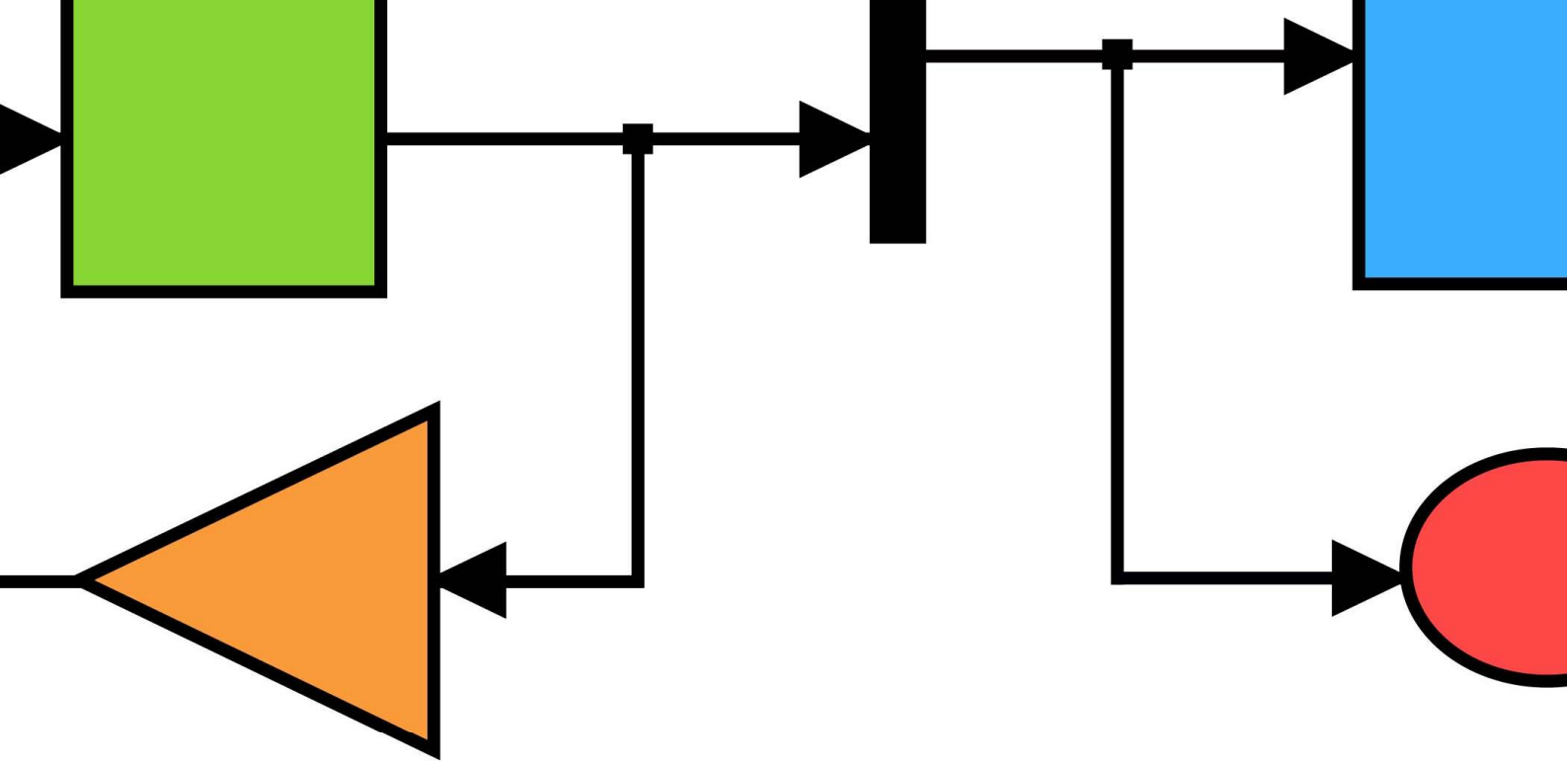
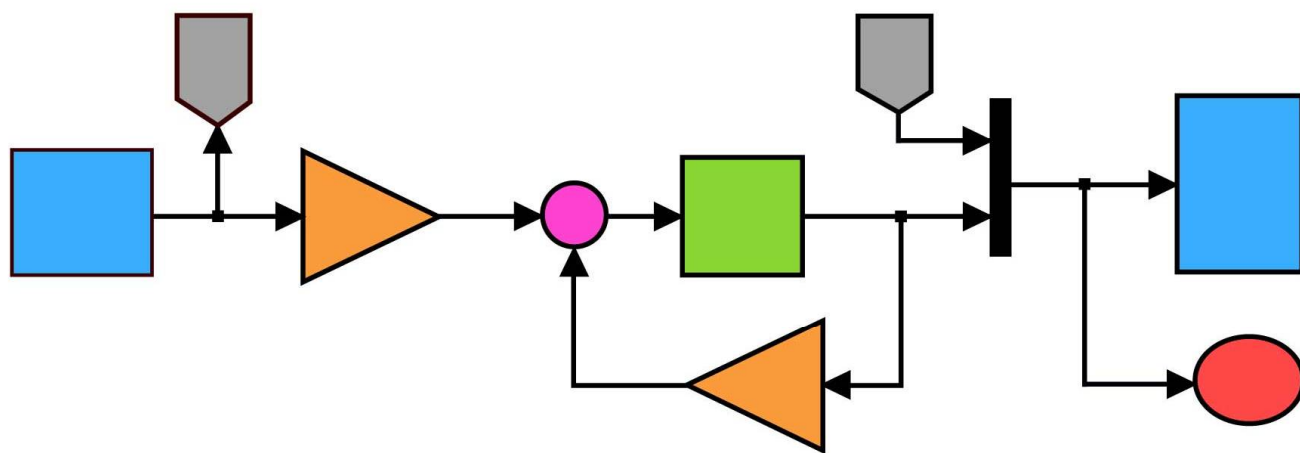




MATLAB[®]

Začínáme se systémy



Karel Zaplatílek

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

Jméno a příjmení kupujícího:

E-mail kupujícího:

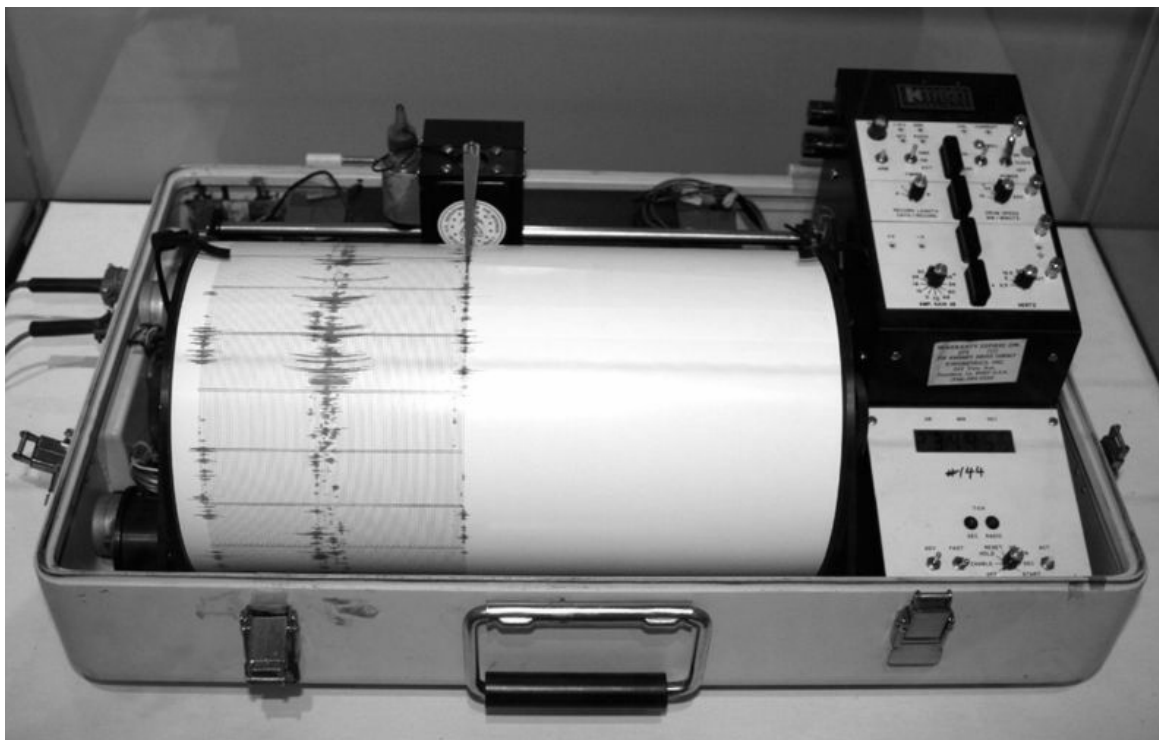
Upozorňuji, že elektronická kniha je dílem chráněným podle autorského zákona a je určena jen pro osobní potřebu kupujícího. Kniha jako celek ani žádná její část nesmí být volně šířena na internetu, ani jinak dále zveřejňována.

2.1.3 Vzájemný vztah systémů a signálů

V kapitole 2.1.2 jsem zmínil jednu důležitou souvislost, vztahující se k pojmu systém. Jde o vzájemný vztah systémů a signálů. Tento vztah hraje v teorii i praxi velmi důležitou roli. Jak jsem uvedl výše, vzájemné svázání systému se signály je neodlučitelné a je typické pro všechny obory lidské činnosti.

2.1.3.1 Určování stavu systémů pozorováním signálů

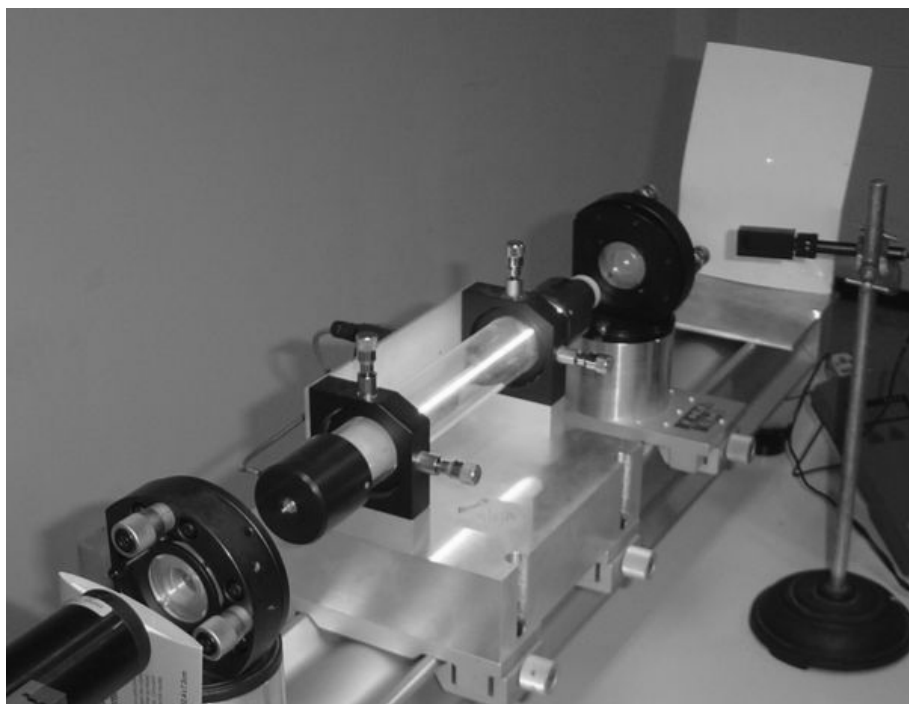
Stav systému může být určen pouze pozorováním, resp. měřením vybraných signálů. Jde např. o určení barvy vozidla, jeho hmotnosti, rychlosti, zrychlení, spotřeby, teploty kapalin, výšky stromu, zralosti jablka apod. Pro učení těchto parametrů systému je třeba znát vybrané signály, tedy fyzikální veličiny, jež nám přinášejí o daném parametru zprávu, resp. informaci. Tyto signály mohou být různé povahy; v současné době však převažuje snaha převádět je ve finální podobě na elektrické signály pro jejich snadnou archivaci, následnou analýzu, přenos apod. K tomu je třeba převodníků signálů optických, mechanických, chemických aj. právě na elektrické. My lidé vyhodnocujeme systémy prostřednictvím našich pěti smyslů. Mozek pak ve spolupráci s dalšími systémy našeho organismu může provádět jejich vyhodnocování, uchování apod. s následnou reakcí. Pro rozšíření možností je pak třeba příslušných měřicích přístrojů, opět svého druhu systémů. Tak parametry jednoho systému vyhodnocujeme díky vhodným signálům opět za pomoci jiných systémů. Takové analýze jsme učeni od útlého dětství, kdy měříme délku úsečky pravítkem, měříme poloměr s následným výpočtem obvodu kružnice, vážíme ve školní laboratoři na speciálních vahách, sami jsme po celý život měření a vážení atd. Opět jde o ukázkou těsného spojení dvojice systém – signál. Zkratka *signály nám slouží ke zjištění určitých parametrů systému*, k určení jeho *stavu*. Zajímavou otázkou je přesnost takového určování parametrů. Z praxe známe určování od orientačního pozorování až po vysoce přesná měření. Kromě samotných údajů (dat) jde zde také o určení možných chyb při měření a tzv. *nepřesností (správně nejistot) při měření*, kterými se zabývá *metrologie*. Na internetu se můžete inspirovat např. na serveru encyklopedie www.wikipedia.cz, kde naleznete řadu zajímavých odkazů pod termíny *metrologie*, *měřicí přístroj* apod. Na obr.2.2 je ilustrační příklad měření, tedy zjišťování parametru nějakého systému.



Obr.2.2. Ilustrační příklad reálných měření a záznamu signálů seismografem.

2.1.3.2 Systémy jako zdroje signálů

Systémy poznáváme díky pozorování, resp. měření příslušných signálů. To je do jisté míry *pasivní přístup*. Mnohdy ale konstruujeme systémy právě s cílem generování potřebných signálů. Takový generátor může mít povahu elektrickou (generované signály jsou elektrickými veličinami), ale také optickou (laserová dioda pro optický kabel, světlo pro signalizaci mezi loděmi), mechanickou (vibrace pro účely zkoušek), chemickou či jinou. Přitom jsou někdy různé druhy signálů mezi sebou transformovány např. z elektrického na mechanický či optický. Účelem konstrukce generátorů signálů může být nutnost provádět zkoušky systémů, kdy vyhodnocujeme změnu, kterou signál nabyl průchodem systémem, viz další kapitola. Dalším motivem může být přenos dat (zpráv) s využitím některé modulační techniky, signalizace apod. Typickým příkladem je zmíněný přenos dat optickým kabelem, kde potřebujeme kvalitní zdroj záření ve viditelném či jiném intervalu spektra. Nebo generátory pro účely vysílání s využitím elektromagnetického vlnění (televize, rozhlas). Jde o *aktivní přístup k tvorbě signálů*, viz obr.2.3.



Obr.2.3. Helium-neonový laser jako zdroj kvalitního světelného záření.

2.1.3.3 Změna stavu systému působením signálu

V předchozí kapitole jsem uvedl, že signály vyrábíme také za účelem zjišťování stavu systémů. Jde o *aktivní přístup*, kdy generovaný signál přivedeme na vstup systému, abychom následně zjišťovali jeho stav vyhodnocením změn signálu, procházejícího systémem. Mimoto však je skutečností, že procházející, resp. zpracovávaný signál, způsobí v systému jisté změny. Jde o *změny jeho stavu, které se promítají do měřených parametrů*. Jak vypadá taková vnitřní změna stavu systému? Záleží na konkrétní povaze systému. Může jít o změnu napětí či proudu některými součástkami u elektrického systému, změnu průřezu, délky či průhybu u mechanického nosníku apod. Změna těchto *stavů, tedy typických parametrů, popisujících chování systému*, je způsobena změnou stavu některých subsystémů (součástek, částí). Pokud působící signál způsobuje změnu velikosti těchto *stavových veličin*, hovoříme o *kvantitativním působení*. Působení signálu však může vyvolat i *kvalitativní změny*. Signál může způsobit přepnutí elektrického přepínače, zahlcení komunikačního kanálu, destrukci mechanické konstrukce atd. V tomto případě jde nejen o změnu stavu subsystémů, ale i o změnu jejich vazeb (propojení). A to se samozřejmě promítne do chování celého systému. Takový vliv signálů na systémy může být chtěným jevem, např. u zkoušek odolnosti či měření, ale může jít také o nechtěný, tedy parazitní jev. To vždy záleží na konkrétním posouzení. Typickým příkladem je působení

mechanických kmitů či napětí na různé konstrukce s cílem rozpoznat jejich vlastnosti, příp. mezní stavy. Zajímavými jsou metody nedestruktivního testování materiálů. U řady komerčních produktů *podobné zkoušky dokonce určují normy*. Podobné je to s teplotním namáháním, korozí, odolností proti působení chemických vlivů, s ochranou proti střelným zbraním aj. Shrnutí, změny stavů systémů působením signálů jsou důležitou vlastností. Jejich vyhodnocením lze získat cenné informace o systémech a jejich vlastnostech. Velkou roli zde hraje *zpětná vazba*. Zjišťování stavů systémů signály může vést ke zpětné úpravě vlastností systémů. Toho se využívá při konstrukci systémů a jejich modelů. Na obr.2.4 vidíte mechanické železniční návěstidlo, jehož stav je ovládán mechanicky pohybem drátů.



Obr.2.4. Stav mechanického dvojramenného návěstidla.

2.2 Modelování systémů

V této kapitole se seznámíme s pojmem *model systému*. Ukážeme si význam modelů, jejich základní druhy a použití a poukáží na rozdíly mezi reálnými systémy a jejich modely.

2.2.1 Skutečný, reálně existující systém

Jak bylo vysvětleno v kapitole 2.1, systémem je vzájemné propojení určité množiny subsystémů (prvků, součástek, částí) tak, že celek plní určitou funkci. Reálně existující systémy pracují vždy s reálnými signály. *Jen pomocí reálných signálů můžeme zjišťovat stav reálných systémů*. Tak např. zjistit stav akumulátoru automobilu lze jen měřením s využitím měřicích přístrojů. Podobně stav počasí zjistíme měřením signálů, jako jsou teplota, tlak apod. Někdy je třeba vyhodnotit menší množství signálů, jindy větší. Počet základních parametrů, které charakterizují vlastnosti systému,

nazýváme *stavem systému*. A jak již víme, ke zjištění každého stavu je třeba znát příslušné signály. *Stav systému je tedy vektorem (souborem) jeho klíčových parametrů*. Přitom počet parametrů systému je, jak je popsáno v kapitole 2.1.2, *relativní vzhledem k pozorovateli a účelu analýzy*. V našem příkladu s počasím je zřejmé, že někomu stačí teplota, jiný vyžaduje ještě tlak a meteorolog bude získávat další parametry. Určování stavu systémů je v praxi často poměrně nákladné, zejména při vývoji nových systémů, kdy se analýza opakuje v jistých cyklech a je zdrojem dat pro úpravu vlastností subsystémů. Typické je to při vývoji nového modelu osobního automobilu, architektonicky neobvyklé mostní konstrukce aj.



Obr.2.5. Příklad reálného systému - meteorologická stanice v katastru obce Sobotín v Jeseníkách (převzato z www.meteo.tombrou.com).

► Reálné systémy a práce s nimi:

1. skutečný, reálně existující systém pracuje vždy s reálnými signály.
2. Určit vlastnosti reálného systému znamená zjistit (změřit) reálné signály pomocí přístrojů.
3. Stavem systému rozumíme souhrn (vektor) jeho klíčových parametrů.
4. Počet parametrů, potřebných ke stanovení vlastností systému, je relativní vzhledem k pozorovateli a účelu pozorování.
5. Praktické měření parametrů systémů může být značně nákladné, zejména při jejich vývoji.

2.3 Klasifikace systémů a jejich modelů

Podobně jako u signálů platí, že každá snaha něco rozdělit, klasifikovat je poplatná tomu, kdo rozdělení provádí, resp. účelu, s nímž je k rozdělení přistupováno. Někdo dělí letadla v základním přístupu podle druhu pohonu (pístová, proudová), jiný preferuje primární rozdělení podle účelu stroje (dopravní, bojová, víceúčelová) a další podle doletu. Je zřejmé, že na reálné systémy i jejich modely se můžete dívat z mnoha úhlů a hledisek a to vás pak přivede k jejich rozdělení. Aby však byl v klasifikaci signálů a jejich modelů jistý řád, je vhodné se přidržet jednak logiky práce a pak také u nás i ve světě běžně užívaného a osvědčeného rozdělení. Jednotlivé podkapitoly psané níže jsou vždy jedním úhlem pohledu na systémy a modely systémů. V praxi je pochopitelné, že se některé doplňují a

kombinují. Tak může jít o systém či model analogový a současně nestabilní, nebo číslicový a současně lineární a stacionární. Z terminologického hlediska je dále uváděná klasifikace systémů vždy pohledem na různé druhy jejich modelů. Jak již víte, lze reálné systémy modelovat různými druhy modelů podle účelu a možností. Ke každému druhu systému se váže jistý matematický modelový aparát. Ten nám ukazuje řadu zajímavých a užitečných vlastností systémů a zejména jejich souvislostí. V této kapitole se věnuji rozdělení a matematikou budeme výrazně šetřit. Půjde primárně o úvod do problematiky, který bude blíže rozveden v dalších partiích knihy. Je vhodné se také obohatit studiem další odborné literatury.

2.3.1 Systémy analogové, diskrétní a číslicové

Začneme tuto kapitolu zajímavou otázkou. Je reálný svět okolo nás spojitý (plynulý) anebo diskrétní, tedy projevující se „skokově“, resp. mající konečný počet svých stavů? Zkuste se nad položenou otázkou zamyslet. Naše každodenní zkušenost nám říká, že vše okolo nás je plynulé, spojitě. Že v reálné praxi nenajdeme žádný systém, jehož stav by se měnil ideálně skokově, resp. že by zaujímal jen některé stavy a jiné ne.

Představte si však železniční výhybku. Jde o systém, který má běžně dva (výjimečně více) pracovní stavy. Má-li jakýkoli systém konečný počet svých stavů (hodnot klíčových parametrů), nazýváme jej *systémem diskrétním*. Takovou výhybku můžeme tedy popsat dvěma stavy, např. přehozeno vlevo a přehozeno vpravo. Přisoudíme-li těmto stavům nikoli slovní, ale číselné vyjádření, hovoříme o *systému číslicovém (stav 0, stav 1)*. Výhybka přehozená vlevo bude mít např. číslo 0, vpravo číslo 1. Číslicový systém je tedy zvláštním případem systému diskrétního, kdy jeho stavy jsou vyjádřeny čísly.

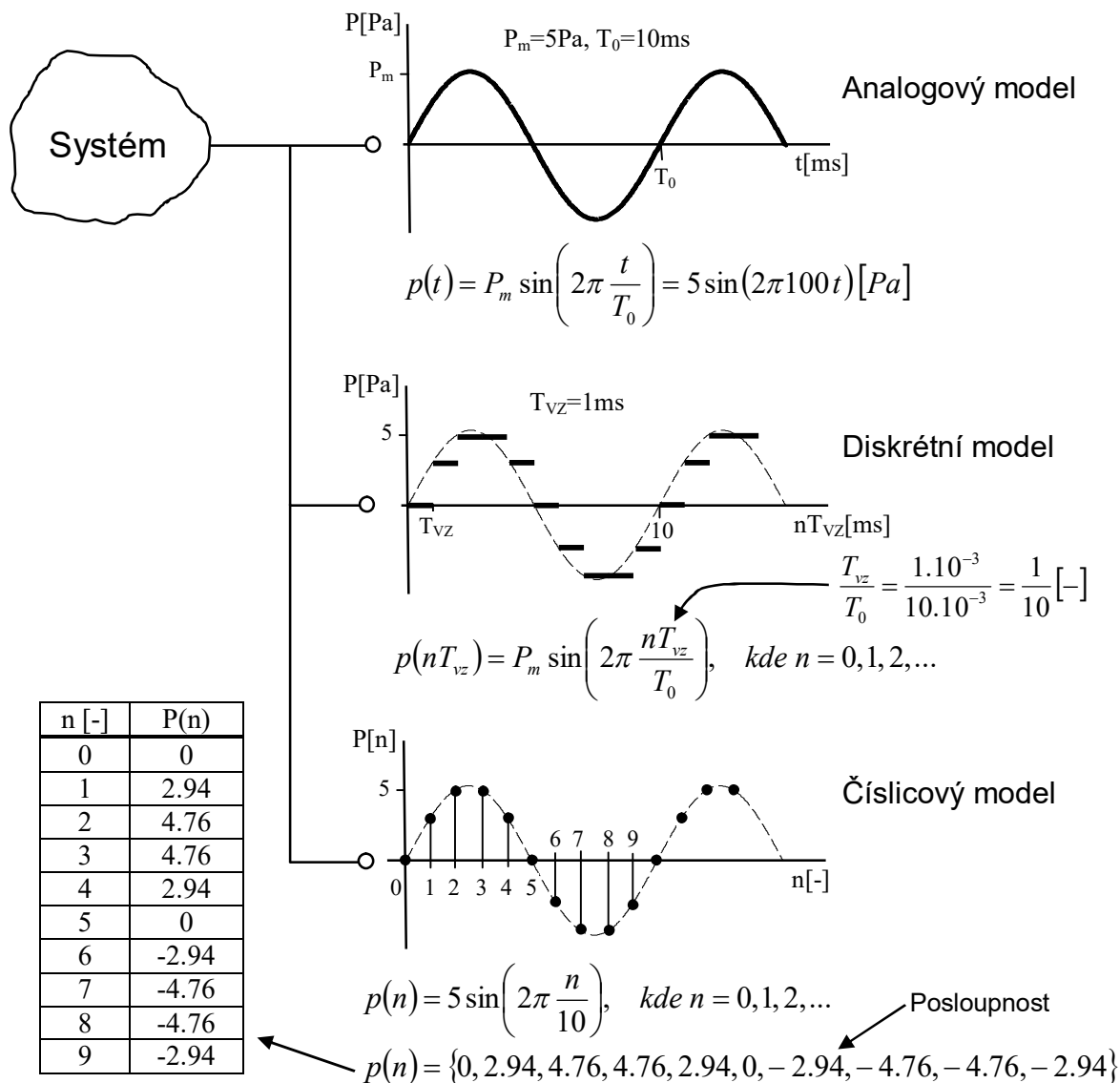
Při bližším pohledu však zjistíme, že přechod mezi stavy u naší výhybky není ideální. Je plynulý, tzn., že mezi dvěma pracovními stavy můžeme nalézt libovolné množství (prakticky nekonečné) jiných stavů, vlastně poloh ramene výhybky, než dosáhne koncového stavu. Přechod je tedy plynulý, říkáme spojitý (souvislý). Z tohoto úhlu pohledu je výhybka *systémem analogovým*, spojitě se chovajícím. Pro takový systém je typické, že hodnoty jeho parametrů mají v jistém intervalu nekonečné množství hodnot (mění se spojitě).

Je tedy naše pokusná výhybka systémem analogovým, diskrétním nebo číslicovým? Odpověď je překvapivá. Může být kterýmkoliv z uvedených systémů. Na čem to záleží? Na nás samotných, tedy roli pozorovatele, subjektu, jak jsem o tom psal již v kapitole 2.1.2. *Záleží jen na tom, jakým modelem budeme reálný systém popisovat*. Naše smysly nám denní zkušenosti říkají, že všechny reálně existující systémy jsou analogové. Naše zkušenost je spojitá. Na základě vyhodnocení potřeb a možností však můžeme i takovým reálným systémům přisoudit model diskrétní, resp. číslicový. Děláme to tehdy, když jsou pro nás významné právě pracovní stavy systémů a ne přechody mezi nimi. To vede často k výraznému zjednodušení modelů, neboť se můžeme soustředit na principy a nezabývat se detaily. *Číslicové modely systémů jsou v současné éře číslicových počítačů vyžadovány stále častěji*. Počítače pracují s čísly, a proto je přirozené pracovat na nich s číslicovými modely. Kdy je vhodnější pracovat s modely analogovými? Samozřejmě tehdy, když pracujeme s analogovými systémy, kdy by číslicový model poskytoval příliš zjednodušené výsledky, nepostihující výrazné analogové aspekty, často nazývané *reálné vlastnosti*. Analogové modely mohou být však složitější a práce s nimi náročnější. V současné době se můžeme setkat např. s elektronickými systémy, které jsou analogové, avšak pracují s polovodičovými spínači, majícími konečný počet (zpravidla dva) stavů. Tyto systémy jsou obecně analogové, avšak periodické spínání umožňuje pohlížet na ně také jako na diskrétní, resp. číslicové systémy. Záleží na tom, co chcete se systémy dělat, jaké jejich parametry modelovat.

V souvislosti s pojmy analogový, diskrétní a číslicový systém jde tedy o relativitu termínu vzhledem k volbě jeho modelu.

Uveďme si zajímavý příklad. Uvažujme nějaký systém. Ve snaze po vyhodnocení jeho parametrů budeme zaznamenávat jeho signály, viz obr.2.9. Zde jsou ukázány tři modely signálu, produkovaného nějakým systémem. Analogový model (signál, rovnice) je přístupem spojitým. Závisle proměnná, tedy svislá osa grafu, může v intervalu $p \in \langle -5,5 \rangle$ nabývat libovolné hodnoty. Tato *spojitost závisle*

proměnné je typická právě pro analogové systémy, viz [9], [16] a [23]. Spojité jsou přitom obě osy (proměnné).



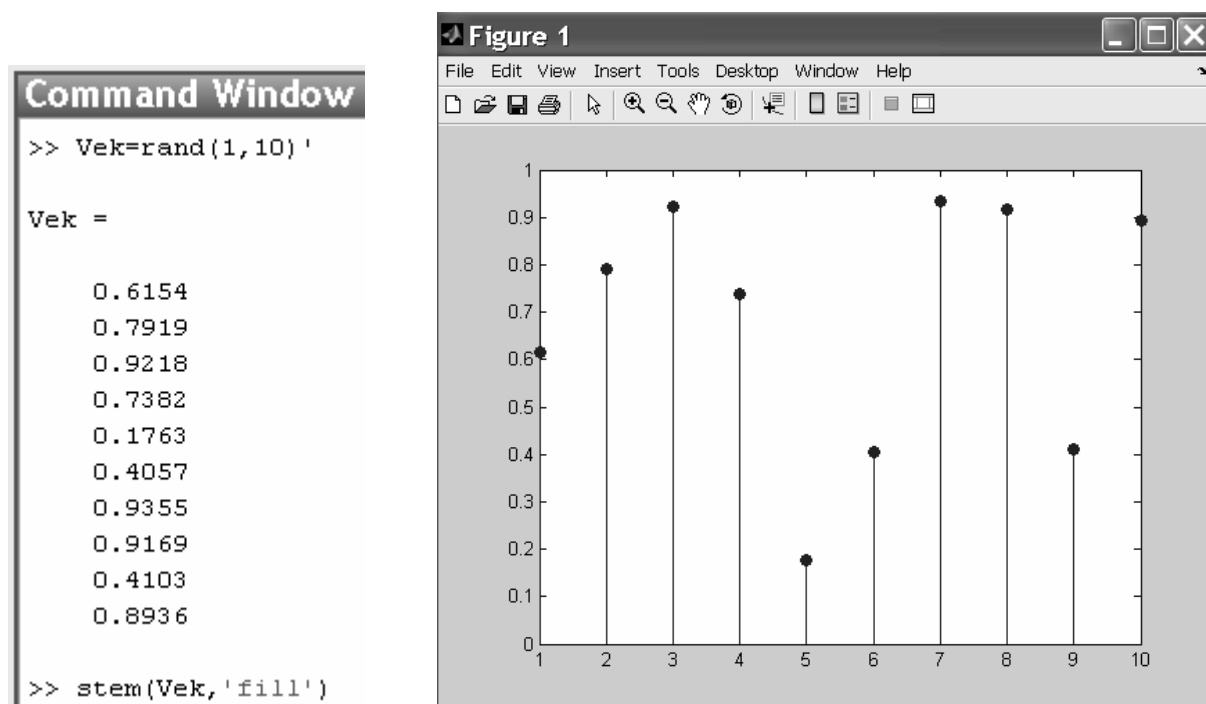
Obr.2.9. K pojmům analogový, diskrétní a číslicový systém.

Jestliže závisle proměnná modelu systému může nabývat jen omezený, konečný počet hodnot (kvant), jde obecně o *systém diskrétní*. V našem případě je volena také diskrétní osa nezávisle proměnné (zde času), avšak to není důležité, může být i spojitá. To, že jednotlivé stavy trvají stejnou dobu, není rozhodující. Mohou být každý jinak „dlouhý“, dokonce náhodně dlouhý. Konečný počet stavů je rozhodujícím kritériem, ne jejich délka. Jestliže budou obě (nebo více) proměnné (osy) diskrétní a závisle proměnná bude vyjádřena číselně, jde o *číslíkový model*. Pak, v souladu s tabulkou v rámci obr.2.9, získáváme dva vektory čísel a to vektor nezávisle a vektor závisle proměnné. Pokud zapíšeme funkční hodnoty našeho signálu tlaku do závorky, hovoříme o *posloupnosti*. Posloupnost je tedy seznam čísel, odpovídající hodnotám (vzorkům) nějakého parametru.

Jak je to s určením typu signálu při práci v systému MATLAB? Číslicový signál vyžaduje obě osy diskrétní. Navíc, abychom byli přesní, také kvantování (konečný počet hladin) a kódování do dnes nejčastěji využívané binární (dvojkové) soustavy. To je typické pro práci číslicových počítačů běžného typu. Pokud tedy budete mít v systému MATLAB vektor nějakých čísel, *jde správně vzato o*

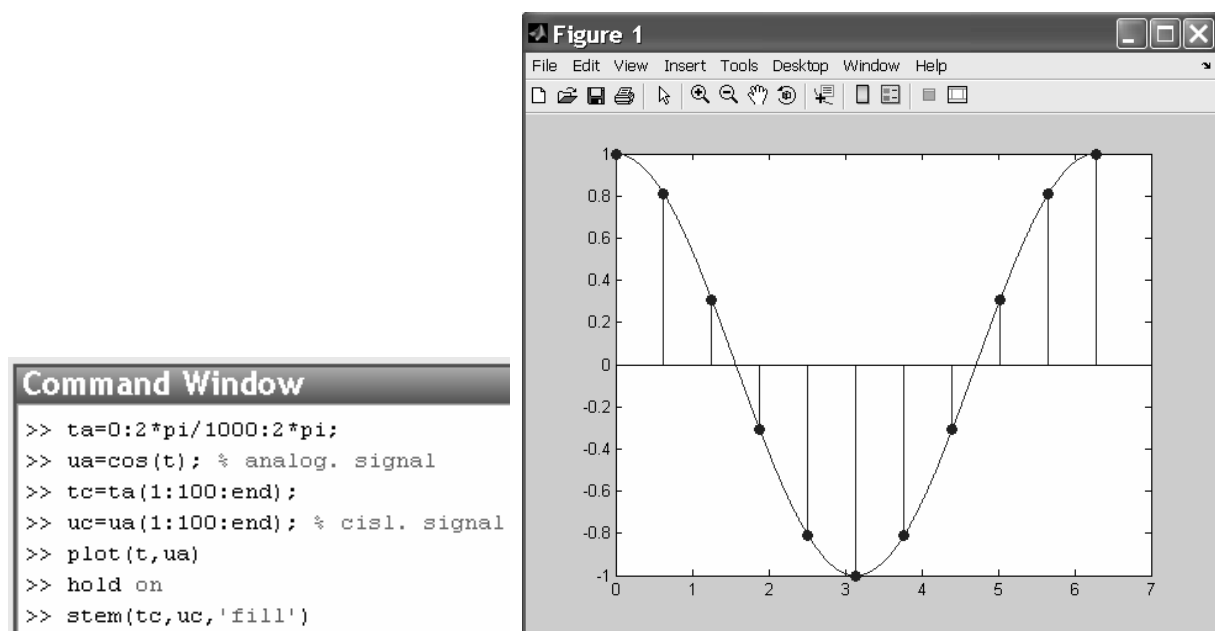
diskrétní signál, neboť závisle proměnná nemusí mít vždy zaručen konečný počet stavů (kvantování), např. signál $s(n) = \{1.1, -0.6, 0.2, 4\}$, viz dále obr.2.10. Bude-li navíc zaručeno kvantování svislé osy grafu (konečný a daný počet úrovní v rámci nějakého intervalu), pak můžeme hovořit o číslicovém signálu i bez kódování do některé varianty dvojkového kódu (zjednodušení). Všimněte si, že v případě diskrétní osy času je na obr.2.9 uvedena hodnota tzv. *vzorkovací periody* T_{vz} , tedy vzdálenosti mezi vzorky signálu. V případě periodických signálů je důležitým parametrem poměr $\frac{T_{vz}}{T_0} [-]$, viz např. [5] a [19].

V rámci jistého zjednodušení však budeme hovořit o číslicovém signálu všeobecně v případě práce s čísly v systému MATLAB (i bez kvantování hodnot). Navíc, použijete-li velký počet hodnot prvků vektoru čísel, bude se graf signálu „tvářit“ spojitě, i když ve skutečnosti není. Lze tím však simulovat práci s analogovými systémy (kromě volby příslušných metod analýzy apod.).



Obr.2.10. Příklad tvorby číslicového signálu v prostředí MATLAB.

Při práci v MATLAB budeme pracovat s číslicovými signály (již znáte podrobnosti), u kterých je na vodorovné ose grafů pouhé pořadí vzorků (posloupnost, vektor čísel), viz obr.2.10. Je-li třeba jiná veličina (čas, teplota aj.), je třeba ji mít k dispozici ve stejném počtu vzorků (funkce). Na obr.2.11 je dokumentována možnost práce s analogovými signály, které jsou však simulovány velkým počtem hodnot posloupností. Tento přístup bude použit také u dalších typů grafů, nejen časových průběhů signálů (kmitočtové charakteristiky apod.).



Obr.2.11. Práce s analogovými signály v prostředí MATLAB.

Poznamenejme, že při klasifikaci systémů jsme zatím vyhodnocovali signály, které získáme měřením či simulací nějakého systému. Popsat systém, jak jsem již zmínil, lze však také jiným druhem modelu, viz dále v knize. Vysvětlení pojmů analogový, diskrétní a číslicový však platí univerzálně.

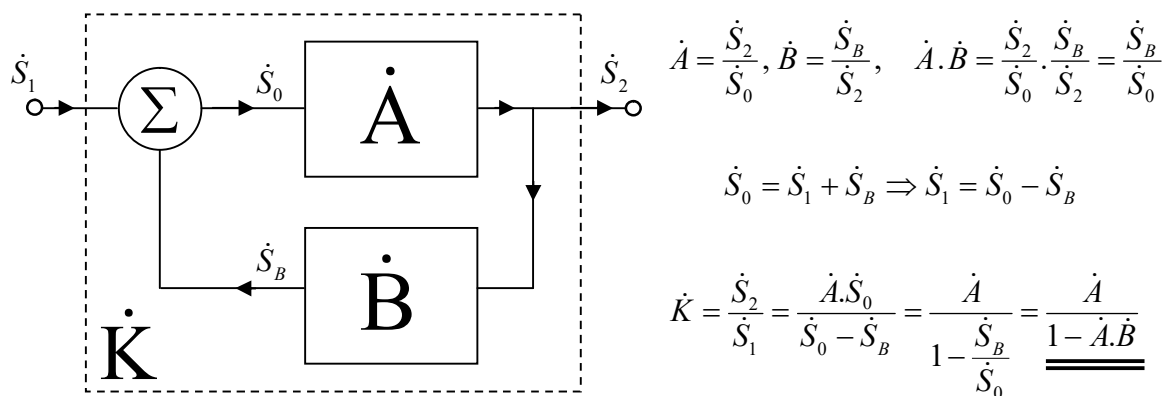
2.3.4 Systémy se zpětnou vazbou a systémy bez zpětné vazby

Zpětná vazba je nám lidem známa odpradávná. Naše běžná životní zkušenost je jí stále naplněna. Při chůzi, běhu, řízení vozidla apod. neustále sledujeme řadu vjemů (rychlost, vzdálenost od okolí aj.) a na základě vyhodnocení těchto mnoha parametrů dokážeme korigovat pohyb či jinou činnost. Bez zpětné vazby bychom automobilem jeli stále rovně anebo jak by nás napadlo bez ohledu na okolní či vnitřní situaci. Neustálá korekce naší činnosti vyhodnocováním řady vjemů je běžnou denní zkušeností. Tento koncept je blízký řídicím systémům, které vlastně pracují analogicky, jen k tomu využívají více či méně složitých technických (tedy neživých) strojů, přístrojů a dalších zařízení včetně software.

2.3.4.1 Systémy se zpětnou vazbou

Z hlediska technických věd můžeme napsat, že moderní pojetí termínu *zpětná vazba* (*Feedback*) se datuje přibližně od roku 1932, kdy byl zaregistrován *U.S. Patent 2, 102, 671* na zesilovač s pevným zesílením. Podstatou patentu bylo odstranění tehdejších problémů s kolísáním zesílení zesilovače v závislosti na kmitočtu zpracovávaného signálu. Autorem patentovaného řešení byl *Harold Stephen Black*. H. Blacka napadlo, trdilo se že při ranní procházce po mostě přes řeku Hudson v roce 1927, přivést část výstupního signálu zesilovače zpět na vstup. Tímto opatřením došlo k výrazné stabilizaci zesílení zesilovače. Podrobnosti např. na <http://cs.wikipedia.org>.

Zpětnou vazbou rozumíme přivedení části výstupního signálu systému zpět na jeho vstup podle obr.2.31.



Obr.2.31. Signálová zpětná vazba systému.

Na obr.2.31 předpokládáme pro jednoduchost lineární systém v harmonickém ustáleném stavu (HUS). Symboly $\dot{K}$, $\dot{A}$ a $\dot{B}$ představují tzv. komplexní přenosy celého systému ($\dot{K}$) a jeho dvou dílčích částí ($\dot{A}$ a $\dot{B}$). Jsou definovány poměrem výstupního signálu k signálu vstupnímu. Signály zde představují harmonické průběhy, definované tzv. fázory, viz blíže v [12]. Tečkami nad symboly naznačujeme, že jde o reálné funkce komplexní proměnné. Původní systém s přenosem $\dot{A}$ je systém bez zpětné vazby. Z jeho výstupu je část signálu $\dot{S}_2$ přivedena na vstup přes tzv. *zpětnovazební subsystém* (člen, prvek) s přenosem $\dot{B}$. Vpravo vedle schématu je odvozeno, jak se původní přenos systému bez zpětné vazby $\dot{A}$ změní na $\dot{K}$ zavedením zpětné vazby díky $\dot{B}$. Subsystém $\dot{B}$ určuje, jaká část výstupního signálu a v jaké fázi se přičte ke vstupnímu signálu.

Dvojitě podtržený výsledný vztah je znám jako tzv. *Blackův vztah*. Udává nám komplexní přenos zpětnovazebního systému (systému se zpětnou vazbou). Ve vztahu figurují dílčí přenosy $\dot{A}$ a $\dot{B}$. Protože v této knize není cílem probírat celou teorii zpětné vazby, uvedme, jaké důsledky zavedení zpětné vazby na systém může vyvolat.

Příklad 2.1.

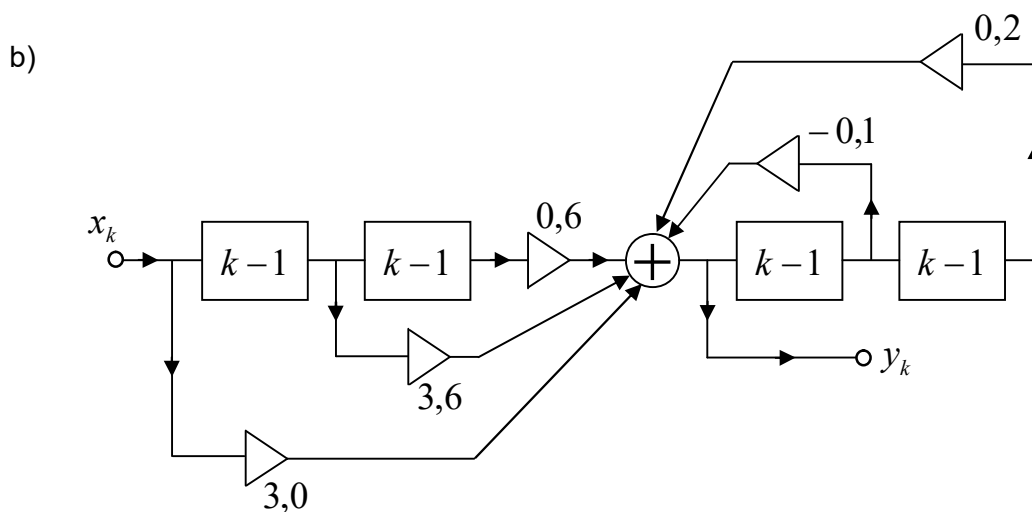
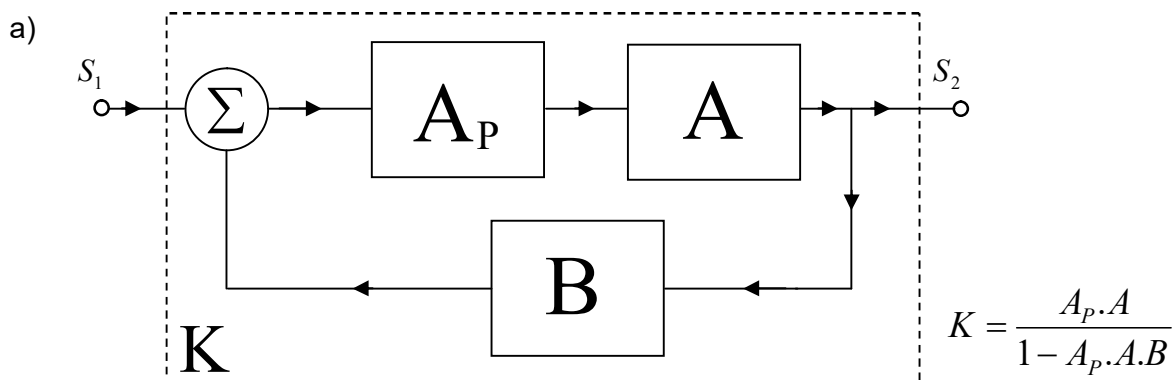
Vypočítejte celkové zesílení systému se zpětnou vazbou podle obr.2.31 je-li dáno:

$$A = 90[-], B_1 = 0,01[-] \text{ a } B_2 = -0,5[-].$$

$$\text{Řešení: } K_1 = \frac{A}{1 - A \cdot B_1} = \frac{90}{1 - 90 \cdot 0,01} = \frac{90}{0,1} = 900[-],$$

$$K_2 = \frac{A}{1 - A \cdot B_2} = \frac{90}{1 - 90 \cdot (-0,5)} = \frac{90}{46} = 1,96[-].$$

Poznámka: sami se zamyslete, který z výsledků představuje jaký typ zpětné vazby.



$$y_k = 3x_k + 3.6x_{k-1} + 0.6x_{k-2} - 0.1y_{k-1} + 0.2y_{k-2}$$

Obr.2.32. Zpětná vazba u analogových a) a číslicových systémů b).

2.4.2 Vnitřní (stavový) popis systémů

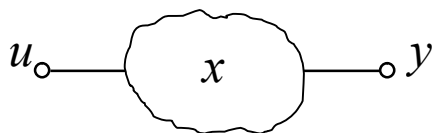
Vnitřní popis (model) systému předpokládá znalost vnitřní struktury systému tedy jeho subsystémů. Oproti vnějšímu popisu nás primárně zajímají vlastnosti signálů (veličin), které reprezentují výstupy vybraných subsystémů. Přitom je možné přejít z modelu vnitřního k modelu vnějšímu a tento přechod je vždy jednoznačný. Naopak, tedy přeměna vnějšího modelu na model vnitřní je cesta nejednoznačná, neboť jednomu vnějšímu popisu může obecně odpovídat nekonečně mnoho druhů vnitřní struktury.

2.4.2.1 Stavový model systému

Pro ilustraci uvedeme tři stavové modely různých systémů. Kromě vlastního zápisu rovnic bude zajímavá také jejich maticová podoba.

Příklad 2.2.

Jde o lineární dynamický systém s jedním vnitřním stavem (veličinou) x , jednou budicí veličinou u a jednou veličinou výstupní y . Symboly K_1 , K_2 jsou konstantami (číslly).

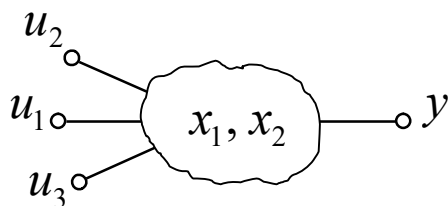


$$\begin{aligned} \frac{dx}{dt} &= K_1 x + K_2 y \\ y &= x \end{aligned} \quad \Rightarrow \quad \begin{aligned} \boxed{\frac{dx}{dt}} &= \boxed{K_1} \cdot \boxed{x} + \boxed{K_2} \cdot \boxed{u} \\ \boxed{y} &= \boxed{1} \cdot \boxed{x} + \boxed{0} \cdot \boxed{u} \end{aligned}$$

První rovnici modelu, ve které se vyskytuje derivace, budeme nazývat *rovnici dynamiky*. Zde je obsažen vlastní dynamický model chování systému. Druhá rovnice je nazývána *rovnici výstupu*, neboť specifikuje transformaci veličiny x na výstup systému (rovnice výstupu nemusí být vždy určena). Veličinu (signál) x budeme nazývat *stavovou veličinou* či stavovou proměnnou. Jak je vidět, rovnice dynamiky je diferenciální rovnici (dynamický systém), v daném případě 1. řádu, a je psána tak, aby na levé straně byla pouze derivace. Maticové vyjádření je získáno intuitivně přímo z rovnic.

Příklad 2.3.

Nyní jde o lineární dynamický systém se dvěma vnitřními stavy (veličinami) x_1 a x_2 , třemi budícími veličinami u_1 , u_2 a u_3 a jednou veličinou výstupní y .



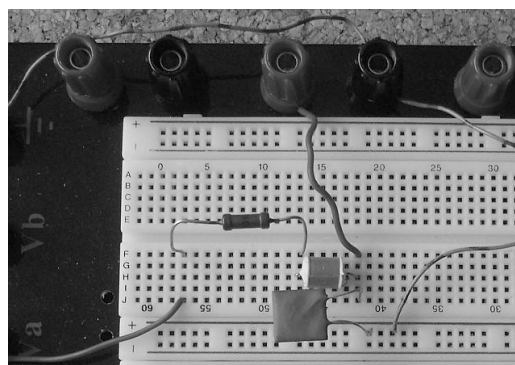
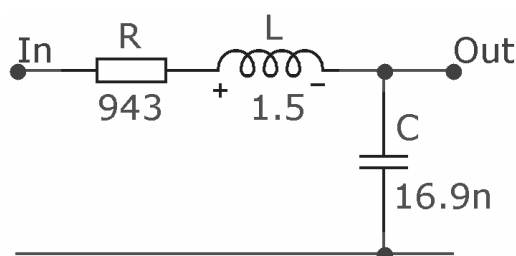
$$\begin{aligned} \frac{dx_1}{dt} &= K_1 x_1 + K_2 x_2 + K_3 (u_1 - u_2) \\ \frac{dx_2}{dt} &= K_4 x_1 + K_5 x_2 + K_6 (u_3 - u_2) \\ y &= K_7 x_2 + u_3 \end{aligned} \quad \Rightarrow \quad \begin{aligned} \boxed{\frac{dx_1}{dt}} &= \begin{bmatrix} K_1 & K_2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} K_3 & -K_3 & 0 \\ 0 & -K_6 & K_6 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \\ \boxed{y} &= \begin{bmatrix} 0 & K_7 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \end{aligned}$$

2.5.6 Realizace a implementace systémů

Realizací či implementací rozumíme postupy uvedení něčeho do praxe. Ve slovnících a databázích můžete nalézt např. tvrzení, že „... *Implementace je uvedení teoreticky stanovené myšlenky do praxe za účelem jejího dalšího použití. Implementaci předchází plánování postupu a očekávaných výsledků*“, viz <http://cs.wikipedia.org/wiki/Implementace>.

V širším smyslu jde tedy o praktické využití. V případě analogových systémů hovoříme spíše o realizaci, u číslicových o implementaci. Je zde však ještě další, možno napsat užší chápání uvedených termínů. V analogové technice se pod pojmem *realizace systému* může rozumět nejen jeho skutečné vytvoření, ale také navržení modelu k praktické realizaci přímo směřující. Jako příklad z našeho oboru – elektrotechniky – můžeme uvést analogové kmitočtové filtry. Zatímco návrh filtru končí známou přenosovou funkcí, o realizaci hovoříme v případě známého schématu zapojení součástek spolu s jejich konkrétními hodnotami a parametry. Fyzická tvorba jakožto logické završení procesu realizace je samozřejmě možná, viz obr.2.77.

$$K(s) = \frac{3,95 \cdot 10^7}{s^2 + 6,28 \cdot 10^2 s + 3,95 \cdot 10^7}$$



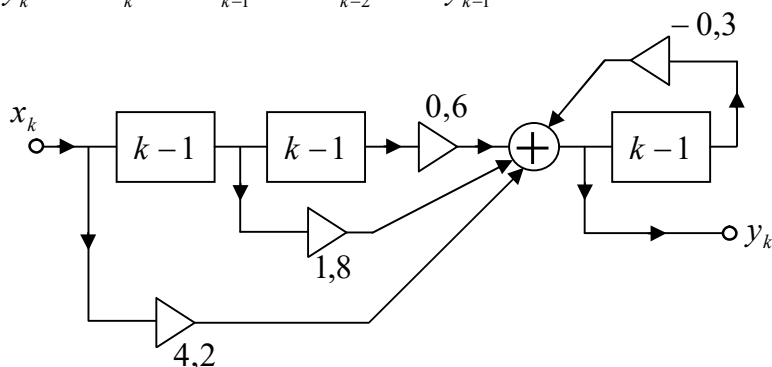
Obr.2.77. Práce s analogovým systémem od návrhu po realizaci.

Samozřejmě že *není realizace jako realizace*. Vždy je nutné přihlédnout k řadě aspektů jako např. ke kmitočtovému rozsahu buzení (parazitní prvky a jevy), kvalitě, velikosti a ceně použitých součástek (subsystémů), způsobu jejich propojení a vzájemného vlivu, typu prostředí (teplota, vibrace, prach), vlastní způsob využití systému (výuka, průmyslová aplikace) aj. Je zřejmé, že tento proces vyžaduje jistou zkušenost.

V případě číslicových systémů je proces podobný tomu na obr.2.77. Zde také vycházíme z výsledku návrhu, kterým může být přenosová (systémová) funkce, diferenční rovnice, blokové schéma aj. Jak jsem však psal výše, číslicový systém je algoritmem, tedy myšlenkovým procesem. Tento proces není sám o sobě hmotně fyzický, nýbrž spíše informačního typu. Pro jeho praktické využití je však nezbytné, aby byl vhodným způsobem realizován. Praktická realizace číslicového algoritmu je nazývána *implementací*. Implementovat číslicový algoritmus znamená realizovat jej ve vhodném programovacím jazyku a s využitím vhodného hardware (stolní či přenosný počítač, jednočipový

mikropočítač, programovatelné hradlové pole, logické obvody aj.). V současné době je často dáována přednost programovatelným implementacím z důvodů snadné změny algoritmu.

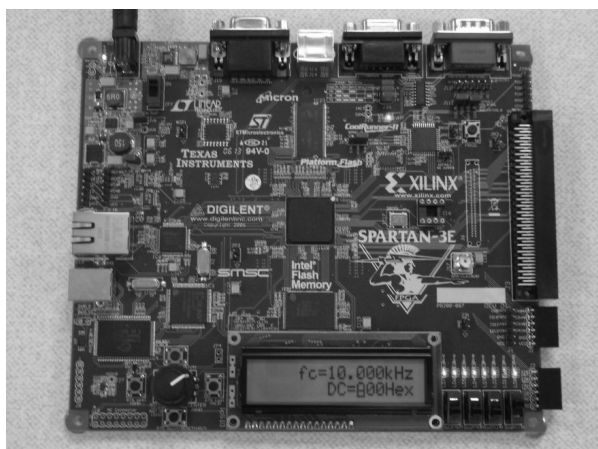
$$y_k = 4.2x_k + 1.8x_{k-1} + 0.6x_{k-2} - 0.3y_{k-1}$$



```

N = length(x); % počet vzorků vst. signálu x
x1=0; % počáteční podmínky  $x_{k-1}$ 
x2=0; % počáteční podmínky  $x_{k-2}$ 
y1=0; % počáteční podmínky  $y_{k-1}$ 

for k=1:N % cyklus od 1 do N
    y(k)=4.2*x(k)+1.8*x1+0.6*x2-0.3*y1;
    x2=x1;
    x1=x(k);
    y1=y(k);
end % konec cyklu
    
```



Obr.2.78. Práce s číslicovým systémem od návrhu po realizaci.

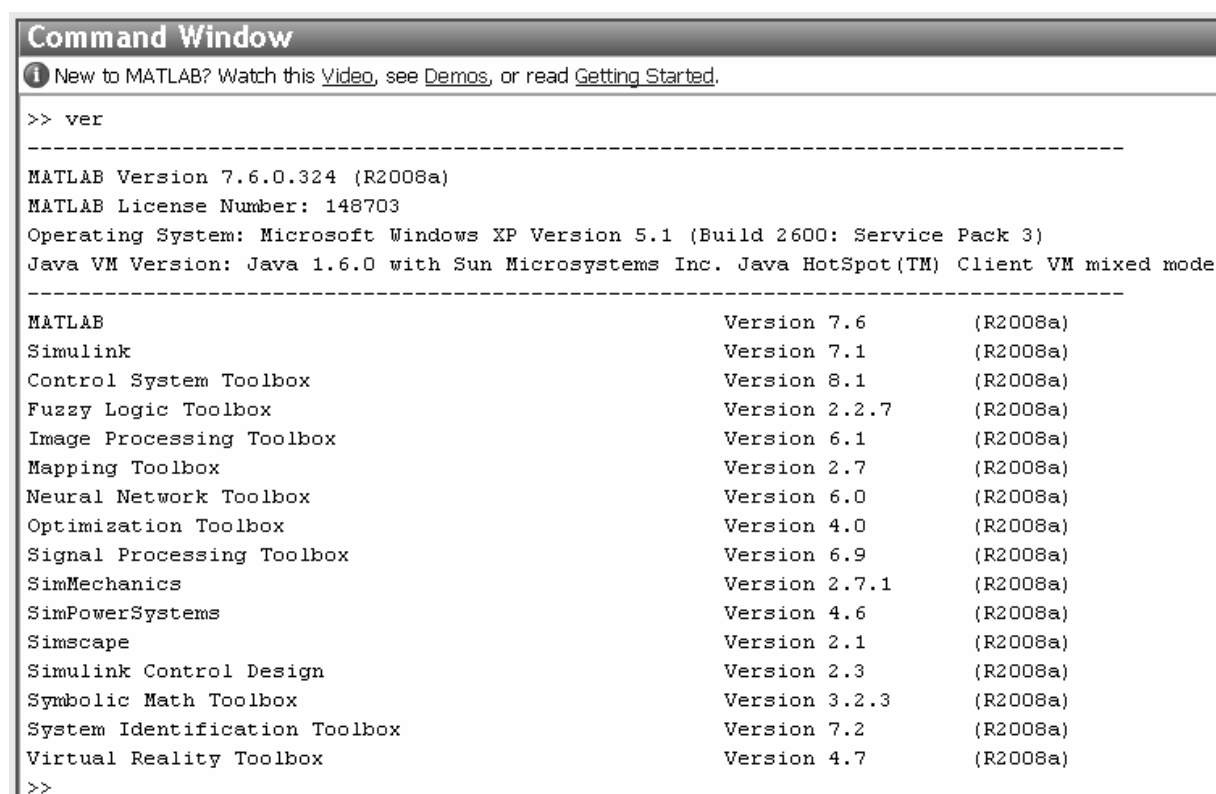
4 Stručný úvod do systému Simulink

V kapitole 3.3 jsem představil systém Simulink jakožto graficky orientovanou nadstavbu vlastního MATLAB. Systémy se zde tvoří pomocí bloků, které jsou obsaženy v jejich tématicky uspořádané knihovně. Stručně napsáno, pracuje se především pomocí myši; bloky se z knihovny přetahují do tzv. modelu, pospojují podle logiky práce a po zadání maximálního času lze spustit simulaci s vhodně nastaveným tzv. ODE solverem, viz kapitola 2.4.1.2. Je zřejmé, že je-li potřeba, lze editovat celou řadu parametrů a měnit nastavení. Při prvních pokusech však doporučuji ponechat implicitní nastavení a věnovat se podstatě věci. Simulace systémů v časové doméně je založena na numerickém řešení sestavených obecně nelineárních diferenciálních rovnic. To vše vykonává systém skrytě. Chybová hlášení či varování slouží k řádnému odladění algoritmu.

V této kapitole chci být maximálně stručný a nezatěžovat čtenáře sice zajímavými, avšak zpočátku nadbytečnými informacemi. Jako i při jiné práci, budeme postupovat od jednoduššího ke složitějšímu. Cílem je předložit vám podstatu věci a motivovat vás k samostatné práci, vlastní tvorbě a pokusům. V rámci pravidelných aktualizací se některé dílčí položky mohou pozměnit. Buďte proto pružnými čtenáři a uživateli. Zde půjde hlavně o to pochopit princip a základní možnosti.

4.1 Spuštění systému Simulink

Aby bylo možné Simulink spustit, je třeba jej mít nainstalován a tedy legálně vlastnit jeho licenci. O tom, zda je Simulink součástí vaší instalace se lze nejsnáze přesvědčit již známým příkazem `ver`, viz obr.4.1, resp. obr.1.1.



```
Command Window
i New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> ver

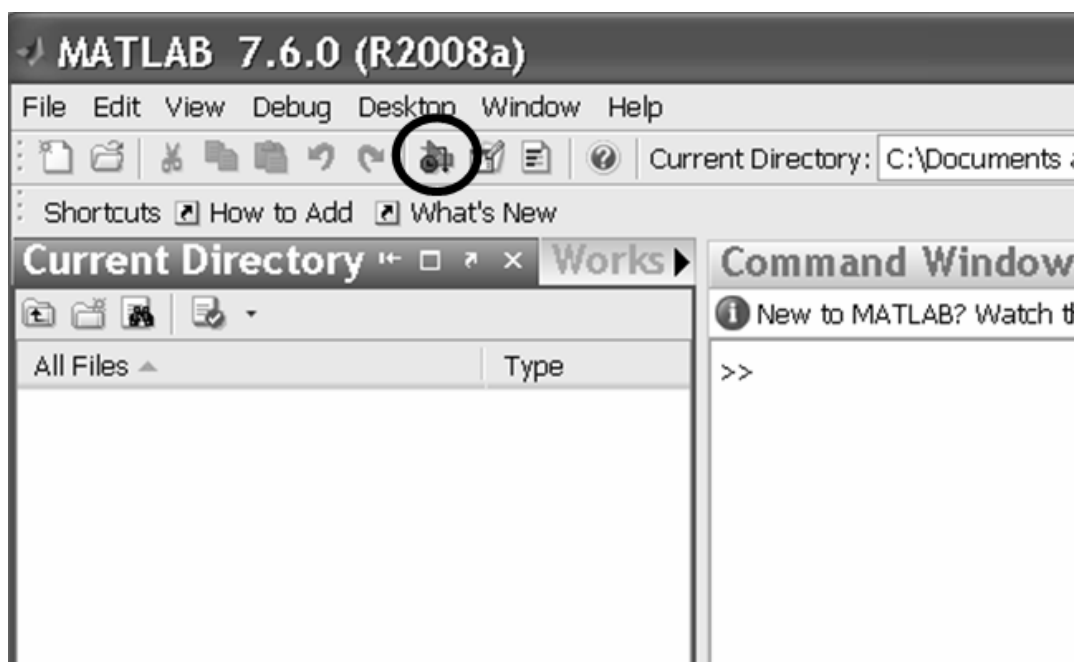
-----
MATLAB Version 7.6.0.324 (R2008a)
MATLAB License Number: 148703
Operating System: Microsoft Windows XP Version 5.1 (Build 2600: Service Pack 3)
Java VM Version: Java 1.6.0 with Sun Microsystems Inc. Java HotSpot(TM) Client VM mixed mode
-----

MATLAB                               Version 7.6           (R2008a)
Simulink                             Version 7.1           (R2008a)
Control System Toolbox               Version 8.1           (R2008a)
Fuzzy Logic Toolbox                 Version 2.2.7         (R2008a)
Image Processing Toolbox             Version 6.1           (R2008a)
Mapping Toolbox                     Version 2.7           (R2008a)
Neural Network Toolbox              Version 6.0           (R2008a)
Optimization Toolbox               Version 4.0           (R2008a)
Signal Processing Toolbox           Version 6.9           (R2008a)
SimMechanics                       Version 2.7.1         (R2008a)
SimPowerSystems                    Version 4.6           (R2008a)
Simscape                           Version 2.1           (R2008a)
Simulink Control Design             Version 2.3           (R2008a)
Symbolic Math Toolbox               Version 3.2.3         (R2008a)
System Identification Toolbox       Version 7.2           (R2008a)
Virtual Reality Toolbox            Version 4.7           (R2008a)

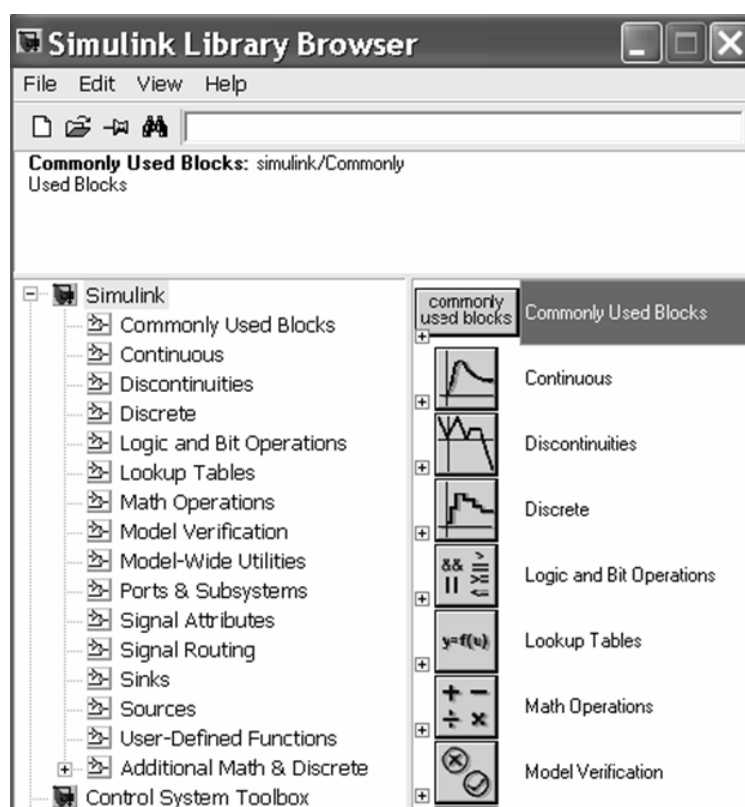
>>
```

Obr.4.1. Dotaz na obsah licence systému MATLAB.

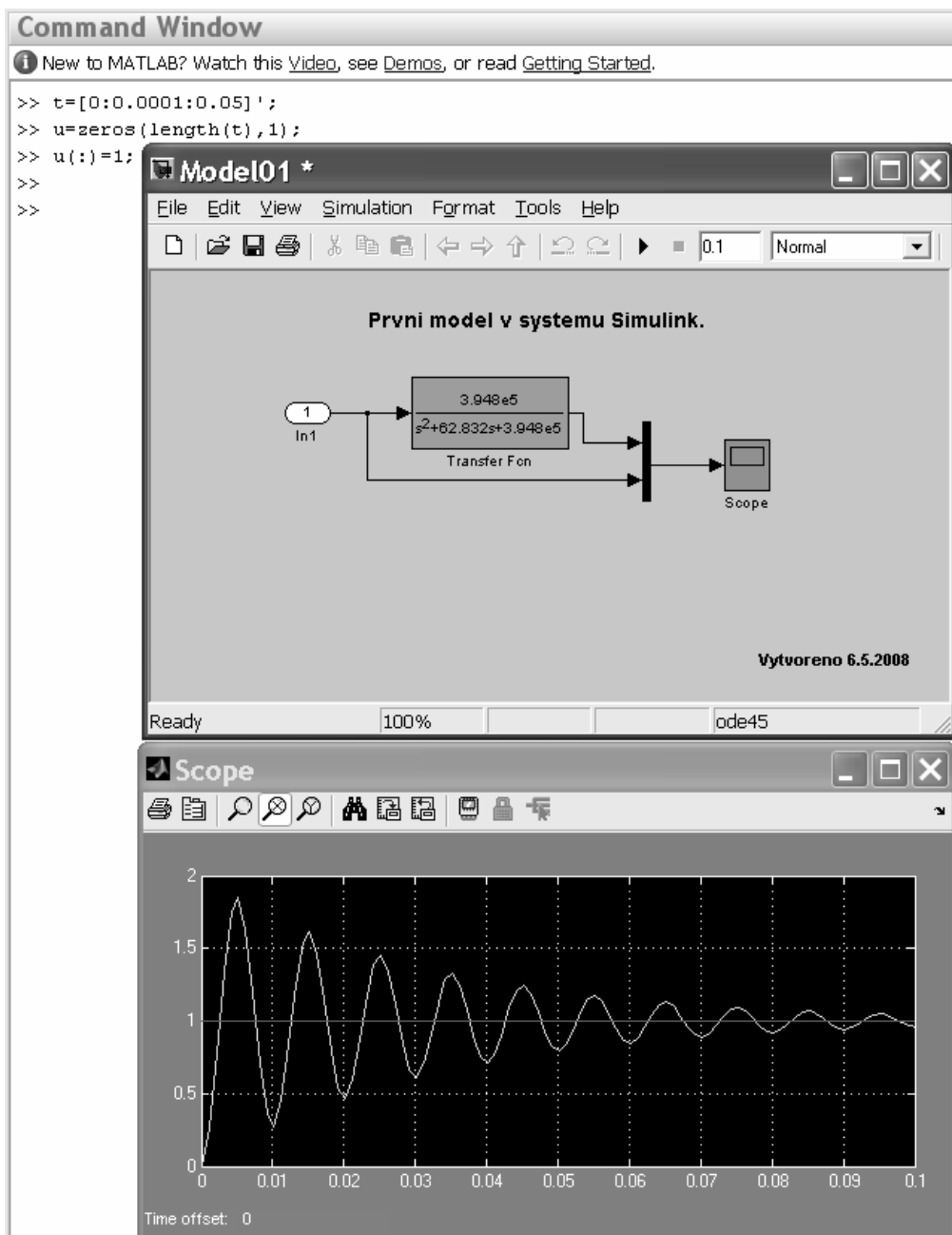
Je-li Simulink nainstalován, spustíte jej jednoduše myší díky tlačítku v levém horním rohu prostředí MATLAB, viz obr.4.2.



Obr.4.2. Tlačítko pro spuštění systému Simulink.



Obr.4.3. Část nabídky po spuštění systému Simulink.



Obr.4.28. Import dat z *Workspace* s využitím bloku *In*.

5 Příklady práce se systémy

V této kapitole uvedu sbírku vybraných příkladů práce se systémy. Vzhledem k názornosti budu preferovat modely v prostředí Simulink. Je třeba připomenout, že řadu věcí lze v MATLAB udělat více způsoby a různými technikami. Je proto pochopitelné, že si předložené a další příklady můžete řešit vlastními postupy (místo modelů v Simulink napsat m-soubor, namísto vnitřních funkcí si napsat vlastní, realizovat *Callback* různým způsobem atp.). Mimoto, sám pracuji v oblasti elektrotechniky a elektroniky a proto si ne zcela detailně umím představit mnohost různých variant systémů ve vašich oborech. Jak jsem uvedl při diskusi o číslicových systémech, pojem systém již dnes není chápán pouze jako fyzicky existující zařízení. Může jít o myšlenkový postup, který si potřebujete ověřit. Z principu je to typické např. u finančních systémů. V MATLAB však vždy pracujeme s čísly, byť se někdy snažíme o napodobení analogových soustav, jak jsem uvedl. Rozlišení druhu systému je však třeba respektovat při volbě matematického modelu (rovnice diferenciální a diferenční, statické a dynamické, lineární a nelineární, deterministické a stochastické atd.).

V dále uvedených příkladech se zaměřujeme na podstatu věci. Nebudu se již věnovat tvorbě *Callback* a doprovodných m-souborů, pokud to nebude mít souvislost s principem. V případě vaší potřeby není jistě problém si příklady obohatit a rozšířit.

5.1 Analogové systémy

Rozdělit systémy lze podle mnoha různých hledisek, jak jsem uvedl výše. Na základě zkušeností s prací s modely systémů se mně však jeví vhodné jako primární rozdělit systémy na analogové a číslicové. Důvodů je více. Mezi ty hlavní patří principiální odlišnosti matematických modelů, respektování spojitého a diskrétního charakteru signálů a dějů, diskuse o vzájemných souvislostech či transformacích systémových modelů a primárně časově orientované vnímání světa člověkem. V prostředí MATLAB lze pracovat s oběma hlavními druhy systémů současně, tedy s tzv. *smíšenými systémy*. Práce s nimi však vyžaduje jistou zkušenost.

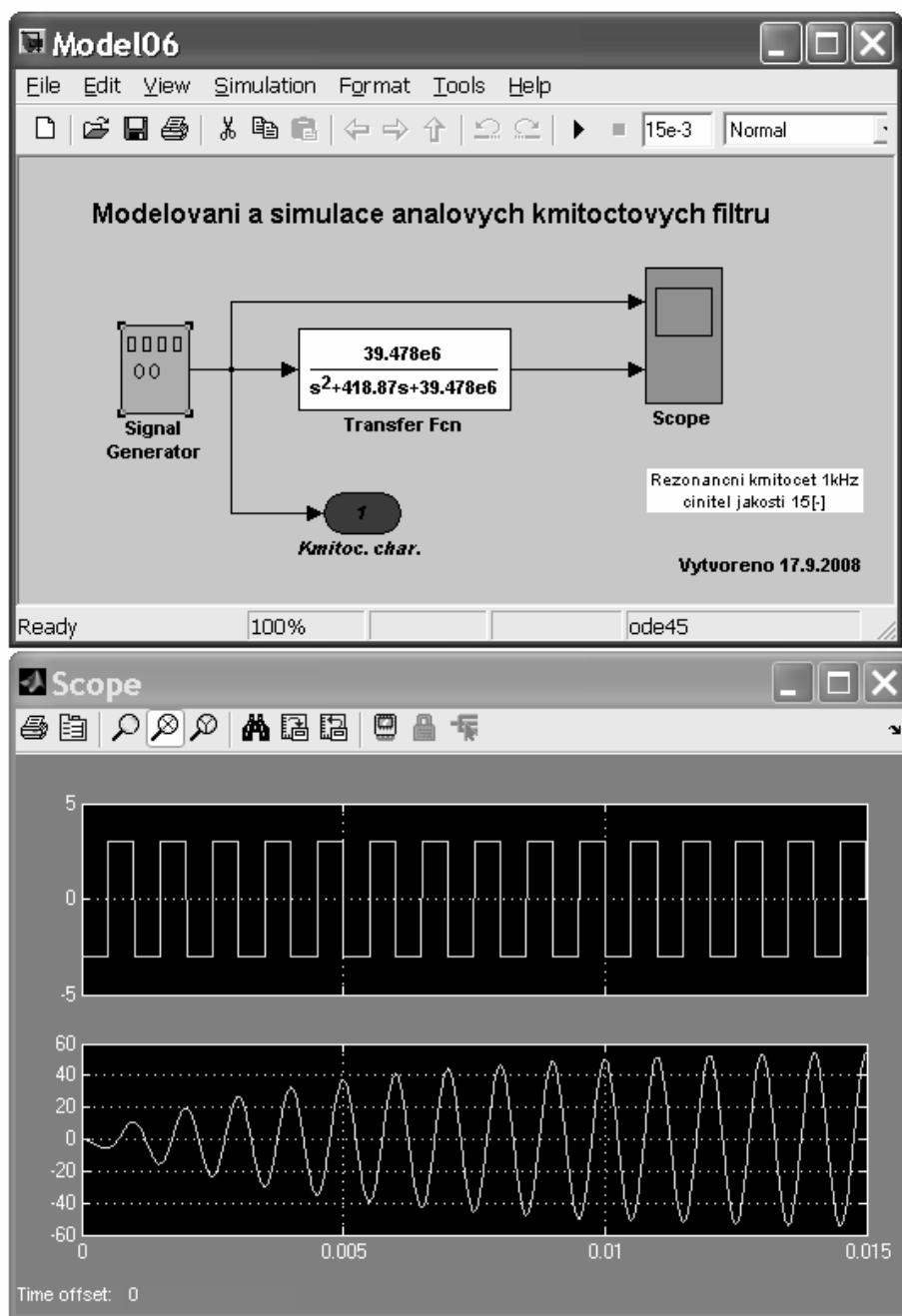
5.1.1 Kmitočtové filtry

Analogové kmitočtové filtry (AKF) patří mezi nejpoužívanější analogové systémy. Jsou *lineárními dynamickými systémy*, takže pro ně platí vše výše uvedené, např. v kapitolách 2.3 a 2.4. Na těchto systémech si přiblížíme práci s podobnými systémy. Analogové filtry jsou používány přibližně sto let. Je zřejmé, že v současné době budou velmi dobře zvládnuté metody jejich analýzy, návrhu, příp. optimalizace a fyzické realizace. U nás i ve světě existuje množství kvalitní literatury, zmiňme alespoň [3], [4] a [8]. Pro práci s modely AKF jsou často využívány speciální programy, avšak i v systému MATLAB lze s nimi na jisté úrovni pracovat. Je však pravdou, že MATLAB je specializován zejména na filtry číslicové a ty analogové nemají zdaleka takovou podporu.

5.1.1.1 Modelování filtrů a jejich analýza

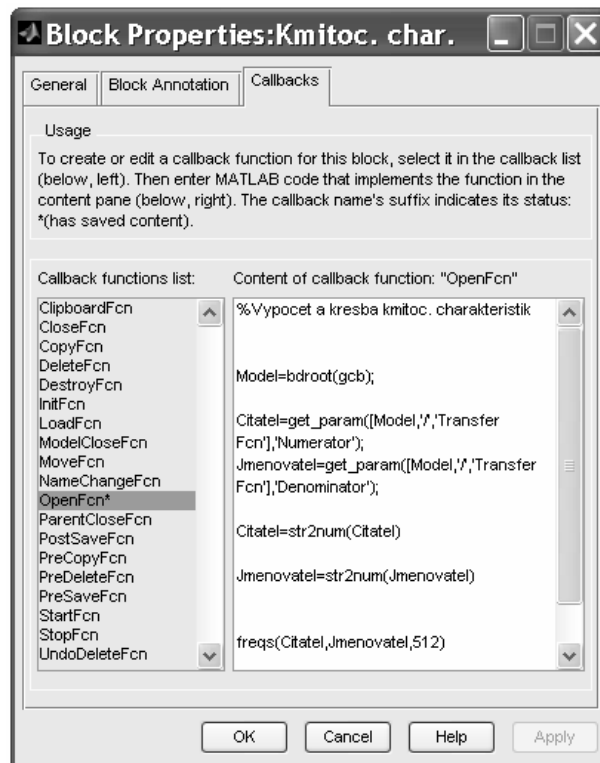
Analogové kmitočtové filtry (AKF) jsou, jak jsem zmínil, systémy lineárními a dynamickými. K jejich popisu se nejčastěji používají *přenosové funkce* jakožto vstupně-výstupní (vnější) model, viz kapitola 2.4.1.3. Vlastní prostředí MATLAB lze využít k modelování *kmitočtových charakteristik*, tedy v kmitočtové rovině, blíže v kapitole 2.3.2.2. Simulace AKF v rovině časové, tedy získání časových průběhů sledovaných signálů, je naopak snazší a z principu vhodnější v prostředí Simulink. Obě prostředí lze tedy s výhodou kombinovat a vzájemně provazovat.

Protože práce s přenosovými funkcemi byla prakticky uvozena v kapitole 2.4.1.3, uvedu příklad, kombinující obě zmíněná prostředí. Na obr.5.1 je ukázán model s AKF, modelovaným pomocí bloku *Transfer Fcn* (zadání parametrů diskutováno výše).



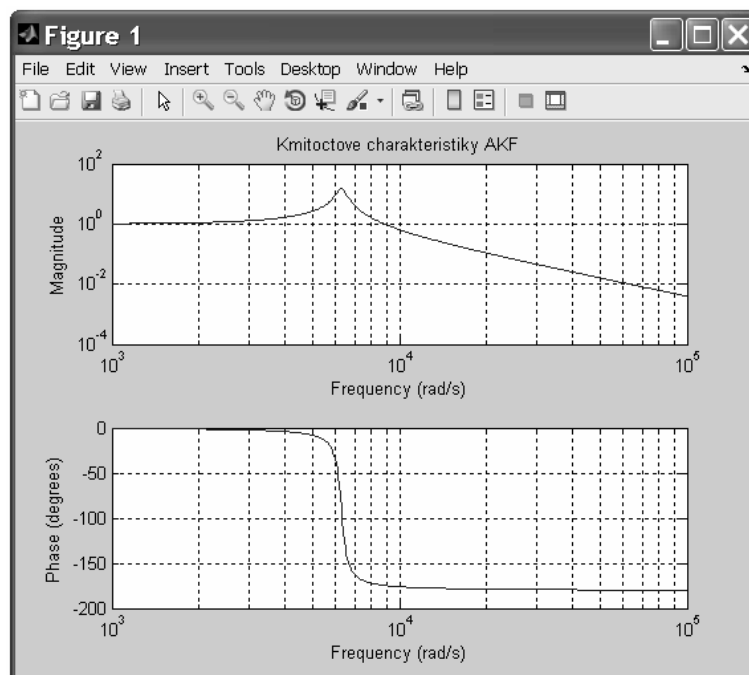
Obr.5.1. Modelování a simulace analogového filtru.

Jedná se o filtr 2. řádu s rezonančním kmitočtem $f_r = 1\text{kHz}$ a činitelem jakosti $Q = 15[-]$. Filtr je buzen z generátoru obdélníkových periodických pulzů s velikostí $3V$ (signál pokládáme za napětí) a kmitočtem 1kHz (nastavení po dvojkliku na blok *Signal Generator*). Kromě osciloskopu *Scope* obsahuje model již jen blok s názvem *Kmitoc. char.* Ten je výše popsáným způsobem nastaven tak, že po dvojkliku myši na něj dojde k vykreslení kmitočtových charakteristik filtru. Model *Model06* tedy spojuje simulaci v časové doméně spolu s modelováním v doméně kmitočtové. Časové průběhy vstupu a výstupu zajistí sám Simulink (maximální čas simulace je nastaven na 15ms, jak je vidět v horní části modelu). O vykreslení kmitočtových charakteristik se musíme postarat sami a to tvorbou *Callback*, viz kapitola 4.10.2. Obsah pole pro definici *Callback* ukazuje obr.5.2. Připomínám, že uvedené pole získáte tak, že označíte blok *Kmitoc. char.* a z menu modelu vyberete položky *Edit/Block Properties...* a následným výběrem položek *Callbacks* a *OpenFcn**.



Obr.5.2. Zdrojový text Callback pro kreslení kmitočtových charakteristik filtru.

Uvedeným postupem vytvoříte *Callback* bloku *Kmitoc. char.* Vše uložte, spusťte simulaci a poté dvakrát klikněte na onen blok. Dojde k zobrazení kmitočtových charakteristik daného filtru, viz obr.5.3. Význam příkazů by neměl činit potíže. Pomocí *get_param* se zjišťují parametry bloku *Transfer Fcn*.



Obr.5.3. Kmitočtové charakteristiky filtru.

5.2 Číslicové systémy

Číslicové systémy mají v prostředí MATLAB podstatně větší podporu, než jejich analogové ekvivalenty. Je to pochopitelné, vzhledem k zaměření celého systému a soudobým trendům, preferujícím digitalizaci prakticky všude kolem nás. Aplikační možnosti číslicových systémů jsou značně stejně jako univerzálnost použití. Zatímco analogové systémy lze prakticky provozovat pouze na základě reálných zařízení, číslicové systémy mají tu přednost, že jsou vlastně *algoritmy*, tedy myšlenkovými pochody. A ty lze realizovat, píšeme spíše implementovat, na nejrozličnějším hardware od univerzálních mikropočítačů, přes PC až ke specializovaným signálovým procesorům a hradlovým polím. V poslední době jsme také svědky jisté integrace analogových a číslicových systémů na jedné platformě. Hovoříme o *smíšených systémech* (pole, sdružující digitální i analogové stavební bloky).

K nejrozšířenějším číslicovým systémům náleží kromě modulačních a kódovacích systémů číslicové filtry. Jsou poměrně dobře propracovány jak po stránce teoretické, tak aplikační (implementační). Kromě rychlosti jsou málo závislé na hardware a umožňují snadnou modifikaci přeprogramováním. Proto se v této kapitole zaměříme zejména na ně.

5.2.1 Lineární číslicové filtry

Při práci s číslicovými filtry (ČF) je třeba, analogicky filtrům analogovým, jistou sumu teoretických znalostí a zkušeností. Na stránkách této knihy se jimi nemůžeme blížeji věnovat. Naštěstí o kvalitní literaturu není nouze jak ve světě kolem nás, tak v ČR, viz např. [3], [5], [11], [17], [18] a [22]. Některé základy jsem nastínil v kapitolách 2.3.1.2 a 2.5.7. V této knize se proto zaměříme na základní přístup a praktickou realizaci tak, abyste pak již mohli dále své znalosti a zkušenosti relativně sami rozšiřovat a zkvalitňovat.

Číslicový filtr (ČF) je, jak jsem naznačil výše, algoritmus, pracující s čísly. Jeho úkolem je *načítat čísla z paměti (proměnné), podrobit je jistým úpravám a poslat je na svůj výstup*, tedy opět do nějakého paměťového úložiště (registru, proměnné apod.). Zpravidla se tak děje v určitých pravidelných krocích či intervalech (vzorcích), které jsou řízeny oscilátorem hardwarových zařízení (hodinový či vzorkovací kmitočet). Tyto filtry jsou systémy dynamickými, ale z principu mohou být systémy lineárními i nelineárními na rozdíl od analogových filtrů. My si zde budeme všimnout ČF *lineárních* a to z důvodů jejich rozšíření a menších nároků na předchozí znalosti. Za již delší dobu jejich používání bylo vypracováno poměrně velké množství nejrozličnějších přístupů k jejich návrhu a implementaci (praktickému nasazení, tedy filtrování číslicových dat). Je třeba si zvyknout na jeden důležitý fakt a sice to, že při práci s ČF (ale nejen s nimi) bude hrát často důležitou roli *vzorkovací kmitočet* nebo *perioda*. Je to typické především, ale nejen, u těch ČF, které pracují s čísly, jež představují *vzorky* původně analogových signálů. Při číslicovém zpracování (DSP - Digital Signal Processing) analogových dat je třeba respektovat některé důležité vlastnosti těchto signálů a těm návrh ČF přizpůsobit.

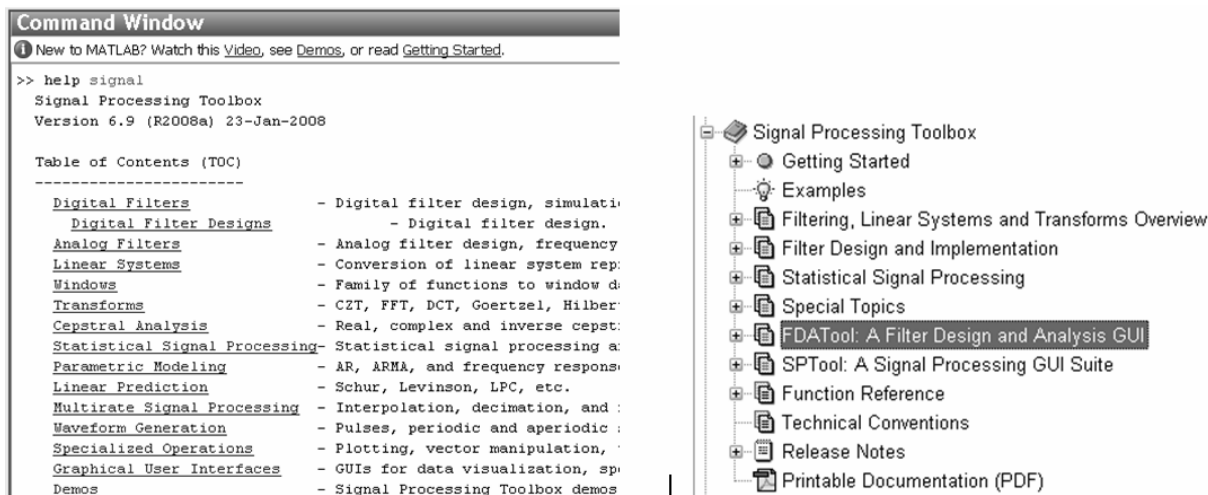
5.2.1.1 Modelování filtrů a jejich analýza

Nástroje k základní práci s číslicovými filtry (ČF) jsou soustředěny především v knihovnách *Signal Processing ToolboxTM* (vlastní MATLAB) a *Signal Processing BlocksetTM* (Simulink). Další specializované knihovny jsou zaměřeny na zpracování obrazů, videa, na oblast komunikační techniky apod. Pro práci s filtry existují navíc zvláštní balíky funkcí *Filter Design ToolboxTM* a *Filter Design HDL CoderTM*. Tyto knihovny však nevlastníme. I tak lze ale s ČF pracovat, avšak nezbytnou výbavou je alespoň zmíněný *Signal Processing Toolbox*.

Pracovat s ČF lze nejméně dvěma způsoby:

- ručně psát příslušné příkazy s požadovanými parametry,
- využívat interaktivní podporu (práce s myší a menu zvláštních oken).

Na obr. 5.13 je ukázka nabídky systému na dotaz *help signal* a hlavní struktury knihovny *Signal Processing Toolbox* po jejím zobrazení v interaktivním helpu. Zkuste si dále rozbalit dílčí podnabídky, např. *Digital Filters* nebo *FDATool*. Možnosti jsou poměrně široké.



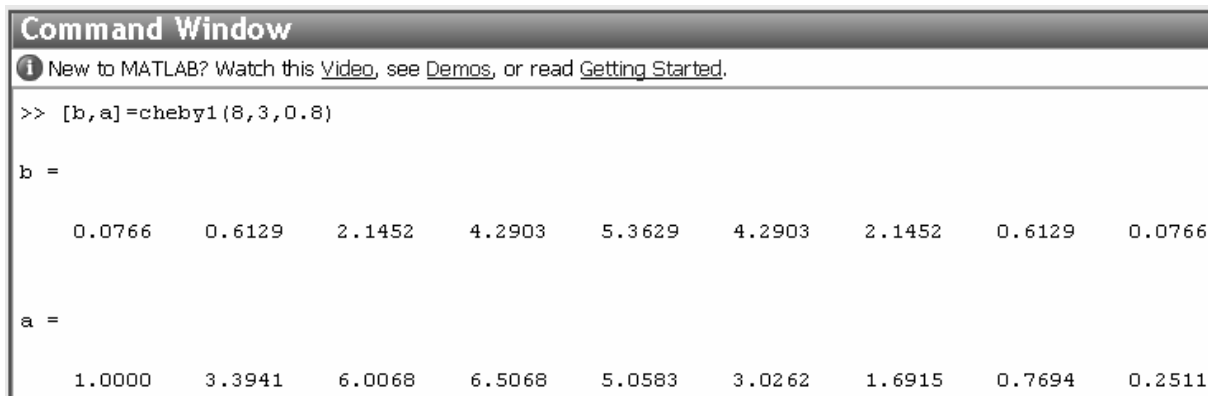
Obr.5.13. Základní nabídka knihovny *Signal Processing Toolbox*.

V této kapitole si blíže všimneme analýzy lineárních ČF. Filtr bude zadán svou přenosovou (systémovou) funkcí analogicky svému analogovému protějšku. Na rozdíl od něj však přenosová funkce nyní nebude mít souvislost s Laplaceovou transformací, viz kapitolu 2.4, ale s transformací Z . Transformací Z se nebudeme blíže zabývat. Postačí informace, že lineární ČF je definován koeficienty (nebo nulovými body a póly) své přenosové funkce, jež je opět racionální funkcí lomenou. Nezávisle proměnná těchto funkcí souvisí právě se zmíněnou Z transformací. Blíže např. v [3], [5] a [18].

Mezi *typické výstupy analýzy ČF* náleží zejména:

- kmitočtové charakteristiky (také číslicový algoritmus má kmitočtově závislý přenos),
- průběh impulzní odezvy (reakce na diskretní jednotkový impulz),
- odezva na vybrané budící signály (posloupnosti vzorků na vstupu),
- rozložení nulových bodů a pólů v Gaussově komplexní rovině, resp. rovině Z (stabilita),
- další možné průběhy.

Uvažujme číslicový filtr (ČF) 8. řádu. Tento filtr je definován svou přenosovou (systémovou) funkcí ve formě jejích koeficientů, viz obr.5.14.



Obr.5.14. Koeficienty čitatele a jmenovatele přenosové funkce ČF 8. řádu.

Na obrázku výše vidíte použití funkce *cheby1*, kterou jsem již aplikoval v kapitole 5.1.1.2 při návrhu analogového filtru. Není-li v rámci jejich parametrů použit symbol 's', pak jde vždy o návrh ČF. Přenosová funkce navrženého ČF vypadá takto:

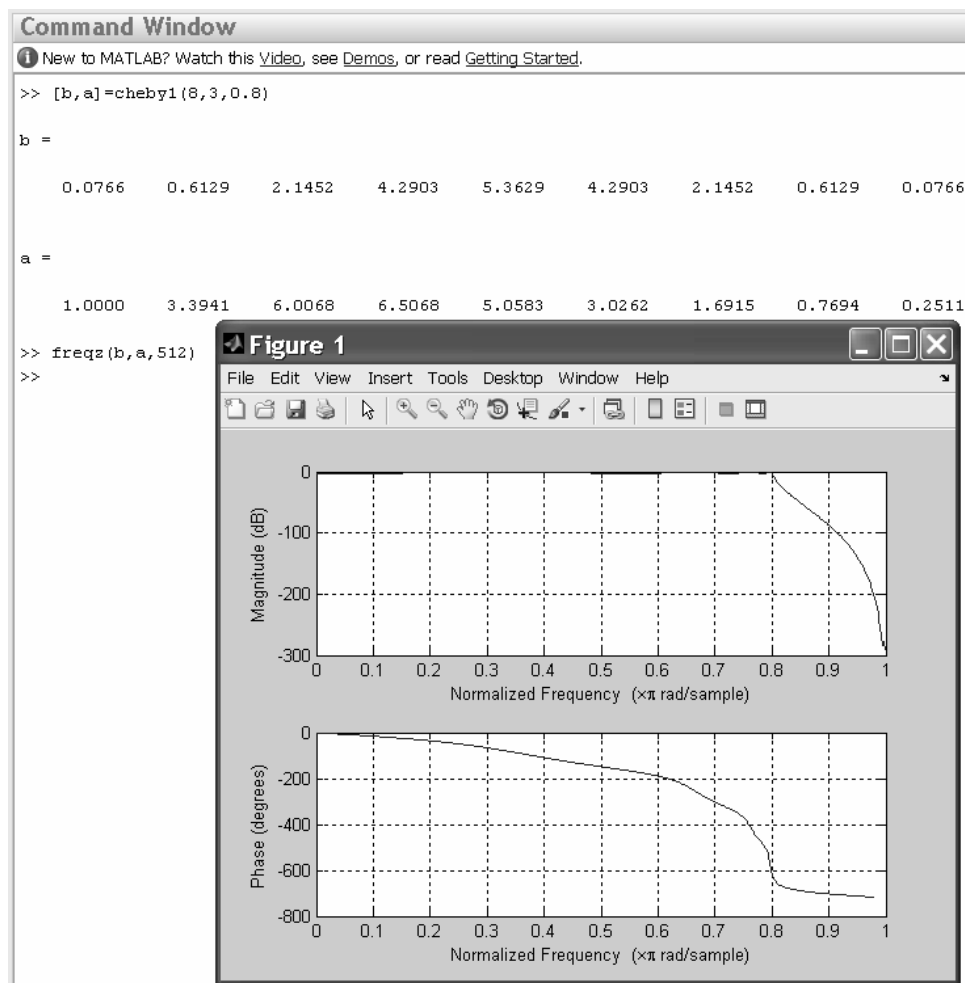
$$K(z) = \frac{0.0766 + 0.6129z^{-1} + 2.1452z^{-2} + 4.2903z^{-3} + \dots + 0.0766z^{-8}}{1 + 3.3941z^{-1} + 6.0068z^{-2} + 6.5068z^{-3} + \dots + 0.2511z^{-8}}. \quad (5.1)$$

Nezávisle proměnnou je v případě ČF operátor Z transformace, psaný s malým z . Platí:

$$z = e^{j\omega T_{vz}}, \quad (5.2)$$

kde T_{vz} je vzorkovací perioda. Jako parametry ČF z obr.5.14 dosazujeme řád filtru (zde 8.), dále zvlnění modulové kmitočtové charakteristiky v propustném pásmu v [dB], v našem případě -3dB. Třetím parametrem je mezní kmitočet filtru zadáný v intervalu $\langle 0,1 \rangle$. Je-li tento kmitočet zadán na hodnotu 1, pak odpovídá polovině vzorkovacího kmitočtu (Nyquistův kmitočet). Je třeba si zvyknout, že ČF se navrhují relativně vzhledem k uvažované vzorkovací frekvenci. Na *obr.5.15* je výpočet a zobrazení kmitočtových charakteristik filtru. Je využito vnitřní funkce *freqz* s analogickým použitím jako *freqs*, viz kapitola 2.4.1.3. Parametry příkazu *freqz* jsou číselník a jmenovatel přenosové funkce a počet bodů, ve kterých je charakteristika vypočtena a zobrazena.

Na základě vztahu (5.2) můžeme konstatovat, že kmitočtové charakteristiky ČF jsou periodickými funkcemi, které se opakují s periodou T_{vz} . Toto opakování si můžeme zobrazit, pokud necháme kmitočtové charakteristiky vypočítat a zobrazit nikoli do poloviny, nýbrž do celé hodnoty vzorkovací frekvence podle obr.5.16. Charakteristiky se jakoby periodicky zrcadlí, důkazy viz nabízené publikace.

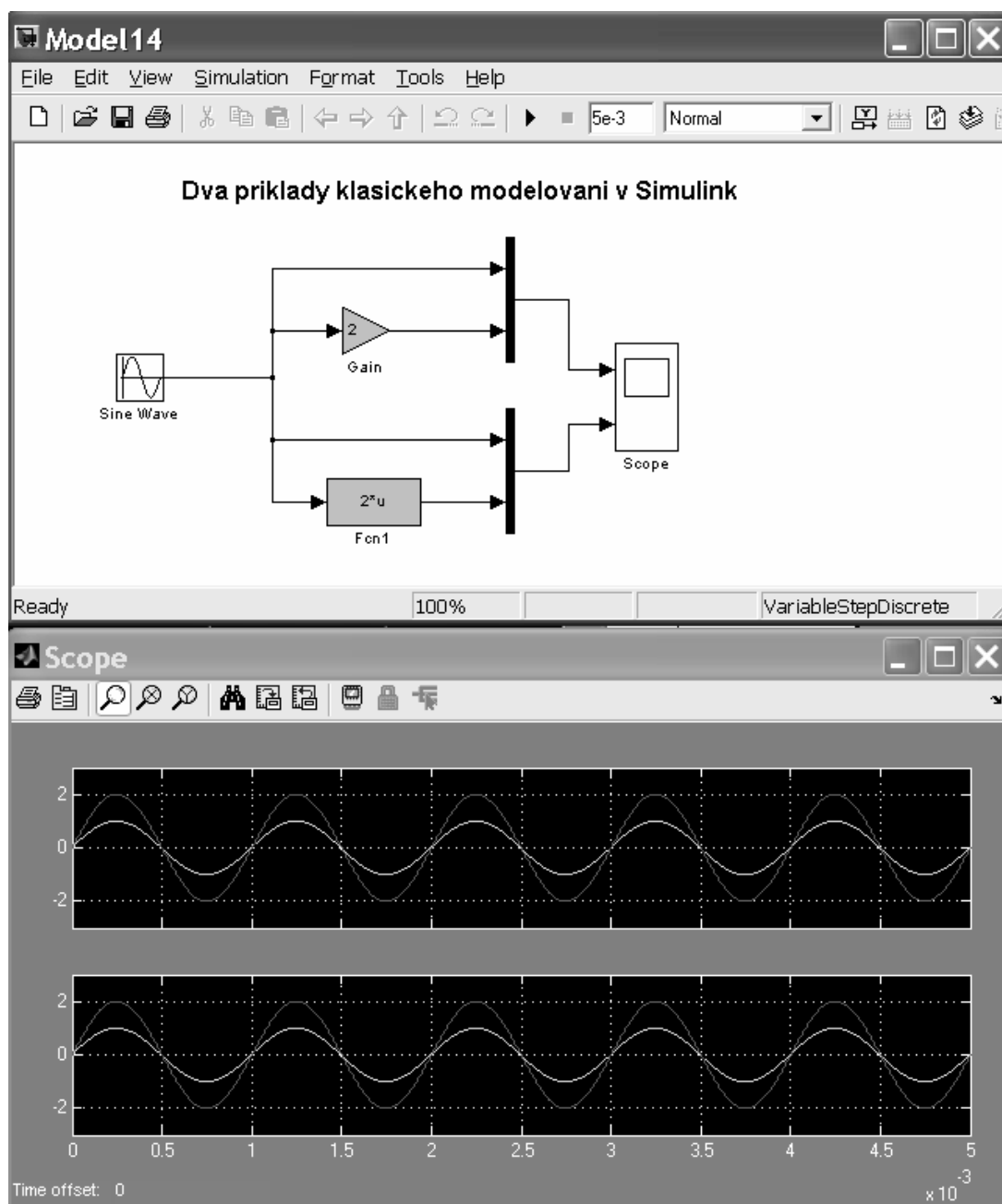


Obr.5.15. Získání kmitočtových charakteristik ČF z obr.5.14.

5.3 Poznámka k fyzikálnímu modelování

Jednou z knihoven systému MATLAB, zobrazených na obr.1.1, je i *Simscape*TM. Jedná se o rozšíření systému Simulink a nabízí modelování a simulaci systémů z různých oborů, tzv. *multidomain* systémů. Uživatel může pracovat s bloky systémů elektrických, mechanických, hydraulických, tepelných aj. Práce s tímto tzv. *fyzikálním modelováním* má svá specifika. Především jde o práci na úrovni součástek (subsystémů, prvků), ze kterých se skládá celý složitější systém. Např. v našem oboru, elektrotechnice, jde vlastně o práci na úrovni schématu zapojení, kdy jsou k dispozici jak vlastní topologie obvodu (logika zapojení), tak hodnoty parametrů součástek. Uživatel si z knihovny *Simscape* vybírá součástky (prvky) a zapojuje je podle logiky práce. Zajištění vlastní simulace se rovněž odlišuje od běžné práce s bloky v prostředí Simulink. Pro styk prvků *Simscape* s běžnými bloky Simulink jsou k dispozici zvláštní konvertory *SPS* a *PSS*. Simulace je zajištěna díky bloku *Solver Configuration*, jenž musí být v každém modelu obsažen. Signály, které uživatel požaduje k zobrazení (výpočtu), resp. které si přeje do modelu přivést jako zdrojové z běžných bloků, jsou přístupné díky tzv. *senzorům* (v elektrotechnice senzory napětí a proudů). Tvorba modelu se tak v některých rysech odlišuje od běžné práce s modely.

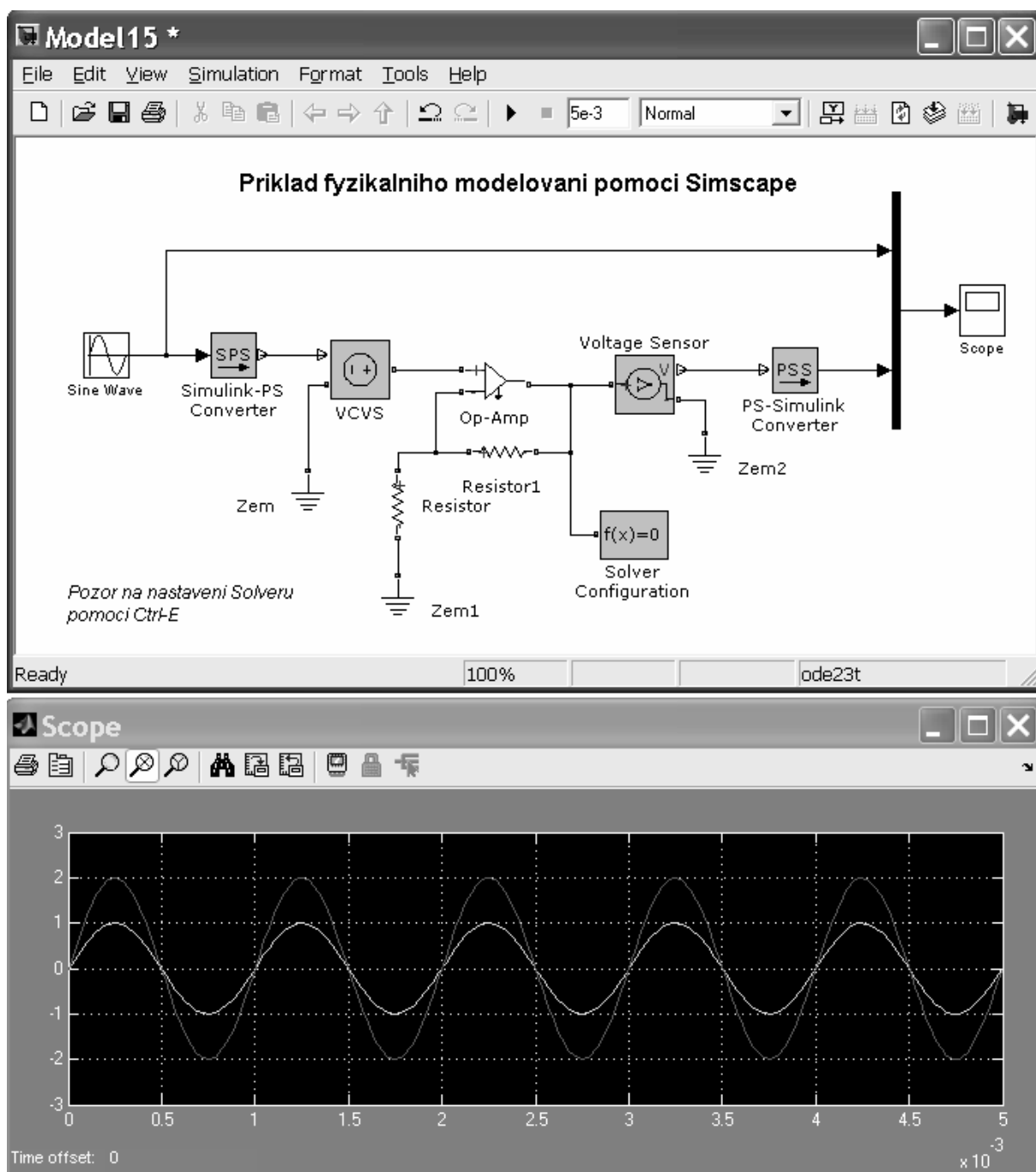
Na obr.5.39 vidíte běžný jednoduchý model zesilovače se zesílením 2.



Obr.5.39. Příklad běžného modelování zesilovače v prostředí Simulink.

Zesílení 2 je jednoduše definované pomocí bloků *Gain* a *Fcn*. Tento model je *modelem ideálním*, zahrnujícím pouze jeden parametr a to zesílení. Podobný přístup je výhodný pro svou jednoduchost a rychlost včetně pohodlné změny zesílení. Je vhodný při ověřování principů, kdy je možné se vyhnout parazitním nebo lépe napsané reálným vlastnostem zesilovačů, které však nejsou principiální.

Podívejte se nyní na obr.5.40.



Obr.5.40. Fyzikální model zesilovače se zesílením 2.