hat{\mathbf{x}}_{t|t-1}}$ is the Jacobian of $\mathbf{h}(\cdot)$ evaluated at $\hat{\mathbf{x}}_{t|t-1}$ and $\mathbf{S}_t = \mathbb{E}[(\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1})(\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1})^\top]$ is the covariance matrix of the innovation. Using the Kalman gain in (5), *a posteriori* mean and covariance are updated as

$$\begin{aligned} \hat{\mathbf{x}}_{t|t} &= \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t (\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}), \\ \Sigma_{t|t} &= \Sigma_{t|t-1} - \mathbf{K}_t \mathbf{S}_t \mathbf{K}_t^\top. \end{aligned} \quad (6)$$

The model-based EKF algorithm experiences significant degradation in the presence of model mismatch. To address this issue, the conventional solutions of worst case error minimization and system identification methods such as covariance matrix adaptation and expectation-maximization algorithm has been used [18], [19], [20]. However, these methods often lead to reduced performance as they struggle to adapt to multiple SS models. An alternative solution is to adopt a data-driven approach, which leverages available data to overcome the model mismatch while minimizing performance loss. KalmanNet uses a model-based data-driven approach to overcome the degradation caused by model mismatch.

B. KalmanNet - A Hybrid Approach

The KalmanNet [13] is a hybrid approach that exploits the power of a DNN while maintaining the recursive EKF algorithm structures in (4), (5), and (6). The noticeable difference from the model-based EKF algorithm is in computing the Kalman gain matrix. Instead of using the Kalman gain matrix in (5) for the measurement-update, KalmanNet performs the measurement-update using a learned Kalman gain matrix, denoted by $\mathcal{K}_t(\Theta)$, with trainable parameters $\Theta \in \mathbb{R}^\Xi$, i.e.,

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathcal{K}_t(\Theta) (\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}). \quad (7)$$

The Kalman gain is trained by optimizing the squared-error loss function with respect to the parameters Θ in an end-to-end fashion.

Let $\mathcal{D} = \{(\mathbf{X}^{(\ell)}, \mathbf{Y}^{(\ell)})\}_{\ell=1}^L$ be a dataset comprised of L different trajectories of the state-measurement pairs. The ℓ th trajectory contains T_ℓ data samples $(\mathbf{X}^{(\ell)}, \mathbf{Y}^{(\ell)})$ with $\mathbf{X}^{(\ell)} = [\mathbf{x}_1^{(\ell)}, \mathbf{x}_2^{(\ell)}, \dots, \mathbf{x}_{T_\ell}^{(\ell)}]$ and $\mathbf{Y}^{(\ell)} = [\mathbf{y}_1^{(\ell)}, \mathbf{y}_2^{(\ell)}, \dots, \mathbf{y}_{T_\ell}^{(\ell)}]$. We also define the empirical loss function for the i th training trajectory as

$$\mathcal{L}(\Theta) = \frac{1}{L} \sum_{\ell=1}^L \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \left\| \mathbf{x}_t^{(\ell)} - \hat{\mathbf{x}}_{t|t}(\mathbf{y}_t^{(\ell)}; \mathcal{K}_t(\Theta)) \right\|_2^2. \quad (8)$$

Let $\Delta \mathbf{x}_t^{(\ell)} = \mathbf{x}_t^{(\ell)} - \hat{\mathbf{x}}_{t|t-1}^{(\ell)}$ and $\Delta \mathbf{y}_t^{(\ell)} = \mathbf{y}_t^{(\ell)} - \hat{\mathbf{y}}_{t|t-1}^{(\ell)}$ be the state prediction error and the measurement innovation at time t for trajectory l . The partial derivative of the loss function with respect to the Kalman gain matrix is

$$\begin{aligned} \frac{\partial \mathcal{L}(\Theta)}{\partial \mathcal{K}_t(\Theta)} &= \frac{1}{LT_\ell} \sum_{\ell=1}^L \sum_{t=1}^{T_\ell} \frac{\partial \left\| \Delta \mathbf{x}_t^{(\ell)} - \mathcal{K}_t(\Theta) \Delta \mathbf{y}_t^{(\ell)} \right\|_2^2}{\partial \mathcal{K}_t(\Theta)} \\ &= \frac{2}{LT_\ell} \sum_{\ell=1}^L \sum_{t=1}^{T_\ell} \left(\mathcal{K}_t(\Theta) \Delta \mathbf{y}_t^{(\ell)} - \Delta \mathbf{x}_t^{(\ell)} \right) \left(\Delta \mathbf{y}_t^{(\ell)} \right)^\top. \end{aligned} \quad (9)$$

By plugging (9) into the following chain rule:

$$\frac{\partial \mathcal{L}(\Theta)}{\partial \Theta} = \frac{\partial \mathcal{L}(\Theta)}{\partial \mathcal{K}_t(\Theta)} \frac{\partial \mathcal{K}_t(\Theta)}{\partial \Theta}, \quad (10)$$

one can find a local optimal parameter Θ^* such that $\frac{\partial \mathcal{L}(\Theta^*)}{\partial \Theta^*} = 0$ using a stochastic gradient descent algorithm. To track the covariance matrices implicitly, recurrent neural networks (RNNs) were used to train the parameter using the innovation difference $\Delta \mathbf{y}_t = \mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}$ and the state update difference $\Delta \hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t} - \hat{\mathbf{x}}_{t|t-1}$ as input features. We refer to [13] for the detailed structure of the KalmanNet.

This DNN-aided EKF algorithm does not require knowledge of the covariance matrices for process and measurement noise, *i.e.*, \mathbf{Q}_t and \mathbf{R}_t . Rather, these matrices are implicitly learned from data for the Kalman gain computation; thereby, it can diminish statistical model mismatch effects compared to the model-based EKF method. In addition, it simplifies the state and time update procedures because *a priori* and *a posteriori* covariance matrices $\Sigma_{t|t-1}$ and $\Sigma_{t|t}$ can be discarded. Notwithstanding, this joint learning approach of the Kalman gain can lead to performance loss when one of \mathbf{Q}_t and \mathbf{R}_t mismatch effects dominates the other. For instance, the dominant mismatch error in measurement \mathbf{R}_t hinders learning the covariance matrix of states \mathbf{Q}_t . This fact motivates us to consider a split learning structure when computing the Kalman gain matrix.

IV. SPLIT-KALMANNET

In this section, we put forth a DNN-aided EKF algorithm, called Split-KalmanNet. To illustrate the proposed idea, it is instructive to rethink the Kalman gain matrix in the model-based EKF algorithm. As in (5), the Kalman gain matrix is computed by multiplying three factors: i) *a priori* covariance matrix of the state $\Sigma_{t|t-1}$, ii) the Jacobian matrix of the measurement function \mathbf{H}_t , iii) the inverse of the covariance matrix of the innovation \mathbf{S}_t^{-1} . The first term, $\Sigma_{t|t-1}$, is a function of both \mathbf{Q}_t and \mathbf{F}_t . In addition, the last term, \mathbf{S}_t^{-1} , changes according to the second-order moment of the measurement noise \mathbf{R}_t . When the uncertainties in \mathbf{Q}_t and \mathbf{R}_t are heterogeneous, the end-to-end KalmanNet can be less robust because $\mathcal{K}_t(\Theta)$ cannot separate the statistical model mismatch effects in \mathbf{Q}_t and \mathbf{R}_t .

Harnessing the statistical independence property between the process and measurement noise vectors and the model-based knowledge in computing the Kalman gain matrix, our approach is to train the Kalman gain matrix with two separate DNNs to form the Kalman gain matrix:

$$\mathcal{G}_t(\Theta_1, \Theta_2, \mathbf{H}_t) = \mathcal{G}_t^1(\Theta_1) \mathbf{H}_t^\top \mathcal{G}_t^2(\Theta_2), \quad (11)$$

where $\mathcal{G}_t^1(\Theta_1)$ is the DNN that learns *a priori* covariance matrix of the state $\Sigma_{t|t-1}$ implicitly with parameter $\Theta_1 \in \mathbb{R}^{\mathbb{E}_1}$. In addition, $\mathcal{G}_t^2(\Theta_2)$ is another DNN that learns the inverse of the covariance matrix of the innovation \mathbf{S}_t implicitly with parameter $\Theta_2 \in \mathbb{R}^{\mathbb{E}_2}$. Using this trained Kalman gain matrix, the Split-KalmanNet performs the state-update as

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathcal{G}_t(\Theta_1, \Theta_2, \mathbf{H}_t) (\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}). \quad (12)$$

The overall structure of Split-KalmanNet is illustrated in Fig. 1.

The Split-KalmanNet allows independently learning the model uncertainties caused by the state update ($\mathbf{F}_t, \mathbf{Q}_t$) and the measurement update ($\mathbf{H}_t, \mathbf{R}_t$). Furthermore, the Split-KalmanNet decouples the effect of aforementioned model uncertainties from the Jacobian matrix of the measurement function \mathbf{H}_t in computing (11). As a result, it is more robust than the end-to-end KalmanNet in the presence of heterogeneous model mismatch effects thanks to the split learning structure.

Loss function: In analogy to (8), given dataset $\mathcal{D} = \{(\mathbf{X}^{(\ell)}, \mathbf{Y}^{(\ell)})\}_{\ell=1}^L$, we define a loss function in terms of two parameter sets Θ_1 and Θ_2 :

$$\mathcal{L}_2(\Theta_1, \Theta_2) = \frac{1}{L} \sum_{\ell=1}^L \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \left\| \mathbf{x}_t^{(\ell)} - \hat{\mathbf{x}}_{t|t} \left(\mathbf{y}_t^{(\ell)}; \mathcal{G}_t(\Theta_1, \Theta_2, \mathbf{H}_t) \right) \right\|_2^2. \quad (13)$$

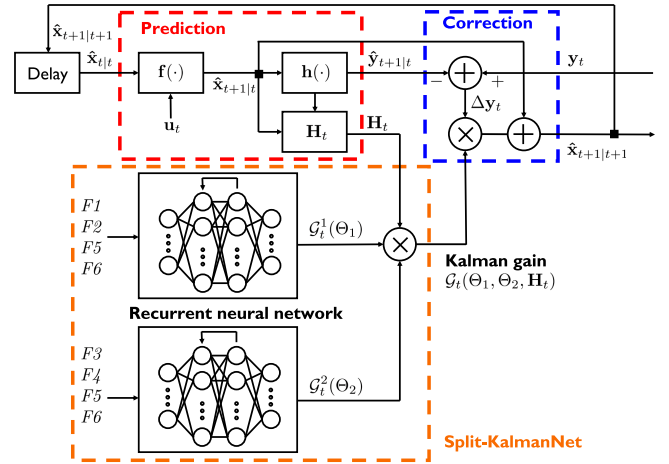


Fig. 1. Block diagram of the proposed Split-KalmanNet algorithm.

Alternating optimization for training: We also provide an alternating optimization method to train two DNNs. We start by computing the partial derivative of the loss function with respect to the Kalman gain matrix:

$$\begin{aligned} & \frac{\partial \mathcal{L}_2(\Theta_1, \Theta_2)}{\partial \mathcal{G}_t(\Theta_1, \Theta_2, \mathbf{H}_t)} \\ &= \frac{1}{LT_\ell} \sum_{\ell=1}^L \sum_{t=1}^{T_\ell} \frac{\partial \left\| \Delta \mathbf{x}_t^{(\ell)} - \mathcal{G}_t(\Theta_1, \Theta_2, \mathbf{H}_t) \Delta \mathbf{y}_t^{(\ell)} \right\|_2^2}{\partial \mathcal{G}_t(\Theta_1, \Theta_2, \mathbf{H}_t)} \\ &= \frac{2}{LT_\ell} \sum_{\ell=1}^L \sum_{t=1}^{T_\ell} \left(\mathcal{G}_t(\Theta_1, \Theta_2, \mathbf{H}_t) \Delta \mathbf{y}_t^{(\ell)} - \Delta \mathbf{x}_t^{(\ell)} \right) \left(\Delta \mathbf{y}_t^{(\ell)} \right)^\top. \end{aligned} \quad (14)$$

Then, for given $\bar{\Theta}_2$, we first optimize Θ_1 by solving the first-order optimality condition:

$$\frac{\partial \mathcal{L}_2(\Theta_1, \bar{\Theta}_2)}{\partial \Theta_1} = \frac{\partial \mathcal{L}_2(\Theta_1, \bar{\Theta}_2)}{\partial \mathcal{G}_t(\Theta_1, \bar{\Theta}_2, \mathbf{H}_t)} \frac{\partial \mathcal{G}_t^1(\Theta_1)}{\partial \Theta_1} \mathbf{H}_t^\top \mathcal{G}_t^2(\bar{\Theta}_2) = 0. \quad (15)$$

Using the updated $\bar{\Theta}_1$, we optimize Θ_2 such that

$$\frac{\partial \mathcal{L}_2(\bar{\Theta}_1, \Theta_2)}{\partial \Theta_2} = \frac{\partial \mathcal{L}_2(\bar{\Theta}_1, \Theta_2)}{\partial \mathcal{G}_t(\bar{\Theta}_1, \Theta_2, \mathbf{H}_t)} \mathcal{G}_t^1(\bar{\Theta}_1) \mathbf{H}_t^\top \frac{\partial \mathcal{G}_t^2(\Theta_2)}{\partial \Theta_2} = 0. \quad (16)$$

Using the updated $\bar{\Theta}_2$, we again optimize Θ_1 by solving (15). We repeat (15) and (16) until the parameters converge.

Complexity: In contrast to KalmanNet, our algorithm needs to compute \mathbf{H}_t at every time step. Nevertheless it should be noted that in numerous scenarios, closed form expression \mathbf{H}_t can be readily obtained and its evaluation results in only a minimal increase in complexity.

Input features: The DNN $\mathcal{G}_t^1(\Theta_1)$ learns *a priori* covariance matrix of the state $\Sigma_{t|t-1}$ and the DNN $\mathcal{G}_t^2(\Theta_2)$ learns the inverse of the covariance matrix of the innovation \mathbf{S}_t implicitly. To track these covariance matrices, we consider the following input features:

- F1 The state update difference $\Delta \hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t} - \hat{\mathbf{x}}_{t|t-1}$. The available feature at time t is $\Delta \hat{\mathbf{x}}_{t-1}$.
- F2 The state evolution difference $\Delta \hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t} - \hat{\mathbf{x}}_{t-1|t-1}$. The available feature at time t is $\Delta \hat{\mathbf{x}}_{t-1}$.
- F3 The innovation difference $\Delta \mathbf{y}_t = \mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}$.

F4 The measurement difference $\Delta \tilde{\mathbf{y}}_t = \mathbf{y}_t - \mathbf{y}_{t-1}$.

F5 The linearization error $\mathbf{h}(\hat{\mathbf{x}}_{t|t-1}) - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1}$.

F6 The Jacobian matrix of measurement function \mathbf{H}_t .

Features F1 and F3 characterize the uncertainty of state estimates, while features F2 and F4 characterize the information about the state and measurement evolution process. In addition, features F5 and F6 encapsulate the local behavior of the measurement function.

V. SIMULATION RESULTS

We use the uniform circular motion and Michigan NCLT dataset [21] to demonstrate the effect of the proposed split architecture in computing the Kalman gain and applicability to real world dynamics.¹

State estimation algorithm: We compare the following algorithms:

- EKF (perfect): The model-based EKF for reference purpose. The exact model parameters are used.
- EKF (mismatch): The EKF using predetermined model parameters.
- KalmanNet: The KalmanNet using input features {F1, F2, F3, F4} as presented in [13].
- Split-KalmanNet: The proposed Split-KalmanNet. The input features {F1, F2, F3, F4, F5, F6} are used.

We omit the comparison with other state estimation algorithms such as unscented Kalman filter and particle filter, because the KalmanNet outperforms them according to [13].

Test metric: We use the MSE, which is defined as

$$\text{MSE [dB]} = 10 \log_{10} \left(\frac{1}{LT_\ell} \sum_{\ell=1}^L \sum_{t=1}^{T_\ell} \|\mathbf{x}_t - \hat{\mathbf{x}}_{t|\ell}\|^2 \right). \quad (17)$$

A. Uniform Circular Motion

To illustrate the impact of split architecture on noise heterogeneity and non-linearity, we adopt the following two-dimensional state evolution model:

$$\mathbf{x}_{t+1} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \mathbf{x}_t + \mathbf{w}_t, \quad (18)$$

where $\mathbf{x}_t = [x_t \ y_t]^\top \in \mathbb{R}^2$ is the position vector in the Euclidean plane and θ is constant rotation speed.

As a measurement model, we consider both linear and non-linear measurement

$$\mathbf{y}_t = \begin{cases} \mathbf{x}_t + \mathbf{v}_t, & \text{linear,} \\ \left[\|\mathbf{x}_t\|_2^2 \ \text{atan2}(\mathbf{x}_t) \right]^\top + \mathbf{v}_t, & \text{non-linear,} \end{cases} \quad (19)$$

where the angle between the x -axis and the state vector \mathbf{x}_t in the Euclidean plane is represented by $\text{atan2}(\mathbf{x}_t)$ in radians. The noise covariance matrices, \mathbf{Q}_t and \mathbf{R}_t , are set to $\sigma_w^2 \mathbf{I}$ and $\sigma_v^2 \mathbf{I}$ respectively. To model the variability of the noise, we set a fixed value of $\sigma_w^2 = 10^{-3}$ and vary $\nu = \sigma_v^2 / \sigma_w^2$. In the case of mismatched EKF, we use $\sigma_w^2 = 10^{-3}$ and $\sigma_v^2 = 1$, resulting in a ν value of 30 dB.

Convergence: To demonstrate the convergence of our Split-KalmanNet, we present the validation error during each training round for both linear and non-linear measurement models in Fig. 2. We set significant noise heterogeneity (i.e., $\nu = 40$ dB). As evident from the figures, Split-KalmanNet converges at a faster pace compared to KalmanNet and almost reaches the MSE of a perfect EKF.

¹The source code used in this simulation with the complete set of hyperparameters can be found online at <https://github.com/geonchoi/Split-KalmanNet>.

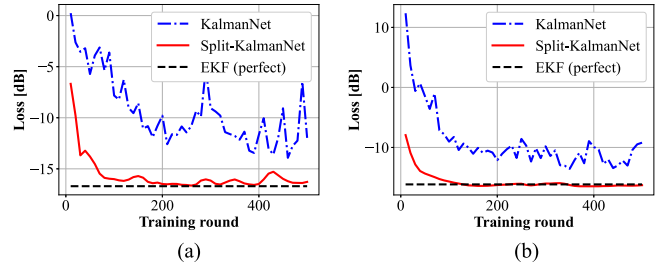


Fig. 2. Convergence rates of different algorithms as increasing training rounds. Noise statistics are $\sigma_w^2 = -30$ dB and $\nu = 40$ dB. The length of trajectory is $T_\ell = 15$ for training, and $T_\ell = 50$ for validation. (a) and (b) correspond to linear and non-linear measurement models.

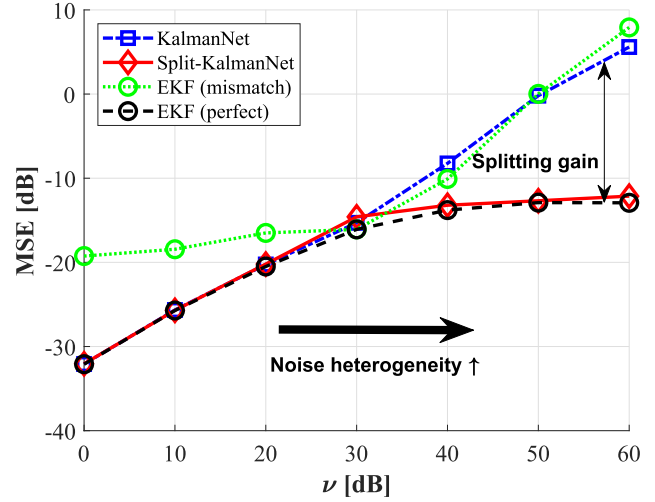


Fig. 3. MSE (y-axis) comparison according to noise parameters ($\sigma_w^2 = -30$ dB, $\sigma_v^2 = \nu \sigma_w^2$) for linear measurement model. The length of trajectory is $T_\ell = 15$ for training, and $T_\ell = 100$ for testing.

Noise heterogeneity and non-linearity: We investigate the MSE performance of Split-KalmanNet and KalmanNet for different levels of noise heterogeneity ν . As illustrated in Fig. 3, when the noise heterogeneity is low, both algorithms achieve MMSE, equivalent to the perfect EKF. However, as the noise heterogeneity increases, KalmanNet fails to learn the Kalman gain matrix and its MSE exceeds that of perfect EKF. On the other hand, Split-KalmanNet continues to achieve MMSE, demonstrating its robustness to noise heterogeneity.

In the non-linear measurement model, as seen in Fig. 4, the perfect EKF does not achieve the optimal MSE and the data-driven methods surpass it for low levels of noise heterogeneity. Specifically, the Split-KalmanNet outperforms KalmanNet with slightly lower MSE, as it exploits the nonlinearity to a greater extent in optimizing the Kalman gain. Even as the noise heterogeneity increases, the Split-KalmanNet remains effective in learning the Kalman gain, while KalmanNet struggles. This highlights the superior robustness of Split-KalmanNet to noise heterogeneity compared to KalmanNet.

Time-varying noise: To simulate time-varying noise statistics, we set $\mathbf{R}_t = \sigma_{v,t}^2 \mathbf{I}$ and $\sigma_{v,t}^2$ to be

$$\sigma_{v,t}^2 [\text{dB}] = \begin{cases} \lfloor \frac{t}{2} \rfloor + t_0 \bmod 50, & 0 \leq t \leq 14, \text{ train,} \\ \lfloor \frac{t}{10} \rfloor + 30 \bmod 50, & 0 \leq t \leq 999, \text{ test,} \end{cases} \quad (20)$$

where $t_0 \in \{0, 10, 20, 30\}$ is randomly selected for each training sequence.

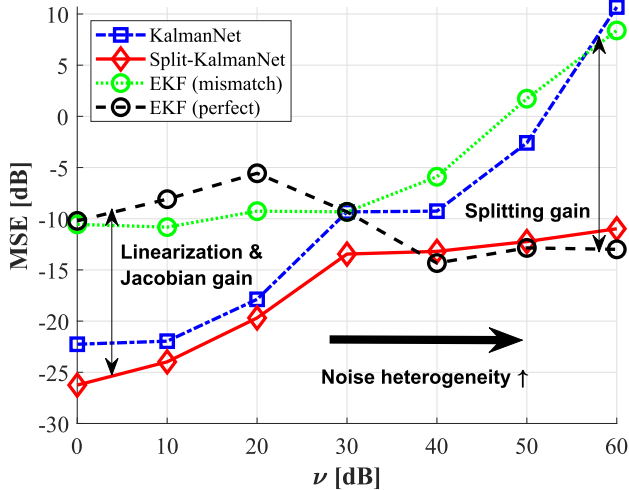


Fig. 4. MSE (y-axis) comparison according to noise parameters ($\sigma_w^2 = -30$ dB, $\sigma_v^2 = \nu\sigma_w^2$) for non-linear measurement model. The length of trajectory is $T_\ell = 15$ for training, and $T_\ell = 100$ for testing.

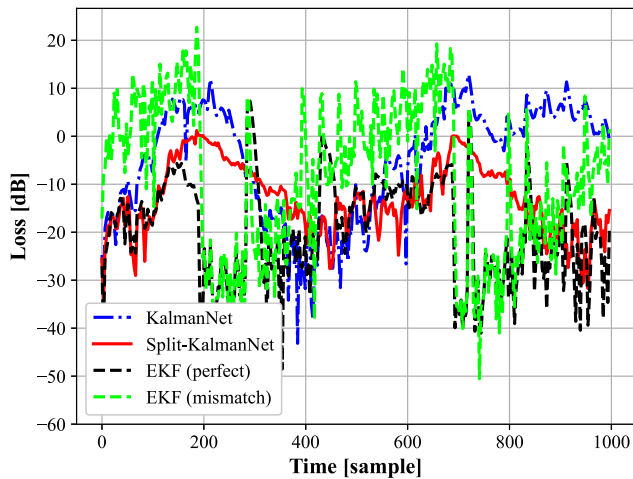


Fig. 5. Instantaneous loss (MSE) for a particular trajectory with time-varying statistics. The non-linear measurement model is used.

In Fig. 5, we present the plot of the instantaneous loss ($\|\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}\|^2$) for a specific test sequence. For the mismatched EKF, we arbitrarily set $\sigma_{v,t}^2$ to 0 [dB]. The results indicate that Split-KalmanNet has the ability to adapt to slow variations in noise statistics mismatch. However, the degree of adaptation may vary depending on the nature of the training dataset. Alternatively, we can adopt online semi-supervised learning approach as in [22]. After applying the supervised training technique, Split-KalmanNet can optimize the parameters in an unsupervised manner to adapt to variations in the SS model.

B. Real World Dynamics: Michigan NCLT Dataset

The effectiveness of the Split-KalmanNet is evaluated using the Michigan NCLT dataset [21]. This dataset consists of moving trajectories of a Segway robot, with the ground truth locations and odometry readings being impacted by noise. Our aim is to localize the position of the robot by utilizing the noisy velocity readings obtained through odometry. To achieve this, we adopt the nearly constant acceleration motion model for the Segway robot, as described in [23, Ch. 19].

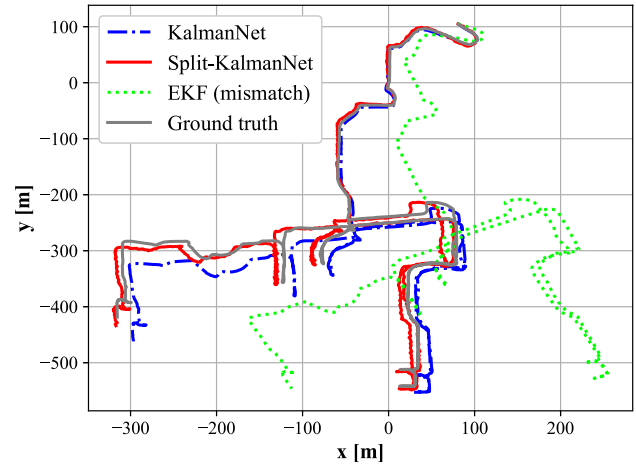


Fig. 6. NCLT dataset trajectory. Training and validation data with a sampling rate of 1 Hz and a date of 2012-01-22. Test data with a sampling rate of 1 Hz and a date of 2012-04-29. The length and number of trajectory is $T_\ell = 50$, $L = 80$ for training, $T_\ell = 200$, $L = 5$ for validation, and $T_\ell = 2000$, $L = 1$ for testing.

We employed the trajectory with a sampling rate of 1 Hz and a date of 2012-01-22 for the training and validation of KalmanNet and Split-KalmanNet. For testing purposes, we used a trajectory with a sampling rate of 1 Hz and a date of 2012-04-29. The effectiveness of Split-KalmanNet was compared to other algorithms in Fig. 6 which highlights its superiority. The trajectory obtained through the EKF was rotated due to the accumulation of velocity error. However, Split-KalmanNet overcomes this limitation by learning the Kalman gain matrix from the data and providing more accurate positioning results. This demonstrates the potential of Split-KalmanNet for real-world applications and its ability to provide improved state estimates.

VI. CONCLUSION

In this paper, we introduced a reliable sequential state estimation algorithm called Split-KalmanNet, which is designed to handle SS models with model mismatch. The Split-KalmanNet uses a unique approach of training the covariance matrices of the prior state estimate and innovation through two parallel DNNs. By utilizing the structure of the Kalman gain, the algorithm composes the Kalman gain matrix using the trained covariance matrices and the measurement function's Jacobian matrix. Simulation results demonstrate that the proposed Split-KalmanNet significantly improves the MSE compared to traditional methods in various model mismatch scenarios.

REFERENCES

- [1] S. I. Roumeliotis and G. A. Bekey, "Bayesian estimation and kalman filtering: A unified framework for mobile robot localization," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, vol. 3, pp. 2985–2992.
- [2] M. Bertozzi, A. Broggi, A. Fascioli, A. Tibaldi, R. Chapuis, and F. Chausse, "Pedestrian localization and tracking system with kalman filtering," in *Proc. IEEE Intell. Veh. Symp.*, 2004, pp. 584–589.
- [3] S. Chen, "Kalman filter for robot vision: A survey," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4409–4420, Nov. 2012.
- [4] I. Ullah, Y. Shen, X. Su, C. Esposito, and C. Choi, "A localization based on unscented Kalman filter and particle filter localization algorithms," *IEEE Access*, vol. 8, pp. 2233–2246, 2020.
- [5] R. E. Kalman, "A new approach to linear filtering and prediction problems," *ASME J. Basic Eng.*, vol. 82, pp. 34–45, Mar. 1960.
- [6] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.

- [7] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F (Radar Signal Process.)*, vol. 140, no. 2, pp. 107–113, Apr. 1993.
- [8] O. Cappé, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential monte carlo," *Proc. IEEE*, vol. 95, no. 5, pp. 899–924, May 2007.
- [9] I. Arasaratnam and S. Haykin, "Cubature kalman filters," *IEEE Trans. Autom. Control*, vol. 54, no. 6, pp. 1254–1269, Jun. 2009.
- [10] G. A. Terejanu et al., "Extended Kalman filter tutorial," University at Buffalo, 2008. [Online]. Available: <https://homes.cs.washington.edu/todorov/courses/cseP590/readings/tutorialEKF.pdf>
- [11] B. Hajek, *Random Processes for Engineers*. New York, NY, USA: Cambridge Univ. Press, 2015.
- [12] G. A. Einicke and L. B. White, "Robust extended Kalman filtering," *IEEE Trans. Signal Process.*, vol. 47, no. 9, pp. 2596–2599, Sep. 1999.
- [13] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. V. Sloun, and Y. C. Eldar, "KalmaNet: Neural network aided Kalman filtering for partially known dynamics," *IEEE Trans. Signal Process.*, vol. 70, pp. 1532–1547, 2022.
- [14] N. Shlezinger, Y. C. Eldar, and S. P. Boyd, "Model-based deep learning: On the intersection of deep learning and optimization," *IEEE Access*, vol. 10, pp. 115384–115398, 2022.
- [15] A. L. Escoriza, G. Revach, N. Shlezinger, and R. J. V. Sloun, "Data-driven Kalman-based velocity estimation for autonomous racing," in *Proc. IEEE Int. Conf. Auton. Syst.*, 2021, pp. 1–5.
- [16] B.-N. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4091–4104, Nov. 2006.
- [17] J. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo, "A random-finite-set approach to Bayesian SLAM," *IEEE Trans. Robot.*, vol. 27, no. 2, pp. 268–282, Apr. 2011.
- [18] L. Xie, Y. C. Soh, and C. E. De Souza, "Robust Kalman filtering for uncertain discrete-time systems," *IEEE Trans. Autom. Control*, vol. 39, no. 6, pp. 1310–1314, Jun. 1994.
- [19] M. Zorzi, "Robust Kalman filtering under model perturbations," *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 2902–2907, Jun. 2017.
- [20] S. Akhlaghi, N. Zhou, and Z. Huang, "Adaptive adjustment of noise covariance in Kalman filter for dynamic state estimation," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2017, pp. 1–5.
- [21] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of michigan north campus long-term vision and lidar dataset," *Int. J. Robot. Res.*, vol. 35, no. 9, pp. 1023–1035, Aug. 2016.
- [22] G. Revach, N. Shlezinger, T. Locher, X. Ni, R. J. V. Sloun, and Y. C. Eldar, "Unsupervised learned Kalman filtering," in *Proc. 30th Eur. Signal Process. Conf.*, 2022, pp. 1571–1575.
- [23] M. A. Richards, J. Scheer, and W. A. Holm, *Principles of Modern Radar: Basic Principles*. New York, NY, USA: Scitech, 2010.