Track 1: AI-based Joint Channel Estimation and Channel State Information Feedback

Organizer: CAICT OPPO

Background

Accurate channel estimation (CE) and effective channel state information (CSI) feedback are the basic conditions to ensure the performance of wireless communication system. These two important topics were separately designed and studied in previous competitions, but will be jointly considered in the 3rd Wireless Communication AI Competition. Track "AI-based Joint Channel Estimation and Channel State Information Feedback" mainly focuses on joint design of channel estimation and CSI feedback, and constructs the corresponding competitive theme.

The deep integration of wireless communication and artificial intelligence (AI) has become an important direction for the construction of future wireless communication systems. In the evolution stage of 5G, utilizing AI to solve the specific modularization problems of the wireless communication system by following the existing architecture has become a popular research topic. For instance, many researches from academic and industry area have improved the performance of channel estimation and CSI feedback separately by introducing AI technology. At the same time, focusing on the impact of AI on the future communication systems, researchers have once again obtained the opportunity to relax or even break some inherent constraints, in order to realize a new generation of wireless communication systems which oriented to intelligence and built on intelligence. Reconstructing the modular design of the wireless communication systems can be carefully considered, such as AI-based end-to-end communication system design, joint design of channel estimation and feedback, joint design of data scheduling and detection, etc.

This competition takes the received pilot signal as input and optimal CSI feedback as target output. The participants will have great degree of freedom to explore joint design solutions for AI-based channel estimation and channel state information feedback, in order to achieve good final performance.

Task

In this competition, we consider to make use of the information compression, nonlinear recovery and end-to-end optimization capability of AI, to takes the received pilot signal (CSI-RS) on small number of time-frequency resources at user equipment (UE) side as the input, and strive to realize the recovery of channel state information as accurately as possible at base station (BS) side, so that enabling the construction of high-quality wireless communication link.

Specifically, the characteristics of this competition are as follows:

a. Following the practical requirements of wireless communication system, the competition focuses on the improvement of CSI feedback performance based on received pilot signal with limited number.

b. The solution design with two different pilot densities (sufficient and insufficient pilot) are required, and the performance will be evaluated separately.

c. In the traditional communication module partitioning, channel estimation and CSI feedback are independently designed and optimized. With the empowerment of AI, there are actually a variety of potential implementation methods to improve the overall performance, such as based on integrated design or modular design. Participants can freely select the most appropriate implementation method to achieve the best performance.

Dataset

The dataset is generated under the MIMO channel environment of 32 transmitting antennas and 4 receiving antennas (32T4R), which face to channel estimation and channel state information feedback. Participants will obtain the following types of data for the solution design and verification:

- Received pilot signal (CSI-RS) at UE side

CSI-RS only occupies limited time-frequency resources, but need to support channel measurement between multiple transmitting and receiving antenna ports. Following the classic signal transmission equation, y=hx+n, in which y represent the received signal, h represent the wireless channel, x represent the transmission signal and n represent the noise. For the received signal y on CSI-RS resources, x is the known CSI-RS sequence, and then h, which is the channel information on CSI-RS resources, could be estimated based on y and x. Further, the channel information on other time-frequency resources can be estimated by interpolation, so as to achieve more accurate channel state information feedback.

In this competition, there are in total 624 subcarriers in the frequency domain, every 12 subcarriers belong to a resource block, and every 4 resource blocks constitute a subband. Within one resource block, which corresponding to 12 subcarriers and 14 OFDM symbols, the pattern of CSI-RS is illustrated as Figure 1. In order to realize channel measurement between multiple antenna ports, CSI-RS corresponding to different transmitting antenna ports (TX) need to be orthogonalized via time division, frequency division, and code division. As can be seen from Figure 1, the time-frequency resources occupied by the 32 CSI-RSs are divided into 16 groups. On each group of time-frequency resources, only two TX transmit signals and the rest are silent. And, the signals between these two TX achieve resource reuse based on orthogonal cover codes. For example, the CSI-RS sequences sent by TX 1 and 2 on the two shared time-frequency resources are [+1,+1] and [+1,-1] respectively, and the other transmitting antenna ports sharing time-frequency resources following the same way.

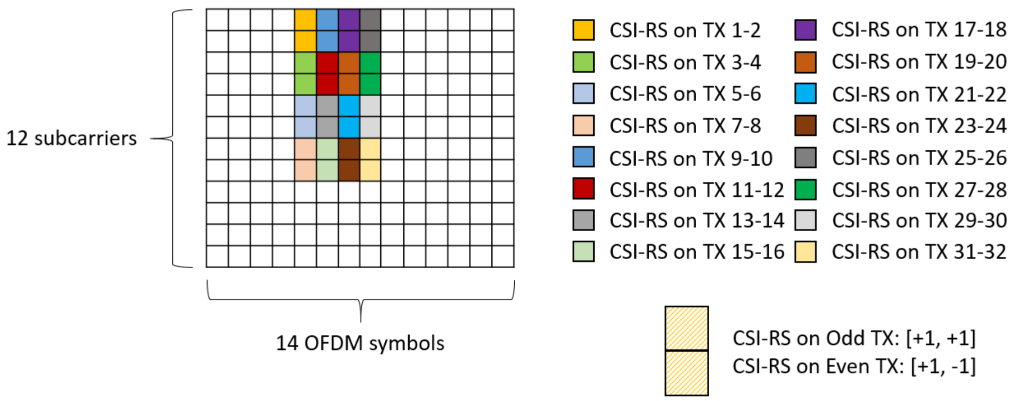


Figure 1: CSI-RS pattern in a resource block

The competition provides two different densities of received pilots as input information in the dataset, each including 300,000 samples. Among them, the high-density pilot (pilot_1.npz) occupies 26 resource blocks (1, 3, 5, ..., 49, 51) of the total 52 resource blocks to send CSI-RS, and the size of each data sample is 4*208*4, which corresponding to 4 receiving antennas, 208 pilot subcarriers (26 resource blocks, each resource block has 8 subcarriers with pilots), 4 OFDM symbols; and for low-density pilots (pilot_2.npz), only 6 resource blocks (7, 15, 23, 31, 39, 47) out of the total 52 resource blocks are occupied to send CSI-RS, and the size of each data sample is 4*48*4, which corresponding to 4 receiving antennas, 48 pilot subcarriers (6 resource blocks, each resource block has 8 subcarriers with pilot), 4 OFDM symbols.

- Channel state information (eigenvectors of channel)

The channel state information (eigenvectors of the channel) are generated based on the whole channel information under 32 transmitting antennas and 4 receiving antennas (32T4R). Considering both the feedback performance and feedback overhead, the transmission bandwidth is equally divided into 13 subbands, and corresponding eigenvectors of channel are constructed in each subband as the label in the competition dataset. In order to make it easier for participants to understand the construction process of the label, the competition provides an example code for generating channel state information based on whole channel information for reference (ht_to_w.py). In addition, it should be noted that since there is a certain time interval between the pilot signal and the target data transmission moment (the channel eigenvector is generated at this moment), the time-variance of channel will also cause challenge.

This part of the dataset (w.npz) include 300,000 samples as common labels for both high-density and low-density pilots. The size of each data sample is 32*13, which corresponding to the length of each eigenvector and number of subbands respectively.

- Whole channel information

The whole channel information (time domain) is provided to participants as an auxiliary information which can only be used in model training phase, so that the participants can have more flexibility in the design of the solution. Meanwhile, the example code (ht_to_hf.py) which demonstrate how to transform the time-domain whole channel information to frequency-domain is also provided to participants for use.

This part of the dataset (h_time.npz) include 300,000 samples as common auxiliary information for both high-density and low-density pilots. The size of each data sample is 4*32*64, which corresponding to 4 receive antennas, 32 transmit antennas, and 64 delay samples respectively.

All the provided data and example code is summarized as in the following table,

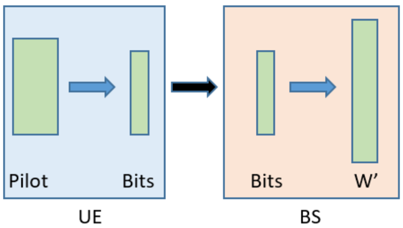
File name	Number of samples	Size	Description
pilot_1.npz	300,000	4*208*4	High-density received pilot signal
pilot_2.npz	300,000	4*48*4	Low-density received pilot signal
w.npz	300,000	32*13	Channel state information (eigenvector of channel)
h_time.npz	300,000	4*32*64	Time-domain whole channel information
ht_to_w.py	\	\	Calculate channel state information based on whole channel information
ht_to_hf.py	\	\	Transform channel information from time to frequency domain

Note: The competition does not allow the use of external datasets. After the online evaluation phase, it is necessary for the winning teams to illustrate the training process during the reviewing phase. The reproduce of training process will be conducted if necessary. In the event of non-reproducible performance, illegal use of external data, etc., the results will be cancelled.

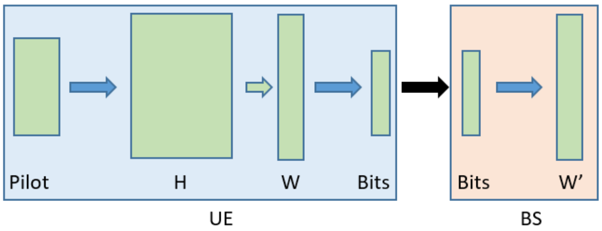
Solution Examples

Towards the competition task, there are a variety of potential solutions. This section provides some examples of potential solutions. Participants can freely choose the most appropriate way (**not limited to the examples**) in order to achieve the best performance.

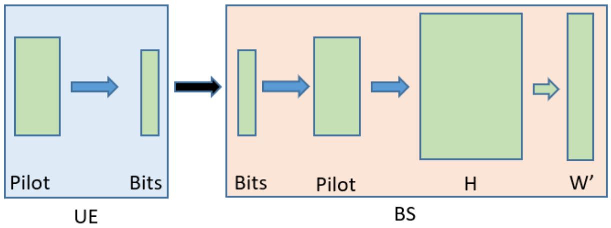
1. Solution 1: Integrated design, using AI to achieve end-to-end CSI feedback based on pilot signal as input



2. Solution 2: Modular design refer to the traditional module partitioning. Firstly, the AI model is used for channel estimation to achieve the purpose of frequency domain interpolation and denoising. Then, based on the estimated channel, the eigenvectors of the channels to be feedback on different subbands are calculated. Finally, another AI model is used to realize the CSI compression and recovery.



3. Solution 3: Modular design. Firstly, the AI model is used to directly compress and feedback the received pilot, and then another AI model is used at the base station to conduct channel estimation based on the feedbacked pilot information, and finally the channel eigenvectors on different subbands are calculated corresponding to the estimated channel.



Result Submission

Participants should design the model according to the following rules, and upload the zip package of the model to the scoring system of the competition platform.

- Recommended programming language version: Python 3.7，
- Recommended package version: TensorFlow 2.1.0; PyTorch>1.7.0; Numpy 1.18.1; h5py 2.10.0; Sklearn 0.23.2; Scipy 1.4.1
- Maximum upload file size: 400MB,
- Maximum model inference time: 400s.

The scoring system supports the submission of model using TensorFlow or PyTorch. We provide the templates of both as baseline for reference, where

1. modelTrain_example1.py: For model constructing and training. This example is used to train an integrated model. The training data address is set as 'pilot_*.npz' and 'w.npz'; and the model saving address is set as './modelSubmit/*.h5' (for TensorFlow) or './modelSubmit/*.pth.tar' (for PyTorch).
2. modelDesign_example1.py: For model designing. It includes encoding functions 'encFunction_1' and 'encFuntion_2' and decoding functions 'decFunction_1' and 'decFunction_2', which corresponding to the two different densities of input pilots respectively and will be called by the scoring system. Participants can self-define the content of these functions, but the interface of these functions cannot be modified. Specifically, the parameters that the encoding function required as input include pilot data, two model file paths, and the encoded bit stream needs to be output; the parameters that the decoding function required as input include the encoded bit stream and two model file paths, and the reconstructed CSI (channel eigenvectors of multiple subbands) needs to be output. It is worth noting that, within each encoding or decoding function, whether to use and how to use the AI model needs to be designed by the participants themselves. In this example, the encoding and decoding functions call the 'encoder' and

'decoder' models respectively.

3. modelTrain_example2.py: This example is used to separately train the channel estimation model and the CSI feedback model and perform necessary data processing during the process. Specifically, in this example, the time-domain whole channel information is firstly transformed into frequency-domain that can be used to train the frequency-domain channel estimation model. And then the channel eigenvectors are calculated based on the inference result of the trained channel estimation model, and finally the model for CSI feedback is trained. The training data address is set as 'pilot_*.npz' , 'h_time.npz' and 'w.npz'; and the model saving address is set as './modelSubmit/*.h5' (for TensorFlow) or './modelSubmit/*.pth.tar' (for PyTorch).
4. modelDesign_example2.py: Similar to the example 1, including the encoding functions 'encFunction_1' and 'encFuntion_2' and decoding functions 'decFunction_1' and 'decFunction_2', which corresponding to the two different densities of input pilots respectively. The usage rule and interface definitions are the same with example 1. In this example, the encoding function first use the channel estimation model 'channel_est' to estimate channel in frequency domain, then calculates the channel eigenvectors on different subbands based on the channel estimation results, and finally use the 'encoder' model for encoding. The decoding function use the 'decoder' model to reconstruct the channel eigenvectors based on the input bit stream.
5. evaluation.py: It is used by the scoring system to evaluate the performance and effectiveness of the submitted solution. The script will firstly call the encoding functions 'encFunction_1' and 'encFunction_2' which defined in modelDesign.py to finish the encoding; then call the decoding functions 'decFunction_1' and 'decFunction_2' which defined in modelDesign.py to reconstruct CSI based on the encoded bit stream. Finally, finish the score calculation. Note that the interface of encoding and decoding functions cannot be modified, but participants can freely decide whether to use and how to use the AI model file.

Files need to be submitted,

1. modelDesign.py (for TensorFlow or PyTorch)
2. encModel_*.h5（for TensorFlow）or encModel_*.pth.tar（for PyTorch）
3. decModel_*.h5（for TensorFlow）or decModel_*.pth.tar（for PyTorch）

Submission Example

For TensorFlow: Zip the files with the following structure, name the zip package as 'submit_tf' and upload it, i.e.,

```
submit_tf.zip
├── modelDesign.py
└── modelSubmit (folder）
    ├── encModel_p1_1.h5
    ├── encModel_p1_2.h5 (optional)
```

├── decModel_p1_1.h5

├── decModel_p1_2.h5 (optional)

├── encModel_p2_1.h5

├── encModel_p2_2.h5 (optional)

├── decModel_p2_1.h5

└── decModel_p2_2.h5 (optional)

For PyTorch: Zip the files with the following structure, name the zip package as ‘submit_pt’ and upload it, i.e.,

submit_pt.zip

├── modelDesign.py

└── modelSubmit (folder）

├── encModel_p1_1.pth.tar

├── encModel_p1_2. pth.tar (optional)

├── decModel_p1_1. pth.tar

├── decModel_p1_2. pth.tar (optional)

├── encModel_p2_1. pth.tar

├── encModel_p2_2. pth.tar (optional)

├── decModel_p2_1. pth.tar

└── decModel_p2_2. pth.tar (optional)

Score Rules

Participants need to train and submit models separately in the two given cases,

- Case 1: CSI feedback based on high-density pilot signal ‘pilot_1’
- Case 2: CSI feedback based on low-density pilot signal ‘pilot_2’

For each case, the score is defined as,

$$score = \frac{1}{N_{sp}} \sum_{j=1}^{N_{sp}} \frac{1}{N_{sb}} \sum_{i=1}^{N_{sb}} \frac{||\mathbf{w}_{i,j}^{\mathrm{H}} \mathbf{w'}_{i,j}||^2}{||\mathbf{w}_{i,j}||^2 ||\mathbf{w'}_{i,j}||^2}$$

Where N_{sp} denotes the number of samples in the test set, N_{sb} denotes the number of subbands of each sample, w_{ij} and w_{ij}' denote the true eigenvectors and the predicted eigenvector of the ith subband in jth sample, respectively. $(\cdot)^H$ denotes the Hermitian matrix/vector of the input matrix/vector.

Final score is defined as score_final = (score_case1 + score_case2) / 2.

Note: Both the single case score and the final score will be shown on the leaderboard.

■ Bonus

Golden Prize:	**300,000 RMB**	**for 1 team**
Silver Prize:	**50,000 RMB (per team)**	**for 2 teams**
Bronze Prize:	**20,000 RMB (per team)**	**for 3 teams**
Winning Prize:	**10,000 RMB (per team)**	**for 4 teams**

Note 1– the award can be settled in US dollars according to the exchange rate on the settlement date

Note 2– the individual income tax or other forms of tax on the bonus will be borne by the winners, and will be withheld and paid by the organizer of the competition. The participating teams shall be responsible for the distribution and distribution of the bonus among their members, and the organizer will not be held responsible for this.

■ Contact

Tel: 010-62381637

Email: liujia@datafountain.cn

Community:

Official WeChat Communication Group:

2022 WAIC-AI-based Joint Channel Estimation and Channel State Information Feedback