sensors

MDPI

Article

Reinforcement Learning-Aided Channel Estimator in Time-Varying MIMO Systems

Tae-Kyoung Kim [1] **and Moonsik Min** [2,*]

[1] Department of Electronic Engineering, Gachon University, Seongnam 13120, Republic of Korea; tk415kim@gmail.com
[2] School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, Republic of Korea
[*] Correspondence: msmin@knu.ac.kr

Abstract: This paper proposes a reinforcement learning-aided channel estimator for time-varying multi-input multi-output systems. The basic concept of the proposed channel estimator is the selection of the detected data symbol in the data-aided channel estimation. To achieve the selection successfully, we first formulate an optimization problem to minimize the data-aided channel estimation error. However, in time-varying channels, the optimal solution is difficult to derive because of its computational complexity and the time-varying nature of the channel. To address these difficulties, we consider a sequential selection for the detected symbols and a refinement for the selected symbols. A Markov decision process is formulated for sequential selection, and a reinforcement learning algorithm that efficiently computes the optimal policy is proposed with state element refinement. Simulation results demonstrate that the proposed channel estimator outperforms conventional channel estimators by efficiently capturing the variation of the channels.

Keywords: data-aided channel estimation; non-iterative approach; first-order Gaussian—Markov channel model; reinforcement learning

check for updates

Citation: Kim, T.-K.; Min, M. Reinforcement Learning-Aided Channel Estimator in Time-Varying MIMO Systems. *Sensors* **2023**, *23*, 5689. https://doi.org/10.3390/s23125689

Academic Editors: Xinyu Zhou, Wing Shing Chan and Liheng Zhou

Received: 17 April 2023
Revised: 8 June 2023
Accepted: 16 June 2023
Published: 18 June 2023

Copyright: © 2023 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

The multi-input multi-output (MIMO) system is a key technology in modern communication and can significantly improve channel capacity and communication reliability by using multiple antennas [1–7]. Spatial multiplexing and diversity gain are representative schemes for this improvement [1,2]. Notably, the channel capacity increases linearly with the number of either transmitter and receiver antennas. However, this increase is based on the unrealistic assumption of perfect channel state information (PCSI) at both the transmitter and receiver.

Many studies have proposed improving the channel estimation accuracy with limited time and frequency resources [8–15]. A representative method is pilot-aided channel estimation, which exploits the information shared between a transmitter and receiver. Linear minimum-mean square-error (LMMSE) channel estimation is a well-known method for pilot-aided channel estimation, owing to its simple structure [8]. However, LMMSE channel estimation exhibits unsatisfactory performance with a limited number of pilots. Thus, many pilots are required to satisfy the performance requirement, which decreases the spectral efficiency.

To overcome this problem, data-aided channel estimation has been investigated in which the detected data symbols are exploited as additional pilot symbols [16–26]. However, the detected data symbols may have errors that degrade the accuracy of channel estimation. The iterative turbo equalizer can overcome this degradation by increasing the maximum-a-posteriori probability (MAP) [16–22]. However, such an iterative turbo equalizer has considerable complexity and latency at the receivers.

As a non-iterative approach, the reinforcement learning (RL)-aided channel estimator was introduced in [27–33]. The basic concept of this approach is the sequential selection of detected data symbols to minimize the channel estimation errors. Hence, a Markov decision process (MDP) was defined to solve the sequential selection, and the corresponding optimal policy was derived in a closed-form expression in [31]. In [32], a low-complexity algorithm was investigated by introducing sub-blocks and finite backup samples, and the computational complexity and latency were significantly reduced without performance loss. Recently, a general framework for RL-aided channel estimation was studied in [33] based on Monte Carlo tree search. However, the RL-aided channel estimators in [31–33] were originally considered in time-invariant channels; they perform insufficiently in time-varying channels.

In this paper, we propose an RL-aided channel estimator for time-varying MIMO channels. To achieve this, we first introduce an optimization problem for an RL-aided channel estimator in time-varying channels. We then formulate an MDP to solve the optimization problem, and propose an RL algorithm for the MDP that considers the time-varying nature of the channel. The main contributions of this paper are as follows:

- We propose an RL-aided channel estimator for time-varying channels modeled using a first-order Gaussian—Markov process. First, we define the optimization problem in time-varying channels to select the detected data symbols and minimize the estimation error between the estimated and current channels. This optimization problem is different from those in [31–33], where the selection of the detected data symbols is unchanged because the current channel remains unchanged with the time slot index.
- We propose an RL algorithm for the optimization problem that captures the time-varying nature of a channel. Because the optimization problem minimizes the estimation error between the estimated and current channels, we adjust the weights of the data symbols to improve the channel accuracy of the current channel. Using this adjustment, we derive the optimal policy as a closed-form solution. Note that the proposed optimal policy differs from those in [31–33] because the influence of soft-decision symbols in the virtual state for future rewards gradually diminishes as the time slot index increases.
- We propose a further performance improvement scheme to refine the state elements. This is because the previously selected data symbol degrades the estimation accuracy of the current channel. To improve the estimation accuracy, we refine the previously selected data symbol by reflecting the channel variation. In addition, we remove selected data symbols that are too old by introducing a sliding window, because they have a large noise variance to estimate the current channel. Through simulations, we demonstrated the effectiveness of the proposed channel estimator compared with conventional channel estimators in time-varying channels.

The remainder of this paper is organized as follows. In Section 1, we introduce the system model, optimization problem, and the MDP. The proposed channel estimator, which determines the optimal policy for time-varying channels, is described in Section 2. We propose a further performance improvement scheme in Section 3. In Section 4, we present simulation results to demonstrate the effectiveness of the proposed channel estimator. Finally, we provide our conclusions in Section 5.

2. Preliminaries

This section describes the system model of a data-aided channel estimator for time-varying MIMO channels. We present the considered channel estimation and data detection schemes based on the model and introduce an optimization problem for data-aided channel estimation.

2.1. System Model

We consider MIMO systems in which a transmitter with a number of transmit antennas N_t communicates with a receiver with a number of receive antennas N_r (Figure 1).

The information is first encoded and mapped to the symbol constellation where $\mathcal{X}$ is the symbol constellation set. The transmitted symbol at time n denoted by $\mathbf{x}[n] \in \mathcal{X}^{N_t}$ is then sent over a wireless channel. We model the wireless channel using a first-order Gaussian—Markov process as a time-varying channel model [34–38], where the channel matrix $\mathbf{H}[n] \in \mathbf{C}^{N_t \times N_r}$ has its (t, r)-th component between the t-th and r-th antennas following a Rayleigh fading $\mathcal{CN}(0, 1)$ distribution. The temporal correlation of the wireless channel, denoted by $\epsilon \in [0, 1]$, increases with velocity. Based on this model, the channel matrix $\mathbf{H}[n]$ at time slot n is given by

$$\mathbf{H}[n] = \sqrt{1-\epsilon^2}\mathbf{H}[n-1] + \epsilon\Delta[n], \tag{1}$$

where $\Delta[n]$ follows a $\mathcal{CN}(0, 1)$ distribution.

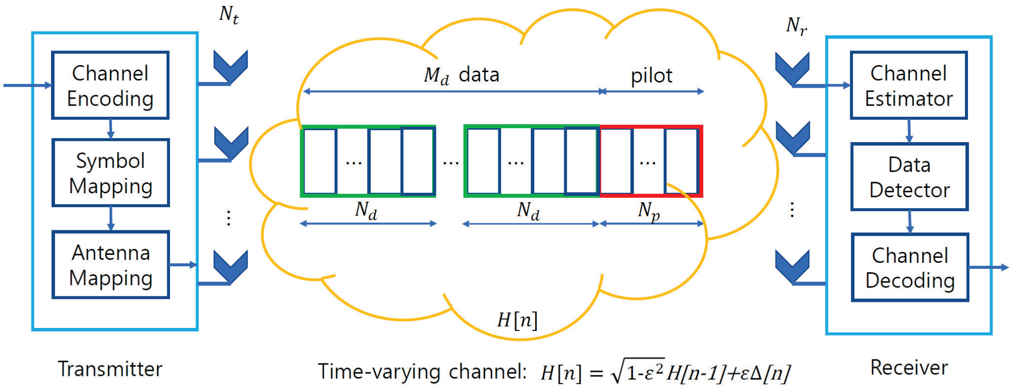

Figure 1. Considered system model and frame structure in time-varying channels.

When the transmitter sends the symbol $\mathbf{x}[n]$ to the receiver over the wireless channel $\mathbf{H}[n]$, the received symbol $\mathbf{z}[n]$ is given by

$$\mathbf{z}[n] = \mathbf{H}^H[n]\mathbf{x}[n] + \mathbf{n}[n], \tag{2}$$

where $(\cdot)^H$ denotes the conjugate transpose. $\mathbf{n}[n]$ is the additive white Gaussian noise (AWGN) at time slot n, with distribution $\mathcal{CN}(\mathbf{0}_{N_r}, \sigma_n^2\mathbf{I}_{N_r})$, where $\mathbf{0}_m$ and $\mathbf{I}_m$ respectively denote $m \times m$ zero and identity matrices.

The frame consists of one pilot and M_d data blocks (Figure 1). The pilot block contains N_p symbols, whereas each data block contains N_d symbols. $\mathcal{M}_p = \{1, \ldots, N_p\}$ is defined as the pilot index set and $\mathcal{M}_d = \{(d-1)N_d + 1, \ldots, dN_d\}$ is defined as the data index set. We consider data-aided channel estimation, where the receiver obtains the initial channel estimates using pilot symbols, and the accuracy of the initial channel estimates is improved by exploiting data symbols.

We adopt the LMMSE method as the basic channel estimation method because it has a simple structure and provides a reasonable performance. Based on the LMMSE method, $\hat{\mathbf{h}}_r$ of the r-th row for the initial channel estimate $\hat{\mathbf{H}}$ can be obtained as

$$\hat{\mathbf{h}}_r = \left(\mathbf{X}^p(\mathbf{X}^p)^H + \sigma_n^2\mathbf{I}_{N_t}\right)^{-1}\mathbf{X}^p(\mathbf{z}_r^p)^H, \tag{3}$$

where $(\cdot)^{-1}$ is the inverse operation. $\mathbf{X}^p = [\mathbf{x}[1], \ldots, \mathbf{x}[N_p]]^T$ and $\mathbf{z}_r^p = [\mathbf{z}_r[1], \ldots, \mathbf{z}_r[N_p]]^T$ are the pilot and corresponding received symbols in the pilot block, respectively.

The conventional channel estimator performs data detection at the receiver using the initial channel estimates $\hat{\mathbf{h}}$. Because the MAP rule guarantees optimal performance, we adopt it for data detection, which is given by

$$\hat{\mathbf{x}}[n] = \underset{\mathbf{x}_k \in \mathcal{X}^{N_t}}{\operatorname{argmax}}\, \theta_k[n]. \tag{4}$$

where $|\cdot|$ is the cardinality of a set. $\mathbf{x}_k \in \mathcal{X}^{N_t}$ where k belongs to the index set of the symbol vector candidate $\mathcal{K} = \{1, \dots, |\mathcal{X}^{N_t}|\}$. $\theta_k[n]$ denotes a posteriori probability (APP), which is given by

$$\theta_k[n] = \frac{\mathbb{P}[\mathbf{z}[n]|\mathbf{x}[n] = \mathbf{x}_k]\mathbb{P}[\mathbf{x}[n] = \mathbf{x}_k]}{\sum\limits_{j \in \mathcal{K}} \mathbb{P}[\mathbf{z}[n]|\mathbf{x}[n] = \mathbf{x}_j]\mathbb{P}[\mathbf{x}[n] = \mathbf{x}_j]}, \tag{5}$$

where the likelihood probability in (5) is calculated by assuming the AWGN channel as

$$\mathbb{P}[\mathbf{z}[n]|\mathbf{x}[n] = \mathbf{x}_k] = \frac{1}{(\pi\sigma_n^2)^{N_r}} e^{-\frac{\|\mathbf{z}[n] - \hat{\mathbf{h}}^H \mathbf{x}_k\|^2}{\sigma_n^2}}. \tag{6}$$

where $\|\cdot\|^2$ denotes the norm operation and $\mathbb{P}(\cdot)$ is the probability of an event. The a priori probability in (5) is also assumed to have an equal probability for possible candidate transmitted symbol $\mathbb{P}[\mathbf{x}[n] = \mathbf{x}_k] = \frac{1}{|\mathcal{X}|^{N_t}}$.

2.2. Problem

In a time-varying channel, the estimation accuracy of $\hat{\mathbf{h}}$ decreases gradually as time slot index n increases. This degradation results in poor detection performance at the receiver. Because the detected data symbol may have an error owing to the channel, an incorrect use of the detected data symbol severely degrades performance. To overcome this degradation, we consider a data-aided channel estimator that selects the detected data symbols for data-aided channel estimation.

For the selection, we define action $a \in \mathcal{A} = \{0, 1\}$ where the detected data symbol is used in channel estimation when $a = 1$; otherwise, the detected data symbol is not used. When we define $\mathbf{a} \in \{0, 1\}^{N_d}$ as a set of actions, the considered data-aided channel estimation can be obtained using this set as

$$\hat{\mathbf{h}}_r(\mathbf{a}) = \left(\hat{\mathbf{X}}(\mathbf{a})\hat{\mathbf{X}}^H(\mathbf{a}) + \sigma_n^2 \mathbf{I}_{N_t}\right)^{-1} \hat{\mathbf{X}}(\mathbf{a})\mathbf{z}_r^H(\mathbf{a}) \tag{7}$$

where $\mathbf{z}_r(\mathbf{a}) = [\mathbf{z}_r^p, \mathbf{z}_r[e_1(\mathbf{a})], \dots, \mathbf{z}_r[e_{\|\mathbf{a}\|_0}(\mathbf{a})]]^T$ and $\hat{\mathbf{X}}(\mathbf{a}) = [\mathbf{X}^p, \hat{\mathbf{x}}[e_1(\mathbf{a})], \dots, \hat{\mathbf{x}}[e_{\|\mathbf{a}\|_0}(\mathbf{a})]]^T$. The time slot index of the i-th nonzero element is denoted as $e_i(\cdot)$. We then define the optimization problem as

$$\mathbf{a}^\star = \underset{\mathbf{a}}{\operatorname{argmin}}\, \mathcal{E}\{\|\hat{\mathbf{h}}(\mathbf{a}) - \mathbf{H}[n]\|^2\}, \tag{8}$$

where $\mathcal{E}(\cdot)$ is the expectation of a random variable.

Compared with previous studies [31–33], the optimization problem in (8) considers the selection to minimize the MSE between the estimated channel and $\mathbf{H}[n]$. Because the channel is variant with time slot index n, the best action $\mathbf{a}^\star$ may be different with time slot index n. That is, the best action in the previous time slot index may be invalid in the next time slot index. In addition, the optimization problem is difficult to solve because the number of candidate actions increases exponentially with the data symbol length. An exhaustive search for action candidates is not feasible in practical applications. To resolve these difficulties, we introduce a sequential selection of the detected data symbols and a refinement of the selected data.

2.3. Markov Decision Process

We formulate an MDP that solves the optimization problem in (8). To achieve this, we define state $\mathcal{S}_n$, transition function $\mathcal{T}_{n+1}^{(a,j)}(\mathsf{S}_n)$, action $\mathcal{A}$, and reward $\mathcal{R}(\mathsf{S}_n, \mathsf{S}_{n+1})$ [39]. Subsequently, the Q-value function $\mathcal{Q}(\mathsf{S}_n, a)$ and the optimal policy $\pi^\star(\mathsf{S}_n)$ will be presented. The basic definitions for the MDP are adopted from those in [31–33]; however, the

RL solution for the MDP is different from those in previous studies, which will be explained in the next section.

The state set $\mathcal{S}_n$ is defined as

$$\begin{aligned}\mathcal{S}_n = \Big\{ (\mathbf{X}_n, \hat{\mathbf{X}}_n, \mathcal{C}) \,\Big|\, &\mathbf{X}_n = \Big[\mathbf{x}[1] \cdots \mathbf{x}[N_p], \mathbf{x}_{k_{\mathcal{C}(1)}}, \cdots, \mathbf{x}_{k_{\mathcal{C}(|\mathcal{C}|)}}\Big], \\ &\hat{\mathbf{X}}_n = \big[\mathbf{x}[1] \cdots \mathbf{x}[N_p], \hat{\mathbf{x}}[\mathcal{C}(1)], \cdots, \hat{\mathbf{x}}[\mathcal{C}(|\mathcal{C}|)]\big], \\ &\mathcal{C} \subset \{1, \cdots, n-1\} \Big\}, \end{aligned} \tag{9}$$

where $\mathcal{C}$ is the set of time slot indices where the symbol is used in channel estimation, and $\mathcal{C}(i)$ is the i-th smallest element. $k_n \in \mathcal{K} = \{1, \ldots, |\mathcal{X}|\}$ is the transmitted symbol index at time slot n. Based on the expression, we can obtain the proposed channel estimate using the state $\mathsf{S}_n \in \mathcal{S}_n$ as

$$\hat{\mathbf{h}}_r(\mathsf{S}_n) = \left(\hat{\mathbf{X}}_n \hat{\mathbf{X}}_n^H + \sigma_n^2 \mathbf{I}_{N_t}\right)^{-1} \hat{\mathbf{X}}_n \mathbf{z}_r^H(\mathsf{S}_n) \tag{10}$$

where $\mathbf{z}_r(\mathsf{S}_n) = [\mathbf{z}_r^p, \mathbf{z}_r[\mathcal{C}(1)], \ldots, \mathbf{z}_r[\mathcal{C}(|\mathcal{C}|)]]^T$. Note that $\mathcal{S}_n$ is the set of all states and S_n is the state.

The action set is defined as $\mathcal{A} = \{0, 1\}$. As explained in the previous subsection, the detected data symbol is used in the proposed channel estimation when $a = 1$; otherwise, the detected data symbol is not used. The transition function $\mathcal{T}^{(a,j)}(\mathsf{S}_n)$ from state $\mathsf{S}_n \in \mathcal{S}_n$ is defined as

$$\mathcal{T}^{(a,j)}(\mathsf{S}_n) = \mathbb{P}\Big[\mathsf{U}_{n+1}^{(a,j)}(\mathsf{S}_n) \,\Big|\, \mathsf{S}_n, a\Big] = \begin{cases} \mathcal{I}[\mathbf{x}[n] = \mathbf{x}_j], & j \in \mathcal{J}_a, a = 1, \\ 1, & j \in \mathcal{J}_a, a = 0. \end{cases} \tag{11}$$

where $\mathcal{I}(\cdot)$ equals one when the event is true and zero otherwise. $\mathcal{J}_0 \in= \{0\}$ and $\mathcal{J}_1 \in \{1, \ldots, K\}$. $\mathsf{U}_{n+1}^{(a,j)}(\mathsf{S}_n) \in \mathcal{U}_{n+1}^{(a,j)}(\mathsf{S}_n)$ is a possible candidate for the next state from state S_n and is defined as

$$\mathcal{U}_{n+1}^{(a,j)}(\mathsf{S}_n) = \begin{cases} \big([\mathbf{X}_n, \mathbf{x}_j], [\hat{\mathbf{X}}_n, \hat{\mathbf{x}}[n]], [\mathcal{C} \cup n]\big), & j \in \mathcal{J}_a, a = 1, \\ (\mathbf{X}_n, \hat{\mathbf{X}}_n, \mathcal{C}), & j \in \mathcal{J}_a, a = 0. \end{cases} \tag{12}$$

The reward $\mathcal{R}(\mathsf{S}_n, \mathsf{S}_{n+1})$ is defined as the difference between the MSEs at the current state $\mathsf{S}_n \in \mathcal{S}_n$ and the next state $\mathsf{S}_{n+1} \in \mathcal{S}_{n+1}$, which is given by

$$\begin{aligned} \mathcal{R}(\mathsf{S}_n, \mathsf{S}_{n+1}) &= \mathcal{E}\Big[\|\hat{\mathbf{h}}_r(\mathsf{S}_n) - \mathbf{h}_r[n]\|^2\Big] - \mathcal{E}\Big[\|\hat{\mathbf{h}}_r(\mathsf{S}_{n+1}) - \mathbf{h}_r[n+1]\|^2\Big] \\ &= \mathrm{Tr}[\mathbf{B}(\mathsf{S}_n)] - \mathrm{Tr}[\mathbf{B}(\mathsf{S}_{n+1})] = \mathrm{Tr}[\mathbf{B}(\mathsf{S}_n) - \mathbf{B}(\mathsf{S}_{n+1})], \end{aligned} \tag{13}$$

where $\mathbf{B}(\mathsf{S}_n) = \mathcal{E}\Big[(\hat{\mathbf{h}}_r(\mathsf{S}_n) - \mathbf{h}_r[n])(\hat{\mathbf{h}}_r(\mathsf{S}_n) - \mathbf{h}_r[n])^H\Big]$ is error covariance. Unlike in [31–33], the error covariance is defined between the estimated channel $\hat{\mathbf{h}}_r(\mathsf{S}_n)$ and $\mathbf{h}_r[n]$ at time slot index n.

The Q-value function $\mathcal{Q}(\mathsf{S}_n, a)$ is the sum of the rewards, which is given by

$$\mathcal{Q}(\mathsf{S}_n, a) = \sum_{j \in \mathcal{J}_a} \mathcal{T}^{(a,j)}(\mathsf{S}_n) \Big[\mathcal{R}\Big(\mathsf{S}_n, \mathsf{U}_{n+1}^{(a,j)}(\mathsf{S}_n)\Big) + \gamma \mathcal{V}^\star\Big(\mathsf{U}_{n+1}^{(a,j)}(\mathsf{S}_n)\Big)\Big], \tag{14}$$

where $\mathrm{Tr}(\cdot)$ is a trace operation. $\mathcal{V}^\star\Big(\mathsf{U}_{n+1}^{(a,j)}(\mathsf{S}_n)\Big)$ is the optimal sum of future reward after $\mathsf{U}_{n+1}^{(a,j)}(\mathsf{S}_n)$. γ is a discounting factor whose value is assumed as one because the proposed channel estimator also considers the effect of future rewards at the ending state [31].

The optimal policy maximizes the Q-value function, which is expressed as

$$\pi^{\star}(\mathsf{S}_n) = \underset{a\in\mathcal{A}}{\operatorname{argmax}}\ \mathcal{Q}(\mathsf{S}_n, a). \tag{15}$$

Solving the optimization problem in (15) is highly difficult because the transition probability $\mathcal{T}^{(a,j)}(\mathsf{S}_n)$ is unknown, and the number of candidate states exponentially increases with the data length. An effective method to solve this problem is to use a reinforcement learning algorithm. Therefore, the proposed channel estimator also adopts a reinforcement learning algorithm, but the effect of the time-varying channel is also considered in comparison with [31–33].

A deep reinforcement learning (DRL) approach is a promising solution for dealing with the dimension explosion of the states by leveraging deep neural networks. To apply the DRL approach to our MDP, an agent needs to interact with an environment to obtain an action-value function for a given action and state. However, both the states and rewards of our MDP are not observable at the receiver. This means that the agent cannot acquire training samples, each of which consists of the state (or the state transition) and the corresponding reward. Consequently, the DRL approach and other data-driven approaches are not directly applicable to solving our MDP.

3. Proposed Optimal Policy

This section describes the proposed optimal policy. The basic concept of the derivation is similar to that in [31–33]. However, its direct extension is difficult for time-varying channels. This is because capturing time-variant channels using previously selected data symbols is difficult. To address this, we approximate the first-order Gaussian—Markov process and propose a computationally efficient algorithm.

We employ the approximation in [31–33] for the transition function, which is given by

$$\hat{\mathcal{T}}^{(a,j)}(\mathsf{S}_n) = \begin{cases} \theta_j[n], & j \in \mathcal{J}_a, a = 1, \\ 1, & j \in \mathcal{J}_a, a = 0, \end{cases} \tag{16}$$

where $\hat{\mathcal{T}}^{(a,j)}(\mathsf{S}_n) \to \mathcal{T}^{(a,j)}(\mathsf{S}_n)$ as $\theta_j[n] \to 1$.

The main difficulty in analyzing the time-varying channel model is solving element $\Delta[n]$. To resolve this difficulty, we approximate the first-order Gaussian—Markov process in (1) as follows:

$$\mathbf{H}[n] \approx \sqrt{1-\epsilon^2}\mathbf{H}[n-1], \tag{17}$$

where $H[n-1] \gg \Delta[n]$. This approximation is often adopted in studies because it provides analytical tractableness [36–38]. Using this approximation, the received symbol $\mathbf{z}[n+m]$ for $1 \leq m$ can be expressed in terms of $\mathbf{H}[n]$ as follows:

$$\begin{aligned} \mathbf{z}[n+m] &= \mathbf{H}^H[n+m]\mathbf{x}[n+m] + \mathbf{n}[n+m] \\ &\approx \mathbf{H}^H[n]\left(\sqrt{1-\epsilon^2}^m \mathbf{x}[n+m]\right) + \mathbf{n}[n+m], \end{aligned} \tag{18}$$

From approximation (18), the virtual state in [31] that mimics the optimal behavior from state $\mathsf{U}_{n+1}^{(a,j)}(\mathsf{S}_n)$ can be obtained as follows:

$$\tilde{\mathsf{U}}_m^{(a,j)}(\mathsf{S}_n) = \left(\mathbf{X}_m^{(a,j)}, \hat{\mathbf{X}}_m^{(a)}, \mathcal{C}_m^{(a)}\right), \tag{19}$$

where

$$\mathbf{X}_m^{(a,j)} = \begin{cases} \left[\mathbf{X}_n, \mathbf{x}_j, \left(\sqrt{1-\epsilon^2}\tilde{\mathbf{x}}[n+1]\right), \cdots, \left(\sqrt{1-\epsilon^2}^{m-n-1}\tilde{\mathbf{x}}[m-1]\right)\right], & a=1, \\ \left[\mathbf{X}_n, \left(\sqrt{1-\epsilon^2}\tilde{\mathbf{x}}[n+1]\right), \cdots, \left(\sqrt{1-\epsilon^2}^{m-n-1}\tilde{\mathbf{x}}[m-1]\right)\right], & a=0. \end{cases}$$

$$\hat{\mathbf{X}}_m^{(a)} = \begin{cases} \left[\hat{\mathbf{X}}_n, \hat{\mathbf{x}}[n], \left(\sqrt{1-\epsilon^2}\tilde{\mathbf{x}}[n+1]\right), \cdots, \left(\sqrt{1-\epsilon^2}^{m-n-1}\tilde{\mathbf{x}}[m-1]\right)\right], & a=1, \\ \left[\hat{\mathbf{X}}_n, \left(\sqrt{1-\epsilon^2}\tilde{\mathbf{x}}[n+1]\right), \cdots, \left(\sqrt{1-\epsilon^2}^{m-n-1}\tilde{\mathbf{x}}[m-1]\right)\right], & a=0. \end{cases}$$

$$\mathcal{C}_m^{(a)} = \begin{cases} [\mathcal{C} \cup \{n+1, \ldots, m\}], & a=1, \\ [\mathcal{C} \cup \{n+2, \ldots, m\}], & a=0. \end{cases}$$

The soft-decision symbol $\tilde{\mathbf{x}}[m]$ for $m \geq n+1$ is define as

$$\tilde{\mathbf{x}}[m] = \sum_{k=1}^{K} \theta_k[m]\mathbf{x}_j. \tag{20}$$

In (19), because $0 \leq \epsilon \leq 1$, the effect of soft decision symbol $\tilde{\mathbf{x}}[n+m]$ for estimating $\mathbf{H}[n]$ is diminished as m increases. Based on the virtual state, the state-action diagram for the proposed channel estimator is shown in Figure 2. In this figure, the number of state transitions at state S_n are one and K for $a=0$ and $a=1$, respectively. However, after $n+2$, the state transition is simplified to one because the virtual state mimics the behavior of state $\mathsf{U}_{n+1}^{(a,j)}(\mathsf{S}_n)$.

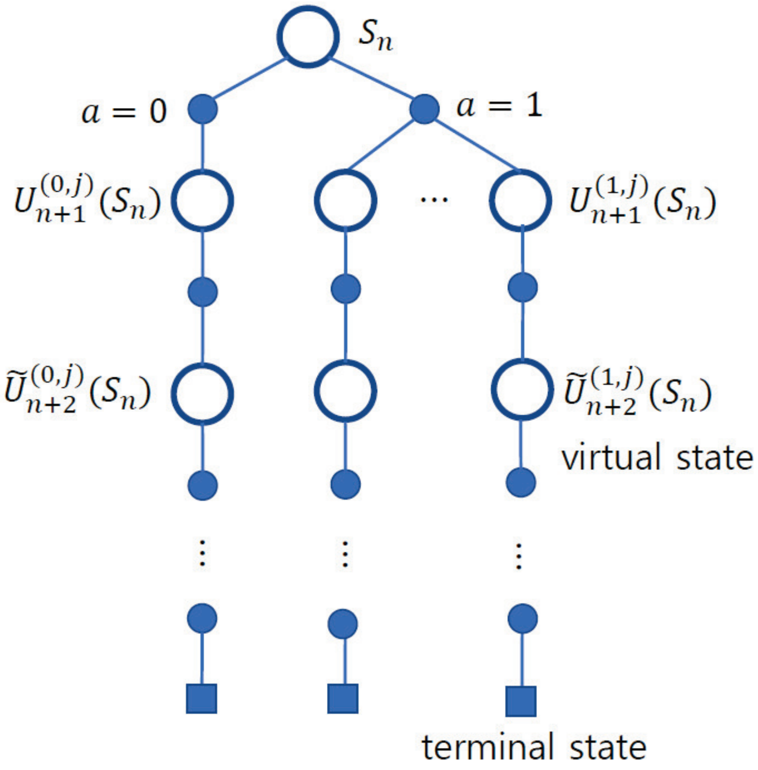

Figure 2. State-action diagram of the proposed channel estimator. After the time slot index $n+2$, the virtual state is applied such that the state transition is simplified.

Using the definition of virtual state (19), we can compute the future reward $\mathcal{V}^{\star}\left(\mathsf{U}_{n+1}^{(a,j)}(\mathsf{S}_n)\right)$ as

$$\mathcal{V}^{\star}\left(\mathsf{U}_{n+1}^{(a,j)}(\mathsf{S}_n)\right) \approx \mathcal{R}\left(\mathsf{U}_{n+1}^{(a,j)}(\mathsf{S}_n), \tilde{\mathsf{U}}_{n+2}^{(a,j)}(\mathsf{S}_n)\right) + \sum_{m=n+2}^{\mathcal{M}_d(N_d)} \mathcal{R}\left(\tilde{\mathsf{U}}_{m}^{(a,j)}(\mathsf{S}_n), \tilde{\mathsf{U}}_{m+1}^{(a,j)}(\mathsf{S}_n)\right). \tag{21}$$

By applying (13) to the future reward, the future reward is simplified as

$$\begin{aligned}\mathcal{V}^{\star}\left(\mathsf{U}_{n+1}^{(a,j)}(\mathsf{S}_n)\right) &= \mathrm{Tr}\left[\mathbf{B}\left(\mathsf{U}_{n+1}^{(a,j)}(\mathsf{S}_n)\right) - \mathbf{B}\left(\tilde{\mathsf{U}}_{n+2}^{(a,j)}(\mathsf{S}_n)\right) + \sum_{m=n+2}^{\mathcal{M}_d(N_d)} \mathbf{B}\left(\tilde{\mathsf{U}}_{m}^{(a,j)}(\mathsf{S}_n)\right) - \mathbf{B}\left(\tilde{\mathsf{U}}_{m+1}^{(a,j)}(\mathsf{S}_n)\right)\right] \\ &= \mathrm{Tr}\left[\mathbf{B}\left(\mathsf{U}_{n+1}^{(a,j)}(\mathsf{S}_n)\right) - \mathbf{B}\left(\tilde{\mathsf{U}}_{\mathcal{M}_d(N_d)+1}^{(a,j)}(\mathsf{S}_n)\right)\right].\end{aligned} \tag{22}$$

Using the approximations (16) and (22), the Q-value function in (14) is obtained as follows:

$$\mathcal{Q}(\mathsf{S}_n, a) = \sum_{j\in\mathcal{J}_a} \hat{\mathcal{T}}^{(a,j)}(\mathsf{S}_n)\mathrm{Tr}\left[\mathbf{B}(\mathsf{S}_n) - \mathbf{B}\left(\tilde{\mathsf{U}}_{\mathcal{M}_d(N_d)+1}^{(a,j)}(\mathsf{S}_n)\right)\right]. \tag{23}$$

The error covariance matrix $\mathbf{B}\left(\tilde{\mathsf{U}}_m^{(a,j)}\right)$ can be computed as

$$\begin{aligned}\mathbf{B}\left(\tilde{\mathsf{U}}_m^{(a,j)}\right) &= \mathcal{E}\{\|\hat{\mathbf{h}}_r\left(\tilde{\mathsf{U}}_m^{(a,j)}\right) - \mathbf{h}_r[n]\|^2\} \\ &= \mathcal{E}\{(\hat{\mathbf{h}}_r\left(\tilde{\mathsf{U}}_m^{(a,j)}\right) - \mathbf{h}_r[n])(\hat{\mathbf{h}}_r\left(\tilde{\mathsf{U}}_m^{(a,j)}\right) - \mathbf{h}_r[n])^H\} \\ &= \mathcal{E}\left\{\hat{\mathbf{h}}_r\left(\tilde{\mathsf{U}}_m^{(a,j)}\right)\hat{\mathbf{h}}_r^H\left(\tilde{\mathsf{U}}_m^{(a,j)}\right) - \mathbf{h}_r[n]\hat{\mathbf{h}}_r^H\left(\tilde{\mathsf{U}}_m^{(a,j)}\right) - \hat{\mathbf{h}}_r\left(\tilde{\mathsf{U}}_m^{(a,j)}\right)\mathbf{h}_r^H[n] + \mathbf{h}_r[n]\mathbf{h}_r^H[n]\right\} \\ &\overset{(a)}{=} \mathbf{Q}_m^{(a)}\hat{\mathbf{X}}_m^{(a,j)}\left(\left(\mathbf{X}_m^{(a,j)}\right)^H\mathbf{X}_m^{(a,j)} + \sigma_n^2\mathbf{I}_{|\mathcal{C}_m^{(a)}|}\right)(\hat{\mathbf{X}}_m^{(a,j)})^H\mathbf{Q}_m^{(a)} \\ &- \mathbf{X}_m^{(a,j)}(\hat{\mathbf{X}}_m^{(a,j)})^H\mathbf{Q}_m^{(a)} - \mathbf{Q}_m^{(a)}\hat{\mathbf{X}}_m^{(a,j)}\left(\mathbf{X}_m^{(a,j)}\right)^H + \mathbf{I}_{N_t} \\ &= \left(\mathbf{I}_{N_t} - \mathbf{Q}_m^{(a)}\hat{\mathbf{X}}_m^{(a,j)}\left(\mathbf{X}_m^{(a,j)}\right)^H\right)\left(\mathbf{I}_{N_t} - \mathbf{Q}_m^{(a)}\hat{\mathbf{X}}_m^{(a,j)}\left(\mathbf{X}_m^{(a,j)}\right)^H\right)^H + \sigma_n^2\mathbf{Q}_m^{(a)} - \sigma_n^4\left(\mathbf{Q}_m^{(a)}\right)^2 \\ &\overset{(b)}{=} \mathbf{Q}_m^{(a)}\mathbf{D}_m^{(a,j)}\left(\mathbf{D}_m^{(a,j)}\right)^H\mathbf{Q}_m^{(a)} + \sigma_n^2\mathbf{Q}_m^{(a)} - \sigma_n^4\left(\mathbf{Q}_m^{(a)}\right)^2\end{aligned} \tag{24}$$

where the distribution of $\mathbf{z}_r^H\left(\tilde{\mathsf{U}}_m^{(a,j)}(\mathsf{S}_n)\right)$ is given by $\mathcal{CN}\left(\mathbf{0}_{|\mathcal{C}_m^{(a)}|}, \left(\mathbf{X}_m^{(a,j)}\right)^H\mathbf{X}_m^{(a,j)} + \sigma_n^2\mathbf{I}_{|\mathcal{C}_m^{(a)}|}\right)$ and $\mathbf{Q}_m^{(a)} = \left(\hat{\mathbf{X}}_n\hat{\mathbf{X}}_n^H + \sum_{k=n+1}^{m-1}(1-\epsilon^2)^{m-n}\tilde{\mathbf{x}}[k]\tilde{\mathbf{x}}^H[k] + \sigma_n^2\mathbf{I}_{N_t}\right)^{-1}$ is applied in (a). $\mathbf{D}_m^{(a,j)} = (\mathbf{Q}_m^{(a)})^{-1} - \hat{\mathbf{X}}_m\mathbf{X}_m^H = \hat{\mathbf{X}}_m\left(\hat{\mathbf{X}}_m - \mathbf{X}_m\right)^H + \sigma_n^2\mathbf{I}_{N_t}$ is used in (b).

By applying (24) to the Q-value function, the optimal policy at S_n is computed as

$$\begin{aligned}\pi^{\star}(\mathsf{S}_n) &= \underset{a\in\{0,1\}}{\mathrm{argmax}}\ \mathcal{Q}(\mathsf{S}_n, a) = \mathcal{I}[(\mathcal{Q}(\mathsf{S}_n,1) - \mathcal{Q}(\mathsf{S}_n,0)) \geq 0] \\ &= \mathcal{I}\left[\mathrm{Tr}\left[\left(\sum_{j=1}^{K}\mathbf{B}\left(\tilde{\mathsf{U}}_{\mathcal{M}_d(N_d)+1}^{(0,0)}(\mathsf{S}_n)\right) - \theta_j[n]\mathbf{B}\left(\tilde{\mathsf{U}}_{\mathcal{M}_d(N_d)+1}^{(1,j)}(\mathsf{S}_n)\right)\right)\right] \geq 0\right].\end{aligned} \tag{25}$$

where $\mathbf{B}\left(\tilde{\mathsf{U}}^{(a,j)}_{\mathcal{M}_d(N_d)+1}(\mathsf{S}_n)\right) = \sigma_n^2\mathbf{Q}^{(a)} - \sigma_n^4\left(\mathbf{Q}^{(a)}\right)^2 + \mathbf{Q}^{(a)}\mathbf{D}^{(a,j)}\left(\mathbf{D}^{(a,j)}\right)^H\mathbf{Q}^{(a)}$. $\mathbf{Q}^{(a)} = \mathbf{Q}^{(a)}_{\mathcal{M}_d(N_d)+1}$ and $\mathbf{D}^{(a,j)} = \mathbf{D}^{(a,j)}_{\mathcal{M}_d(N_d)+1}$ are defined as

$$\begin{aligned}
\mathbf{Q}^{(a)} &= \left(\hat{\mathbf{X}}^{(a)}_{\mathcal{M}_d(N_d)+1}\left(\hat{\mathbf{X}}^{(a)}_{\mathcal{M}_d(N_d)+1}\right)^H + \sigma_n^2\mathbf{I}_{N_{\mathrm{tx}}}\right)^{-1} \\
&\overset{(a)}{=} \begin{cases} \left(\hat{\mathbf{X}}_n\hat{\mathbf{X}}_n^H + \sum\limits_{m=n+1}^{\mathcal{M}_d(N_d)}(1-\epsilon^2)^{m-n}\tilde{\mathbf{x}}[m]\tilde{\mathbf{x}}^H[m] + \sigma_n^2\mathbf{I}_{N_t}\right)^{-1}, & a = 0, \\ \left(\left(\mathbf{Q}^{(0)}\right)^{-1} + \tilde{\mathbf{x}}[n]\tilde{\mathbf{x}}^H[n]\right)^{-1}, & a = 1. \end{cases} \\
\mathbf{D}^{(a,j)} &= \hat{\mathbf{X}}^{(a)}_{\mathcal{M}_d(N_d)+1}\left(\hat{\mathbf{X}}^{(a)}_{\mathcal{M}_d(N_d)+1} - \mathbf{X}^{(a,j)}_{\mathcal{M}_d(N_d)+1}\right)^H + \sigma_n^2\mathbf{I}_{N_t} \\
&\overset{(b)}{=} \begin{cases} \hat{\mathbf{X}}_n\left(\hat{\mathbf{X}}_n - \mathbf{X}_n\right)^H + \sigma_n^2\mathbf{I}_{N_t}, & j \in \mathcal{J}_a, a = 0, \\ \mathbf{D}^{(0,0)} + \hat{\mathbf{x}}[n]\left(\hat{\mathbf{x}}[n] - \mathbf{x}_j\right)^H, & j \in \mathcal{J}_a, a = 1. \end{cases}
\end{aligned}$$

Similar to [31], $\mathbf{Q}^{(1)}$ and $\mathbf{Q}^{(0)}$ satisfy $\mathbf{Q}^{(1)} = \mathbf{Q}^{(0)} - \frac{\mathbf{Q}^{(0)}\hat{\mathbf{x}}[n]\hat{\mathbf{x}}^H[n]\mathbf{Q}^{(0)}}{1+\hat{\mathbf{x}}^H[n]\mathbf{Q}^{(0)}\hat{\mathbf{x}}[n]}$. In addition, $\mathbf{D}^{(1,j)}$ and $\mathbf{D}^{(0,0)}$ satisfy $\sum\limits_{j=1}^{K}\theta_j[n]\mathbf{D}^{(1,j)}\left(\mathbf{D}^{(1,j)}\right)^H = \left(\mathbf{D}^{(0,0)} + \hat{\mathbf{d}}_n\right)\left(\mathbf{D}^{(0,0)} + \hat{\mathbf{d}}_n\right)^H + \delta_n\hat{\mathbf{x}}[n]\hat{\mathbf{x}}^H[n]$ where $\hat{\mathbf{d}}_n = \hat{\mathbf{x}}[n](\hat{\mathbf{x}}[n] - \tilde{\mathbf{x}}[n])^H$, and $\delta_n = \sum\limits_{j=1}^{K}\theta_j[n]\|\hat{\mathbf{x}}[n] - \mathbf{x}_j\|^2 - \|\hat{\mathbf{x}}[n] - \tilde{\mathbf{x}}[n]\|^2$.

Finally, similar to [32], by applying the results in (23) and (24) to (25), we obtain the proposed optimal policy in closed-form as

$$\pi^\star(\mathsf{S}_n) = \mathcal{I}\left[\frac{\sigma_n^2(1+\alpha_n) + \sigma_n^4\|\mathbf{a}_n\|^2 + \|\mathbf{d}_n\|^2}{2\sigma_n^4\beta_n + \gamma_n + \|\mathbf{c}_n - \mathbf{b}_n + \mathbf{d}_n\|^2} \geq 1\right]. \tag{26}$$

When we define $\mathbf{Q} = \mathbf{Q}^{(0)}$ and $\mathbf{D} = \mathbf{D}^{(0,0)}$, vectors are computed as $\mathbf{a}_n = \frac{\mathbf{Q}\hat{\mathbf{x}}[n]}{\sqrt{1+\alpha_n}}$, $\mathbf{b}_n = \mathbf{D}^H\mathbf{b}_n$, $\mathbf{c}_n = \frac{\hat{\mathbf{x}}[n]-\tilde{\mathbf{x}}[n]}{\sqrt{1+\alpha_n}}$, and $\mathbf{d}_n = \frac{\mathbf{D}^H\mathbf{Q}\mathbf{a}_n}{\|\mathbf{a}_n\|^2}$. In addition, the constants are computed as $\alpha_n = \hat{\mathbf{x}}^H[n]\mathbf{Q}\hat{\mathbf{x}}[n]$, $\beta_n = \frac{\mathbf{a}_n^H\mathbf{Q}\mathbf{a}_n}{\|\mathbf{a}_n\|^2}$, and $\gamma_n = \frac{\delta_n}{1+\alpha_n}$. Note that the expression of the optimal policy in (26) is similar to that in [32]. However, the vectors and constants in the optimal policy is different from those in [32] because the temporal correlation ϵ is considered in $\mathbf{Q}$ and $\mathbf{D}$. When $\epsilon = 0$, the optimal policy in (26) is equivalent to that in [32].

4. Further Performance Improvement

In this section, we propose a practical method to improve the estimation accuracy of the proposed channel estimator. The proposed method refines state elements to capture the time-varying nature of the channel.

4.1. State Element Refinement

Elements $\mathbf{X}_n$ and $\hat{\mathbf{X}}_n$ in state S_n are updated when the detected data symbol is selected based on the optimal policy. However, the elements gradually lose their effectiveness in estimating $\mathbf{H}[n]$ as time slot index n increases. To address this, we first represent the received symbol for $1 \leq m$ in terms of $\mathbf{H}[n]$ as

$$\begin{aligned}
\mathbf{z}[n-m] &= \mathbf{H}^H[n-m]\mathbf{x}[n-m] + \mathbf{n}[n-m] \\
&\approx \mathbf{H}^H[n]\left(\sqrt{1-\epsilon^2}^{\,-m}\mathbf{x}[n-m]\right) + \mathbf{n}[n-m].
\end{aligned} \tag{27}$$

Using (27), we refine the elements $\mathbf{X}_n$ and $\hat{\mathbf{X}}_n$ in state as the time slot index increases, which is given by

$$\begin{aligned} \mathbf{X}_n &\leftarrow \sqrt{1-\epsilon^2}^{-1}\mathbf{X}_n \\ \hat{\mathbf{X}}_n &\leftarrow \sqrt{1-\epsilon^2}^{-1}\hat{\mathbf{X}}_n. \end{aligned} \tag{28}$$

Regardless of the above refinement, the previously selected data symbols lose their effectiveness as the time slot index increases, particularly for large data lengths. This is because the term $\Delta[n]$ in (1) becomes dominant, increasing the uncertainty in estimating the channel. To overcome this, we remove too-old selected data symbols in state by introducing a window size N_w. In other words, we maintain the size of the set of time slot indices as $|\mathcal{C}| = N_w$. Thus, when the optimal action is one at time slot index n, n is included, whereas the first index $\mathcal{C}(1)$ is removed from set $\mathcal{C}$, which can be expressed as

$$\begin{aligned} \mathbf{X}_n &\leftarrow \mathbf{X}_n \setminus \mathbf{X}_n[\mathcal{C}(1)], \\ \hat{\mathbf{X}}_n &\leftarrow \hat{\mathbf{X}}_n \setminus \hat{\mathbf{X}}_n[\mathcal{C}(1)], \\ \mathcal{C} &\leftarrow \mathcal{C} \setminus \mathcal{C}(1). \end{aligned} \tag{29}$$

4.2. Algorithm

Using the proposed optimal policy and performance improvement strategy, the proposed channel estimator is summarized in Algorithm 1. The receiver obtains the initial channel estimation during pilot transmission. Subsequently, during the data transmission, the receiver sequentially selects a data symbol based on the optimal policy. When the optimal action $a^\star = 1$, the state S_n is updated using the most-probable state transition [31]. In addition, the state element refinement is performed based on this condition. After each data block ends, the channel estimate is updated using the state S_n.

Algorithm 1: Proposed channel estimator

```
Obtain the initial channel estimate $\mathbf{H} \leftarrow \hat{\mathbf{h}} = \left[\hat{\mathbf{h}}_1, \cdots, \hat{\mathbf{h}}_{N_r}\right]$ from (3)
Initialize the state $\mathsf{S}_1 = (\mathbf{X}^p, \mathbf{X}^p, \phi)$.
for $d = 1$ to $M_d$ do
    for $n \in \mathcal{M}_d$ do
        Compute the optimal policy $a^\star = \pi^\star(\mathsf{S}_n)$ from (26).
        Set the optimal values $j^\star = 0$ for $a^\star = 0$ and $\mathbf{x}_{j^\star} = \hat{\mathbf{x}}[n]$ for $a^\star = 1$.
        Update $\mathsf{S}_{n+1} \leftarrow \mathsf{U}_{n+1}^{(a^\star, j^\star)}(\mathsf{S}_n)$ from (12).
        if $a^\star == 1$ and $N_w < |\mathcal{C}|$ then
            Remove the state elements in $\mathsf{S}_{n+1}$.
            $\mathbf{X}_{n+1} \leftarrow \mathbf{X}_{n+1} \setminus \mathbf{X}_{n+1}[\mathcal{C}(1)]$,
            $\hat{\mathbf{X}}_{n+1} \leftarrow \hat{\mathbf{X}}_{n+1} \setminus \hat{\mathbf{X}}_n[\mathcal{C}(1)]$,
            $\mathcal{C} \leftarrow \mathcal{C} \setminus \mathcal{C}(1)$.
        end
        Refine the state elements in $\mathbf{X}_{n+1} \leftarrow \sqrt{1-\epsilon^2}^{-1}\mathbf{X}_{n+1}$ and
         $\hat{\mathbf{X}}_{n+1} \leftarrow \sqrt{1-\epsilon^2}^{-1}\hat{\mathbf{X}}_{n+1}$.
    end
    Update the channel estimate $\mathbf{H} \leftarrow \hat{\mathbf{h}} = \left[\hat{\mathbf{h}}_1(\mathsf{S}_n), \cdots, \hat{\mathbf{h}}_{N_r}(\mathsf{S}_n)\right]$ from (10).
end
```

In Figure 3, we show a block diagram of the proposed channel estimator, which consists of the LMMSE channel estimator, optimal policy calculator, and state element refinement. The LMMSE channel estimator obtains the initial estimate at pilot transmission and updates the estimate at data transmission using state S_n. The optimal policy calculator obtains the optimal action of (26) from the channel estimates and APP from the data detector. The state elements are then refined based on the obtained optimal action, and the refined state is used to estimate the channel and optimal policy for the next step.

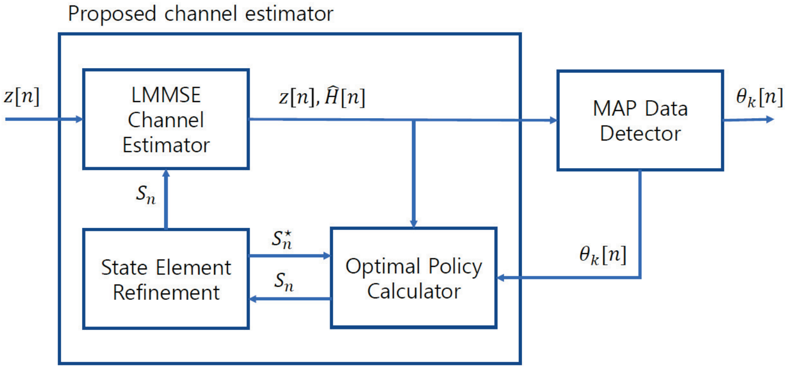

Figure 3. Proposed channel estimator using a further performance improvement strategy.

Application of other data detection: The proposed RL-aided channel estimator can be universally applied to any other soft-output data detection method. To achieve this, the proposed RL-aided channel estimator relies on the availability of APPs, which can be directly derived from the MAP data detection method. In the case of using other soft-output data detections, the proposed RL-aided channel estimator can utilize the APPs that are computed from the log-likelihood ratios.

Complexity analysis: Complexity is analyzed in terms of real multiplications to provide an implementation perspective. Figure 3 shows the hardware structure of the proposed RL-aided channel estimator, which consists of the LMMSE channel estimator, state element refinement, and optimal policy calculator. Because the exact complexity can vary depending on the implementation details, the complexity order ($\mathcal{O}(\cdot)$) of each component is analyzed.

The complexity order of the LMMSE channel estimator in (7) is $\mathcal{O}((N_p + |\mathcal{C}(\mathbf{a}^\star)|)(N_t^2 + N_t N_r))$ where $\mathcal{C}(\mathbf{a}^\star)$ is the set of selected data symbol vectors. The complexity order of state element refinement in Section 4.1 is $\mathcal{O}(4(N_p + |\mathcal{C}(\mathbf{a}^\star)|)$. The complexity of the optimal policy in (25) is primarily determined by the computation of $\mathbf{Q}^{(a)}$. Consequently, the complexity order of the optimal policy in (25) is $\mathcal{O}\big(2N_t^2 T_d^2\big)$. It is important to note that among the components of the proposed channel estimator, the optimal policy calculator has the highest complexity because it performs every data symbol index n, while the other components perform every data block index d.

5. Simulation Results

This section presents the effectiveness of the proposed channel estimator using simulations. The numbers of transmit and receive antennas used were $N_t = 2$ and $N_r = 4$. The transmission frame consisted of one pilot block with $N_p = 8$ symbols and $M_d = 20$ data blocks with $N_d = 128$ symbols. Each symbol used 4-quadrature amplitude modulation (QAM) symbol mapping. We adopted turbo channel code with a rate of 1/2 and 16 cyclic redundancy check bits. For the proposed channel estimator, the window size was set to $N_w = 2 \times N_d$. The signal-to-noise ratio (SNR) was defined as $E_b/N_0 = 1/(\log_2|\mathcal{X}|\sigma_n^2)$ under the power constraint $\mathbb{E}\{\|\mathbf{x}[n]\|^2\} = 1$. The proposed channel estimator was compared with the following methods.

- PCSI: This method is ideal for time-invariant channels in which a perfect initial channel estimate is available at the receiver. Because the initial channel changes during data transmission, it is not optimal for time-varying channels.
- Pilot: This method uses a conventional pilot-aided channel estimator using (3).

- Soft: This method is a data-aided channel estimator when all symbols in (20) are used as additional pilot symbols.
- Conv-RL [31]: This method is a data-aided channel estimator in which the detected data symbol is selected using the RL approach developed for time-invariant channels.

The performance of the methods was compared with that of the proposed channel estimator in terms of the block-error rate (BLER) and normalized MSE (NMSE). In addition, we considered the time-invariant channel $\epsilon = 0$ and time-variant channel with $\epsilon = 0.005$ and $\epsilon = 0.01$. Note that channel was more severely variant when $\epsilon = 0.01$ than when $\epsilon = 0.005$.

Figure 4 shows the BLERs for the proposed and other channel estimators in the time-invariant channel, i.e., $\epsilon = 0$. The conventional pilot-aided channel estimator exhibited a poor performance when the number of pilots was small. Data-aided channel estimators can overcome performance degradation caused by pilot-aided channel estimators. In particular, the RL-based channel estimator [31] showed an outstanding performance compared with other channel estimators. The BLER of the proposed channel estimator was slightly worse than that of [31] because of a reduced window size $N_w = 2 \times N_d$.

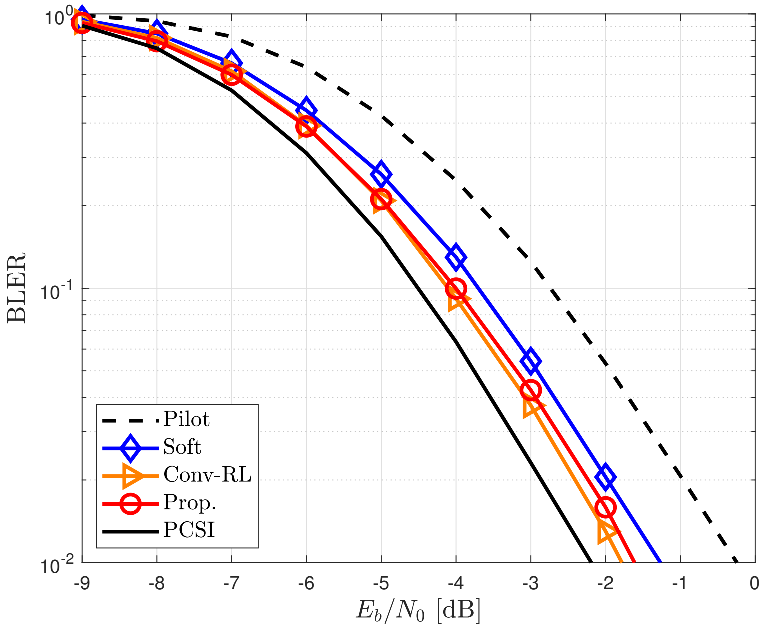

Figure 4. BLER for different channel estimators in time-invariant channels.

In Figure 5, the proposed channel estimator is compared with other channel estimators in time-varying channels. The proposed channel estimator had a better BLER improvement than the conventional pilot-aided channel estimator. In particular, the performance improvement is more prominent at $\epsilon = 0.01$ than that at $\epsilon = 0.005$. This is because the proposed channel estimator can efficiently capture channel variations by selecting and refining detected data symbols. In addition, in time-variant channels, the proposed channel estimator had a slightly higher BLER than the RL-based channel estimator, primarily due to the utilization of a reduced window size (see Figure 5). However, in time-varying channels, this reduction in window size actually contributed to an improvement in BLER by effectively leveraging the most recent data symbols. Consequently, the proposed channel estimator had a lower BLER compared to the RL-based channel estimator. In Figure 6, we show the BLER of the proposed channel estimator for different window sizes N_w in time-varying channels with $\epsilon = 0.01$ The BLER of the proposed channel estimator gradually degraded as N_w increased. This is because the selected data symbol is undesirable as an additional pilot symbol in fast fading channels; therefore, only the usage of the latest selected data symbol can improve the performance.

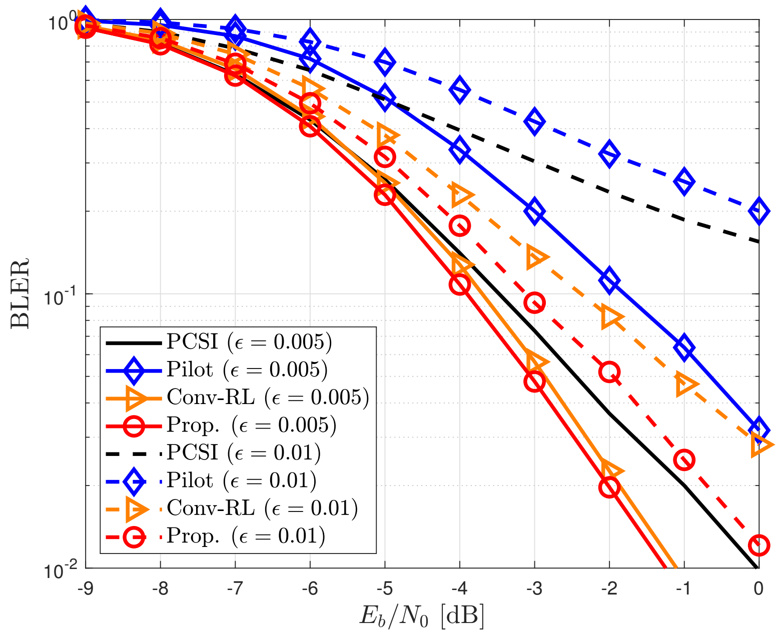

Figure 5. BLER for different channel estimators in time-varying channels ($\epsilon = 0.005$ and $\epsilon = 0.01$).

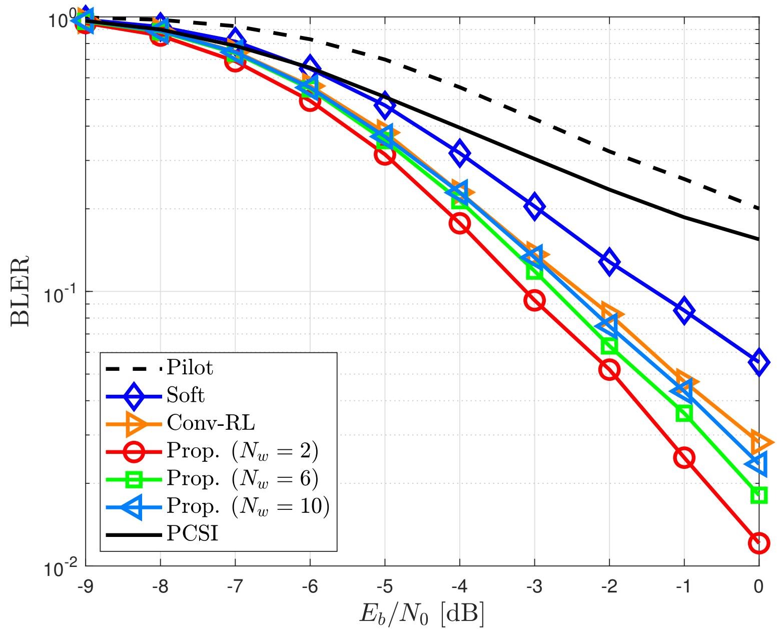

Figure 6. BLER of the proposed channel estimator for different window sizes N_w in time-varying channels with $\epsilon = 0.01$.

To further investigate the effect of window size, we investigated the NMSE of the proposed channel estimator for different window sizes N_w at $\epsilon = 0.005$ and $E_b/N_0 = -2$ dB (Figure 7). We observed that the NMSE improved until $M_d = 2$ but decreased as the data block length increased. This is because the old selected data symbol is ineffective for estimating the channel. Thus, when we discard the old data symbol, we can further improve the estimation accuracy (Figure 7).

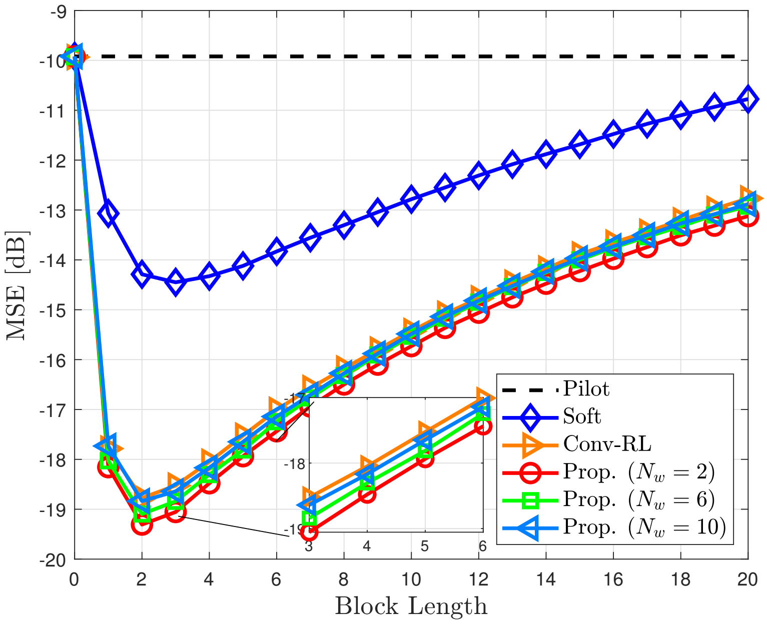

Figure 7. NMSE of the proposed channel estimator for different window sizes N_w in time-varying channels with $\epsilon = 0.005$.

6. Conclusions

A data-aided channel estimator was proposed for time-varying channels, which involves selecting the detected data symbol. To facilitate efficient selection of the detected data symbol, an optimization problem was initially formulated to minimize the channel estimation error. Subsequently, the MDP for this optimization problem was formulated, and its optimal policy was derived using an RL algorithm. In the derivation process, approximations for the transition probability and a first-order Gaussian–Markov process were utilized. To improve estimation accuracy, a state element refinement was introduced to capture the time-varying nature of the channel by incorporating a window size. Simulation results demonstrated that the proposed channel estimator provides similar performance to the conventional RL-based channel estimator in time-invariant channels when $\epsilon = 0$, while showing improved performance in time-varying channels when $\epsilon = 0.01$ and $\epsilon = 0.005$ compared to conventional RL-based channel estimator.

An interesting direction for further research involves optimizing the frame structure in terms of the spectral efficiency. In this study, the frame structure comprises one pilot and D data block. The proposed RL-aided channel estimator is applied to the data blocks to capture the time-varying nature of the channel. However, in fast fading channels, it can be challenging for the proposed channel estimator to accurately track channel variations. In such cases, reducing the value of D in the frame structure can potentially improve the performance. However, this reduction also leads to a degradation in spectral efficiency. To find an appropriate value for D in time-varying channels, an optimization problem that maximizes spectral efficiency while maintaining acceptable performance levels becomes a suitable criterion. To address this, one approach is to first derive the performance of the RL-aided channel estimator. Subsequently, the solution to the optimization problem can be obtained using the derived performance.

Author Contributions: Conceptualization, M.M.; Methodology, T.-K.K.; Software, T.-K.K.; Validation, M.M.; Formal analysis, T.-K.K.; Investigation, T.-K.K.; Resources, T.-K.K.; Data curation, T.-K.K.; Writing—original draft, T.-K.K.; Writing—review & editing, M.M.; Visualization, M.M.; Supervision, M.M.; Project administration, M.M.; Funding acquisition, M.M. All authors have read and agreed to the published version of the manuscript.

Funding: The work of Tae-Kyoung Kim was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MIST) (No. 2021R1F1A1063273). The work of Moonsik Min was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea government (MSIT) (No. 2023R1A2C1004034), and in part by the BK21 FOUR Project funded by the Ministry of Education, Korea (4199990113966).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Goldsmith, A.; Jafar, S.A.; Jindal, N.; Vishwanath, S. Capacity Limits of MIMO Channels. *IEEE J. Sel. Commun.* **2003**, *21*, 684–702. [CrossRef]
2. Zheng, L.; Tse, D.N.C. Diversity and Multiplexing: A Fundamental Tradeoff in Multiple-Antenna Channels. *IEEE Trans. Inf. Theory* **2003**, *49*, 1073–1096. [CrossRef]
3. Paulraj, A.J.; Gore, D.A.; Nabar, R.U.; Bolcskei, H. An Overview of MIMO Communications-a Key to Gigabit Wireless. *Proc. IEEE* **2004**, *92*, 198–218. [CrossRef]
4. Sanayei, S.; Nosratinia, A. Antenna Selection in MIMO Systems. *IEEE Commun. Mag.* **2004**, *42*, 68–73. [CrossRef]
5. Larsson, E.G.; Edfors, O.; Tufvesson, F.; Marzetta, T.L. Massive MIMO for Next Generation Wireless Systems. *IEEE Commun. Mag.* **2014**, *52*, 186–1954. [CrossRef]
6. Zheng, K.; Zhao, L.; Mei, J.; Shao, B.; Xiang, W.; Hanzo, L. Survey of Large-Scale MIMO Systems. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 1738–1760. [CrossRef]
7. Yang, S.; Hanzo, L. Fifty Years of MIMO Detection: The Road to Large-Scale MIMOs. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 1941–1988. [CrossRef]
8. Morelli, M.; Mengali, U. A Comparison of Pilot-Aided Channel Estimation Methods for OFDM System. *IEEE Trans. Signal Process.* **2001**, *49*, 3065–3073. [CrossRef]
9. Coleri, S.; Ergen, M.; Puri, A.; Bahai, A. Channel Estimation Techniques Based on Pilot Arrangement in OFDM Systems. *IEEE Trans. Broadcast.* **2002**, *48*, 223–229. [CrossRef]
10. Mostofi, Y.; Cox, D.C. ICI Mitigation for Pilot-Aided OFDM Mobile Systems. *IEEE Trans. Wirel. Commun.* **2005**, *4*, 765–774. [CrossRef]
11. Biguesh, M.; Gershman, A.B. Training-based MIMO Channel Estimation: A Study of Estimator Tradeoffs and Optimal Training Signals. *IEEE Trans. Signal Process.* **2006**, *54*, 884–893. [CrossRef]
12. Ozdemir, M.K.; Arslan, H. Channel Estimation for Wireless OFDM Systems. *IEEE Commun. Surv. Tutor.* **2007**, *9*, 18–48. [CrossRef]
13. Soltani, M.; Pourahmadi, V.; Mirzaei, A.; Sheikhzadeh, H. Deep Learning-based Channel Estimation. *IEEE Commun. Lett.* **2019**, *23*, 652–655. [CrossRef]
14. Le, H.A.; Van Chien, T.; Nguyen, T.H.; Choo, H.; Nguyen, V.D. Machine Learning-Based 5G-and-Beyond Channel Estimation for MIMO-OFDM Communication Systems. *Sensors* **2021**, *21*, 4861. [CrossRef]
15. Yuan, J.; Ngo, H.Q.; Matthaiou, M. Machine Learning-Based Channel Prediction in Massive MIMO with Channel Aging. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 2960–2973. [CrossRef]
16. Valenti, M.C.; Woerner, B.D. Iterative Channel Estimation and Decoding of Pilot Symbol Assisted Turbo Codes over Flat-Fading Channels. *IEEE J. Sel. Commun.* **2001**, *19*, 1697–1705. [CrossRef]
17. Dowler, A.; Nix, A.; McGeehan, J. Data-derived Iterative Channel Estimation with Channel Tracking for a Mobile Fourth Generation Wide Area OFDM System. In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM), San Francisco, CA, USA, 1–5 December 2003.
18. Cozzo, C.; Hughes, B.L. Joint Channel Estimation and Data Detection in Space-Time Communications. *IEEE Trans. Commun.* **2003**, *51*, 1266–1270. [CrossRef]
19. Song, S.; Singer, A.C.; Sung, K.M. Soft Input Channel Estimation for Turbo Equalization. *IEEE Trans. Signal Process.* **2004**, *52*, 2885–2894. [CrossRef]
20. Nicoli, M.; Ferrara, S.; Spagnolini, U. Soft-Iterative Channel Estimation: Methods and Performance Analysis. *IEEE Trans. Signal Process.* **2007**, *55*, 2993–3006. [CrossRef]
21. Zhao, M.; Shi, Z.; Reed, M.C. Iterative Turbo Channel Estimation for OFDM System over Rapid Dispersive Fading Channel. *IEEE Trans. Wirel. Commun.* **2008**, *7*, 3174–3184. [CrossRef]
22. Guo, Q.; Ping, L.; Huang, D. A Low-Complexity Iterative Channel Estimation and Detection Technique for Doubly Selective Channels. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 4340–4349.
23. Ma, J.; Ping, L. Data-Aided Channel Estimation in Large Antenna Systems. *IEEE Trans. Signal Process.* **2014**, *62*, 3111–3124.
24. Wen, C.K.; Wang, C.J.; Jin, S.; Wong, K.K.; Ting, P. Bayes-Optimal Joint Channel-and-Data Estimation for Massive MIMO with Low-Precision ADCs. *IEEE Trans. Signal Process.* **2015**, *64*, 2541–2556. [CrossRef]

25. Park, S.; Shim, B.; Choi, J.W. Iterative Channel Estimation Using Virtual Pilot Signals for MIMO-OFDM Systems. *IEEE Trans. Signal Process.* **2015**, *63*, 3032–3045. [CrossRef]
26. Huang, C.; Liu, L.; Yuen, C.; Sun, S. Iterative Channel Estimation Using LSE and Sparse Message Passing for mmWave MIMO Systems. *IEEE Trans. Signal Process.* **2018**, *67*, 245–259. [CrossRef]
27. Li, X.; Wang, Q.;Yang, H.; Ma, X. Data-Aided MIMO Channel Estimation by Clustering and Reinforcement-Learning. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Austin, TX, USA, 10–13 April 2022.
28. Naeem, M.; De Pietro, G.; Coronato, A. Application of Reinforcement Learning and Deep Learning in Multiple-Input and Multiple-Output (MIMO) Systems. *Sensors* **2022**, *22*, 309. [CrossRef]
29. Oh, M.S.; Hosseinalipour, S.; Kim, T.; Brinton, C.G.; Love, D.J. Channel Estimation via Successive Denoising in MIMO OFDM Systems: A Reinforcement Learning Approach. In Proceedings of the IEEE International Conference on Communications (ICC), Montreal, QC, Canada, 14–23 June 2021.
30. Chu, M.; Liu, A.; LAu, V.K.N.; Jiang, C.; Yang, T. Deep Reinforcement Learning based End-to-End Multi-User Channel Prediction and Beamforming. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 10271–10285. [CrossRef]
31. Jeon, Y.S.; Li, J.; Tavangaran, N.; Poor, H.V. Data-Aided Channel Estimator for MIMO Systems via Reinforcement Learning. In Proceedings of the IEEE International Conference on Communications (ICC), Prayagraj, India, 27–29 November 2020.
32. Kim, T.K.; Min, M. A Low-Complexity Algorithm for Reinforcement Learning-Based Channel Estimator for MIMO Systems. *Sensors* **2022**, *21*, 4379. [CrossRef]
33. Kim, T.K.; Jeon, Y.S.; Li, J.; Tavangaran, N.; Poor, H.V. Semi-Data-Aided Channel Estimation for MIMO Systems via Reinforcement Learning. *IEEE Trans. Wirel. Commun.* 2022, *early access*. [CrossRef]
34. Dong, M.; Tong, L.; Sadler, B.M. Optimal Insertion of Pilot Symbols for Transmissions over Time-Varying Flat Fading Channels. *IEEE Trans. Signal Process.* **2004**, *52*, 1403–1418. [CrossRef]
35. Kim, T.K.; Jeon, Y.S.; Min, M. Training Length Adaptation for Reinforcement Learning-Based Detection in Time-Varying Massive MIMO Systems With One-Bit ADCs. *IEEE Trans. Veh. Technol.* **2021**, *70*, 6999–7011. [CrossRef]
36. Li, C.C.; Lin, Y.P. Predictive Coding of Bit Loading for Time Correlated MIMO Channels with A Decision Feedback Receiver. *IEEE Trans. Signal Process.* **2015**, *63*, 3376–3386. [CrossRef]
37. Kim, H.; Yu, H.; Lee, Y. Limited Feedback for Multicell Zero-Forcing Coordinated Beamforming in Time-Varying Channels. *IEEE Trans. Veh. Technol.* **2015**, *64*, 2349–2359. [CrossRef]
38. Mirza, J.; Dmochowski, P.A.; Smith, P.J.; Shafi, M. A Differential Codebook with Adaptive Scaling for Limited Feedback MU MISO Systems. *IEEE Wirel. Commun. Lett.* **2014**, *3*, 2–5. [CrossRef]
39. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; The MIT Press: Cambridge, MA, USA, 2018.

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