



MATLAB® & SIMULINK®

Kurzy pro začátečníky
a mírně pokročilé

prof. Karel Zaplatílek



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

Jméno a příjmení kupujícího: **Karel Zaplatílek**

E-mail: **slunce007@seznam.cz**

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.

Jak pracovat s publikací 6

Kurz 1. MATLAB pro úplné začátečníky (7x1 h)

1 h	1. Prostředí MATLAB, help, příkazová řádka, proměnné, textové řetězce	7
1 h	2. Práce s vektory	15
1 h	3. Práce s maticemi	18
1 h	4. Pole buněk (<i>Structure Array</i>) a Struktury (<i>Structure Array</i>)	20
1 h	5. Ukládání a načítání dat	23
1 h	6. Základní programátorské techniky (cykly, podmínky, ad.)	38
1 h	7. Grafika 2-D a 3-D.....	55
	Datové soubory pro použití v kurzu 1	76

Kurz 2. Ruční tvorba GUI (5x1 h)

1 h	Princip tvorby GUI, práce v příkazové řádce, grafické objekty <i>figure</i> , <i>axes</i> a <i>uicontrol</i> , pojem <i>handle</i> , příkazy <i>set</i> a <i>get</i>	77
1 h	Programování GUI metodou <i>Switched Board Programming</i> , tvorba aplikace <i>Music Box</i> - úvod	80
1 h	Postupné doplňování aplikace <i>Music Box</i> a její ladění; pole pro výpis textu. zatržítka typu <i>checkbox</i> a <i>radiobutton</i> , automatická detekce souborů *.wav ve složce, rozbalovací menu	86
1 h	Další doplňování a ladění aplikace <i>Music Box</i> , vnitřní funkce pro načtení a přehrání souborů *.wav, pole pro výpis parametrů nahrávky, zatržítka <i>radio</i> pro volbu typu vodorovné osy, tahový potenciometr slider pro změnu odstínu šedé barvy plátna, tlačítko <i>play</i> pro přehrání hudby	89
1 h	Tvorba funkce pro zobrazení všech grafických objektů (bez událostí)	95
	Datové soubory pro použití v kurzu 2	97

Kurz 3. Matematické rovnice (4 h = 1+1+2)

1 h	Matematické rovnice o jedné a dvou neznámých	98
1 h	Práce s rovnicemi s využitím symbolické matematiky. Řešení soustav lineárních a nelineárních algebraických rovnic	101
2 h	Řešení obyčejných diferenciálních rovnic (ODE)	104

Kurz 4. Čtení dat z reálné praxe (4 h = 2+1+1)

2 h	Čtení textově orientovaných souborů typu *.txt, *.csv, *.xls	112
1 h	Čtení dat ve formě struktur ze souborů typu *.mat	119
1 h	Základy práce se zvukovými a obrazovými daty	122
	Datové soubory pro použití v kurzu 4.....	127

Kurz 5. Simulink pro úplné začátečníky (4 h = 2+1+1)

2 h	Úvod do prostředí, knihovna bloků, tvorba a simulace modelu, export a import signálů, spolupráce s <i>Workspace</i> , nástroj <i>Data Inspector</i>	128
1 h	Řízení modelů Simulink z prostředí MATLAB	141
1 h	Tvorba dalších modelů: analogový kmitočtový filtr, číslicový filtr, modelování ODE 1. řádu, tvorba subsystémů	143
	Datový soubor pro použití v kurzu 5	150

Kurz 6. Spektrální analýza a číslicová filtrace (4x1 h)

1 h	Monofonní audio-nahrávka: načtení a vykreslení dat, základní analýza v časové rovině, spektrální analýza algoritmem FFT, Parsevalův teorém, práce v interaktivní aplikaci <i>Signal Analyzer</i>	151
1 h	Výkonové spektrum a výkonová spektrální hustota (PSD), výpočet a kreslení 3-D spektrogram, práce v aplikaci <i>Signal Analyzer</i>	155
1 h	Číslicová filtrace monofonní audionahrávky: návrh číslicového filtru s využitím vestavěných funkcí, vlastní číslicová filtrace, práce v aplikaci <i>Signal Analyzer</i>	158
1 h	Číslicová filtrace monofonní audionahrávky: návrh číslicového filtru interaktivní aplikací <i>filterDesigner</i> , vlastní číslicová filtrace, tvorba bloku s navrženým filtrem pro model v Simulink	162
	Datový soubor pro použití v kurzu 6	162

Kurz 7. Programování aplikací (6x1 h)

1 h	Programování skriptů v klasickém editoru: základní vlastnosti skriptů, správa paměti, sdílení proměnných. Základní programátorské techniky: cykly <i>for-end</i> a <i>while-end</i>	164
1 h	Programování skriptů: podmíněný skon <i>if-elseif-else-end</i> , přepínač <i>switch-case-otherwise-end</i> , ladění zdrojového kódu, rozdělení zdrojového kódu do sekcí, vzájemné volání skriptů	170
1 h	Programování skriptů: vstupy z klávesnice a myši, příkazy <i>assignin</i> a <i>evalin</i> , měření doby trvání zdrojového kódu, podrobný časový snímek aplikace, použití záložek v editoru, základní příkazy pro datum a čas	177
1 h	Programování funkcí: správa paměti, lokální a globální proměnné, funkce bez vstupních a výstupních parametrů, funkce se vstupními a výstupními parametry, vzájemné volání funkcí, více funkcí v jednom m-souboru, nepřímé volání funkcí technikou <i>function handle</i> , anonymní funkce, matematická rovnice o dvou neznámých jako příklad uživatelské funkce	183
1 h	Příklady uživatelských aplikací: funkce pro zápis a řešení obyčejné diferenciální rovnice 1. řádu, skript pro polynomiální aproximaci naměřených dat	190
1 h	Práce v <i>Live Editoru</i> : otevření <i>Live Editoru</i> , tvorba dokumentu s využitím živého kódu, možnosti spuštění kódu, ladění zdrojového kódu, vložení řídicích prvků do dokumentu, rozdělení dokumentu do sekcí, export dokumentu do jiných formátů (PDF, HTML, Word, LaTeX), využití aplikací, otevření běžných skriptů a funkcí v <i>Live Editoru</i>	196
	Datový soubor pro použití v kurzu 7	203

Kurz 8. Pokročilá práce v prostředích MATLAB&Simulink (5 h = 1+2+1+1)

1 h	Číslicová (diskrétní) filtrace signálu s aditivním šumem v prostředí Simulink. Číslicová filtrace s využitím interaktivního bloku <i>Digital Fiter Design</i> a s využitím bloku <i>Biquad</i>	204
2 h	Modelování a simulace obyčejné lineární diferenciální rovnice (ODE) 3. řádu s pravou stranou. Modelování textovým způsobem v MATLAB. Modelování blokovým způsobem v Simulink. Blokový způsob modelování v Simulink s využitím přenosové funkce: přímá transformace ODE 3. řádu na přenosovou funkci a transformace s využitím stavového přístupu. Řízení modelu z prostředí MATLAB	207

1 h	Spektrální analýza diskretních (číslíkových) signálů v prostředí Simulink v reálném čase. Jednorázové řízení frekvence harmonického zdroje. Řízení frekvence pomocí GUI (tahový potenciometr). Přehrávání zvuku v reálném čase s využitím bloku <i>Audio Device Writer</i>	216
1 h	Další vybrané modely systémů v prostředí Simulink. Pulzně-šířkový modulátor (PWM), skákající míč (hybridní dynamický systém)	220
Kurz 9. Tvorba GUI pomocí aplikace <i>appdesigner</i> (1,5 h = 0,5+1)		
0,5 h	Spuštění aplikace, výběr, rozmístění a nastavení základních parametrů grafických objektů. Seznam a struktura grafických objektů, automaticky generovaný zdrojový kód aplikace	225
1 h	Programování událostí: tvorba a použití globálních proměnných (nabídka <i>Property</i>), tvorba a aplikace tzv. <i>StartupFcn</i> funkce, tvorba a použití uživatelských funkcí (<i>Private Functions</i>). Spuštění vyvinuté aplikace z prostředí MATLAB	228
	Datový soubor pro použití v kurzu 9	236
Kurz 10. Číselné soustavy (2,5 h = 1,5+1)		
1,5 h	Dvojková (binární) soustava. Převod kladného celého desítkového (dekadického) čísla na dvojkové: teoretické odvození a aplikace v MATLAB (podobně u všech dalších příkladů). Převod záporného celého desítkového čísla na dvojkové - použití dvojkového doplňku a symbolický zápis. Převod kladného desetinného čísla na dvojkové. Volba počtu bitů při převodu na dvojková čísla. Převod kladného a záporného dvojkového (binárního) čísla na desítková čísla	237
1 h	Osmičková (octalová) a šestnáctková (hexadecimální) soustava. Srovnání osmičkové, desítkové a šestnáctkové soustavy - základní symboly (číslíky). Převod kladného celého desítkového čísla na dvojkové, osmičkové a šestnáctkové: teoretické odvození a aplikace v MATLAB (podobně u dalších příkladů). Převod šestnáctkového čísla na dekadické. Převod osmičkového čísla na dekadické. Převod šestnáctkového čísla na dvojkové, pojem <i>nibble</i> . Převod dvojkového čísla na šestnáctkové. Symbolický převod záporného desetinného šestnáctkového čísla na dvojkové, volba počtu bitů. Převod desetinného dvojkového čísla na šestnáctkové	242
	Tip na webové stránky	245
	Publikace	245

Předmluva

Tato kniha je určena pro lektory kurzů a školení prostředí MATLAB&Simulink. Obsahuje celkem deset kurzů pro začátečníky a mírně pokročilé, které byly vyzkoušeny a ověřeny jak v prezenční formě v prostředích poslucháren a zasedacích místností, tak ve formě distanční s využitím internetu. Odborný obsah, formát A4 a zvolený styl jsou přizpůsobeny vedení kurzů.

Lektor při vedení kurzů a školení potřebuje znát časový rozsah celého kurzu i jednotlivých kapitol a řadu popůrných návěstí, tipů a doporučení tak, aby průběh každé kapitoly byl plynulý a aby jednotlivé kapitoly měly odbornou i časovou kontinuitu. Zvolené fonty, barvy, zvýraznění a celkový styl vyplynuly mých z mnohaletých zkušeností s různými formami kurzů a školení před posluchači s různou odbornou úrovní a zaměřením.

Větší formát knihy a větší použité fonty velmi usnadňují práci v pološeru zasedacích místností nebo učeben, jak ví každý, kdo někdy vedl kurzy podobného typu. Přes velká usnadnění, která nám poskytuje výpočetní technika a informační technologie se domnívám, že naní nad tištěné podklady kurzu. Lektor tak nemusí v případě nutnosti přepínat různá okna na své obrazovce a způsobovat zmatek na promítacím plátně dataprojektoru v případě, kdy si potřebuje osvěžit paměť nebo nalézt zapomenutý příkaz či funkci.

Neskromně se domnívám, že moje dlouholetá učitelská s lektorská praxe, spojená se zpětnou vazbou od studentů, účastníků kurzů a učitelů, skýtá záruku kvalitního pečlivě rozmyšleného a pedagogickou praxí vyzkoušeného obsahu.

Jak pracovat s publikací

Jak je zmíněno výše, obsahuje kniha deset kurzů prostředí MATLAB&Simulink. Již v obsahu naleznete kromě odborných témat také časové rozsahy pro celý daný kurz i jeho jednotlivé kapitoly. Časový rozsah každé kapitoly je navíc uveden přímo na jejím začátku za názvem. Doporučuji si celý kurz před vlastním vedením projít a pokusit se odhadnout časové možnosti vaše a vašich posluchačů. Sám dobře vím, že s některou skupinou čas jen letí, zatímco s jinou se spíše vleče. Proto je třeba být flexibilní, nespěchat a nechat si určitou odbornou rezervu pro případ, že vám zbyde na konci více času. To vše je součástí praxe lektora, která se vyvíjí postupně s množstvím odvedených kurzů a školení. Největší výzvou je nesnažit se spěchat a neříkat posluchačům příliš mnoho informací naráz. Spíše doporučuji pomalejší tempo s dostatkem času na provedení příkazů s jejich dobrým zažitím.

Přeji si, aby se tato kniha stala příručkou pro lektory kurzů a školení prostředí MATLAB&Simulink. Prakticky každý software se dnes poměrně rychle vyvíjí; prostředí MATLAB není výjimkou. Buďte proto flexibilní tam, kde to bude v budoucnu třeba.

Autor

MATLAB®, Simulink®, Signal Processing Toolbox™ a DSP System Toolbox™
jsou registrovanými ochrannými známkami společnosti The MathWorks, Inc.

;

Kurz 4/10

Čtení dat z reálné praxe (2+1+1 h)

1. Čtení textově orientovaných souborů typu *.txt, *.csv, *.xls

a) Automatizované měření magnetické indukce

Prog_cteni_1_indukce1.m, soubor Indukce.xls, použití příkazu readmatrix

```
% cteni automatizovaneho mereni
% magneticke indukce podel magnetu
% pouziti prikazu readmatrix
%-----
close all
clc
%--- cteni dat ---
data=readmatrix('Indukce.xls','Sheet','List1','Range','A2:
B67');
whos data
[r,s]=size(data); % 66x2
d=data(:,1); % 1. sloupec
B=data(:,2);
%--- kresleni ---
h1=plot(d,B,'ro-','linewidth',1,'MarkerEdgeColor',[1 0
0]);
grid on
axis([d(1) d(end) -170 170])
xlabel('Vzdálenost [mm]','fontsize',12)
ylabel('Indukce [mT]','fontsize',12)
title('Průběh magnetické indukce podél
magnetu','fontsize',14)
set(gca,'fontsize',12) % cisla na osach
set(gca,'Color',[0.9 0.9 0.9]) % barva plata axes
%--- ulozeni grafu na disk ---
set(gcf,'InvertHardcopy','off') % nastavení pro správné
uložení
saveas(gcf,'Obr_01','png')
%-----
```

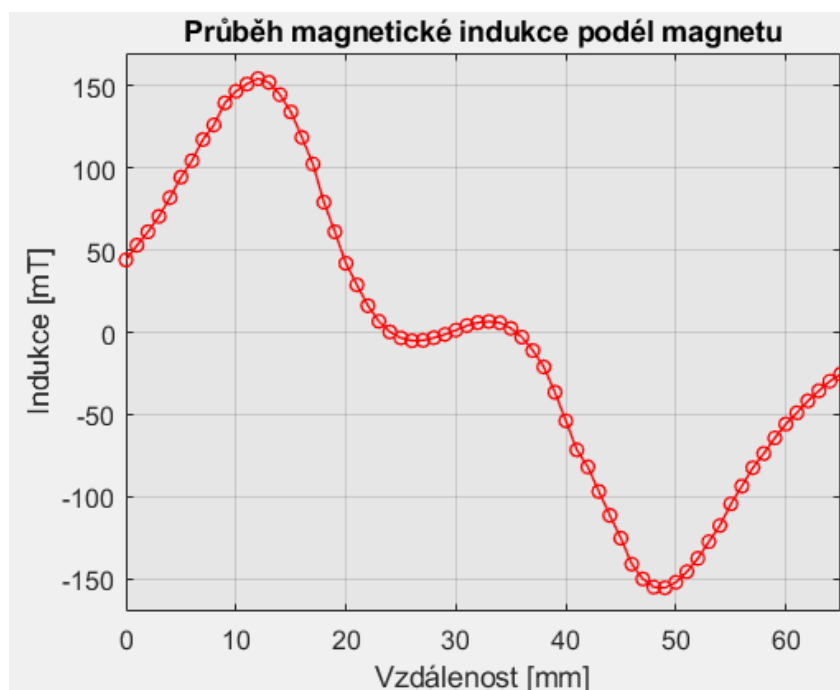
Microsoft Excel - Indukce

Soubor Úpravy Zobrazit

Arial

E1

	A	B	C
1	d [mm]	B [mT]	
2	0	44,1	
3	1	53	
4	2	61,2	
5	3	70,6	
6	4	82	
7	5	94,5	
8	6	104,5	
9	7	117,4	
10	8	126,3	
11	9	139,7	
12	10	146,8	
13	11	151,2	
14	12	154,5	
15	13	152,2	



b) Automatizované měření magnetické indukce

Prog_cteni_2_indukce2.m, soubor Indukce.xls, použití příkazu readtable

```
% cteni automatizovaneho mereni
% magneticke indukce podel magnetu
% pouziti prikazu readtable
%-----
close all
clc
%--- cteni dat ---
data=readtable('Indukce.xls','Sheet','List1','Range','A2:B
67');
whos data % table
[r,s]=size(data); % 66x2
d=data(:,1); % 1. sloupec z tabulky
B=data(:,2);
%--- kresleni ---
h1=plot(d,B,'ro-','linewidth',1,'MarkerEdgeColor',[1 0
0]);
grid on
axis([d(1) d(end) -170 170])
xlabel('Vzdálenost [mm]','fontsize',12)
ylabel('Indukce [mT]','fontsize',12)
title('Průběh magnetické indukce podél
magnetu','fontsize',14)
set(gca,'fontsize',12) % cisla na osach
set(gca,'Color',[0.9 0.9 0.9]) % barva plata axes
```

```
%--- uložení grafu na disk ---
set(gcf,'InvertHardcopy','off') % nastavení pro správné
uložení
saveas(gcf,'Obr_01','png')
%-----

doc readmatrix, doc readtable
```

c) Čtení dat z akcelerometru chytrých hodinek,

Prog_cteni_3_krokomer.m, soubor Krokomer.csv

```
% ctení dat z akcelerometru chytrych hodinek
%-----
close all
clc
%--- ctení dat ---
data=readmatrix('Krokomer.csv');
whos data % table
[r,s]=size(data); % 3344x3
x=data(:,1); % 1. sloupec z tabulky, zrychlení ve smeru x
y=data(:,2);
z=data(:,3);
%--- kreslení ---
h1=plot(x,'k');
hold on
h2=plot(y,'r');
hold on
h3=plot(z,'b');
grid on
axis([1 r -Inf Inf])
xlabel('Pořadový index [-]','fontsize',12)
ylabel('Zrychlení [m.s-2'],'fontsize',12)
title('Záznam zrychlení tříosého
akcelerometru','fontsize',14)
set(gca,'fontsize',12) % čísla na osách
set(gca,'Color',[0.9 0.9 0.9]) % barva plotu axes
%--- analýza a zpracování ---
a=sqrt(x.^2+y.^2+z.^2); % celkové zrychlení
a=a-mean(a); % odstranění střední hodnoty (sum)
figure
h4=plot(a,'r');
grid on
axis([1 r -Inf Inf])
xlabel('Pořadový index [-]','fontsize',12)
ylabel('Zrychlení [m.s-2'],'fontsize',12)
```

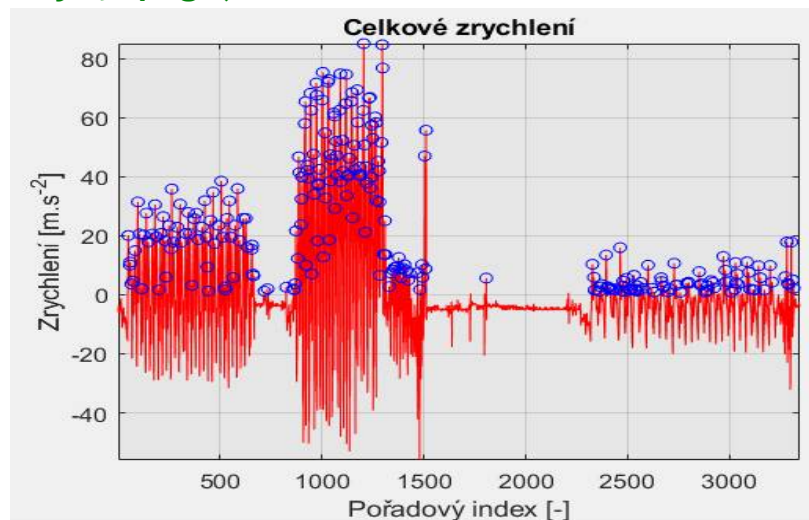
```

title('Celkové zrychlení','fontsize',14)
set(gca,'fontsize',12) % čísla na osách
set(gca,'Color',[0.9 0.9 0.9]) % barva plota axes
%---
[m1,m2]=max(a);
prah=1/100*m1; % hranice sumy, experiment
[vyska,kroky]=findpeaks(a,'MINPEAKHEIGHT',prah); % detekce
piku nad sumou
whos vyska kroky
hold on
h5=plot(kroky,vyska,'bo'); % detekované kroky
%--- uložení grafu na disk ---
%set(gcf,'InvertHardcopy','off') % nastavení pro správné
uložení
%saveas(gcf,'Obr_kroky','png')
%-----

```

Krokometr – Poznámkový blok

Soubor	Úpravy	Formát	Zobrazení
339,262,329			
338,260,335			
338,260,335			
338,260,334			
337,261,334			



d) Čtení dat ze snímače rychlosti hydraulického stroje,

Prog_cteni_4_hydraulika.m, soubor Hydraulika.txt

```

% čtení dat ze snímače rychlosti
% hydraulického stroje
%-----
close all
clc
%--- čtení dat ---
data=readmatrix('Hydraulika.txt');
whos data % 17861x5 double
[r,s]=size(data); % 17861x5
cas=data(:,1); % absolutní čas měření
rychlost=data(:,5);
%--- kreslení ---
h1=plot(cas,rychlost,'r');

```

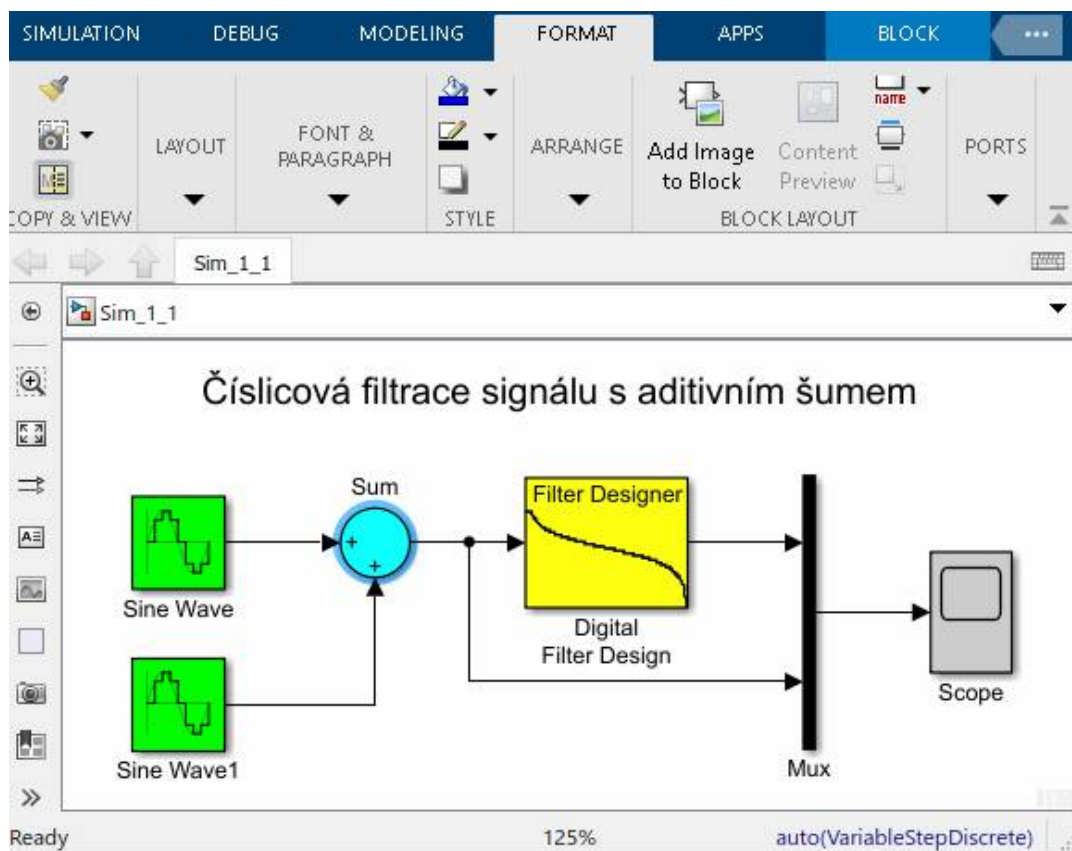
Kurz 8/10

Pokročilá práce v prostřecích MATLAB&Simulink (5 h = 1+2+1+1)

1. Číslicová (diskrétní) filtrace signálu s aditivním šumem v prostředí Simulink

1.1. Číslicová filtrace s využitím bloku „Digital Filter Design“

V Simulink založíme nový model s názvem **Sim_1_1.slx** (příponu dodá systém):

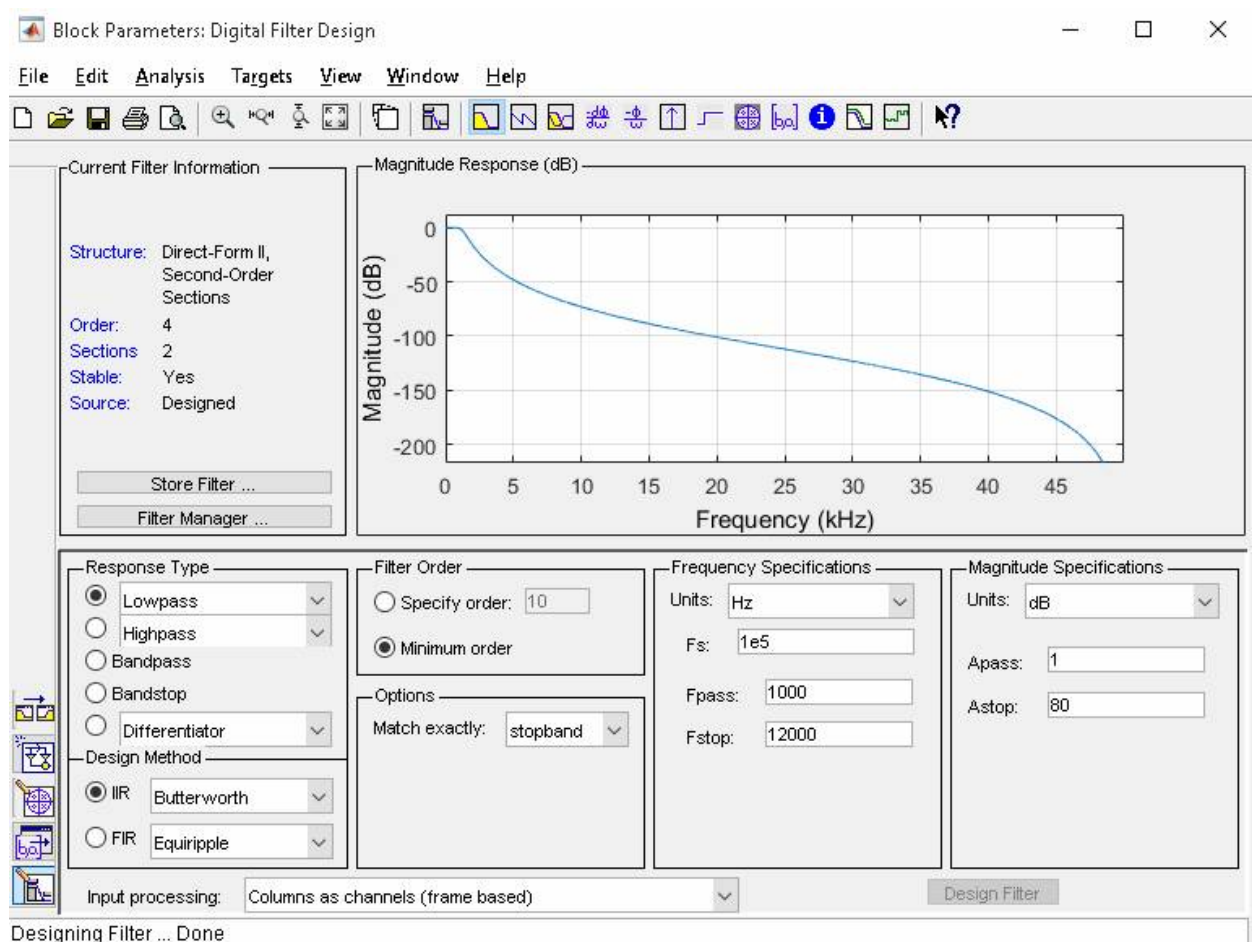


Bloky zdrojů harmonického signálu (**Sine Wave**) z knihovny **Sources** mají frekvence 1 kHz a 10 kHz (parametry **Frequency (rad/sec)**: jsou tedy $2\pi \cdot 10^3$ a $2\pi \cdot 10^4$), **amplitudy** 1 a 0,2 a **Sample Time** 10^{-5} (oba shodné).

Jde tedy o harmonický signál s frekvencí 1 kHz, na který je nasuperponován (přičten) vysokofrekvenční aditivní (součtový) šum s frekvencí 10 kHz. Oba harmonické signály jsou **diskrétní** se vzorkovací frekvencí 100 kHz.

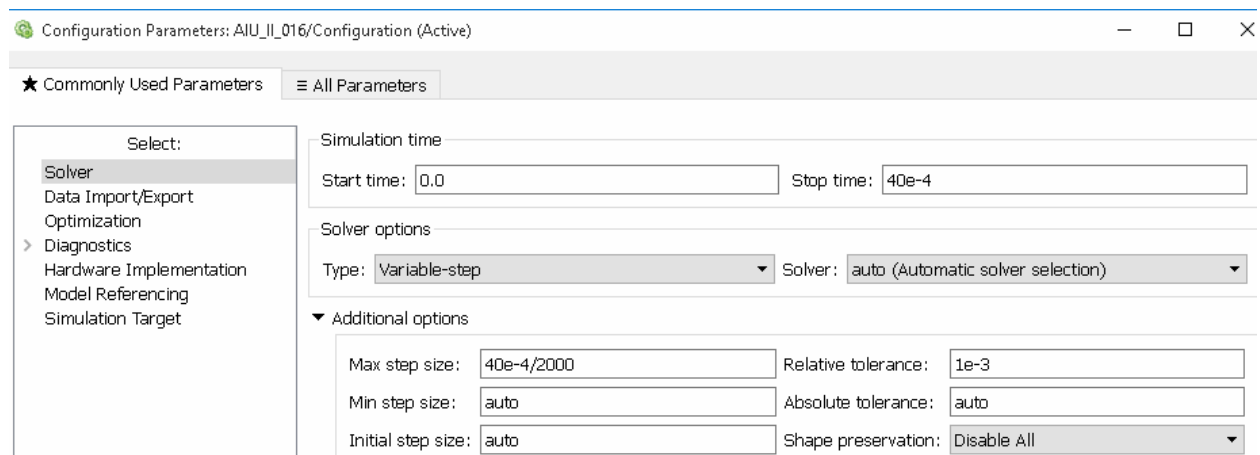
K filtraci je použit blok **Digital Filter Design** (knihovna *DSP System Toolbox / Filtering / Filter Designs*).

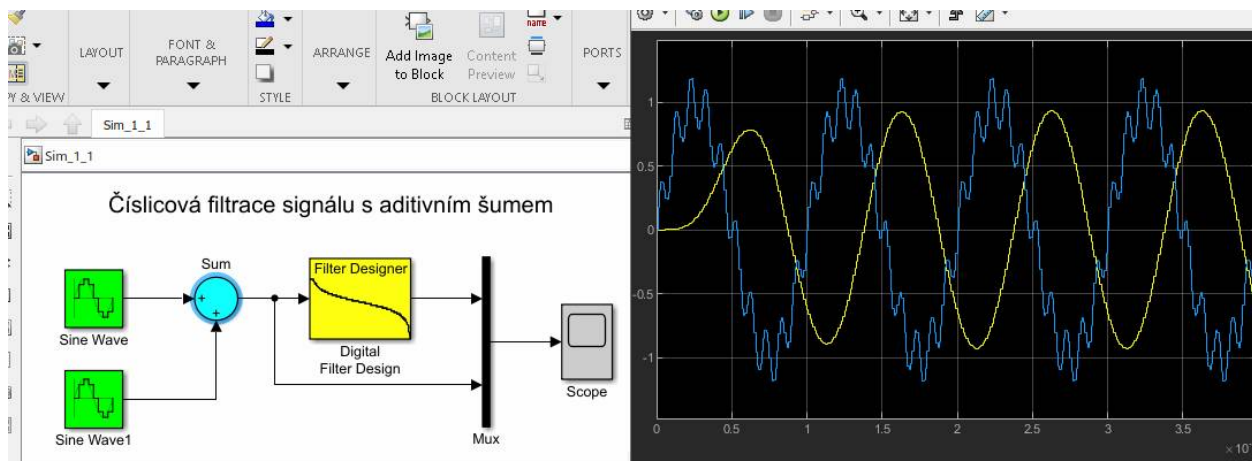
Poklepáním myši na blok **Digital FilterDesign** se otevře již známé okno pro návrh číslicového filtru:



Vyplníme parametry a klikneme na **Design Filter**. Pak stačí celé okno zavřít křížkem vpravo nahoře a navržený filtr se automaticky přenese do bloku **Digital Filter Design**.

Nastavíme parametry simulace (**Ctrl-E**), stačí jen **Stop time** a **Max step size**, a spustíme simulaci:





V časových průbězích je vidět krátký přechodný děj na počátku simulace a **fázový posun** výstupního signálu (zpoždění vlivem filtru).

Experimentujte s různými druhy filtru a jejich parametry sledujte kvalitu filtrace.

1.2. Číslicová filtrace s využitím bloku „Biquad“

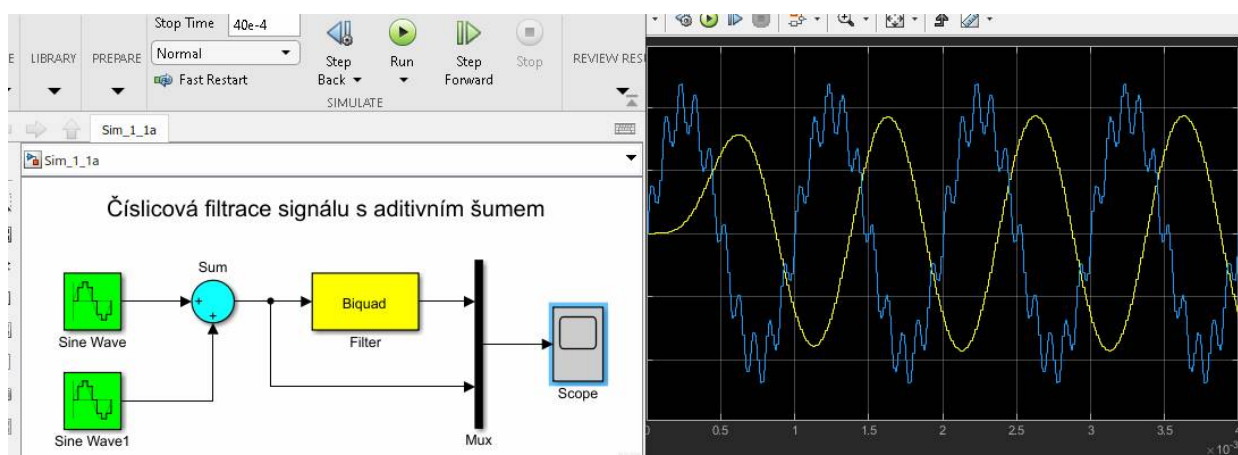
Jedná se variantu výše uvedené filtrace. Soubor **Sim_1_1.slx** uložíme pod novým názvem **Sim_1_1a.slx**

Poklepejte znovu myši na blok **Digital Filter Design**, zobrazí se dříve navržený číslicový filtr.

Nyní zvolte nabídku **File / Export to Simulink Model**.

V novém okně vpravo dole klikněte na **Realize Model**:

Přímo v našem modelu se objeví nový blok **Biquad**, obsahující koeficienty stejného filtru. Zaměníme blok **Digital Filter Design** za **Biquad** a spustíme simulaci. Bude shodná:



Poklepáním na blok **Biquad** můžete zobrazit jeho parametry a pomocí **Help** nastudovat detaily:

Kurz 10/10

Číselné soustavy (1,5+1 h)

1. Dvojková (binární) soustava

Základ dvojkové soustavy: 2

Číslice (symboly): 0 1

Ilustrační příklad:

0	1	0	1	1	0	1	1
$0 \cdot 2^7$	$1 \cdot 2^6$	$0 \cdot 2^5$	$1 \cdot 2^4$	$1 \cdot 2^3$	$0 \cdot 2^2$	$1 \cdot 2^1$	$1 \cdot 2^0$

8 bitů = 1 Byte (bajt, slovo)
součet: $2^6 + 2^4 + 2^3 + 2^1 + 2^0 = 91$

Příklad 1.1

Převeďte desítkové (dekadické) číslo +458 na dvojkové (binární).

Symbolicky:

$$(+458)_{10} \rightarrow X_2$$

a) analyticky (papír a tužka) metodou dělení číslem 2 (tzv. Euklidův algoritmus):

458:2=229	zbytek 0 (LSB)	MATLAB: fix(458/2), mod(458,2)
229:2=114	zbytek 1	
114:2= 57	zbytek 0	
57:2= 28	zbytek 1	
28:2= 14	zbytek 0	LSB: <i>Least Significant Bit</i> (nejméně významný bit)
14:2= 7	zbytek 0	MSB: <i>Most Significant Bit</i> (nejvýznamnější bit)
7:2= 3	zbytek 1	
3:2= 1	zbytek 1	
1:2= 0	zbytek 1 (MSB)	

Výsledek: $(+458)_{10} \rightarrow (111001010)_2$

b) řešení v MATLAB: **b1=dec2bin(458)** % vrací '111001010'
b2=dec2bin(458,16) % vrací '0000000111001010', 16 bitů
b3=dec2base(458,2) % '111001010'
b4=dec2base(458,2,16) % '0000000111001010'

```
Command Window
>> b1=dec2bin(458)

b1 =

    '111001010'

>> b2=dec2bin(458,16)

b2 =

    '0000000111001010'
```

Příklad 1.2

Převeďte desítkové (dekadické) číslo +13 na dvojkové (binární) na 8 bitů (1 Byte).
Symbolicky:

$$(+13)_{10} \rightarrow X_2$$

a) analyticky (papír a tužka) metodou dělení číslem 2 (tzv. Euklidův algoritmus):

13:2= 6	zbytek 1 (LSB)	MATLAB: fix(13/2), mod(13,2)
6:2= 3	zbytek 0	
3:2= 1	zbytek 1	
1:2= 0	zbytek 1 (MSB)	

Výsledek: $(+13)_{10} \rightarrow (1101)_2$

b) řešení v MATLAB: **b1=dec2bin(13)** % vrací '1101'
b2=dec2bin(13,8) % vrací '00001101', 8 bitů
b3=dec2base(13,2) % '1101'
b4=dec2base(13,2,8) % '00001101'

```
Command Window
>> b1=dec2bin(13)

b1 =

    '1101'

>> b2=dec2bin(13,8)

b2 =

    '00001101'
```

Příklad 1.3

Převeďte desítkové číslo -71 na dvojkové (binární) na 16 bitů (2 Bytes).
Symbolicky:

$$(-71)_{10} \rightarrow X_2$$

a) analyticky s využitím tzv. dvopjkového doplňku (*two's complement*),

- vyjádříme nejprve číslo kladné dělím 2:

$$(+71)_{10} \rightarrow X_{21}$$

71:2= 35	zbytek 1 (LSB)	MATLAB: fix(71/2), mod(71,2)
35:2= 17	zbytek 1	
17:2= 8	zbytek 1	
8:2= 4	zbytek 0	
4:2= 2	zbytek 0	
2:2= 1	zbytek 0	
1:2= 0	zbytek 1 (MSB)	

Dílčí výsledek: $(+71)_{10} \rightarrow (0000000001000111)_2$

- znegujeme všechny číslice (symboly):

$$(1111111110111000)_2$$

- přičteme číslo 1 k LSB (zcela vpravo):

$$(1111111110111001)_2$$

Výsledek: $(-71)_{10} \rightarrow (1111111110111001)_2$

b) symbolické vyjádření záporného čísla

Výsledek: $(-71)_{10} \rightarrow (-0000000001000111)_2$

c) řešení v MATLAB: **b1=dec2bin(-71)** % vrací '10111001'

b2=dec2bin(-71,16) % vrací '1111111110111001', 16 bitů

```
Command Window
>> b1=dec2bin(-71)

b1 =

    '10111001'

>> b2=dec2bin(-71,16)

b2 =

    '1111111110111001'
```

Příklad 1.4

Převeďte desítkové číslo -19 na dvojkové (binární) na 16 bitů (2 Bytes).

Symbolicky:

$$(-19)_{10} \rightarrow X_2$$

a) analyticky s využitím tzv. dvojkového doplňku (*two's complement*),

- vyjádříme nejprve číslo kladné dělím 2:

$$(+19)_{10} \rightarrow X_{21}$$

19:2= 9	zbytek 1 (LSB)	MATLAB: fix(19/2), mod(19,2)
9:2= 4	zbytek 1	
4:2= 2	zbytek 0	
2:2= 1	zbytek 0	
1:2= 0	zbytek 1 (MSB)	

Dílčí výsledek: $(+19)_{10} \rightarrow (0000000000010011)_2$

- znegujeme všechny číslice (symboly):

$$(111111111101100)_2$$

- přičteme číslo 1 k LSB (zcela vpravo):

$$(111111111101101)_2$$

Výsledek: $(-19)_{10} \rightarrow (111111111101101)_2$

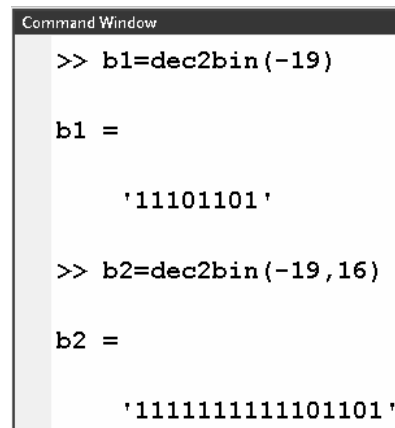
poznámka: první číslice zleva je tzv. znaménkový bit; je-li „1“, jde o záporné číslo.

b) symbolické vyjádření záporného čísla

Výsledek: $(-19)_{10} \rightarrow (-0000000000010011)_2$

c) řešení v MATLAB: **b1=dec2bin(-19)** % vrací '11101101'

b2=dec2bin(-19,16) % vrací '111111111101101', 16 bitů



```
Command Window
>> b1=dec2bin(-19)

b1 =

    '11101101'

>> b2=dec2bin(-19,16)

b2 =

    '111111111101101'
```

Tip: www.rapidtables.com/convert (vybrat: Hex/decimal/octal/binary converter)

MATLAB® & Simulink®

Kurzy pro začátečníky a mírně pokročilé

prof. Ing. Karel Zaplatílek, Ph.D.