Vážení zákazníci,

dovolujeme si Vás upozornit, že na tuto ukázku knihy se vztahují autorská práva, tzv. copyright.

To znamená, že ukázka má sloužit výhradně pro osobní potřebu potenciálního kupujícího (aby čtenář viděl, jakým způsobem je titul zpracován a mohl se také podle tohoto, jako jednoho z parametrů, rozhodnout, zda titul koupí či ne).

Z toho vyplývá, že není dovoleno tuto ukázku jakýmkoliv způsobem dále šířit, veřejně či neveřejně např. umisťováním na datová média, na jiné internetové stránky (ani prostřednictvím odkazů) apod.

redakce nakladatelství BEN – technická literatura
redakce@ben.cz

Programový balík MATLAB je software a pro práci s ním jej musíte nainstalovat na disk počítače. Soudobé počítače, dnes zřejmě nejčastěji stolní osobní počítače PC či notebooky, jsou počítači číslicovými. To znamená, že pracují s čísly. Proto také každý takový software, MATLAB nevyjímaje, je primárně nástrojem, určeným pro práci se signály v číselné podobě.

Chcete-li pracovat v prostředí MATLAB se signály, resp. jejich modely (dále již budeme používat souhrnný název *signál* pro zjednodušení textu), musíte počítat s tím, že vždy budete pracovat se signály v podobě čísel. To platí jak pro případy, kdy signály přímo v MATLABu tvoříte, tak pro případy, kdy signály do MATLABu načítáte z jiných zdrojů, např. měřicích zařízení. V této kapitole se zamyslíme nad některými důsledky, které číselná podoba signálů v prostředí MATLAB přináší a upozorníme na vybrané odborné termíny a zajímavosti, které se v této souvislosti používají a jimž je vhodné věnovat pozornost.

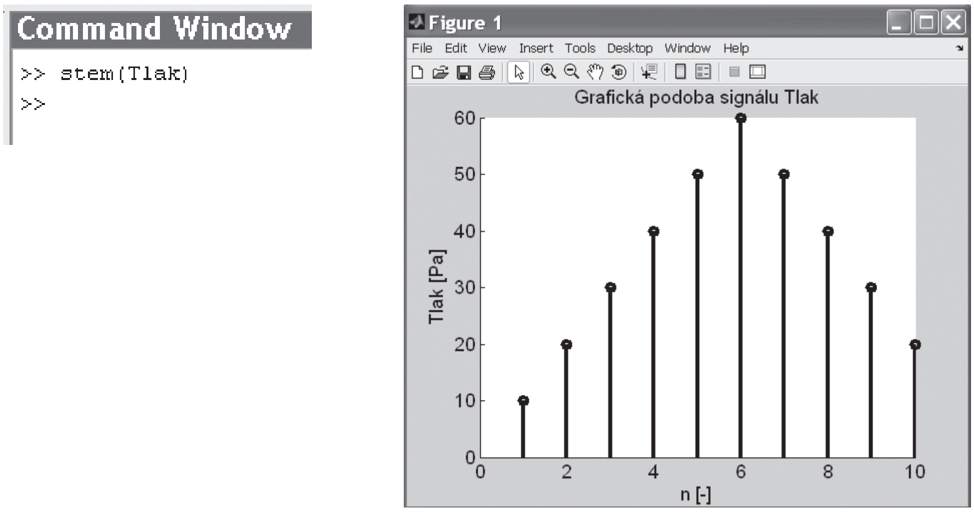
3.1 Číslicové signály a jejich zobrazení

Pojem číslicový signál byl zaveden a objasněn v kapitole 2.3.1, příklady jsou na *obr. 2.6d*, *obr. 2.8e*, *obr. 2.9g*. Protože MATLAB je určen primárně pro práci s čísly, bude číslicový signál ten nejvhodnější a nejpřirozenější pro práci v jeho prostředí. Na *obr. 3.1* a *obr. 3.2* vidíte jednoduchý příklad tvorby číslicového signálu a jeho zobrazení ve formě *posloupnosti čísel*, *obr. 3.1* a ve formě grafu, *obr. 3.2*. Podrobněji budou tvorba a generování signálů rozebrány v kapitole 4.

```
Command Window
>> Tlak=[10 20 30 40 50 60 50 40 30 20]

Tlak =

    10    20    30    40    50    60    50    40    30    20

>>
```

Obr. 3.1 *Příklad tvorby číslicového signálu a jeho zobrazení ve formě posloupnosti čísel*

Při pohledu na *obr. 3.1* a *obr. 3.2* můžete vidět, že signál *Tlak* je tvořen 10 čísly. Jde tedy o posloupnost čísel. Termínem *posloupnost* tedy budeme rozumět prostý výčet hodnot signálu, tedy soubor čísel, které jej tvoří (vektor). Bude-li signál *Tlak* zobrazen ve formě grafu, např. příkazem *stem*, pak bude patrné, že číslicový signál je signálem diskrétním na ose nezávisle proměnné. Na této ose, tedy vodorovné ose, je vlastně vyneseno pouhé pořadí jednotlivých čísel – vzorků. Logicky vás jistě nepřekvapí, že mezi těmito pořadovými čísly není nic, přesněji není signál vůbec definován. I kdybychom zvýšili počet čísel v signálu *Tlak* mnohonásobně více, vždy to bude stejné. Mimo své vzorky nebude signál definován. Vzdálenost mezi čísly grafu je konstantní. Je to pochopitelné, neboť jde o pořadové hodnoty jednotlivých čísel. Shrňme si tedy, že číslicový signál je tvořen řadou čísel. Výpis těchto čísel můžete provádět ve formě *posloupnosti čísel nebo ve formě grafické* (či jiné).



Obr. 3.2 *Zobrazení číslicového signálu z obr. 3.1 ve formě grafu*

3.1.1 Cejchování vodorovné osy grafu

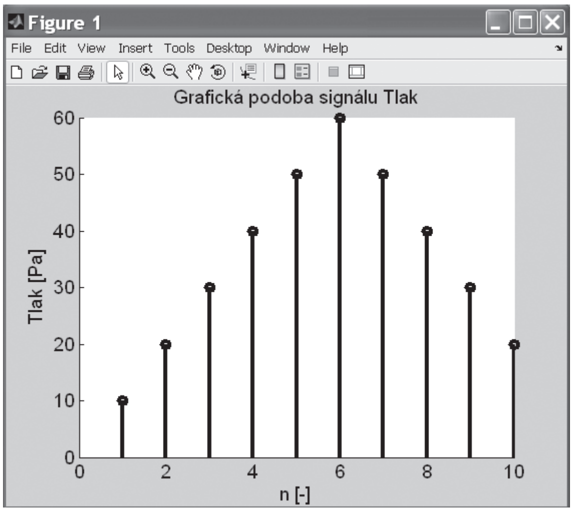
Na svislé ose grafu z *obr. 3.2* jsou hodnoty jednotlivých prvků signálu, např. tlaku v Pascalech *Tlak [Pa]*, na vodorovné pak pořadový index čísel v signálu, označený symbolem *n [–]*. Svislá osa grafu, tedy osa závisle proměnné, může nabývat libovolných hodnot. Je to přirozené, neboť velikost tlaku v Pascalech může být v principu jakákoli. Tato osa je tedy spojitá, resp. souvislá. Čísla, která tvoří signál, mohou nabývat libovolných hodnot nebo být také definována diskrétně, vždy to budou čísla a jejich zobrazení bude prováděno naprosto stejně.

Při práci se signály v jednotlivých technických či vědních oborech může být požadováno, aby na vodorovné ose grafu číslicového signálu nebyla vynášena veličina, odpovídající pořadí jednotlivých čísel – vzorků. Podle povahy signálu zde může být např. čas, jas, poloha, teplota, intenzita apod. pochopitelně vždy ve svých příslušných jednotkách. Na *obr. 3.3* vidíte dva příklady tvorby signálů a jejich grafů, které naplňují takové požadavky. Příslušný graf náleží vždy pod příklad jeho tvorby.

Z *obr. 3.3* je zřejmé, aby mohla být vodorovná osa cejchovaná jinak, než jako prosté pořadí čísel – vzorků, *je třeba ji před kreslením grafu vytvořit*. Taková osa (*Teplota*, *Čas*) bude opět řadou (vektorem) čísel a počet jejich vzorků musí být logicky stejný, jako počet vzorků na svislé ose. *Tedy, počet hodnot na obou osách musí být shodný*. Vzdálenost mezi vzorky bude na vodorovné ose vizuálně stále stejná, pokud bude tato osa lineární. Při volbě logaritmické osy pak již nikoliv. Obě osy jsou tedy tvořeny řadou čísel, *jsou tedy vektory o stejném počtu prvků neboli stejné délce*. *Přitom je zcela lhostejné, jakou fyzikální povahu oněm číslům na obou osách přisoudíme*. U číslicových signálů to nehraje žádnou roli, MATLAB prostě pracuje s čísly. Pokud uživatel nebude mít vektor vodorovné osy k dispozici, bude se muset spokojit se základním zobrazením v podobě pořadového čísla vzorků na této ose nebo s prostým výpisem hodnot, tedy s posloupností.

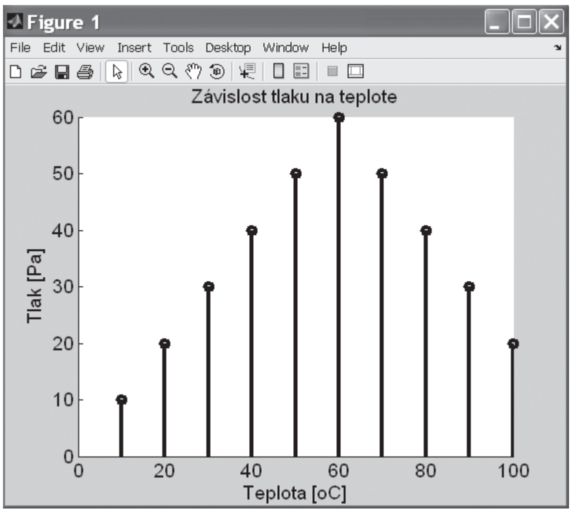
Command Window

```
>> Tlak=[10 20 30 40 50 60 50 40 30 20];
>> Teplota=[10 20 30 40 50 60 70 80 90 100];
>> stem(Teplota,Tlak)
>>
```



Command Window

```
>> Tlak=[10 20 30 40 50 60 50 40 30 20];
>> Cas=[0 5 10 15 20 25 30 35 40 45];
>> stem(Cas,Tlak)
>>
```



Obr. 3.3 *Příklady cejchování vodorovné osy číslicového signálu*

Jistě vás napadne řada příkladů, kdy měříte či jinak vytváříte číslicový signál a to bez zvláštních veličin na vodorovné ose nebo s nimi. Můžete měřit obvody stromů v lese, kdy nás vodorovná osa nezajímá, vážit děti ve škole, nebo měřit teplotu vzduchu v několika časových okamžicích po sobě. V tomto případě je k dispozici také veličina na vodorovné ose – čas již z principu měření. Vidíte, že někdy je veličina na vodorovné ose důležitá (čas měření teploty), jindy nás nezajímá (měření obvodu stromů pro účely výpočtu průměrného obvodu). Shrňme, že vždy, když máte k dispozici řadu čísel, jde o číslicový signál – jistý počet čísel. *Jejich účelnost, využitelnost, fyzikální význam či dopad těmto číslům přisuzujeme my lidé*.

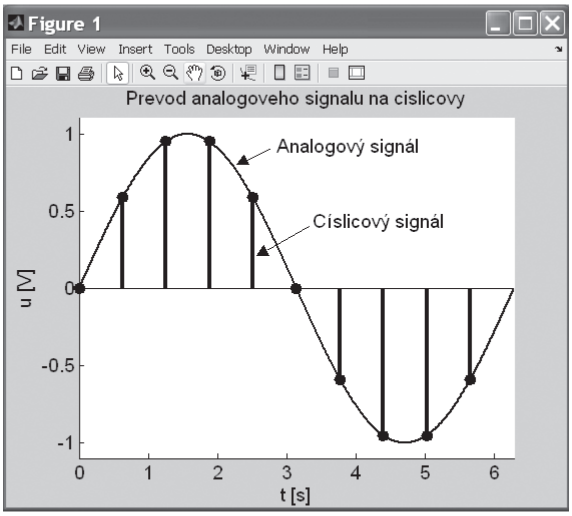
3.1.2 Vzorkování a diskrétní čas

Jak jsme již předeslali 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 obvody 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).

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 vý-

Command Window

```
>> ta=0:2*pi/1000:2*pi;
>> ua=sin(ta);
>> td=0:2*pi/10:2*pi;
>> ud=sin(td);
>> td=td(1:end-1);
>> ud=ud(1:end-1);
>> plot(ta,ua);
>> hold on
>> stem(td,ud,'fill')
>>
```



Obr. 3.4 *Převod analogového signálu do číslicové podoby*

znam času. Při pohledu na oba signály, analogový i číslicový, je patrné, že *číslicový 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 MATLABu.

```
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 *Číslicový 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 číslicový signál vznikl z analogového, pak představuje vzorky původního analogového prototypu. Z hlediska vodorovné osy, času, jde tedy o vzorky, vzniklé 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 jsme psali v kapitole 2.3.2. Odtud víme, že jedním z parametrů těchto signálů je opakovací perioda T_0 či opakovací kmitočet F_0. U číslicových 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*.

Symboly T_{vz} a F_{vz} na *obr. 3.6* označují vzorkovací periodu a vzorkovací kmitočet číslicových vzorků. Jde tedy o parametry, vztahující se k vlastním vzorkům. Symbolem T_0 je v grafech označena perioda původního analogového periodického signálu, tedy spojité obálky. Pokud znáte vzorkovací periodu, tedy vzdálenost vzorků a počet vzorků v rámci periody původního analogového signálu, snadno vypočtete jeho původní periodu. 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

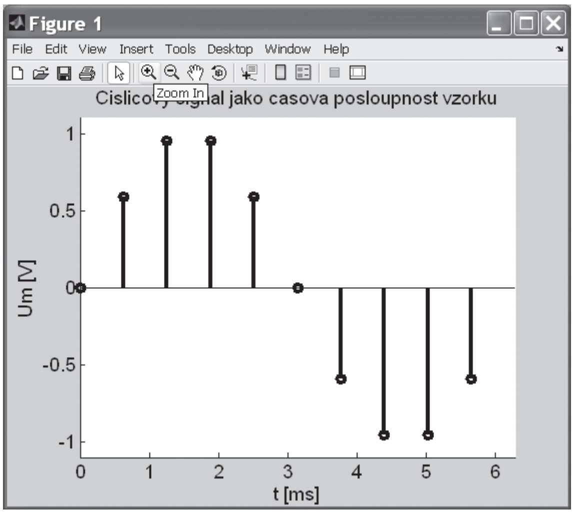
$$\frac{T}{T_{vz}} = \frac{F_{vz}}{F_0}\,[-],$$

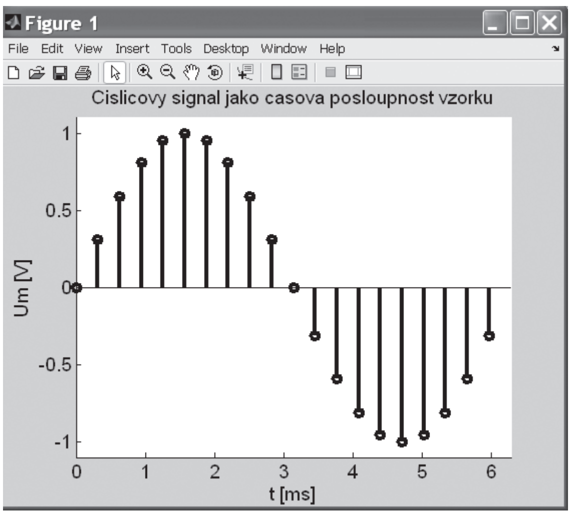
symbolem F_0 označujeme opakovací kmitočet analogového periodického signálu a platí

$$F_0 = \frac{1}{T_0}\,[\mathrm{Hz}].$$

V teorii jsou definovány a užívány také normované veličiny, kdy platí:

$$\bar{F} = \frac{F_0}{F_{vz}}\,[-] \quad \text{a} \quad \bar{T} = \frac{T_0}{T_{vz}}\,[-].$$





Obr. 3.6 *K pojmům „vzorkovací perioda“ a „vzorkovací kmitočet“*

V případě *obr. 3.6* lze u prvního grafu vypočítat:

$$\frac{T_0}{T_{vz}} = \frac{6{,}283 \cdot 0^{-3}}{0{,}6283 \cdot 10^{-3}} = \frac{6{,}283}{0{,}6283} = 10\ [-],$$

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

Řešení:

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

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

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?

Řešení:

$$\frac{F_{vz}}{F_0} = 200[-] \quad \Rightarrow \quad F_{vz} = 200 \cdot F, \quad F_{vz} = 200 \cdot 400, \quad F_{vz} = 80\ \text{kHz}.$$

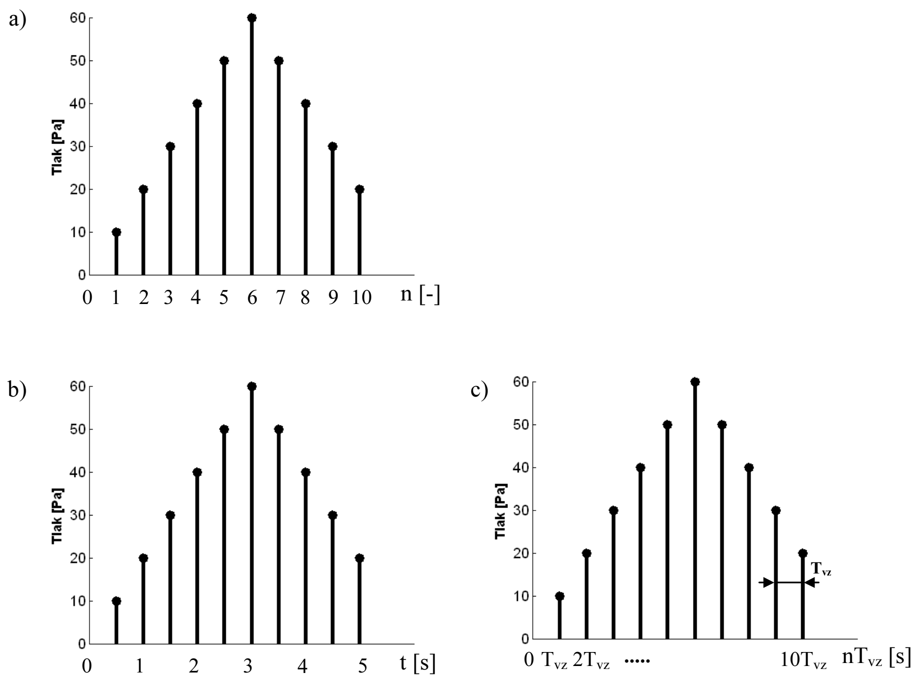
Poučení: je jistě zřejmé, že zvýšení vzorkovacího kmitočtu na dvojnásobek způsobí dvojnásobný nárůst počtu vzorků v rámci jedné periody analogového signálu.

Poznámky:

- pokud vzorkovaný signál není periodický, nemá smysl používat symbol periody T_0,
- v praxi může být proces vzorkování realizován různým způsobem; v konečném důsledku však na konci procesu převodu budou pouhá čísla,
- bez důkazů uveďme, aby byl proces vzorkování realizován korektně, je třeba dodržet podmínku $F_{vz} > 2 \cdot F_0$, tedy je třeba nejméně dvou vzorků na periodu analogového periodického harmonického signálu (podrobnosti v kapitole 5.3.2.1).

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.*

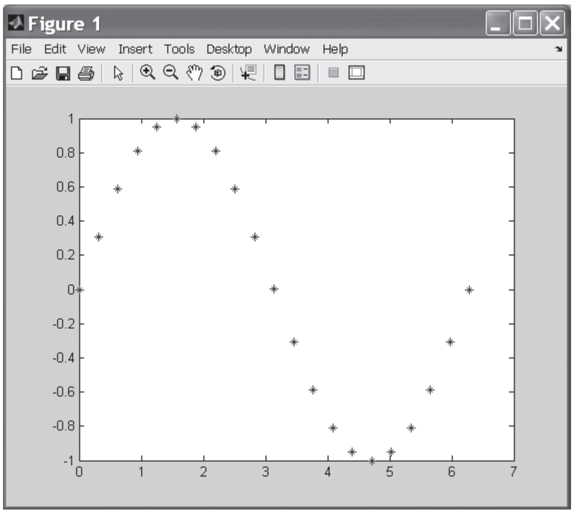
Je zřejmé, že pokud bude potřeba mít na vodorovné ose čas, bude třeba informaci o něm někde získat. Např. u souborů zvukového formátu **.wav*, používaném v operačním systému Windows, je kromě vlastních zvukových vzorků přítomna informace o vzorkovacím kmitočtu F_{vz}. Bez toho by nebylo zřejmé, s jakou 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.

3.2 Možnosti práce s analogovými signály

Z hlediska jisté komplexnosti se nelze vyhnout práci s analogovými signály a to zejména v kapitolách 4 a 5, které jsou věnovány tvorbě signálů a jejich analýze.

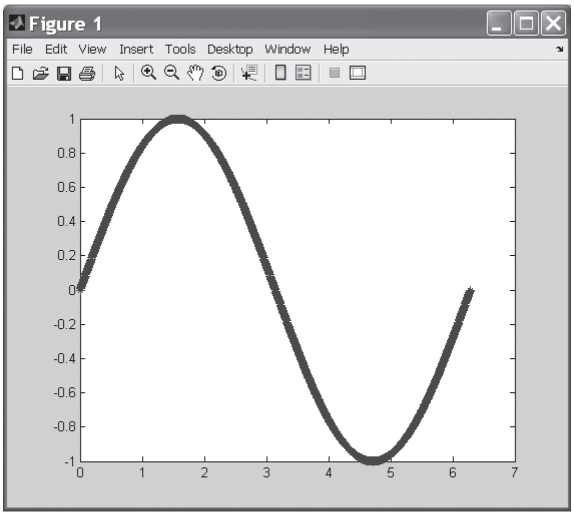
Command Window

```
>> t=0:2*pi/20:2*pi;
>> u=sin(t);
>> plot(t,u,'*')
```



Command Window

```
>> t1=0:2*pi/1000:2*pi;
>> u1=sin(t1);
>> plot(t1,u1,'*')
```



Obr. 3.8 *Simulace analogových průběhů velkým zvětšením počtu vzorků*

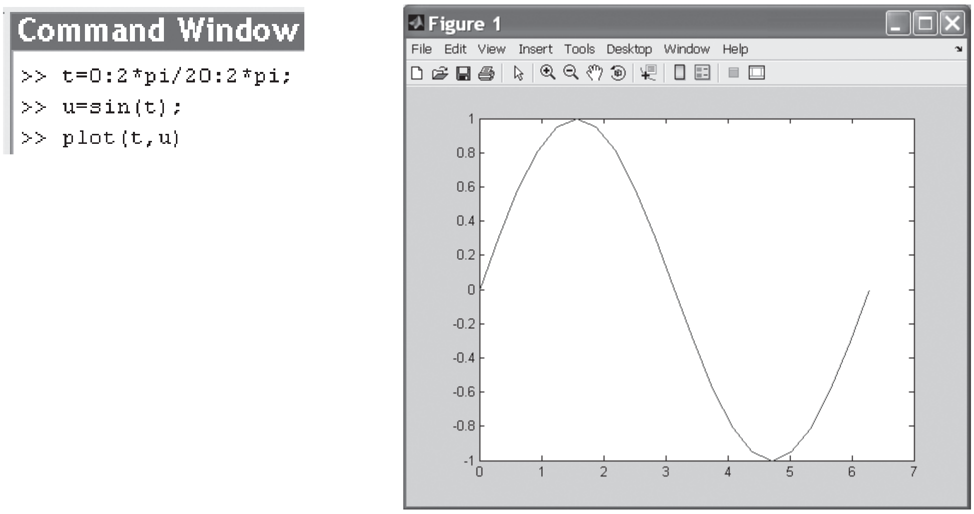
Jak jsme zmínili v kapitole 3.1.2, pravé spojité (souvislé) veličiny v MATLABu používat nelze. Lze však analogové signály jistým způsobem *simulovat* tím, že k jejich zobrazování budeme používat velký počet hodnot tak, že se ve výsledku budou tvářit jako analogové, viz *obr. 3.8* a *obr. 3.9*.

Na druhém grafu *obr. 3.8* je ukázána simulace analogových průběhů tím, že se velkou měrou zvýší počet vzorků. Graf takového průběhu *se tváří jako spojitý* v obou osách. Vy však již budete vědět, že je to jistým způsobem napodobení pravých analogových průběhů. Tento způsob nám však umožní zabývat se těmito signály a poukazovat na řadu jejich vlastností.

MATLAB má však jednu zajímavou vlastnost, která umožní zobrazovat analogové signály či spojité veličiny bez toho, aniž by uživatel musel explicitně zvyšovat počet vzorků. Stačí použít příkaz *plot* při vykreslování grafů bez dalších parametrů, viz *obr. 3.9*.

Srovnáte-li *obr. 3.9* s prvním případem na *obr. 3.8* shledáte, že příkaz *plot*, použitý bez dalších parametrů, automaticky *aproximuje průběh mezi definovanými body*, že tedy zvyšuje počet vzorků. Tímto jednoduchým způsobem lze efektivně kreslit grafy analogových signálů a platí to i v případě dvourozměrných signálů, viz např. příkazy *plot3, mesh* a *surf*.

Budete-li chtít zvýraznit číslicový, tedy diskrétní charakter signálů, použijte k vykreslení nejlépe příkaz *stem*, viz např. *obr. 3.2*, nebo příkaz *plot* podle prvního grafu na *obr. 3.8*.



Obr. 3.9 *Využití příkazu „plot“ k simulaci analogových průběhů*