



MATLAB[®]

Začínáme se signály



Karel Zaplatílek

Tato elektronická publikace byla zakoupena u: Karel Zaplatílek, slunce007@seznam.cz

Jméno a příjmení kupujícího:

E-mail kupujícího:

Upozorňuji, že elektronická kniha je dílem chráněným podle autorského zákona a je určena jen pro osobní potřebu kupujícího. Kniha jako celek ani žádná její část nesmí být volně šířena na internetu, ani jinak dále zveřejňována.

Obsah

Předmluva	7
1 Úvod	8
1.1 Proč jsem knihu napsal a pro koho je určena	8
1.2 Co je potřeba k práci	8
1.3 Práce s literaturou	10
1.4 Licenční politika tvůrců systému MATLAB	10
2 Signály a jejich modely	11
2.1 Pojem signál	11
2.1.1 Praktické příklady signálů	11
2.1.2 Zajímavé souvislosti, vztahující se k pojmu signál	13
2.2 Modely signálů	14
2.2.1 Skutečný, reálně existující signál	14
2.2.2 Modely signálů	15
2.3 Klasifikace signálů a jejich modelů	18
2.3.1 Signály se spojitým a diskrétním průběhem veličin	18
2.3.2 Signály periodické a aperiodické	19
2.3.2.1 Periodické signály	19
2.3.2.2 Aperiodické signály	21
2.3.3 Signály deterministické a stochastické	22
2.3.3.1 Deterministické signály	22
2.3.3.2 Stochastické signály	23
2.3.4 Signály jednorozměrné a vícerozměrné	25
3 MATLAB – nástroj pro práci s čísly	26
3.1 Číslíkové signály a jejich zobrazení	26
3.1.1 Cejchování vodorovné osy grafu	27
3.1.2 Vzorkování a diskrétní čas	29
3.1.2.1 Vzorkování	30
3.1.2.2 Diskrétní čas	32
3.2 Možnosti práce s analogovými signály	33
3.3 Shrnutí používaných termínů a symbolů	35
4 Tvorba a získávání signálů v prostředí MATLAB	37
4.1 Vektory a matice jako číslíková reprezentace signálů	37
4.1.1 Tvorba jednoduchých signálů a práce s nimi	38
4.1.2 Tvorba a využití vícerozměrných matic	40
4.1.3 Tvorba a využití pole buněk (Cell Arrays)	41
4.1.4 Práce se strukturami	42
4.1.5 Práce s různými formáty čísel	43
4.2 Ukládání a načítání proměnných	44
4.2.1 Ukládání dat do souboru	44
4.2.1.1 Ukládání dat v ASCII podobě	45

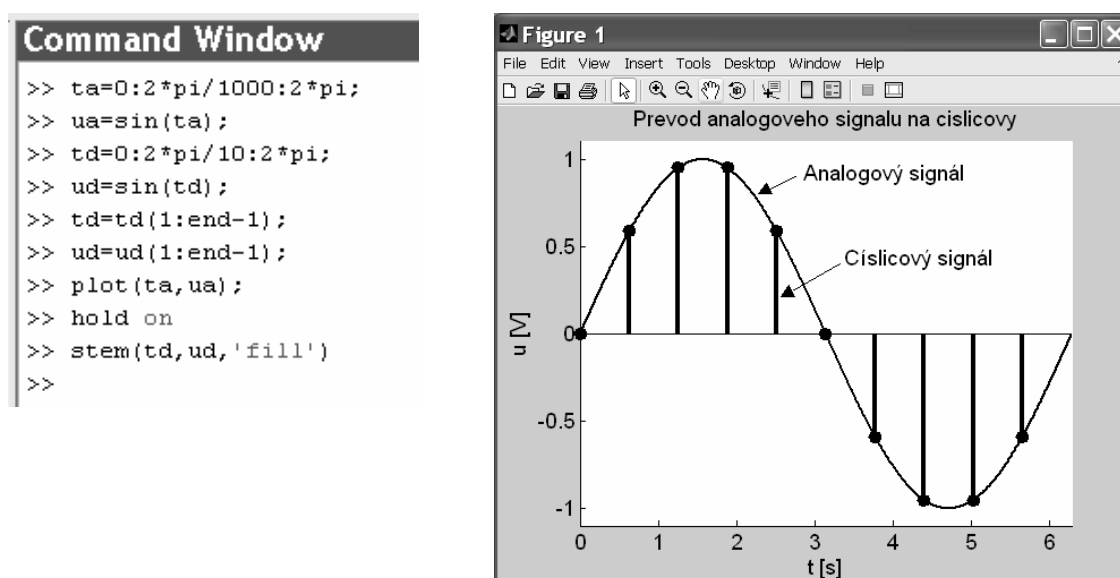
4.2.1.2	Ukládání dat v binární podobě	50
4.2.2	Čtení dat ze souboru	51
4.2.2.1	Čtení dat v ASCII podobě	51
4.2.2.2	Čtení dat v binární podobě	53
4.2.3	Interaktivní podpora práce s proměnnými	58
4.2.3.1	Práce s panelem „Workspace“	58
4.2.3.2	Práce s oknem „Import Wizard“	60
4.3	Signal Processing Toolbox – základní modul pro práci se signály	62
4.4	Použití vestavěných generátorů signálů	63
4.4.1	Generování harmonických signálů	63
4.4.1.1	Práce s příkazy „sin“ a „cos“	64
4.4.1.2	Harmonický signál s proměnným kmitočtem	65
4.4.1.3	Napětově řízený oscilátor (vco)	66
4.4.2	Generování obdélníkových signálů	67
4.4.2.1	Generování periodických obdélníkových signálů	67
4.4.2.2	Generování aperiodických obdélníkových signálů	68
4.4.3	Diskrétní jednotkový impuls a diskrétní jednotkový skok	69
4.4.4	Generování pilovitých signálů	70
4.4.4.1	Generování periodických pilovitých signálů	71
4.4.4.2	Generování aperiodických pilovitých signálů	72
4.4.5	Generování Gaussovsky modulovaného harmonického signálu	74
4.4.5.1	Generování periodických průběhů	74
4.4.5.2	Generování aperiodických průběhů	75
4.4.6	Funkce pro periodické opakování vybraných aperiodických průběhů	76
4.4.7	Generování signálů s průběhem tvaru „sinc“	78
4.4.8	Generování modulovaných signálů	78
4.4.8.1	Amplitudová modulace (AM) s potlačenou nosnou	79
4.4.8.2	Kmitočtová modulace (FM).....	80
4.4.8.3	Pulzní šířková modulace (PWM).....	82
4.4.8.4	Pulzní polohová modulace (PPM)	83
4.4.9	Demodulace modulovaných signálů	84
4.4.9.1	Demodulace signálu AM s přenášenou nosnou	84
4.4.9.2	Demodulace signálu FM	85
4.4.9.3	Demodulace signálu PWM	86
4.4.10	Generování náhodných signálů	86
4.4.10.1	Stručné shrnutí teorie	86
4.4.10.2	Diskrétní náhodné veličiny	87
4.4.10.3	Spojité náhodné veličiny	89
4.4.10.4	Generování signálů s rovnoměrným rozdělením hustoty pravděpodobnosti..	90
4.4.10.5	Generování signálů s normálním rozdělením hustoty pravděpodobnosti	93
4.4.10.6	Výpočet střední hodnoty a směrodatné odchylky	95
4.4.10.7	Výpočet mediánu	95
4.4.11	Generování zašuměných signálů	96
4.4.11.1	Stručné shrnutí teoretických základů	97
4.4.11.2	Generování harmonického signálu, zašuměného aditivním šumem	98
4.4.12	Generování harmonického signálu s definovaných harmonickým zkreslením	100
4.4.13	Práce v interaktivním prostředí SPTOOL	103
4.4.13.1	Prohlížení signálů, jejich export a import.....	104

4.4.13.2	<i>Práce s filtry</i>	106
4.4.13.3	<i>Spektrální analýza signálů</i>	107
4.5	Získávání dat z externích zdrojů	108
4.5.1	Přenos dat s podporou rozhraní RS-232	108
4.5.2	Přenos dat s podporou protokolu FTP	116
4.5.3	Přenos dat pomocí systému DDE v prostředí operačního systému MS Windows	120
4.6	Použití uživatelských funkcí jako zdroje dat	123
4.6.1	Příklady m-souborů s využitím příkazů systému MATLAB	124
4.6.2	Příklady m-souborů pro definici vlastních funkčních závislostí	127
5	Analýza signálů	129
5.1	Rozdělení metod analýzy	132
5.2	Analýza signálů v originální (časové) oblasti	133
5.2.1	Základní charakteristiky signálů	133
5.2.2	Globální a další charakteristiky signálů	139
5.2.2.1	<i>Mohutnost impulsu</i>	140
5.2.2.2	<i>Střední hodnota</i>	141
5.2.2.3	<i>Směrodatná odchylka</i>	143
5.2.2.4	<i>Medián</i>	144
5.2.2.5	<i>Okamžitý výkon impulsu</i>	144
5.2.2.6	<i>Činný výkon (střední výkon, výkon impulsu)</i>	145
5.2.2.7	<i>Efektivní hodnota</i>	146
5.2.2.8	<i>Energie impulsu</i>	147
5.2.2.9	<i>Vzájemná energie dvou impulsů</i>	147
5.2.2.10	<i>Derivace</i>	149
5.2.2.11	<i>Integrace</i>	151
5.2.3	Výpočet konvoluce	153
5.2.3.1	<i>Lineární diskrétní konvoluce</i>	153
5.2.3.2	<i>Kruhová (cyklická) diskrétní konvoluce</i>	158
5.2.3.3	<i>Využití algoritmu FFT pro výpočet kruhové konvoluce – rychlá konvoluce</i>	161
5.2.3.4	<i>Výpočet lineární konvoluce pomocí algoritmu FFT</i>	161
5.3	Spektrální analýza	163
5.3.1	Spektrální analýza analogových signálů	166
5.3.1.1	<i>Spektrum harmonického signálu</i>	167
5.3.1.2	<i>Spektrum neharmonických periodických signálů</i>	175
5.3.1.3	<i>Spektrum aperiodických signálů</i>	186
5.3.2	Spektrální analýza deterministických číslicových signálů	196
5.3.2.1	<i>Spektrum vzorkovaného signálu, vzorkovací poučka</i>	196
5.3.2.2	<i>Fourierova transformace diskrétních signálů</i>	200
5.3.2.3	<i>Spektrální analýza periodických signálů</i>	206
5.3.2.4	<i>Spektrální analýza aperiodických signálů</i>	207
5.3.2.5	<i>Důsledky aplikace algoritmu DFT</i>	208
5.3.3	Spektrální analýza stochastických číslicových signálů	225
5.3.3.1	<i>Odhad výkonových spekter jednotlivých procesů</i>	226
5.3.3.2	<i>Vzájemná spektra dvojice procesů</i>	229
5.4	Časově frekvenční analýza	230

6 Závěr	233
8 Literatura a odkazy	234
9 Přehled použitých symbolů a zkratk	236

3.1.2 Vzorkování a diskrétní čas

Jak jsem již předeslal výše, na vodorovné ose grafů číslicových signálů mohou být různé veličiny. Jedna veličina se však vyskytuje obzvláště často a tou je *čas*. Řada číslicových signálů je totiž získávána měřením či výpočtem časových průběhů - záznamů signálů a to analogových nebo přímo číslicových. Při měření teploty každou hodinu během dne si budete zapisovat do tabulky dvě řady čísel - teplotu a příslušný čas. Takový signál je již číslicový a hodnoty času na vodorovné ose jsou již tímto měřením dány. Jiná situace nastane, když bude měřen signál analogový a do číslicové podoby bude převeden pomocí převodníků A/D, často nazývaných obvod ADC (Analog to Digital Converter - převodník z analogové do digitální podoby). Na výstupu obvodu ADC má tedy signál již číslicovou podobu. Čísla, obsažená v tomto signálu jistě souvisejí s původním analogovým signálem. Ilustrační příklad vzniku číslicového signálu z analogového ukazuje obr.3.4 (signál není kvantován).



Obr.3.4. Převod analogového signálu do číslicové podoby.

Obrázek vlevo ukazuje příkazy, použité k vytvoření pravého grafického průběhu, který byl poté přímo v menu obrázku dotvořen (název, popisky apod.). Z časového průběhu je vidět, že pokud převedete analogový signál na číslicový, nabude podobu prosté posloupnosti čísel. Vodorovná osa je vytvořena a veličina na ní má význam času. Při pohledu na oba signály, analogový i číslicový, je patrné, že *číslíkový signál je tvořen vzorky analogového signálu v určitých, pravidelně se opakujících časových intervalech, daných vektorem časové osy*. Na obr.3.5 je číslicový signál z obr.3.4 ukázán v podobě posloupnosti. V horním řádku jsou čísla, odpovídající svislé ose (funkční hodnoty signálu), druhý řádek představuje vodorovnou osu grafu (čas). Z obr.3.4 plyne, že tzv. „analogový“ signál je ve skutečnosti také tvořen konečnou řadou čísel (vektor u_a), avšak počet čísel je velký a proto se tento signál tváří jako analogový. Tato „finta“ je jedinou možností, jak pracovat s „analogovými“ signály v MATLAB.

```

Command Window

>> [ud;td]

ans =

    0    0.5878    0.9511    0.9511    0.5878    0.0000   -0.5878   -0.9511   -0.9511   -0.5878
    0    0.6283    1.2566    1.8850    2.5133    3.1416    3.7699    4.3982    5.0265    5.6549

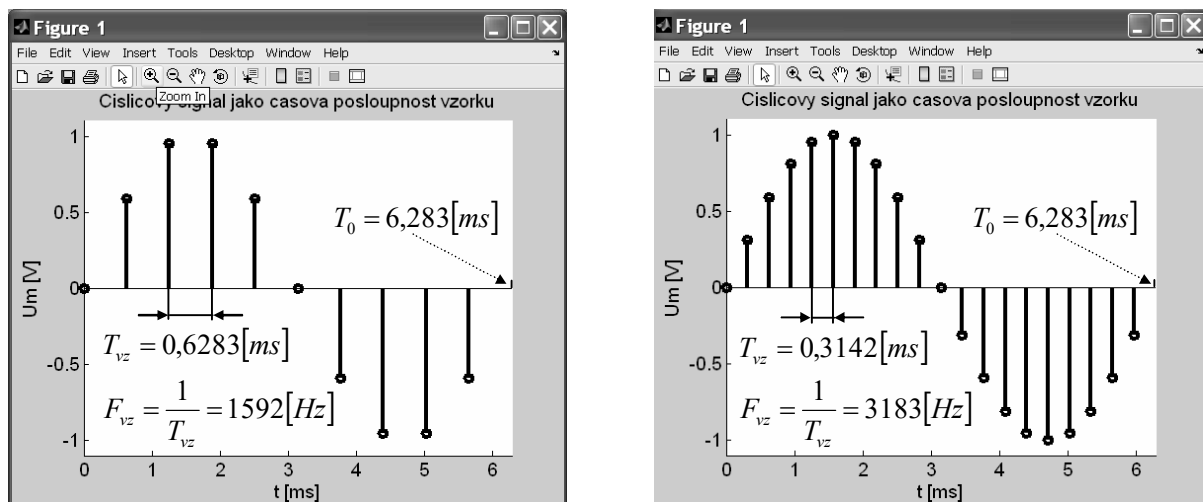
>>

```

Obr.3.5. Číslíkový signál z obr.3.4 v podobě posloupnosti spolu s časovou osou.

3.1.2.1 Vzorkování

Na základě předchozího příkladu můžeme napsat, že pokud číslíkový signál vznikl z analogového, pak představuje vzorky původního analogového prototypu. Z hlediska vodorovné osy, času, jde tedy o vzorky, vniklé procesem vzorkování. *Vzorkování je tedy procesem, kterým z analogového signálu získáme jeho vzorky na časové ose.* Tyto vzorky jsou definovány jen v určitých časových okamžicích a v běžných případech (ale ne ve všech) jsou podél časové osy rozprostřeny pravidelně. O periodických signálech píší v kapitole 2.3.2. Odtud víme, že jedním z parametrů těchto signálů je opakovací perioda T_0 či opakovací kmitočet F_0 . U číslíkových signálů, vzniklých vzorkováním analogových signálů, je jedním z nejdůležitějších parametrů vzorkovací perioda T_{vz} nebo vzorkovací kmitočet F_{vz} . *Vzorkovací perioda* je doba mezi jednotlivými vzorky, tedy vzdálenost mezi nimi v sekundách [s]. Převrácená hodnota vzorkovací periody je *vzorkovacím kmitočtem* v jednotkách Hertz (Hz = s⁻¹). Tyto základní pojmy jsou ilustrovány na obr.3.6.



Obr.3.6. K pojmům vzorkovací perioda a vzorkovací kmitočet.

Symboły T_{vz} a F_{vz} na obr.3.6 označují vzorkovací periodu a vzorkovací kmitočet číslíkových vzorků. Jde tedy o parametry, vztahující se k vlastním vzorkům. Symbolem T_0 je v grafech označena perioda původního analogového periodického signálu, tedy spojitě obálky. Pokud znáte vzorkovací periodu, tedy vzdálenost vzorků a počet vzorků v rámci periody původního analogového signálu, snadno vypočtete jeho původní periodu T_0 . A naopak, je-li známa původní perioda T_0 , pak na základě znalosti T_{vz} nebo F_{vz} lze vypočítat počet vzorků v rámci jedné periody původního analogového signálu.

Jinými slovy, namísto toho, abychom definovali tři parametry, tedy T_{vz} , počet vzorků a T_0 , lze definovat pouze dva a ten třetí dopočítat, bude-li to potřeba. Při práci se signály se velmi často používá poměr $\frac{T}{T_{vz}} = \frac{F_{vz}}{F_0} [-]$, symbolem F_0 označujeme opakovací kmitočet analogového periodického signálu a platí $F_0 = \frac{1}{T_0} [Hz]$. V teorii jsou definovány a užívány také normované veličiny, kdy platí: $\bar{F} = \frac{F_0}{F_{vz}} [-]$ a $\bar{T} = \frac{T_0}{T_{vz}} [-]$. V případě obr.3.6 lze u prvního grafu vypočítat: $\frac{T_0}{T_{vz}} = \frac{6,283 \cdot 10^{-3}}{0,6283 \cdot 10^{-3}} = \frac{6,283}{0,6283} = 10 [-]$, u druhého grafu $\frac{T_0}{T_{vz}} = \frac{6,283 \cdot 10^{-3}}{0,3142 \cdot 10^{-3}} = \frac{6,283}{0,3142} = 20 [-]$. Čísla 10 a 20 jsou tedy počty vzorků v rámci jedné periody původního analogového signálu (obálky). Výhodou je, že znáte-li uvedené poměry $\frac{T_0}{T_{vz}} [-]$, pak při zadání T_{vz} lze ihned dopočítat T_0 a naopak. Ukažme si to v následujících příkladech.

Příklad 3.1.

Periodický analogový signál byl převodníkem ADC převeden do číslicové podoby a načten do souboru na pevném disku počítače. Bylo zjištěno, že počet vzorků v rámci jedné periody analogového signálu byl 100. Jaký byl použitý vzorkovací kmitočet a vzorkovací perioda, byl-li kmitočet analogového signálu $F_0 = 400 Hz$.

$$\text{Řešení: } \frac{F_{vz}}{F_0} = 100 [-] \Rightarrow F_{vz} = 100 \cdot F_0, F_{vz} = 100 \cdot 400, \underline{F_{vz} = 40 kHz}$$

$$T_{vz} = \frac{1}{F_{vz}}, T_{vz} = \frac{1}{40 \cdot 10^3}, T_{vz} = 25 \cdot 10^{-6}, \underline{T_{vz} = 25 \mu s}$$

Poznámka: vzorkovací kmitočet je dán použitým převodníkem ADC. Někdy má uživatel možnost jej měnit, jindy je pevně vázán na daný typ převodníku či zařízení.

Příklad 3.2.

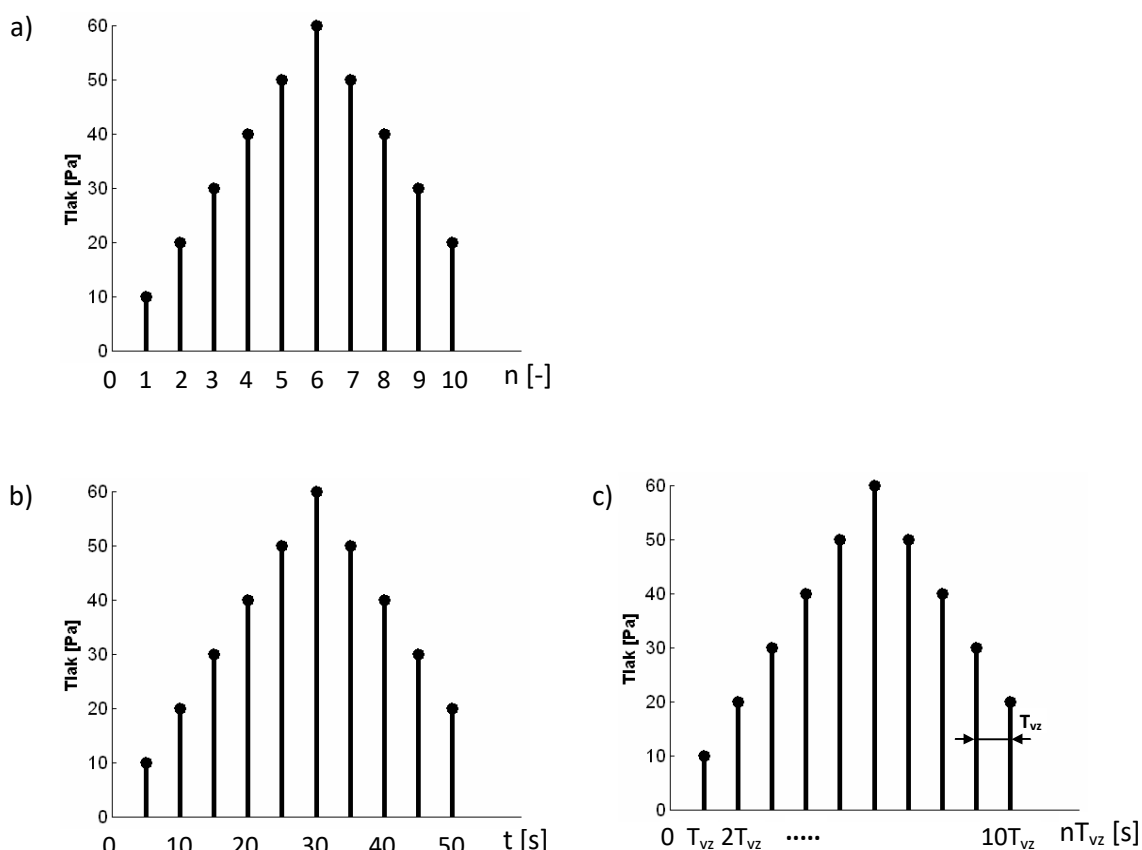
U signálu z příkladu 3.1 potřebujeme zajistit, aby v rámci jedné periody analogového signálu byl dvojnásobně vyšší počet vzorků. Jaký vzorkovací kmitočet je potřebné zvolit?

$$\text{Řešení: } \frac{F_{vz}}{F_0} = 200 [-] \Rightarrow F_{vz} = 200 \cdot F_0, F_{vz} = 200 \cdot 400, \underline{F_{vz} = 80 kHz}$$

Poučení: je jistě zřejmé, že zvýšení vzorkovacího kmitočtu na dvojnásobek způsobí dvojnásobný nárůst počtu vzorků v rámci jedné periody analogového signálu.

3.1.2.2 Diskrétní čas

Podívejme se ještě jednou na obr.3.2 a obr.3.3. Na vodorovné ose 2D grafu mohou být tři veličiny. Jde o prosté *pořadí vzorků* n , spojitý čas t (continuous-time) a tzv. *diskrétní čas* nT_{vz} (discrete-time), viz obr.3.7. Rozdíl mezi spojitým a diskrétním časem je zřejmý. Spojitý čas je spojitou veličinou, je tedy definován v každém bodě sledovaného časového intervalu. Diskrétní čas je definován pouze v okamžicích vzorků.

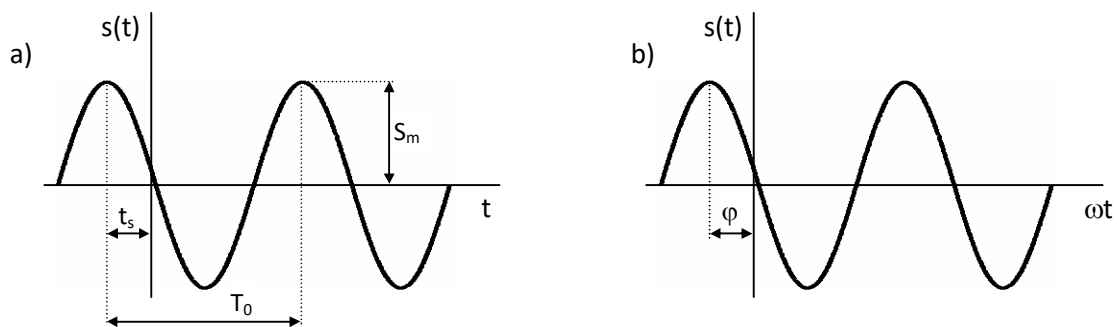


Obr.3.7. Různé veličiny na vodorovné ose grafu u vzorkovaných signálů,
a) pořadí vzorků, b) spojitý čas, c) diskrétní čas.

Je zřejmé, že pokud bude potřeba mít na vodorovné ose čas, bude třeba informaci o něm někde získat. Např. u souborů zvukového formátu **.wav*, používaném v operačním systému Windows, je kromě vlastních zvukových vzorků přítomna informace o vzorkovací kmitočtu F_{vz} . Bez toho by nebylo zřejmé, s jakou četností byla data získána, a nebylo by možné je stejnou rychlostí přehrát, viz např. příklad 5.5 v kapitole 5.3.2.5. Čísla bez dalších informací jsou pouhými čísly.

4.4.1 Generování harmonických signálů

Harmonickým signálem označujeme signál, jehož časový průběh lze popsat (modelovat) funkcemi *sinus* nebo *kosinus*, viz kapitoly 2.2.2, 2.3.2 a 2.3.3. Harmonický signál je zcela definován třemi parametry, kterými jsou amplituda S_m , perioda T_0 a počáteční fáze φ , viz obr.4.38.



$$s(t) = S_m \cdot \cos(\omega \cdot t + \varphi), \quad \omega = \frac{2 \cdot \pi}{T_0}$$

Obr.4.38. Časový průběh harmonického signálu a jeho parametry,

a) na vodorovné ose je čas t ,

b) na vodorovné ose je ωt (S_m a T_0 beze změny).

4.4.1.1 Práce s příkazy „sin“ a „cos“

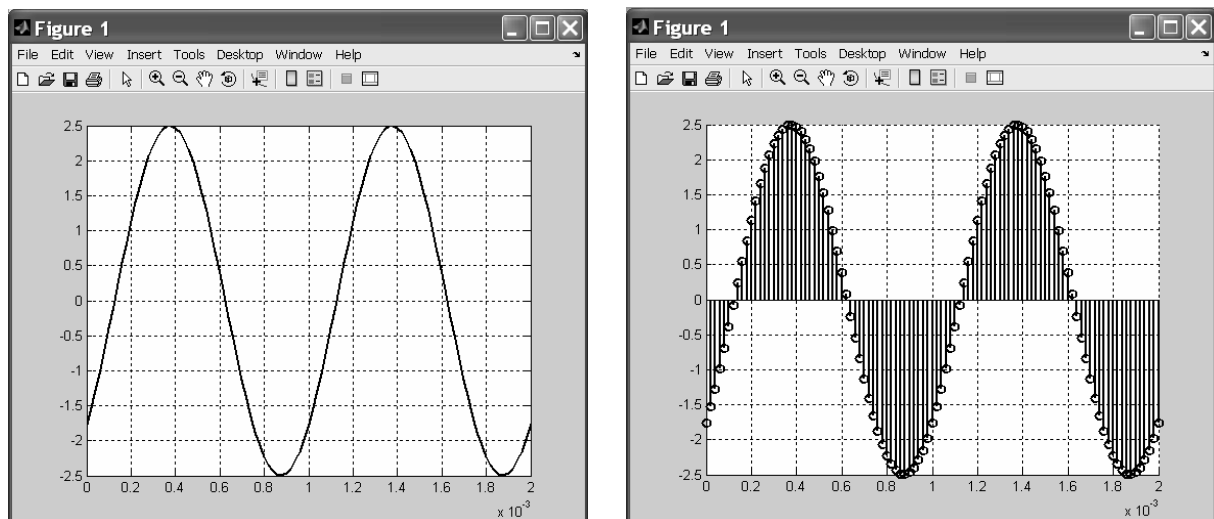
Vytvořit „obyčejný“ harmonický signál lze velmi jednoduše použitím funkcí jádra MATLAB *sin* a *cos*. Připomeňme, že tyto funkce patří mezi tzv. elementární matematické funkce, jejichž přehled lze obdržet např. pomocí *help elfun* a jejichž syntaxe použití je velmi jednoduchá, viz *help sin* a *help cos*. Funkce *sin* a *cos* mohou mít jako vstupní parametr buďto jedno číslo (skalár) nebo vektor čísel a dokonce matici. Kolik čísel vložíme jako vstupní parametr, tolik jich obdržíme zpět. Na obr.4.39 jsou ukázány dva příklady tvorby signálu harmonického průběhu. Při použití příkazu *stem* je zdůrazněn diskretní charakter signálu. Oba harmonické signály mají stejné parametry, tedy amplitudu 2.5V, kmitočet 1000 Hz a počáteční fázi $\pi/4$ [rad]. Jsou vykresleny dvě periody.

Command Window

```
>> t1=0:0.002/100:0.002;
>> u1=2.5*sin(2*pi*1000*t1-pi/4);
>> plot(t1,u1)
>> grid on
```

Command Window

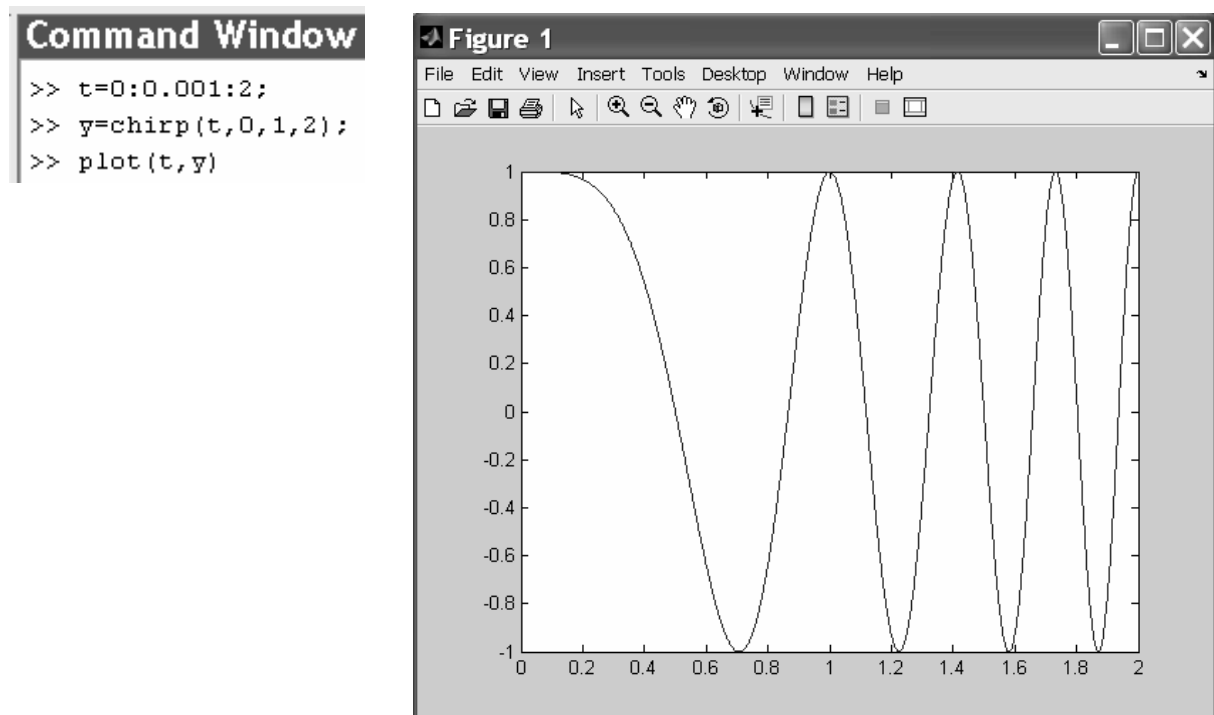
```
>> t1=0:0.002/100:0.002;
>> u1=2.5*sin(2*pi*1000*t1-pi/4);
>> stem(t1,u1)
>> grid on
```

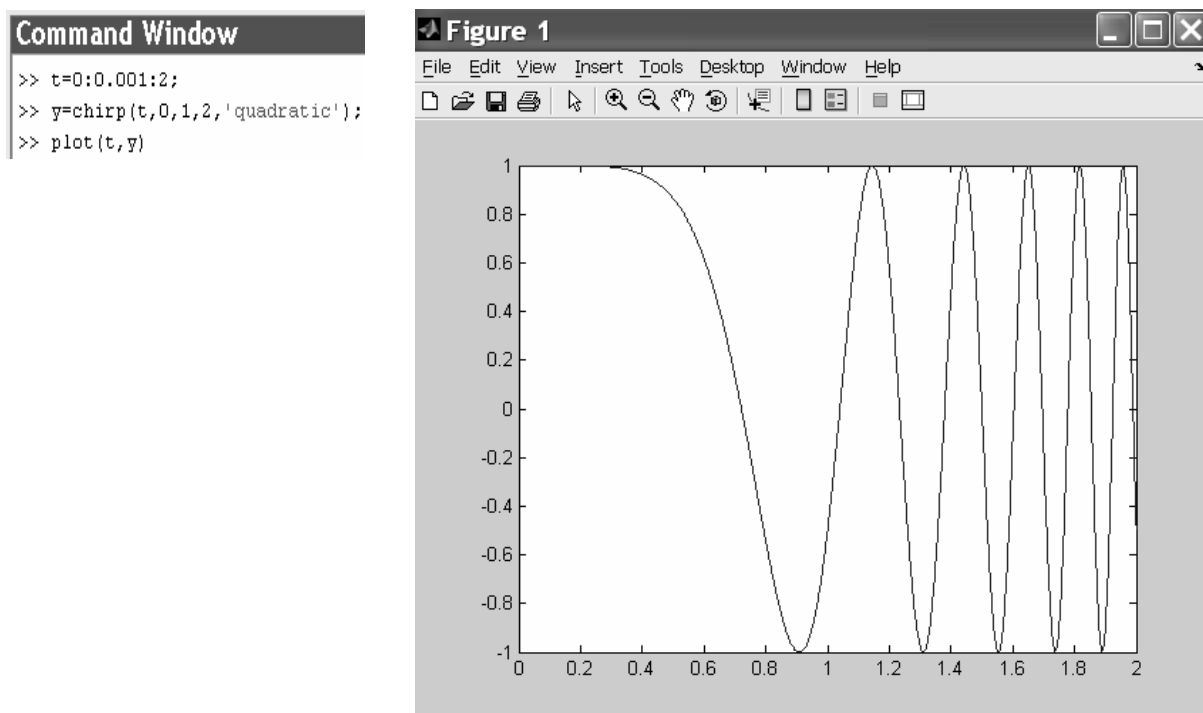


Obr.4.39. Generování harmonických signálů.

4.4.1.2 Harmonický signál s proměnným kmitočtem

Kmitočet harmonického signálu se bude měnit v mezích a metodou, stanovenými uživatelem. Změna kmitočtu může být lineární, kvadratická nebo logaritmická, viz obr.4.40. Klíčovým příkazem je příkaz *chirp*. V prvním příkladu na obr.4.40 jsou jako jeho parametry zadány vektor časové osy t , počáteční kmitočet 0 Hz , cílový čas 1 s , při němž bude kmitočet 2 Hz . Nebudou-li definovány žádné další parametry, bude změna kmitočtu od 0 Hz do 2 Hz lineární. Ve druhém příkladu je díky parametru '*quadratic*' změna kmitočtů v uvedených mezích kvadratická. Při volbě '*logarithmic*' půjde o logaritmickou změnu.



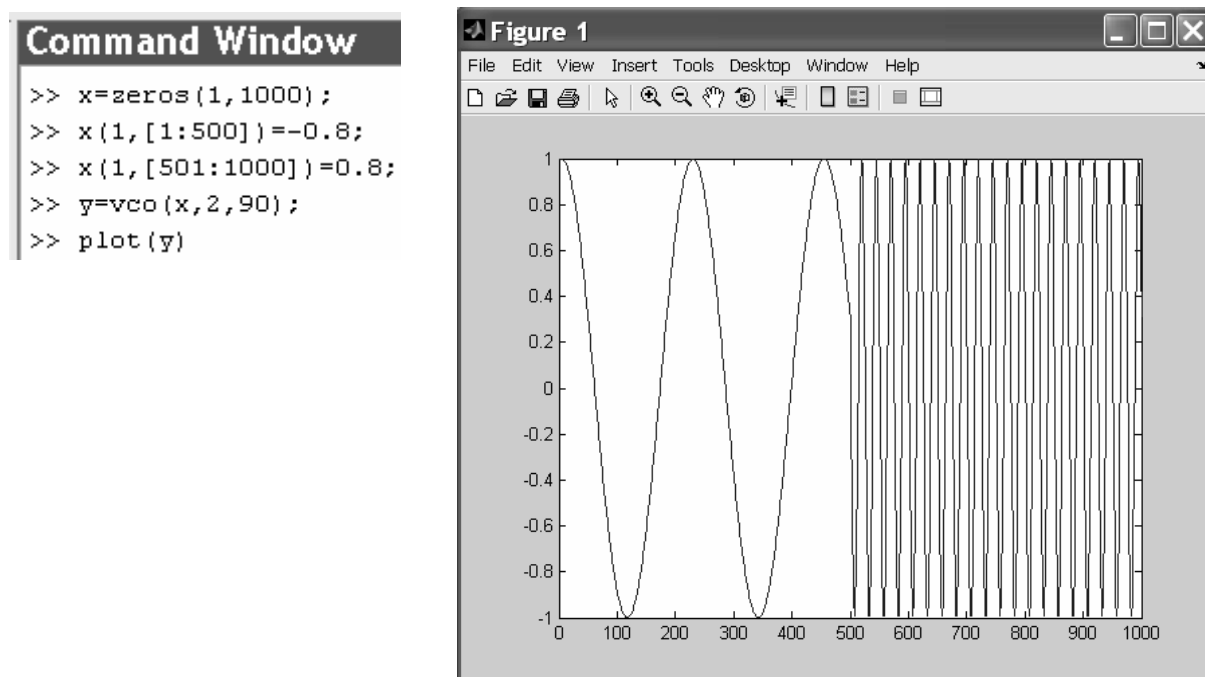


Obr.4.40. Generování harmonických signálů s proměnným kmitočtem příkazem *chirp*.

4.4.1.3 Napěťově řízený oscilátor (vco)

Podstatou je generování harmonického signálu, jehož okamžitý kmitočet závisí na okamžité hodnotě jistého vektoru. Termíny „napěťově řízený oscilátor“ (*voltage controlled oscillator - vco*) jsou zvoleny z praxe, kde oscilátor (generátor harmonického signálu) je často řízen veličinou, jejímž fyzikálním významem je elektrické napětí. Jde však o již tak všeobecně vžitý a uznávaný název (termín), že se běžně používá. Z hlediska MATLAB by se také dalo napsat, že jde o oscilátor s kmitočtem řízeným hodnotou.

Klíčovým příkazem je příkaz *vco*. Mezi jeho hlavní parametry patří vektor hodnot, které řídí kmitočet harmonického signálu, označme jej x , referenční (nosný) kmitočet f_c a vzorkovací kmitočet f_s . Pro vysvětlení principu si představte, že vidíte dva průběhy dat. Jeden náleží vektoru řídicích hodnot x a druhý výstupu příkazu *vco*. Oba průběhy plynou současně. Podle okamžité hodnoty vektoru x je nastaven kmitočet výstupu. Změní-li se hodnota vektoru x , okamžitě se změní kmitočet výstupního harmonického signálu. N obr.4.41 je ukázán princip řízení.



Obr.4.41. Základní princip použití příkazu *vco*.

Je definován vektor x ze samých nul o délce 1000 prvků. Poté je polovině prvků vektoru přiřazena hodnota -0.8, druhé polovině hodnota 0.8. Příkaz *vco* obsahuje jako své parametry vektor x , dále kmitočet referenčního signálu 2 Hz a kmitočet vzorkovacího signálu 90 Hz. Na obrázku, získaném příkazem *plot*, je vidět vlastní řízení kmitočtu. Do poloviny je kmitočet řízen číslem -0.8, od poloviny číslem 0.8. Jde tedy o dva rozdílné kmitočty. Jaká je jejich hodnota? Rozsah použitelných hodnot vektoru x je $x \in \langle -1, 1 \rangle$. Pro hodnotu -1 je kmitočet na výstupu roven 0 Hz, pro hodnotu 0 je kmitočet roven kmitočtu referenčního signálu (tedy 2 Hz) a pro hodnotu 1 je kmitočet roven dvojnásobku kmitočtu referenčního signálu. Je zřejmé, že mezi hodnotami -1 a 1 budou kmitočty úměrné konkrétní hodnotě. Je např. možné definovat řídicí vektor x jako harmonický signál. Potom se bude kmitočet na výstupu měnit harmonicky. Příkazu *vco* se využívá např. při tvorbě některých typů modulovaných signálů. Pro další možnosti viz help.

5.2.2 Globální a další charakteristiky signálů

Globální charakteristiky charakterizují číslicový signál podrobněji a poskytují další cenné informace o něm. Jejich vlastností je, že k jejich výpočtu je třeba zpravidla *pracovat se všemi vzorky* a navíc některé z nich mají *integrální charakter*, tedy určují vlastnosti signálu jako celku za určitý časový interval, resp. interval veličiny nezávisle proměnné. Zaujímají nás zejména tyto charakteristiky:

- mohutnost,
- střední hodnota,
- směrodatná odchylka,
- medián,
- okamžitý výkon,
- činný výkon (střední výkon, výkon signálu),

- efektivní hodnota,
- energie signálu,
- vzájemná energie dvou signálů,
- derivace,
- integrál.

Následující části kapitoly uvádějí vždy stručně teoretické zázemí a na příkladech výpočty jednotlivých globálních a dalších charakteristik.

Poznámka: protože chápeme a priori všechny číslicové signály jako aperiodické, budeme je v dalších částech analýzy v časové oblasti nazývat *impulzy*.

5.2.2.1 Mohutnost impulzu

V případě analogových signálů je mohutnost impulzu dána plochou, kterou vymezuje jeho grafický průběh v rámci doby trvání. Matematicky lze plochu spojitě funkce, jíž je daný signál modelován, vypočítat určitým integrálem. V případě číslicových signálů je integrál nahrazen sumou (součtem). Pak můžeme napsat, že *mohutnost je dána součtem hodnot všech vzorků impulzu*, viz vztah (5.1).

$$M = \sum_{k=-\infty}^{\infty} s(k). \quad (5.1)$$

Jednotkou mohutnosti je jednotka signálu, je-li k k dispozici. Teoretické meze sumace $k \in \langle -\infty, \infty \rangle$ se v případě praktických signálů mění na konkrétní počet vzorků. Na obr.5.5 je ilustrován výpočet mohutnosti, který je jistě velmi snadný. Číslicová data jsou získána načtením souboru *recycle.wav* ze složky *Windows/Media*. Jde o soubor zvukových dat, který slýcháte při vysypání koše v operačním systému Windows XP. Po načtení příkazem *wavread* je zjištěn rozměr proměnné data, která obsahuje vlastní vzorky zvukových dat a velikost vzorkovacího kmitočtu. Data jsou nahrána pouze monofonně, tedy jednokanálově. Pomocí *sum* je pak získán součet hodnot všech vzorků, tedy vlastní mohutnost M . Připomeňme, že získání součtu hodnot vzorků pomocí cyklu *for* v m-souboru je sice alternativou, avšak pro velkou časovou náročnost tohoto přístupu jej příliš nedoporučujeme.

```
Command Window
>> [data, fvz]=wavread('c:\Windows\Media\recycle');
>> size(data)

ans =

    12671     1

>> fvz

fvz =

    22050

>> M=sum(data)

M =

    0.2337
```

Obr.5.5. Výpočet mohutnosti M číslicového aperiodického signálu.

5.2.2.2 Střední hodnota

Střední hodnota je v teorii analogových signálů definována jako průměrná hodnota signálu v rámci jeho jedné periody. Vztahuje se tedy *historicky k periodickým signálům*. My zde budeme tento termín chápat ve vztahu k signálům číslicovým a to stochastickým. Střední hodnota tedy bude jednou z charakteristik náhodné veličiny, viz kapitolu 4.4.10. Zde je konstatováno, že střední hodnota je teoreticky definována jako číslo, kolem kterého kolísají hodnoty výběrových průměrů, jež se počítají vždy ze série hodnot náhodné veličiny (mnoho realizací náhodného procesu). Tento fakt je vyjádřen rovnicí (4.1). Bude-li k dispozici číslicový signál pouze v jediné realizaci a navíc všechny funkční hodnoty náhodné veličiny (vzorky) budou mít stejnou pravděpodobnost, pak se celý problém redukuje do výpočtu běžného aritmetického průměru, který je vyjádřen vztahem (4.6), viz obr.5.6. Toto tvrzení by mělo být správně podloženo vědomím, že střední hodnota by měla být nezávislá na konkrétní realizaci, tedy že jde o ergodický náhodný proces. Pokud to není známo či zaručeno, vyjde nám výpočtem pouze více či méně přesný *odhad střední hodnoty* konkrétního signálu. Jednotkou střední hodnoty je jednotka signálu.

```
Command Window
>> Vek=rand(1,10)

Vek =

    0.9501    0.2311    0.6068    0.4860    0.8913    0.7621    0.4565    0.0185    0.8214    0.4447

>> E=mean(Vek)

E =

    0.5669
```

Obr.5.6a. Výpočet střední hodnoty číslicového aperiodického signálu ve formě vektoru.

```

Command Window
>> Mat=rand(5,10)

Mat =

    0.6154    0.4057    0.0579    0.2028    0.0153    0.4186    0.8381    0.5028    0.1934    0.6979
    0.7919    0.9355    0.3529    0.1987    0.7468    0.8462    0.0196    0.7095    0.6822    0.3784
    0.9218    0.9169    0.8132    0.6038    0.4451    0.5252    0.6813    0.4289    0.3028    0.8600
    0.7382    0.4103    0.0099    0.2722    0.9318    0.2026    0.3795    0.3046    0.5417    0.8537
    0.1763    0.8936    0.1389    0.1988    0.4660    0.6721    0.8318    0.1897    0.1509    0.5936

>> E=mean(Mat)

E =

    0.6487    0.7124    0.2745    0.2953    0.5210    0.5330    0.5501    0.4271    0.3742    0.6767

```

Obr.5.6b. Výpočet střední hodnoty číslcového aperiodického signálu ve formě matice; je počítána střední hodnota jednotlivých sloupců matice.

```

Command Window
>> E=mean(Mat,5)

E =

    0.4966    0.6602    0.7271    0.7027    0.7948    0.9797    0.1365    0.6614    0.5828    0.2259
    0.8998    0.3420    0.3093    0.5466    0.9568    0.2714    0.0118    0.2844    0.4235    0.5798
    0.8216    0.2897    0.8385    0.4449    0.5226    0.2523    0.8939    0.4692    0.5155    0.7604
    0.6449    0.3412    0.5681    0.6946    0.8801    0.8757    0.1991    0.0648    0.3340    0.5298
    0.8180    0.5341    0.3704    0.6213    0.1730    0.7373    0.2987    0.9883    0.4329    0.6405

>> E=mean(Mat,2)

E =

    0.5968
    0.4625
    0.5809
    0.5132
    0.5615

```

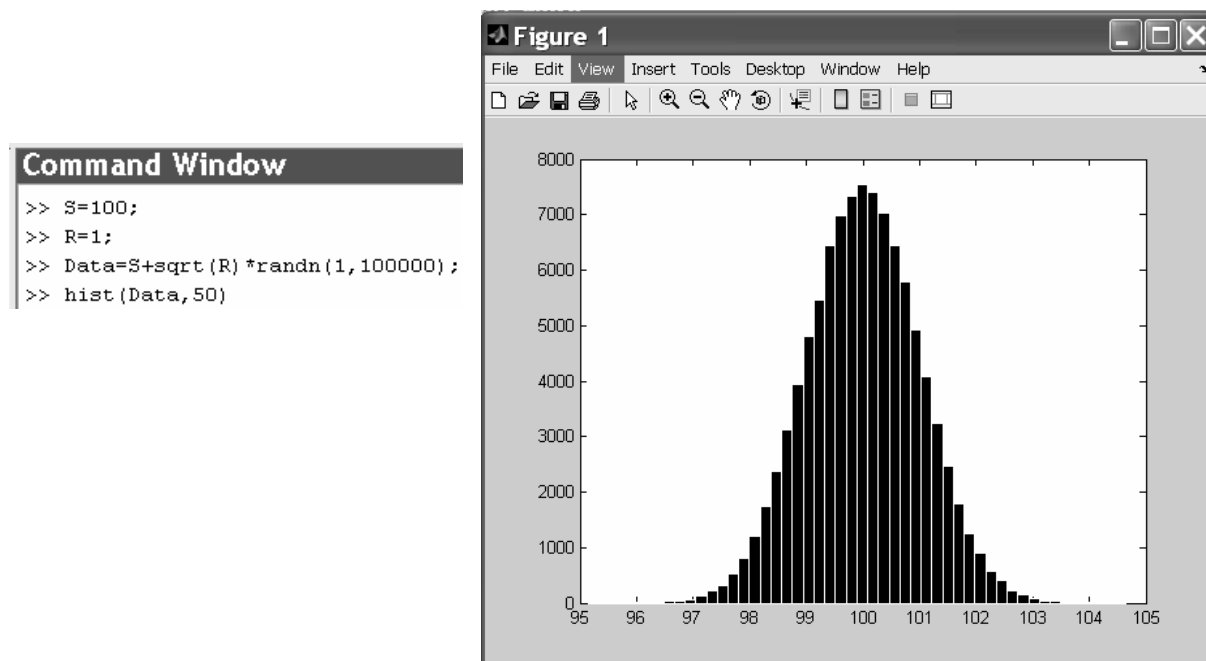
Obr.5.6c. Výpočet střední hodnoty číslcového aperiodického signálu ve formě matice; je počítána střední hodnota jednotlivých řádků matice.

Je-li příkaz *mean* aplikován na matici, pak vrátí vektor čísel, jejichž významem je střední hodnota jednotlivých sloupců, resp. řádků.

Jak jsem ukázal v kapitole 4.4.10, jsou pojmy jako střední hodnota, rozptyl apod. číselnými charakteristikami náhodné veličiny a zastupují tak mnohdy velmi pracné až nemožné získání rozložení hustoty pravděpodobnosti náhodné veličiny. Jinou možností je výpočet a kreslení histogramů, ze kterých je patrná i střední hodnota, nejlépe je-li histogram dostatečně „hustý“, tedy jsou-li intervaly datových hodnot na vodorovné ose dostatečně malé, viz obr.5.7 (názorné jen u některých rozdělení). Pak histogram již svým tvarem připomíná konkrétní rozložení, v našem případě *normální*. Konstanty *S* a *R* mají význam střední hodnoty a rozptylu. Není-li k dispozici jiná možnost, lze tedy střední hodnotu odhadovat i takto. Střední hodnota v histogramu signálu s Gaussovským rozdělením hustoty

pravděpodobnosti reprezentuje opravdu střed, tedy číslo 100 na vodorovné ose (největší pravděpodobnost výskytu).

Terminologicky je střední hodnota někdy označována jako *stejnoseměrná hodnota* či *stejnoseměrná složka signálu*. Toto označení se používá např. v elektrotechnice z důvodů vyzdvížení polarity posunutí jinak v čase proměnného signálu. Jindy termínem pro stejnosměrný signál rozumíme signál konstantní, tedy v čase neproměnný.



Obr.5.7. Přibližný odhad střední hodnoty pomocí histogramu.

5.2.2.3 Směrodatná odchylka

Směrodatná odchylka (anglicky *standard deviation*) je matematicky odmocninou z rozptylu náhodné veličiny. Je definována vztahem (4.3), resp. (4.2). V MATLAB je dosažitelná díky funkci *std*, viz obr.5.8.

```
Command Window  
  
>> S=100;  
>> R=1;  
>> Data=S+sqrt(R)*randn(1,100000);  
>> std(Data)  
  
ans =  
  
1.0013
```

Obr.5.8. Výpočet směrodatné odchylky pomocí příkazu *std*.

Platí podobné úvahy o ergodicitě, jako v případě střední hodnoty. Aplikován na matici vrací příkaz *std* vektor hodnot, jejichž významem je směrodatná odchylka dat ve sloupcích matice. Příkaz může mít ještě dva další parametry, viz *help*. Jednotkou směrodatné odchylky je jednotka signálu.

5.2.2.4 Medián

V kapitole 4.4.10 jsem se věnoval také termínu *medián*. Odtud víte, že medián je 50 %-ním kvantilem náhodné veličiny. Aplikujme postup zde uvedený v obr.5.9. Opět platí výše zmíněné úvahy o ergodicitě.

```

Command Window
>> S=100;
>> R=1;
>> Data=S+sqrt(R)*randn(1,100000);
>> median(Data)

ans =

    100.0040

>>
>> Data=S+sqrt(R)*randn(3,10)

Data =

    100.7509    99.6754    99.2129    100.3416    101.4616    101.2314    100.8059    100.6769    100.5692    100.6647
    99.9053    100.3041    99.3066    99.1842    99.5242    100.0631    101.1863    99.0095    99.0418    99.4245
    97.9245    101.0753    100.4934    99.6951    100.4677    100.9520    99.4237    100.3745    100.3967    98.6014

>> median(Data)

ans =

    99.9053    100.3041    99.3066    99.6951    100.4677    100.9520    100.8059    100.3745    100.3967    99.4245

>>
>> median(Data,2)

ans =

    100.6708
    99.4744
    100.3856

```

Obr.5.9. Výpočet mediánu číslicového stochastického signálu.

Komentář k výsledkům by byl analogický výpočtu střední hodnoty (aplikace na vektor a matici).

5.2.2.5 Okamžitý výkon impulzu

V teorii analogových signálů je okamžitý výkon dán kvadrátem funkce, kterou signál modelujeme. Analogicky v případě číslicových signálů půjde tedy o výpočet nového signálu, jehož vzorky budou mít velikost, danou kvadrátem těch původních. Výsledkem výpočtu tedy není jeden parametr-skalár, ale opět řada vzorků stejného počtu, jako původní signál. Platí tedy vztah:

Ilustrační výpočet je na obr.5.10. Jednotkou okamžitého výkonu je kvadrát jednotky signálu. Upozorníme, že při výpočtu je užito operátoru $\wedge$, který zajistí výpočet kvadrátů hodnot jednotlivých vzorků. Bez použití tečky vrátí MATLAB chybové hlášení, neboť by šlo o maticové násobení, kteréžto předpokládá jistý rozměr obou matic, resp. vektorů.

V literatuře je někdy okamžitý výkon doplněn přívlastkem *normovaný*, což souvisí s jeho historickou definicí, kdy je u analogových signálů v elektrotechnice předpokládáno jeho působení do odporové zátěže s jednotkovým (normovaným) odporem.

```

Command Window
>> Vektor=rand(1,10)

Vektor =

    0.0579    0.3529    0.8132    0.0099    0.1389    0.2028    0.1987    0.6038    0.2722    0.1988

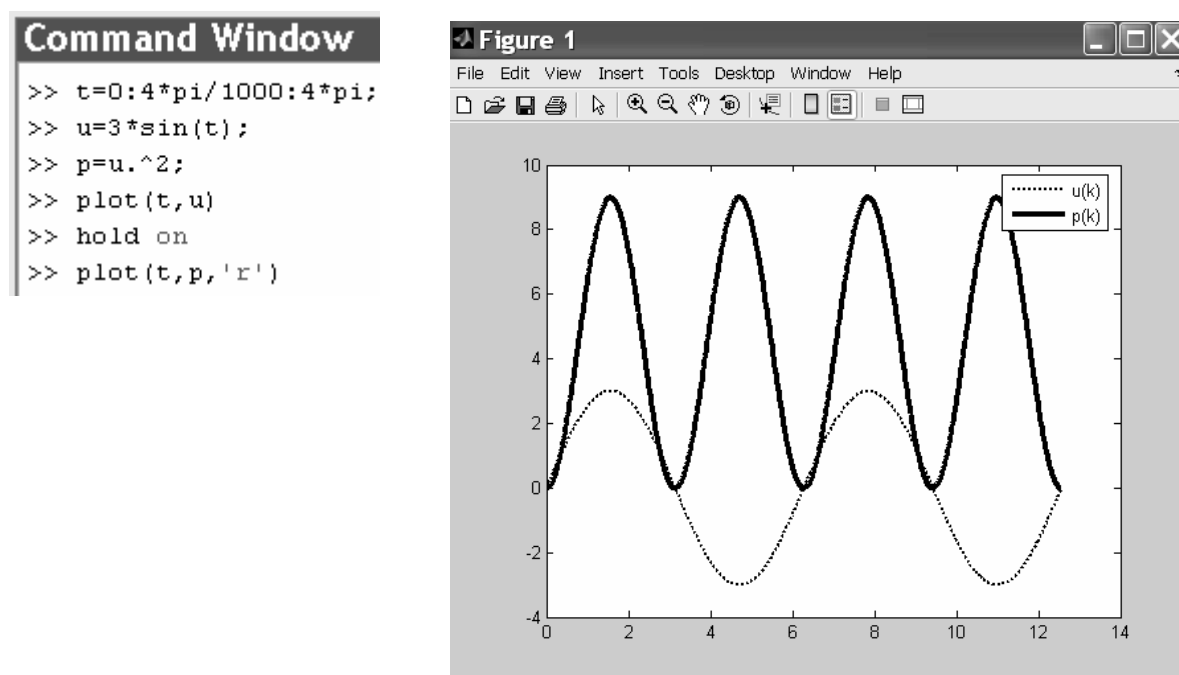
>> p=Vektor.^2

p =

    0.0034    0.1245    0.6612    0.0001    0.0193    0.0411    0.0395    0.3646    0.0741    0.0395

```

Obr.5.10a. Výpočet okamžitého výkonu číslicového stochastického signálu ve formě vektoru.



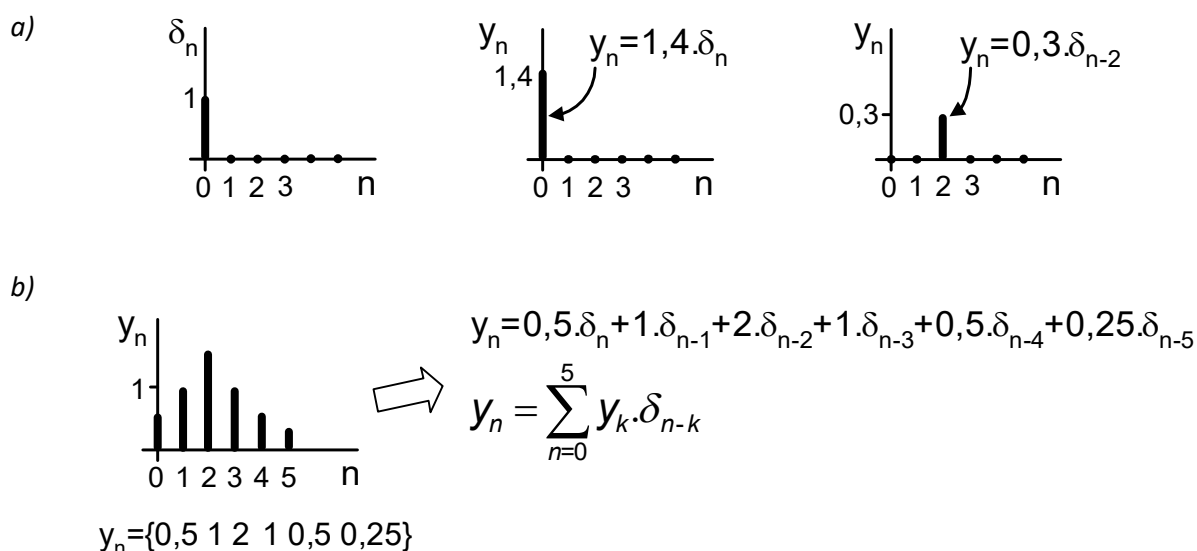
Obr.5.10b. Průběhy signálu a jeho okamžitého výkonu.

5.2.3 Výpočet konvoluce

Algoritmy pro výpočet *konvoluce* dvou číslicových signálů tvoří důležitou součást DSP. Konvoluce se využívá dosti často a to přímo anebo jako součást rozsáhlejších výpočtů. Z hlediska fyzikální podstaty a porozumění ji zavedeme v následující části knihy spíše intuitivně s minimem matematiky.

5.2.3.1 Lineární diskrétní konvoluce

Naším cílem je zavést pojem *lineární diskrétní konvoluce* spíše intuitivně. Abychom však toto mohli realizovat, nevyhneme se následujícímu popisu číslicového signálu. Číslicový signál je posloupností čísel - vzorků. V pořadí první číslo dané posloupnosti je možné vyjádřit samostatně pomocí diskrétního jednotkového impulzu, viz kapitola 4.4.3. Bude-li hodnota tohoto čísla různá od jedničky, pak bude muset být diskrétní jednotkový impulz násoben tímto číslem, neboť jeho hodnota je rovna 1. V pořadí druhé číslo posloupnosti, tedy o jeden krok zpožděné, lze opět vyjádřit diskrétním jednotkovým impulzem. Ten bude taktéž vynásoben hodnotou druhého čísla posloupnosti a navíc zpožděn (posunut) o jeden krok-pořadové číslo. Tento postup, tedy náhradu či popis každého čísla signálu diskrétním jednotkovým impulzem, s hodnotou vynásobenou tímto číslem a příslušně posunutým, provedeme myšlenkově pro každé číslo posloupnosti. Sečteme-li poté všechny vynásobené a posunuté diskrétní jednotkové impulzy, dostaneme původní posloupnost čísel. Uvedený postup dokumentuje obr.5.20.



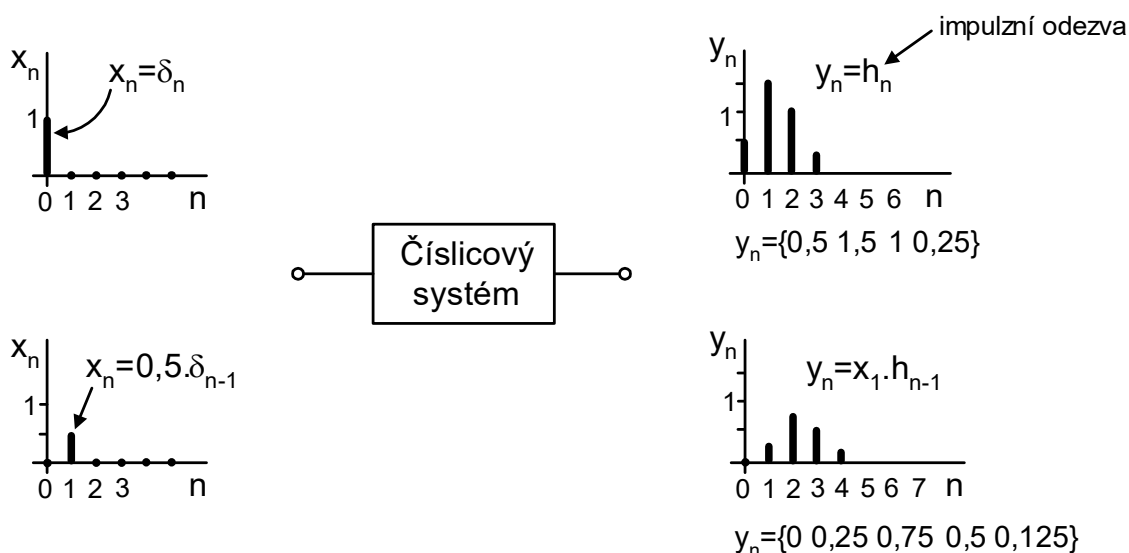
Obr.5.20. Číslicový signál jako suma vážených posunutých diskrétních jednotkových impulzů,
a) diskrétní jednotkový impulz δ_n a jeho využití k popisu jednotlivých vzorků (čísel),
b) konkrétní číslicový signál y_n , vyjádřený pomocí diskrétních jednotkových impulzů.

Násobení diskrétních jednotkových impulzů hodnotami čísel posloupnosti nazýváme často *vážením* (násobení *vahou-hodnotou* prvků posloupnosti). Vyjádříme-li tedy libovolný číslicový signál výše uvedeným postupem, říkáme, že daný signál byl vyjádřen součtem (sumou) vážených posunutých diskrétních jednotkových impulzů. Výše uvedený postup využijeme v následujících partiích.

Uvažujte nyní libovolný číslicový systém (soustavu). Tímto systémem je v praxi vždy nějaký algoritmus, tedy postup, při němž je ze vstupní posloupnosti čísel vytvořena posloupnost výstupní bez ohledu na konkrétní fyzickou realizaci (implementaci). Nechť je daný číslicový systém lineární, tedy hodnoty vzorků na výstupu jsou v přímé úměře k hodnotám vstupu (zjednodušeně). Na takový

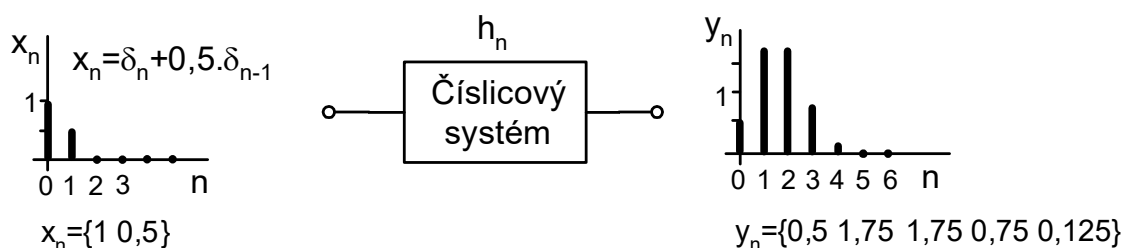
číslicový systém přicházejí postupně vzorky (čísla) vstupní posloupnosti x_n a na jeho výstupu se objevují vzorky odezvy y_n . Zamysleme se nad možností vypočítat vzorky výstupní posloupnosti, budou-li známy vzorky posloupnosti vstupní a příslušné parametry onoho číslicového systému. Jde tedy o výpočet odezvy lineárního číslicového systému (algoritmu) na vstupní budící posloupnost.

Důležitým termínem v této souvislosti je *impulzní odezva lineárního číslicového systému*. Impulzní odezva h_n je odezvou systému v případě, kdy na vstupu působí pouze jednotkový impulz δ_n . Je-li systém buzen vzorkem (číslem) s jinou hodnotou, než 1 a navíc posunutým na vodorovné ose (často v čase), pak na výstupu bude posloupnost vzorků, které tvarově odpovídají impulzní odezvě, ale všechny její vzorky jsou násobeny (váženy) hodnotou vstupního vzorku a celá impulzní odezva bude posunuta stejně, jako budící vzorek, viz obr.5.21. Pro usnadnění předpokládáme, že počet nenulových vzorků impulzní odezvy je 4 s pořadovými čísly (indexy) na vodorovné ose 0 až 3. Všimněte si, že vlivem posunutí budícího vzorku x_n ve druhém případě dojde také k posunutí vážené impulzní odezvy, tedy k celkovému prodloužení počtu vzorků na výstupu. Poznamenejme, že jednotkou vzorků výstupní posloupnosti se nezabýváme, neboť tuto nelze stanovit obecně jednoznačně (může být různá podle konkrétní fyzikální povahy vzorků, chápáné uživatelem).



Obr.5.21. Odezva lineárního číslicového systému na jednotkový impulz a na obecný vstupní vzorek.

Bude zajímavé nyní sledovat, jaká bude odezva v případě, kdy na vstupu systému budou působit oba vstupní vzorky z obr.5.21, tedy když bude vstupní posloupnost sestávat z obou vzorků, viz obr.5.22.



Obr.5.22. Odezva lineárního číslicového systému na obecnou vstupní posloupnost.

Dokázat vypočítat výstupní posloupnost y_n z obr.5.22 lze několika způsoby. Jak jsem naznačil výše, jde o lineární systém. V lineárních případech vždy platí tzv. *princip superpozice*. Ten nám říká, že působí-li na vstup lineárního systému součet více budících podnětů (vzorků, zdrojů apod.), pak lze odezvu získat jako součet dílčích odezev na ony podněty, působící samostatně. Jednoduše napsáno, tahají-li dva siláci za lano, je celková síla daná součtem sil, působí-li každý zvlášť. Podle tohoto principu je jasné, že k výpočtu odezvy na vstupní posloupnost podle obr.5.22 stačí sečíst dílčí odezvy y_n z obr.5.21, kdy působí každý vzorek samostatně. Sečtení je možné provést buďto graficky nebo číselně. Budeme-li sčítat dílčí odezvy z obr.5.21, pak můžeme psát:

$$y_n = \{0,5 + 0 \quad 1,5 + 0,25 \quad 1 + 0,75 \quad 0,25 + 0,5 \quad 0 + 0,125\} = \{0,5 \quad 1,75 \quad 1,75 \quad 0,75 \quad 0,125\}.$$

Uvedený, intuitivně zavedený způsob výpočtu celkové odezvy na libovolnou vstupní posloupnost lze aplikovat obecně. V tom případě lze zapsat algoritmus výpočtu v podobě sumy podle vztahu (5.8).

$$y_n = \sum_{m=0}^n x_m \cdot h_{n-m} \quad (5.8)$$

Vztah (5.8) umožňuje vypočítat hodnoty vzorků výstupní posloupnosti lineárního číslicového systému, jsou-li známy vstupní posloupnost x_n a impulzní odezva systému h_n . Pokusme se nyní vztah (5.8) aplikovat a vypočítat analyticky hodnoty vzorků výstupu podle obr.5.22. Výpočet je uveden jako obr.5.23.

$$N_y = N_x + N_h - 1 = 2 + 4 - 1 = 5 \quad \text{počet členů výstupní posloupnosti (indexy)}$$

$$y_n = \sum_{m=0}^n x_m \cdot h_{n-m} \quad \text{výchozí vztah pro lineární diskretní}$$

$$y_0 = \sum_{m=0}^0 x_m \cdot h_{0-m} = x_0 \cdot h_0 = 1 \cdot 0,5 = 0,5$$

$$y_1 = \sum_{m=0}^1 x_m \cdot h_{1-m} = x_0 \cdot h_1 + x_1 \cdot h_0 = 1 \cdot 1,5 + 0,5 \cdot 0,5 = 1,75$$

$$y_2 = \sum_{m=0}^2 x_m \cdot h_{2-m} = x_0 \cdot h_2 + x_1 \cdot h_1 + x_2 \cdot h_0 = 1 \cdot 1 + 0,5 \cdot 1,5 + 0 \cdot 0,5 = 1,75$$

$$y_3 = \sum_{m=0}^3 x_m \cdot h_{3-m} = x_0 \cdot h_3 + x_1 \cdot h_2 + x_2 \cdot h_1 + x_3 \cdot h_0 = 1 \cdot 0,25 + 0,5 \cdot 1 + 0 \cdot 1,5 + 0 \cdot 0,5 = 0,75$$

$$y_4 = \sum_{m=0}^4 x_m \cdot h_{4-m} = x_0 \cdot h_4 + x_1 \cdot h_3 + x_2 \cdot h_2 + x_3 \cdot h_1 + x_4 \cdot h_0 = 1 \cdot 0 + 0,5 \cdot 0,25 + 0 \cdot 1 + 0 \cdot 1,5 + 0 \cdot 0,5 = 0,125$$

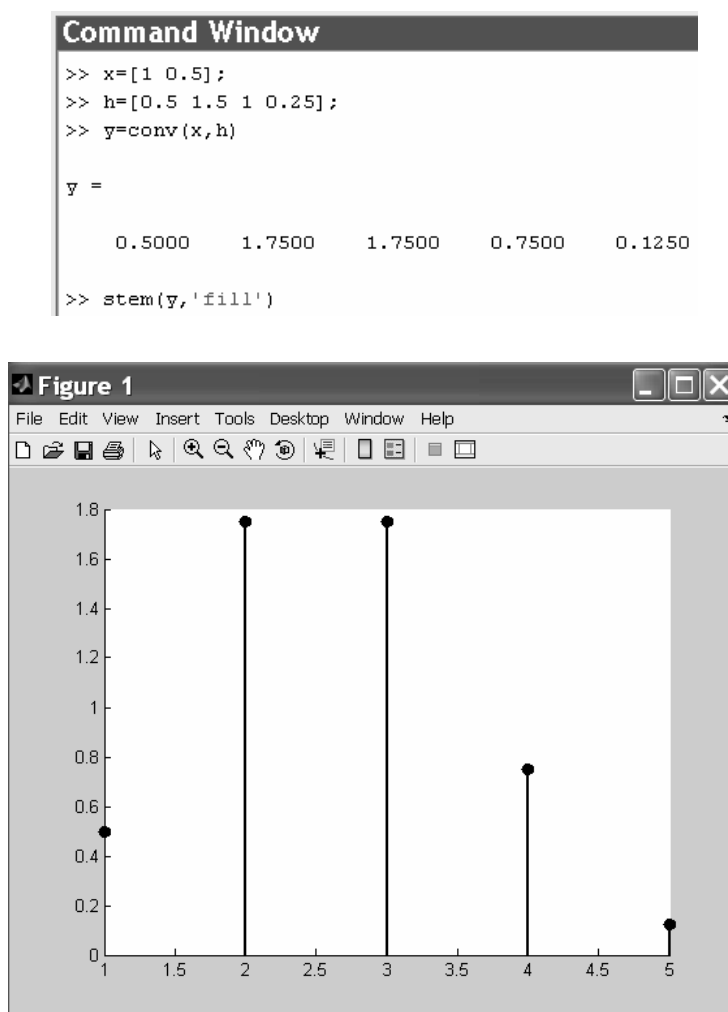
$$y = \{0,5 \quad 1,75 \quad 1,75 \quad 0,75 \quad 0,125\}$$

Obr.5.23. Aplikace vztahu (5.8) k výpočtu vzorků výstupní posloupnosti z obr.5.22.

Podívejme se nyní na počet vzorků výstupní posloupnosti, výše počítané. Počet nenulových vstupních vzorků z obr.5.22 je $N_x = 2$, délka impulzní odezvy na obr.5.21 je $N_h = 4$ a délka výstupní

posloupnosti pak $N_y = 5$. Z toho plyne jednoduchý vztah pro počet vzorků výstupní posloupnosti: $N_y = N_x + N_h - 1$, použitý v obr.5.23. Vztah (5.8) je často nazýván *lineární diskretní konvolucí*. Prozatím jsme se nezamýšleli nad termínem *konvoluce*. Bude diskutován níže v kapitole o spektrální analýze v souvislosti s větou o součinu obrazů, viz kapitola 5.3.

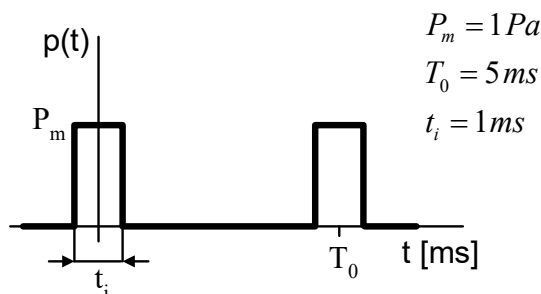
Výpočet lineární diskretní konvoluce v MATLAB je snadný díky intuitivnímu příkazu *conv*, viz obr.5.24. Jako své parametry potřebuje příkaz *conv* vektory dvou posloupností obecně různé délky. Je zajímavé, že pořadí obou vektorů lze zaměnit.



Obr.5.24. Výpočet lineární diskretní konvoluce v systému MATLAB, srovnejte s obr.5.22 a obr.5.23.

Příklad 5.2

Vypočtěte prvních 11 složek amplitudového a fázového spektra periodického obdélníkového signálu podle obr.5.57.



Obr.5.57. Časový průběh periodického signálu z příkladu 5.2.

Řešení: vyjdeme ze vztahu v tab.5.2 pro výpočet komplexních spektrálních koeficientů:

$$\dot{c}_k = \frac{1}{T_0} \int_{T_0} p(t) e^{-j.k.\Omega_0.t} dt, \text{ je zadáno } T_0 = 5 \text{ ms}, t_i = 1 \text{ ms} \text{ a } P_m = 1 \text{ Pa}. \text{ Pro další}$$

postup je třeba znát analyticky funkci časového průběhu $p(t)$. V rámci jedné periody je v případě obdélníka touto funkcí konstanta, tedy platí: $p(t) = P_m$. Výchozí vztah lze poté upravit:

$$\begin{aligned} \dot{c}_k &= \frac{1}{T_0} \int_{T_0} p(t) e^{-j.k.\Omega_0.t} dt = \frac{1}{T_0} \int_{-\frac{t_i}{2}}^{\frac{t_i}{2}} P_m e^{-j.k.\Omega_0.t} dt = \frac{P_m}{T_0} \int_{-\frac{t_i}{2}}^{\frac{t_i}{2}} e^{-j.k.\Omega_0.t} dt = -\frac{P_m}{T_0 j k \Omega_0} \left[e^{-j.k.\Omega_0.t} \right]_{-\frac{t_i}{2}}^{\frac{t_i}{2}} = \\ &= -\frac{P_m}{j k \Omega_0 T_0} \left[e^{-j.k.\Omega_0 \cdot \frac{t_i}{2}} - e^{+j.k.\Omega_0 \cdot \frac{t_i}{2}} \right] \quad \text{podle Eulerovy věty platí} \quad \sin \alpha = \frac{e^{j\alpha} - e^{-j\alpha}}{2j} \\ c_k &= \frac{P_m}{j k \Omega_0 T_0} \left[e^{j.k.\Omega_0 \cdot \frac{t_i}{2}} - e^{-j.k.\Omega_0 \cdot \frac{t_i}{2}} \right] = \frac{2P_m}{k \Omega_0 T_0} \cdot \frac{e^{j.k.\Omega_0 \cdot \frac{t_i}{2}} - e^{-j.k.\Omega_0 \cdot \frac{t_i}{2}}}{2j} = \frac{2P_m}{k \Omega_0 T_0} \cdot \sin \left(k \Omega_0 \frac{t_i}{2} \right) \end{aligned}$$

Výsledek lze upravit do často používané funkce $\frac{\sin \alpha}{\alpha} = \text{sinc}(\alpha)$, neboť její průběh je dobře znám a umožňuje snadné vyhodnocení výsledku.

$$\dot{c}_k = \frac{2P_m}{k \Omega_0 T_0} \cdot \sin \left(k \Omega_0 \frac{t_i}{2} \right) = P_m \frac{t_i}{T_0} \cdot \frac{\sin \left(k \Omega_0 \frac{t_i}{2} \right)}{k \Omega_0 \frac{t_i}{2}} = P_m \frac{t_i}{T_0} \cdot \text{sinc} \left(k \Omega_0 \frac{t_i}{2} \right).$$

rozšíření výrazem $\frac{t_i}{t_i}$

Podle posledního vztahu lze již určovat amplitudy a fáze jednotlivých harmonických

složek. Ještě než toto uvedeme, je třeba vypočítat parametry první spektrální složky, tedy konstanty, jejíž význam je *střední hodnota časového průběhu*. Tuto konstantu nemůžeme jednoduše vypočítat z odvozeného vztahu dosazením za $k = 0$, neboť bychom obdrželi neurčitý výraz $\frac{0}{0}$. Z teorie však je známo, že střední hodnotu lze vypočítat následovně:

$$P_0 = \frac{1}{T_0} \cdot \int_{T_0} p(t) \cdot dt = \frac{1}{T_0} \int_{-\frac{t_i}{2}}^{\frac{t_i}{2}} P_m \cdot dt = P_m \frac{t_i}{T_0} [Pa]$$

Výsledky odvození můžeme již shrnout:

$$\dot{c}_k = P_m \frac{t_i}{T_0} \cdot \sin c \left(k \Omega_0 \frac{t_i}{2} \right), \quad P_k = 2 \cdot |\dot{c}_k|, \quad P_0 = P_m \frac{t_i}{T_0} \quad (5.23)$$

Podle rovnice (5.23) lze již jednoduše vypočítat konkrétní hodnoty spektrálních složek.

Jednotky vždy odpovídají jednotkám zadané veličiny, v našem případě Pa .

Vztah (5.23) lze konkretizovat dosazením zadaných číselných hodnot takto:

$$\dot{c}_k = 0,2 \cdot \sin c(0,2k\pi) = 0,2 \cdot \frac{\sin(0,2k\pi)}{0,2k\pi}, \quad P_k = 2 \cdot |\dot{c}_k|, \quad P_0 = c_0 = 0,2.$$

↙
spektrální koeficienty $\dot{c}_k$
↙
harmonické složky spektra

Dosazováním za $k = 1, 2, 3, \dots, 10$ vypočteme potřebné hodnoty, viz tabulka tab.5.3.

Tab.5.3. Konkrétní vypočtené hodnoty spektrálních čar průběhu na obr.5.57.

k	$\dot{c}_k [Pa]$	$ \dot{c}_k [Pa]$	$P_k [Pa]$	$\varphi_k [rad]$
0	0,2	0,2	0,2	0
1	0,1871	0,1871	0,3742	0
2	0,1514	0,1514	0,3027	0
3	0,1009	0,1009	0,2018	0
4	0,0468	0,0468	0,0935	0
5	0	0	0	x
6	-0,03118	-0,03118	-0,06237	π
7	-0,04325	-0,04325	-0,08649	π
8	-0,03784	-0,03784	-0,07568	π
9	-0,02079	-0,02079	-0,04158	π
10	0	0	0	x

Spektrální koeficienty $\dot{c}_k$ nám posloužily jako výchozí hodnoty pro výpočet cílových parametrů harmonických složek spektra P_k .

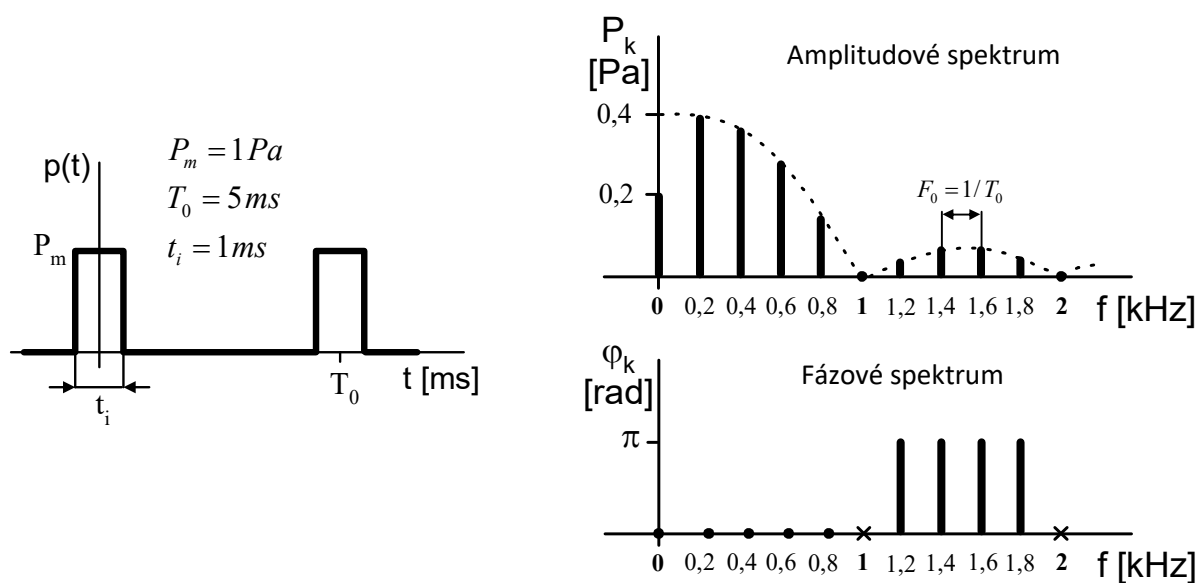
Spektrální koeficienty $\dot{c}_k$ jsou *obecně funkcemi komplexní proměnné* a lze je psát ve

tvaru $\dot{c}_k = |c_k| e^{j\varphi_k}$, kde $\operatorname{tg} \varphi_k = \frac{\operatorname{Im} \dot{c}_k}{\operatorname{Re} \dot{c}_k}$. V našem konkrétním příkladu však vyšly

koeficienty jako reálná čísla, tedy platí: $\operatorname{Im} c_k = 0 \Rightarrow \operatorname{tg} \varphi_k = 0$. Z toho plynou hodnoty fázového spektra 0 nebo π . V obou případech je tangens roven nule, avšak je třeba zohlednit fakt, že fázové spektrum je lichou funkcí (důkaz přesahuje záměr kapitoly).

Poznámka: budete-li spektrální koeficienty a harmonické složky počítat za pomoci kalkulačky, přepněte ji do režimu *rad*. V případě výpočtů za pomoci počítače vhodným software bývá toto často zajištěno automaticky (implicitně).

Časový průběh a průběhy obou spekter jsou uvedeny na obr.5.58.



Obr.5.58. Výsledná podoba spektra signálu z příkladu 5.2.

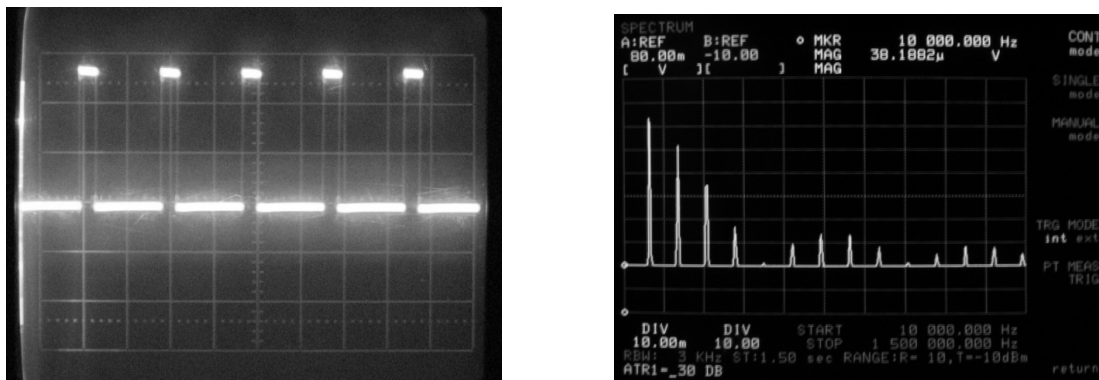
Je zajímavé, že v případě obdélníkových pulzů nabývá obálka amplitudového spektra v jistých bodech nulových hodnot. Tyto body na vodorovné ose, tedy ose kmitočtu, souvisejí se šířkou pulzu.

Nulových hodnot obálka nabývá vždy v bodech $f = \frac{k}{t_i}$, $k = 1, 2, 3, \dots$, jinak napsáno, ve spektru

vymizí vždy každá $\frac{T_0}{t_i}$ -tá harmonická složka, v našem případě každá pátá (nepočítáme tu úplně

první), viz tab.5.3. Maxima nabývá obálka amplitudového spektra pro nulový kmitočet a jde o

hodnotu $2 \cdot P_m \cdot \frac{t_i}{T_0}$, v příkladu 5.2 vychází konkrétně $0,4 Pa$. První čára, tedy střední hodnota (matematicky váha), je vždy poloviční. S těmito shrnutími lze spektrum nakreslit velmi snadno pro libovolné parametry časového průběhu. Na obr.5.59 je ilustrace k praktickému měření spekter pomocí spektrálního analyzátoru.



Obr.5.59. Časový průběh a amplitudové spektrum reálného obdélníkového signálu.

Jak je naším zvykem na stránkách knihy, doporučujeme v případě potřeby prostudovat další hodnotné prameny, např. [2], [6] nebo [10]. Dobrým místem pro poučení mohou být také vhodné internetové stránky.

Spektrum neharmonických periodických signálů - dílčí shrnutí:

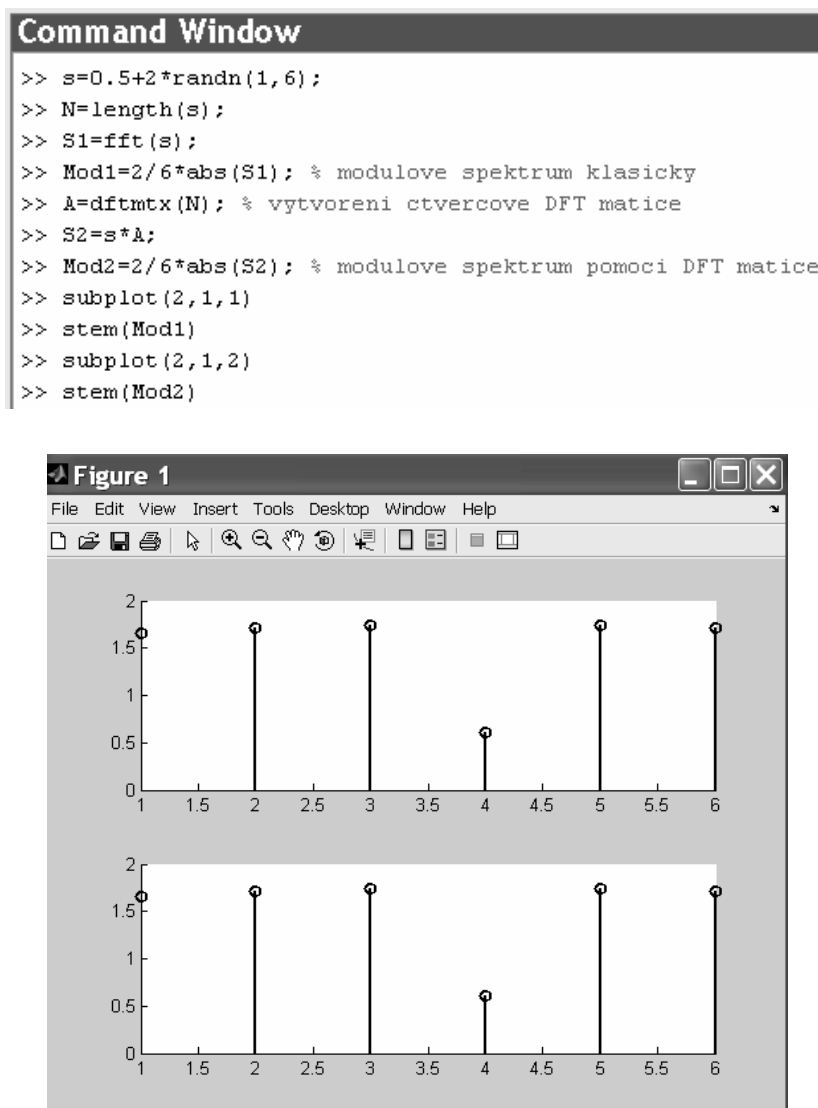
- každý periodický signál, splňující jisté matematické podmínky, lze vyjádřit ve formě nekonečné řady harmonických funkcí; tuto řadu nazýváme Fourierovou řadou*,
- Fourierova řada má tři tvary a každý tvar obsahuje jiný tvar koeficientů,
- určit spektrum znamená analyticky vypočítat koeficienty Fourierovy řady, tedy provést integraci,
- spektrum periodických signálů je čárové (diskrétní); vzdálenost mezi čarami je vždy rovna kmitočtu časového průběhu,
- výpočet koeficientů a harmonických složek spektra vyžaduje znalost funkčního vztahu pro zadaný časový průběh; pokud tento vztah není znám, nelze spektrum uvedeným způsobem počítat,
- aby mohl být periodický neharmonický signál podroben spektrální analýze, musí splňovat tzv. Dirichletovy podmínky,
- při odvozování vztahů pro koeficienty je třeba jisté zkušenosti, aby byly výsledky snadno interpretovatelné; naštěstí pro mnoho časových průběhů jsou již koeficienty odvozeny,
- v případě obdélníkových pulzů lze spektrum nakreslit snadno se znalostí několika údajů,
- takto pojaté spektrum je teoreticky správné a budeme se k němu odkazovat při zavádění spektrální analýzy diskrétních, resp. číslicových signálů.

***Jean-Baptiste Joseph de Fourier** (1768-1830) – proslulý holandský matematik a fyzik. Kolem roku 1822 formuloval ve své práci „Analytická teorie tepla“ základy Fourierovy metody řešení parciálních diferenciálních rovnic. Jeho trigonometrické řady funkcí dnes využíváme, mimo jiné,

k vyjádření, popř. k aproximaci, periodických funkcí a spektrální analýze. Taktéž Fourierův integrál (5.24) je Fourierovým objevem. Jeho životní osudy jsou velmi zajímavé a inspirativní.

Příklad 5.7

Tento příklad je zařazen pro demonstraci spektrální analýzy s využitím tzv. diskrétní Fourierovy transformační matice. Připomeňme, že případ byl již naznačen v poznámce kapitoly 5.3.2.2. Na obr.5.78 je ukázán výpočet komplexních koeficientů a zobrazeno amplitudové spektrum číslicového signálu dvojím postupem. První je klasický, v předchozích příkladech užívaný způsob, využívající funkce vnitřní *fft*. Ve druhém případě půjde o využití zmíněné transformační matice. Cílem je ukázat, že výsledky se neliší.



Obr.5.78. Spektrální analýza číslicového signálu s využitím Fourierovy transformační matice.

Jak je zřejmé z obr.5.78, platí při práci s transformační maticí vztah $S = A \cdot s$, kde s je původní vektor s číslicovými vzorky a S je vektor komplexních koeficientů jeho spektra.

Další používané varianty příkazu fft:

- pomocí *fftshift* namísto *fft* dojde k tomu, že první složka spektra bude zobrazena ve středu,
- při práci s dvou a vícerozměrnými signály je třeba použít příkazy *fft2*, resp. *fftn*, kde *n* je rozměr,
- je-li příkaz *fft* aplikován na matici, provede výpočet komplexního spektra pro každý sloupec.

5.4 Časově - frekvenční analýza

Klasická spektrální analýza signálů ve Fourierově smyslu předpokládá jeden důležitý předpoklad. *Signál, který analyzujeme, se v průběhu analýzy z hlediska frekvenčního obsahu nemění.* Jinak napsáno, vždy při výpočtech spekter uvažujeme celý signál. V praxi však existuje řada případů, kdy je délka signálu neúměrně dlouhá či v principu nekonečná nebo jde o signál s parametry proměnnými v čase (*nestacionární signál*). Představte si např. situaci, že je třeba počítat spektra signálů z nějakého snímače, jenž snímá mechanické napětí nosníku mostu při průjezdu vozidel. Situace vás povede k tomu, že připojíte záznamové zařízení na výstup snímače a zaznamenáte např. 24 hodin záznamu. Nebo budete mít k dispozici data ze snímače signálů EKG či EEK či záznam koncertu na nosiči CD. Takové signály jsou *velmi dlouhé* na to, aby byl proveden výpočet DFT (FFT) z celého záznamu. Vzhledem k ohromnému počtu vzorků by výpočet trval mnoho hodin, desítek hodin nebo i dnů; o výpočtu v reálném čase pak nemá smyslu uvažovat vůbec. Mimoto, některé signály ani celé k dispozici nemáme, neboť *jsou v principu nekonečné* (snímač mostu poskytuje data nepřetržitě). A nakonec ještě jedno omezení klasické Fourierovy spektrální analýzy. Fourierova analýza vede na kmitočtové spektrum, kdy na ose nezávisle proměnné je kmitočet jednotlivých složek spektra (analogie s duhou). Ale co když vás bude zajímat, *jak se jednotlivé spektrální složky mění v čase?* To je typické např. v oblasti rozpoznání řečového signálu, kdy nás může zajímat časový vývoj některých typických složek pro jednotlivé hlásky apod. To však klasická Fourierova analýza v principu nedokáže, neboť *analyzuje signál jako celek*.

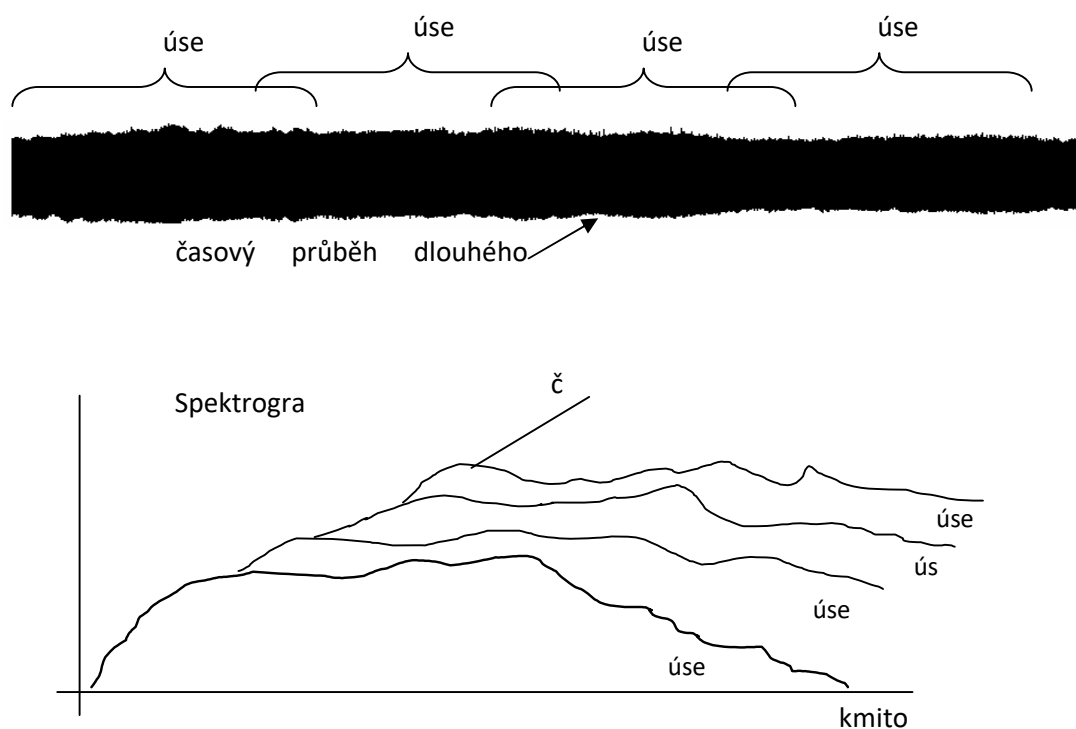
Ze všech výše uvedených důvodů, tedy přílišné délky signálu či potřebě znalosti časových změn spektrálních složek, je třeba provádět spektrální analýzu jiným způsobem. Máme k dispozici v principu dvě možnosti. Budťo využijeme výpočet spektra v jiném, než Fourierově smyslu, anebo Fourierovu analýzu použijeme, ale *rozdělíme originální signál na úseky konečné délky a nad každým úsekem provedeme námi probíranou analýzu s využitím FFT*. Získáme tím časově řazená spektra dílčích úseků (krátkodobá spektra), z nichž můžeme usuzovat na jistý, časově podmíněný vývoj spektra. Na rozdíl od kapitoly 5.3.3 však neprovádíme průměrování, ale snažíme se zachytit časový vývoj spektrálních složek.

Analýzovaný číslicový, pro naše účely velmi dlouhý signál, je rozdělen na dílčí úseky. Nad každým je provedena DFT standardně pomocí FFT algoritmu. V rámci zvýšení rozlišení ve spektru a zachování návaznosti mezi úseky jsou úseky voleny co nejdelší a navíc s určitým přesahem, viz obr.5.87. Mimoto, je možné provést analýzu několikrát a pokaždé vybrat jinou délku signálů (oken) tak, aby bylo možné ještě zvýšit rozlišení ve spektrální oblasti (podrobnější analýza). Podrobnosti uvádět nebudeme, neboť je třeba vyřešit řadu dílčích problémů, např. spojených s redukcí objemu dat.

Je-li požadavek či potřeba vypočítat takové spektrum, pak výsledek bývá zobrazen ve formě tzv. *spektrogramu*. Spektrogram je 3D graf, který má dvě osy nezávisle proměnné a to kmitočet a čas (pořadí spekter úseků). Jak je vidět z obr.5.87, jde o krátkodobá spektra jednotlivých úseků signálu

řazena za sebou. Tato krátkodobá spektra jsou na obrázku naznačena jako spojitá obálka z důvodů přehlednosti, jinak jde samozřejmě o vzorky. Na svislé ose bývá často amplitudové spektrum dané veličiny nebo výkonové spektrum.

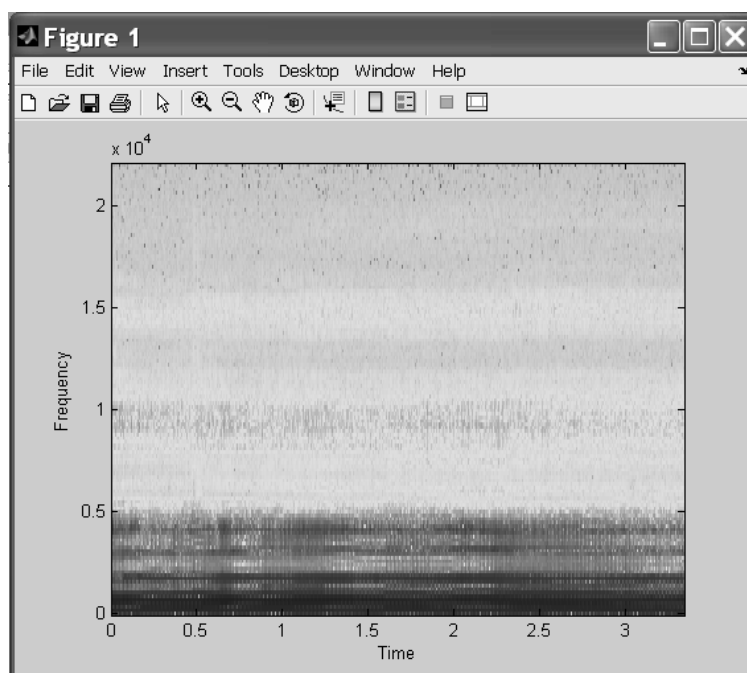
Spektrogram bývá často zobrazován jako 2D graf s tím, že jde o pohled shora na původní graf 3D. Velikosti jednotlivých spektrálních čar jsou pak rozlišeny v reálu barevně, viz ilustrace na obr.5.87. Příkaz *specgram* má řadu parametrů, minimálně jde o vlastní data a dále dvě čísla, udávající délku úseků, resp. počet bodů algoritmu FFT (zde 256) a vzorkovací kmitočet. Překrytí úseků (*overlap*) je implicitně voleno jako polovina jejich délky. Úseky jsou ze signálu vybírány implicitně pomocí Hammingova okna zadané délky, viz příklad 5.9. Uživatel má možnost tyto a další parametry ovlivnit podle povahy řešené úlohy.



Obr.5.87. Ilustrace k rozdělení dlouhého signálu na úseky a pojmu spektrogram.

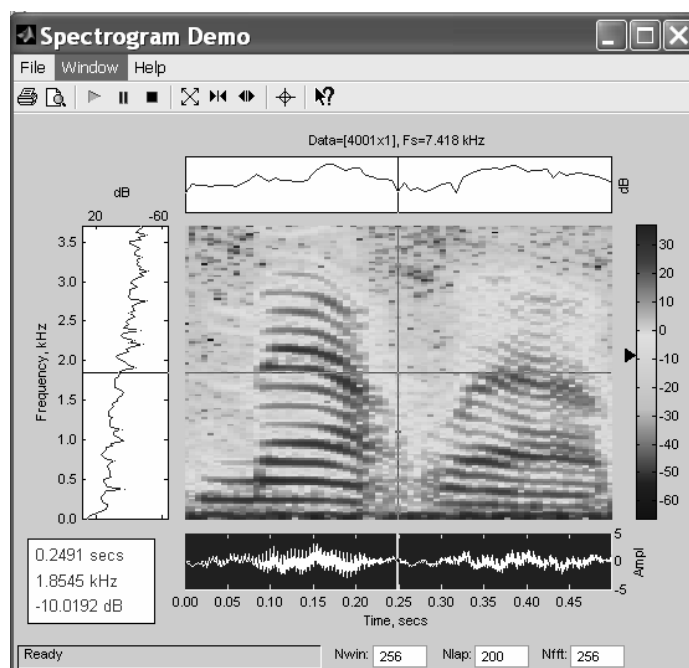
Command Window

```
>> [s,Fvz]=wavread('hlaskae');
>> s=s(:,1);
>> specgram(s,256,Fvz)
```



Obr.5.88. Základní tvorba spektrogramu číslicového signálu.

Pro účely výuky či podrobnějšího seznámení se s pojmem spektrogram nabízí MATLAB interaktivní demonstrační systém. Můžete jej spustit několika způsoby. Buďto příkazem *specgramdemo* z hlavního okna nebo z nabídky *Start/Demos* a v zobrazeném okně interaktivního helpu vybrat *Toolboxes/Signal Processing/Spectral analysis and ... /Spectrogram Demo*. Zobrazí se okno podle obr.5.89 v reálu pěkně v barvách. Pak lze myší pohybovat kurzorem a sledovat souřadnice, měnit měřítko (zoom), počet bodů algoritmu FFT (délka okna), překrytí úseků apod.



Obr.5.89. Interaktivní demonstrační systém pro práci se spektrogramy.

Literatura a odkazy pro další studium

Vybrané publikace v českém a slovenském jazyce

- [5] DAVÍDEK Vratislav a PŠENÍČKA Bohumil. *Prostředky diskrétního zpracování signálů* [cvičení]. Praha: Vydavatelství ČVUT, 1995. ISBN 80-01-01139-9.
- [6] BIOLKOVÁ Viera, BIOLEK Dalibor a PROKEŠ Aleš. *Systémy, procesy a signály I* [sbírka příkladů]. Brno: Vysoké učení technické v Brně, 1996. ISBN 80-214-0832-4.
- [7] JAN Jiří. *Číslicová filtrace, analýza a restaurace signálů*. Druhé rozšířené vydání. Brno: VUTIMUM, 2002 (dotisk 2005). ISBN 80-214-2911-9.
- [8] KOZÁK Štefan a KAJAN Slavomír. *MATLAB-SIMULINK 1*. Bratislava: Slovenská technická univerzita v Bratislave, 1999. ISBN 80-227-1213-2.
- [9] NEVŘIVA Pavel. *Analýza signálů a soustav*. Praha: BEN-technická literatura, 2000. ISBN 80-7300-004-0.
- [10] ŠEBESTA Vladimír. *Systémy, procesy a signály*. Třetí přepracované vydání. Brno: Vysoké učení technické v Brně, 2001. ISBN 80-214-1925-3.
- [11] KOZUMPLÍK, J., KOLÁŘ, R., JAN, J. *Číslicové zpracování signálů v prostředí MATLAB®* [skriptum]. Brno: VUT FEKT, Ústav biomedicínského inženýrství, 2001. ISBN 80-214-1964-4.
- [12] UHLÍŘ Jan a SOVKA Pavel. *Číslicové zpracování signálů*. Druhé vydání. Praha: Vydavatelství ČVUT, 2002. ISBN 80-01-02613-2.
- [13] ONDRÁČEK Oldřich. *Diskrétní signály a systémy*. Bratislava: Slovenská technická univerzita v Bratislave, 2002. ISBN 80-227-1649-9.
- [14] ONDRÁČEK Oldřich. *Signály a systémy*. Bratislava: Slovenská technická univerzita v Bratislave, 2003. ISBN 80-227-1875-0.
- [15] SEDLÁČEK Miloš a ŠMÍD Radislav. *MATLAB® v měření*. Praha: Vydavatelství ČVUT, 2003. ISBN 80-01-02851-8.
- [16] REKTORYS Karel a kol. *Přehled užití matematiky I*. Sedmé vydání. Praha: Prometheus, 2000. ISBN 80-7196-180-9.
- [17] REKTORYS Karel a kol. *Přehled užití matematiky II*. Sedmé vydání. Praha: Prometheus, 2000. ISBN 80-7196-181-7.
- [18] KOLDA Stanislav. *Matematika: úvod do počtu pravděpodobnosti a matematická statistika*. Páté vydání. Univerzita Pardubice, 1998. ISBN 80-7194-162-X.
- [19] Žalud Václav. *Moderní radioelektronika*. Praha: BEN-technická literatura, 2000. ISBN 80-86056-47-3.

Webové stránky v ČR

<http://www.humusoft.cz/> stránky prodejce systému MATLAB

Vybrané publikace v anglickém jazyce

- [20] BALMER L. *Signals and Systems. An Introduction*. Prentice-Hall, Inc., 1991. ISBN 0-13-809351-2.
- [21] OPPENHIEM, A.V., SCHAFER, R.W. and BUCK, J.R. *Discrete-Time Signal Processing*. Prentice-Hall, Inc., 1998. ISBN 0-13-754920-2.
- [22] POULARIKAS, A.D. *The Handbook of Formulas and Tables for Signal Processing*. RCR Press LLC, 1999. ISBN 0-8493-8579-2.
- [23] BAHER, H. *Analog&Digital Signal Processing*. John-Wiley&Sons, Ltd., 2001. ISBN 0-471-62354-7.
- [24] HANSELMAN, D., LITTLEFIELD, B. *Mastering MATLAB®*. Prentice Hall, Inc., 2001. ISBN 0-13-019468-9.
- [25] STEARNS, S.D. *Digital Signal Processing with Examples in MATLAB®*. CRC Press, Inc., 2003. ISBN 0-8493-1091-1.
- [26] McCLELLAN, J.H., SCHAFER, R.W. and YODER, M.A. *Signal Processing First*. Pearson Education, Inc. & Prentice Hall, Inc., 2003. ISBN 0-13-090999-8.

Webové stránky tvůrce systému MATLAB

<http://www.mathworks.com/> stránky tvůrce MATLAB v USA
<http://www.mathworks.com/matlabcentral/> výměna zkušeností, diskuse

MATLAB®

Začínáme se signály

prof. Ing. Karel Zaplatílek, Ph.D.