Beamforming in Antenna Technology

Indian Institute of Technology Madras, India

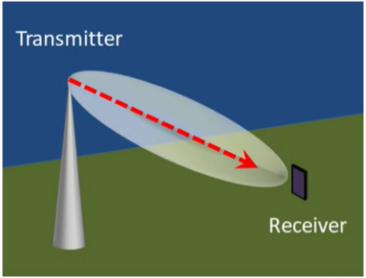
Outline

1. Forming beam pattern
2. Beam steering
3. Adaptive beamforming
4. Beamforming in wireless communication (5G) and beyond
5. Precoding
6. An application using precoder

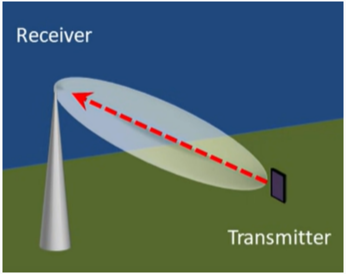
Introduction

- Increasing demand for mobile communication services
  - More users per cell
  - Higher data rate
- Radio operators choose higher frequencies with newer technologies, e.g., millimeter-wave (mmWave, wavelength 1–10 mm) and terahertz (THz, wavelength 100 μm–1 mm) communications.
- Strength: smaller antenna height
- Challenges: severe path loss
- Solution:
  - Use high-gain antennas
  - Concentrate the radio wave power towards the line linking the base station and the mobile equipment - **Beamforming**

Beamforming



(1.1) Transmission beamforming: concentrate the transmit energy towards mobile receiver



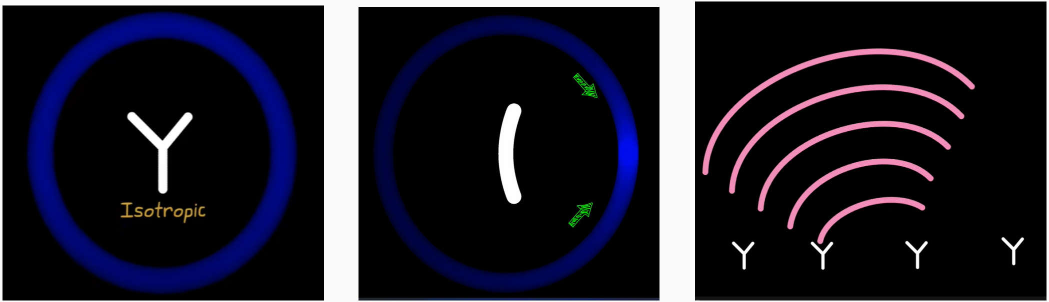
(1.2) Receive beamforming: allows the receiver antennas to focus towards the mobile transmitter

Figure 1: Transmission and receive beamforming

What is beamforming?

- Beamforming is a signal processing technique used in sensor arrays for directional signal transmission or reception.
- Beamforming
  - improve the signal-to-noise ratio of received signals
  - eliminate undesirable interference sources
  - focus transmitted signals to specific locations

Why are phased arrays needed?



(2.1) Isotropic antenna **(2.2)** Dish antenna **(2.3)** Antenna arrays

Figure 2: Different types of antennas [1]

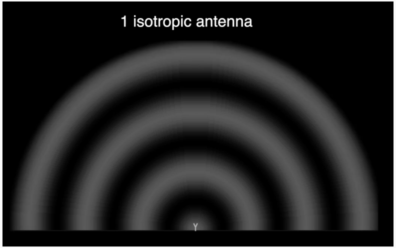
[1] Image source: `https://www.youtube.com/playlist?list=PLn8PRpmsu08q9U0y7_63Dfz5cawEnicxi`

Why are phased arrays needed?

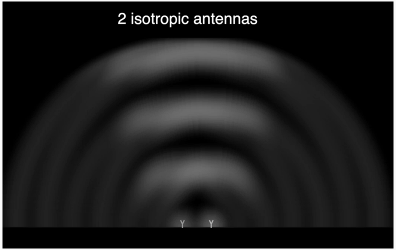
Figure 3: Antenna arrays are generating multiple beams targeted to multiple users and can steer them independently of each other using phase shifting

Forming beam pattern

Beamforming with phased array



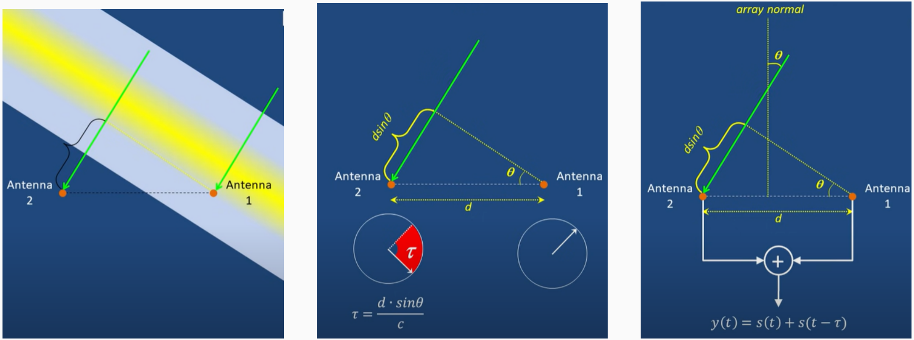
(4.1) Single antenna



(4.2) Two antenna with same phase

Figure 4: Beamforming by constructive and destructive interference

Mathematical expression of beamforming with two elements



(5.1) Incoming wave

(5.2) Delay between array elements

(5.3) Composite signal

Figure 5: Beamforming with 2 antenna elements at the receiver[2]

[2] Source: `https://www.youtube.com/watch?v=HKpQP8H4JRc&t=1058s`

Mathematical expression of Beamforming with two elements

- Modulated wave with carrier frequency ω_0

$$s(t) = x(t)\cos(\omega_0 t)$$

- Array output is

$$\begin{aligned} y(t) &= s(t) + s(t-\tau) \\ &= x(t)\cos(\omega_0 t) + x(t-\tau)\cos[\omega_0(t-\tau)] \\ &\cong x(t)\cos(\omega_0 t) + x(t)\cos[\omega_0(t-\tau)], \end{aligned}$$

considering $x(t-\tau) \cong x(t)$ because the distance between two antenna elements is very less compared to the path the signal has travelled

Contd.

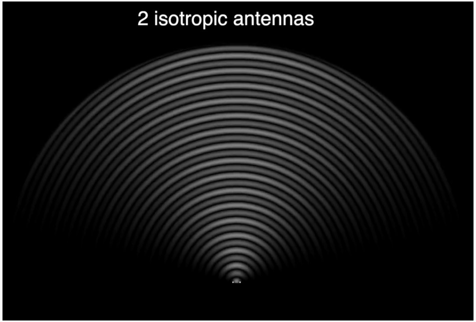
- Considering $\omega_0 \tau = \psi$,

$$y(t) = x(t)\cos(\omega_0 t) + x(t)\cos(\omega_0 t - \psi)$$

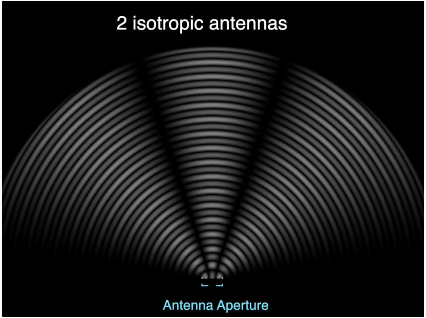
- Considering the analytic Phasor representation of R.H.S.

$$\begin{aligned}
\tilde{y}(t) &= x(t)e^{j\omega_0 t} + x(t)e^{j\omega_0 t}e^{-j\psi} \quad \text{where } \psi = \frac{2\pi d}{\lambda}\sin\theta \quad \text{(Phasor notation)} \\
&= x(t)e^{j\omega_0 t}\sum_{i=0}^{1} e^{-ji\psi} \quad \text{where } ArrayFactor(\theta) = \sum_{i=0}^{1} e^{-ji\psi} \\
&= x(t)e^{j\omega_0 t}\sum_{i=0}^{1} a_i e^{-ji\psi} \quad \text{if } i^{th} \text{ antenna has gain } a_i
\end{aligned}$$

Narrowing the main beam by increasing the gap between antennas



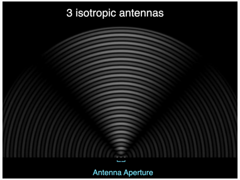
(6.1) $d = \lambda/2$



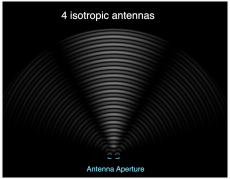
(6.2) $d = 3\lambda/2$

Figure 6: Beamforming with two antennas with different d (distance between two antenna elements)

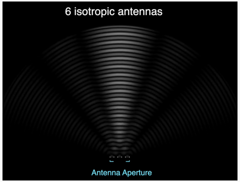
Effect of adding more antenna elements



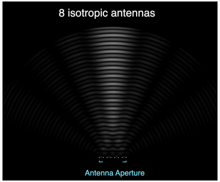
(7.1) 3 antennas



(7.2) 4 antennas



(7.3) 6 antennas



(7.4) 8 antennas

Figure 7: Beam gets narrower and the side lobes become insignificant with more antenna elements where distance between two elements is $d = \lambda/2$

Beam pattern

- The constructive/destructive interference pattern depends on
  - Number of elements
  - Element spacing
  - Geometry of the array layout
- These contribute to **array factor**
- The interference pattern also depends on individual antenna patterns (e.g., isotropic, sinc) and can be tuned to get the desired pattern
- Specifically, array pattern is a function of array factor and antenna pattern

Beam steering

Steer a beam electronically by changing the phase

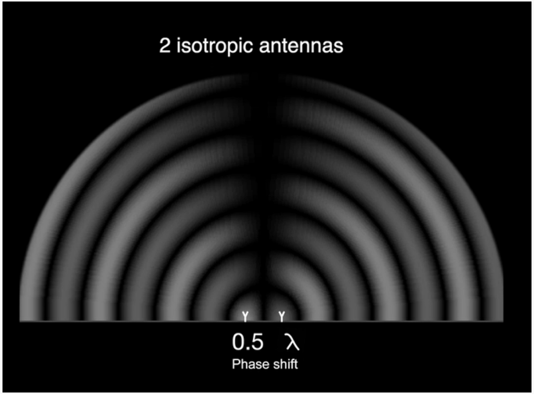


Figure 8: Beam steering - by delaying the signal by any amount through different phase shifts, we can steer the beam in any direction.

Mathematical expression for beam steering

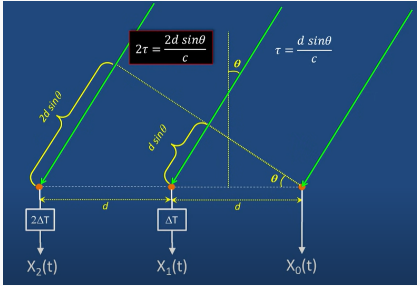


Figure 9: Signal is incident on the receiving antenna

- Modulated signal $X_0(t) = x(t)\cos\omega_0 t$
- User can tune ΔT to vary the beam direction
- Output of the first element

$$\begin{aligned} X_1(t) &= x(t-\tau-\Delta T)\cos[\omega_0(t-\tau-\Delta T)] \\ &= x(t)\cos[\omega_0(t-\tau-\Delta T)], \\ &\qquad (\text{considering } x(t-\tau-\Delta T) = x(t)) \\ &= x(t)e^{j\omega_0 t}e^{-j(\omega_0\tau+\omega_0\Delta T))} \text{ (phasor notation)} \\ &= x(t)e^{j\omega_0 t}e^{-j(\psi+\delta))} \end{aligned}$$

where $\delta = \omega_0 \Delta T$

- Similarly, output of second element is

$$X_2(t) = x(t)e^{j\omega_0 t}e^{-2j(\psi+\delta))}$$

Beam steering

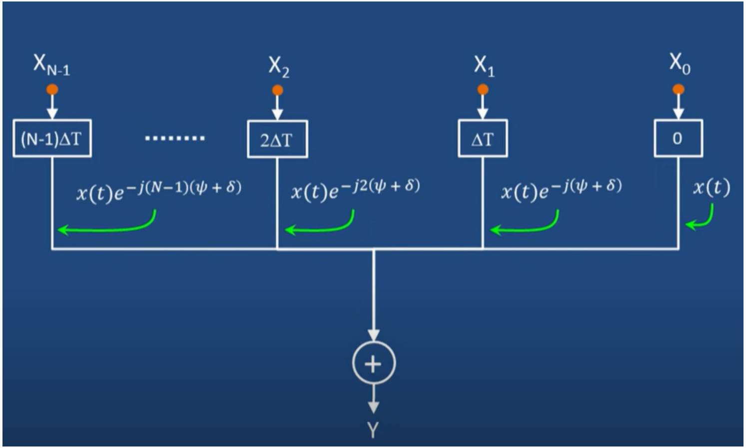
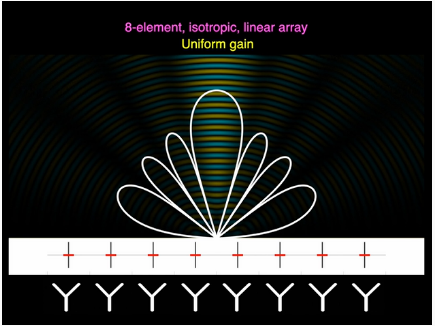
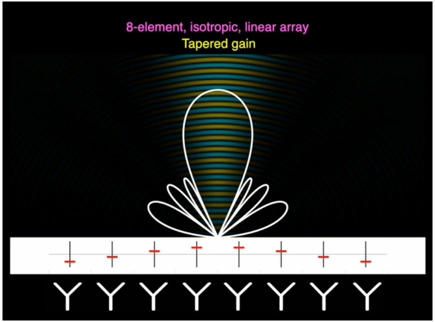


Figure 10: Delay applied to N elements of an antenna array. By changing the ΔT, the beam can be steered towards any direction.

Take away till now



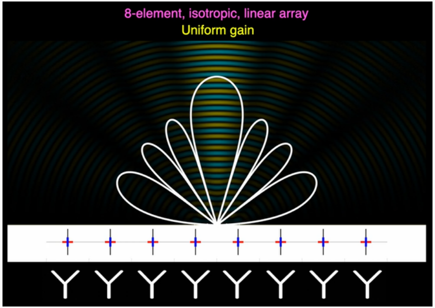
(11.1) Uniform gain



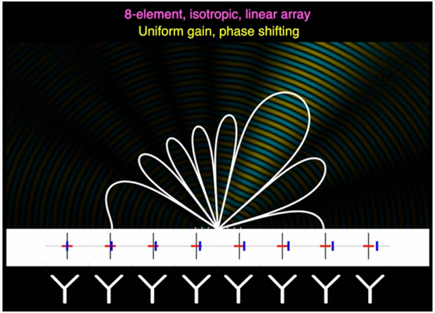
(11.2) Tapered gain

Figure 11: Gain of array creates the beam pattern

Take away till now



(12.1) No phase shift



(12.2) With uniform phase shift

Figure 12: Phase shifts steers that beam

Adaptive beamforming

How to have more control over a beam?

- Have different gain of each element
- Have different phases of each element unevenly
- With a fast analog-to-digital converter, we can have full digital control over the gain and phase of each antenna element
- For every antenna element, a complex number denotes the gain and phase - the vector consisting of these complex numbers is the weight of the antenna array
- How to set the weights
  - **Conventional beamforming** - Tapered gain for side lobes, phase shifts to steer
  - **Adaptive Beamforming** - Optimize the weight of the array antenna based on the received data, which in turn depends on the environment, e.g., MVDR

Minimum Variance Distortionless Response (MVDR)

- Preserves the gain in the direction of arrival of a desired signal and attenuates interference from other directions.
- Model the unknown nonrandom signal propagating along the known direction, so the frequency domain snapshot is given by

$$\mathbf{x} = a\mathbf{v}_0 + \mathbf{n}$$

where $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \mathbf{S}_n)$, such that $\mathbf{S}_n \neq \sigma_n^2\mathbf{I}$ and $a\mathbf{v}_0$ is the signal. The array manifold vector (look direction) $\mathbf{v}_0$ is deterministic, and $a \sim \mathcal{CN}(0, \sigma_a^2)$ is the frequency domain snapshot of the source signal.

- If beamformer is $\mathbf{w}$ where $\mathbf{w}^H \in \mathbb{C}^{1\times N}$, then the received signal is

$$y = \mathbf{w}^H\mathbf{x} = \mathbf{w}^H(a\mathbf{v}_0 + \mathbf{n}) = a\mathbf{w}^H\mathbf{v}_0 + \mathbf{w}^H\mathbf{n}$$

MVDR (Contd.)

- In the absence of noise $y = a$
- So, we want to keep gain

$$\mathbf{w}^H \mathbf{v}_0 = 1$$

whereas minimizing variance

$$\mathbb{E}\{|\mathbf{w}^H \mathbf{n}|^2\} = \mathbf{w}^H \mathbb{E}\{\mathbf{n}\mathbf{n}^H\}\mathbf{w} = \mathbf{w}^H \mathbf{S}_n \mathbf{w}$$

- Choose $\mathbf{w}$ to minimize $\mathbf{w}^H \mathbf{S}_n \mathbf{w}$ such that $\mathbf{w}^H \mathbf{v}_0 = 1$
- So, the Lagrange problem becomes

$$J = \mathbf{w}^H \mathbf{S}_n \mathbf{w} + \gamma(\mathbf{w}^H \mathbf{v}_0 - 1) + \gamma^*(\mathbf{v}_0^H \mathbf{w} - 1)$$

MVDR (Contd.)

- Solving for $\mathbf{w}$ in $\nabla_{\mathbf{w}^H} J = 0$ gives

$$\mathbf{w} = -\gamma \mathbf{S}_n^{-1} \mathbf{v}_0$$

- Next, solve for γ in the constraint to obtain

$$\gamma = -\frac{1}{\mathbf{v}_0^H \mathbf{S}_n^{-1} \mathbf{v}_0} \tag{1}$$

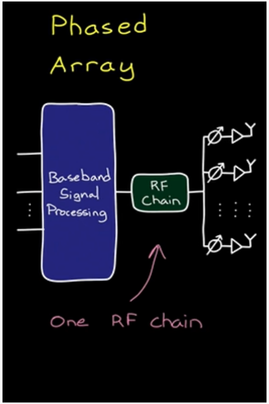
- Therefore, the MVDR beamformer is given by

$$\mathbf{w} = \frac{\mathbf{S}_n^{-1} \mathbf{v}_0}{\mathbf{v}_0^H \mathbf{S}_n^{-1} \mathbf{v}_0} = \alpha \mathbf{S}_n^{-1} \mathbf{v}_0$$

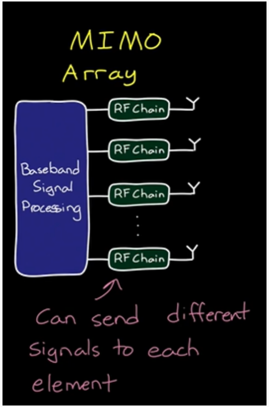
where $\alpha = \frac{1}{\mathbf{v}_0^H \mathbf{S}_n^{-1} \mathbf{v}_0}$ is the gain factor and is a scaler.

Beamforming in wireless communication (5G) and beyond

Multichannel beamforming using MIMO array



(13.1) Analog phased array



(13.2) MIMO array

Figure 13: In the MIMO array, each element is connected to its own RF chain, so we have more control over the beams. This flexibility allows us to do **Multi-channel beamforming**.

Multichannel Beamforming for Spatial diversity

Wireless communication needs to provide **robust communication link** to serve in scattering-rich environments in urban cities that have **multiple scattered paths**

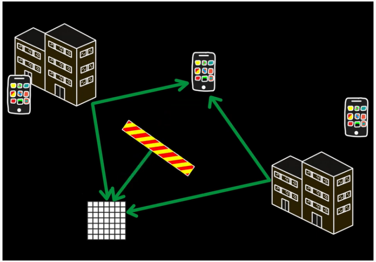


Figure 14: Multi-channel beamforming takes advantage of multi-path environments

In a scatter-rich environment, a better approach is to use different channels to **send the same information along different paths - Spatial diversity**. In this way, when one path is blocked, at least some power reaches the receiver through other paths.

Multichannel Beamforming for capacity enhancement

Apart from the robust link, 5G also needs to serve **many users**

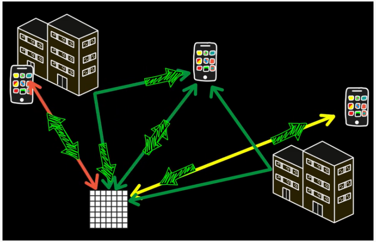


Figure 15: Beamforming to many users to enhance capacity

Allows to send **completely different signals on different channels in different spatial directions** using a single antenna array to serve many 5G radios - **capacity enhancement**

Serving multiple users with MIMO array

- Both capacity enhancement and multi-path communication need massive MIMO that has hundreds of elements and, therefore, has hundreds of channels (theoretically), each pointing in a different direction and serving different users
- In 5G, to direct the signals in a crowded area, we need very sharp beams too - achieved by **massive MIMO array** with many antenna elements
- Compared to an analog phased array, in a massive MIMO array, we want to have multiple channels, so we need to have multiple RF chains
- This incurs cost issues in massive MIMO as 5G operates in mm wavelength so all RF components need to be deployed in a very small area
- Solution: **Hybrid array**

Hybrid array

- A trade-off between analog phased array and massive MIMO array
- It has more antenna elements, therefore, sharper beams as expected in 5G
- But it does not have as many channels as massive MIMO because of limitations in deployment yet enough to serve many users simultaneously

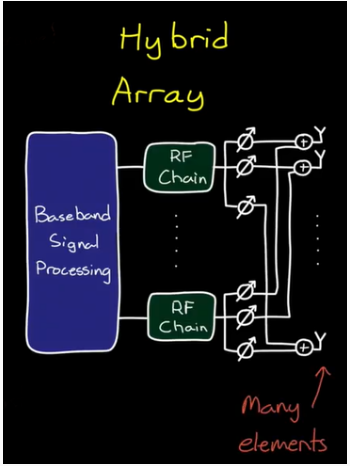
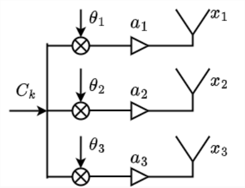


Figure 16: Hybrid array

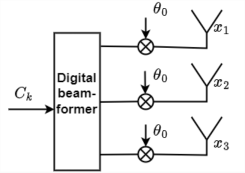
Precoding

How are beamforming and precoding related? [3]

- Analog beamforming: when C_k is the transmitted symbol, find **a** and $\boldsymbol{\theta}$ such that SNR is maximized; so that
$$\begin{bmatrix} x_{1k} \\ x_{2k} \\ x_{3k} \end{bmatrix} = \begin{bmatrix} a_1 e^{j\theta_1} \\ a_2 e^{j\theta_2} \\ a_3 e^{j\theta_3} \end{bmatrix} C_k$$
- Digital beamforming: if the above operation is done in digital form
- By exploiting the power of digital beamforming, we can consider other types of cost functions than maximize SNR - that is when we do **precoding**
- Beamforming is a special case of precoding



(17.1) Analog beamforming



(17.2) Digital beamforming

[3] Source: https://www.youtube.com/playlist?list=PLx7-Q20A1VYKwoWNCyWfErLArGLtKdS37

Precoding in wireless communication

- With precoding with digital beamformers, theoretically, we can have the same number of inputs as the number of antennas (MIMO arrays)
- It is also called spatial coding, as the symbols are sent to spatially different users

$$\begin{bmatrix} x_{1k} \\ x_{2k} \\ x_{3k} \end{bmatrix} = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} \begin{bmatrix} C_{1k} \\ C_{2k} \\ C_{3k} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{d}_1 & \mathbf{d}_2 & \mathbf{d}_3 \end{bmatrix} \begin{bmatrix} C_{1k} \\ C_{2k} \\ C_{3k} \end{bmatrix}$$

where $\mathbf{d}_1$, $\mathbf{d}_2$ and $\mathbf{d}_3$ are beamformers for C_{1k}, C_{2k} and C_{3k} respectively.

- To find the parameters of the beamformer, i.e., to find $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3]$ one needs the **knowledge of channel**

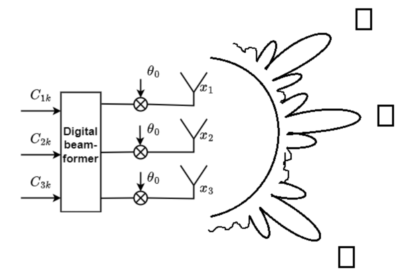


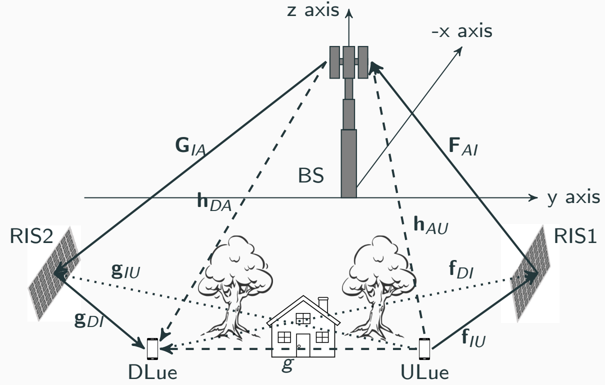
Figure 17: Precoding

Different types of precoders

- While finding **D**, one can maximize SNR for each user - **Maximum Ratio Transmission**, **Maximum Ratio Combining** precoders
- We can also steer nulls towards other users to minimize interference to other users - results in **Zero forcing precoder**

An application using precoder

Application of precoders in modern Full Duplex communication systems with RIS[4]



[4] Nayak, Nancy, Sheetal Kalyani, and Himal A. Suraweera. "A DRL Approach for RIS-Assisted Full-Duplex UL and DL Transmission: Beamforming, Phase Shift and Power Optimization." arXiv preprint arXiv:2212.13854 (2022).

System model

- Full-duplex (FD) setup one FD BS, one Half duplex (HD) ULue, one HD DLue, and two Reconfigurable Intelligent Surfaces (RIS) to facilitate communication when the users are not in LoS with the BS
- BS has a uniform linear antenna (ULA) array with M_t transmit antenna elements and M_r receive antenna elements
- ULue and DLue are single-antenna HD user
- Both the RISs are deployed as Uniform Planar Antenna (UPA)
- RIS1 and RIS2 has $N_1 = N_{1_h} N_{1_v}$ and $N_2 = N_{2_h} N_{2_v}$ reflecting elements
- The direct paths are blocked therefore has high pathloss
- User to user interference and inter-RIS interference is present, but paths have high pathloss
- DLue is HD, therefore it cannot transmit and receive at the same time

System model

- The diagonal phase-shift matrices $\mathbf{\Theta}_U$ and $\mathbf{\Theta}_D$ of the two RISs are

$$\begin{aligned}\mathbf{\Theta}_U &= \mathrm{diag}\{\bar{\Theta}_U\} = \mathrm{diag}\{\phi_{U1} \ \ldots \ \phi_{UN_1}\}, \text{ and} \\ \mathbf{\Theta}_D &= \mathrm{diag}\{\bar{\Theta}_D\} = \mathrm{diag}\{\phi_{D1} \ \ldots \ \phi_{DN_2}\}\end{aligned} \tag{2}$$

  with $\phi_n = e^{j\theta_n}$
- s_U denotes the transmit signal from the ULue
- $p_U > 0$ denotes the transmit power of the ULue
- s_D denotes the transmit signal of the BS
- $p_A > 0$ denotes the transmit power of the BS
- Channels are in the figure

System model

- The received signal at the BS is given by

$$
\begin{aligned}
y_A = {} & \mathbf{w}_R \mathbf{h}_{AU} \sqrt{p_U} s_U + \mathbf{w}_R \mathbf{F}_{AI} \boldsymbol{\Theta}_U \mathbf{f}_{IU} \sqrt{p_U} s_U + \mathbf{w}_R \mathbf{G}_{IA}^T \boldsymbol{\Theta}_D \mathbf{g}_{IU} \sqrt{p_U} s_U \\
& + \underbrace{\mathbf{w}_R \mathbf{H}_{AA} \mathbf{w}_T \sqrt{p_A} s_D}_{\text{residual SI}} + \mathbf{w}_R n_A,
\end{aligned} \tag{3}
$$

- The signal received by the DLue is

$$
\begin{aligned}
y_D = {} & \mathbf{h}_{DA} \mathbf{w}_T \sqrt{p_A} s_D + \mathbf{g}_{DI} \boldsymbol{\Theta}_D \mathbf{G}_{IA} \mathbf{w}_T \sqrt{p_A} s_D + \mathbf{f}_{DI} \boldsymbol{\Theta}_U \mathbf{F}_{AI}^T \mathbf{w}_T \sqrt{p_A} s_D \\
& + \underbrace{\mathbf{g}_{DI} \boldsymbol{\Theta}_D \mathbf{g}_{IU} \sqrt{p_U} s_U + \mathbf{f}_{DI} \boldsymbol{\Theta}_U \mathbf{f}_{IU} \sqrt{p_U} s_U}_{\text{inter-RIS}} + \underbrace{g \sqrt{p_U} s_U}_{\text{inter-user}} + n_D.
\end{aligned} \tag{4}
$$

- Here transmit and receiver beamformers are denoted by $\mathbf{w}_T$ and $\mathbf{w}_R$ respectively
- Highlighted term leads to a very high level of interference if not canceled

SINR and Data rate

- The SINR at the BS, γ_{BS} and the DL user, γ_{DL} are given by

$$\begin{aligned}\gamma_{BS} &= \frac{p_U||\mathbf{w}_R(\mathbf{h}_{AU} + \mathbf{F}_{AI}\mathbf{\Theta}_U\mathbf{f}_{IU} + \mathbf{G}_{IA}^T\mathbf{\Theta}_D\mathbf{g}_{IU})||^2}{p_A||\mathbf{w}_R\mathbf{H}_{AA}\mathbf{w}_T||^2 + \mathbf{w}_R^2\sigma_A^2}, \text{ and} \\ \gamma_{DL} &= \frac{p_A||(\mathbf{h}_{DA} + \mathbf{g}_{DI}\mathbf{\Theta}_D\mathbf{G}_{IA}\mathbf{w}_T + \mathbf{f}_{DI}\mathbf{\Theta}_U\mathbf{F}_{AI}^T\mathbf{w}_T)||^2}{p_U||(g + \mathbf{g}_{DI}\mathbf{\Theta}_D\mathbf{g}_{IU} + \mathbf{f}_{DI}\mathbf{\Theta}_U\mathbf{f}_{IU})||^2 + \sigma_D^2},\end{aligned} \tag{5}$$

  respectively.
- Accordingly, the data rate at the DLue and the BS are given by

$$\begin{aligned} r_{DL} &= \log_2\left(1 + \gamma_{DL}\right), \text{ and} \\ r_{BS} &= \log_2\left(1 + \gamma_{BS}\right).\end{aligned} \tag{6}$$

Objective

The optimization problem is formulated as follows:

$$\begin{aligned} \mathcal{P}_1 : \quad & \max_{\boldsymbol{\Theta}_D, \boldsymbol{\Theta}_U, \mathbf{w}_T, \mathbf{w}_R, p_A, p_U} r_{BS} + r_{DL} \\ & \text{s.t. } p_A^{max} \geq p_A \geq 0, \quad p_U^{max} \geq p_U \geq 0, \\ & |\phi_n| = 1, 1 \leq n \leq N_1, 1 \leq n \leq N_2 \end{aligned} \tag{7}$$

In our work, we solve for RIS phase shifts, transmit and receive beamformers, and transmit powers using a two-stage Deep Reinforcement Learning (DRL) algorithm that does not need any CSI, unlike other precoding techniques.

Existing works

- Either assumes negligible residual self-interference due to excellent SI mitigation technique
- Or assumes the presence of very little residual SI and then
  - cancels the residual SI by using beamformers
  - however, designing beamformers needs CSI
- What happens if the CSI is noisy?
- What happens if the residual SI is high/unknown?
- Solution is **two-stage Deep Reinforcement Learning (DRL) algorithm**

What are the strengths of DRL algorithms?

- DRL algorithms learns from data and does not need any additional knowledge of parameters like CSI
- DRL methods enable end-to-end learning, where the agent directly learns policies or value functions from raw input to action output, without relying on manual feature engineering
- DRL methods can effectively handle complex and non-linear environments
- When a problem is non-convex, it is either relaxed or broken into smaller problems and then solved. DRL algorithms are powerful enough to give a single-shot solution to any difficult optimization problem.
- With DRL, it is more flexible to change the optimization objective and still have a simple solution.

DRL based two-stage algorithm

- Main **challenge in the FD communication** system: the **self-interference (SI)** imposed by the transmit antenna on the receive antenna of the BS
- If SI mitigation scheme is not good, the residual SI has high power
- As the **DRL agent learns from feedback**, it is difficult for the DRL agent to learn anything **if the received signal has too much interference** due to the high residual SI
- So first stage is to **cancel a major part of the SI interference by sending a pilot symbol** (next slide)
- Second stage is to **feed the data to an Intelligent agent** which learns to predict the RIS phaseshifts, beamformers and the transmit powers (agent: DRL algorithm)

Two-stage DRL algorithm

- By using DRL, one can smartly can avoid estimating CSI and can **reduce the overhead**
- Uses only one **scalar feedback**, i.e., the weighted sum of UL and DL SINR from the environment instead of CSI which is in the form of Matrix for MIMO channels
- For our simulation setup i.e. 10 BS antenna elements and 36 RIS elements per RIS, the number of instantaneous CSI that need to be fed to the algorithm is 813
- In contrast, the proposed CSI oblivious Minimum Signaling Feedback (MSF) DRL method needs to transmit one pilot signal and needs to receive one reward values in a single time step

First stage: Least square-based SI-cancellation (LSSIC)

- The signal due to SI can be canceled with an estimate of $\mathbf{w}_R \mathbf{H}_{AA} \mathbf{w}_T$ denoted by $\hat{h}$ as

$$y_A = \mathbf{w}_R \mathbf{F}_{AI} \mathbf{\Theta}_U \mathbf{f}_{IU} \sqrt{p_U} s_U + (\mathbf{w}_R \mathbf{H}_{AA} \mathbf{w}_T - \hat{h}) \sqrt{p_A} s_D + \mathbf{w}_R n_A. \tag{8}$$

- At every epoch, the scalar $\hat{h}$ is estimated by sending a pilot signal $s_D^p \in \mathcal{C}$ at the BS from the transmitter to the receiver antenna
- The corresponding received signal $y_A^p \in \mathcal{C}$ at the receiver antenna of BS can be expressed as

$$y_A^p = \mathbf{w}_R \mathbf{H}_{AA} \mathbf{w}_T \sqrt{p_A} s_D^p + v_A, \tag{9}$$

where $v_A \in \mathcal{C}$ is the AWGN and the scalar $\mathbf{w}_R \mathbf{H}_{AA} \mathbf{w}_T$ needs to be estimated.

Contd. First stage: LSSIC

- To estimate $\mathbf{w}_R \mathbf{H}_{AA} \mathbf{w}_T$, we need to minimise the error $J(h)$ where

$$
\begin{aligned}
J(h) &= \overline{(y_A^p - \sqrt{p_A}\, s_D^p\, h)}(y_A^p - \sqrt{p_A}\, s_D^p\, h), \\
&= \overline{y_A^p}\, y_A^p - 2\sqrt{p_A}\, h\, \overline{y_A^p}\, s_D^p + p_A\, h^2\, \overline{s_D^p}\, s_D^p.
\end{aligned}
\tag{10}
$$

  where $\bar{x}$ denotes the conjugate transpose of x.

- By taking the derivative of $J(h)$ with respect to h and equating it to zero to obtain $\hat{h}$,

$$
\begin{aligned}
\frac{\partial J(h)}{\partial h} &= 0 - 2\sqrt{p_A}\, \overline{y_A^p}\, s_D^p + 2 p_A\, h\, \overline{s_D^p}\, s_D^p = 0, \\
\text{therefore, } \hat{h} &= \frac{1}{\sqrt{p_A}} (\overline{s_D^p} s_D^p)^{-1}\, \overline{s_D^p}\, y_A^p.
\end{aligned}
\tag{11}
$$

- The derived $\hat{h}$ can be used in (8) to cancel a significant amount of SI.

Second stage: DRL based method

- The DRL agent located at the BS initializes the phaseshift, beamformers and transmit powers randomly, together called as **actions**
- Using these, the BS and ULue transmit and DLue and BS receives signals
- After receiving the signal at the BS, a significant amount of residual SI is cancelled[5] using a pilot signal (LSSIC); then the SINR at BS is obtained; used as **observation or state**
- Note, if an estimate of $\mathbf{H}_{AA}$ is already available at the BS, $\hat{h}$ can be estimated using this $\tilde{\mathbf{H}}_{AA}$ - $\tilde{\mathbf{H}}_{AA}$ based SI cancellation (HSIC)
- The **SINR at BS after SI cancellation** is used by the DRL agent for further learning in the second stage
- The SINR at DLue (a scalar) is also calculated and the SINR is fed back to the agent to be used in second stage as **observation or state**

[5] DLue is HD, so no SI for DLue

Second stage: DRL based method

- Weighted sum of the corresponding data rates of these two SINRs are used as **reward**
- Now based on this feedback, the DRL agent predicts a different set of actions, according to which again the signals are transmitted from BS and ULue, achieving a new pair of UL and DL SINR (after LSSIC/HSIC at the first stage for UL) which takes the system to a **next state**
- The quality of the beamformers learned by the DRL agent in the second stage depends on the SI-cancelled signal from the first stage
- At the same time, with accurate learning of the beamformers, the SI-cancellation is also better

Proposed: DRL-based complete online solution

This method predicts RIS phases, beamformers at BS, and transmit powers of BS and ULue such that the SI that is not canceled at the first stage is attempted to cancel by the predicted beamformers without the CSI knowledge

- How do we formulate it using some mathematical model?
- **Markov Decision Process (MDP)**

Formulating MDP

- An MDP has a **state** space $\mathcal{S}$, an **action** space $\mathcal{A}$, an initial distribution of space $p(\mathbf{s}^{\{1\}})$ and a stationary distribution for state transition that obeys Markov property i.e., $p(\mathbf{s}^{\{t+1\}}|\mathbf{s}^{\{t\}}, \mathbf{a}^{\{t\}}) = p(\mathbf{s}^{\{t+1\}}|\mathbf{s}^{\{t\}}, \mathbf{a}^{\{t\}}, \ldots, \mathbf{s}^{\{1\}}, \mathbf{a}^{\{1\}})$ and a **reward** function $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$.
- The algorithm (deployed at BS) gives action $\mathbf{a}^{\{t\}}$ based on the state $\mathbf{s}^{\{t\}}$ generated at a previous time step
- The environment reacts to these **actions** and gives back the SINRs as the **observations/states** indicate how good the actions are
- Finally, the **reward** is calculated and fed as input to the learning agent
- The SINR observations, along with the actions at time step t give the state for time step $(t + 1)$.

MDP for our system model

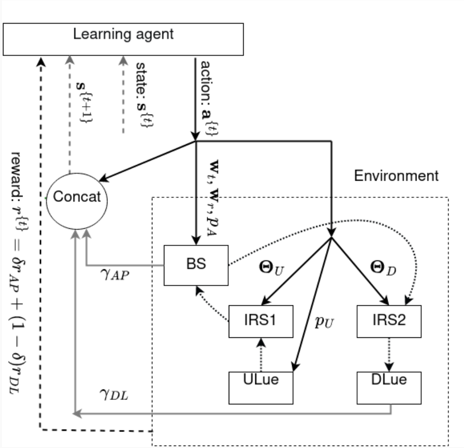


Figure 18: MDP formulation for RIS-based FD communication.

Contd. Second stage: DRL based method

- The job of the RL agent is to learn the policy $\boldsymbol{\pi} : \mathcal{S} \rightarrow \mathcal{A}$ from the observations corresponding to each of the actions while maximizing the return $\mathbb{r}^{\{t\}}(\gamma) = \sum_{t'=t}^{\infty} \gamma^{t'-t} r^{\{t\}}(\mathbf{a}^{\{t\}}, \mathbf{s}^{\{t\}})$ where $\gamma \in [0, 1]$ is the discounting factor.
- The learning agent calculates the quality of action by using Q-function given by $Q^{\pi}(s, a) = \mathbb{E}\left[\mathbb{r}_1(\gamma) | S_1 = s, A_1 = a; \boldsymbol{\pi}\right]$ indicating how rewarding each action a is when taken from a state s. At each timestep, the agent takes action which maximizes the Q-value
- A better agent is an algorithm that approximates the Q-value well by predicting a good action/policy
- The policy can be approximated by deep neural networks - **actor-critic method namely deep deterministic policy gradient (DDPG)**

DDPG

DDPG has two neural networks:

- An **actor-network** $\mathbb{A}$ parameterized by $\boldsymbol{\omega}^a$ which predicts the action $\mathbf{a}^{\{t\}}$ based on the current state $\mathbf{s}^{\{t\}}$
- A **critic network** $\mathbb{C}$ parameterized by $\boldsymbol{\omega}^c$ which computes $Q(\mathbf{s}^{\{t\}}, \mathbf{a}^{\{t\}})$ that is essentially the quality of the action taken by actor-network $\mathbb{A}$
- The agent encourages the actor-network to take **better actions through its feedback**
- The critic network $\mathbb{C}$ trains itself for **better prediction by observing the rewards after each action** [6]
- It first computes the Q-value of each action and then calculates the gradient of the error on its prediction of the Q-value of actions

[6] For more details regarding how these networks are trained, please refer to the paper.

Proposed actor network

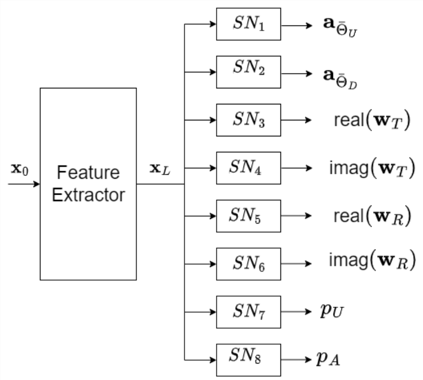


Figure 19: The proposed action predictor network. SN represents sub-network

Actor network

- The state for next time step $\mathbf{s}^{\{t+1\}}$ takes the SINR at BS and DLue $\gamma_{BS}^{\{t\}}$ and $\gamma_{DL}^{\{t\}}$, RIS phases $\bar{\Theta}_U^{\{t\}}$, $\bar{\Theta}_D^{\{t\}}$, transmit and receive beamformers $\mathbf{w}_T^{\{t\}}$ and $\mathbf{w}_R^{\{t\}}$ and transmit powers of the BS and ULue given by $p_A^{\{t\}}$ and $p_U^{\{t\}}$
- Input to actor network $\mathbf{x}_0$ is the state

$$\begin{aligned} s^{\{t\}} = [&\gamma_{BS}^{\{t-1\}}, \gamma_{DL}^{\{t-1\}}, \theta_{11}^{\{t-1\}}, \theta_{12}^{\{t-1\}}, \dots, \theta_{1N_1}^{\{t-1\}}, \theta_{21}^{\{t-1\}}, \theta_{22}^{\{t-1\}}, \dots, \theta_{2N_2}^{\{t-1\}}, \\ &\mathrm{real}(w_{t1}^{\{t-1\}}, w_{t2}^{\{t-1\}}, \dots, w_{tM_t}^{\{t-1\}}), \mathrm{imag}(w_{t1}^{\{t-1\}}, w_{t2}^{\{t-1\}}, \dots, w_{tM_t}^{\{t-1\}}), \\ &\mathrm{real}(w_{r1}^{\{t-1\}}, w_{r2}^{\{t-1\}}, \dots, w_{rM_r}^{\{t-1\}}), \mathrm{imag}(w_{r1}^{\{t-1\}}, w_{r2}^{\{t-1\}}, \dots, w_{rM_r}^{\{t-1\}}), p_A^{\{t-1\}}, p_U^{\{t-1\}}]. \end{aligned} \tag{12}$$

- The predicted action by the actor network for time step t is given by

$$\begin{aligned} \mathbf{a}^{\{t\}} = [&\theta_{11}^{\{t\}}, \theta_{12}^{\{t\}}, \dots, \theta_{1N_1}^{\{t\}}, \theta_{21}^{\{t\}}, \theta_{22}^{\{t\}}, \dots, \theta_{2N_2}^{\{t\}}, \mathrm{real}(w_{T1}^{\{t\}}, w_{T2}^{\{t\}}, \dots, w_{TM_t}^{\{t\}}), \\ &\mathrm{imag}(w_{T1}^{\{t\}}, w_{T2}^{\{t\}}, \dots, w_{TM_t}^{\{t\}}), \mathrm{real}(w_{R1}^{\{t\}}, w_{R2}^{\{t\}}, \dots, w_{RM_r}^{\{t\}}), \\ &\mathrm{imag}(w_{R1}^{\{t\}}, w_{R2}^{\{t\}}, \dots, w_{RM_r}^{\{t\}}), p_A^{\{t\}}, p_U^{\{t\}}]. \end{aligned} \tag{13}$$

Predicted actions

- **RIS phases**: The feature $\mathbf{x}_L$ is passed via the first two subnetworks, and the outputs are:

$$\mathbf{a}_{\bar{\Theta}_U} = \tanh(\mathbf{W}_{\bar{\Theta}_U}\mathbf{x}_L + \mathbf{b}_{\bar{\Theta}_U}), \text{ and } \mathbf{a}_{\bar{\Theta}_D} = \tanh(\mathbf{W}_{\bar{\Theta}_D}\mathbf{x}_L + \mathbf{b}_{\bar{\Theta}_D}). \tag{14}$$

  The tanh is used to get the normalized actions between $[-1, +1]$ are then shifted and scaled to take values in $[0, 2\pi]$.

- **Beamformers:** The in-phase and quadrature part of beamforming vectors can take any value between $[-1, +1]$, so the action corresponding to the in-phase and quadrature components of the transmit beamforming vector [7]

$$\mathbf{a}_{M_t,I} = \text{real}(\mathbf{w}_T) = \tanh(\mathbf{W}_{2,M_t,I}\text{ReLU}(\mathbf{W}_{1,M_t,I}\mathbf{x}_L + \mathbf{b}_{1,M_t,I}) + \mathbf{b}_{2,M_t,I}) \text{ and}$$
$$\mathbf{a}_{M_t,Q} = \text{imag}(\mathbf{w}_T) = \tanh(\mathbf{W}_{2,M_t,Q}\text{ReLU}(\mathbf{W}_{1,M_t,Q}\mathbf{x}_L + \mathbf{b}_{1,M_t,Q}) + \mathbf{b}_{2,M_t,Q}),$$

[7] similar way for receive beamforming vectors

Contd. Predicted actions

- **Transmit powers:** The output of the last two sub-networks are:

$$a_{p_U} = \tanh(\mathbf{w}_{p_U}\mathbf{x}_L + b_{p_U}), \text{ and } a_{p_A} = \tanh(\mathbf{w}_{p_A}\mathbf{x}_L + b_{p_A}), \tag{15}$$

  The output from sub-network is in the range $[-1, +1]$ which is shifted and scaled to the range $[0, P_u]$ and $[0, P_a]$ before using them in the environment

$$p_a = (a_{p_A} + 1)/2 \times P_a, \text{ and } p_u = (a_{p_U} + 1)/2 \times P_u, \tag{16}$$

  where P_u and P_a are the maximum allowable transmit powers of the ULue and BS, respectively.

- Addition of Gaussian noise to the action explores the action space well which gives faster convergence

Alternative solution using MRC-ZF - needs CSI

- Maximize sum of UL and DL SINR by maximizing the SNR towards reception

$$\begin{aligned} \mathcal{P}_2 : \quad & \max_{\mathbf{w}_T} \; r_{DL} + r_{BS} \\ & \text{s.t.} \; ||\mathbf{w}_R^{MRC} \mathbf{H}_{AA} \mathbf{w}_T||^2 = 0, \quad ||\mathbf{w}_T||^2 = 1, \end{aligned} \tag{17}$$

where

$$\mathbf{w}_r^{MRC} = \frac{(\mathbf{h}_{AU} + \mathbf{F}_{AI} \mathbf{\Theta}_U \mathbf{f}_{IU} + \mathbf{G}_{IA}^T \mathbf{\Theta}_D \mathbf{g}_{IU})^H}{||\mathbf{h}_{AU} + \mathbf{F}_{AI} \mathbf{\Theta}_U \mathbf{f}_{IU} + \mathbf{G}_{IA}^T \mathbf{\Theta}_D \mathbf{g}_{IU}||}. \tag{18}$$

- Note that the knowledge of every interferer is not available so $\mathbf{g}_{IU}$ is not available

Contd.

- We want to minimize the self-interference using Zero-Forcing. The precoder $\mathbf{w}_T$ is in the orthogonal complement space of $\mathbf{w}_R^{MRC}\mathbf{H}_{AA}$. The orthogonal projection onto the orthogonal complement of the column space of $\mathbf{w}_R^{MRC}\mathbf{H}_{AA}$ is given by[8]

$$\Pi^{\perp}_{\mathbf{H}_{AA}^{\dagger}\mathbf{w}_R^{MRC\dagger}} = \mathbf{I}_{M_t} - \mathbf{H}_{AA}^{\dagger}\mathbf{w}_R^{MRC\dagger}(\mathbf{w}_R^{MRC}\mathbf{H}_{AA}\mathbf{H}_{AA}^{\dagger}\mathbf{w}_R^{MRC\dagger})^{-1}\mathbf{w}_R^{MRC}\mathbf{H}_{AA}.$$

- The optimal solution for transmit beamforming is

$$\mathbf{w}_T^{ZF} = \frac{\Pi^{\perp}_{\mathbf{H}_{AA}^{\dagger}\mathbf{w}_R^{MRC\dagger}}(\mathbf{h}_{DA} + \mathbf{g}_{DI}\mathbf{\Theta}_D\mathbf{G}_{IA} + \mathbf{f}_{DI}\mathbf{\Theta}_U\mathbf{F}_{AI}^T)^{\dagger}}{||\Pi^{\perp}_{\mathbf{H}_{AA}^{\dagger}\mathbf{w}_R^{MRC\dagger}}(\mathbf{h}_{DA} + \mathbf{g}_{DI}\mathbf{\Theta}_D\mathbf{G}_{IA} + \mathbf{f}_{DI}\mathbf{\Theta}_U\mathbf{F}_{AI}^T)^{\dagger}||}. \tag{19}$$

- This work can be extended to multiple users, too, and precoding will help us to beamform towards every user using the same antenna array.

[8] † represents conjugate transpose

Simulation setup and results

- BS situated at $(0, 0)$
- No direct path from the BS to the UL and DL users
- To promote communication, two RISs are placed at $(50, 22)$ and $(50, -22)$
- Static ULue and DLue at $(50, 20)$ and $(50, -20)$ respectively
- Maximum transmit power allowable at ULue and BS are $p_U^{max} = 50$ mW and $p_A^{max} = 1$ W
- For moving UEs, UEs move in a square area of 100 m^2 with an average speed of $1m$ per time step
- BS-RIS and the RIS-user channels have LOS, so modeled as Rician channel

Contd.

- For example, the channel between RIS2 and DLue is given by

$$\mathbf{g}_{DI} = \sqrt{\frac{\beta_{UI}}{1+\beta_{UI}}}\mathbf{g}_{DI}^{LOS} + \sqrt{\frac{1}{1+\beta_{UI}}}\mathbf{g}_{DI}^{NLOS}. \tag{20}$$

  where β_{UI} is the Rician K-factor for the channels between RIS and users [9]
- The other channels don't have an LOS, so modeled as Rayleigh
- The path loss between two points with distance d is modeled as,

$$PL(f_c, d)_{dB} = -20\log_{10}(4\pi f_c/c) - 10\alpha\log(d/D_0), \tag{21}$$

  where f_c is the carrier frequency, $D_0 = 1$ m, α is the path loss exponent.

[9] For more detail on how the channels are simulated in our setup, please refer to the paper

Contd.

- The bandwidth where the system operates is 100 MHz, the noise power density is -174 dBm/Hz and the carrier frequency f_c is 3.5 GHz.
- For the DRL agent, the discounting factor $\gamma = 0.6$, buffer size $\tau = 10000$, and the learning rates of actor and critic networks are 0.0001 and 0.001 respectively
- The experiment is run for initial 50 episodes and each episode has 1000 time steps. The results are averaged over 4 independent runs
- The experiments are performed on an NVIDIA GeForce RTX 2080 Ti GPU
- The benchmark metrics used for studying the performance are UL and DL data rates with the unit bits/sec/Hz.

Our method and baseline competitor methods

- **RandPSBF**: Agent does not receive any SINR feedback from the environment; predict the RIS phases, the beamformers, and transmit powers randomly
- **OUPSBF**: Makes use of the same DRL framework as ours, except for the action-noise where it adds the Ornstein Uhlenbeck noise to the RIS phases and beamformer
- Proposed Minimum Signalling Feedback (MSF) DRL method with LSSIC and HSIC (**MSF-DRL-LSSIC** and **MSF-DRL-HSIC**):
  - The critic network and the feature extractor actor-network are feed-forward networks with two layers, each with 100 neurons.
  - At the beginning of every episode[10], the MSF-DRL agent chooses actions randomly.
  - MSF-DRL uses a Gaussian action noise with zero mean and linearly decaying standard deviation (SD) with an initial SD of 0.3 decaying over 100 episodes.

[10] Episode is an independent game or sequence of states where the agent and environment interacts. It starts at an initial state and ends at a terminal state.

Contd.

- **PerfCSI-DRL** and **NoisCSI-DRL**:
  - A hypothetical experiment called "PerfCSI-DRL" to predict only the RIS phases where the perfect CSI knowledge including residual SI is available to the agent therefore serves as benchmark
  - The agent calculates the beamformers based on the ZF and MMSE principle that needs perfect CSI
  - Periodically receive the CSI to calculate the beamformers and therefore incurs overhead
  - CSI estimation methods may not be exact - NoisCSI-DRL
- **MSF-DRL-LSSIC-pos**: Along with previous actions, UL SINR and DL SINR, the past positions for a window is also given to the MSF-DRL-LSSIC agent

Comparison with static UE

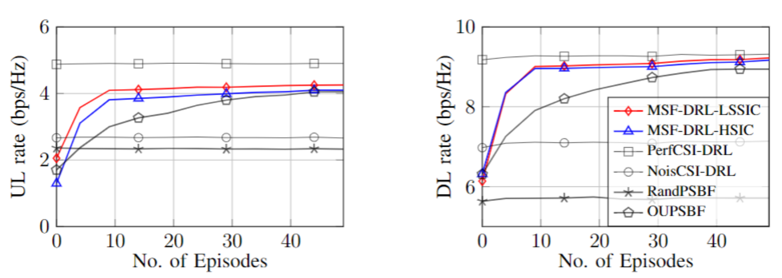


Figure 20: Rate evolution during learning in the static UE scenario. MSF-DRL-LSSIC and HSIC learns to predict and tries to reach the benchmark PerfCSI-DRL, performs much better than the case of NoisCSI-DRL.

Comparison with moving UE

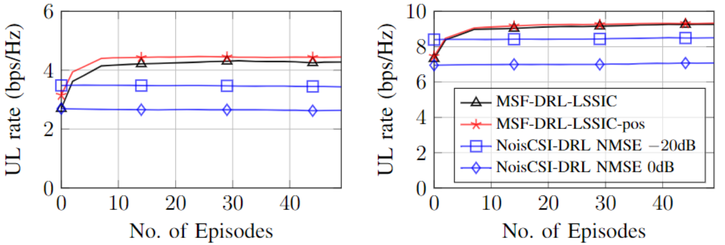


Fig. 7: Rate evolution during learning in the moving UE scenario

Figure 21: Rate evolution during learning in the moving UE scenario. In moving UE case also, the proposed method does not need any CSI and the knowledge of residual SI still performs better than NoisCSI-DRL methods.

Contributions

- The two-stage learning algorithm **assumes the absence of a good SI mitigation scheme** and the costly CSI overhead but **still performs almost as well as perfect CSI-based semi-oracle DRL methods**.
- To overcome the challenge of SI, a **least square-based method** is proposed when a good estimate of SI is not present.
- The performances are shown in scenario with moving UEs as well

Summary

- Discussed beamforming and beam steering
- Application of beamforming in modern wireless communication systems - multi-user and multi-path
- Analog digital and hybrid beamformers -introduction to precoding
- An application of beamforming in full-duplex communication using RIS

Sources

Some of the slides and figures are picked from the following:

- MATLABs introduction to beamforming
- Some simple explanations by Iain Collings
- An introduction to Radio Beamforming with mathematical expressions
- Van Trees, Harry L. Optimum array processing: Part IV of detection, estimation, and modulation theory. John Wiley & Sons, 2002.
- Nayak, Nancy, Sheetal Kalyani, and Himal A. Suraweera. "A DRL Approach for RIS-Assisted Full-Duplex UL and DL Transmission: Beamforming, Phase Shift and Power Optimization." arXiv preprint arXiv:2212.13854 (2022).

Thank you!

DDPG contd.

- To get stable, uncorrelated gradients for policy improvement, DDPG maintains a replay buffer of finite size τ and samples the observations from the buffer in mini-batches to update the parameters.
- At each timestep, the state $\mathbf{s}^{\{i\}}$ and the action taken $\mathbf{a}^{\{i\}}$ along with the reward obtained $r^{\{i\}}$ and the next state $\mathbf{s}^{\{i+1\}}$ is stored as an experience $(\mathbf{s}^{\{i\}}, \mathbf{a}^{\{i\}}, r^{\{i\}}, \mathbf{s}^{\{i+1\}})$ to the buffer $\mathcal{B}$.
- DDPG also uses target networks with parameters $\bar{\boldsymbol{\omega}}^a$ and $\bar{\boldsymbol{\omega}}^c$ to avoid divergence in value estimation
- For the critic network $\mathbb{C}(\cdot|\boldsymbol{\omega}^c)$ to compute the Q-value for each state action-pair, an estimate of return for state s_i in each sample is computed as

$$y^{\{i\}} = r^{\{i\}} + \gamma \mathbb{C}(\mathbf{s}^{\{i+1\}}, \mathbb{A}(\mathbf{s}^{\{i+1\}}|\bar{\boldsymbol{\omega}}^a)|\bar{\boldsymbol{\omega}}^c). \tag{22}$$

DDPG contd.

- Once we observe a reward r_i after taking an action a_i, based on the estimate for return, the mean squared Bellman error (MSBE) is computed as

$$\mathcal{L} = \frac{1}{N} \sum_i \left(y^{\{i\}} - \mathbb{C}(\mathbf{s}^{\{i\}}, \mathbf{a}^{\{i\}} | \boldsymbol{\omega}^c) \right)^2, \tag{23}$$

  where, $\mathbb{C}(\cdot)$ is the predicted output value of critic network with parameter $\boldsymbol{\omega}^c$ for the state $\mathbf{s}^{\{i\}}$ and action $\mathbf{a}^{\{i\}}$ before seeing the reward.

- Then, the critic network parameters are updated as

$$\boldsymbol{\omega}^c \leftarrow \boldsymbol{\omega}^c - \eta_c \nabla_{\boldsymbol{\omega}^c} \mathcal{L}, \tag{24}$$

  where $\eta_c \ll 1$ is the stepsize for the stochastic update.

- In our case the critic network is a **fully connected network** which gives a scalar output as Q-value

DDPG contd.

- For the actor-network, the update depends on both the gradient of action as well as the improvement in Q-value. The final update for updating parameters of actor-network ω^a is given by

$$\omega^a \leftarrow \omega^a + \eta_a \frac{1}{N} \sum_i \left(\nabla_{\omega^a} \mathbb{A}(s) \nabla_a \mathbb{C}(s, a)|_{a=\mathbb{A}(s)} \right), \tag{25}$$

  where $\eta_a \ll 1$ is the update stepsize.
- Finally, the target network parameters are updated in every U timestep to provide stable value estimates using an exponentially weighted update as $\bar{\omega}^c \leftarrow \lambda\omega^c + (1-\lambda)\bar{\omega}^c$, and $\bar{\omega}^a \leftarrow \lambda\omega^a + (1-\lambda)\bar{\omega}^a$, with $\lambda \ll 1$.

Quantized MSF-DRL-LSSIC

- Every element of the real-valued phases is to be represented with just n bits so that instead of a real (i.e., 64 bit) phase-shift value, only n ($n << 64$) bit information is transmitted from the BS to the RIS
- Number of phase values that each of the passive elements can take is $Q = 2^n$ and are given by $\mathbf{p} = 2\pi/2^n \times [0, \ldots, 2^n - 1]$ radians
- The architecture of the first two sub-networks for predicting RIS phases are now modified so that instead of a single phase value, they predict the probabilities of picking that phase value out of the possible values of discrete RIS phase angles of length $Q = 2^n$
- The sub-network for predicting the phase-shift of RIS1 takes the feature $\mathbf{x}_L$ as input and passes it through N_1 fully connected layers, each with 2^n neurons, followed by a softmax activation on each of the N_1 outputs.

Results with quantized phase MSF-DRL-LSSIC

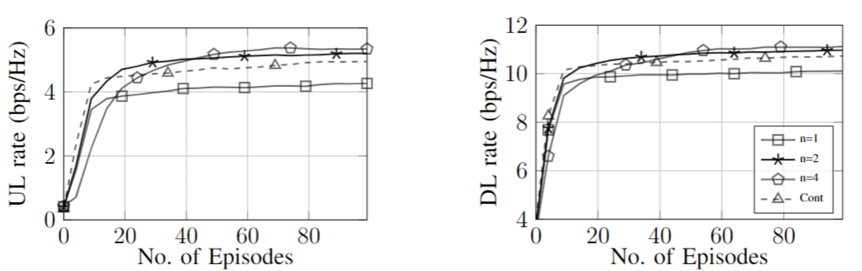


Figure 22: Effect of quantization on phaseshifts of RISs on MSF(Q)-DRL methods. When $n = 1$, the phase-shifts can take $Q = 2^n = 2$ values $\{0, \pi\}$; for $n = 0$, the phase-shifts are fixed to 0 and only the beamformers are learned.