

# 訊號資料輸入/輸出

## 資料格式與訊號繪製

盧家鋒 助理教授

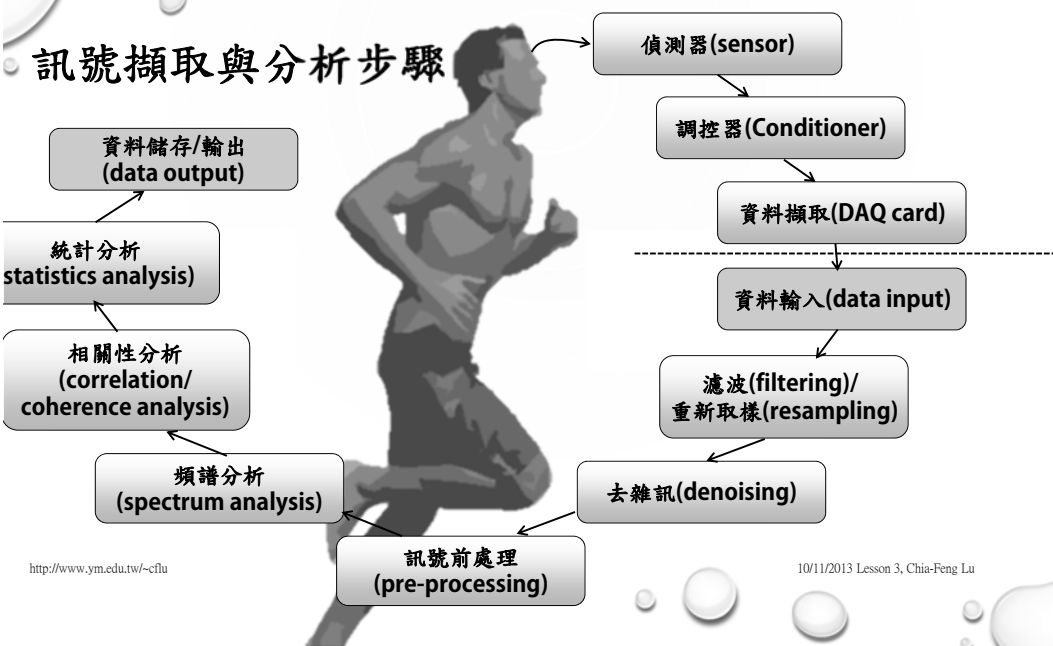
國立陽明大學 物理治療暨輔助科技學系

alvin4016@ym.edu.tw

## 請先下載本週上課資料

- <http://www.ym.edu.tw/~cflu>
- 點選左欄 [ 課程資料 ]
- 下載第4週上課資料 [ [demodata.zip](#) ]，檔案大小約107MB
- 下載完成後請解壓縮資料夾，並點擊 [pxBinaryViewerSetup.exe](#) 安裝 Binary Viewer 程式

## 訊號擷取與分析步驟



## 本週課程內容

- 認識資料格式
- MATLAB 資料輸入
- MATLAB 訊號繪製與確認
- MATLAB 檔案覆寫與儲存

## 認識資料格式

## 電腦記憶體

- 最小單位：位元 (**bit**)
- 二進位制 (**0 or 1, binary format**)
- 常用單位 **1 byte = 8 bits**

- **1 byte**可用不同進位制表示，列如：

|  |  |
|--|--|
| 二進位 <b>binary</b><br>(0,1)                                   | 00100110 <sub>(2)</sub>  |
| 十進位 <b>decimal</b><br>(0,1,2,3,4,5,6,7,8,9)                  | 00100110 <sub>(2)</sub> = $1*2^5 + 1*2^2 + 1*2^1$<br>= 037 <sub>(10)</sub>               |
| 十六進位 <b>hexadecimal</b><br>(0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F) | 00100110 <sub>(2)</sub> = 37 <sub>(10)</sub> = $2*16^1 + 5*16^0$<br>= 25 <sub>(16)</sub> |

## 電腦記憶體

- **1 byte = 8 bits** 可儲存 $2^8=256$ 個數字
- **2 bytes = 16 bits** 可儲存 $2^{16}=65536$ 個數字
- **4 bytes = 32 bits** 可儲存 $2^{32}=4.2950e9$ 個數字
- **8 bytes = 64 bits** 可儲存 $2^{64}=1.8447e19$ 個數字
- 字元/文字如何被電腦儲存？
  - **ASCII**字元對照表
- 字元與數字在電腦的眼中是不一樣的!!
  - 避免用錯誤的方式讀取資料

## ASCII字元編碼

- **America Standard Code for Information Interchange**，美國資訊交換標準碼
- 前**32**個字元 (**000<sub>(10)</sub>~031<sub>(10)</sub>**)，一般用於通訊或控制使用，大部分**Windows**系統無法顯示。
- 接下來**96**個字元 (**032<sub>(10)</sub>~127<sub>(10)</sub>**)，用於表示阿拉伯數字、英文字母大小寫和常用符號。

# ASCII字元對照表

十六進位的 0D 0A 連續出現代表換行!!

| ASCII 碼 |      |    |      |                 | ASCII 碼 |      |    |      |        |
|---------|------|----|------|-----------------|---------|------|----|------|--------|
| 十進位     | 十六進位 | 字元 | 控制字元 | 意義              | 十進位     | 十六進位 | 字元 | 控制字元 | 意義     |
| 000     | 00   |    | NULL | 空字元             | 016     | 10   | ▶  | DLE  |        |
| 001     | 01   | ⊖  | SOH  |                 | 017     | 11   | ◀  | DC1  |        |
| 002     | 02   | ⊕  | STX  |                 | 018     | 12   | ↑  | DC2  |        |
| 003     | 03   | ♥  | ETX  |                 | 019     | 13   | !! | DC3  |        |
| 004     | 04   | ♦  | EOT  |                 | 020     | 14   | ¶  | DC4  |        |
| 005     | 05   | ♣  | ENQ  |                 | 021     | 15   | §  | NAK  |        |
| 006     | 06   | ♠  | ACK  |                 | 022     | 16   | -  | SYN  |        |
| 007     | 07   | •  | BELL | 鈴聲              | 023     | 17   | ⏏  | ETB  |        |
| 008     | 08   | □  | BS   | 倒退鍵             | 024     | 18   | ↑  | CAN  |        |
| 009     | 09   |    | HT   | 定位鍵             | 025     | 19   | ↓  | EM   |        |
| 010     | 0A   |    | LF   | line feed       | 026     | 1A   | →  | SUB  | 檔案結束   |
| 011     | 0B   | ♂  | VT   | home            | 027     | 1B   | ←  | ESC  | escape |
| 012     | 0C   | ♀  | FF   | form feed       | 028     | 1C   | L  | FS   | 向右游標   |
| 013     | 0D   |    | CR   | carriage return | 029     | 1D   | ↔  | GS   | 向左游標   |
| 014     | 0E   | ♫  | SO   |                 | 030     | 1E   | ▲  | RS   | 向上游標   |
| 015     | 0F   | ☼  | SI   |                 | 031     | 1F   | ▼  | US   | 向下游標   |

# ASCII字元對照表

| ASCII 碼 |      | ASCII 碼 |     | ASCII 碼 |    | ASCII 碼 |      |   |
|---------|------|---------|-----|---------|----|---------|------|---|
| 十進位     | 十六進位 | 字元      | 十進位 | 十六進位    | 字元 | 十進位     | 十六進位 |   |
| 032     | 20   |         | 056 | 38      | 8  | 080     | 50   | P |
| 033     | 21   | !       | 057 | 39      | 9  | 081     | 51   | Q |
| 034     | 22   | "       | 058 | 3A      | :  | 082     | 52   | R |
| 035     | 23   | #       | 059 | 3B      | ;  | 083     | 53   | S |
| 036     | 24   | \$      | 060 | 3C      | <  | 084     | 54   | T |
| 037     | 25   | %       | 061 | 3D      | =  | 085     | 55   | U |
| 038     | 26   | &       | 062 | 3E      | >  | 086     | 56   | V |
| 039     | 27   | '       | 063 | 3F      | ?  | 087     | 57   | W |
| 040     | 28   | (       | 064 | 40      | @  | 088     | 58   | X |
| 041     | 29   | )       | 065 | 41      | A  | 089     | 59   | Y |
| 042     | 2A   | *       | 066 | 42      | B  | 090     | 5A   | Z |
| 043     | 2B   | +       | 067 | 43      | C  | 091     | 5B   | [ |
| 044     | 2C   | ,       | 068 | 44      | D  | 092     | 5C   | \ |
| 045     | 2D   | -       | 069 | 45      | E  | 093     | 5D   | ] |
| 046     | 2E   | .       | 070 | 46      | F  | 094     | 5E   | ^ |
| 047     | 2F   | /       | 071 | 47      | G  | 095     | 5F   | _ |
| 048     | 30   | 0       | 072 | 48      | H  | 096     | 60   | ` |
| 049     | 31   | 1       | 073 | 49      | I  | 097     | 61   | a |
| 050     | 32   | 2       | 074 | 4A      | J  | 098     | 62   | b |
| 051     | 33   | 3       | 075 | 4B      | K  | 099     | 63   | c |
| 052     | 34   | 4       | 076 | 4C      | L  | 100     | 64   | d |
| 053     | 35   | 5       | 077 | 4D      | M  | 101     | 65   | e |
| 054     | 36   | 6       | 078 | 4E      | N  | 102     | 66   | f |
| 055     | 37   | 7       | 079 | 4F      | O  | 103     | 67   | g |
|         |      |         |     |         |    | 104     | 68   | h |
|         |      |         |     |         |    | 105     | 69   | i |
|         |      |         |     |         |    | 106     | 6A   | j |
|         |      |         |     |         |    | 107     | 6B   | k |
|         |      |         |     |         |    | 108     | 6C   | l |
|         |      |         |     |         |    | 109     | 6D   | m |
|         |      |         |     |         |    | 110     | 6E   | n |
|         |      |         |     |         |    | 111     | 6F   | o |
|         |      |         |     |         |    | 112     | 70   | p |
|         |      |         |     |         |    | 113     | 71   | q |
|         |      |         |     |         |    | 114     | 72   | r |
|         |      |         |     |         |    | 115     | 73   | s |
|         |      |         |     |         |    | 116     | 74   | t |
|         |      |         |     |         |    | 117     | 75   | u |
|         |      |         |     |         |    | 118     | 76   | v |
|         |      |         |     |         |    | 119     | 77   | w |
|         |      |         |     |         |    | 120     | 78   | x |
|         |      |         |     |         |    | 121     | 79   | y |
|         |      |         |     |         |    | 122     | 7A   | z |
|         |      |         |     |         |    | 123     | 7B   | { |
|         |      |         |     |         |    | 124     | 7C   |   |
|         |      |         |     |         |    | 125     | 7D   | } |
|         |      |         |     |         |    | 126     | 7E   | ~ |
|         |      |         |     |         |    | 127     | 7F   | ☐ |

# ASCII字元對照表

| Bits | b <sub>7</sub> | b <sub>6</sub> | b <sub>5</sub> | b <sub>4</sub> | b <sub>3</sub> | b <sub>2</sub> | b <sub>1</sub> | b <sub>0</sub> |
|------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 0    | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| 1    | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 0              |
| 2    | 0              | 0              | 0              | 0              | 0              | 1              | 0              | 0              |
| 3    | 0              | 0              | 0              | 0              | 1              | 0              | 0              | 0              |
| 4    | 0              | 0              | 0              | 1              | 0              | 0              | 0              | 0              |
| 5    | 0              | 0              | 1              | 0              | 0              | 0              | 0              | 0              |
| 6    | 0              | 1              | 0              | 0              | 0              | 0              | 0              | 0              |
| 7    | 0              | 1              | 0              | 0              | 1              | 0              | 0              | 0              |
| 8    | 0              | 1              | 0              | 1              | 0              | 0              | 0              | 0              |
| 9    | 0              | 1              | 1              | 0              | 0              | 0              | 0              | 0              |
| 10   | 0              | 1              | 1              | 0              | 1              | 0              | 0              | 0              |
| 11   | 0              | 1              | 1              | 1              | 0              | 0              | 0              | 0              |
| 12   | 1              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| 13   | 1              | 0              | 0              | 0              | 1              | 0              | 0              | 0              |
| 14   | 1              | 0              | 1              | 0              | 0              | 0              | 0              | 0              |
| 15   | 1              | 0              | 1              | 0              | 1              | 0              | 0              | 0              |

# [MATLAB RULE] 建立矩陣

- 如要輸入字元/字串，須將輸入文字以前後單引號標註
  - '1000 is a string not a number'
  - '1000'
- 如要輸入數字，僅需直接鍵入數字
  - 1000

試試看：

```
abs('s')
char(115)
%如 input 數值介於 127~65535 則按照 unicode 編碼

abs('我很欣賞你')
char([25105 24456 27427 36062 20320])
```

在電腦語言中，字串與數字是不一樣的!!

## 常用字元編碼

- **ASCII**碼是使用最廣泛的標準碼，使用**7個bits**來記錄每個符號(標準為**128**個字元)
- **ANSI(American National Institute 美國國家標準協會)**，使用**8個bits**，前**128**個符號和**ASCII**一樣，新增**128**個符號。用於表示框線、音標與非英語系歐洲字母。
- **UNICODE**使用**2或4個bytes**來表示每一個符號，共可表示**65536**個或**1677**萬個字元符號，除英文外，還可以包含數量最多的中文、日文，及全世界各國的文字符號。
- **BIG5**是中文系統下的編碼方式，**BIG5**只能容納一萬三千多個漢字，相對於國字整理小組整理出的七萬多個字，不夠使用。

## 記憶體位置順序

- **Little-Endian (Intel, AMD 等其他常見CPU廠牌使用)**
  - 儲存記憶體位置時，小位數的**byte**會寫在前面
  - 所以在解讀編碼時，位置需要前後互換
  - 例如： $15294_{(10)} = 00111011\ 10111110_{(2)}$  但實際記憶體位置為  
**10111110 00111011**
- **Big-Endian (Sun Sparc使用)**
  - 儲存記憶體位置時，大位數的**byte**會寫在前面
  - 所以在解讀編碼時，位置不需要調整
  - 例如： $15294_{(10)} = 00111011\ 10111110_{(2)}$  但實際記憶體位置仍為  
**00111011 10111110**

## 常用資料精準度 (PRECISION)

| C or Fortran    | Description                | MATLAB   | C or Fortran     | Description                     |
|-----------------|----------------------------|----------|------------------|---------------------------------|
| 'unsigned char' | unsigned integer, 8 bits.  | 'char'   | 'char*1'         | character.                      |
| 'signed char'   | signed integer, 8 bits.    | 'short'  | 'short'          | integer, 16 bits.               |
| 'integer*1'     | integer, 8 bits.           | 'int'    | 'int'            | integer, 32 bits.               |
| 'integer*2'     | integer, 16 bits.          | 'long'   | 'long'           | integer, 32 or 64 bits.         |
| 'integer*4'     | integer, 32 bits.          | 'ushort' | 'unsigned short' | unsigned integer, 16 bits.      |
| 'integer*8'     | integer, 64 bits.          | 'uint'   | 'unsigned int'   | unsigned integer, 32 bits.      |
| 'integer*1'     | unsigned integer, 8 bits.  | 'ulong'  | 'unsigned long'  | unsigned integer, 32 bits or 64 |
| 6*'integer*2'   | unsigned integer, 16 bits. | 'float'  | 'float'          | floating point, 32 bits.        |
| 2*'integer*4'   | unsigned integer, 32 bits. |          |                  |                                 |
| 4*'integer*8'   | unsigned integer, 64 bits. |          |                  |                                 |
| e*'real*4'      | floating point, 32 bits.   |          |                  |                                 |
| 32*'real*4'     | floating point, 32 bits.   |          |                  |                                 |
| e*'real*8'      | floating point, 64 bits.   |          |                  |                                 |
| 64*'real*8'     | floating point, 64 bits.   |          |                  |                                 |

## MATLAB資料輸入實作

## 常見原始資料格式

- ASCII純文字格式
- Binary二進位資料
- Excel活頁簿
- 醫學影像與訊號通用協定 (DICOM, MFER...)
- 其他廠牌自定義與未知格式

ASCII純文字格式→

Binary Viewer !!

Binary二進位資料→



## 醫學訊號必要資訊

- 取樣頻率 (Sampling rates)
- 紀錄單位 (Physical units)
- 數值範圍 (Dynamic ranges)
- 資料長度 (Data length)
- 事件紀錄 (Events, Annotations, Markers)
- 其他資訊 (Scanning date, Subject ID/name, Demography, Manufacturer, Operator...)

## [MATLAB RULE] 指定檔案完整路徑

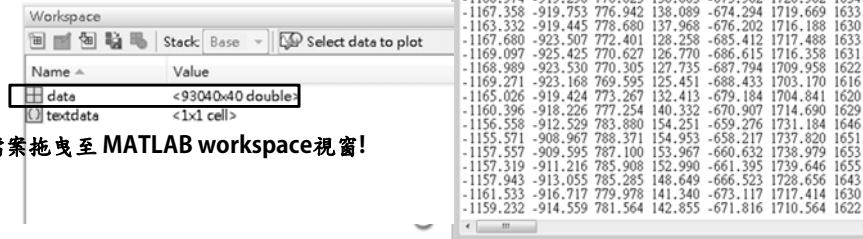
- 方式一：
  - 手動輸入完整路徑字串
  - `full_filepath='C:\Users\Alvin\Desktop\demodata\ASCII\MAX80_TEST.txt';`
- 方式二：
  - 呼叫檔案開啟視窗選取檔案路徑
  - `[filename filepath]=uigetfile('*.');`  
`full_filepath=[filepath filename];`

## [MATLAB RULE] 變數名稱

- 變數名稱是由英文字母、數字或底線所組成，不能使用特殊符號
- 開頭的第一個字元必須是英文字母，不能是數字
- 名稱長度不能超過32個字元
- 會區分變數的大小寫
- 變數不必宣告便可直接使用

## 訊號範例2：ASCII FORMAT

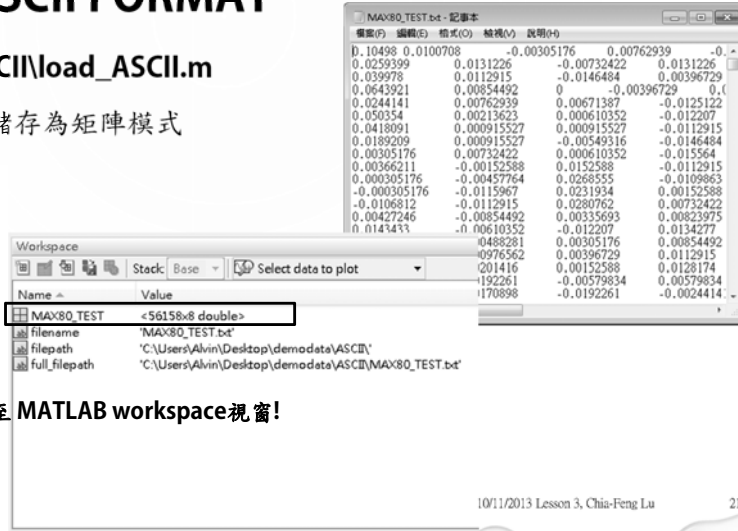
- 請開啟 demodata\ASCII\_mix\load\_ASCII.m
- 數值排列非矩陣模式 → 直接使用 load 會錯誤!
- fid=fopen(full\_filepath, 'r');  
fseek(fid, offset, 'bof');  
data=fscanf(fid, '%f', [channelno datapoint]);  
fclose(fid);



也可直接將ASCII檔案拖曳至 MATLAB workspace視窗!

## 訊號範例1：ASCII FORMAT

- 請開啟 demodata\ASCII\load\_ASCII.m
- 數值排列以分隔符號儲存為矩陣模式
- load(full\_filepath);



也可直接將ASCII檔案拖曳至 MATLAB workspace視窗!

## [MATLAB RULE] 向量/矩陣轉置 (行列互換)

- 單一單引號放在向量/矩陣變數後，可以達到行列互換的效果

>> A=[1 2 3;4 5 6]

A =

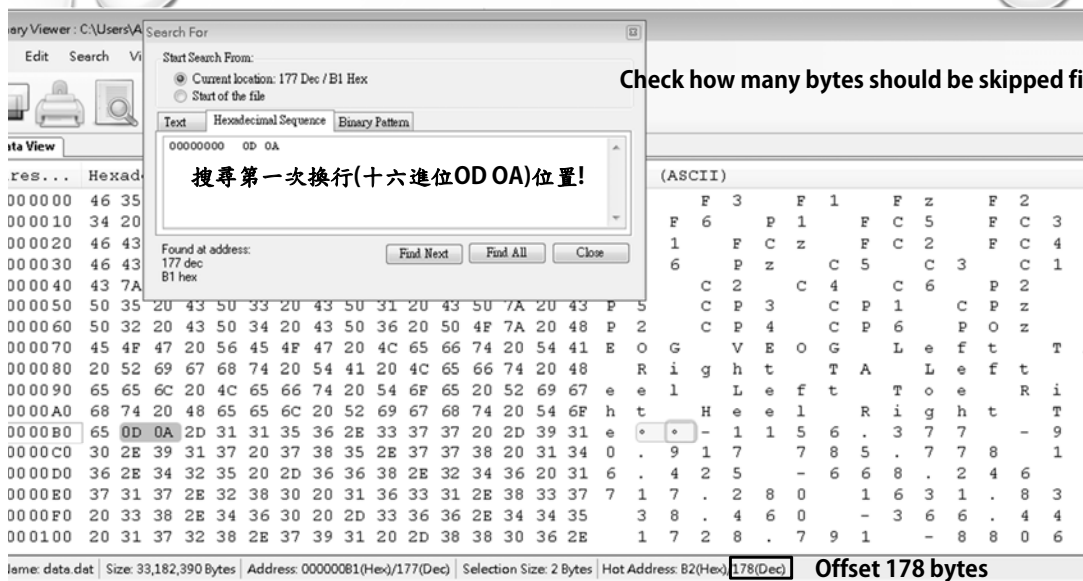
```
1 2 3
4 5 6
```

>> A=A'

A =

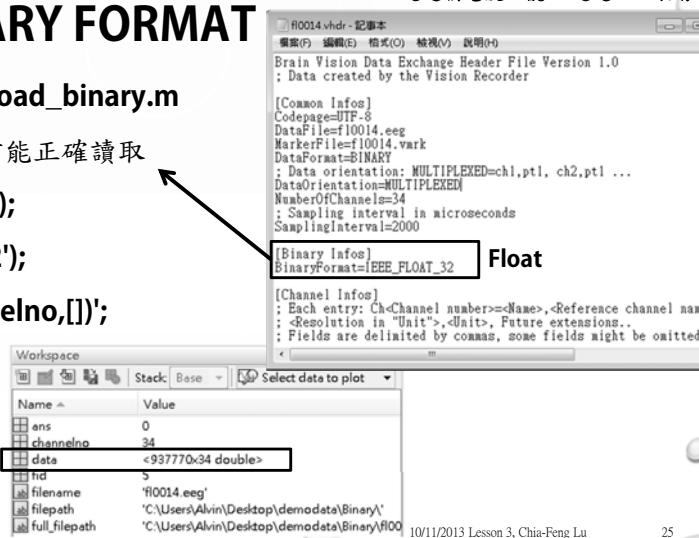
```
1 4
2 5
3 6
```

Check how many bytes should be skipped fir



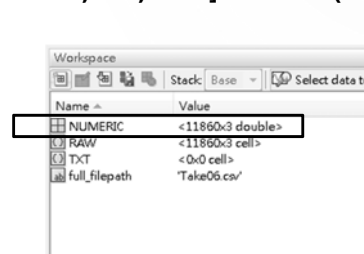
## 訊號範例3：BINARY FORMAT

- 請開啟 `demodata\Binary\load_binary.m`
- 必須先知道資料的精準度方能正確讀取
- `fid=fopen(full_filepath,'r');`
- `data=fread(fid,inf,'float32');`
- `data=reshape(data,channelno,[]);`
- `fclose(fid);`



## 訊號範例4：EXCEL格式

- 請開啟 `demodata\Excel\load_excel.m`
- 可以讀取指定活頁簿及指定欄位的資料
- `[NUMERIC,TXT,RAW]=xlsread(full_filepath,1,'H11887:J23746');`



|       | E   | F   | G | H        | I        | J        | K       | L        | M       |
|-------|-----|-----|---|----------|----------|----------|---------|----------|---------|
| 11887 | 254 | 232 | 0 | -4.13259 | 20.42184 | 883.9454 | 6641.17 | -48374   | 2131.38 |
| 11888 | 254 | 232 | 0 | -4.37233 | 20.68468 | 883.2789 | 6665.67 | -48381.4 | 2147.57 |
| 11889 | 254 | 232 | 0 | -4.20109 | 20.90181 | 882.9234 | 6669.67 | -48370.8 | 2143.02 |
| 11890 | 254 | 232 | 0 | -3.97277 | 20.68468 | 883.4122 | 6688.3  | -48358.4 | 2145.38 |
| 11891 | 254 | 232 | 0 | -4.16684 | 20.43327 | 883.7677 | 6708.39 | -48390.4 | 2166.13 |
| 11892 | 254 | 232 | 0 | -4.14401 | 20.25042 | 883.8121 | 6743.47 | -48400.4 | 2173.06 |
| 11893 | 254 | 232 | 0 | -4.26958 | 20.45612 | 883.2789 | 6702.73 | -48408.3 | 2135.98 |
| 11894 | 254 | 232 | 0 | -4.33808 | 20.74182 | 883.0567 | 6730.89 | -48391.8 | 2145.27 |
| 11895 | 254 | 232 | 0 | -4.05268 | 20.70754 | 883.7232 | 6719.03 | -48359.4 | 2163.89 |
| 11896 | 254 | 232 | 0 | -3.96135 | 20.59326 | 883.5899 | 6710.41 | -48385.8 | 2166.22 |
| 11897 | 254 | 232 | 0 | -4.07551 | 20.31899 | 883.19   | 6740.3  | -48367.3 | 2175.41 |
| 11898 | 254 | 232 | 0 | -4.44082 | 20.46755 | 882.8345 | 6720.17 | -48405.2 | 2161.42 |
| 11899 | 254 | 232 | 0 | -4.53215 | 20.70754 | 883.0123 | 6718.82 | -48383.3 | 2186.92 |
| 11900 | 254 | 232 | 0 | -4.281   | 20.59326 | 883.5455 | 6737.99 | -48396.8 | 2159.17 |

## FILE INPUT 常用函式列表

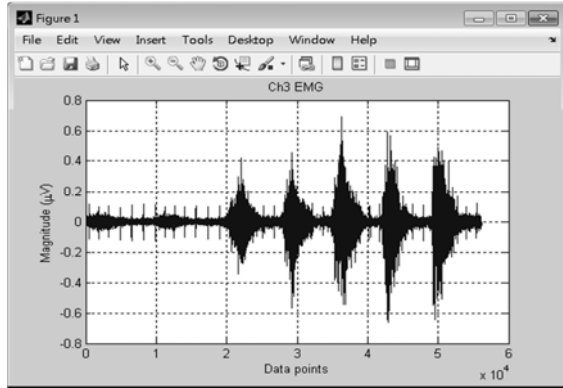
- `uigetfile` % 呼叫開啟檔案對話視窗選取檔案路徑
- `load` % 讀取ASCII 或 \*.mat 檔案 (MATLAB 專用檔案格式)
- `fopen` % 指派檔案ID (以正整數代表) 給所選路徑檔案
- `fread` % 以特定 precision 讀取 binary 格式檔案
- `fscanf` % 讀取 ASCII 檔案內容
- `fseek` % 移動檔案游標位置
- `fclose` % 釋放檔案ID 與所選路徑檔案
- `reshape` % 將向量/矩陣大小重新調整
- `xlsread` % 讀取 \*.xls 或 \*.csv 等試算表或分隔符號檔案

請在 **command window** 中 **help** 各函式並學會其用法!!

## MATLAB 訊號繪製與確認

## 繪製曲線

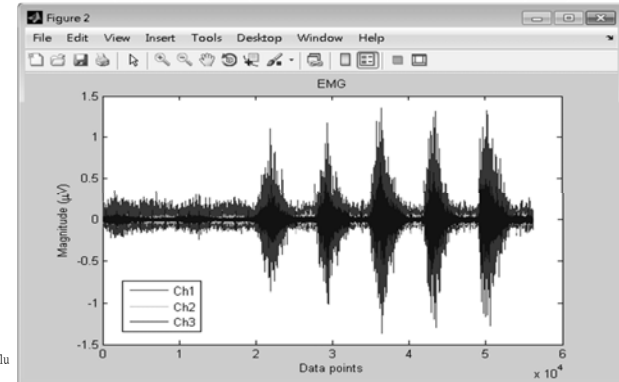
- 請執行 `demodata\ASCII\load_ASCII.m` 讀取呼吸肌肌電訊號
- 請執行 `demodata\ASCII\plot_EMG.m` 繪製呼吸肌肌電訊號



←單一視窗單一圖軸單一訊號

## 繪製曲線

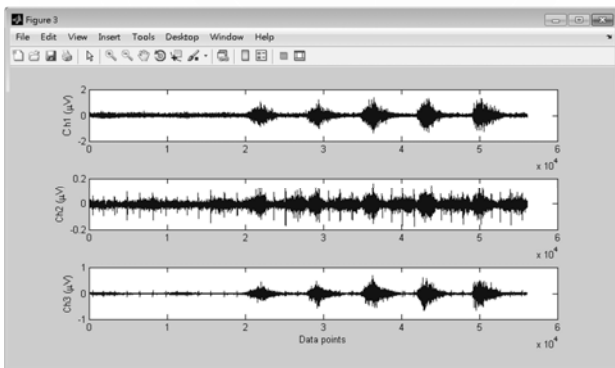
- 請執行 `demodata\ASCII\load_ASCII.m` 讀取呼吸肌肌電訊號
- 請執行 `demodata\ASCII\plot_EMG.m` 繪製呼吸肌肌電訊號



←單一視窗單一圖軸多訊號

## 繪製曲線

- 請執行 `demodata\ASCII\load_ASCII.m` 讀取呼吸肌肌電訊號
- 請執行 `demodata\ASCII\plot_EMG.m` 繪製呼吸肌肌電訊號



←單一視窗多圖軸單一訊號

## 訊號繪製常用函式列表

- `figure` % 開啟新視窗
- `subplot` % 指定子圖軸
- `plot` % 繪製資料點或曲線
- `title` % 圖軸標題
- `xlabel` % X軸標註
- `ylabel` % Y軸標註
- `xlim` % X軸呈現範圍
- `ylim` % Y軸呈現範圍
- `grid` % `grid on` 顯示網格線
- `hold` % `hold on` 可讓 `plot` 指令不會抹除已繪製曲線
- `legend` % 在圖軸中加上圖例

請在 **command window** 中 **help** 各函式並學會其用法!!



## MATLAB檔案覆寫與儲存

## 儲存MATLAB專用檔案格式

- 請開啟 `demodata/save_data.m`
- \*.mat格式可以直接用 `save` 儲存，用 `load` 讀取
- `save(filename)`
- `save(filename,variable1,variable2,...)`
- `save(filename,variable1,variable2,...,format)`

## ASCII資料覆寫

- 請開啟並執行 `demodata/fprintf_ASCII.m`

- 認識 `fprintf`

|   |                      |
|---|----------------------|
| <code>fprintf(fid,'%15s','This is a test.')</code>          | % 寫入15個字元的字串         |
| <code>fprintf(fid,'\r\n')</code>                            | % 換行 即十六位元的0D 0A     |
| <code>fprintf(fid,'%11s %8d','The date is',20131011)</code> | % 寫入11個字元的字串與8位數字的整數 |
| <code>fprintf(fid,'\r\n')</code>                            | % 換行 即十六位元的0D 0A     |
| <code>fprintf(fid,'%5.4f',12345.1234)</code>                | % 寫入浮點數(5位整數、4位小數)   |

## BINARY資料覆寫

- 請開啟並執行 `demodata/fwrite_binary.m`

- 認識 `fwrite`

- `data=[23.456 19.664 -14.123 27.548 -10.168];`
- `fwrite(fid,data,'double');`

切記：要讀取binary資料必須要知道儲存格式  
與他人做binary資料交換時必須一併告知格式!!

## EXCEL 資料覆寫

- 請開啟並執行 `demodata/xlswrite_excel.m`
- 認識 `xlswrite`
  - `xlswrite(full_filepath,data,3,'B10');` % 將資料儲存在第3活頁簿的B10欄位

儲存成EXCEL相容格式可以方便使用其他軟體進行資料統計分析

## FILE OUTPUT 常用函式列表

- `uiputfile` % 呼叫儲存檔案對話視窗選取檔案路徑
- `save` % 儲存 ASCII 或 \*.mat 檔案 (MATLAB 專用檔案格式)
- `fopen` % 指派檔案ID (以正整數代表) 給所選路徑檔案
- `fseek` % 移動檔案游標位置
- `fwrite` % 以特定 precision 覆寫 binary 格式檔案
- `fprintf` % 覆寫 ASCII 檔案內容
- `fclose` % 釋放檔案ID 與所選路徑檔案
- `xlswrite` % 覆寫 \*.xls 或 \*.csv 等試算表或分隔符號檔案

請在 **command window** 中 **help** 各函式並學會其用法!!

# THE END

<http://www.ym.edu.tw/~cflu>