



Úvod do Matlabu

Prednáška č. 9

- **Load a Save – práca s .mat súborom**
- Import a export dát z/do súborov s formátovaným textom
- Import a export súborov MS Excel
- Import pomocou funkcie **uimport()**
- Načítanie a uloženie obrazových súborov
- Načítanie a prehrávanie zvukových súborov

Load a Save – práca s .mat súborom

Pre Matlab najprirodzenejším spôsobom ukladania a načítavania dát je práca s súbormi typu **.mat**. Tieto súbory v sebe uchovávajú premenné v takej podobe, ako sú uložené vo **workspace**. Uloženie vybraných premenných je možné dosiahnuť pomocou príkazu

- `save('názov_súboru.mat', 'premenná_1', ... , 'premenná_N')`

Naopak, pokiaľ je potrebné načítať premenné uložené na pevnom disku, použijeme príkaz

- `load('názov_súboru.mat')`

Spôsob použitia oboch príkazov je demonštrovaný v nasledovnom skripte a výpise príkazového riadka.

Ukladanie údajov do súborov .mat je výhodné hlavne vtedy, ak vieme, že tieto dáta budeme vždy spracovávať pomocou Matlabu. Hlavná výhoda spočíva v tom, že uložené údaje sú uložené v jednom súbore, ale dáta sú uložené ako jednotlivé premenné workspace.

```
1      % Definícia premenných
2 -    A = [1 2 3 4];
3 -    B = 5;
4 -    c = 'ahoj svet';
5      % Zobrazenie premenných v % workspace
6 -    who
7      % Export premenných
8 -    save('moje_premenne.mat','A','B');
9      % Vymazanie obsahu workspace
10 -   clear;
11     % Import premenných
12 -   load('moje_premenne.mat');
13     % Zobrazenie premenných v % workspace
14 -   who
```

Current Folder		Command Window
		>> load_save
		Your variables are:
		A B c
		Your variables are:
		A B



Úvod do Matlabu

Prednáška č. 9

- Load a Save – práca s .mat súborom
- **Import a export dát z/do súborov s formátovaným textom**
- Import a export súborov MS Excel
- Import pomocou funkcie **uimport()**
- Načítanie a uloženie obrazových súborov
- Načítanie a prehrávanie zvukových súborov

Import a export z/do súborov s formátovaným textom

Častokrát je potrebné spracovávať údaje, ktoré sú do súboru uložené vo forme formátovaného textu. Jednotlivé dáta uložené v takomto súbore sú od seba oddelené separátorom – **Delimiter** (bodka, čiarka, bodkočiarka, medzera a pod.)

Je nevyhnutné, aby tieto dáta boli uložené konzistentne!

Prípona samotného súboru nie je dôležitá (.txt, .dat, .fpv, .csv a vlastné prípony), dôležité je len to, aby údaje boli uložené ako formátovaný text.

Bežne sa stretávame s formátom **CSV - Comma-separated values**.

Dáta separované medzerou a desatinná časť je za čiarkou:

```
No Sensor1 Sensor2 Sensor3 Sensor4 Sensor5 Sensor6
1 0,0102 1,0816 0,1428 -0,5204 0,8061 -0,2346
2 0,2040 0,8979 0,1122 0,4081 0,7040 -0,1020
3 0,1734 1,0510 0,0816 0,0102 0,5612 0,1836
4 -0,2040 0,8979 0,1122 -0,3979 0,7040 -0,1020
5 -0,1428 1,0510 0,0816 0,5204 0,2857 -0,1224
6 0,5306 0,8061 -0,2346 0,6326 0,4897 -0,7040
```

Nejedná sa o najlepší spôsob uloženia dát, nakoľko po načítaní bude potrebný postprocessing. No niekedy si nepomôžeme a merací prístroj ma takýto výstup.

Dáta separované čiarkou a desatinná časť je za bodkou:

```
No,Sensor1,Sensor2,Sensor3,Sensor4,Sensor5,Sensor6
1,0.0102,1.0816,0.1428,-0.5204,0.8061,-0.2346
2,0.204,0.8979,0.1122,0.4081,0.704,-0.102
3,0.1734,1.051,0.0816,0.0102,0.5612,0.1836
4,-0.204,0.8979,0.1122,-0.3979,0.704,-0.102
5,-0.1428,1.051,0.0816,0.5204,0.2857,-0.1224
6,.0.5306,0.8061,-0.2346,0.6326,0.4897,-0.704
7,0.5306,0.8061,-0.2346,0.6326,0.4897,-0.704
```

Omnoho vhodnejší spôsob uloženia dát.

Import z textového súboru - `readtable()`, `readcell()`

Na načítanie textového súboru môžeme použiť napríklad funkciu `readtable()`. Z výpisu súboru je zrejmé, že ide o dáta, ktorých prvý riadok obsahuje názvy premenných a ostatné riadky sú hodnoty týchto premenných.

Dáta sú uložené v údajovom type `table` (tabuľka). V prípade potreby je ich možné previesť na iný údajový typ.

Delimiter je v tomto prípade medzera - ' '

Data.txt

```
No Sensor1 Sensor2 Sensor3 Sensor4 Sensor5 Sensor6
1 0,0102 1,0816 0,1428 -0,5204 0,8061 -0,2346
2 0,2040 0,8979 0,1122 0,4081 0,7040 -0,1020
3 0,1734 1,0510 0,0816 0,0102 0,5612 0,1836
4 -0,2040 0,8979 0,1122 -0,3979 0,7040 -0,1020
5 -0,1428 1,0510 0,0816 0,5204 0,2857 -0,1224
6 0,5306 0,8061 -0,2346 0,6326 0,4897 -0,7040
```

```
T = readtable('Data.txt', 'Delimiter', ' ')
```

```
T =
```

No	Sensor1	Sensor2	Sensor3	Sensor4	Sensor5	Sensor6
1	'0,0102'	'1,0816'	'0,1428'	'-0,5204'	'0,8061'	'-0,2346'
2	'0,2040'	'0,8979'	'0,1122'	'0,4081'	'0,7040'	'-0,1020'
3	'0,1734'	'1,0510'	'0,0816'	'0,0102'	'0,5612'	'0,1836'
4	'-0,2040'	'0,8979'	'0,1122'	'-0,3979'	'0,7040'	'-0,1020'
5	'-0,1428'	'1,0510'	'0,0816'	'0,5204'	'0,2857'	'-0,1224'
6	'0,5306'	'0,8061'	'-0,2346'	'0,6326'	'0,4897'	'-0,7040'

Import z textového súboru - `readtable()`, `readcell()`

Dáta sú načítané do údajového typu `table` a je zrejmé, že jednotlivé hodnoty sú načítané ako reťazce a navyše desatinná čiarka jednotlivých údajov je reprezentovaná znakom čiarky.

Ak chceme s dátami ďalej pracovať, je potrebné **konvertovať tabuľku na pole buniek**, následne na to pomocou príkazu pre prácu s reťazcami **vymeniť znak čiarky za bodku** a nakoniec jednotlivé údaje **konvertovať z reťazcov uložených v poly buniek na maticu čísel**.

```
T = readtable('Data.txt', 'Delimiter', ' ')
```

```
T =
```

No	Sensor1	Sensor2	Sensor3	Sensor4	Sensor5	Sensor6
1	'0,0102'	'1,0816'	'0,1428'	'-0,5204'	'0,8061'	'-0,2346'
2	'0,2040'	'0,8979'	'0,1122'	'0,4081'	'0,7040'	'-0,1020'
3	'0,1734'	'1,0510'	'0,0816'	'0,0102'	'0,5612'	'0,1836'
4	'-0,2040'	'0,8979'	'0,1122'	'-0,3979'	'0,7040'	'-0,1020'
5	'-0,1428'	'1,0510'	'0,0816'	'0,5204'	'0,2857'	'-0,1224'
6	'0,5306'	'0,8061'	'-0,2346'	'0,6326'	'0,4897'	'-0,7040'

```
k = table2cell(T);
```

```
K = str2double(strrep(k(:,2:end), ',', '.'));
```

```
>> disp(K);
```

```
    0.0102    1.0816    0.1428   -0.5204    0.8061   -0.2346
    0.2040    0.8979    0.1122    0.4081    0.7040   -0.1020
    0.1734    1.0510    0.0816    0.0102    0.5612    0.1836
   -0.2040    0.8979    0.1122   -0.3979    0.7040   -0.1020
   -0.1428    1.0510    0.0816    0.5204    0.2857   -0.1224
    0.5306    0.8061   -0.2346    0.6326    0.4897   -0.7040
```

Export do textového súboru - writetable(), writecell()

Export dát reprezentovaných tabuľkou môžeme vykonať pomocou funkcie **writetable()**

Najprv je potrebné dáta uložené v matici (double) **previesť na údajový typ table**. Pritom je možné **zadefinovať aj názvy jednotlivých stĺpcov tabuľky**.

```
A = array2table(K, 'VariableNames', {'Sensor1', 'Sensor2', 'Sensor3', 'Sensor4', 'Sensor5', 'Sensor6'});  
writetable(A, 'New_data.dat');
```

Výpis súboru New_data.dat

```
Sensor1, Sensor2, Sensor3, Sensor4, Sensor5, Sensor6  
0.0102, 1.0816, 0.1428, -0.5204, 0.8061, -0.2346  
0.204, 0.8979, 0.1122, 0.4081, 0.704, -0.102  
0.1734, 1.051, 0.0816, 0.0102, 0.5612, 0.1836  
-0.204, 0.8979, 0.1122, -0.3979, 0.704, -0.102  
-0.1428, 1.051, 0.0816, 0.5204, 0.2857, -0.1224  
0.5306, 0.8061, -0.2346, 0.6326, 0.4897, -0.704  
0.5306, 0.8061, -0.2346, 0.6326, 0.4897, -0.704
```



Úvod do Matlabu

Prednáška č. 9

- Load a Save – práca s .mat súborom
- Import a export dát z/do súborov s formátovaným textom
- **Import a export súborov MS Excel**
- Import pomocou funkcie **uimport()**
- Načítanie a uloženie obrazových súborov
- Načítanie a prehrávanie zvukových súborov

Import z .xls/.xlsx súboru - xlsread()

data.xls

Ďalšou možnosťou importu a exportu údajov sú súbory programového balíka MS Excel. MS Excel je veľmi obľúbeným, nakoľko je používateľsky priateľský a umožňuje pomerne zložité výpočty. Súbory MS Excel môžeme importovať pomocou funkcie `xlsread()`. Vstupnými parametrami tejto funkcie sú **názov vstupného súboru**, **názov hárku** a **tiež môžu byť špecifikované aj bunky**, ktoré sa majú načítať.

Príkaz `xlsread()` môže vrátiť až tri výstupné premenné.

```
[T_n, T_t, T_raw] = xlsread('data.xls', 'Harok1');
```

Príkaz `xlsread()` môže vrátiť až tri výstupné premenné.

`T_n` - iba číselné hodnoty

`T_t` - iba textové hodnoty

`T_raw` - údaje „v prirodzenej podobe“. Takto načítané dáta sú uložené ako bunky (cell). Prázdne riadky sú nahradené hodnotou NaN.

Ak chceme načítať mapr. len RAW dáta tak môžeme použiť znak tilda „~“

```
[~, ~, T_raw] = xlsread(...
```

	A	B	C	D	E	F	G
1							
2		No	Sensor1	Sensor2	Sensor3	Sensor4	Sensor6
3		1	0,742788	0,516566	0,208592	0,073685	0,867435
4		2	0,353162	0,900247	0,165524	0,879975	0,0845
5		3	0,131603	0,613179	0,735611	0,389162	0,931715
6		4	0,77361	0,822411	0,925677	0,055386	0,934585
7		5	0,523947	0,264227	0,242776	0,422668	0,118324
8		6	0,426022	0,325796	0,359274	0,673656	0,785632
9							

```
[T_n, T_t, T_raw] = xlsread('data.xls', 'Harok1')
```

```
T_n =
```

```
1.0000    0.7428    0.5166    0.2086    0.0737    0.8674
2.0000    0.3532    0.9002    0.1655    0.8800    0.0845
3.0000    0.1316    0.6132    0.7356    0.3892    0.9317
4.0000    0.7736    0.8224    0.9257    0.0554    0.9346
5.0000    0.5239    0.2642    0.2428    0.4227    0.1183
6.0000    0.4260    0.3258    0.3593    0.6737    0.7856
```

```
T_t =
```

```
'No'      'Sensor1'    'Sensor2'    'Sensor3'    'Sensor4'    'Sensor6'
```

```
T_raw =
```

```
[NaN]     [NaN]     [NaN]     [NaN]     [NaN]     [NaN]
'No'      'Sensor1' 'Sensor2' 'Sensor3' 'Sensor4' 'Sensor6'
[ 1]     [ 0.7428] [ 0.5166] [ 0.2086] [ 0.0737] [ 0.8674]
[ 2]     [ 0.3532] [ 0.9002] [ 0.1655] [ 0.8800] [ 0.0845]
[ 3]     [ 0.1316] [ 0.6132] [ 0.7356] [ 0.3892] [ 0.9317]
[ 4]     [ 0.7736] [ 0.8224] [ 0.9257] [ 0.0554] [ 0.9346]
[ 5]     [ 0.5239] [ 0.2642] [ 0.2428] [ 0.4227] [ 0.1183]
[ 6]     [ 0.4260] [ 0.3258] [ 0.3593] [ 0.6737] [ 0.7856]
```

Export do .xls/.xlsx súboru - `xlswrite()` a `writetable()`

Export dát dosiahneme pomocou príkazu `xlswrite()`. Povedzme, že chceme zapísať pozmenené dáta `T_raw` do ďalšieho hárku toho istého súboru.

```
xlswrite('data.xls', T_raw, 'Harok2')
```

Tiež je možné použiť funkciu `writetable()` v takom prípade je potrebné urobiť prevod cell na table.

```
writetable(cell2table(T_raw), 'data2.xlsx')
```

Alebo tiež je možné použiť aj `writecell()`.

```
writecell(T_raw, 'data2.xlsx')
```

Použitím všetkých troch metód je možné dosiahnuť porovnateľný výsledok.

Dáta je možné ukladať aj mnohými inými spôsobmi. Detaily ako jednotlivé funkcie použiť je možné nájsť na [MathWorks-e](#).

```
>> disp(T_raw)
    [NaN]    [    NaN]    [    NaN]    [    NaN]    [    NaN]    [    NaN]
    'No'    'Sensor1'    'Sensor2'    'Sensor3'    'Sensor4'    'Sensor6'
    [  1]    [ 0.7428]    [ 0.3532]    [ 0.1316]    [ 0.7736]    [ 0.5239]
    [  2]    [ 0.5166]    [ 0.9002]    [ 0.6132]    [ 0.8224]    [ 0.2642]
    [  3]    [ 0.2086]    [ 0.1655]    [ 0.7356]    [ 0.9257]    [ 0.2428]
    [  4]    [ 0.0737]    [ 0.8800]    [ 0.3892]    [ 0.0554]    [ 0.4227]
    [  5]    [ 0.8674]    [ 0.0845]    [ 0.9317]    [ 0.9346]    [ 0.1183]
    [  6]    [ 0.3892]    [ 0.0554]    [ 0.4227]    [ 0.0554]    [ 0.4227]

>> xlswrite('data.xls', T_raw, 'Harok2')
>> writetable(cell2table(T_raw), 'data2.xlsx');
>> writecell(T_raw, 'data3.xlsx');
```



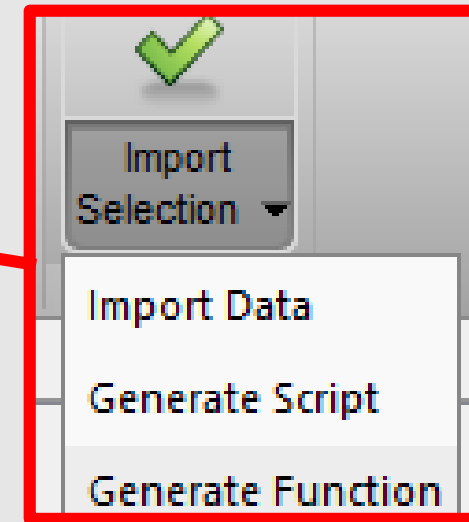
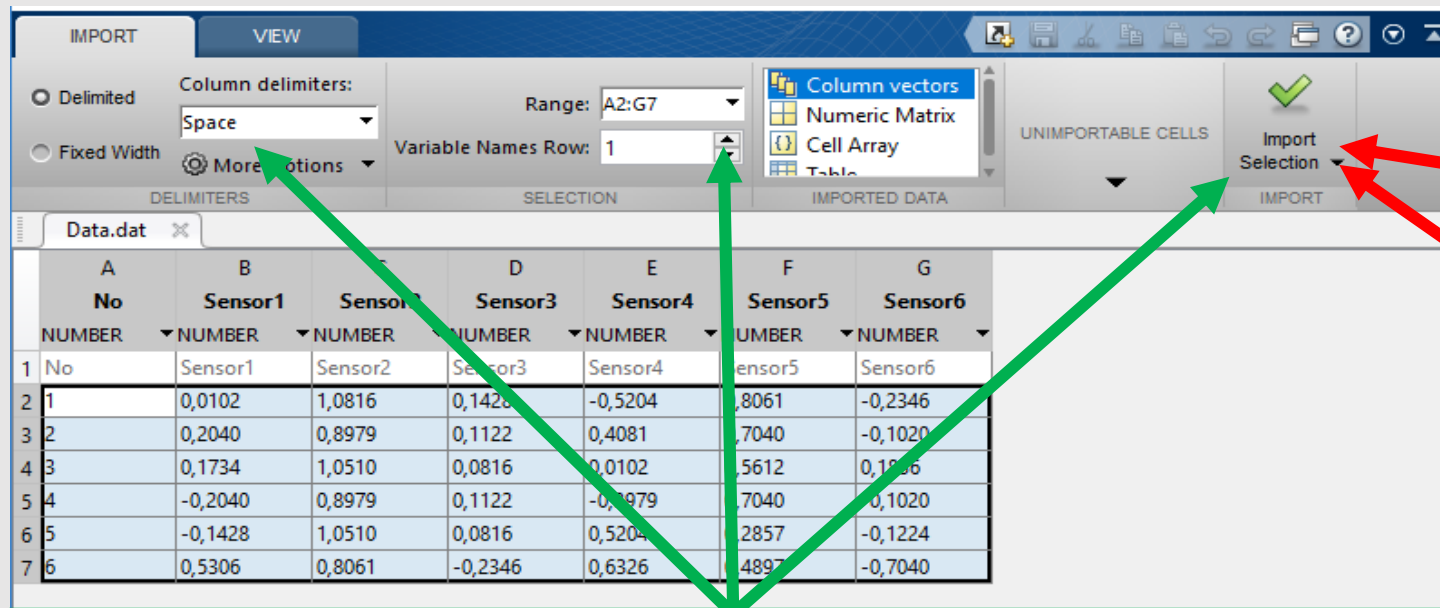
Úvod do Matlabu

Prednáška č. 9

- Load a Save – práca s .mat súborom
- Import a export dát z/do súborov s formátovaným textom
- Import a export súborov MS Excel
- **Import pomocou funkcie uiimport()**
- Načítanie a uloženie obrazových súborov
- Načítanie a prehrávanie zvukových súborov

Import pomocou funkcie uiimport()

V predchádzajúcich príkladoch sme ukázali, ako je možné importovať údaje priamo zo skriptu alebo funkcie. To je výhodné vtedy, ak chceme, aby napísaný kód importoval dáta automatizovane. Matlab však má na import dát tiež implementovanú funkciu s grafickým rozhraním. Funkcia podporuje širokú škálu súborových formátov. Pre načítanie dát zo súboru data.dat ju použijeme nasledovne: `uiimport('data.dat')`. Vykonaním uvedeného príkazu sa zobrazí hlavné okno nástroja na import dát:



V hornej lište je možné vybrať znak, ktorým sú od seba jednotlivé údaje oddelené (delimiter), rozsah údajov a údajový typ, ktorého bude výsledná premenná. Po tom, ako sú nastavené všetky parametre importu, je možné dáta priamo importovať stlačením tlačidla **Import Selection**

Tiež je možné vygenerovať funkciu, ktorú potom môžeme použiť vo svojom riešení na import zvolených dát. **Tento nástroj je veľmi šikovný, ale určite nie je optimálny.**



Úvod do Matlabu

Prednáška č. 9

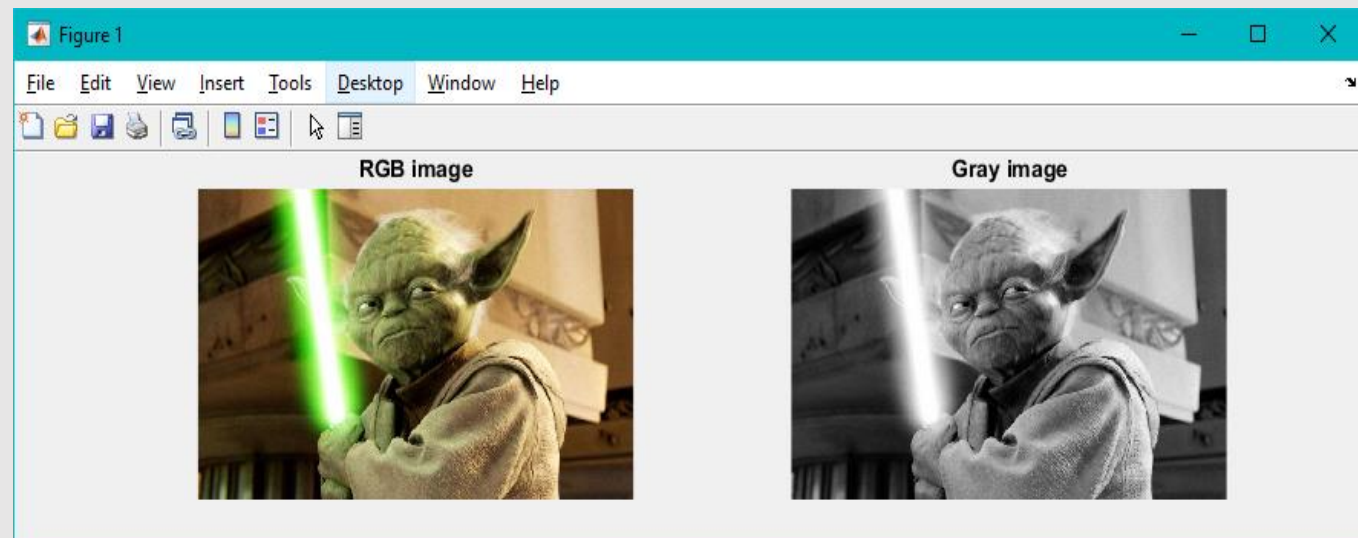
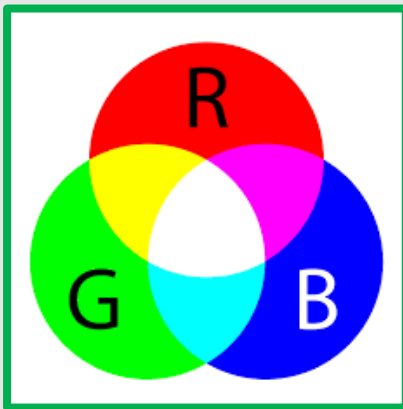
- Load a Save – práca s .mat súborom
- Import a export dát z/do súborov s formátovaným textom
- Import a export súborov MS Excel
- Import pomocou funkcie uimport()
- **Načítanie a uloženie obrazových súborov**
- Načítanie a prehrávanie zvukových súborov

Načítanie a uloženie obrazových súborov

Častokrát sa môžete stretnúť s úlohami, ktoré budú vyžadovať prácu s obrazovým súborom rastrovej grafiky. Ide napríklad o .jpg, .bmp, .png a mnoho ďalších. Preto je potrebné vedieť minimálne to ako sa taký súbor načíta a ako je potom reprezentovaný vo workspace. Pre načítanie súboru používame funkciu `imread()`. Táto spravidla načíta obraz ako maticu typu `uint8` o rozmere `MxN` alebo `MxNx3` v prípade farebného obrazu. Pri farebnom obraze si to môžeme predstaviť ako 3 matice uložené na sebe. Každá matica zobrazuje iný farebný kanál (RGB).

```
2 - file = 'StarWars.jpg';
3 - % Načítanie farebného obrazu
4 - I_RGB = imread(file);
5 - % Prevod obrazu na odtiene sivej
6 - I_gray = rgb2gray(I_RGB);
7 - % Uloženie nového obrazu
8 - imwrite(I_gray, 'StarWarsG.jpg');
9 - % Zobrazenie oboch obrazov
10 - figure(1),
11 - subplot(1,2,1), imshow(I_RGB); title('RGB image');
12 - subplot(1,2,2), imshow(I_gray); title('Gray image');
```

```
>> whos
Name          Size          Bytes  Class  Attributes
I_RGB         614x959x3     1766478  uint8
I_gray        614x959       588826   uint8
```





Úvod do Matlabu

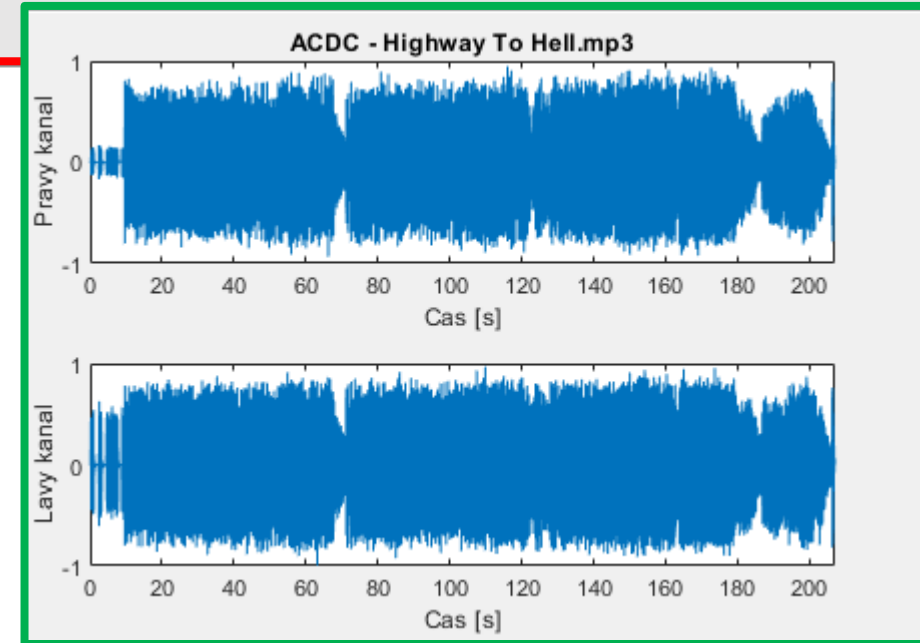
Prednáška č. 9

- Load a Save – práca s .mat súborom
- Import a export dát z/do súborov s formátovaným textom
- Import a export súborov MS Excel
- Import pomocou funkcie uimport()
- Načítanie a uloženie obrazových súborov
- **Načítanie a prehrávanie zvukových súborov**

Načítanie, prehrávanie a uloženie audio súboru

V Matlabe je možné efektívne pracovať aj so zvukovými súbormi. V nasledujúcom príklade je zobrazený kód pre načítanie audio-súboru, zobrazenie informácií o tomto súbore a tiež jeho prehrávanie.

```
1 % Vlastný súbor mp3 si každý zaobstará ako vie ;-)  
2 file = "ACDC - Highway To Hell.mp3";  
3 % Zobrazit informácie o audionahrávke  
4 info = audioinfo(file);  
5 disp(info);  
6 % Načítať audionahrávku  
7 [y,Fs] = audioread(file);  
8 % y - samotné dáta  
9 % Fs- vzorkovacia frekvencia (zvyčajne 44kHz)  
10  
11 % Vytvorenie vektoru času od 0 do počet vzoriek predelený Fs. Je tam krok  
12 % 10 vzoriek, aby sa to dalo "normalne" zobraziť.  
13 time = 0:10/Fs:length(y)/Fs;  
14 % Samotné zobrazenie priebehov oboch kanálov (každá 10. vzorka sa vykresluje)  
15 figure(1),  
16 subplot(2,1,1), plot(time,y(1:10:end,1)); xlabel('Cas [s]'); ylabel('Pravy kanal'); xlim([ 0 length(y)/Fs]);title(file);  
17 subplot(2,1,2), plot(time,y(1:10:end,2)); xlabel('Cas [s]'); ylabel('Lavy kanal');xlim([ 0 length(y)/Fs]);  
18 % Načítanie nahrávky do prehrávača  
19 player = audioplayer(y, Fs);  
20 % Spustiť skladbu  
21 play(player);  
22 % Tu sa program pozastaví a skladba sa prehráva. Po stlačení enter v  
23 % príkazovom riadku sa pokračuje na ďalší krok, ktorý ukončí prehrávanie.  
24 pause  
25 stop(player);
```



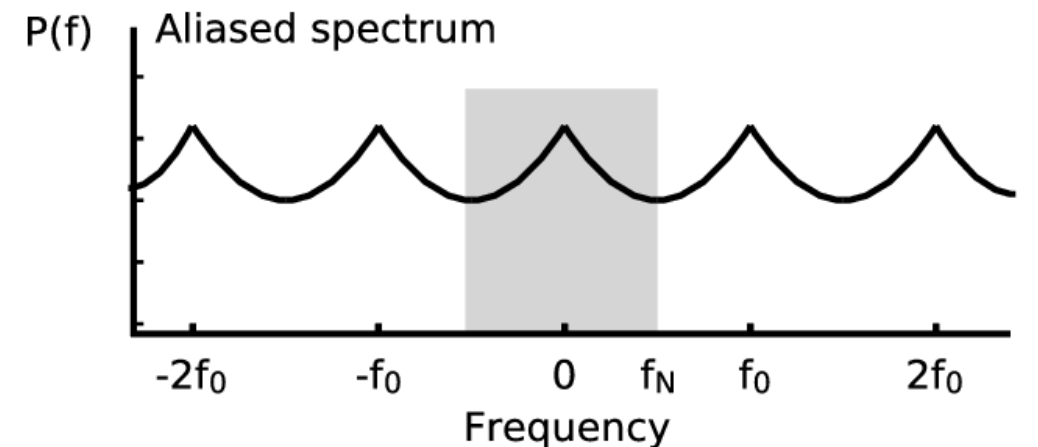
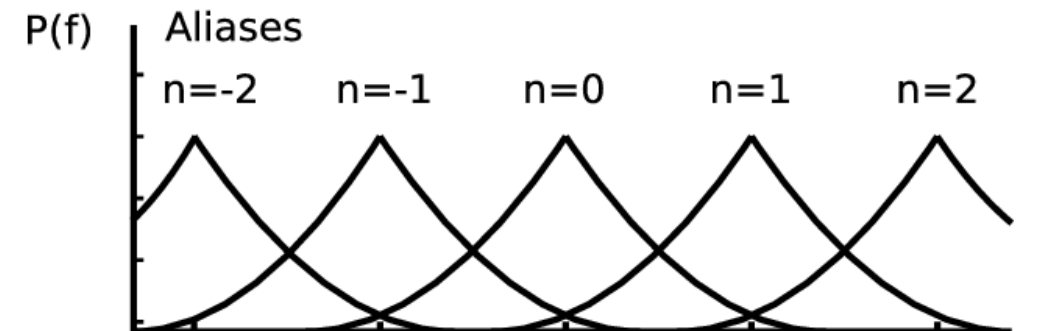
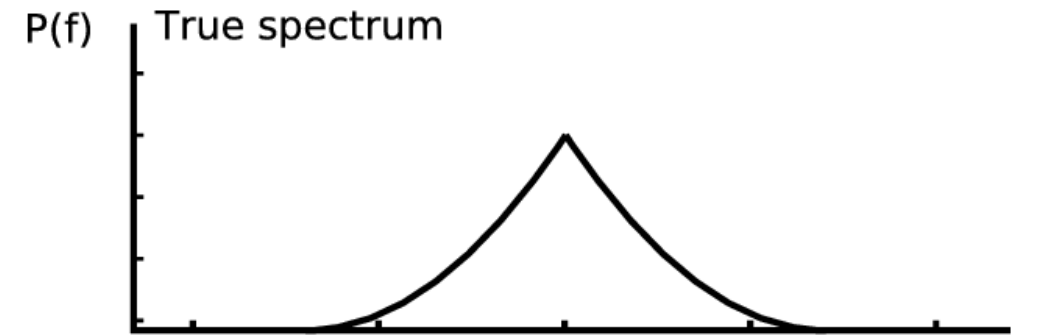
```
>> disp(info);  
      Filename: 'ACDC - Highway To Hell.mp3'  
      CompressionMethod: 'MP3'  
      NumChannels: 2  
      SampleRate: 44100  
      TotalSamples: 9131904  
      Duration: 207.0727  
      Title: 'Highway to Hell'  
      Comment: []  
      Artist: 'AC/DC'  
      BitRate: 192
```


Načítanie, prehrávanie a uloženie audio súboru

Nahrávku je možné uložiť pomocou funkcie audiowrite().

Nasledujúci príklad demonštruje podvzorkovanie zvukového signálu. Z pôvodného záznamu sa uloží len každá desiatu vzorka. Simuluje sa tak zmena vzorkovacej frekvencie z 44kHz na 4,4kHz. Na SAS ste sa učili o aliasingu, tak takto je ten aliasing „počut“.

```
27 % Zoberie sa každá 10 vzorka
28 - Y_d = y(1:10:end,:);
29 - Fs_d = Fs/10;
30 % Podvzorkovaná nahrávka sa uloží
31 - audiowrite('Audio.wav',Y_d,Fs_d);
32 % Ak pustíte nahrávku napr. cez VLC malo by byť počut výrazné skreslenie
33 % oproti originálu
```



Nabudúce

- Matlab a periférie
- Sériová linka a Arduino

