

# 情報管理

第7回：【番外編】勾配降下法の応用  
エコーキャンセリングによる雑音除去

# 今回の講義内容

今回は教師有りクラスタリングの方法としてロジスティック回帰を説明しました。

ロジスティック回帰関数を学習するための方法として、二乗誤差最小化基準と勾配降下法について解説しました。

今回は、二乗誤差最小化＋勾配降下法の応用例として、雑音除去技術のひとつである「エコーキャンセリング」について解説します。

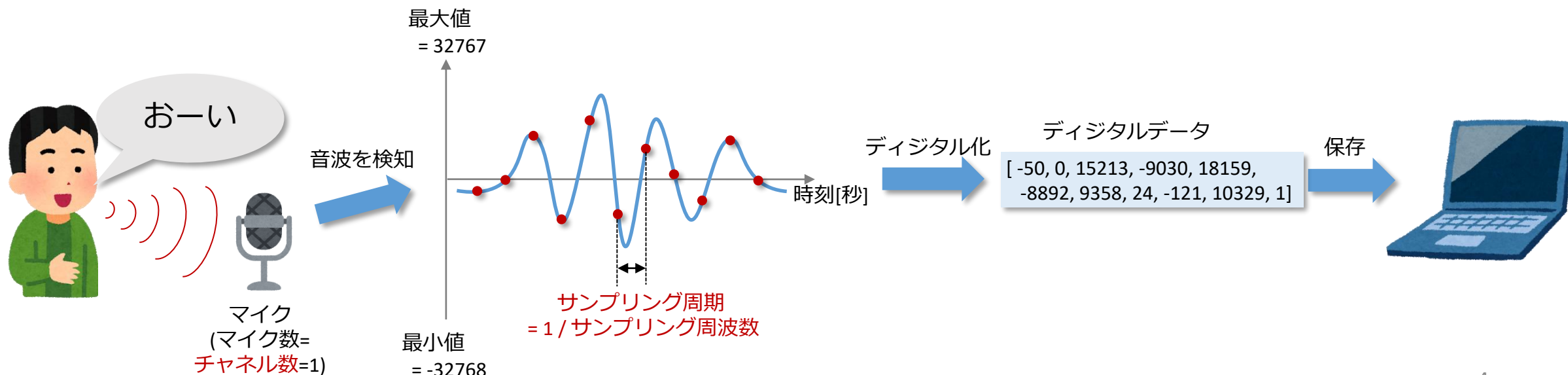
前回までとは打って変わって信号処理の知識なども出てきますので、理解できるように頑張りましょう。

事前準備

Python で音声データを扱おう

# 音声のデータとは？

- 音声は、空気を振動して伝わる「波」です。
- 音波はマイクによって検知されます。マイクの数「チャンネル数」と呼びます。今回はチャンネル数=1（モノラル信号）の音声のみを扱います。
- マイク信号をコンピュータに取り込むため、デジタル信号に変換します。
  - 「何秒間隔でデータ化するか」を「サンプリング周期(秒)」と呼び、その逆数を「サンプリング周波数(Hz)」と呼びます。  
本講義では、サンプリング周波数=8000Hz（つまり、1秒間に8000個の値をデータ化）の音声を扱います。
  - 各時刻における波の大きさは、多くの場合16bit整数(-32768～32767)に表現(量子化と呼ぶ)されて、コンピュータに保存されます。



# Pythonで音声データを処理してみよう

第三回のレポート課題では、音声データをcsvファイルにしたものを処理していましたが、今回は実際に音声データとして使われる「wavファイル」を使用します。

07\_01\_wave\_file.ipynb を動かして、音声データをPythonで入出力してみましよう。

このプログラムでは音声の逆再生音声を作成します。

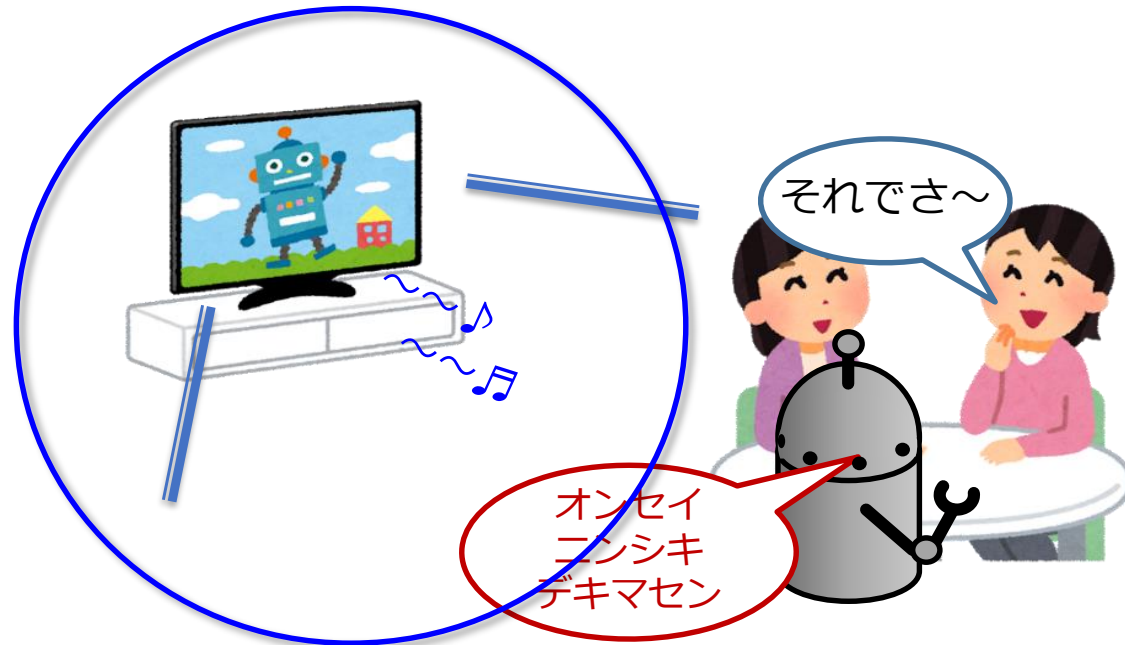
# エコーキャンセリング技術とは

# エコーキャンセリング技術とは？

雑音除去技術とは、マイクに収録された雑音を抑圧し、音声を聞き取りやすくする技術です。

エコーキャンセリングは、**スピーカから出力される雑音の除去**に適した雑音除去技術の一つです。

除去したい雑音  
(テレビのスピーカから出力される音)



# エコーキャンセリングのデモ

(英語のラジオニュースを聞きながらマイクに話しかけているシーンを想定)





# スピーカから出力される雑音って？

例えば、以下のようなケースがあります。

- カーナビゲーションシステムや、スマートフォン、テレビなどが、音を再生している状態で音声認識を行う場合。
- ロボットが発話中に割り込んで話しかけて音声認識を行う場合。
- 電話での自分の話し声が、相手の受話器のマイクを回り込んで戻ってくる。

再生している音楽が雑音になる



スピーカ

ロボット自身の発話が雑音になる

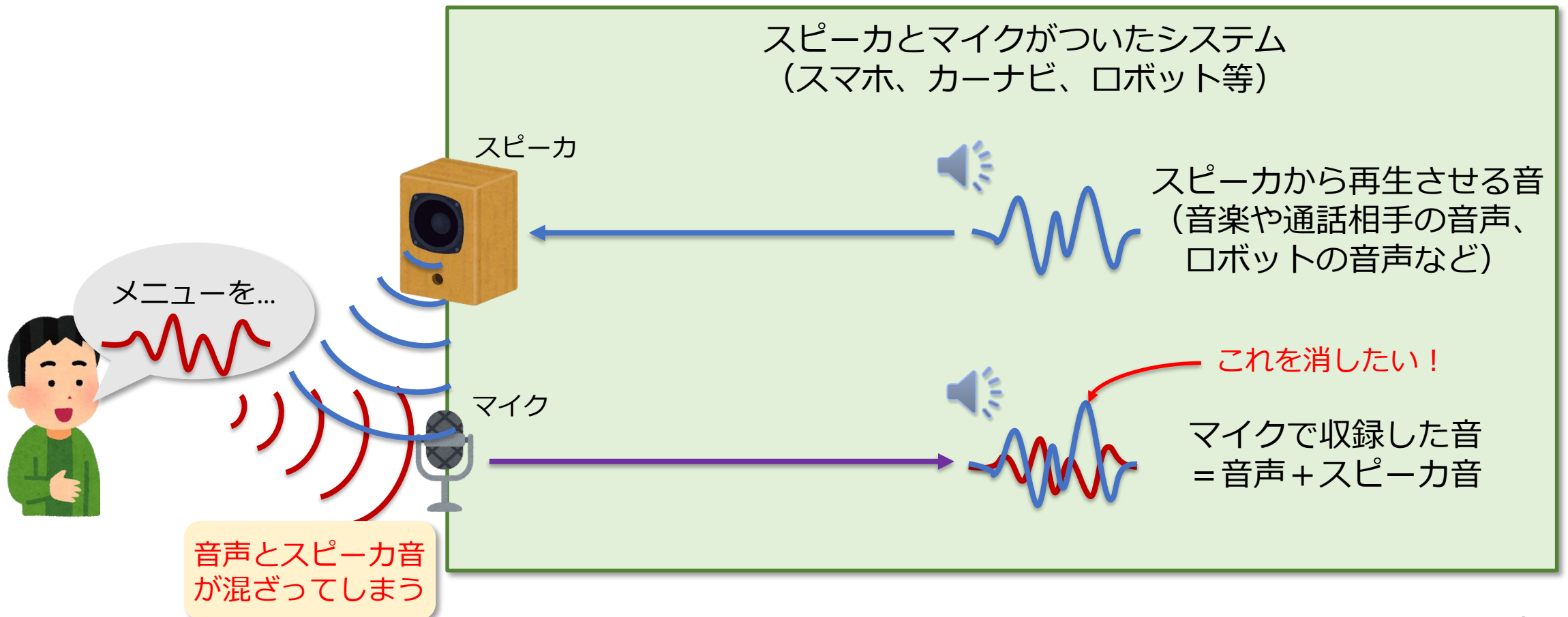


電話で自分の声が聞こえてくる



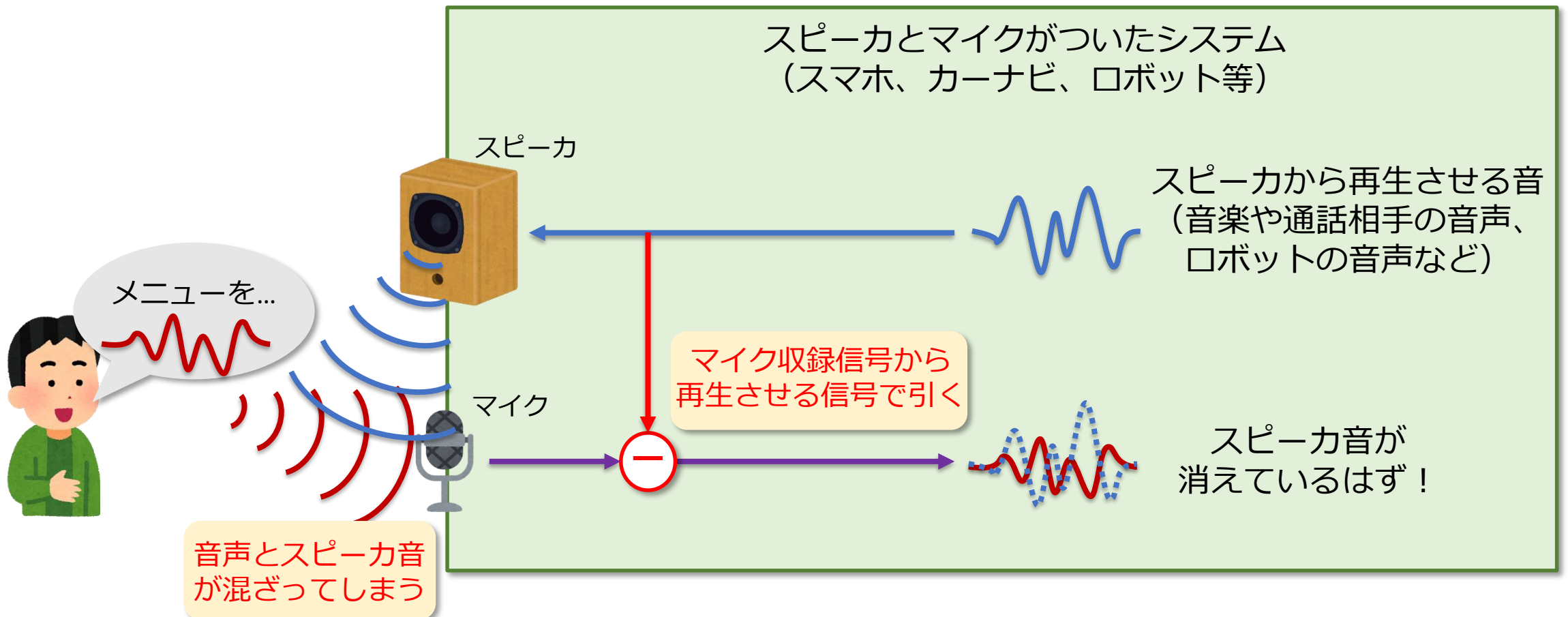
# 問題を図式化すると・・・

以下のようにになります。マイクで収録した音からスピーカ出力音を消すことが目的です。



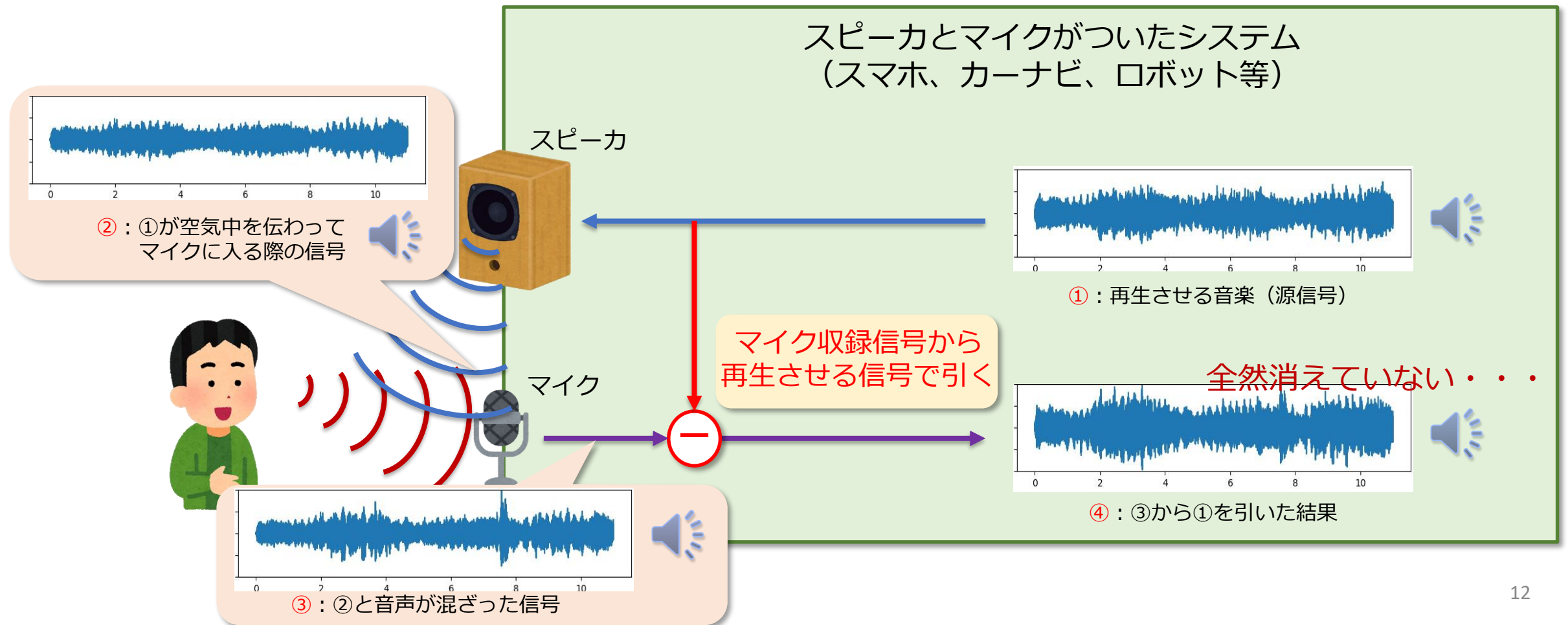
# 基本アイディア

再生させる音の信号は持っているんだから、マイク収録信号から引けばスピーカ音が消えるはず！



# 本当に消えるの？

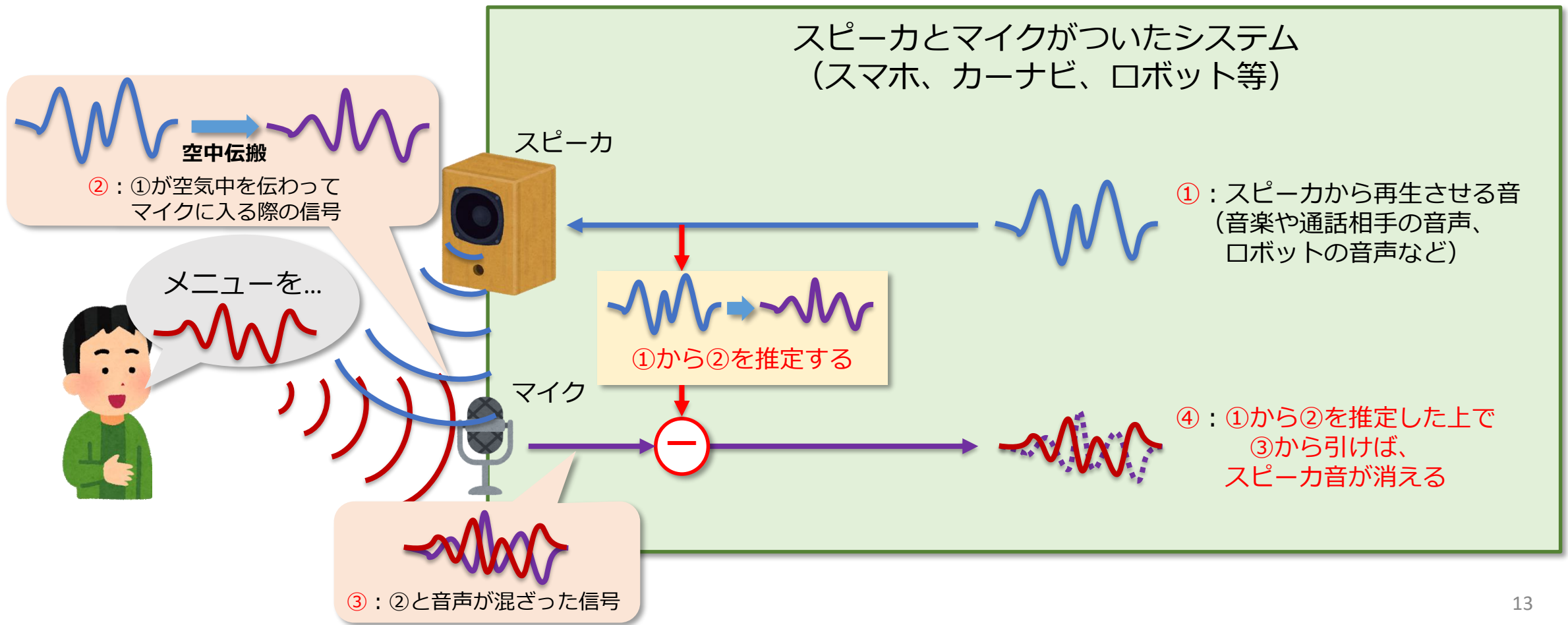
全く消えません。再生させる源信号（①）と、その信号がスピーカから空气中を伝わってマイクに収録された信号（②）では全く異なる波形になっているからです。



# じゃあどうすればいいの？

再生させる源信号(①)から、その信号が空气中を伝わってマイクに収録される信号(②)を推定できれば、正確にスピーカ音を消せるはずです。

いかにして①から②を正確に推定するかが、エコーキャンセリングの肝となります。



# エコーキャンセリングの考え方（結論）

$x$ : スピーカから再生させる元々の波形（音楽ファイルなど）

$s$ : 人の音声

$y$ : マイクで収録された音声（スピーカ音 + 人の音声）

間違った考え方：  $y = s + x$  だから、 $y - x$  で  $s$  が取り出せる！

正しい考え方：  $x$  はスピーカからマイクへ空中伝播する過程で波が  $z$  に変化する。  
つまり  $y = s + z$ 。ただし、 $z = f(x)$   
 $f()$  が分かれば、 $y - f(x)$  で  $s$  が取り出せる！

以降のページで、

- $f()$  はどういう形をしているのか？
- $f()$  はどうやって推定するのか？

について説明していきます。

# エコーキャンセリング問題の 定式化

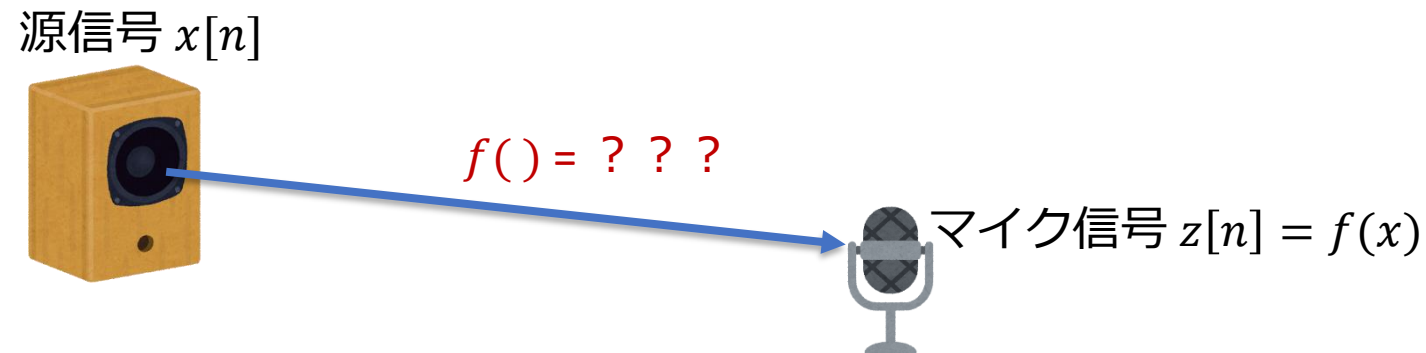
# 空中伝搬する音はどのように変化するのか

スピーカから再生させる源信号から、それが空中伝搬してマイクで収録された信号を推定するためには、空中伝搬の過程で信号がどう変化するのかわかる必要があります。

源信号を  $x[n]$ 、マイクに収録される信号を  $z[n]$  とします。 $n$  は時刻です。

空中伝搬の過程で起こる音の変化を  $f()$  とすると、 $z[n] = f(x)$  です。

では  $f()$  とはいったい何なのかわを見ていきましょう。





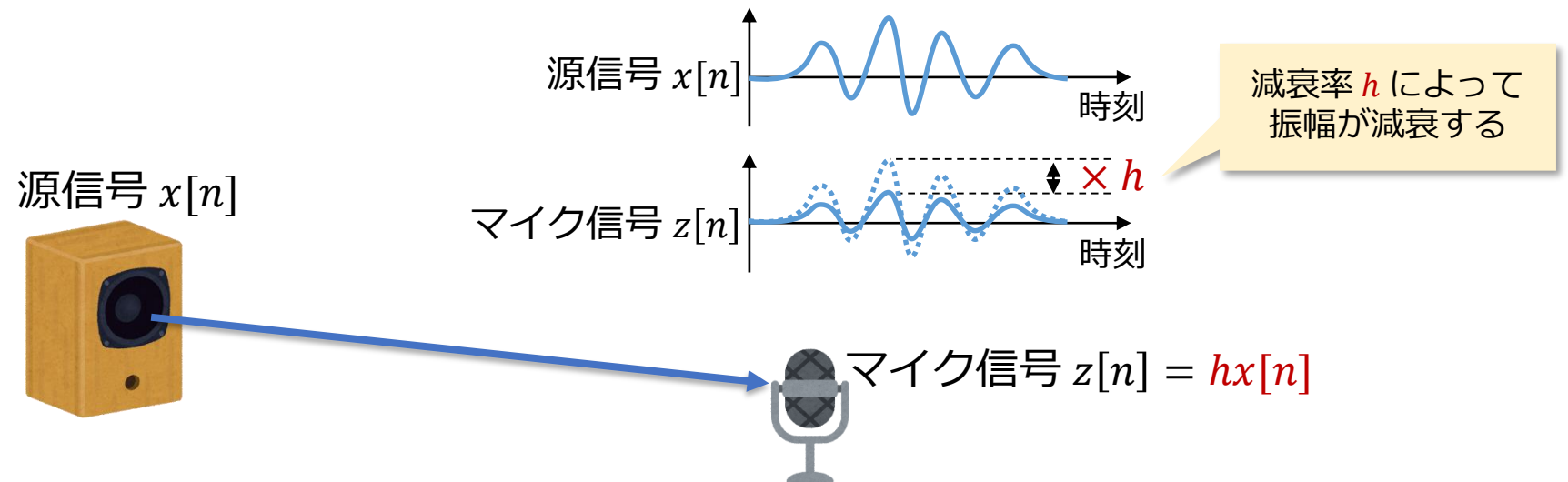
# 変化要素 1 : 音の減衰

変化する要素として、音の減衰があります。

スピーカから生じた音声は、空中伝搬する過程で音の振幅が減衰します。

この減衰率を  $h$  とすると、

減衰率を反映させたマイク信号  $z[n]$  は、 $z[n] = hx[n]$  と表せます。



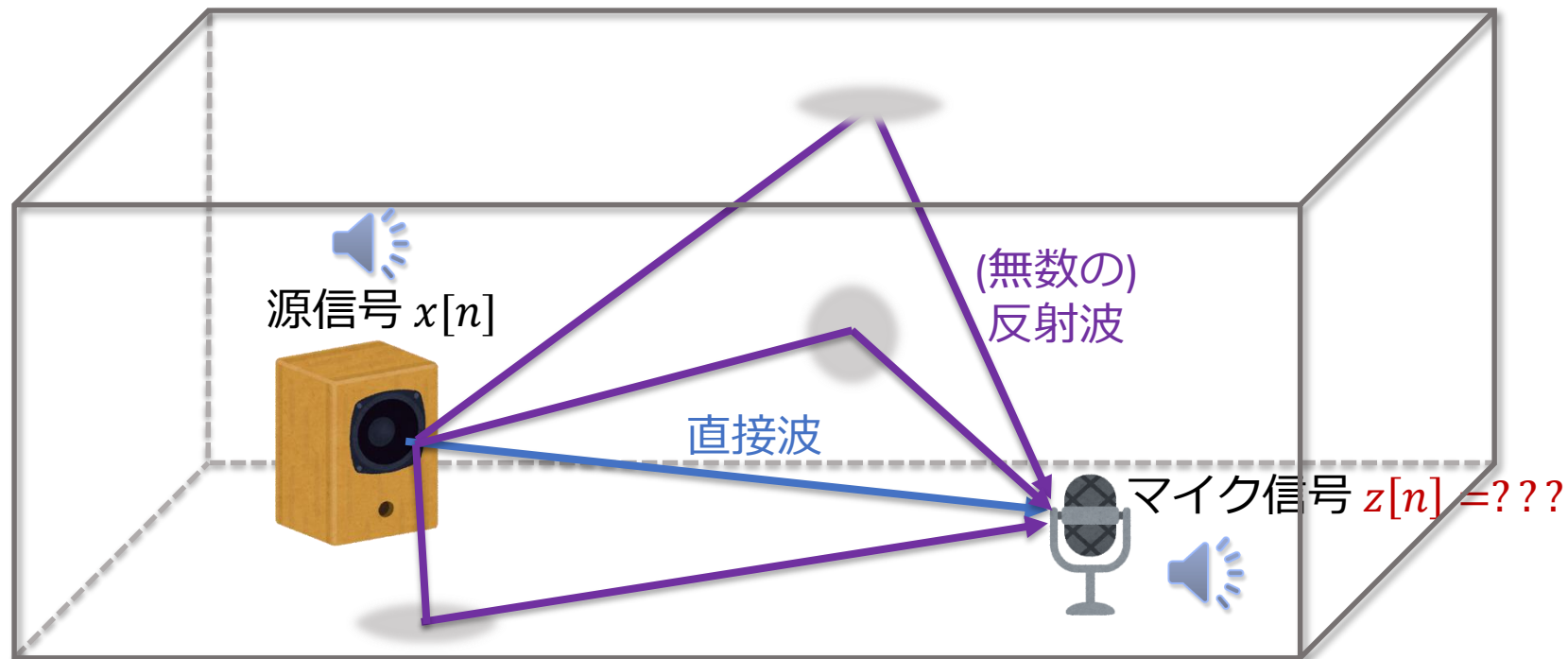
## 変化要素 2：残響

もう一つ、変化する最も重要な要素として、**残響**があります。

マイクに入力される信号は、スピーカから直接伝わる音（直接波）だけでなく、**壁や天井、床など様々な固体を跳ね返ってマイクに伝わる無数の音（反射波）**もあります。

反射波のことを**残響**と呼びます。残響の加算によって源信号の波形は大きく変わります。

このとき、マイク信号  $z[n]$  はどのようなになるのでしょうか。



# 残響の定式化

直接波は一番最初にマイクに到達します。

減衰率（変化要素1）を  $h_0$  とすると、

マイクで観測される直接波は、  $z[n] = h_0 x[n]$  です。

源信号  $x[n]$



直接波（減衰率 =  $h_0$ ）



マイク信号  $z[n] = h_0 x[n]$



# 残響の定式化

壁や天井、床などを介して伝わる反射波は、直接波より遅れてマイクに伝わります。

どれだけ遅れるかは、反射経路によって異なります。また反射波の減衰率も反射経路によって異なります。

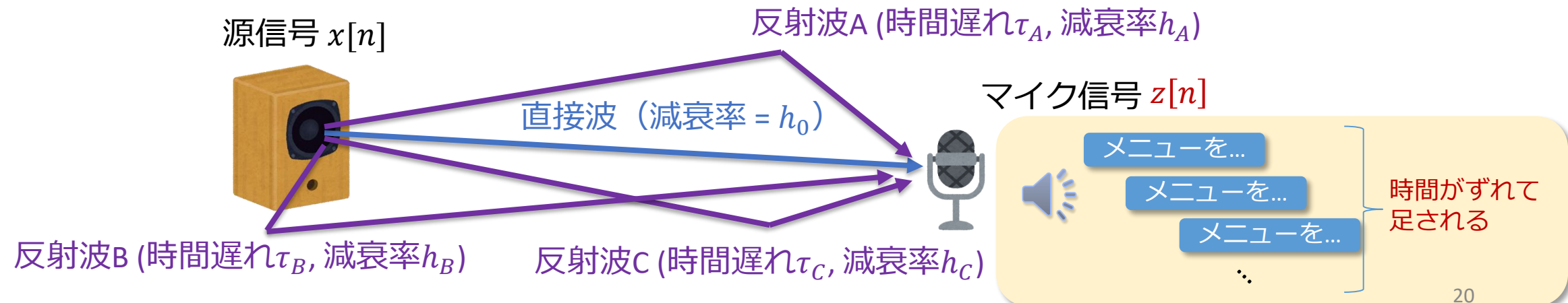
仮に反射波が A, B, C の3つだけだとします。

反射波 A, B, C が直接波よりそれぞれ  $\tau_A, \tau_B, \tau_C$  遅れて、さらに減衰率  $h_A, h_B, h_C$  で減衰してマイクに伝わるとしたとき、マイク信号  $z[n]$  は以下ようになります。

$$z[n] = \underbrace{h_0 x[n]}_{\text{直接波}} + \underbrace{\dots}_{\text{反射波 (残響)}}$$

直接波

反射波 (残響)



# 音の減衰と残響の定式化

実際の反射波は、直接波との時間遅れが 1 の反射波、2 の反射波、・・・と、無数に存在します。きりが無いので、時間遅れが  $1, \dots, K-1$  までの反射波のみを扱うことにします。

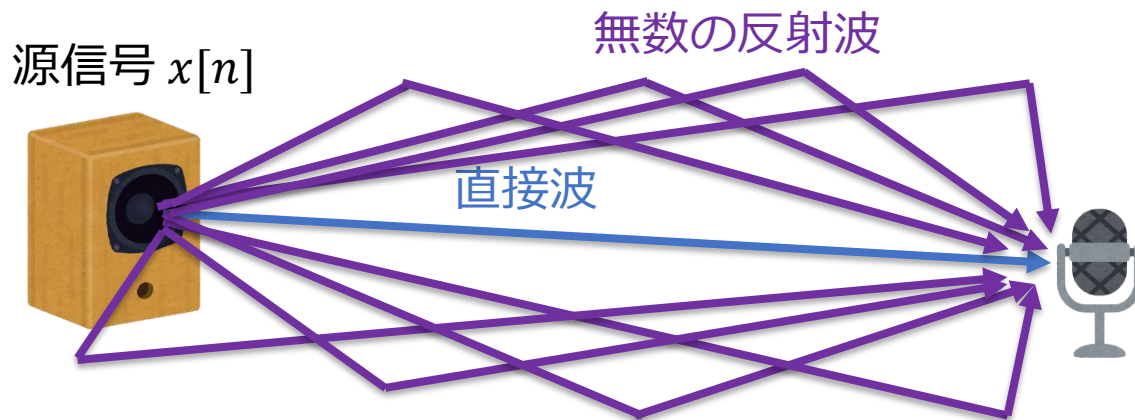
また、直接波の減衰率を  $h[0]$ 、時間遅れ  $1, \dots, K-1$  の反射波の減衰率を  $h[1], \dots, h[K-1]$  とします。このとき、マイク信号  $z[n]$  は以下のようになります。

直接波の減衰率  
(変化要素 1)

残響(変化要素 2)

$$\begin{aligned} z[n] &= h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + \dots + h[K-1]x[n-(K-1)] \\ &= \sum_{k=0}^{K-1} h[k]x[n-k] \end{aligned}$$

この式を、 $h$  と  $x$  の**畳み込み**と呼びます。



マイク信号  $z[n] = \sum_{k=0}^{K-1} h[k]x[n-k]$

# 空中伝搬する音の変化：結論

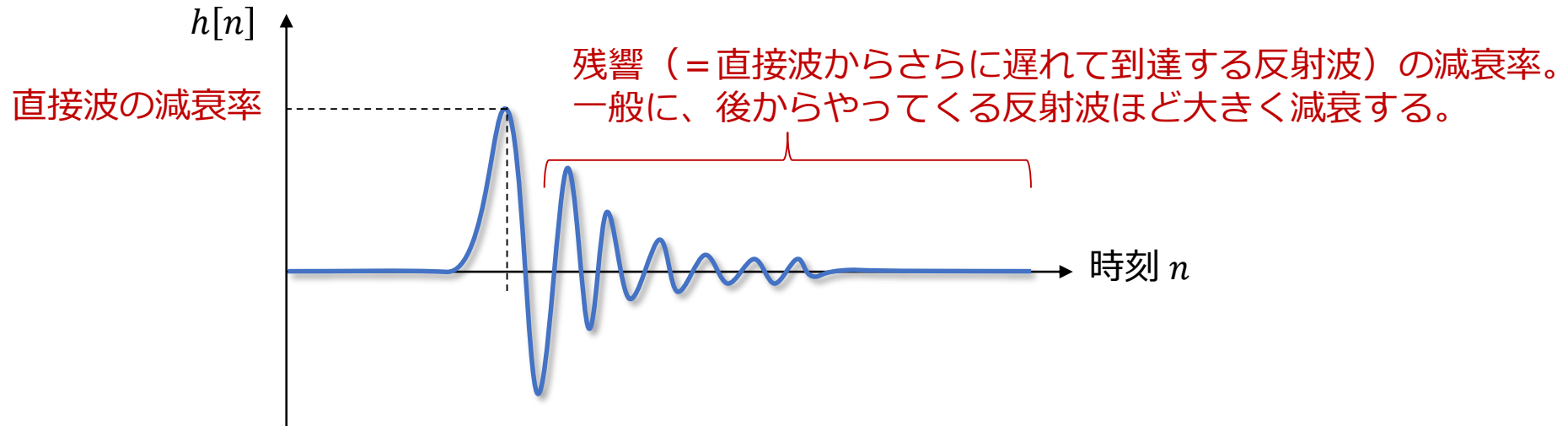
源信号  $x[n]$  を再生し、それが空中伝搬してマイクで収録されたときの信号  $z[n]$  は、畳み込みの式によって表すことができる！

$$z[n] = \sum_{k=0}^{K-1} h[k]x[n-k]$$

このとき、 $h[n]$  ( $n = 0, \dots, K-1$ ) を、**インパルス応答**と呼びます。

インパルス応答は、下のような形をしています。

最初のピーク値は直接波の減衰率（変化要素1）、以降の値は残響（変化要素2）を表します。

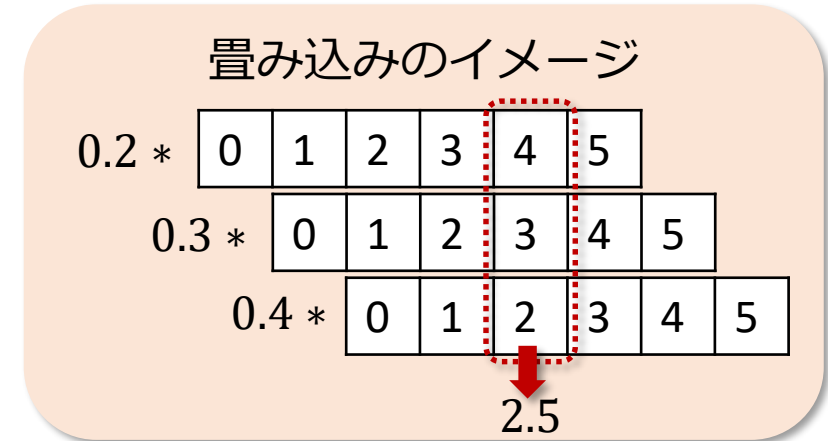


# 畳み込みの計算例


例えば、畳み込みの式  $z[n] = \sum_{k=0}^{K-1} h[k]x[n-k]$

について、 $x = [0, 1, 2, 3, 4, 5]$  ( $n = 0, \dots, 5$ )、 $h = [0.2, 0.3, 0.4]$  ( $K = 3$ ) としたとき、 $z[4]$  を計算してみましょう。

$$\begin{aligned} z[4] &= \sum_{k=0}^2 h[k]x[4-k] = h[0]x[4] + h[1]x[3] + h[2]x[2] \\ &= 0.2 * 4 + 0.3 * 3 + 0.4 * 2 = 2.5 \end{aligned}$$



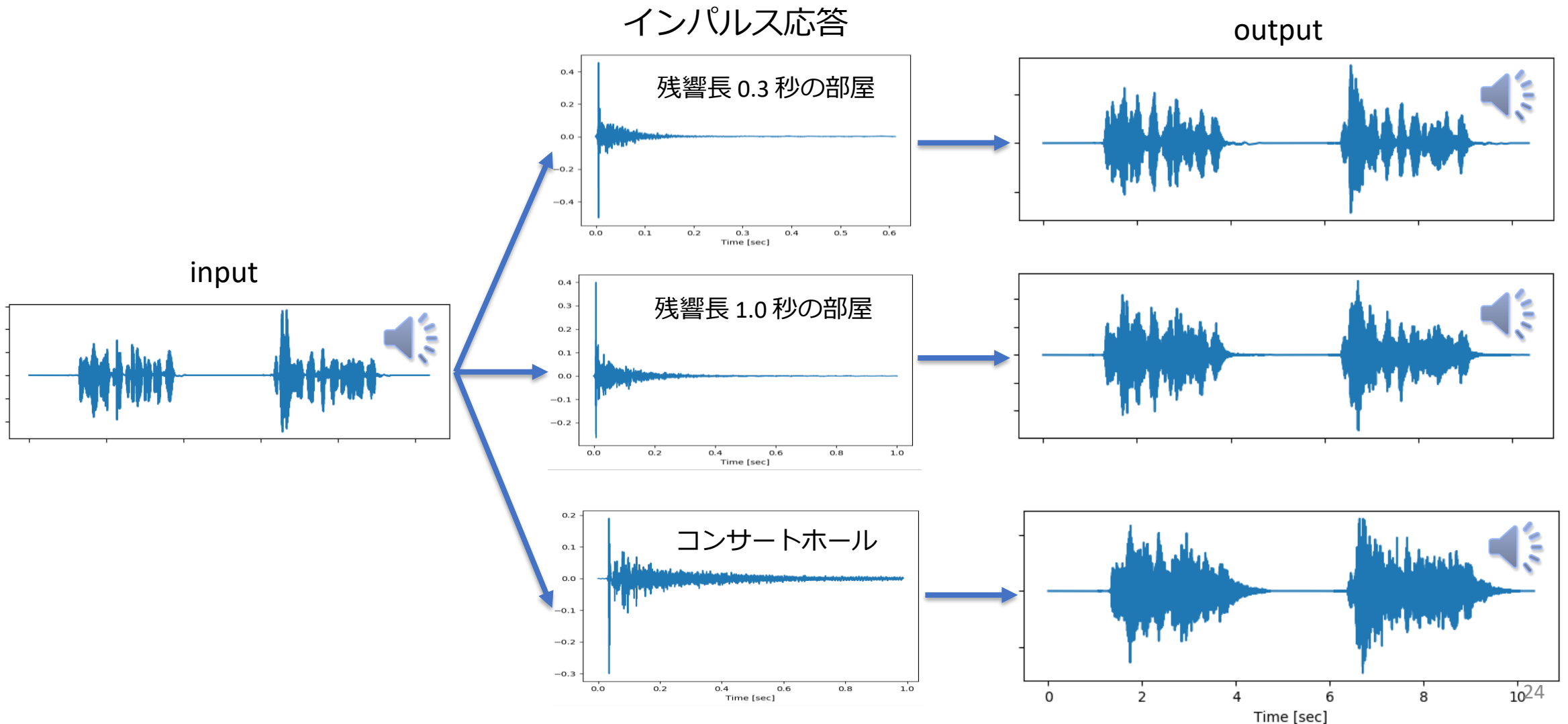
これは、 $x$  の  $n = 4$  から 3 つ前の要素(=2)～ 4 番目までの要素  $[2, 3, 4]$  を逆順に並び替えて、 $h$  と内積を計算することに等しいです。

$[2, 3, 4]$    $[4, 3, 2]$   
逆順に並び替え

$$\underbrace{[0.2, 0.3, 0.4] \cdot [4, 3, 2]^T}_{\text{内積}} = 0.2 * 4 + 0.3 * 3 + 0.4 * 2 = 2.5$$

実際にインパルス応答を畳み込んでみよう

07\_02\_convolve\_rir.ipynb を動かして、音声にインパルス応答を畳み込んでみましょう。



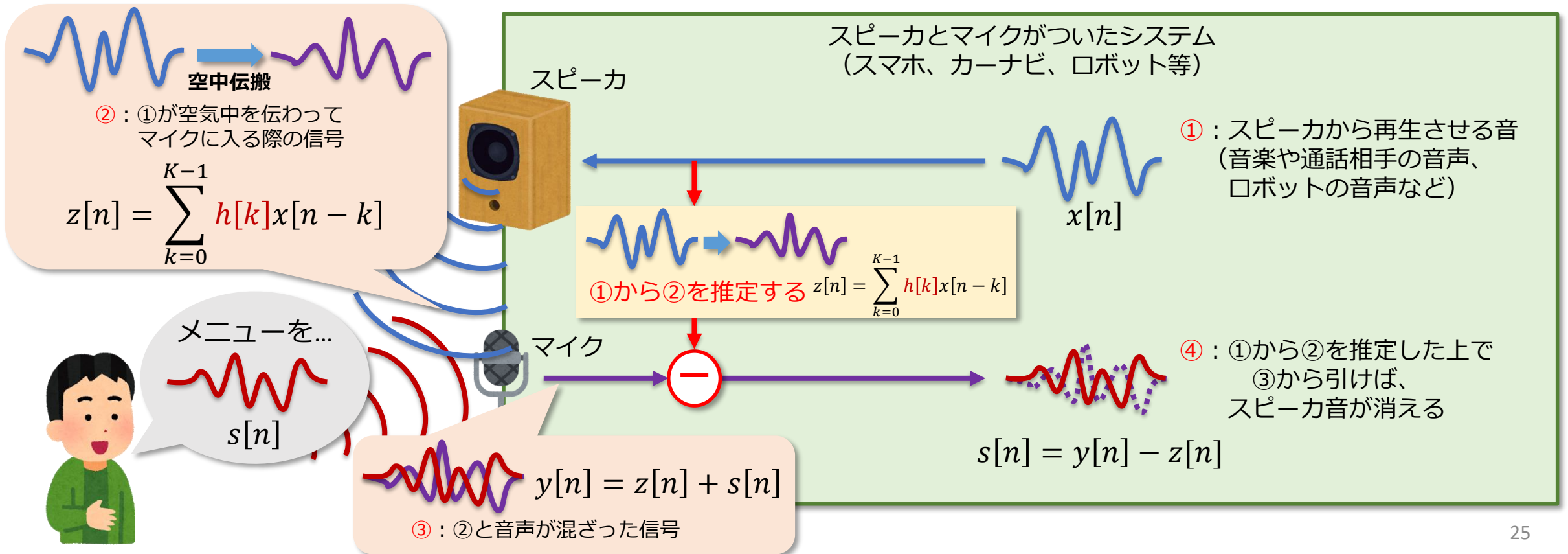


# エコーキャンセリング問題の定式化：結論

エコーキャンセリングを行うためには、スピーカから再生させる源信号(①)から、それが空中伝搬してマイクで収録された信号(②)を推定する必要がありました。

①から②への変換は、インパルス応答  $h$  を畳み込むことでできることが分かりました。

よって、次は「どうやってインパルス応答  $h$  を推定するのか」を考える必要があります。



# エコーキャンセリングの アルゴリズム

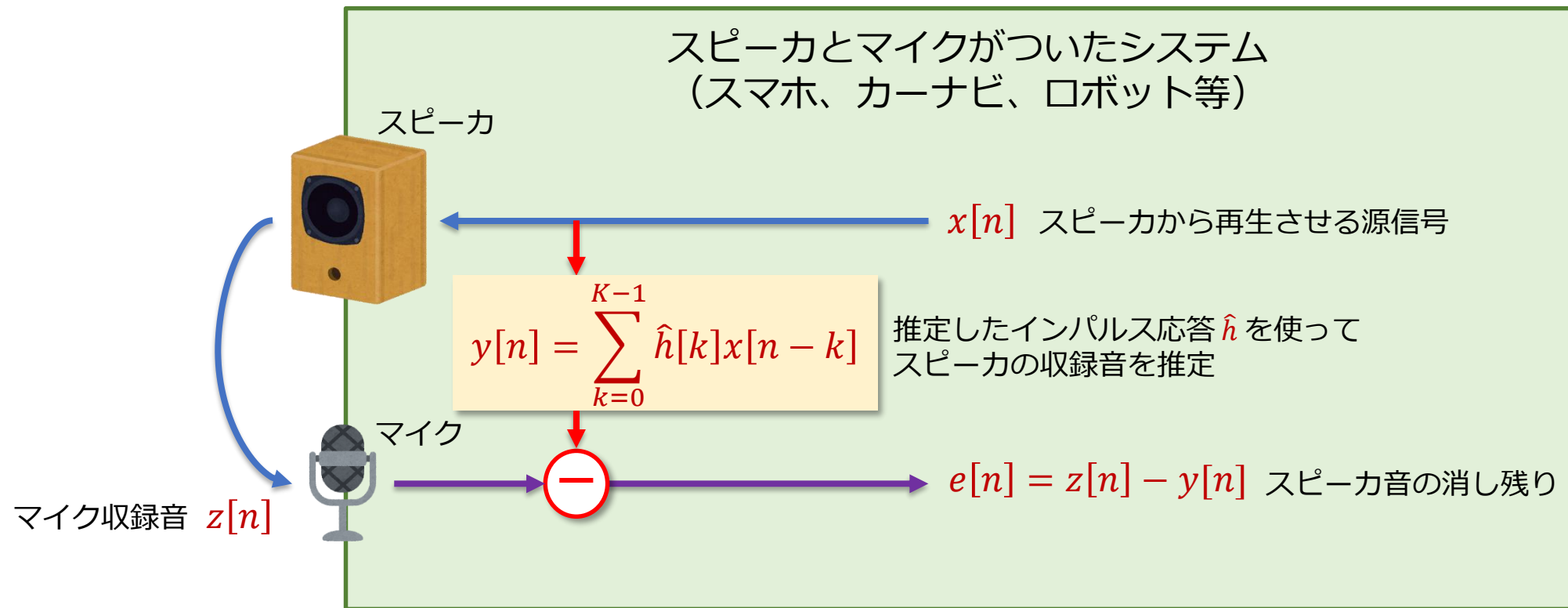
(LMSアルゴリズムとNLMSアルゴリズム)

# 問題の定式化

仮に人が話しておらず、スピーカの音声だけがマイクに収録されているとします。

何らかの方法で推定したインパルス応答  $\hat{h}$  を使ってスピーカの収録音を推定し（推定値 =  $y[n]$ ）、実際のマイク収録音  $z[n]$  から引きます。

引いたときに残った音  $e[n]$  がスピーカ音の消え残りとなります。



# インパルス応答を推定するための方針

人が話していない時の、エコーキャンセリングの式は以下の通りです。

$$y[n] = \sum_{k=0}^{K-1} \hat{h}[k]x[n-k]$$

$$e[n] = z[n] - y[n]$$

$x[n]$  : スピーカから再生させる源信号

$\hat{h}[n]$  : 推定したインパルス応答

$y[n]$  : 推定したスピーカの収録信号

$z[n]$  : マイク収録音

$e[n]$  : スピーカ音の消し残り

もしインパルス応答の推定が正確であれば、消し残り  $e[n]$  の音量は0に近くなるはずです。  
そこで、 $e[n]$  のエネルギー（二乗した値）を最小にするように  $\hat{h}$  を推定します。

$$\begin{aligned} L &= (e[n])^2 \\ &= (z[n] - y[n])^2 \end{aligned}$$

$$= \left( z[n] - \sum_{k=0}^{K-1} h[k]x[n-k] \right)^2$$



最小にする(0に近づける)  $\hat{h}$  を推定する。  
= 二乗誤差最小化基準

損失関数  $L$  を最小化するパラメータ  $h$  は**勾配降下法**で求めます。

# 勾配降下法を使ったインパルス応答の推定

勾配降下法をエコーキャンセリングの問題に適用してみます。  
 $h[k]$  を推定するため、 $h[k]$  に対する  $L$  の勾配を計算します。

$$\begin{aligned}\frac{\partial L}{\partial h[k]} &= \frac{\partial}{\partial h[k]} \left( z[n] - \sum_{k=0}^{K-1} h[k]x[n-k] \right)^2 \\ &= \frac{\partial}{\partial h[k]} \left( -2z[n]h[k]x[n-k] + \left( \sum_{k=0}^{K-1} h[k]x[n-k] \right)^2 \right) \quad (\text{展開して} h[k] \text{に関する項以外を削除}) \\ &= -2z[n]x[n-k] + 2x[n-k] \sum_{k=0}^{K-1} h[k]x[n-k] \quad (h[k] \text{に関する偏微分を計算}) \\ &= -2x[n-k] \left( z[n] - \sum_{k=0}^{K-1} h[k]x[n-k] \right) \quad (x[n-k] \text{の項でくくる}) \\ &= -2e[n]x[n-k] \quad (e[n] = z[n] - \sum_{k=0}^{K-1} \hat{h}[k]x[n-k] \text{ を利用})\end{aligned}$$

シンプルな勾配の式が得られる。

# 勾配降下法を使ったインパルス応答の推定

勾配降下法によるパラメータ更新は以下の式で定義されます。

$$\hat{h}_{new}[k] \leftarrow \hat{h}_{old}[k] - \mu \frac{\partial L}{\partial h[k]}$$

前頁で導出された勾配  $\frac{\partial L}{\partial h[k]} = -2e[n]x[n-k]$  を使って、以下の更新式が得られます。

(係数の2は省略します)

$$\hat{h}_{new}[k] \leftarrow \hat{h}_{old}[k] + \mu e[n]x[n-k]$$

この更新式によってインパルス応答を推定する手法を、**LMS (Least Mean Square) アルゴリズム**と呼びます。

# LMSの改良手法：NLMSアルゴリズム

LMSアルゴリズムによる更新式

$$\hat{h}_{new}[k] \leftarrow \hat{h}_{old}[k] + \mu e[n]x[n-k]$$

に対して、更新項を源信号  $x$  のパワーで正規化すると、より安定してインパルス応答を推定できることが知られています。

$x$  のパワー(二乗和)は、 $\|x[n]\| = \sum_{k=0}^{K-1} (x[n-k])^2$  によって求められます。

大きさによる正規化を使ったLMSアルゴリズムの更新式は、以下のようになります。

$$\hat{h}_{new}[k] \leftarrow \hat{h}_{old}[k] + \frac{\mu}{\|x[n]\| + \epsilon} e[n]x[n-k]$$

$\epsilon$  はゼロ除算を防ぐための小さな値 (例:  $\epsilon = 1E-10$ )

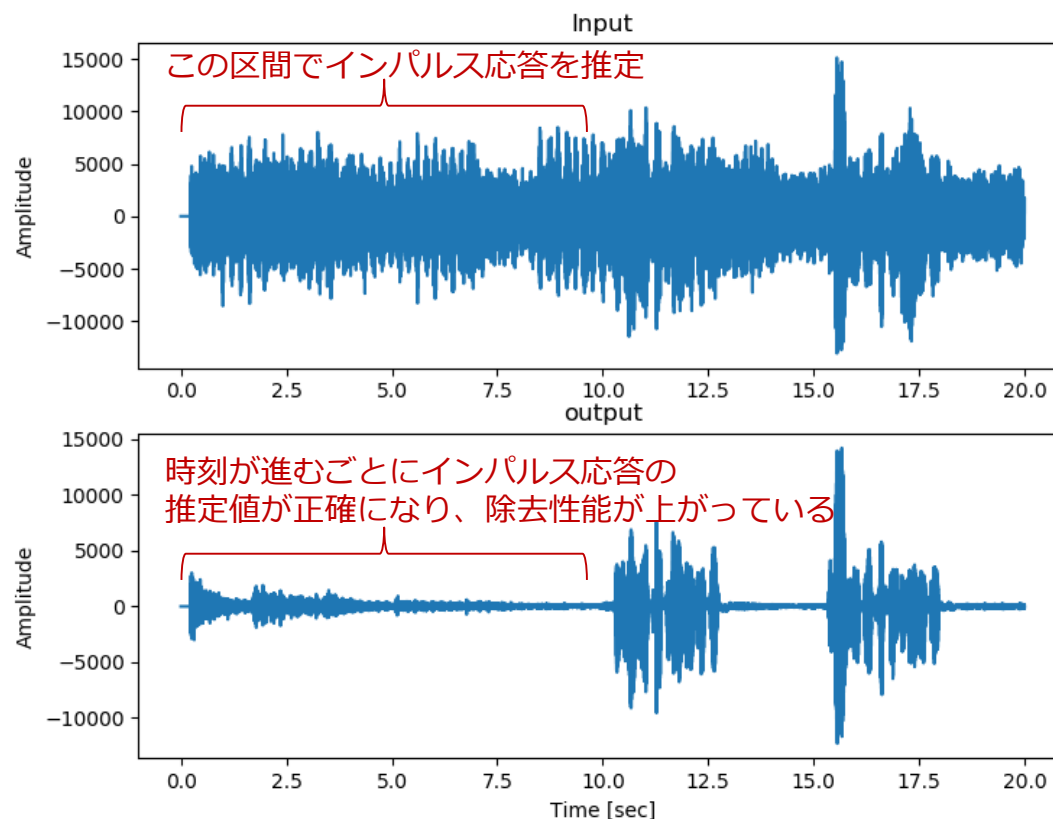
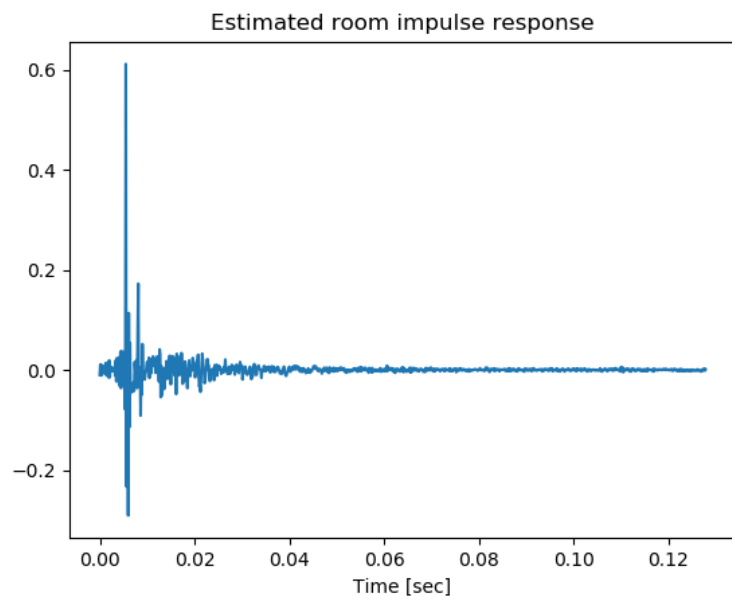
このアルゴリズムを、**NLMS (Normalized LMS) アルゴリズム**と呼びます。

ステップサイズ  $\mu$  は  $0.0 < \mu \leq 1.0$  の値をとります。小さいほど推定精度は高くなりますが、推定にかかる時間が大きくなります。

# NLMSアルゴリズムを実行してみよう

07\_03\_echo\_cancelling.ipynb を動かして, NLMSエコーキャンセリングを実行してみましよう。

実装したNLMSアルゴリズムでは、人の音声が始まる直前の10.0秒間でインパルス応答を推定（学習）し、それ以降は推定値の修正を止めています。



echo\_mixed\_input.wav



output.wav





# おわりに

今回は、最小二乗誤差基準＋勾配降下法の応用例として、エコーキャンセリングによる雑音除去を紹介しました。

データ処理において重要なことは「**モデリング**」と「**最適化**」です。

エコーキャンセリング問題における

モデリング：インパルス応答との畳み込み（残響）

最適化：二乗誤差最小化基準と勾配降下法

最適化が重要なのは当然ですが、どれだけ精度よく最適化しても、モデリングが不正確だと問題は解けないということです。

次回は再びクラスタリングの話に戻ります。

# レポート課題

echo\_mixed\_input\_2.wav に対して、NLMSアルゴリズムによるエコーキャンセリングを実行せよ。

サンプルプログラム07\_03\_echo\_cancelling.ipynbで使ったデータは、インパルス応答が時間によって変化しないことを前提としたシミュレーションデータである。

一方echo\_mixed\_input\_2.wavは、インパルス応答が時間によって微小に変動する、より実環境に近いシミュレーションデータであり、サンプルプログラムほどきれいに雑音除去がされない。

きれいに除去されない点について、何故うまくいかないのか、どうすれば（どういふことができれば）うまくいくのか考察せよ。

レポート提出期限：6/21(火) AM10:30, ipynbファイルをhtmlファイルに変換して提出