

## About ADSR

用於合成音色的波形封包產生器(envelope generator)，應用於合成音色的樂器上使用，像是 keyboard 能夠藉由操作按鈕(控制 A,D,S,R)，達到更換音色的目的。

主要架構:

**Attack :**

起始音，是樂器剛被觸發，能量最大，所以斜率最大。  
影響振幅大小。

**Decay :**

在起始音的結束後的短暫且快速的衰退量。  
由 maximum 降至 sustain

**Sustain :**

在衰退一陣後，維持一定的音(穩定狀態)。

**Release :**

在演奏者手指放開鍵盤後，尾端自然震動的尾音。

我們便可以把一段聲波(自訂)，由 ADSR 濾波器套用在我們所產生的 sin or cos 波上面，讓他可以產生不同音色的變化。

而藉由調整 A、D、S、R，我們可以得到想要的波形變化，進而得到所希望得到的音色。

不同的音色，可由不同的 ADSR 所得到。

所以我參考了一下題目的定義，寫了以下幾個不同的 function:

**ADSR\_generator.m =>**

用來產生需要的長度(和 source wave 相同)的 ADSR

**Create\_cos.m =>**

依據給定的參數製造 cosine wave

**Sum\_2\_Wave.m =>**

相加兩個波(用來相加計算出的 cosine 跟 ADSR filter 做相加)

**Lab6\_script.m =>**

執行主程式

這邊主程式的部分，因為我不太確定 ADSR 的用法，所以我寫了兩個版本的

**ADSR:**

第一張圖 ADSR 的長度跟 cosine wave 長度相同，所以相加是以整段長度做相加。

第二張圖則是針對每個波做 ADSR。

而以結果來看，應該是第一種作法比較正確，第二種出現抖音的問題(每個 cos

都做 ADSR)，我認為是因為頻率更加明顯，造成抖音的問題。

然後，我再針對 ADSR 該用加法還是乘法來實作，在作一次實驗(第三張圖)，發現乘法的效果好像比加法來的像(更接近樂器發出來的聲音)。我認為原因在於第三種波形是凸顯(用乘法放大)原本的值，所以相對前面兩種都要來的明顯。

經過實際觀察，控制 ADSR 的比例，可以對於整個音的圓潤度(起音→暫留→消逝)更加接近目標樂器。而調整  $W0$  (在程式裡是  $F$ ，cosine 的頻率)，越高越刺耳；我覺得除了  $F$  之外，調整其他 `Create_cos` 部分的參數比較沒有影響，重點比較在於 ADSR 的比例，因為決定音色就在於整體波形，而 ADSR 正是針對波形作改變。