

Review

The New Trend of State Estimation: From Model-Driven to Hybrid-Driven Methods

Xue-Bo Jin ^{1,2} , Ruben Jonhson Robert Jeremiah ³, Ting-Li Su ^{1,2,*}, Yu-Ting Bai ^{1,2}  and Jian-Lei Kong ^{1,2} 

¹ Artificial Intelligence College, Beijing Technology and Business University, Beijing 100048, China; jinxuebo@btbu.edu.cn (X.-B.J.); baiyuting@btbu.edu.cn (Y.-T.B.); kongjianlei@btbu.edu.cn (J.-L.K.)

² China Light Industry Key Laboratory of Industrial Internet and Big Data, Beijing Technology and Business University, Beijing 100048, China

³ School of Food and Health, Beijing Technology and Business University, Beijing 100048, China; rubenjohnson1988@gmail.com

* Correspondence: sutingli@btbu.edu.cn

Abstract: State estimation is widely used in various automated systems, including IoT systems, unmanned systems, robots, etc. In traditional state estimation, measurement data are instantaneous and processed in real time. With modern systems' development, sensors can obtain more and more signals and store them. Therefore, how to use these measurement big data to improve the performance of state estimation has become a hot research issue in this field. This paper reviews the development of state estimation and future development trends. First, we review the model-based state estimation methods, including the Kalman filter, such as the extended Kalman filter (EKF), unscented Kalman filter (UKF), cubature Kalman filter (CKF), etc. Particle filters and Gaussian mixture filters that can handle mixed Gaussian noise are discussed, too. These methods have high requirements for models, while it is not easy to obtain accurate system models in practice. The emergence of robust filters, the interacting multiple model (IMM), and adaptive filters are also mentioned here. Secondly, the current research status of data-driven state estimation methods is introduced based on network learning. Finally, the main research results for hybrid filters obtained in recent years are summarized and discussed, which combine model-based methods and data-driven methods. This paper is based on state estimation research results and provides a more detailed overview of model-driven, data-driven, and hybrid-driven approaches. The main algorithm of each method is provided so that beginners can have a clearer understanding. Additionally, it discusses the future development trends for researchers in state estimation.

Keywords: state estimation; model-driven; data-driven; hybrid-driven; Kalman filter; deep learning



Citation: Jin, X.-B.; Robert Jeremiah, R.J.; Su, T.-L.; Bai, Y.-T.; Kong, J.-L. The New Trend of State Estimation: From Model-Driven to Hybrid-Driven Methods. *Sensors* **2021**, *21*, 2085. <https://doi.org/10.3390/s21062085>

Academic Editor: Geoff Merrett

Received: 14 January 2021

Accepted: 10 March 2021

Published: 16 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Some of the states in a system cannot be directly measured, or else, some measurements are not accurate enough due to sensor uncertainty. A sensor is a detection device that converts an actual physical quantity (the so-called state) into an electrical signal output and provides measurement data. The conversion process introduces some errors. These errors include drift error and measurement noise. The sensor's correction can eliminate the drift error, but the measurement noise cannot be eliminated fundamentally. Thus, the output of the sensor cannot be completely consistent with the state to be measured.

State estimation is one of the most common methods for estimating the most likely value of a state based on measurements and a system. It is a prerequisite for high-level information extraction and control and has been widely used in mobile robots [1], unmanned aerial vehicles (UAVs) [2], sensor networks [3], smart grids [4], health performance detection and evaluation [5,6], etc.

State estimation methods have a long history. In 1809, Gauss proposed an optimization method called the least-squares method to determine a celestial body's orbit from

measurements [7]. Since the least-squares method does not need to know the signal's prior statistical knowledge when estimating, the least-squares method has a wide range of applications in many fields.

In the 1940s, to control firepower, Wiener et al. [8] designed the Wiener filter in the frequency domain to achieve linear optimal dynamic estimation in a stationary random process system. Through the calculation of the Wiener–Hopf equation, the Wiener filter obtains the analytical solution of the optimal transfer function of the filter, which can suppress or gate signals containing a variety of information. However, because the Wiener filter requires that both the estimated state and the measurement conform to a stationary random process, and the Wiener–Hopf equation needs to be solved in the filtering process, the amount of calculation and the storage space required are extensive. The project is challenging to realize; therefore, it limits the applications of Wiener filters.

To overcome the shortcomings of Wiener filters, in 1960, Rudolf Emil Kalman proposed modern filtering theory [9]. He introduced state space in stochastic estimation theory, used state models to describe the relationship between states and measurements, and estimated states based on measurements using predictions and updates. Kalman filtering does not need to store all historical data. According to the state estimation at the previous moment and the current measurement information, a new estimate can be calculated according to the recursive method, which reduces the computer's storage and calculation capacity requirements and improves real-time processing. Simultaneously, the Kalman filter can estimate one-dimensional, stationary random processes and multidimensional, nonstationary random processes [10].

The Kalman filter has been widely used because of its simplicity and ease of implementation. Because a computer will continue to accumulate and transmit rounding errors and truncation errors during the calculation process, the error covariance matrix loses its positive definiteness and results in unstable filter estimation. Therefore, researchers have successively proposed a series of numerical, robust filtering algorithms, such as square root filtering, UD (Upper-Diagonal) decomposition filtering, and singular value decomposition filtering [11]. These methods effectively improve the Kalman filter's numerical stability while also increasing computational efficiency [12].

On the other hand, the standard Kalman filter requires an accurate model, the known statistical characteristics of the system noise. These requirements are relatively harsh in applications. In actual systems, the system cannot meet these conditions due to some uncertainties, making the Kalman filter lose its optimality, which reduces the estimation accuracy and even leads to divergence. Researchers have introduced the idea of robust control in filtering to solve this problem, thus forming a robust estimation [13].

Moreover, the standard Kalman filtering theory is only applicable to linear systems and requires that the observation equations are linear. Meanwhile, in actual engineering practice, the system is generally nonlinear. Therefore, in the 1970s, Bucy and Sunahara proposed the extended Kalman filter (EKF) [14–16]. The nonlinear system is linearized first and then uses the generalized Kalman filter to estimate the state. However, because the linearization process will introduce errors in the nonlinear system due to calculating the Jacobian matrix, the final state estimation accuracy will decrease. When the Jacobian matrix calculation is inaccurate, the problem of filtering divergence will also occur.

The sigma point Kalman filter (SPKF) method is a class of approximate nonlinear filtering methods based on the Gaussian distribution, including the unscented Kalman filter (UKF), central difference Kalman filter (CDKF), square-root unscented Kalman filter (SRUKF), etc. [17]. The ideas of these methods are roughly the same, of which the UKF is the most famous. It uses several sigma points for nonlinear systems to obtain the accuracy with second order.

With continuous research, the original methods have improved in the Gaussian domain filtering system. Some new methods have been proposed for approximating nonlinear characteristics. In Bayesian filtering's general framework, the filtering process involves predictions and updates, including two integral operations. Under the Gaussian assump-

tion, the integral is a multidimensional integral operation of the product of the equivalent Gaussian function and the system function. A new idea is provided to realize approximate Gaussian filtering: the nonlinear filtering is realized through an integral calculation. For example, Simon Haykin et al. proposed a new filtering strategy independent of the EKF and UKF, named the cubature Kalman filter (CKF) [18].

The above methods all describe the noise of a system using the Gaussian distribution. Studies have shown that the effectiveness of these methods is only valid for single-mode problems. They assume that the system state and noise are both Gaussian distributions, where one Gaussian component corresponds to one mode. However, many issues faced in the real world are often much more complicated than single-mode problems. A typical example is system noise, including process noise and observation noise, which is not a Gaussian distribution, but a gamma distribution or other complex mixed distributions [19]. Therefore, with the filtering of the above mentioned algorithms based on the Gaussian noise assumption, it is difficult to obtain a satisfactory estimation performance due to the model mismatch. In general, complex noise can be expressed as the sum of multiple Gaussian noises. Then, the EKF, UKF, and CKF are applied to obtain the Gaussian mixture (GM)-EKF, UKF, and CKF [20].

It is necessary to provide the system with the specific expression of the model and the distribution characteristics of noise for the standard Kalman filter, EKF, UKF, CKF, etc., as well as the multimode, complex, approximate Gaussian mixture filtering methods, such as GM-EKF, etc. However, when the system is complex, it is challenging to obtain these system models. Therefore, based on random sampling, another type of filter has appeared, which realizes the state's estimation by calculating the conditional probability of the system state and realizes the conditional probability transfer through Bayes' theorem. Its algorithm is not based on the state function's distribution characteristics, observation function, and noise (such as the mean and variance, etc.), but the state's conditional probability density. Its purpose is to provide the state's probability distribution, not the functional expression of the state. This algorithm is based on Monte Carlo simulation technology and is usually called the sequential Monte Carlo method or particle filter method [21].

In this type of method, a large number of particles are generated by Monte Carlo simulation, and their distribution is used to approximate the probability distribution of the state. Its advantage lies in its strong applicability, and it can be applied to any complex system through calculating particles. Its accuracy can approach optimal estimation. It can overcome the shortcomings of traditional Gaussian filtering algorithms such as the EKF, which are more sensitive to initial value selection. By contrast, particle filters easily capture the actual state within a specific error range due to particles' dispersion, thereby improving the filter system's stability and convergence speed. However, the Monte Carlo simulation method is challenging to realize recursive filtering with, and the calculation amount is much more tremendous than that with the Gaussian approximation method, which affects its real-time application to some extent.

With the development of sensor technology and storage technology, especially in recent years, modern intelligent systems, such as unmanned aerial vehicles (UAVs), autonomous driving systems, etc., have been widely used. We found that there are more and more sensors in the system, and the measurement is also increasing. In the military and civil fields, many processes, such as target monitoring, detection, tracking, and identification, are based on multiple sensors' measurements and are completed by information fusion technology [22]. Therefore, the data-driven learning network has also become an important research direction in state estimation theory.

Compared with model-based methods that require known system information, data-based methods can use big data to obtain system characteristics that cannot be described by the model. However, they abandon the established knowledge of the system and rely too much on data. When the data are insufficient or of low quality, the result is not better than that from traditional model-based estimation methods. Therefore, these two methods have their advantages and disadvantages. It is crucial to effectively combine model-driven

and data-driven methods to improve the performance of state estimation effectively, i.e., use a hybrid-driven state estimation method.

Estimation methods such as the Kalman filter series, EKF, UKF, CKF, and particle filters have obtained rich published research results. They are widely used in various detection, tracking, and control systems. However, data-driven methods based on shallow networks cannot fundamentally improve the estimation performance, so related research has not been the mainstream part of research into estimation methods. Recently, the rapid development of other artificial intelligence methods was highlighted in recent years [23,24], prompting researchers to reconsider improving estimation performance by the hybrid modeling method. However, related research in this area has just begun, so there are not many related review papers.

Although there are not many, the authors are still willing to recommend the following reviews that are worth reading: Hong et al. [25] provides a survey for estimators that can produce accurate estimations for complex dynamic systems, such as airbag, debris detection, and active blast protection systems. From an application perspective, Dehghanpour et al. [26] focuses on distribution system state estimation in power systems. A few critical topics are discussed, such as mathematical problem formulation, the application of pseudomeasurements, metering instrument placement, network topology issues, the impacts of renewable penetration, and cybersecurity. It is worth emphasizing that both conventional and modern data-driven methods are reviewed in this survey. Similarly, Jin et al. [27,28] are also based on the applications. Simultaneously, the latter emphasizes the research progress in artificial intelligence in moving objects, which is more general and includes deep learning networks in unmanned autonomous mobile systems.

Unlike the above review papers, this review introduces state estimation methods without emphasizing a specific system or application background. We try to provide the development path of the state estimation method from the perspective of the method and discuss the future development trend under the premise of the current system's universal characteristics.

This review is based on the problem of state estimation, starting with the introduction of classic estimation methods. We analyze the problems of classic estimation methods, the current research progress and problems of model-driven and data-driven methods, and discuss the state based on the combination of data-driven and model-driven methods. Finally, we put forward our ideas for future state estimation research.

The structure of this article is as follows. Sections 2 and 3 are both for model-driven estimation methods. Section 2 describes the Kalman filter and Gaussian mixture filter, which are based on given system models. Section 3 describes the state estimation method when the system model is not accurate enough, including a robust filter and closed-loop adaptive filter. Section 4 discusses data-driven methods, including deep neural networks, the hyperparameter optimization of network models, and the ability to suppress sensor noise. Section 5 provides a detailed introduction to current hybrid drive estimation methods. Finally, in the Conclusion, we discuss the future development research direction.

2. State Estimation Based on a Distinct Model

The state estimation method discussed in this section is based on the unified framework of Bayesian filtering. These methods require a known model of the system: the expression of the state (the so-called process model) and the relationship between the measurements and state (the so-called measurement model). Furthermore, the distribution of the process noise and observation noise of the system must be known. When the system's noise is Gaussian noise, the Kalman filter series can obtain satisfactory estimation results (mentioned in Section 2.1). However, if the noise becomes complex and cannot be approximated by the sum of Gaussian noise, the mixed filter is needed (mentioned in Section 2.2).

2.1. Kalman Filter Family

The standard Kalman filter requires a model of the given system, and the state of the system is usually described as the following process model:

$$x_{k+1} = A_k x_k + w_k \quad (1)$$

where x_k is the state, A_k is the state transition matrix, w_k is the state noise with a given covariance of Q_k , and the measurement model is the following:

$$z_k = C_k x_k + v_k \quad (2)$$

where C_k is the measurement matrix, z_k is the measurement data, and v_k is the measurement noise with a given covariance of R_k . Kalman filtering provides the estimated state $\hat{x}(k|k)$ by the state's initialization $\hat{x}(0|0)$ and covariance $P(0|0) = P_0$:

$$\begin{aligned} \hat{x}_{k|k-1} &= A_{k-1} \hat{x}_{k-1|k-1} \\ P_{k|k-1} &= A_{k-1} P_{k-1|k-1} A_{k-1}^T + Q_{k-1} \end{aligned} \quad (3)$$

and

$$\begin{aligned} K_k &= P_{k|k-1} C_k^T [C_k P_{k|k-1} C_k^T + R_k]^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k [z_k - C_k \hat{x}_{k|k-1}] \\ P_{k|k} &= [I - K_k C_k] P_{k|k-1} \end{aligned} \quad (4)$$

Equations (3) and (4) show that the Kalman filter algorithm has a two-step process. Equation (3) denotes the prediction step, in which the prediction estimates $\hat{x}(k|k-1)$ are obtained based on the last estimate $\hat{x}_{k-1|k-1}$. The second step is the update step shown in Equation (4), in which the estimates $\hat{x}(k|k)$ are updated using a filter gain K_k based on $\hat{x}(k|k-1)$.

Since most practical systems have nonlinear characteristics, researchers have improved the nonlinear system based on the standard Kalman filter. The nonlinear filtering algorithm, the EKF, uses the Taylor expansion of a nonlinear function to retain the first-order linearization. The study [29] used the EKF to estimate the real-time traffic state in freeway stretches for the nonlinear macroscopic traffic flow model. Next, we provide the system model and then illustrate the EKF estimation method.

The nonlinear relation of the system is described by the following general model:

$$\begin{aligned} x_{k+1} &= f(x_k, w_k) \\ z_k &= h(x_k, v_k) \end{aligned} \quad (5)$$

where $f(\cdot)$ and $h(\cdot)$ are the nonlinear process and measurement equations, respectively. We cannot find the process matrix A_k and measurement matrix C_k in Equations (3) and (4), which are replaced by $f(\cdot)$ and $h(\cdot)$. Therefore, the Kalman filter in Equations (3) and (4) cannot be used to estimate the nonlinear systems in Equation (5).

Then, the EKF obtains the prediction $\hat{x}_{k|k-1}$ from the nonlinear transfer $f(\hat{x}_{k-1|k-1})$. In addition, the innovation $z_k - C_k \hat{x}_{k|k-1}$ can be changed to $z_k - h(\hat{x}_{k|k-1})$. The update of the state then becomes

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - h(\hat{x}_{k|k-1})) \quad (6)$$

We note that the other equations in the standard Kalman filter, such as Equation (4), must use the matrices A_k and C_k . The EKF method obtains these matrices as the following first-order Taylor series expansions: The first-order Taylor expansion of the nonlinear process equation $f(x_k, w_k)$ as $\frac{\partial f}{\partial x} |_{\hat{x}_{k-1|k-1}} = F_k$ and $\frac{\partial f}{\partial w} |_{\hat{x}_{k-1|k-1}} = L_k$ and the Taylor expansion of the nonlinear measurement equation $h(x_k, w_k)$ as $\frac{\partial h}{\partial x} |_{x=\hat{x}_{k|k-1}} = H_k$ and $\frac{\partial h}{\partial v} |_{x=\hat{x}_{k|k-1}} = M_k$,

where F_k , L_k , H_k , and M_k are the transformations from the Taylor expansion based on the nonlinear process equation $f(x_k, w_k)$ and the measurement equation $h(x_k, w_k)$. Then, similarly to the standard Kalman filter in Equation (4), the estimation covariance and the filter gain are obtained by F_k , L_k , H_k , and M_k for the nonlinear system models in Equation (5).

Because the Taylor expansion only has the first order, an EKF is commonly applied to a nonlinear system's weak case. If the system's nonlinearity is strong, the error between the approximate linear system of the Taylor expansion and the original nonlinear system leads to a decrease in estimation performance and even to the phenomenon of filter divergence. To maintain the estimation performance, Yang et al. [30] provided the improved federated EKF for multisensor-integrated navigation according to the near-ground short distance navigation applications of small unmanned aerial vehicles (UAVs). The study [31] employed the EKF to fuse the data from multiple heterogeneous sensors, including an inertial measurement unit (IMU), a magnetometer, a barometer, a GNSS receiver, an optical flow sensor (OFS), light detection and ranging (LiDAR), and an RGB-D camera for an unmanned aerial vehicle.

Unlike the EKF, the UKF does not use linearization. Instead, several effective adoption points are used to simulate the state transfer relationship of the nonlinear model. By effectively selecting sampling points, the mean and covariance of the state can be better captured.

In 2002, Julier proposed the UKF method and selected sampling points as the unscented transformation [32]. Later, researchers proposed other unscented transformations such as spherical unscented transformation. The difference from the standard Kalman filter is that the UKF is not just a calculation of the mean and covariance of the state estimation but also the calculation of each sigma point generated by the unscented transformation. Therefore, in addition to the standard prediction and update process, the filter includes generating sigma points and calculating the mean and variance of the current step based on sigma points [33]:

Assuming that the state estimate and its covariance for the previous step have been obtained, namely, $\hat{x}_{k-1|k-1}$ and $P_{k-1|k-1}$, the process of unscented transformation is calculated as follows:

$$\hat{x}_{k-1|k-1}^{(i)} = \hat{x}_{k-1|k-1} + \tilde{x}^{(i)} \quad (7)$$

where

$$\begin{aligned} \tilde{x}^{(i)} &= 0, \quad i = 0 \\ \tilde{x}^{(i)} &= \left(\sqrt{(n+k)P_{k-1|k-1}} \right)_i^T, \quad i = 1, \dots, n \\ \tilde{x}^{(i)} &= -\left(\sqrt{(n+k)P_{k-1|k-1}} \right)_i^T, \quad i = n+1, \dots, 2n \end{aligned} \quad (8)$$

Then, each sigma point $\hat{x}_{k-1|k-1}^{(i)}$ is predicted and is weighted as

$$\hat{x}_{k|k-1} = \sum_{i=0}^{2n} w^{(i)} \hat{x}_{k|k-1}^{(i)} \quad (9)$$

with $w^{(0)} = \frac{a}{n+a}$, $w^{(i)} = \frac{1}{2(n+a)}$, $i = 1, 2, \dots, 2n$, and a being a positive constant. Similarly, the forward prediction of the measurement and the estimated weight also requires calculating sigma points and the weighted summation. The details of the calculation process can be found in Jin et al. [28]. In our former research [34], we proposed two nonlinear estimation methods based on the EKF and UKF, respectively, for real-time indoor RFID (Radio-frequency identification) tracking. The results show that compared to the EKF, the UKF method can obtain a lower covariance, while the EKF can cost less in calculation. The study [35] developed an adaptive unscented Kalman filter (AUKF) as a vehicle driving

state observer based on the seven-degree-of-freedom vehicle model and the Dugoff tire model.

The CKF also uses several points to expand the previous step's estimation and variance, which is just a special case of the UKF with $2n$ sigma points. Differently from the sigma point of the UKF in Equation (8), $\tilde{x}^{(i)}$ is obtained by

$$\begin{aligned}\tilde{x}^{(i)} &= (\sqrt{P_{k-1|k-1}})_i^T, \quad i = 1, \dots, n \\ \tilde{x}^{(i)} &= -(\sqrt{P_{k|k-1}})_i^T, \quad i = n + 1, \dots, 2n\end{aligned}\quad (10)$$

with $w^{(i)} = \frac{1}{2n}$.

We can find that the CKF and UKF are very similar in the form of the algorithm. The CKF is simply a special case of the UKF with $2n$ sigma points (with no center sigma point). To obtain that, though a different path is followed, in the end, we have a similar algorithm. Therefore, the CUK and UKF are also relatively similar in estimation performance. As an application for the CKF, [36] proposed a strong tracking mixed-degree cubature Kalman filter (STMCKF) method according to the system characteristics of the quadruped robot to obtain the fusion estimation of forward kinematics and the IMU track.

Finally, we would like to discuss the performance of each one of these Kalman filters. The classic Kalman filter requires the system to be linear with Gaussian white noise. It is obviously difficult to meet these requirements in an actual system. Therefore, in practical applications, the classic Kalman filter's performance cannot reach the optimal estimation performance. However, the classic Kalman filter plays a pioneering role in research, so it is the theoretical basis of other filters. If the system is nonlinear and contains Gaussian white noise, the performance of the EKF, UKF, and CKF is different. Most research results indicate that the UKF and CKF are better than the EKF [34]. Meanwhile, the UKF and CKF exhibit almost the same performance. Recently, the Kalman filter has been developed in practical systems, such as the distributed Kalman filter [37] for sensor networks, hybrid Kalman filter [38], and adaptive ensemble square root Kalman filter [39].

2.2. Gaussian Mixture Filter and Random Sampling Filter

The Kalman filter given in Section 2.1 is based on single Gaussian noise. However, in existing systems, the distribution characteristics of noise are often more complicated. Suppose the system's noise can be decomposed into the sum of multiple Gaussian noises, which is what we call a multimode problem. In that case, we can decompose the noise first and then use the filters described in the previous section for each Gaussian noise component.

Suppose $p(x)$ is a mixed Gaussian distribution with n Gaussian components

$$p(x) = \sum_{i=1}^n \omega_i N(x; \mu_i, P_i) \quad (11)$$

where $N(x; \mu_i, P_i)$ is each Gaussian component and the weight satisfies $\sum_{i=1}^n \omega_i = 1$.

Next, we need to determine the weight ω_i , the mean μ_i , and variance P_i of each Gaussian component. Then, the estimation methods in Section 2.1 can be used to obtain the estimations based on each component with the Gaussian noise. At last, all the estimations can be weight-summed by ω_i . A flowchart of the Gaussian mixture filter is shown in Figure 1.

To avoid too many components making subsequent calculations too complicated, it is usually necessary to reduce the number of Gaussian components. In other words, it is necessary to describe as accurately as possible with as few components as possible. The methods of component reduction include retaining those significant components that affect the distribution function by selecting some of the components with larger weights and discarding those that are not significant. Furthermore, by combining similar components, the total number of components is reduced.

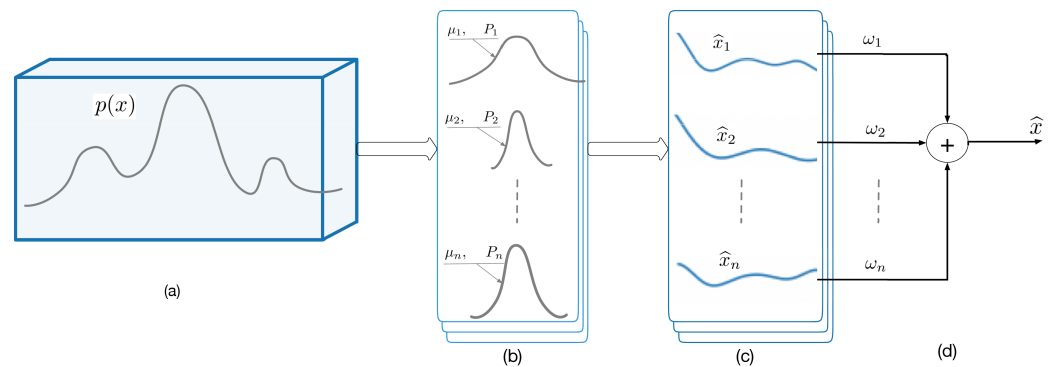


Figure 1. A flowchart of the Gaussian mixture filter. (a) The system noise is a complicated one with multiple Gaussian components. (b) The complex Gaussian noise of the system is decomposed into several standard Gaussian noises P_i , $i = 1, 2, \dots, n$. (c) The state \hat{x}_i is estimated based on each Gaussian noise. (d) The state estimation result based on the complex mixed high noise is obtained by using the estimated result and the weight.

Suppose the system's noise is more complex and cannot be expressed as the approximate sum of multiple Gaussian noises. The particle filters appear for such a problem [40,41]. In that case, a numerical approximation method is required, that is, Monte Carlo simulation. The probability distribution of the state is calculated by selecting multiple samples and using statistical methods. However, the computational overhead is very high. Additionally, in the process of calculation, due to particle degradation, sequential importance resampling processing is required.

Figure 2 shows the flow of the particle filter algorithm. A set of particles represents the samples in the distribution. Each particle is assigned a likelihood weight, representing the probability of sampling. The weights tending to be the same is a common problem with these filtering algorithms. However, this can be alleviated by resampling before the weights become too uneven. In the resampling step, particles with smaller weights will be replaced by new particles.

Two reviews provide details about particle filters, [42,43]. The first one focuses on particle filters based on the optimal and suboptimal Bayesian algorithms for nonlinear and non-Gaussian tracking problems. Within the general framework of the sequential importance sampling (SIS) algorithm, several variants of the particle filter are introduced, such as sampling importance resampling (SIR), auxiliary sampling importance resampling (ASIR), and regularized particle filter (RPF). The second one discusses new developments and demonstrates that particle filters are useful in even the largest dimensional geophysical data assimilation problems.

At the end of this subsection, we want to discuss an interesting fact about particle filters and Gaussian mixture filters. That is, particle filters were popular about 20 years ago [44], while Gaussian mixture filters have only become a hot topic in the past decade [45]. The reason is that when the system encounters complex noise, people first use particle filters to complete the task. However, the computational complexity of the particle filter is too large. To improve real-time performance, many research results show a compromise between real-time performance and accuracy. On the other hand, people find that it is possible to use Gaussian mixture filters to complete the task by decomposing complex noise, improving state estimation's real-time performance while ensuring estimation performance. However, when encountering difficult and incredibly complex noise, particle filters can show better advantages. Therefore, there are still many application examples for particle filters in video tracking [46,47]. However, in target tracking, the Gaussian mixture filter has attracted more attention from researchers due to its small amount of calculation [48,49].

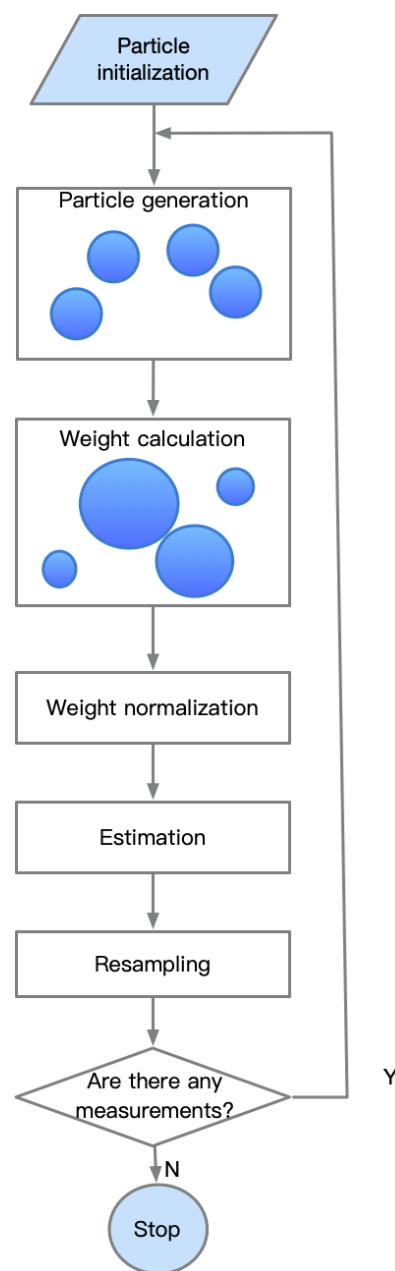


Figure 2. The flow of the particle filter algorithm. The particles are generated firstly, and then, the weights need to be calculated according to the states' accuracy. Next, the weights are normalized, the state is estimated, and the particles are resampled. The loop is continued until all the measurement data are used.

2.3. Discussion

Table 1 provides some performance comparison of the filters. We can find that the Kalman filter requires the system to be linear, with Gaussian white noise, and has a low calculation cost. Meanwhile, in practice, it is hard to meet such requirements for the system; therefore, in an actual system, the obtained estimation accuracy is low.

When the system is nonlinear and contains Gaussian noise, the UKF and CKF are the first choices. This is because they do not require a large amount of calculation and can obtain more accurate estimation results than the EKF. Of course, if the system's nonlinearity is not strong, the EKF is also a good choice. However, if the system contains complex noise, the Gaussian mixture filter is the first consideration. As mentioned earlier, we need to decompose the complex noise into Gaussian noise series and then use the Gaussian mixture

filter to complete the estimation task. If the decomposition cannot be achieved effectively, we suggest selecting the particle filter to ensure the actual estimation performance.

Table 1. Comparison of each filter.

Filter	Requirements for the System	Accuracy for a Practical System	Calculation Cost	Description
Kalman filter	Linear, with Gaussian white noise	Low	Low	The requirements for the system are very high, so it is difficult to achieve high accuracy in the actual application system.
EKF	Nonlinear, with Gaussian noise	Medium	Low	The performance of UKF and CKF is better than that of EKF, but their calculation amount is slightly larger than that of EKF.
UKF	Nonlinear, with Gaussian noise	Medium	Medium	
CKF	Nonlinear, with Gaussian noise	Medium	Medium	These filters have low requirements for the system. However, the amount of calculation is large.
Gaussian mixture filters	Nonlinear, with non-Gaussian noise	Medium	Medium	
Particle filters	Nonlinear, with non-Gaussian noise	High	High	

To obtain an accurate system model, a system identification method is used to establish the mathematical models of dynamical systems [50–53], and some identification approaches can be used to establish the prediction models and soft sensor models [54–57]. Nevertheless, it is still difficult to obtain an accurate model in the existing system. Estimation methods for inaccurate models are also attracting attention.

3. State Estimation Based on a Blurry Model

The estimation method in Section 2.1 requires a given system model, including system parameters and the probability distribution of the noise contained in the system. However, the system model parameters and noise distribution characteristics are usually not accurately obtained due to the large scale of the existing system, the complicated relationship between variables, and the influence of the state process, measurement method, surrounding environment, etc.

The estimation performance degradation caused by the low degree of model matching has always existed in actual system applications. It makes the estimation results unable to reach the ideal performance of theoretical research. Simultaneously, it has also promoted the development of classical estimation theory in the past few decades. Below, we introduce the estimation theory and method based on a blurry model from two aspects of a “robust filter” and “closed-loop filter”.

3.1. Robust Filter

When the system model parameters are inaccurate, the state can be corrected by modeling the uncertainty term. Based on this, a robust filtering method has been designed to improve the estimation performance. This method first describes the incorrect system model parameters as the perturbation range and then designs the filter gain based on it. The estimated covariance is bounded within the perturbation range of the parameters. The specific methods include guaranteed performance estimation, H_∞ filtering, etc. [11,13,58].

To guarantee performance estimation based on the system’s maximum uncertainty, the estimator is designed to minimize the steady-state filter covariance’s upper bound. For example, under the Bayesian framework, Dehghannasiri et al. [59] proposed a robust Kalman filter based on the orthogonal theory, Nishanthi et al. [60] studied the guaranteed performance filtering method for uncertain discrete systems under random hysteresis

conditions, and Roy et al. [61] designed a guaranteed performance filter for quantum phase change.

Wang et al. [62] designed a time-varying estimator to realize distributed state estimation based on a type of sensor network with random model parameters and nonlinearity. Under the premise of known topology information, based on Bernoulli's random distribution, Li and Chang [63] studied the problem of robust filtering and feedback control with random occurrence characteristics for uncertain systems.

The above research results are mainly based on Lyapunov stability. The basic idea is to transform the robust filtering problem into a feasible solution of the Riccati matrix equation and then provide the conditions and robust filters to ensure robust filtering performance. Although the Riccati equation can achieve the method, some parameters still need to be provided in advance. The choice of parameters affects the performance of the result and affects the solvability of the equation. Due to the lack of suitable parameter optimization methods, these parameters need to be set in the design process, which brings conservativeness to the results. On the other hand, Riccati matrix equations are mostly solved by iterative methods, which cannot guarantee the solution process' convergence.

In the 1990s, with the introduction of the interior point method for solving convex optimization problems, the linear matrix inequality (LMI) method attracted the academic community's attention. This method can overcome the shortcomings in solving the Riccati equation, provide a convex constraint condition for the problem to be solved, and then obtain the robust filtering problem's solution by solving the convex optimization problem [64].

The robust H_∞ filtering considers the following uncertain discrete linear time-invariant system:

$$\begin{aligned}x_{k+1} &= (A + \Delta A_k)x_k + B\omega_k \\ y_k &= (C + \Delta C_k)x_k + Dv_k\end{aligned}\quad (12)$$

where x_k is the state to be estimated, and ΔA_k and ΔC_k are the perturbation of the parameter, and they can be written as

$$\begin{bmatrix} \Delta A_k \\ \Delta C_k \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} F_k E \quad (13)$$

where H_1 , H_2 , and E are the constant matrix with appropriate dimensions, and F_k satisfies the following:

$$F_k^T F_k \leq I \quad (14)$$

If we have the constant β ; symmetric positive definite matrix X , Y ; symmetric matrix U ; and asymmetric matrix M , S , T based on a given constant $\gamma > 0$, satisfying the following,

$$\begin{aligned} & \begin{bmatrix} X & I \\ I & Y \end{bmatrix} > 0 \\ & \begin{bmatrix} X & M \\ M^T & U \end{bmatrix} > 0 \\ & \begin{bmatrix} -Y & * & * & * & * & * & * & * \\ -I & -X & * & * & * & * & * & * \\ 0 & 0 & -\gamma^2 I & * & * & * & * & * \\ A^T & A^T X^T + C^T T^T & I & -X + \beta E^T E & * & * & * & * \\ 0 & S^T & -I & -M^T & -U & * & * & * \\ B^T & B^T X^T & 0 & 0 & 0 & -I & * & * \\ 0 & D^T T^T & 0 & 0 & 0 & 0 & -I & * \\ H_1^T & H_1^T X^T + H_1^T T^T & 0 & 0 & 0 & 0 & 0 & -\beta \end{bmatrix} < 0 \end{aligned} \quad (15)$$

then we can have the robust filter

$$\hat{\mathbf{x}}_{k+1} = \mathbf{A}_f \hat{\mathbf{x}}_k + \mathbf{B}_f \mathbf{y}_k \quad (16)$$

with the parameter

$$\begin{aligned} \mathbf{A}_f &= \mathbf{M}^{-1} \mathbf{S} \\ \mathbf{B}_f &= \mathbf{M}^{-1} \mathbf{T} \end{aligned} \quad (17)$$

satisfying the estimation covariance

$$E\{e_k e_k^T\} < \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{X} & \mathbf{M} \\ \mathbf{M}^T & \mathbf{U} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \end{bmatrix} \quad (18)$$

In summary, based on the system's uncertainty model, robust filtering can perform state estimation based on minimizing the error covariance or performance indicators for all the possible uncertainties of the actual system. However, for a specific practical application, the state estimation accuracy provided is not high.

Therefore, although robust filtering was once a hot research direction in state estimation at the beginning of this century, it has not been widely used in practice because it cannot meet the existing system requirements for state estimation performance. The reason is that it is still very difficult to accurately model the uncertain part first. If the uncertain part cannot be accurately described, then the robust filtering performance is not high. Recently, some research has tried to develop the performance for practice. For example, Liu et al. [65] combined the unscented Kalman filter with the H_∞ filter for state estimation for nonlinear non-Gaussian systems.

3.2. IMM and Closed-Loop Adaptive Filter

Based on the fact that "complex system parameters and structures will vary with different applications", researchers have provided another solution to improve the estimation performance: improve the model to make it more in line with the actual system.

There are two categories for the related methods:

- (1) Multiple system models describe parts of the system and then combine to describe the whole system;
- (2) Measurement information is applied to continuously optimize the online system model to make the system model as consistent as possible with the current situation.

The former is represented by the "multimodel method" proposed by Bar-Shalom, which simultaneously uses multiple models for estimation, evaluates the estimation results of different models, and then designs a reasonable weighting strategy to obtain the final estimation result [66].

The interacting multiple model (IMM) includes a series of basic models. Then, the estimated results are generated by weighting each model's estimation so that the tracking performance is better than that of any single model.

The IMM supposes that the probability switching among models obeys a Markov chain, and assumes the prior model number of the system is N , the sets of the models are $M(k) = \{m_i(k)\}$, $i = 1 \cdots N$, and the transition probability from model i at k time switching to model j at $k + 1$ time is PT_{ij} . Then, the model transition probability matrix is as follows

$$PT = \begin{bmatrix} PT_{11} & PT_{12} & \cdots & PT_{1N} \\ PT_{21} & PT_{22} & \cdots & PT_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ PT_{N1} & PT_{N2} & \cdots & PT_{NN} \end{bmatrix} \quad (19)$$

where each row denotes the current model of the k sampling time; each column denotes the prediction model of the $k + 1$ sampling time, i.e., the next sampling time. Note that PT is a given matrix based on prior knowledge.

Then, the probability prediction of the model i to model j is calculated:

$$PT_{\mu}(k+1|k) = \begin{bmatrix} PT_{11}\mu_1(k) & PT_{12}\mu_1(k) & \cdots & PT_{1N}\mu_1(k) \\ PT_{21}\mu_2(k) & PT_{22}\mu_2(k) & \cdots & PT_{2N}\mu_2(k) \\ \vdots & \vdots & \vdots & \vdots \\ PT_{N1}\mu_N(k) & PT_{N2}\mu_N(k) & \cdots & PT_{NN}\mu_N(k) \end{bmatrix} \quad (20)$$

where $PT_{ij}\mu_i(k)$ denotes the prediction joint probability of the model i transferred to model j . Each element in the matrix is added at each column; then, the prediction probability of model j at the $k + 1$ time can be obtained:

$$\mu_j(k+1|k) = \sum_{i=1}^N PT_{ij}\mu_i(k) \quad (21)$$

Then, the probability prediction of the model is given as

$$\mu_{j/i}(k) = \frac{PT_{ij}\mu_i(k)}{\mu_j(k+1|k)} \quad (22)$$

The interactive probability is the probability of the model i at the current time k related to the model j in the future $k + 1$ time.

Based on this, researchers have developed multimodel methods for nonlinear and uncertainty parameters. However, this type of method requires multiple models to be calculated in parallel and weighted according to each model's matching degree with actual system, so the amount of calculation is relatively large. Worse, in practical applications, if the model group contains models with low matching degrees, it will lead to a decrease in the estimation accuracy, so using too many models will not improve the estimation performance.

Another method is to update model parameters in real-time based on state estimation results, such as the expectation–maximization (EM) method and hybrid grid multimodel [67]. Based on this idea, the applicant has also proposed a closed-loop adaptive model [68,69]. For example, Bai et al. [70] built a framework of dynamic noise based on a gyroscope's noises, and developed the dynamic Allan variance to build the adaptive Kalman filter.

This type of model can continuously optimize the system model's system parameters based on real-time estimation results so that the system model matches the actual system as much as possible. Figure 3 shows the optimization loop of the system model and state. In this type of system, optimal convergence is the most critical issue. We can find that, in such a system, the state and the model are optimized sequentially. The model is initialized first, and then, the state is updated; based on the estimated state obtained, the model parameters are further updated.

However, the optimization based on the current estimation results will cause the model update to lag behind the actual system. The simultaneous update of the state and model parameters may also make the system parameter optimization process unable to converge. These methods can combine identification approaches [71–75] for studying the modeling of dynamic time series and stochastic systems with colored noises [76–80] and can be applied to other fields such as signal modeling, tracking and control systems [81–83].

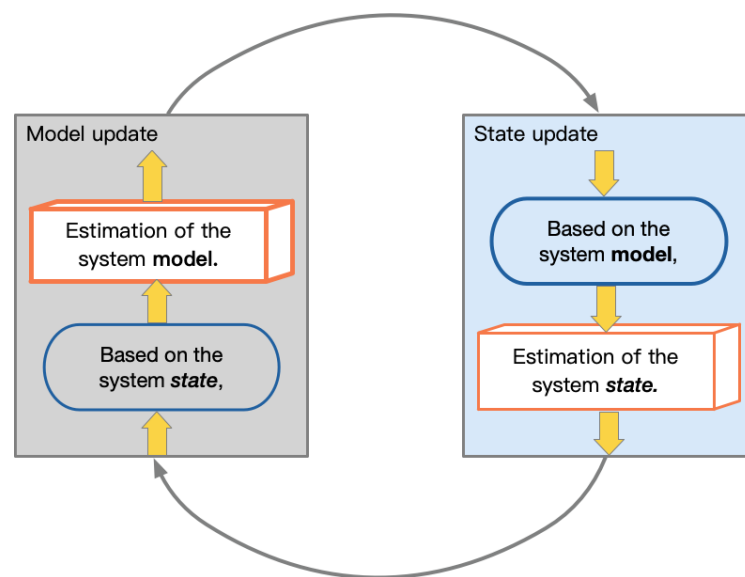


Figure 3. The optimization loop of the system model and state.

4. Data-Driven Modeling by Learning

In this section, we introduce three parts: different deep learning networks, the optimization of their hyperparameters, and one of their learning methods: Bayesian learning.

The network cell is the main part for the data-driven model, which decides the performance of the model. Meanwhile, the choice of hyperparameters has a great influence on the convergence of the network. Therefore, we focus on the hyperparameter optimization in Section 4.2.

As for the measurement for the sensors, the measurement noise has to be firmly considered. Unlike the classic deep learning network, the neural network's weight should have the capability to model the input noise characteristics. Section 4.3 describes the so-called Bayesian deep learning network: the methods for Bayesian variational inference and some applications. Table 2 provides the whole picture of the state of the art for the data-driven model method, from which we can find that as the most general application field, “prediction” has the most papers. Meanwhile, recently, some research work has focused on the estimation.

Table 2. Data-driven modeling methods with deep learning networks.

References	Network Cell	Hyperparameter Optimization	Type of Network	Purpose
[84]	Long short-term memory (LSTM)	Not mentioned	Classic deep learning network	Classify sequence
[85]	Gated recurrent unit (GRU)	Not mentioned	Classic deep learning network	Forecasting time-series data
[86–88]	Recurrent neural network (RNN)	Not mentioned	Classic deep learning network	Machine translation
[89]	Attention-based LSTM	Not mentioned	Classic deep learning network	Machine translation
[90]	Convolution network	Bayesian optimization	Classic deep learning network	Prediction
[91–93]	GRU	Bayesian optimization	Classic deep learning network	Prediction
[94]	Bidirectional RNN	Not mentioned	Classic deep learning network	Detection
[95]	ConvLSTM	Not mentioned	Classic deep learning network	Prediction
[96]	RNN	Not mentioned	Classic deep learning network	State estimation
[97]	GRU	Manual search	Classic deep learning network	Prediction
[98]	LSTM	Manual search	Bayesian deep learning network	Prediction
[99]	GRU	Bayesian optimization	Classic deep learning network	Prediction
[100,101]	Multi-layer forward neural network	Not mentioned	Bayesian deep learning network	State estimation

4.1. Deep Learning Network

Unlike the traditional models in Section 3, the neural network model based on machine learning does not require other prior physical information or model assumptions [102,103]. In the model training stage, the model is obtained by learning the data's hidden relation-

ships and knowledge. This type of model can process data with nonlinear characteristics and has been effectively applied to prediction tasks in different fields [104,105].

As the most important machine learning branch, deep learning methods break through the bottleneck of traditional machine learning methods and promote artificial intelligence development. The recurrent neural network (RNN) has become an important process modeling method because it can capture a nonlinear relationship. However, due to the RNN structure's limitation, the effect will worsen when dealing with longer sequences. The emergence of long short-term memory (LSTM) solved the long-term dependence problem of RNNs [84].

The interior of the LSTM cell has three gating units, which are the input gate, forget gate, and output gate. The calculation process is

$$\begin{aligned}
 f_t &= \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \\
 i_t &= \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \\
 \tilde{c}_t &= \tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \\
 c_t &= f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \\
 o_t &= \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \\
 h_t &= o_t \cdot \tanh(c_t)
 \end{aligned} \tag{23}$$

Among them, o_t represents the output of LSTM, and h_t represents the output of the last layer of LSTM.

Differently from the LSTM, the gated recurrent unit (GRU) network further simplifies the structure of the LSTM network while maintaining the accuracy of the prediction. The GRU model consists of two parts, the update gate and the reset gate.

The formula of the GRU cell is shown below:

$$\begin{aligned}
 z_t &= \sigma(x_t U^z + h_{t-1} W^z + b^z) \\
 r_t &= \sigma(x_t U^r + h_{t-1} W^r + b^r) \\
 \tilde{h}_t &= \tanh(x_t U^h + (h_{t-1} \odot r_t) W^h + b^h) \\
 h_t &= (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1}
 \end{aligned} \tag{24}$$

With the GRU and LSTM cell, the network can be constructed. Figure 4 provides an example of a GRU for PM2.5 prediction with two layers [85], in which we find that the deep learning networks still cannot capture the complex dynamic relationship of the data, so the data are decomposed into several components.

In recent years, encoders and decoders based on data modeling have been able to obtain more stable information from data and been suitable for the lengths of input data and output data. One reason is that the encoder extracts features from the input data: the extracted features are the hidden state of the RNN, and the output state of the encoder is used as the initial state of the decoder to obtain the target signal [86]. The other is that the encoder encodes the input data into a context vector and outputs the variable-length sequence as a fixed-length sequence [87].

As the dependency of the input data increases, the performance of data-based modeling declines rapidly. Therefore, by introducing an attention mechanism, different attention weights can be assigned to all the time steps. The relevant encoder hidden state can be adaptively selected in all the time steps, and highly relevant features and output sequences can be extracted. There are two common attention mechanisms: Bahdanau attention [88] and Luong attention [89], also known as additive attention and multiplicative attention.

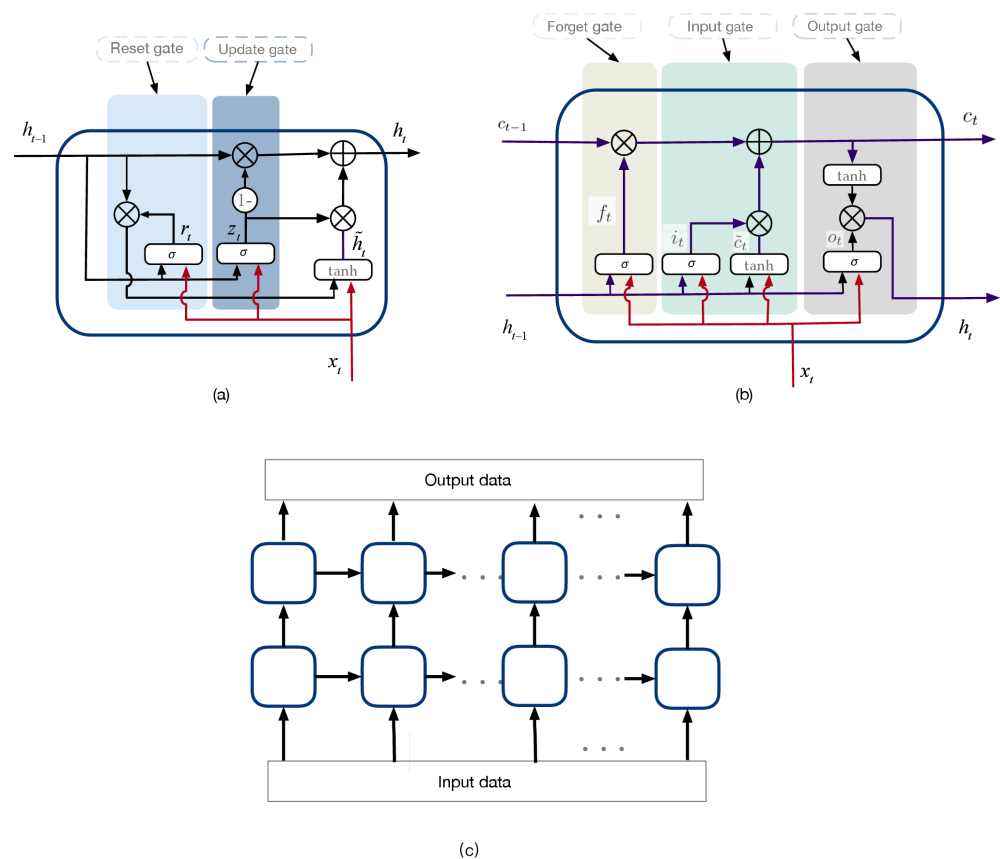


Figure 4. Data-driven model. (a) GRU cell; (b) LSTM cell; (c) deep learning network.

4.2. Hyperparameter Optimization

Due to the increase in data complexity, deep learning methods' modeling performance needs to be improved. Especially, the hyperparameters of the network greatly influence the modeling performance, so setting the network's hyperparameters is very important. Hyperparameters determine the model's performance in deep neural networks; therefore, it is extremely important to optimize the hyperparameters [106].

Hyperparameters need to be set before starting the learning process, not the parameter data obtained through training, but including the number of layers of the network, the number of neurons in each layer, the type of activation function, etc. One hyperparameter, the weight's training process, is shown in Figure 5, where Figure 5a is the training process for the standard deep learning network to obtain the weights. Based on them, the part shown in Figure 5b evaluates the network performance, defines the weights' optimization space, and makes it easier to converge to the optimal weights.

At present, hyperparameter optimization methods mainly include manual search, web search, random search, and Bayesian optimization. Among them, a manual search is a process of optimizing hyperparameters manually through trial and error. This method is very time-consuming and relies on experience to determine the optimal hyperparameters. Grid search is traversing all the candidate parameters, trying every possibility, and comparing the values of the objective function one by one to the points that meet the constraints, which is the final result. It is also very time-consuming. Random search is a method of using random numbers to find the optimal solution for function approximation. It is different from the "violent" search method of grid search. This method is based on the theory of probability and continuous randomness within a certain interval. Instead of generating random points with a tendency to calculate the value of the constraint function and the objective function, the values of the objective function are compared one by one for the points that meet the constraint conditions.

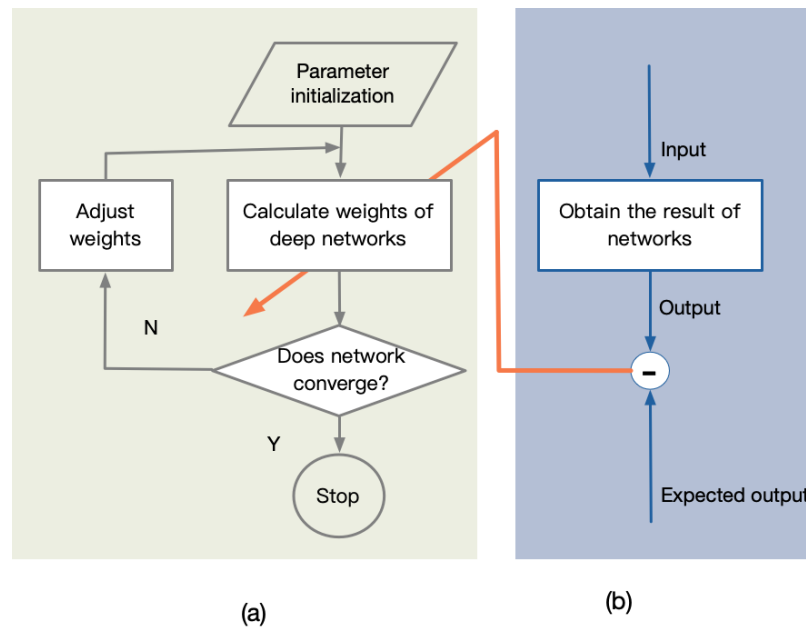


Figure 5. Hyperparameter optimization process. (a) The training process; (b) The weights' optimization process.

Unlike these methods, the Bayesian method will refer to the previous evaluation results when trying the next set of hyperparameters, so it can save a lot of calculations. The Bayesian optimization problem has four parts, which are the objective function, domain space, optimization algorithm, and result history. The principle is to establish a distribution function (probability model) based on the past evaluation results for the objective function, in the constructed domain space, to find the value that minimizes the objective function. Wang et al. [90] used the Bayesian algorithm to optimize the convolution prediction network's hyperparameters by modeling the complex time-series data sets. The study [91] used a GRU network for each component of power load data, in which Bayesian optimization parameters were used for each subpredictor. The results showed that the Bayesian optimization could effectively improve the prediction accuracy.

4.3. The Ability to Model System Noise

The neural network's ability to model nonlinear relationships has been fully proven theoretically; i.e., the neural network can fit any nonlinear relationship. In practical applications, neural networks, especially deep neural networks, have shown strong advantages in practical application systems, with their powerful nonlinear modeling capabilities. However, an important problem to be faced in the estimation system is the system's complex noise. Overcoming the influence of complex noise on the estimation performance is one of the important attributes for estimation methods' performance.

We believe that the way the neural network obtains the weight should consider the input noise characteristics. The Bayesian training process can replace the fixed weights and bias of the traditional neural network. The network training is conducted by sampling, thereby constructing the model's uncertainty, suppressing the influence of noise, and obtaining more accurate predictions. The neural network output is no longer a fixed sequence but a sequence with a fluctuation range, thereby improving the prediction results' reliability. Bayesian deep learning networks' training methods usually include two methods: approximating the integral with Markov chain Monte Carlo (MCMC) and black-box variational inference. We can estimate the true full cost function by Monte Carlo sampling and then backpropagate using the estimated value.

Bayesian deep learning believes that every weight and bias should be a random distribution, not a certain value. Different from the constant weights and biases obtained by non-Bayesian neural network training, the training will obtain the distribution of

weights and biases for Bayesian deep learning. The difference between the Bayesian deep network and non-Bayesian deep network is shown in Figure 6.

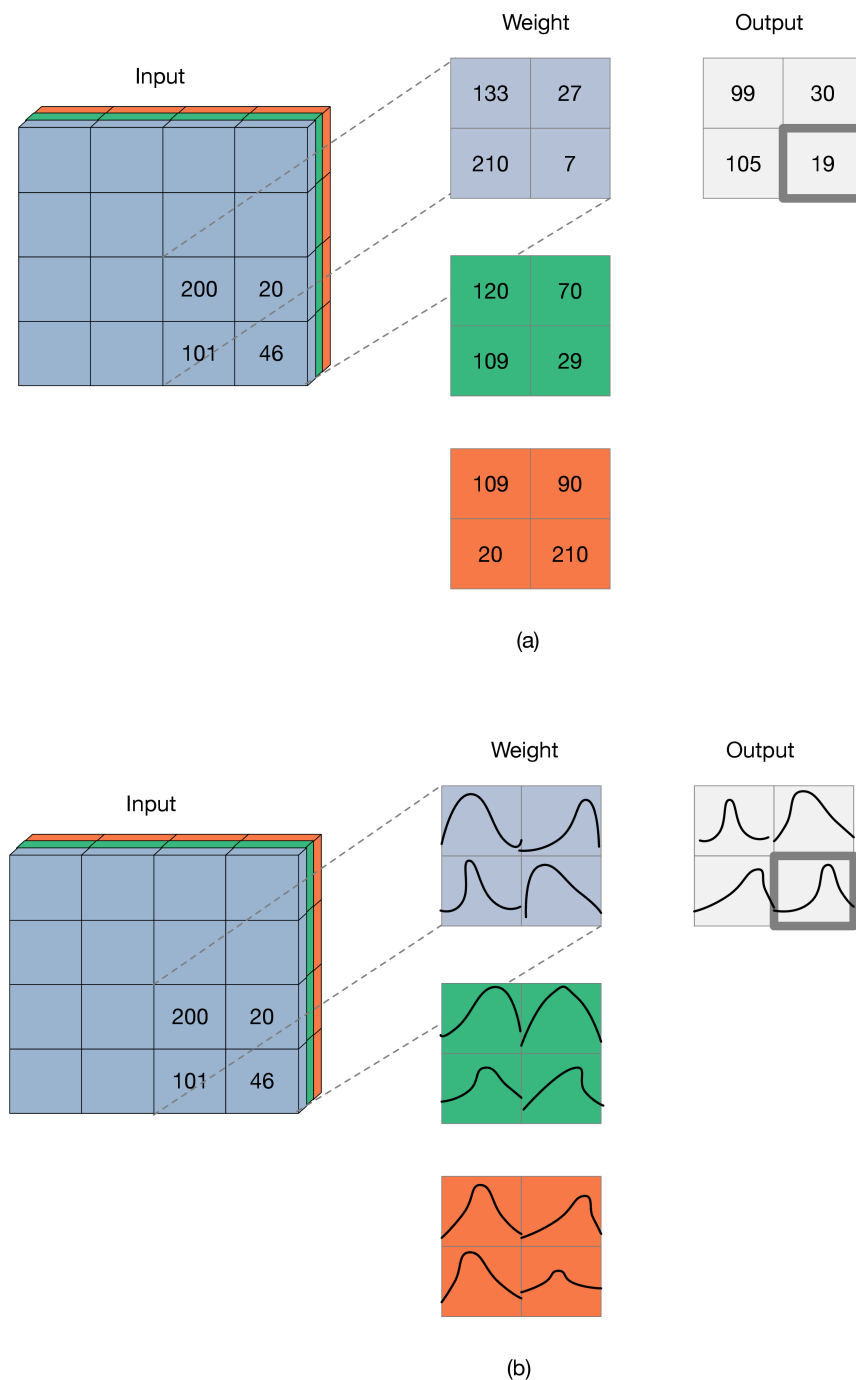


Figure 6. The differences between the Bayesian deep network and non-Bayesian deep network are (a) the weight optimization process for the non-Bayesian deep network, where the weight is a certain value, and (b) the weight optimization process for the Bayesian deep network, where the weight's distribution is obtained.

Given a training data set $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$, suppose we want to obtain the posterior probability $p(w | x, y)$. Then, according to the Bayes theory, we get

$$p(w | x, y) = \frac{p(y | x, w)p(w)}{\int p(y | x, w)p(w)dw} \quad (26)$$

However, the fact is that the denominator has to integrate on the value space of w , and the space formed by these weights together will be quite complicated. In general, variational inference is used to solve the integral expression of the denominator. This method uses a simple point distribution $q(w)$ to approximate the posterior probability distribution $p(w | x, y)$, and uses Kullback–Leibler (KL) divergence to calculate the difference between minimizing the distribution $q(w)$ and $p(w | x, y)$.

Data-driven methods have been used in state estimation research, which greatly improve the performance of state estimation. The study [107] proposed training a shallow neural network with historical data to “learn to initialize”, that is, to map the available measured values to a point in the neighborhood of the true latent states. The results show that, compared with traditional optimization methods, this machine learning-based optimization method produces superior performance in terms of stability, accuracy, and runtime efficiency. The study [108] was based on the Bayesian inference method for state estimation and used it for sensor transmission power control. Experiments prove that this method can improve the estimation performance.

5. State Estimation Based on Hybrid-Driven Methods

With the development of technologies such as sensors, wireless communications, and data storage, more and more historical data are stored in modern complex large systems, forming measurement “big” data. These data contain process information based on the system parameters and noise characteristics that cannot be expressed by the system’s mathematical model. We believe that these data will be used to develop the performance of the simplified system model.

As long as we have enough measurement data, models describing data relationships can be obtained through offline learning without prior knowledge. In recent years, in state estimation, researchers have made some attempts to explore state estimation algorithms that combine the Kalman filter and neural network.

Specifically, they are mainly divided into the following three categories:

Firstly, the network output is the difference between the reference state and the classical estimation, which is used to correct the estimated state based on the system model. For example, Liu et al. [109] trained an LSTM network to predict the difference between the estimated result and the reference trajectory and modify the estimated result. The study [110] proposed a multisensor fusion algorithm for underwater vehicle localization by the augmentation of the radial basis function (RBF) neural network with the EKF, in which the RBF neural network was utilized to compensate for the lack of EKF performance. It simply used the classic state estimation method to evaluate the difference between the actual complex system’s reference trajectory and the estimation result. Similar methods include using learning algorithms to improve the extended Kalman filter [111] and the multimodel extended Kalman filter [112]. In principle, this type of method does not modify the model but uses the difference between the reference state and the classical estimation result to describe the model’s matching degree. When the model error is large, this compensation cannot completely offset the model error.

The second method is to learn the parameters of the system’s model under the framework of Bayesian estimation [113,114]. For example, Zhao et al. [113] trained the network to obtain the state transition matrix, state noise variance, measurement matrix, and measurement noise variance of the system model’s monocular visual tracking. After obtaining these parameters by learning LSTM networks, the state is still estimated based on the system model and the Kalman filter framework. This type of method has the same purpose as the optimization parameter method in Section 3.2. The difference is that the method in Section 3.2 uses a statistical optimization strategy, while here, the learning method is used to obtain system parameters after offline training. However, this method is essentially system parameter identification, which cannot fundamentally solve the estimation performance degradation caused by the system model’s low matching degree, especially the model structure.

The last one is to obtain the estimation state directly [115–118]. For example, Gao used measurement data as input and the reference state as output data to train the LSTM network [115]; Bai used the NARX (Nonlinear autoregressive with external input) network to learn the complex relationship between the measurement, state, and filter gain in nonlinear systems [116]. This method's characteristics are as follows: It is necessary to obtain the state reference value through simulation. The system is modeled through supervised learning, and then, the network output is used as the result of the state estimation. This method can use the network model's ability to model complex data relationships and improve state estimation performance to a certain extent. However, it must be pointed out that the degree of matching between the simulation reference in the method and the actual application system determines the practicability of the model.

6. Conclusions

When the model can be accurately obtained or completely consistent with a practical system, the state estimation method using the Kalman filter can produce the best result. For the nonlinear model, the EKF, UKF, and CKF are effective supplements to the standard Kalman filter, and the estimation results for the nonlinear system with Gaussian noise can be obtained. Although this estimation result is no longer optimal, this method is simple and easy to perform effective real-time estimation with, so it has always been used in many applications. However, such methods in existing complex systems, especially in modern complex systems, are not good enough. To reduce the estimation method's dependence on the model, researchers have provided robust filters, IMM, and closed-loop adaptive filter methods that can use multiple models to describe the system effectively.

Nevertheless, we found that with modern complex systems, the system contains more and more sensors. The system's internal structure becomes more and more complex, making the modeling of the system more difficult. On the other hand, the system contains more and more sensors, and we can obtain more and more sensor signals. Simultaneously, due to the development of storage technology, we can store these data and generate so-called big data. Based on these big data, we can further describe the model with high accuracy and obtain a more accurate model. Therefore, the state estimation method based on the learning network has been one of the hot spots of people's attention in recent years.

However, because these methods only use the network to fit the nonlinear system, the network modeling results are not good enough when the system is too complex. Moreover, the traditional neural network modeling ability is further reduced due to sensor noise. Therefore, in recent years, state estimation research has tended to combine the traditional model-based estimation method and the data-driven network model method, in the so-called hybrid-driven state estimation method. In the past three years, related research has shown an increasing trend. It has brought a lot of vitality to the relatively quiet traditional state estimation methods for many years and opened up a new field for studying state estimation methods.

In this review, we summarized and analyzed the three current methods of this research direction. We believe that these studies are very preliminary. There is still a lot of research to be conducted on how to effectively combine the traditional state estimation method based on Bayesian estimation theory with a strict mathematical foundation and the model-driven modeling method that uses the big data complex network model to describe the system. In this regard, we believe that the following research trends are worthy of attention.

The first is how to use sensor-based big data. Compared with the problems studied by traditional deep learning networks, the network model applied to state estimation needs to overcome sensor data noise, which is a more difficult subject for traditional learning networks. Traditional neural network methods can effectively solve nonlinear modeling problems. In theory, researchers have even proved that neural networks can best fit any nonlinearity. Traditional neural networks are prone to overfitting to noise.

Second, the state estimation method should be developed for complex networks. The current hybrid state estimation method does not fundamentally use the advantages of the

model-driven and data-driven method. Therefore, the author believes that the research to solve this problem effectively will be of great significance. It can fundamentally combine the advantages of the estimation framework and the nonlinear learning network.

Third, the theory based on the hybrid model needs to be studied in depth. Based on the state of the hybrid drive, the convergence of the estimation method, the requirements for training data, the discussion of real-time performance, etc., are all worthy of our attention.

This review aimed to study the development context of state estimation and describe the development trend of future state estimation. We believe that state estimation will still be an important part of practical systems. For such complex systems, it will bring more difficulties and new solutions to the state estimation methods.

Author Contributions: Conceptualization, X.-B.J.; data curation, Y.-T.B.; formal analysis, T.-L.S. and Y.-T.B.; methodology, R.J.R.J. and J.-L.K.; software, R.J.R.J.; supervision, J.-L.K.; validation, T.-L.S.; visualization, X.-B.J.; writing—original draft, R.J.R.J.; writing—review and editing, X.-B.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (No. 61903009 and 62006008), Beijing Municipal Education Commission (No. KM201910011010 and KM201810011005), Young Teacher Research Foundation Project of BTBU (No. QNJJ2020-26), Defense Industrial Technology Development Program (No. 6142006190201), and Beijing Excellent Talent Training Support Project for Young Top-Notch Teams (No. 2018000026833TD01).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Rubio, F.; Valero, F.; Llopis-Albert, C. A review of mobile robots: Concepts, methods, theoretical framework, and applications. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 172988141983959. [\[CrossRef\]](#)
- Xin, Z.; Pei, L.; Junwei, C. Multi-UAV Cooperative Target Tracking Control Based on Nonlinear Guidance. *Command. Inf. Syst. Technol.* **2019**, *10*, 47–54.
- Muzammal, M.; Talat, R.; Sodhro, A.H.; Pirbhulal, S. A multi-sensor data fusion enabled ensemble approach for medical data from body sensor networks. *Inf. Fusion* **2020**, *53*, 155–164. [\[CrossRef\]](#)
- Ma, X.; Guo, R.; Liu, J.; Fang, C.; Zhu, P.; Zhang, Y. State estimation of AC and DC distribution network under three-phase unbalance. *Autom. Electr. Power Syst.* **2019**, *43*, 65–71.
- Zhang, X.; Zhao, Z.; Wang, Z.; Wang, X. Fault Detection and Identification Method for Quadcopter Based on Airframe Vibration Signals. *Sensors* **2021**, *21*, 581. [\[CrossRef\]](#)
- Zhao, Z.Y.; Wang, X.Y.; Yao, P.; Bai, Y.T. A health performance evaluation method of multirotors under wind turbulence. *Nonlinear Dyn.* **2020**, *102*, 1701–1715. [\[CrossRef\]](#)
- Sorenson, H.W. *Kalman Filtering: Theory and Application*; IEEE Press: New York, NY, USA, 1985.
- Wiener, N. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*; John Wiley & Sons: New York, NY, USA, 1949.
- Kalman, R.E. A new approach to linear filtering and prediction problems. *Trans. ASME J. Basic Eng.* **1960**, *82*, 35–45. [\[CrossRef\]](#)
- Qin, Y.Y.; Zhang, H.Y.; Wang, S.H. *Principles of Kalman Filtering and Integrated Navigation*; Northwestern Polytechnical University Press: Xi'an, China, 1998.
- Zorzi, M. Robust Kalman Filtering Under Model Perturbations. *IEEE Trans. Autom. Control* **2017**, *62*, 2902–2907. [\[CrossRef\]](#)
- Fu, M.; Deng, Z.H.; Zhang, J.W. *Kalman Filtering Theory and Its Application in Navigation System*; Science Press: Beijing, China, 2010.
- Hedayati, M.; Rahmani, M. Robust distributed H_∞ filtering over an uncertain sensor network with multiple fading measurements and varying sensor delays. *Int. J. Robust Nonlinear Control* **2020**, *30*, 538–566. [\[CrossRef\]](#)
- Julier, S.J.; Uhlmann, J.K. A new approach for filtering nonlinear system. In Proceedings of the 1995 American Control Conference, Seattle, WA, USA, 21–23 June 1995; pp. 1628–1632.
- Julier, S.J.; Uhlmann, J.K. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Trans. Autom. Control* **2000**, *45*, 477–482. [\[CrossRef\]](#)
- Norgarrd, M.; Poulsen, N.K.; Ravn, O. New developments in state estimation for nonlinear systems. *Automatica* **2000**, *36*, 1627–1638. [\[CrossRef\]](#)

17. Julier, S.J.; Uhlmann, J. Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations. In Proceedings of the American Control Conference, Anchorage, AK, USA, 8–10 May 2002; pp. 887–892.
18. Arasaratnam, I.; Haykin, S. Cubature Kalman smoothers. *Automatica* **2011**, *47*, 2245–2250. [[CrossRef](#)]
19. Zhang, P.; Li, B.; Boudaren, M.E.Y.; Yan, J.; Li, M.; Wu, Y. Parameter Estimation of Generalized Gamma Distribution Toward SAR Image Processing. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 3701–3717. [[CrossRef](#)]
20. Jin, Z.; Zhao, J.; Chakrabarti, S.; Ding, L.; Terzija, V. A hybrid robust forecasting-aided state estimator considering bimodal Gaussian mixture measurement errors. *Int. J. Electr. Power Energy Syst.* **2020**, *120*, 105962. [[CrossRef](#)]
21. Walia, G.S.; Kumar, A.; Saxena, A. Robust object tracking with crow search optimized multi-cue particle filter. *Pattern Anal. Appl.* **2020**, *23*, 1439–1455. [[CrossRef](#)]
22. Jin, X.B.; Sun, S.L.; Wei, H.; Yang, F.B. Advances in multi-sensor information fusion: Theory and applications 2017. *Sensors* **2018**, *18*, 1162. [[CrossRef](#)]
23. Bai, Y.T.; Wang, X.Y.; Sun, Q. Spatio-temporal prediction for the monitoring-blind area of industrial atmosphere based on the fusion network. *Int. J. Environ. Res. Public Health* **2019**, *16*, 3788. [[CrossRef](#)]
24. Wang, L.; Zhang, T.; Wang, X.; Jin, X.; Xu, J.; Yu, J.; Zhang, H.; Zhao, Z. An approach of improved multivariate timing-random deep belief net modelling for algal bloom prediction. *Biosyst. Eng.* **2019**, *177*, 130–138. [[CrossRef](#)]
25. Hong, J.; Laflamme, S.; Dodson, J.; Joyce, B. Introduction to State Estimation of High-Rate System Dynamics. *Sensors* **2018**, *18*, 217. [[CrossRef](#)] [[PubMed](#)]
26. Dehghanpour, K.; Wang, Z.; Wang, J.; Yuan, Y.; Bu, F. A Survey on State Estimation Techniques and Challenges in Smart Distribution Systems. *IEEE Trans. Smart Grid* **2018**, *10*, 2312–2322. [[CrossRef](#)]
27. Jin, X.; Yin, G.; Chen, N. Advanced Estimation Techniques for Vehicle System Dynamic State: A Survey. *Sensors* **2019**, *19*, 4289. [[CrossRef](#)]
28. Jin, X.B.; Su, T.L.; Kong, J.L.; Bai, Y.T.; Miao, B.B.; Dou, C. State-of-the-art mobile intelligence: Enabling robots to move like humans by estimating mobility with artificial intelligence. *Appl. Sci.* **2018**, *8*, 379. [[CrossRef](#)]
29. Wang, Y.; Papageorgiou, M. Real-time freeway traffic state estimation based on extended Kalman filter: A general approach. *Transp. Res. Part B Methodol.* **2005**, *39*, 141–167. [[CrossRef](#)]
30. Yang, Y.; Liu, X.; Zhang, W.; Liu, X.; Guo, Y. A Nonlinear Double Model for Multisensor-Integrated Navigation Using the Federated EKF Algorithm for Small UAVs. *Sensors* **2020**, *20*, 2974. [[CrossRef](#)] [[PubMed](#)]
31. Du, H.; Wang, W.; Xu, C.; Xiao, R.; Sun, C. Real-Time Onboard 3D State Estimation of an Unmanned Aerial Vehicle in Multi-Environments Using Multi-Sensor Data Fusion. *Sensors* **2020**, *20*, 919. [[CrossRef](#)] [[PubMed](#)]
32. Julier, S.J. The scaled unscented transformation. In Proceedings of the American Control Conference, Anchorage, AK, USA, 8–10 May 2002; Volume 6, pp. 4555–4559.
33. Wang, F.; Su, T.; Jin, X.; Zheng, Y.; Kong, J.; Bai, Y. Indoor tracking by RFID fusion with IMU data. *Asian J. Control* **2019**, *21*, 1768–1777. [[CrossRef](#)]
34. Jin, X.B.; Dou, C.; Su, T.L.; Lian, X.F.; Shi, Y. Parallel Irregular Fusion Estimation Based on Nonlinear Filter for Indoor RFID Tracking System. *Int. J. Distrib. Sens. Netw.* **2016**. [[CrossRef](#)]
35. Luo, Z.; Fu, Z.; Xu, Q. An Adaptive Multi-Dimensional Vehicle Driving State Observer Based on Modified Sage-Husa UKF Algorithm. *Sensors* **2020**, *20*, 6889. [[CrossRef](#)]
36. Liu, J.; Wang, P.; Zha, F.; Guo, W.; Jiang, Z.; Sun, L. A Strong Tracking Mixed-Degree Cubature Kalman Filter Method and Its Application in a Quadrupe Robot. *Sensors* **2020**, *20*, 2251. [[CrossRef](#)]
37. Zhang, X.; Shen, Y. Distributed Kalman Filtering Based on the Non-Repeated Diffusion Strategy. *Sensors* **2020**, *20*, 6923. [[CrossRef](#)]
38. Zhao, Y.; Zhang, J.; Hu, G.; Zhong, Y. Set-Membership Based Hybrid Kalman Filter for Nonlinear State Estimation under Systematic Uncertainty. *Sensors* **2020**, *20*, 627. [[CrossRef](#)] [[PubMed](#)]
39. Nan, D.; Wang, W.; Wang, K.; Mahfoud, R.J.; Alhelou, H.H.; Siano, P. Dynamic State Estimation for Synchronous Machines Based on Adaptive Ensemble Square Root Kalman Filter. *Appl. Sci.* **2019**, *9*, 5200. [[CrossRef](#)]
40. Santos, N.P.; Lobo, V.; Bernardino, A. Unmanned Aerial Vehicle Tracking Using a Particle Filter Based Approach. In Proceedings of the IEEE International Underwater Technology Symposium, UT 2019—Proceedings, Kaohsiung, Taiwan, 16–19 April 2019; pp. 1–10.
41. Zhao, Z.; Yao, P.; Wang, X.; Xu, J.; Wang, L.; Yu, J. Reliable flight performance assessment of multirotor based on interacting multiple model particle filter and health degree. *Chin. J. Aeronaut.* **2019**, *32*, 444–453. [[CrossRef](#)]
42. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188. [[CrossRef](#)]
43. Leeuwen, P.J.V.; Künsch, H.R.; Nerger, L.; Potthast, R.; Reich, S. Particle filters for high-dimensional geoscience applications: A review. *Q. J. R. Meteorol. Soc.* **2019**, *145*, 2335–2365. [[CrossRef](#)] [[PubMed](#)]
44. Del Moral, P. Non Linear Filtering: Interacting Particle Solution. *Markov Process. Relat. Fields* **1996**, *2*, 555–580.
45. Stordal, A.S.; Karlsen, H.A.; Nævdal, G.; Skaug, H.J.; Vallès, B. Bridging the ensemble Kalman filter and particle filters: The adaptive Gaussian mixture filter. *Comput. Geosci.* **2011**, *15*, 293–305. [[CrossRef](#)]
46. Zhang, T.; Liu, S.; Xu, C.; Liu, B.; Yang, M.H. Correlation Particle Filter for Visual Tracking. *IEEE Trans. Image Process.* **2018**, *27*, 2676–2687. [[CrossRef](#)] [[PubMed](#)]

47. Jing, Y.; Chen, Y. Distributed Color-Based Particle Filter for Target Tracking in Camera Network. In Proceedings of the International Conference on Collaborative Computing: Networking, Applications and Worksharing, Shanghai, China, 16–18 October 2020; pp. 396–406.
48. Bilik, I.; Tabrikian, J. Maneuvering Target Tracking in the Presence of Glint using the Nonlinear Gaussian Mixture Kalman Filter. *IEEE Trans. Aerosp. Electron. Syst.* **2010**, *46*, 246–262. [[CrossRef](#)]
49. Bengua, J.A.; Tuan, H.D.; Duong, T.Q.; Poor, H.V. Joint Sensor and Relay Power Control in Tracking Gaussian Mixture Targets by Wireless Sensor Networks. *IEEE Trans. Signal Process.* **2018**, *66*, 492–506. [[CrossRef](#)]
50. Ding, F.; Wang, X.H.; Mao, L.; Xu, L. Joint state and multi-innovation parameter estimation for time-delay linear systems and its convergence based on the Kalman filtering. *Digit. Signal Process* **2017**, *62*, 211–223. [[CrossRef](#)]
51. Ding, F.; Xu, L.; Zhu, Q. Performance analysis of the generalised projection identification for time-varying systems. *IET Control Theory Appl.* **2016**, *10*, 2506–2514. [[CrossRef](#)]
52. Ding, F.; Xu, L.; Meng, D.D. Gradient estimation algorithms for the parameter identification of bilinear systems using the auxiliary model. *J. Comput. Appl. Math.* **2020**, *369*, 112575. [[CrossRef](#)]
53. Xu, L. The parameter estimation algorithms based on the dynamical response measurement data. *Adv. Mech. Eng.* **2017**, *9*, 1687814017730003. [[CrossRef](#)]
54. Pan, J.; Jiang, X.; Wan, X.K.; Ding, W. A filtering based multi-innovation extended stochastic gradient algorithm for multivariable control systems. *Int. J. Control Autom. Syst.* **2017**, *15*, 1189–1197. [[CrossRef](#)]
55. Zhang, X. Recursive parameter estimation and its convergence for bilinear systems. *IET Control Theory Appl.* **2020**, *14*, 677–688. [[CrossRef](#)]
56. Li, M.H.; Liu, X.M. The least squares based iterative algorithms for parameter estimation of a bilinear system with autoregressive noise using the data filtering technique. *Signal Process* **2018**, *147*, 23–34. [[CrossRef](#)]
57. Zhang, X.; Liu, Q.Y. Recursive identification of bilinear time-delay systems through the redundant rule. *J. Frankl. Inst.* **2020**, *257*, 726–747. [[CrossRef](#)]
58. Frezzatto, L.; de Oliveira, M.C.; Oliveira, R.C.; Peres, P.L. Robust H_∞ filter design with past output measurements for uncertain discrete-time systems. *Automatica* **2016**, *71*, 151–158. [[CrossRef](#)]
59. Dehghannasiri, R.; Esfahani, M.S.; Dougherty, E.R. Intrinsically Bayesian Robust Kalman Filter: An Innovation Process Approach. *IEEE Trans. Signal Process.* **2017**, *65*, 2531–2546. [[CrossRef](#)]
60. Nishanthi, D.; Banu, L.J.; Balasubramaniam, P. Robust guaranteed cost state estimation for discrete-time systems with random delays and random uncertainties. *Int. J. Adapt. Control Signal Process.* **2017**, *31*, 1361–1372. [[CrossRef](#)]
61. Roy, S.; Berry, D.W.; Petersen, I.R.; Huntington, E.H. Robust guaranteed-cost adaptive quantum phase estimation. *Phys. Rev. A* **2017**, *95*. [[CrossRef](#)]
62. Ding, D.; Wang, Z.; Dong, H.; Shu, H. Distributed H_∞ state estimation with stochastic parameters and nonlinearities through sensor networks: The finite-horizon case. *Automatica* **2012**, *48*, 1575–1585. [[CrossRef](#)]
63. Li, Z.; Chang, X. Robust H_∞ control for networked control systems with randomly occurring uncertainties: Observer-based case. *ISA Trans.* **2018**, *83*, 13–24. [[CrossRef](#)]
64. Pal, B.C.; Coonick, A.H.; Jaimoukha, I.M.; El-Zobaidi, H. A linear matrix inequality approach to robust damping control design in power systems with superconducting magnetic energy storage device. *IEEE Trans. Power Syst.* **2000**, *15*, 356–362. [[CrossRef](#)]
65. Liu, Y.; Fu, Y.; Lin, H.; Liu, J.; Gao, M.; He, Z. A New Constrained State Estimation Method Based on Unscented H_∞ Filtering. *Appl. Sci.* **2020**, *10*, 8484. [[CrossRef](#)]
66. Daeipour, E.; Bar-Shalom, Y. IMM tracking of maneuvering targets in the presence of glint. *IEEE Trans. Aerosp. Electron. Syst.* **1998**, *34*, 996–1003. [[CrossRef](#)]
67. Xu, L.; Li, X.R.; Duan, Z. Hybrid grid multiple-model estimation with application to maneuvering target tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 122–136. [[CrossRef](#)]
68. Jin, X.B.; Lian, X.F.; Su, T.L.; Shi, Y.; Miao, B.B. Closed-Loop Estimation for Randomly Sampled Measurements in Target Tracking System. *Math. Probl. Eng.* **2014**, *2014*, 315908.
69. Ouyang, W.; Wu, Y.; Member, S.; Chen, H. INS/Odometer Land Navigation by Accurate Measurement Modeling and Multiple-Model Adaptive Estimation. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *57*, 245–262. [[CrossRef](#)]
70. Bai, Y.; Wang, X.; Jin, X.; Su, T.; Kong, J. Adaptive filtering for MEMS gyroscope with dynamic noise model. *ISA Trans.* **2020**, *101*, 430–441. [[CrossRef](#)] [[PubMed](#)]
71. Xu, L.; Ding, F.; Zhu, Q.M. Hierarchical Newton and least squares iterative estimation algorithm for dynamic systems by transfer functions based on the impulse responses. *Int. J. Syst. Sci.* **2019**, *50*, 141–151. [[CrossRef](#)]
72. Xu, L. Iterative parameter estimation for signal models based on measured data. *Circuits Syst. Signal Process* **2018**, *37*, 3046–3069. [[CrossRef](#)]
73. Gu, Y.; Liu, J.; Li, X.; Chou, Y.; Ji, Y. State space model identification of multirate processes with time-delay using the expectation maximization. *J. Frankl. Inst.* **2019**, *356*, 1623–1639. [[CrossRef](#)]
74. Xu, L.; Xiong, W.; Alsaedi, A.; Hayat, T. Hierarchical Parameter Estimation for the Frequency Response Based on the Dynamical Window Data. *Int. J. Control Autom. Syst.* **2018**, *16*, 1756–1764. [[CrossRef](#)]
75. Xu, L.; Ding, F.; Lu, X.; Wan, L.; Sheng, J. Hierarchical multi-innovation generalised extended stochastic gradient methods for multivariable equation-error autoregressive moving average systems. *IET Control Theory Appl.* **2020**, *14*, 1276–1286. [[CrossRef](#)]

76. Pan, J.B.; Ma, H.; Zhang, X.; Liu, Q.Y. Recursive coupled projection algorithms for multivariable output-error-like systems with coloured noises. *IET Signal Process* **2020**, *14*, 455–466. [[CrossRef](#)]
77. Xu, L.; Ding, F.; Wan, L.; Sheng, J. Separable multi-innovation stochastic gradient estimation algorithm for the nonlinear dynamic responses of systems. *Int. J. Adapt. Control Signal Process.* **2020**, *34*, 937–954. [[CrossRef](#)]
78. Zhang, X.; Ding, F.; Alsaadi, F.E.; Hayat, T. Recursive parameter identification of the dynamical models for bilinear state space systems. *Nonlinear Dyn.* **2017**, *89*, 2415–2429. [[CrossRef](#)]
79. Zhang, X.; Xu, L.; Ding, F.; Hayat, T. Combined state and parameter estimation for a bilinear state space system with moving average noise. *J. Frankl. Inst.* **2018**, *355*, 3079–3103. [[CrossRef](#)]
80. Gu, Y.; Zhu, Q.; Nouri, H. Bias compensation-based parameter and state estimation for a class of time-delay non-linear state-space models. *IET Control Theory Appl.* **2020**, *14*, 2176–2185. [[CrossRef](#)]
81. Jin, X.B.; Du, J.J.; Bao, J. Target Tracking of a Linear Time Invariant System under Irregular Sampling. *Int. J. Adv. Robot. Syst.* **2012**, *9*, 219.
82. Zhang, X.; Ding, F.; Xu, L.; Yang, E.F. State filtering-based least squares parameter estimation for bilinear systems using the hierarchical identification principle. *IET Control Theory Appl.* **2018**, *12*, 1704–1713. [[CrossRef](#)]
83. Wang, L.; Ji, Y.; Wan, L.; Bu, N. Hierarchical recursive generalized extended least squares estimation algorithms for a class of nonlinear stochastic systems with colored noise. *J. Frankl. Inst.* **2019**, *356*, 10102–10122. [[CrossRef](#)]
84. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
85. Jin, X.B.; Yang, N.X.; Wang, X.Y.; Bai, Y.T.; Su, T.L.; Kong, J.L. Integrated predictor based on decomposition mechanism for PM2.5 long-term prediction. *Appl. Sci.* **2019**, *9*, 4533. [[CrossRef](#)]
86. Jin, X.B.; Zheng, W.Z.; Kong, J.L.; Wang, X.Y.; Bai, Y.L.; Su, T.L.; Lin, S. Deep-Learning Forecasting Method for Electric Power Load via Attention-Based Encoder-Decoder with Bayesian Optimization. *Energies* **2021**, *14*, 1596. [[CrossRef](#)]
87. Sutskever, I.; Vinyals, O.; Le, Q. Sequence to Sequence Learning with Neural Networks. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 3104–3112.
88. Chorowski, J.K.; Bahdanau, D.; Serdyuk, D.; Cho, K.; Bengio, Y. Attention-based models for speech recognition. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 577–585.
89. Pham Luong, M.T.; Manning, C.H. Effective Approaches to Attention-based Neural Machine Translation. *arXiv* **2015**, arXiv:1508.04025.
90. Wang, Y.; Liu, Z.; Hu, D. Multivariate Time Series Prediction Based on Optimized Temporal Convolutional Networks with Stacked Auto-encoders. *Mach. Learn.* **2019**, *2019*, 157–172.
91. Jin, X.B.; Wang, H.X.; Wang, X.Y.; Bai, Y.T.; Su, T.L.; Kong, J.L. Deep-Learning Prediction Model with Serial Two-Level Decomposition Based on Bayesian Optimization. *Complexity* **2020**, *2020*, 4346803. [[CrossRef](#)]
92. Jin, X.B.; Yang, N.X.; Wang, X.; Bai, Y.; Su, T.L.; Kong, J. Deep Hybrid Model Based on EMD with Classification by Frequency Characteristics for Long-Term Air Quality Prediction. *Mathematics* **2020**, *8*, 214. [[CrossRef](#)]
93. Jin, X.B.; Yu, X.H.; Wang, X.Y.; Bai, Y.T.; Su, T.L.; Kong, J.L. Deep Learning Predictor for Sustainable Precision Agriculture Based on Internet of Things System. *Sustainability* **2020**, *12*, 1433. [[CrossRef](#)]
94. Niu, X.; Li, J.; Sun, J. Dynamic Detection of False Data Injection Attack in Smart Grid using Deep Learning. In Proceedings of the 2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 18–21 February 2019; pp. 1–6.
95. Zhang, C.; Zhang, H.; Qiao, J.; Yuan, D.; Zhang, M. Deep Transfer Learning for Intelligent Cellular Traffic Prediction Based on Cross-Domain Big Data. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1389–1401. [[CrossRef](#)]
96. Al-Sharman, M.; Murdoch, D.; Cao, D.; Lv, C.; Zweiri, Y.; Rayside, D.; Melek, W. A Sensorless State Estimation for A Safety-Oriented Cyber-Physical System in Urban Driving: Deep Learning Approach. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 169–178. [[CrossRef](#)]
97. Jin, X.B.; Yang, N.X.; Wang, X.Y.; Bai, Y.T.; Su, T.L.; Kong, J.L. Hybrid Deep Learning Predictor for Smart Agriculture Sensing Based on Empirical Mode Decomposition and Gated Recurrent Unit Group Model. *Sensors* **2020**, *20*, 1334. [[CrossRef](#)] [[PubMed](#)]
98. Jin, X.B.; Yu, X.H.; Su, T.L.; Yang, D.N.; Bai, Y.T.; Kong, J.L.; Wang, L. Distributed Deep Fusion Predictor for a Multi-Sensor System Based on Causality Entropy. *Entropy* **2021**, *23*, 219. [[CrossRef](#)]
99. Zhang, K.; Zheng, L.; Liu, Z.; Jia, N. A deep learning based multitask model for network-wide traffic speed prediction. *Neurocomputing* **2020**, *396*, 438–450. [[CrossRef](#)]
100. Mestav, K.R.; Luengo-Rozas, J.; Tong, L. Bayesian State Estimation for Unobservable Distribution Systems via Deep Learning. *IEEE Trans. Power Syst.* **2019**, *34*, 4910–4920. [[CrossRef](#)]
101. Mestav, K.R.; Tong, L. Learning the Unobservable: High-Resolution State Estimation via Deep Learning. In Proceedings of the 2019 57th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 24–27 September 2019; pp. 171–176.
102. Wang, L.; Zhang, T.; Jin, X.; Xu, J.; Wang, X.; Zhang, H.; Yu, J.; Sun, Q.; Zhao, Z.; Xie, Y. An approach of recursive timing deep belief network for algal bloom forecasting. *Neural Comput. Appl.* **2020**, *32*, 163–171. [[CrossRef](#)]
103. Yu, W.; De, E. Deep Boltzmann machine for nonlinear system modelling. *Int. J. Mach. Learn. Cybern.* **2018**, *10*, 1705–1716. [[CrossRef](#)]
104. Shi, Z.; Bai, Y.; Jin, X.; Wang, X.; Su, T.; Kong, J. Parallel deep prediction with covariance intersection fusion on non-stationary time series. *Knowl. Based Syst.* **2021**, *2021*, 106523. [[CrossRef](#)]

105. Bai, Y.; Jin, X.; Wang, X.; Wang, X.; Xu, J. Dynamic correlation analysis method of air pollutants in spatio-temporal analysis. *Int. J. Environ. Res. Public Health* **2020**, *17*, 360. [[CrossRef](#)] [[PubMed](#)]
106. Bergstra, J.S.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for hyper-parameter optimization. *Adv. Neural Inf. Process. Syst.* **2011**, *24*, 2546–2554.
107. Zamzam, A.S.; Fu, X.; Sidiropoulos, N.D. Data-Driven Learning-Based Optimization for Distribution System State Estimation. *IEEE Trans. Power Syst.* **2019**, *34*, 4796–4805. [[CrossRef](#)]
108. Wu, J.; Li, Y.; Quevedo, D.E.; Lau, V.; Shi, L. Data-driven power control for state estimation: A Bayesian inference approach. *Automatica* **2015**, *54*, 332–339. [[CrossRef](#)]
109. Liu, J.; Wang, Z.; Xu, M. DeepMTT: A deep learning maneuvering target-tracking algorithm based on bidirectional LSTM network. *Inf. Fusion* **2020**, *53*, 289–304. [[CrossRef](#)]
110. Shaukat, N.; Ali, A.; Iqbal, M.J.; Moinuddin, M.; Otero, P. Multi-Sensor Fusion for Underwater Vehicle Localization by Augmentation of RBF Neural Network and Error-State Kalman Filter. *Sensors* **2021**, *21*, 1149. [[CrossRef](#)] [[PubMed](#)]
111. Khuntia, P.; Hazra, R. An efficient Deep reinforcement learning with extended Kalman filter for device-to-device communication underlying cellular network. *Trans. Emerg. Telecommun. Technol.* **2019**, *30*, e3671. [[CrossRef](#)]
112. Zhang, L.; Mao, D.; Niu, J.; Wu, Q.M.; Ji, Y. Continuous tracking of targets for stereoscopic HFSWR based on IMM filtering combined with ELM. *Remote Sens.* **2020**, *12*, 272. [[CrossRef](#)]
113. Zhao, C.; Sun, L.; Yan, Z.; Neumann, G.; Duckett, T.; Stolkin, R. Learning Kalman Network: A deep monocular visual odometry for on-road driving. *Robot. Auton. Syst.* **2019**, *121*, 103234. [[CrossRef](#)]
114. Li, Q.; Wu, Z.Y.; Rahman, A. Evolutionary Deep Learning with Extended Kalman Filter for Effective Prediction Modeling and Efficient Data Assimilation. *J. Comput. Civ. Eng.* **2019**, *33*, 04019014. [[CrossRef](#)]
115. Gao, C.; Yan, J.; Zhou, S.; Varshney, P.K.; Liu, H. Long short-term memory-based deep recurrent neural networks for target tracking. *Inf. Sci.* **2019**, *502*, 279–296. [[CrossRef](#)]
116. Bai, Y.; Wang, X.; Jin, X.; Zhao, Z.; Zhang, B. A neuron-based Kalman filter with nonlinear auto-regressive model. *Sensors* **2020**, *20*, 299. [[CrossRef](#)] [[PubMed](#)]
117. Sharman, M.K.S.A. Deep Learning-Based Neural Network Training for State Estimation Enhancement: Application to Attitude Estimation. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 24–34. [[CrossRef](#)]
118. Yu, Y.; Liu, Q.; Chambon, S.; Hamzah, M. Using deep Kalman filter to predict drilling time series. In Proceedings of the International Petroleum Technology Conference, Beijing, China, 26–28 March 2019. [[CrossRef](#)]