

ASP Final Project

黃柏維 機械碩一 R10522815

Methods :

1.

在課堂中有學到以下幾種 beamformers

➤ The beamformer with uniform weights

選擇權重

$$\mathbf{w} = \frac{1}{N} [1 \ 1 \ 1 \ 1 \ \dots \ 1]^T$$

使得輸出

$$y(t) = \mathbf{w}^H \mathbf{x}(t) = (Ae^{j2\pi ft}) + \left(\frac{1}{N} \mathbf{1}^H \mathbf{n}(t)\right)$$

如此一來可以把雜訊平均，盡量的消除雜訊。

➤ The beamformer with array steering

選擇權重

$$\mathbf{w} = \frac{1}{N} \mathbf{a}(\theta_s) = \frac{1}{N} [1 \ e^{j(\pi \sin \theta_s)} \ e^{2j(\pi \sin \theta_s)} \ e^{3j(\pi \sin \theta_s)} \ \dots \ e^{(N-1)j(\pi \sin \theta_s)}]$$

這可以使得最大的 $B_\theta(\theta)$ 會落在 θ_s 上，較能準確抓出 source 的訊號。

➤ The MVDR beamformer

由於輸出 $y(t)$ 可以寫成

$$\mathbf{w}^H \mathbf{x}(t) = \mathbf{w}^H (\mathbf{a}(\theta_s) s_1(t) + \mathbf{n}(t)) = (\mathbf{w}^H \mathbf{a}(\theta_s)) s_1(t) + (\mathbf{w}^H \mathbf{n}(t))$$

當我們要降低 noise 造成的影響，我們必須盡量降低 $E[\mathbf{n}(t)\mathbf{n}^H(t)]$ ，但是 noise 的 R 矩陣不好計算，因此選擇改用降低 $E[|y|^2]$ ，並建立最佳化問題

$$\mathbf{w}_{\text{MVDR}} = \arg \min E[|y|^2]$$

subject to

$$\mathbf{w}^H \mathbf{a}(\theta_s) = 1,$$

$$y(t) = \mathbf{w}^H \mathbf{x}(t)$$

將此最佳化問題解得如下

$$\mathbf{w}_{\text{MVDR}} = \frac{R^{-1} \mathbf{a}(\theta_s)}{\mathbf{a}^H(\theta_s) R^{-1} \mathbf{a}(\theta_s)}$$

因此只要知道 θ_s 及 R^{-1} ，即可計算 beamformer 的 weight。

➤ The LCMV beamformer

在訊號的接收中除了雜訊之外可能還會有來自 θ_i 的干擾訊號，但是 MVDR 比較沒辦法考慮到這個狀況，因此在做 MVDR 的最佳化問題時，將

Constraint 改成

$$\mathbf{C}^H \mathbf{w} = \mathbf{g}$$

此時

$$\begin{cases} \mathbf{C} = [\mathbf{a}(\theta_s) & \mathbf{a}(\theta_i)] \\ \mathbf{g} = \begin{bmatrix} 1 \\ g_i^* \end{bmatrix} \end{cases}, \text{ where } g_i < 1$$

將該最佳化問題解出可得

$$\mathbf{w}_{\text{LCMV}} = \mathbf{R}^{-1} \mathbf{C} (\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C})^{-1} \mathbf{g}$$

2.

在角度的估測上，課堂上有提供了幾種方法

- MVDR spectrum
- Multiple Signal Classification (MUSIC)
- Estimation of Signal Parameters via Rational Invariance Techniques (ESPRIT)

R 矩陣估計

不管在哪種方法中，必須先知道訊號的 R 矩陣，但因為其中涉及期望值的計算，然而我們只知道訊號源的量測值，因此必須透過這些量測值去估計 R 矩陣。有以下方法可以用來估計 R 矩陣

$$\begin{aligned} \hat{R} &= \frac{1}{K} \sum \hat{\mathbf{x}}(k) \hat{\mathbf{x}}^H(k) \\ \hat{R} &= \frac{1}{K} \sum \hat{\mathbf{x}}(k) \hat{\mathbf{x}}^H(k) + \delta \mathbf{I} \\ \hat{R} &= \sum_{k=1}^K \lambda^{K-k} \hat{\mathbf{x}}(k) \hat{\mathbf{x}}^H(k) + \delta \lambda^K \mathbf{I} \end{aligned}$$

我後來選擇採用第三種方法來計算估計 R 矩陣，因為第一種較容易計算出 singular 的矩陣，此時無法計算反矩陣，因此可以再加上一個微小的對角矩陣，使得矩陣變成 singular，易於計算反矩陣，第二及第三種方法都是這種類型的，但是第三種方法會另外乘上係數 λ ，使得最近的訊號會有較大的權重，因為 source 的訊號源位置會隨時間變化，因此太久之前的訊號相對就沒那麼重要，所以最後決定使用第三種方法估計 R 矩陣，並選擇 $\lambda = 0.99$ 做為權重的衰減率。

DOA

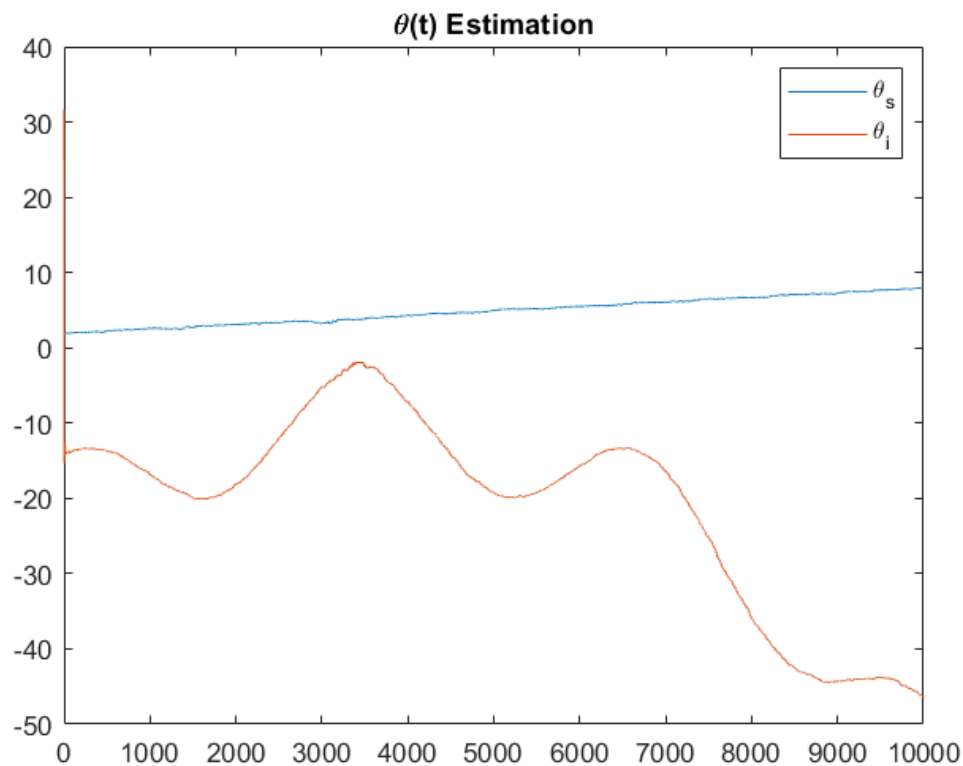
有了 R 矩陣，我們便可計算該時間點下的 MVDR spectrum，公式如下

$$\hat{P}_{\text{MVDR}}(\theta) = \frac{1}{\mathbf{a}^H(\theta) \hat{\mathbf{R}}^{-1} \mathbf{a}(\theta)}$$

接著即可找出此 MVDR spectrum 的 local maximum，找出第一大及第二大的峰值及對應的 θ 值即可估測出 θ_s 及 θ_i 。

在實作的過程中，由於在前幾個時間點所知道的資料量還太少，因此很容易被雜訊影響，因此我在計算 MVDR spectrum 的時候只會取到小數點第一位，粗略估計 θ_s 及 θ_i ，到約 5 個 sample 之後，我會直接選取前一個時刻估計出的角度，並取其正負一度的範圍計算 MVDR spectrum，計算該範圍內的 local maximum，並取到小數點後兩位，如此一來計算時間及精度可以取得平衡。

3.



4.

在課堂中有介紹了 4 種方法，其中 LCMV beamformer 最能處理這個 project 中的情況，因為只有 LCMV 有考慮 θ_s 及 θ_i 造成的影響，而其餘演算法僅僅只有考慮 θ_s 。因此我在設計 beamformer 時，優先參考 LCMV 的演算法，並依此下去做延伸。

在 LCMV 演算法中，beamformer 的 weighting 為

$$\mathbf{w}_{\text{LCMV}} = \mathbf{R}^{-1} \mathbf{C} (\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C})^{-1} \mathbf{g}$$

$$\text{where } \begin{cases} \mathbf{C} = [\mathbf{a}(\theta_s) & \mathbf{a}(\theta_i)] \\ \mathbf{g} = \begin{bmatrix} 1 \\ 0.001 \end{bmatrix} \end{cases}$$

即可計算出

$$\hat{\mathbf{y}}(t) = \mathbf{w}^H \mathbf{x}(t)$$

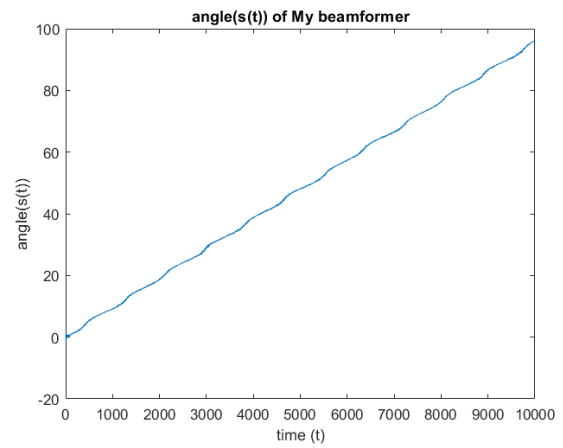
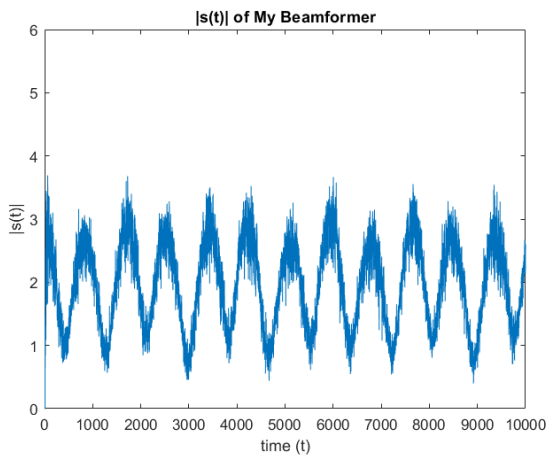
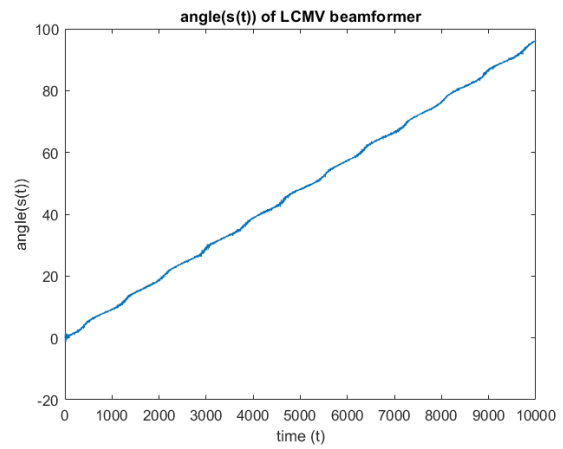
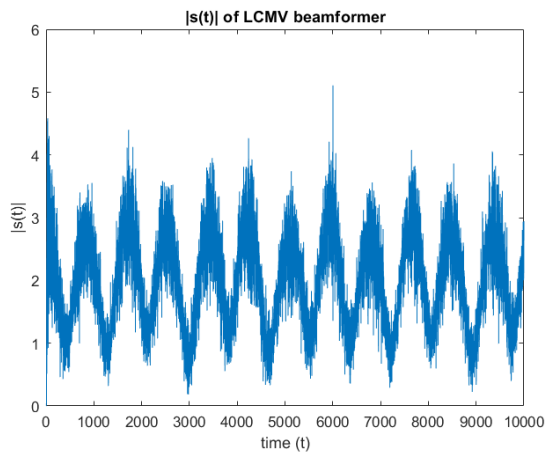
在這個基礎之下，再對輸出的 $\hat{\mathbf{y}}(t)$ 做平滑化處理，公式如下

$$\hat{s}(t) = \frac{\hat{s}(t-1) + \hat{\mathbf{y}}(t)}{2}, \text{ where } \hat{s}(0) = 0$$

這樣可以使在該時刻的輸出值可以參考前一個時間點的輸出值，使得每個時間點之間的輸出值不會有過大的差距，較能呈現連續性的波型。

5.

由於輸出是複數向量，因此我先取絕對值再畫出圖形，並把 unwrapped phase 畫出來



Additional Discussions :

DOA estimators

在我的 DOA estimator 裡面，我會先將 MVDR spectrum 計算出來，並取到小數點後第 2 位，因此會需要跑 for 迴圈來計算每個角度之下的值，此外也需要計算 R 矩陣的 inverse，所以會需要較大的計算量，複雜度為 n 。

LCMV beamformers

在 LCMV beamformer 中，就僅僅只是利用課本的公式進行矩陣運算，主要計算量是在於反矩陣的計算。

My beamformer

我所設計的 beamformer 是基於 LCMV beamformer 所延伸出，將 output 去做 smoothing，由於我做 smoothing 的算法是直接看上個平滑化後的資料並將其與 LCMV 新預測的資料取平均，因此只有 Constant time 的計算時間。因此計算複雜度與 LCMV beamformer 差不多。