Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 54-20 (2021) 366–373

Reinforced Optimal Estimator [★]

Wenhan Cao [*] **Jianyu Chen** [**] **Jingliang Duan** [*]
Shengbo Eben Li [*] **Yao Lyu** [*] **Ziqing Gu** [*] **Yuhang Zhang** [*]

[*] *School of Vehicle and Mobility, Tsinghua University, Beijing, China(e-mail: lisb04@gmail.com)*
[**] *Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China*

Abstract: Estimating the state of a stochastic system is a long-lasting issue in the areas of engineering and science. Existing methods either use approximations or yield a high computation burden. In this paper, we propose reinforced optimal estimator (ROE), which is an offline estimator for general nonlinear and non-Gaussian stochastic models. This method solves optimal estimation problems offline, and the learned estimator can be applied online efficiently. Firstly, we demonstrate that minimum variance estimation requires us to solve the estimation problem online, which causes low computation efficiency. To overcome this drawback, we propose an infinite horizon optimal estimation problem, called reinforcement estimation problem, to obtain the offline estimator. The time-invariant filter of linear systems is shown as an example to analyze the equivalence between reinforcement estimation problem and minimum variance estimation problem. We show that such equivalence can only be found for linear systems, and the proposed problem formulation actually enables us to find the time-invariant estimator for general nonlinear systems. Then, we propose the ROE algorithm, inspired by reinforcement learning, and develop an actor-critic architecture to find a nearly optimal estimator of the reinforcement estimation problem. The estimator is approximated by recurrent neural networks, which has high online computation efficiency. The convergence is proved using contraction mapping and extended policy improvement theorem. Experiment results on complex nonlinear system estimation problems show that our method achieves higher estimation accuracy and computation efficiency than the unscented Kalman filter and particle filter.

Keywords: State estimation, reinforcement learning, minimum variance estimation, time-invariant filter, recurrent neural network.

1. INTRODUCTION

In recent years, state estimation (also known as filtering) of dynamic systems draws much attention in different domains such as signal processing, robotics as well as econometrics (Musoff and Zarchan (2009)). Statistical techniques for state estimation such as Bayesian filtering provide a natural way to tackle difficult issues for robotics such as simultaneous localization and mapping (Thrun (2002)). Such filtering methods can also be used to fuse the redundant and complementary sensor data to enhance the perception system's reliability and capability (Sun (2004)). For example, it can be used to increase the position accuracy of the Global Positioning System and the Inertial Navigation System.

State estimation for linear stochastic systems has actually been elegantly solved with convergence guarantee since the prominent Kalman filter (KF) was proposed in the late 1950s (Kalman (1960)). KF is the optimal filter for linear systems with Gaussian noises in the sense of minimum variance estimation (MVE). It can also be considered optimal regardless of the probability distribution function of the noise in the sense of linear minimum-variance estimation (Simon (2006)).

However, a closed form estimator like KF does not exist for nonlinear stochastic system because the probability distribution of state does not preserve the property of Gaussian. As a result, nonlinear filtering is far more challenging. Extended Kalman filter (EKF) introduces first-order Taylor series expansion of the nonlinear stochastic state space models and derives a suboptimal filter (Smith et al. (1962)). Instead of directly linearizing the models, Unscented Kalman filter (UKF) uses a set of sigma points to parameterize the mean and covariance of the posterior distribution and can be comparable to the second-order Gauss filter (Julier and Uhlmann (1997)). Unfortunately, both methods are only valid in high signal-to-noise ratio situations and are not applicable to highly nonlinear systems.

Some more accurate estimators which better approximate model non-linearities and non-Gaussian noises have been proposed. Gaussian sum filter (GSF) fuses multiple EKFs to deal with a highly nonlinear and noisy environment (Alspach and Sorenson (1972)). Although a weighted sum

[★] This work is supported by International Science & Technology Cooperation Program of China under 2019YFE0100200, NSF China with 51575293, and U20A20334. It is also partially supported by Geekplus Technology Co., Ltd. All correspondence should be sent to S. Eben Li with lisb04@gmail.com.

10.1016/j.ifacol.2021.11.201

of Gaussian probability density functions has been proved to be able to approximate arbitrarily close to another density function, the computational overhead of the GSF can be significantly large and it relies on some adhoc rules to keep consistent with its theoretical properties. Particle filter (PF) uses sequential Monte Carlo methods which sample a set of particles to approximate the posterior distribution (Liu and Chen (1998)). Although PF has proved convergence to the true posterior distribution with the increasing number of particles, it requires a lot of computation resources which hinders its online applications.

This paper proposes an offline optimal estimation problem, called Reinforcement Estimation Problem (REP), to find an estimator with the minimized cumulative square estimate error. Then we design the ROE algorithm which is inspired by reinforcement learning and develop an actor-critic architecture to find the optimal solution of the REP. The main contributions of this paper are summarized as follows:

(1) We point out that the problem formulation of MVE is a one-step horizon estimation problem and requires us to solve it online, causing low computation efficiency. To overcome this drawback, we propose an infinite horizon estimation problem called REP. This problem formulation enables us to obtain the offline estimator. We take the time-invariant filter of linear systems as an example to analyze the relationship between REP and MVE. We illustrate that the gain matrix of the time-invariant filter can be seen as the stationary policy of the REP. Such a special form can only be found in linear systems because the Markov property of the estimate error only holds when the system and the estimator are linear. Thus, the problem formulation of the REP enables us to find the time-invariant estimator for general nonlinear systems.
(2) We stress that KF has the special recurrent form in which the current estimate can be calculated only by the last estimate and the innovation. This form is optimal due to the property of linear systems and Gaussian noises. However, such an optimal analytical form does not exist for nonlinear systems. This paper finds the recurrent form for general nonlinear estimators. The proposed form employs the hidden state of recurrent neural network (RNN) to encode the historical information, which ensures the optimality of the estimator due to the universal approximation ability of RNN (Hammer (2000); Schäfer and Zimmermann (2006)).
(3) We utilize contraction mapping and extended policy improvement theorem to prove the convergence of the ROE algorithm. Compared with EKF and UKF, ROE can be applied to complex nonlinear systems and non-Gaussian noises. Besides, unlike existing methods such as PF and GSF that require huge online computation resources, ROE has high computation efficiency since it is an offline estimator represented by neural networks.

The remainder of the paper is organized as follows. Section 2 describes REP based on the general discrete-time stochastic model. Section 3 illustrates the relationship between REP and MVE. The ROE algorithm is proposed in section 4. The convergence of ROE algorithm is proved in section 5. Experiments are described in section 6. Section 7 makes a conclusion in the end.

2. PROBLEM FORMULATION

2.1 Preliminaries

Consider the following system with process noise and measurement noise:

$$\begin{aligned} x_{t+1} &= f(x_t) + \xi_t \\ y_t &= g(x_t) + \zeta_t. \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the state, $y \in \mathbb{R}^m$ is the observation. $f(\cdot)$ and $g(\cdot)$ can be arbitrary time-invariant functions. $\xi \in \mathbb{R}^n$ is the process noise and $\zeta \in \mathbb{R}^m$ is the measurement noise. As for the general state estimation problem, the stochastic system has the following assumptions (Anderson and Moore (2012)):

(1) $\{\xi_t, t \geq 0\}$ is independent and identically distributed (iid) and the distribution is accessible:
$$\xi_t \sim p_\xi(\xi_t).$$
(2) $\{\zeta_t, t \geq 0\}$ is iid and the distribution is accessible:
$$\zeta_t \sim p_\zeta(\zeta_t).$$
(3) $\{\xi_t, t \geq 0\}$ and $\{\zeta_t, t \geq 0\}$ are independent with each other.
(4) The distribution of the initial state is independent to $\{\xi_t, t \geq 0\}$ and $\{\zeta_t, t \geq 0\}$.

Denote the estimate of x_t as $\hat{x}_t$, then the estimation error is $e_t = x_t - \hat{x}_t$. We consider the following generic form of state estimator:

$$\hat{x}_t = \varphi(h_t),$$

where $h_t = (\hat{x}_0, y_1, \hat{x}_1, y_2, \dots, \hat{x}_{t-1}, y_t)$ is the history information containing all past estimations and observations. To estimate the true state, existing filtering algorithms, such as KF and PF, aims to find the estimator that minimizes the square estimation error:

$$\varphi^*(h_t) = \underset{\hat{x}_t = \varphi(h_t)}{\arg\min}\, \mathbb{E}\left\{\|e_t\|_2^2 | h_t\right\}. \tag{2}$$

Remark 1. Such problem formulation requires us to calculate the posterior distribution of Bayesian filtering online. However, for complex nonlinear systems, this process is often intractable with low computation efficiency. Actually, state estimation or filtering means the recovery from $y(\cdot)$ of $x(\cdot)$ or even some information about $x(\cdot)$ (Anderson and Moore (2012)). In this sense, MVE is just one type of estimation criterion. It's reasonable for us to find other estimation criterions.

2.2 Reinforcement Estimation Problem

RL has received remarkable success in a wide variety of challenging control problems (Duan et al. (2020); Guan et al. (2019)). RL can learn a nearly optimal policy offline and the learned policy can be directly used for online application, leading to high computation efficiency (Li (2020)). It is known that optimal estimation and optimal control are dual problems in linear Gaussian settings. Inspired by this duality, if we can reframe the nonlinear optimal estimation problem as an optimal control problem and then use RL's techniques to solve it, we might be

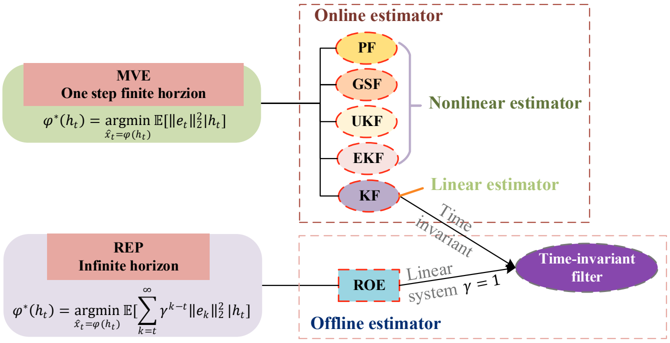

Online estimator
MVE
One step finite horzion
$\varphi^*(h_t) = \operatorname{argmin}_{\hat{x}_t=\varphi(h_t)} \mathbb{E}[\|e_t\|_2^2|h_t]$
PF
GSF
UKF
EKF
KF
Nonlinear estimator
Linear estimator
Time invariant
REP
Infinite horizon
$\varphi^*(h_t) = \operatorname{argmin}_{\hat{x}_t=\varphi(h_t)} \mathbb{E}[\sum_{k=t}^{\infty} \gamma^{k-t}\|e_k\|_2^2 |h_t]$
ROE
Linear system $\gamma = 1$
Time-invariant filter
Offline estimator

Fig. 1. Problem formulation of MVE and REP.

able to obtain the optimal offline estimator for general nonlinear stochastic state space models.

RL was first utilized to solve the estimation problem in 2007 and achieved great performance (Morimoto and Doya (2007)). However, its cost function designed is actually only suitable for zero-mean Gaussian noise. Recently, the RL-based state estimator with bounded estimate error performance guarantee is proposed (Hu et al. (2020)). This work assumes that the estimate error dynamics can be described by a Markov decision process. However, the error dynamics generated by the nonlinear filter actually does not satisfy the Markov property. Besides, the structure of the estimator is predefined as the combination of the last step's estimate and pseudo-innovations, which loses the guarantee of optimality.

To deal with all these issues, our work formulates a new offline estimation problem called REP. In contrast to the previous work, our work finds the true state of Markov decision process and has no prior assumptions on the estimator's structure.

Inspired by RL, the value function of the offline state estimation problem is defined as:

$$V^{\varphi}(h_t) = \mathop{\mathbb{E}}_{\hat{x}_t=\varphi(h_t)} \left\{ \sum_{k=t}^{\infty} \gamma^{k-t} \|e_k\|_2^2 \mid h_t \right\},$$

where $\gamma \in (0, 1]$ is discount factor.

This paper aims to find an optimal estimator φ^* with the minimized value function $V^*(h_t)$, i.e.,

$$\varphi^*(h_t) = \mathop{\arg\min}_{\hat{x}_t=\varphi(h_t)} V^{\varphi}(h_t). \tag{3}$$

We refer to problem (3) for systems in (1) as REP. $\gamma \to 0$ means the REP is "myopic" and concerns only immediate error. As $\gamma \to 1$, it becomes more "farsighted". The relationship between REP and MVE is shown in Fig. 1. Considering the standard RL setting, an agent interacts with the environment $\mathcal{E}$ which can be modeled as Markov decision process. It is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{C}, \mathcal{P})$. In REP, the state is defined as $s_t = h_t \in \mathcal{S}$. Action is described as $a_t = \hat{x}_t \in \mathcal{A}$, which is selected according to the estimator (or policy) $\varphi(h_t)$, i.e., $\hat{x}_t = \varphi(h_t)$. The $\mathbb{E}\left\{\|e_t\|_2^2 \mid h_t\right\}$ can be seen as the cost function $l_t \sim l(s_t, a_t)$, i.e., $l_t = l(h_t, \hat{x}_t) = \mathbb{E}\left\{\|e_t\|_2^2 \mid h_t\right\}$. Besides, $P(h_{t+1}|h_t, \hat{x}_t)$ maps a given state-action tuple (s_t, a_t) to the probability distribution over s_{t+1}. Therefore, the Bellman equation of the REP can be written as

$$V^*(h_t) = \min_{\hat{x}_t=\varphi(h_t)} \Bigg\{ \mathbb{E}\left\{\|e_t\|_2^2 \mid h_t\right\} + \gamma \mathop{\mathbb{E}}_{h_{t+1}\sim p(h_{t+1}|h_t,\hat{x}_t)} \{V^*(h_{t+1})\} \Bigg\}.$$

The optimal value function and estimator can be found by solving the Bellman equation.

Remark 2. In keeping with the terminologies in the control community, we use "minimize the cost", which is equal to "maximize the reward" as used in the RL community.

3. RELATIONSHIP BETWEEN REP AND MVE

Section 2.2 transforms the traditional state estimation problem (2) into REP (3). To demonstrate the correctness of this transformation, this section takes the time-invariant filter of linear systems as an example to analyze the relationship between REP and MVE.

3.1 Time-invariant solution of MVE

First, consider the following linear system

$$\begin{aligned} x_{t+1} &= Fx_t + G\xi_t \\ y_t &= H^\top x_t + \zeta_t, \end{aligned} \tag{4}$$

where $\mathbb{E}[\xi_t] = 0$, $\mathbb{E}[\zeta_t] = 0$, $\mathbb{E}[\xi_t\xi_t^\top] = Q$ and $\mathbb{E}[\zeta_t\zeta_t^\top] = R$. A standard and widely used estimation algorithm for linear system is the well-known KF. KF is the optimal filter for linear systems with Gaussian noises in the sense of MVE. In KF, the state estimator for systems in (4) is given as follows:

$$\hat{x}_{t+1} = F\hat{x}_t + K_t(y_{t+1} - H^\top F\hat{x}_t), \tag{5}$$

where $K_t = P_tH(H^\top P_tH + R)^{-1}$. The predicted error covariance matrix P_t can be solved via Riccati equation

$$P_t = FP_{t-1}F^\top - FP_{t-1}H(H^\top P_{t-1}H + R)^{-1}H^\top P_{t-1}F^\top + GQG^\top.$$

Suppose the cholesky decomposition of Q is $Q = G_1G_1^\top$. If (F, H) is completely detectable and (F, GG_1) is completely stabilizable, the matrix P_t would converge to $\bar{P}$, which is the solution of the discrete-time algebraic Riccati equation (DARE)

$$\bar{P} = F\bar{P}F^\top - F\bar{P}H(H^\top\bar{P}H + R)^{-1}H^\top\bar{P}F^\top + GQG^\top.$$

For the time-invariant filter, the gain matrix K_t is invariant, i.e.,

$$\bar{K} = \bar{P}H(H^\top\bar{P}H + R)^{-1}. \tag{6}$$

According to (5), the time-invariant filer can be designed in the following form

$$\hat{x}_{t+1} = F\hat{x}_t + \bar{K}(y_{t+1} - H^\top F\hat{x}_t). \tag{7}$$

The time-invariant filter is obviously not the optimal filter, but $\bar{K}$ is optimal among all the fixed gains. However, it can be proved that the limit of the error covariance matrix is the same as that of the KF (Sinopoli et al. (2004)). Suppose that when $t > t_{\text{stationary}}$, the system come to the stationary distribution and $\bar{K}$ satisfies

$$\begin{aligned} \bar{K} = &\mathop{\arg\min}_{K} \mathbb{E}\left\{\|x_t - \hat{x}_t\|_2^2\right\} \\ &\text{s.t.} \quad t > t_{\text{stationary}} \end{aligned} \tag{8}$$

3.2 Time-invariant solution of REP

Assuming that the time-invariant solution of REP has a similar form with (7), one has

$$\hat{x}_{t+1} = \varphi(h_{t+1};\eta) = F\hat{x}_t + \eta(y_{t+1} - H^\top F\hat{x}_t), \tag{9}$$

where η is the parameter to be solved. According to (3), the goal of REP is to find the optimal η^* such that

$$\eta^* = \arg\min_{\eta} \mathbb{E}\left\{\sum_{k=t}^{\infty} \gamma^{k-t}\|e_k\|_2^2 \mid h_t\right\}. \tag{10}$$

When $\gamma = 1$, problem (10) is equivalent to minimizing the average-cost, i.e.,

$$\begin{aligned}\eta^* &= \arg\min_{\eta} \mathbb{E}\left\{\sum_{k=t}^{\infty} \|e_k\|_2^2 \mid h_t\right\}\\ &= \arg\min_{\eta} \lim_{T\to\infty}\frac{1}{T}\mathbb{E}\left\{\sum_{k=t}^{\mathrm{T}} \|e_k\|_2^2 \mid h_t\right\}\\ &= \arg\min_{\eta} \lim_{T\to\infty}\mathbb{E}\left\{\frac{1}{T}\sum_{k=t}^{\mathrm{T}} \|e_k\|_2^2\right\}\\ &= \arg\min_{\eta} \lim_{T\to\infty}\mathbb{E}\left\{\mathop{\mathbb{E}}_{e_k\sim d(e_k)}\left\{\|e_k\|_2^2\right\}\right\}\\ &= \arg\min_{\eta} \mathop{\mathbb{E}}_{e_k\sim d(e_k)}\left\{\|e_k\|_2^2\right\},\end{aligned} \tag{11}$$

where $d(e_k)$ is the stationary distribution of estimation errors. The key point of (11) is that e_t obeys Markov property when the system and estimator are both linear, i.e.,

$$e_{t+1} = (I - \eta H^\top)Fe_t + (I - \eta H^\top)G\xi_t - \eta\zeta_{t+1}. \tag{12}$$

From (8) and (11), it is clear that $\eta^* = \bar{K}$, which means that REP and MVE lead to a same time-invariant filer for linear systems when $\gamma = 1$. Based on this finding, if we do not limit the estimator $\varphi(h_t;\eta)$ to the form shown in (9) and assume that it can be characterized as an arbitrarily complex function, we can further derive that

$$\min_{\varphi}\lim_{T\to\infty}\mathbb{E}\left\{\frac{1}{T}\sum_{k=t}^{\mathrm{T}}\|e_k\|_2^2\right\} \le \lim_{T\to\infty}\mathbb{E}\left\{\frac{1}{T}\sum_{k=t}^{\mathrm{T}}\|e_k\|_2^2\right\}\Big|_{\bar{K}=\bar{P}H(H^\top\bar{P}H+R)^{-1}}.$$

This indicates that the optimal filter obtained by REP is at least not worse than the time-invariant filter.

Remark 3. More generally, average-cost problems ($\gamma = 1$) are ill-conditioned, which usually need a few strong assumptions on environment models and cost functions to find optimal policies. For practical applications, the vanishing discount factor $\gamma \in (0, 1)$ is a widely used alternating procedure. In fact, if $\gamma \to 1$, the discounted cost is approximately linear to an average cost (Sutton and Barto (1998)).

Remark 4. This special equivalence $\eta^* = \bar{K}$ can only be found in linear systems because e_t obeys Markov property in this special case. Thus, the problem formulation of the REP actually enables us to extend the time-invariant filter to the general nonlinear systems.

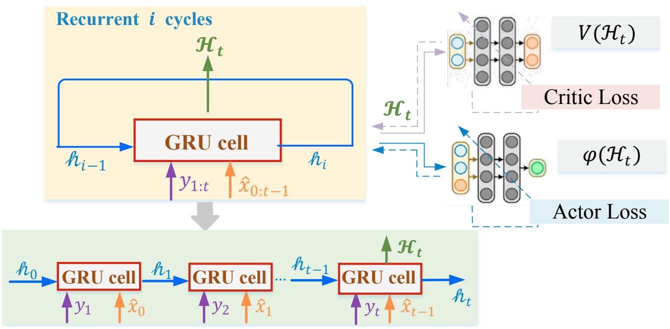


Fig. 2. Gradient flow of GRU-based value function and policy. The gradient $\frac{\partial V}{\partial \omega}$ or $\frac{\partial \varphi}{\partial \eta}$is calculated via backpropagation through time (BPTT) which is a gradient-based technique for training certain types of RNN.

4. REINFORCED OPTIMAL ESTIMATOR

This section proposes the ROE algorithm to find the nearly optimal solution of the REP for general nonlinear systems.

4.1 Policy Iteration for REP

Existing RL algorithms usually employ the policy iteration techniques to find the nearly optimal policy and value function. Policy iteration involves two revolving iteration procedures: 1) policy evaluation (PEV) and 2) policy improvement (PIM).

For the REP, given an estimator φ, PEV seeks to numerically solve its corresponding value function $V^\varphi(h_t)$ via the self-consistency condition:

$$V^\varphi_{k+1}(h_t) = l(h_t, \varphi(h_t)) + \gamma \mathop{\mathbb{E}}_{h_{t+1}\sim p(h_{t+1}|h_t,\varphi(h_t))}\left\{V^\varphi_k(h_{t+1})\right\}. \tag{13}$$

PIM aims to search for a better estimator by minimizing the current value function:

$$\varphi^{k+1}(h_t) = \arg\min_{\varphi}\Big\{l(h_t,\varphi(h_t)) + \gamma \mathop{\mathbb{E}}_{h_{t+1}\sim p(h_{h+1}|h_t,\varphi(h_t))}\left\{V^k(h_{t+1})\right\}\Big\}. \tag{14}$$

Here, V^k is the value function of the estimator φ_k.

4.2 Algorithm

To learn an analytic estimator, both value function and estimator should be approximated using parameterized functions. In this paper, we employ RNN to represent the value function and estimator, called value network and estimator network, due to its strong ability to fit complex nonlinear functions and handle sequential inputs (Elman (1990)). RNNs are now broadly used in various fields such as language modeling, machine translation as well as speech recognition (Lipton et al. (2015)). It is proved that the hidden state of RNN can store the information of previous inputs (Allen-Zhu et al. (2018)).

The value network is also called the critic, denoted as

$$V(h_t) \cong V(h_t;\omega), \tag{15}$$

where ω is the parameter. According to (13), the value network can be updated by directly minimizing the following critic loss:

$$J_{\text{critic}} = \mathop{\mathbb{E}}_{\substack{h_t\sim d(h_t),\\ h_{t+1}\sim d(h_{t+1})}} \left\{\frac{1}{2}[l_t + V(h_{t+1};\omega) - V(h_t;\omega)]^2\right\}.$$

where $h_t \sim d(h_t)$ is the stationary distribution derived by the current estimator. Based on the semi-gradient trick, the update gradient for the value network can be derived as

$$\frac{\partial J_{\text{critic}}}{\partial \omega} = - \mathop{\mathbb{E}}_{\substack{h_t\sim d(h_t),\\ h_{t+1}\sim d(h_{t+1})}} \Big\{[l_t + \gamma V(h_{t+1};\omega) - V(h_t;\omega)]\frac{\partial V(h_t;\omega)}{\partial \omega}\Big\}. \tag{16}$$

The estimator network is also called the actor, denoted as

$$\varphi(h_t) \cong \varphi(h_t;\eta), \tag{17}$$

where η is the parameter. According to (14), a better estimator network can be obtained by minimizing the following actor loss:

$$J_{\text{actor}} = \mathop{\mathbb{E}}_{h_t\sim d(h_t), h_{t+1}\sim d(h_{t+1})} \{l_t + \gamma V(h_{t+1};\omega)\}.$$

Then, the update gradient for the estimator can be expressed as

$$\frac{\partial J_{\text{actor}}}{\partial \eta} = \mathop{\mathbb{E}}_{\substack{h_t\sim d(h_t),\\ h_{t+1}\sim d(h_{t+1})}} \left\{\frac{\partial l}{\partial \varphi}\frac{\partial \varphi}{\partial \eta} + \gamma \frac{\partial V(h_{t+1};\omega)}{\partial h_{t+1}}\frac{\partial h_{t+1}}{\partial \varphi}\frac{\partial \varphi}{\partial \eta}\right\}, \tag{18}$$

which contains two parts: 1) the first part comes from the cost $l_t = l(h_t, \hat{x}_t)$ and 2) the second part comes from h_{t+1}.

We refer to the algorithm that uses the gradients (16) and (18) to alternately update the value network and the estimator network, as ROE algorithm. Algorithm 1 and Fig. 3 show the pseudocode and diagram of ROE algorithm, respectively.

Algorithm 1 ROE algorithm

```
Initialize parameters $\eta_0$, $\omega_0$
Initialize state $h_0 \in \mathcal{S}$
repeat
    Rollout with estimator $\varphi_\eta$ from $h_t$
    Receive and store $h_{t+1}$
  PEV step:
  Calculate critic gradient (16)
  Update value function with: $\omega_{k+1} = \omega_k - \alpha\frac{\partial J_{critic}}{\partial \omega}$
  PIM step:
  Calculate actor gradient (18)
  Update estimator with: $\eta_{k+1} = \eta_k - \beta\frac{\partial J_{actor}}{\partial \eta}$
until Convergence
```

5. CONVERGENCE PROOFS OF ROE ALGORITHM WITH FUNCTION APPROXIMATION

Inspired by the proof of the convergence of approximate dynamic programming with discrete finite state (Bertsekas and Tsitsiklis (1996)), we prove the convergence of ROE algorithm with continuous state space in this section.

Definition 1. Define operators $\mathcal{T}_\varphi$ and $\mathcal{T}$:

$$\mathcal{T}_\varphi V(h_t) = l(h_t, \varphi(h_t)) + \gamma \mathop{\mathbb{E}}_{h_{t+1}\sim p(h_{t+1}|h_t,\varphi(h_t))} \{V(h_{t+1})\}$$

$$\mathcal{T} V(h_t) = \min_{\hat{x}_t}\left\{l(h_t, \hat{x}_t) + \gamma \mathop{\mathbb{E}}_{h_{t+1}\sim p(h_{t+1}|h_t,\hat{x}_t)} \{V(h_{t+1})\}\right\}.$$

Definition 2. If function f defined on a set $\mathcal{H}$ is bounded, the infinite norm is defined as

$$\|f(\cdot)\|_\infty = \sup_{h\in\mathcal{H}}(|f(h)|).$$

Lemma 1. Operator $\mathcal{T}_\varphi$ is a contraction mapping with respect to the infinite norm.

$$\|\mathcal{T}_\varphi V(\cdot) - \mathcal{T}_\varphi U(\cdot)\|_\infty \leq \gamma\|V(\cdot) - U(\cdot)\|_\infty.$$

Proof.

$$\begin{aligned}
&\|\mathcal{T}_\varphi V(\cdot) - \mathcal{T}_\varphi U(\cdot)\|_\infty \\
&= \sup_{h_t}\left(\gamma\left|\mathop{\mathbb{E}}_{h_{t+1}\sim p(h_{t+1}|h_t,\varphi(h_t))}\{V(h_{t+1}) - U(h_{t+1})\}\right|\right) \\
&\leq \sup_{h_t}\left(\gamma \mathop{\mathbb{E}}_{h_{t+1}\sim p(h_{t+1}|h_t,\varphi(h_t))}\{|V(h_{t+1}) - U(h_{t+1})|\}\right) \\
&\leq \sup_{h_t}\left(\gamma \mathop{\mathbb{E}}_{h_{t+1}\sim p(h_{t+1}|h_t,\varphi(h_t))}\{\|V(\cdot) - U(\cdot)\|_\infty\}\right) \\
&= \gamma\|V(\cdot) - U(\cdot)\|_\infty.
\end{aligned}$$

■

Lemma 2. If there exist some $\epsilon, \delta \geq 0$ which satisfy $\|V^k(\cdot;\omega_k) - V^k(\cdot)\|_\infty \leq \epsilon, \forall k$ and $\|\mathcal{T}_{\varphi_{k+1}}V^k(\cdot;\omega_k) - \mathcal{T}V^k(\cdot;\omega_k)\|_\infty \leq \delta$, $\forall k$, then

$$V^{k+1}(h) \leq V^k(h) + \frac{\delta + 2\gamma\epsilon}{1-\gamma}.$$

Here, $V_k(\cdot)$ is the real value function of the estimator φ_k and $V_k(\cdot;\omega_k)$ is the value function with function approximation.

Proof. Let $\rho_k = \sup_h\big(V^{k+1}(h) - V^k(h)\big)$. Obviously, we have

$$\begin{aligned}
V^{k+1}(h) &\leq V^k(h) + \rho_k \\
V^{k+1}(h) &= \mathcal{T}_{\varphi_{k+1}}V^{k+1}(h) \\
&\leq \mathcal{T}_{\varphi_{k+1}}\big(V^k(h) + \rho_k\big) \\
&= \mathcal{T}_{\varphi_{k+1}}V^k(h) + \gamma\rho_k.
\end{aligned} \tag{19}$$

Using assumption on φ_{k+1}, we have

$$\begin{aligned}
0 &\leq -\mathcal{T}_{\varphi_{k+1}}V^k(\cdot;\omega_k) + \mathcal{T}V^k(\cdot;\omega_k) + \delta \\
&\leq -\mathcal{T}_{\varphi_{k+1}}V^k(\cdot;\omega_k) + \mathcal{T}_{\varphi_k}V^k(\cdot;\omega_k) + \delta.
\end{aligned} \tag{20}$$

Combine (19) with (20), we have

$$\begin{aligned}
&V^{k+1}(h) - V^k(h) \\
&\leq \mathcal{T}_{\varphi_{k+1}}V^k(h) + \gamma\rho_k - V^k(h) \\
&\leq 2\gamma\|V^k(\cdot) - V^k(\cdot;\omega_k)\|_\infty + \gamma\rho_k + \delta \\
&\leq 2\gamma\epsilon + \gamma\rho_k + \delta.
\end{aligned}$$

The inequality makes use of the fact stated in the Lemma 1. As a result

$$\begin{aligned}
\rho_k &\leq 2\gamma\epsilon + \gamma\rho_k + \delta \\
\rho_k &\leq \frac{\delta + 2\gamma\epsilon}{1-\gamma}.
\end{aligned}$$

■

Lemma 3. Let $\tau_k = \sup_h(V^k(h) - V^*(h))$, then the sequence τ_k satisfies

$$\tau_{k+1} \leq \gamma\tau_k + \gamma\rho_k + \delta + 2\gamma\epsilon. \tag{21}$$

Proof. We first note that $V^k(h) \leq V^*(h) + \tau_k, \forall h$, which leads to

$$\begin{aligned}
\mathcal{T}V^k(h) &\leq \mathcal{T}(V^*(h) + \tau_k) \\
&= \mathcal{T}V^*(h) + \gamma\tau_k \\
&= V^*(h) + \gamma\tau_k.
\end{aligned}$$

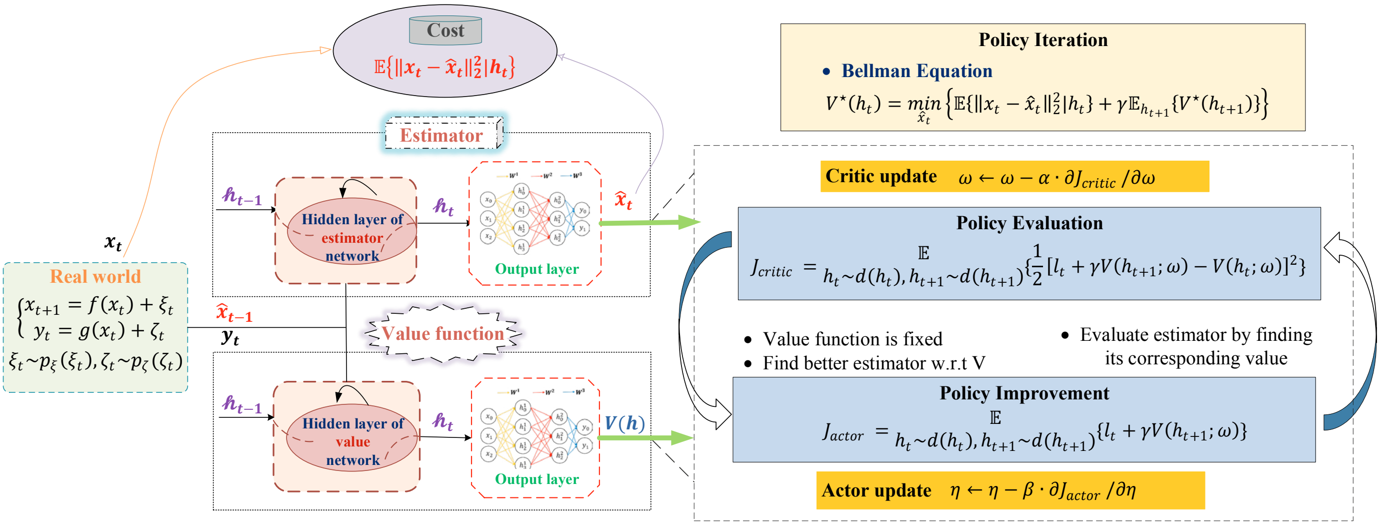

Cost
$\mathbb{E}\{\|x_t - \hat{x}_t\|_2^2 | h_t\}$
Estimator
h_{t-1}
Hidden layer of estimator network
h_t
Output layer
$\hat{x}_t$
x_t
Real world
$x_{t+1} = f(x_t) + \xi_t$
$y_t = g(x_t) + \zeta_t$
$\xi_t \sim p_\xi(\xi_t), \zeta_t \sim p_\zeta(\zeta_t)$
$\hat{x}_{t-1}$
y_t
Value function
h_{t-1}
Hidden layer of value network
h_t
Output layer
$V(h)$
Policy Iteration
Bellman Equation
$V^\star(h_t) = \min_{\hat{x}_t} \{\mathbb{E}\{\|x_t - \hat{x}_t\|_2^2 | h_t\} + \gamma \mathbb{E}_{h_{t+1}}\{V^\star(h_{t+1})\}\}$
Critic update $\omega \leftarrow \omega - \alpha \cdot \partial J_{critic} / \partial \omega$
Policy Evaluation
$J_{critic} = \mathbb{E}_{h_t \sim d(h_t), h_{t+1} \sim d(h_{t+1})} \{\frac{1}{2}[l_t + \gamma V(h_{t+1}; \omega) - V(h_t; \omega)]^2\}$
Value function is fixed
Find better estimator w.r.t V
Evaluate estimator by finding its corresponding value
Policy Improvement
$J_{actor} = \mathbb{E}_{h_t \sim d(h_t), h_{t+1} \sim d(h_{t+1})} \{l_t + \gamma V(h_{t+1}; \omega)\}$
Actor update $\eta \leftarrow \eta - \beta \cdot \partial J_{actor} / \partial \eta$

Fig. 3. Training procedure of ROE algorithm. Note that during training, we need to roll out from h_t to h_{t+1}. It seems that we can never obtain the transition probability for h_t analytically. However, it is not the case. It is reasonable to obtain x_t during training using the stochastic model which can be seen as "real world" and $\hat{x}_t$ can be obtained from the "estimator". The learned estimator is called ROE and can be applied online.

We then have

$$\begin{aligned}
\mathcal{T}_{\varphi_{k+1}} V^k(h) &\le \mathcal{T}_{\varphi_{k+1}} \left(V^k(h; \omega_k) + \epsilon\right). \\
&= \mathcal{T}_{\varphi_{k+1}} V^k(h; \omega_k) + \gamma\epsilon \\
&\le \mathcal{T} V^k(h; \omega_k) + \delta + \gamma\epsilon \\
&\le \mathcal{T}\left(V^k(h) + \epsilon\right) + \delta + \gamma\epsilon \\
&= \mathcal{T} V^k(h) + \delta + 2\gamma\epsilon \\
&\le \mathcal{T}\left(V^*(h) + \tau_k\right) + \delta + 2\gamma\epsilon \\
&= V^*(h) + \gamma\tau_k + \delta + 2\gamma\epsilon.
\end{aligned}$$

Thus

$$\begin{aligned}
V^{k+1}(h) &= \mathcal{T}_{\varphi_{k+1}} V^{k+1}(h) \\
&\le \mathcal{T}_{\varphi_{k+1}} \left(V^k(h) + \rho_k\right) \\
&= \mathcal{T}_{\varphi_{k+1}} V^k(h) + \gamma\rho_k \\
&\le V^*(h) + \gamma\tau_k + \delta + 2\gamma\epsilon + \gamma\rho_k.
\end{aligned}$$

That is to say

$$\tau_{k+1} \le \gamma\tau_k + \delta + 2\gamma\epsilon + \gamma\rho_k.$$

■

Theorem 1. The sequence of estimator φ_k generated by the ROE algorithm satisfies

$$\limsup_{k\to\infty} \|V^k(\cdot) - V^*(\cdot)\|_\infty \le \frac{\delta + 2\gamma\epsilon}{(1-\gamma)^2}.$$

Proof. According to Lemma 3, take the limit on both sides of (21), we have

$$\limsup_{k\to\infty} \tau_k \le \frac{\delta + 2\gamma\epsilon + \gamma\rho_k}{1-\gamma}.$$

Combining with Lemma 2, we have

$$\rho_k \le \frac{\delta + 2\gamma\epsilon}{1-\gamma}.$$

Thus

$$\limsup_{k\to\infty} \tau_k \le \frac{\delta + 2\gamma\epsilon}{(1-\gamma)^2}.$$

■

According to Theorem 1, when δ,$\epsilon \to 0$, the estimator φ_k will finally convergence to $\varphi^\star$.

6. EXPERIMENT RESULTS

In this section, we first adopt the linear pendulum task to demonstrate the equivalence between the time-invariant solutions of REP and MVE. Then, we evaluate the performance of ROE by estimating the centroid side slip angle and yaw rate of a nonlinear vehicle system.

6.1 Experiment I: Estimation of Pendulum System

The linear dynamics of the pendulum is shown as below:

$$\begin{aligned}
x_{t+1} &= F x_t + B u_t + \xi_t \\
y_t &= x_t + \zeta_t,
\end{aligned}$$

where

$$\xi_t \sim \begin{bmatrix} \mathcal{N}(0,\ 0.005) \\ \mathcal{N}(0,\ 0.01) \end{bmatrix},\ \zeta_t \sim \begin{bmatrix} \mathcal{N}(0,\ 0.1) \\ \mathcal{N}(0,\ 0.3) \end{bmatrix}.$$

And

$$F = \begin{bmatrix} 1 & T \\ -\dfrac{gT}{l} & 1 - \dfrac{\mu T}{ml^2} \end{bmatrix},\ B = \begin{bmatrix} 0 \\ \dfrac{T}{ml^2} \end{bmatrix},$$

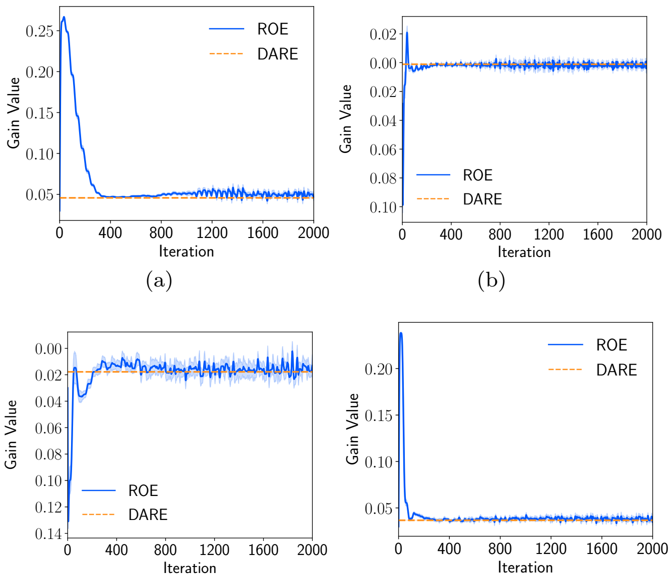
with $g = 9.81$, $T = 0.01$, $\mu = 0.01$, $m = 1$ and $l = 1$.

Given the time-invariant estimator form shown in (7) or (9), we use the DARE (6) and ROE (Algorithm 1) to solve their matrix gain $\bar{K}$ and η^* respectively. The experiment is repeated 40 times independently and Fig. 4 plots the training curves. Table 1 shows the average results over the last 100 iterations.

Table 1. Final Solution of DARE and ROE

	k_{11}	k_{12}	k_{21}	k_{22}
DARE	0.0457	-0.00110	-0.0178	0.0367
ROE	0.0495	-0.00109	-0.0152	0.0379

As shown in Fig. 4 and Table 1, the four elements of the gain matrix obtained by the ROE algorithm converges to the gain of time-invariant filter after 1000 iterations.



(a) (b) (c) (d)

Fig. 4. Training curves of experiment I. (a) $\bar{K}_{11}$ and η_{11}^* (b) $\bar{K}_{12}$ and η_{12}^* (c) $\bar{K}_{21}$ and η_{21}^* (d) $\bar{K}_{22}$ and η_{22}^*. Solid lines are average values over 40 runs. Shaded regions correspond to 95% confidence interval. The gain matrix solved by ROE finally convergences to the solution of DARE.

6.2 Experiment II: Estimation of Vehicle State

The employed vehicle dynamics (Bakker et al. (1987)) is as follows:

$$\theta_{t+1} = f_1(\theta_t, \Omega_t) + \xi_{1,t}$$
$$\Omega_{t+1} = f_2(\theta_t, \Omega_t) + \xi_{2,t},$$

where θ_t is the slide slip angle and Ω_t is the yaw rate. The observations are calculated as

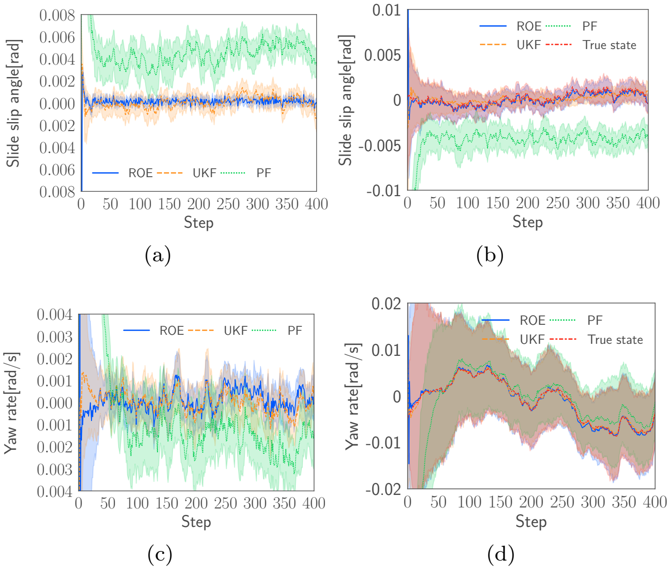
$$y_{1,t} = \theta_t + \zeta_{1,t}$$
$$y_{2,t} = \Omega_t + \zeta_{2,t}.$$

Due to the characteristics of tires, the model has a strong degree of non-linearity,

$$\begin{aligned} f_1(\theta_t, \Omega_t) =& A_1 b \cos\delta_t \sin\{C \arctan[B(-\delta_t + \theta_t + \frac{a\Omega_t}{u})]\} \\ &+ A_1 a \sin\{C \arctan[B(\theta_t - \frac{b\Omega_t}{u})]\} - \Omega_t T, \\ f_2(\theta_t, \Omega_t) =& - A_2 \sin\{C \arctan[B(-\delta_t + \theta_t + \frac{a\Omega_t}{u})]\} \\ &+ A_2 \sin\{C \arctan[B(\theta_t - \frac{b\Omega_t}{u})]\}. \end{aligned}$$

To evaluate the performance of ROE, we compare it with the UKF and PF. Note that although the value function should be GRU or LSTM and the inputs are the history state h_t, we find that using true state x_{t-1} and estimate $\hat{x}_{t-1}$ of the last step to represent the history state works better. In this case, only a multi-layer perceptron (MLP) is needed to parameterize the value function.

Each algorithm is evaluated using the Monte Carlo experiment (5000 steps per experiment), which is repeated 100 times from different initial states. The experiment results are shown in Fig. 5. Table 2 gives the RMSE (Root Mean Square Error) during experiments of all algorithms. Besides, We use RTX3090 to calculate the estimated state, and the average time consumption per step is also shown in Table 2.



(a) (b) (c) (d)

Fig. 5. Results for experiment II. (a) Estimation error of state 1. (b) The true state and estimates of 3 methods for state 1. (c) Estimation error of state 2. (d) The true state and estimates of 3 methods for state 2. The solid lines correspond to the mean and the shaded regions correspond to 95% confidence interval over 20 runs.

Table 2. RMSE and Average Time

	ROE	UKF	PF
θ(rad)	0.000989	0.002239	0.005742
Ω(rad/s)	0.002371	0.002597	0.005986
Time consumption(s)	0.000541	0.000777	1.0736

As shown in Fig. 5 and Table 2, the accuracy of ROE outperforms that of UKF and PF (1000 particles), especially for the estimation of slide slip angle. The time-consuming is superior because ROE is trained offline, and there is no need for online sampling and calculation for ROE.

The detailed parameters for the 2-DOF system and noise are listed in Table 3. The detailed parameters for training procedure is listed in Table 4.

7. CONCLUSION

This work proposed an offline estimation problem called REP for general nonlinear systems and non-Gaussian noises. Besides, we proposed ROE algorithm to find an approximated optimal solution of REP. This algorithm employs RNN to tackle the historical sequential information simultaneously. The estimator is trained offline and can be used efficiently without online computations. Simulation results show the superiority of our method over existing nonlinear estimators in terms of accuracy and time-consuming. However, there are still quite a few issues that have not yet been addressed in the paper. For example, how to describe uncertainty of the estimate and will the algorithm still convergence considering the property of neural networks and finite samples.

Table 3. Detailed Parameters for Vehicle Dynamics

Model parameters	**Value**
A_1	$\frac{DgT}{uL}$
A_2	$\frac{DmgabT}{I_{zz}L}$
δ_t	$-\theta_t$
T	0.01
B	14
C	1.43
D	0.75
m	1500
u	20
I_{zz}	2420
L	2.54
a	1.14
b	1.4
g	9.81
ξ_t	$E(\xi_{1,t} + \xi_{2,t})$
$\xi_{1,t}$	$\begin{bmatrix} \mathcal{N}\left(100, 10^4\right) \\ \mathcal{N}\left(100, 10^4\right) \end{bmatrix}$
$\xi_{2,t}$	$\begin{bmatrix} \mathcal{U}(-1126.25, 1326.25) \\ \mathcal{U}(-900, 1100) \end{bmatrix}$
E	$\begin{bmatrix} 3.33 * 10^{\ 7} & 3.33 * 10^{\ 7} \\ 0 & 5.4 * 10^{\ 7} \end{bmatrix}$
ζ_t	$\begin{bmatrix} 8.33 * 10^{\ 3}\chi^2(1) \\ 2.47 * 10^{\ 2}\chi^2(1) \end{bmatrix}$

Table 4. Detailed Parameters for Training

Training parameters	**Value**
Optimizer	ADAM (β_1=0.9, β_2=0.99)
Batch size	512
Discounted factor	0.9
Learning rate	0.00001
Number of hidden layers	2
Number of hidden units per layer	512
ValueNet	
Approximation function	MLP
Activation function of hidden layer	Elu
Activation function of output layer	Softplus
PolicyNet	
Approximation Function	GRU
Length of series	20
Activation function of hidden layer	Tanh
Activation function of output layer	Identity

REFERENCES

Allen-Zhu, Z., Li, Y., and Song, Z. (2018). On the convergence rate of training recurrent neural networks. *arXiv preprint arXiv:1810.12065.*

Alspach, D. and Sorenson, H. (1972). Nonlinear bayesian estimation using gaussian sum approximations. *IEEE transactions on automatic control*, 17(4), 439–448.

Anderson, B.D. and Moore, J.B. (2012). *Optimal filtering.* Courier Corporation.

Bakker, E., Nyborg, L., and Pacejka, H.B. (1987). Tyre modelling for use in vehicle dynamics studies. *SAE Transactions*, 190–204.

Bertsekas, D.P. and Tsitsiklis, J.N. (1996). *Neuro-dynamic programming.* Athena Scientific.

Duan, J., Guan, Y., Li, S.E., Ren, Y., and Cheng, B. (2020). Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors. *arXiv preprint arXiv:2001.02811.*

Elman, J.L. (1990). Finding structure in time. *Cognitive science*, 14(2), 179–211.

Guan, Y., Li, S.E., Duan, J., Li, J., Ren, Y., and Cheng, B. (2019). Direct and indirect reinforcement learning. *arXiv preprint arXiv:1912.10600.*

Hammer, B. (2000). On the approximation capability of recurrent neural networks. *Neurocomputing*, 31(1-4), 107–123.

Hu, L., Wu, C., and Pan, W. (2020). Lyapunov-based reinforcement learning state estimator. *arXiv preprint arXiv:2010.13529.*

Julier, S.J. and Uhlmann, J.K. (1997). New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, 182–193. International Society for Optics and Photonics.

Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82D, 35–45.

Li, S.E. (2020). Reinforcement Learning and Control. Tsinghua University: Lecture Notes. http://www.idlabtsinghua.com/thulab/labweb/publications.html.

Lipton, Z.C., Berkowitz, J., and Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019.*

Liu, J.S. and Chen, R. (1998). Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443), 1032–1044.

Morimoto, J. and Doya, K. (2007). Reinforcement learning state estimator. *Neural computation*, 19(3), 730–756.

Musoff, H. and Zarchan, P. (2009). *Fundamentals of Kalman filtering: a practical approach.* American Institute of Aeronautics and Astronautics.

Schäfer, A.M. and Zimmermann, H.G. (2006). Recurrent neural networks are universal approximators. In *International Conference on Artificial Neural Networks*, 632–640. Springer.

Simon, D. (2006). *Optimal state estimation: Kalman, H infinity, and nonlinear approaches.* John Wiley & Sons.

Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Jordan, M.I., and Sastry, S.S. (2004). Kalman filtering with intermittent observations. *IEEE transactions on Automatic Control*, 49(9), 1453–1464.

Smith, G.L., Schmidt, S.F., and McGee, L.A. (1962). *Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle.* National Aeronautics and Space Administration.

Sun, S.l. (2004). Multi-sensor optimal information fusion kalman filters with applications. *Aerospace Science and Technology*, 8(1), 57–62.

Sutton, R.S. and Barto, A.G. (1998). *Reinforcement learning: An introduction.* MIT Press, Cambridge. doi: 10.1109/TNN.1998.712192.

Thrun, S. (2002). Probabilistic robotics. *Communications of the ACM*, 45(3), 52–57.