Part A: Single Source (Basics Concepts of RTF)

To calculate the relative room, transfer function (RTF) between two microphone channels, you typically want to analyze the difference in the acoustical signals captured by each microphone. The RTF gives you insights into the spatial relationship between the two microphones, including any delays, gains, or distortions caused by the room.

Below is described the step-by-step (as I understood) process to calculate the relative room transfer function between two microphone channels

1. Collect the signals from both microphones

You have two microphones, each recording signals $x_1(t)$ and $x_2(t)$, where t is the time variable. These signals are the acoustic signals recorded by the microphones (two channels)

2. Preprocess the signals

Ensure the signals are properly aligned. This may involve removing any DC component and normalizing the signals to ensure that the comparison is valid (remember to normalize relatively otherwise you will be introducing additional energies)

3. Compute the cross-correlation between the two signals

Compute the Cross-correlation which is the method to measure the similarity between two signals as a function of time lag

$$R_{12}(\tau) = \int x_1(t) \cdot x_2(t+\tau)dt$$

This gives the time delay τ between the signals. The peak of the cross-correlation function indicates the time shift between the two signals

4. Estimate the time delay

From the cross-correlation function, extract the time lag τ_{max} at which the peak occurs. This is the estimated relative delay between the two microphones. The delay corresponds to the time it takes for the sound to travel between the microphones.

5. Perform spectral analysis (Fourier transform)

Apply the Fourier transform to both signals to analyze their frequency content. This will convert the signals from the time domain into the frequency domain. In the frequency domain, you can better observe the relationship between the two signals in terms of phase and amplitude.

6. Calculate the Room Transfer Function

The Room Transfer Function $H_{12}(f)$ between the two microphones can be computed as the ratio of the frequency domain representations of the two signals:

$$H_{12}(f) = \frac{X_2(f)}{X_1(f)}$$

This gives you the relative gain and phase shift between the two microphones for each frequency. It describes how the acoustical environment has modified the signals as they traveled from the sound source to the microphones.

Magnitude of $H_{12}(f)$ represents the relative amplitude gain (or attenuation) between the two microphones across different frequencies. Phase of $H_{12}(f)$ represents the phase shift between the signals, which can be influenced by the spatial positioning of the microphones relative to the sound source.

7. Inverse Filtering

We need to find the inverse transfer function (e.g., in our case, to recover the sound at one microphone from the other), we can compute the inverse of $H_{12}(f)$

8. Considerations:

Room effects: The transfer function will also include the effects of room acoustics such as reflections and reverberations. This is especially important in a reverberant environment.

Microphone placement: The relative positioning of the microphones will affect the transfer function, as sound may travel differently to each microphone depending on their distances and angles from the sound source.

Part B: Crosstalk Cancellation (CTC): Inverse Filtering based (Basic Concepts)

Now in the context of crosstalk cancellation (CTC) for two microphones with two sound sources in the room, the Room Transfer Function (RTF) plays a crucial role in modeling the acoustical relationship between the microphones and the sound sources

Crosstalk Cancellation Concept

When there are two sound sources in a room, each microphone captures a mixture of the sounds from both sources. If you want to isolate the sound from one source while suppressing the sound from the other (crosstalk cancellation), you need to model the way sound from each source propagates to both microphones. The RTF is key in this process because it describes how each microphone receives the signals from each sound source.

Key Steps for Crosstalk Cancellation Using RTF:

1. Model the Room Transfer Functions for Each Sound Source:

- Let's assume you have two sound sources in the room: S_1 and S_2
- Each microphone captures a mixture of sounds from both sources, influenced by the room's acoustics.
- For each microphone (i.e. M_1 and M_2), you have the following transfer functions:
  - $H_{S_1 \to M_1}$: The RTF from source S_1 to microphone M_1
  - $H_{S_2 \to M_1}$: The RTF from source S_2 to microphone M_1
  - $H_{S_1 \to M_2}$: The RTF from source S_1 to microphone M_2
  - $H_{S_2 \to M_2}$: The RTF from source S_2 to microphone M_2

These transfer functions capture the effects of the room acoustics (such as reflections, absorption, and diffraction) and the relative positioning of the microphones and sources.

2. Record the Microphone Signals:
   - Let the signals captured by the microphones be:
     - $x_1(t)$: The signal recorded by M_1 microphone 1
     - $x_2(t)$: The signal recorded by M_2 microphone 2

These signals are a mixture of the sound from both sources:

$$x_1(t) = H_{S_1 \to M_1} * S_1(t) + H_{S_2 \to M_1} * S_2(t)$$

$$x_2(t) = H_{S_1 \to M_2} * S_1(t) + H_{S_2 \to M_2} * S2(t)$$

Here, $*$ represents convolution, and $S_1(t)$ and $S_1(t)$ are the signals from the sources

3. Inverse Filter Design for Crosstalk Cancellation: To isolate the sound from one source and cancel the other, we need to design an **inverse filter** for each microphone signal. The idea is to cancel the signal from one source (i.e. S_2) by creating a filter that models how that source would contribute to the microphone signal, and then subtracting it from the microphone's output.

   For microphone 1:

   - To cancel the contribution of S_2 we need to subtract the influence of S_2 from the microphone signal.
   - The **inverse filter** for microphone 1, denoted as $H_{S_2 \to M_1}$ is designed based on the transfer function $H_{S_2 \to M_1}$. This filter can be derived from the room transfer function between the second source and the first microphone.
   - The signal from microphone 1 after cancellation would be:

$$y_1(t) = x_1(t) - \widehat{H}_{S_2 \to M_1} * x_2(t)$$

   - The term $\widehat{H}_{S_2 \to M_1} * x_2(t)$ represents the modeled contribution of S_2 at microphone 1. By subtracting this, we aim to cancel out the crosstalk from source 2

For microphone 2:

- Similarly, for microphone 2, we design an inverse filter $H_{S_1 \to M_2}$ to cancel the contribution of S_1 from the second microphone signal as

$$y_2(t) = x_2(t) - \widehat{H}_{S_1 \to M_2} * x_1(t)$$

- Again, $\widehat{H}_{S_1 \to M_2} * x_1(t)$ is the modeled contribution of S_1 at microphone 2, which is subtracted to cancel the crosstalk.

4. Formulate the CTC Filter Using RTF: The inverse filters $\widehat{H}_{S_2 \to M_1}$ and $\widehat{H}_{S_1 \to M_2}$ can be designed based on the transfer functions $H_{S_2 \to M_1}$ and $H_{S_1 \to M_2}$, respectively. These are typically derived by using the least squares or adaptive filtering techniques. The goal is to estimate the exact transfer function from the unwanted source and apply it as an inverse filter

Part C: How Relative Impulse Response (RIR) is Used in Blind Source Separation (BSS) with Two Microphones

In the case of two microphones, the Relative Impulse Response (RIR) describes how the signal from a source propagates differently to each microphone due to spatial differences, reflections, and reverberation. This concept plays a crucial role in Blind Source Separation (BSS), which aims to separate multiple sound sources from mixed microphone recordings without knowing the mixing parameters beforehand.

1. Two-Microphone Mixing Model:
   Consider a room where two sound sources ($s_1(t)$ and $s_2(t)$) are being recorded by two microphones ($x_1(t)$ and $x_2(t)$) placed at different locations. The recorded signals can be modeled as:

$$x_1(t) = h_{11}(t) * s_1(t) + h_{12}(t) * s_2(t) + n_1(t)$$

$$x_2(t) = h_{21}(t) * s_1(t) + h_{22}(t) * s_2(t) + n_2(t)$$

   Where:
   $h_{ij}(t)$ is the impulse response from source jj to microphone i
   $n_1(t)$ and $n_2(t)$ are noise components

   Each microphone records a mixture of both sources, and the goal of BSS is to estimate the individual source signals $s_1(t)$ and $s_2(t)$

2. Role of Relative Impulse Response (RIR):
   BSS algorithms use RIR information to separate sources effectively.

   A. Estimating the Mixing System (Cross-Talk Estimation):

- Since the two microphones capture the sources with different filtering and delay effects, estimating the Relative Transfer Function (RTF) (frequency-domain equivalent of RIR) helps determine how signals are mixed.
- The RTF is estimated using techniques like Generalized Eigenvalue Decomposition (GEVD) or Blind System Identification.
- Knowing the RIR/RTF allows us to construct spatial filters to suppress cross-talk.

B. Beamforming for Source Enhancement:
- Using RIR, we can apply beamforming (e.g., MVDR or GSC beamforming) to enhance one source while suppressing the other.
- The delay-and-sum beamformer aligns signals from a specific direction while canceling interference based on the RIR model.

C. Frequency-Domain Independent Component Analysis (FD-ICA):
- RIR helps estimate the mixing filters in the frequency domain.
- The BSS algorithm then applies Independent Component Analysis (ICA) or Independent Vector Analysis (IVA) to separate sources.

D. Multichannel Wiener Filtering:
- Once the RIR is estimated, Multichannel Wiener Filtering (MWF) can be applied to suppress unwanted signals while preserving target signals.

E. Dereverberation for Improved BSS:
- In reverberant environments, multiple reflections increase cross-talk, making separation harder.
- Using RIR, we can apply Weighted Prediction Error (WPE) Dereverberation to suppress late reflections and improve separation quality.

Part D: Estimating the Mixing System (Cross-Talk Estimation) in Two-Microphone BSS

The goal of cross-talk estimation in Blind Source Separation (BSS) is to characterize how multiple sources are mixed in the observed microphone signals. This involves estimating the mixing filters, often represented in terms of the Relative Impulse Response (RIR) or its frequency-domain counterpart, the Relative Transfer Function (RTF)

Mixing Model in Two-Microphone System:

See Part C Section 1

Method 1: **Relative Impulse Response (RIR) and Relative Transfer Function (RTF):**

Instead of estimating each impulse response individually, we estimate the Relative Impulse Response (RIR) as

$$RIR_{s_1}(t) = \frac{h_{21}(t)}{h_{11}(t)}$$

$$RIR_{s_2}(t) = \frac{h_{22}(t)}{h_{12}(t)}$$

These describe the relative filtering effects between microphones. In the frequency domain, we define the Relative Transfer Function (RTF) as:

$$RTF_{s_1}(f) = \frac{H_{21}(f)}{H_{11}(f)}$$

$$RTF_{s_2}(f) = \frac{H_{22}(f)}{H_{12}(f)}$$

By estimating the RTF, we can describe how each source leaks into the second microphone relative to the first, which helps in constructing filters for separation.

Method 2: **Estimating RTF Using Cross-Power Spectra:**

A common way to estimate the RTF is by using cross-power spectral density (CPSD) of the recorded signals. The cross-power spectral density between $x_1(t)$ and $x_2(t)$ is:

$$S_{x_1x_2}(f) = H_{11}(f)S_{s_1s_1}(f)H_{21} * (f) + H_{12}(f)S_{s_2s_2}(f)H_{22} * (f)$$

With $S_{s_1s_1}(f)$ and $S_{s_2s_2}(f)$ are power spectral densities of the sources.

If we assume the sources are uncorrelated, we can estimate $RTF_{s_1}(f)$ as

$$RTF_{s_1}(f) = \frac{S_{x_1x_2}(f)}{S_{x_1x_1}(f)}$$

This provides a practical way to estimate the RTF without requiring explicit knowledge of the original sources.

Method 3: **Eigenvalue-Based RTF Estimation:**

Another approach involves using Generalized Eigenvalue Decomposition (GEVD), which is useful in reverberant environments. Given the spatial covariance matrices:

$$R_x(f) = E[x(f)x^H(f)]$$

With $x(f) = [X_1(f), X_2(f)]^T$ as the vector of microphone signals, we solve as

$$R_x(f)v = \lambda R_n(f)v$$

Where $R_n(f)$ is the noise covariance matrix and v is the eigenvector corresponding to the largest eigenvalue.

The estimated RTF is given by

$$RTF(f) = \frac{v_2}{v_1}$$

This method is particularly effective in noisy and reverberant environments.

Part E: Practical Implementation of Cross-Talk Estimation Using Relative Transfer Function (RTF) in BSS

To estimate cross-talk between two microphones and use it for Blind Source Separation (BSS), we can follow a practical approach using short-time Fourier transform (STFT), cross-power spectral estimation, and Generalized Eigenvalue Decomposition (GEVD).

Step 1: Collect Audio Data:

We have the code in Repository and recorded the different sounds from two loudspeakers and two channel microphone. Let's us load them as x_1 and x_2.

Step 2: Compute STFT for Time-Frequency Representation:

Since room reverberation and mixing are convolutive, we work in the frequency domain. We compute the Short-Time Fourier Transform (STFT). Write a code for this part in our main repository.

Step 3: Estimate Cross-Power Spectral Density (CPSD):

To estimate the Relative Transfer Function (RTF), we compute the cross-power spectral density (CPSD) and auto-power spectral density (APSD). Again write the code for this part as well.

Step 4: Estimate Mixing Model Using Eigenvalue Decomposition (GEVD):

For more robust estimation, we use Generalized Eigenvalue Decomposition (GEVD) on the spatial covariance matrix. Again write the code for this part as well.

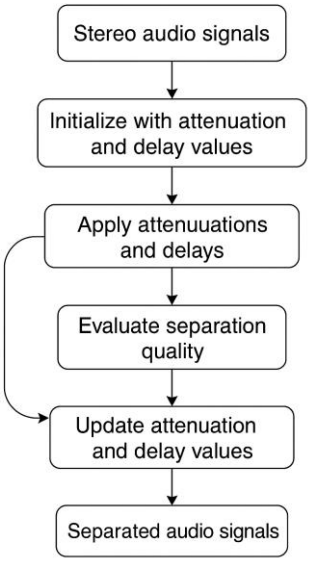
Step 5: Apply Wiener Filtering for Cross-Talk Cancellation: (One Approach)

Once the RTF is estimated, we can use a Wiener filter as one method to suppress unwanted signals.

Step 6: Integrate the RTF into Our Unmixing Matrix for Cross-Talk Cancellation: (AMS Approach)

Think about this and let us discuss

Below is the flow chat of our BSS code “ICAabskl_puredelay_online2.py”



Function Flow Chart