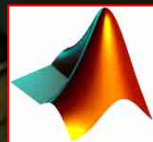


Učebnice

MATLAB

Simulink

řešené příklady



Libor Kupka
Josef Janeček



TENTO PROJEKT JE SPOLUFINANCOVÁN
EVROPSKÝM SOCIÁLNÍM FONDEM A STÁTNÍM
ROZPOČTEM ČESKÉ REPUBLIKY



Matlab & Simulink

řešené příklady

Libor Kupka, Josef Janeček

Obsah

Předmluva	7
Úvod.....	9
1 Analýza elektrického obvodu	11
1.1 Text pro studenta	12
1.2 Text pro pedagoga	16
1.2.1 Řešení pomocí lineární algebry	17
1.2.2 Řešení v prostředí MATLABu	19
2 Rezonanční vlastnosti RLC členu	23
2.1 Text pro studenta	24
2.2 Text pro pedagoga	28
2.2.1 Vytvoření simulačního modelu.....	29
2.2.2 Spuštění simulace a zobrazení výsledků.....	32
2.2.3 Frekvenční charakteristiky	34
3 Dynamika sedačky řidiče	39
3.1 Text pro studenta	40
3.2 Text pro pedagoga	43
3.2.1 Vytvoření simulačního modelu.....	45
3.2.2 Spuštění simulace a zobrazení výsledků.....	47
3.2.3 Matematická analýza pohybu sedačky	48
3.2.4 Ověření analytického řešení v Simulinku.....	52
3.2.5 Dynamický systém na mezi periodicity	53
3.2.6 Kmitavý dynamický systém.....	54
4 Balistická křivka projektilu	59
4.1 Text pro studenta	60
4.2 Text pro pedagoga	63
4.2.1 Vytvoření simulačního modelu.....	66
4.2.2 Vykreslení několika křivek do jednoho grafu	69
4.2.3 Model inverzní dynamiky	73

5	Řešené příklady	79
5.1	Elektron v homogenním elektrickém poli	80
5.2	Základní vlastnosti střídavého elektrického signálu	81
5.3	Výkon v RLC obvodu.....	82
5.4	Impedance RLC obvodu.....	84
5.5	Řešení elektrického obvodu přímou aplikací Kirchhoffových zákonů	85
5.6	Řešení elektrického obvodu metodou smyčkových proudů.....	87
5.7	Řešení elektrického obvodu metodou uzlových napětí.....	90
5.8	Rovnoměrný přímočarý pohyb hmotného tělesa	93
5.9	Rovnoměrně zrychlený přímočarý pohyb.....	97
5.10	Volný pád tělesa	100
5.11	Vrh svislý vzhůru	103
5.12	Pohyb tělesa po nakloněné rovině	106
5.13	Matematické kyvadlo	108
5.14	Let kosmické lodi.....	112
5.15	Ohyb jednostranně vetknutého nosníku.....	115
5.16	Vývoj výšky hladiny v nádrži.....	118
5.17	Vývoj výšky hladin dvou spojených nádrží.....	123
5.18	Navíjecí zařízení.....	127
5.19	Stejnoseměrný motor s cizím buzením	130
5.20	Stejnoseměrný motor s buzením permanentními magnety	134
5.21	Kombinační logický obvod.....	137
6	Příklady z oblasti teorie automatického řízení.....	143
6.1	Přechodové charakteristiky dynamických systémů 1. řádu	144
6.2	Přechodové charakteristiky dynamických systémů 2. řádu	149
6.3	Konstrukce dynamických charakteristik systému s využitím Control System Toolbox.....	156
6.4	Frekvenční charakteristiky dynamických systémů 2. řádu	161
6.5	Přechodové charakteristiky PID regulátoru.....	171
6.6	Regulační obvod se spojitým PID regulátorem	175

6.7	Modifikace PID algoritmu, seřízení regulátoru metodou Zieglera a Nicholse.....	182
6.8	Oblast stability regulačního obvodu	190
6.9	Dvoupolohová a třípolohová regulace.....	195
6.10	Číslicové řízení, regulátor PSD	201
6.11	Numerická optimalizace parametrů regulátoru	206
Literatura		215
PŘÍLOHA A: Laplaceova integrální transformace		217
PŘÍLOHA B: Seznam souborů		223

Předmluva

Sbírka příkladů řešených s využitím prostředí MATLAB – Simulink orientovaná na výuku mechatroniky je jedním z hlavních výstupů projektu „Podpora odborné přípravy středoškolské mládeže pro podmínky automatické i automatizované výroby (mechatronika)“, řešeného v letech 2005 až 2007 s podporou operačního programu EU Rozvoj lidských zdrojů, opatření 3.1.

Vydání sbírky je výsledkem několika systémových kroků. Především byl podroben široké diskusi výběr vhodného software – základu pro modelování, simulaci a výpočty částí mechatronických systémů. Po rozhodnutí o orientaci na prostředí MATLAB byla v rámci programu připravena a vydána metodická příručka – MATLAB & Simulink: úvod do použití. Operační program nám umožnil i uspořádání několika seminářů, na kterých byli účastníci, především učitelé středních odborných škol, seznámeni s možnostmi vybraných programových prostředků i s některými typickými úlohami. Finální seminář v červnu 2007 ukončuje etapu vývoje sbírky úloh. Sled těchto logických kroků k vytvoření ucelené sbírky úloh by nebylo možné uskutečnit bez grantové podpory z ESF a MŠMT.

Sbírku úloh uvádějí čtyři metodicky podrobně zpracované úlohy, které mohou sloužit i jako studijní materiál pro odborné učitele. Následující kapitola obsahuje úlohy orientované na modelování elementů mechatronických systémů na bázi elektroniky, elektrotechniky i mechaniky. Kapitola šestá se zabývá simulací systémů automatického řízení jak v analogové, tak v digitální formě. Úlohy mají bezprostřední návaznost na obsah základních odborných kompetencí středoškolských oborů zaměřených na automatizaci, automatizační techniku a mechatroniku. Odborní učitelé na technických středních školách mohou, samozřejmě diferencovaně, využít předloženou sbírku úloh ve výuce s tím, že použijí hotové programy. Nemusí tedy vlastní programy vytvářet. Není rovněž nutné, aby žáci znali matematický popis modelovaných jevů, který je často nad rámec jejich disponibilních znalostí. Ve virtuální podobě však mohou s modely pracovat a analyzovat tak vlastnosti mechatronických prvků.

Pro aplikaci je nutné mít k dispozici počítač, programový produkt MATLAB – Simulink, sbírku úloh a nosič s připravenými úlohami. Po zkušenosti s jejich aplikací si mohou aktivní učitelé připravit své další úlohy v souvislosti s potřebami a zaměřením školy. Prostředí MATLAB mohou ve výuce použít i učitelé matematiky, fyziky, chemie, biologie a dalších odborných předmětů. V tomto smyslu připravili autoři sbírky, zkušení pedagogové, didaktické pomůcky mající širší uplatnění na středních školách.

doc. Ing. Ladislav Maixner, CSc.
manažer projektu

Úvod

Motto: „To, co držíte právě v ruce, není učebnice, ale inspiromat, návod na to, jak vidět jednoduše i věci mnohdy složité.“

Předkládaná kniha navazuje na již vydanou příručku Matlab & Simulink: úvod do použití, ve které byly podrobně vyloženy základní funkce prostředí MATLAB – Simulink. Snahou autorů sbírky je především ukázat na nesporné výhody využití MATLABu a Simulinku při výuce. Velmi efektivně je totiž možné s pomocí tohoto prostředí řešit mnohdy i značně komplikované úlohy. Díky dobrým grafickým možnostem MATLABu lze obdržené výsledky názorně prezentovat a více tak přiblížit probíranou látku studentům.

V první části sbírky jsou uvedeny čtyři modelové příklady (každý v samostatné kapitole), které jsou podrobně po metodické stránce rozpracovány. Příklady jsou strukturovány do dvou navzájem oddělených úrovní. Součástí úrovně první, určené pro studenta, je formulace zadání konkrétního technického problému. Studentovi je následně nabídnuto řešení pomocí připraveného skriptu nebo modelu. Nejsou kladeny prakticky žádné nároky na zvládnutí složitějšího matematického aparátu (např. lineární algebry či diferenciálního počtu). Činnost studenta je především směřována ke zvládnutí práce s prostředím MATLAB a Simulink. Jsou uvedeny i kontrolní otázky a úkoly vybízející studenta k aktivnímu přístupu k řešení problematice, a to směrem, který je v části určené pro pedagoga dále podrobněji rozebrán.

V druhé úrovni, která je určena pro pedagoga, jsou příklady (uvedené již ve studentské části) detailně rozpracovány, tedy včetně teoretického rozboru a diskuze možných variant jejich řešení. Uživatel je krok po kroku provázen simulačním experimentem a zároveň je také veden i k případným dalším aplikacím uvedených přístupů. Po zkušenostech získaných při práci s připravenými úlohami, mohou aktivní učitelé, v závislosti na konkrétním zaměření vyučovaného předmětu, tyto úlohy dále modifikovat. Měli by být ale schopni vytvářet i úlohy vlastní.

V následující části sbírky je uveden ucelený soubor příkladů, které byly voleny tak, aby byly přímo využitelné při výuce, zejména v matematice, fyzice, mechanice, elektrotechnice, automatizaci a řídicí technice. Teoretický rozbor řešení je obvykle velmi zkrácen. Příklady jsou zaměřeny zejména na sestavování matematických modelů pomocí tzv. matematicko-fyzikální analýzy. Jsou vyloženy základní přístupy umožňující vytvoření odpovídajících simulačních modelů v prostředí MATLAB – Simulink.

V závěru je uvedeno několik úloh, které jsou orientovány na oblast teorie automatického řízení. Zvláštní pozornost je věnována jednak způsobům modelování regulované soustavy, ať již na základě příslušné diferenciální rovnice nebo na základě obrazového přenosu, ale především možnostem realizace regulátoru. Uvedeno je několik různých realizací spojitého PID regulátoru a jeho číslíkové obdoby PSD, ale i regulátorů nespojitých – dvoupolohových a třípolohových. V prostředí Simulinku jsou pak vytvářeny modely uzavřeného regulačního obvodu. Ve stručnosti jsou popsány a simulačně ověřeny i některé přístupy vedoucí k nazelení

optimálních parametrů regulátoru. Je také ilustrována jednoduchá, ale přesto velmi užitečná aplikace MATLABu, která umožňuje velmi efektivně podle zvoleného kritéria najít parametry regulátoru na základě numerické (iterační) optimalizace, tedy bez použití obtížných matematických postupů.

Pro konstrukci dynamických charakteristik modelovaných systémů je velmi užitečné také použití speciálních příkazů doplňkového Control System Toolboxu. Tento toolbox poskytuje značné množství příkazů pro teorii automatického řízení. V textu je prostor věnován zejména charakteristikám frekvenčním – je demonstrováno několik zajímavých situací při různém rozložení pólů charakteristické rovnice systému 2. řádu. V jednom z příkladů je diskutována i oblast stability jednoduchého regulačního obvodu.

Všechny v textu popisované modely a jim příslušné soubory skriptu jsou čtenáři dostupné na přiloženém datovém médiu. Naprogramované úlohy jsou také k dispozici ke stažení na internetové adrese <http://www.spslan.cz/tech>. Programy jsou v uvedené formě okamžitě použitelné, příslušné soubory postačuje umístit do aktuálního adresáře MATLABu. Velmi snadno lze uvedené modely upravovat a následně je dle konkrétní potřeby použít ve výuce.

Milou povinností autorů sbírky je poděkovat panu doc. Ing. Ladislavu Maixnerovi, CSc., hlavnímu manažerovi projektu, za poskytnutý prostor a finanční podporu, bez níž by tento text nemohl vzniknout. Autoři by rádi vyjádřili poděkování i panu Ing. Gunnaru Künzelovi za poskytnutí podkladů k úlohám uvedeným v kapitolách 5.8 až 5.15 a v neposlední řadě také panu doc. Ing. Vladislavu Singulemu, CSc. za dodání podkladů a konkrétních simulačních parametrů ke kapitole 5.20.

Pokud shledáte tuto publikaci za užitečnou, řekněte to svým kolegům. Na druhou stranu budou autoři velmi vděční za všechny připomínky, názory a tipy na doplnění či zlepšení textu. Uvítají i upozornění na jakékoliv drobné chyby a nepřesnosti (i edičního charakteru) či prosté překlepy. Tyto připomínky budou použity při vytváření případného dalšího vydání. Cílem autorů je vytvořit knihu, která by čtenáři přinesla inspiraci a alespoň trochu pomáhala při jeho práci. Reakce a připomínky lze zasílat na e-mailové adresy

libor.kupka@tul.cz
josef.janecek@tul.cz

1 Analýza elektrického obvodu

Základem všech metod analýzy elektrických obvodů jsou Kirchhoffovy zákony. Pro řešení složitějších obvodů bylo v minulosti navrženo několik různých metod. Mezi základní patří metoda smyčkových proudů, metoda uzlových napětí, metoda větví stromu, Théveninova a Nortonova věta a další.

Rovnice popisující libovolný elektrický obvod lze samozřejmě získat přímou aplikací obou Kirchhoffových zákonů. **1. Kirchhoffův zákon** (proudový) je speciálním případem rovnice kontinuity. Říká, že algebraický součet proudů ve větvích spojených v libovolném uzlu je roven nule

$$\sum_k i_k(t) = 0$$

Obvykle uvažujeme, že proudy tekoucí do uzlu mají znaménko plus a proudy tekoucí z uzlu pak znaménko minus.

2. Kirchhoffův zákon (napěťový) říká, že algebraický součet všech napětí ve větvích tvořících libovolnou smyčku je roven nule

$$\sum_k u_k(t) = 0$$

Napětí orientované souhlasně se smyčkou uvažujeme jako kladné, napětí orientované opačně jako záporné.

V dalším textu budou pro řešení elektrických obvodů použity také metody smyčkových proudů a uzlových napětí. Uvedme ve stručnosti jejich popis. **Metoda smyčkových proudů** vychází z druhého Kirchhoffova zákona. Neznámými jsou v tomto případě fiktivní (myšlené) smyčkové proudy, které zavedeme do nezávislých smyček analyzovaného obvodu. Tyto proudy automaticky splňují první (proudový) Kirchhoffův zákon. Metoda je výhodná zvláště tehdy, má-li obvod menší počet nezávislých smyček než uzlů.

Metoda uzlových napětí vychází z prvního Kirchhoffova zákona a je vhodná zejména v případě, že analyzovaný elektrický obvod má několik paralelních větví sbíhajících se v jednom uzlu. V obvodu zvolíme jeden referenční uzel a zavedeme napětí zbývajících uzlů (tzv. uzlová) vůči němu. Referenční uzel volíme tak, abychom s výhodou využili znalosti velikosti napětí co nejvíce napěťových zdrojů ke snížení počtu neznámých. K případnému dalšímu studiu může čtenář využít např. knihy [1 až 3].

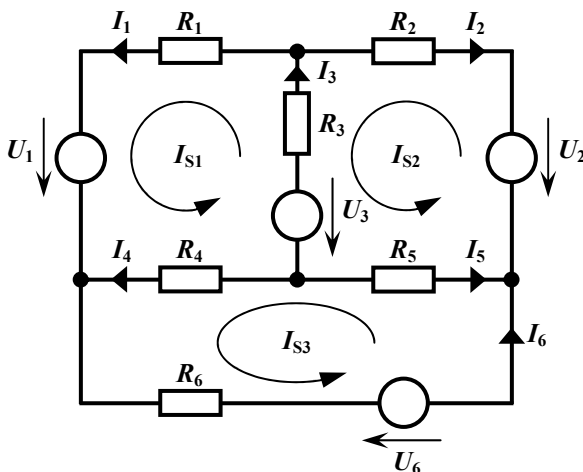
Je zřejmé, že i popisované základní metody analýzy vedou u složitějších elektrických obvodů k velkému počtu rovnic. Řešení těchto rovnic je bez použití výpočetní techniky relativně složité a zdlouhavé. Značné možnosti poskytuje programové prostředí MATLAB, které je k řešení soustav lineárních algebraických rovnic, které aplikací některé z uvedených metod obdržíme, přímo předurčeno. MATLAB integruje maticové výpočty do uživatelsky příjemného prostředí, ve kterém se řešené problémy zapisují podobně jako v matematice,

tedy bez známých obtíží klasického programování. Práce s vektory a maticemi je podrobně popisována v [4].

1.1 Text pro studenta

Příklad

Vypočítejte velikost všech neznámých proudů a napětí v obvodu na obr. 1.1-1. Jsou zadány hodnoty odporů jednotlivých rezistorů $R_1 = 4 \Omega$, $R_2 = 20 \Omega$, $R_3 = 8 \Omega$, $R_4 = 15 \Omega$, $R_5 = 5 \Omega$ a $R_6 = 10 \Omega$ a napětí stejnosměrných zdrojů $U_1 = 6 \text{ V}$, $U_2 = 3 \text{ V}$, $U_3 = 12 \text{ V}$ a $U_6 = 5 \text{ V}$.



Obr. 1.1-1: Schéma zapojení elektrického obvodu

Řešení

Příklad budeme řešit metodou **smyčkových proudů**, která je výhodná zvláště tehdy, má-li analyzovaný obvod menší počet nezávislých smyček než uzlů. Jako neznámé zavedeme smyčkové proudy I_{S1} , I_{S2} a I_{S3} . K sestavení rovnic použijeme **druhý Kirchhoffův zákon**

$$\begin{aligned} R_1 I_{S1} + U_1 + R_4 (I_{S1} - I_{S3}) - U_3 + R_3 (I_{S1} - I_{S2}) &= 0 \\ R_2 I_{S2} + R_3 (I_{S2} - I_{S1}) + U_3 + R_5 (I_{S2} - I_{S3}) - U_2 &= 0 \\ R_4 (I_{S3} - I_{S1}) + R_6 I_{S3} - U_6 + R_5 (I_{S3} - I_{S2}) &= 0 \end{aligned}$$

a po úpravě

$$\begin{aligned} (R_1 + R_3 + R_4) I_{S1} - R_3 I_{S2} - R_4 I_{S3} &= -U_1 + U_3 \\ -R_3 I_{S1} + (R_2 + R_3 + R_5) I_{S2} - R_5 I_{S3} &= U_2 - U_3 \\ -R_4 I_{S1} - R_5 I_{S2} + (R_4 + R_5 + R_6) I_{S3} &= U_6 \end{aligned}$$



Takto vzniklou soustavu lineárních algebraických rovnic zapíšeme v maticovém tvaru

$$\begin{bmatrix} R_1 + R_3 + R_4 & -R_3 & -R_4 \\ -R_3 & R_2 + R_3 + R_5 & -R_5 \\ -R_4 & -R_5 & R_4 + R_5 + R_6 \end{bmatrix} \cdot \begin{bmatrix} I_{S1} \\ I_{S2} \\ I_{S3} \end{bmatrix} = \begin{bmatrix} -U_1 + U_3 \\ U_2 - U_3 \\ U_6 \end{bmatrix}$$



po dosazení pak

$$\begin{bmatrix} 27 & -8 & -15 \\ -8 & 33 & -5 \\ -15 & -5 & 30 \end{bmatrix} \cdot \begin{bmatrix} I_{S1} \\ I_{S2} \\ I_{S3} \end{bmatrix} = \begin{bmatrix} 6 \\ -9 \\ 5 \end{bmatrix}$$

Označíme-li matici soustavy symbolem **A**, vektor neznámých smyčkových proudů **I_S** a vektor pravých stran **b**, lze vyjádřit uvedenou soustavu rovnic ve tvaru

$$\mathbf{A} \mathbf{I}_S = \mathbf{b}$$

$$\mathbf{I}_S = \mathbf{A}^{-1} \mathbf{b}$$

S použitím MATLABu (obr. 1.1-2) velmi snadno maticovou rovnici vyřešíme (matici **A**⁻¹ nazýváme inverzní maticí k matici **A**).

```

Command Window
>> A=[27 -8 -15; -8 33 -5; -15 -5 30]
A =
    27    -8   -15
    -8    33    -5
   -15    -5    30
>> b=[6; -9; 5]
b =
     6
    -9
     5
>> Is=inv(A)*b
Is =
    0.3630
   -0.1354
    0.3256
>> |

```

Obr. 1.1-2: Řešení v MATLABu

Hodnoty námi zavedených smyčkových proudů vypočtené pomocí MATLABu jsou $I_{S1} = 0,3630$ A, $I_{S2} = -0,1354$ A, $I_{S3} = 0,3256$ A. Proudů v jednotlivých větvích pak jsou

$$I_1 = I_{S1} = 0,3630 \text{ A}$$

$$I_2 = -I_{S2} = 0,1354 \text{ A}$$

$$I_3 = I_{S1} - I_{S2} = 0,3630 + 0,1354 = 0,4984 \text{ A}$$

$$I_4 = I_{S3} - I_{S1} = 0,3256 - 0,3630 = -0,0374 \text{ A}$$

$$I_5 = I_{S2} - I_{S3} = -0,1354 - 0,3256 = -0,4610 \text{ A}$$

$$I_6 = I_{S3} = 0,3256 \text{ A}$$

Napětí na jednotlivých rezistorech vypočteme pomocí **Ohmova zákona**

$$U_{R_1} = R_1 I_1 = 4 \cdot 0,3630 = 3,6684 \text{ V}$$

$$U_{R_2} = R_2 I_2 = 20 \cdot 0,1354 = 2,7080 \text{ V}$$

$$U_{R_3} = R_3 I_3 = 8 \cdot 0,4984 = 3,9872 \text{ V}$$

$$U_{R_4} = R_4 I_4 = 15 \cdot (-0,0374) = -0,5610 \text{ V}$$

$$U_{R_5} = R_4 I_5 = 5 \cdot (-0,4610) = -2,3050 \text{ V}$$

$$U_{R_6} = R_6 I_6 = 10 \cdot 0,3256 = 3,2560 \text{ V}$$

Záporná znaménka u proudů I_4 a I_5 jsou způsobena opačnou volbou orientace jejich působení (směru proudových šipek) ve schématu. Z toho vychází i orientace napětí U_{R_4} a U_{R_5} .

```

Command Window
>> R1=4; R2=20; R3=8; R4=15; R5=5; R6=10;
>> U1=6; U2=3; U3=12; U6=5;
>> A=[R1+R3+R4 -R3 -R4;-R3 R2+R3+R5 -R5;-R4 -R5 R4+R5+R6]
A =
    27    -8   -15
   -8    33    -5
  -15    -5    30
>> b=[-U1+U3; U2-U3; U6]
b =
     6
    -9
     5
>> Is=inv(A)*b
Is =
    0.3630
   -0.1354
    0.3256
>> |

```

Obr. 1.1-3: Zadáni parametrů a výpočet smyčkových proudů v MATLABu

Celé řešení v prostředí MATLAB (obr. 1.1-3) výpočet značně zjednoduší a umožní velmi jednoduchou analýzu obvodu s různými modifikacemi pouhou změnou zadaných hodnot odporů jednotlivých rezistorů a napětí napájecích zdrojů.

Středník (;) potlačuje v příkazovém řádku zobrazení zadávaných hodnot a výsledků. Vynecháme-li jej, MATLAB vypočtené hodnoty zobrazí. Používá se pro zobrazení nebo potlačení mezivýsledků, zejména výpisů rozsáhlých matic či vektorů. Středník také slouží jako oddělovač řádků při zadávání matic resp. jako oddělovač jednotlivých prvků při zadávání sloupcových vektorů.

Command Window	Command Window
>> I1=Is (1)	>> UR1=R1*I1
I1 =	UR1 =
0.3630	1.4520
>> I2=-Is (2)	>> UR2=R2*I2
I2 =	UR2 =
0.1354	2.7079
>> I3=Is (3) -Is (1)	>> UR3=R3*I3
I3 =	UR3 =
-0.0374	-0.2992
>> I4=Is (3) -Is (1)	>> UR4=R4*I4
I4 =	UR4 =
-0.0374	-0.5609
>> I5=Is (2) -Is (3)	>> UR5=R5*I5
I5 =	UR5 =
-0.4610	-2.3050
>> I6=Is (3)	>> UR6=R6*I6
I6 =	UR6 =
0.3256	3.2560
>>	>>

Obr. 1.1-4: Výpočet proudů v jednotlivých větvích a napětí na rezistorech v MATLABu

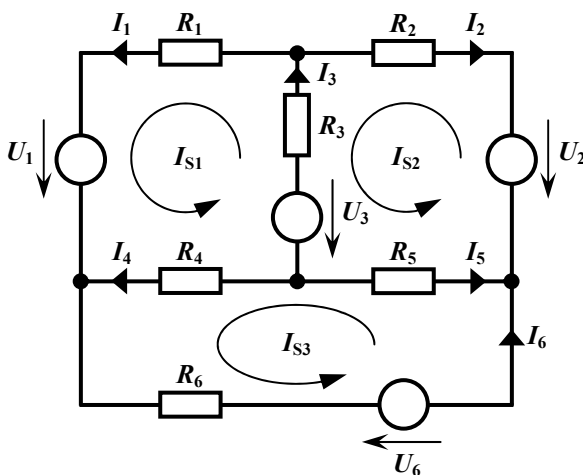
Kontrolní otázky a úkoly

- Jak se změní proud I_1 rezistorem R_1 jestliže se změní jeho odpor na polovinu?
- Jak se změní napětí U_{R_1} při této změně?
- Jak se změní výkonové zatížení tohoto rezistoru?
- Sledujte všechny tyto změny jestliže se odpor rezistoru R_1 mění v rozsahu 1 až 10 Ω . Závislosti vyjádřete tabulkou a grafem.
- Vyšetřete, jak se změní poměry v obvodu, jestliže dojde k přerušení rezistoru R_5 .
- Vyšetřete, jak se změní poměry v obvodu, jestliže se odpory všech rezistorů změní na polovinu.
- Jak se změní proudy v obvodu, jestliže se změní všechna napájecí napětí na dvojnásobek? Ověřte váš předpoklad.
- Klad'te si podobné otázky a váš předpoklad vždy ověřte výpočtem.
- Pokuste se o vytvoření jednoduchého skriptu, který vám usnadní ovládání výpočtu.
- Realizujte obvod s reálnými rezistory a srovnajte teoretický výpočet s měřením.
- Aplikujte výše popsany přístup na jiný elektrický obvod.

1.2 Text pro pedagoga

Příklad

Vypočítejte velikost všech neznámých proudů a napětí v obvodu na obr. 1.2-1. Jsou zadány hodnoty odporů jednotlivých rezistorů $R_1 = 4 \Omega$, $R_2 = 20 \Omega$, $R_3 = 8 \Omega$, $R_4 = 15 \Omega$, $R_5 = 5 \Omega$ a $R_6 = 10 \Omega$ a napětí stejnosměrných zdrojů $U_1 = 6 \text{ V}$, $U_2 = 3 \text{ V}$, $U_3 = 12 \text{ V}$ a $U_6 = 5 \text{ V}$.



Obr. 1.2-1: Schéma zapojení elektrického obvodu

Řešení

Příklad budeme řešit metodou **smyčkových proudů**, která je výhodná zvláště tehdy, má-li analyzovaný obvod menší počet nezávislých smyček než uzlů. Jako neznámé zavedeme smyčkové proudy I_{S1} , I_{S2} a I_{S3} . K sestavení rovnic použijeme **druhý Kirchhoffův zákon**

$$\begin{aligned} R_1 I_{S1} + U_1 + R_4 (I_{S1} - I_{S3}) - U_3 + R_3 (I_{S1} - I_{S2}) &= 0 \\ R_2 I_{S2} + R_3 (I_{S2} - I_{S1}) + U_3 + R_5 (I_{S2} - I_{S3}) - U_2 &= 0 \\ R_4 (I_{S3} - I_{S1}) + R_6 I_{S3} - U_6 + R_5 (I_{S3} - I_{S2}) &= 0 \end{aligned}$$

a po úpravě

$$\begin{aligned} (R_1 + R_3 + R_4) I_{S1} - R_3 I_{S2} - R_4 I_{S3} &= -U_1 + U_3 \\ -R_3 I_{S1} + (R_2 + R_3 + R_5) I_{S2} - R_5 I_{S3} &= U_2 - U_3 \\ -R_4 I_{S1} - R_5 I_{S2} + (R_4 + R_5 + R_6) I_{S3} &= U_6 \end{aligned}$$

1.2.1 Řešení pomocí lineární algebry

Výše uvedenou soustavu lineárních algebraických rovnic zapíšeme v maticovém tvaru

$$\begin{bmatrix} R_1 + R_3 + R_4 & -R_3 & -R_4 \\ -R_3 & R_2 + R_3 + R_5 & -R_5 \\ -R_4 & -R_5 & R_4 + R_5 + R_6 \end{bmatrix} \cdot \begin{bmatrix} I_{S1} \\ I_{S2} \\ I_{S3} \end{bmatrix} = \begin{bmatrix} -U_1 + U_3 \\ U_2 - U_3 \\ U_6 \end{bmatrix}$$

a dosadíme konkrétní hodnoty dle zadání

$$\begin{bmatrix} 27 & -8 & -15 \\ -8 & 33 & -5 \\ -15 & -5 & 30 \end{bmatrix} \cdot \begin{bmatrix} I_{S1} \\ I_{S2} \\ I_{S3} \end{bmatrix} = \begin{bmatrix} 6 \\ -9 \\ 5 \end{bmatrix}$$

Označíme-li matici soustavy symbolem $\mathbf{A}$, vektor neznámých smyčkových proudů $\mathbf{I}_S$ a vektor pravých stran $\mathbf{b}$, lze vyjádřit uvedenou soustavu rovnic ve tvaru

$$\mathbf{A} \mathbf{I}_S = \mathbf{b}$$

Poznámka k řešitelnosti uvedené soustavy rovnic

Řešitelnost soustavy lineárních rovnic je určena tzv. **Frobeniovou větou**. Soustava má řešení tehdy a jenom tehdy, pokud hodnost h_S matice soustavy $\mathbf{A}$, v tomto případě

$$h_S = \text{hodn } \mathbf{A} = \text{hodn} \begin{bmatrix} 27 & -8 & -15 \\ -8 & 33 & -5 \\ -15 & -5 & 30 \end{bmatrix} = 3,$$

a hodnost h_R rozšířené matice soustavy $\mathbf{A}_R = [\mathbf{A}, \mathbf{b}]$, zde

$$h_R = \text{hodn } \mathbf{A}_R = \text{hodn} \begin{bmatrix} 27 & -8 & -15 & 6 \\ -8 & 33 & -5 & -9 \\ -15 & -5 & 30 & 5 \end{bmatrix} = 3,$$

jsou **shodné**. Podmínka, že $h_S = h_R = 3$ je splněna, soustava tedy řešení má. Pokud by $h_S \neq h_R$, uvažovaná soustava by řešení neměla. Jestliže dále $h_S = h_R = n$, kde n je počet neznámých (zde $n = 3$) existuje řešení jediné. Pokud však $h_S = h_R \neq n$ existuje ∞ mnoho řešení. V našem konkrétním případě tedy řešení existuje a je jediné.

Řešení soustavy algebraických rovnic

Poznamenejme, že výše uvedený (odstavec 1.1) způsob řešení soustavy $\mathbf{A} \mathbf{I}_S = \mathbf{b}$, tedy $\mathbf{I}_S = \mathbf{A}^{-1} \mathbf{b}$ není jediný možný.

Připomeňme tzv. **Gaussovu eliminaci**, tj. postupné řádkové úpravy rozšířené matice soustavy $\mathbf{A}_R = [\mathbf{A}, \mathbf{b}]$ tak, aby matice $\mathbf{A}$ přešla v matici jednotkovou, řešení je pak dáno posledním sloupcem takto vzniklé matice.

Dalším, často používaným efektivním způsobem, je použití tzv. **Cramerova pravidla**. Jestliže matice soustavy **A** má plnou hodnotu (je regulární, její determinant je nenulový), má soustava jednoznačné řešení, které můžeme postupně pro jednotlivé proměnné vyjádřit jako podíl D_i / D , kde $D = \det \mathbf{A}$ a D_i je determinant matice, která vznikne z matice **A** záměnou i -tého sloupce sloupcem pravých stran rovnic.

$$D = \det \mathbf{A} = \begin{vmatrix} 27 & -8 & -15 \\ -8 & 33 & -5 \\ -15 & -5 & 30 \end{vmatrix} = 15510$$

$$I_{S1} = \frac{D_1}{D} = \frac{\begin{vmatrix} 6 & -8 & -15 \\ -9 & 33 & -5 \\ 5 & -5 & 30 \end{vmatrix}}{15510} = \frac{5630}{15510} = 0,3630 \text{ A}$$

$$I_{S2} = \frac{D_2}{D} = \frac{\begin{vmatrix} 27 & 6 & -15 \\ -8 & -9 & -5 \\ -15 & 5 & 30 \end{vmatrix}}{15510} = -\frac{2100}{15510} = -0,1354 \text{ A}$$

$$I_{S3} = \frac{D_3}{D} = \frac{\begin{vmatrix} 6 & -8 & -15 \\ -9 & 33 & -5 \\ 5 & -5 & 30 \end{vmatrix}}{15510} = \frac{5050}{15510} = 0,3256 \text{ A}$$

Proudy v jednotlivých větvích pak jsou

$$I_1 = I_{S1} = 0,3630 \text{ A}$$

$$I_2 = -I_{S2} = 0,1354 \text{ A}$$

$$I_3 = I_{S1} - I_{S2} = 0,3630 + 0,1354 = 0,4984 \text{ A}$$

$$I_4 = I_{S3} - I_{S1} = 0,3256 - 0,3630 = -0,0374 \text{ A}$$

$$I_5 = I_{S2} - I_{S3} = -0,1354 - 0,3256 = -0,4610 \text{ A}$$

$$I_6 = I_{S3} = 0,3256 \text{ A}$$

Napětí na jednotlivých rezistorech vypočteme pomocí **Ohmova zákona**

$$U_{R_1} = R_1 I_1 = 4 \cdot 0,3630 = 3,6684 \text{ V}$$

$$U_{R_2} = R_2 I_2 = 20 \cdot 0,1354 = 2,7080 \text{ V}$$

$$U_{R_3} = R_3 I_3 = 8 \cdot 0,4984 = 3,9872 \text{ V}$$

$$U_{R_4} = R_4 I_4 = 15 \cdot (-0,0374) = -0,5610 \text{ V}$$

$$U_{R_5} = R_4 I_5 = 5 \cdot (-0,4610) = -2,3050 \text{ V}$$

$$U_{R_6} = R_6 I_6 = 10 \cdot 0,3256 = 3,2560 \text{ V}$$

Záporná znaménka u proudů I_4 a I_5 jsou způsobena opačnou volbou orientace jejich působení (směru proudových šipek) ve schématu. Z toho vychází i orientace napětí U_{R_4} a U_{R_5} .

Poznámka k výpočtu inverzní matice

Inverzní matice $\mathbf{A}^{-1}$ je taková čtvercová matice, pro kterou platí $\mathbf{A}^{-1} \mathbf{A} = \mathbf{A} \mathbf{A}^{-1} = \mathbf{E}$, kde $\mathbf{E}$ je jednotková matice (jedničky na hlavní diagonále, ostatní prvky nulové). Pro každou čtvercovou regulární matici $\mathbf{A}$ ($\det \mathbf{A} \neq 0$) existuje právě jediná $\mathbf{A}^{-1} = \text{adj } \mathbf{A} / \det \mathbf{A}$. Matice $\text{adj } \mathbf{A}$ je tzv. adjungovanou maticí sestavenou z algebraických doplňků matice $\mathbf{A}$ (prvek na pozici (i, j) této matice dostaneme jako determinant matice vzniklé vypuštěním i -tého řádku a j -tého sloupce transponované matice $\mathbf{A}^T$ vynásobený koeficientem $(-1)^{i+j}$). V našem případě

$$\text{adj } \mathbf{A} = \begin{bmatrix} 965 & 315 & 535 \\ 315 & 585 & 255 \\ 535 & 255 & 827 \end{bmatrix}, \quad \det \mathbf{A} = \begin{vmatrix} 27 & -8 & -15 \\ -8 & 33 & -5 \\ -15 & -5 & 30 \end{vmatrix} = 15510$$

$$\mathbf{A}^{-1} = \frac{\text{adj } \mathbf{A}}{\det \mathbf{A}} = \begin{bmatrix} 0,062 & 0,020 & 0,034 \\ 0,020 & 0,038 & 0,016 \\ 0,034 & 0,016 & 0,053 \end{bmatrix}$$

Výpočet inverzní matice pro nejčastější případ rozměru matice 2×2 je zcela triviální (záměna prvků ležících na hlavní diagonále a inverze znaménka prvků ostatních, vše děleno determinantem).

Výpočet determinantu Sarrusovým pravidlem je možný pouze u matic o rozměru 2×2 a 3×3 . U matic větších rozměrů je nutné při výpočtu použít rozvoj podle libovolného řádku nebo sloupce.

1.2.2 Řešení v prostředí MATLABu

S použitím MATLABu lze maticovou rovnici $\mathbf{A} \mathbf{I}_S = \mathbf{b}$ velmi snadno vyřešit. Řešení hledáme ve tvaru $\mathbf{I}_S = \mathbf{A}^{-1} \mathbf{b}$ nebo případně ve tvaru $\mathbf{I}_S = \mathbf{A} \backslash \mathbf{b}$, použijeme-li dělení zleva, viz obr. 1.2-2.

```
Command Window
>> A=[27 -8 -15;-8 33 -5;-15 -5 30];
>> b=[6;-9;5];
>> Is=inv(A)*b
Is =
    0.3630
   -0.1354
    0.3256
>> |
```

```
Command Window
>> A=[27 -8 -15;-8 33 -5;-15 -5 30];
>> b=[6;-9;5];
>> Is=A\b
Is =
    0.3630
   -0.1354
    0.3256
>> |
```

Obr. 1.2-2: Dvě různé varianty řešení v MATLABu

Celý postup výpočtu smyčkových proudů, skutečných proudů v jednotlivých větvích a napětí na rezistorech se značně zjednoduší, využijeme-li k řešení prostředí MATLAB, viz obr. 1.2-3.

```

Command Window
>> R1=4; R2=20; R3=8; R4=15; R5=5; R6=10;
>> U1=6; U2=3; U3=12; U6=5;
>> A=[R1+R3+R4 -R3 -R4;-R3 R2+R3+R5 -R5;-R4 -R5 R4+R5+R6];
>> b=[-U1+U3;U2-U3;U6];
>> Is=inv(A)*b;
>> I1=Is(1), I2=-Is(2), I3=Is(3)-Is(1), I4=Is(3)-Is(1), I5=Is(2)-Is(3), I6=Is(3)
I1 =
    0.3630
I2 =
    0.1354
I3 =
   -0.0374
I4 =
   -0.0374
I5 =
   -0.4610
I6 =
    0.3256
>> UR1=R1*I1, UR2=R2*I2, UR3=R3*I3, UR4=R4*I4, UR5=R5*I5, UR6=R6*I6
UR1 =
    1.4520
UR2 =
    2.7079
UR3 =
   -0.2992
UR4 =
   -0.5609
UR5 =
   -2.3050
UR6 =
    3.2560
>> |

```

Obr. 1.2-3: Kompletní řešení v prostředí MATLAB

Výše uvedené řešení v MATLABu (obr. 1.2-3) lze také zapsat do skriptu, který je pak možné opakovaně volat. Jednoduchým způsobem je takto možné provést analýzu obvodu pro různé hodnoty zadaných parametrů. Skript je také možné doplnit o komentáře, které jsou uvozené znakem procenta (%). Komentáře mohou být na samostatných řádcích, ale i za běžnými příkazy. Skript (obr. 1.2-4) je rozdělen na několik částí. V první části jsou zadávány parametry odporů jednotlivých rezistorů a napětí stejnosměrných zdrojů. V druhé části je pak proveden vlastní výpočet. Pomocí příkazu **disp** je logicky členěn i výstup na obrazovku. Vstupní parametry mohou být zadávány i přímo z příkazové řádky MATLABu. Např. zadání hodnoty odporu rezistoru R_1 lze realizovat pomocí příkazu **input** následujícím způsobem: **R1 = input('hodnota odporu R1: ');** Proměnné R1 bude takto přiřazena zadaná hodnota, viz obr. 1.2-5.

```

1 %
2 % Kirchhoffovy zákony
3 % Skript pro výpočet napěťových a proudových poměrů v el. obvodu
4 %
5 clear all, close all, clc
6
7 disp(' ')
8 disp(' ')
9 disp('*****')
10 disp(' Kirchhoffovy zákony')
11 disp(' VÝPOČET NAPĚŤOVÝCH A PROUDOVÝCH POMĚRŮ V ELEKTRICKÉM OBVDU')
12 disp('*****')
13 disp(' ')
14 disp(' Parametry obvodu : ')
15 disp(' ')
16
17 % Zadání parametrů obvodu
18 R1=4, R2=20, R3=8, R4=15, R5=5, R6=10
19 U1=6, U2=3, U3=12, U6=5
20 % Konec pole zadání parametrů
21
22 disp('*****')
23 disp(' ')
24 disp(' Vypočtené hodnoty :')
25
26 A=[R1+R3+R4 -R3 -R4;-R3 R2+R3+R5 -R5;-R4 -R5 R4+R5+R6];
27 b=[-U1+U3;U2-U3;U6];
28 Is=inv(A)*b;
29 I1=Is(1), I2=-Is(2), I3=Is(1)-Is(2), I4=Is(3)-Is(1), I5=Is(2)-Is(3), I6=Is(3)
30 UR1=R1*I1, UR2=R2*I2, UR3=R3*I3, UR4=R4*I4, UR5=R5*I5, UR6=R6*I6
31
32 disp('*****')
33 disp(' Konec příkladu')
34 disp('*****')
35 disp(' ')

```

Obr. 1.2-4: Příklad možného skriptu

```

1 % Zadání parametrů obvodu
2 R1=input('hodnota odporu R1: ')
3 R2=input('hodnota odporu R2: ')
4
5

```

Command Window

```

hodnota odporu R1: 4
R1 =
    4
hodnota odporu R2: 20
R2 =
    20
>>

```

Obr. 1.2-5: Zadávání hodnot proměnných z příkazového řádku

Probraná témata

- metoda smyčkových proudů, Kirchhoffovy zákony,
- řešení soustav lineárních algebraických rovnic,
- maticové vyjádření soustavy lineárních algebraických rovnic,
- matematické aspekty řešení (řešitelnost soustavy, Frobeniova věta, Cramerovo pravidlo, výpočet inverzní matice),
- realizace řešení v prostředí MATLABu, práce se skriptem.

Stručné shrnutí

- metoda smyčkových proudů je výhodná zvláště tehdy, má-li analyzovaný obvod menší počet nezávislých smyček než uzlů, k sestavení rovnic použijeme druhý Kirchhoffův zákon,
- řešitelnost soustavy lineárních rovnic je určena tzv. Frobeniovou větou,
- často používaným efektivním způsobem řešení soustavy lineárních algebraických rovnic, je použití Cramerova pravidla,
- při práci v MATLABu rozlišujeme malá a velká písmena,
- jednotlivé příkazy jsou odděleny čárkou, středníkem nebo novým řádkem,
- středník potlačuje zobrazení vypočtených hodnot na obrazovce počítače,
- prvky matice se zadávají do hranatých závorek po řádcích, jednotlivé prvky jsou na řádku odděleny mezerou nebo čárkou, řádky jsou navzájem odděleny středníkem,
- chceme-li opakovat výpočet a případně také měnit vstupní parametry, je vhodné pro tyto účely vytvořit skript.

Přiložené soubory

- soubor skriptu: `analiza_obvodu.m`

2 Rezonanční vlastnosti RLC členu

RLC členy jsou obvody, které jsou připojeny ke zdroji střídavého napětí a jsou obecně tvořeny rezistorem o odporu R , ideální cívku s indukčností L a ideálním kondenzátorem s kapacitou C . Obvykle se tyto obvody označují jako obvody rezonanční. Jednoduché rezonanční obvody, sériové nebo paralelní, tvoří vždy komplexní jednobran. Obvody serio-paralelní pak tvoří komplexní dvojbran. Při tzv. rezonanční frekvenci se v těchto obvodech navzájem vyrovná působení indukční a kapacitní reaktance (induktance a kapacitance) a celý obvod se chová jako čistá rezistance. Rezonanční obvody se používají jako pasivní filtry, horní a dolní propusti, pásmové propusti či zádrže.

Problémem při řešení takovýchto obvodů může být komplikovanost obdržených rovnic a to mnohdy i ve velmi jednoduchých případech. V teorii obvodů existuje několik způsobů, jak problém elegantně zjednodušit.

Řešení střídavých obvodů, napájených jedním nebo více zdroji se stejnou frekvencí, představuje dvojrozměrný problém. Napájíme-li obvod napětím $u(t) = U_0 \sin \omega t$, budou jednotlivá napětí a proudy záviset na čase. Je tedy postačující popsat každou veličinu v každé větvi dvěma parametry, velikostí a fází. Používá se jeden z matematických nástrojů: dvojrozměrné vektory nebo komplexní čísla v Gaussově rovině. Popis pomocí komplexních čísel je výhodnější, protože je pro ně definováno více operací (např. dělení, mocnění a odmocňování). Souřadná soustava nebo Gaussova rovina rotuje s ωt , takže zobrazujeme jen velikost a fázi veličin – hovoříme tedy o fázorech. Popis oběma způsoby je podobný. Velikost příslušné veličiny (napětí nebo proudu) je popsána velikostí fázoru nebo absolutní hodnotou komplexního čísla a fáze je popsána úhlem, který svírají s kladnou částí osy x nebo reálné osy.

Pro komplexní impedanci resp. admitanci platí následující vztahy. j je imaginární jednotka definovaná $j^2 = -1$.

$R:$	$Z_R = R$	$Y_R = 1/R$
$L:$	$Z_L = j\omega L$	$Y_L = -j/\omega L$
$C:$	$Z_C = -j/\omega C$	$Y_C = j\omega C$

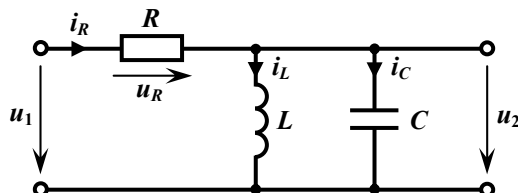
Na základě uvedeného lze jednoduše odvodit celkovou impedanci obvodu a tedy i příslušný přenos a lze zkonstruovat frekvenční charakteristiky popisovaného obvodu.

Značné možnosti poskytuje programové prostředí MATLAB, především jeho grafická nadstavba Simulink. Chování rezonančních obvodů lze velmi jednoduchým způsobem v tomto prostředí simulovat. Můžeme také měnit hodnoty vstupních parametrů a sledovat průběh odezvy obvodu při buzení nejrůznějšími typy signálů. Konstrukce frekvenčních charakteristik je v MATLABu triviální.

2.1 Text pro studenta

Příklad

Analyzujte a v prostředí MATLAB – Simulink simulujte chování rezonančního obvodu na obr. 2.1-1. Hodnoty prvků pro jednoduchost zpočátku uvažujte $R = L = C = 1$.

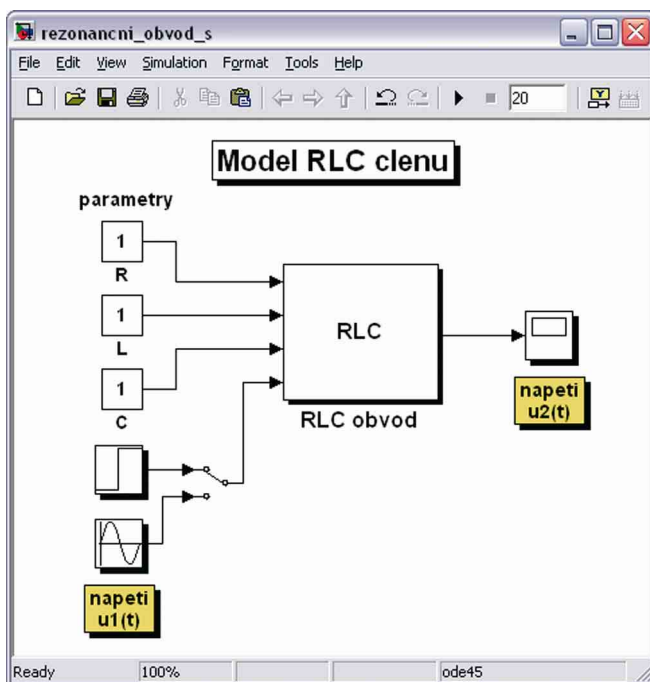


Obr. 2.1-1: Schéma zapojení elektrického obvodu



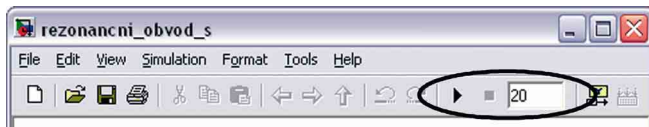
Řešení

Při zkoumání rezonančních vlastností RLC členu na obr. 2.1-1, vycházíme ze základních vztahů pro střídavé napětí a proud. Odvození diferenciální rovnice je ale relativně složité. K analýze vlastností RLC obvodu tedy použijeme již vytvořený model, který je v souboru rezonanci_obvod_s.mdl. Naším úkolem bude sledovat odezvu modelu obvodu při buzení různými typy vstupních signálů. Simulační model RLC obvodu je na obr. 2.1-2.



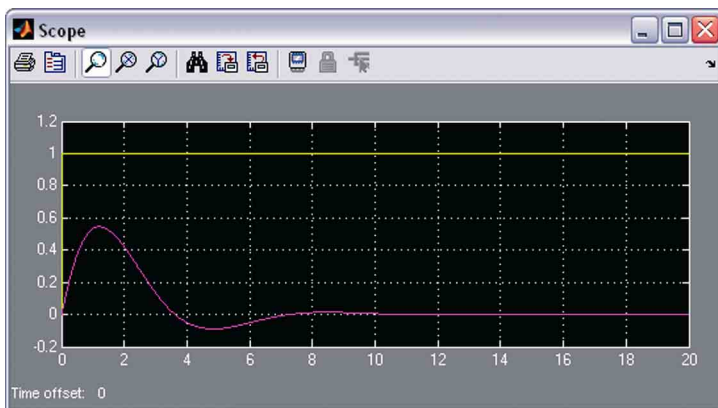
Obr. 2.1-2: Analýza vlastností RLC členu v Simulinku

Model RLC obvodu na obr. 2.1-2 je již ve stavu, kdy je možné přejít k vlastní simulaci. Všechny potřebné parametry jsou zadány. Simulaci lze spustit z nabídky okna modelu **Simulation** → **Start** (nebo Ctrl + T). Rychlejší je ale spuštění prostřednictvím ikony ► v toolbaru okna, viz obr. 2.1-3. Napravo od této ikony je také možné měnit délku simulace. Pokud dojde k neočekávaným problémům v průběhu simulace nebo pokud je zvolen příliš dlouhý čas výpočtu, lze simulaci zastavit pomocí ikony ■.

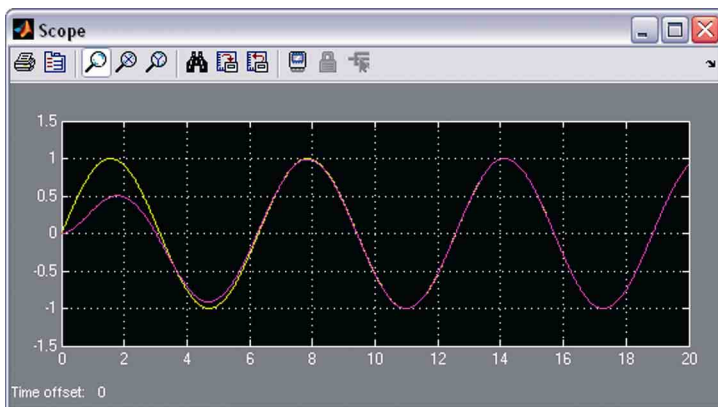


Obr. 2.1-3: Rychlé ovládání běhu simulace

Po ukončení výpočtu můžeme dvojitým kliknutím na ikonu zobrazovače (napeti $u_2(t)$) otevřít okno s grafickým záznamem řešení. Na obr. 2.1-4 vidíme průběh skokové budicí funkce (jednotkový skok) a průběh odezvy, zde konkrétně přechodové charakteristiky.

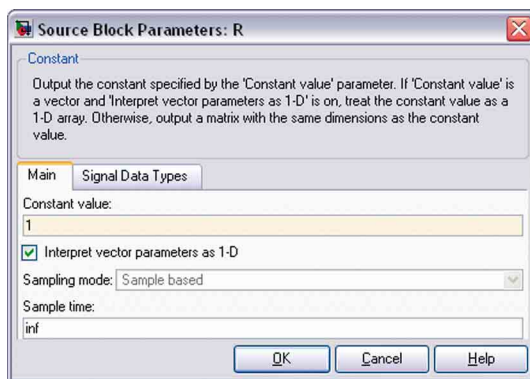


Obr. 2.1-4: Zobrazení řešení

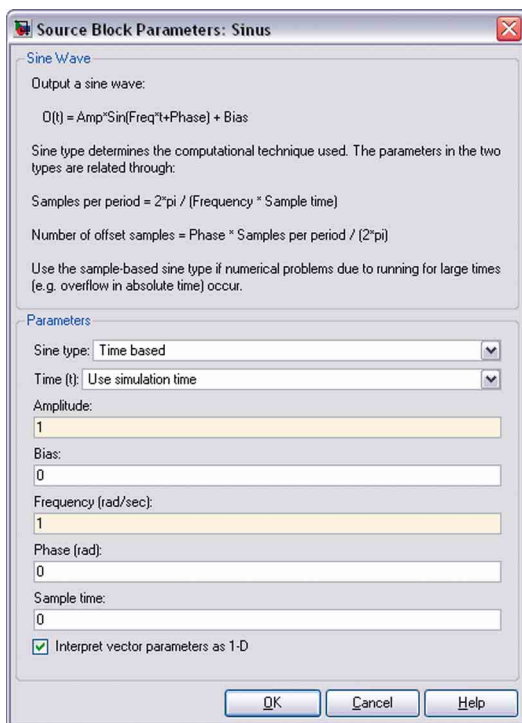
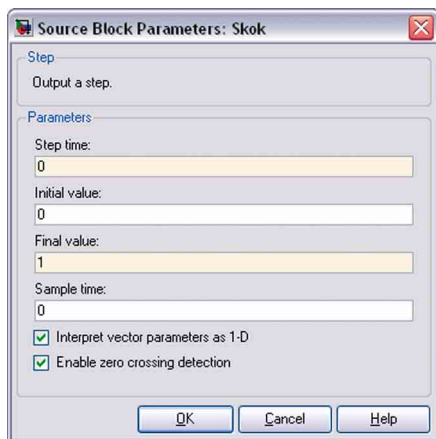


Obr. 2.1-5: Záznam řešení při harmonickém buzení modelu

Pomocí dvoupolohového přepínače je také možné zvolit jiný typ vstupního signálu. Dvojitým kliknutím na bloku přepínače zvolíme signál harmonický (sinus). Po proběhnutí simulace můžeme opět v bloku zobrazovače sledovat průběh budicí funkce a odezvy, viz obr. 2.1-5.



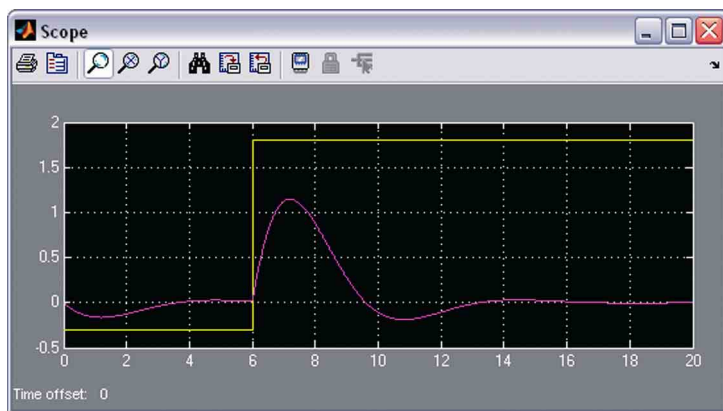
Obr. 2.1-6: Změna hodnot vstupních parametrů



Obr. 2.1-7: Změna parametrů budících funkcí

V modelu je samozřejmě možné měnit některé parametry (např. hodnoty R , L , C) podle potřeby. Na obr. 2.1-6 je uvedeno dialogové okno pro zadávání hodnoty odporu rezistoru. V záložce **Main** tohoto okna zapíšeme v položce **Constant value** požadovanou hodnotu.

Jednoduše lze změnit i parametry obou budicích funkcí (obr. 2.1-7). V případě skokové funkce lze měnit v položce **Step time** dobu skoku a v položkách **Initial value** a **Final value** počáteční a konečnou hladinu skoku. Chceme-li např. vykonat skok v čase 6 s z hodnoty $-0,3$ na hodnotu $1,8$, zadáme postupně **Step time** 6, **Initial value** $-0,3$ a **Final value** $1,8$. Výsledné průběhy jsou uvedeny na obr. 2.1-8. V případě harmonické funkce (sinus) lze měnit v položce **Amplitude** amplitudu, dále v položce **Bias** velikost stejnosměrné složky, v položce **Frequency** frekvenci v rad/s a v položce **Phase** fázový posuv v radiánech.



Obr. 2.1-8: Jiný průběh skokové budicí funkce

Kontrolní otázky a úkoly

- Simulujte chování zadaného RLC členu s uvažováním obou typů budicích funkcí, měňte vhodně hodnoty R , L a C a sledujte odezvu modelu.
- Předpokládejme zadané hodnoty parametrů, při frekvenci budicího harmonického signálu $f_{u1} = 1$ rad/s je obvod v rezonanci. Jaká bude perioda vybuzeného signálu (napětí $u_2(t)$) v ustáleném stavu? Vaši hypotézu ověřte odečtem z grafu.
- Uvažujte tyto hodnoty prvků: $R = 0,5 \Omega$, $L = 5$ H, $C = 200 \mu\text{F}$. Nalezněte frekvenci, při které bude uvažovaný RLC v rezonanci. Výpočet následně ověřte simulací.
- Simulujte chování RLC členu s reálnými hodnotami použitých součástek.
- Podle pokynů učitele vypracujte krátkou technickou zprávu.

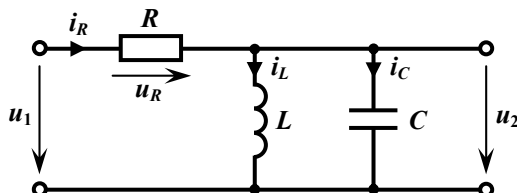
Přiložené soubory

- soubor modelu: rezonancni_obvod_s.mdl

2.2 Text pro pedagoga

Příklad

Analyzujte a v prostředí MATLAB – Simulink simulujte chování rezonančního obvodu na obr. 2.2-1. Hodnoty prvků pro jednoduchost zpočátku uvažujte $R = L = C = 1$.



Obr. 2.2-1: Schéma zapojení elektrického obvodu

Řešení

Při zkoumání rezonančních vlastností RLC členu na obr. 2.2-1, vycházíme ze základních vztahů pro střídavé napětí a proud. Je-li v obvodu více prvků (R , L nebo C), je pro každý prvek možné sestavit odpovídající integrální a diferenciální rovnici

$$\left. \begin{aligned} u_R &= i_R R \\ u_C &= \frac{1}{C} \int_0^t i_C(\tau) d\tau \\ u_L &= L \frac{di_L}{dt} \end{aligned} \right\} u_2 = u_L = u_C \quad \begin{aligned} i_C &= C \frac{du_C}{dt} \\ i_L &= \frac{1}{L} \int_0^t u_L(\tau) d\tau \end{aligned}$$

Dosazením do rovnice pro napětí v obvodu

$$u_1 = u_R + u_2 = i_R R + u_2 = (i_L + i_C) R + u_2$$

$$u_1 = \left(C \frac{du_2}{dt} + \frac{1}{L} \int_0^t u_L(\tau) d\tau \right) R + u_2$$

Výše uvedenou integro-diferenciální rovnici dále můžeme derivovat podle času, formálně označíme $du_2/dt = u_2'$, $d^2u_2/dt^2 = u_2''$, a získat tak diferenciální rovnici popisující zkoumaný systém – RLC člen

$$RCu_2'' + u_2' + \frac{R}{L}u_2 = u_1'$$

Výslednou diferenciální rovnici budeme řešit v prostředí MATLAB – Simulink. Chování RLC členu budeme simulovat, nejprve však musíme navrhnout simulační schéma.

2.2.1 Vytvoření simulačního modelu

Vycházíme z výše uvedené diferenciální rovnice a nejprve vyjádříme nejvyšší derivaci u_2'' jako lineární kombinaci nižších derivací a buzení (napájecí napětí u_1)

$$u_2'' = -\frac{1}{RC} u_2' - \frac{1}{LC} u_2 + \frac{1}{RC} u_1'$$

Sestavení simulačního schématu provedeme s využitím **metody postupné integrace**, která je v tomto případě vynucena derivací na pravé straně diferenciální rovnice, tj. u_1' . Rovnici budeme dvakrát integrovat

$$u_2 = -\frac{1}{RC} \int u_2' - \frac{1}{LC} \iint u_2 + \frac{1}{RC} \int u_1$$

Z důvodu zjednodušení zápisu jsou jednotlivé integrály zapsány bez příslušných mezí.

Např. integrálu $\int_0^t u_2(\tau) d\tau$ odpovídá zápis $\int u_2$, obdobně $\iint_0^t u_2(\tau) d\tau^2$ odpovídá $\iint u_2$.

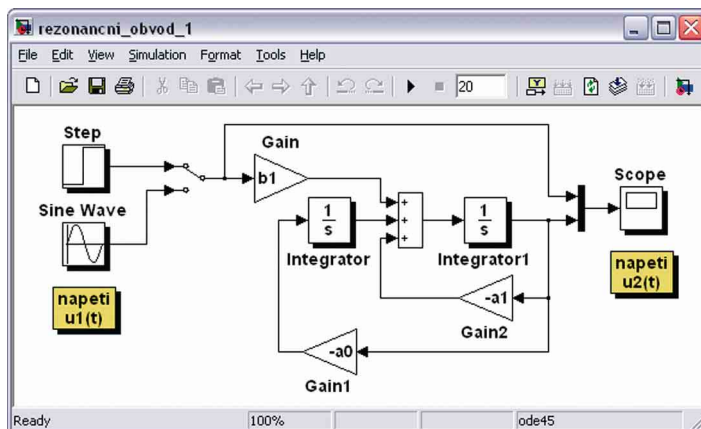
Zavedeme-li označení

$$a_0 = \frac{1}{LC} ; \quad a_1 = \frac{1}{RC} ; \quad b_1 = \frac{1}{RC}$$

lze psát

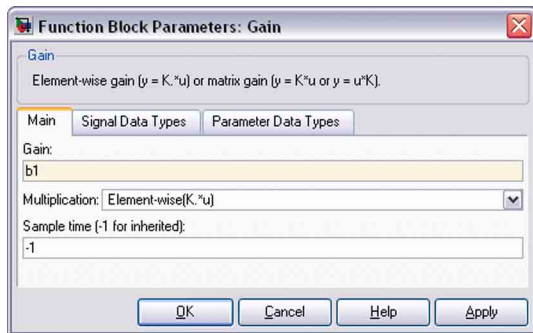
$$u_2 = -a_1 \int u_2 - a_0 \iint u_2 + b_1 \int u_1$$

Simulační schéma (model) RLC členu je na obr. 2.2-2. Technice vytváření simulačních modelů se podrobně věnuje kniha [4]. Kliknutím na symbol (ikonu) některého z bloků v programovém okně Simulinku se otevře příslušné dialogové okno, ve kterém nastavíme požadované parametry.



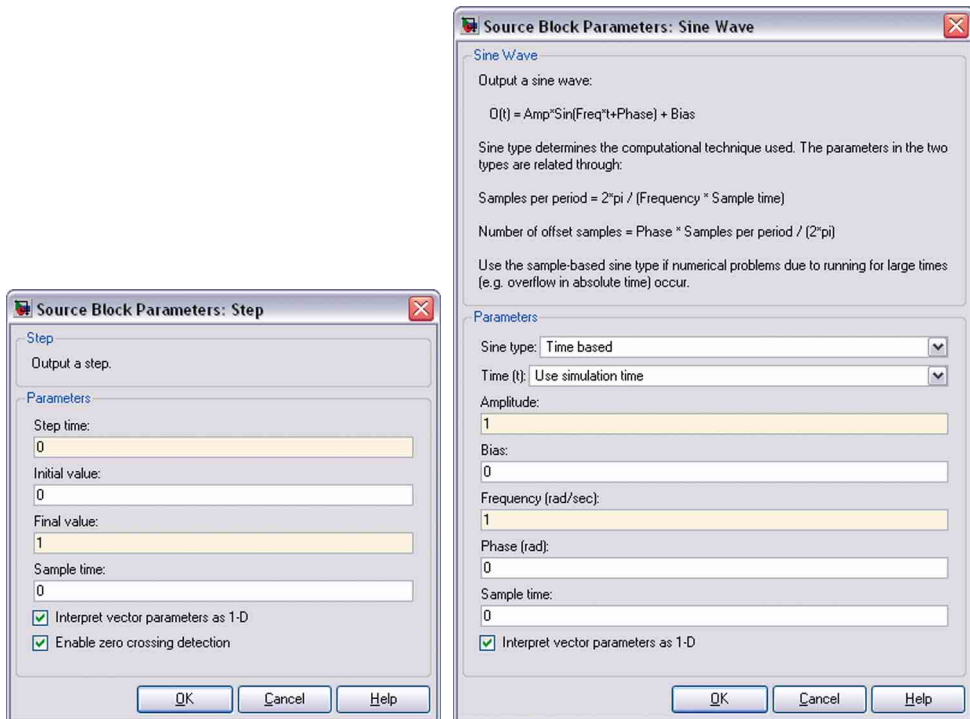
Obr. 2.2-2: Simulační model RLC členu

Koeficienty v zesilovačích (bloky *Gain*) budeme definovat pomocí proměnných. Na obr. 2.2-3 je okno parametrů jednoho ze zesilovačů. V položce **Gain** záložky okna **Main** je v tomto případě zadána proměnná *b1*.



Obr. 2.2-3: Změna parametrů zesilovače

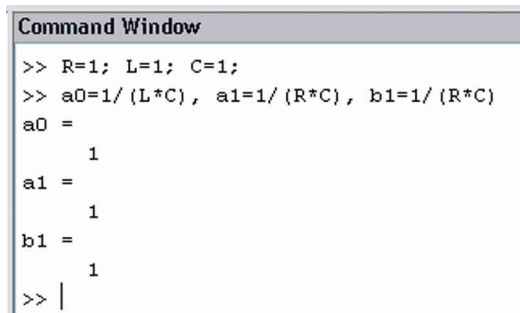
Pomocí dvoupolohového přepínače, dvojitým kliknutím na ikoně bloku, je také možné zvolit typ vstupního signálu. Jednoduše lze také změnit parametry obou budících funkcí (obr. 2.2-4). V případě skokové funkce lze měnit v položce **Step time** dobu skoku a v položkách **Initial value** a **Final value** počáteční a konečnou hladinu skoku.



Obr. 2.2-4 Změna parametrů budících funkcí

V případě harmonické funkce (sinus) lze měnit v položce **Amplitude** amplitudu, dále v položce **Bias** velikost stejnosměrné složky signálu (posun), v položce **Frequency** frekvenci v rad/s a v položce **Phase** fázový posuv v radiánech.

Hodnoty koeficientů a_0 , a_1 a b_1 jsou zadány pomocí proměnných. Konkrétní hodnoty je tedy třeba ještě definovat v příkazovém řádku v okně *Command Window* MATLABu, viz obr. 2.2-5.



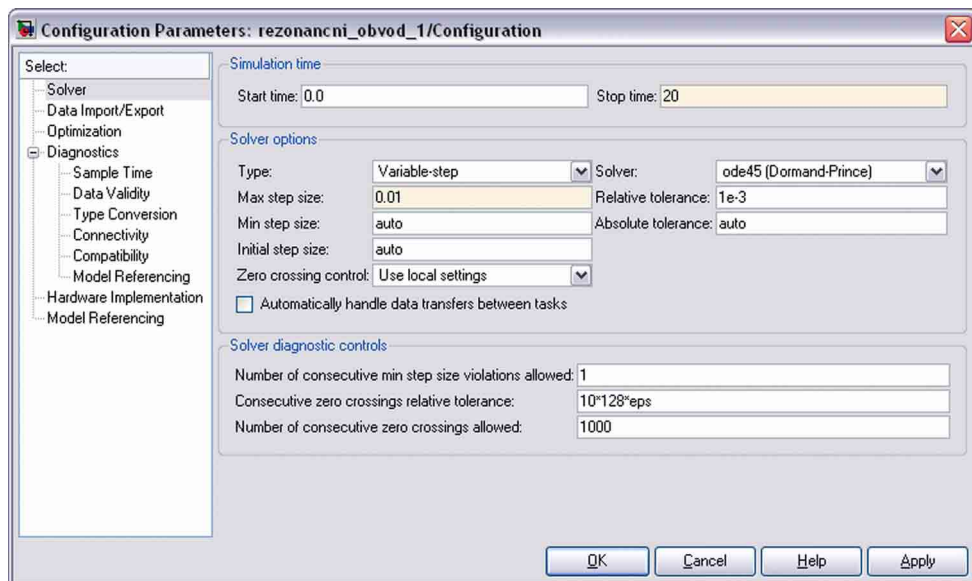
```

Command Window

>> R=1; L=1; C=1;
>> a0=1/(L*C), a1=1/(R*C), b1=1/(R*C)
a0 =
    1
a1 =
    1
b1 =
    1
>> |
  
```

Obr. 2.2-5: Definování proměnných a výpočet koeficientů diferenciální rovnice

Před spuštěním simulace je možné nastavit základní parametry simulace. Po otevření nabídky **Simulation** z okna modelu a výběru položky **Configuration Parameters...** (nebo po stisku Ctrl + E) se otevře okno parametrů simulace (obr. 2.2-6) s několika záložkami (jsou v levé části okna v nabídce *Select*). Pro základní práci s prostředím Simulink obvykle postačuje změnit nastavení v záložce **Solver**.



Obr. 2.2-6: Změna základních parametrů simulace

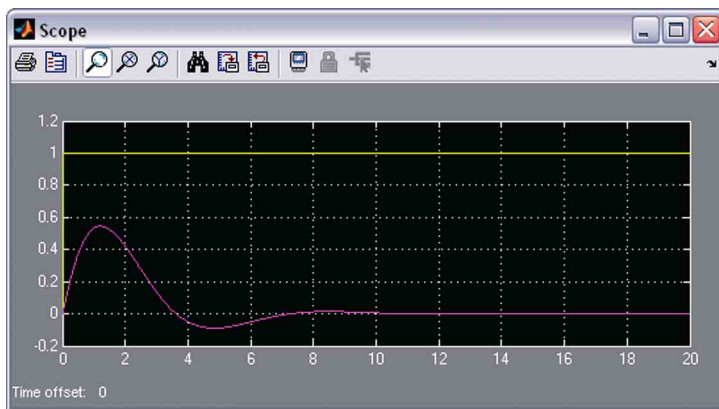
V záložce **Solver** je nejdůležitější nastavit v části *Simulation time* počátek simulace **Start time**, zde zpravidla ponecháme nulu, a zejména pak konec resp. délku simulace **Stop time**. Nastavení numerické metody výpočtu, kroku výpočtu a tolerancí je možné v části okna *Solver options*. Zde obvykle ponecháme parametr týkající se velikosti kroku (**Variable-step**, druhou variantou je **Fixed-step**) a pokud chceme zvýšit přesnost výpočtu, změníme pouze hodnoty **Max step size**, **Min step size** a **Initial step size**.

Implicitní nastavení u všech těchto parametrů je **auto**. Parametr **Max step size** je Simulinkem volen automaticky podle zvolené doby simulace $dt_{\max} = (t_{\text{stop}} - t_{\text{start}}) / 50$. V našem případě, kdy jsme zvolili čas simulace 20 s, Simulink nastaví automaticky podle uvedeného vzorce tuto hodnotu na 0,4 s (v příkazovém okně MATLABu je dokonce zobrazeno i hlášení – Warning). Pro zvýšení přesnosti nastavíme **Max step size** na hodnotu 0,01, viz obr. 2.2-6.

2.2.2 Spuštění simulace a zobrazení výsledků

Simulaci lze spustit z nabídky okna modelu **Simulation** → **Start** (Ctrl + T). Rychlejší je ale spuštění prostřednictvím ikony ► v toolbaru okna. Napravo od této ikony je také možné měnit délku simulace. Pokud dojde k neočekávaným problémům v průběhu simulace nebo pokud je zvolen příliš dlouhý čas výpočtu (resp. příliš jemný krok výpočtu), lze simulaci zastavit pomocí ikony ■.

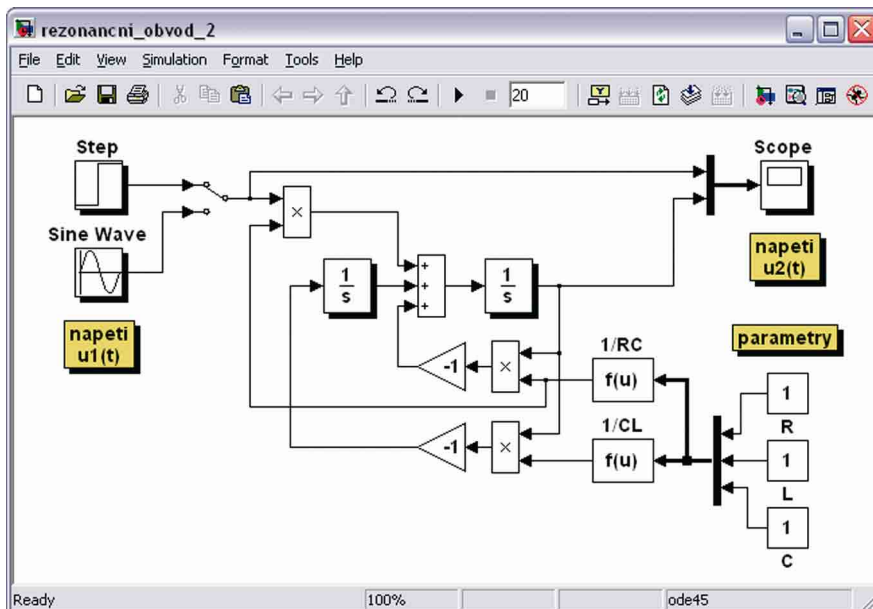
Po ukončení výpočtu, dvojitým kliknutím na ikonu zobrazovače ve schématu (v našem případě blok *Scope*) otevřeme okno s grafickým záznamem řešení diferenciální rovnice. Na obr. 2.2-7 jsou průběhy budící skokové funkce (jednotkový skok) a příslušné odezvy.



Obr. 2.2-7: Zobrazení řešení

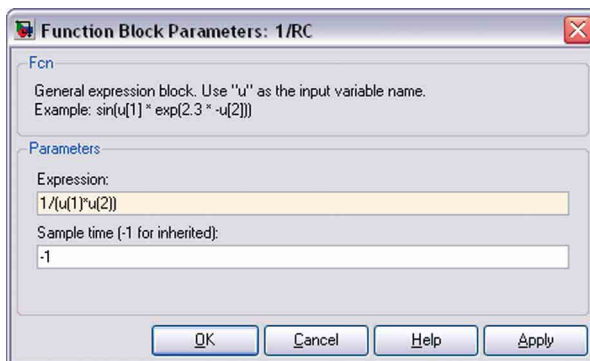
Pokud by nás zajímala reakce systému vybuzeného jinou než skokovou budící funkcí, jednoduše nahradíme blok *Step* jiným blokem z bohaté nabídky Simulinku. Další možností je umístění více bloků budících funkcí a použití analogového přepínače (*Manual Switch*) k volbě mezi nimi, tak jak je to provedeno v demonstrovaném příkladě na obr. 2.2-2.

Chceme-li nyní zkoumat vlastnosti RLC členu v závislosti na parametrech R , L a C a zároveň se i vyhnout poněkud složitějšímu zadávání koeficientů z příkazového řádku MATLABu, otevřeme z prostředí Simulinku příložený soubor `rezonanci_obvod_2.mdl`. Upravené simulační schéma je na obr. 2.2-8. Hodnoty R , L a C jsou zde zadávány pomocí bloků *Constant*. Pomocí bloku *Mux* (multiplexor) jsou jednotlivé signály sdruženy do jednoho. Vektorové spoje je možno pro přehlednost zvýraznit. V položce menu **Format Port/Signal Displays** zvolíme **Wide Nonscalar Lines**.



Obr. 2.2-8: Modifikovaný simulační model

Výpočet koeficientů a_0 , a_1 a b_1 je ve schématu realizován pomocí bloků *Fcn* (označeny $1/RC$ a $1/CL$) sloužících pro definování funkcí. K hodnotám R , L a C je třeba přistupovat výběrem z vektoru. Ukázka zápisu výrazu do funkčního bloku je uvedena na obr. 2.2-9.



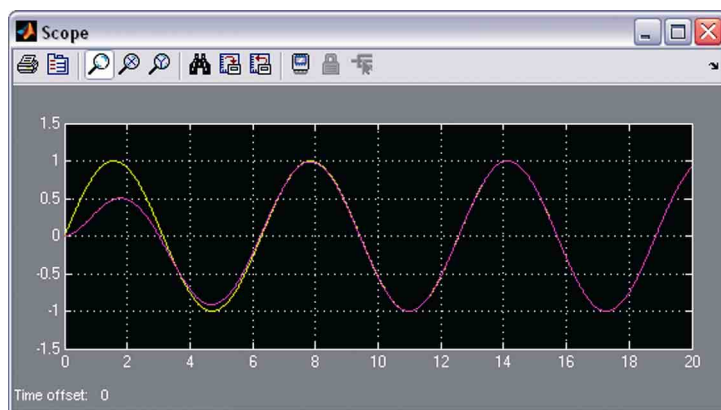
Obr. 2.2-9: Zápis výrazu v bloku *Fcn*

Nyní se budeme podrobněji zabývat rezonančními vlastnostmi RLC členu. Jako budící funkci zvolíme funkci sinus, použijeme blok *Sine Wave* resp. přepneme přepínač do příslušné polohy. Konfigurace bloku byla provedena již dříve. Připomeňme pouze, že pro jednoduchost byly nastaveny tyto hodnoty: amplituda $U_1 = 1$, frekvence $f_{u_1} = 1$ rad/s. Fáze a střední hodnota byly ponechány nulové.

Na obr. 2.2-10 jsou průběhy budící sinové funkce a odpovídající odezvy. Je zřejmé, že ve stavu rezonance ($\omega L = 1/\omega C$) a pro $R = L = C = 1$ platí

$$\omega = \frac{1}{\sqrt{LC}} = 1; \quad T = \frac{2\pi}{\omega} = 2\pi$$

Snadno se můžeme přesvědčit z průběhu získaného simulací, že perioda vybuzeného signálu v ustáleném stavu je skutečně 2π a při impedanci jdoucí k nekonečnu je $u_2 = u_1$.



Obr. 2.2-10: Záznam řešení při harmonickém buzení modelu

2.2.3 Frekvenční charakteristiky

Vyjdeme-li z diferenciální rovnice popisující RLC člen ve tvaru

$$u_2'' + a_1 u_2' + a_0 u_2 = b_1 u_1'$$

$$a_0 = \frac{1}{LC}; \quad a_1 = \frac{1}{RC}; \quad b_1 = \frac{1}{RC}$$

lze s využitím Laplaceovy transformace samozřejmě stanovit odpovídající obrazový přenos systému – RLC členu. S využitím základních korespondencí (viz příloha A) můžeme psát

$$[s^2 U_2(s) - s u_2(0+) - u_2'(0+)] + a_1 [s U_2(s) - u_2(0+)] + a_0 U_2(s) = b_1 [s U_1(s) - u_1(0+)]$$

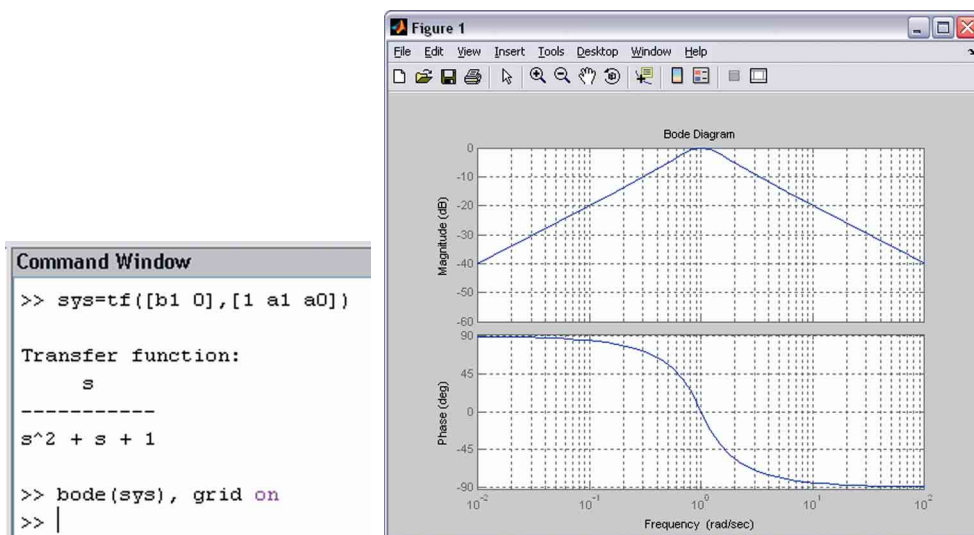
Obrazový přenos je definován jako poměr Laplaceova obrazu výstupu ku Laplaceovu obrazu vstupu při nulových počátečních podmínkách

$$G(s) = \frac{U_2(s)}{U_1(s)} = \frac{b_1 s}{s^2 + a_1 s + a_0}$$

Frekvenční přenos získáme nahrazením parametru $s \rightarrow j\omega$

$$G(j\omega) = \frac{b_1 j\omega}{(j\omega)^2 + a_1 j\omega + a_0}$$

Z frekvenčního přenosu můžeme samozřejmě vyjádřit pomocí jednoduchých úprav frekvenční charakteristiky. Zejména nás zajímají logaritmické frekvenční charakteristiky – amplitudová a fázová frekvenční charakteristika (často označovány jako tzv. BODEho charakteristiky). MATLAB umožňuje tyto charakteristiky kreslit přímo, pokud definujeme obrazový přenos zkoumaného systému. K definování obrazového přenosu použijeme příkaz **tf** (Transfer Function), k zobrazení charakteristik pak příkaz **bode**, viz obr. 2.2-11. Tyto příkazy jsou ale přístupné pouze tehdy, máme-li nainstalován Control System Toolbox.

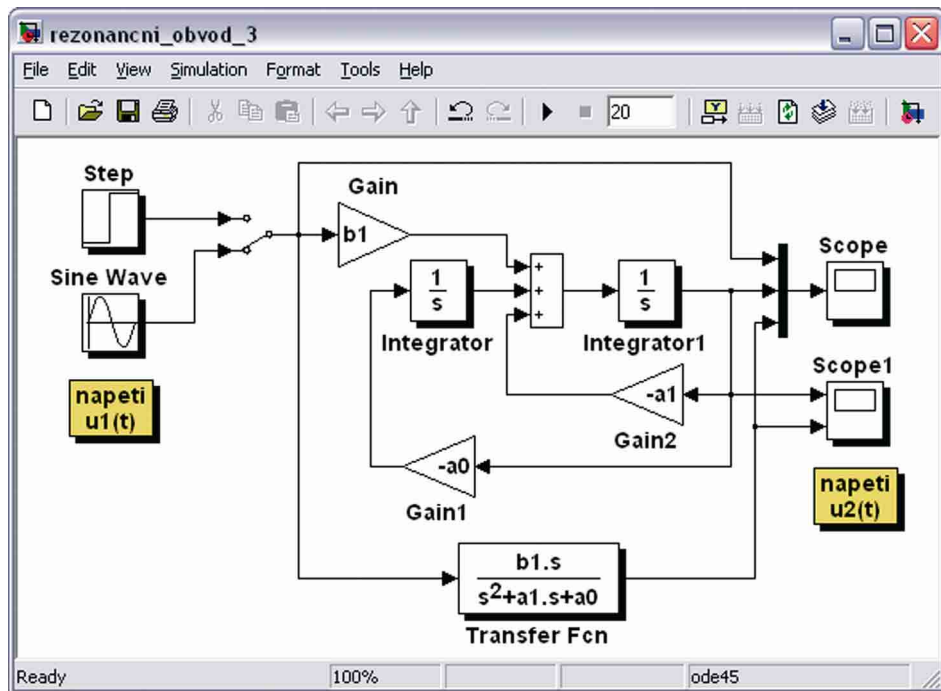


Obr. 2.2-11: Frekvenční charakteristiky

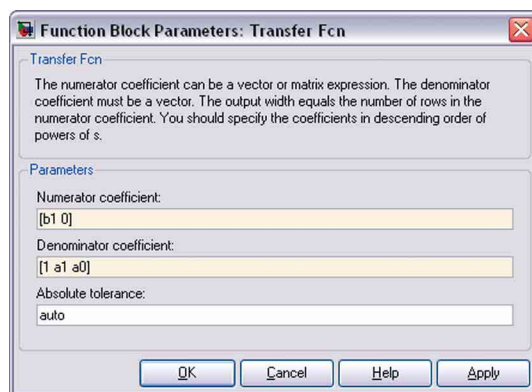
Simulink umožňuje provádět simulaci systému, který je definován pouze v jednom bloku, na základě znalosti obrazového přenosu. Nemusíme tedy složitě programovat odpovídající diferenciální rovnici, použijeme pouze blok *Transfer Fcn*. Simulační schéma s příkladem je součástí souboru rezonanci_obvod_3.mdl. Pro srovnání jsou uvedeny obě varianty řešení (obr. 2.2-12). Praktická shoda je prokázána překrytím obou průběhů zakreslených v jednom grafu (obr. 2.2-14).

Zadání koeficientů obrazového přenosu odpovídá zápisu do příkazového řádku v okně MATLABu, viz obr. 2.2-11 a 2.2-13. Velmi důležité je dodržení správného zápisu vektorů

čitatele a jmenovatele obrazového přenosu – vždy sestupně od koeficientu u nejvyšší mocniny operátoru s .

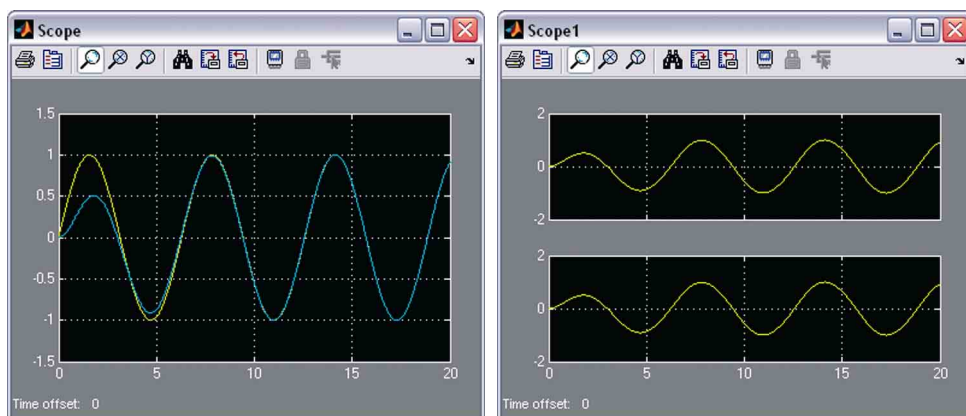


Obr. 2.2-12: Simulace na základě znalosti obrazového přenosu



Obr. 2.2-13: Definice obrazového přenosu v bloku *Transfer Fcn*

Zobrazení průběhů do paralelních oken grafu zajistíme nastavením parametru **Number of axes** v záložce **General** okna parametrů zobrazovače. Zvolíme-li jemný krok simulace, je z důvodu většího rozsahu dat vhodné v záložce **Data history** zrušit přednastavenou limitaci počtu zobrazovaných vzorků na posledních 5000.



Obr. 2.2-14: Porovnání obou variant řešení

Probraná témata

- základní integrální a diferenciální rovnice (pro napětí a proud) pro jednotlivé prvky, odvození diferenciální rovnice,
- řešení lineární diferenciální rovnice s derivací na pravé straně v prostředí MATLAB – Simulink, metoda postupné integrace,
- změna parametrů simulace a budících funkcí, možnosti zobrazení výsledků,
- frekvenční charakteristiky.

Stručné shrnutí

- při sestavování diferenciální rovnice vycházíme z integrálních a diferenciálních vztahů pro napětí a proud,
- vzhledem k derivaci na pravé straně rovnice používáme k sestavení simulačního modelu obvykle metodu postupné integrace,
- výsledný simulační model může obsahovat i několik různých bloků vstupních funkcí (jejichž parametry lze samozřejmě měnit), přepínání mezi nimi lze realizovat pomocí přepínačů (manuálních nebo řízených),
- Simulink umožňuje měnit jednoduchým způsobem krok simulace (přesnost výpočtu), všechny parametry lze definovat prostřednictvím proměnných,
- pokud máme nainstalován i Control system toolbox, MATLAB umožňuje velmi jednoduše vykreslovat frekvenční charakteristiky,
- analyzovaný dynamický systém může být popsán obrazovým přenosem, při simulaci pak postačuje systém modelovat pouze pomocí jednoho bloku.

Přiložené soubory

- soubory modelu: rezonancni_obvod_1.mdl, rezonancni_obvod_2.mdl a rezonancni_obvod_3.mdl

3 Dynamika sedačky řidiče

S neustálým rozvojem automobilové dopravy se logicky zvyšují také nároky kladené na pohodlí a bezpečnost jízdy osádky vozidel. Kmity a otřesy vyvolané jízdou vozidel po nerovných vozovkách nebo polním a lesním terénem zvyšují únavu řidičů. Při dlouhodobém a intenzivním působení mohou vážně poškozovat lidské zdraví a bývají častou příčinou nemocí z povolání. Sedadlo řidiče musí splňovat náročné požadavky, které jsou kladeny na zdraví, pohodlí, komfort a bezpečí sedící osoby. Během jeho vývoje byl kladen důraz na ergonomické, užité a estetické vlastnosti a na schopnost tlumit otřesy vznikající při jízdě. Sedadlo je vybaveno systémem posuvu a zdvihu, zvýšený boční okraj opěry zaručuje uvolněné a pohodlné sezení.

Základní konstrukce vodicích mechanismů odpružených sedadel jsou dvojího typu. Jedním typem je nůžkový mechanismus, druhým typem je čtyřkloubový mechanismus – paralelogram. Oba tyto typy mechanismů sedadel jsou obvykle doplněny pneumatickými pružinami a hydraulickými tlumiči. Vyráběná sedadla umožňují nastavovat jejich výšku, tuhost tlumičů a nebo je umožněno úplné vypuštění vzduchu ze vzduchové pružiny. Sedadla jsou vybavena také pružnými omezovači zdvihu, obvykle v podobě pryžových dorazů či zádržných pásů, které zabráňují poškození vibroizolačního systému.



Obr. 3-1: Sedačka řidiče

Chování nejruznějších dynamických systémů resp. jejich zjednodušených modelů (např. zjednodušený model sedačky řidiče či model pružného závěsu kola automobilu), můžeme poměrně jednoduše simulovat v prostředí MATLAB – Simulink. Abychom mohli sestavit odpovídající model nějakého dynamického systému v Simulinku, je nutné se nejdříve zabývat diferenciálním popisem tohoto systému. Pomocí matematicko-fyzikální analýzy tedy obvykle hledáme diferenciální rovnici, která by vhodně popisovala chování reálného fyzikálního systému. V dalším textu se omezíme na konkrétní lineární časově invariantní systém (jeho vlastnosti se v čase nemění) s jednou vstupní a jednou výstupní veličinou. Ten bývá obvykle popsán lineární diferenciální rovnicí s konstantními koeficienty, obecně ve tvaru

$$a_n y^{(n)}(t) + a_{n-1} y^{(n-1)}(t) + \dots + a_1 y'(t) + a_0 y(t) = b_m u^{(m)}(t) + \dots + b_1 u'(t) + b_0 u(t)$$

kde a_i , b_i jsou konstantní koeficienty a $u(t)$ představuje vstupní a $y(t)$ výstupní veličinu.

Z podmínky fyzikální realizovatelnosti systému musí být $m \leq n$, tj. stupeň nejvyšší derivace výstupní veličiny musí být větší nebo roven stupni nejvyšší derivace vstupní veličiny. Pro řešení uvedené diferenciální rovnice musíme znát počáteční podmínky systému $y(0), y'(0), \dots, y^{(n-1)}(0)$ a průběh vstupní veličiny $u(t)$ včetně jejich počátečních podmínek $u(0), u'(0), \dots, u^{(m-1)}(0)$.

Přenos dynamického systému je definován jako poměr Laplaceova obrazu výstupní veličiny k Laplaceově obrazu vstupní veličiny při nulových počátečních podmínkách systému $y(0) = y'(0) = \dots = y^{(n-1)}(0) = 0$ a vstupního signálu $u(0) = u'(0) = \dots = u^{(m-1)}(0) = 0$. Lineární diferenciální rovnici s konstantními koeficienty můžeme transformovat použitím pravidel Laplaceovy transformace (viz příloha A) při splnění výše uvedených podmínek do tvaru

$$[a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0] Y(s) = [b_m s^m + \dots + b_1 s + b_0] U(s)$$

Vyjdeme-li z této rovnice, tak podle výše uvedené definice má obrazový přenos systému tvar

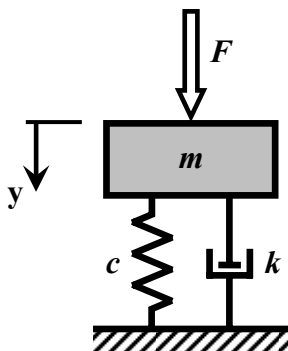
$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + \dots + b_1 s + b_0}{a_n s^n + \dots + a_1 s + a_0}$$

K případnému dalšímu studiu může čtenář využít např. knihu [5].

3.1 Text pro studenta

Příklad

Analyzujte a v prostředí Simulink simulujte chování zjednodušeného modelu sedačky řidiče na obr. 3.1-1. Uvažujte konkrétní hodnoty: hmotnost sedačky $m_1 = 20$ kg, hmotnost řidiče $m_2 = 80$ kg, gravitační zrychlení $g \approx 10 \text{ m/s}^2$, koeficient tuhosti pružiny $c = 8 \cdot 10^3 \text{ N/m}$ a koeficient tlumení tlumiče $k = 2,4 \cdot 10^3 \text{ Ns/m}$. Sledujte také vliv změny parametrů c a k na pohyb sedačky při dosednutí řidiče.



Obr. 3.1-1: Zjednodušený model sedačky

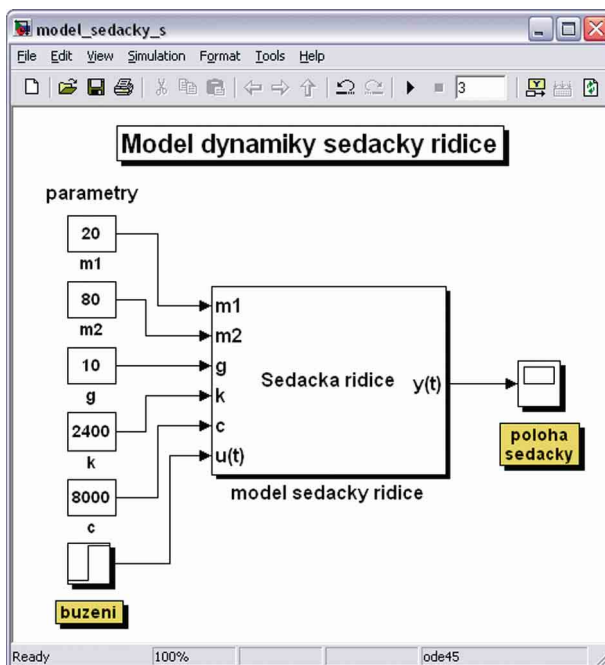


Řešení

Při zkoumání dynamického chování pracovní sedačky řidiče budeme vycházet z jejího zjednodušeného modelu. Jedná se v podstatě o pohyb hmoty (zahrnující hmotnost řidiče, sedačky a jejího mechanismu) na tlumeném pružném závěsu. Poloha sedačky je vztažena k ustálenému stavu, odpovídajícímu deformaci pružiny vyvolané samotnou její hmotností bez vnějšího zatížení. Hmotu řidiče lze považovat za vnější sílu F působící na sedačku. Uvažujeme-li konkrétní zadané hodnoty

$$F = m_2 g = 80 \cdot 10 = 800 \text{ N}$$

Simulační model (obr. 3.1-2) byl vytvořen v prostředí MATLAB – Simulink. Model je uložen v příloženém souboru model_sedacky_s.mdl.



Obr. 3.1-2: Model dynamiky sedačky řidiče v Simulinku

Model dynamiky sedačky řidiče je již ve stavu, kdy je možné přejít k vlastní simulaci. Všechny potřebné parametry jsou zadány, můžeme spustit simulaci (viz také kapitola 2). Po ukončení výpočtu lze dvojitým kliknutím na ikoně zobrazovače otevřít okno s grafickým záznamem řešení.



Obr. 3.1-3: Zobrazení řešení

Poznamenejme, že kladnou výchylku y simulované reakce je nutné chápat tak, jak je naznačeno na obr. 3.1-3, jako odchylku od ustáleného stavu bez zatížení, tedy prosednutí směrem dolů.

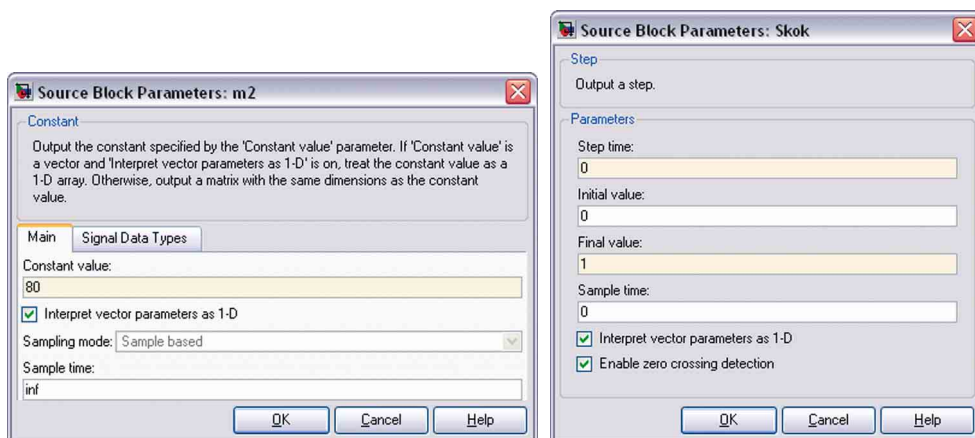
Pohyb sedačky je v tomto případě plynulý, tj. bez překmitu a kmitavých složek. K praktickému ustálení pohybu dochází asi za 1,5 sekundy. Ustálená hodnota polohy sedačky po odeznění přechodového děje je $y(\infty) = 0,1 \text{ m} = 10 \text{ cm}$. Tuto hodnotu jsme mohli (měli) předpokládat. V ustáleném stavu se totiž neprojeví vliv tlumiče a poloha sedačky je dána pouze tuhostí pružiny c . V rovnovážném stavu jsou síla reakce pružiny a síla vyvolaná hmotností řidiče ve vzájemné rovnováze

$$y(\infty) c = F = m_2 g$$

$$y(\infty) = (m_2 g) / c = (80 \cdot 10) / 8000 = 0,1 \text{ m}$$

Koeficient tlumení tlumiče k ovlivní pouze průběh a dynamiku přechodového děje.

V modelu je samozřejmě možné měnit vstupní parametry (hodnoty m_1 , m_2 , g , k , c). Na obr. 3.1-4 je uvedeno dialogové okno pro zadávání hmoty řidiče m_2 . V záložce **Main** tohoto okna zapíšeme v položce **Constant value** požadovanou hodnotu. Jednoduše lze změnit i parametry budicí funkce. V položce **Step time** lze měnit dobu skoku a v položkách **Initial value** a **Final value** počáteční a konečnou hladinu skoku.



Obr. 3.1-4: Změna vstupních parametrů a parametrů skokové budicí funkce

Kontrolní otázky a úkoly

- Simulujte pohyb sedačky za stejné situace, ovšem se správnou (nezaokrouhlenou) hodnotou gravitačního zrychlení $g = 9,800665 \text{ m/s}^2$. Jakou změnu jste předpokládali? Potvrdila se simulací?
- Jak by se chovala sedačka řidiče za stejné situace na Měsíci? Gravitační zrychlení na povrchu Měsíce je přibližně šestkrát menší, uvažujte $g = 1,624 \text{ m/s}^2$.

- Simulujte chování sedačky pro různé hodnoty parametrů. Co jednotlivé parametry ovlivňují? Jak by se chovala sedačka s nulovým tlumením? Konfrontujte vždy váš fyzikální náhled s výsledky simulace.
- Experimentálně určete při jakém tlumení tlumiče k se začíná projevovat kmitavý pohyb sedačky.
- Podle pokynů učitele vypracujte krátkou technickou zprávu.

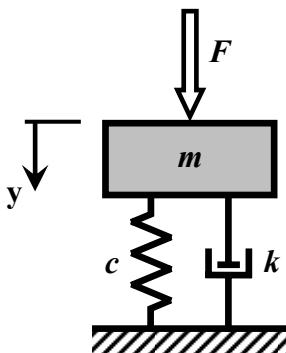
Přiložené soubory

- soubor modelu: model_sedacky_s.mdl

3.2 Text pro pedagoga

Příklad

Analyzujte a v prostředí Simulink simulujte chování zjednodušeného modelu sedačky řidiče na obr. 3.2-1. Uvažujte konkrétní hodnoty: hmotnost sedačky $m_1 = 20$ kg, hmotnost řidiče $m_2 = 80$ kg, gravitační zrychlení $g \approx 10$ m/s², koeficient tuhosti pružiny $c = 8 \cdot 10^3$ N/m a koeficient tlumení tlumiče $k = 2,4 \cdot 10^3$ Ns/m. Sledujte také vliv změny parametrů c a k na pohyb sedačky při dosednutí řidiče.



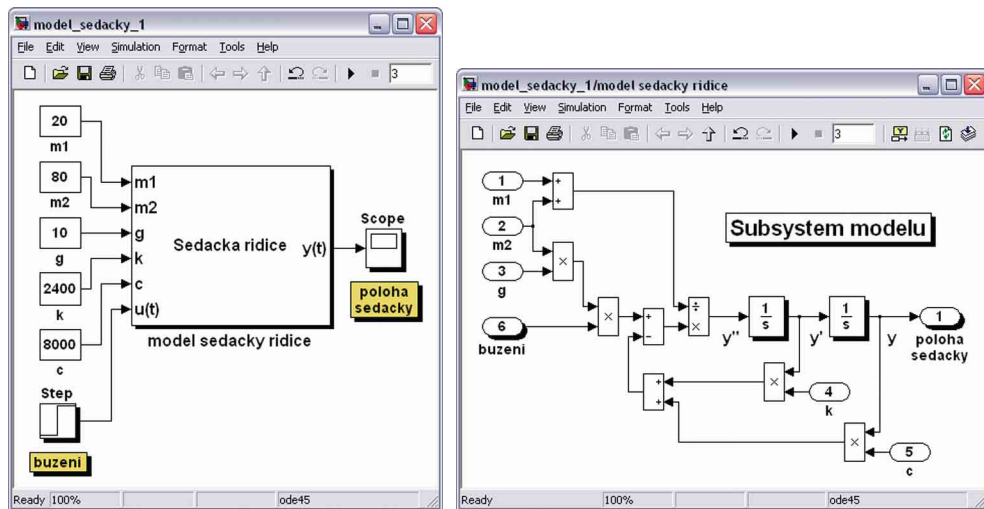
Obr. 3.2-1: Zjednodušený model sedačky

Řešení

Při zkoumání dynamického chování pracovní sedačky řidiče budeme vycházet z jejího zjednodušeného modelu. Jedná se v podstatě o pohyb hmoty (zahrnující hmotnost řidiče, sedačky a jejího mechanismu) na tlumeném pružném závěsu.

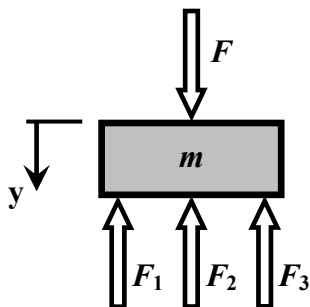
V příloze tohoto textu naleznete soubor model_sedacky_1.mdl. Otevřete tento soubor v prostředí Simulinku a dvakrát klikněte na ikonu symbolizující model sedačky řidiče, otevře se vám skrytá struktura subsystému vlastního simulačního modelu, viz obr. 3.2-2. Tato struktura může (ale také nemusí) být uživateli, kterého nezajímají podrobnosti modelu,

skrytá. Realizuje v grafickém prostředí Simulinku jednu z možností simulačního modelu reálného systému, jehož pohyb je popsán diferenciální rovnicí.



Obr. 3.2-2: Model dynamiky sedačky řidiče v Simulinku

Zabývejme se nejprve **diferenciálním popisem** dynamického systému, tedy diferenciální rovnicí (představuje matematický model), která popisuje chování reálného fyzikálního systému. Získáme ji jednoduchou aplikací d' Alembertova principu vyrovnání sil působících na hmotu m , viz obr. 3.2-3. V pracovním rozsahu předpokládáme lineární chování tlumiče i pružiny.



Obr. 3.2-3: Rovnováha sil

V tomto případě se jedná o rovnováhu čtyř sil

$$F = m_2 g \quad \dots \text{ síla vyvolaná hmotností řidiče}$$

$$F_1 = c y \quad \dots \text{ reakce pružiny}$$

$$F_2 = k y' = k \frac{dy}{dt} \quad \dots \text{ reakce tlumiče}$$

$$F_3 = m y'' = m \frac{d^2 y}{dt^2} \quad \dots \text{ setrvačný odpor}$$

Uvažujeme-li jejich společné působíště v těžišti tělesa

$$F_3 + F_2 + F_1 = F$$

$$m y'' + k y' + c y = F$$

Pro konkrétní hodnoty uvedené v zadání je

$$F = m_2 g = 80 \cdot 10 = 800 \text{ N}$$

$$m = m_1 + m_2 = 20 + 80 = 100 \text{ kg}$$

a diferenciální popis nabývá tvaru

$$100 y'' + 2400 y' + 8000 y = F$$

Poloha sedačky je vztažena k ustálenému stavu, odpovídajícímu deformaci pružiny vyvolané samotnou její hmotností bez vnějšího zatížení. Hmotu řidiče lze považovat za vnější sílu F působící na sedačku.

Statická úvaha

Zabývejme se nyní souvislostmi mezi budící silou F a reakcí y v ustáleném stavu, tedy po odeznění přechodového děje. Z fyzikální podstaty je zřejmé, že se jedná o **stabilní** systém v tom smyslu, že po dosednutí řidiče na sedačku dojde k přechodovému ději, který se za konečný čas prakticky ustálí.

Po dostatečně dlouhé době (čas $t \rightarrow \infty$) proto platí, že $F \rightarrow F(\infty) = 800$, $y'' \rightarrow y''(\infty) = 0$, $y' \rightarrow y'(\infty) = 0$ a $y \rightarrow y(\infty) \neq 0$. Diferenciální popis tak přechází do tvaru

$$100 y''(\infty) + 2400 y'(\infty) + 8000 y(\infty) = F(\infty)$$

a s ohledem na výše uvedené

$$8000 y(\infty) = 800$$

$$y(\infty) = 0,1 \text{ m} = 10 \text{ cm}$$

Ustálená hodnota polohy sedačky po odeznění přechodového děje je $y(\infty) = 10 \text{ cm}$. Srovnajte s výsledkem simulace v kapitole 3.2.2 (obr. 3.2-6)!

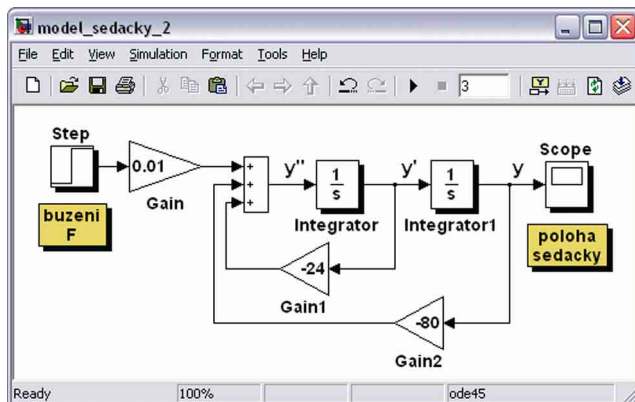
3.2.1 Vytvoření simulačního modelu

Při vytváření simulačního modelu (tvorbě simulačního schéma v Simulinku) vycházíme z dynamického popisu systému. Nejprve z výše uvedené diferenciální rovnice vyjádříme nejvyšší derivaci polohy sedačky y'' jako lineární kombinaci nižších derivací a buzení, tj. síly F

$$y'' = -24 y' - 80 y + 0,01 F$$

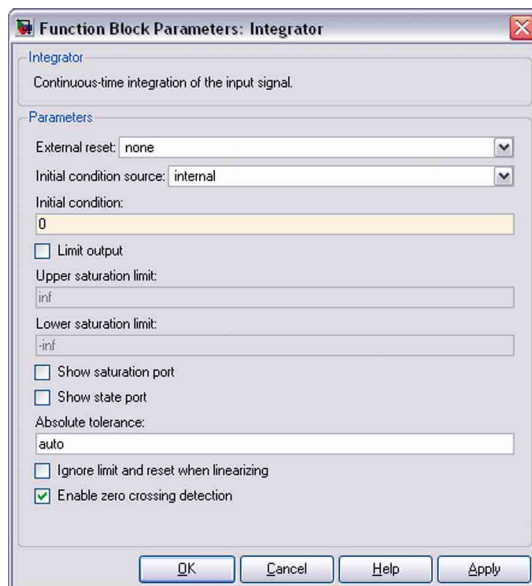
Realizaci této závislosti nám snadno umožní sériové zapojení dvou integrátorů na obr. 3.2-4. Technice vytváření simulačních modelů se podrobně věnuje kniha [4]. Kliknutím

na symbol (ikonu) některého z bloků v programovém okně Simulinku se otevře příslušné dialogové okno, ve kterém nastavíme požadované parametry.



Obr. 3.2-4: Simulační schéma

V modelu je samozřejmě možné měnit hodnoty zesílení v jednotlivých zesilovačích (bloky *Gain*, *Gain1* a *Gain2*). V záložce **Main** dialogového okna lze v položce **Gain** požadovanou hodnotu. Jednoduše lze změnit i parametry budicí funkce (blok *Step*). V položce **Step time** lze měnit dobu skoku a v položkách **Initial value** a **Final value** počáteční a konečnou hladinu skoku (obr. 3.1-4). Podrobnější popis techniky nastavení parametrů jednotlivých bloků a nastavení parametrů simulace je uveden v kapitole 2.2.1.



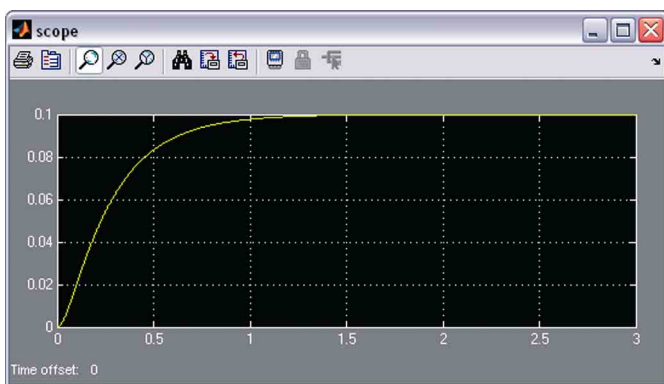
Obr. 3.2-5: Možnosti nastavení počátečních podmínek a dalších parametrů integrátoru

Blok *Integrator* umožňuje i zadání počátečních podmínek a limitaci výstupu (tzv. saturaci), viz obr. 3.2-5. Obojí je možno zadat jednak přímo hodnotou parametru, ale také pomocí dalších vstupních signálů. Blok dovoluje i externí reset (nastavení do počátečního stavu) na základě změny na řídicím logickém signálu (na náběžnou, sestupnou nebo obě hrany). Další možností je definování stavového výstupu (state portu), který se využívá právě ve spojitosti s externím resetem integrátoru a poskytuje hodnotu, která by byla na výstupu, pokud by nedošlo k resetu.

3.2.2 Spuštění simulace a zobrazení výsledků

Simulaci spustíme prostřednictvím položky **Start** v nabídce **Simulation** nebo kliknutím na ikonu ► v toolbaru okna. Stejného efektu dosáhneme i kombinací kláves Ctrl + T. Pokud dojde v průběhu simulace k problémům nebo je-li zvolen příliš dlouhý čas (resp. jemný krok), je možné simulaci předčasně ukončit pomocí ikony ■.

Po ukončení výpočtu a kliknutím na ikonku zobrazovače ve schématu (v našem případě blok *Scope*) se otevře okno s grafickým záznamem simulovaného průběhu pohybu sedačky (tj. grafické řešení diferenciální rovnice), viz obr. 3.2-6. V liště okna je k dispozici sada ikon pro případný zoom a editaci průběhu.



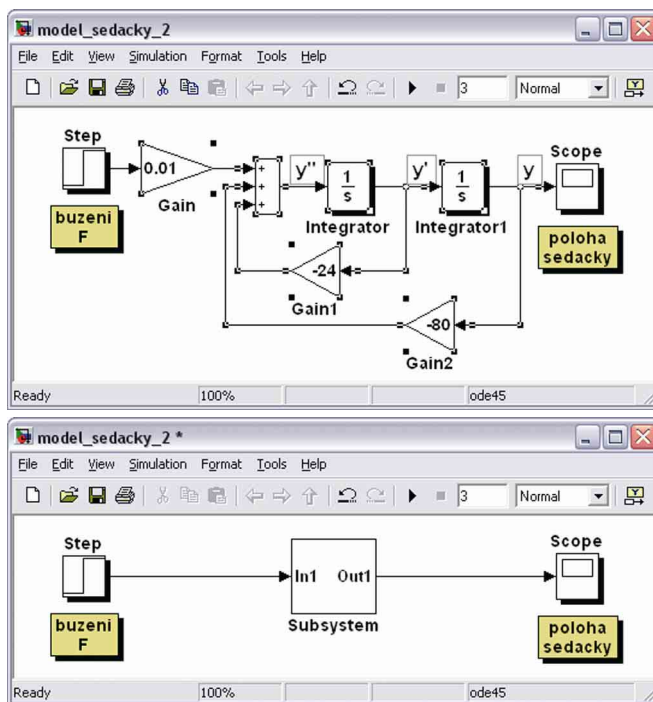
Obr. 3.2-6: Záznam řešení

Připomeňme, že kladná hodnota reakce modelu odpovídá prosednutí sedačky, tedy pohybu směrem dolů. Povšimněme si také ustálené hodnoty řešení $y(\infty) = 0,1$, která potvrzuje správnost úvahy o statickém chování systému (viz statická úvaha výše).

Pokud by nás zajímala reakce systému vybuzeného jinou než skokovou budicí funkcí, jednoduše nahradíme blok skokové změny (blok *Step*) jiným blokem z bohaté nabídky Simulinku.

Model sedačky řidiče uvedený v úvodu této kapitoly (obr. 3.2-2) a v textu určeném pro studenta (obr. 3.1-2, kapitola 3.1) byl skryt do tzv. subsystému, který je reprezentován jediným blokem. Zahrnutí části simulačního schéma do subsystému (tj. do jednoho bloku) dosáhneme tak, že myši označíme požadovanou část schématu a následně z menu **Edit**

vybereme položku **Create Subsystem**. Tímto způsobem můžeme zjednodušit a zpřehlednit i velmi složité a komplikované struktury simulačních modelů.



Obr. 3.2-7: Zahnutí části modelu do subsystému

3.2.3 Matematická analýza pohybu sedačky

Matematickým modelem sedačky řidiče je diferenciální rovnice, jejímž řešením je časová funkce, která analyticky popisuje její pohyb. Velmi účinným nástrojem pro řešení lineárních diferenciálních rovnic je Laplaceova transformace. V předchozích odstavcích byl nalezen diferenciální popis ve tvaru

$$m y'' + k y' + c y = F$$

a pro zadané hodnoty pak

$$100 y'' + 2400 y' + 8000 y = F$$

Poloha sedačky y je vztažena k ustálenému stavu, odpovídajícímu deformaci pružiny vyvolané samotnou její hmotností bez vnějšího zatížení. Hmotu řidiče lze považovat za vnější sílu F působící na sedačku.

Předpokládejme, že v ustáleném klidovém stavu sedačky (nulové počáteční podmínky, tj. $y(0) = 0$, $y'(0) = 0$) dosedne na sedačku náhle řidič a vyvolá tak její pohyb. Řešení diferenciální rovnice za těchto podmínek a s tímto buzením popisuje vyvolaný pohyb.

Nejprve připomeňme statickou úvahu provedenou již v úvodu této kapitoly. Jedná se o určení $y(\infty) = \lim_{t \rightarrow \infty} y(t)$, tj. celkového prosednutí sedačky v ustáleném stavu, po odeznění přechodového děje. Tuto hodnotu můžeme určit velmi snadno i jednoduchou úvahou bez použití složité matematiky (matematický rozbor však musí tento závěr samozřejmě potvrdit).

Z fyzikální podstaty je zřejmé, že se jedná o stabilní systém, a to v tom smyslu, že po dosednutí řidiče na sedačku dojde k přechodovému ději, který se za konečný čas prakticky ustálí. Po dostatečně dlouhé době ($t \rightarrow \infty$) proto platí, že $F \rightarrow F(\infty) = 800$, $y'' \rightarrow y''(\infty) = 0$, $y' \rightarrow y'(\infty) = 0$ a $y \rightarrow y(\infty) \neq 0$. Diferenciální popis tak přechází do tvaru

$$100 y''(\infty) + 2400 y'(\infty) + 8000 y(\infty) = F(\infty)$$

a s ohledem na výše uvedené

$$y(\infty) = 0,1 \text{ m} = 10 \text{ cm}$$

Laplaceova transformace

Zabývejme se dále pohybem sedačky po dosednutí řidiče. Její pohyb je v čase popsán řešením uvedené diferenciální rovnice s předpokládanými počátečními podmínkami a budící funkcí. Řešme ji pomocí Laplaceovy transformace. Je to velmi jednoduchý a efektivní aparát, který nám umožní převést lineární diferenciální rovnici na rovnici algebraickou, v tomto tvaru pak nalézt její řešení (resp. jeho obraz) a opět ho převést zpět do časové oblasti. Byť tento postup vypadá na první pohled složitě, řešení diferenciální rovnice tímto způsobem je velmi jednoduché. K formálnímu řešení vystačíme s několika málo korespondencemi a vlastnostmi transformace (příloha A)

$$y(t) \doteq Y(s)$$

$$y'(t) \doteq sY(s) - y(0+)$$

$$y''(t) \doteq s^2 Y(s) - sy(0+) - y'(0+)$$

$$F(t) \doteq 800/s = F(s)$$

Obraz diferenciální rovnice nalezneme snadno

$$100 [s^2 Y(s) - sy(0+) - y'(0+)] + 2400 [sY(s) - y(0+)] + 8000 Y(s) = F(s)$$

Obrazem diferenciální rovnice je rovnice algebraická s jedinou neznámou $Y(s)$, obrazem řešení je

$$Y(s) = \frac{1}{100s^2 + 2400s + 8000} F(s) + \frac{(100s + 2400)y(0+) + 100y'(0+)}{100s^2 + 2400s + 8000}$$

Vyjádřili jsme tak relativně snadno obraz reakce dynamického systému obecně pro jakékoliv buzení a počáteční podmínky. V našem konkrétním případě však

$$y(0+) = y'(0+) = 0$$

$$F(s) = \frac{800}{s}$$

$$Y(s) = \frac{1}{100s^2 + 2400s + 8000} \cdot \frac{800}{s} = \frac{8}{(s^2 + 24s + 80)s} = \frac{8}{(s+4)(s+20)s}$$

Limitní korespondence

Hodnotu $y(\infty)$ reakce sedačky po odeznění přechodového děje můžeme ověřit již v této fázi řešení z nalezeného obrazu $Y(s)$ využitím limitních korespondencí mezi obrazem a předmětem (viz příloha A).

$$y(\infty) = \lim_{s \rightarrow 0} s Y(s) = \lim_{s \rightarrow 0} \frac{8}{(s+4)(s+20)} = 0,1$$

Použitím druhé limitní korespondence určíme naopak začátek pochodu, který musí být v souladu se zadanými počátečními podmínkami.

$$y(0+) = \lim_{s \rightarrow \infty} s Y(s) = \lim_{s \rightarrow \infty} \frac{8}{(s+4)(s+20)} = 0$$

$$y'(t) \doteq s Y(s) - y(0+) = \frac{8}{(s+4)(s+20)}$$

$$y'(0+) = \lim_{s \rightarrow \infty} \frac{8s}{(s+4)(s+20)} = 0$$

Ověření shody těchto limitních hodnot v této fázi řešení je mj. i vhodnou kontrolou správnosti nalezeného obrazu. Poznamenejme však, že shoda hodnot ještě nedokazuje správnost obrazu, ale naopak rozpor by prokázal chybu.

Zpětná Laplaceova transformace

Řešení popisující pohyb sedačky v reálném čase získáme zpětnou Laplaceovou transformací. Nejprve obraz rozložíme na součet parciálních zlomků, které pak postupně podle slovníku korespondencí uvedených v příloze A převedeme do hledaného předmětu.

$$Y(s) = \frac{8}{s(s+4)(s+20)} = \frac{A}{s} + \frac{B}{(s+4)} + \frac{C}{(s+20)}$$

$$A = \lim_{s \rightarrow 0} Y(s)s = \lim_{s \rightarrow 0} \frac{8}{(s+4)(s+20)} = 0,1$$

$$B = \lim_{s \rightarrow -4} Y(s)(s+4) = \lim_{s \rightarrow -4} \frac{8}{s(s+20)} = -\frac{1}{8} = -0,125$$

$$C = \lim_{s \rightarrow -20} Y(s)(s+20) = \lim_{s \rightarrow -20} \frac{8}{s(s+4)} = \frac{1}{40} = 0,025$$

Koeficienty parciálních zlomků lze určit i jiným způsobem. Sečteme-li formálně uvažované parciální zlomky a srovnáme-li koeficienty u příslušných mocnin vzniklého polynomu v čitateli s polynomem čitatele rozkládaného obrazu, dostaneme soustavu algebraických rovnic, jejímž řešením jsou hledané koeficienty (jde o tzv. metodu neurčitých koeficientů).

$$Y(s) = \frac{A}{s} + \frac{B}{(s+4)} + \frac{C}{(s+20)} = \frac{A(s+4)(s+20) + Bs(s+20) + Cs(s+4)}{s(s+4)(s+20)} =$$

$$= \frac{s^2(A+B+C) + s(24A+20B+4C) + 80A}{s(s+4)(s+20)} = \frac{8}{s(s+4)(s+20)}$$

$$\left. \begin{array}{l} A+B+C=0 \\ 24A+20B+4C=0 \\ 80A=8 \end{array} \right\} \begin{array}{l} A=0,1 \\ B=-0,125 \\ C=0,025 \end{array}$$

Metoda neurčitých koeficientů se zdá být jednodušší (snad je i častěji používaná), výpočet je však v obecném případě pracnější. Uvedený přístup s limitními manipulacemi je vysoce efektivní zejména v případě reálných jednonásobných kořenů jmenovatele rozkládané funkce.

Povšimněme si, že koeficienty A , B , C parciálních zlomků jsou rezidua obrazu $Y(s)$ v jeho singulárních bodech (pólech) – a také se tak jako rezidua i počítají. Vzhledem ke stupňům polynomů čitatele (nula) a jmenovatele (tři) obrazu musí být jejich součet roven nule. Součet reziduí je roven podílu koeficientů $(n-1)$ -ní mocniny operátoru s čitatele a n -té mocniny s jmenovatele, kde n je nevyšší mocnina s ve jmenovateli – obraz musí být racionální lomená funkce ryzi.

Tato kontrola je vhodným ověřením správnosti rozkladu. V případě vícenásobných pólů nejsou obecně koeficienty všech parciálních zlomků rezidua rozkládaného obrazu. V tomto případě je nutné i uvedený výpočet koeficientů parciálních zlomků poněkud modifikovat.

Pokud máme obraz $Y(s)$ rozložený na součet parciálních zlomků, jsme již blízko od hledaného řešení diferenciální rovnice. Z integrální podstaty Laplaceovy transformace totiž vyplývá, že obraz součtu dílčích funkcí se rovná součtu jejich obrazů (a obráceně). Můžeme proto velmi jednoduše nalézt pomocí slovníku L-transformace předměty odpovídající jednotlivým parciálním zlomkům a vyjádřit hledané řešení jejich součtem.

$$Y(s) = \frac{0,1}{s} + \frac{-0,125}{(s+4)} + \frac{0,025}{(s+20)}$$

$$y(t) = (0,1 - 0,125e^{-4t} + 0,025e^{-20t})\eta(t)$$

Symbol $\eta(t)$ nabývá hodnoty 0 pro čas $t < 0$ a hodnoty 1 pro $t \geq 0$. Tzv. Heavisideův jednotkový skok formálně ošetřuje platnost uvedeného vztahu jen pro $t \geq 0$.

Rychlost pohybu sedačky určíme jako derivaci $y(t)$

$$y'(t) = (0,5e^{-4t} + 0,5e^{-20t})\eta(t)$$

Ověřme nyní správnost limitních hodnot řešení na začátku a na konci přechodového děje

$$y(0+) = \lim_{t \rightarrow 0+} y(t) = \lim_{t \rightarrow 0+} (0,1 - 0,125 e^{-4t} + 0,025 e^{-20t}) = 0$$

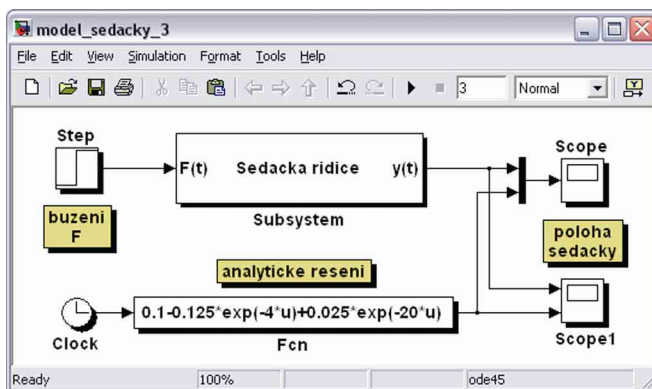
$$y'(0+) = \lim_{t \rightarrow 0+} y'(t) = \lim_{t \rightarrow 0+} (0,5 e^{-4t} + 0,5 e^{-20t}) = 0$$

$$y(\infty) = \lim_{t \rightarrow \infty} y(t) = \lim_{t \rightarrow \infty} (0,1 - 0,125 e^{-4t} + 0,025 e^{-20t}) = 0,1$$

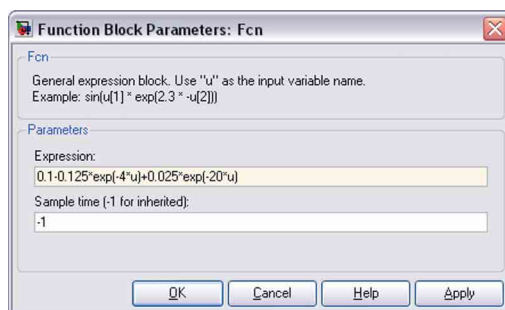
$$y'(\infty) = \lim_{t \rightarrow \infty} y'(t) = \lim_{t \rightarrow \infty} (0,5 e^{-4t} + 0,5 e^{-20t}) = 0$$

3.2.4 Ověření analytického řešení v Simulinku

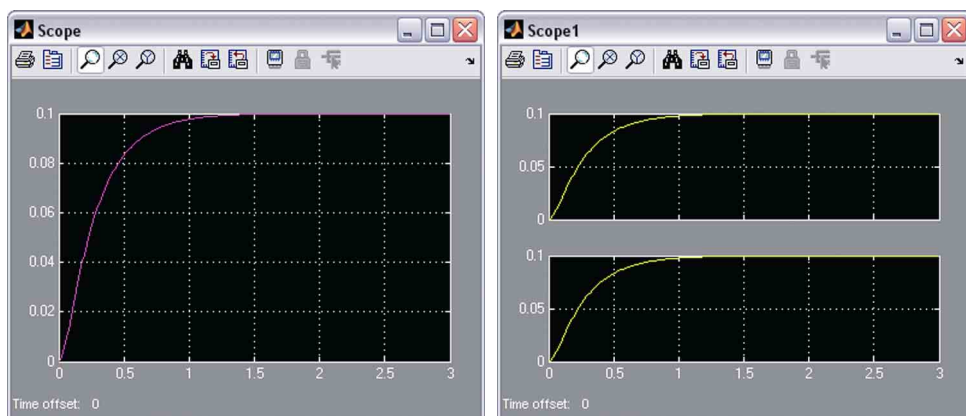
Simulační model sedačky řidiče doplníme funkčním blokem *Fcn* realizujícím analytické řešení diferenciální rovnice (obr. 3.2-8 a 3.2-9). Praktická shoda obou průběhů je prokázána na obr. 3.2-10 jejich vzájemným překrytím v jednom grafu. Další možností je zobrazení průběhů ve dvou paralelních grafech. V bloku zobrazovače je třeba nastavit v okně parametrů bloku v záložce **General** položku **Number of Axes** na hodnotu 2.



Obr. 3.2-8: Analytické řešení v Simulinku



Obr. 3.2-9: Zápis analytického řešení v bloku *Fcn*



Obr. 3.2-10: Porovnání analytického řešení se simulací

3.2.5 Dynamický systém na mezi periodicity

Hledáme minimální hodnotu tlumení k , při které se ještě neobjevují kmitavé složky průběhu. Kořeny charakteristické rovnice jsou **reálné násobné**. Uvažujeme obraz řešení diferenciální rovnice za stejných okolností s jinými hodnotami konstrukčních parametrů

$$Y(s) = \frac{1}{m s^2 + k s + c} \cdot \frac{800}{s} = \left. \begin{array}{l} \text{spec.} \\ m = 100 \\ c = 8000 \end{array} \right| = \frac{8}{(s^2 + k \cdot 10^{-2} s + 80) s}$$

charakteristická rovnice a její kořeny

$$s^2 + k \cdot 10^{-2} s + 80 = 0; \quad s_{1,2} = \frac{-k \cdot 10^{-2} \pm \sqrt{k^2 \cdot 10^{-4} - 4 \cdot 80}}{2}$$

a pro dvojnásobný kořen

$$k^2 \cdot 10^{-4} - 4 \cdot 80 = 0$$

$$k = \sqrt{4 \cdot 80 \cdot 10^{+4}} \approx 17,89 \cdot 10^2 \text{ Ns/m}$$

$$s_{1,2} = \frac{-17,89 \cdot 10^2 \cdot 10^{-2}}{2} \approx -8,94$$

Obraz řešení v tomto případě je

$$Y(s) = \frac{8}{(s^2 + k \cdot 10^{-2} s + 80) s} = \frac{8}{(s + 8,94)^2 s} = \frac{A}{s} + \frac{B}{(s + 8,94)^2} + \frac{C}{s + 8,94}$$

$$A = \lim_{s \rightarrow 0} Y(s) s = \lim_{s \rightarrow 0} \frac{8}{(s + 8,94)^2} = 0,1$$

$$B = \lim_{s \rightarrow -8,94} Y(s)(s + 8,94)^2 = \lim_{s \rightarrow -8,94} \frac{8}{s} \approx -0,89$$

$$C = \lim_{s \rightarrow -8,94} \frac{1}{1!} \cdot \frac{d}{ds} \{Y(s)(s + 8,94)^2\} = \lim_{s \rightarrow -8,94} \frac{d}{ds} \frac{8}{s} = \lim_{s \rightarrow -8,94} \frac{-8}{s^2} \approx -0,1$$

V případě vícenásobných pólů obrazu by se koeficienty parciálních zlomků s klesající mocninou kořenového činitele počítaly jako limita ze stejně rostoucí derivace vynásobené faktoriálním koeficientem $1/n!$, kde n je řád derivace. Poznamenejme, že uvedené limity jsou v regulárních bodech, takže se i zde jedná jen o pouhé dosazení hodnot.

V tomto případě jsou rezidua obrazu $Y(s)$ pouze koeficienty A a C , tedy $-B$ není reziduum. I zde samozřejmě platí, že součet reziduí je roven nule, tj. $A + C = 0$. Hledané řešení je

$$Y(s) = \frac{0,1}{s} - \frac{0,89}{(s + 8,94)^2} - \frac{0,1}{(s + 8,94)}$$

$$y(t) = (0,1 - 0,89t e^{-8,94t} - 0,1e^{-8,94t})\eta(t)$$

3.2.6 Kmitavý dynamický systém

Uvažujme kmitavý dynamický systém s vlastními harmonickými složkami pohybu. Kořeny charakteristické rovnice jsou v tomto případě **komplexní**. Předpokládejme hodnoty parametrů $m = 100$ kg, $k = 400$ Ns/m, $c = 8 \cdot 10^3$ N/m.

$$Y(s) = \frac{1}{ms^2 + ks + c} \cdot \frac{800}{s} = \left| \begin{array}{l} m = 100 \\ k = 400 \\ c = 8000 \end{array} \right| = \frac{8}{(s^2 + 4s + 80)s}$$

Charakteristická rovnice a její kořeny

$$s^2 + 4s + 80 = 0; \quad s_{1,2} = \frac{-4 \pm \sqrt{4^2 - 4 \cdot 80}}{2} \approx -2 \pm j 8,72$$

Kořeny charakteristické rovnice jsou komplexní, dynamický systém má vlastní kmitý s frekvencí 8,72 rad/s a s exponenciálním tlumením e^{-2t} (určeno imaginární a reálnou částí kořenů).

Rozklad na parciální zlomky předpokládáme z důvodu komplexních kořenů ve tvaru

$$Y(s) = \frac{8}{(s^2 + 4s + 80)s} = \frac{A}{s} + \frac{Bs + C}{s^2 + 4s + 80}$$

Koeficient A určíme nejlépe reziduálním výpočtem, koeficienty B a C metodou neurčitých koeficientů

$$A = \lim_{s \rightarrow 0} Y(s)s = \lim_{s \rightarrow 0} \frac{8}{s^2 + 4s + 80} = 0,1$$

$$Y(s) = \frac{0,1}{s} + \frac{Bs + C}{s^2 + 4s + 80} = \frac{s^2(B + 0,1) + s(C + 0,4) + 8}{(s^2 + 4s + 80)s} = \frac{8}{(s^2 + 4s + 80)s}$$

$$\left. \begin{array}{l} B + 0,1 = 0 \\ C + 0,4 = 0 \end{array} \right\} \quad B = -0,1; \quad C = -0,4$$

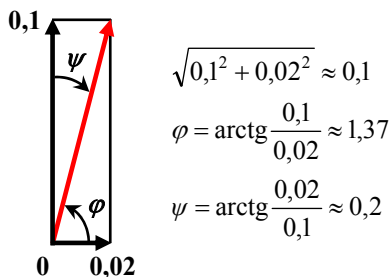
$$Y(s) = \frac{0,1}{s} - \frac{0,1s + 0,4}{s^2 + 4s + 80}$$

Následuje důležitá úprava druhé části rozkladu $Y(s)$ (parciálního zlomku odpovídajícího dvojici komplexně sdružených kořenů) do tvaru podle posledních dvou korespondencí uvedených ve slovníku v příloze A.

$$\begin{aligned} Y(s) &= \frac{0,1}{s} - \frac{0,1s + 0,4}{(s + 2)^2 + 76} = \frac{0,1}{s} - \frac{0,1(s + 2) + 0,2}{(s + 2)^2 + 76} = \\ &= \frac{0,1}{s} - \frac{0,1(s + 2)}{(s + 2)^2 + 76} - \frac{0,2}{\sqrt{76}} \cdot \frac{\sqrt{76}}{(s + 2)^2 + 76} \end{aligned}$$

Dále už jen jednoduše převedeme obraz podle slovníku L-transformace do předmětu

$$\begin{aligned} y(t) &= (0,1 - 0,1e^{-2t} \cos \sqrt{76}t - \frac{0,2}{\sqrt{76}} e^{-2t} \sin \sqrt{76}t) \eta(t) = \\ &= \{0,1 - e^{-2t} [0,1 \cos(8,72t) + 0,02 \sin(8,72t)]\} \eta(t) \end{aligned}$$



Obr. 3.2-11: Stanovení φ a ψ

Výsledek lze také v souladu s obr. 3.2-11 upravit do tvaru

$$y(t) = [0,1 - 0,1e^{-2t} \sin(8,72t + \varphi)] \eta(t) = [0,1 - 0,1e^{-2t} \cos(8,72t - \psi)] \eta(t)$$

Jiný způsob řešení

Předpokládejme stejnou situaci, stejnou diferenciální rovnici, ukažme si však jinou variantu přechodu od obrazu k řešení v časové oblasti. Předpokládáme stejné hodnoty parametrů $m = 100$ kg, $k = 400$ Ns/m, $c = 8 \cdot 10^3$ N/m.

$$Y(s) = \frac{1}{m s^2 + k s + c} \cdot \frac{800}{s} = \left. \begin{array}{l} m = 100 \\ k = 400 \\ c = 8000 \end{array} \right| = \frac{8}{(s^2 + 4s + 80)s}$$

Stejná je samozřejmě i charakteristická rovnice a její kořeny

$$s^2 + 4s + 80 = 0; \quad s_{1,2} = \frac{-4 \pm \sqrt{4^2 - 4 \cdot 80}}{2} \approx -2 \pm j 8,72$$

Rozklad na parciální zlomky však předpokládáme ve tvaru s komplexními koeficienty

$$Y(s) = \frac{8}{(s^2 + 4s + 80)s} = \frac{A}{s} + \frac{B}{(s + 2 - j 8,72)} + \frac{C}{(s + 2 + j 8,72)}$$

Všechny koeficienty rozkladu A , B i C jsou v tomto případě rezidua obrazu v jeho pólech a určíme je

$$A = \lim_{s \rightarrow 0} Y(s)s = \lim_{s \rightarrow 0} \frac{8}{s^2 + 4s + 80} = 0,1$$

$$B = \lim_{s \rightarrow -2 + j 8,72} Y(s)(s + 2 - j 8,72) = \lim_{s \rightarrow -2 + j 8,72} \frac{8}{s(s + 2 + j 8,72)} = \dots \approx -0,05 + j 0,01$$

$$C = \lim_{s \rightarrow -2 - j 8,72} Y(s)(s + 2 + j 8,72) = \lim_{s \rightarrow -2 - j 8,72} \frac{8}{s(s + 2 - j 8,72)} = \dots \approx -0,05 - j 0,01$$

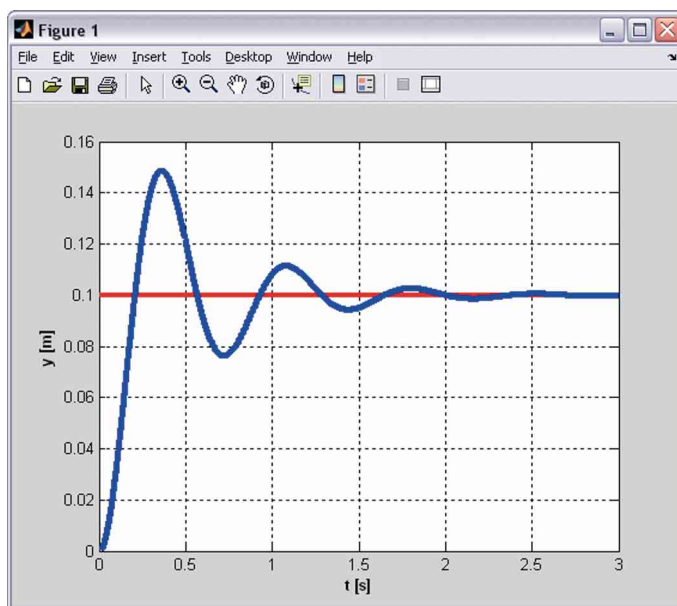
Koeficienty B a C musejí být nutně komplexně sdružené, nebylo proto ani třeba C znovu počítat. Zde jsou všechny tři koeficienty parciálních zlomků rezidua obrazu $Y(s)$ v jeho pólech, proto také platí $A + B + C = 0$.

$$Y(s) = \frac{8}{(s^2 + 4s + 80)s} = \frac{0,1}{s} + \frac{-0,05 + j 0,01}{(s + 2 - j 8,72)} + \frac{-0,05 - j 0,01}{(s + 2 + j 8,72)}$$

Pomocí slovníku (příloha A) jednoduše nalezneme předmět

$$\begin{aligned} y(t) &= [0,1 + (-0,05 + j 0,01)e^{(-2 + j 8,72)t} + (-0,05 - j 0,01)e^{(-2 - j 8,72)t}] \eta(t) = \\ &= \{0,1 + e^{-2t} [(-0,05 + j 0,01)(\cos 8,72t + j \sin 8,72t) + \\ &\quad + (-0,05 - j 0,01)(\cos 8,72t - j \sin 8,72t)]\} \eta(t) = \\ &= \dots = \{0,1 - e^{-2t} \sin[0,1 \cos(8,72t) + 0,02 \sin(8,72t)]\} \eta(t) \end{aligned}$$

Samozřejmě, že výsledek je naprosto stejný jako v předchozí variantě řešení. Obě varianty se liší jen přístupem ke zpětné Laplaceově transformaci. První přístup, bez nutnosti práce v komplexní oblasti, se zdá být poněkud jednodušší a příjemnější. Průběh polohy sedačky v závislosti na čase je uveden na obr. 3.2-12.



Obr. 3.2-12: Časová závislost pohybu sedačky řidiče

Probraná témata

- diferenciální popis sedačky, odvození její diferenciální rovnice,
- řešení lineární diferenciální rovnice v Simulinku, metoda snižování řádu derivace,
- změna parametrů simulace a budících funkcí, zobrazení výsledků,
- matematická analýza pohybu sedačky, aplikace Laplaceovy integrální transformace, limitní korespondence,
- nalezení analytického řešení diferenciální rovnice, zpětná Laplaceova transformace, rozklad na parciální zlomky a metoda neurčitých koeficientů, ověření nalezeného řešení v Simulinku,
- matematický rozbor dynamického systému na mezi periodicity (tj. násobné reálné kořeny charakteristické rovnice),
- matematický rozbor kmitavého dynamického systému s vlastními harmonickými složkami pohybu (tj. komplexní kořeny charakteristické rovnice).

Stručné shrnutí

- při sestavování diferenciální rovnice vycházíme z d'Alembertova principu rovnováhy sil zjednodušeného modelu sedačky řidiče,

- k sestavení simulačního modelu, v případě, že na pravé straně diferenciální rovnice není derivace, používáme metodu snižování řádu derivace, jinými slovy postačuje vyjádřit nejvyšší derivaci polohy sedačky,
- výsledný simulační model může být buzen nejrůznějšími typy vstupních funkcí, změna parametrů modelu je velmi jednoduchá,
- jednotlivé části simulačního schématu můžeme zahrnout z důvodu přehlednosti do subsystému, zvolená část modelu je pak reprezentována jedním blokem se vstupy a výstupy,
- efektivním prostředkem pro řešení lineárních diferenciálních rovnic s konstantními koeficienty je Laplaceova transformace, která umožňuje převést diferenciální rovnici na rovnici algebraickou, v tomto tvaru pak nalézt její řešení, resp. jeho obraz a ten převést zpět do časové oblasti,
- limitní hodnoty na počátku a po odeznění přechodového děje lze stanovit pomocí limitních korespondencí mezi obrazem a předmětem,
- řešení popisující pohyb sedačky v reálném čase získáme zpětnou Laplaceovou transformací, v popisovaném příkladu je nejprve ale nutné provést rozklad obrazu řešení na parciální zlomky, které pak postupně podle slovníku korespondencí převedeme do hledaného předmětu,
- koeficienty rozkladu lze vypočítat pomocí limit (resp. reziduálním výpočtem), další možností je tzv. metoda neurčitých koeficientů.

Přiložené soubory

- soubory modelu: model_sedacky_1.mdl, model_obvod_2.mdl
a model_sedacky_3.mdl

4 Balistická křivka projektilu

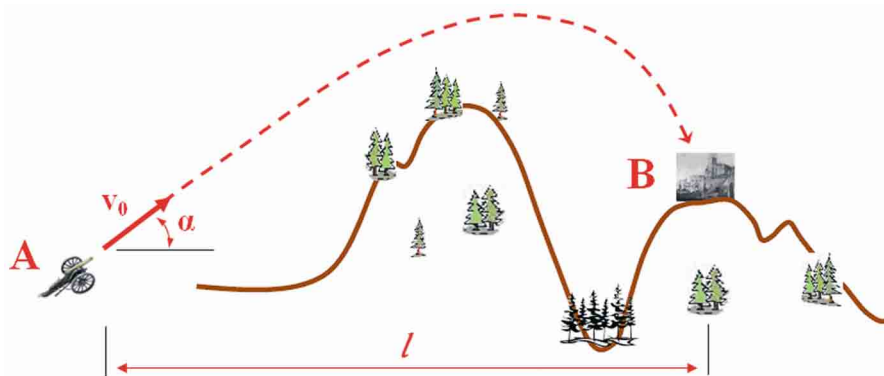
Psal se rok 1468 a Jan Boček Kuna z Kunštátu právě v bezpečí svého hradu Buchlova dobře krytého chřibskými hvozdy otevíral již třetí láhev jiskřivého vína na počest svého strýce Jiřího z Poděbrad. Náhle se ozvala obrovská rána a první dělová koule uherského krále Matyáše I. Korvína otřásla druhým nádvořím hradu. Začalo nemilosrdné obléhání, během kterého silné dělostřelectvo Buchlov ze dvou třetin zcela zničilo



Obr. 4-1: Hrad Buchlov

Balistická křivka je křivka popisující dráhu střely. Je výslednicí dynamického pohybu vystřeleného hmotného projektilu respektující odpor okolního vzduchu. Základními parametry dráhy střely jsou její hmotnost m [kg], její počáteční rychlost v okamžiku výstřelu v_0 [m/s] a úhel (náměr) α [deg], který svírá osa hlavně s vodorovnou rovinou. V dělostřelecké praxi se obvykle vyjadřuje úhel v tzv. gradech, jeden grad je jedna setina kvadrantu. Pravý úhel má tedy 100 grad, 90 deg (stupňů) a $\pi/2 \approx 1,57$ rad (míra oblouková).

Pan Boček podcenil nebezpečí a nečekal, že šelma Matyáš pošle své zvědy na nejvyšší kopec a náhle zaútočí ze zálohy bezpečně krytý okolními vrchy. Pojďme si to také zkusit, pojďme si zastřílet. Bez prachu, koulí a kanónů, bez krve, rizik a ničení.



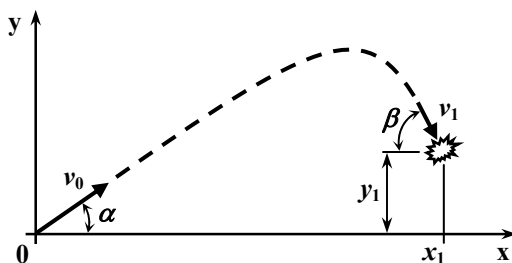
Obr. 4-2: Střelba ve členitém terénu

Je zřejmé, že to lidi na Buchlově neměli snadné. Přesvědčte se ale také o tom, že to neměli moc lehké ani Korvínovi kanonýři – i s tím špehem na kopci. Zkuste se jen trefit, když není na cíl vidět. Jen tak, „pokus omyl“. Můžete si vybrat místo (vzdálenost l místa A od místa B), množství střelného prachu (počáteční rychlost střely v_0), náměr α i hmotnost střely m . Máme k tomu dobrý prostředek – počítač a simulaci, nic nemůžete pokazit ani zničit. Nevadí, když se netrefíte. A kulí máme tolik, kolik si jen budete přát.

4.1 Text pro studenta

Příklad

Vypracujte jednoduchý návod odkud a jak střílet tak, aby se bezpečně zasáhl cíl. Formu si zvolte sami. Vypracujte tabulku nebo vhodný graf. Pokuste se o co nejpřehlednější návod, který by tehdy jistě velmi ocenili uherští žoldáci (ještě, že ho neměli, a neuměli také simulovat jako my, to by tenkrát Buchlov dopadl jistě mnohem hůře). Volte: vzdálenost od cíle $l > 500$ m vzhledem k topologii terénu, počáteční rychlost střely $100 < v_0 < 500$ m/s (tedy množství prachu), náměr $0 < \alpha < 90$ deg (stupňů), hmotnost střely $10 < m < 50$ kg a koeficient odporu vzduchu $k = 0,0134$ Ns²/m². Sami také vhodně zvolte výšku cíle (hradu) y_1 nad úrovní ústí děla. Náskres celé situace je na obr. 4.1-1.



Obr. 4.1-1: Základní parametry



Řešení

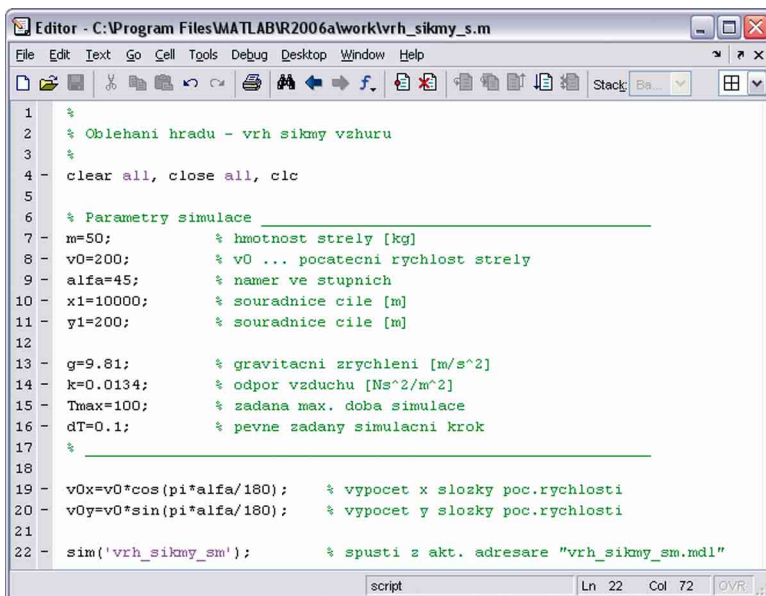
Zvolíme metodu „pokus omyl, pokus trefá“. S tolika parametry to není ale jednoduché. K simulaci střelby použijeme již předem připravený simulační model uložený v souboru `vrh_sikmy_sm.mdl` spolupracující se skriptem v souboru `vrh_sikmy_s.m`.

Oba soubory musejí být umístěny v aktuálním adresáři MATLABu. O tom se jednoduše přesvědčíme např. tak, že v pracovním okně MATLABu zadáme příkaz **dir** (a potvrdíme klávesou ENTER) a necháme vypsat obsah aktuálního adresáře. Neobjeví-li se názvy obou souborů, musíme aktuální adresář správně nastavit.

Nejprve otevřeme skript `vrh_sikmy_s.m` psaný v jednoduchém interpretačním jazyce MATLABu. Skript (podle své přípony někdy označován také jako tzv. m-soubor či m-file) slouží k ovládání sekvence příkazů prováděných MATLABem. V tomto skriptu, tak jak je napsán, viz obr. 4.1-2, lze mj. i měnit hodnoty parametrů výpočtu. Po uložení provedených změn lze výpočet jednoduše spustit v příkazovém okně zadáním názvu skriptu, ovšem bez přípony (!), tedy **vrh_sikmy_s**. Automaticky (a velmi rychle) se provede sekvence příkazů, které bychom také mohli (pracně a velmi pomalu) zadávat jeden po druhém v příkazovém okně.

Jedním z příkazů uvedeného skriptu je i příkaz pro spuštění simulace připraveného modelu v souboru `vrh_sikmy_sm.mdl` (obr. 4.1-3). Je vhodné, zejména kvůli lepší orientaci v simulačním experimentu, si model nejprve prohlédnout. V příkazovém okně MATLABu zadáme příkaz **simulink** (s malým „s“ na začátku) a v novém okně Simulinku otevřeme

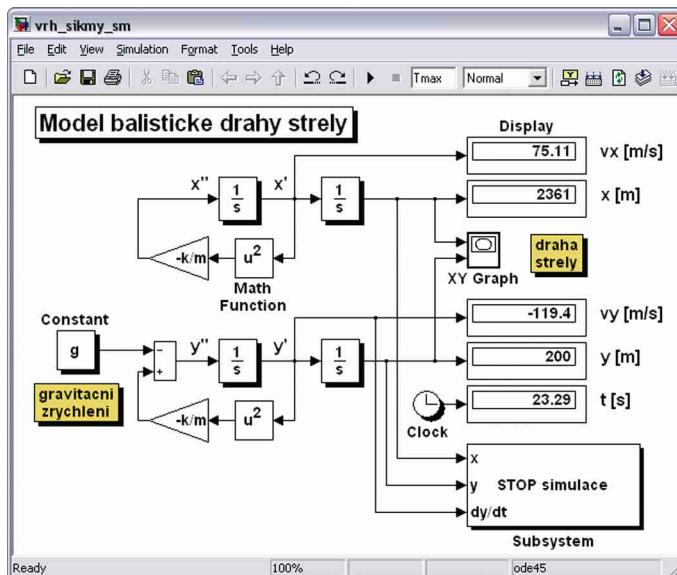
příslušný soubor modelu, tedy `vrh_sikmy_sm.mdl`. Parametry simulace můžeme měnit ve skriptu nebo také přímo v okně simulačního modelu. Limity měřítek os grafu volíme vhodně podle situace v okně parametrů (otevřeme jej dvojitým kliknutím na příslušné ikoně) zobrazovače (blok **XY Graph**) přímo v simulačním modelu.



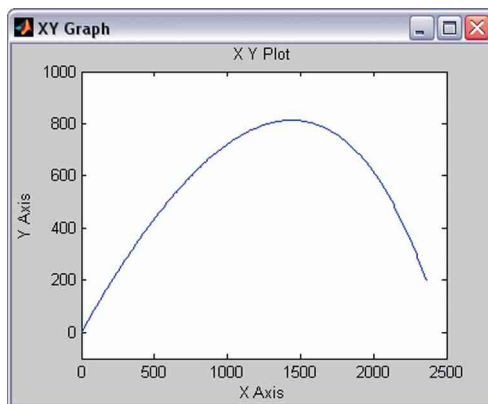
```

1 %
2 % Oblehání hradu - vrh sikmy vzhuru
3 %
4 clear all, close all, clc
5
6 % Parametry simulace
7 m=50; % hmotnost strely [kg]
8 v0=200; % v0 ... pocatecni rychlost strely
9 alfa=45; % namer ve stupnich
10 x1=10000; % souradnice cile [m]
11 y1=200; % souradnice cile [m]
12
13 g=9.81; % gravitacni zrychleni [m/s^2]
14 k=0.0134; % odpor vzduchu [Ns^2/m^2]
15 Tmax=100; % zadana max. doba simulace
16 dt=0.1; % pevne zadany simulacni krok
17 %
18
19 vx=v0*cos(pi*alfa/180); % vypocet x slozky poc.rychlosti
20 vy=v0*sin(pi*alfa/180); % vypocet y slozky poc.rychlosti
21
22 sim('vrh_sikmy_sm'); % spusti z akt. adresare "vrh_sikmy_sm.mdl"

```

Obr. 4.1-2: Skript `vrh_sikmy_s.m`Obr. 4.1-3: Model `vrh_sikmy_sm.mdl`

Názvy skriptu a modelu (byť se již liší příponou) nesmějí být jinak shodné. Na obr. 4.1-4 je uveden průběh balistické křivky při počátečních parametrech $m = 50$ kg, $v_0 = 200$ m/s, $\alpha = 45$ deg, $x_1 = 10$ km a $y_1 = 200$ m, viz obr. 4.1-2. Jak je patrné z obdrženého průběhu, cíl o souřadnicích $[x_1, y_1] = [10000, 200]$ m nebyl zasažen. Hodnoty x , v_x , y , v_y , včetně času simulace t , jsou v průběhu simulace zobrazovány pomocí číslicových displejů. Konečné hodnoty uvedených veličin jsou na obr. 4.1-3.



Obr. 4.1-4: Simulace dráhy střely

Kontrolní otázky a úkoly

- Zvolte vhodné parametry v_0 , α , m . Uvažujte vždy dva z parametrů za konstantní, třetí parametr měňte (alespoň tři různé hodnoty). Nakreslete průběhy odpovídajících balistických křivek pro $y_1 = 0$.
- Uvažujte stejně zvolené parametry jako v předchozím bodě. Nakreslete průběhy balistických křivek v normální atmosféře ($k = 0,0134 \text{ N s}^2/\text{m}^2$) a ve vzduchoprázdnu ($k = 0$). Balistická křivka střely ve vzduchoprázdnu připomíná parabolu, je tomu skutečně přesně tak? Pokuste se o vysvětlení a volte vhodnou argumentaci, ať už se domníváte, že tomu tak je nebo není.
- Uvažujte stejně zvolené parametry jako v předchozím bodě. Nakreslete opět dvě balistické křivky, jednu v normální zemské atmosféře, druhou za stejných podmínek na Měsíci. Předpokládejte, že je Měsíc prakticky bez atmosféry (vzduchoprázdno), hodnotu gravitačního zrychlení na povrchu Měsíce zjistíte sami. Zhodnoťte průběh obou křivek a pokuste se o fyzikální zdůvodnění jejich rozdílů.
- Podle pokynů učitele vypracujte krátkou technickou zprávu.

Přiložené soubory

- soubor skriptu: vrh_sikmy_s.mdl
- soubor modelu: vrh_sikmy_sm.mdl

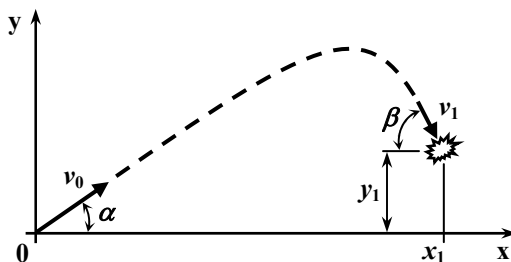
Otázky a úkoly pro pokročilé

- Pokuste se nakreslit křivky odpovídající měnícímu se parametru do jednoho grafu.
- Upravte skript a příslušné simulační schéma tak, aby bylo možné:
 - a) zadávat náměr α v gradech (jak je u dělostřelců obvyklé),
 - b) respektovat vliv větru,
 - c) respektovat měnící se gravitační zrychlení s kvadrátem výšky střely (pro střelení středověkým dělem by nemělo valný smysl, pro aplikace v raketové nebo kosmické dynamice však rozhodně ano),
 - d) respektovat reliéf terénu tak, aby nedošlo k nárazu střely do překážky.
- Bude absolutní rychlost střely v_1 při dopadu stejná jako její absolutní rychlost v_0 v okamžiku výstřelu? Může tomu tak za jistých okolností být? Může se stát, že by byla dokonce i větší? Pokuste se fyzikálně zdůvodnit vaši úvahu a ověřit ji simulací.
- Zvykněte si klást sami sobě podobné otázky a hledat na ně odpovědi, aniž by vás k tomu někdo nutil (to je velmi důležité).
- Jste-li schopni matematické analýzy, proveďte ji a konfrontujte vždy výsledek se simulací a s vaším technickým citem.

4.2 Text pro pedagoga

Příklad

Vypracujte jednoduchý návod odkud a jak střílet tak, aby se bezpečně zasáhl cíl. Formu si zvolte sami. Vypracujte tabulku nebo vhodný graf. Pokuste se o co nejpřehlednější návod, který by tehdy jistě velmi ocenili uherští žoldáci (ještě, že ho neměli, a neuměli také simulovat jako my, to by tenkrát Buchlov dopadl jistě mnohem hůře). Volte: vzdálenost od cíle $l > 500$ m vzhledem k topologii terénu, počáteční rychlost střely $100 < v_0 < 500$ m/s (tedy množství prachu), náměr $0 < \alpha < 90$ deg (stupňů), hmotnost střely $10 < m < 50$ kg a koeficient odporu vzduchu $k = 0,0134 \text{ N s}^2/\text{m}^2$. Sami také vhodně zvolte výšku cíle (hradu) y_1 nad úrovní ústí děla. Náskres celé situace je na obr. 4.2-1.



Obr. 4.2-1: Základní parametry

Řešení

Balistická křivka je výslednice dynamického pohybu vystřeleného hmotného projektilu ve dvou osách. Základními parametry jsou m [kg], v_0 [m/s] a α [deg]. Uvažujeme odpor okolního vzduchu a konstantní gravitační zrychlení, ostatní vlivy (jako je rychlost větru, vliv rotace střely atd.) pro jednoduchost zanedbáváme. Pokud bychom chtěli tyto vlivy zohlednit, dostali bychom poněkud složitější diferenciální pohybové rovnice.

Dráha pohybu střely je v tomto případě rovinnou křivkou a je určena výslednicí současného pohybu střely v ose x i v ose y . Dynamika jejího pohybu v obou směrech je dána jednoduchými diferenciálními rovnicemi.

Pohyb střely v ose x

Setrvačnou sílu lze vyjádřit obvyklým způsobem jako

$$F_1 = m \, d^2x/dt^2 = m \, x''$$

a jelikož odpor vzduchu je závislý na čtverci rychlosti

$$F_2 = k \, (dx/dt)^2 = k \, (x')^2$$

Aplikací d'Alembertova principu vyrovnání sil působících na hmotu m získáme rovnost

$$F_1 = -F_2$$

$$m \, x'' = -k \, (x')^2$$

tedy po jednoduché úpravě

$$m \, x'' + k \, (x')^2 = 0$$

Pohyb střely v ose x je dán řešením výše uvedené homogenní diferenciální rovnice s počátečními podmínkami

$$x(0) = 0$$

$$x'(0) = v_{0x} = v_0 \cos \alpha$$

Obdrželi jsme **nelineární** diferenciální rovnici, která popisuje pohyb střely ve směru x . Analytické řešení nelineárních diferenciálních rovnic je na rozdíl od rovnic lineárních obecně dosti nepříjemné. Analytickými prostředky dokážeme řešit jen některé speciální typy nelineárních rovnic. Klasické přístupy rozkladu na homogenní a partikulární část řešení či použití Laplaceovy transformace zde obecně nelze použít. Zbývají pouze metody numerické matematiky a aplikace iteračních procedur.

Na rozdíl od analytické matematiky však není pro simulaci (díky její numerické podstatě) nelinearita diferenciálních rovnic významnou komplikací. Tento příklad je zařazen mj. pro ilustraci toho, že nám simulace nabízí jednoduché elegantní řešení i v případech, kdy analytické možnosti matematiky jsou omezeny.

Pohyb střely ve **vzduchoprázdnu** ($k = 0$) je však popsán diferenciální rovnicí **lineární** $m x'' = 0$, Laplaceova transformace nám v tomto případě poskytuje elegantní a jednoduché řešení

$$x(t) \doteq X(s)$$

$$x'(t) \doteq s X(s) - x(0+)$$

$$x''(t) \doteq s^2 X(s) - s x(0+) - x'(0+)$$

S uvažováním uvedených základních vztahů transformace

$$m [s^2 X(s) - s x(0+) - x'(0+)] = 0$$

$$s^2 X(s) - v_{0x} = 0$$

Obraz a příslušný originál řešení pak je

$$X(s) = v_{0x} / s^2$$

$$x(t) = v_{0x} t \quad \eta(t)$$

a platí, že

$$v_x(t) = x'(t) = v_{0x} = \text{konst}$$

Pohyb střely v ose y

V tomto případě budeme uvažovat následující tři síly

$$F_1 = m y'' \quad \dots \text{setrvačná síla}$$

$$F_2 = k (y')^2 \quad \dots \text{odpor vzduchu}$$

$$F_3 = m g \quad \dots \text{tíhová síla}$$

Opět jednoduchou aplikací d'Alembertova principu vyrovnnání sil získáme rovnost

$$F_1 = -F_2 - F_3$$

$$m y'' = -k (y')^2 - m g$$

a po jednoduché úpravě

$$m y'' + k (y')^2 = -m g$$

Pohyb v ose y je dán řešením následující nelineární diferenciální rovnice s počátečními podmínkami

$$y'' + k/m (y')^2 = -g$$

$$y(0+) = 0$$

$$y'(0+) = v_{0y} = v_0 \sin \alpha$$

Pro pohyb střely ve vzduchoprázdnu (tj. pro $k = 0$) můžeme i zde s výhodou použít Laplaceovu transformaci

$$y(t) \doteq Y(s)$$

$$y'(t) \doteq s Y(s) - y(0+)$$

$$y''(t) \doteq s^2 Y(s) - s y(0+) - y'(0+)$$

S uvažováním uvedených vztahů

$$s^2 Y(s) - s y(0+) - y'(0+) = -g \frac{1}{s}$$

$$s^2 Y(s) - v_{0y} = -g \frac{1}{s}$$

Obraz a příslušný originál řešení je

$$Y(s) = \frac{v_{0y}s - g}{s^3} = \frac{v_{0y}}{s^2} - \frac{g}{s^3}$$

$$y(t) = (v_{0y}t - \frac{1}{2}gt^2)\eta(t)$$

a rychlost v ose y

$$v_y(t) = y'(t) = (v_{0y} - gt)\eta(t)$$

Srovnaj se známými vzorci pro volný pád ve vzduchoprázdnu – pohyb střely v ose y je rovnoměrně zrychlený.

Ve vzduchoprázdnu je balistickou křivkou parabola protínající osu x v čase $t_1 = 2 v_{0y}/g$, tj. $t_1 = (2 v_0 \sin \alpha) / g$ v bodě $x_1 = 2 v_{0x} v_{0y} / g = (2 v_0^2 \sin \alpha \cos \alpha) / g$ bez ohledu na hmotnost střely m (záleží na v_0 , neboli na energii, kterou střele na začátku udělíme). Vyloučením času t z obou pohybových rovnic dostaneme její rovnici $y = (v_{0y} / v_{0x}) x - (g / 2 v_{0x}^2) x^2$. Vrchol paraboly je $y_{\max} = (v_0^2 \sin^2 \alpha) / 2g$.

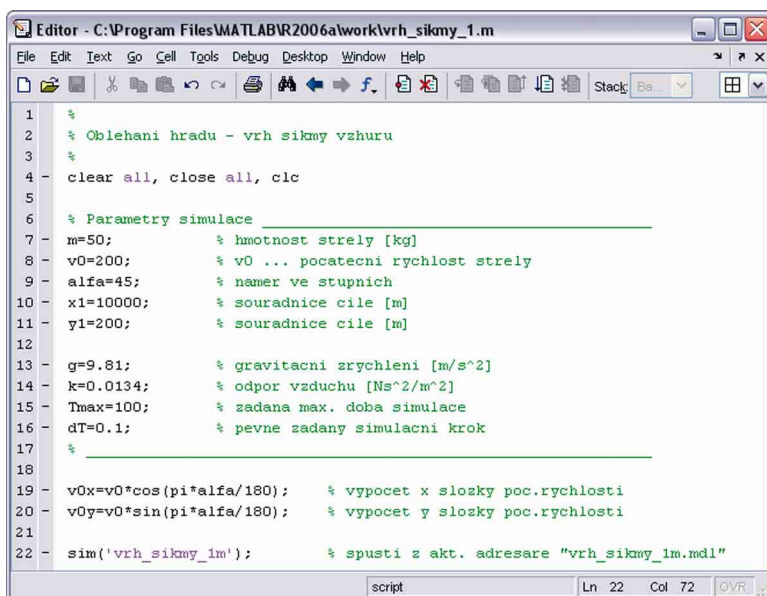
Proveďme nyní diskusi pro $\alpha \in \langle 0; \pi/2 \rangle$, $g \rightarrow 0$ a srovnajme s vrhem kolmým vzhůru. Maximální dostřel je zřejmě při daném v_0 pro $\alpha = \pi/2$, protože

$$dx_1/d\alpha = 2v_0^2 (\cos^2 \alpha - \sin^2 \alpha) / g = (2v_0^2 \cos 2\alpha) / g = 0 \Rightarrow \alpha = \pi/2$$

Při pohybu tělesa ve vzduchoprázdnu je nulová reakce prostředí. Složka rychlosti v horizontálním směru je proto konstantní, dráha je lineární funkcí času. Ve vertikálním směru je těleso stejnou gravitační silou ve stoupající části letu zpomalováno a v sestupné části naopak zrychlováno. Vertikální složka jeho dráhy odpovídající působení gravitačního zrychlení je úměrná kvadrátu času. Balistická křivka musí být proto symetrická a je popsána parabolou.

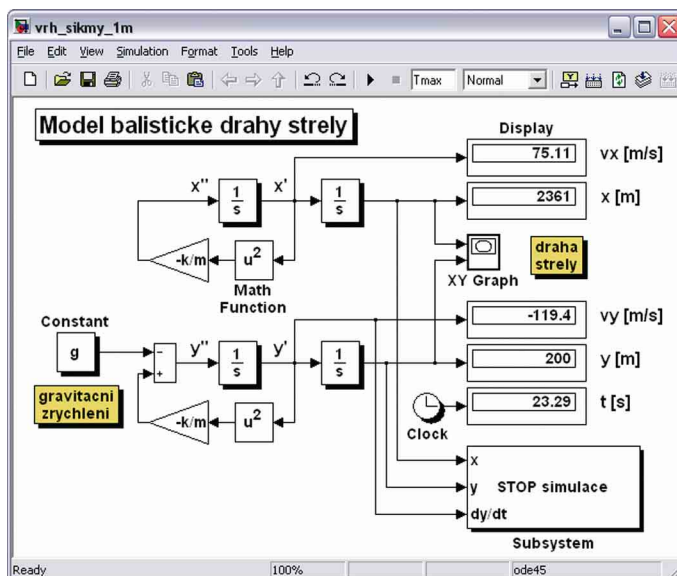
4.2.1 Vytvoření simulačního modelu

Pro řízení výpočtu v MATLABu je užitečné vytvořit tzv. skript (m-soubor), který interpretačním způsobem, jeden po druhém, vykonává sekvenci zadaných příkazů. Na obr. 4.2-2 je např. jednoduchý skript nazvaný `vrh_sikmy_1.m`. Názvy souborů v MATLABu nesmí obsahovat znak „mezera“, proto je obvykle nahrazen znakem „podtržítko“.



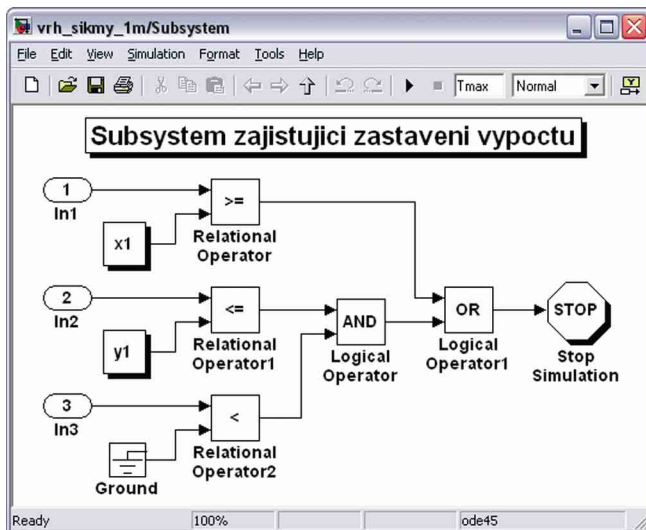
Obr. 4.2-2: Skript pro zadávání vstupních parametrů a spuštění simulace

Do skriptu je možné vepsat na libovolné místo (tedy i za příkaz) komentář uvozený znakem %, který MATLAB při vykonávání skriptu ignoruje. Způsobem popisovaným již v kapitolách 2 a 3 a podrobněji také v knize [4] vytvoříme simulační schéma na obr. 4.2-3.



Obr. 4.2-3: Model balistické křivky projektilu v Simulinku

V modelu (obr. 4.2-3) je také blok *Subsystem*, který jako celek není součástí standardní nabídky Simulinku. Je však vytvořen ze standardních logických členů pro účely konkrétní simulační úlohy. Budiž jednoduchým příkladem realizace logických funkcí v prostředí Simulink. Jeho konstrukce je z uvedené struktury zřejmá i bez komentáře. Nezapojený vstup bloku se chová jako by na něj byla přivedena trvale nulová hodnota, vhodnější je ale na takovýto vstup připojit výstup bloku *Ground* příp. blok *Constant* s hodnotou 0, čímž zamezíme varovnému hlášení ve tvaru Warning: Input port 2 of 'vrh_sikmy_1m/Subsystem/Relational Operator2' is not connected. Uvedená konstrukce (obr. 4.2-4) subsystému není jedinou možnou realizací logické funkce, pokuste se o realizaci jinou.



Obr. 4.2-4: Subsystem pro zastavení simulace při splnění definovaných podmínek

K zastavení výpočtu (běhu simulace) dojde, jestliže

- souřadnice x ve vodorovném směru dosáhne předem nastavené hodnoty x_1 ,
- je-li souřadnice y ve svislém směru při klesající fázi letu ($y' < 0$) menší než y_1 .

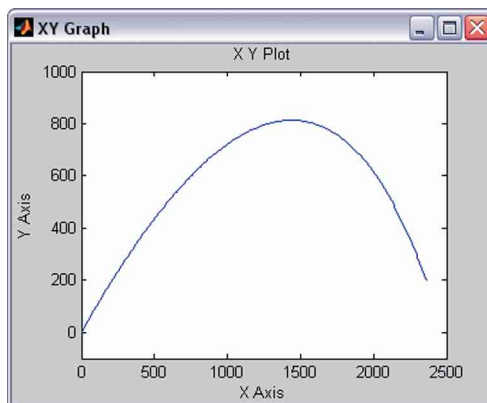
Pro spolupráci skriptu řídicího simulační výpočet se simulačním modelem je nutné zajistit přenos hodnot nastavením parametrů jednotlivých bloků modelu (je v souboru *vrh_sikmy_1m.mdl* je již nastaveno):

- počáteční podmínky integrátorů (parametr **Initial condition**) na v_{0x} (proměnná v_{0x}) resp. 0 pro pohyb v ose x a v_{0y} (v_{0y}) resp. 0 pro pohyb v ose y ,
- konstantu gravitačního zrychlení (**Constant value**) na g ,
- zesílení obou použitých zesilovačů (**Gain**), v tomto případě shodně na $-k/m$,
- parametry simulace (přístup z menu **Simulation** → **Configuration Parameters...**), maximální doba výpočtu (položka **Stop time**) na $T_{\max}$ ($T_{\max}$) a maximální délka simulačního kroku (**Max step size**) na dT .

V subsystému přerušujícím výpočet (obr. 4.2-4) po splnění zadané logické podmínky jsou prostřednictvím skriptu zadávány konstanty předpokládaných souřadnic cíle x_1 ($x1$) resp. y_1 ($y1$).

Význam jednotlivých parametrů řídících výpočet je zřejmý již z vlastního komentáře skriptu. Koeficient odporu vzduchu ($k = 0,0134 \text{ N s}^2/\text{m}^2$) zřejmě také závisí na tvaru střely a nadmořské výšce. Maximální doba výpočtu ($T_{\max} = 100 \text{ s}$) udává dobu simulace v případě, že nebude do té doby splněna logická podmínka pro přerušení výpočtu. Maximální zadaná délka simulačního kroku ($dT = 0,1 \text{ s}$) není povinným parametrem. Pokud není zadána, MATLAB délku simulačního kroku nastaví automaticky podle zvolené doby simulace a to dle vzorce $dT_{\max} = (t_{\text{stop}} - t_{\text{start}}) / 50$. Volbou kroku je samozřejmě ovlivněna rychlost simulace. V některých případech se může stát, že sice výpočet proběhne rychle, ale výsledný průběh získaný simulací není interpretován dostatečně hladkou křivkou.

Oba soubory, tj. `vrh_sikmy_1m.mdl` a `vrh_sikmy_1.m` musí být umístěny v aktuálním adresáři. Musí se lišit nejen příponou, ale i názvem. Po otevření skriptu ve vestavěném editoru můžeme měnit parametry výpočtu (a pochopitelně i celý skript). Změny je vhodné uložit a spustit výpočet v příkazovém okně MATLABu zadáním názvu skriptu bez přípony, zde tedy **vrh_sikmy_1** (potvrdíme klávesou ENTER). Výpočet lze spustit také i přímo z okna editoru, příkazem **Run** (popř. **Save and Run**) z menu **Debug** nebo stiskem klávesy F5. Simulovaný průběh dráhy střely při zachování původních parametrů (obr. 4.2-2) je uveden na obr. 4.2-5. K vykreslení byl použit blok *XY Graph*. Křivka je kreslena průběžně již při běhu simulace. Měřítka os grafu lze nastavit podle dané situace v okně parametrů bloku (otevřeme jej dvojitým kliknutím na jeho ikoně).

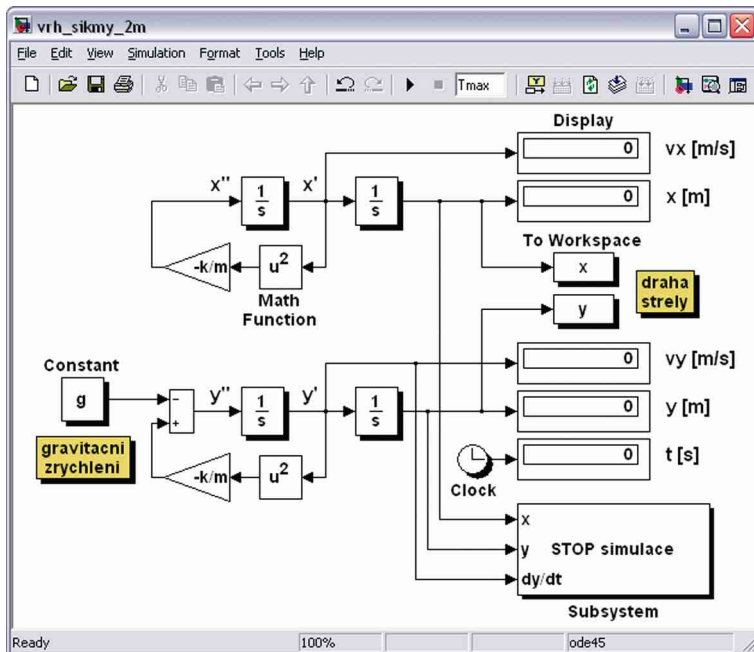


Obr. 4.2-5: Simulace dráhy střely

4.2.2 Vykreslení několika křivek do jednoho grafu

Chceme-li vykreslit několik křivek dráhy letu střely odpovídající měnícímu se parametru do jednoho obrázku je vhodné oba soubory (skript i model) poněkud modifikovat. Model rozšíříme tak, abychom při výpočtu ukládali souřadnice dráhy do příslušných polí, která nám umožní následně dráhu vykreslit, viz model `vrh_sikmy_2m.mdl` na obr. 4.2-6.

Uložení obou souřadnic zajistíme prostřednictvím speciálních bloků **To Workspace**. V okně parametrů těchto bloků je třeba zadat název proměnné, v našem případě x resp. y a také změnit formát ukládaných dat z implicitního typu **Structure** na typ **Array**.



Obr. 4.2-6: Upravený model balistické křivky

Skript řídicí průběh výpočtu upravíme tak, aby v cyklu opakovaně spouštěl simulační výpočet a vykresloval křivku dráhy letu, viz skript `vrh_sikmy_2.m` na obr. 4.2-7. Modifikace skriptu je zřejmá z jeho komentáře. Příkaz **pause**, jako poslední v těle cyklu, realizuje přerušení výpočtu po proběhnutí jednoho kroku cyklu. Výpočet pokračuje po stisknutí libovolné klávesy. Alternativou je např. příkaz **pause(n)**, který spouští výpočet automaticky po n sekundách nebo příkaz **pause** vůbec nepoužít. Pokud chceme z jakéhokoliv důvodu celý výpočet přerušit a úplně ukončit, učiníme tak stiskem kláves **Ctrl + C**.

Zde uvedená varianta skriptu (obr. 4.2-7) odpovídá měnící se hodnotě náměru α . Varianty příslušné měnícím se hodnotám jiných parametrů jsou analogické (pokuste se realizovat). Zajímavá je zejména situace s měnící se hmotností střely m a tomu odpovídající dráha letu ve vzduchoprázdnu (pro $k = 0$).

Export grafu do textového editoru

Velmi často potřebujeme pro dokumentaci výpočtu umístit obrázek simulované situace do textového editoru (MS Word apod.). Po ukončení simulačního výpočtu se nám objeví na obrazovce monitoru obrázek odpovídající nastaveným parametrům příkazu **plot** ve skriptu, kterým jsme řídili výpočet. V daném případě obrázek, viz obr. 4.2-8.

```

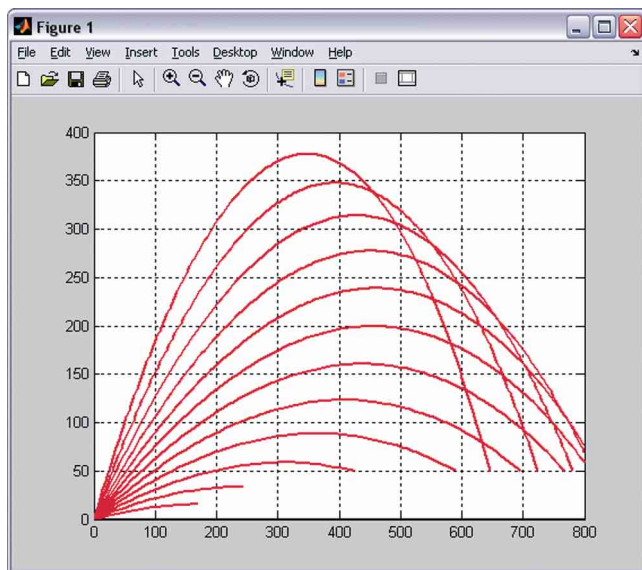
1 %
2 % Oblehání hradu - vrh síkmy vzhůru
3 % Varianta 2: simulace několika průběhů balistické křivky
4 % pro různé hodnoty nameru
5 %
6 clear all, close all, clc
7
8 % Parametry simulace
9 % Variace hodnoty nameru
10 alfa0=10; % počáteční namer ve stupních
11 alfad=5; % posazovaný krok změny nameru
12 alfa12; % počet kroku
13 %
14 m=50; % hmotnost střely [kg]
15 v0=100; % počáteční rychlost střely [m/s]
16 x1=800; % souřadnice cíle [m]
17 y1=50; % souřadnice cíle [m]
18
19 g=9.81; % gravitační zrychlení [m/s^2]
20 k=0.0134; % odpor vzduchu [Ns^2/m^2]
21 %k=0; % nulový odpor vzduchu, tj. vakuoprázdno
22 Tmax=100; % zadána max. doba simulace
23 dt=0.1; % pevně zadán simulací krok
24 % Konec pole pro zadávání parametru simulace
25
26 %vrh_sikmy_2m % otevírá okno se simulacním schématem
27 figure % otevírá okno a vykreslí osy souřadnic
28 plot([0 x1],[0 0], 'k', 'LineWidth', 2)
29 hold on % zachová aktuální okno pro kreslení
30 %break
31 for alfa = alfa0 : alfad : alfa0+alfad*(alfak-1)
32 v0x=v0*cos(pi*alfa/180); % výpočet složky x počáteční rychlosti
33 v0y=v0*sin(pi*alfa/180); % výpočet složky y počáteční rychlosti
34 sim('vrh_sikmy_2m') % spuštění "vrh_sikmy_1m.mdl" z akt. adresáře
35 plot(x,y,'r','LineWidth',2) % vykreslení dráhy, nastavení měřítka, mřížka
36 axis([0 x1 0 ceil(max(y)/100)*100]), grid on
37 pause % přerušování výpočtu po vykreslení jedné křivky
38 % pokračování stiskem libovolné klávesy
39 end % konec cyklu

```

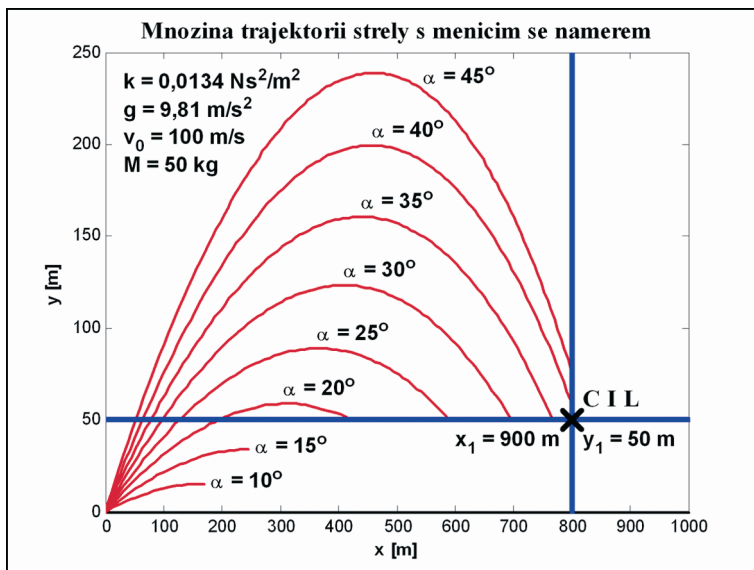
Obr. 4.2-7: Upravený skript

Máme možnost buď jemněji pochopit význam parametrů příkazu pro kreslení křivek, upravit je a zopakovat výpočet nebo využít možností interaktivní editace přímo v grafickém okně *Figure*. MATLAB umožňuje měnit parametry a barvu jednotlivých křivek, vkládat texty a měnit vzhled a měřítko kreslicí plochy. Příklad možné editace je na obr. 4.2-9.

Výsledný graf lze zkopírovat stiskem kombinace kláves Alt + Print Screen a přenést do grafického editoru pro další zpracování nebo přímo do editoru textového. Další možností je využití nabídky grafického okna *Figure* (**File** → **Save**). Obrázek uložíme ve zvoleném formátu do vybraného adresáře a následně jej vložíme standardním způsobem do textu. Lze také pomocí nabídky **Edit** → **Copy Figure** obrázek zkopírovat do schránky a následně běžným způsobem, v závislosti na používaném textovém editoru (např. pomocí kombinace kláves Ctrl + V), vložit do textu, viz obr. 4.2-9.


 Obr. 4.2-8: Několik křivek pro různé náměry α v jednom grafu

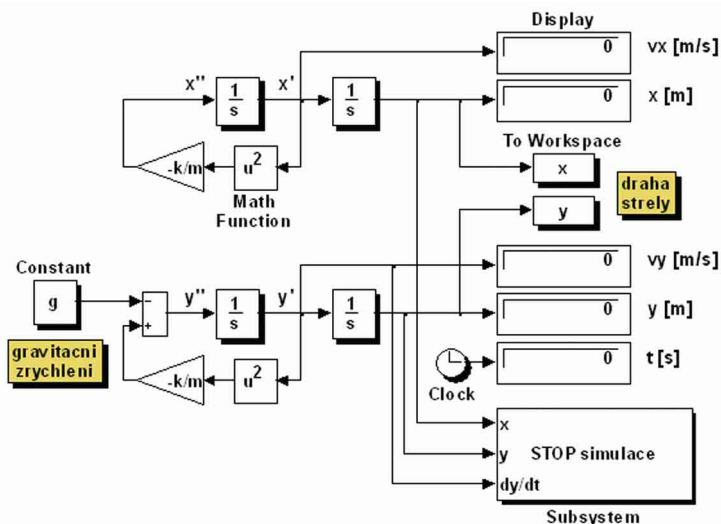
Pozorný čtenář si jistě všiml, že zásadně nevyužíváme české diakritiky v textech ani v názvech souborů. Důvodem je, že i přes deklarovanou podporu češtiny, činí MATLABu verze R2006a v některých případech čárky a háčky jisté potíže, proto se diakritice raději vyhýbáme. Jisté možnosti ovšem poskytují speciální formátovací příkazy, které dovolují měnit typ fontu a umožní i použití diakritiky [4].



Obr. 4.2-9: Úprava vzhledu grafu

Export simulačního schématu do textového editoru

Přenos simulačního schématu z pracovního okna Simulinku do textového editoru je velmi jednoduchý. Nejprve raději vypneme případný zoom zobrazení (**View → Normal (100%)**) a přesuneme motiv schématu do levého horního rohu okna (označíme vše stiskem kombinace kláves Ctrl + A a přetáhneme pomocí myši). Poté motiv schématu zkopírujeme do schránky (**Edit → Copy Model to Clipboard**) a zkopírovaný obrázek vložíme (např. stiskem Ctrl + V) do libovolného grafického editoru nebo přímo do textového editoru. Tímto způsobem bylo do tohoto textu vloženo schéma modelu na obr. 4.2-10.



Obr. 4.2-10: Exportované simulační schéma

4.2.3 Model inverzní dynamiky

Pan Boček podcenil nebezpečí a nečekal, že šelma Matyáš pošle své zvědy na nejvyšší kopec a náhle zaútočí ze zálohy bezpečně krytý okolními vrchy... Ani obránci hradu nebyli tehdy úplně bez šancí. My jsme na tom v současnosti s naší technikou o dost lépe. Pomocí simulace dokážeme zázraky, tedy téměř. Jednoduchým způsobem dokážeme měnit měřítko času, dokážeme řešení diferenciální rovnice zrychlit či zpomalit nebo dokonce tok času zcela obrátit, realizovat tzv. inverzní dynamiku. S jistou mírou nadsázky realizovat něco jako stroj času. Simulovat dynamické děje „pozpátku“, jakoby čas plynul obráceně.

Diferenciální rovnici inverzní dynamiky, tedy téhož děje běžícího „pozpátku“, získáme jednoduchou transformací času

$$\tau = k_t t, \quad k_t = -1 \Rightarrow \tau = -t, \quad d\tau = -dt$$

$$x' = dx/dt = -dx/d\tau$$

$$x'' = d^2x/dt^2 = d^2x/d\tau^2$$

$$m (d^2x/dt^2) + k (dx/dt)^2 = m (d^2x/d\tau^2) + k (-dx/d\tau)^2 = 0$$

$$m (d^2x/d\tau^2) + k (dx/d\tau)^2 = 0$$

Počáteční podmínky takto transformované diferenciální rovnice jsou podmínkami okrajovými původní rovnice v okamžiku dopadu, tedy

$$x(0) = x_1, \quad x'(0) = (dx/d\tau)|_{\tau=0} = (-dx/dt)|_{t=t_1} = -v_{1x} = -v_1 \cos \beta$$

kde x_1 , y_1 , v_1 resp. β jsou souřadnice, rychlost, resp. úhel dopadu střely.

Díky kvadratické struktuře diferenciální rovnice se její tvar transformací času formálně nezměnil. V obecném případě však ke změně dochází. U stabilní lineární diferenciální rovnice dochází touto transformací dokonce vždy k její nestabilitě.

Zcela obdobně i pro souřadnici y

$$y'' + k/m (y')^2 = -g$$

$$m (d^2y/dt^2) + k (dy/dt)^2 = m (d^2y/d\tau^2) + k (-dy/d\tau)^2 = -mg$$

$$m (d^2y/d\tau^2) + k (dy/d\tau)^2 = -mg$$

Počáteční podmínky transformované diferenciální rovnice jsou okrajovými podmínkami původní rovnice v okamžiku dopadu

$$y(0) = y_1, \quad y'(0) = (dy/d\tau)|_{\tau=0} = (-dy/dt)|_{t=t_1} = -v_{1y} = -v_1 \sin \beta$$

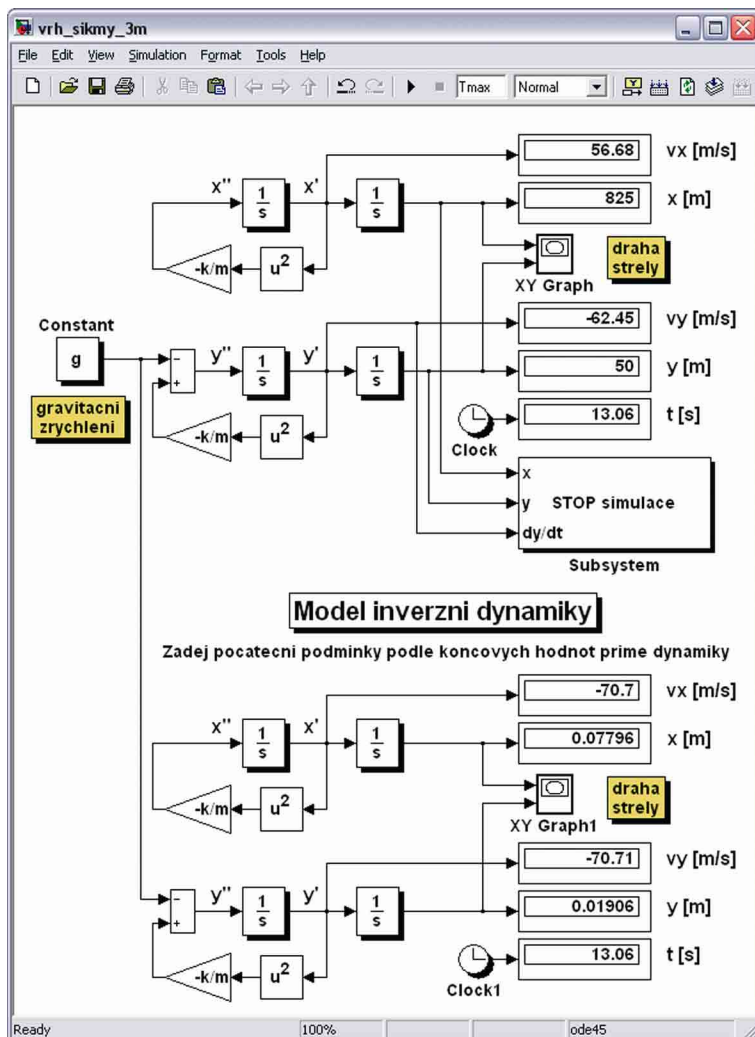
Ani v tomto případě se pohybová rovnice ve směru y ze stejných důvodů transformací času formálně nezměnila.

Rovnici inverzní dynamiky můžeme v obecném případě řešit buď analyticky nebo simulačně. Pro simulační řešení v prostředí MATLAB – Simulink rozšíříme simulační model o část inverzní dynamiky, viz `vrh_sikmy_3m.mdl`. Rozšíření simulačního schématu provedeme velmi jednoduše. Pomocí levého tlačítka myši vybereme část, kterou chceme zkopírovat, stiskneme pravé tlačítko myši (u kurzoru se objeví symbol plus), přesuneme kurzor na požadované místo a uvolníme stisknutí tlačítka. Výsledný model je na obr. 4.2-11.

Simulační výpočet můžeme realizovat buď přímo z okna Simulinku tak, že nastavíme parametry simulace na jednotlivých blocích ve schématu a spustíme řešení, nebo upravíme skript, viz `vrh_sikmy_3.m` na obr. 4.2-12, kterým řídíme výpočet. Jedná se o variantu původního skriptu `vrh_sikmy_1.m`, lišící se jen tím, že spouští výpočet jiného modelu. Práce se skriptem je jednodušší, převážně z důvodu možnosti snadno měnit parametry simulace a výpočet opakovat.

Pro ověření inverzní dynamiky spustíme nejprve výpočet řízený skriptem z pracovního okna MATLABu (v příkazovém řádku zapíšeme `vrh_sikmy_3` a potvrdíme klávesou ENTER). Po ukončení výpočtu (dopad střely) nejprve přesuneme navzájem překrytá okna obou XY zobrazovačů. Na jednom z nich bude vykreslena balistická křivka vypočtená se zadanými parametry. Dále na příslušných displejích přímé dynamiky odečteme parametry dráhy letu v okamžiku dopadu střely. Pokud simulace proběhla s parametry, které jsou uvedeny ve skriptu na obr. 4.2-12, pak: doba letu $t_1 = 13,06$ s, $x = x_1 = 825$ m, $y = y_1 = 50$ m,

$v_x = v_{x_1} = 56,68 \text{ m/s}$, $v_y = v_{y_1} = -62,45 \text{ m/s}$. Tyto hodnoty nastavíme ručně v příslušných integrátorech inverzní dynamiky jako počáteční podmínky. Při zadávání numerických hodnot je v MATLABu povinná desetinná tečka. Údaj o době letu střely je jen informativní, údajů na displejích inverzního modelu si v této fázi nemusíme všimnout. Pokud simulace inverzního modelu proběhla s nulovými počátečními podmínkami (implicitní nastavení), tak se na těchto displejích objevují parametry prostého volného pádu střely v čase t_1 .



Obr. 4.2-11: Původní model doplněný o inverzní dynamiku

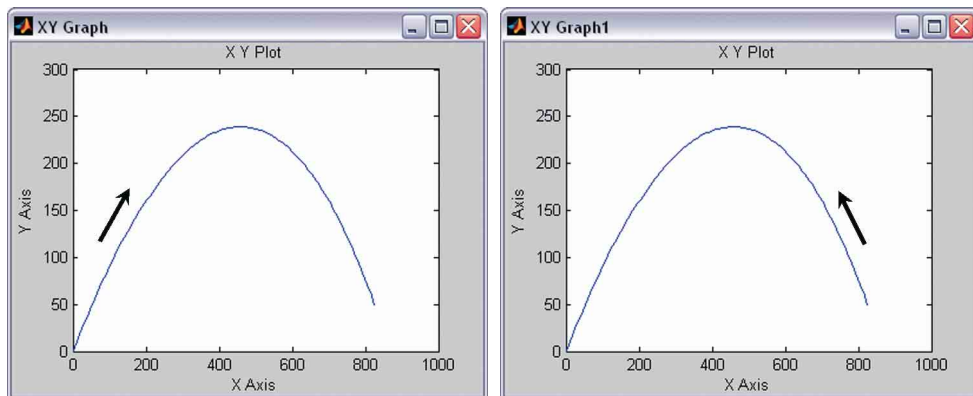
Po zadání počátečních podmínek inverzního modelu (obr. 4.2-11) spustíme výpočet z okna Simulinku. Probíhající výpočet generuje dvě identické balistické křivky, z hlediska běžícího času však s inverzním vývojem (proti sobě), viz obr. 4.2-13.

```

Editor - C:\Program Files\MATLAB\R2006a\work\vrh_sikmy_3.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1 %
2 % Oblehání hradu - vrh sikmy vzhuru
3 % Varianta 3: overeni inverzni dynamiky
4 %
5 clear all, close all, c1c
6
7 % Parametry simulace
8 m=50; % hmotnost strelly [kg]
9 v0=100; % pocatecni rychlost strelly [m/s]
10 alfa=45; % namer ve stupnich
11 x1=900; % souradnice cile [m]
12 y1=50; % souradnice cile [m]
13
14 g=9.81; % gravitacni zrychleni [m/s^2]
15 k=0.0134; % odpor vzduchu [Ns^2/m^2]
16 Tmax=100; % zadana max. doba simulace
17 dT=0.1; % pevne zadany simulacni krok
18 % Konec pole pro zadavani parametru simulace
19
20 v0x=v0*cos(pi*alfa/180); % vypocet x složky poc. rychlosti
21 v0y=v0*sin(pi*alfa/180); % vypocet y složky poc. rychlosti
22
23 vrh_sikmy_3m % otevře okno se simulacním schématem
24 sim('vrh_sikmy_3m'); % spustí z akt. adresare vrh_sikmy_3m.mdl
script Ln 24 Col 70 OVR

```

Obr. 4.2-12: Skript řídící výpočet



Obr. 4.2-13: Zobrazení výsledků simulace přímé a inverzní dynamiky střely

Probraná témata

- diferenciální popis pohybu střely v osách x a y , odvození příslušných nelineárních diferenciálních rovnic,
- nalezení analytického řešení pomocí Laplaceovy transformace ve speciálním případě pohybu střely ve vzduchoprázdnu ($k = 0$, diferenciální rovnice jsou lineární),

- řešení původních nelineárních diferenciálních rovnic v Simulinku,
- použití skriptů pro řízení výpočtu, předávání hodnot mezi skriptem a modelem, použití cyklu v těle skriptu,
- grafické zpracování výsledků simulace, uložení výsledků simulace do pole, vykreslení simulovaných křivek, vykreslení několika křivek s různými parametry do jednoho grafu,
- přenos grafu a simulačního schéma do textového editoru,
- ovládání výpočtu, přerušení simulace pomocí příkazu **pause**, použití logických funkcí v prostředí Simulinku,
- rozbor inverzní dynamiky systému a následná simulace.

Stručné shrnutí

- dráha pohybu střely je určena výslednicí současného pohybu střely v ose x i v ose y , dynamiku jejího pohybu v obou směrech lze popsat jednoduchými nelineárními diferenciálními rovnicemi,
- při sestavování obou diferenciálních rovnic vycházíme z d'Alembertova principu rovnováhy sil,
- na rozdíl od analytické matematiky není pro simulaci (díky její numerické podstatě) nelinearita diferenciálních rovnic významnou komplikací,
- během simulace můžeme souřadnice dráhy střely ukládat do prostoru proměnných a následně dráhu vykreslit pomocí standardních funkcí MATLABu, výsledný graf lze pak editovat (měřítko, popis os, legenda apod.),
- jednoduchou transformací času můžeme naleznout diferenciální rovnici inverzní dynamiky, tzn. téhož děje běžícího ale „pozpátku“, počáteční podmínky takto transformované diferenciální rovnice jsou podmínkami okrajovými původní rovnice v okamžiku dopadu,
- v názvech souborů skriptu i modelu nesmí být mezera (i jiné znaky, které se však tak často nepoužívají), českou diakritiku ve verzi R2006a raději nepoužíváme, pro spolupráci skriptu a simulačního modelu musí být oba příslušné soubory umístěny v aktuálním adresáři a nesmějí mít stejné jméno (byť se liší příponou).

Přiložené soubory

- soubory skriptu: vrh_sikmy_1.mdl, vrh_sikmy_2.mdl a vrh_sikmy_3.mdl
- soubory modelu: vrh_sikmy_1m.mdl, vrh_sikmy_2m.mdl a vrh_sikmy_3m.mdl

5 Řešené příklady

Předešlé čtyři kapitoly na konkrétních příkladech ilustrují bohaté možnosti MATLABu a Simulinku, vedou uživatele krok po kroku simulačním experimentem a vybízejí jej také k případným dalším aplikacím uvedených přístupů. Příklady jsou strukturovány do dvou úrovní. Jednak na část určenou pro pedagoga, obsahující podrobný rozbor řešení (včetně matematické analýzy a případného řešení pomocí Laplaceovy transformace) a dále na část určenou pro studenta, ve které nejsou naopak kladeny nároky na aktivní zvládnutí mnohdy složitého matematického aparátu.

V následujících odstavcích je uveden ucelený soubor příkladů, které je možné využít při výuce na středních školách technického směru s různým zaměřením. Všechny příklady jsou řešeny v prostředí MATLAB – Simulink. Teoretický rozbor je zde velmi omezen a příklady již nejsou rozděleny do dvou úrovní, tj. pro studenta a pro pedagoga. Jejich řazení vychází částečně ze struktury zavedené v knize [4], na kterou text logicky navazuje. Nejprve jsou uvedeny příklady, k jejichž řešení potřebujeme pouze prostředí MATLAB (základy práce v MATLABu, matice a vektory, datové struktury, řešení lineárních algebraických rovnic, použití grafiky, apod.), následují příklady orientované především na grafickou nadstavbu Simulink. Nejprve je ukázána realizace lineárních modelů dynamických systémů, později modelů nelineárních, příp. i složitějších modelů elektromechanických systémů.

Řazení příkladů ale z části také vychází z členění modelovaných systémů na systémy elektrické, hydraulické a mechanické či jejich vzájemné kombinace. S ohledem na uvedené lze příklady rozdělit do následujících skupin:

- I. základní použití MATLABu (např. základní vlastnosti elektrického pole, aplikace Ohmova zákona, vlastnosti střídavého signálu, impedance a výkon v RLC obvodu),
- II. maticový počet (Kirchhoffovy zákony, metoda smyčkových proudů, metoda uzlových napětí),
- III. simulace lineárních systémů (průhyb nosníku, pohyb tělesa na nakloněné rovině, dráha kyvadla),
- IV. simulace nelineárních systémů (vývoj výšky hladiny nádrže, spojené nádrže, volný pád tělesa),
- V. složitější modely mechanických a elektromechanických systémů (model navijecího zařízení, model stejnosměrného motoru).

Je zřejmé, že tématické členění příkladů tak, aby se vyhovělo struktuře vyučovaných předmětů na různých typech středních škol a zároveň byla zachována celková koncepce výkladu funkcí a vlastností samotného MATLABu a následně i Simulinku použitá již v [4], je věc prakticky nemožná. Hlavní důraz je kladen na postupné osvojování si základních technik práce s prostředím, nejprve při realizaci jednoduchých výpočtů a později i při tvorbě složitějších simulačních modelů realizovaných s využitím uživatelských skriptů. Snahou autorů sbírky je především ukázat jednoduchou a přehlednou formou na nesporné výhody využití MATLABu a Simulinku při řešení mnohdy i značně komplikovaných úloh.

Příklady jsou voleny tak, aby byly přímo využitelné při výuce, zejména v matematice, fyzice, mechanice, elektrotechnice, automatizaci a řídicí technice. Učitelům umožní tento přístup podrobně seznámit studenty s vlastnostmi mechatronických systémů, které nejsou efektivně realizovatelné na fyzikálních učebních pomůckách. Všechny v textu popisované modely a jim příslušné soubory skriptu jsou čtenáři dostupné na přiloženém datovém médiu. V následujících kapitolách nejsou ovšem konkrétní soubory popisovány. Jejich úplný přehled je uveden v příloze B.

5.1 Elektron v homogenním elektrickém poli

Určete, na jak dlouhé dráze a za jaký čas získá elektron rychlost $v = 6 \cdot 10^6 \text{ m s}^{-1}$, je-li urychlován homogenním elektrickým polem intenzity $E = 540 \text{ V m}^{-1}$. Předpokládejme, že elektron byl původně v klidu, jeho klidová hmotnost je $m = 9,1 \cdot 10^{-31} \text{ kg}$, a je nositelem elementárního náboje $q = e \doteq 1,6 \cdot 10^{-19} \text{ C}$.

V elektrickém poli intenzity o velikosti E působí na náboj q elektrická síla o velikosti

$$F_e = q E = 1,6 \cdot 10^{-19} \cdot 540 = 8,6400 \cdot 10^{-17} \text{ N}$$

Ta podle druhého Newtonova pohybového zákona udílí částici o hmotnosti m zrychlení

$$a = \frac{F_e}{m} = \frac{8,64 \cdot 10^{-17}}{9,1 \cdot 10^{-31}} \doteq 9,4945 \cdot 10^{13} \text{ m s}^{-2}$$

V homogenním elektrickém poli působí na náboj konstantní elektrická síla, která udílí částici nesoucí náboj q konstantní zrychlení. Částice (elektron) byla původně v klidu a její pohyb bude tedy přímočarý (rovnoběžný s vektorem E) a rovnoměrně zrychlený.

```

Command Window
>> v=6E6; E=540; m=9.1E-31; q=1.6E-19;
>> Fe=q*E, a=Fe/m
Fe =
    8.6400e-017
a =
    9.4945e+013
>> t=v/a, s=0.5*a*t^2
t =
    6.3194e-008
s =
    0.1896
>> |

```

Obr. 5.1-1: Řešení příkladu v MATLABu

Dráhu a čas lze za uvedených podmínek vypočítat podle známých vztahů

$$t = \frac{v}{a} = \frac{6 \cdot 10^6}{9,4945 \cdot 10^{13}} = 6,3194 \cdot 10^{-8} \doteq 63 \text{ ns}$$

$$s = \frac{1}{2} a t^2 = \frac{1}{2} \cdot 9,4945 \cdot 10^{13} \cdot (6,3194 \cdot 10^{-8})^2 = 0,1896 \doteq 0,19 \text{ m}$$

Ukázka možného řešení v MATLABu je uvedena na obr. 5.1-1.

5.2 Základní vlastnosti střídavého elektrického signálu

Střídavé elektrické napětí je popsáno vztahem

$$u(t) = 120 \cos \left(942t + \frac{\pi}{6} \right)$$

Uveďte všechny jeho vlastnosti, tj. amplitudu, frekvenci, periodu a fázi. Vypočítejte také jeho střední a efektivní hodnotu. Zapište $u(t)$ v symbolickém tvaru a uveďte jeho fázor.

Ve vztahu pro okamžitou hodnotu napětí je funkce kosinus, jedná se tedy o napětí harmonické. Amplituda $U_0 = 120 \text{ V}$, okamžitá hodnota $u(t)$ se mění od -120 do 120 V .

Úhlová frekvence $\omega = 942 \text{ rad s}^{-1}$. Frekvenci a periodu lze jednoduše vypočítat

$$f = \frac{\omega}{2\pi} = \frac{942}{2\pi} = 149,9240 \doteq 150 \text{ Hz}$$

$$T = \frac{1}{f} = \frac{1}{150} = 6,6667 \cdot 10^{-3} \doteq 6,7 \text{ ns}$$

Fáze je rovna

$$\varphi = 942t + \frac{\pi}{6}; \quad \varphi_0 = \frac{\pi}{6} = 30^\circ$$

Jelikož je uvedeno pouze jedno napětí, nelze mluvit o fázovém posuvu.

Střední a efektivní hodnotu lze určit z amplitudy pomocí známých vztahů. Připomeňme, že tyto vztahy platí pouze pro harmonický průběh.

$$U_s = \frac{2}{\pi} U_0 = \frac{2}{\pi} \cdot 120 = 76,3944 \text{ V}$$

$$U_{ef} = \frac{U_0}{\sqrt{2}} = \frac{120}{\sqrt{2}} = 84,8528 \text{ V}$$

Symbolický tvar zadaného napětí $u(t)$ je

$$u(t) = U_0 e^{j(\omega t + \varphi_0)} = 120 e^{j(942t + \frac{\pi}{6})}$$

U fázoru je nutné použít efektivní hodnotu. Můžeme také vyjádřit napětí ve složkovém tvaru

$$U = U e^{j\varphi} = 84,8528 e^{j\frac{\pi}{6}} = 84,8528 e^{j30^\circ} = (73,4847 + j42,4264)$$

Řešení příkladu v MATLABu je na obr. 5.2-1.

```

Command Window
>> U0=120; w=942; fi=30;
>> f=w/(2*pi)
f =
    149.9240
>> T=1/f
T =
    0.0067
>> |

Command Window
>> Us=2/pi*U0
Us =
    76.3944
>> Uef=U0/sqrt(2)
Uef =
    84.8528
>> |

Command Window
>> U=Uef*cosd(fi)+j*Uef*sind(fi)
U =
    73.4847 +42.4264i
>> abs(U)
ans =
    84.8528
>> angle(U)*180/pi
ans =
    30.0000
>> |

```

Obr. 5.2-1: Řešení příkladu v MATLABu

5.3 Výkon v RLC obvodu

Sériový obvod RLC má tyto parametry: $R = 84 \, \Omega$, $L = 0,4 \, \text{H}$, $C = 64 \, \mu\text{F}$, $U = 230 \, \text{V}$, $f = 50 \, \text{Hz}$. Vypočítejte jeho činný, jalový a zdánlivý výkon a účinník.

Nejdříve musíme vypočítat celkovou komplexní impedanci obvodu a následně stanovit protékající proud

$$Z = R + j\omega L + \frac{1}{j\omega C} = R + j\left(\omega L - \frac{1}{\omega C}\right)$$

$$Z = 84 + j75,9278 = 113,2300 e^{j42,1105^\circ} \, \Omega$$

$$I = \frac{U}{Z} = \frac{230 e^{j0^\circ}}{113,2300 e^{-j42,1105^\circ}} = 2,0313 e^{-j42,1105^\circ} \, \text{A}$$

Výkony P , Q a S lze určit z komplexních efektivních hodnot napětí a proudu. Označme $U = U e^{j\varphi_1}$ a $I = I e^{j\varphi_2}$. Rozdíl $\varphi = \varphi_1 - \varphi_2$ je fázový posun mezi napětím a proudem (tj. úhel, který svírají fázory U a I). Jelikož činný výkon a jalový výkon jsou definovány

$$P = UI \cos \varphi, \quad Q = UI \sin \varphi$$

musíme vytvořit takový součin fázorů napětí a proudu, aby v něm vystupoval rozdíl $\varphi = \varphi_1 - \varphi_2$. Tento rozdíl nevystupuje ve vztahu UI , protože

$$UI = U e^{j\varphi_1} I e^{j\varphi_2} = UI e^{j(\varphi_1 + \varphi_2)}$$

Jelikož však $I^* = I e^{-j\varphi_2}$ a podle Eulerovy věty $e^{j\varphi} = \cos \varphi + j \sin \varphi$, je zřejmé

$$UI^* = U e^{j\varphi_1} I e^{-j\varphi_2} = UI e^{j(\varphi_1 - \varphi_2)} = UI e^{j\varphi} = UI \cos \varphi + j UI \sin \varphi$$

Komplexní výkon S je roven

$$S = UI^*$$

takže

$$P = \operatorname{Re}[S], \quad Q = \operatorname{Im}[S], \quad S = |S|$$

Komplexní výkon S po dosazení konkrétních hodnot je

$$S = UI^* = 230 e^{j0^\circ} \cdot 2,0313 e^{j42,1105^\circ} = 467,1990 e^{j42,1105^\circ} = (346,5930 + j 313,2862) \text{ VA}$$

S využitím výše uvedeného můžeme nyní vypočítat výkony P , Q a S a účinník $\cos \varphi$

$$P = \operatorname{Re}[S] = 346,5930 \text{ W}$$

$$Q = \operatorname{Im}[S] = 313,2862 \text{ var}$$

$$S = |S| = 467,1990 \text{ VA}$$

$$\cos \varphi = P / S = 0,7419$$

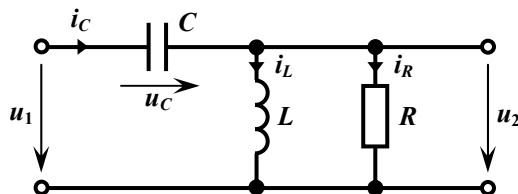
Řešení v MATLABu bude, obdobně jako v minulém příkladě, značně jednodušší. Postup je uveden na obr. 5.3-1.

Command Window	Command Window
>> R=84; L=0.4; C=64E-6;	>> S=abs(Sk)
>> U=230; f=50; w=2*pi*f;	S =
>> Z=R+j*(w*L-1/(w*C))	467.1907
Z =	>> P=real(Sk)
84.0000 +75.9278i	P =
>> I=U/Z	346.5869
I =	>> Q=imag(Sk)
1.5069 - 1.3621i	Q =
>> Sk=U*conj(I)	313.2806
Sk =	>> cos_fi=P/S
3.4659e+002 +3.1328e+002i	cos_fi =
>>	0.7419
	>>

Obr. 5.3-1: Řešení příkladu v MATLABu

5.4 Impedance RLC obvodu

Vypočítejte celkovou impedanci sérioparalelního zapojení rezistoru o odporu $R = 200 \, \Omega$, induktoru o indukčnosti $L = 1 \, \text{H}$ a kapacitoru o kapacitě $C = 5 \, \mu\text{F}$ při frekvenci $f = 50 \, \text{Hz}$ a napětí $U_1 = 230 \, \text{e}^{j0^\circ} \, \text{V}$. Zapojení tohoto RLC členu je na obr. 5.4-1. Vypočítejte také velikost napětí U_2 a proudu I_C (ve složkovém i exponenciálním tvaru).



Obr. 5.4-1: Schéma zapojení elektrického obvodu

Pro vyřešení příkladu je nutné nejprve vyjádřit impedanci zadaného RLC obvodu, provést úpravy a rozšíření komplexně sdruženým (konjugovaným) číslem osamostatnit reálnou a imaginární část. Pro zjednodušení dalších výpočtů (násobení a dělení) je vhodné výsledný složkový tvar převést na tvar exponenciální. Celá situace se poněkud zjednoduší, pokud řešení provedeme v MATLABu (obr. 5.4-2). Drobné rozdíly v některých výsledcích jsou způsobeny zaokrouhlováním.

$$Z = \frac{1}{j\omega C} + Z_2 = \frac{1}{j\omega C} + \frac{R j\omega L}{R + j\omega L}$$

$$Z_2 = \frac{R j\omega L}{R + j\omega L} \cdot \frac{R - j\omega L}{R - j\omega L} = \frac{R^2 j\omega L - R j^2 \omega^2 L^2}{R^2 - j^2 \omega^2 L^2} = \left| j^2 = -1 \right| = \frac{jR^2 \omega L + R\omega^2 L^2}{R^2 + \omega^2 L^2}$$

$$Z_2 = \frac{R\omega^2 L^2}{R^2 + \omega^2 L^2} + j \frac{R^2 \omega L}{R^2 + \omega^2 L^2} = 142,3199 + j 90,6037 = 168,7127 \, \text{e}^{j32,4816^\circ} \, \Omega$$

$$Z = 142,3199 + j 546,0161 = 564,2593 \, \text{e}^{-j 75,3908^\circ} \, \Omega$$

Proud I_C a napětí U_2 vypočteme z Ohmova zákona

$$I_C = \frac{U_1}{Z} = \frac{230 \, \text{e}^{j0^\circ}}{564,2593 \, \text{e}^{-j 75,3908^\circ}} = 0,4076 \, \text{e}^{-j 75,3908^\circ} = (0,1028 + j 0,3944) \, \text{A}$$

$$U_2 = Z_2 I_C = 168,7127 \, \text{e}^{j32,4816^\circ} \cdot 0,4076 \, \text{e}^{-j 75,3908^\circ}$$

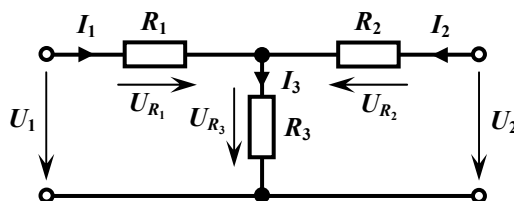
$$U_2 = 68,7673 \, \text{e}^{j107,8724^\circ} = (-21,1046 + j 65,4488) \, \text{V}$$

Command Window	Command Window
<pre>>> R=200; L=1; C=5e-6; >> U1=230; f=50; w=2*pi*f; >> Z2=(R*j*w*L)/(R+j*w*L) Z2 = 1.4232e+002 +9.0604e+001i >> abs(Z2) ans = 168.7127 >> angle(Z2)*180/pi ans = 32.4816 >> Z=Z2+1/(j*w*C) Z = 1.4232e+002 -5.4602e+002i >> abs(Z) ans = 564.2593 >> angle(Z)*180/pi ans = -75.3908 >> </pre>	<pre>>> Ic=U1/Z Ic = 0.1028 + 0.3944i >> abs(Ic) ans = 0.4076 >> angle(Ic)*180/pi ans = 75.3908 >> U2=Z*Ic U2 = -21.1054 +65.4510i >> abs(U2) ans = 68.7697 >> angle(U2)*180/pi ans = 107.8725 >> </pre>

Obr. 5.4-2: Řešení příkladu v MATLABu

5.5 Řešení elektrického obvodu přímou aplikací Kirchhoffových zákonů

Pomocí Kirchhoffových zákonů vypočítejte velikost všech neznámých proudů a napětí v obvodu z obr. 5.5-1. Konkrétní hodnoty odporu jednotlivých rezistorů jsou $R_1 = R_2 = 24 \Omega$ a $R_3 = 8 \Omega$. Napětí zdrojů jsou $U_1 = 12 \text{ V}$ a $U_2 = 6 \text{ V}$.



Obr. 5.5-1: Schéma zapojení elektrického obvodu

Aplikací Kirchhoffových zákonů postupně obdržíme soustavu lineárních algebraických rovnic

$$\begin{aligned}
 I_1 + I_2 - I_3 &= 0 \\
 -U_1 + R_1 I_1 + R_3 I_3 &= 0 \\
 -R_2 I_2 - R_3 I_3 + U_2 &= 0
 \end{aligned}$$

Rovnice lze zapsat v maticovém tvaru, dále dosadit konkrétní hodnoty

$$\begin{bmatrix} 1 & 1 & -1 \\ R_1 & 0 & R_3 \\ 0 & -R_2 & -R_3 \end{bmatrix} \cdot \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} 0 \\ U_1 \\ -U_2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & -1 \\ 24 & 0 & 4 \\ 0 & -8 & -4 \end{bmatrix} \cdot \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 12 \\ -6 \end{bmatrix}$$

a řešit např. pomocí Gaussovy eliminační metody. Pokud není determinant matice soustavy nulový, můžeme k řešení využít Cramerovo pravidlo (více viz kapitola 1.2.1)

$$\det \mathbf{A} = \begin{vmatrix} 1 & 1 & -1 \\ 24 & 0 & 8 \\ 0 & -24 & -8 \end{vmatrix} = 576 - (-192 - 192) = 960$$

$$I_1 = \frac{\begin{vmatrix} 0 & 1 & -1 \\ 12 & 0 & 8 \\ -6 & -24 & -8 \end{vmatrix}}{960} = \frac{288 - 48 - (-96)}{960} = 0,35 \text{ A}$$

$$I_2 = \frac{\begin{vmatrix} 1 & 0 & -1 \\ 24 & 12 & 8 \\ 0 & -6 & -8 \end{vmatrix}}{960} = \frac{-96 + 144 - (-96)}{960} = 0,1 \text{ A}$$

$$I_3 = \frac{\begin{vmatrix} 1 & 1 & 0 \\ 24 & 0 & 12 \\ 0 & -24 & -6 \end{vmatrix}}{960} = \frac{-(-288 - 144)}{960} = 0,45 \text{ A}$$

Napětí na jednotlivých rezistorech vypočteme pomocí Ohmova zákona

$$U_{R_1} = R_1 I_1 = 24 \cdot 0,35 = 8,4 \text{ V}$$

$$U_{R_2} = R_2 I_2 = 24 \cdot 0,1 = 2,4 \text{ V}$$

$$U_{R_3} = R_3 I_3 = 8 \cdot 0,45 = 3,6 \text{ V}$$

Řešení v prostředí MATLAB je značně jednodušší a je uvedeno na obr. 5.5-2.

Command Window

```
>> R1=24; R2=24; R3=8;
>> U1=12; U2=6;
>> A=[1 1 -1;R1 0 R3;0 -R2 -R3]
A =
    1    1   -1
   24    0    8
    0   -24   -8
>> b=[0;U1;-U2]
b =
    0
   12
   -6
>> I=A\b
I =
    0.3500
    0.1000
    0.4500
>> |
```

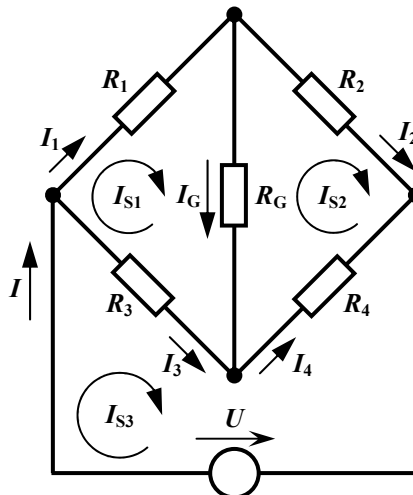
Command Window

```
>> UR1=R1*I(1), UR2=R2*I(2), UR3=R3*I(3)
UR1 =
    8.4000
UR2 =
    2.4000
UR3 =
    3.6000
>> |
```

Obr. 5.5-2: Řešení příkladu v MATLABu

5.6 Řešení elektrického obvodu metodou smyčkových proudů

Určete všechny neznámé proudy v obvodu podle obr. 5.6-1, jsou-li zadány hodnoty odporů jednotlivých rezistorů $R_1 = R_3 = 20 \Omega$, $R_2 = 40 \Omega$, $R_4 = 10 \Omega$ a $R_G = 25 \Omega$ a napětí zdroje $U = 2 \text{ V}$. Řešení proveďte metodou smyčkových proudů.



Obr. 5.6-1: Schéma zapojení elektrického obvodu

Jako neznámé zavedeme smyčkové proudy I_{S1} , I_{S2} a I_{S3} . K sestavení rovnic použijeme druhý Kirchhoffův zákon

$$R_1 I_{S1} + R_G (I_{S1} - I_{S2}) + R_3 (I_{S1} - I_{S3}) = 0$$

$$R_G (I_{S2} - I_{S1}) + R_2 I_{S2} + R_4 (I_{S2} - I_{S3}) = 0$$

$$R_3 (I_{S3} - I_{S1}) + R_4 (I_{S3} - I_{S2}) - U = 0$$

a po úpravě

$$(R_1 + R_3 + R_G) I_{S1} - R_G I_{S2} - R_3 I_{S3} = 0$$

$$-R_G I_{S1} + (R_2 + R_4 + R_G) I_{S2} - R_4 I_{S3} = 0$$

$$-R_3 I_{S1} - R_4 I_{S2} + (R_3 + R_4) I_{S3} = U$$

Soustavu lineárních algebraických rovnic zapíšeme v maticovém tvaru

$$\begin{bmatrix} R_1 + R_3 + R_G & -R_G & -R_3 \\ -R_G & R_2 + R_4 + R_G & -R_4 \\ -R_3 & -R_4 & R_3 + R_4 \end{bmatrix} \cdot \begin{bmatrix} I_{S1} \\ I_{S2} \\ I_{S3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ U \end{bmatrix}$$

$$\begin{bmatrix} 65 & -25 & -20 \\ -25 & 75 & -10 \\ -20 & -10 & 30 \end{bmatrix} \cdot \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

K řešení můžeme využít Cramerova pravidla, determinant matice **A** není nenulový

$$\det \mathbf{A} = \begin{vmatrix} 65 & -25 & -20 \\ -25 & 75 & -10 \\ -20 & -10 & 30 \end{vmatrix} = 81000$$

$$I_{S1} = \frac{\begin{vmatrix} 0 & -25 & -20 \\ 0 & 75 & -10 \\ 2 & -10 & 30 \end{vmatrix}}{81000} = \frac{3500}{81000} = 0,0432 \text{ A} = 43,2 \text{ mA}$$

$$I_{S2} = \frac{\begin{vmatrix} 65 & 0 & -20 \\ -25 & 0 & -10 \\ -20 & 2 & 30 \end{vmatrix}}{81000} = \frac{2300}{81000} = 0,0284 \text{ A} = 28,4 \text{ mA}$$

$$I_{S3} = \frac{\begin{vmatrix} 65 & -25 & 0 \\ -25 & 75 & 0 \\ -20 & -10 & 2 \end{vmatrix}}{81000} = \frac{8500}{81000} = 0,1049 \text{ A} = 104,9 \text{ mA} .$$

Proudy v jednotlivých větvích jsou

$$I_1 = I_{S1} = 43,2 \text{ mA}$$

$$I_2 = I_{S2} = 28,4 \text{ mA}$$

$$I_3 = I_{S3} - I_{S1} = 61,7 \text{ mA}$$

$$I_4 = I_{S3} - I_{S2} = 76,5 \text{ mA}$$

$$I_G = I_{S1} - I_{S2} = 14,8 \text{ mA}$$

Řešení v prostředí MATLAB je uvedeno na obr. 5.6-2.

```

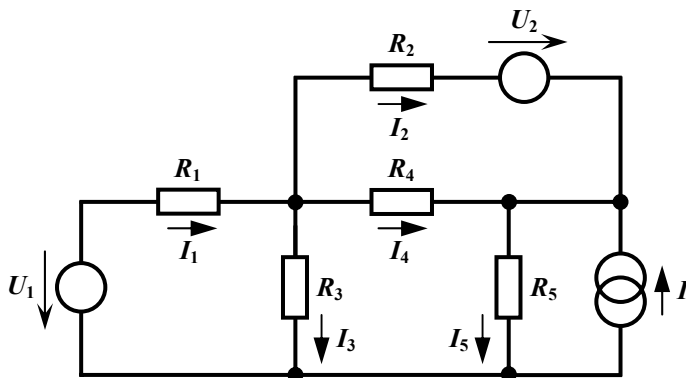
Command Window
>> R1=20; R2=40; R3=20; R4=10; RG=25; U=2;
>> A=[R1+R3+RG -RG -R3;-RG R2+R4+RG -R4;-R3 -R4 R3+R4]
A =
    65    -25    -20
   -25     75    -10
   -20    -10     30
>> b=[0;0;U]
b =
     0
     0
     2
>> Is=A\b
Is =
    0.0432
    0.0284
    0.1049
>> I1=Is(1), I2=Is(2), I3=Is(3)-Is(1), I4=Is(3)-Is(2), IG=Is(1)-Is(2)
I1 =
    0.0432
I2 =
    0.0284
I3 =
    0.0617
I4 =
    0.0765
IG =
    0.0148
>> |

```

Obr. 5.6-2: Řešení příkladu v MATLABu

5.7 Řešení elektrického obvodu metodou uzlových napětí

Vypočítejte proudy všemi rezistory v obvodu na obr. 5.7-1, jsou-li napětí zdrojů $U_1 = 10 \text{ V}$, $U_2 = 20 \text{ V}$ a proud zdroje $I = 0,2 \text{ A}$. Hodnoty odporu jednotlivých rezistorů jsou $R_1 = 20 \Omega$, $R_2 = 50 \Omega$, $R_3 = 10 \Omega$, $R_4 = 20 \Omega$ a $R_5 = 20 \Omega$.



Obr. 5.7-1: Schéma zapojení elektrického obvodu

Reálné zdroje napětí U_1 a U_2 nejprve nahradíme ekvivalentními zdroji proudu (viz obr. 5.7-2) s níže uvedenými parametry a vypočteme také vodivosti všech rezistorů

$$I_{Z1} = \frac{U_1}{R_1} = \frac{10}{20} = 0,5 \text{ A}; \quad G_1 = \frac{1}{R_1} = \frac{1}{20} = 50 \text{ mS}$$

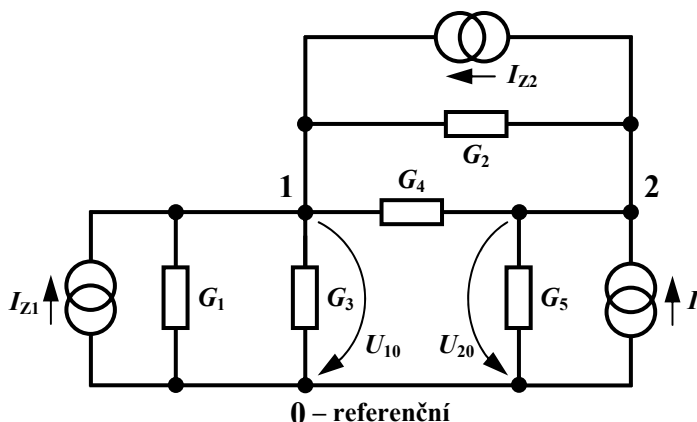
$$I_{Z2} = \frac{U_2}{R_2} = \frac{20}{50} = 0,4 \text{ A}; \quad G_2 = \frac{1}{R_2} = \frac{1}{50} = 20 \text{ mS}$$

$$G_3 = \frac{1}{R_3} = \frac{1}{10} = 100 \text{ mS}$$

$$G_4 = \frac{1}{R_4} = \frac{1}{20} = 50 \text{ mS}$$

$$G_5 = \frac{1}{R_5} = \frac{1}{20} = 50 \text{ mS}$$

Příklad budeme řešit metodou uzlových napětí, která je vhodná v případě, že analyzovaný elektrický obvod má několik paralelních větví sbíhajících se v jednom uzlu. V obvodu zvolíme jeden referenční uzel a zavedeme napětí zbývajících uzlů vůči referenčnímu – U_{10} a U_{20} .



Obr. 5.6-2: Modifikace zapojení a zavedení uzlových napětí

Pro nezávislé uzly napíšeme podle prvního Kirchhoffova zákona příslušné rovnice

$$\frac{U_{10}}{R_1} + \frac{U_{10}}{R_3} + \frac{U_{10} - U_{20}}{R_4} + \frac{U_{10} - U_{20}}{R_2} - I_{Z1} - I_{Z2} = 0$$

$$\frac{U_{20} - U_{10}}{R_4} + \frac{U_{20} - U_{10}}{R_2} + \frac{U_{20}}{R_5} + I_{Z2} - I = 0$$

analogicky pro vodivosti

$$G_1 U_{10} + G_3 U_{10} + G_4 (U_{10} - U_{20}) + G_2 (U_{10} - U_{20}) + I_{Z1} - I_{Z2} = 0$$

$$G_4 (U_{20} - U_{10}) + G_2 (U_{20} - U_{10}) + G_5 U_{20} + I_{Z2} - I = 0$$

a po úpravě

$$(G_1 + G_2 + G_3 + G_4) U_{10} - (G_2 + G_4) U_{20} = I_{Z1} + I_{Z2}$$

$$-(G_2 + G_4) U_{10} + (G_2 + G_4 + G_5) U_{20} = I - I_{Z2}$$

Soustavu lineárních algebraických rovnic zapíšeme v maticovém tvaru

$$\begin{bmatrix} G_1 + G_2 + G_3 + G_4 & -G_2 - G_4 \\ -G_2 - G_4 & G_2 + G_4 + G_5 \end{bmatrix} \cdot \begin{bmatrix} U_{10} \\ U_{20} \end{bmatrix} = \begin{bmatrix} I_{Z1} + I_{Z2} \\ I - I_{Z2} \end{bmatrix}$$

po dosazení

$$\begin{bmatrix} 220 & -70 \\ -70 & 120 \end{bmatrix} \cdot \begin{bmatrix} U_{10} \\ U_{20} \end{bmatrix} = \begin{bmatrix} 900 \\ -200 \end{bmatrix}$$

Obdobně jako v předchozích příkladech můžeme využít k řešení Cramerova pravidla, neboť determinant matice **A** není nenulový

$$\det \mathbf{A} = \begin{vmatrix} 220 & -70 \\ -70 & 120 \end{vmatrix} = 21500$$

$$U_{10} = \frac{\begin{vmatrix} 900 & -70 \\ -200 & 120 \end{vmatrix}}{21500} = \frac{94000}{21500} = 4,3721 \text{ V}$$

$$U_{20} = \frac{\begin{vmatrix} 220 & 900 \\ -70 & -200 \end{vmatrix}}{21500} = \frac{19000}{21500} = 0,8837 \text{ V}$$

Proudové rezistory jsou

$$I_3 = U_{10} G_3 = 437,2 \text{ mA}$$

$$I_4 = (U_{10} - U_{20}) G_4 = 174,4 \text{ mA}$$

$$I_5 = U_{20} G_5 = 44,2 \text{ mA}$$

$$I_2 = I_5 - I_3 + I_4 = -330,2 \text{ mA}$$

$$I_1 = I_2 + I_3 + I_4 = 281,4 \text{ mA}$$

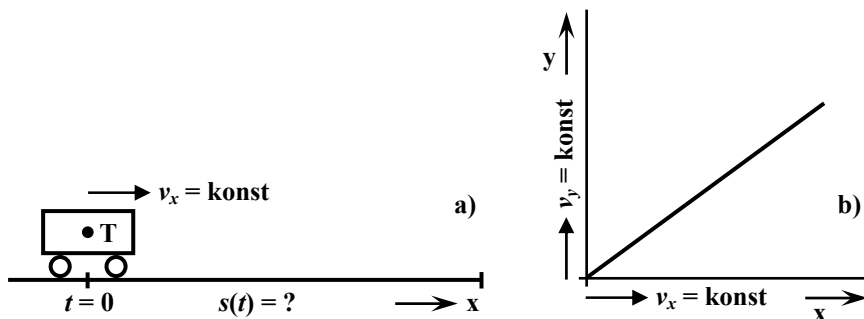
Řešení v prostředí MATLAB je uvedeno na obr. 5.6-3.

Command Window	Command Window
>> U1=10; U2=20; I=0.2;	>> U=A\b
>> R1=20; R2=50; R3=10; R4=20; R5=20;	U =
>> Iz1=U1/R1, Iz2=U2/R2	4.3721
Iz1 =	0.8837
0.5000	>> I3=U(1)*G3
Iz2 =	I3 =
0.4000	0.4372
>> G1=1/R1, G2=1/R2, G3=1/R3, G4=1/R4, G5=1/R5	>> I4=(U(1)-U(2))*G4
G1 =	I4 =
0.0500	0.1744
G2 =	>> I5=U(2)*G5
0.0200	I5 =
G3 =	0.0442
0.1000	>> I2=I5-I-I4
G4 =	I2 =
0.0500	-0.3302
G5 =	>> I1=I2+I3+I4
0.0500	I1 =
>> A=[G1+G2+G3+G4 -G2-G4;-G2-G4 G2+G4+G5]	0.2814
A =	>>
0.2200 -0.0700	
-0.0700 0.1200	
>> b=[Iz1+Iz2; I-Iz2]	
b =	
0.9000	
-0.2000	
>>	

Obr. 5.6-3: Řešení příkladu v MATLABu

5.8 Rovnoměrný přímočarý pohyb hmotného tělesa

Předpokládáme, že jde o pohyb hmotného tělesa ve směru osy x , nebo o složený pohyb ve směru osy x a osy y . Vyznačuje se konstantní rychlostí. Cílem je ukázat závislost dráhy tělesa na čase při různé konstantní rychlosti a také vznik složeného pohybu, daného součtem jednotlivých složek pohybu ve směru os x , y .



Obr. 5.8-1: Rovnoměrný přímočarý pohyb hmotného tělesa

a) pohyb tělesa ve směru osy x

b) výsledný složený pohyb

Definujme nejprve základní veličiny:

t [s]	– čas
s [m]	– dráha tělesa
$s(t)$ [m]	– okamžitá dráha tělesa v čase t
$s(0) = s_0$ [m]	– dráha na počátku pohybu v čase $t = 0$, často lze položit $s_0 = 0$
v [ms ⁻¹]	– rychlost pohybu
$v(t)$ [ms ⁻¹]	– okamžitá rychlost tělesa v čase t

Vzájemné vztahy těchto veličin odvodíme na základě následující úvahy. V krátkém časovém okamžiku Δt [s] budeme předpokládat, že se v tomto čase pohybuje těleso stálou průměrnou rychlostí v a urazí dráhu Δs [m], platí tedy vztah

$$v = \frac{\Delta s}{\Delta t}$$

Jestliže budeme časový úsek Δt zmenšovat k nule, dostaneme po tzv. limitním přechodu vztah pro okamžitou rychlost tělesa

$$v = \frac{ds}{dt}$$

kde ds , dt představují tzv. diferenciál dráhy s a diferenciál času t .

Derivaci časové funkce lze fyzikálně interpretovat tak, že okamžitá rychlost tělesa je rovna derivaci dráhy podle času. Probíhá-li pohyb tělesa konstantní rychlostí, upravíme výše uvedenou rovnici takto

$$v_x = \frac{ds(t)}{dt} \Rightarrow ds(t) = v_x dt$$

ds představuje elementární úsek dráhy. Chceme-li určit okamžitou dráhu tělesa v čase t , musíme použít matematickou operaci integrace

$$s_x(t) = \int_0^t v_x d\tau = v_x \int_0^t d\tau = v_x t = x(t)$$

Z uvedené závislosti je zřejmé, že dráha narůstá přímo úměrně s časem. Obdobná rovnost platí i pro pohyb ve směru osy y .

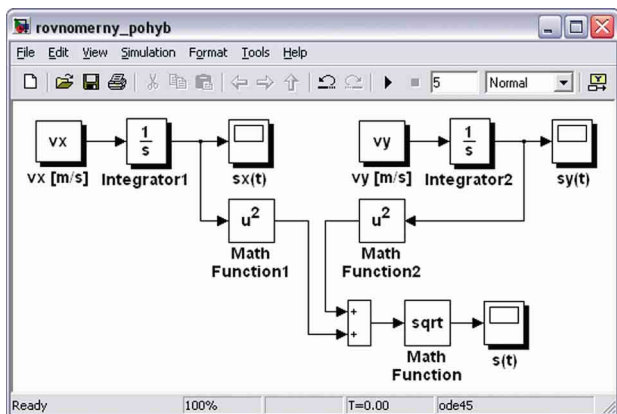
$$s_y(t) = \int_0^t v_y d\tau = v_y \int_0^t d\tau = v_y t = y(t)$$

Pro výslednou dráhu platí na základě vektorového součtu rovnice

$$s(t) = \sqrt{s_x^2(t) + s_y^2(t)}$$

Okamžitá dráha tělesa je tedy dána integrálem okamžité rychlosti vzhledem k času.

Simulační model v prostředí MATLAB – Simulink je uveden na obr. 5.8-2. Nejprve musíme vytvořit schéma pro výpočet dráhy tělesa v obou osách a následně stanovit dráhu výslednou.



Obr. 5.8-2: Struktura bloků pro výpočet dráhy pohybu tělesa

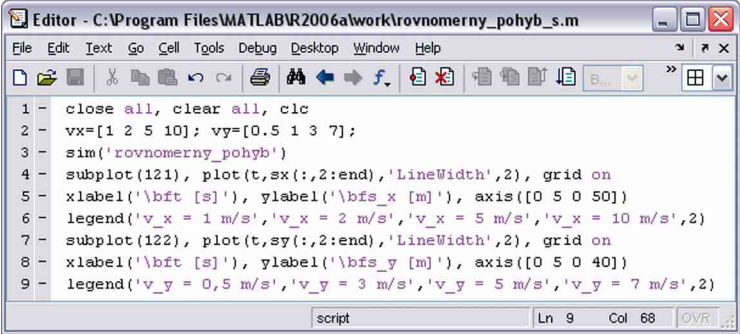
Za vstupní parametry simulace volíme postupně počáteční rychlosti $v_x = 1, 2, 5$ a 10 ms^{-1} , $v_y = 0,5, 1, 3$ a 7 ms^{-1} . Simulaci můžeme provést i pro všechny počáteční rychlosti najednou. Stačí, když definujeme jednotlivé rychlosti jako vektory, viz obr. 5.8-3.

Command Window

```
>> vx=[1 2 5 10]; vy=[0.5 1 3 7]; sim('rovnomerny_pohyb')
>> plot(t,sx(:,2:end)), figure, plot(t,sy(:,2:end))
>> figure, plot(t,s(:,2:end))
>> |
```

Obr. 5.8-3: Zadání hodnot počátečních rychlostí a vykreslení výsledků

Pokud nastavíme v zobrazovačích ukládání výsledků do proměnných – v okně parametru bloku v záložce **Data history** zvolíme **Save data to workspace**, zadáme název proměnné (s_x , s_y či s) a zvolíme strukturu ukládaných dat **Array** – lze jednoduše simulované průběhy vykreslit. Uložení času zajistíme prostřednictvím nabídky **Simulation** → **Configuration Parameters**. V záložce **Data Import/Export** v části *Save to workspace* zvolíme položku **Time** a zadáme název proměnné, v našem případě t . Vykreslení průběhů je možné pomocí příkazu **plot**, viz obr. 5.8-3.



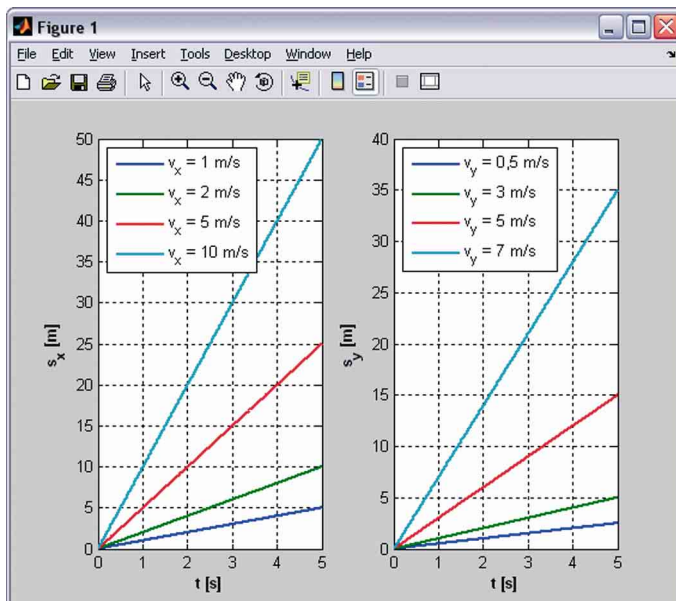
```

1 - close all, clear all, clc
2 - vx=[1 2 5 10]; vy=[0.5 1 3 7];
3 - sim('rovnomerne_pohyb')
4 - subplot(121), plot(t,sx(:,2:end),'LineWidth',2), grid on
5 - xlabel('\bft [s]'), ylabel('\bfs_x [m]'), axis([0 5 0 50])
6 - legend('v_x = 1 m/s','v_x = 2 m/s','v_x = 5 m/s','v_x = 10 m/s',2)
7 - subplot(122), plot(t,sy(:,2:end),'LineWidth',2), grid on
8 - xlabel('\bft [s]'), ylabel('\bfs_y [m]'), axis([0 5 0 40])
9 - legend('v_y = 0,5 m/s','v_y = 3 m/s','v_y = 5 m/s','v_y = 7 m/s',2)

```

Obr. 5.8-4: Příklad uživatelského skriptu

Simulační výpočet lze také zcela zautomatizovat. Zadání počátečních rychlostí, vlastní spuštění simulace a přehledné vykreslení simulovaných průběhů je výhodné provést pomocí jednoduchého uživatelského skriptu na obr. 5.8-4. Grafické zobrazení závislosti $s_x(t)$ resp. $s_y(t)$ může být podobné jako na obr. 5.8-5. Průběhy lze doplnit např. i o legendu.



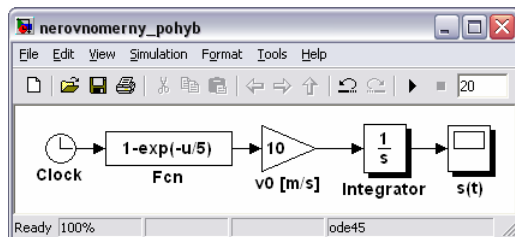
Obr. 5.8-5: Přehledné zobrazení výsledků

Se zvyšující se rychlostí narůstá lineárně dráha tělesa (za stejnou dobu vykoná těleso delší dráhu). V případě, že $v(t) = at$ (lineární nárůst rychlosti), jde o pohyb rovnoměrně zrychlený s konstantním zrychlením a . Rovnoměrný přímočarý pohyb je vlastně pohyb se zrychlením $a = 0$.

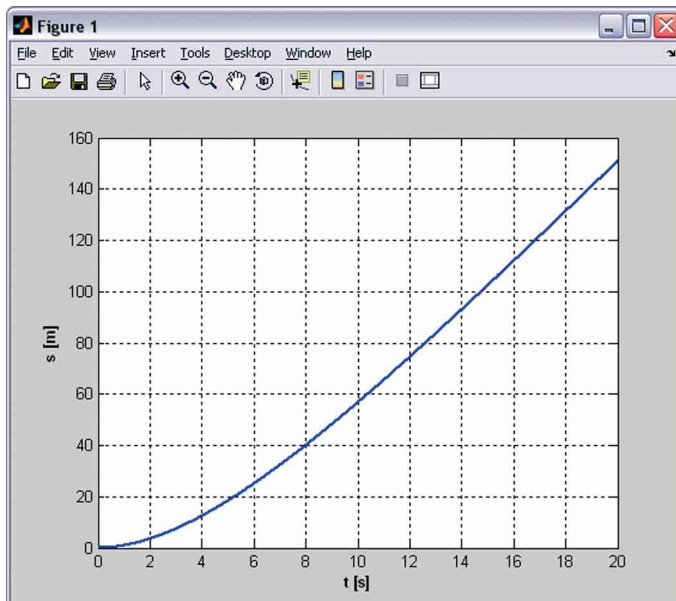
Lze vyšetřovat i nerovnoměrný pohyb tělesa, známe-li konkrétní závislost $v(t)$. Ukažme případ, kdy $v(t) = v_0 (1 - e^{-t/T})$ pro $v_0 = 10 \text{ m s}^{-1}$ a $T = 5 \text{ s}$. Pak platí

$$s(t) = \int_0^t v_0 (1 - e^{-\tau/T}) d\tau$$

Simulační schéma a příslušný záznam grafického řešení je na obr. 5.8-6 resp. 5.8-7.



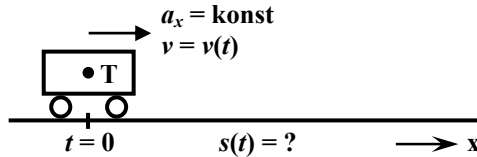
Obr. 5.8-6: Simulace nerovnoměrného pohybu



Obr. 5.8-7: Závislost dráhy na čase při proměnné rychlosti

5.9 Rovnoměrně zrychlený přímočarý pohyb

Rovnoměrně zrychlený přímočarý pohyb se děje s konstantním zrychlením a . Cílem je ukázat grafické závislosti dráhy a rychlosti pohybu tělesa na čase při tomto konstantním zrychlení, případně závislost rychlosti na probíhající dráze.



Obr. 5.9-1: Rovnoměrně zrychlený přímočarý pohyb tělesa ve směru osy x

Definujeme základní veličiny

t [s]	– čas
s [m]	– dráha tělesa
$s(t)$ [m]	– okamžitá dráha tělesa v čase t
$s(0) = s_0$	– dráha na počátku pohybu v čase $t = 0$, často lze položit $s_0 = 0$
v [ms ⁻¹]	– rychlost pohybu
$v(t)$ [ms ⁻¹]	– okamžitá rychlost tělesa v čase t
$v(0) = v_0$ [ms ⁻¹]	– počáteční rychlost tělesa, nejčastěji $v_0 = 0$
$a = \text{konst}$ [ms ⁻²]	– zrychlení rovnoměrně zrychleného pohybu tělesa

Předpokládáme, že v krátkém časovém okamžiku Δt působí na těleso konstantní zrychlení a a dojde ke zvýšení rychlosti o Δv , pak platí

$$a = \frac{\Delta v}{\Delta t} = \frac{\Delta^2 s}{\Delta t^2}$$

Jestliže budeme zmenšovat časový úsek Δt k nule, pak po limitním přechodu dostaneme vztah (tj. pro $\Delta t \rightarrow dt$)

$$a = \frac{dv(t)}{dt} = \frac{d^2 s(t)}{dt^2}$$

Obdrželi jsme diferenciální rovnici druhého řádu. Jejím řešením můžeme stanovit dráhu, kterou urazí těleso, je-li mu uděleno zrychlení. Předpokládejme, že čas se začíná počítat od okamžiku počátku pohybu. Dráhu a rychlost tělesa v čase t tedy určíme integrací

$$s(t) = \int_0^t v(\tau) d\tau + s_0$$

$$v(t) = \int_0^t a(\tau) d\tau + v_0$$

Uvedené rovnice jsou matematickým modelem rovnoměrně zrychleného přímočarého pohybu tělesa.

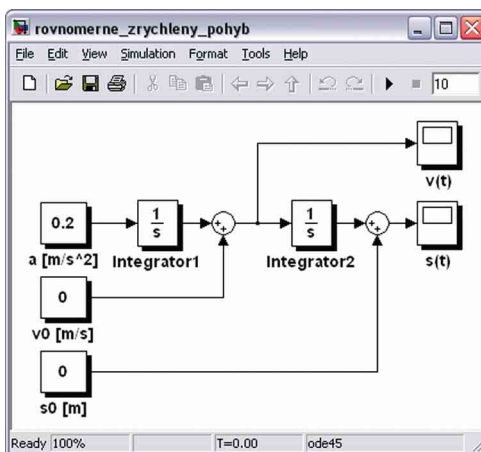
Fyzikálně lze výše uvedené vztahy interpretovat tak, že okamžitá dráha tělesa je dána integrálem z okamžité rychlosti tělesa vzhledem k času; okamžitá rychlost tělesa je pak dána integrálem zrychlení vzhledem k času.

Je-li $a(t) = a = \text{konst}$, pak

$$v(t) = \int_0^t a \, d\tau = a \int_0^t d\tau = at$$

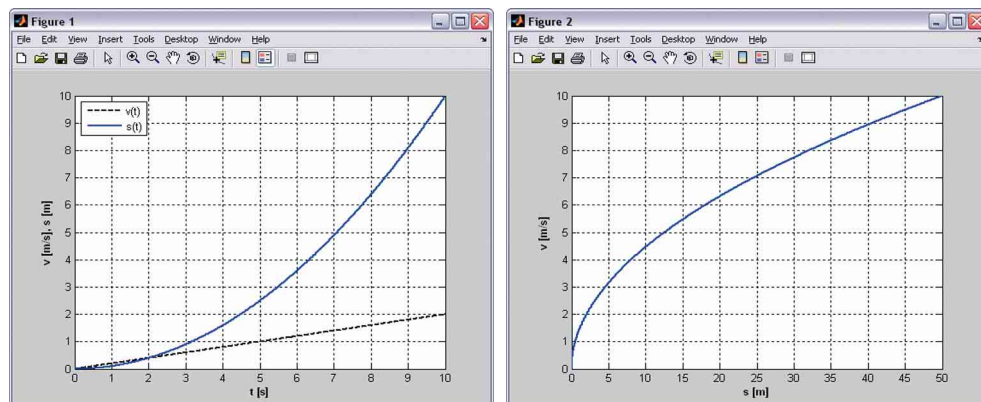
což je známá lineární závislost mezi rychlostí a časem.

Uvedenou soustavu rovnic lze řešit pomocí struktury dvou sériově zapojených integrátorů s konstantním vstupem (obr. 5.9-2), daným hodnotou zrychlení a .



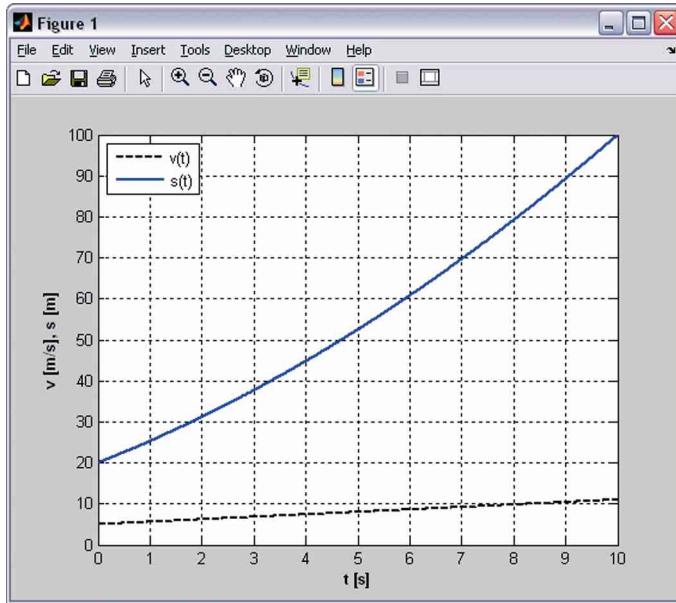
Obr. 5.9-2: Simulace rovnoměrně zrychleného pohybu

Simulaci provedeme pro tyto parametry: $s_0 = 0 \text{ m}$, $v_0 = 0 \text{ m s}^{-1}$, v čase $t = 0$ začne působit konstantní zrychlení $a = 0,2 \text{ m s}^{-2}$. Řešení provedeme v čase $t = 0$ až 10 s. Úlohu budeme řešit též pro $a = 0,6 \text{ m s}^{-2}$, $v_0 = 5 \text{ m s}^{-1}$, $s_0 = 20 \text{ m}$.



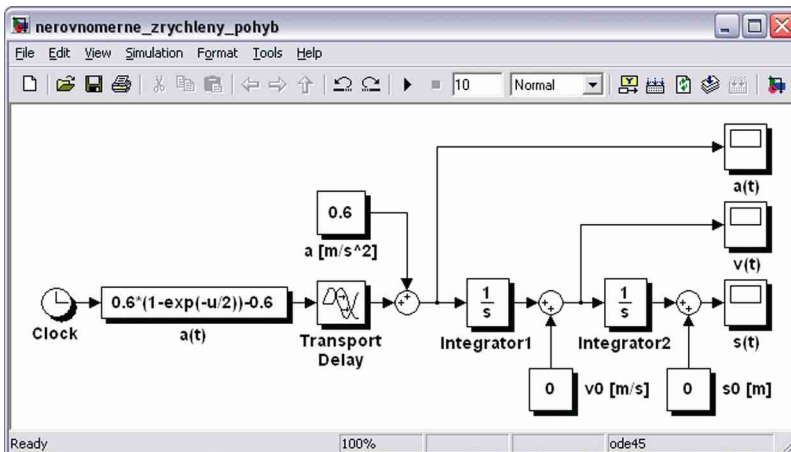
Obr. 5.9-3: Časový průběh dráhy a rychlosti tělesa při $a = 0,2 \text{ m s}^{-2}$ a závislost $v = f(s)$

Z prvního grafu na obr. 5.9-3 vyplývá, že u přímočarého rovnoměrně zrychleného pohybu s konstantním zrychlením rychlost tělesa lineárně narůstá, zatímco dráha tělesa narůstá kvadraticky.

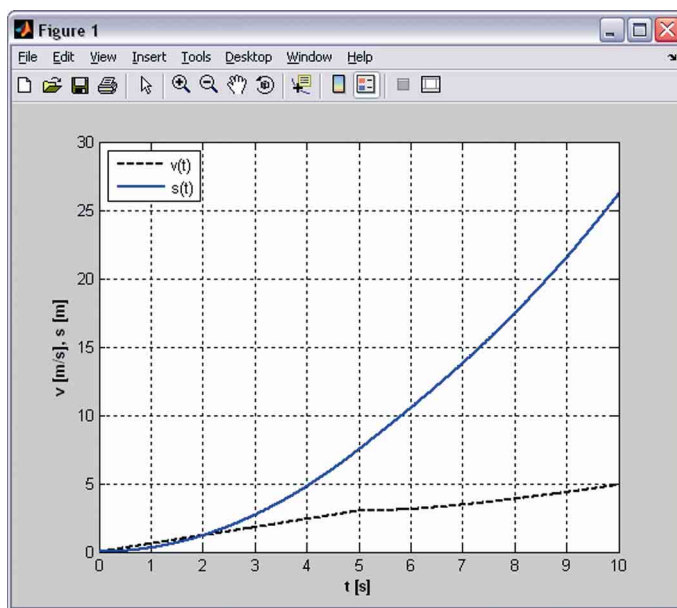


Obr. 5.9-4: Časový průběh dráhy a rychlosti tělesa při $a = 0,6 \text{ ms}^{-2}$, $v_0 = 5 \text{ ms}^{-1}$ a $s_0 = 20 \text{ m}$

Simulační model lze po úpravě (obr. 5.9-5) využít i pro řešení nerovnoměrně zrychleného pohybu, kdy zrychlení a není konstantní, ale je známou funkcí času, např. $a(t) = 0,6 \text{ m s}^{-2}$ pro $t = 0$ až 5 s , pro $t = 5$ až 10 s je $a(t) = 0,6(1 - e^{-t/2})$.



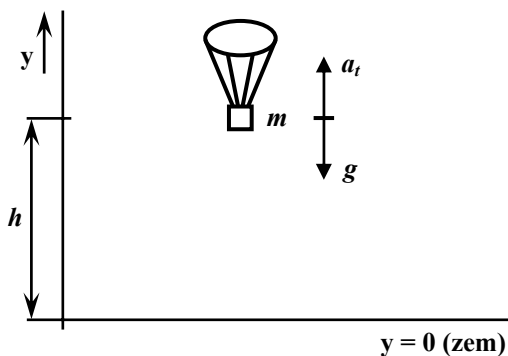
Obr. 5.9-5: Simulace nerovnoměrně zrychleného pohybu



Obr. 5.9-6: Časový průběh dráhy a rychlosti u nerovnoměrně zrychleného pohybu

5.10 Volný pád tělesa

Volný pád tělesa je svislý rovnoměrně zrychlený pohyb z výšky h s konstantním gravitačním zrychlením $a = g = 9,81 \text{ ms}^{-2}$. Vyšetřeme závislost dráhy a rychlosti na čase, pohybuje-li se těleso reálně v atmosféře a ve speciálním případě ve vakuu. Úloha umožňuje simulovat vliv odporující síly atmosférického ovzduší na průběh dráhy a rychlosti tělesa.



Obr. 5.10-1: Volný pád tělesa

Při volném pádu tělesa v atmosféře klade ovzduší pohybu tělesa odporovou (brzdící) sílu

$$F_t = \frac{1}{2} C S \rho v^2$$

Označíme-li $k = (CS\rho)/2$, pak odporová síla F_t udělí tělesu zrychlení

$$a_t = \frac{k v^2}{m}$$

Definujme veličiny

t [s]	– čas
m [kg]	– hmotnost tělesa
g [ms ⁻²]	– gravitační zrychlení $g = 9,81 \text{ ms}^{-2}$
F_t [N]	– odporová (brzdící) síla ovzduší
C [–], k [kg m ⁻¹]	– konstanty úměrnosti
S [m ²]	– průřez tělesa kolmý ke směru pohybu
ρ [kg m ⁻³]	– hustota vzduchu
$y(t)$ [m]	– okamžitá výška tělesa nad zemí
$y(0) = h$ [m]	– počáteční výška, ve které se nachází těleso (v čase $t = 0$)
$y'(t) = dy(t)/dt$	– okamžitá rychlost padajícího tělesa
v [ms ⁻¹]	– rychlost tělesa
$v(0) = v_0$ [ms ⁻¹]	– počáteční rychlost tělesa $v_0 = 0 \text{ ms}^{-1}$
a_t [ms ⁻²]	– zrychlení působící proti směru pohybu
$y''(t) = dv(t)/dt = d^2y(t)/dt^2$ [ms ⁻²]	– okamžité zrychlení padajícího tělesa

Odporová síla závisí na tvaru tělesa, což zohledníme koeficientem C . Pro kouli platí přibližně (zjištěno na základě experimentů) $C = 0,48$, pro dutou polokouli (např. padák) volíme $C = 1,33$, pro obdélníkový profil je $C = 0,55$ a pro dokonalý aerodynamický tvar je $C = 0,01$.

Dynamika volného pádu tělesa o hmotnosti m v atmosféře, charakterizované konstantou k , je dán diferenciální rovnicí druhého řádu

$$y''(t) = \frac{k}{m} [y'(t)]^2 - g$$

$$y(0) = h, \quad y'(0) = 0$$

Vzhledem k tomu, že se v rovnici vyskytuje kvadratický člen $(y')^2$, jde o nelineární diferenciální rovnici, avšak pomocí počítače v prostředí Simulink dobře řešitelnou.

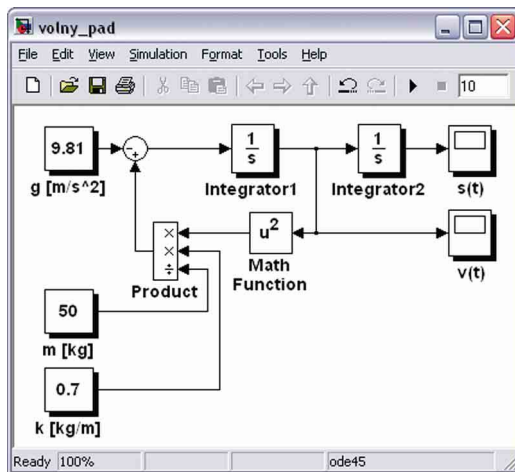
Označíme-li $y'(t) = v(t)$ a $y''(t) = v'(t)$, převedeme uvedenou diferenciální rovnici na soustavu dvou rovnic prvního řádu

$$y'(t) = v(t)$$

$$v'(t) = \frac{k}{m} [v(t)]^2 - g$$

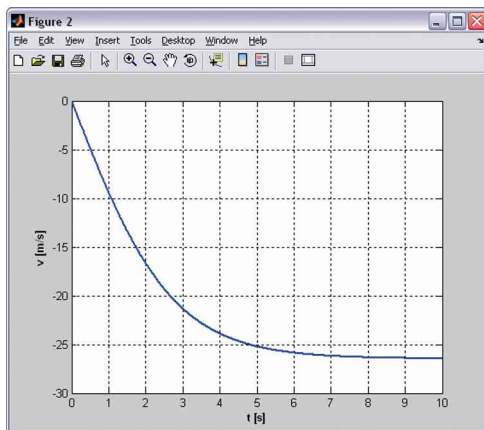
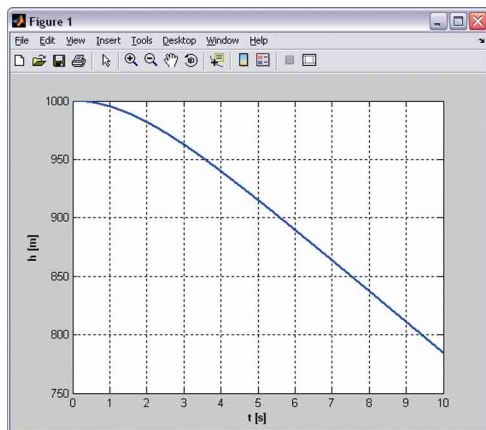
$$y(0) = h, \quad y'(0) = v(0) = 0$$

Každou rovnici řeší jeden integrátor, kompletní blokové schéma (obr. 5.10-2) je tvořeno strukturou dvou integrátorů se vstupem $-g$ a zpětnou vazbou přes kvadratický člen. Simulaci provedeme pro tyto parametry: $m = 50 \text{ kg}$, $k = 0,7 \text{ kg m}^{-1}$ a $h = 1000 \text{ m}$. Počáteční výšku $y(0) = h$ lze přičíst k signálu $y(t)$ za druhým integrátorem, obdobně jako v příkladě 5.9, nebo ji lze přímo zadat jako počáteční podmínku v druhém integrátoru.



Obr. 5.10-2: Simulace volného pádu tělesa

Model na obr. 5.10-2 umožňuje graficky znázornit časové závislosti dráhy (výšky) tělesa $y(t)$ a jeho rychlosti $v(t)$ při daných parametrech m , k a h . Uvedené závislosti jsou v grafech na obr. 5.10-3.



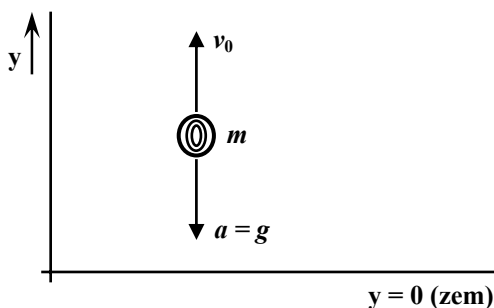
Obr. 5.10-3: Závislost výšky a rychlosti tělesa na čase

Na výstupech obou integrátorů můžeme sledovat časové průběhy klesající výšky tělesa až po dosažení země, tj. $y(t_d) = 0$ a okamžité rychlosti tělesa v závislosti na zvolených parametrech pohybu m , k , h . Změnou konstanty k zohledňujeme vliv atmosféry. Je-li $k = 0$ a tedy i $F_t = 0$, pohyb probíhá ve vakuu a platí obvyklé rovnice dráhy a rychlosti pro volný pád.

Při pomalých pohybech v odporujícím prostředí je brzdná síla vyvolána vnitřním třením ve vazké tekutině (kapalina, vzduch) a je přímo úměrná rychlosti tělesa – jde tedy o lineární závislost. V tomto speciálním případě by ve zpětné vazbě prvního integrátoru odpadl nelineární kvadratický člen.

5.11 Vrh svislý vzhůru

Vrh svislý je přímočarý rovnoměrně zrychlený pohyb, při kterém těleso o hmotnosti m vypustíme s určitou počáteční rychlostí v_0 vzhůru. Těleso se pohybuje v gravitačním poli, které působí proti směru pohybu. Cílem úlohy je určit časový průběh dráhy a rychlosti tělesa.



Obr. 5.11-1: Vrh svislý vzhůru

Definujme veličiny

t [s]	– čas
g [ms^{-2}]	– gravitační zrychlení $g = 9,81 \text{ ms}^{-2}$
$h(t)$ [m]	– okamžitá výška tělesa nad zemí
$h(0) = h_0$ [m]	– počáteční výška, ve které se nachází těleso (v čase $t = 0$)
$y'(t) = dh(t)/dt$	– okamžitá rychlost padajícího tělesa
v [ms^{-1}]	– rychlost tělesa
$v(0) = v_0$ [ms^{-1}]	– počáteční rychlost tělesa
$y''(t) = dv(t)/dt =$ $= d^2h(t)/dt^2$ [ms^{-2}]	– okamžité zrychlení tělesa
a [ms^{-2}]	– zrychlení tělesa

Svislý vrh vzhůru se skládá z rovnoměrného přímočarého pohybu směrem vzhůru a volného pádu. Pro okamžitou rychlost v a výšku nad zemí h v čase t platí vztahy

$$v = v_0 - g t$$

$$h = v_0 t - \frac{1}{2} g t^2$$

Část svislého vrhu vzhůru, kdy těleso stoupá, se nazývá výstup; těleso při něm koná *rovnoměrně zpomalený pohyb*. Výstup končí, je-li okamžitá rychlost rovna 0. Potom následuje volný pád. Čas, který bude trvat výstup je doba výstupu T_v , a těleso vystoupá do výšky vrhu $h_{\max}$.

$$0 = v_0 - g T_v \Rightarrow T_v = v_0 / g$$

$$h_{\max} = v_0 T_v - \frac{1}{2} g T_v^2 = \frac{v_0^2}{g} - \frac{1}{2} g \frac{v_0^2}{g^2} = \frac{v_0^2}{2g}$$

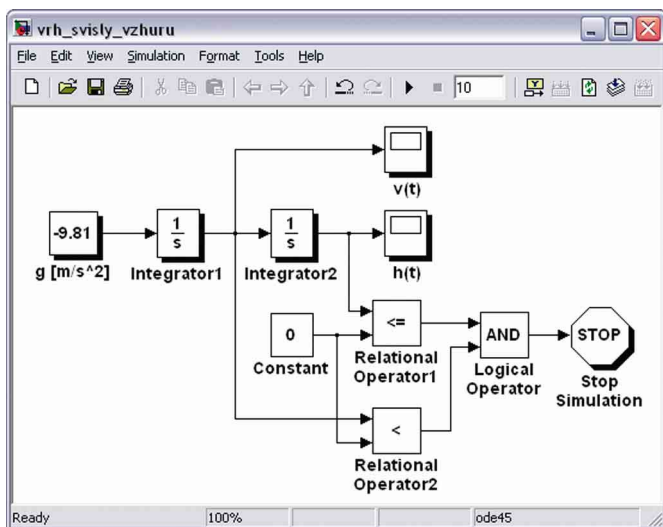
Matematický model bude obdobný jako u přímočarého rovnoměrně zrychleného pohybu, zrychlení má ale opačný smysl.

$$y'' = a = -g$$

$$y' = -\int_0^t g d\tau = -g t + v_0$$

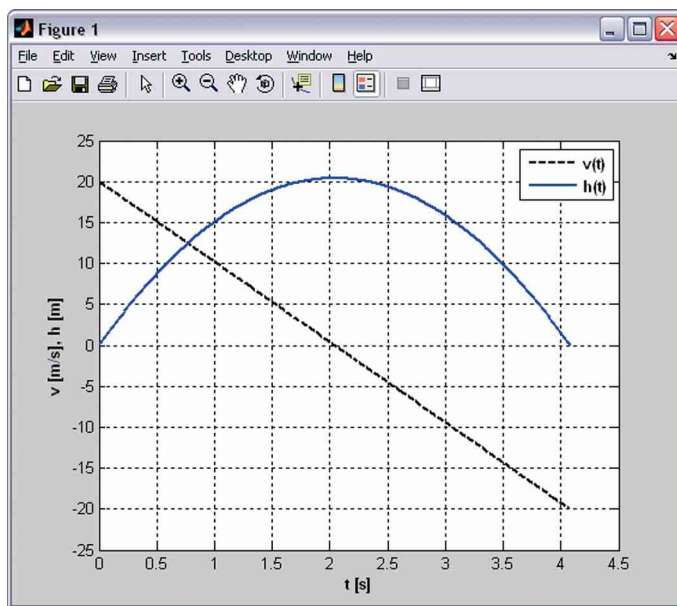
$$h = y = \int_0^t y' d\tau = \int_0^t g \tau d\tau + \int_0^t v_0 d\tau = -\frac{g t^2}{2} + v_0 t + h_0$$

Simulaci provedeme pro tyto parametry: $v_0 = 20 \text{ ms}^{-1} = 72 \text{ kmh}^{-1}$, $h_0 = 0 \text{ m}$. Počáteční rychlost v_0 a výšku h_0 zadáme jako počáteční podmínky v integrátorech.



Obr. 5.11-2: Simulace svislého vrhu vzhůru

Model je vhodné doplnit o logiku zastavení simulace v okamžiku dopadu tělesa na zem, viz obr. 5.11-2. V okamžiku, kdy nastane dopad, tedy $h = 0$, dojde prostřednictvím bloku *Stop Simulation* k zastavení simulace. Aby nedošlo k zastavení simulace ihned na počátku, je výhodné sledovat, zda je těleso již ve fázi volného pádu, tj. platí-li, že $v(t) < 0$.

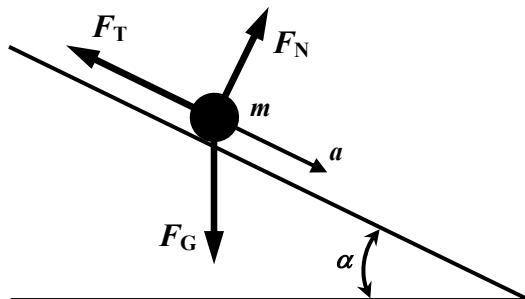


Obr. 5.11-3: Závislost výšky a rychlosti tělesa na čase

Z grafů na obr. 5.11-3 vyplývá, že dráha nejprve narůstá do doby až těleso dosáhne maximální výšky, pak začíná klesat. Způsobuje to vliv gravitačního zrychlení. Rychlost pohybu klesá lineárně s časem. Model umožňuje určit maximální výšku $h_{\max}$, kdy je rychlost $v = v' = 0$ a celkovou dobu trvání pohybu, tedy do okamžiku $h = 0$, kdy těleso opět dosáhne země.

5.12 Pohyb tělesa po nakloněné rovině

Pohyb tělesa po nakloněné rovině je přímočarý rovnoměrně zrychlený pohyb. Těleso o hmotnosti m položené na nakloněnou rovinu se pohybuje směrem dolů vlivem zemské gravitace. Cílem úlohy je simulovat průběh dráhy a rychlosti pohybu tělesa po nakloněné rovině a vyšetřit vliv počáteční rychlosti v_0 , úhlu sklonu φ a tření zohledněného koeficientem smykového tření f . Schématicky je celá situace znázorněna na obr. 5.12-1.



Obr. 5.12-1: Pohyb tělesa po nakloněné rovině

Definujeme základní veličiny

t [s]	– čas
m [kg]	– hmotnost tělesa
φ [°]	– úhel sklonu roviny
F_G, F_N, F_T [N]	– tíhová, normálová a třecí síla
g [ms ⁻²]	– gravitační zrychlení $g = 9,81 \text{ ms}^{-2}$
f [–]	– koeficient smykového tření
$s(t)$ [m]	– dráha tělesa v čase t
$s(0) = s_0$ [m]	– počáteční dráha (poloha) tělesa
$v(0) = v_0$ [ms ⁻²]	– počáteční rychlost tělesa
a [ms ⁻²]	– zrychlení tělesa

Pohybová rovnice pro pohyb tělesa na nakloněné rovině je

$$m a = F_G + F_N - F_T$$

Uvedenou rovnici rozepíšeme do jednotlivých složek v pravoúhlém souřadném systému, kde osa t (tečná) je shodná se sklonem roviny a osa n (normálová) je na rovinu kolmá. Tedy platí

$$m a = F_G \sin \varphi - F_T$$

$$0 = -F_G \cos \varphi + F_N$$

Pro třecí a normálovou sílu platí vzájemný vztah

$$F_T = f F_N$$

a po dosazení

$$m a = -F_G f \cos \varphi + F_G \sin \varphi$$

Je-li gravitační síla je dána vztahem

$$F_G = m g$$

pak po dosazení pro zrychlení a platí vztah

$$a = s''(t) = g(\sin \varphi - f \cos \varphi)$$

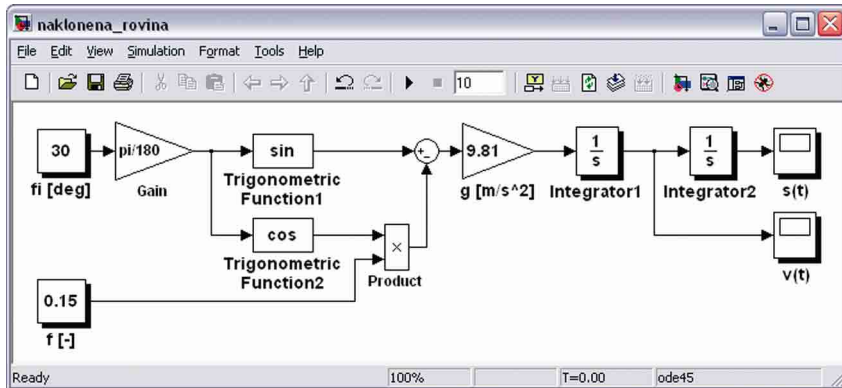
Jelikož parametry g , f a φ jsou konstanty, je zrychlení a též konstantní.

Postupnou integrací výše uvedené rovnice získáme vztah pro určení dráhy s v čase t . Přitom předpokládáme, že čas se začíná počítat od okamžiku počátku pohybu. Dráha je dána vztahem

$$s'(t) = \int_0^t g(\sin \varphi - f \cos \varphi) d\tau = g t (\sin \varphi - f \cos \varphi) + v_0$$

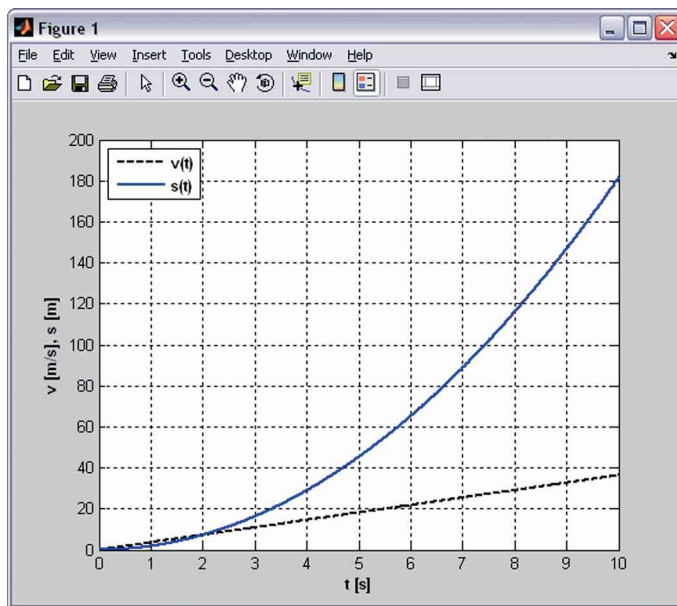
$$s(t) = \int_0^t g \tau (\sin \varphi - f \cos \varphi + v_0) d\tau + s_0$$

Tento systém rovnic je matematickým modelem pohybu tělesa po nakloněné rovině. Simulaci provedeme pro tyto parametry: $v_0 = 0 \text{ ms}^{-1}$, $s_0 = 0 \text{ m}$, $\varphi = 30^\circ \approx 0,524 \text{ rad}$ a $f = 0,15$. Příslušné simulační schéma je na obr. 5.12-2.



Obr. 5.12-2: Simulace pohybu po nakloněné rovině

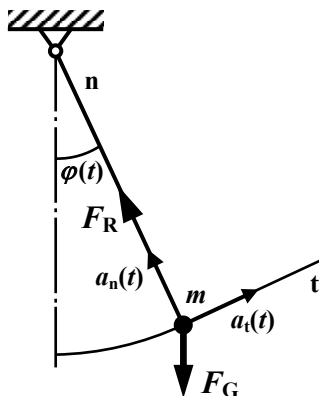
Model umožňuje sledovat časový průběh okamžitého zrychlení na vstupu prvního integrátoru, rychlost na jeho výstupu a dráhu na výstupu druhého integrátoru při různých parametrech pohybu v_0 , φ , f .



Obr. 5.12-3: Časový průběh rychlosti a dráhy tělesa při pohybu po nakloněné rovině

5.13 Matematické kyvadlo

Hmotný bod o zanedbatelné hmotnosti m se pohybuje po svislé kružnici, realizované otočně uloženou tyčí konstantní délky l za působení tíhové síly F_G . Cílem je vyšetřit kmitání tohoto tzv. matematického kyvadla, tedy zobrazit úhlovou dráhu φ a úhlovou rychlost ω v závislosti na čase při proměnném parametru φ_0 . Zobrazíme také tzv. fázovou trajektorii, tedy závislost úhlové rychlosti na úhlové dráze při $\varphi_0 = \text{konst.}$



Obr. 5.13-1: Matematické kyvadlo

Definujeme základní veličiny

t [s]	– čas
m [kg]	– hmotnost
F_G, F_R [N]	– tíhová síla a síla reakce
g [ms^{-2}]	– gravitační zrychlení $g = 9,81 \text{ ms}^{-2}$
$\varphi(t)$ [rad]	– úhel natočení kyvadla
φ_0 [rad]	– počáteční úhel natočení kyvadla
$\omega(t)$ [rad s^{-1}]	– úhlová rychlost
$\varepsilon(t)$ [rad s^{-2}]	– úhlové zrychlení
l [m]	– délka kyvadla
a, a_t, a_n [ms^{-2}]	– zrychlení hmotného bodu, jeho tečná a normálová složka

Sestavíme pohybovou rovnici a rozepíšeme ji do os vhodného souřadného systému. Po zavedení reakčního účinku tyče F_R bude její pohybová rovnice

$$ma = F_R + F_G$$

Ve směru tečny a normály (viz obr. 5.13-1) platí

$$ma_n(t) = F_R - mg \cos \varphi(t)$$

$$ma_t(t) = -mg \sin \varphi(t)$$

Pohyb bodu po kružnici je v kinematice popsán základními veličinami: φ – úhlová dráha [rad], ω – úhlová rychlost [rad s^{-1}], ε – úhlové zrychlení [rad s^{-2}]. V krátkém časovém úseku Δt budeme předpokládat, že se bod pohybuje stálou rychlostí ω a urazí dráhu $\Delta\varphi$, pak platí

$$\omega = \frac{\Delta\varphi}{\Delta t}$$

Pokud bude v tomto krátkém časovém úseku na bod působit konstantní úhlové zrychlení ε dojde ke zvýšení rychlosti o $\Delta\omega$, pak

$$\varepsilon = \frac{\Delta\omega}{\Delta t} = \frac{\Delta^2\varphi}{\Delta t^2}$$

Jestliže budeme v uvedené rovnici zmenšovat časový úsek Δ , tedy provedeme-li limitní přechod, obdržíme diferenciální rovnici pro určení úhlové dráhy φ , kterou bod urazí při působení úhlového zrychlení ε

$$\varepsilon(t) = \frac{d^2\varphi(t)}{dt^2}$$

Jelikož normálové zrychlení $a_n = l\omega^2 = l(d\varphi/dt)^2$ a tečné zrychlení $a_t = l(d^2\varphi/dt^2)$

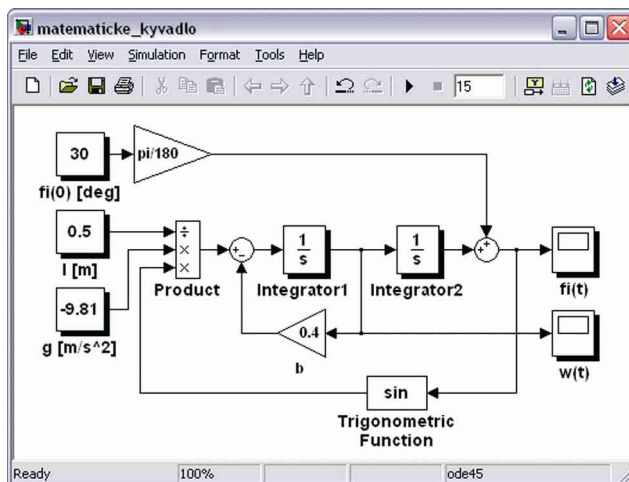
$$ml[\varphi'(t)]^2 = F_R - mg \cos \varphi(t)$$

$$ml\varphi''(t) = -mg \sin \varphi(t)$$

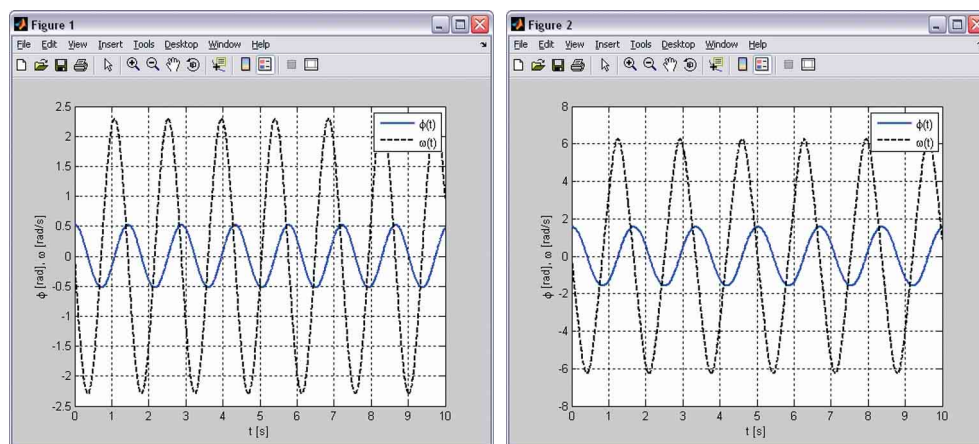
Z pohybové rovnice (pro směr tečny) vyjádříme úhlové zrychlení

$$\varepsilon(t) = \varphi''(t) = -\frac{g}{l} \sin \varphi(t)$$

Simulaci provedeme pro počáteční úhel $\varphi_0 = 30^\circ$, $\varphi_0 = 90^\circ$ a $\varphi_0 = 180^\circ$. Úhel φ_0 lze zadávat ve stupních, i když Simulink pracuje s úhly vyjádřenými v radiánech. Přepočtení je realizováno násobením konstantou $\pi/180$. V grafech (obr. 5.13-3) je však úhlová dráha vyjádřena v radiánech z důvodu přehledného zobrazení úhlové dráhy a úhlové rychlosti v jednom grafu. Délku kyvadla volíme $l = 0,5$ m.



Obr. 5.13-2: Simulace matematického kyvadla



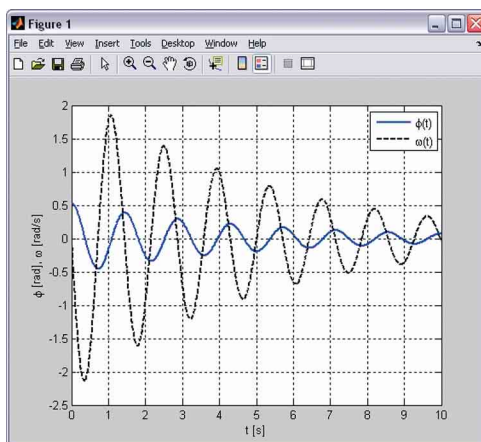
Obr. 5.13-3: Průběh $\varphi(t)$ a $\omega(t)$ při $\varphi_0 = 30^\circ$ a $\varphi_0 = 90^\circ$ bez tlumení (zde $\phi \equiv \varphi$)

Kmitání kyvadla probíhá pro nastavený počáteční úhel v čase $t = 0$, tj. φ_0 až do ustálení kyvadla do polohy, kdy je $\varphi = 0^\circ$. Uvedený model simuluje chování kyvadla ve vakuu (bez

působení tlumení). V případě, že chceme uvažovat koeficient tlumení b , musíme do pohybové rovnice doplnit tlumící sílu

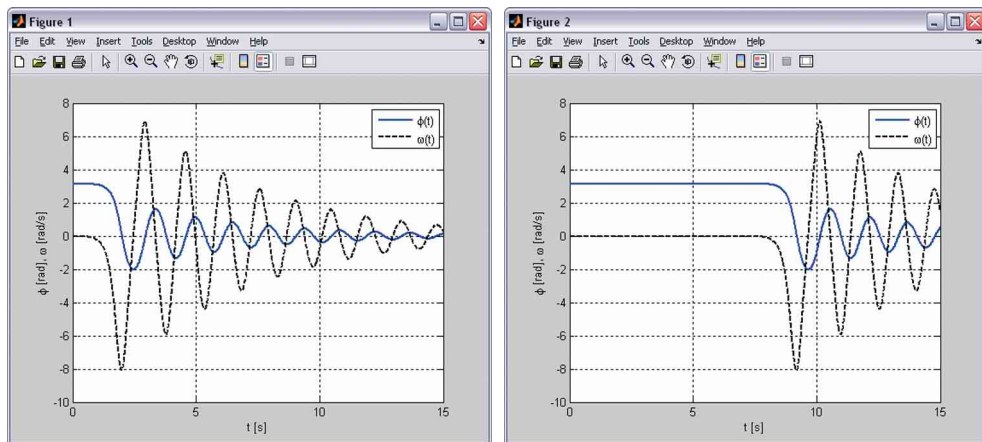
$$F_B = b \frac{d\varphi}{dt}$$

Tato síla působí proti rychlosti pohybu. Schéma doplníme o zápornou zpětnou vazbu z výstupu prvního integrátoru přes koeficient b na vstup tohoto integrátoru. Průběh $\varphi(t)$ a $\omega(t)$ kyvadla pro $b = 0,4 \text{ Ns/m}$ je uveden na obr. 5.13-4.



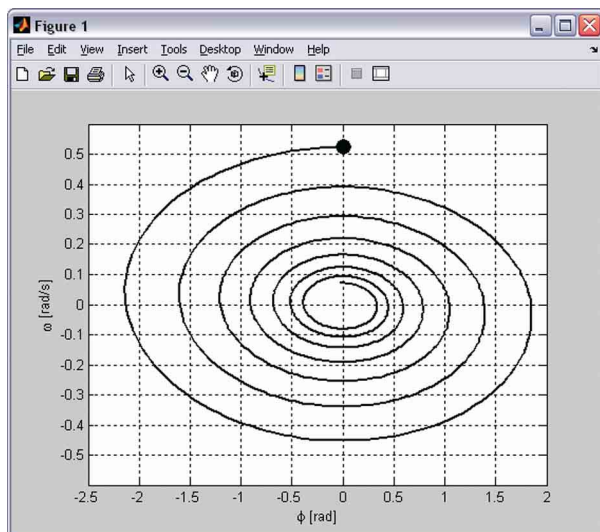
Obr. 5.13-4: Průběh $\varphi(t)$ a $\omega(t)$ při $\varphi_0 = 30^\circ$ s tlumením $b = 0,4 \text{ Ns/m}$ (zde $\phi \equiv \varphi$)

Pohyb kyvadla závisí na přesnosti zadané hodnoty π , což je nejvíce viditelné v případě, kdy $\varphi_0 = 180^\circ$. Experimentováním s nastavením hodnoty π různé přesnosti pak získáme různé průběhy pohybu kyvadla v závislosti na čase, viz obr. 5.13-5.



Obr. 5.13-5: Vliv přesnosti zadané hodnoty π při $\varphi_0 = 180^\circ$ (zde $\phi \equiv \varphi$)

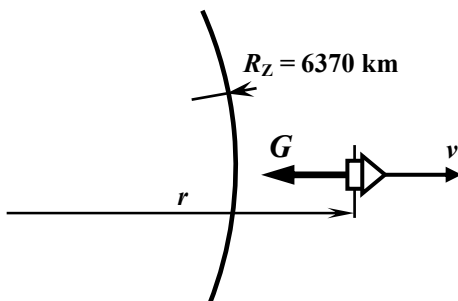
V prvním případě (graf na obr. 5.13-5 vlevo) bylo zadáno $\pi = 3,14$, v případě druhém (graf vpravo) bylo využito maximální přesnosti MATLABu, tedy $\pi = 3,14159265358979$. Zajímavý je také průběh úhlové rychlosti v závislosti na úhlové dráze (tzv. fázová resp. stavová trajektorie), viz obr. 5.13-6.



Obr. 5.13-6: Fázová trajektorie při $\varphi_0 = 30^\circ$ (zde $\phi \equiv \varphi$)

5.14 Let kosmické lodi

Ve větších vzdálenostech od povrchu Země nelze již její gravitační pole považovat za homogenní, tj. tíhové zrychlení g se s rostoucí vzdáleností rychle zmenšuje. V kosmonautice je ale důležitý zejména pohyb v radiálním gravitačním zemském poli. Tělesu (raketě) je udělena ve značné vzdálenosti od Země rychlost v_0 , ve směru kolmém k zemskému povrchu. Cílem úlohy je prozkoumat dráhy kosmických lodí, kterým by poslední díl nosné rakety udělil ve výši 100 km nad Zemí rychlost v intervalu 8 až 15 km s^{-1} .



Obr. 5.14-1: Poloha kosmické lodi vůči Zemi

Definujme základní veličiny

t [s]	– čas
m, M_Z [kg]	– hmotnost kosmické lodi, hmotnost Země
r, R_Z [m]	– vzdálenost lodi od středu Země, poloměr Země
h [m]	– vzdálenost lodi od povrchu Země
r_0 [m]	– počáteční vzdálenost lodi od středu Země, $r_0 = R_Z + h$
G [N]	– gravitační síla
g [ms^{-2}]	– gravitační zrychlení $g = 9,81 \text{ ms}^{-2}$
κ [$\text{kg}^{-1} \text{m}^3 \text{s}^{-2}$]	– gravitační konstanta
v [ms^{-1}]	– rychlost kosmické lodi
v_0 [ms^{-1}]	– počáteční rychlost kosmické lodi
a [ms^{-2}]	– zrychlení kosmické lodi

Pro pohyb kosmické lodi platí podle II. Newtonova zákona pohybová rovnice

$$ma = -G$$

a podle gravitačního zákona je

$$G = \kappa \frac{mM_Z}{r^2}$$

Zrychlení kosmické lodi je dle definice druhou derivací vzdálenosti podle času, tj. $a = r''$. Dosazením do pohybové rovnice dostaneme

$$m \frac{d^2 r}{dt^2} = \kappa \frac{mM_Z}{r^2}$$

a po úpravě

$$r^2 \frac{d^2 r}{dt^2} + \kappa M_Z = 0$$

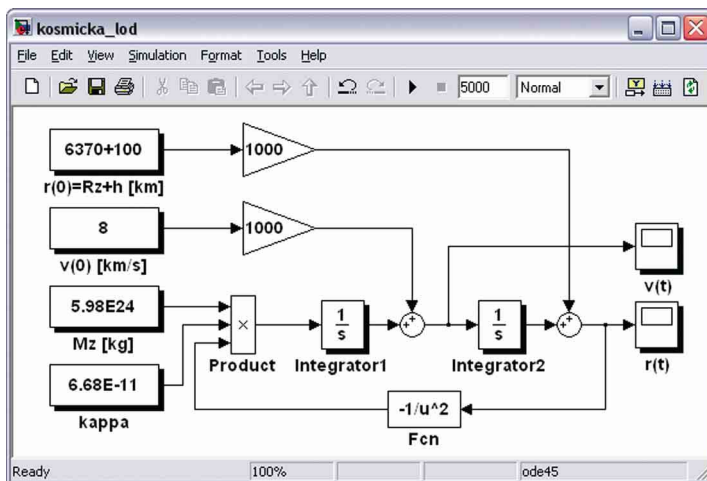
Tato diferenciální rovnice druhého řádu je nelineární a jejím řešením je hledaná vzdálenost rakety $r(t)$ od středu Země. Dráha letu nezávisí na hmotnosti kosmické lodi.

Vyjádříme nejvyšší derivaci r

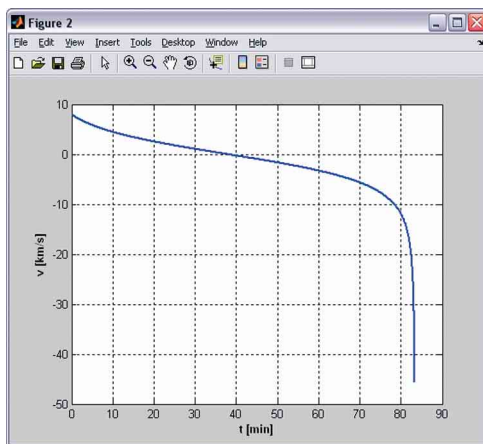
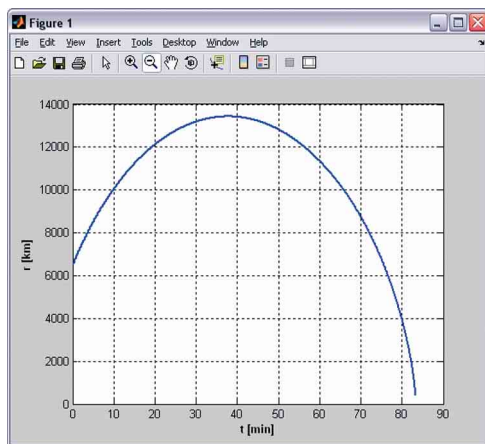
$$\frac{d^2 r}{dt^2} = -\frac{\kappa M_Z}{r^2}$$

Simulaci provedeme pro $M_Z = 5,98 \cdot 10^{24} \text{ kg}$, $R_Z = 6370 \text{ km}$, $\kappa = 6,68 \cdot 10^{-11} \text{ kg}^{-1} \text{m}^3 \text{s}^{-2}$ a $v_0 = 8$ až 15 km s^{-1} . Model (obr. 5.14-2) umožňuje sledovat rychlost $v(t)$ kosmické lodi na výstupu prvního integrátoru a vzdálenost $r(t)$ od středu Země na výstupu druhého integrátoru v závislosti na čase. Parametrem je počáteční rychlost v_0 . Závislost dráhy na čase není parabolická, neboť tíhové zrychlení se s rostoucí vzdáleností rychle zmenšuje. Lze se o tom přesvědčit, zobrazíme-li současně $r(t)$, $v(t)$ a $a(t)$. Je-li počáteční rychlost větší, než 12 km s^{-1} , kosmická loď se již nevrátí a trvale se od Země vzdaluje. Zajímavé je též sledovat rychlost rakety v závislosti na vzdálenosti od Země. Pokud je počáteční rychlost menší, než tzv.

úniková, rychlost lodi zpočátku letu klesá rychle, potom pozvolněji až k nule. To je vrchol dráhy letu. Potom rychlost změní znaménko a postupně se zvětšuje, loď se vrací k Zemi. Při počáteční rychlosti větší než únikové klesá $v(t)$ k určité mezní hodnotě.



Obr. 5.14-2: Simulace dráhy letu kosmické lodi


Obr. 5.14-3: Závislost vzdálenosti od středu Země a rychlosti kosmické lodi na čase při $v_0 = 8 \text{ km s}^{-1}$

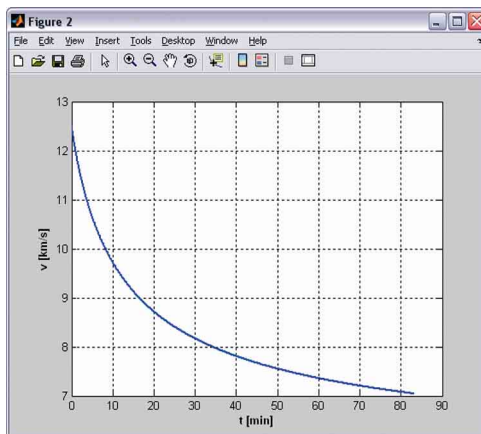
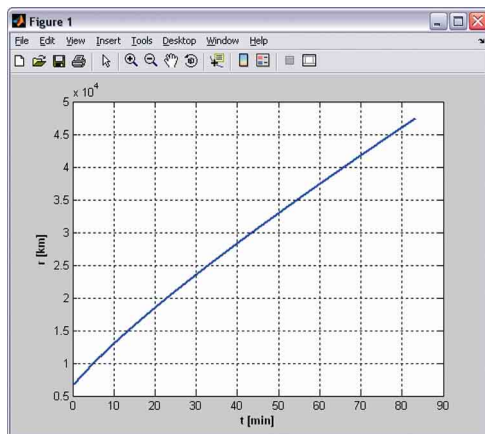
Aby těleso ve výšce h obíhalo kolem Země jako družice po kruhové dráze, musí být odstředivá síla v rovnováze s gravitační silou Země. Tedy

$$\frac{v_1^2}{R_z + h} = \frac{\kappa M_Z}{(R_z + h)^2} \quad \Rightarrow \quad v_1 = \sqrt{\frac{\kappa M_Z}{R_z + h}}$$

Pro $h \rightarrow 0$, po dosazení za $M_Z = 5,98 \cdot 10^{24} \text{ kg}$ a $R_z = 6,37 \cdot 10^6 \text{ m}$, je $v_1 = 7,9 \text{ km s}^{-1}$, tedy tzv. I. kosmická rychlost, k ní náleží doba oběhu $T = (2\pi R_z)/v_1 \approx 5066 \text{ s} = 84,4 \text{ minut}$.

Je-li tělesu ve výšce h nad zemským povrchem udělena počáteční rychlost $v_0 > v_1$, pak se pohybuje po elipse a při hodnotě $v_0 = v_2$ přejde na parabolickou dráhu a těleso se trvale vzdaluje od Země tzv. únikovou – II. kosmickou rychlostí, pro kterou platí

$$v_2 = \sqrt{\frac{2\kappa M_Z}{R_Z}} = v_1 \sqrt{2} \doteq 11,2 \text{ km s}^{-1}$$



Obr. 5.14-4: Závislost vzdálenosti od středu Země a rychlosti kosmické lodi na čase při $v_0 = 12,5 \text{ km s}^{-1}$

5.15 Ohyb jednostranně vetknutého nosníku

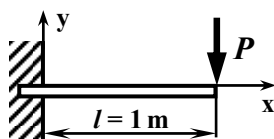
Pro tzv. ohybovou čáru nosníků platí diferenciální rovnice druhého řádu

$$\frac{d^2 y}{dx^2} = -\frac{M}{EJ}$$

kde

- y [m] – průhyb nosníku
- M [Nm] – ohybový moment
- E [Pa] – modul pružnosti
- J [kg m²] – moment setrvačnosti profilu nosníku

Cílem úlohy je vyšetřit průhyb, úhel sklonu ohybové čáry a ohybový moment krakorcového nosníku ve směru osy x při zatížení na volném konci silou P při zadaných parametrech J a E .



Obr. 5.15-1: Krakorcový nosník zatížený silou P

Definujme základní veličiny

- l [m] – délka nosníku
- P [N] – síla působící na volný konec nosníku
- x [m] – délková souřadnice nosníku ($x = 0$ v místě vetknutí)
- $y(x)$ [m] – průhyb nosníku v místě x

Nejprve je nutné vyjádřit průběh ohybového momentu. Počítáme-li tento moment od konce nosníku

$$M = -P(l - x)$$

Uvedený vztah dosadíme do rovnice pro ohybovou čáru

$$\frac{d^2 y}{dx^2} = \frac{P}{EJ}(l - x)$$

Rovnici dále derivujeme podle x

$$\frac{d^3 y}{dx^3} = -\frac{P}{EJ}$$

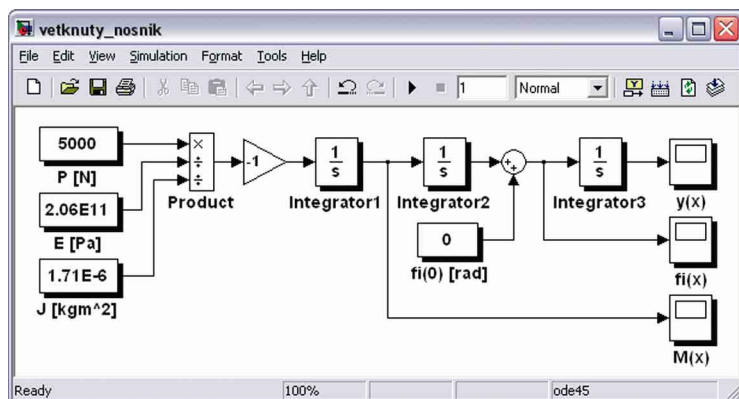
a výsledný vztah postupně integrujeme, tedy

$$\frac{d^2 y}{dx^2} = -\int_0^x \frac{P}{EJ} dx$$

$$\frac{dy}{dx} = \tan \varphi \approx \varphi + \varphi_0 \quad (\text{platí pro malé úhly a prohnutí})$$

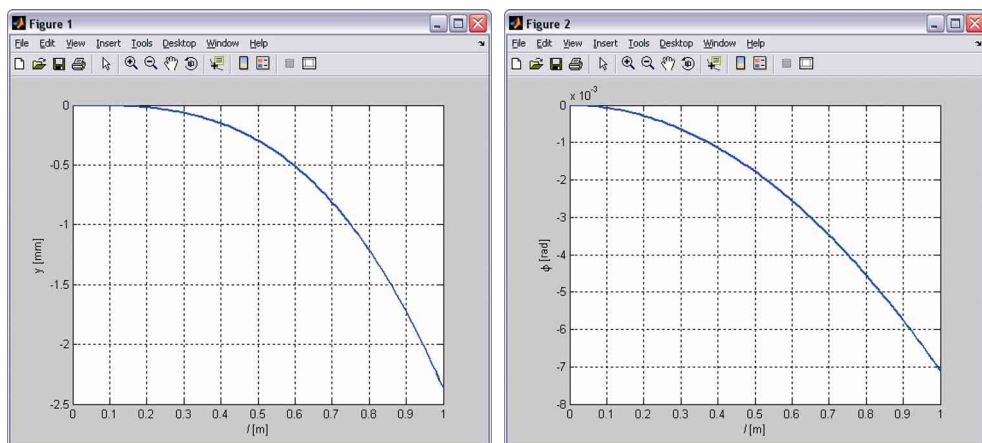
$$y(x) = \int_0^x \varphi dx$$

Simulaci provedeme pro tyto hodnoty: $P = 5$ kN, $J = 1,71 \cdot 10^{-6}$ kg m², $E = 2,06 \cdot 10^{11}$ Pa a $\varphi_0 = 0$ resp. $\varphi_0 = -4 \cdot 10^{-3}$ rad. Úhel φ_0 představuje úhel vetknutí nosníku.

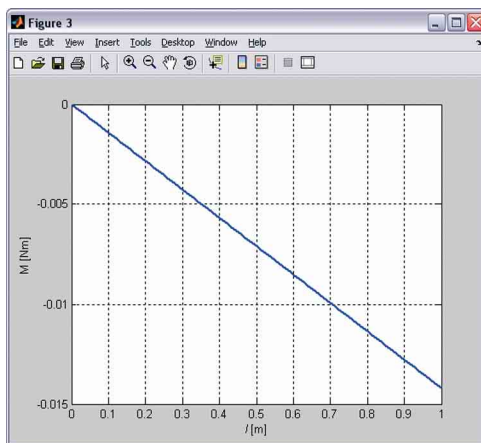


Obr. 5.15-2: Simulace průhybu nosníku

Model (obr. 5.15-2) umožňuje sledovat na výstupu prvního integrátoru časový průběh ohybového momentu nosníku, na výstupu druhého integrátoru úhel sklonu ohybové čáry a na výstupu třetího integrátoru průběh ohybové čáry podél souřadnice x nosníku. V modelu lze měnit parametry φ_0 (úhel vetknutí nosníku), P , J a E .

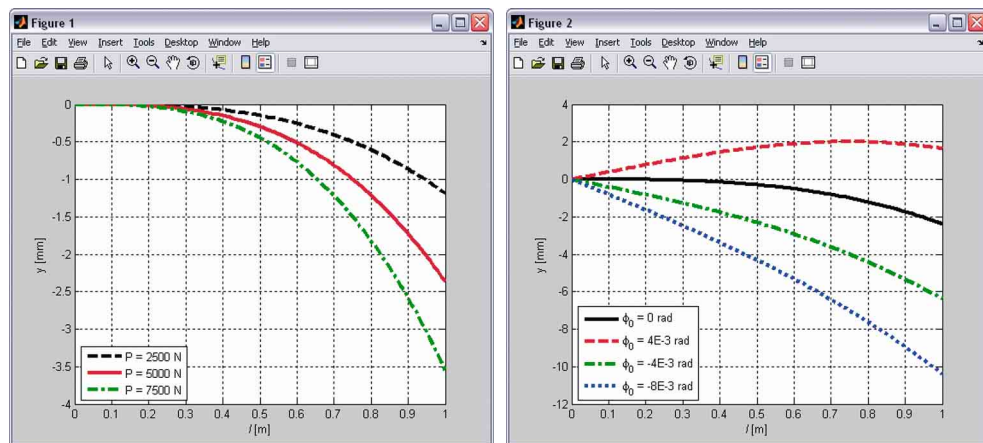


Obr. 5.15-3: Průhyb a úhel (zde $\phi \equiv \varphi$) nosníku v závislosti na souřadnici x



Obr. 5.15-4: Ohybový moment v závislosti na souřadnici x

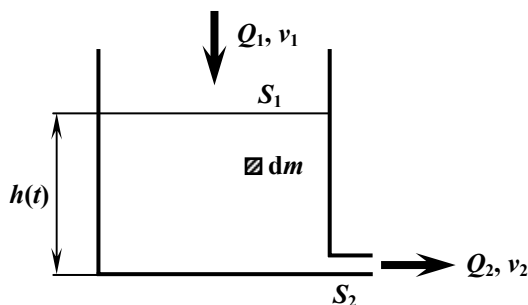
Zajímavé jsou též průhyby podél nosníku pro různé úhly vetknutí a průhyb pro různé zatěžovací síly, viz obr. 5.15-5.



Obr. 5.15-5: Průhyb nosníku různé zatěžovací síly a pro různé úhly vetknutí (zde $\phi_0 \equiv \phi_0$)

5.16 Vývoj výšky hladiny v nádrži

Úkolem je simulovat vývoj hladiny v nádrži na obr. 5.16-1, do které napouštíme vodu definovanou rychlostí a která má u dna otvor. Uvažujeme ideální kapalinu a homogenní gravitační zrychlení.



Obr. 5.16-1: Nádrž s výtakovým otvorem u dna

Definujeme základní veličiny

t [s]	– čas
Q_1, Q_2 [$\text{m}^3 \text{s}^{-1}$]	– přítok do nádrže, odtok z nádrže otvorem
v_1, v_2 [ms^{-1}]	– rychlost přitékající a rychlost odtékající kapaliny
S_1, S_2 [m^2]	– plocha půdorysu nádrže a plocha výtakového otvoru
m [kg]	– hmotnost kapaliny
V [m^3]	– objem kapaliny
g [ms^{-2}]	– gravitační zrychlení, $g = 9,81 \text{ ms}^{-2}$
$h(t)$ [m]	– výška hladiny v čase t

Ze zákona o zachování energie nejprve odvodíme výtokovou rychlost v_2 . Tato rychlost samozřejmě není konstantní, neboť závisí na aktuálním množství vody v nádrži. Uvažujeme-li malý element kapaliny o hmotnosti dm , který má na hladině nulovou rychlost a na dně při výtoku má nulovou potenciální energii, platí, že

$$\frac{1}{2} dm v_2(t)^2 = dm g h(t)$$

a tedy

$$v_2(t) = \sqrt{2 g h(t)}$$

Celkový objemový tok při změně výšky hladiny je

$$Q(t) = Q_1(t) - Q_2(t) = \frac{dV}{dt}$$

Platí-li dále, že

$$dV = S_1 \frac{dh(t)}{dt} = S_1 h'(t)$$

$$Q_2(t) = S_2 v_2(t)$$

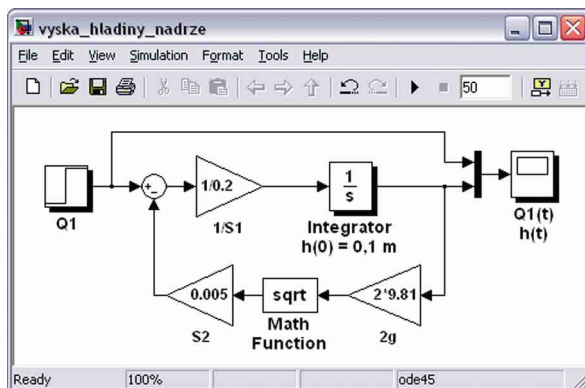
Ize po dosazení do rovnice pro objemový tok psát

$$S_1 h'(t) = Q_1(t) - S_2 v_2(t)$$

Dosadíme-li do uvedených rovnic dále za rychlost v_2 , tak po jednoduché úpravě obdržíme výslednou diferenciální rovnici

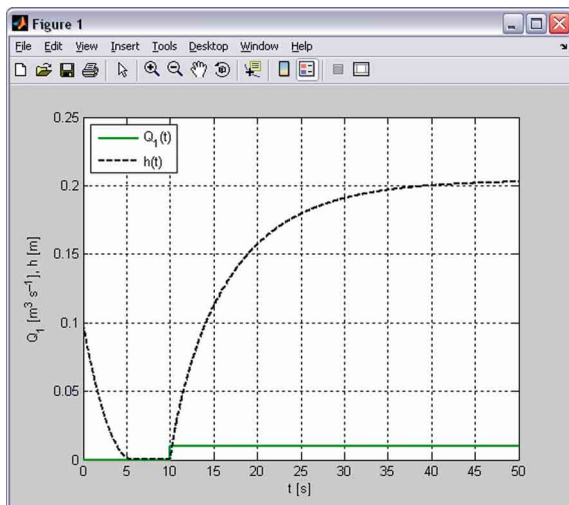
$$h'(t) = \frac{1}{S_1} [Q_1(t) - S_2 \sqrt{2 g h(t)}]$$

Simulaci provedeme pro tyto hodnoty: $Q_1 = 0,01 \text{ m}^3 \text{ s}^{-1}$, $S_1 = 0,2 \text{ m}^2$ a $S_2 = 0,005 \text{ m}^2$. Příslušný simulační model je na obr. 5.16-2. Při simulaci je možné zvolit počáteční výšku hladiny a libovolný přítok, případně ponechat přítok vypnutý a sledovat průběh vypouštění nádrže.



Obr. 5.16-2: Simulace vývoje výšky hladiny v nádrži

Výsledky simulace při počáteční výšce hladiny $h_0 = 0,1$ m a přítoku, který je nejprve nulový a v čase $t = 10$ s je skokově změněn na $Q_1 = 0,01 \text{ m}^3 \text{ s}^{-1}$ jsou na obr. 5.16-3. V grafu je zobrazen mimo průběhu simulované výšky hladiny $h(t)$ také popisovaný průběh přítoku $Q_1(t)$.



Obr. 5.16-3: Závislost výšky hladiny na čase

Pro kontrolu správnosti výsledku simulace je vhodné výpočtem stanovit ustálenou výšku hladiny $h(\infty)$ při konstantním přítoku Q_1 . V tomto případě by se měl přítok rovnat odtoku, tedy pro $t \rightarrow \infty$, $h'(\infty) \rightarrow 0$

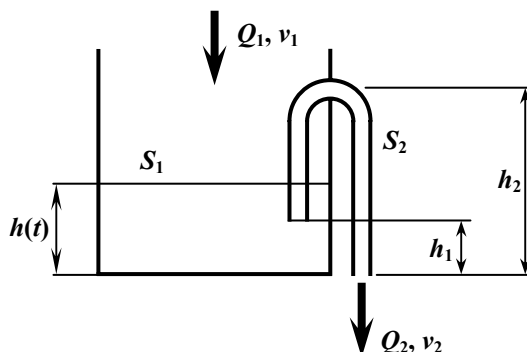
$$h(\infty) = \frac{1}{2g} \left(\frac{Q_1(\infty)}{S_2} \right)^2 = \frac{1}{2 \cdot 9,81} \left(\frac{0,01}{0,005} \right)^2 = 0,2039 \text{ m}$$

V grafu na obr. 5.16-3 se snadno přesvědčíme, že ustálená hodnota výšky hladiny je v souladu s hodnotou $h(\infty)$ vypočtenou výše. Ustálená hodnota není závislá na ploše půdorysu nádrže S_1 , na velikosti této plochy závisí pouze rychlost ustálení. V modelu je uvažováno, že kapalina vytéká vždy celou plochou S_2 , přestože nemusí být při úplném vypouštění v nádrži dostatečné množství kapaliny. Hladina nádrže se v tomto případě ustálí na kladné hodnotě blízko nuly, model není v tomto ohledu zcela dokonalý.

Modifikujme nyní úlohu tak, že namísto otvorem u dna bude nádrž vypouštěna **pomocí hadice**, viz obr. 5.16-4. Konec hadice, který je uvnitř nádrže, je ve výšce h_1 od dna, hadice je ohnuta ve výšce h_2 a druhý konec hadice, kterým kapalina vytéká, je na úrovni dna.

Vzhledem k okolnosti, že kapalina vytéká z hadice na úrovni dna nádrže, není nutné měnit vzorec pro výtokové rychlosti v_2 . Musíme, ale v rámci modelu ošetřit stavy, kdy kapalina vytéká a kdy ne. Stanovme následující pravidla:

- kapalina začne vytékát v okamžiku, kdy výška hladiny překročí výšku h_2 ,
- kapalina vytékát přestane v okamžiku, kdy výška hladiny klesne pod úroveň h_1 .



Obr. 5.16-4: Vypouštění nádrže hadicí

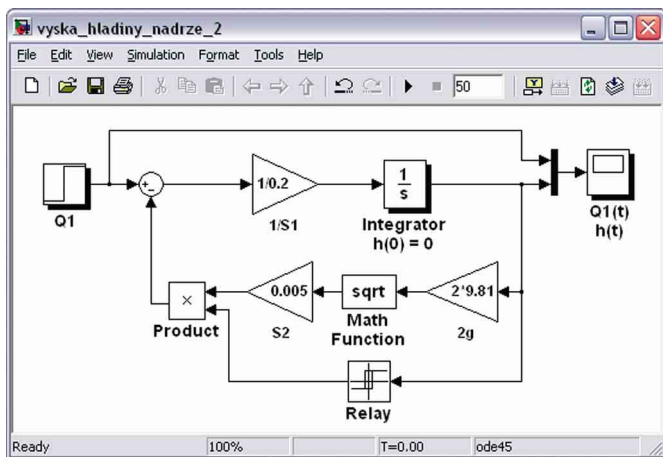
Zápis uvedených podmínek pomocí logické proměnné $L(t)$ může být např. následující

$$h(t) = \frac{1}{S_1} [Q_1(t) - L(t) S_2 \sqrt{2gh(t)}]$$

$$h(t) > h_2 \rightarrow L(t) = 1$$

$$h(t) > h_1 \rightarrow L(t) = 0$$

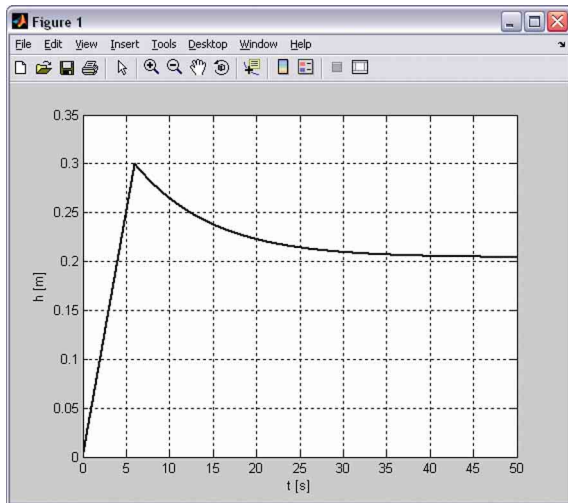
Simulační schéma na obr. 5.16-2 musíme upravit tak, aby zpětnovazební smyčka byla aktivní pouze tehdy, když kapalina vytéká hadicí z nádrže. Nejjednodušší je zřejmě použití bloku *Relay*. V bloku je nutné nastavit hodnoty, při kterých dochází k sepnutí (h_2) a vypnutí (h_1) a hodnotu při sepnutém (1) resp. vypnutém stavu (0). Upravené simulační schéma je na obr. 5.16-5.



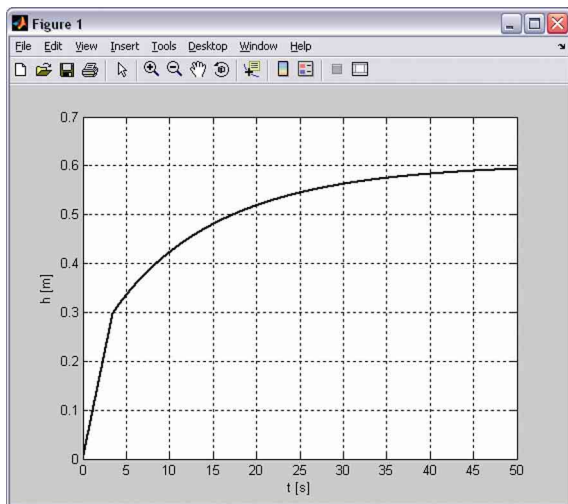
Obr. 5.16-5: Upravené simulační schéma

S modelem je možné značně experimentovat, lze měnit např. přítok $Q_1(t)$, výšky h_1 a h_2 . Mohou nastat následující základní tři eventuality:

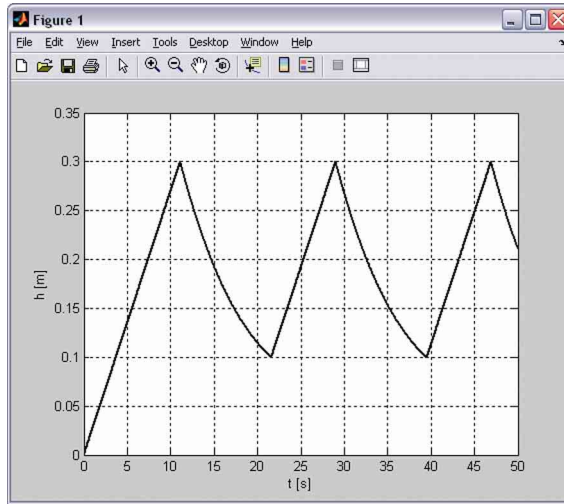
- $h_1 \leq h(\infty) < h_2$, tj. situace na obr. 5.16-4 – v okamžiku, kdy se nádrž se naplní nad úroveň h_2 , začne kapalina vytékat a ustálí se na hodnotě $h(\infty)$, viz obr. 5.16-6,
- $h(\infty) > h_2$ – nádrž se naplní nad úroveň h_2 a výška hladiny zůstane nad touto úrovní, kapalina bude neustále vytékat, viz obr. 5.16-7,
- $h(\infty) > h_1$ – jestliže se nádrž naplní nad úroveň h_2 , začne kapalina vytékat až do okamžiku, kdy hladina klesne pod h_1 , v tom okamžiku se nádrž přestane vypouštět a hladina začne stoupat opět až na úroveň h_2 a následně začne opět klesat, tento jev se bude periodicky opakovat a hladina se nikdy neustálí, viz obr. 5.16-8.



Obr. 5.16-6: Průběh $h(t)$ při $Q_1 = 0,01 \text{ m}^3 \text{ s}^{-1}$

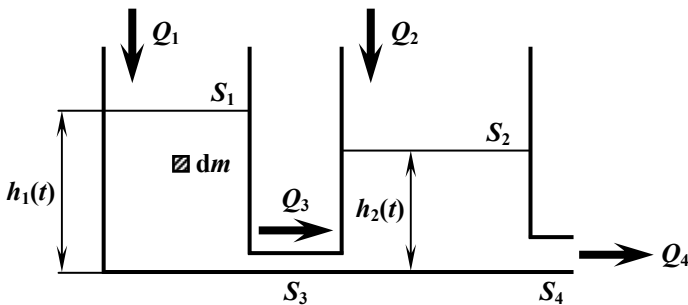


Obr. 5.16-7: Průběh $h(t)$ při $Q_1 = 0,0167 \text{ m}^3 \text{ s}^{-1}$


Obr. 5.16-8: Průběh $h(t)$ při $Q_1 = 0,0054 \text{ m}^3 \text{ s}^{-1}$

5.17 Vývoj výšky hladin dvou spojených nádrží

Uvažujme dvě navzájem spojené nádrže, viz obr. 5.17-1. Každá nádrž má příslušné rozměry (navzájem odlišné) a svůj vlastní přítok. Voda z první nádrže může vytékat do nádrže druhé, voda z druhé nádrže může téci zpět do první nádrže nebo odtékat ven. Ztráty ve vedeních zanedbáme. Zajímá nás vývoj výšky hladiny v obou nádržích.



Obr. 5.17-1: Spojené nádrže

Definujme základní veličiny

$t \text{ [s]}$	– čas
$Q_1, Q_3 \text{ [m}^3 \text{ s}^{-1}\text{]}$	– přítok do první nádrže a odtok ven z nádrže otvorem
$Q_2, Q_4 \text{ [m}^3 \text{ s}^{-1}\text{]}$	– přítok do druhé nádrže a odtok ven z nádrže otvorem
$v_1, v_2, v_3, v_4 \text{ [ms}^{-1}\text{]}$	– rychlosti přitékající a rychlosti odtékající kapaliny
$S_1, S_3 \text{ [m}^2\text{]}$	– plocha půdorysu první nádrže a jejího výtokového otvoru

S_2, S_4 [m ²]	– plocha půdorysu druhé nádrže a jejího výtokového otvoru
m [kg]	– hmotnost kapaliny
V_1, V_2 [m ³]	– objem kapaliny v první resp. druhé nádrži
g [ms ⁻²]	– gravitační zrychlení, $g = 9,81 \text{ ms}^{-2}$
$h_1(t)$ [m]	– výška hladiny v první nádrži v čase t
$h_2(t)$ [m]	– výška hladiny v druhé nádrži v čase t

Jelikož se jedná o dva navzájem propojené systémy, musíme jejich chování popsat pomocí dvou diferenciálních rovnic. Uvažujeme-li, že přítok (resp. odtok) lze vyjádřit jako derivaci objemu podle času, obdržíme následující dvě rovnice

$$\frac{dV_1(t)}{dt} = S_1 h_1'(t) = Q_1(t) - Q_3(t) = Q_1(t) - S_3 v_3(t)$$

$$\frac{dV_2(t)}{dt} = S_2 h_2'(t) = Q_2(t) + Q_3(t) - Q_4(t) = Q_2(t) + S_3 v_3(t) - S_4 v_4(t)$$

Rychlost $v_4(t)$ odvodíme ze zákona o zachování energie (viz také příklad 5.16)

$$v_4(t) = \sqrt{2g h_2(t)}$$

Je zřejmé, že rychlost $v_3(t)$ je ovlivněna rozdílem tlaků u dna nádob a je tedy možné psát

$$\frac{1}{2} dm v_3(t) = dm g [h_1(t) - h_2(t)]$$

$$v_3(t) = \sqrt{2g [h_1(t) - h_2(t)]}$$

V okamžiku, kdy ve druhé nádrži bude vyšší hladina než v nádrži první, objeví se pod odmocninou ve výše uvedeném vztahu pro výpočet rychlosti záporné číslo. Tuto situaci je nutné ošetřit. Odmocninu lze počítat z absolutní hodnoty rozdílu $|h_1(t) - h_2(t)|$ a výsledek pak vynásobit funkcí *signum*

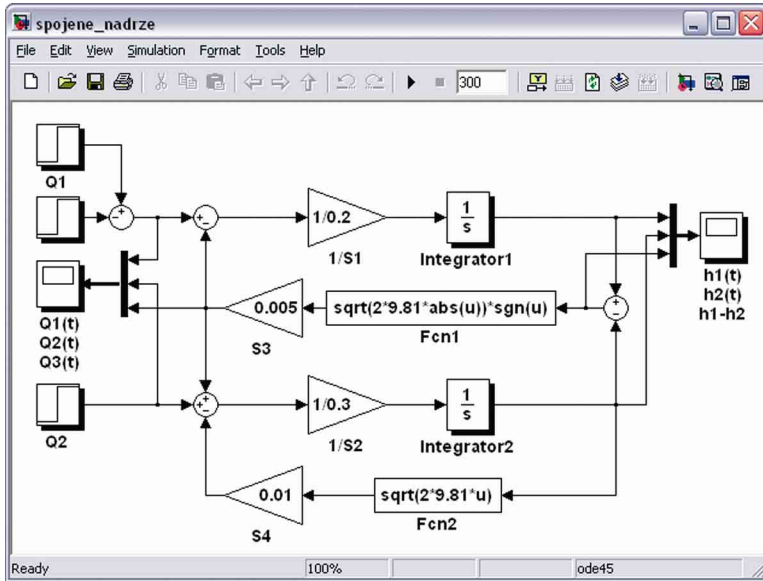
$$v_3(t) = \sqrt{2g |h_1(t) - h_2(t)|} \cdot \text{sign}[h_1(t) - h_2(t)]$$

Výsledné diferenciální rovnice získáme dosazením za rychlosti $v_3(t)$ a $v_4(t)$ do základních rovnic

$$h_1'(t) = \frac{1}{S_1} \{Q_1(t) - Q_1(t) - S_3 \sqrt{2g |h_1(t) - h_2(t)|} \cdot \text{sign}[h_1(t) - h_2(t)]\}$$

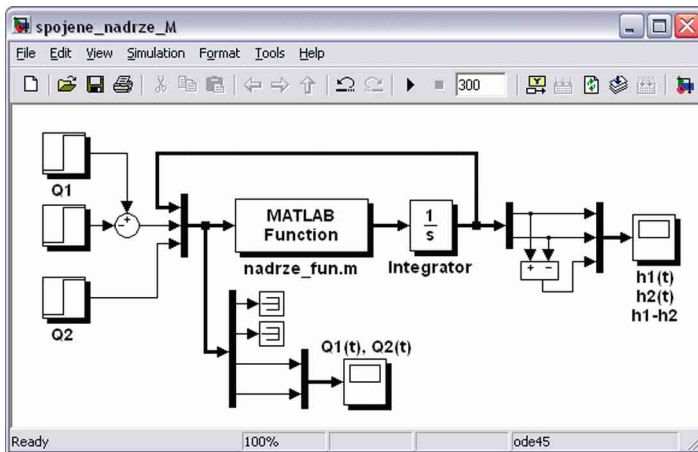
$$h_2'(t) = \frac{1}{S_2} \{Q_2(t) + S_3 \sqrt{2g |h_1(t) - h_2(t)|} \cdot \text{sign}[h_1(t) - h_2(t)] - S_4 \sqrt{2g h_2(t)}\}$$

Simulaci provedeme pro tyto hodnoty: $Q_1 = 0,01 \text{ m}^3 \text{ s}^{-1}$, $Q_2 = 0,02 \text{ m}^3 \text{ s}^{-1}$, $S_1 = 0,2 \text{ m}^2$, $S_2 = 0,3 \text{ m}^2$, $S_3 = 0,005 \text{ m}^2$ a $S_4 = 0,01 \text{ m}^2$. Jedna z možných realizací simulačního modelu je na obr. 5.17-2. Pro výpočet průtoku $Q_3(t)$ resp. $Q_4(t)$ jsou použity bloky *Fcn*. Vztah je také možné sestavit ze standardních bloků Simulinku; pro výpočet absolutní hodnoty použijeme blok *Abs*, odmocninu vypočteme pomocí bloku *Math Function* (zvolíme v něm odmocninu, tedy **sqrt**), *signum* pak pomocí bloku *Sign*.



Obr. 5.17-2: Simulace vývoje výšky hladin dvou spojených nádrží

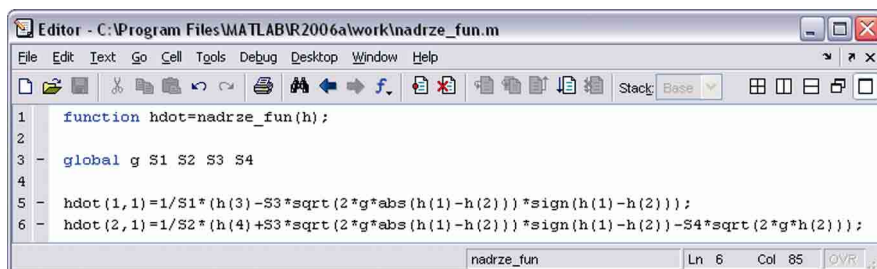
Řešení diferenciálních rovnic lze také realizovat v bloku *MATLAB Function*. V rámci tohoto bloku vypočítáváme derivace obou výšek hladin, tj. $h_1'(t)$, $h_2'(t)$ a za blok zařadíme integrátor, viz obr. 5.17-3. Zápis rovnic pro výpočet $h_1'(t)$, $h_2'(t)$ je uveden na obr. 5.17-4, příklad možného skriptu pro řízení výpočtu, v jehož rámci jsou zadávány konkrétní hodnoty, je spouštěna simulace a jsou zobrazeny i výsledky, je na obr. 5.17-5.



Obr. 5.17-3: Realizace pomocí MATLAB Function

Při zápisu funkce (obr. 5.17-4) je na prvním řádku m-souboru vždy deklarace funkce, která musí obsahovat slovo **function**. Vstupní proměnné (zde pouze h) obsahují data předaná

funkci a výstupní proměnné (zde hdot) obsahují data funkcí vrácená. Všechny další řádky funkce tvoří tělo funkce. Tělo obsahuje příkazy MATLABu, které zpracovávají vstupní argumenty a výsledky ukládají do výstupních argumentů funkce. Běh funkce končí poté, co je zpracována poslední řádka m-souboru.



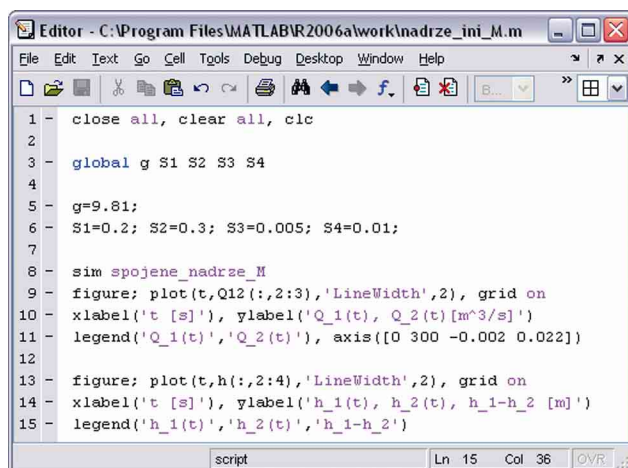
```

1 function hdot=nadrze_fun(h);
2
3 global g S1 S2 S3 S4
4
5 hdot(1,1)=1/S1*(h(3)-S3*sqrt(2*g*abs(h(1)-h(2))))*sign(h(1)-h(2));
6 hdot(2,1)=1/S2*(h(4)+S3*sqrt(2*g*abs(h(1)-h(2))))*sign(h(1)-h(2))-S4*sqrt(2*g*h(2));

```

Obr. 5.17-4: Definování funkce

Jakákoliv funkce definovaná v m-souboru pracuje s proměnnými v přiděleném lokálním prostoru, který je vždy oddělen od prostoru jiné funkce a také od základního prostoru MATLABu. Funkce mohou ale sdílet proměnné s jinými funkcemi, základním pracovním prostorem a rekurzivně volanými funkcemi přes proměnné, které jsou deklarovány jako globální. Je nutné, aby sdílená proměnná byla definována v každém požadovaném pracovním prostoru. Při deklaraci se globální proměnné uvozují slovem **global**. Deklarace se provede obvykle na začátku skriptu. Na obr. 5.17-4 jsou takto definovány proměnné g, S1, S2, S3 a S4. Deklaraci je nutné provést nejen přímo ve vytvářené funkci, ale také i v řídicím skriptu (obr. 5.17-5), kde jsou proměnné inicializovány (tj. jsou jim přiřazeny konkrétní hodnoty).



```

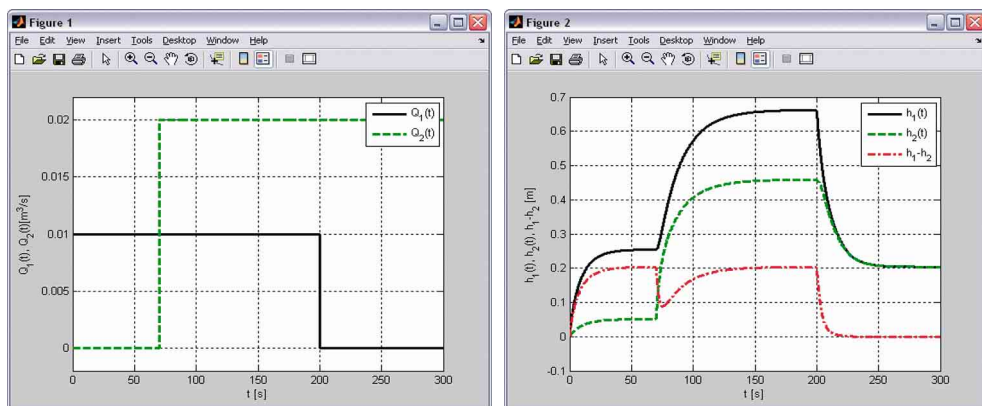
1 close all, clear all, clc
2
3 global g S1 S2 S3 S4
4
5 g=9.81;
6 S1=0.2; S2=0.3; S3=0.005; S4=0.01;
7
8 sim_spojene_nadrze_M
9 figure; plot(t,Q12(:,2:3),'LineWidth',2), grid on
10 xlabel('t [s]'), ylabel('Q_1(t), Q_2(t) [m^3/s]')
11 legend('Q_1(t)', 'Q_2(t)'), axis([0 300 -0.002 0.022])
12
13 figure; plot(t,h(:,2:4),'LineWidth',2), grid on
14 xlabel('t [s]'), ylabel('h_1(t), h_2(t), h_1-h_2 [m]')
15 legend('h_1(t)', 'h_2(t)', 'h_1-h_2')

```

Obr. 5.17-5: Příklad řídicího skriptu

Jak již bylo řečeno, výstupním argumentem popisované funkce je proměnná hdot, tedy vektor derivací výšek hladin obou nádrží. Po integraci je nutné zavést tento vektor zpět na vstup bloku *MATLAB Function*. V bloku je třeba také definovat kromě názvu funkce, tedy nadrze_fun.m, také v položce **Output Dimensions** rozměr výstupu, v našem případě 2.

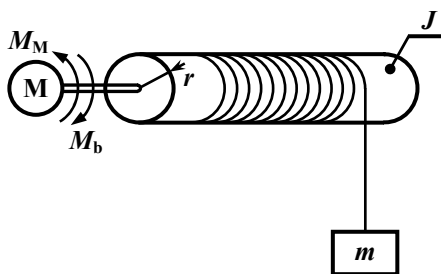
Ve funkci přistupujeme k prvkům vstupního vektoru (proměnná h) běžným způsobem, tj. pomocí kulatých závorek. Podrobněji je práce s funkcemi vysvětlena v [4]. Simulaci spustíme přímo ve skriptu pomocí příkazu **sim**. Obdržené výsledky lze samozřejmě v rámci řídicího skriptu také přehledně zobrazit, viz obr. 5.17-6.



Obr. 5.17-6: Zobrazení výsledků simulace

5.18 Navíjecí zařízení

Předpokládáme, že mechanická soustava (obr. 5.18-1) s rotačním pohybem se skládá ze stejnosměrného motoru s hnacím momentem M_M , který pohání navíjak s momentem setrvačnosti J_m . Navíjecí zařízení zvedá pomocí lana těleso o hmotnosti m . Cílem úlohy je simulovat chování navíjaku, tj. zobrazit závislosti úhlové rychlosti a úhlu natočení navíjecího válce na čase při zadaných parametrech J , b a zatěžovacím momentu M_Z .



Obr. 5.18-1: Schéma poháněného navíjaku

Definujme základní veličiny

m [kg]	– hmotnost tělesa
r [m]	– poloměr navíjecího válce
g [ms^{-2}]	– gravitační zrychlení, $g = 9,81 \text{ ms}^{-2}$
J_m [kg m^2]	– moment setrvačnosti navíjecího válce vzhledem k jeho ose

$J \text{ [kg m}^2\text{]}$	– celkový moment setrvačnosti navíjecího válce
$\varphi \text{ [rad]}$	– úhel natočení válce
$\omega = d\varphi/dt \text{ [rad s}^{-1}\text{]}$	– úhlová rychlost pohybu
$M_M, M_Z, M_b \text{ [Nm]}$	– hnací moment motoru, zatěžovací moment a tlumicí moment
$b_r \text{ [Nms]}$	– koeficient rotačního tlumení

Pohybovou rovnici sestavíme na základě momentové rovnováhy. Podle obr. 5.18-1 uvažujeme následující momenty:

- hnací moment motoru M_M ,
- tlumicí moment $M_b = b_r \frac{d\varphi(t)}{dt} = b_r \omega(t)$,
- zatěžovací moment $M_Z = m g r$,
- setrvačný moment $M_J = \frac{d\omega(t)}{dt} = \frac{d^2\varphi(t)}{dt^2}$.

Protože hmota m je zvedána vertikálním pohybem mimo osu rotace navíjecího válce, je podle tzv. Steinerovy věty celkový moment setrvačnosti soustavy roven

$$J = J_m + m r^2$$

kde J je moment setrvačnosti navíjecího válce vzhledem k ose procházející jeho těžištěm a r je vzdálenost osy v těžišti od osy rotace.

Rovnice momentové rovnováhy má tedy tvar

$$M_J = M_M - M_Z - M_b$$

po dosazení jednotlivých momentů

$$\frac{d^2\varphi(t)}{dt^2} = M_M - M_Z - b_r \frac{d\varphi(t)}{dt}$$

Po úpravě dostaneme

$$J \frac{d^2\varphi(t)}{dt^2} + b_r \frac{d\varphi(t)}{dt} = M_M - M_Z$$

Předpokládáme-li nulové počáteční podmínky, lze v Laplaceově transformaci odvodit tzv. obrazový přenos dynamického systému, který je definován jako poměr Laplaceova obrazu výstupu ku Laplaceově obrazu vstupu, viz také kapitola 3. S využitím základních pravidel transformace (příloha A) lze postupně psát

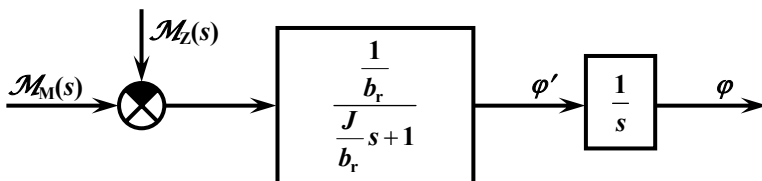
$$Js^2\Phi(s) + b_r s\Phi(s) = \mathcal{M}_M(s) - \mathcal{M}_Z(s)$$

$$s\Phi(s)[Js + b_r] = \mathcal{M}_M(s) - \mathcal{M}_Z(s)$$

$$\Phi(s) = \frac{\mathcal{M}_M(s) - \mathcal{M}_Z(s)}{s(Js + b_r)}$$

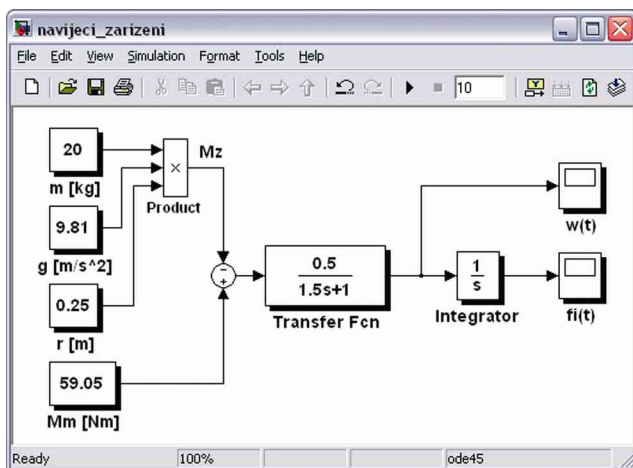
$$\Phi(s) = \frac{\frac{1}{b_r}}{s \left(\frac{J}{b_r} s + 1 \right)} [\mathcal{M}_M(s) - \mathcal{M}_Z(s)] = \frac{K}{s(Ts + 1)} [\mathcal{M}_M(s) - \mathcal{M}_Z(s)]$$

Ve výše uvedené rovnici lze označit $K = 1/b_r$ jako zesílení dynamického systému a $T = J/b_r$ jako jeho časovou konstantu. Na základě obdržené rovnice lze sestavit blokové schéma (obr. 5.18-2), které je obvyklé v teorii řízení



Obr. 5.18-2: Blokové schéma modelu rotační soustavy navijáku

Simulaci provedeme pro tyto hodnoty: $m = 20$ kg, $J_m = 1,75$ kg m², $r = 0,25$ m, $b_r = 2$ Nms a $M_M = 59,05$ Nm. Ze zadaných hodnot vypočteme zesílení systému $K = 1/b_r = 1/2 = 0,5$, celkový moment setrvačnosti dle Steinerovy věty $J = J_m + m r^2 = 1,75 + 20 \cdot 0,25^2 = 3$ kg m² a časovou konstantu systému $T = J/b_r = 3/2 = 1,5$ s. Simulační model vytvořený v prostředí Simulink je na obr. 5.18-3. Závislosti úhlové rychlosti a úhlu natočení navijáku na čase jsou na obr. 5.18-4.

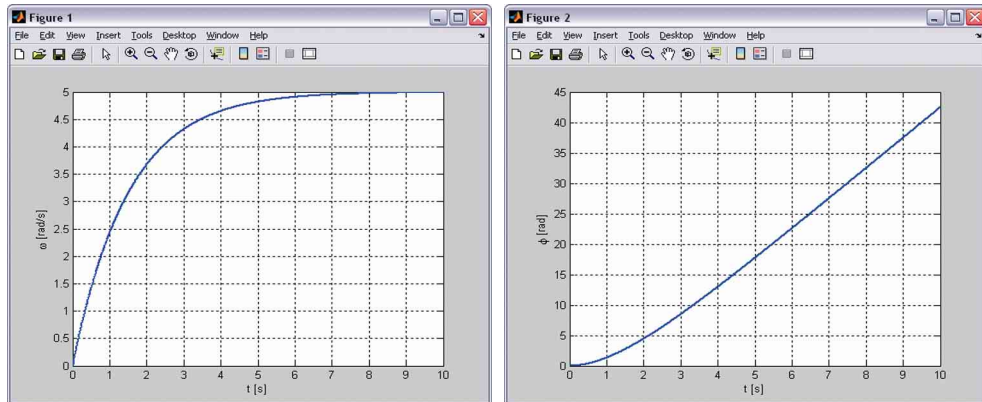


Obr. 5.18-3: Simulační model navijáku s břemenem

Z hlediska teorie řízení je popisovaná rotační soustava setrvačným blokem prvního řádu s přenosem

$$G(s) = \frac{K}{Ts + 1}$$

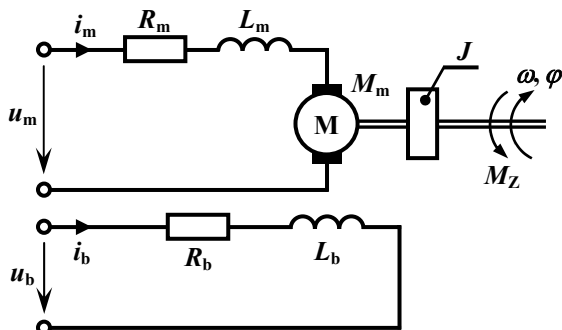
Do soustavy vstupuje obraz rozdílu momentů $\mathcal{M}_M(s) - \mathcal{M}_Z(s)$ a výstupem je $\omega(s) = s\varphi(s)$, tedy obraz úhlové rychlosti. Dalším blokem, zapojeným do série, je integrační blok, na jehož výstupu je obraz úhlu natočení $\varphi(s)$. Zde se samozřejmě nejedná o regulaci, proto chybí zpětná vazba. Změnou konstanty rotačního tlumení b_r měníme zesílení K soustavy, změnou b_r a J pak ovlivňujeme časovou konstantu T .



Obr. 5.18-4: Průběh úhlové rychlosti a úhlu natočení navijáku v závislosti na čase (zde $\phi \equiv \varphi$)

5.19 Stejnosměrný motor s cizím buzením

Stejnosměrné motory s cizím buzením se vyznačují dobrými regulačními vlastnostmi. Umožňují jednoduché řízení rychlosti změnou napětí kotvy popř. budicího proudu, přitom se otáčky mohou pohybovat v širokém rozsahu, který není nijak vázán na kmitočet sítě. Při řízení napětím kotvy se navíc jedná o v zásadě lineární prvek. Smysl otáčení lze snadno měnit změnou polarity napětí kotvy či budicího proudu. Výhodný je i velký točivý moment, zvláště při nízkých otáčkách. Značným problémem je ale napájení rotoru přes komutátor, v jehož důsledku je stejnosměrný motor relativně méně spolehlivý a s většími nároky na údržbu než např. motor asynchronní.



Obr. 5.19-1: Náhradní schéma stejnosměrného elektromotoru s cizím buzením

Stejnoseměrný motor s cizím buzením tvoří obvod kotvy, budicí obvod a mechanická část pohonu. Podrobný matematický model, který by zahrnoval všechny elektromagnetické vazby motoru, je složitý. Pro potřeby technické praxe a pro řešení většiny otázek spjatých s návrhem regulace je však dostatečné použití modelu, který vychází z jistých zjednodušení. Obvykle zanedbáváme rozptylový magnetický tok budicího vinutí, vzájemné transformační působení jednotlivých vinutí, vliv vířivých proudů v magnetickém obvodu a úbytek napětí v kartáčcích. Východiskem pro odvození modelu, např. podle [3, 10, 11], je náhradní schéma stejnosměrného motoru s cizím buzením na obr. 5.19-1.

Definujeme základní veličiny

u_m [V]	– napětí na kotvě motoru (svorkové)
i_m [A]	– proud kotvy motoru
u_i [V]	– indukované napětí na kotvě motoru
R_m [Ω]	– celkový odpor všech vinutí v kotvě motoru
L_m [H]	– celková indukčnost všech vinutí v kotvě motoru
u_b [V]	– napětí budicího obvodu
i_b [A]	– budicí proud motoru
R_b [Ω]	– celkový odpor budicího obvodu
L_b [H]	– celková indukčnost budicího obvodu
J [kg m^2]	– celkový moment setrvačnosti vztažený k hřídeli motoru
ξ [Vs rad^{-1}]	– konstrukční konstanta motoru
Φ [Wb]	– magnetický tok
φ [rad]	– úhel natočení hřídele motoru
$\omega = d\varphi/dt$ [rads^{-1}]	– úhlová rychlost pohybu
M_m, M_Z [Nm]	– elektromagnetický a zatěžovací moment motoru

Elektrické parametry obvodu kotvy motoru jsou charakterizovány celkovým odporem R_m a indukčností L_m vinutí kotvy i dalších vedení a vinutí, která jsou s ním v sérii. Obvod kotvy lze tak popsat napětíovou rovnicí

$$u_m(t) = R_m i_m(t) + L_m \frac{di_m(t)}{dt} + u_i(t)$$

Indukované napětí na kotvě motoru $u_i(t)$ určené magnetickým tokem závislým na budicím proudu je rovno

$$u_i(t) = c \Phi(i_b) \omega(t)$$

S využitím Laplaceovy transformace můžeme rovnici pro napětí kotvy $u_m(t)$ převést do přenosového vyjádření

$$U_m(s) - U_i(s) = R_m I_m(s) + L_m s I_m(s) = I_m(s) [R_m + L_m s]$$

$$G_m(s) = \frac{I_m(s)}{U_m(s) - U_i(s)} = \frac{1}{L_m s + R_m} = \frac{\frac{1}{R_m}}{\frac{L_m}{R_m} s + 1} = \frac{K_m}{T_m s + 1}$$

kde

$$K_m = \frac{1}{R_m}, \quad T_m = \frac{L_m}{R_m}$$

Závislost magnetického toku Φ na proudu budicího obvodu i_b vyjadřuje magnetizační charakteristika motoru $\Phi = f(i_b)$, která je nelineární. Ve schématu na obr. 5.19-1 jsou dvě vstupní veličiny: napájecí napětí kotvy u_m a napájecí napětí budicího obvodu u_b . V obecném případě lze obě použít k regulaci. Matematický model je pak třeba doplnit o popis budicího obvodu

$$u_b(t) = R_b i_b(t) + L_b \frac{di_b(t)}{dt}$$

Obvod buzení lze rovněž pomocí Laplaceovy transformace vyjádřit přenosem

$$U_b(s) = R_b I_b(s) + L_b s I_b(s) = I_b(s) [R_b + L_b s]$$

$$G_b(s) = \frac{I_b(s)}{U_b(s)} = \frac{1}{L_b s + R_b} = \frac{\frac{1}{R_b}}{\frac{L_b}{R_b} s + 1} = \frac{K_b}{T_b s + 1}$$

kde

$$K_b = \frac{1}{R_b}, \quad T_b = \frac{L_b}{R_b}$$

Mechanická část motoru koná rotační pohyb vyvolaný elektromagnetickým momentem motoru M_m za působení momentu zátěže M_Z . Je-li J celkový moment setrvačnosti pohonu, pak úhlová rychlost je určena pohybovou rovnicí

$$J \frac{d\omega(t)}{dt} = M_m - M_Z$$

kde moment motoru M_m je určen pomocí magnetického toku obecně závislého na budícím proudu podle vztahu

$$M_m = c \Phi(i_b) i_m(t)$$

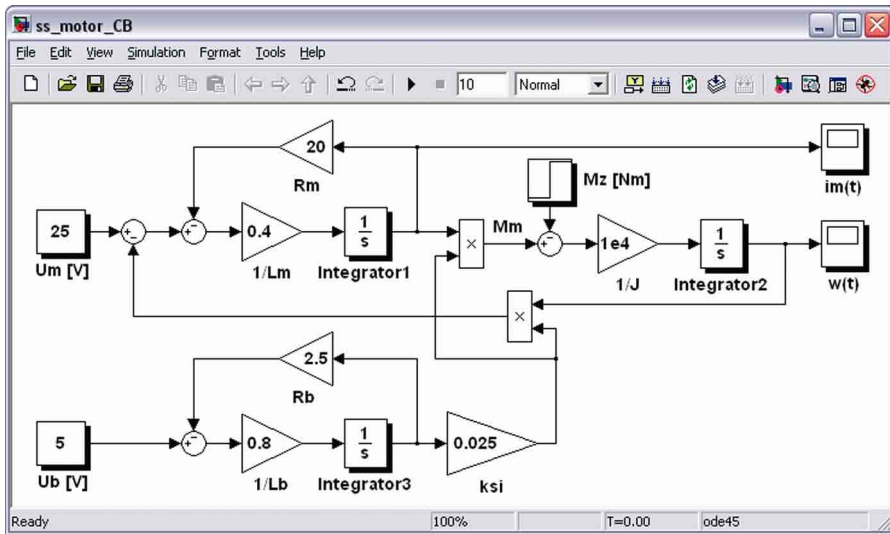
Následující tři rovnice tvoří matematický model stejnosměrného motoru s cizím buzením, jehož simulační schéma je na obr. 5.19-2.

$$\frac{di_m(t)}{dt} = \frac{1}{L_m} [u_m(t) - c \Phi(i_b) \omega(t) - R_m i_m(t)]$$

$$\frac{di_b(t)}{dt} = \frac{1}{L_b} [u_b(t) - R_b i_b(t)]$$

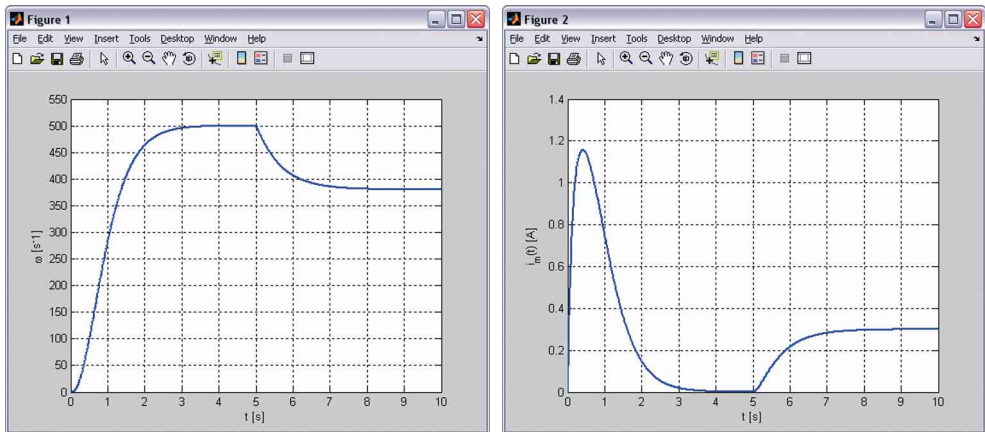
$$\frac{d\omega(t)}{dt} = \frac{1}{J} [c \Phi(i_b) i_m(t) - M_Z]$$

Simulaci provedeme pro tyto hodnoty: $u_m = 25 \text{ V}$, $R_m = 20 \text{ } \Omega$, $L_m = 2,5 \text{ H}$, $J = 10^{-5} \text{ kg m}^2$, $u_b = 5 \text{ V}$, $R_b = 2,5 \text{ } \Omega$, $L_b = 1,25 \text{ H}$ a $\xi = c \Phi = 0,025 \text{ Vs rad}^{-1}$. Moment zátěže M_Z se v čase $t = 5 \text{ s}$ změní z hodnoty 0 Nm na hodnotu $0,015 \text{ Nm}$.



Obr. 5.19-2: Simulační schéma stejnosmerného motoru s cizím buzením

Závislosti úhlové rychlosti a proudu kotvy na čase jsou na obr. 5.19-3.



Obr. 5.19-3: Průběh úhlové rychlosti a proudu kotvy motoru v závislosti na čase

5.20 Stejnosměrný motor s buzením permanentními magnety

Stejnosměrný motor patří mezi nejstarší stejnosměrné stroje. Přitom je ale stejnosměrný motor s cizím buzením ideálně regulovatelný, jelikož lze jeho otáčky plynule měnit změnou napětí přiváděného na kotvu motoru.

Pro servopohony se používají především motory s buzením permanentními magnety na statoru. Pro výrobu permanentních magnetů se používá magnetických tvrdých materiálů na bázi vzácných zemin, dnes nečastěji spěkaných materiálů Sm-Co (samárium-kobalt) nebo Fe-Nd-B (železo-neodym-bór).

Definujme základní veličiny

u_m [V]	– napětí na kotvě motoru (svorkové)
i_m [A]	– proud kotvy motoru
R_m [Ω]	– celkový odpor všech vinutí v kotvě motoru
L_m [H]	– celková indukčnost všech vinutí v kotvě motoru
J [kg m^2]	– celkový moment setrvačnosti vztažený k hřídeli motoru
k [Nm A^{-1}]	– konstanta motoru
Φ [Wb]	– magnetický tok
φ [rad]	– úhel natočení hřídele motoru
$\omega = d\varphi/dt$ [rads^{-1}]	– úhlová rychlost pohybu
M_m, M_Z [Nm]	– elektromagnetický a zatěžovací moment motoru

Je-li motor (schéma je na obr. 5.19-1, kapitola 5.19) řízený pouze změnou napětí kotvy při stálém budičím proudu $i_b = \text{konst}$ resp. je-li buzen permanentními magnety, lze jej popsat pomocí následujících dvou rovnic

$$u_m(t) = R_m i_m(t) + L_m \frac{di_m(t)}{dt} + c \Phi \omega(t)$$

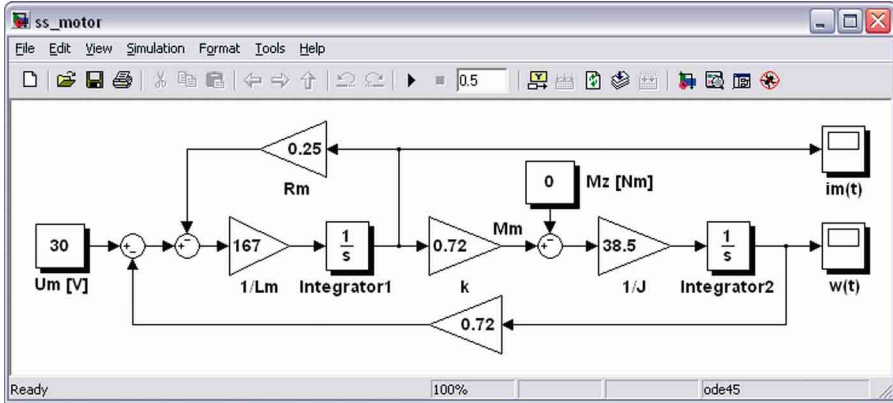
$$J \frac{d\omega(t)}{dt} + M_Z = M_m = c \Phi i_m(t)$$

Magnetický tok je konstantní (zanedbáváme reakci kotvy, resp. předpokládáme, že je kompenzována). Nemusíme tedy uvažovat nelineární funkci $\Phi = f(i_b)$, viz také kapitola 5.19, a zavedeme konstantu motoru $k = c \Phi$. Indukované napětí kotvy je pak přímo úměrné rychlosti otáčení. Upravíme-li výše uvedené rovnice do tvaru

$$\frac{di_m(t)}{dt} = \frac{1}{L_m} [u_m(t) - k \omega(t) - R_m i_m(t)]$$

$$\frac{d\omega(t)}{dt} = \frac{1}{J} [k i_m(t) - M_Z]$$

je možné strukturu modelu realizovat v Simulinku pomocí schéma na obr. 5.20-1.



Obr. 5.20-1: Simulační schéma stejnosměrného motoru s buzením permanentními magnety

Aplikujeme-li dále na uvedené rovnice Laplaceovu transformaci, lze celý model převést do přenosového vyjádření

$$U_m(s) = R_m I_m(s) + L_m s I_m(s) + k \Omega(s)$$

$$J s \Omega(s) = k I_m(s) - \mathcal{M}_Z(s)$$

z prvního vztahu vyjádříme proud kotvy a dosadíme do druhého. Po úpravě dostaneme

$$\Omega(s) = \frac{1}{k} \frac{1}{T_e T_m s^2 + T_m s + 1} U_m(s) - \frac{R_m}{k^2} \frac{T_e s + 1}{T_e T_m s^2 + T_m s + 1} \mathcal{M}_Z(s)$$

kde T_m označuje elektromagnetickou a T_e elektromechanickou časovou konstantu motoru

$$T_e = \frac{L_m}{R_m}, \quad T_m = \frac{R_m J}{k}$$

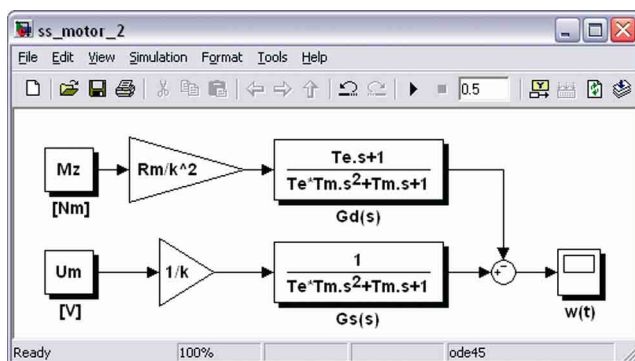
Napětí kotvy můžeme v tomto případě považovat za akční a zatěžovací moment za poruchovou veličinu. S ohledem na strukturu zpětnovazebního obvodu, tak jak je znázorněna na obr. 5.20-1, je možné pro přenos soustavy $G_S(s)$ a pro přenos poruchy $G_d(s)$ psát

$$G_S(s) = \frac{1}{k} \frac{1}{T_e T_m s^2 + T_m s + 1}$$

$$G_d(s) = -\frac{R_m}{k^2} \frac{T_e s + 1}{T_e T_m s^2 + T_m s + 1}$$

Z hlediska potlačení vlivu poruchy na průběh regulace rychlosti je jistou nepříjemností skutečnost, že relativní řád přenosu $G_d(s)$ je roven jedné, zatímco přenosu $G_S(s)$ dvěma. Rychlost odezvy na změnu poruchy (v tomto případě momentové zátěže) je tak větší než na změnu akční veličiny.

Simulační schéma motoru můžeme samozřejmě nyní také realizovat pomocí odvozených přenosů $G_S(s)$ a $G_d(s)$, viz obr. 5.20-2. Struktura modelu se v tomto případě podstatně zjednoduší.



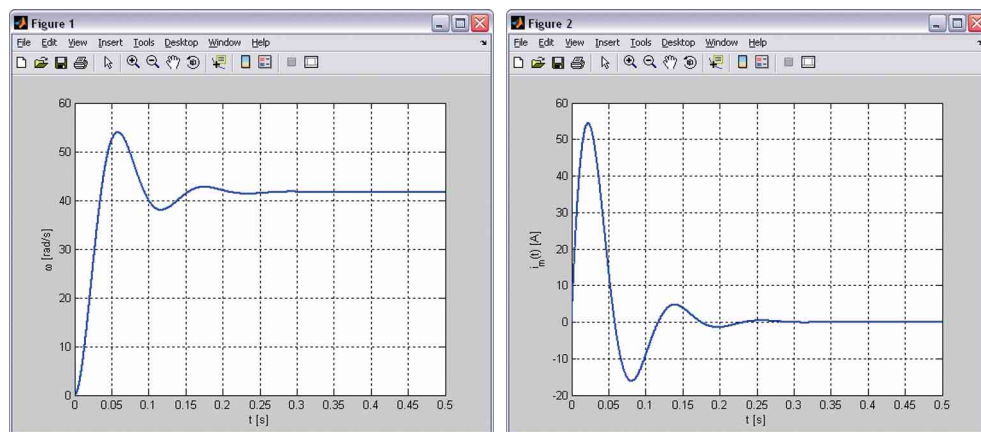
Obr. 5.20-2: Blokové schéma modelu stejnosměrného motoru s buzením permanentními magnety

S využitím věty o konečné hodnotě Laplaceovy transformace (limitní korespondence, viz kapitola 3.2.3 a příloha A) můžeme pomocí vtahu pro obraz úhlové rychlosti $\Omega(s)$ získat výraz pro ustálenou hodnotu úhlové rychlosti

$$\omega(\infty) = \frac{1}{k} U_m(\infty) - \frac{R_m}{k^2} M_Z(\infty)$$

Ustálená hodnota rychlosti lineárně klesá s rostoucím zatěžovacím momentem motoru M_Z . Vlastnosti motoru se z hlediska momentové zatížitelnosti zhoršují s narůstajícím odporem R_m v obvodu kotvy. Ustálenou hodnotu rychlosti $\omega(\infty)$ snadno ověříme v grafu na obr. 5.20-3; podle uvedeného výrazu je rovna $\omega(\infty) \cong 41,7$ rad/s.

Při simulaci vyjdeme z parametrů skutečného stejnosměrného motoru: $M_n = 13$ Nm, $n_n = 500$ min⁻¹, $U_{an} = 56$ V, $I_{an} = 18$ A, $R_m = 0,5$ Ω , $L_m = 6$ mH, $J = 0,026$ kg m². Vlastní simulaci provedeme pro napětí $u_m = 30$ V a moment zátěže $M_Z = 0$ Nm. Dále musíme také stanovit konstantu motoru $k = M_n / I_{an} = 0,72$ Nm A⁻¹.

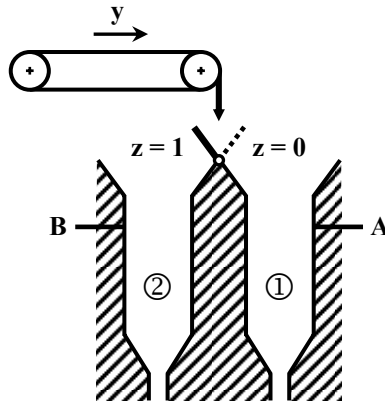


Obr. 5.20-3: Průběh úhlové rychlosti a proudu kotvy motoru v závislosti na čase

5.21 Kombinační logický obvod

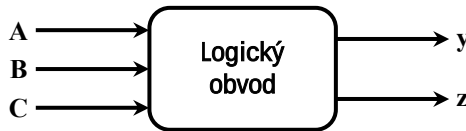
Sypký materiál se dopravuje pomocí pásového dopravníku do dvou zásobníků, viz obr. 5.21-1. Rozdělování materiálu se provádí pomocí hradítka, které zavádí materiál vždy do zásobníku, který není plný. Nejsou-li oba zásobníky plné, zavádí se materiál do prvního z nich. Po naplnění obou zásobníků se musí pásový dopravník automaticky zastavit. Dispečer může pásový dopravník zastavit kdykoliv.

Navrhněte a minimalizujte logickou funkci obvodu, který zajišťuje popsanou činnost. Realizujte ji pomocí prvků NAND (negace logického součinu) a NOR (negace logického součtu).



Obr. 5.21-1: Princip plnění zásobníků

Pokud není uvedeno předem, musíme nejprve vstupním a výstupním logickým veličinám přiřadit konkrétní fyzikální význam (tzv. výrokový kalkúl). Vstupní a výstupní veličiny logického obvodu jsou znázorněny na obr. 5.21-2.



Obr. 5.21-2: K definici logického obvodu

Pro vstupní proměnné platí

$$\begin{aligned}
 A &= \begin{cases} 1 \\ 0 \end{cases} \quad \text{materiál 1. zásobníku je} \quad \begin{cases} \text{nad} \\ \text{pod} \end{cases} \quad \text{požadovanou úroveň} \\
 B &= \begin{cases} 1 \\ 0 \end{cases} \quad \text{materiál 2. zásobníku je} \quad \begin{cases} \text{nad} \\ \text{pod} \end{cases} \quad \text{požadovanou úroveň} \\
 C &= \begin{cases} 1 \\ 0 \end{cases} \quad \text{požadavek dispečera je} \quad \begin{cases} \text{zastavit (stop)} \\ \text{automatický provoz} \end{cases}
 \end{aligned}$$

a pro výstupní proměnné

$$y = \begin{cases} 1 \\ 0 \end{cases} \quad \text{dopravník} \quad \begin{cases} \text{běží} \\ \text{stojí} \end{cases}$$

$$z = \begin{cases} 1 \\ 0 \end{cases} \quad \text{poloha hradítka} \quad \begin{cases} \text{zásobník č. 1} \\ \text{zásobník č. 2} \end{cases}$$

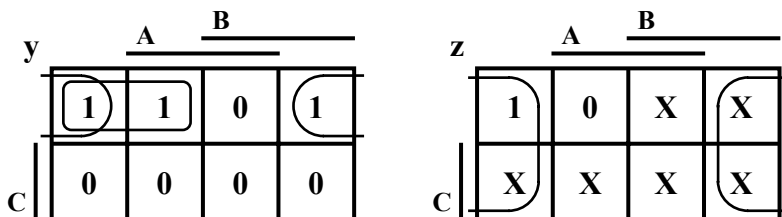
Dále musíme definovat logickou funkci. Učiníme tak pomocí tzv. pravdivostní tabulky, která představuje výčet kombinací vstupních proměnných a jim odpovídajících hodnot výstupních proměnných. V pravdivostní tabulce (tab. 5.21-1) jsou označeny symbolem X stavy, pro které je z hlediska funkce zařízení lhostejné, zda konkrétní hodnota výstupní proměnné z bude 0 nebo 1. Konkrétní hodnotu zvolíme později tak, aby vyšla realizace co nejjednodušší.

Tab. 5.21-1: Pravdivostní tabulka

A	B	C	y	z
0	0	0	1	1
1	0	0	1	0
0	1	0	1	1
1	1	0	0	X
0	0	1	0	X
1	0	1	0	X
0	1	1	0	X
1	1	1	0	X

Logická funkce se dá obecně vyjádřit (a také realizovat) různými způsoby. Pokud nám nezáleží na její složitosti, např. při realizaci pomocí PLC automatu, můžeme jen dodefinovat neurčité stavy a funkci realizovat přímo z pravdivostní tabulky (finální logická funkce musí být jednoznačná). Pokud však máme navrženou funkci realizovat pomocí skutečných součástek (a jde-li nám např. i o cenu její realizace), je potřeba vyjádření logické funkce minimalizovat.

K minimalizaci vyjádření logické funkce využijeme tzv. **Karnaughovu mapu**. Nejprve budeme logickou funkci realizovat pomocí hradel **NAND**.



Obr. 5.21-3: Minimalizace logických funkcí pomocí Karnaughovy mapy

Metoda je intuitivní, pokrýváme v mapě co největší tělesa (skupiny sousedních jedniček) pomocí tzv. termů. Tělesa se mohou překrývat. Do tělesa můžeme sdružit jen 2^n sousedních prvků.

V našem případě

$$y = \overline{B} \overline{C} + \overline{A} \overline{C}$$

$$z = \overline{A}$$

V této fázi řešení jsme jednoznačně určili, pro které kombinace vstupních proměnných nabývá funkce z hodnotu 1 a pro které 0. Při této volbě bude hradítka nastaveno vždy tak, aby bylo připraveno plnění preferovaného zásobníku číslo 1 vždy, pokud není naplněn, a to i v případě zásahu dispečera.

Tab. 5.21-2: Doplněná pravdivostní tabulka

A	B	C	y	z
0	0	0	1	1
1	0	0	1	0
0	1	0	1	1
1	1	0	0	0
0	0	1	0	1
1	0	1	0	0
0	1	1	0	1
1	1	1	0	0

		A		B	
z					
		1	0	0	1
C		1	0	0	1

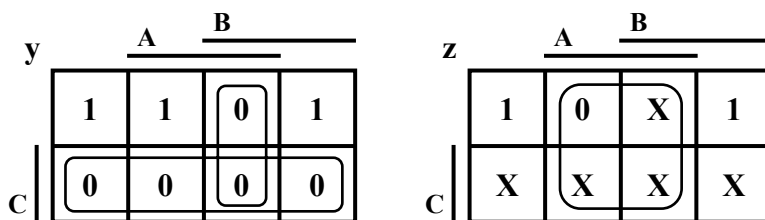
Obr. 5.21-4: Doplněná mapa

Povšimněme si, jak je vyjádřen nadřazený požadavek dispečera v součinném tvaru funkce. K úpravám logických výrazů používáme základní pravidla Booleovy dvouhodnotové algebry.

$$y = \overline{B} \overline{C} + \overline{A} \overline{C} = (\overline{B} + \overline{A}) \overline{C}$$

Uvedeným způsobem jsme získali vyjádření logické funkce ve tvaru součtu součinů, který je vhodný pro jednoduchou realizaci pomocí prvků NAND ve dvou úrovních.

Pro realizaci logické funkce pomocí prvků **NOR** je naopak vhodné její vyjádření ve tvaru součinů součtů. V tomto případě postupujeme podobně, pouze s tím rozdílem, že v Karnaughově mapě pokrýváme skupiny nul místo skupin jedniček.



Obr. 5.21-5: Minimalizace logických funkcí

Vyjádření obou logických funkcí v tomto případě je

$$\bar{y} = C + AB$$

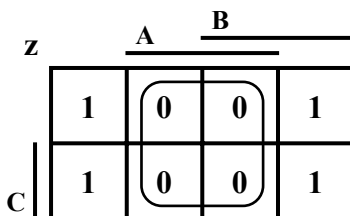
$$\bar{z} = A$$

a provedeme-li úpravu pomocí de Morganova zákona

$$y = \overline{C + AB} = \bar{C}(\bar{A} + \bar{B})$$

$$z = \bar{A}$$

Konkrétní tvar funkce z nabývá tvaru na obr. 5.21-6



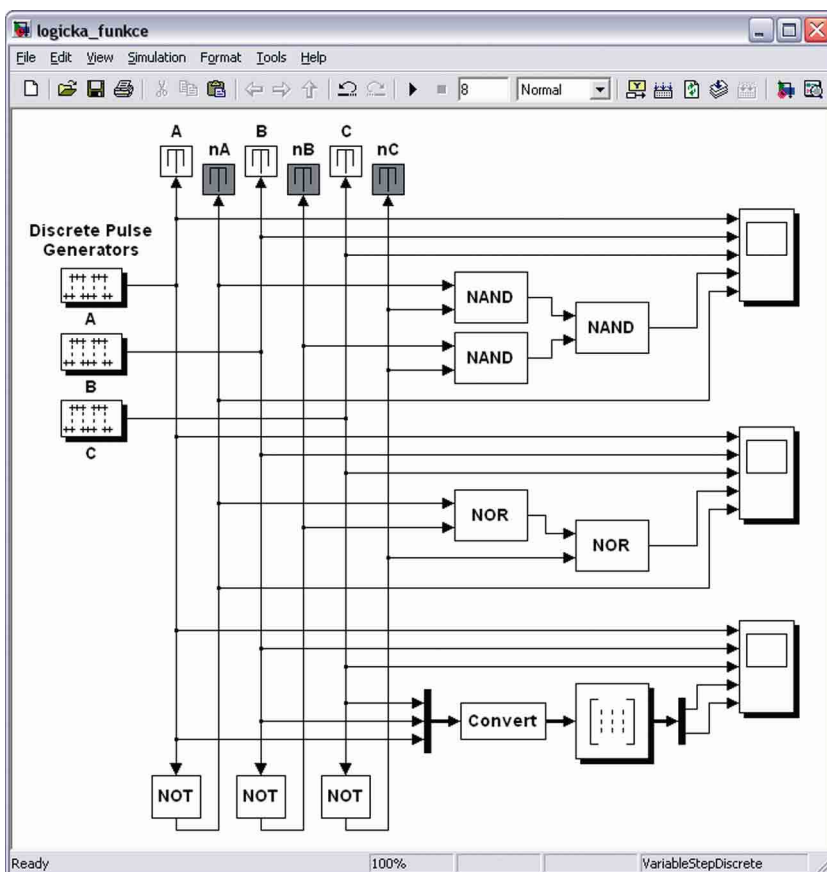
Obr. 5.21-6: Doplněná mapa

I tentokrát jsme obdrželi stejný konkrétní tvar funkce. Hradítka bude tedy i v tomto případě připraveno k plnění preferovaného zásobníku číslo 1 vždy, není-li naplněn.

Obě realizace logického obvodu je možné jednoduše simulovat v prostředí Simulink. V simulačním schématu na obr. 5.21-7 jsou uvedeny tři realizace logické funkce, pomocí prvků NAND a NOR (podle zadání), navíc ještě pro ilustraci realizace třetí, pomocí pravdivostní tabulky. Všechny realizace jsou z hlediska vnějšího chování identické.

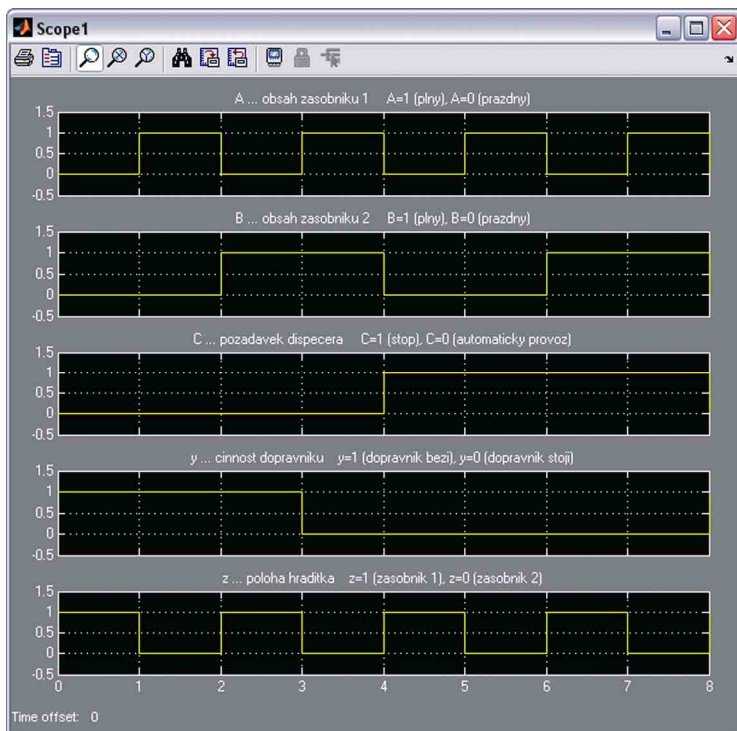
Generování vstupních logických proměnných A, B a C zajistíme prostřednictvím bloků *Pulse Generator*, ve kterých je nutné změnit parametr **Pulse type** z původního **Time based** na **Sample based**. Dále změníme nastavení parametru **Pulse width (number of samples)** z původních 50 na 1 a parametru **Sample time** – pro proměnnou A na hodnotu 1, pro proměnnou B na 2 a pro C na 4. Průběh takto definovaných proměnných je na obr. 5.21-8.

Pro přehlednost je v simulačním schématu na obr. 5.21-7 vytvořena speciální sběrnice pro jednotlivé vstupní proměnné a jejich negace, např. pro proměnnou A je její negace označena jako nA.



Obr. 5.21-7: Simulace kombinačního logického obvodu

Bloky pro realizaci logických funkcí NAND a NOR vytvoříme vložením univerzálního bloku *Logical Operator*, ve kterém zvolíme příslušný typ operátoru. V případě realizace pomocí pravdivostní tabulky využijeme pro definování této tabulky blok *Combinatorial Logic*. Datový typ vstupních veličin (jsou sdruženy do vektorového spoje) musíme změnit (je požadován typ Boolean) pomocí bloku *Convert*. Volbu typu můžeme prostřednictvím tohoto bloku provádět zcela automaticky, podle požadovaného typu na výstupu bloku, tedy podle typu, který je požadován na vstupu bloku následujícího (zde *Combinatorial Logic*). V bloku stačí nastavit parametr **Output data type mode** na **Inherit via back propagation**. Zobrazení všech sledovaných proměnných provedeme pomocí standardního zobrazovače *Scope*, pouze změníme počet jeho os (parametr **Number of axes**); blok bude následně mít tomu odpovídající počet vstupů.



Obr. 5.21-8: Průběhy všech sledovaných veličin

6 Příklady z oblasti teorie automatického řízení

V kapitolách 5.8 až 5.20 byly uvedeny konkrétní řešené příklady zaměřené především na techniky sestavování matematických modelů pomocí analytické identifikace, tedy pomocí tzv. matematicko-fyzikální analýzy. Byly vyloženy základní přístupy umožňující sestavení simulačních modelů dynamických systémů v prostředí MATLAB – Simulink.

V následujících kapitolách je uvedeno několik úloh, které jsou orientovány na oblast teorie automatického řízení. V prostředí Simulinku jsou názorným způsobem vytvářeny modely uzavřeného regulačního obvodu. Ten se skládá ze dvou základních částí – regulované soustavy a regulátoru. Pozornost je věnována zejména možnostem realizace spojitěho PID regulátoru. Jsou uvedeny i jiné implementace samotného PID algoritmu a některé jeho nejpoužívanější modifikace. Regulovaná soustava je modelována na základě příslušné diferenciální rovnice, jejíž řešení může být realizováno pomocí metody snižování řádu derivace či metody postupné integrace, nebo na základě obrazového přenosu. Modelování dynamických systémů na základě jejich diferenciálního popisu bylo podrobně popsáno již v kapitolách 2, 3 a 4. Ve stručnosti jsou uvedeny i některé klasické přístupy vedoucí k nastavení optimálních parametrů regulátoru.

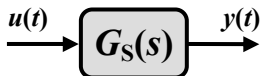
Dále jsou také zařazeny i úlohy, ve kterých jsou k vlastní regulaci modelovaného technologického procesu použity nespojitě (dvoupolohové a třípolohové) resp. číslicové (PSD) regulátory. Všechny řešené příklady byly opět voleny tak, aby byly přímo využitelné při výuce, zejména v předmětech zaměřených na automatizaci a řídicí techniku.

V závěru kapitoly jsou na konkrétním příkladě zformulovány a prakticky realizovány základní techniky pro numerickou optimalizaci dynamické funkce určené svými parametry. Je ilustrována jednoduchá, ale přesto velmi užitečná aplikace MATLABu, která umožňuje velmi efektivně podle zvoleného kritéria naléznout např. optimální parametry regulátoru. Jako příklad je uvedena úloha optimalizace parametrů číslicového regulátoru (spojitě popsaný proces je řízen diskretním regulátorem) se zvoleným kvadratickým kritériem s penalizací akčních zásahů. Úlohu je samozřejmě možné řešit i analyticky. To je však velmi pracné. Iterační simulační výpočet naopak tuto úlohu řeší velmi jednoduše, a to bez použití obtížných matematických postupů, s minimálním teoretickým vybavením. Problémem navíc není ani složitější dynamika regulované soustavy vzhledem k řádu, nelinearitám, omezením nebo komplikovanému kritériu, která by analytické řešení zcela znemožnila.

Soubory s modely a skripty, pomocí nichž jsou úlohy uvedené v následujících odstavcích v prostředí MATLAB – Simulink řešeny, jsou opět dostupné na přiloženém datovém nosiči. Přehled názvů v příslušné kapitole použitých souborů je uveden v příloze B.

6.1 Přechodové charakteristiky dynamických systémů 1. řádu

Abychom mohli sestavit model nějakého dynamického systému v Simulinku, je nutné se nejdříve zabývat diferenciálním popisem tohoto systému. Pomocí matematicko-fyzikální analýzy tedy obvykle hledáme diferenciální rovnici, která by vhodně popisovala dynamiku reálného fyzikálního systému.



Obr. 6.1-1: Dynamický systém

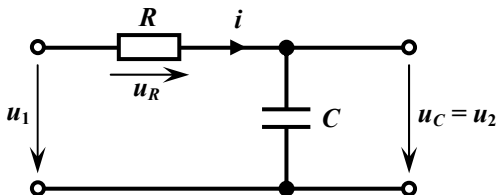
Chování dynamického systému 1. řádu na obr. 6.1-1 (někdy označovaného jako systém se zpožděním či setrvačností 1. řádu příp. jako setrvačný člen 1. řádu) je popsáno nejčastěji diferenciální rovnici

$$Ty'(t) + y(t) = Ku(t)$$

kde K je zesílení a T časová konstanta systému, případně pomocí odpovídajícího obrazového přenosu ve tvaru

$$G_S(s) = \frac{K}{Ts + 1}$$

Uvedená diferenciální rovnice může popisovat např. lineární elektrický obvod obsahující jediný kapacitor nebo induktor, viz obr. 6.1-2 a 6.1-3. Podobné dynamické systémy, ať již elektrické, mechanické, pneumatické nebo hydraulické, se pak označují také jako systémy s jedním akumulátorem energie (kondenzátor s kapacitou C resp. cívka s indukčností L) nebo jako jednodukapacitní systémy.



Obr. 6.1-2: Elektrický obvod – RC článek

Ukažme nyní, jakým způsobem lze odvodit diferenciální rovnici konkrétního systému, v tomto případě jednoduchého elektrického obvodu (RC článku) na obr. 6.1-2. Vyjdeme ze základních vztahů pro střídavé napětí a proud

$$u_R = Ri$$

$$i = C \frac{du_2}{dt}$$

Podle druhého Kirchhoffova zákona musí v každém okamžiku platit

$$u_1 = u_R + u_2$$

Postupným dosazováním za napětí u_R (z Ohmova zákona) a následně za proud i obdržíme následující diferenciální rovnici

$$u_1 = RC \frac{du_2}{dt} + u_2$$

Tuto rovnici můžeme formálně přepsat do tvaru

$$RC u_2'(t) + u_2(t) = u_1(t)$$

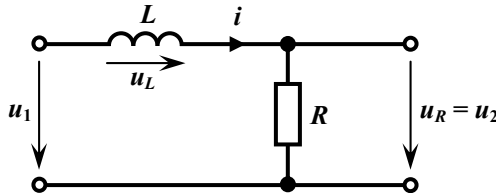
a pomocí Laplaceovy transformace (základní vzorce jsou v příloze A) ji převést na funkci operátoru s

$$RC s U_2(s) + U_2(s) = U_1(s)$$

Přenos dynamického systému je definován jako poměr Laplaceova obrazu výstupní veličiny k Laplaceově obrazu vstupní veličiny při nulových počátečních podmínkách

$$G(s) = \frac{U_2(s)}{U_1(s)} = \frac{1}{RCs + 1} = \frac{K}{Ts + 1}$$

kde $K = 1$ je zesílení systému a $T = RC$ je jeho časová konstanta.



Obr. 6.1-3: Elektrický obvod – LR článek

Obdobným způsobem lze odvodit diferenciální rovnici a přenos elektrického obvodu na obr. 6.1-3. Nejprve opět vyjdeme ze základních vztahů pro napětí a proud v obvodu

$$u_L = L \frac{di}{dt}$$

$$i = \frac{u_R}{R} = \frac{u_2}{R}$$

Dosazením do rovnice pro napětí v obvodu lze postupně psát

$$u_1 = u_L + u_2 = L \frac{di}{dt} + u_2$$

$$u_1 = \frac{L}{R} \frac{du_2}{dt} + u_2$$

Uvedenou diferenciální rovnici můžeme dále formálně přepsat do tvaru

$$\frac{L}{R} u_2'(t) + u_2(t) = u_1(t)$$

Přenos systému lze po provedení Laplaceovy transformace a elementárních úpravách zapsat jako

$$G(s) = \frac{U_2(s)}{U_1(s)} = \frac{1}{\frac{L}{R}s + 1} = \frac{K}{Ts + 1}$$

kde $K = 1$ je zesílení systému a $T = L/R$ je jeho časová konstanta.

Oba elektrické obvody (obr. 6.1-2 a 6.1-3) lze tedy popsat prostřednictvím stejné diferenciální rovnice resp. stejného obrazového přenosu. Vlastní dynamika konkrétního systému je určena jeho statickým zesílením K a časovou konstantou T (reálná nenulová čísla), viz také obr. 6.1-4.

Přechodová charakteristika $h(t)$ dynamického systému je obecně reakce dynamického systému vybuzeného z klidu (nulové počáteční podmínky) jednotkovou skokovou změnou vstupního (budicího) signálu. Její Laplaceův obraz je

$$H(s) = G_S(s) \frac{1}{s}$$

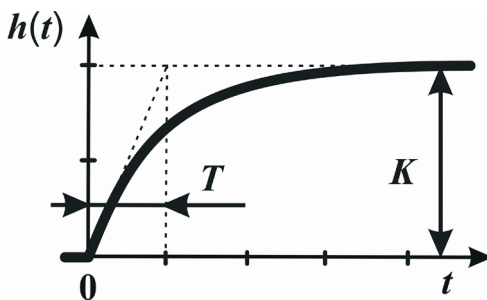
a konkrétně pro systém 1. řádu

$$H(s) = \frac{K}{Ts + 1} \frac{1}{s} = K \frac{\frac{1}{T}}{s(s + \frac{1}{T})}$$

Zpětnou Laplaceovou transformací, s využitím slovníku korespondencí v příloze A, získáme její analytické vyjádření v časové oblasti (tzv. přechodovou funkci)

$$h(t) = K(1 - e^{-\frac{t}{T}})\eta(t)$$

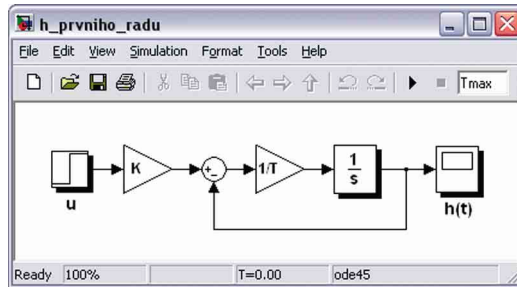
Možný průběh přechodové funkce je na obr. 6.1-4. Výše uvedený popis odpovídá nejčastější situaci ryze dynamického statického systému. Je stabilní pro $T > 0$.



Obr. 6.1-4: Přechodová charakteristika systému 1. řádu

Naším úkolem je nyní simulovat přechodovou charakteristiku popsaného dynamického systému a sledovat vliv jednotlivých parametrů na její průběh.

Příkladem realizace simulačního modelu dynamického systému prvního řádu je simulační schéma na obr. 6.1-5. Simulační model může sloužit k simulaci libovolné reakce tohoto dynamického systému. Speciálně však pro simulaci přechodové charakteristiky nastavíme nulové počáteční podmínky použitého integrátoru a za budicí signál $u(t)$ zvolíme jednotkový skok. Parametry systému definujeme pomocí hodnot zesílení příslušných zesilovačů. Pomocí zobrazovače (*Scope*) můžeme po provedení simulace sledovat průběh přechodové charakteristiky $y(t) = h(t)$.



Obr. 6.1-5: Simulace přechodové charakteristiky systému 1. řádu

```

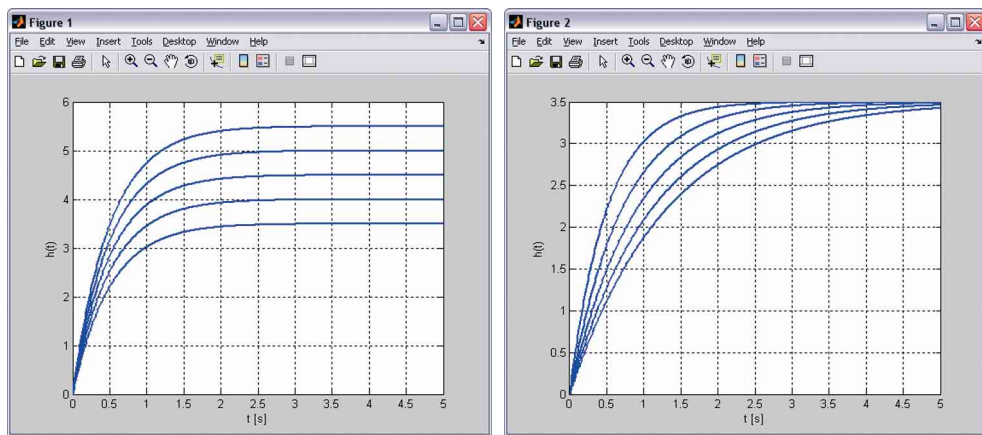
1  %
2  % Přechodova charakteristika dynamického systému 1. řádu
3  % Varianta: simulace několika průběhů přechodové charakteristiky
4  %           pro různé hodnoty jeho parametru
5  %
6  clear all, close all, clc
7  % Pole parametru simulace
8  % Nastavení parametru systému
9  K=3.5;      % statické zesílení
10 T=0.5;      % časová konstanta (T různé od nuly)
11 % Vyber příslušné variace parametru systému
12 P0=K; Krok=0.5; Pocet=5; Q=0; % variace parametru K
13 %P0=T; Krok=0.2; Pocet=5; Q=1; % variace parametru T
14 % (znak % blokuje provedení příkazu uvedených za ním až do konce řádku)
15 %
16 Tmax=8;    % maximální doba simulace
17 % Konec pole parametru simulace
18
19 h_prvniho_radu % otevře okno se simulačním schématem
20 for P=P0:Krok:P0+Krok*(Pocet-1) % začátek cyklu s měnícím se parametrem
21     switch Q % přiřazení podle příznaku vybrané variace
22         case 0, K=P;
23         otherwise, T=P;
24     end
25     sim('h_prvniho_radu') % spustí simulaci "h_prvniho_radu.mdl"
26     plot(t,h(:,2),'LineWidth',2) % vykreslení přechodové charakteristiky
27     hold on % umožní vykreslování do stejného okna
28     %pause % prerušení po vykreslení jedné křivky
29 end % konec cyklu % (pokračování libovolnou klávesou)
30 grid on % vykreslení mřížky
31 xlabel('t [s]'), ylabel('h(t)') % popis os

```

Obr. 6.1-6: Skript pro řízení výpočtu

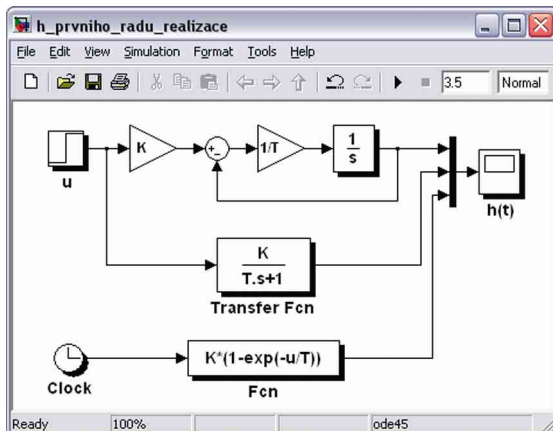
V uvedeném schématu ukládáme navíc časový průběh $h(t)$ a odpovídající čas t pomocí bloku *Scope* do pracovního prostoru MATLABu. Důvodem je možnost dalšího zpracování numerické informace o průběhu přechodové charakteristiky. Např. lze průběh přechodové funkce zobrazit do grafu a po případné editaci jej přenést do textového editoru.

Skript na obr. 6.1-6 řeší jednoduchým způsobem ovládání výpočtu a vykreslení sítě průběhů přechodových charakteristik odpovídajících měnícímu se parametru systému. Pro spolupráci skriptu a simulačního modelu musejí být oba soubory umístěny v aktuálním adresáři. Struktura skriptu je zřejmá z vepsaných komentářů.



Obr. 6.1-7: Simulované přechodové charakteristiky (variací parametrů K a T)

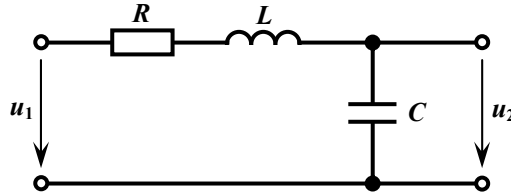
Uvedená realizace (obr. 6.1-5) není samozřejmě jediná možná. Další možností je simulace tohoto dynamického systému na základě jeho obrazového přenosu. K tomu lze použít blok *Transfer Fcn*. Simulink umožňuje i simulaci průběhu přechodové charakteristiky definované pomocí analytické funkce (originálu či předmětu přechodové funkce). Funkci lze zapsat v bloku *Fcn*, vektor času generujeme pomocí bloku *Clock*, viz obr. 6.1-8.



Obr. 6.1-8: Možnosti simulace přechodové odezvy systému 1. řádu

6.2 Přechodové charakteristiky dynamických systémů 2. řádu

Zabýváme se podrobněji dynamickými systémy 2. řádu. Tyto systémy bývají označovány také jako setrvačné resp. kmitavé členy 2. řádu. Představiteli těchto systémů jsou obvody či zařízení, která obsahují dva akumulátory energie (obecně např. dvojice kapacita-indukčnost nebo hmota-pružina). Příkladem takového systému může být RLC článek na obr. 6.2-1 nebo sedačka řidiče, jejíž zjednodušený model byl podrobně popsán v kapitole 3.



Obr. 6.2-1: Příklad systému 2. řádu – RLC článek

Chování dynamického systému 2. řádu lze popsat diferenciální rovnicí ve tvaru

$$a_2 y''(t) + a_1 y'(t) + a_0 y(t) = b_2 u''(t) + b_1 u'(t) + b_0 u(t)$$

Z podmínky fyzikální realizovatelnosti systému musí být stupeň nejvyšší derivace výstupní veličiny $y(t)$ větší nebo roven stupni nejvyšší derivace vstupní veličiny $u(t)$. Budeme-li dále uvažovat ryze dynamický proporcionalní systém ($b_2 = b_1 = 0$), nabude diferenciální rovnice tvaru

$$a_2 y''(t) + a_1 y'(t) + a_0 y(t) = b_0 u(t)$$

Uvedenou diferenciální rovnici můžeme při nulových počátečních podmínkách, použitím pravidel Laplaceovy transformace (příloha A), transformovat a následně vyjádřit obrazový přenos systému

$$G_S(s) = \frac{b_0}{a_2 s^2 + a_1 s + a_0}$$

Přenos lze dále postupně upravovat

$$G_S(s) = \frac{\frac{b_0}{a_0}}{\frac{a_2}{a_0} s^2 + \frac{a_1}{a_0} s + 1} = \frac{K}{(Ts)^2 + 2\xi Ts + 1} = \frac{K}{\left(\frac{s}{\omega_n}\right)^2 + 2\xi \frac{s}{\omega_n} + 1} = \frac{p(s)}{q(s)}$$

kde $K = b_0/a_0$ je statické zesílení systému, ξ je jeho **poměrné (relativní) tlumení** a $\omega_n = 1/T$ je **přirozená (vlastní) frekvence netlumeného systému**, viz také obr. 6.2-2.

Laplaceův obraz přechodové charakteristiky systému je obecně roven

$$H(s) = G_S(s) \frac{1}{s}$$

a v případě popisovaného systému 2. řádu pak

$$H(s) = \frac{K}{\left(\frac{s}{\omega_n}\right)^2 + 2\xi \frac{s}{\omega_n} + 1} \frac{1}{s} = \frac{p(s)}{q(s)} \frac{1}{s}$$

Odezva systému je charakterizována kořeny polynomu $q(s)$ ve jmenovateli přenosu $H(s)$, které jsou označovány jako tzv. **póly** přenosu. Poznamenejme, že kořeny polynomu čitatele $p(s)$ se označují jako tzv. **nuly** přenosu; v našem případě je ale v čitateli pouze konstanta.

Polynom ve jmenovateli přenosu $q(s)$ se nazývá **charakteristický polynom** a pokud je $q(s) = 0$, hovoříme o **charakteristické rovnici**. Upravíme-li dále přenos systému do tvaru

$$G_S(s) = K \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} = \frac{p(s)}{q(s)}$$

je pak jeho charakteristická rovnice

$$q(s) = s^2 + 2\xi\omega_n s + \omega_n^2 = 0$$

Charakteristickou rovnici lze vyjádřit jako součin kořenových činitelů

$$s^2 + 2\xi\omega_n s + \omega_n^2 = (s - s_1)(s - s_2)$$

a pro relativní tlumení $\xi < 1$ určit oba kořeny této rovnice

$$s_{1,2} = -\xi\omega_n \pm j\omega_n\sqrt{1-\xi^2} = -\xi\omega_n \pm j\omega_d$$

kde $\omega_d = \omega_n\sqrt{1-\xi^2}$ je tzv. **přirozená frekvence tlumeného systému**.

Obraz přechodové charakteristiky $H(s)$ můžeme s ohledem na uvedené vyjádřit

$$H(s) = K \frac{\omega_n^2}{s(s + \xi\omega_n - j\omega_n\sqrt{1-\xi^2})(s + \xi\omega_n + j\omega_n\sqrt{1-\xi^2})}$$

Je zřejmé, že ke kořenům charakteristické rovnice systému s_1 a s_2 přibude ještě navíc kořen třetí $s_3 = 0$.

Pomocí zpětné Laplaceovy transformace, s využitím věty o reziduích [14, 15], lze nyní vyjádřit přechodovou funkci v časové oblasti

$$h(t) = K \left(\frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \right)_{s=0} + 2 \operatorname{Re} \left[\left(\frac{\omega_n^2}{s(s + \xi\omega_n - j\omega_n\sqrt{1-\xi^2})} \right)_{s=-\xi\omega_n - j\omega_n\sqrt{1-\xi^2}} e^{-\xi\omega_n t} e^{-j(\omega_n\sqrt{1-\xi^2})t} \right]$$

$$h(t) = K \left\{ 1 - \frac{1}{\sqrt{1-\xi^2}} e^{-\xi\omega_n t} \sin \left[(\omega_n \sqrt{1-\xi^2}) t + \arctg \frac{\sqrt{1-\xi^2}}{\xi} \right] \right\} \eta(t)$$

Názorná grafická interpretace parametrů kmitavého proporcionálního systému 2. řádu je na obr. 6.2-2. S ohledem na výše uvedené souvislosti lze přechodovou funkci v závislosti na čase zjednodušeně zapsat

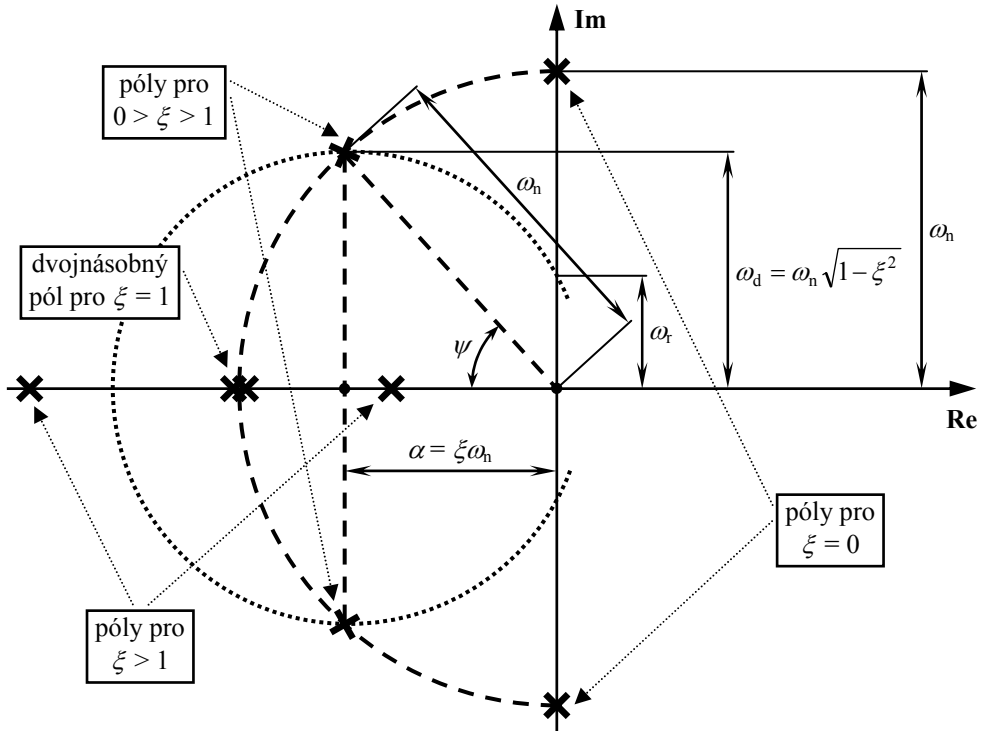
$$h(t) = K [1 - C e^{-\alpha t} \sin(\omega_d t + \psi)] \eta(t)$$

kde α je tzv. **absolutní tlumení** a dále platí, že

$$C = \frac{1}{\omega_d T} = \frac{\omega_n}{\omega_d}, \quad \alpha = \frac{\xi}{T} = \xi \omega_n, \quad \omega_n = \frac{1}{T}, \quad \omega_d = \frac{1}{T} \sqrt{1-\xi^2} = \omega_n \sqrt{1-\xi^2}$$

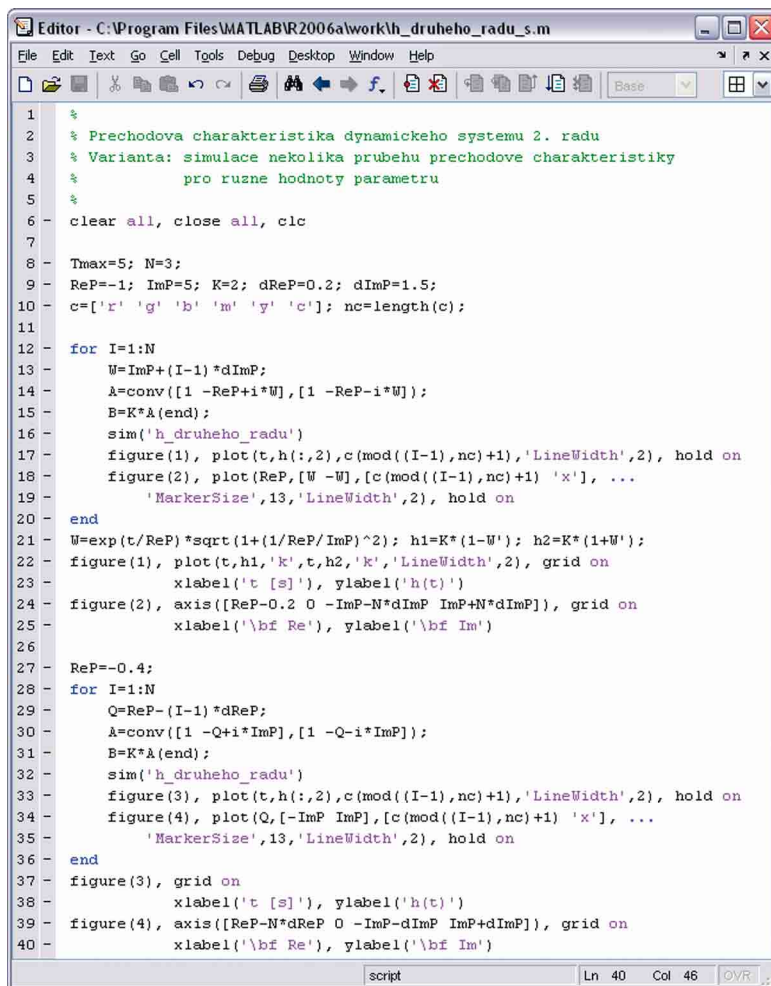
$$\omega_r = \omega_n \sqrt{1-2\xi^2}, \quad \psi = \arctg \frac{\omega_d}{\alpha} = \frac{\sqrt{1-\xi^2}}{\xi} = \arccos \xi$$

Na obr. 6.2-2 jsou zakresleny základní polohy pólů v závislosti na relativním tlumení ξ . Diskutujme nyní vliv jejich polohy na průběh odezvy systému. Budeme předpokládat, že systém je stabilní, tj. že všechny jeho póly leží v levé polorovině komplexní roviny.



Obr. 6.2-2: Geometrická interpretace parametrů a rozložení pólů systému druhého řádu

Zkoumejme nejprve vliv absolutního tlumení α , někdy také označovaného jako stupeň stability, který vyjadřuje vzdálenost dvojice komplexně sdružených pólů od imaginární osy. Pomocí jednoduchého skriptu (obr. 6.2-3), pouze s využitím základních příkazů MATLABu, definujeme tři dvojice pólů tak, aby měly vždy stejnou reálnou část (tj. absolutní tlumení) a rozdílnou část komplexní.



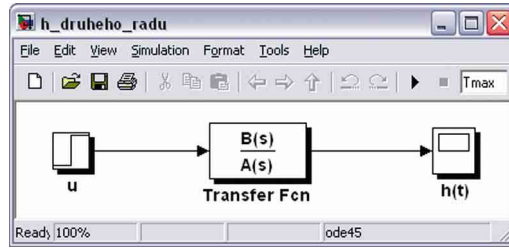
```

1 %
2 % Prechodova charakteristika dynamickeho systemu 2. radu
3 % Varianta: simulace nekolika prubehu prechodove charakteristiky
4 % pro ruzne hodnoty parametru
5 %
6 clear all, close all, clc
7
8 Tmax=5; N=3;
9 ReP=-1; ImP=5; K=2; dReP=0.2; dImP=1.5;
10 c=['r' 'g' 'b' 'm' 'y' 'c']; nc=length(c);
11
12 for I=1:N
13     W=ImP+(I-1)*dImP;
14     A=conv([1 -ReP+i*W], [1 -ReP-i*W]);
15     B=K*A(end);
16     sim('h_druheho_radu')
17     figure(1), plot(t, h(:,2), c(mod((I-1),nc)+1), 'LineWidth',2), hold on
18     figure(2), plot(ReP, [W -W], [c(mod((I-1),nc)+1) 'x'], ...
19         'MarkerSize',13, 'LineWidth',2), hold on
20 end
21 W=exp(t/ReP)*sqrt(1+(1/ReP/ImP)^2); h1=K*(1-W); h2=K*(1+W);
22 figure(1), plot(t, h1, 'k', t, h2, 'k', 'LineWidth',2), grid on
23 xlabel('t [s]'), ylabel('h(t)')
24 figure(2), axis([ReP-0.2 0 -ImP-N*dImP ImP+N*dImP]), grid on
25 xlabel('\bf Re'), ylabel('\bf Im')
26
27 ReP=-0.4;
28 for I=1:N
29     Q=ReP-(I-1)*dReP;
30     A=conv([1 -Q+i*ImP], [1 -Q-i*ImP]);
31     B=K*A(end);
32     sim('h_druheho_radu')
33     figure(3), plot(t, h(:,2), c(mod((I-1),nc)+1), 'LineWidth',2), hold on
34     figure(4), plot(Q, [-ImP ImP], [c(mod((I-1),nc)+1) 'x'], ...
35         'MarkerSize',13, 'LineWidth',2), hold on
36 end
37 figure(3), grid on
38 xlabel('t [s]'), ylabel('h(t)')
39 figure(4), axis([ReP-N*dReP 0 -ImP-dImP ImP+dImP]), grid on
40 xlabel('\bf Re'), ylabel('\bf Im')

```

Obr. 6.2-3: Skript pro řízení výpočtu

Konkrétní rozložení zvolených pólů je na obr. 6.2-5. Skript umožňuje jednoduchou změnou parametrů N , ReP , ImP , dImP a K měnit počet dvojic komplexně sdružených pólů, velikost jejich reálné složky, počáteční velikost imaginární složky a krok její změny a také zesílení soustavy. Nejprve je na základě definovaných pólů charakteristické rovnice a zesílení K sestaven odpovídající přenos. Simulace je pak provedena s využitím jednoduchého modelu na obr. 6.2-4. V bloku *Transfer Fcn* jsou načítány vektory koeficientů přenosu B a A .

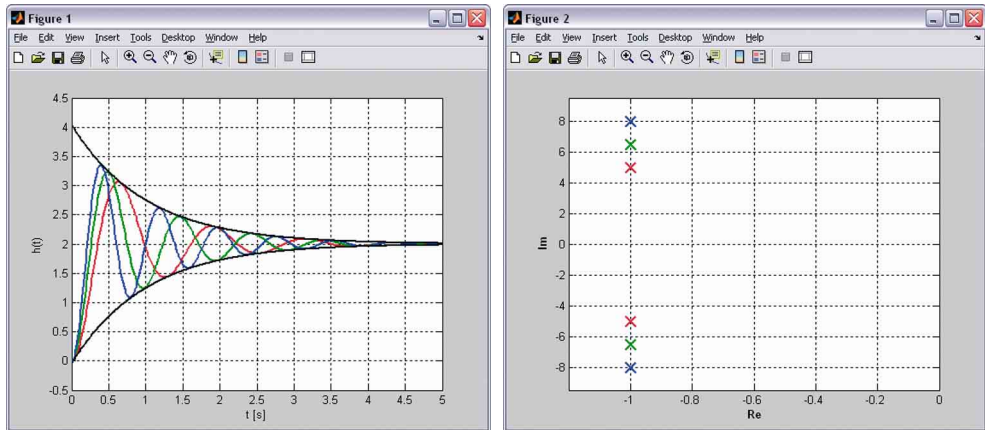


Obr. 6.2-4: Simulace přechodové odezvy systému

Na obr. 6.2-5 jsou také uvedeny i příslušné průběhy přechodových funkcí. V grafu je vykreslena navíc obalová křivka této sítě charakteristik, jejíž vyjádření je

$$c(t) = K \left[1 \pm e^{-\alpha t} \sqrt{1 + \left(\frac{\alpha}{\omega_d} \right)^2} \right]$$

Připomeňme, že absolutní tlumení α je rovno reálné části pólů (ReP) a přirozená frekvence tlumeného systému ω_d je rovna části imaginární (ImP), neboť $s_{1,2} = -\alpha \pm j\omega_d$.



Obr. 6.2-5: Přechodové charakteristiky systému 2. řádu při stejném absolutním tlumení α

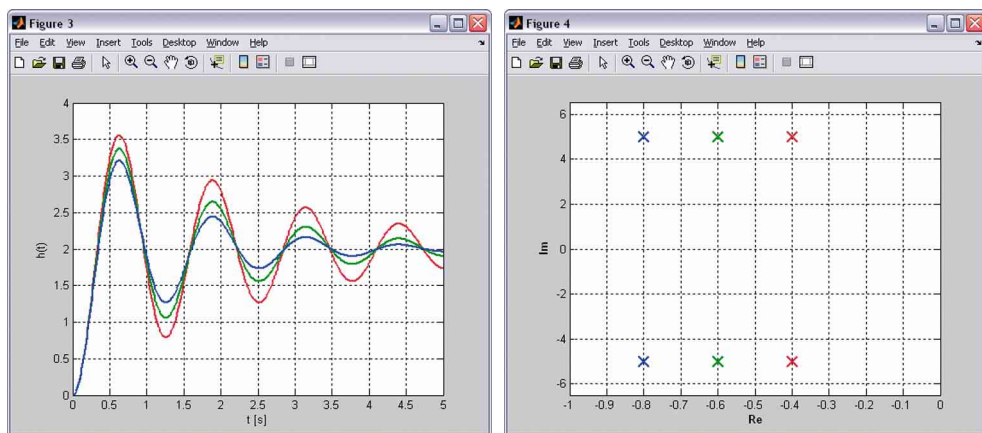
Pomocí skriptu na obr. 6.2-4 můžeme obdobným způsobem měnit naopak reálnou složku pólů při zachování složky imaginární (neboli při zachování přirozené frekvence tlumeného systému ω_d). Změnou parametrů N, ReP, dReP, ImP lze měnit počet dvojic komplexně sdružených pólů, počáteční velikost jejich reálné složky, krok její změny a velikost složky imaginární. Grafické znázornění polohy pólů a příslušné přechodové charakteristiky jsou na obr. 6.2-6. V tomto případě platí, že vzdálenost dvou sousedních vrcholů kmitů přechodové charakteristiky je vždy stejná, tedy

$$T_k = \frac{2\pi}{\omega_d} = \frac{2\pi}{\omega_n \sqrt{1 - \zeta^2}}$$

Lze samozřejmě stanovit i maximum charakteristiky a jemu odpovídající čas

$$h_m = K \left(1 + e^{-\frac{\pi\alpha}{\omega_d}}\right) = K \left(1 + e^{-\frac{\pi\xi}{\sqrt{1-\xi^2}}}\right), \quad t_m = \frac{\pi}{\omega_d} = \frac{\pi}{\omega_n \sqrt{1-\xi^2}}$$

V případě, kdy např. volíme $s_{1,2} = -0,4 \pm j5$ (na obr. 6.2-6 dvojice komplexně sdružených pólů zcela vpravo) je maximum tomu odpovídající přechodové charakteristiky $h_m \approx 3,56$ v čase $t_m \approx 0,63$ s.



Obr. 6.2-6: Přechodové charakteristiky systému 2. řádu při stejné frekvenci ω_d

Relativní tlumení ξ určuje velikost relativního překmitu κ (obr. 6.2-8). Konstantní hodnotě relativního tlumení odpovídají dvě polopřímky vedené z počátku souřadného systému, které svírají se zápornou reálnou poloosou úhel ψ (komplexní kořeny vždy vystupují v komplexně sdružených dvojicích).

Pokud je úhel $\psi = 45^\circ$, pak kružnice vedená dvojicí navzájem komplexně sdružených pólů (o poloměru ω_d) prochází zároveň i počátkem souřadnic. Rezonanční frekvence systému ω_r bude v tomto případě nulová. Tato situace nastane, budeme-li komplexně sdružené póly na obr. 6.2-2 posouvat z výchozí pozice (je zakreslena) směrem od počátku.

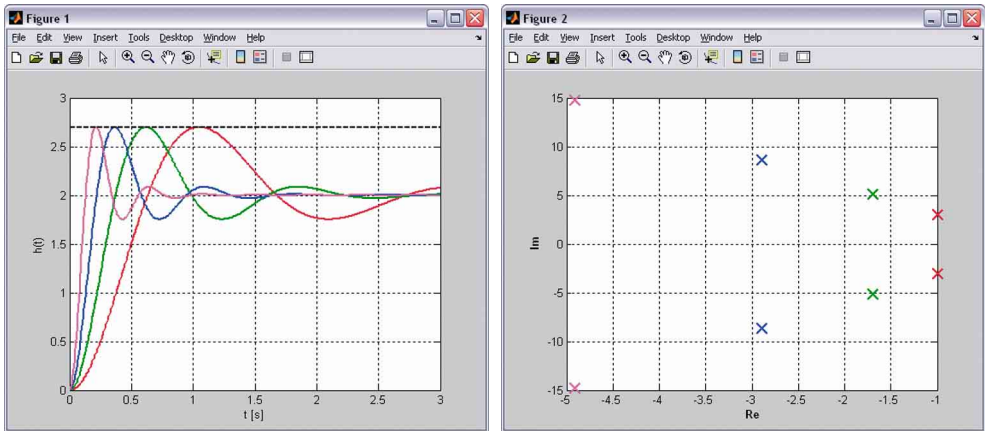
K názorné ilustraci problematiky lze využít skript na obr. 6.2-7. K simulaci použijeme opět jednoduchý model na obr. 6.2-4. Ve skriptu lze změnou parametrů N , ReP , ImP , p , mp měnit počet dvojic komplexně sdružených pólů, počáteční velikost jejich reálné a imaginární složky a koeficienty jejich změny. Grafické znázornění polohy pólů a příslušné přechodové charakteristiky jsou na obr. 6.2-8. Je zřejmé, že všechny přechodové charakteristiky mají stejně velký relativní překmit, dochází pouze ke změně časového měřítka.

```

Editor - C:\Program Files\MATLAB\R2006a\work\h_druheho_radu_ksi.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1 %
2 % Přechodova charakteristika dynamickeho systemu 2. radu
3 % Varianta: simulace nekolika prubehu prechodove charakteristiky
4 %      pri stejnem relativnim tlumeni
5 %
6 clear all, close all, clc
7
8 Tmax=3; N=4; p=1; mp=1.7;
9 ReP=-1; ImP=3; K=2;
10 c=['r' 'g' 'b' 'm' 'y' 'c']; nc=length(c);
11
12 for I=1:N
13     W=p*(ReP+i*ImP); p=p*mp;
14     A=conv([1 -W],[1 conj(-W)]);
15     B=K*A(end);
16     sim('h_druheho_radu')
17     figure(1), plot(t,h(:,2),c(mod((I-1),nc)+1),'LineWidth',2), hold on
18     figure(2), plot(W,[c(mod((I-1),nc)+1) 'x'], ...
19         'MarkerSize',13,'LineWidth',2), hold on
20     figure(2), plot(conj(W),[c(mod((I-1),nc)+1) 'x'], ...
21         'MarkerSize',13,'LineWidth',2)
22 end
23 kappa=max(h(:,2));
24 figure(1), plot([min(t) max(t)],[kappa kappa],'k--','LineWidth',2)
25 grid on, xlabel('t [s]'), ylabel('h(t)')
26 figure(2), grid on, xlabel('\bf Re'), ylabel('\bf Im')

```

Obr. 6.2-7: Skript pro vykreslení charakteristik při stejném relativním tlumení ξ



Obr. 6.2-8: Přechodové charakteristiky systému 2. řádu při stejném relativním tlumení ξ

6.3 Konstrukce dynamických charakteristik systému s využitím Control System Toolboxu

Rozšiřující modul MATLABu **Control System Toolbox** poskytuje značné množství příkazů pro teorii automatického řízení (více **help control**). Funkce z oblasti analýzy a návrhu řídicích systémů jsou založeny nejen na klasickém popisu pomocí přenosů, ale i na popisu systémů ve stavovém prostoru. Toolbox zavádí také tzv. lineární časově invariantní objekty (LTI), což jsou struktury popisující jednorozměrové i mnohorozměrové lineární systémy.

Z důvodu omezeného rozsahu tohoto textu, není samozřejmě možné popsat všechny dostupné příkazy Control System Toolboxu. Uvedme tedy pouze několik nejdůležitějších příkazů, pomocí kterých lze vytvořit jednoduchý skript pro vykreslování dynamických charakteristik systému libovolného řádu. Za dynamické charakteristiky systému považujeme charakteristiky **přechodové**, **váhové** a **frekvenční**. Konkrétní systém budeme definovat pomocí obrazového přenosu, nejprve zadáním příslušných koeficientů, později definováním pólů a nul.

Dynamický systém n -tého řádu je obvykle popsán lineární diferenciální rovnicí s konstantními koeficienty, obecně ve tvaru

$$a_n y^{(n)}(t) + a_{n-1} y^{(n-1)} + \dots + a_1 y'(t) + a_0 y(t) = b_m u^{(m)}(t) + \dots + b_1 u'(t) + b_0 u(t)$$

Z podmínky fyzikální realizovatelnosti systému musí být $m \leq n$, tj. stupeň nejvyšší derivace výstupní veličiny musí být větší nebo roven stupni nejvyšší derivace vstupní veličiny. Obrazový přenos systému je definován jako poměr Laplaceova obrazu výstupní veličiny k Laplaceově obrazu vstupní veličiny při nulových počátečních podmínkách. Jak již bylo řečeno v kapitole 3, diferenciální rovnici můžeme použitím pravidel Laplaceovy transformace při splnění výše uvedených podmínek transformovat do tvaru

$$[a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0] Y(s) = [b_m s^m + \dots + b_1 s + b_0] U(s)$$

Obrazový přenos systému je tedy

$$G_S(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + \dots + b_1 s + b_0}{a_n s^n + \dots + a_1 s + a_0}$$

Přenos systému lze v MATLABu definovat pomocí příkazu **tf**. Označme dále polynom čitatele přenosu $B(s)$ a polynom jmenovatele (charakteristický) $A(s)$. Příkaz zapíšeme ve tvaru **Gs = tf(B,A)**.

Pokud polynomy $B(s)$ a $A(s)$ rozložíme na kořenové činitele, lze přenos systému zapsat ve tvaru

$$G_S(s) = \frac{b_m (s - s_{b1})(s - s_{b2}) \dots (s - s_{bm})}{a_n (s - s_{a1})(s - s_{a2}) \dots (s - s_{an})}$$

kde $s_{b1}, s_{b2}, \dots, s_{bm}$ jsou kořeny čitatele (nuly) a $s_{a1}, s_{a2}, \dots, s_{an}$ jsou kořeny jmenovatele (póly). Zesílení systému je dáno poměrem b_m/a_n .

```

1 %
2 % Dynamické charakteristiky systému n-teho řadu
3 %
4 clear all, close all, clc, format compact
5
6 Z=-0.6; P=[-0.5+sqrt(3)/2*i -0.5-sqrt(3)/2*i -1.5 -3]; K=12;
7 Gs1=zpk(Z,P,K)
8 [B1,A1]=tfdata(Gs1,'v')
9 K1=dcgain(Gs1)
10 Gs1=tf(B1,A1)
11
12 Z=0.6; P=[-1 -0.4 -0.4]; K=0.7;
13 Gs2=zpk(Z,P,K)
14 [B2,A2]=tfdata(Gs2,'v')
15 K2=dcgain(Gs2)
16 Gs2=tf(B2,A2)
17
18 figure, pzmap(Gs1,Gs2), grid on
19 figure, step(Gs1,Gs2), grid on
20 figure, impulse(Gs1,Gs2), grid on
21 figure, nyquist(Gs1,Gs2), grid on
22 figure, bode(Gs1,Gs2), grid on

```

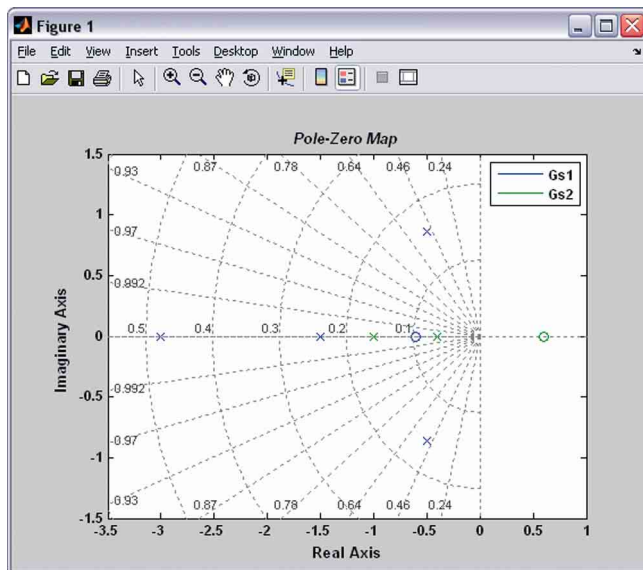
Obr. 6.3-1: Skript pro vykreslení dynamických charakteristik systému

Pro sestavení obrazového přenosu ze zadaných nul (zeros), pólů (poles) a zesílení (gain) lze v MATLABu použít příkaz **zpk**. Příkaz zapíšeme ve tvaru **Gs = zpk(Z, P, K)**. Přenos vytvořený příkazem **zpk** je možné převést na přenos s polynomy v čitateli a ve jmenovateli. Pomocí příkazu **[B, A] = tfdata(Gs, 'v')** nejprve tyto polynomy vypočteme a pomocí příkazu **tf** z nich následně vytvoříme obrazový přenos. Ve skriptu na obr. 6.3-1 jsou takto postupně definovány dva dynamické systémy, jejichž přenosy jsou uvedeny na obr. 6.3-2.

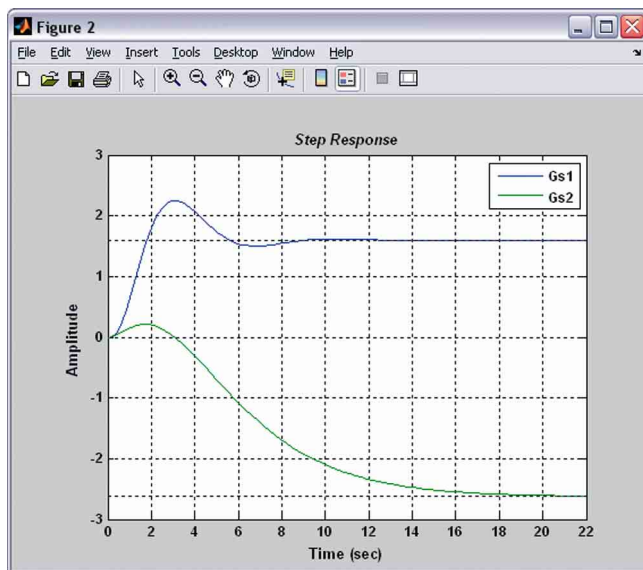
Command Window	Command Window
Zero/pole/gain: 12 (s+0.6) ----- (s+1.5) (s+3) (s^2 + s + 1)	Zero/pole/gain: 0.7 (s-0.6) ----- (s+1) (s+0.4)^2
B1 = 0 0 0 12.0000 7.2000	B2 = 0 0 0.7000 -0.4200
A1 = 1.0000 5.5000 10.0000 9.0000 4.5000	A2 = 1.0000 1.8000 0.9600 0.1600
K1 = 1.6000	K2 = -2.6250
Transfer function: 12 s + 7.2 ----- s^4 + 5.5 s^3 + 10 s^2 + 9 s + 4.5	Transfer function: 0.7 s - 0.42 ----- s^3 + 1.8 s^2 + 0.96 s + 0.16
>>	>>

Obr. 6.3-2: Definice přenosu systému

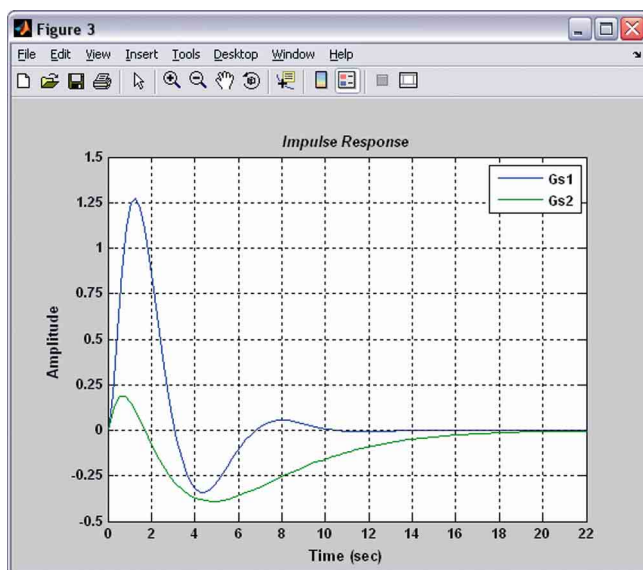
Pomocí příkazu **pzmap(Gs)** můžeme přehledně zobrazit rozložení předem definovaných pólů a nul v komplexní rovině. Pokud chceme zobrazit rozložení kořenů více systémů, zapíšeme příkaz ve tvaru **pzmap(Gs1, Gs2, ...)**. Póly přenosu jsou vyznačeny křížky, nuly přenosu pak kroužky. Systémy lze navzájem odlišit použitím různých barev. Výsledný graf (obr. 6.3-3) můžeme doplnit o legendu.



Obr. 6.3-3: Rozložení pólů a nul

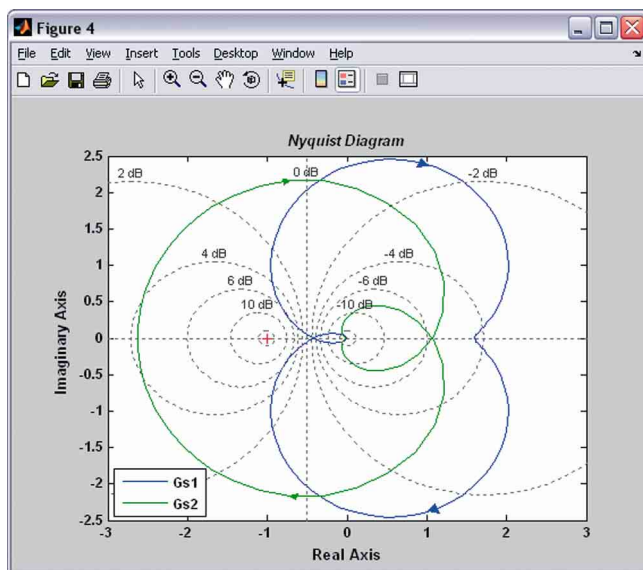


Obr. 6.3-4: Přechodové charakteristiky



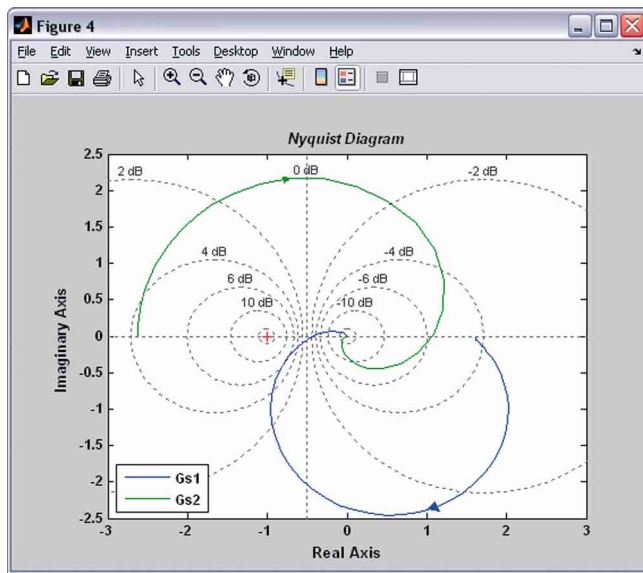
Obr. 6.3-5: Váhové (impulsní) charakteristiky

Přechodové charakteristiky (obr. 6.3-4) obou systémů získáme zadáním příkazu **step(Gs)**, charakteristiky váhové (obr. 6.3-5) pak zadáním **impulse(Gs)**. Vytvářený skript doplníme o zobrazení frekvenčních charakteristik v Gaussově komplexní rovině a v logaritmických souřadnicích (obr. 6.3-6 až 6.3-8).

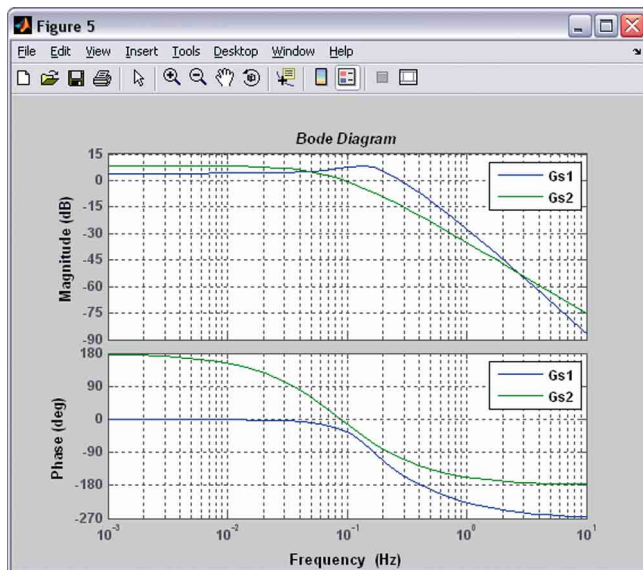


Obr. 6.3-6: Frekvenční charakteristiky v komplexní rovině (včetně větví pro záporné frekvence)

Frekvenční charakteristiku dynamického systému v komplexní rovině získáme zadáním příkazu **nyquist(Gs)** případně **nyquist(Gs, {Wmin, Wmax})**, chceme-li definovat rozsah frekvencí (v rad/s). Charakteristika je vykreslena včetně větve pro záporné frekvence, která je souměrná podle reálné osy s větví pro frekvence kladné, viz obr. 6.3-6.

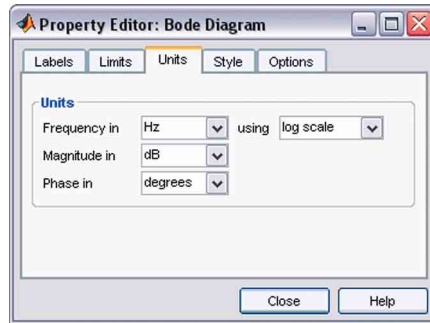


Obr. 6.3-7: Frekvenční charakteristiky v komplexní rovině



Obr. 6.3-8: Amplitudové a fázové frekvenční charakteristiky v logaritmických souřadnicích

Obdobně vykreslíme i frekvenční charakteristiky v logaritmických souřadnicích uvedené na obr. 6.3-8, tj. amplitudovou a fázovou, které bývají často označovány jako tzv. BODEho charakteristiky. K jejich zobrazení použijeme příkaz **bode(Gs)** či **bode(Gs,{Wmin,Wmax})**, chceme-li definovat také frekvenční rozsah.



Obr. 6.3-9: Editor vlastností grafu

Prostřednictvím plovoucího menu, které je přístupné po stisku pravého tlačítka myši nad plochou grafu, můžeme měnit vzhled všech popisovaných charakteristik. Pokud např. nechceme zobrazit sdruženou část frekvenční charakteristiky v komplexní rovině (obr. 6.3-6), odznačíme položku plovoucího menu **Show → Negative Frequencies**. Výsledný průběh je uveden na obr. 6.3-7. V dialogovém okně (obr. 6.3-9), které je přístupné zvolíme-li položku **Properties...**, lze měnit popisky, rozsahy os, použité jednotky (v tomto případě rad/s nebo Hz), fonty apod.

Na závěr poznamenejme, že druhý definovaný systém, na obr. 6.3-2 vlevo (je označen indexem 2), je tzv. **systém s neminimální fází**. Tento systém má nestabilní nulu (kořen čitatele), tj. nulu v pravé polorovině komplexní roviny, viz obr. 6.3-3. Charakter systému je patrný i z přechodové charakteristiky (obr. 6.3-4), z frekvenční charakteristiky v komplexní rovině (obr. 6.3-7) a zejména pak z fázové frekvenční charakteristiky (obr. 6.3-8). Fáze se zde pro nízké frekvence blíží ke 180° , není tedy minimální (záporná).

6.4 Frekvenční charakteristiky dynamických systémů 2. řádu

V teorii automatického řízení se k posouzení vlastností regulačních obvodů i jejich jednotlivých členů někdy používají tzv. frekvenční charakteristiky. Pokud na vstup systému zavedeme harmonické kmity $u(t)$ o určité frekvenci, tak po určité době nutné k utlumení přechodového děje vzniknou na výstupu tohoto systému harmonické kmity $y(t)$ se stejnou frekvencí, ale obecně s jinou amplitudou a s fázovým posunem. Utlumení přechodového děje nastane vždy, pokud je systém stabilní.

Jelikož lze harmonický pohyb znázornit v komplexní rovině vektorem s konstantní amplitudou, který se otáčí v kladném smyslu (ve směru hodinových ručiček) úhlovou rychlostí ω , lze rovnici vstupních kmitů zapsat ve tvaru

$$\mathbf{u} = u_0 e^{j\omega t}$$

a rovnici výstupních kmitů

$$\mathbf{y} = y_0 e^{j(\omega t + \varphi)}$$

kde u_0 a y_0 jsou amplitudy vstupních a výstupních kmitů a φ je fázový posuv výstupních kmitů vůči vstupním. S výjimkou derivačních členů (systémů s neminimální fází) je $\varphi < 0$, tj. výstupní kmitý jsou zpožděny za vstupními.

Vytvoříme-li poměr výstupních kmitů ke vstupním, dostáváme

$$G(j\omega) = \frac{\mathbf{y}}{\mathbf{u}} = \frac{y_0}{u_0} e^{j\varphi}$$

Komplexní funkce $G(j\omega)$ udává, jak se mění výstupní kmitý $y(t)$ co do amplitudy a fáze v závislosti na frekvenci, udržujeme-li amplitudu vstupních kmitů konstantní. Modulem funkce $G(j\omega)$ je poměr amplitud vstupních a výstupních kmitů

$$|G(j\omega)| = \frac{y_0}{u_0} = A(\omega)$$

a jejím argumentem je fázový posuv

$$\arg[G(j\omega)] = \varphi(\omega)$$

Funkce $G(j\omega)$ se nazývá **frekvenční přenos**. Frekvenční přenos získáme také z přenosu obrazového dosazením za $s = j\omega$, tedy

$$G(j\omega) = \frac{b_m(j\omega)^m + \dots + b_1(j\omega) + b_0}{a_n(j\omega)^n + \dots + a_1(j\omega) + a_0}$$

Výrazy v čitateli a ve jmenovateli přenosu $G(j\omega)$ rozdělíme na reálnou a imaginární část, takže bude

$$G(j\omega) = \frac{e(\omega) + jf(\omega)}{c(\omega) + jd(\omega)}$$

kde

$$c(\omega) = a_0 - a_2 \omega^2 + a_4 \omega^4 - \dots$$

$$d(\omega) = a_1 \omega - a_3 \omega^3 + a_5 \omega^5 - \dots$$

$$e(\omega) = b_0 - b_2 \omega^2 + b_4 \omega^4 - \dots$$

$$f(\omega) = b_1 \omega - b_3 \omega^3 + b_5 \omega^5 - \dots$$

Jestliže čitatele i jmenovatele přenosu $G(j\omega)$ vynásobíme $[c(\omega) - jd(\omega)]$, tedy výrazem komplexně sdruženým (konjugovaným) ke jmenovateli přenosu, můžeme $G(j\omega)$ zapsat přímo jako komplexní číslo – ve složkovém popř. v exponenciálním tvaru

$$G(j\omega) = P(\omega) + jQ(\omega) = A(\omega)e^{j\varphi(\omega)}$$

Z výše uvedeného dále plyne, že

$$P(\omega) = A(\omega) \cos \varphi(\omega)$$

$$Q(\omega) = A(\omega) \sin \varphi(\omega)$$

$$A(\omega) = \sqrt{P^2(\omega) + Q^2(\omega)}$$

$$\varphi(\omega) = \arctg \frac{Q(\omega)}{P(\omega)}$$

Pro jednotlivé složky přenosu $G(j\omega)$ zavádíme tyto názvy:

$P(\omega)$ – reálná část frekvenční charakteristiky,

$Q(\omega)$ – imaginární část frekvenční charakteristiky,

$A(\omega)$ – amplitudová frekvenční charakteristika,

$\varphi(\omega)$ – fázová frekvenční charakteristika.

Jak vlastní funkci $G(j\omega)$, tak i funkce $A(\omega)$ a $\varphi(\omega)$ lze znázornit graficky a nazýváme je pak **frekvenční charakteristiky**. Frekvenční charakteristika systému v komplexní rovině je geometrické místo, které opíše koncový bod vektoru $G(j\omega)$ v komplexní rovině při změně frekvence vstupního harmonického signálu v rozsahu $-\infty \leq \omega \leq +\infty$. Tato charakteristika se někdy označuje jako tzv. **amplitudo-fázová** a je též grafickým zobrazením přenosu $G(j\omega)$ v komplexní rovině. Větev charakteristiky pro záporné frekvence je souměrná podle reálné osy s větví pro frekvence kladné a zpravidla se neuvádí. (V MATLABu je ovšem tato větev implicitně vždy vykreslována.)

Amplitudovou $A(\omega)$ a **fázovou** $\varphi(\omega)$ frekvenční charakteristiku je vhodné znázorňovat v logaritmických souřadnicích. Z toho důvodu jsou často tyto charakteristiky označovány jako **logaritmické frekvenční charakteristiky**. U amplitudové charakteristiky bývá zvykem vynášet absolutní hodnotu (modul) velikosti přenosu v decibelech. Hodnotu nějaké veličiny v decibelech [dB] vypočteme jako dvacetinásobek dekadického logaritmu této veličiny. Na logaritmické stupnici pro frekvenci ω nazýváme interval mezi určitou frekvencí a jejím desetinásobkem **dekádou**. V praxi se často přesné charakteristiky nahrazují asymptotami. U asymptot pak hovoříme o poklesu resp. vzrůstu vztaženém právě na dekádu.

V dalším textu se omezíme na **stabilní kmitavé systémy 2. řádu**. Budeme uvažovat přenos ve tvaru

$$G_S(j\omega) = K \frac{\omega_n^2}{(j\omega)^2 + 2\xi\omega_n j\omega + \omega_n^2}$$

který získáme z obrazového přenosu uvedeného v kapitole 6.2 dosazením za $s = j\omega$. Přenos lze dále upravit

$$G_S(j\omega) = K \frac{\omega_n^2}{(\omega_n^2 - \omega^2) + j2\xi\omega_n\omega} = K \frac{1}{(1 - \varpi^2) + j2\xi\varpi}$$

kde $\varpi = \omega/\omega_n$ je tzv. normalizovaná frekvence.

Z uvedeného přenosu vypočteme amplitudo-fázovou frekvenční charakteristiku

$$G_s(j\omega) = K \frac{1 - \varpi^2}{(1 - \varpi^2)^2 + 4\xi^2\varpi^2} - jK \frac{2\xi\varpi}{(1 - \varpi^2)^2 + 4\xi^2\varpi^2}$$

a logaritmickou amplitudovou charakteristiku

$$\begin{aligned} 20 \log A(\omega) &= 20 \log K - 20 \log \sqrt{(1 - \varpi^2)^2 + 4\xi^2\varpi^2} = \\ &= 20 \log K - 10 \log [(1 - \varpi^2)^2 + 4\xi^2\varpi^2] \end{aligned}$$

Přesný průběh charakteristiky nahradíme asymptotami. Rovnice první asymptoty pro $\omega \rightarrow 0$ je

$$20 \log A(\omega) \approx 20 \log K - 10 \log 1 = 20 \log K$$

Rovnice druhé asymptoty pro $\omega \rightarrow \infty$, kdy zřejmě $[(1 - \varpi^2)^2 + 4\xi^2\varpi^2] \rightarrow \varpi^4$, je

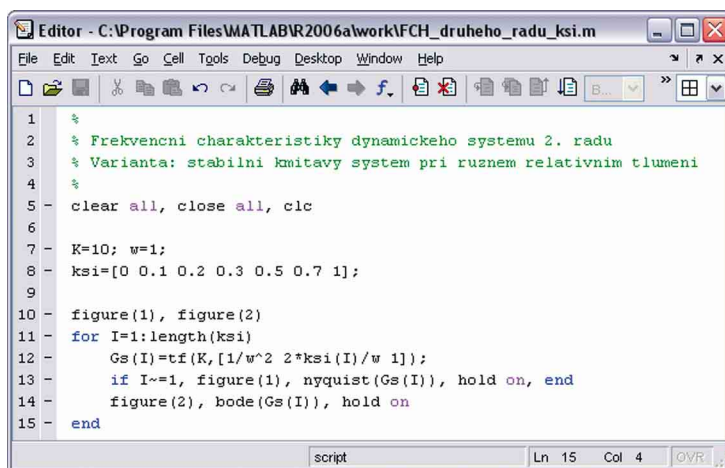
$$20 \log A(\omega) \approx 20 \log K - 10 \log \varpi^4 = 20 \log K - 40 \log \varpi$$

Pro zakreslení přesných charakteristik je nutné znát chybu, které se dopouštíme při náhradě asymptotami. Chyba je závislá na koeficientu ξ . Podle [17] u dynamických systémů, u kterých ξ leží v intervalu $\langle 0,4; 0,7 \rangle$ chyba nepřesáhne hodnotu 3 db.

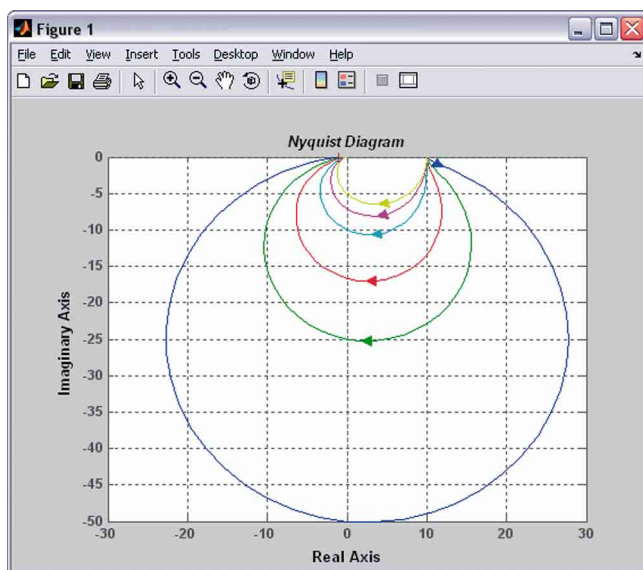
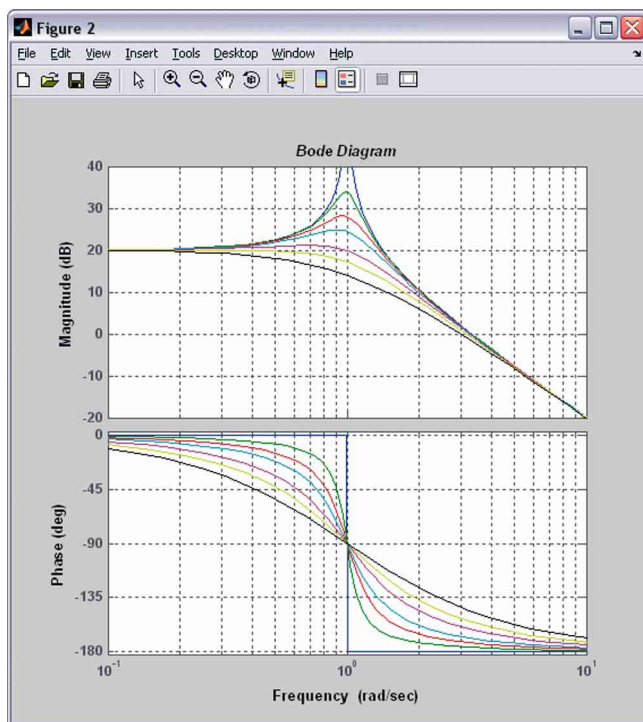
Zbývá určit fázovou charakteristiku. Ta je dána vztahem

$$\varphi(\omega) = -\arg[(1 - \varpi^2) + j2\xi\varpi] = -\arctg \frac{2\xi\varpi}{1 - \varpi^2}$$

Uvedený postup je poměrně složitý. Pokud ale k zobrazení frekvenčních charakteristik využijeme MATLAB resp. jeho Control System Toolbox, situace se velmi zjednoduší. Zcela odpadne např. konstrukce asymptot u amplitudové charakteristiky.

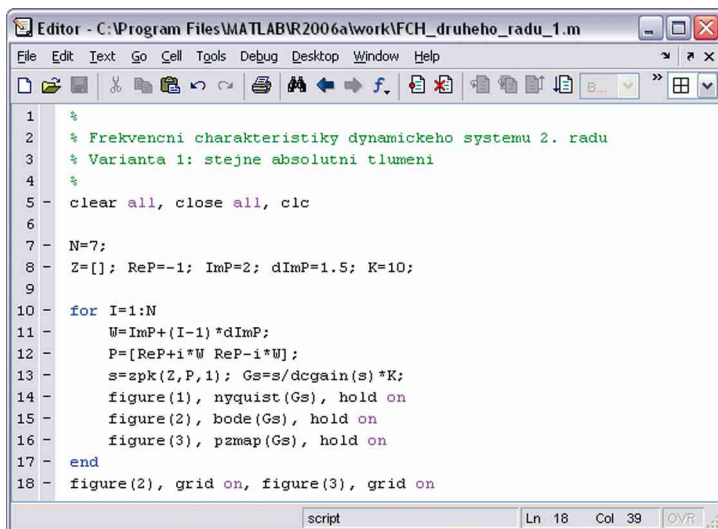


Obr. 6.4-1: Skript pro vykreslení frekvenčních charakteristik


 Obr. 6.4-2: Frekvenční charakteristika v komplexní rovině pro různé hodnoty ξ

 Obr. 6.4-3: Amplitudová a fázová frekvenční charakteristika v logaritmických souřadnicích pro různé hodnoty ξ

Na obr. 6.4-1 je uveden jednoduchý skript, který vykreslí frekvenční charakteristiky v komplexní rovině (obr. 6.4-2) a frekvenční charakteristiky v logaritmických souřadnicích (obr. 6.4-3). Parametrem jednotlivých křivek je relativní tlumení ξ , které je postupně měněno v intervalu $(0; 1)$. Zesílení systému je $K = 10$. Je možné měnit vektor hodnot tlumení ξ , pro které mají být charakteristiky vykresleny. Ve skriptu jsou použity příkazy **nyquist** a **bode** Control System Toolboxu, které byly podrobněji popsány již v kapitole 6.3.

Sledujme nyní vliv polohy pólů přenosu systému na průběh frekvenčních charakteristik. Podobně jako v kapitole 6.2 budeme nejprve uvažovat komplexně sdružené póly, které mají stejnou reálnou část (tj. absolutní tlumení) a rozdílnou imaginární. Grafické znázornění jejich možné polohy v komplexní rovině a tomu odpovídající přechodové charakteristiky byly již uvedeny na obr. 6.2-5. S využitím Control System Toolboxu můžeme vytvořit jednoduchý skript pro vykreslení frekvenčních charakteristik takto definovaných stabilních kmitavých systémů 2. řádu, viz obr. 6.4-4. Pro sestavení obrazového přenosu ze zadaných pólů a zesílení byl použit příkaz **zp**. Frekvenční charakteristiky při shodném zesílení $K = 10$ a při zachování stejného absolutního tlumení α jsou na obr. 6.4-5 a 6.4-6.



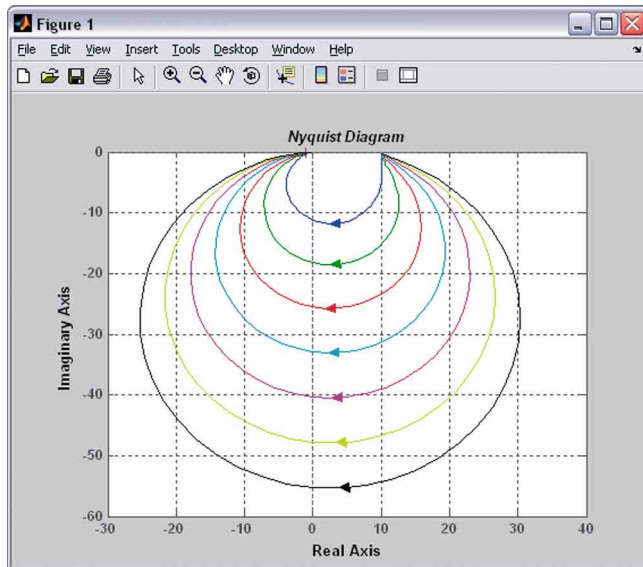
```

1 %
2 % Frekvenční charakteristiky dynamického systému 2. řádu
3 % Varianta 1: stejné absolutní tlumení
4 %
5 clear all, close all, clc
6
7 N=7;
8 Z=[]; ReP=-1; ImP=2; dImP=1.5; K=10;
9
10 for I=1:N
11     W=ImP+(I-1)*dImP;
12     P=[ReP+i*W ReP-i*W];
13     s=zpk(Z,P,1); Gs=s/dcgain(s)*K;
14     figure(1), nyquist(Gs), hold on
15     figure(2), bode(Gs), hold on
16     figure(3), pzmap(Gs), hold on
17 end
18 figure(2), grid on, figure(3), grid on

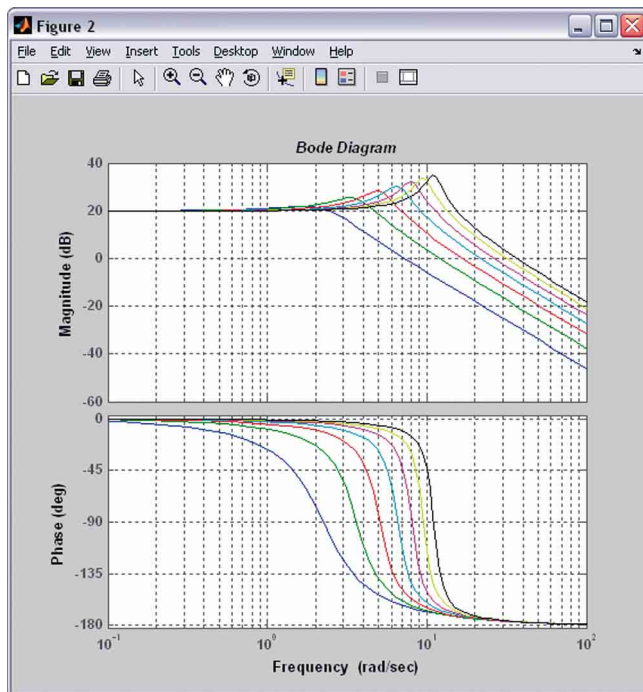
```

Obr. 6.4-4: Skript pro vykreslení frekvenčních charakteristik systému při stejném absolutním tlumení α

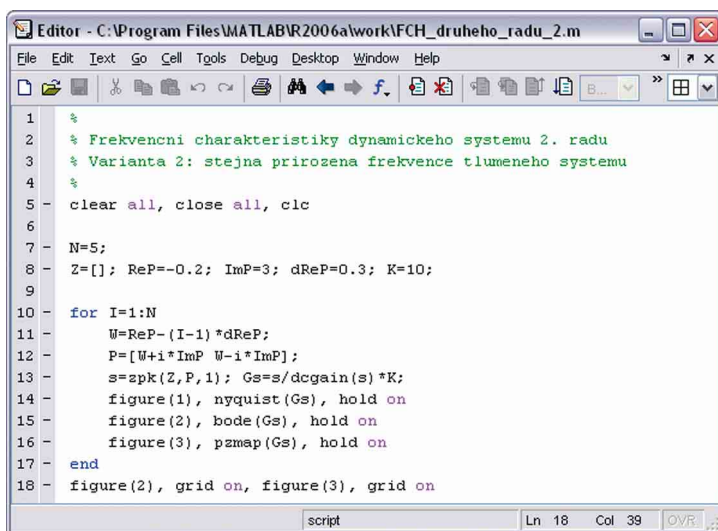
Pomocí skriptu na obr. 6.4-7 můžeme obdobným způsobem měnit reálnou složku pólů při zachování složky imaginární (tj. při zachování přirozené frekvence tlumeného systému ω_d). Grafické znázornění možné polohy pólů v komplexní rovině a odpovídající přechodové charakteristiky jsou uvedeny na obr. 6.2-6. Frekvenční charakteristiky při shodném zesílení systému $K = 10$ a při stejné přirozené frekvenci tlumeného systému ω_d jsou na obr. 6.4-8 a 6.4-9.



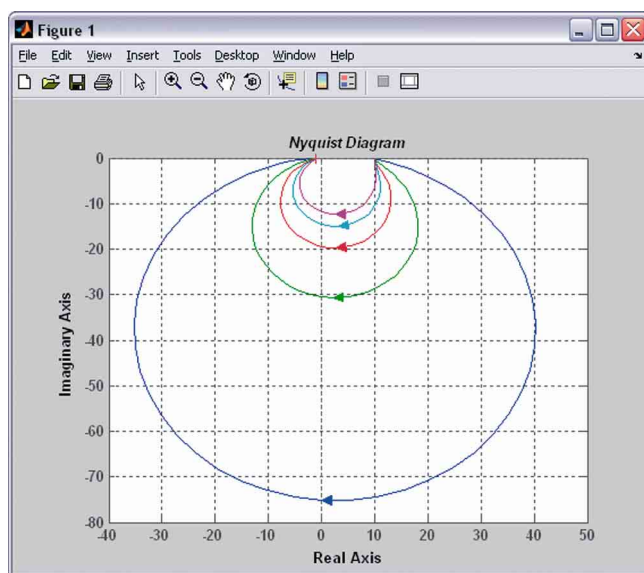
Obr. 6.4-5: Frekvenční charakteristiky systému v komplexní rovině při stejném absolutním tlumení α



Obr. 6.4-6: Amplitudové a fázové frekvenční charakteristiky v logaritmických souřadnicích při stejném absolutním tlumení α



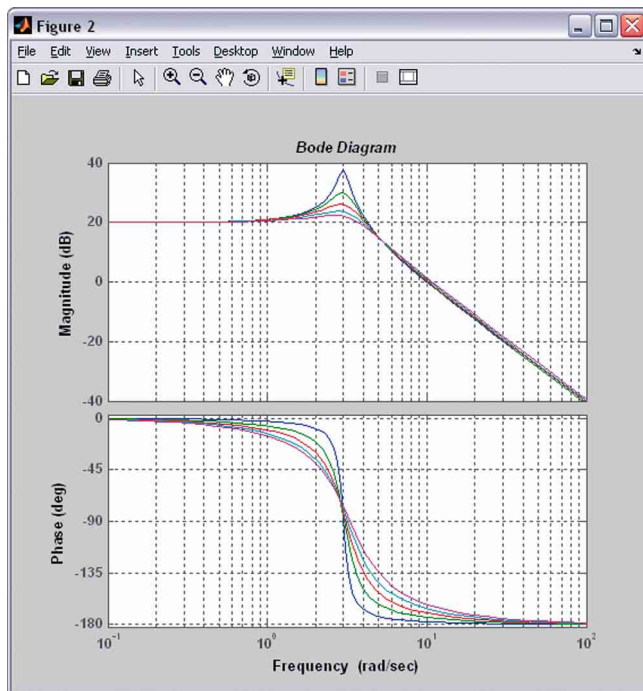
Obr. 6.4-7: Skript pro vykreslení frekvencních charakteristik systému při stejné přirozené frekvenci ω_d



Obr. 6.4-8: Frekvencní charakteristiky systému v komplexní rovině při stejné přirozené frekvenci ω_d

Poslední variantou rozložení pólů přenosu kmitavého systému 2. řádu, kterou budeme simulovat, je rozložení pólů v komplexní rovině tak, aby výsledné systémy měly vždy stejné relativní tlumení ξ . Připomeňme, že konstantní hodnotě relativního tlumení odpovídají dvě polopřímky vedené z počátku souřadného systému, viz obr. 6.2-2. Grafické znázornění možné

polohy pólů v komplexní rovině a odpovídající přechodové charakteristiky jsou na obr. 6.2-8. K vykreslení frekvenčních charakteristik použijeme skript na obr. 6.4-10.

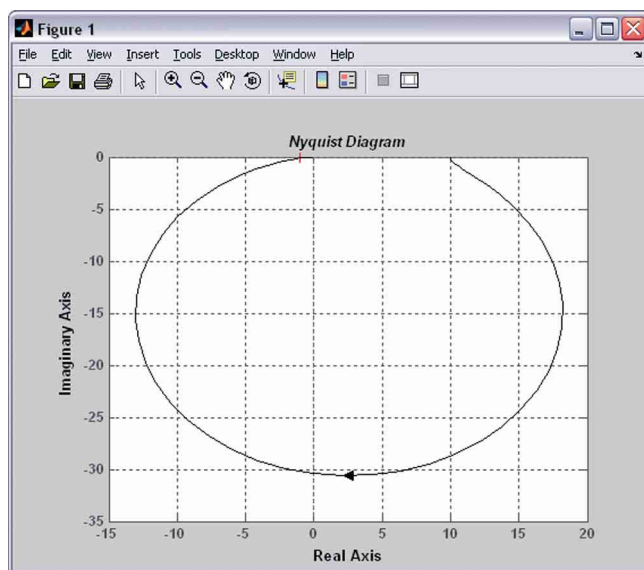


Obr. 6.4-9: Amplitudové a fázové frekvenční charakteristiky v logaritmických souřadnicích při stejné přirozené frekvenci ω_d

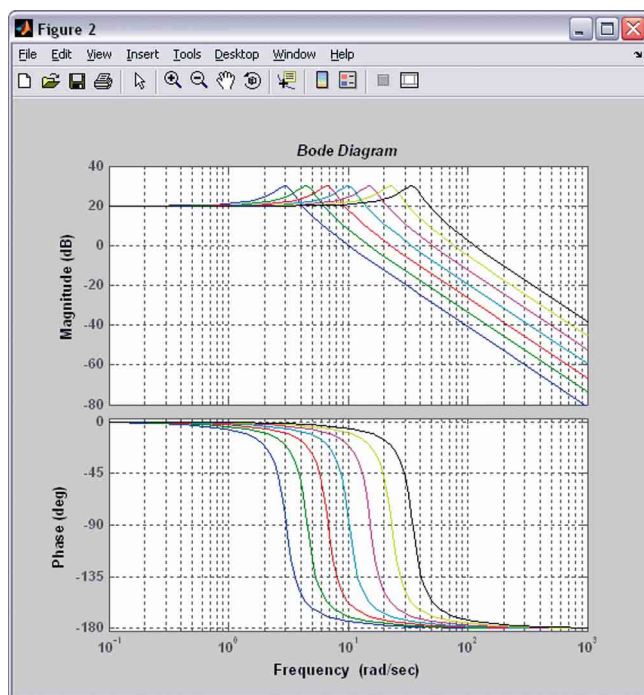
```

1  %
2  % Frekvenční charakteristiky dynamického systému 2. řádu
3  % Varianta 3: stejné relativní tlumení
4  %
5  clear all, close all, cld
6
7  N=7;
8  Z=[]; ReP=-0.5; ImP=3; p=1; mp=1.5; K=10;
9
10 for I=1:N
11     W=p*(ReP+i*ImP); p=p*mp;
12     P=[W conj(W)];
13     s=zpk(Z,P,1); Gs=s/dcgain(s)*K;
14     figure(1), nyquist(Gs), hold on
15     figure(2), bode(Gs), hold on
16     figure(3), pzmap(Gs), hold on
17 end
18 figure(2), grid on, figure(3), grid on
    
```

Obr. 6.4-10: Skript pro vykreslení frekvenčních charakteristik systému při stejném relativním tlumení ξ



Obr. 6.4-11: Frekvenční charakteristiky systému v komplexní rovině při stejném relativním tlumení ξ

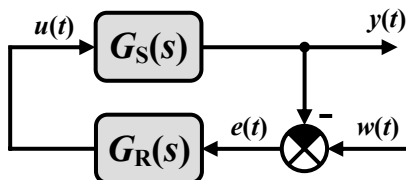


Obr. 6.4-12: Amplitudové a fázové frekvenční charakteristiky v logaritmických souřadnicích při stejném relativním tlumení ξ

Při stejném relativním tlumení mají všechny definované systémy totožné frekvenční charakteristiky v komplexní rovině (obr. 6.4-11). Na amplitudových a fázových frekvenčních charakteristikách (obr. 6.4-12) je patrný jejich vzájemný posun při zachování tvaru křivek.

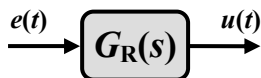
6.5 Přechodové charakteristiky PID regulátoru

Regulátor je ve své podstatě uměle vytvořený dynamický prvek zařazený do zpětné vazby k regulované soustavě (k technologickému procesu, který se snažíme řídit). Změna parametrů regulátoru nám umožňuje ovlivňovat dynamické vlastnosti celého uzavřeného regulačního obvodu, viz obr. 6.5-1.



Obr. 6.5-1: Blokové schéma regulačního obvodu

Vstupem do regulátoru je regulační odchylka $e(t)$ vyjádřená rozdílem $e(t) = w(t) - y(t)$, kde $y(t)$ představuje regulovanou a $w(t)$ řídicí veličinu.



Obr. 6.5-2: Blok regulátoru

Vlastnosti regulátoru typu PID mohou být poněkud zjednodušeně popsány vztahem

$$u(t) = r_0 e(t) + r_1 \int_0^t e(\tau) d\tau + r_2 \frac{de(t)}{dt}$$

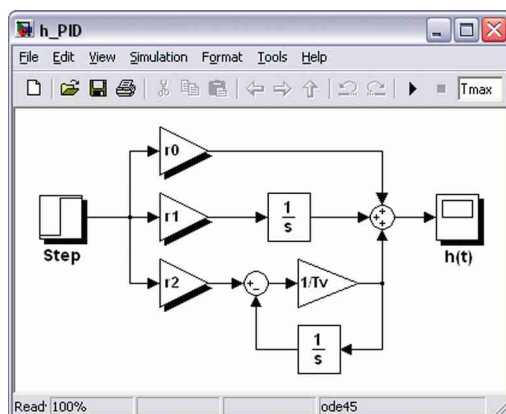
Přenos regulátoru je

$$G_R(s) = r_0 + \frac{r_1}{s} + r_2 s$$

Derivační složku lze u reálných průmyslových regulátorů realizovat pouze přibližně s dynamickým zpožděním prvního (nebo někdy i vyššího) řádu. Obrazový přenos reálného regulátoru nabývá pak poněkud komplikovanějšího tvaru (zde pro zpoždění 1. řádu)

$$G_R(s) = r_0 + \frac{r_1}{s} + \frac{r_2 s}{T_v s + 1}$$

kde T_v je **časová konstanta zpožďovacího členu**. Takto modifikovaný regulátor se pak často označuje jako **PID regulátor s aproximovanou derivací**. Parametry r_0 , r_1 , r_2 jsou stavitelné parametry regulátoru, po řadě jeho proporcionální, integrační a derivační složky.



Obr. 6.5-3: Realizace PID regulátoru v Simulinku

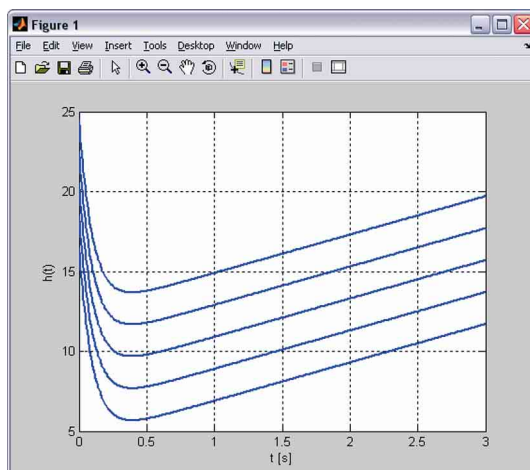
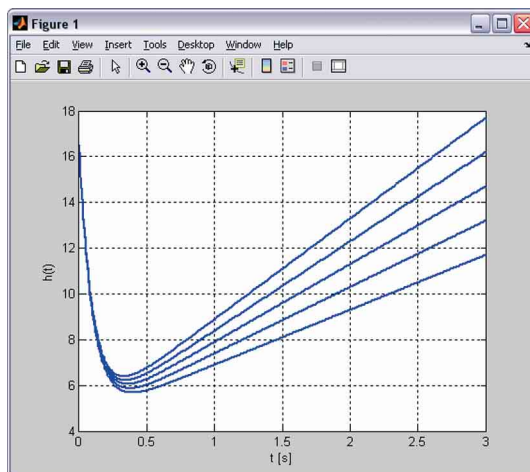
```

1  %
2  % Prechodova charakteristika PID regulatoru
3  %
4  clear all, close all, cld
5  % Pole parametru simulace
6  % Nastaveni parametru systemu
7  r0=4.5; % proporcionalni zesileni
8  r1=2.4; % integracni zesileni
9  r2=1.2; % derivacni zesileni
10 Tv=0.1; % parazitni casova konstanta (Tv ruzna od nuly)
11 % Vyber prislusne variace parametru regulatoru
12 J0=r0; Krok=2; Pocet=5; Q=0; % variace parametru r0
13 %J0=r1; Krok=0.5; Pocet=5; Q=1; % variace parametru r1
14 %J0=r2; Krok=0.5; Pocet=5; Q=2; % variace parametru r2
15 %J0=Tv; Krok=0.1; Pocet=5; Q=3; % variace parametru r0
16 % (znak % blokuje provedeni prikazu uvedenych za nim az do konce radku)
17 %
18 Tmax=3; % doba simulace
19 % Konec pole parametru simulace
20
21 h_PID % otevre okno se simulacnim schematem
22 for J=J0:Krok:J0+Krok*(Pocet-1) % zacatek cyklu s menicim se parametrem
23     switch Q % prirazeni podle priznaku vybrane variace
24         case 0, r0=J;
25         case 1, r1=J;
26         case 2, r2=J;
27         otherwise, Tv=J;
28     end
29     sim('h_PID') % spousti simulaci "h_prvniho_radu.mdl"
30     plot(t,h(:,2),'LineWidth',2) % vykresleni prechodove charakteristiky
31     hold on % umozni vykreslovani do stejneho okna
32     %pause % preruseni po vykresleni jedne krivky
33     end % konec cyklu % (pokracovani libovolnou klavesou)
34     grid on % vykresleni mriscky
35     xlabel('t [s]'), ylabel('h(t)') % popis os

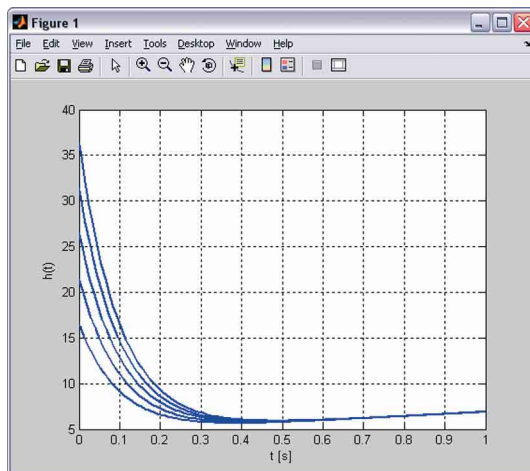
```

Obr. 6.5-4: Skript pro řízení simulačního výpočtu

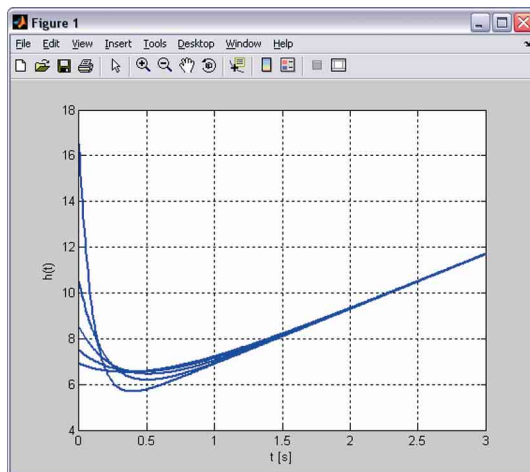
Přechodová charakteristika $h(t)$ je obecně reakce dynamického systému vybuzeného z klidu (nulové počáteční podmínky) jednotkovou skokovou změnou vstupního budicího signálu. Úkolem je simulovat konkrétně přechodovou charakteristiku regulátoru a sledovat vliv jednotlivých parametrů na její průběh. Příkladem simulačního modelu regulátoru v prostředí Simulink může být např. model na obr. 6.5-3. Zvolíme-li jako budicí signál jednotkový skok a parametry PID regulátoru prostřednictvím hodnot zesílení příslušných zesilovačů, můžeme pomocí zobrazovače (*Scope*) sledovat po provedení simulaci průběh přechodové charakteristiky. V uvedeném modelu ukládáme časový průběh $h(t)$ do pracovního prostoru (také pomocí *Scope*). Můžeme ho pak následně např. vykreslit do grafu, editovat a pro další účely přenést do textového editoru nebo jiným způsobem ve formě pole dále zpracovat.

Obr. 6.5-5: Variace parametru r_0 Obr. 6.5-6: Variace parametru r_1

Pro účely vykreslení sítě přechodových charakteristik odpovídajících měnícímu se parametru do jednoho obrázku byl vytvořen jednoduchý skript (obr. 6.5-4), kterým lze řídit výpočet opakované simulace. Struktura i ovládání tohoto skriptu jsou zřejmé z vepsaných komentářů. Přechodové charakteristiky PID regulátoru při variaci parametrů r_0 , r_1 , r_2 a T_v jsou na obr. 6.5-5 až 6.5-8.

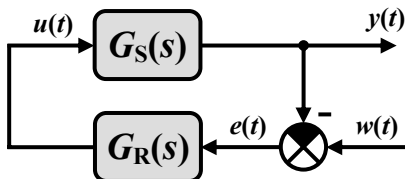

 Obr. 6.5-7: Variace parametru r_2

Vliv jednotlivých parametrů regulátoru lze pomocí uvedeného skriptu vyšetřovat i na jednodušších strukturách regulátoru P ($r_1 = r_2 = 0$), I ($r_0 = r_2 = 0$), D ($r_0 = r_1 = 0$) a PI ($r_2 = 0$) a PD ($r_1 = 0$).


 Obr. 6.5-8: Variace parametru T_v

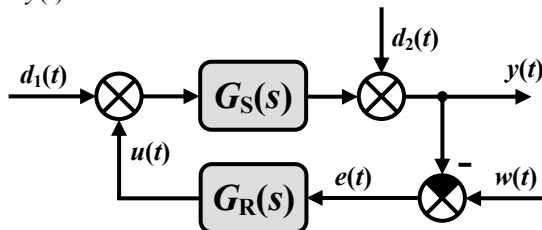
6.6 Regulační obvod se spojitým PID regulátorem

Hlavní funkcí regulátoru s přenosem $G_R(s)$ ve zpětnovazebním regulačním obvodu na obr. 6.4-1 je působit na technologický proces (tzv. regulovanou soustavu) s přenosem $G_S(s)$ vhodně tak, aby celý uzavřený obvod měl dobré statické i dynamické vlastnosti.



Obr. 6.4-1: Blokové schéma regulačního obvodu

Regulovaná veličina $y(t)$ musí s přijatelnou přesností reagovat na požadavek změny její hodnoty vyjádřený žádanou hodnotou regulované veličiny $w(t)$ (řídící veličinou). Kromě této základní funkce ještě regulátor eliminuje i vliv většinou nežádoucích vlivů, tzv. poruch, které také vstupují do obvodu. Vstupem do regulátoru je regulační odchylka $e(t)$ vyjádřená rozdílem $e(t) = w(t) - y(t)$.



Obr. 6.4-2: Regulační obvod s poruchami

V blokovém schéma na obr. 6.4-2 jsou vyznačeny dva nejčastější typy poruch. Porucha na vstupu do regulované soustavy $d_1(t)$, někdy označovaná jako porucha na akční veličině $u(t)$, a na jejím výstupu $d_2(t)$.

Chování regulačního obvodu můžeme ovlivnit parametry regulátoru $G_R(s)$. V technické praxi jsou nejčastěji používány spojitý regulátory typu **PID** (Proporcionálně Integračně Derivační), jejichž diferenciální rovnice je obvykle ve tvaru

$$u(t) = r_0 e(t) + r_1 \int_0^t e(\tau) d\tau + r_2 \frac{de(t)}{dt}$$

kde r_0 , r_1 a r_2 představují proporcionální, integrační resp. derivační zesílení regulátoru, nebo případně ve tvaru

$$u(t) = r_0 [e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt}]$$

Vzájemný vztah mezi konstantami obou uvedených variant regulátoru je $r_1 = I = r_0 / T_i$ a $r_2 = D = r_0 T_d$, kde T_i je integrační a T_d derivační časová konstanta.

Nastavením parametrů regulátoru r_0, r_1, r_2 resp. r_0, T_i, T_d můžeme ovlivnit chování celého regulačního obvodu, nevhodným nastavením může však dojít i k jeho špatné funkci nebo dokonce k jeho nestabilitě.

Pro určení vhodné kombinace hodnot parametrů regulátoru (tzv. optimální seřízení regulátoru) existuje několik metod a přístupů lišících se zejména tím, jak posuzují kvalitu dynamického chování celého uzavřeného regulačního obvodu.

Sestavme nyní simulační model regulačního obvodu složeného z regulované soustavy třetího řádu popsané diferenciální rovnicí

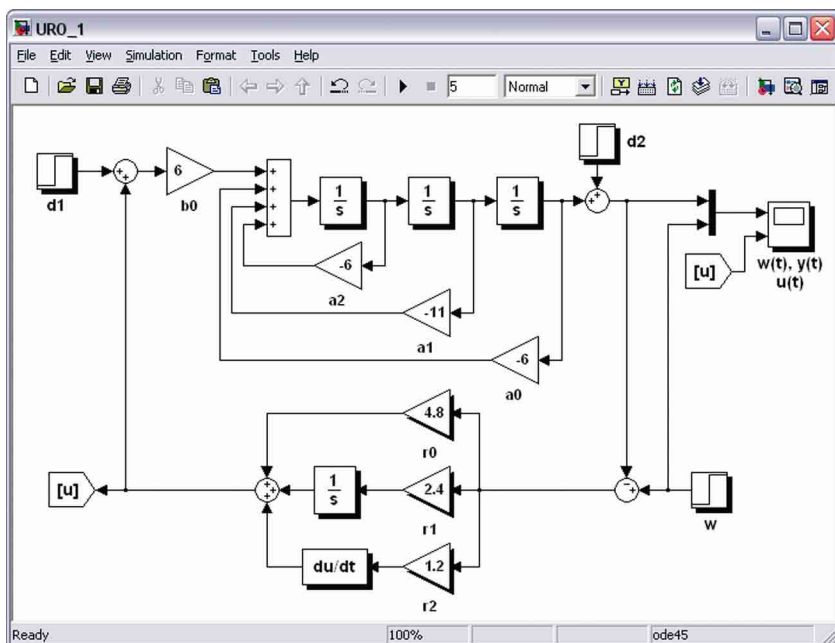
$$y'''(t) + 6y''(t) + 11y'(t) + 6y(t) = 6u(t)$$

resp. obrazovým přenosem

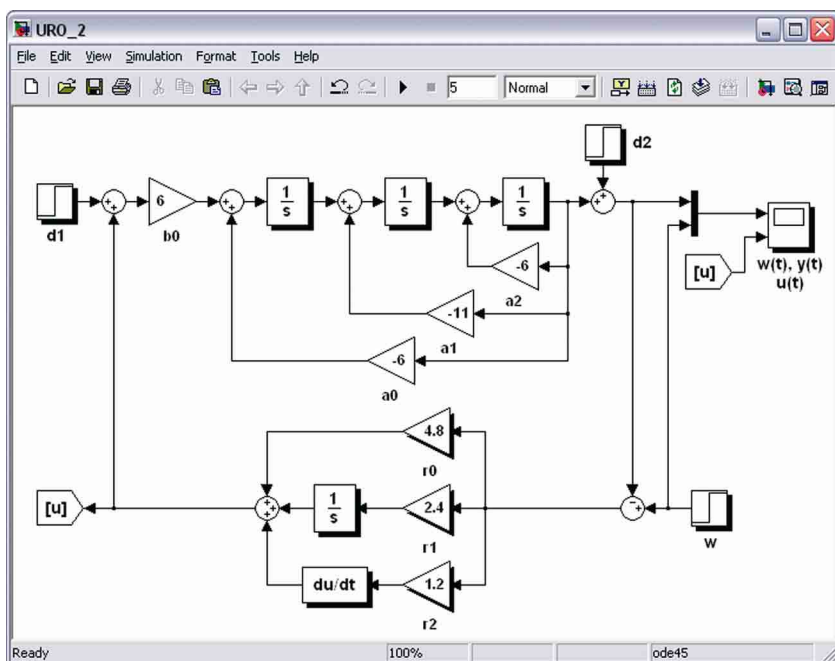
$$G_S(s) = \frac{6}{s^3 + 6s^2 + 11s + 6}$$

a regulátoru typu PID. Ověřte jeho statické i dynamické chování pro různé hodnoty parametrů regulátoru (mohou být i nulové) a různé poruchy $d_1(t)$, $d_2(t)$ a změny $w(t)$.

Model regulačního obvodu lze realizovat různými způsoby. Regulovaná soustava může být modelována na základě příslušné diferenciální rovnice, jejíž řešení může být např. realizováno pomocí **metody snižování řádu derivace** (obr. 6.4-3) či **metody postupné integrace** (obr. 6.4-4). Další možností je realizace pomocí **obrazového přenosu** (obr. 6.4-5). K realizaci regulátoru je možné použít standardní bloky Simulinku, případně přímo blok *PID Controller* (knihovna *Simulink Extras / Additional Linear*).

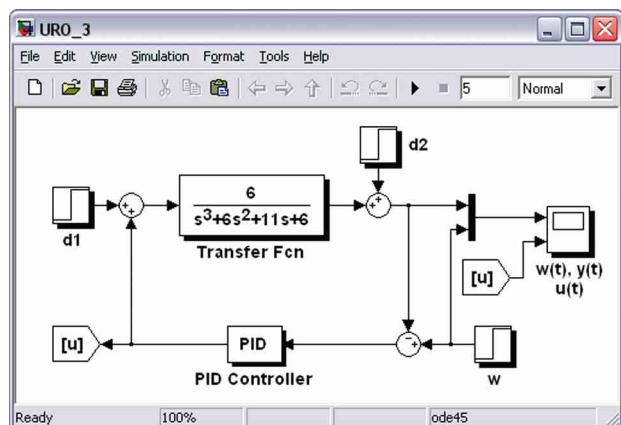


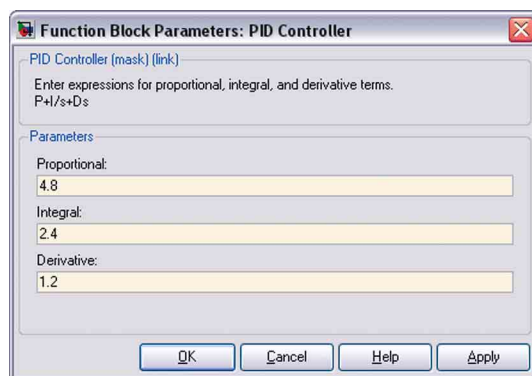
Obr. 6.4-3: Realizace regulačního obvodu v Simulinku – metoda snižování řádu derivace



Obr. 6.4-4: Realizace regulačního obvodu v Simulinku – metoda postupné integrace

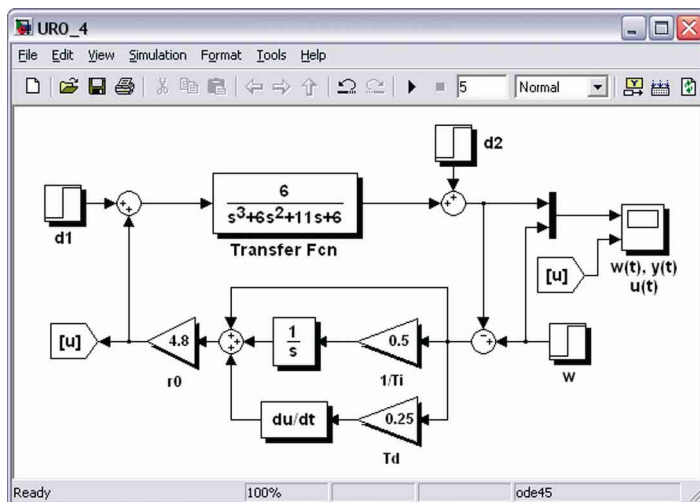
Blok *PID Controller* představuje běžnou spojitou variantu PID regulátoru s přenosem v paralelním tvaru $G_R(s) = P + I/s + Ds$, ve kterém konstanty P , I a D regulátoru odpovídají zesílením r_0 , r_1 a r_2 používaným ve výše uvedených modelech. Tyto konstanty definujeme přímo prostřednictvím okna parametrů bloku, které je uvedeno na obr. 6.4-6. Nevýhodou použití bloku PID regulátoru je ale nemožnost modifikace struktury PID algoritmu a ztráta možnosti názorné ilustrace jeho realizace.

Obr. 6.4-5: Realizace regulačního obvodu v Simulinku – obrazový přenos, použití bloku *PID Controller*



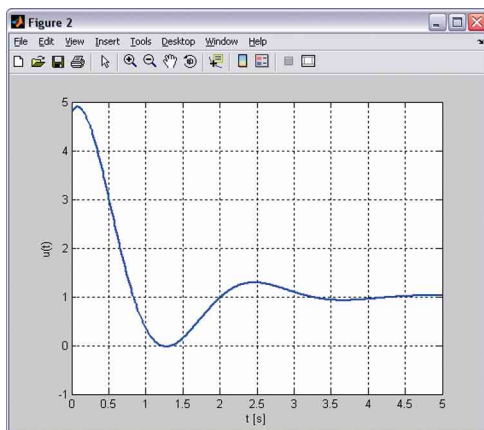
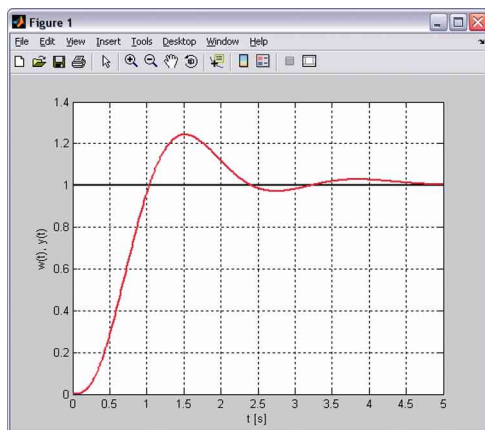
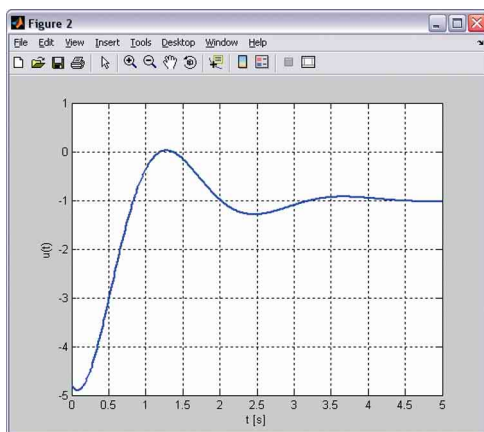
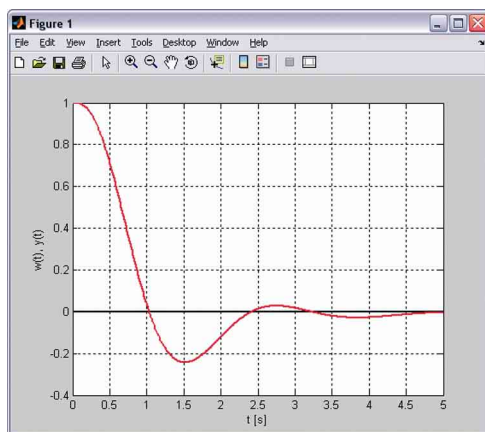
Obr. 6.4-6: Zadání konstant PID regulátoru

Na obr. 6.4-7 je implementován odlišný PID algoritmus. Jedná se o tzv. ideální paralelní tvar s interakcí konstant (také standardní či ISA tvar), který je podle [12] v praxi používán v cca 31 % průmyslových PID regulátorů, zatímco PID algoritmus realizovaný na obr. 6.4-3 až obr. 6.4-5 je používán v cca 22 % regulátorů. Konstanty regulátoru na obr. 6.4-7 byly přepočteny podle vztahů, které lze odvodit porovnáním obou tvarů rovnic PID regulátoru, tedy $r_0 = 4,8$, $T_i = r_0 / r_1 = 4,8 / 2,4 = 2$ s a $T_d = r_2 / r_0 = 1,2 / 4,8 = 0,25$ s.



Obr. 6.4-7: Realizace regulačního obvodu v Simulinku – obrazový přenos, jiná implementace PID algoritmu

Provedeme-li simulaci chování regulačních obvodů na obr. 6.4-3 až 6.4-5 a obr. 6.4-7, obdržíme naprosto identické průběhy, viz obr. 6.4-8. Všechny použité metody modelování, ať již samotné dynamické soustavy či PID regulátoru, jsou navzájem ekvivalentní. Seřízení regulátoru není samozřejmě optimální, regulační pochod je relativně kmitavý. Jednoduchou změnou parametrů v blocích w , $d1$ a $d2$ můžeme také sledovat vliv vstupujících poruch na průběh regulace. Odezva systému při poruše na jeho výstupu ($d_2 = \eta(t)$) je na obr. 6.4-9.


 Obr. 6.4-8: Průběh regulované veličiny a akční veličiny regulátoru při $w = \eta(t)$ a $d_1 = d_2 = 0$

 Obr. 6.4-9: Průběh regulované veličiny a akční veličiny regulátoru při $w = d_1 = 0$ a $d_2 = \eta(t)$

Dalším možným tvarem rovnic PID regulátoru je tzv. sériový (také klasický) tvar, jehož obrazové vyjádření je

$$U(s) = r_{0s} \left(1 + \frac{1}{T_{is}s} \right) (1 + T_{ds}s) E(s)$$

Tento tvar se v průmyslových PID používá nejčastěji, cca ve 47 % regulátorů. Konstanty regulátoru lze samozřejmě opět přepočítat na konstanty ideálního paralelního tvaru. Vztahy pro přepočet odvodíme vzájemným porovnáním obou tvarů rovnic regulátoru

$$r_0 \left(1 + \frac{1}{T_{is}s} + T_{ds}s \right) = r_0 + \frac{r_0}{T_{is}s} + r_0 T_{ds}s = r_{0s} \left(1 + \frac{1}{T_{is}s} \right) (1 + T_{ds}s) = r_{0s} + r_{0s} \frac{T_{ds}}{T_{is}} + \frac{r_{0s}}{T_{is}s} + r_{0s} T_{ds}s$$

Porovnáme-li nyní členy s odpovídající mocninou operátoru s , pak

$$r_0 = r_{0s} \left(1 + \frac{T_{ds}}{T_{is}} \right)$$

$$\frac{r_0}{T_i} = \frac{r_{0s}}{T_{is}} \Rightarrow T_i = r_0 \frac{T_{is}}{r_{0s}} = r_{0s} \left(1 + \frac{T_{ds}}{T_{is}} \right) \frac{T_{is}}{r_{0s}} = T_{is} + T_{ds}$$

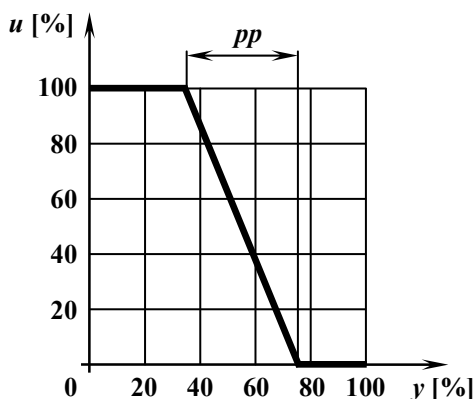
$$r_0 T_d = r_{0s} T_{ds} \Rightarrow T_d = \frac{r_{0s} T_{ds}}{r_0} = \frac{r_{0s} T_{ds}}{r_{0s} \left(1 + \frac{T_{ds}}{T_{is}} \right)} = \frac{T_{is} T_{ds}}{T_{is} + T_{ds}}$$

Převod parametrů regulátoru v sériovém tvaru r_{0s} , T_{is} , T_{ds} na parametry regulátoru ve tvaru paralelním r_0 , T_i , T_d je tedy triviální. Opačný přepočít ovšem není vždy možný, neboť sériový tvar odpovídá regulátoru s interakcí a určité kombinace hodnot T_i , T_d neumožňuje nastavit vůbec. Ze samotného faktu, že sériový tvar představoval po desetiletí jediný prakticky používaný typ PID algoritmu, je ale zároveň zřejmé, že tyto kombinace nebývají zapotřebí příliš často.



Obr. 6.4-10: Průmyslové procesní regulátory KS 90-1 a KS 92-1 firmy PMA

Většina v současnosti vyráběných procesních regulátorů používá PID algoritmus založený na sériovém tvaru rovnic. Např. regulátory KS 90-1 a KS 92-1 (obr. 6.4-10) německé firmy PMA mají také implementován sériový tvar rovnic, pouze s tím rozdílem, že na místo proporcionálního zesílení r_0 (proportional gain) je velikost proporcionální složky regulátoru udávána tzv. pásmem proporcionality (proportional band).



Obr. 6.4-11: K pásnu proporcionality ($w = 55\%$, $pp = 40\%$, $r_0 = 2,5$)

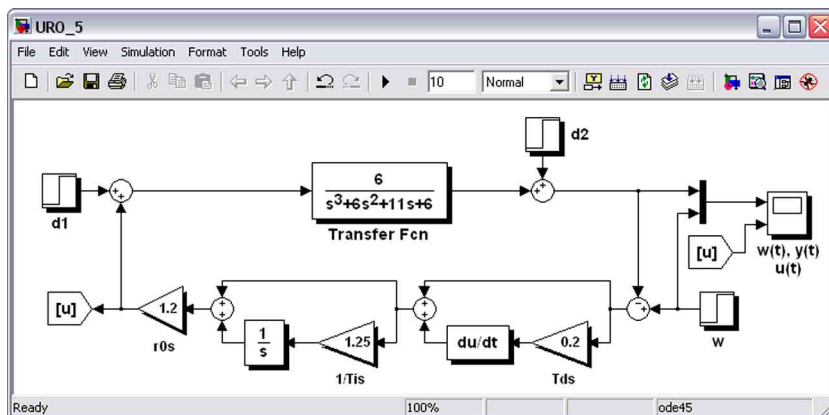
V přístrojové technice mají všechny veličiny určitý rozsah, proto je obvyklé je vyjadřovat jejich poměrnou hodnotou v rámci tohoto rozsahu. Budeme-li např. podle [13] uvažovat pouze proporcionální P regulátor, můžeme rovnici regulátoru zapsat obecně ve tvaru

$$u - u_0 = r_0 (w - y) = r_0 e$$

kde u_0 představuje referenční hodnotu akční veličiny u . Tento vztah lze graficky znázornit podle obr. 6.4-11. Skutečně proporcionální funkci má uvedený regulátor jen v části svého rozsahu, v tzv. **pásmu proporcionality**, které se obvykle označuje δ či pp . Při poměrném vyjádření veličin w , u a y je zřejmé splnění následující vztah mezi proporcionálním zesílením r_0 a pásmem proporcionality

$$\delta = pp = \frac{1}{r_0} 100 [\%]$$

Přestavováním žádané hodnoty dochází k posunu celé charakteristiky. Na obr. 6.4-11 je uvedena charakteristika regulátoru pro $w = 55\%$.



Obr. 6.4-12: Realizace sériového tvaru rovnic PID regulátoru

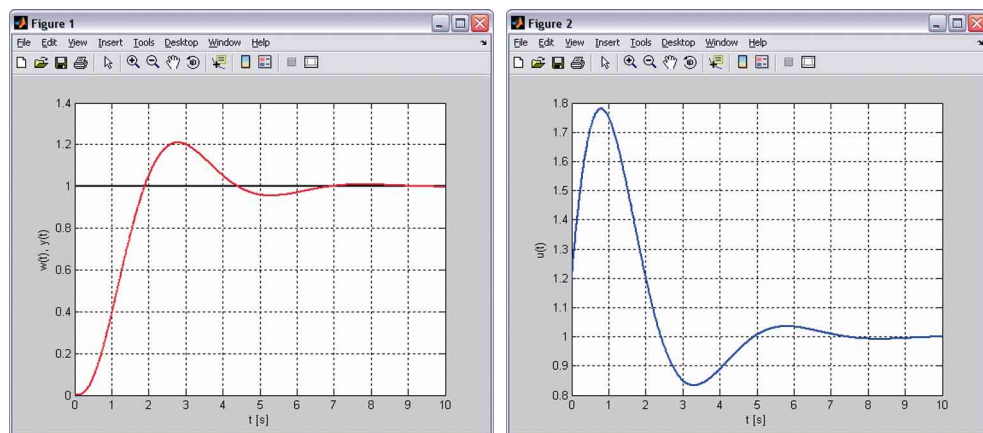
Realizace PID regulátoru v sériové struktuře v prostředí Simulink je na obr. 6.4-12. Regulovaná soustava je opět modelována pomocí obrazového přenosu, regulátor je sestaven ze základních bloků Simulinku. Navrženy byly tyto konstanty regulátoru: $r_{0s} = 1,2$, $T_{is} = 0,8$ s a $T_{ds} = 0,2$ s. Pásmo proporcionality regulátoru je $pp = 100 / r_{0s} = 100 / 1,2 \approx 83$ %. Přepoččet na koeficienty regulátoru v paralelní struktuře (ideální paralelní tvar) je pak podle výše uvedených vztahů následující

$$r_0 = r_{0s} \left(1 + \frac{T_{ds}}{T_{is}} \right) = 1,2 \left(1 + \frac{0,2}{0,8} \right) = 1,5$$

$$T_i = T_{is} + T_{ds} = 0,8 + 0,2 = 1 \text{ s}$$

$$T_d = \frac{T_{is} T_{ds}}{T_{is} + T_{ds}} = \frac{0,8 \cdot 0,2}{0,8 + 0,2} = 0,16 \text{ s}$$

Výsledky simulace chování uzavřeného regulačního obvodu s regulátorem v sériové struktuře (obr. 6.4-12) při buzení jednotkovým skokem $w(t) = \eta(t)$ jsou na obr. 6.4-13.



Obr. 6.4-13: Průběh regulované veličiny a akční veličiny regulátoru PID v sériové struktuře při $w = \eta(t)$ a $d_1 = d_2 = 0$

6.7 Modifikace PID algoritmu, seřízení regulátoru metodou Zieglera a Nicholse

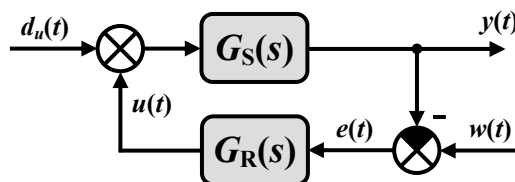
Derivační složka PID regulátoru reaguje na rychlé změny žádané hodnoty regulované veličiny (v běžném provozu se často vyskytují i náhlé skokové změny požadavku) ostrými derivačními špičkami akční veličiny, které mají negativní vliv na chování celého regulačního obvodu. Tomuto negativnímu jevu jednoduše zabrání často používaná modifikace algoritmu regulátoru. Derivační složka regulátoru je odvozena pouze od změn regulované veličiny $y(t)$ a negativní vliv rychlých změn žádané hodnoty $w(t)$ je tak eliminován. Diferenciální rovnice upraveného PID regulátoru je pak ve tvaru

$$u(t) = r_0 e(t) + r_1 \int_0^t e(\tau) d\tau - r_2 \frac{dy(t)}{dt}$$

nebo případně ve tvaru

$$u(t) = r_0 \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau - T_d \frac{dy(t)}{dt} \right]$$

Budeme uvažovat regulační obvod na obr. 6.7-1 s poruchou na vstupu do regulovaného systému $d_u(t)$, která je někdy označována jako porucha na akční veličině $u(t)$.

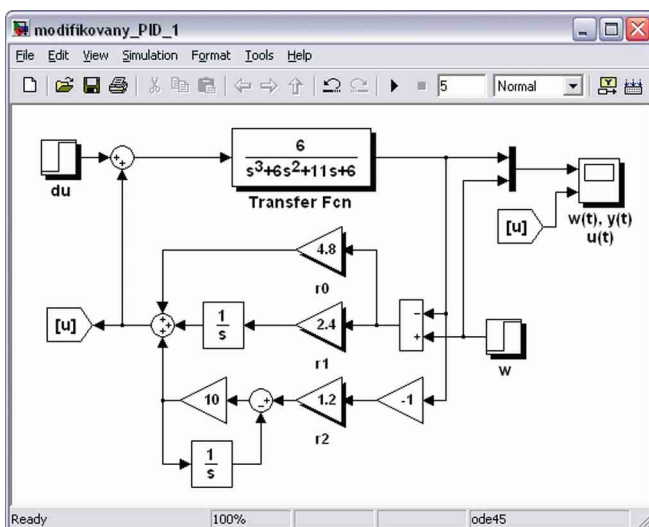


Obr. 6.7-1: Regulační obvod s poruchou na vstupu systému

Simulační model regulačního obvodu složeného ze systému třetího řádu popsaného obrazovým přenosem

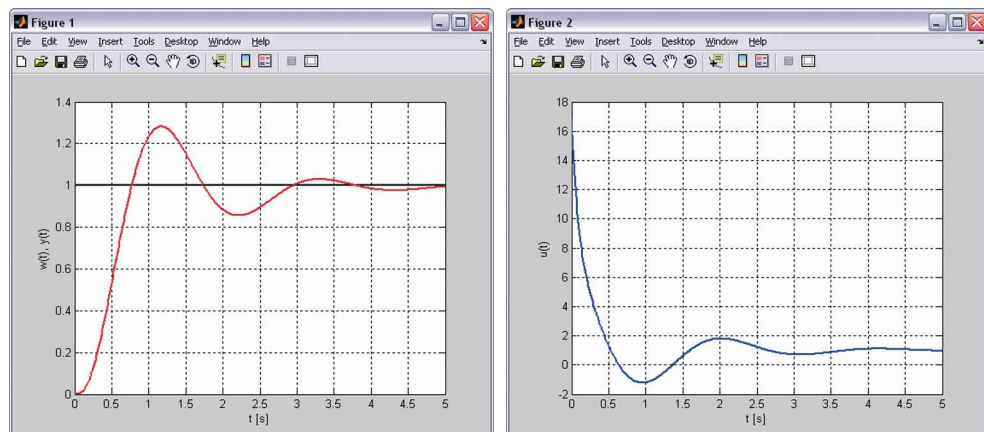
$$G_S(s) = \frac{6}{s^3 + 6s^2 + 11s + 6}$$

a modifikovaného regulátoru typu PID s aproximovanou derivací je na obr. 6.7-2. Model systému je realizován pomocí bloku *Transfer Fcn*. Nejprve budeme uvažovat, že na systém nepůsobí porucha, tj. $d_u(t) = 0$. Koeficienty regulátoru převzeme z příkladu 6.6.

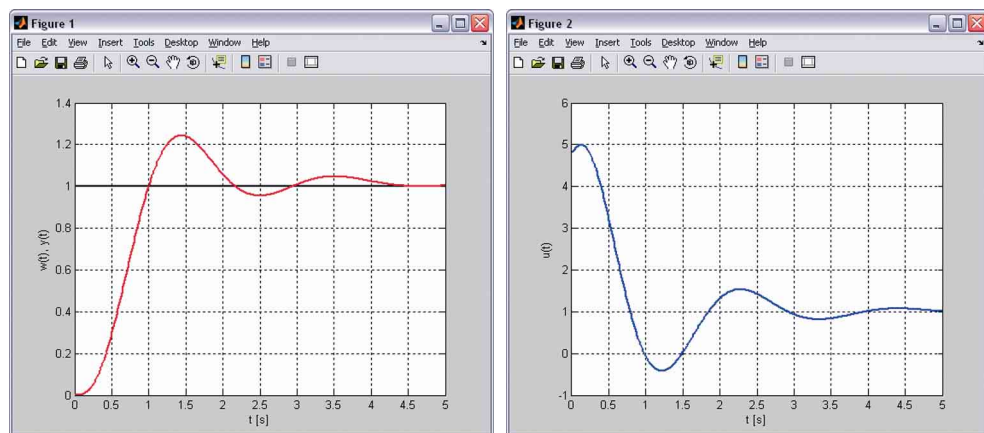


Obr. 6.7-2: Regulační obvod s modifikovaným PID regulátorem

Na obr. 6.7-3 jsou uvedeny průběhy odezvy systému (při $w(t) = \eta(t)$) a akční veličiny klasického PID regulátoru s aproximovanou derivací (je podrobněji popsán v kapitole 6.5). Na průběhu akční veličiny je patrná ostrá špička způsobená derivační složkou regulátoru. Na obr. 6.7-4 jsou pak průběhy získané při použití modifikovaného regulátoru. Odezva systému v tomto případě vykazuje co do velikosti menší kmity, ustálení systému je celkově pomalejší. Akční veličina regulátoru již nevykazuje výše zmiňovanou špičku, neboť derivační složka regulátoru je v tomto případě odvozena pouze od změn regulované veličiny.



Obr. 6.7-3: Průběh regulované veličiny a akční veličiny klasického PID regulátoru s aproximovanou derivací při $w = \eta(t)$ a $d_u = 0$

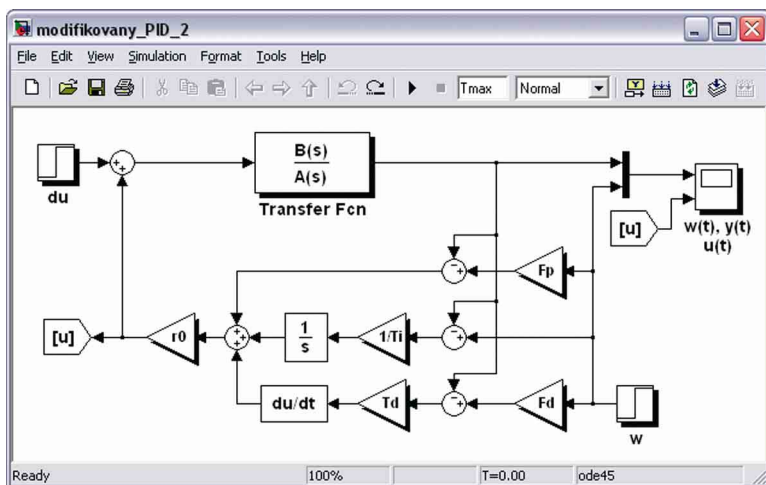


Obr. 6.7-4: Průběh regulované veličiny a akční veličiny modifikovaného PID regulátoru s aproximovanou derivací při $w = \eta(t)$ a $d_u = 0$

Další variantou regulátoru, kterou budeme realizovat je **PID s vážením žádané hodnoty**. Struktura tohoto regulátoru je již poněkud složitější a umožňuje odvozovat proporcionální a derivační složku od regulované veličiny (nikoliv od regulační odchylky jako u klasického PID regulátoru). Akční veličina $u(t)$ je vytvářena následujícím způsobem

$$u(t) = r_0 [F_p w(t) - y(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{d}{dt} (F_d w(t) - y(t))]$$

Je-li regulátor nastaven především s ohledem na dobré vyregulování poruchy a odezva na skokovou změnu žádané hodnoty je proto příliš kmitavá, uvedená modifikace způsobí zpomalení náběhu na žádanou hodnotu a zmenšení překmitů. Simulační model regulačního obvodu s modifikovaným regulátorem je na obr. 6.7-5. Koeficienty (váhy) F_p a F_d bývají běžně voleny rovny buď jedné či nule, obecně lze ovšem uvažovat celý interval $\langle 0; 1 \rangle$.



Obr. 6.7-5: Regulační obvod s PID regulátorem s vážením žádané hodnoty

Zajímavé jistě bude porovnání klasického a modifikovaného PID regulátoru. Přenos regulovaného systému může být např.

$$G_S(s) = \frac{1}{(T_1 s + 1)(T_2 s + 1)(T_3 s + 1)}$$

s časovými konstantami $T_1 = 2$ s, $T_2 = 0,8$ s a $T_3 = 0,2$ s. Přenos systému můžeme převést do obvyklé podoby s polynomy v čitateli a ve jmenovateli a poté modelovat standardním způsobem pomocí bloku *Transfer Function*.

Dále je nutné definovat také hodnoty konstant PID regulátoru. Na základě průběhu odezvy uzavřeného regulačního obvodu můžeme provést např. ruční seřizení regulátoru nebo využít některou z mnoha metod. V našem případě byly konstanty regulátoru navrženy na základě metody kritického zesílení Zieglera-Nicholse. Pro tyto účely byl vytvořen speciální skript, který je na obr. 6.7-7.

Pravidla **Zieglera a Nicholse** jsou např. v [11] uváděna ve dvou variantách. Obě tyto varianty vycházejí z velmi zjednodušeného popisu regulovaného systému. V prvním případě se vychází z tzv. kritické frekvence systému (tj. frekvence, při níž je fázové zpoždění rovné 180°) resp. z odpovídající kritické periody T_k a ze zesílení systému při této frekvenci.

K nalezení obou parametrů se používá jednoduchý experiment v uzavřeném regulačním obvodu s P regulátorem (vyřadíme složky I a D), jehož zesílení se postupně zvyšuje až do okamžiku, kdy se obvod dostane na mez stability a objeví se netlumené kmity. Frekvence těchto kmitů je rovna kritické frekvenci a zesílení P regulátoru (tzv. kritické zesílení r_k) je převrácenou hodnotou zesílení systému. Na základě výsledků tohoto experimentu jsou pak podle pravidel uvedených v tab. 6.7-1 určeny hodnoty parametrů regulátoru.

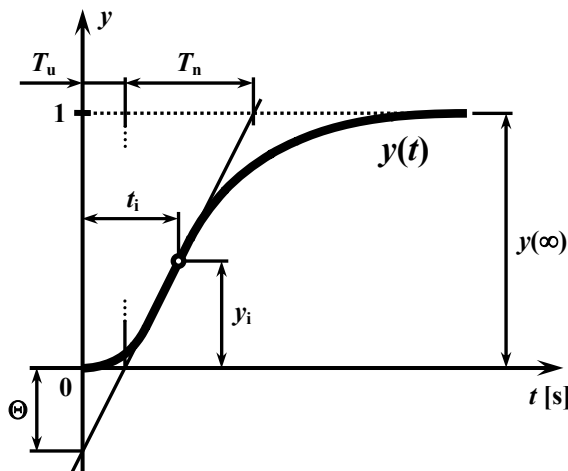
Tab. 6.7-1: Pravidla Zieglera-Nicholse, metoda ustálených kmitů

Regulátor	r_0	T_i	T_d
P	$0,5 r_k$	–	–
PI	$0,45 r_k$	$0,85 T_k$	–
PID	$0,6 r_k$	$0,5 T_k$	$0,125 T_k$

Tab. 6.7-2: Pravidla Zieglera-Nicholse, metoda přechodové odezvy

Regulátor	r_0	T_i	T_d
P	$1 / (K \Theta)$	–	–
PI	$0,9 / (K \Theta)$	$3 T_u$	–
PID	$1,2 / (K \Theta)$	$2 T_u$	$0,5 T_u$

Uvedení systému na mez stability může být ovšem mnohdy z technologických či provozních důvodů nepřijatelné. Pak lze použít druhou metodu založenou na vyhodnocení přechodové charakteristiky regulovaného systému v okolí pracovního bodu. Na základě doby průtahu T_u , doby náběhu T_n (resp. normalizovaného dopravního zpoždění $\Theta = T_u / T_n$) a statického zesílení K (viz obr. 6.7-6) pak lze nastavit regulátor podle pravidel uvedených v tab. 6.7-2.



Obr. 6.7-6: Základní parametry přechodové odezvy statického nekomitavého systému

```

%
% PID regulator s vazenim zadane hodnoty, metoda Zieglera-Nicholse
%
clear all, close all, clc, format compact

%system G(s)=K/[(2s+1)*(0.8s+1)*(0.2s+1)]
K=1; B=K; A=conv(conv([2 1],[0.8 1]),[0.2 1]); Gs=tf(B,A);
Tm=15; n=1e4; t=linspace(0,Tm,n);

y=step(Gs,t);          %vypocet jednotlivych bodu h(t)
dy=impulse(Gs,t);      %vypocet g(t), tj. derivace funkce h(t)
%dt=Tm/(n-1);          %vypocet kroku
%dy=diff(y)/dt; dy=dy'; dy=[dy dy(end)]; %alternativni vypocet derivace

[k i]=max(dy);          %nalezeni maxima na prubehu derivace a jeho index
ti=t(i);                %vyber z vektoru hodnot casu t
tecna=k*(t-ti)+y(i);    %vypocet tecny - rovnice primky kx+q
                        %[prochazi bodem s indexem i, tj. [ti,yi]]

figure, plot(t,dy,'LineWidth',2), grid on
figure, plot(t,y,t,tecna,'LineWidth',2), grid on, axis([0 Tm -0.2 1.2])

I=find(tecna>=0); Tu=t(I(1)), clear I      %doba prutahu
I=find(tecna>=K); Tn=t(I(1))-Tu, clear I   %doba nabehu

%kriticke hodnoty
theta=Tu/Tn;
%rk=sqrt(((pi*Tn)/(2*Tu))^2+1)/K; Tk=4*Tu;

%model v Simulinku
Tmax=ceil(Tm/10)*20; w=1; tdu=Tmax/2; du=1;

%serizeni PID podle metody Z-N
%r0=0.6*rk; Ti=0.5*Tk; Td=0.125*Tk;
r0=1/theta; Ti=2*Tu; Td=0.5*Tu; Fp=1; Fd=1;
sim modifikovany_PID_2;
y1=v.signals(1).values(:,1); w1=v.signals(1).values(:,2);
u1=v.signals(2).values; t1=t;
figure, plot(t1,y1,'k--','LineWidth',2), hold on
xlabel('t [s]'), ylabel('y(t)')

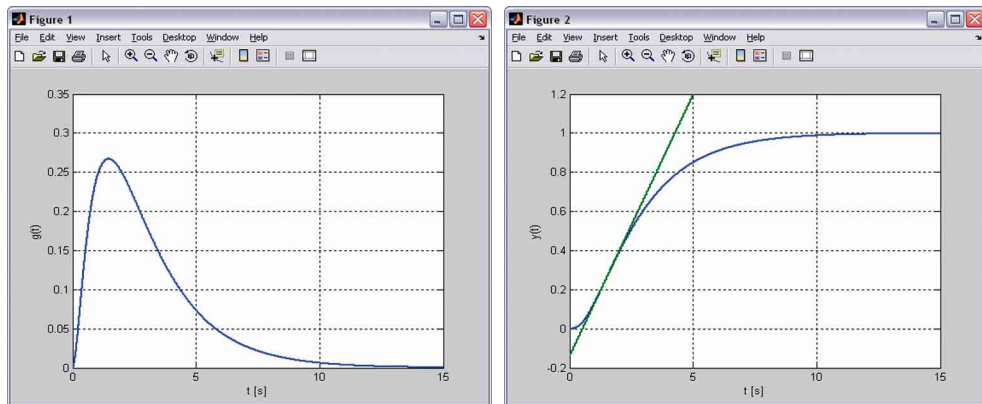
%modifikace Z-N (40x snizen koeficient u r0 - bez prekmitu)
%r0=0.015*rk; Ti=0.5*Tk; Td=0.33*Tk;
r0=1/theta/40; Ti=2*Tu; Td=0.5*Tu;
sim modifikovany_PID_2;
y2=v.signals(1).values(:,1); w2=v.signals(1).values(:,2);
u2=v.signals(2).values; t2=t;
plot(t2,y2,'b','LineWidth',2), grid on
xlabel('t [s]'), ylabel('y(t)')
legend('serizeni podle Z-N','aperiodicky prubeh',4)

%serizeni PID podle metody Z-N, vazeni zadane hodnoty
%r0=0.6*rk; Ti=0.5*Tk; Td=0.125*Tk;
r0=1/theta; Ti=2*Tu; Td=0.5*Tu; Fp=0; Fd=0;
sim modifikovany_PID_2;
y3=v.signals(1).values(:,1); w3=v.signals(1).values(:,2);
u3=v.signals(2).values; t3=t;
figure; plot(t1,y1,'k--',t3,y3,'b','LineWidth',2), grid on
xlabel('t [s]'), ylabel('y(t)')
legend('serizeni podle Z-N','modifikovany PID',4)
figure; plot(t1,u1,'r--',t3,u3,'g','LineWidth',2), grid on
xlabel('t [s]'), ylabel('u(t)')
legend('serizeni podle Z-N','modifikovany PID',4)

```

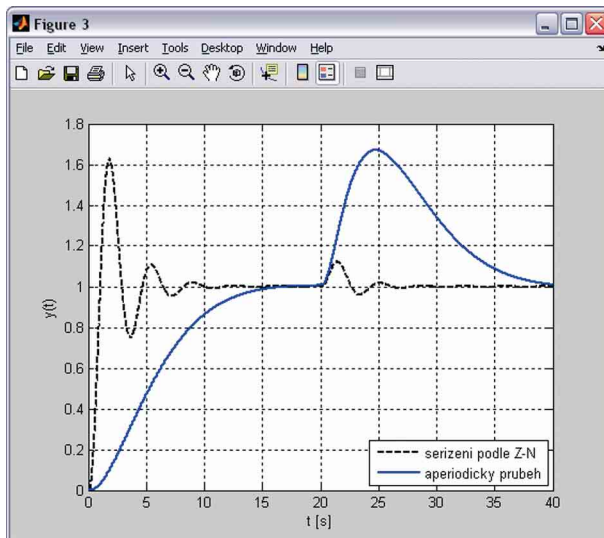
Obr. 6.7-7: Skript pro porovnání různých variant návrhu PID regulátoru (modifikovany_PID_ZN.m)

Ve skriptu je nejprve přenos systému převeden do podoby s polynomy v čitateli a ve jmenovateli (je využit příkaz **conv**); přenos systému je pak vytvořen příkazem **tf**. Dále je vypočtena přechodová a váhová (impulsní) odezva tohoto systému, viz obr. 6.7-8.



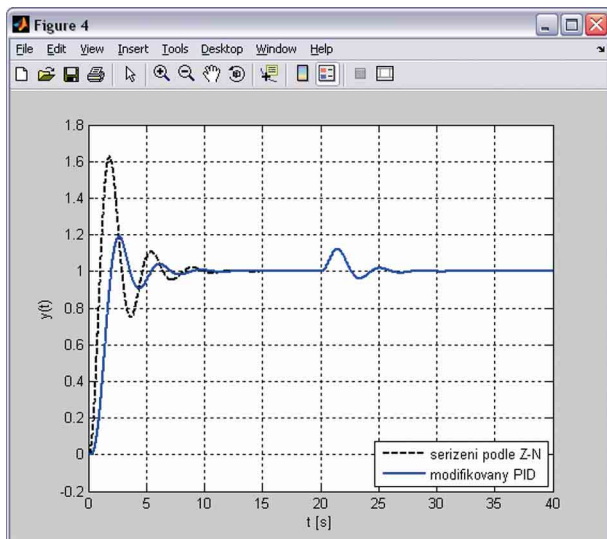
Obr. 6.7-8: K nalezení doby průtahu a doby náběhu systému

Pro seřízení regulátoru byla použita druhá varianta pravidel Zieglera-Nicholse – **metoda přechodové odezvy**. Chceme-li určit z průběhu přechodové funkce dobu průtahu T_u a dobu náběhu T_d , potřebujeme zkonstruovat tečnu v inflexním bodě funkce (obr. 6.7-7). Inflexní bod odpovídá maximu derivace přechodové funkce $h(t)$, tj. maximu funkce váhové $g(t)$. K jeho nalezení lze použít příkaz **impulse**, který váhovou charakteristiku systému přímo vypočte, případně příkaz **diff** k numerickému výpočtu derivace funkce $h(t)$. Konstrukce tečny je již jednoduchá. Doby průtahu resp. náběhu lze pak určit s využitím příkazu **find**.

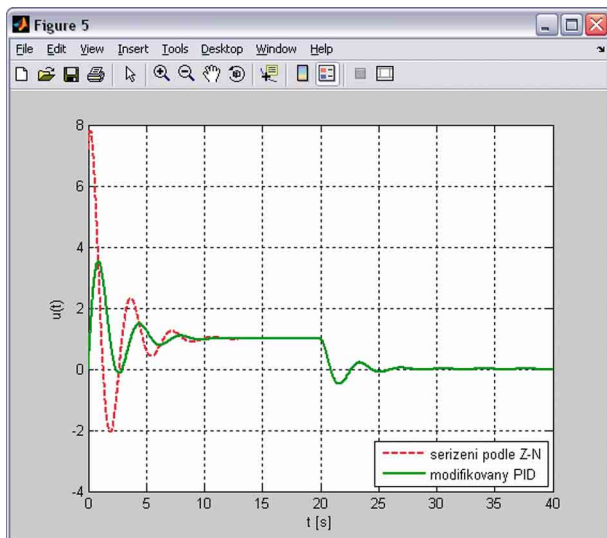


Obr. 6.7-9: Odezva při seřízení regulátoru metodou Zieglera-Nicholse a její porovnání s aperiodickým průběhem

Ve skriptu jsou dále z nalezených hodnot T_u a T_n resp. ze zpoždění $\Theta = T_u/T_n$ vypočteny podle pravidel Zieglera-Nicholse optimální hodnoty koeficientů regulátoru. Konkrétně je použita metoda přechodové odezvy (tab. 6.7-1), alternativně je uvedeno také řešení pomocí metody ustálených kmitů (tab. 6.7-2). Průběh obdržené odezvy při $w(t) = \eta(t)$ a poruše $d_u(t)$ vstupující v čase $t = 20$ s je zobrazen na obr. 6.7-9. Pro porovnání je uveden i aperiodický průběh odezvy dosažený snížením velikosti proporcionálního zesílení regulátoru.



Obr. 6.7-10: Odezva při použití PID regulátoru s vážením žádané hodnoty

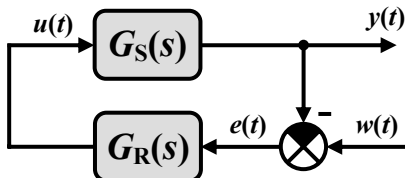


Obr. 6.7-11: Průběh akční veličiny při použití PID regulátoru s vážením žádané hodnoty

Pravidla pro nalezení vhodných konstant regulátoru jsou u metody Zieglera-Nicholse optimalizována z hlediska **dobrého potlačení poruch** a pro sledování skokových změn žádané hodnoty jsou nevhodná a prakticky nepoužitelná. V regulátoru s vážením žádané hodnoty zvolíme váhy u proporcionální a derivační složky $F_p = F_d = 0$ a tím zajistíme jejich odvození od regulované veličiny $y(t)$ (nikoliv od regulační odchylky $e(t)$ jako u integrační složky). Názorné zobrazení obou odezev uzavřeného obvodu v jednom grafu, tedy odezvy při použití klasického PID regulátoru ($F_p = F_d = 1$) seřazeného metodou Zieglera-Nicholse a regulátoru modifikovaného ($F_p = F_d = 0$), je na obr. 6.7-10. Na obr. 6.7-11 jsou odpovídající průběhy akční veličiny regulátoru v těchto dvou případech. Jak vidíme, tak shora uvedená hypotéza se potvrdila, neboť modifikovaný PID regulátor podstatně lépe vyreguluje změnu žádané hodnoty při zachování dobrého potlačení poruchy.

6.8 Oblast stability regulačního obvodu

Oblastí stability regulačního obvodu (obr. 6.8-1) rozumíme oblast vymezující takové možné kombinace stavitelných parametrů regulátoru $G_R(s)$, aby uzavřený regulační obvod byl s těmito kombinacemi parametrů stabilní. Stabilita je základní vlastností, kterou od regulačního obvodu očekáváme. U jednodušších regulačních obvodů lze oblast stability vyšetřit analytickými prostředky.



Obr. 6.8-1: Blokové schéma regulačního obvodu

Vlastnosti regulátoru typu PID jsou popsány diferenciální rovnicí

$$u(t) = r_0 e(t) + r_1 \int_0^t e(\tau) d\tau + r_2 \frac{de(t)}{dt}$$

Přenos regulátoru je

$$G_R(s) = r_0 + \frac{r_1}{s} + r_2 s$$

Vyšetřeme nyní a simulačně ověříme oblast stability jednoduchého regulačního obvodu s PI regulátorem a regulovanou soustavou popsanou diferenciální rovnicí

$$y'''(t) + 6y''(t) + 11y'(t) + 6y(t) = 6u(t)$$

resp. přenosem

$$G_S(s) = \frac{6}{s^3 + 6s^2 + 11s + 6}$$

Obrazový přenos PI regulátoru je

$$G_R(s) = r_0 + \frac{r_1}{s} = \frac{r_0 s + r_1}{s}$$

Přenos **uzavřeného regulačního obvodu** na obr. 6.8-1, např. tzv. **přenos řízení** $G_{yw}(s)$ je

$$G_{yw}(s) = \frac{Y(s)}{W(s)} = \frac{G_R(s) G_S(s)}{1 + G_R(s) G_S(s)}$$

Dosadíme-li za přenos regulátoru $G_R(s)$ a soustavy $G_S(s)$, pak

$$G_{uw}(s) = \frac{\frac{r_0 s + r_1}{s} \frac{6}{s^3 + 6s^2 + 11s + 6}}{1 + \frac{r_0 s + r_1}{s} \frac{6}{s^3 + 6s^2 + 11s + 6}} = \dots = \frac{6r_0 s + 6r_1}{s^4 + 6s^3 + 11s^2 + 6s(1 + r_0) + 6r_1}$$

Kořeny **charakteristické rovnice** uzavřeného regulačního obvodu, kterou získáme tak, že charakteristický polynom ve jmenovateli přenosu položíme roven nule, rozhodují o jeho stabilitě. Charakteristická rovnice uvažovaného obvodu je

$$s^4 + 6s^3 + 11s^2 + 6s(1 + r_0) + 6r_1 = 0$$

Jsou-li všechny její kořeny **v levé polorovině** Gaussovy roviny (buď reálné záporné nebo komplexní se zápornou reálnou částí), regulační obvod je stabilní, jestliže alespoň jeden kořen tuto podmínku nesplňuje, obvod stabilní není.

Protože se jedná o charakteristickou rovnici uzavřené regulační smyčky, lze ji získat i z libovolné jiné dynamické závislosti, např. z přenosů

$$G_{yw}(s) = \frac{U(s)}{W(s)} = \frac{G_R(s)}{1 + G_R(s) G_S(s)}, \quad G_{ew}(s) = \frac{E(s)}{W(s)} = \frac{1}{1 + G_R(s) G_S(s)}$$

nebo také podle vztahu

$$1 + G_o(s) = 1 + G_R(s) G_S(s) = 0$$

kde G_o je přenos **otevřeného regulačního obvodu**, který lze vyjádřit

$$G_o(s) = \frac{Y(s)}{E(s)} = G_R(s) G_S(s)$$

Změny parametrů regulátoru r_0 a r_1 ovlivňují nejen koeficienty charakteristické rovnice, ale samozřejmě i její kořeny. Mohou tedy nastat situace, že uzavřený regulační obvod bude vlivem seřízení regulátoru stabilní nebo i nestabilní. Vyšetření kořenů rovnice analytickými prostředky v případě vyšších řádů není možné. Jsme potom nuceni použít některé z kritérií stability. Často používané je kritérium **Hurwitzovo**, které patří mezi **algebraická kritéria**

stability. Tato kritéria vycházejí z charakteristické rovnice uzavřeného regulačního obvodu a nelze je použít při vyšetřování stability obvodů s dopravním zpožděním.

Nutnou podmínkou pro stabilitu obvodu je, aby všechny koeficienty charakteristické rovnice měly stejná znaménka (zde kladná)

$$1, 6, 11 > 0, \quad 6(1+r_0) > 0 \Rightarrow r_0 > -1, \quad 6r_1 > 0 \Rightarrow r_1 > 0$$

Postačující podmínkou je, aby všechny Hurwitzovy subdeterminanty H_i (hlavní rohové minory) Hurwitzovy matice

$$\mathbf{H} = \left[\begin{array}{ccc|cc} a_{n-1} & a_{n-3} & a_{n-5} & 0 & 0 \\ a_n & a_{n-2} & a_{n-4} & 0 & 0 \\ 0 & a_{n-1} & a_{n-3} & 0 & 0 \\ 0 & a_n & a_{n-2} & 0 & 0 \\ \hline 0 & 0 & 0 & a_2 & a_0 \end{array} \right]$$

byly také kladné. Rozměr Hurwitzovy matice je dán stupněm charakteristického polynomu.

V našem případě nabývá Hurwitzova matice tvaru

$$\mathbf{H} = \left[\begin{array}{cccc} 6 & 6(1+r_0) & 0 & 0 \\ 1 & 11 & 6r_1 & 0 \\ 0 & 6 & 6(1+r_0) & 0 \\ 0 & 1 & 11 & 6r_1 \end{array} \right]$$

a pro její jednotlivé subdeterminanty pak platí

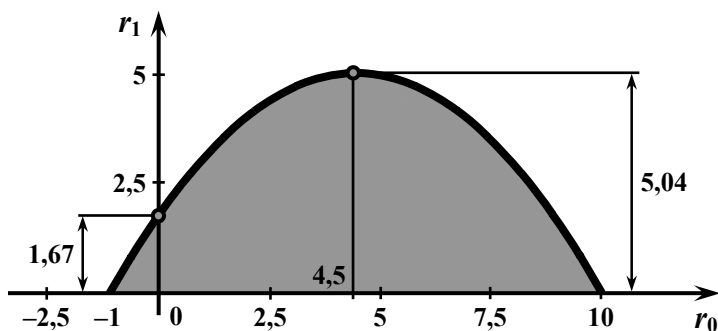
$$H_1 = 6 > 0$$

$$H_1 = \det \begin{bmatrix} 6 & 6(1+r_0) \\ 1 & 11 \end{bmatrix} = 6 \cdot 11 - 1 \cdot 6(1+r_0) = 6(10+r_0) > 0 \Rightarrow \underline{r_0 < 10}$$

$$\begin{aligned} H_3 &= \det \begin{bmatrix} 6 & 6(1+r_0) & 0 \\ 1 & 11 & 6r_1 \\ 0 & 6 & 6(1+r_0) \end{bmatrix} = 6 \cdot 11 \cdot 6(1+r_0) - 6 \cdot 6r_1 \cdot 6 - 6(1+r_0) \cdot 1 \cdot 6(1+r_0) = \\ &= 36(-r_0^2 + 9r_0 + 10 - 6r_1) > 0 \Rightarrow \underline{-r_0^2 + 9r_0 + 10 > 6r_1} \end{aligned}$$

$$H_4 = 6r_1 H_3 > 0$$

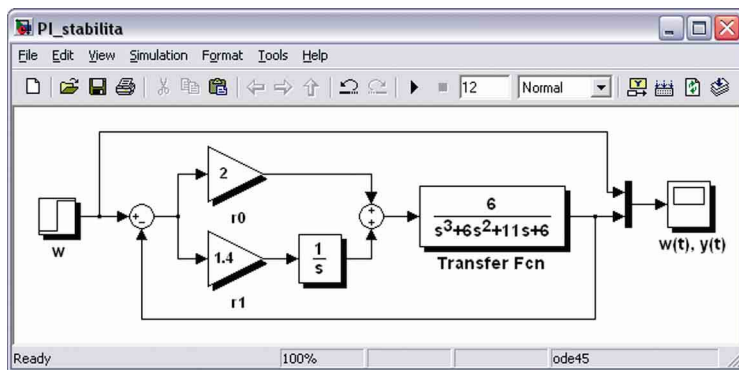
Soustava výše uvedených podtržených nerovností definuje podmínky pro stabilitu uzavřeného regulačního obvodu (oblast stability). Povšimněme si, že jsme určili podmínky stability, aniž bychom znali kořeny charakteristické rovnice. Potřebu analýzy jejich kořenů jsme tak vhodně obešli použitím kritéria stability. Grafické vyjádření oblasti stability je na obr. 6.8-2.



Obr. 6.8-2: Oblast stability regulačního obvodu

Oblast stability téhož obvodu s regulátorem typu P by byla vyjádřena zřejmě úsečkou $1 < r_0 < 10$ (situace pro $r_1 = 0$), pro regulátor typu I úsečka $0 < r_1 < 1,67$ (řez oblastí pro $r_0 = 0$). Vyšetřená oblast na obr. 6.8-2 je řezem složitější 3D oblasti stability regulátoru typu PID při nulové derivační složce.

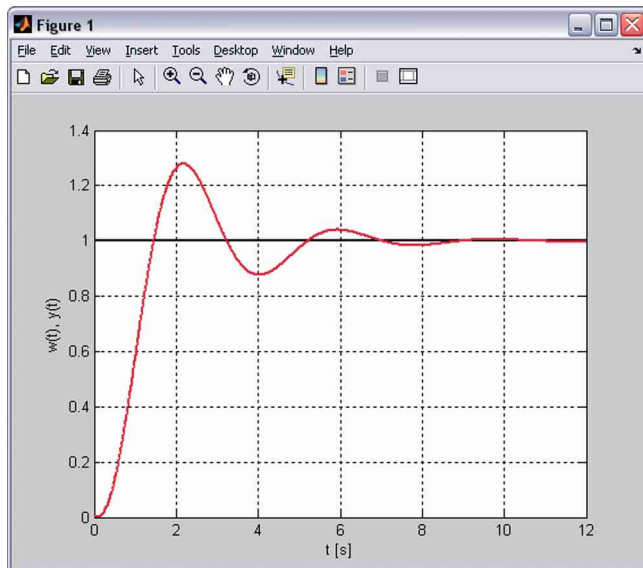
Správnost teoreticky vyšetřené oblasti stability můžeme ověřit simulací (schéma je na obr. 6.8-3) a zároveň také sledovat charakter možných průběhů stabilních i nestabilních regulačních pochodů.



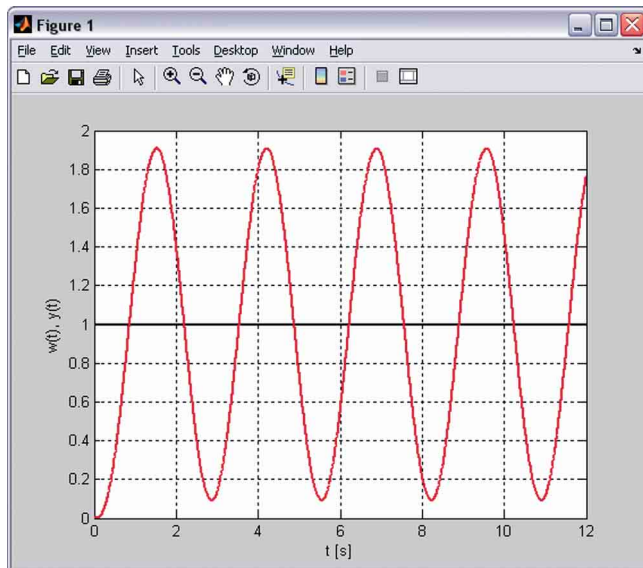
Obr. 6.8-3: Uzavřený regulační obvod s PI regulátorem

Průběh regulačního pochodu při parametrech $r_0 = 2$ a $r_1 = 1,4$ je na obr. 6.8-4. Je zřejmé, že tyto hodnoty koeficientů regulátoru byly zvoleny tak, aby se nacházely ve stabilní oblasti na obr. 6.8-2. Nulová hodnota Hurwitzova subdeterminantu $H_3 = 0$ (všude na parabolické části hranice oblasti) indikuje existenci ryze imaginárních kořenů charakteristické rovnice (ostatní kořeny jsou v levé polorovině). Situace při $r_0 = 4,5$ a $r_1 = 5,04$ (vrchol paraboly) je na obr. 6.8-5. Regulační obvod je na mezí stability, dochází k netlumeným kmitům.

Povšimněme si, že oblast stability zasahuje svou levou částí do záporných hodnot r_0 . To ovšem znamená slabou kladnou zpětnou vazbu proporcionální složky. Dramaticky se tato skutečnost projeví je-li $r_1 = 0$, takovéto seřízení ale nelze doporučit. Důvodem je i fakt, že reálné průmyslové regulátory záporné hodnoty koeficientů ani neumožňují nastavit.



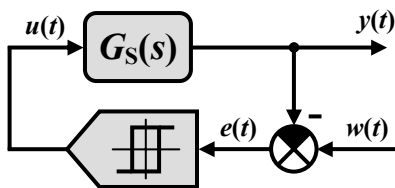
Obr. 6.8-4: Průběh regulačního pochodu při $r_0 = 2$ a $r_1 = 1,4$



Obr. 6.8-5: Průběh regulačního pochodu při $r_0 = 4,5$ a $r_1 = 5,04$

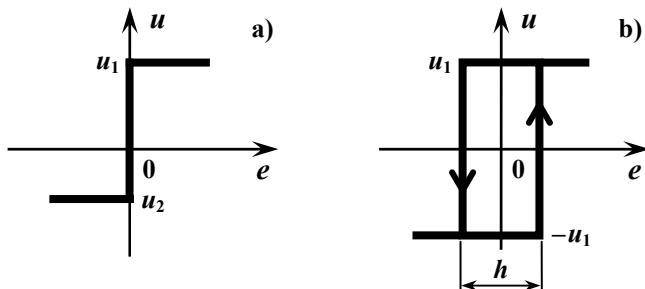
6.9 Dvoupolohová a třípolohová regulace

S regulačními obvody s nespojitým regulátorem ve zpětné vazbě jsme se zcela jistě již mnohokrát setkali. Používají se velmi často – všude tam, kde neklademe příliš vysoké nároky na kvalitu regulačních pochodů. Výhodou je jejich jednoduchost, spolehlivost a zpravidla i nízká cena. Setkáváme se s nimi v řadě případů u domácích spotřebičů, např. u žehliček, ale i v průmyslových aplikacích. Schéma regulačního obvodu s nespojitým regulátorem je na obr. 6.9-1



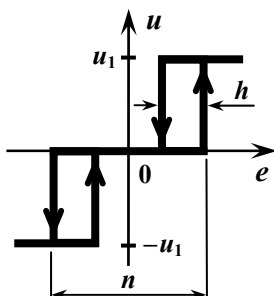
Obr. 6.9-1: Blokové schéma regulačního obvodu s nespojitým regulátorem

Výstupní veličina ústředního členu regulátoru (akční veličina u) nabývá obvykle jen dvou nebo tří hodnot – zapnuto/vypnuto, přidat/ubrat/zastavit. Nejjednodušší typy nespojitých regulátorů bez vnitřní zpětné vazby mají jen statický přepínací charakter. Akční veličina $u(t)$ je odvozena od hodnoty regulační odchylky $e(t)$ podle zvolené přepínací charakteristiky regulátoru. Nejčastější situace jsou na obr. 6.9-2 až 6.9-3.



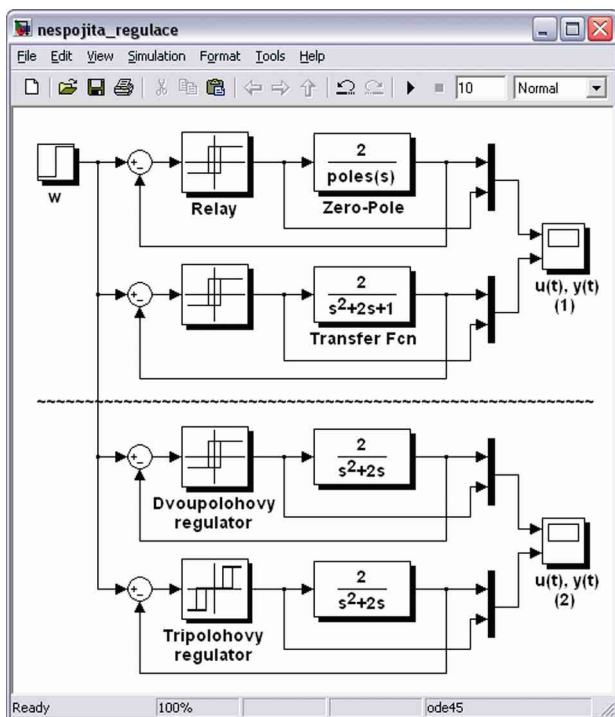
Obr. 6.9-2: Přepínací charakteristiky dvoupolohového regulátoru
a) nesymetrické
b) symetrické s hysterezí h

Zavedení hystereze h do přepínací charakteristiky regulátoru zamezuje extrémně rychlému přepínání akční veličiny u mezi jejími krajními hodnotami (tím i namáhání a opotřebení regulátoru a akčního členu) v případě regulované soustavy 1. řádu nebo soustavy s malými časovými konstantami. Zlepšení regulačního pochodu je na úkor přesnosti regulace, je nutné proto při volbě h vždy volit vhodný kompromis. Maximální hodnoty hystereze, které lze nastavit na průmyslově vyráběných regulátorech se pohybují řádově v desetinách až jednotkách procent rozsahu regulované veličiny (na uvedených obrázcích je tento parametr záměrně nadsazen).



Obr. 6.9-3: Symetrické přepínací charakteristiky třípolohového regulátoru s hystezí h a s pásmem necitlivosti n

V případě, že regulátor může aktivně působit na regulovanou veličinu v obou směrech, např. spínat běh motoru servomechanizmu v jednom či druhém směru, je vhodné zavedení pásma necitlivosti n , které nám umožní zastavení motoru jestliže $|e| < n/2$. V opačném případě by musel motor stále běžet buď jedním nebo druhým směrem, docházelo by tak k nežádoucímu kmitání regulačního pochodu a neúměrnému zatěžování akčního členu i regulátoru. I v tomto případě je nutné hledat vhodný kompromis mezi průběhem regulace a její kvalitou.



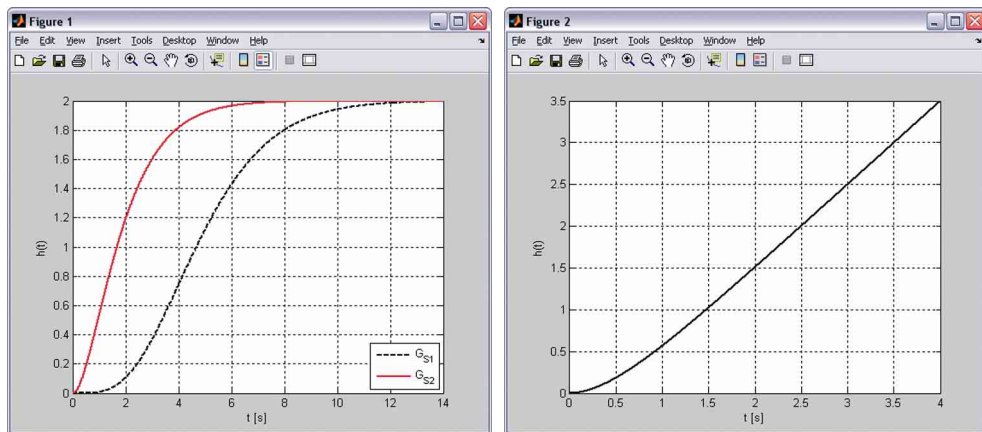
Obr. 6.9-4: Porovnání regulačních pochodů s dvupolohovým a třípolohovým regulátorem

Naším úkolem bude simulovat regulační pochody s různými typy nespojitých regulátorů, s uvažováním jejich různých parametrů a s regulovanými soustavami s různou dynamikou. Zaměříme se na vliv parametrů regulátoru a na přesnost a kvalitu regulace. Jednou z možností realizace modelu je simulační schéma na obr. 6.9-4, které je sestaveno tak, abychom mohli sledovat a srovnávat současně několik různých regulačních pochodů.

První dva simulované případy (ve schématu nahoře) představují dva regulační obvody s různou dynamikou soustav. V druhém případě (ve schématu dole) je porovnáván pochod s dvoupolohovým a třípolohovým regulátorem. Regulovaná soustava je v tomto případě astatická (řád astatismu je 1). Přenosy soustav jsou

$$G_{S1}(s) = \frac{2}{(s+1)^5}, \quad G_{S2}(s) = \frac{2}{s^2 + 2s + 1}, \quad G_{S3}(s) = \frac{2}{s^2 + 2s}$$

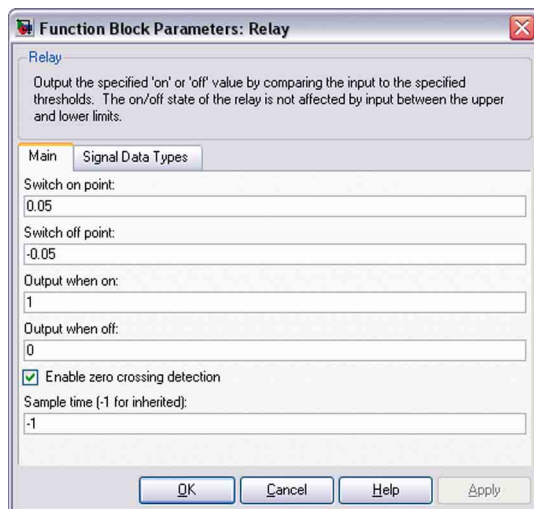
Je zřejmé, že regulované soustavy se navzájem značně liší svojí dynamikou. Odpovídající přechodové charakteristiky všech tří systémů jsou na obr. 6.9-5.



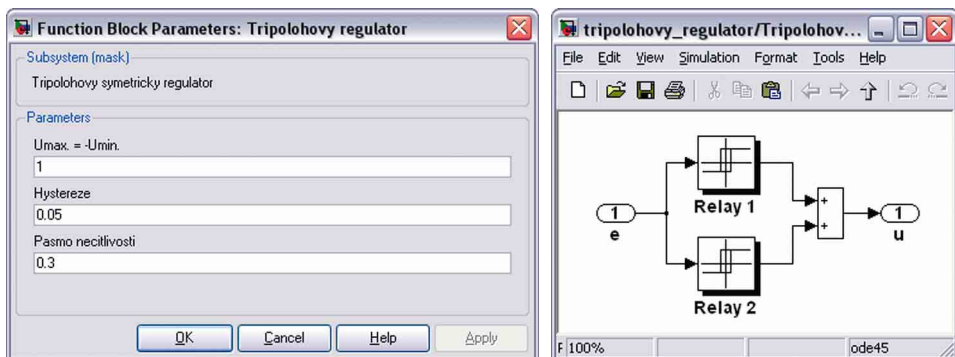
Obr. 6.9-5: Přechodové charakteristiky regulovaných soustav

Dvoupolohový regulátor v Simulinku vytvoříme pomocí bloku *Relay*. V bloku zadáme hodnoty pro přepínání akční veličiny (**Output when on** a **Output when off**) a velikost hystereze h (resp. $h/2$, **Switch on point** a **Switch off point**), viz obr. 6.9-6. Třípolohový regulátor není v Simulinku přímo k dispozici – musíme jej vytvořit pomocí dvou bloků *Relay* (obr. 6.9-7). Nový blok můžeme zahrnout do subsystému a pomocí maskování zajistit, aby se při jeho vybrání otevřelo okno, ve kterém lze přímo zadat příslušné parametry. Postup při maskování subsystému je podrobně vysvětlen v [4]. Jakmile je subsystém zamaskován, není již možný obvyklý přístup k jeho vnitřnímu schématu. Úprava takto zamaskovaného systému je ale možná, např. volbou z plovoucího menu **Look Under Mask** příp. pomocí kombinace kláves **Ctrl + U**. Ikonu bloku lze také doplnit např. o přepínací charakteristiky regulátoru (podobně jako u bloku *Relay*).

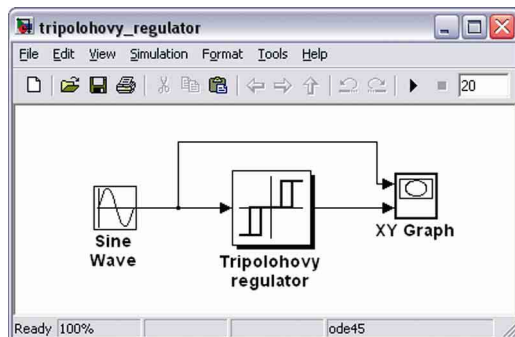
Přepínací charakteristiky regulátoru můžeme také zobrazit. Odpovídající schéma, které jejich vykreslení pomocí bloku *XY Graph* umožňuje, je na obr. 6.9-8, přepínací charakteristika je pak na obr. 6.9-9.



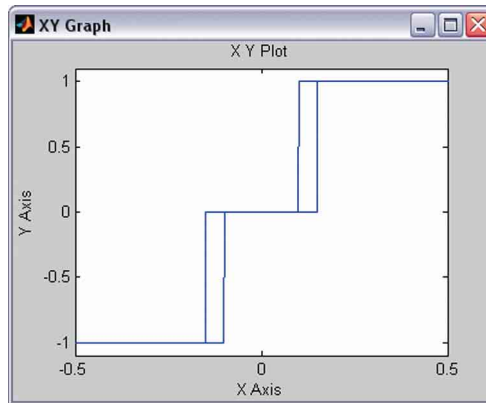
Obr. 6.9-6: Zadání parametrů bloku *Relay*



Obr. 6.9-7: Okno parametrů a vnitřní schéma bloku *Triplokový regulátor*

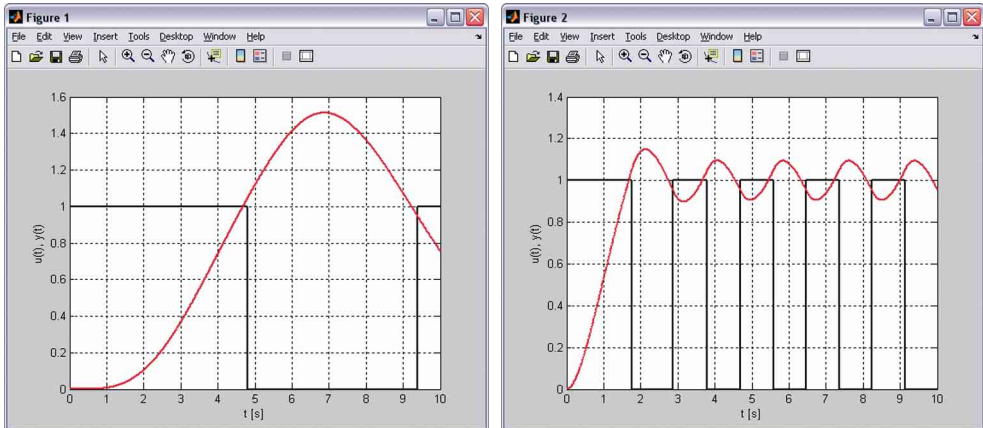


Obr. 6.9-8: Schéma pro zobrazení přepínacích (statických) charakteristik třípolohového regulátoru



Obr. 6.9-9: Přepínací charakteristiky třípolohového regulátoru

Dynamika regulované soustavy má samozřejmě vliv na průběh regulačního pochodu. Výsledky simulace regulačního pochodu při použití dvoupolohového regulátoru pro soustavy G_{S1} a G_{S2} jsou na obr. 6.9-10.



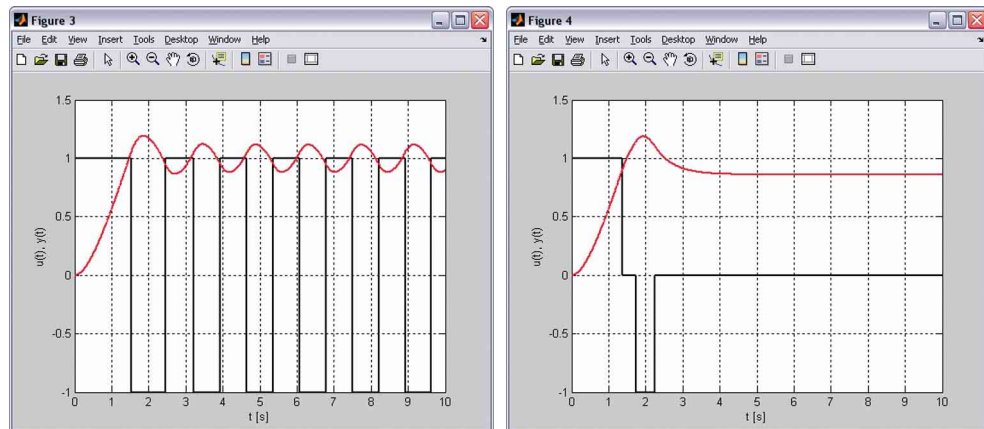
Obr. 6.9-10: Porovnání regulačních pochodů s dvoupolohovým regulátorem pro soustavy G_{S1} a G_{S2}

Na obrázku 6.9-11 jsou pro porovnání uvedeny regulační pochody s dvoupolohovým a třípolohovým regulátorem. Regulovaná soustava je v obou případech stejná (výše byla označena jako G_{S3}) a je astatická.

Zobrazení výsledků je realizováno pomocí speciálního skriptu na obr. 6.9-12. Ukládání jednotlivých veličin, v tomto případě $u(t)$ a $y(t)$ je prováděno v blocích zobrazovačů (*Scope*). Data jsou ukládána do struktury. Princip jejich následného zpracování a zobrazení v grafech je zřejmý přímo z uvedeného skriptu.

Uvedený model regulačního obvodu s nespojitými regulátory umožňuje ověřovat vliv i jiných typů regulovaných soustav. Modifikovat můžeme jejich řád, statické zesílení, časové

konstanty a lze uvažovat i systémy s dopravním zpožděním. Dopravní zpoždění modelujeme pomocí bloku *Transport Delay*, který zařadíme např. za blok soustavy. Vždy sledujeme přesnost regulace a frekvenci přepínání akční veličiny.



Obr. 6.9-11: Porovnání regulačních pochodů s dvoupolohovým regulátorem a třípolohovým regulátorem pro soustavu G_{S3}

Sledovat můžeme i vliv snižující se hystereze h , např. v případě regulované soustavy 1. řádu. Hodnota hystereze v uvedeném modelu regulátoru musí být $h \neq 0$. Na modelu lze demonstrovat důvody uměle zaváděné hystereze.

```

1 - close all, clear all, clc
2 - sim nespojita_regulace
3 - % dvoupolohova regulace - vliv dynamiky soustavy
4 - y1=v1.signals(1).values(:,1); u1=v1.signals(1).values(:,2);
5 - y2=v1.signals(2).values(:,1); u2=v1.signals(2).values(:,2);
6 - figure, plot(t,u1,'k',t,y1,'r','LineWidth',2), grid on
7 - xlabel('t [s]'), ylabel('u(t), y(t)')
8 - figure, plot(t,u2,'k',t,y2,'r','LineWidth',2), grid on
9 - xlabel('t [s]'), ylabel('u(t), y(t)')
10 - % porovnani dvoupolohove a tripolohove regulace
11 - y3=v2.signals(1).values(:,1); u3=v2.signals(1).values(:,2);
12 - y4=v2.signals(2).values(:,1); u4=v2.signals(2).values(:,2);
13 - figure, plot(t,u3,'k',t,y3,'r','LineWidth',2), grid on
14 - xlabel('t [s]'), ylabel('u(t), y(t)')
15 - figure, plot(t,u4,'k',t,y4,'r','LineWidth',2), grid on
16 - xlabel('t [s]'), ylabel('u(t), y(t)')

```

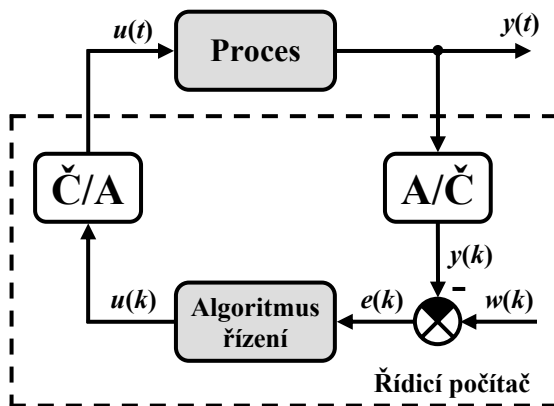
Obr. 6.9-12: Skript pro vykreslení výsledků simulace

Za jistých okolností může pásmo necitlivosti n třípolohového regulátoru způsobit ustálení regulačního pochodu. Na modelu lze ukázat, v jakých případech k tomuto jevu dochází a kdy tomu tak není. Zajímavé je také sledování vlivu pásma necitlivosti na ustálenou hodnotu regulační odchylky $e(\infty)$, viz obr. 6.9-11.

6.10 Číslicové řízení, regulátor PSD

V současné době překotného rozvoje výpočetní techniky a informačních technologií se stále více prosazují v regulační technice specializované počítače a celé počítačové systémy. Plní převážně funkci regulátoru, ale současně zajišťují mnohdy i další funkce – ochranné, signalizační, vizualizační apod. Výhodou těchto řešení je relativně nízká cena, spolehlivost, možnost propojení do průmyslových počítačových sítí, komunikace prakticky bez zkreslení informace i na velké vzdálenosti, téměř neomezená složitost řídicích algoritmů a možnost velmi rychlé změny jejich parametrů.

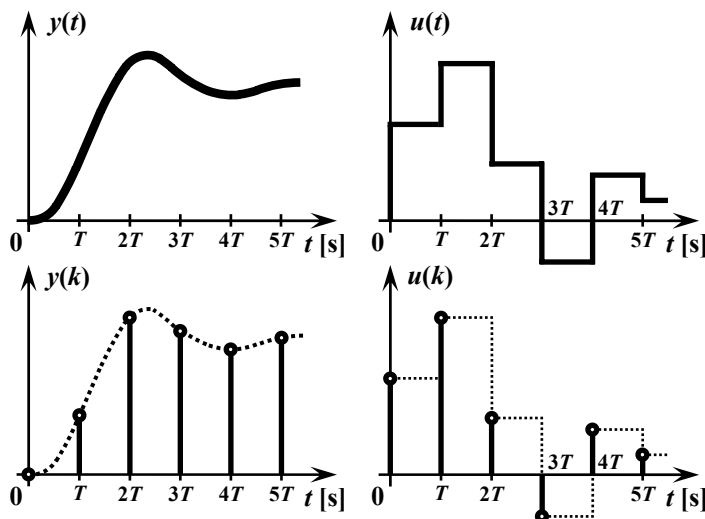
Číslicový počítač pracuje s informací ve formě čísel, kterou mu poskytuje **analogově-číslíkový převodník** A/Č (A/D), viz obr. 6.10-1, který snímá momentální hodnotu spojitého průběhu regulované veličiny $y(t)$. Pracuje na stejném principu jako např. digitální voltmetr, který převádí spojitý průběh měřeného napětí do číselného tvaru. Snímání probíhá obvykle v pravidelných časových okamžicích s tzv. **periodou vzorkování** T . Řídicí počítač dostává informaci o spojitém průběhu $y(t)$ ve formě číselné posloupnosti $y(k)$. Vhodnou hodnotu periody vzorkování získáme jednoduše tak, že dobu praktického ustálení přechodové charakteristiky regulované soustavy (regulovaného technologického procesu) rozdělíme na určitý počet dílků. Pokud je však regulovaná soustava např. kmitavá (tj. má vlastní přirozené frekvence) nebo je regulovaná veličina $y(t)$ zatížena nežádoucím šumem, volíme frekvenci vzorkování alespoň dvojnásobnou než je nejvyšší frekvence, kterou chceme vzorkováním zachytit (tzv. Shannonův či Shannon-Kotělnikovův teorém).



Obr. 6.10-1: Blokové schéma regulačního obvodu s řídicím počítačem

Numerická hodnota regulované veličiny $y(k)$ se porovnává s hodnotou řídicí veličiny $w(k)$ (také v numerickém tvaru). Regulační odchylka $e(k)$ je přirozenou mírou nesouladu mezi požadavkem a skutečnou hodnotou $y(t)$ v aktuálním okamžiku a je postupně po krocích ukládána do registru počítače, ve kterém je vždy několik starších hodnot $e(k)$. Algoritmus řízení je realizován vhodným programem, který z aktuální a minulých hodnot $e(k)$ vypočítává aktuální hodnotu řídicí veličiny $u(k)$. Ta je pomocí **číslíkově-analogového převodníku** Č/A (D/A) převedena na akční veličinu $u(t)$. Vše, se děje opakovaně v pravidelném taktu daném

periodou vzorkování T (někdy též hovoříme o periodě nebo kroku řízení). Z logiky věci vyplývá, že akční veličina $u(t)$ je tzv. **schodová** (po úsecích konstantní) funkce; mění svoji hodnotu vždy v okamžiku nového výpočtu $u(k)$ a zůstává konstantní až do dalšího okamžiku vzorkování, viz obr. 6.10-2.



Obr. 6.10-2: Možné průběhy signálů v regulačním obvodu

Regulátor PSD (Proporcionálně – Sumačně – Diferenční) je diskrétní (numerickou) variantou spojitého PID regulátoru. Spojitý řídicí algoritmus PID regulátoru

$$u(t) = r_0 e(t) + r_1 \int_0^t e(\tau) d\tau + r_2 \frac{de(t)}{dt}$$

je přibližně nahrazen diskrétním výpočtem v k -tém regulačním kroku v čase $t = kT$. Jestliže

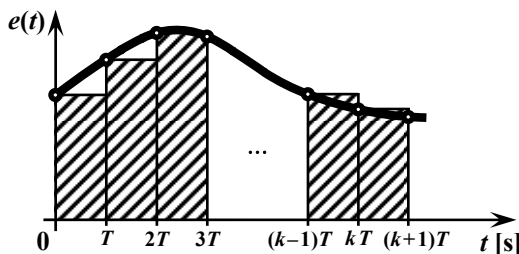
$$r_2 \frac{de(t)}{dt} \approx r_2 \frac{e(kT) - e[(k-1)T]}{T}$$

$$r_1 \int_0^t e(\tau) d\tau \approx r_1 \{e(0)T + e(T)T + e(2T)T + \dots + e[(k-1)T]T\} = r_1 T \sum_{i=0}^{k-1} e(iT)$$

pak

$$u(kT) = r_0 e(kT) + r_1 T \sum_{i=0}^{k-1} e(iT) + r_2 \frac{e(kT) - e[(k-1)T]}{T}$$

Nevýhodou takovéhoho řídicího algoritmu je nutnost si pamatovat celou historii vývoje regulační odchylky $e(iT)$. Velmi elegantně lze tuto nevýhodu obejít výpočtem tzv. **akčního zásahu**, změny (diference) akční veličiny $u(kT)$ v aktuálním k -tém regulačním kroku.



Obr. 6.10-3: K diskrétní náhradě regulátoru

Vypočtíme diferenci

$$\begin{aligned}
 \nabla u(kT) &= u(kT) - u[(k-1)T] = \\
 &= r_0 e(kT) + r_1 T \sum_{i=0}^{k-1} e(iT) + r_2 \frac{1}{T} \{e(kT) - e[(k-1)T]\} - \\
 &\quad - r_0 e(kT) - r_1 T \sum_{i=0}^{k-2} e(iT) - r_2 \frac{1}{T} \{e[(k-1)T] - e[(k-2)T]\} = \\
 &= e(kT) \left[r_0 + \frac{r_2}{T} \right] + e[(k-1)T] \left[r_1 T - r_0 - \frac{2r_2}{T} \right] + e[(k-2)T] \frac{r_2}{T} = \\
 &= b_0 e(kT) + b_1 e[(k-1)T] + b_2 e[(k-2)T]
 \end{aligned}$$

kde

$$b_0 = r_0 + \frac{r_2}{T}, \quad b_1 = r_1 T - r_0 - \frac{2r_2}{T}, \quad b_2 = \frac{r_2}{T}$$

Následným přičtením vypočtené difference k minulé hodnotě akční veličiny $u[(k-1)T]$ získáme její aktuální hodnotu

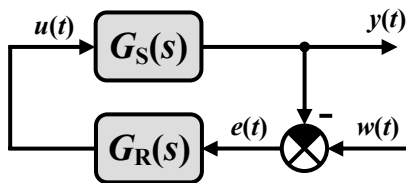
$$u(kT) = u[(k-1)T] + \nabla u(kT) = u(k)$$

Takto vytvořený **rekurzivní algoritmus** potřebuje k výpočtu akční veličiny $u(kT)$ pouze současnou a dvě starší hodnoty regulační odchylky $e(kT)$, $e[(k-1)T]$ a $e[(k-2)T]$. Výpočet je velmi rychlý, bez nároků na paměť počítače. Známe-li seřízení spojitého PID regulátoru, lze velmi snadno nalézt při daném regulačním kroku jeho PSD diskrétní ekvivalent podle uvedených vztahů. Poznamenejme, že existují i jiné metody numerické integrace, než kterou jsme použili výše pro náhradu spojitého integrálu. Zvolíme-li jinou metodu (např. lichoběžníkovou aproximaci), přepočtení vztahy se poněkud modifikují.

Nalezněme nyní číslicovou PS variantu spojitého PI regulátoru regulačního obvodu podle obr. 6.10-4. Uvažujme regulovanou soustavu popsanou přenosem

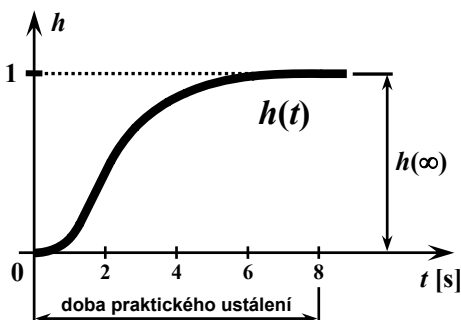
$$G_S(s) = \frac{6}{s^3 + 6s^2 + 11s + 6}$$

Koeficienty PI regulátoru jsou $r_0 = 0,9$ a $r_1 = 1$.



Obr. 6.10-4: Blokové schéma regulačního obvodu

Pro volbu periody vzorkování T vycházíme z doby praktického ustálení přechodové charakteristiky regulované soustavy (obr. 6.10-5, zde ca 8 s), zvolme $T = 8 / 80 = 0,1$ s. Vzorkovací frekvence $\omega_v = 2\pi / T = 2\pi / 0,1 = 20\pi \approx 62,83 \text{ rad s}^{-1} \sim 10 \text{ Hz}$.



Obr. 6.10-5: Přechodová odezva regulované soustavy

Podle Shannonova teorému tímto vzorkováním zachytíme frekvence jen maximálně dvakrát pomalejší, tedy 5 Hz a nižší. Pokud bychom chtěli zachytit frekvence vyšší, musíme volit periodu vzorkování kratší.

Podle výše odvozených vztahů vypočteme koeficienty numerického algoritmu řízení s tím, že v případě regulátoru PI derivační složka chybí, tedy $r_2 = 0$

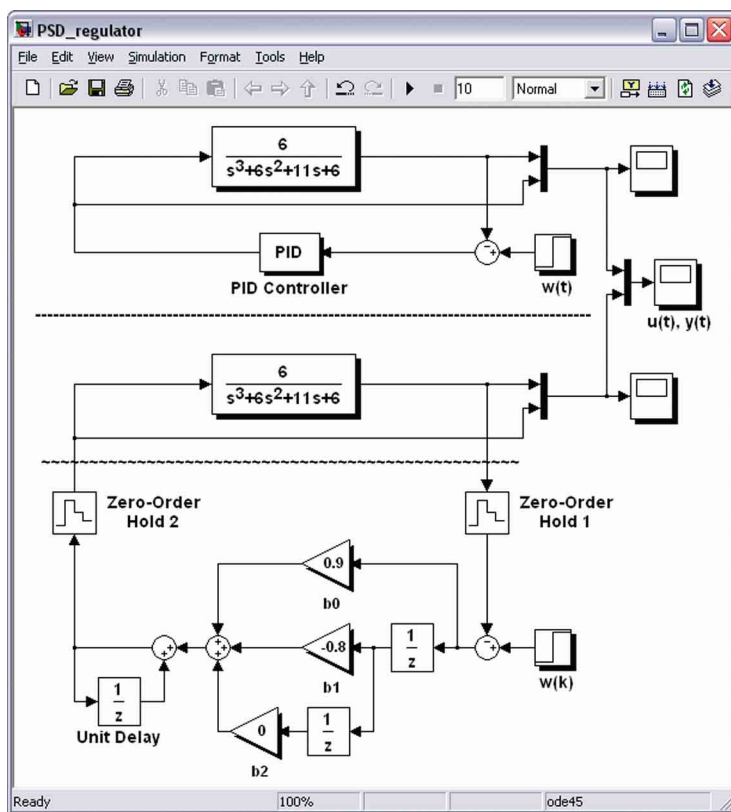
$$b_0 = r_0 + \frac{r_2}{T} = 0,9, \quad b_1 = r_1 T - r_0 - \frac{2r_2}{T} = 1 \cdot 0,1 - 0,9 = -0,8, \quad b_2 = \frac{r_2}{T} = 0$$

$$\nabla u(kT) = b_0 e(kT) + b_1 e[(k-1)T] + b_2 e[(k-2)T] = 0,9e(kT) - 0,8e[(k-1)T]$$

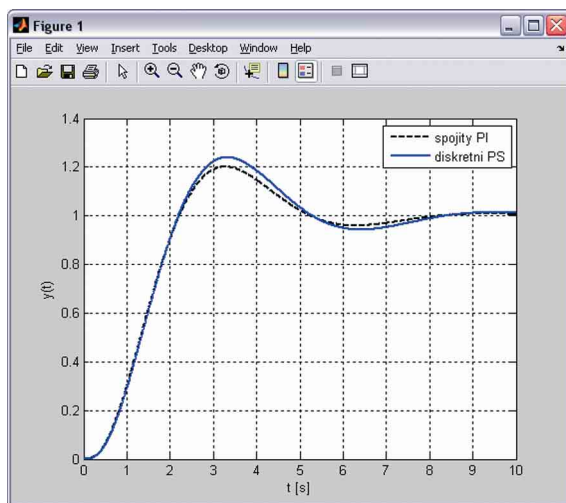
$$u(kT) = u[(k-1)T] + \nabla u(kT)$$

Pro výpočet akční veličiny $u(kT)$ v aktuálním čase $t = kT$ je třeba znalost pouze aktuální hodnoty regulační odchylky $e(kT)$ a její hodnoty v předcházejícím kroku $e[(k-1)T]$, kterou si musíme vždy pamatovat. Algoritmus je velmi jednoduchý. Odpovídající simulační schéma v programu Simulink je na obr. 6.7-7. Je porovnáváno chování spojitého PI regulátoru a diskrétního PS regulátoru.

Simulační blok **Unit Delay** ($1/z$) realizuje diskrétní zpoždění informace o jeden krok, blok **Zero-Order Hold** plní funkci A/Č a Č/A převodníků (vzorkovač, tvarovač). U obou diskrétních bloků je nutné nastavit periodu vzorkování $T = 0,1$ s.

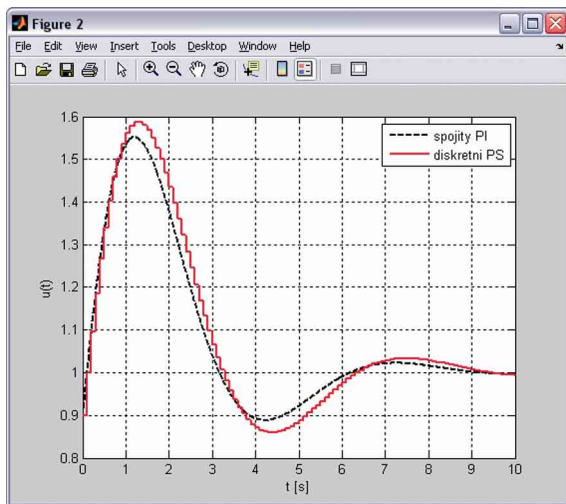


Obr. 6.10-6: Regulační obvod s PSD regulátorem



Obr. 6.10-7: Porovnání průběhů regulované veličiny

Regulační obvod se spojitým PI i číslicovým PS regulátorem vykazuje velmi podobné chování odpovídající přibližné náhradě spojitého regulátoru, viz obr. 6.10-7 a 6.10-8.



Obr. 6.10-8: Porovnání průběhů akční veličiny

6.11 Numerická optimalizace parametrů regulátoru

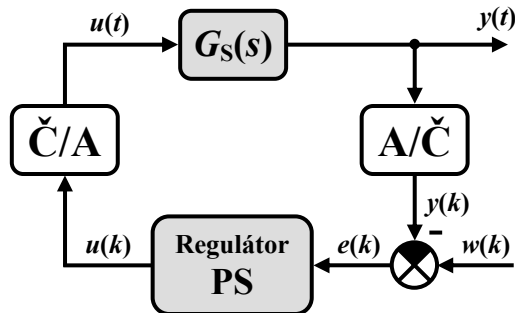
Pokud chceme rozhodnout o tom, zda je něco lepší než něco jiného, musíme mít vždy jasné měřítko, podle kterého kvalitu posuzujeme. V případě regulačních obvodů může být měřítkem kvality např. čas ustálení regulačního pochodu, rychlost náběhu, velikost překyvů či cena spotřebované energie. Častým požadavkem bývá, aby pochody byly bez kmitavých složek (rozjíždění výtahu, brzdění vagonu metra, apod.). Většinou záleží na charakteru řízeného procesu a na tom, co od regulace očekáváme. Často bývají požadavky protichůdné a je potřeba hledat vhodný kompromis. Důležitou roli mnohdy hraje technický cit a pouhé subjektivní posouzení průběhu expertem.

Velmi rozšířenými, klasickými kritérii v regulační technice jsou tzv. **integrální kritéria**. Nejčastěji posuzují průběh regulační odchylky $e(t)$ a akční veličiny $u(t)$. Regulační odchylka je přirozenou kritériální veličinou, která vyjadřuje rozdíl mezi požadavkem a skutečnou momentální hodnotou veličiny, kterou regulujeme. Posuzujeme-li kvalitu regulace podle integrálu regulační odchylky v průběhu celého regulačního pochodu, teoreticky od nuly (od okamžiku vstupu předpokládané poruchy) až do nekonečna (je to plocha sevřená průběhem $e(t)$ a časovou osou), mluvíme o tzv. **lineární regulační ploše**. Pokud je kritériálním zhodnocením integrál z kvadrátu regulační odchylky, jedná se o tzv. **kvadratickou regulační plochu**. Kvadratická regulační plocha, na rozdíl od lineární, obsahuje explicitně i podmínky stability, její minimalizaci zajistíme vždy stabilitu obvodu. Rozšířením kritéria o integrál akční veličiny $u(t)$ zvažujeme i energii potřebnou k realizaci celého regulačního pochodu. Zvyšováním váhy na akční veličinu regulační pochod zpomalujeme a zklidňujeme.

Vyjádření kvadratického funkcionálu v lineárních případech (Parcevalův integrál) není příliš obtížné a existují i analytické prostředky pro jeho optimalizaci. To je i jeden z důvodů velké obliby a rozšíření integrálních kritérií. Nicméně, ve složitějších případech (vyšší řád, rozvětvená struktura regulačního obvodu) se stává optimalizační úloha, byť matematicky velmi zajímavá, bohužel velmi komplikovanou a pracnou. Alternativou je numerický přístup, tedy iterační způsob optimalizace kritéria. Je to velmi účinný a univerzální nástroj, který nám umožňuje nalézt optimální řešení i velmi složitých regulačních úloh s prakticky libovolným kritériem. Je použitelný bez větších problémů i v případě nelineárních obvodů, kdy dříve uvedené analytické přístupy zcela selhávají.

Numerická iterační optimalizace je univerzálním nástrojem, který se uplatní nejen při optimalizaci regulačních obvodů, ale v různých variantách i v široké škále technických aplikací. Všude tam, kde je třeba hledat vhodnou kombinaci volitelných parametrů tak, aby výsledné řešení odpovídalo co nejlépe zvolené podmínce. Metoda je založena na opakovaném (iteračním) výpočtu zadané úlohy s různými parametry a výběrem jejich nejlepší kombinace. Jedná se tak ve své podstatě o úlohu hledání extrému funkce více proměnných, pro kterou existuje v numerické matematice řada účinných metod (simplexová, Newtonova, gradientní metody, atd.) a přirozeně je jimi MATLAB jako standardními procedurami vybaven.

Přestože není iterační princip vázán na konkrétní program nebo programovací jazyk, ukážeme si jeho aplikaci na konkrétním příkladě v prostředí MATLAB – Simulink. Mějme za úkol naleznout optimální kombinaci parametrů číslicového PS regulátoru tak, abychom minimalizovali hodnotu rozšířeného kvadratického kritéria J s penalizací akčních zásahů.



Obr. 6.11-1: Blokové schéma regulačního obvodu

Regulovaná soustava (obr. 6.11-1) je definována přenosem

$$G_S(s) = \frac{6}{s^3 + 6s^2 + 11s + 6}$$

Algoritmus vytváření akční veličiny PS regulátorem (byl popsán již v kapitole 6.10) je

$$\nabla u(kT) = b_0 e(kT) + b_1 e[(k-1)T]$$

$$u(kT) = u[(k-1)T] + \nabla u(kT)$$

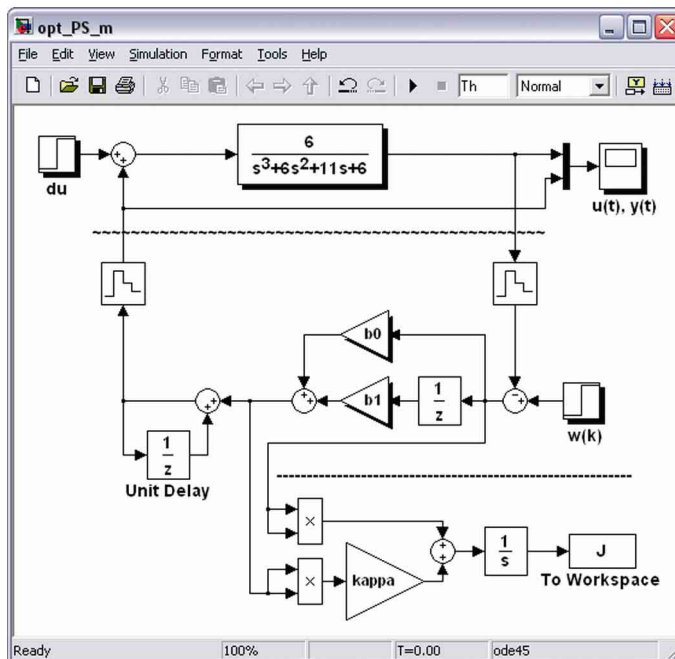
Rozšířené kvadratické kritérium s penalizací akčních zásahů je vyjádřeno vztahem

$$J = \sum_{k=0}^{K_h} e^2(kT) + \kappa \nabla u^2(kT) = J(b_0, b_1) \rightarrow \min$$

kde K_h představuje počet kroků regulačního pochodu, ve kterých kritérium posuzujeme (tzv. kritériální horizont) a κ váhu zásahů ∇u v kritériu.

V našem případě zvolíme $K_h = 100$ (tak, aby během K_h kroků došlo k ustálení regulačního pochodu) a $\kappa = 1$ (velké, regulace bude pomalejší). Počáteční nastavení regulátoru na začátku iteračního výpočtu volíme $b_0 = 4,5$ a $b_1 = -4,215$ (ekvivalent seřízení Ziegler-Nicholsovou metodou kritického zesílení regulátoru, tedy silná zpětná vazba, rychlý kmitavý regulační pochod).

Optimalizační proces je založen na spolupráci tří programů, které musí být umístěny v aktuálním adresáři MATLABu. Model regulačního obvodu (obr. 6.11-2) `opt_PS_m.mdl` rozšířený o výpočet kritériálního funkcionálu J je opakovaně spouštěn v prostředí MATLAB Simulink s různými parametry regulátoru. Proces je řízen hlavním programem `opt_PS.m` (obr. 6.11-3), který využívá standardní proceduru MATLABu pro hledání extrému funkcí více proměnných. Optimalizovanou funkci definuje skript `opt_krit.m` (obr. 6.11-4).



Obr. 6.11-2: Model regulačního obvodu

V bloku *To Workspace* je nutné nastavit parametr **Limit data points to last** na 1, v bloku *Unit Delay* a ve vzorkovači a tvarovači pak nastavit periodu vzorkování $T = 0,1$ s.

```

%
% opt_PS    Optimalizace parametru PS regulatoru
%           Kvadraticke kritérium s penalizací akčních zásahů
%
clear all, close all, clc, format long, format compact
disp(' ')
disp('*****')
disp('* Minimalizace kvadratickeho kriteria *')
disp('* Kvadraticke kritérium s penalizací akčních zásahů *')
disp('*****')
disp(' ')

global b0 b1 Tmax kappa

Th=10;           % zadana doba simulace (kritériální horizont)
dT=0.001;        % pevne zadany simulacni krok pro opt_PS_m.mdl
kappa=1;         % vaha akčních zásahů v kritériu
                % (pro male kappa rychlé reg. pochody a naopak)
w=1; du=0;       % reakce na zmenu zadane hodnoty
%w=0; du=1;      % reakce pri poruse na akci velicine (na vstupu)

b0=4.5; b1=-4.215; % parametry regulatoru na zacatku optimalizace
x0=[b0 b1];       % volba vektoru optimalizovanych parametru
sim('opt_PS_m');  % spusti simulaci (soubor je v akt. adresari)

disp('Vychodi situace:')
disp(' ')
b0, b1, J
disp(' ')
disp('(b0, b1 ... parametry PS regulatoru, J ... hodnota funkcionalu)')
disp(' ')
opt_PS_m         % otevre okno se simulacnim schematem
figure(1);       % otevre okno a vykresli reg. pochod s poc. nast.
plot(t,v(:,2),'k','LineWidth',2), grid on, hold on
figure(2);       % otevre okno a vykresli prubeh akčních veliciny
plot(t,v(:,3),'k','LineWidth',2), grid on, hold on
disp('Pokracuj stiskem libovolne klavesy'), pause

disp(' ')
disp('*****')
disp('Zacatek iterace ...')
disp(' ')
% HLAVNI CAST OPTIMALIZACNIHO VYPOCTU
% OPTIONS - parametry optimalizace (par. MaxIter udava max. pocet kroku)
OPTIONS=optimset('LargeScale','off','MaxIter',20,'Display','iter');
%OPTIONS=optimset('LargeScale','off','MaxIter',200,'Display','iter',...
% 'TolFun',1e-10,'TolX',1e-10);

x=fminsearch('opt_krit',x0,OPTIONS); % optimalizace simplexovou metodou
%(1. par. je minimalizovana ucelova funkce definovana funkci opt_krit.m
%x = fminunc('opt_krit',x0,OPTIONS); % optimalizace gradientni metodou

disp(' ')
disp('Konec iterace')
disp('*****')
disp(' ')

b0opt=x(1); b1opt=x(2);
b0=b0opt; b1=b1opt; sim('opt_PS_m')

```

Obr. 6.11-3: Skript řídicí výpočet (opt_PS.m)

```

disp('Vysledek optimalizace:')
disp(' ')
b0opt, b1opt, J
disp(' ')
disp('(b0opt, b1opt ... nalezene optimalni parametry PS regulatoru)')
disp(' ')

figure(1), plot(t,v(:,2),'r','LineWidth',2) %vykresli opt. reg. pochod
xlabel('t [s]'), ylabel('y(t)')
figure(2), plot(t,v(:,3),'r','LineWidth',2) %vykresli prubeh akcni vel.
xlabel('t [s]'), ylabel('u(t)')

disp('*****')
disp('* Konec optimalizace kvadratickeho funkcionalu *')
disp('*****')
disp(' ')

```

Obr. 6.11-3: pokračování

```

%
% opt_krit   Konstrukce minimalizovane ucelove funkce
%            vektorovy parametr x funkce obsahuje koeficienty regulatoru
%            f je hodnota kriterialniho ohodnoceni J

function f=opt_krit(x)
global b0 b1 Th kappa

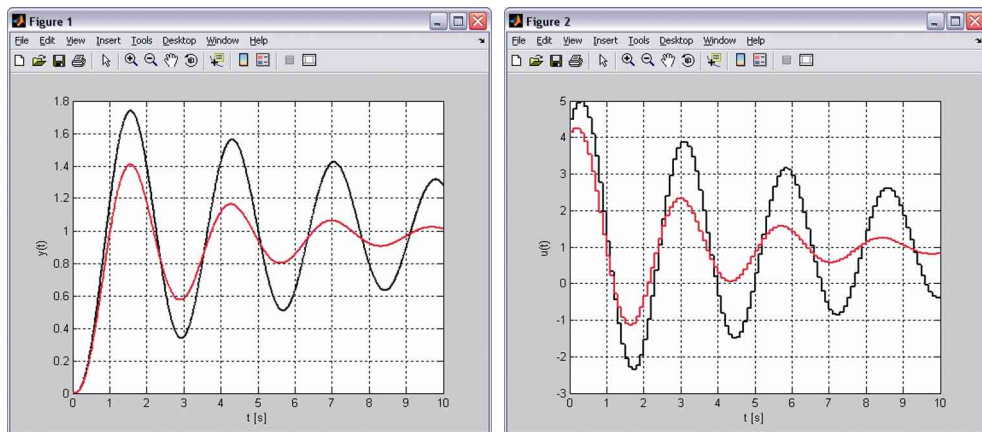
b0=x(1); b1=x(2); % naplneni vektoru x koeficienty regulatoru

%simulace reg. obvodu s aktualnimi koeficienty regulatoru
sim('opt_PS_m',Th) % spusti opt_PS_m.mdl
f=J;               % vraci krit. ohodnoceni vypoctene opt_PS_m.mdl

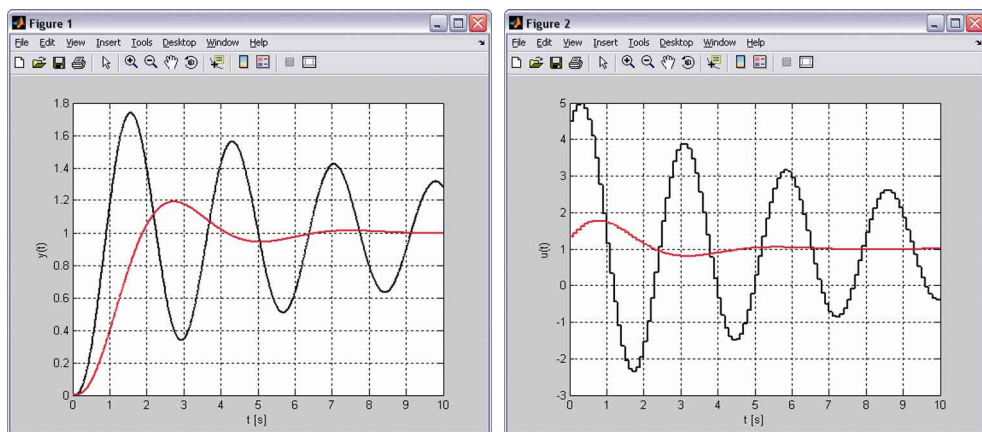
```

Obr. 6.11-4: Funkce pro výpočet hodnoty kritéria (opt_krit.m)

Výsledné regulační pochody, průběhy regulované a akční veličiny, jsou na obr. 6.11-5. Byly obdrženy po provedení 10 iterací (kroků optimalizace). Pro porovnání jsou vykresleny i průběhy při počátečních hodnotách parametrů regulátoru $b_0 = 4,5$ a $b_1 = -4,215$.


Obr. 6.11-5: Průběh regulačního pochodu při $w = \eta(t)$ a $u_v = 0$ na počátku a po 10 iteracích

Průběhy regulované a akční veličiny na obr. 6.11-6 byly získány po provedení 20 iterací. Je zřejmé, že regulační pochod je již výrazně méně kmitavý než v předchozím případě. Na obr. 6.11-8 je také uveden kompletní výpis průběhu výpočtu (výpis okna *Command Window* MATLABu).



Obr. 6.11-6: Průběh regulačního pochodu $w = \eta(t)$ a $d_u = 0$ na počátku a po 20 iteracích

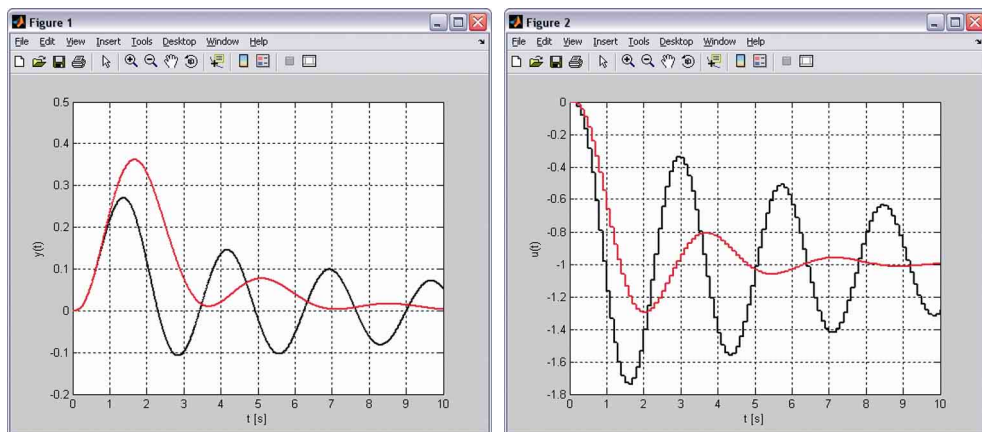
Po ukončení výpočtu byl jako optimální vybrán řídicí algoritmus minimalizující zadané kritérium

$$\nabla u(kT) = 1,343e(kT) - 1,241e[(k-1)T]$$

$$u(kT) = u[(k-1)T] + \nabla u(kT)$$

V průběhu optimalizace došlo k vývoji parametrů PS regulátoru a kritéria jakosti

$$b_0 = 4,5 \rightarrow 1,343, \quad b_1 = -4,215 \rightarrow -1,241, \quad J = 5,518 \rightarrow 1,158$$



Obr. 6.11-7: Průběh regulačního pochodu při $w = 0$ a $d_u = \eta(t)$ na počátku a po 45 iteracích

```

*****
* Minimalizace kvadratickeho kriteria                                     *
* Kvadraticke kritérium s penalizací akčních zásahů                       *
*****

Vychodi situace:

b0 =
    4.500000000000000
b1 =
   -4.215000000000000
J =
    5.51754703414972

(b0, b1 ... parametry PS regulatoru, J ... hodnota funkcionalu)

Pokracuj stiskem libovolne klavesy

*****
Zacatek iterace ...

  Iteration   Func-count   min f(x)      Procedure
        0           1       5.51755
        1           3       3.55176    initial simplex
        2           5       3.37381    contract outside
        3           7       3.37381    contract inside
        4           9       3.37381    contract inside
        5          11       3.37381    contract inside
        6          13       3.30405    reflect
        7          15       3.30405    contract inside
        8          17       3.27281    expand
        9          19       3.10336    expand
       10          21       3.04325    expand
       11          23       2.61162    expand
       12          24       2.61162    reflect
       13          26       1.9294    expand
       14          27       1.9294    reflect
       15          29       1.34946    expand
       16          31       1.22409    reflect
       17          33       1.16306    contract outside
       18          35       1.16306    contract outside
       19          37       1.16306    contract inside
       20          39       1.15837    contract inside

Exiting: Maximum number of iterations has been exceeded
- increase MaxIter option.
Current function value: 1.158374

Konec iterace
*****

Vysledek optimalizace:

b0opt =
    1.34289493560781
b1opt =
   -1.24081651210787
J =
    1.15837446305066

(b0opt, b1opt ... nalezene optimalni parametry PS regulatoru)

*****
* Konec optimalizace kvadratickeho funkcionalu                             *
*****
>> |

```

Obr. 6.11-8: Výpis průběhu optimalizačního výpočtu

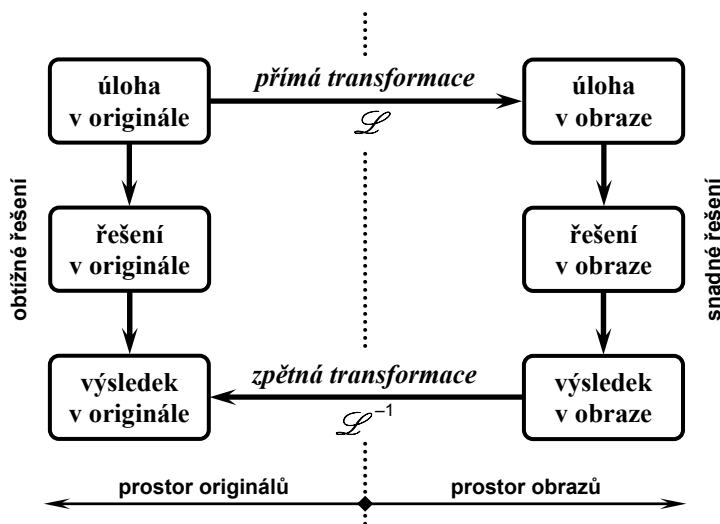
Na obr. 6.11-7 jsou uvedeny také průběhy regulačního pochodu při působení poruchy na vstupu soustavy (akční veličině). V tomto případě byla zvýšena váha zásahů ∇u v kritériu na $\kappa = 10$.

Literatura

- [1] Myslík, J.: *Elektrické obvody: příručka pro praxi a učebnice pro střední a vysoké školy*. 2. vyd. Praha: BEN – technická literatura, 1998. ISBN 80-86056-19-8.
- [2] Myslík, J.: *Elektrické obvody – řešené příklady: příručka pro praxi a učebnice pro střední a vysoké školy*. 2. vyd. Praha: BEN – technická literatura, 1998. ISBN 80-86056-33-3.
- [3] Noskievič, P.: *Modelování a identifikace systémů*. 1. vyd. Ostrava: Montanex, 1999. ISBN 80-7225-030-2.
- [4] Kupka, L.: *Matlab & Simulink: úvod do použití*. 1. vyd. Lanškroun: SOŠ a SOU, 2007. ISBN 978-80-239-8871-0.
- [5] Balátě, J.: *Automatické řízení*. 2. vyd. Praha: BEN – technická literatura, 2004. ISBN 80-7300-148-9.
- [6] Lepil, O. a kol.: *Fyzika pro střední školy, I. díl*. 4. vyd. Praha: Prometheus, 1998. ISBN 80-7196-184-1.
- [7] Bartuška, K.: *Sbírka řešených úloh z fyziky pro střední školy I*. 1. vyd. Praha: Prometheus, 1998. ISBN 80-7196-033-0.
- [8] Barták, F.: *Sbírka úloh z fyziky pro studijní obory SOU a SOŠ*. 1. vyd. Praha: SPN, 1988. ISBN 80-04-24967-1.
- [9] Jor, Z. a kol.: *Malé analogové počítače ve školní praxi: učební pomůcka pro 4. ročník středních průmyslových škol elektrotechnických, chemických a strojnických*. 1. vyd. Praha: SNTL, 1971.
- [10] Topčejev, J. I., Cypljakov, A. P.: *Sbírka příkladů z teorie automatického řízení*. 1. vyd. Praha: SNTL, 1983.
- [11] Hlava, J.: *Prostředky automatického řízení II: analogové a číslicové regulátory, elektrické pohony, průmyslové komunikační systémy*. 1. vyd. Praha: ČVUT, 2000. ISBN 80-01-02221-8.
- [12] Techmation: *Applications Manual*. [Online.] Tempe (USA), 1999. [cit. květen 2007.] Dostupné na URL: <http://www.protuner.com/appman.pdf>
- [13] Zítek, P., Hofreiter, M., Hlava, J.: *Automatické řízení*. 2. vyd. Praha: ČVUT, 2001. ISBN 80-01-02044-4.
- [14] Nagrath, I. J., Gopal, M.: *Control System Engineering*. 2. vyd. New York (USA): John Wiley & Sons, 1982. ISBN 0-471-09814-0.
- [15] Pírko, Z., Veit, J.: *Laplaceova transformace: základy teorie a užití v elektrotechnice*. 1. vyd. Praha: SNTL, 1970.
- [16] Vítečková, M., Víteček, A.: *Základy automatické regulace*. 1. vyd. Ostrava: VŠB – TU, 2006. ISBN 80-248-1068-9.
- [17] Balda, M., Bošek, B., Dráb, Z.: *Základy automatizace I*. 1. vyd. Praha: SNTL, 1968.

PŘÍLOHA A: Laplaceova integrální transformace

Na obr. A-1 je uvedeno schéma řešení úlohy pomocí Laplaceovy transformace. Úloha (technický problém) je obvykle popsána diferenciálními rovnicemi, jejichž klasické řešení je obtížné. S využitím transformace převedeme diferenciální rovnice na rovnice algebraické, jež jsou snadno řešitelné. Obraz výsledku následně převedeme zpět do časové oblasti.



Obr. A-1: Schéma řešení úlohy pomocí Laplaceovy transformace

Současně odpadá potřeba určování hodnot integračních konstant z počátečních podmínek. Počáteční podmínky jsou již ve výsledcích v obraze obsaženy. Odpadají rovněž problémy s nespojitostmi při derivacích na pravé straně.

Laplaceova integrální transformace je definována vztahem

$$\mathcal{L}\{f(t)\} = \int_0^{\infty} f(t) e^{-st} dt = F(s)$$

Aby integrál existoval (tedy aby byl integrabilní ve smyslu Cauchyově-Riemanově), musí být funkce $f(t)$ jednoznačná a po částech spojitá v každém konečném intervalu proměnné t , a to pro všechna $t > 0$, a aby konvergoval, musí být $f(t)$ exponenciálního řádu pro $\lim_{t \rightarrow \infty} t$. Podmínka této konvergence vyžaduje u většiny funkcí, aby $f(t) = 0$ pro $t < 0$. Funkci $f(t)$ nazýváme **originálem** (předmětem), funkci $F(s)$ **obrazem** tohoto originálu. Transformaci značíme symbolicky $\mathcal{L}$. Některé nejběžnější originály a příslušné obrazy jsou uvedeny ve slovníku korespondencí v tab. A-2.

Lineární časově invariantní systém (jeho vlastnosti se v čase nemění) s jednou vstupní a jednou výstupní veličinou bývá obvykle popsán lineární diferenciální rovnicí s konstantními koeficienty. Obecná závislost mezi vstupním signálem $f_1(t)$ a výstupním signálem $f_2(t)$ je dána rovnicí

$$\begin{aligned} a_n f_2^{(n)}(t) + a_{n-1} f_2^{(n-1)}(t) + \dots + a_1 f_2'(t) + a_0 f_2(t) = \\ = b_m f_1^{(m)}(t) + b_{m-1} f_1^{(m-1)}(t) + \dots + b_1 f_1'(t) + b_0 f_1(t) \end{aligned}$$

kde a_i, b_i jsou konstantní koeficienty. Z podmínky fyzikální realizovatelnosti systému musí být $m \leq n$, tj. stupeň nejvyšší derivace výstupní veličiny musí být větší nebo roven stupni nejvyšší derivace vstupní veličiny.

Obraz derivace funkce závisí též na počátečních podmínkách vzatých jako limita zprava, a to podle vztahu

$$\mathcal{L}\left\{\frac{df(t)}{dt}\right\} = sF(s) - f(0+)$$

kde označení $0+$ znamená hodnotu funkce nebo její derivace pro t blížící se $t = 0$ zprava. Podobně pro libovolnou vyšší derivaci obdržíme vztah

$$\mathcal{L}\left\{\frac{d^n f(t)}{dt^n}\right\} = s^n F(s) - [s^{n-1} f(0+) + s^{n-2} f'(0+) + \dots + s f^{(n-2)}(0+) + f^{(n-1)}(0+)]$$

U regulačních obvodů obvykle předpokládáme, že až do okamžiku vstupu poruchy (tj. pro $t < 0$) byl obvod a tedy i všechny jeho členy v klidu. Pak lze provést transformaci rovnice obecné závislosti mezi vstupním a výstupním signálem člen po členu (s ohledem na linearitu transformace). Např. pro člen $a_0 f_2(t)$ bude

$$\mathcal{L}\{a_0 f_2(t)\} = \int_0^{\infty} a_0 f_2(t) e^{-st} dt = a_0 \int_0^{\infty} f_2(t) e^{-st} dt$$

kde však výraz $\int_0^{\infty} f_2(t) e^{-st} dt$ je podle definice obrazem funkce $f_2(t)$, a tedy

$$\mathcal{L}\{a_0 f_2(t)\} = a_0 F_2(s)$$

Pro člen $a_1 f_2'(t)$ bude

$$\mathcal{L}\{a_1 f_2'(t)\} = \int_0^{\infty} a_1 \frac{df_2(t)}{dt} e^{-st} dt = a_1 \int_0^{\infty} \frac{df_2(t)}{dt} e^{-st} dt$$

kde integrál představuje obraz derivace funkce $f_2(t)$ a má při nulové počáteční podmínce hodnotu $sF_2(s)$, bude tedy

$$\mathcal{L}\{a_1 f_2'(t)\} = a_1 s F_2(s)$$

a obdobně pro vyšší derivace.

Transformujeme-li takto postupně všechny členy, dostaneme

$$[a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0] F_2(s) = [b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0] F_1(s)$$

Poměr

$$\frac{F_2(s)}{F_1(s)} = \frac{b_m s^m + b_{m-1} s^{(m-1)} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{(n-1)} + \dots + a_1 s + a_0} = F(s)$$

je tzv. **obrazový přenos**, který je tedy vlastně roven

$$F(s) = \frac{\mathcal{L}\{f_2(t)\}}{\mathcal{L}\{f_1(t)\}}$$

Přenos systému (soustavy) je definován jako poměr Laplaceova obrazu výstupního signálu $f_2(t)$ k Laplaceově obrazu vstupního signálu $f_1(t)$ při nulových počátečních podmínkách $f_2(0) = f_2'(0) = \dots = f_2^{(n-1)}(0) = 0$ a $f_1(0) = f_1'(0) = \dots = f_1^{(m-1)}(0) = 0$.

Zpětná transformace (přechod od obrazu k originálu) je definována integrálem

$$\mathcal{L}^{-1}\{F(s)\} = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} F(s) e^{st} ds = f(t)$$

kde c je úsečka absolutní konvergence (musí být větší, než jsou reálné části všech singularit obrazu $F(s)$).

Pro úplnost uvedme bez důkazu několik pravidel platných v Laplaceově transformaci, která jsou důležitá pro řešení úloh v teorii automatického řízení. Přehled těchto pravidel je uveden v tab. A-1. Předpokládáme, že originál $f(t)$ je funkce po částech spojitá v intervalu $\langle 0, +\infty \rangle$, exponenciálního řádu s indexem růstu α .

Tab. A-1: Přehled základních vzorců Laplaceovy transformace

	Název	Vzorec
1	Definice přímé transformace	$\mathcal{L}\{f(t)\} = \int_0^{\infty} f(t) e^{-st} dt = F(s)$ (regulární pro $\text{Re}[s] > \alpha$)
2	Linearita	$\mathcal{L}\{c_1 f_1(t) \pm c_2 g(t)\} =$ $= c_1 \mathcal{L}\{f_1(t)\} \pm c_2 \mathcal{L}\{f_1(t)\}$ (platí obdobně pro libovolný počet členů)
3	Posun v originále (translace, časové zpoždění)	$\mathcal{L}\{f(t-a)\} = e^{-as} F(s)$ $a \geq 0, f(t) = 0 \text{ pro } t < 0$
4	Posun v obraze (substituce)	$\mathcal{L}\{e^{at} f(t)\} = F(s-a)$
5	Konvoluce	$\mathcal{L}\left\{\int_0^t f_1(t-\tau) f_2(\tau) d\tau\right\} = F_1(s) F_2(s)$

Tab. A-1: pokračování

	Název	Vzorec
6	Změna měřítka	$\mathcal{L}\{f(kt)\} = \frac{1}{k} F\left(\frac{s}{k}\right), \quad \mathcal{L}\left\{f\left(\frac{t}{k}\right)\right\} = k F(sk)$ (platí pro $k > 0$)
7	Věta o počáteční hodnotě	$\lim_{t \rightarrow 0+} f(t) = \lim_{s \rightarrow \infty} s F(s)$
8	Věta o konečné hodnotě	$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} s F(s)$ (platí pouze tehdy, jestliže limita existuje)
9	Derivace obrazu	$\mathcal{L}\{t f(t)\} = -F'(s)$
10	Integrál obrazu	$\mathcal{L}\left\{\frac{f(t)}{t}\right\} = \int_s^{\infty} F(q) dq$
11	Obraz derivace	$\mathcal{L}\{f'(t)\} = s F(s) - f(0+)$ $\mathcal{L}\{f^{(n)}(t)\} = s^n F(s) - [s^{n-1} f(0+) + s^{n-2} f'(0+) + \dots + s f^{(n-2)}(0+) + f^{(n-1)}(0+)]$
12	Obraz integrálu	$\mathcal{L}\left\{\int_0^t f(\tau) d\tau\right\} = \frac{1}{s} F(s)$
13	Obraz periodické funkce (s periodou T)	$\mathcal{L}\{f(t)\} = \frac{\int_0^T f(t) e^{-st} dt}{1 - e^{-sT}}$
14	Definice zpětné transformace	$\mathcal{L}^{-1}\{F(s)\} = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} F(s) e^{st} ds = f(t)$
15	Zpětná transformace pomocí vzorce a pomocí reziduí	$f(t) = \sum_{i=1}^p \lim_{s \rightarrow s_i} \frac{1}{(r_i - 1)!} \frac{d^{r_i-1}}{ds^{r_i-1}} [(s - s_i)^{r_i} F(s) e^{st}]$ $f(t) = \sum_{i=1}^p \operatorname{res}_{s=s_i} [F(s) e^{st}]$ pro $F(s) = \frac{P(s)}{Q(s)}$, p pólů r_i-násobných

Tab. A-2: Slovník základních korespondencí Laplaceovy transformace

	Obraz $F(s)$	Originál $f(t)$
1	1	$\delta(t)$ (jednotkový impuls, Diracova funkce)
2	$\frac{1}{s}$	$\eta(t) = \begin{cases} 0 & \text{pro } t < 0 \\ 1 & \text{pro } t \geq 0 \end{cases}$ (jednotkový skok, Heavisideova funkce)
3	$\frac{1}{s^2}$	t
4	$\frac{1}{s^n}$	$\frac{t^{n-1}}{(n-1)!}$ pro $n = 1, 2, 3, \dots$
5	$\frac{1}{s+a}$	e^{-at}
6	$\frac{1}{(s+a)^2}$	$t e^{-at}$
7	$\frac{1}{(s+a)^n}$	$\frac{t^{n-1}}{(n-1)!} e^{-at}$ pro $n = 1, 2, 3, \dots$
8	$\frac{1}{s(s+a)}$	$\frac{1}{a} (1 - e^{-at})$
9	$\frac{1}{s^2(s+a)}$	$\frac{1}{a^2} (at - 1 + e^{-at})$
10	$\frac{s}{(s+a)^2}$	$(1-at)e^{-at}$
11	$\frac{1}{s(s+a)^2}$	$\frac{1}{a^2} [1 - (1+at)e^{-at}]$
12	$\frac{1}{(s+a)(s+b)}$	$\frac{1}{b-a} (e^{-at} - e^{-bt})$
13	$\frac{s}{(s+a)(s+b)}$	$\frac{1}{b-a} (be^{-bt} - ae^{-at})$
14	$\frac{1}{s(s+a)(s+b)}$	$\frac{1}{ab} \left[1 + \frac{1}{a-b} (be^{-at} - ae^{-bt}) \right]$

Tab. A-2: pokračování

	Obraz $F(s)$	Originál $f(t)$
15	$\frac{\omega}{s^2 + \omega^2}$	$\sin \omega t$
16	$\frac{s}{s^2 + \omega^2}$	$\cos \omega t$
17	$\frac{\omega}{(s + a)^2 + \omega^2}$	$e^{-at} \sin \omega t$
18	$\frac{s + a}{(s + a)^2 + \omega^2}$	$e^{-at} \cos \omega t$
19	$\frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$	$\frac{1}{\sqrt{1 - \xi^2}} e^{-\xi\omega_n t} \sin \left[(\omega_n \sqrt{1 - \xi^2}) t \right]$ (pro $\xi < 1$)
20	$\frac{s}{s^2 + 2\xi\omega_n s + \omega_n^2}$	$\frac{-1}{\sqrt{1 - \xi^2}} e^{-\xi\omega_n t} \sin \left[(\omega_n \sqrt{1 - \xi^2}) t - \psi \right]$ $\psi = \arctg \frac{\sqrt{1 - \xi^2}}{\xi}, \quad \xi < 1$
21	$\frac{\omega_n^2}{s(s^2 + 2\xi\omega_n s + \omega_n^2)}$	$1 - \frac{1}{\sqrt{1 - \xi^2}} e^{-\xi\omega_n t} \sin \left[(\omega_n \sqrt{1 - \xi^2}) t + \psi \right]$ $\psi = \arctg \frac{\sqrt{1 - \xi^2}}{\xi}, \quad \xi < 1$

PŘÍLOHA B: Seznam souborů

V tab. B-1 jsou uvedeny názvy všech vytvořených souborů (modely a skripty), se kterými se v textu pracuje. Všechny naprogramované úlohy jsou čtenáři k dispozici na přiloženém datovém nosiči a jsou také volně ke stažení na internetové adrese <http://www.spslan.cz/tech>. Programy jsou v uvedené formě okamžitě použitelné, postačuje příslušné soubory umístit do aktuálního adresáře MATLABu. Velmi snadno lze uvedené modely upravovat a následně je dle konkrétní potřeby použít ve výuce.

Tab. B-1: Soubory modelu a skriptu

Kapitola	Soubory modelu (*mdl)	Soubory skriptu (*m)
1 až 4	uvedeny přímo v textu	uvedeny přímo v textu
5.1 až 5.7	–	–
5.8	rovnomerny_pohyb nerovnomerny_pohyb	rovnomerny_pohyb_s nerovnomerny_pohyb_s
5.9	rovnomerne_zrychleny_pohyb rovnomerne_zrychleny_pohyb_2 nerovnomerne_zrychleny_pohyb	rovnomerne_zrychleny_pohyb_s rovnomerne_zrychleny_pohyb_2_s nerovnomerne_zrychleny_pohyb_s
5.10	volny_pad	volny_pad_s
5.11	vrh_svisly_vzhuru	vrh_svisly_vzhuru_s
5.12	naklonena_rovina	naklonena_rovina_s
5.13	matematicke_kyvadlo	matematicke_kyvadlo_s
5.14	kosmicka_lod	kosmicka_lod_s
5.15	vetknuty_nosnik vetknuty_nosnik_var	vetknuty_nosnik_s vetknuty_nosnik_var_s
5.16	vyska_hladiny_nadrze vyska_hladiny_nadrze_2	vyska_hladiny_nadrze_s vyska_hladiny_nadrze_2_s
5.17	spojene_nadrze spojene_nadrze_M	spojene_nadrze_s nadrze_ini_M nadrze_fun
5.18	navijeci_zarizeni	navijeci_zarizeni_s
5.19	ss_motor_CB	ss_motor_CB_s
5.20	ss_motor ss_motor_2	ss_motor_s ss_motor_2_s
5.21	logicka_funkce	–

Tab. B-1: pokračování

Kapitola	Soubory modelu (*.mdl)	Soubory skriptu (*.m)
6.1	h_prvniho_radu h_prvniho_radu_realizace	h_prvniho_radu_s
6.2	h_druheho_radu	h_druheho_radu_s h_druheho_radu_ksi
6.3	–	dynamicke_charakteristiky
6.4	–	FCH_druheho_radu_ksi FCH_druheho_radu_1 FCH_druheho_radu_2 FCH_druheho_radu_3
6.5	h_PID	h_PID_s
6.6	URO_1 URO_2 URO_3 URO_4 URO_5	URO_s
6.7	modifikovany_PID_1 modifikovany_PID_2	modifikovany_PID_s modifikovany_PID_ZN
6.8	PI_stabilita	PI_stabilita
6.9	tripolohovy_regulator nespojita_regulace	prechod nespojita_regulace_s
6.10	PSD_regulator	PSD_regulator_s
6.11	opt_PS_m	opt_PS opt_krit

Vydavatel: SOŠ a SOU, Lanškroun
Kollárova 445, 563 01 Lanškroun
tel.: 465 521 081, telefax: 465 521 083
e-mail: info@spslan.cz, www.spslan.cz

Podpora: Tento dokument byl vytvořen za finanční pomoci Evropské unie, Evropského sociálního fondu a Ministerstva školství, mládeže a tělovýchovy. Za obsah tohoto dokumentu je výhradně odpovědná SOŠ a SOU Lanškroun a nelze jej v žádném případě považovat za názor Evropské unie.

Autor: Ing. Libor Kupka
doc. Ing. Josef Janeček, CSc.

Náklad: 150 výtisků
Tisk: TG Tisk s.r.o., Lanškroun
Vyšlo: červen 2007

ISBN 978-80-239-9532-9