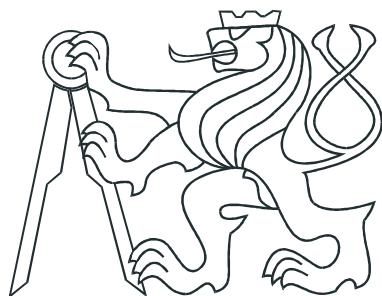


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ



BAKLÁŘSKÁ PRÁCE

Studijní materiály pro modelování systémů

Praha, 2009

Autor: Jiří Machač

Prohlášení

Prohlašuji, že jsem svou diplomovou (bakalářskou) práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne 11.6.2009



podpis

Poděkování

Děkuji především vedoucímu bakalářské práce, Jiřímu Roubalovi za pomoc při psaní této práce a také všem vyučujícím, kteří mi předali zkušenosti a vědomosti během studia. Také bych chtěl poděkovat rodině, která mi umožnila studium na Vysoké škole a vždy mě podporovala.

Abstrakt

Cílem této práce je rozšířit materiály použitelné pro výuku předmětů na Katedře řídicí techniky. Většina práce se zabývá stavovými rovnicemi. Nejen, že popisuje, jak je možné tyto rovnice řešit, ale její součástí je i program, který průběhy stavů umí vykreslovat. Dalsí část práce se popisuje vytváření virtuálního modelu. Virtuální model svým chováním odpovídá skutečnému. Student se tak může seznámit s chováním modelu ještě před tím, než přijde do laboratoře.

Abstract

The purpose of this thesis is to broaden educational materials for subjects taught on Department of Control Engineering. The major part is concerned with the state space representation. It describes how to solve the state space equations and also contains an application capable of drawing courses of states. The rest is related to description of virtual model creation. A virtual model, in the general sense, is a model of a physical system. It is a digital simplified description of the system behaving reasonably equally in the given scope. Thanks to the virtual model, students are able to familiarize with the physical system even before entering the laboratory.

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Jiří Machač**

Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný
Obor: Kybernetika a měření

Název tématu: **Studijní materiály pro modelování systémů v Matlabu**

Pokyny pro vypracování:

1. Připravte sadu řešených a neřešených příkladů, které budou součástí sbírky příkladů k výuce předmětů Systémy a modely a Systémy a řízení.
2. Naprogramujte GUI pro řešení lineárních spojitých a diskrétních dynamických modelů.
3. Namodelujte fyzikální model kuličky v obruci v Simulinku.
4. Připravte k tomuto modelu virtuální realitu.

Seznam odborné literatury:

Dorf, R. C. and Bishop, R. H. Modern Control Systems, 11. vydání, Prentice Hall, 2007, ISBN-10: 0132270285, ISBN-13: 978-0132270281.
Petr Horáček, Systémy a modely, Praha 2000
Web SARI, <http://dce.felk.cvut.cz/sari/>
Jan John, Systémy a řízení, Praha 1999
<http://www.mathworks.com/access/helpdesk/help/toolbox/vr/>

Vedoucí: Ing. Jiří Roubal, Ph.D.

Platnost zadání: do konce letního semestru 2008/09

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry



doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 27. 2. 2009

Obsah

Seznam obrázků	v
1 Úvod	1
2 Popisylienárních dynamických systémů	3
2.1 Vnější přenosový popis	3
2.2 Vnitřní stavový popis	4
2.3 Řešení stavových rovnic	4
2.3.1 Analytické řešení stavových rovnic systému	5
2.3.1.1 Módy systému	6
2.3.1.2 Řešení pomocí Laplaceovy transformace	7
2.3.2 Numerické řešení stavových rovnic	7
2.4 Příklady	8
2.5 Úlohy	18
3 Virtuální model - Kulička v obruci	27
3.1 Popis laboratorního modelu	27
3.2 Matematický popis systému	28
3.3 Identifikace laboratorního modelu	31
3.3.0.1 Otáčky obruče	31
3.3.0.2 Úhel natočení obruče	31
3.3.0.3 Poloha kuličky	32
3.3.1 Měření	32
3.3.1.1 skok + 1V	32
3.3.1.2 skok - 1V	34
3.3.1.3 trojúhelníkový vstup	35
3.3.1.4 sinusový vstup	37

3.3.1.5	obdélník vstup	38
3.3.2	Závěr	40
3.4	Virtuální realita	40
4	GUI pro stavový portrét	43
4.1	Vytvoření GUI	43
4.1.1	Nastavení GUI	44
4.1.2	Kreslení grafů	46
4.1.3	Vykreslení konkrétní počáteční podmínky	48
4.1.4	Export grafu do figure	49
5	Závěr	51
Literatura		54
A Obsah CD		I

Seznam obrázků

1.1	Laboratorní model kulička v obruči TQ	2
2.1	Blokové schéma spojitého lineárního dynamického systému	4
2.2	Odezvy systému pro různé počáteční podmínky x_0 a vstup $u(t) = 1(t)$. .	10
2.3	Simulinkové schéma k řešení příkladu 2.2	12
2.4	Odezva stavových veličin pro $\mathbf{x}(0) = [2, 1]^T$	13
2.5	Odezva systému znázorňující vliv pólu systému	14
2.6	Odezva systému znázorňující vliv nuly systému	15
2.7	Stavový (fázový) portrét systému z příkladu 2.2	16
2.8	Odezva systému pro $\mathbf{x}(0) = \mathbf{0}$ a vstup $u = 1$	17
2.9	Odezva systému pro $\mathbf{x} = \mathbf{0}$ a vstup $u = 1$ pro různé T_s	18
2.10	Model systému	22
2.11	Průběh vstupní, stavových a výstupní veličiny systému	23
2.12	Průběh stavových veličin systému	23
2.13	Model systému	24
2.14	Průběh stavových veličin systému	24
2.15	Stavové portréty	25
2.16	Řešení příkladu 2.29	26
2.17	Řešení příkladu 2.30: první a druhá část	26
3.1	Podobná dynamika kapaliny v cisternách a kuličky v obruči	27
3.2	Schématické znázornění laboratorního modelu kuličky v obruči	28
3.3	Simulinkový model kuličky v obruči	30
3.4	Statická převodní charakteristika u na ω	31
3.5	Vstupní napětí - skok	32
3.6	Otáčky obruče	33
3.7	Poloha kuličky	33
3.8	Vstupní napětí - skok	34

3.9	Otáčky obruče	34
3.10	Poloha kuličky	35
3.11	Vstupní napětí - trojúhelník	35
3.12	Otáčky obruče	36
3.13	Poloha kuličky	36
3.14	Vstupní napětí - sinus	37
3.15	Otáčky obruče	37
3.16	Poloha kuličky	38
3.17	Vstupní napětí - obdélník	38
3.18	Otáčky obruče	39
3.19	Poloha kuličky	39
3.20	Virtuální model	40
3.21	Čelní pohled ve virtuální realitě	41
4.1	Úvodní obrazovka mého GUI	44
4.2	Zadávání vstupních parametrů	46
4.3	Výsledné vykreslené průběhy	48
4.4	Byly přikresleny odezvy na dvě počáteční podmínky	50
4.5	První stav vyexportovaný do figure	50

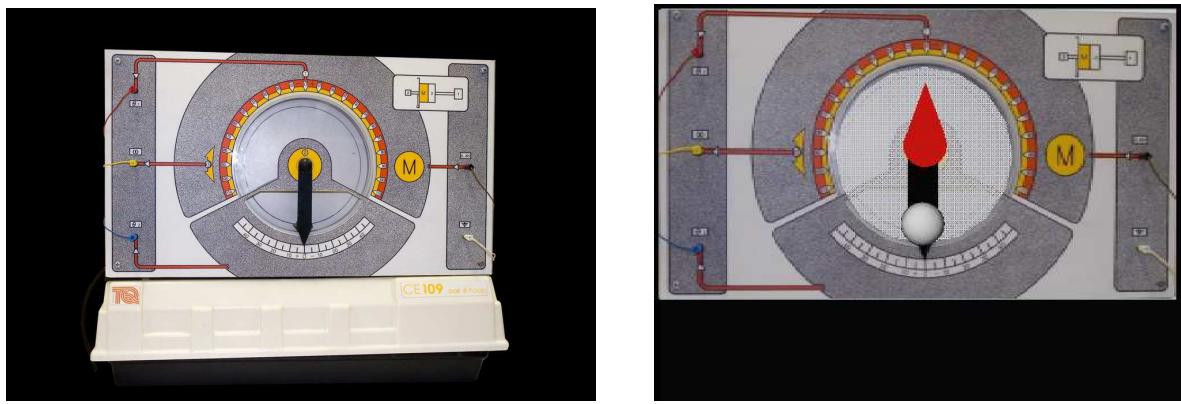
Kapitola 1

Úvod

Řídicí technika se zabývá modelováním a řízením systémů. Chceme-li nějaký systém úspěšně řídit, musíme vědět, jak takový systém funguje. K tomu slouží matematický model, jenž je sestaven podle fyzikálních zákonů, které systém popisují. V matematickém modelu se objeví řada proměnných a konstant. Některé z konstant můžeme změřit, jiné musíme určit experimentálně. V okamžiku, kdy je matematický model kompletní, můžeme vytvořit vnitřní, nebo vnější popis reálného systému. Oba popisy nám o systému prozradí mnoho informací, které jsou nezbytné pro další operace s ním. Čím kvalitnější je fyzikální model, tím více odpovídají průběhy matematického modelu skutečnému. Na základě těchto popisů můžeme např. navrhnout vhodný regulátor.

Bakalářská práce je rozdělena do tří částí. První část se zabývá vnitřním popisem systému. Je v ní ukázáno, jak je možné řešit stavové rovnice. Jednotlivé metody řešení stavových rovnic jsou ilustrované řadou řešených i neřešených příkladů. Ty se staly součástí sbírky příkladů postihující tématiku modelování a řízení dynamických systémů (ROUBAL, J. et al., 200x).

Druhá část bakalářské práce popisuje návrh a vytvoření virtuálního modelu kuličky v obruči, který se náhází v Laboratoři teorie automatického řízení 26 (ROUBAL, J. a HOLEČEK, J., 2008). Chování tohoto laboratorního modelu odpovídá chování kapaliny při přepravě v cisternách nebo tankerech. V této části práce jsou ukázány vzorce nezbytné pro vytvoření fyzikálního modelu a simulinkové schéma, které představuje matematický model systému. K tomuto systému je vytvořena virtuální realita obr. 1.1. Vlastnosti virtuálního modelu jsou velmi podobné skutečnému laboratornímu modelu a znalosti získané z řízení tohoto modelu je pak možné použít v praxi.



(a) Skutečný model

(b) Virtuální realita

Obrázek 1.1: Laboratorní model kulička v obruči TQ

Stavovým rovnicím se věnuje i třetí část bakalářské práce, ve které popisuji vytvoření programu, který umí vykreslit průběhy stavů – tzv. stavový portrét. Tetno program dále zobrazuje vývoj jednotlivých stavových veličin.

Kapitola 2

Popisy lineárních dynamických systémů

Dynamické systémy můžeme popsat několika způsoby. Dále se budu zabývat pouze lineárně časově neproměnnými dynamickými systémy (LTI z anglického Linear Time-Invariant) a na jejich dvěma matematickými způsoby popisu: vnější (přenosový) popis a vnitřní (stavový) popis. Na začátku kapitoly si připomeneme jak oba zápisu vypadají. V další části se budeme věnovat pouze vnitřnímu popisu. U jeho popisu se využívá stavových rovnic. Ukážeme si, jak se tyto rovnice teoreticky řeší. Na úplném konci pak budete tato teorie procvičena na sérii řešených i neřešených příkladů.

2.1 Vnější přenosový popis

Vnější popis systému spočívá ve vytvoření vztahu mezi vstupem \mathbf{u} a výstupem \mathbf{y} . Je určený diferenciální rovnicí

$$a_n \mathbf{y}^{(n)}(t) + \dots + a_1 \mathbf{y}(t) + a_0 = b_m \mathbf{u}^{(m)}(t) + \dots + b_1 \mathbf{u}(t) + b_0.$$

V případě použití Laplaceovy transformace můžeme vytvořit přenos

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

kde $Y(s)$ a $U(s)$ jsou Laplaceovy obrazy výstupu, resp. vstupu. Obě strany rovnice jsem převedl na Laplaceovy obrazy a vytvořil jejich poměr. (DORF, R. C. a BISHOP, R. H., 2007; FRANKLIN, G. F. et al., 2005; ŠTECHA, J.; HAVLENA, V., 1999)

2.2 Vnitřní stavový popis

Vnitřní stavový popis vyjadřuje vztah mezi vstupem \mathbf{u} a výstupem \mathbf{y} přes stav \mathbf{x} . K popisu se využívají stavové rovnice

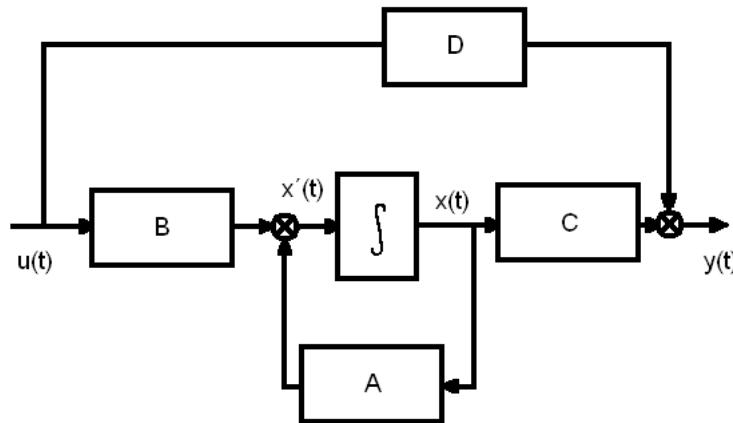
$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \\ \mathbf{y}(t) &= \mathbf{g}(\mathbf{x}, \mathbf{u}, t).\end{aligned}\tag{2.1}$$

Veličina t představuje reálný čas, \mathbf{u} vektory vstupů, \mathbf{y} vektory výstupů a \mathbf{x} vektory vnitřních stavů. (FRANKLIN, G. F. et al., 2005)

Pro lineární systémy nebo linearizované nelineární systémy můžeme stavové rovnice psát ve tvaru :

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t),\end{aligned}\tag{2.2}$$

kde \mathbf{A} je matice systému, \mathbf{B} je matice řízení, \mathbf{C} je výstupní matice a \mathbf{D} je výstupní matice. Na základě stavového popisu můžeme vytvořit blokové schémata



Obrázek 2.1: Blokové schéma spojitého lineárního dynamického systému

2.3 Řešení stavových rovnic

Nyní si ukážeme vztahy, které můžeme pro řešení stavových rovnic použít. Kromě analytických a numerických metod si ukážeme i jak můžeme využít Laplaceovu transformaci a módy systému pro vyřešení rovnic.

2.3.1 Analytické řešení stavových rovnic systému

Pro nulový vstup $u(t) = 0$ je řešení rovnice (2.2)

$$\mathbf{x}(t) = ce^{\mathbf{A}t}, \quad (2.3)$$

kde exponenciální matice $e^{\mathbf{A}t}$ je *fundamentální matici systému*.

Poznámka: V matematice se diferenciální rovnice $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$ nazývá homogení. \square

V případě, že jsou známé počáteční podmínky $x(t_0) = x_0$, řešení se změní

$$\mathbf{x}(t) = e^{\mathbf{A}(t - t_0)}\mathbf{x}_0, \quad (2.4)$$

Poznámka: V případě, že je matice \mathbf{A} diagonální, pak fundamentální matici má tvar

$$e^{\mathbf{A}t} = \begin{bmatrix} e^{\lambda_1 t} & 0 & \dots & 0 \\ 0 & e^{\lambda_2 t} & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & e^{\lambda_n t} \end{bmatrix},$$

kde $\lambda_1 \dots \lambda_n$ jsou čísla na diagonále matice \mathbf{A} . \square

Řešení úplné stavové rovnice (2.2) získáme metodou variace konstant. Předpokládané řešení je ve tvaru

$$\mathbf{x}(t) = \mathbf{c}(t)e^{\mathbf{A}(t - t_0)}, \quad (2.5)$$

kde $\mathbf{c}(t)$ je hledaný vektor, a $\mathbf{c}(t_0) = \mathbf{x}(t_0)$.

Toto řešení dosadíme do rovnice (2.2) a dostaneme

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{c}(t)e^{\mathbf{A}(t - t_0)} + \dot{\mathbf{c}}(t)e^{\mathbf{A}(t - t_0)} = \mathbf{A}\mathbf{c}(t)e^{\mathbf{A}(t - t_0)} + \mathbf{B}\mathbf{u}(t).$$

Odtud si vyjádříme $\dot{\mathbf{c}}(t)$

$$\dot{\mathbf{c}}(t) = e^{-\mathbf{A}(t - t_0)}\mathbf{B}\mathbf{u}(t)$$

a tuto rovnici zintegrujeme

$$\mathbf{c}(t) = \mathbf{c}(t_0) + \int_{t_0}^t e^{-\mathbf{A}(\tau - t_0)} \mathbf{B}\mathbf{u}(\tau) d\tau. \quad (2.6)$$

Po dosazení (2.6) do rovnice (2.5) je řešení rovnice (2.2)

$$\mathbf{x}(t) = e^{\mathbf{A}(t - t_0)} \mathbf{x}(t_0) + e^{\mathbf{A}t} \int_{t_0}^t e^{-\mathbf{A}\tau} \mathbf{B}\mathbf{u}(\tau) d\tau, \quad (2.7)$$

kde $\mathbf{x}(t_0)$ je hodnota stavové veličiny v počátečním čase t_0 a $e^{\mathbf{A}(t - t_0)}$ je tak zvaná *normovaná fundamentální matici* též nazývaná *stavová matici přechodu* lineárního stacionárního systému (ANTSAKLIS, P. J. a MICHEL, A. N., 2007; CHEN, C. T., 1998; KAILATH, T., 1980; ŠTECHA, J.; HAVLENA, V., 1999). První část rovnice (2.7) je odezva na počáteční podmítku $\mathbf{x}(t_0)$ a druhá část je odezva na vstupní signál $\mathbf{u}(t)$. Dosazením do výstupní rovnice v (2.2) můžeme vyjádřit i vývoj výstupu $\mathbf{y}(t)$

$$\mathbf{y}(t) = \mathbf{C}e^{\mathbf{A}(t - t_0)} \mathbf{x}(t_0) + \mathbf{C}e^{\mathbf{A}t} \int_{t_0}^t e^{-\mathbf{A}\tau} \mathbf{B}\mathbf{u}(\tau) d\tau + \mathbf{D}\mathbf{u}(t). \quad (2.8)$$

Poznámka: Exponenciální matici $e^{\mathbf{A}t}$ můžeme vypočítat i pomocí jejího rozvoje v nekonečnou maticovou MacLaurinovou řadu (REKTORYS, K., 1969)

$$e^{\mathbf{A}t} = \mathbf{I} + \mathbf{A}t + \frac{\mathbf{A}^2 t^2}{2!} + \frac{\mathbf{A}^3 t^3}{3!} + \frac{\mathbf{A}^4 t^4}{4!} + \dots. \quad \square$$

2.3.1.1 Módy systému

Hledejme řešení rovnice $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$ pro nulový vstup $\mathbf{u}(t) = \mathbf{0}$. Obecné řešení této rovnice můžeme zapsat ve tvaru

$$\mathbf{x}(t) = \sum_{i=1}^{n_x} k_i \boldsymbol{\nu}_i e^{\lambda_i t}, \quad (2.9)$$

kde k_i jsou reálné konstanty závislé na počátečních podmínkách a $\boldsymbol{\nu}_i$ jsou pravé vlastní vektory odpovídající vlastním číslům λ_i matice \mathbf{A} . Členy $\boldsymbol{\nu}_i e^{\lambda_i t}$ se nazývají *módy dynamického systému* (ANTSAKLIS, P. J. a MICHEL, A. N., 2007; KAILATH, T., 1980; CHEN, C. T., 1998; ŠTECHA, J.; HAVLENA, V., 1999). Mód, odpovídající pólů s největší reálnou částí, se nazývá *dominantní*.

K rovnici (2.9) se váže jeden pojem, který si zde jen slovně uvedeme a více si ho osvětlíme na příkladech v následující kapitole. Jedná se o tak zvaný *stavový portrét*, což je graf znázorňující odezvy počátečních podmínek $\mathbf{x}(t_0)$ ve stavovém prostoru (v prostoru stavů) (ANTSAKLIS, P. J. a MICHEL, A. N., 2007; KAILATH, T., 1980; CHEN, C. T., 1998; ŠTECHA, J.; HAVLENA, V., 1999).

2.3.1.2 Řešení pomocí Laplaceovy transformace

Na stavovou rovnici (2.2) aplikujeme Laplaceovu transformaci

$$sX(s) - x(t_0) = \mathbf{A}X(s) + \mathbf{B}U(s).$$

Tuto rovnici vyřešíme pro neznámou $X(s)$

$$X(s) = (s\mathbf{I} - \mathbf{A})^{-1}x(0) + (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}U(s).$$

Výsledný časový průběh stavu $\mathbf{x}(t)$ je ve tvaru

$$\mathbf{x}(t) = \mathcal{L}^{-1}\{(s\mathbf{I} - \mathbf{A})^{-1}x(0) + (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}U(s)\} \quad (2.10)$$

(ANTSAKLIS, P. J. a MICHEL, A. N., 2007; CHEN, C. T., 1998; KAILATH, T., 1980; ŠTECHA, J.; HAVLENA, V., 1999).

Poznámka: Chceme-li použít inverzní Laplaceovu transformaci na matici, použijeme ji tak, že ji aplikujeme na každý prvek matice. \square

2.3.2 Numerické řešení stavových rovnic

Pro řešení stavových rovnic můžeme také využít simulinkové schéma. Simulink řeší dynamické systémy samozřejmě *numericky*. Simulink používá propracované numerické metody, ale my si v této kapitole ukážeme jen několik jednoduchých numerických metod pro řešení stavových rovnic.

Můžeme použít jednu z následujících tří metod, kdy je derivace $\dot{\mathbf{x}}(t)$ nahrazena diferenčí (BABUŠKA, I. et al., 1966; ANTSAKLIS, P. J. a MICHEL, A. N., 2007; CHEN, C. T., 1998; KAILATH, T., 1980; ŠTECHA, J.; HAVLENA, V., 1999).

Eulerova metoda

$$\dot{\mathbf{x}}(t) \approx \frac{\mathbf{x}(t + T_s) - \mathbf{x}(t)}{T_s} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (2.11)$$

Metoda zpětné diference

$$\dot{\mathbf{x}}(t) \approx \frac{\mathbf{x}(t + T_s) - \mathbf{x}(t)}{T_s} = \mathbf{A}\mathbf{x}(t + T_s) + \mathbf{B}\mathbf{u}(t + T_s) \quad (2.12)$$

Tustinova metoda

$$\dot{\mathbf{x}}(t) \approx \frac{\mathbf{x}(t + T_s) - \mathbf{x}(t)}{T_s} = \frac{1}{2} [\mathbf{A}(\mathbf{x}(t + T_s) + \mathbf{x}(t)) + \mathbf{B}(\mathbf{u}(t + T_s) + \mathbf{u}(t))] \quad (2.13)$$

2.4 Příklady

Příklad 2.1: Nalezněte obecné řešení diferenciální rovnice popisující spojitý lineární stacionární systém

$$\dot{x}(t) = a x(t) + b u(t)$$

s počáteční podmínkou $x(t_0) = x_0$, kde a, b jsou reálné konstanty a $u(t)$ je nezávislý vstup systému. Poté uvažujte vstup $u(t) = 1(t)$ (jednotkový skok) a určete $x(t)$ pro $t \rightarrow \infty$. Diskutujte, zda je nutné znát počáteční podmínu x_0 , pokud chcete určit $x(t)$ pro $t \rightarrow \infty$.

Řešení: Budeme nejprve řešit homogenní rovnici systému $\dot{x}(t) = a x(t)$. Obecné řešení této homogenní rovnice lze zapsat ve tvaru

$$x(t) = k e^{at},$$

kde k je zatím blíže neurčená konstanta. Toto řešení musí být samozřejmě splněno i pro počáteční podmínu $x(t_0) = x_0$

$$x(t_0) = k e^{at_0}.$$

Vyjádříme-li nyní podíl $x(t)$ a $x(t_0)$, získáme řešení pro $x(t)$ ve tvaru

$$\frac{x(t)}{x(t_0)} = \frac{k e^{at}}{k e^{at_0}} = e^{a(t - t_0)}, \quad \Rightarrow \quad x(t) = x_0 e^{a(t - t_0)}. \quad (2.14)$$

Tak jsme získali řešení pro stavovou veličinu $x(t)$ funkcí počáteční podmínky x_0 pro nulový vstup $u = 0$.

Nyní budeme řešit nehomogenní diferenciální rovnici $\dot{x}(t) = a x(t) + b u(t)$. Použijeme metodu variace konstant (BRABEC, J. a HRŮZA, B., 1986) a můžeme psát

$$x(t) = k(t) e^{a(t - t_0)}, \quad \Rightarrow \quad \dot{x}(t) = \dot{k}(t) e^{a(t - t_0)} + k(t) a e^{a(t - t_0)}.$$

Dosadíme-li do původní diferenciální rovnice $\dot{x}(t) = a x(t) + b u(t)$, dostaneme

$$\dot{x}(t) = \dot{k}(t) e^{a(t-t_0)} + k(t) a e^{a(t-t_0)} = a k(t) e^{a(t-t_0)} + b u(t).$$

Odtud platí

$$\dot{k}(t) = e^{-a(t-t_0)} b u(t), \quad \Rightarrow \quad k(t) = k(t_0) + \int_{t_0}^t e^{-a(\tau-t_0)} b u(\tau) d\tau.$$

Dosadíme-li vztah pro $k(t)$ do rovnice pro $x(t)$ (2.14), získáme

$$x(t) = k(t_0) e^{a(t-t_0)} + e^{a(t-t_0)} \int_{t_0}^t e^{-a(\tau-t_0)} b u(\tau) d\tau.$$

Po dosazení počáteční podmínky $x(t_0) = x_0$ do této rovnice, dostaneme řešení původní diferenciální rovnice

$$x(t) = x_0 e^{a(t-t_0)} + e^{at} \int_{t_0}^t e^{-a\tau} b u(\tau) d\tau.$$

Dospěli jsme k řešení, které odpovídá obecnému řešení ve tvaru (2.7), který platí i pro systémy vyšších řádů. Řešení je tedy správné.

Uvažujeme-li vstup $u(t) = u$ podle zadání a počáteční podmínku x_0 v čase $t = 0$, můžeme psát

$$\begin{aligned} x(t) &= x_0 e^{at} + e^{at} \int_0^t e^{-a\tau} d\tau bu = x_0 e^{at} + e^{at} \left[-\frac{b}{a} e^{-a\tau} \right]_0^t u. \\ x(t) &= x_0 e^{at} - \frac{b}{a} (1 - e^{at}) u. \end{aligned}$$

Výše uvedená rovnice potvrzuje, že odezva lineárního stacionárního systému je daná součtem odezvy na počáteční podmínsku x_0 a odezvy na vstup u .

Nyní vyšetříme ustálenou hodnotu stavu x pro $t \rightarrow \infty$. Podíváme-li se na předchozí rovnici, zjistíme, že pro $a > 0$ budou exponenciály divergovat. Naopak pro $a < 0$ budou exponenciály pro $t \rightarrow \infty$ konvergovat k nule z čehož vyplývá, že stav x bude konvergovat k hodnotě

$$x \longrightarrow -\frac{b}{a}, \quad \text{pro } a < 0, \quad \Rightarrow \quad x \longrightarrow \frac{b}{|a|}.$$

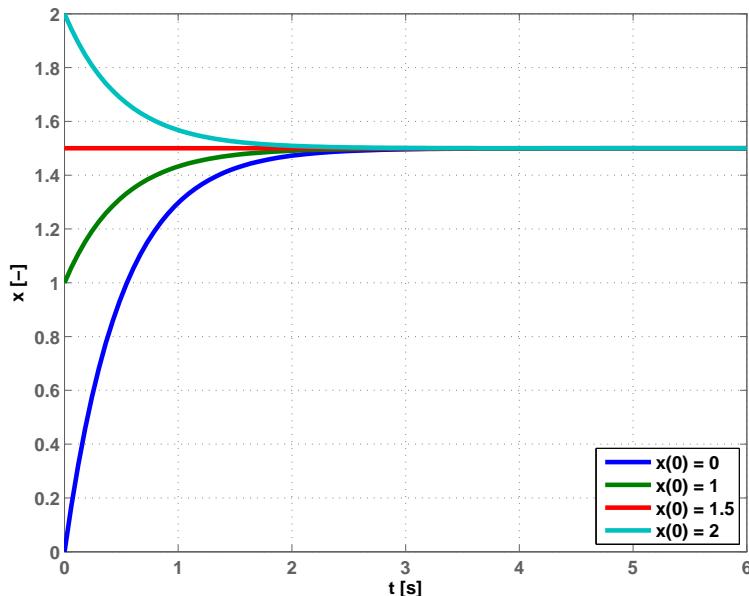
Protože se jedná o lineární systém, platí princip superpozice. Počáteční podmínka tedy oznámi a nepotřebujeme ji pro určení ustálené hodnoty stavu x znát. Jinými slovy, ustálená

hodnota stavu x na počáteční podmínce nezávisí. Rozmyslete si, jak to bude s ustálenou hodnotou stavu x v případě, když bude $a = 0$.

Budeme-li se zabývat stabilitou systémů, zjistíme, že je zadaný systém asymptoticky stabilní právě pro $a < 0$. Pak můžeme použít k určení ustálené hodnoty x úvahu, že v ustáleném stavu platí $\dot{x}(t) = 0$. Potom platí $0 = ax(t) + bu(t)$ pro $t \rightarrow \infty$. Odtud opět vidíme k jaké hodnotě stav x konverguje.

Nyní pro hodnoty parametrů $a = -2$ a $b = 3$ vykreslíme pomocí následujícího kódu odezvy systému pro různé počáteční podmínky a vstup $u(t) = 1(t)$, viz obr. 2.2.

```
t = linspace(0,6); %time axis
u = 1; %system input
%two initial conditions
x0 = [0 1 1.5 2];
Xi = [];
for i = 1 : 1 : max(size(x0))
    x = x0(i)*exp(-2*t) + 3/2*(1-exp(-2*t))*u;
    Xi = [Xi;x];
    Leg{i} = sprintf('x(0) = %1.3g',x0(i));
end
figure(1);
plot(t,Xi);
hold on;
grid on;
legend(Leg,4);
FigPlot(figure(1), 't [s]', 'x [-]', 12, 14, 3);
print(1, '-depsc2', '../Figures/ex1x0');
```



Obrázek 2.2: Odezvy systému pro různé počáteční podmínky x_0 a vstup $u(t) = 1(t)$

✓

Příklad 2.2: Nalezněte řešení stavové rovnice $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$, je-li matice \mathbf{A}

$$\mathbf{A} = \begin{bmatrix} -3 & 1 \\ 0 & -1 \end{bmatrix}.$$

Dále určete odezvu systému pro počáteční podmítku $\mathbf{x}(0) = [2, 1]^T$.

Řešení: Řešení této rovnice budeme očekávat ve tvaru (2.7) a to konkrétně

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}_0.$$

Protože systém sledujeme od času $t_0 = 0$ s, bude řešení ve tvaru $\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0$. Stavovou matici přechodu můžeme určit pomocí (2.10)

$$e^{\mathbf{A}t} = \begin{bmatrix} e^{-3t} & -\frac{1}{2}e^{-3t} + \frac{1}{2}e^{-t} \\ 0 & e^{-t} \end{bmatrix},$$

kterou dosadíme do výše uvedeného řešení a tím obdržíme řešení stavové rovnice

$$\mathbf{x}(t) = \begin{bmatrix} e^{-3t} & -\frac{1}{2}e^{-3t} + \frac{1}{2}e^{-t} \\ 0 & e^{-t} \end{bmatrix} \mathbf{x}_0.$$

Pro zadanou počáteční podmítku $\mathbf{x}(0) = [2, 1]^T$ získáme hledané řešení

$$\mathbf{x}(t) = \begin{bmatrix} e^{-3t} & -\frac{1}{2}e^{-3t} + \frac{1}{2}e^{-t} \\ 0 & e^{-t} \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{3}{2}e^{-3t} + \frac{1}{2}e^{-t} \\ e^{-t} \end{bmatrix}.$$

Zobrazit průběh stavových veličin si můžete sami. ✓

Příklad 2.3: Vyjádřete řešení příkladu 2.2 pomocí módů (2.9).

Řešení: Nejprve určíme vlastní čísla matice \mathbf{A}

$$\lambda_1 = -3, \quad \lambda_2 = -1$$

a jim odpovídající vlastní (pravé) vektory

$$\boldsymbol{\nu}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{\nu}_2 = \begin{bmatrix} 0,5 \\ 1 \end{bmatrix}.$$

Podle (2.9) je řešením

$$\mathbf{x}(t) = k_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} e^{-3t} + k_2 \begin{bmatrix} 0,5 \\ 1 \end{bmatrix} e^{-t} = \begin{bmatrix} k_1 e^{-3t} + k_2 0,5 e^{-t} \\ k_2 e^{-t} \end{bmatrix}.$$

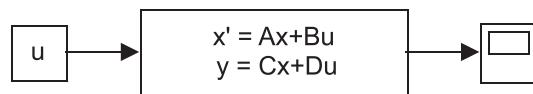
Dosazením počáteční podmínky $\mathbf{x}(0) = [2, 1]^T$ do této rovnice určíme konstanty $k_1 = \frac{3}{2}$ a $k_2 = 1$ a vidíme, že je výsledek stejný, jako v předchozím příkladu. ✓

Příklad 2.4: Nalezněte řešení stavové rovnice $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$, jsou-li matice

$$\mathbf{A} = \begin{bmatrix} -3 & 1 \\ 0 & -1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Dále určete odezvu stavů pro počáteční podmínu $\mathbf{x}(0) = [2, 1]^T$ a vstup $u(t) = 0$.

Řešení: K řešení pomocí Simulinku můžeme využít schéma na obr. 2.3. Matici \mathbf{C} zadáme jako jednotkovou, matici \mathbf{D} nastavíme nulovou tak, aby odpovídaly jejich dimenze. Na výstupu stavového bločku na obr. 2.3 tedy dostaneme průběh obou vnitřních stavů $x_1(t)$ a $x_2(t)$. Simulinkové schéma na obr. 2.3 uložíme do souboru s názvem ex4.mdl a k zobrazení použijeme přiložený zdrojový kód. Průběhy obou stavů jsou na obr. 2.4.



Obrázek 2.3: Simulinkové schéma k řešení příkladu 2.2

```

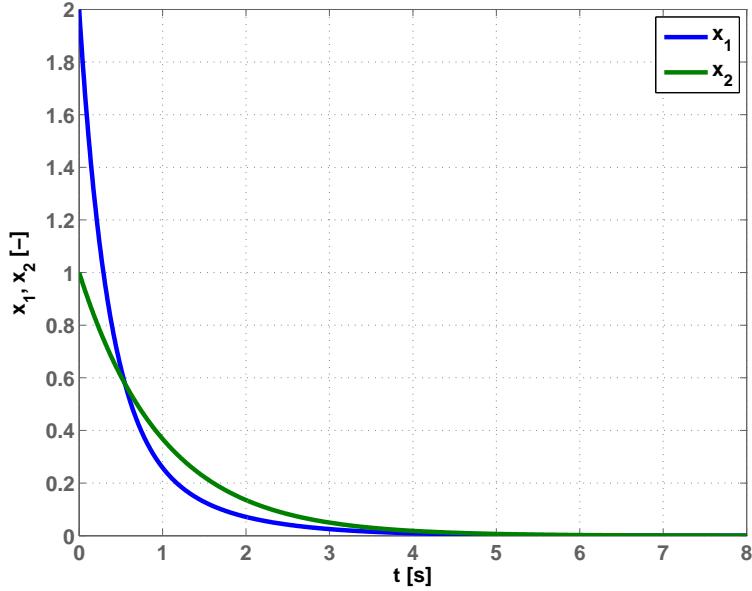
%% system matrices
A = [-3 1; 0 -1];
B = [0; 1];
C = eye(2);
D = [0;0];

%system input
u = 0;
%initial condition
x0 = [2; 1];

%% simulation in simulink
SimModel = fopen('ex4.mdl');
sim('ex4',8);
fclose(SimModel);

figure(1);
plot(ty(:,1),ty(:,2:3));
hold on;
grid on;
legend('x_1','x_2',1);

FigPlot(figure(1),'t [s]', 'x_1, x_2 [-]', 14, ' ', 14, 3);
  
```

Obrázek 2.4: Odezva stavových veličin pro $\mathbf{x}(0) = [2, 1]^T$

✓

Příklad 2.5: Ověřte, že systém s přenosem

$$G(s) = \frac{Y(s)}{U(s)} = \frac{s+5}{s+1}$$

skutečně generuje mód odpovídající jeho pólu, i když tato frekvence není obsažena ve vstupním signálu.

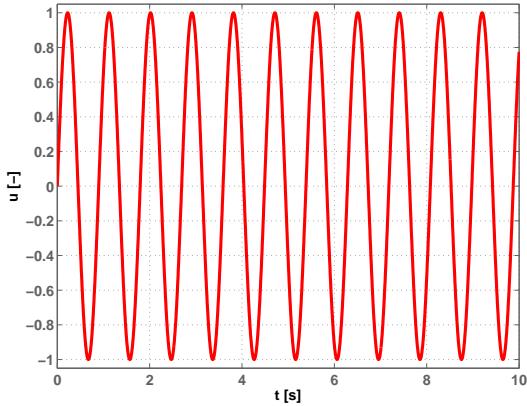
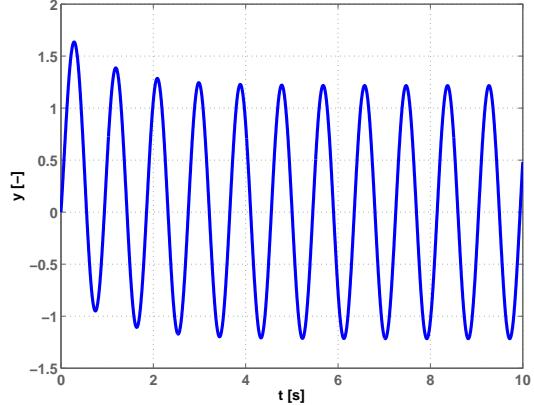
Řešení: Přivedeme na vstup systému například signál $u(t) = U_m \sin(\omega t)$. Výstup systému určíme následovně

$$y(t) = \mathcal{L}^{-1}\{G(s)U(s)\} = \mathcal{L}^{-1}\{G(s)\mathcal{L}\{u(t)\}\},$$

$$y(t) = \mathcal{L}^{-1}\left\{\frac{s+5}{s+1} \frac{U_m \omega}{s^2 + \omega^2}\right\} = U_m \mathcal{L}^{-1}\left\{\frac{\frac{4\omega}{\omega^2+1}}{s+1} - \frac{\frac{4\omega}{\omega^2+1} s}{s^2 + \omega^2} + \frac{\frac{\omega^3+5\omega}{\omega^2+1}}{s^2 + \omega^2}\right\},$$

$$y(t) = U_m \left(\frac{4\omega}{\omega^2+1} e^{-t} - \frac{4\omega}{\omega^2+1} \cos(\omega t) + \frac{\omega^2+5}{\omega^2+1} \sin(\omega t) \right).$$

Vidíme, že výstup obsahuje mód e^{-t} , aniž by tento mód byl obsažen ve vstupním signálu. Pro $U_m = 1$ a $\omega = 7 \text{ rad s}^{-1}$ je vykreslen vstupní a výstupní signál systému na obr. 2.5. Ověřte sami, že výstup obsahuje mód e^{-t} i pro jiné vstupní signály.

(a) vstup systému $u(t)$ (b) výstup systému $y(t)$

Obrázek 2.5: Odezva systému znázorňující vliv pólu systému

Tento příklad byl převzat z (Roubal, n.d.) a následně upraven. ✓

Příklad 2.6: Ovězte, že výstup systému s přenosem

$$G(s) = \frac{Y(s)}{U(s)} = \frac{s^2 + 5}{(s + 1)^2}$$

nebude obsahovat mód odpovídající nule tohoto systému. Dále nalezněte takové počáteční podmínky systému, aby byl výstup systému identicky rovný nule pro tento vstupní signál.

Řešení: Systém ze zadání má dvojici komplexně sdružených nul $z = \pm\sqrt{5}$. Přivedeme tedy na vstup systému signál $u(t) = U_m \sin(\sqrt{5}t)$ a určíme odezvu výstupu

$$y(t) = \mathcal{L}^{-1} \left\{ \frac{s^2 + 5}{(s + 1)^2} \frac{U_m \sqrt{5}}{s^2 + 5} \right\} = \mathcal{L}^{-1} \left\{ \frac{U_m \sqrt{5}}{(s + 1)^2} \right\} = U_m \sqrt{5} t e^{-t}.$$

Odtud vidíme, že výstup systému neobsahuje mód, odpovídající frekvenci vstupního signálu $\omega = \sqrt{5} \text{ rad s}^{-1}$. Odezva systému na tento vstupní signál je na obr. 2.6.

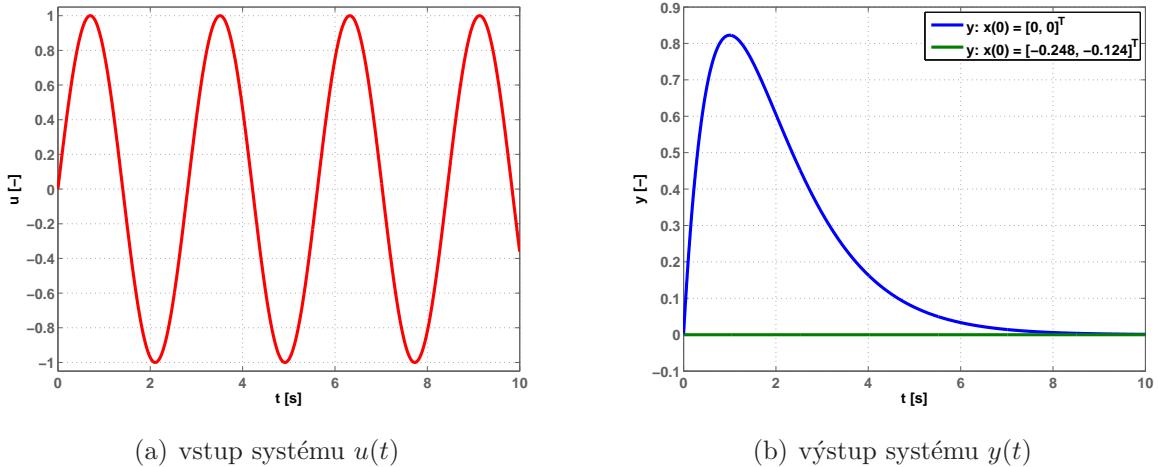
Nyní zvolíme nějakou stavovou realizaci $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ a budeme hledat takové počáteční podmínky $\mathbf{x}(0)$ v této realizaci, aby byl výstup identicky roven nule pro vstupní signál $u(t) = U_m \sin(\sqrt{5}t)$. Vyjdeme z rovnice

$$0 = Y(s) = [\mathbf{C}(\mathbf{sI} - \mathbf{A})^{-1} \mathbf{B} + \mathbf{D}] U(s) + \mathbf{C}(\mathbf{sI} - \mathbf{A})^{-1} \mathbf{x}(0) = G(s) U(s) + \frac{\mathbf{C} \operatorname{adj}(\mathbf{sI} - \mathbf{A})}{(s + 1)^2} \mathbf{x}(0).$$

Odtud určíme $\mathbf{x}(0)$

$$\frac{U_m \sqrt{5}}{(s + 1)^2} = -\frac{\mathbf{C} \operatorname{adj}(\mathbf{sI} - \mathbf{A})}{(s + 1)^2} \mathbf{x}(0), \quad \Rightarrow \quad U_m \sqrt{5} = -\mathbf{C} \operatorname{adj}(\mathbf{sI} - \mathbf{A}) \mathbf{x}(0).$$

Odtud určíme $\mathbf{x}(0)$ takové, že výstup bude identicky nulový, i když vstup nulový není, viz obr. 2.6. Zobrazte sami Bodeho frekvenční charakteristiky systému a diskutujte na nich tyto závěry.



Obrázek 2.6: Odezva systému znázorňující vliv nuly systému

Tento příklad byl převzat z (Roubal, n.d.) a následně upraven. ✓

Příklad 2.7: Načrtněte stavový (fázový) portrét systému z příkladu 2.2.

Řešení: Podívejme se nejprve blíže na rovnici $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$. Matice \mathbf{A} reprezentuje lineární operátor, který každému vektoru $\mathbf{x}(t)$ přiřazuje tečný vektor, tj. vektor, který ukazuje směr příštího vývoje stavové trajektorie.

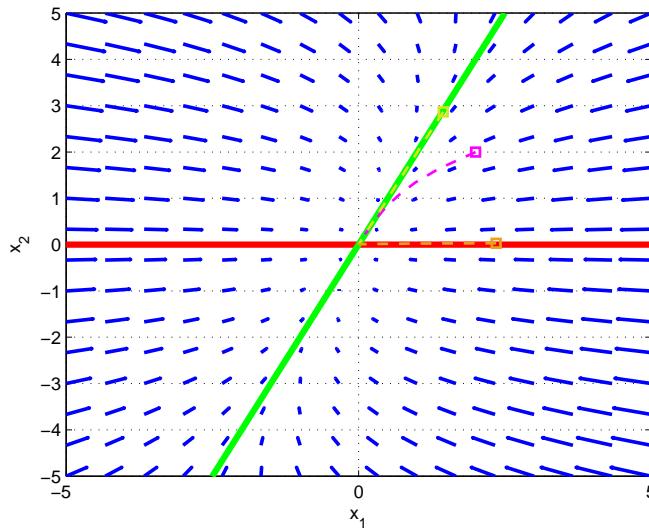
Pro první odhad průběhu stavové trajektorie využijeme vlastní čísla a vlastní vektory matice (operátoru) \mathbf{A} , které jsme určili již v příkladě 2.3

$$\lambda_1 = -3, \quad \boldsymbol{\nu}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \lambda_2 = -1, \quad \boldsymbol{\nu}_2 = \begin{bmatrix} 0,5 \\ 1 \end{bmatrix}.$$

Víme, že ve směru vlastních vektorů má operátor \mathbf{A} pouze účinek změny velikosti (směr zůstává stejný). Zvolíme-li tedy počáteční vektor $\mathbf{x}(0)$ někde ve směru vlastních vektorů, bude ve stejném směru ležet i $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$ pro všechna t . Zda se bude stav pohybovat směrem k počátku nebo od počátku souřadnic rozhodne to, zda je odpovídající vlastní číslo stabilní, či nestabilní.

Začněme tedy nakreslením invariantních podprostorů (přímek) odpovídajícím vlastním vektorům matice \mathbf{A} do obr. 2.7. Protože jsou obě vlastní čísla záporná, víme, že všechny trajektorie budou směřovat s rostoucím časem k počátku souřadnic. Bude-li tedy

počáteční podmínka ležet na některém z vlastních vektorů, bude se stav pohybovat v čase po přímce, odpovídající danému vlastnímu vektoru matice A .



Obrázek 2.7: Stavový (fázový) portrét systému z příkladu 2.2

Bude-li počáteční podmínka ležet mimo vlastní vektor matice \mathbf{A} , bude záležet na dominanci jednotlivých módů. Pro malé časy se bude nejprve projevovat rychlejší mód (s vlastním číslem $\lambda_1 = -3$, $\nu_1 = [1 \ 0]^T$). S rostoucím časem bude převažovat vliv dominantního módu (s vlastním číslem $\lambda_2 = -1$, $\nu_2 = [0,5 \ 1]^T$). Toto je patrné ze stavového portrétu na obr. 2.7.

V obr. 2.7 jsou také naznačeny dva průběhy stavových veličin pro dvě konkrétní počáteční podmínky

$$\boldsymbol{x}_{01} = \begin{bmatrix} 2 \\ -2 \end{bmatrix}, \quad \boldsymbol{x}_{02} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Počáteční podmínka \mathbf{x}_{01} leží mimo vlastní vektory matice \mathbf{A} , kdežto počáteční podmínka \mathbf{x}_{02} leží na vlastním vektoru ν_2 matice \mathbf{A} . ✓

Příklad 2.8: Vyřešte příklad 2.4 pro počáteční podmínu $\mathbf{x}(0) = [0, 0]^T$ a vstup $u = 1$ pomocí některé z numerických metod z kapitoly 2.3.2.

Rешение: Используем метод Евклида (2.11), в котором мы заменяем производную первого порядка дифференциалом.

$$\frac{\boldsymbol{x}(t+T_s) - \boldsymbol{x}(t)}{T_s} = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t).$$

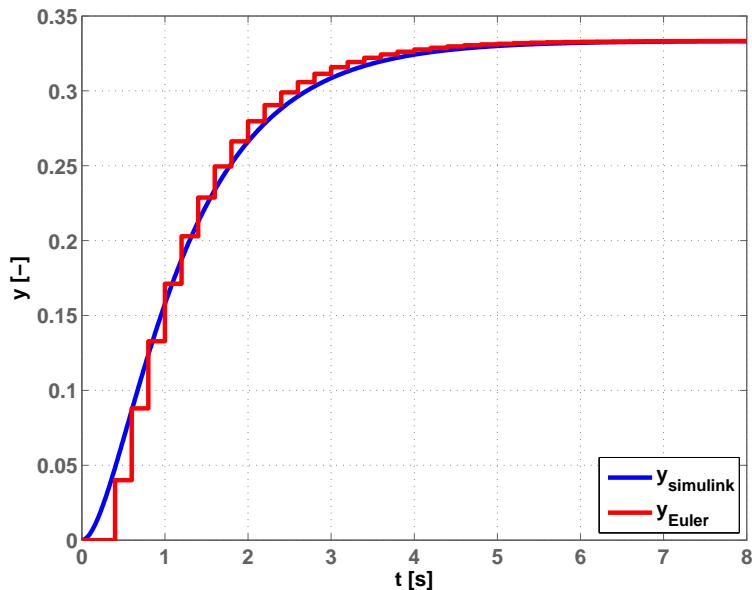
Odtud

$$\boldsymbol{x}(t + T_s) = \boldsymbol{x}(t) + T_s(\boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)).$$

Tuto rovnici stačí cyklicky řešit například pomocí následujícího kódu.

```
%% system matrices
A = [-3 1; 0 -1];
B = [0; 1];
C = [1 0];
D = [0];
Tend = 8; %length of simulation
Ts = 0.2; % sampling time
N = Tend/Ts;
x = [0;0]; %initail state
X = [] ; Y = [];
for k = 1 : 1 : N;
    y = C*x+D*u;
    X = [X; x'];
    Y = [Y; y];
    x = x + Ts*(A*x+B*u);
end
TimeC = [0 : 1 : N-1]*Ts;
figure(1);
plot(ty(:,1),ty(:,2),'b');
hold on;
stairs(TimeC,Y,'r');
grid on;
legend('y_{simulink}', 'y_{Euler}', 4);
FigPlot(figure(1), 't [s]', 'x_1, x_2 [-]', 14, ' ', 14, 3);
```

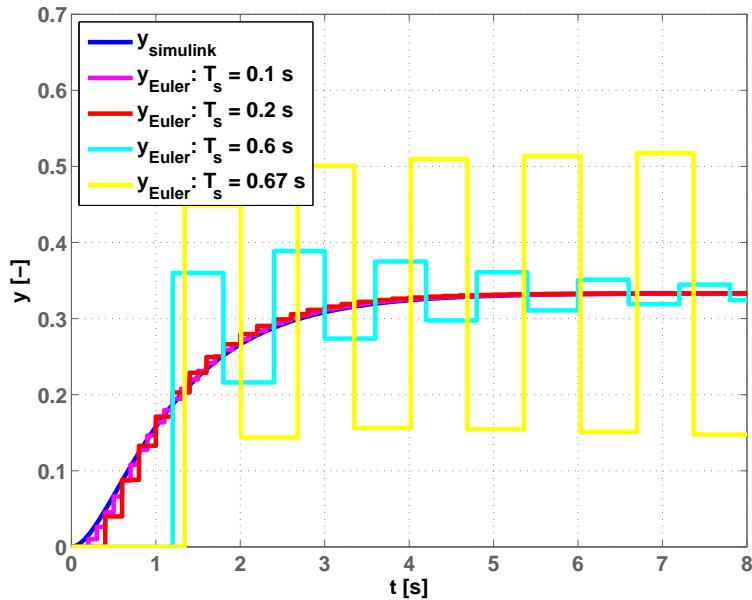
Na následujícím obrázku je porovnáno řešení pomocí simulinkového řešení z příkladu 2.4 a řešení pomocí Eulerovy metody pro $T_s = 0,2$ s.



Obrázek 2.8: Odezva systému pro $x(0) = \mathbf{0}$ a vstup $u = 1$

Příklad 2.9: Analyzujte řešení z příkladu 2.8 pro různé periody vzorkování T_s .

Řešení: Na obr. 2.9 je znázorněno řešení pro různé periody vzorkování T_s [s]. Z těch-



Obrázek 2.9: Odezva systému pro $x = 0$ a vstup $u = 1$ pro různé T_s

to grafů je patrné, že příliš velká perioda vzorkování vede k divergenci řešení. Odtud by tedy plynulo volit periodu vzorkování co nejmenší. Tak ale bude výpočet trvat příliš dlouho a budeme mít také vyšší nároky na paměť. Jak tedy zvolit periodu vzorkování pro numerickou metodu, pokud nemáme již řešení zobrazeno jako na obr. 2.8?

Můžeme postupovat například takto. Zvolíme nějakou periodu vzorkování a provedeme simulaci systému. Poté periodu vzorkování několikrát zmenšíme a novou simulaci porovnáme s předchozím řešením. Pokud se výsledné průběhy podobají, můžeme tuto periodu vzorkování považovat za správně zvolenou. Samozřejmě můžeme využít znalosti o souvislosti pólů systému a rychlosti a stability časové odezvy systému a podle toho vhodně zvolit periodu vzorkování. ✓

2.5 Úlohy

Příklad 2.10: Určete analytické řešení spojitého lineárního stacionárního systému

$$\dot{x}(t) = a x(t) + b u(t)$$

s počáteční podmínkou $x(t_0) = x_0$, kde a, b jsou reálné konstanty a $u(t)$ je nezávislý vstup systému. Uvažujte vstupní signál

$$u(t) = \begin{cases} 0, & t < 5 \text{ s}, \\ 1, & t \in (5, 10) \text{ s}, \\ 5, & t \geq 10 \text{ s} \end{cases}$$

a vypočítejte ustálenou hodnotu stavu $x(t)$ pro $t \rightarrow \infty$ (pokud vůbec existuje).

Dále zvolte konstanty a, b a vykreslete průběh stavu $x(t)$ pro dvě různé počáteční podmínky $x(0)$ a pro výše uvedený vstupní signál $u(t)$. Porovnejte výsledky simulace s analytickým řešením. Dále určete přenos systému a určete jeho statické zesílení. Diskutujte jeho souvislost s ustálenou hodnotou stavu x .

Příklad 2.11: Řešte příklad 2.10 s uvažováním vstupního signálu definovaným jako

$$u(t) = \begin{cases} 1, & t \in (0, 1) \text{ s}, \\ 0, & \text{jinde.} \end{cases}$$

Příklad 2.12: U systému z příkladu 2.10 zvolte dvě kombinace konstant $[a, b]$ a určete odezvu systému pro dvě různé počáteční podmínky x_0 a pro vstupní signál typu jednotkový skok $u(t) = 1(t)$.

Příklad 2.13: Určete odezvu systému z příkladu 2.10 pomocí všech tří numerických metod z kapitoly 2.3.2 pro nějakou kombinaci konstant $[a, b]$ a porovnejte toto řešení s řešením z příkladu 2.10. Diskutujte vlastnosti jednotlivých numerických metod.

Příklad 2.14: Uvažujte systém popsaný přenosem

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1}{s^2 + 3s + 2}.$$

Nalezněte stavový popis tohoto systému a určete jeho módy. Určete, který mód je dominantní. Nalezněte analytický vztah pro $y(t)$ pomocí řešení stavových rovnic. Vykreslete odezvu tohoto systému na jednotkový skok a porovnejte ji s analytickým řešením.

Příklad 2.15: Uvažujte stavový popis lineárního systému s maticemi

$$\mathbf{A} = \begin{bmatrix} -3 & -2 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad \mathbf{D} = [0].$$

Určete módy tohoto systému. Určete, který mód je dominantní. Nalezněte analytický vztah pro $y(t)$ pomocí řešení stavových rovnic. Vykreslete odezvu tohoto systému na jednotkový skok a porovnejte ji s analytickým řešením.

Příklad 2.16: Určete odezvu systému z příkladu 2.15 pomocí všech tří numerických metod z kapitoly 2.3.2 a porovnejte řešení s řešením z příkladu 2.15. Diskutujte vlastnosti jednotlivých numerických metod.

Příklad 2.17: Určete takový vstupní signál $u(t)$ (případně i počáteční podmínu), který systém s přenosem

$$G(s) = \frac{Y(s)}{U(s)} = \frac{s+2}{s+0,1}$$

vůbec nepropouští. Dále určete, který typ signálu se objeví vždy na výstupu tohoto systému nezávisle na vstupu. Určete výstup systému, pokud na vstup systému přivedeme signál odpovídající jeho pólů.

Příklad 2.18: Napište přenos nebo stavový popis spojitého lineárního systému, který splňuje všechny následující podmínky:

- všechny póly mají nekladnou reálnou část,
- systém generuje na výstupu signál $\sin(t)$, nezávisle na vstupu,
- systém vůbec nepropouští signál $u(t) = e^{2t}$.

Příklad 2.19: Napište přenos nebo stavový popis spojitého lineárního systému, který splňuje všechny následující podmínky:

- všechny póly mají zápornou reálnou část,
- systém generuje na výstupu signál $e^{-2t}(\cos(10t) + j\sin(10t))$ nezávisle na vstupu,
- systém vůbec nepropouští signál $u(t) = \sin(5t)$,
- statické zesílení systému je $k = 10$.

Příklad 2.20: Uvažujte lineární systém s jedním vstupem u a jedním výstupem y popsaný stavovými maticemi

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \mathbf{D} = [0].$$

Uvažujte počáteční podmínu systému $\mathbf{x}(0) = [0, 1]^T$ a průběh vstupu $u(t) = 1(t - 10)$ (posunutý jednotkový skok do 10 s). Dále víte, že stav v čase $t = 9$ s je $\mathbf{x}(9) \doteq [1, 0]^T$. Určete hodnotu výstupu $y(t)$ pro $t = 9$ s.

Příklad 2.21: Uvažujte lineární systém s jedním vstupem u a jedním výstupem y popsaný stavovými maticemi

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \mathbf{D} = [0].$$

Uvažujte počáteční podmínu systému $\mathbf{x}(0) = [10, 1]^T$ a průběh vstupu $u(t) = 4 \cdot 1(t-1)$ (posunutý jednotkový skok do 1 s). Určete hodnotu výstupu $y(t)$ pro $t \rightarrow \infty$.

Příklad 2.22: Uvažujte lineární systém jehož řešení má tvar

$$\mathbf{x}(t) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} e^{-t} + \begin{bmatrix} -1 \\ 0,2 \end{bmatrix} e^{-10t}$$

Určete počáteční podmínu $\mathbf{x}(0)$ a vykreslete průběh tohoto řešení. Dále určete matici \mathbf{A} tohoto systému a ověřte pomocí simulace, že jste ji nalezli správně.

Příklad 2.23: Určete módy autonomního systému popsaného stavovou rovnicí

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -10 & 0 \\ 0 & -1 \end{bmatrix} \mathbf{x}(t)$$

a načrtněte jeho stavový (fázový) portrét. Diskutujte, jak se jednotlivé módy projevují v čase.

Příklad 2.24: Určete módy autonomního systému popsaného stavovou rovnicí

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -10 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}(t)$$

a načrtněte jeho stavový (fázový) portrét. Diskutujte, jak se jednotlivé módy projevují v čase.

Příklad 2.25: Určete módy autonomního systému popsaného stavovou rovnicí

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -5 & 1 \\ 0 & -2 \end{bmatrix} \mathbf{x}(t)$$

a načrtněte jeho stavový (fázový) portrét. Diskutujte, jak se jednotlivé módy projevují v čase.

Příklad 2.26: Určete módy autonomního systému popsaného stavovou rovnicí

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -5 & 1 \\ 0 & +2 \end{bmatrix} \mathbf{x}(t)$$

a načrtněte jeho stavový (fázový) portrét. Diskutujte, jak se jednotlivé módy projevují v čase.

Příklad 2.27: Určete módy autonomního systému popsaného stavovou rovnicí

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -3 & -2 \\ 2 & -2 \end{bmatrix} \mathbf{x}(t)$$

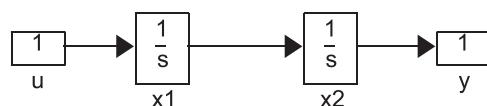
a načrtněte jeho stavový (fázový) portrét. Diskutujte, jak se jednotlivé módy projevují v čase.

Příklad 2.28: Nakreslete stavový (fázový) portrét nelineárního systému (popisujícího vývoj populace dravců x_1 a obětí x_2 v uzavřeném prostoru (VOLTERRA, V., 1931))

$$\begin{aligned}\dot{x}_1(t) &= -x_1(t) + 0,01 x_1(t) x_2(t), \\ \dot{x}_2(t) &= x_2(t) - 0,01 x_1(t) x_2(t).\end{aligned}$$

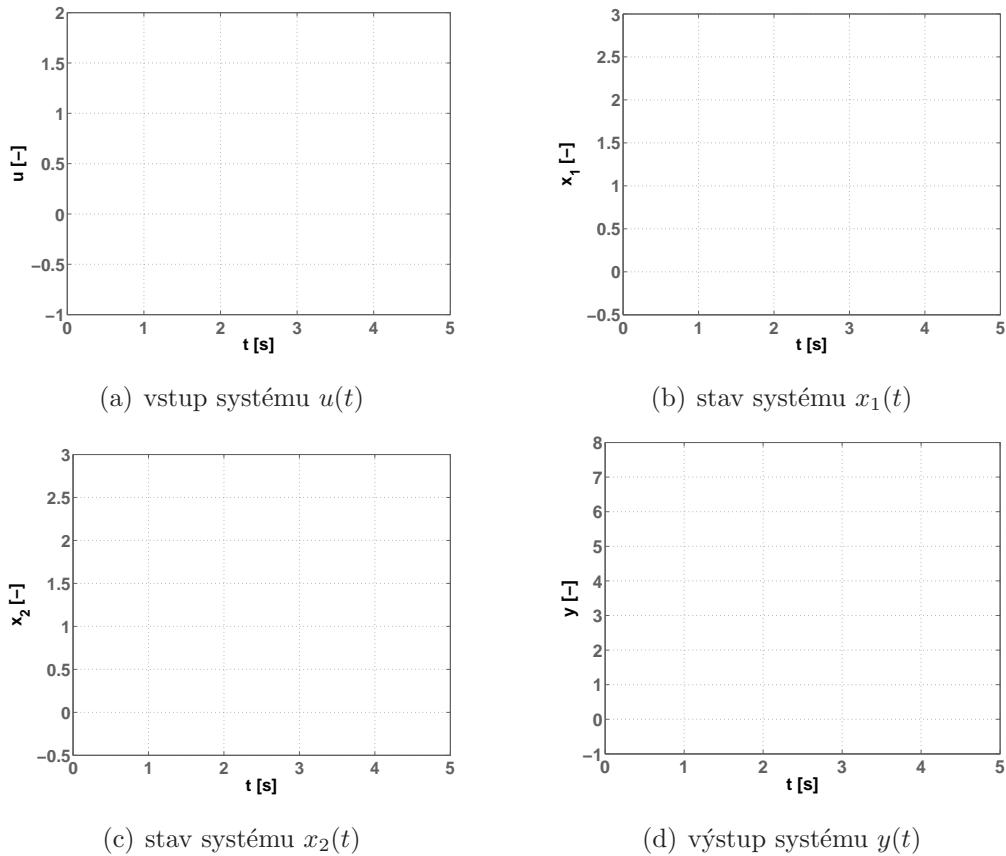
Řešte rovnice numericky. Poté nalezněte rovnovážné body tohoto systému a proved'te v nich linearizaci. Porovnejte stavové portréty nelineárního a linearizovaného systému.

Příklad 2.29: Uvažujte systém s jedním vstupem u a jedním výstupem y podle obr. 2.10.



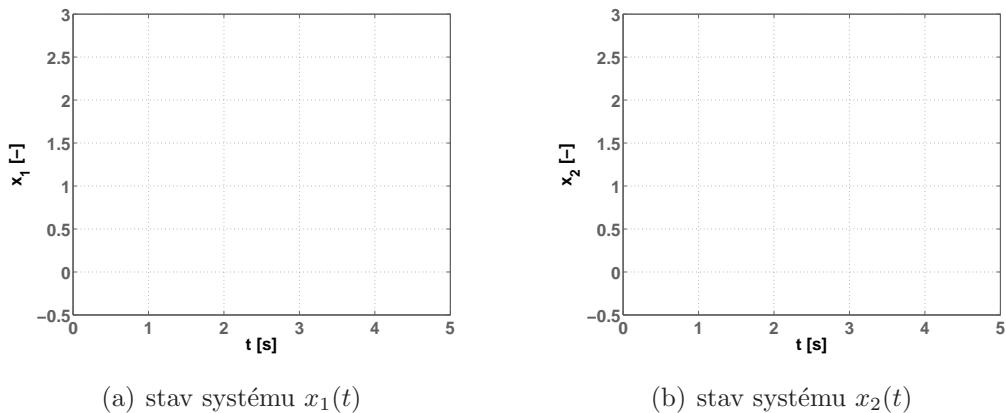
Obrázek 2.10: Model systému

Pro nulové počáteční podmínky načrtněte průběh jednotlivých stavových veličin a výstupu systému pro vstupní jednotkový skok $u(t) = 1(t - 1)$. Poté určete stavový model tohoto systému a odezvy určete výpočtem.



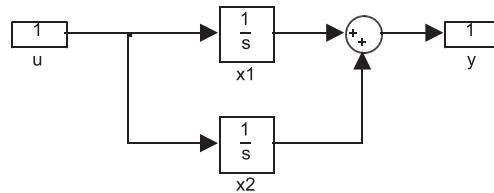
Obrázek 2.11: Průběh vstupní, stavových a výstupní veličiny systému

Příklad 2.30: Uvažujte systém s jedním vstupem u a jedním výstupem y podle obr. 2.10. Pro nulový vstup načrtněte průběh jednotlivých stavových veličin systému pro počáteční podmínky: a) $\mathbf{x}_0 = [0, 1]^T$, b) $\mathbf{x}_0 = [1, 0]^T$. Poté určete stavový model tohoto systému a odezvy určete výpočtem.



Obrázek 2.12: Průběh stavových veličin systému

Příklad 2.31: Uvažujte systém s jedním vstupem u a jedním výstupem y podle obr. 2.13.



Obrázek 2.13: Model systému

Načrtněte do obr. 2.14 průběh jednotlivých stavových veličin systému pro vstupní jednotkový skok $u(t) = 1(t - 1)$ a pro různé počáteční podmínky

- $\mathbf{x}_0 = [0, 0]^T$,

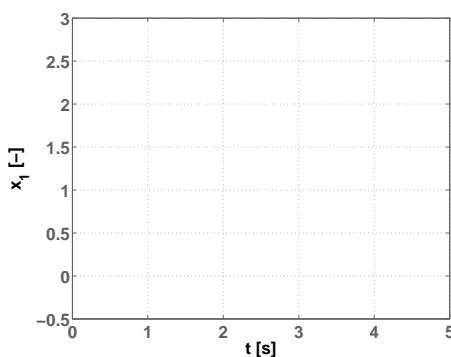
- $\mathbf{x}_0 = [0, 1]^T$,

- $\mathbf{x}_0 = [0, 2]^T$,

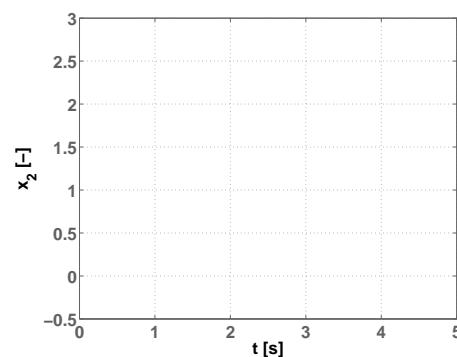
- $\mathbf{x}_0 = [1, 0]^T$,

- $\mathbf{x}_0 = [2, 0]^T$.

Co je na těchto průbězích zajímavého? Poté určete stavový model tohoto systému a odezvy určete výpočtem.



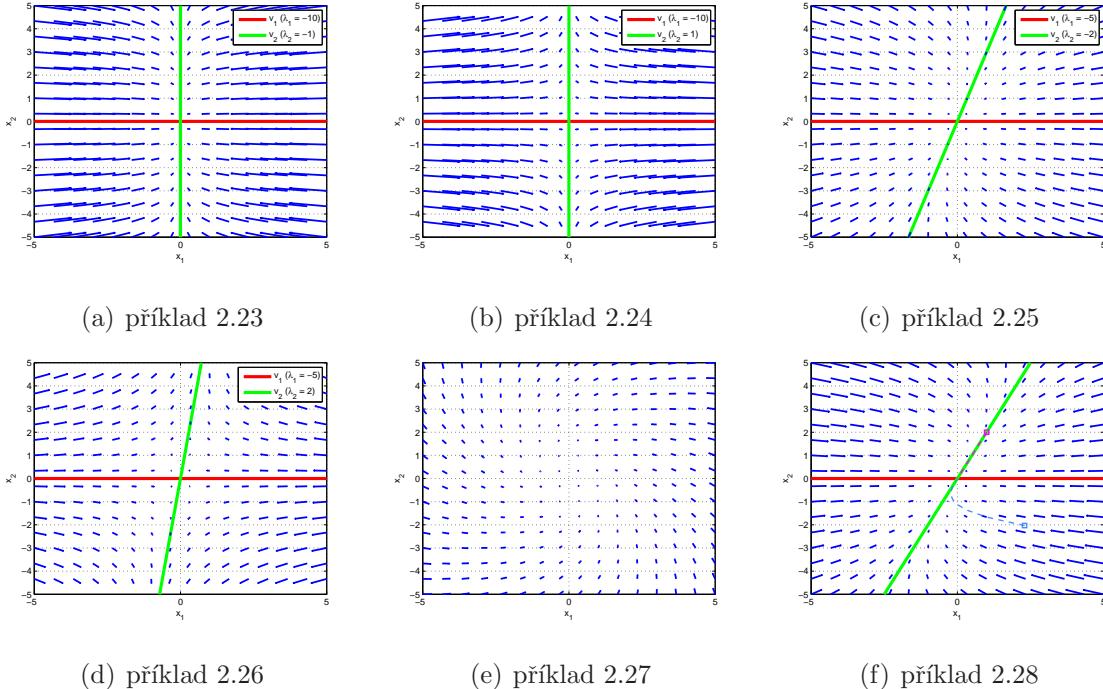
(a) stav systému $x_1(t)$



(b) stav systému $x_2(t)$

Obrázek 2.14: Průběh stavových veličin systému

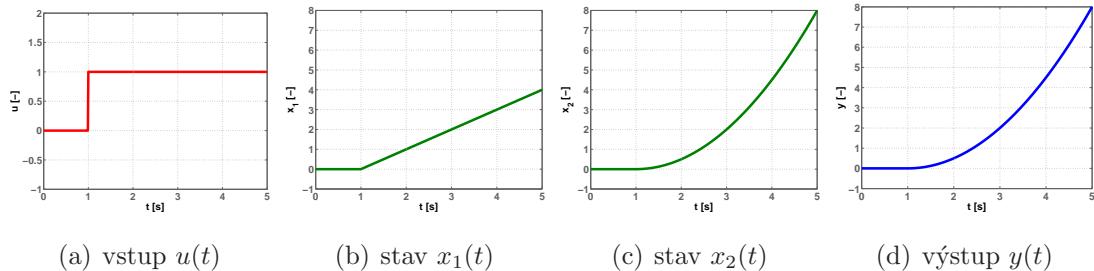
Řešení úloh z kapitoly 2.3 Řešení stavových rovnic systému



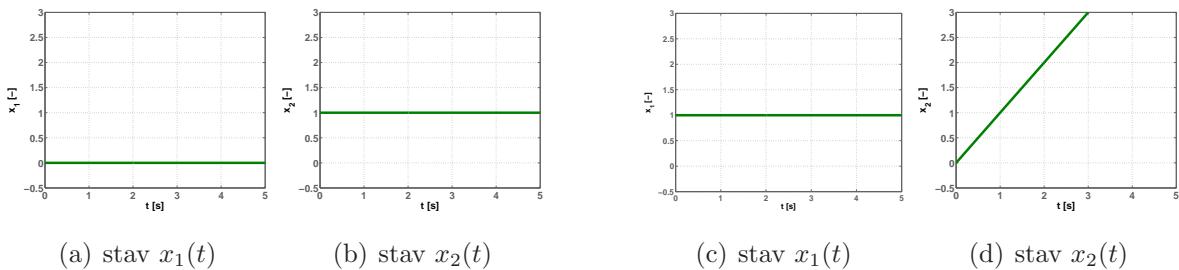
Obrázek 2.15: Stavové portréty

2.10: Analytické řešení je podle (2.7). Je-li systém stabilní ($a < 0$), ustálená hodnota bude $-\frac{b}{a} u$ (na počáteční podmínce nezáleží). Pakliže je systém nestabilní limita v nekonečnu je nekonečno. Přenos je $G(s) = \frac{b}{s-a}$. Statické zesílení je pro stabilní systém $k = \frac{-b}{a}$.; 2.11: Pro stabilní systém ($a < 0$) je ustálená hodnota 0. Pro nestabilní systém každá nenulová počáteční podmínka způsobí divergenci.; 2.12: podobně jako v příkladě 2.1; 2.13: podobně jako v příkladě 2.8; 2.14: stavové matice jsou například $\mathbf{A} = \begin{bmatrix} -3 & -2 \\ 1 & 0 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\mathbf{C} = [0 \ 1]$, $\mathbf{D} = [0]$, póly systému jsou $\lambda_1 = -2$, $\lambda_2 = -1$, dominantní mód je mód s vlastním číslem λ_2 , analytické řešení je $y(t) = 0,5e^{-2t} - e^{-t} + 0,5$; 2.15: stavový popis odpovídá přenosu z příkladu 2.15; 2.16: podobně jako v příkladě 2.8; 2.17: Signál, který systém vůbec nepropustí, je $u(t) = e^{-2t}$ (nula systému). Mód, který se vždy objeví na výstupu, je $e^{-0,1t}$. Výstup pro vstup odpovídající jeho módu je $y(t) = \mathcal{L}^{-1}\left\{\frac{s+2}{(s+0,1)^2}\right\}$. Průběhy si vykreslete sami.; 2.18: $G(s) = \frac{s-2}{s^2+1}$; 2.19: $G(s) = \frac{41,6(s^2+25)}{s^2+4s+104}$; 2.20: $y(9) = \mathbf{C}\mathbf{x}(9) = 1$, ve stavu je veškerá potřebná informace (nepotřebujeme minulá data); 2.21: systém je stabilní, ustálená hodnota výstupu podle řešení příkladu 2.10 tedy nezávisí na počáteční podmínce, $y = (\mathbf{C}\mathbf{A}^{-1}\mathbf{B}+\mathbf{D})4 = 2$; 2.22: $\mathbf{x}(0) = [0 \ 0,2]^T$, matice

$\mathbf{A} = \begin{bmatrix} -1 & 45 \\ 0 & -10 \end{bmatrix}$; 2.23: $\lambda_1 = -10$, $\lambda_2 = -1$, $\mathbf{v}_1 = [1 \ 0]^T$, $\mathbf{v}_2 = [0 \ 1]^T$, viz obr. 2.15(a); 2.24: $\lambda_1 = -10$, $\lambda_2 = 1$, $\mathbf{v}_1 = [1 \ 0]^T$, $\mathbf{v}_2 = [0 \ 1]^T$, viz obr. 2.15(b); 2.25: $\lambda_1 = -5$, $\lambda_2 = -2$, $\mathbf{v}_1 = [1 \ 0]^T$, $\mathbf{v}_2 = [1 \ 3]^T$, viz obr. 2.15(c); 2.26: $\lambda_1 = -5$, $\lambda_2 = 2$, $\mathbf{v}_1 = [1 \ 0]^T$, $\mathbf{v}_2 = [1 \ 7]^T$, viz obr. 2.15(d); 2.27: $\lambda_{1,2} = -2,5 \pm 1,94i$, vlastní vektory jsou komplexní, viz obr. 2.15(e); 2.28: viz obr. 2.15(f), systém je nelineární, takovýto stavový portrét nemůže lineární systém vygenerovat; 2.29: viz obr. 2.16; 2.30: viz obr. 2.17; 2.31: podobně jako v příkladě 2.30, systém je neřiditelný;



Obrázek 2.16: Řešení příkladu 2.29

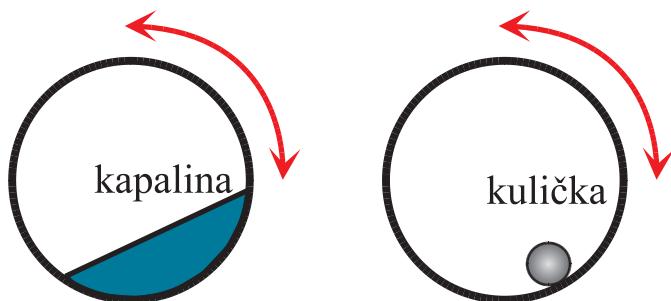


Obrázek 2.17: Řešení příkladu 2.30: první a druhá část

Kapitola 3

Virtuální model - Kulička v obruci

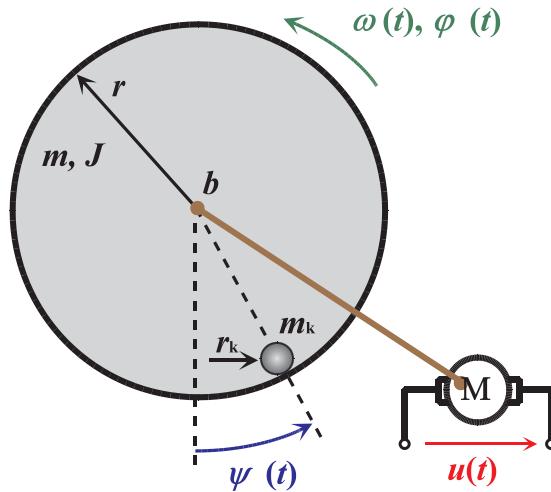
Model kuličky v obruci byl vyroben společností TecQuipment Ltd. Slouží k laboratorní výuce modelování dynamických systémů a jejich řízení. Pohyb kuličky v obruci odpovídá pohybu kapaliny při přepravě v cisternách nebo tankerech. Pohyb kapaliny ovlivní vlastnosti dopravního prostředku, který by měl být schopen vydržet váhu nákladu a také odolávat dynamickým silám, vytvářeným kapalinou při přepravě. Podobnost mezi pohybem kapaliny a kuličky ilustruje obr. 3.1.



Obrázek 3.1: Podobná dynamika kapaliny v cisternách a kuličky v obruci

3.1 Popis laboratorního modelu

Model je sestaven ze servomotoru, ke kterému je pevně připojena obruč, která vytváří setrvačnou zátěž motoru. V drážce obruče je umístěna kovová kulička, která se v ní může pohybovat. Vstupem do systému je napětí na servomotoru $u[V]$. Na výstupu měříme pomocí integrálního tachometru rychlosť otáčení obruče $\omega[\text{rad/s}]$, úhel natočení obruče $\varphi[\text{rad}]$ a potenciometrem polohu kuličky $\psi[\text{rad}]$. Schéma modelu je naznačeno na následujícím obrázku.



Obrázek 3.2: Schématické znázornění laboratorního modelu kuličky v obruči

Všechny měřené veličiny jsou převedeny na bezrozměrná čísla. Komunikace mezi PC a modelem je zajišťována pomocí zásuvné karty a programu MATLAB/SIMULINK. Více informací o modelu můžete získat na webových stránkách Laboratoře teorie automatického řízení (ROUBAL, J. a HOLEČEK, J., 2008).

3.2 Matematický popis systému

Pro popis úhlu natočení obruče a otáček obruče použiji vztahy pro popis servomotoru, který obruč roztáčí

$$L \frac{di(t)}{dt} = -Ri(t) - k_e \omega(t) + u(t), \quad (3.1)$$

$$J \frac{d\omega(t)}{dt} = k_m i(t) - b\omega(t), \quad (3.2)$$

$$\frac{d\varphi(t)}{dt} = \omega(t), \quad (3.3)$$

kde i [A] je proud motoru, ω [s^{-1}] jsou otáčky motoru, φ [rad] je úhel natočení hřídele motoru, u [V] je vstupní napětí motoru, $R[\Omega]$ je elektrický odpor motoru, L [H] je indukčnost motoru, J [$kg\ m^2\ s^{-1}$] je moment setrvačnosti motoru, b [$kg\ m^2\ s^{-1}$] je konstanta tření motoru, k_e [$s\ V^{-1}$] je elektrická konstanta motoru a k_m [$kg\ m^2\ s^{-2}$] je mechanická konstanta motoru.

Z těchto rovnic můžeme určit stavový popis a přenosy

$$\begin{bmatrix} \dot{i}(t) \\ \dot{\omega}(t) \\ \dot{\varphi}(t) \end{bmatrix} = \begin{bmatrix} -R & -k_e & 0 \\ \frac{L}{k_m} & \frac{-b}{L} & 0 \\ \frac{J}{L} & 0 & 1 \end{bmatrix} \begin{bmatrix} i(t) \\ \omega(t) \\ \varphi(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \end{bmatrix} u(t),$$

$$y(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i(t) \\ \omega(t) \\ \varphi(t) \end{bmatrix},$$

$$G_1(s) = \frac{\Omega(s)}{U(s)} = \frac{k_m}{LJs^2 + (Lb + RJ)s + Rb + k_e k_m}. \quad (3.4)$$

Z rovnice (3.3) je vidět, že pro získání úhlu natočení obruče stačí rovnici (3.4) zintegrovat

$$G_2(s) = \frac{\varphi(s)}{U(s)} = \frac{1}{s} \frac{k_m}{LJs^2 + (Lb + RJ)s + Rb + k_e k_m}. \quad (3.5)$$

Pro určení pohybu kuličky si musíme vyjádřit energie kuličky. Nejprve si vyjádřím kynetickou energii kuličky

$$W_{kk} = \frac{1}{2} J_k \omega_k^2(t) + \frac{1}{2} m_k v_k^2(t), \quad (3.6)$$

kde J_k [kg m²] je moment setrvačnosti kuličky, pro kouli platí $J_k = \frac{2}{5} mr_k^2$, ω_k [rad s⁻¹] je úhlová rychlosť kuličky, m_k [kg] je hmotnosť kuličky a v_k [m s⁻¹] je rychlosť kuličky. Potenciální energie je

$$W_{pk} = W_{pk0} + mg[r - (r - r_k) \cos(\psi(t))], \quad (3.7)$$

kde r [m] je poloměr obruče, r_k [m] je poloměr kuličky a $\psi(t)$ [rad] je poloha kuličky v obruči. Rychlosť v_k si mohu vyjádřit pomocí souřadnic

$$x(t) = (r - r_k) \sin(\psi(t)),$$

$$y = r - (r - r_k) \cos(\psi(t)).$$

Odtud

$$v_k^2(t) = v_{kx}^2(t) + v_{ky}^2(t),$$

$$v_{kx}(t) = \frac{dx(t)}{dt} = (r - r_k) \cos(\psi(t)) \frac{d\psi(t)}{dt},$$

$$v_{ky}(t) = \frac{dy(t)}{dt} = (r - r_k) \sin(\psi(t)) \frac{d\psi(t)}{dt}.$$

Nyní si mohu upravit vyjádření kinetické energie

$$W_{kk} = \frac{1}{2} \frac{2}{5} m_k r_k^2 \frac{1}{r_k^2} v_k^2(t) + \frac{1}{2} m_k v_k^2(t),$$

$$W_{kk} = \frac{7}{10} m_k (r - r_k)^2 \left(\frac{d\psi(t)}{dt} \right)^2. \quad (3.8)$$

Pro výsledný popis polohy kuličky použiji Lagrangeovy rovnice. Lagangiánem bude rozdíl kynetické a potenciální energie

$$L = \frac{7}{10} m_k (r - r_k)^2 (\dot{\psi}(t))^2 - W_{pk0} - m_k g [r - (r - r_k) \cos(\psi(t))] \quad (3.9)$$

Samotná Lagrangeova rovnice bude

$$\frac{d}{dt} \frac{\delta L}{\delta \dot{\psi}} - \frac{\delta L}{\delta \psi} = - \frac{dF_t}{d\psi} + J \frac{d\omega(t)}{dt}, \quad (3.10)$$

(Wikipedia – Open encyclopedia [online], 2008) kde F_t [N] je třetí síla

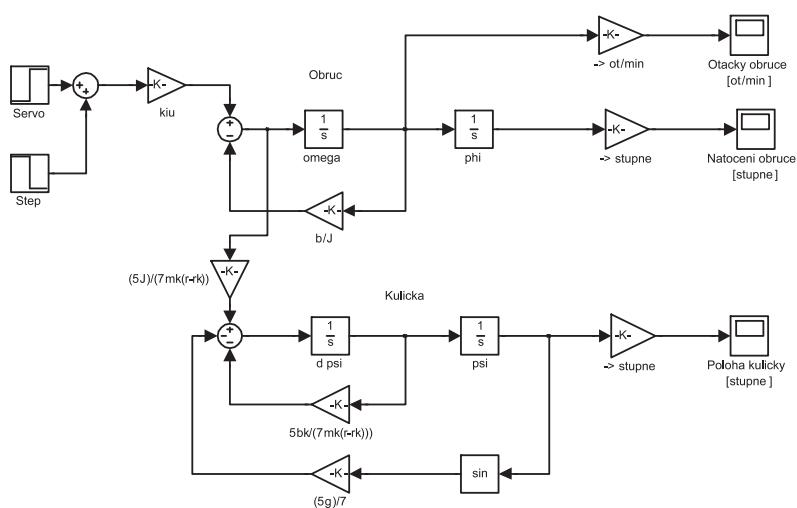
$$F_t = \frac{1}{2} b_k (\dot{\psi}(t))^2, \quad (3.11)$$

kde b_k je koeficient tření kuličky.

Když dosadím rovnice (3.9) a (3.11) do rovnice (3.10) a provedu derivování, dostanu rovnici, která popisuje pohyb kuličky

$$\begin{aligned} \frac{7}{5} m_k (r - r_k)^2 \ddot{\psi}(t) + m_k g (r - r_k) \sin(\psi(t)) &= -b_k \dot{\psi}(t) + J \frac{d\omega(t)}{dt} \\ \frac{7}{5} m_k (r - r_k)^2 \ddot{\psi}(t) + b_k \dot{\psi}(t) + m_k g (r - r_k) \sin(\psi(t)) &= k_m i(t) - b\omega(t) \end{aligned} \quad (3.12)$$

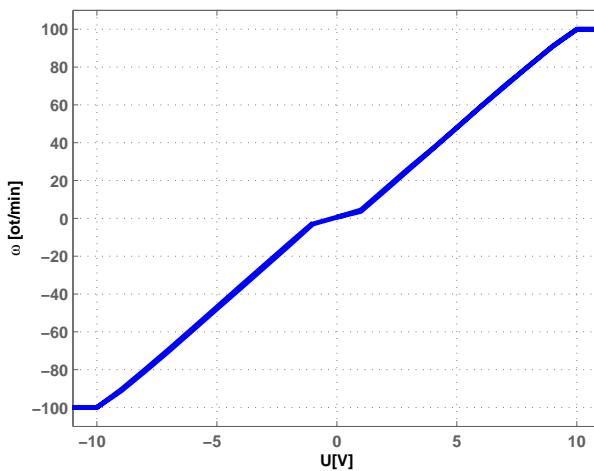
Simulinkový model systému kulička v obruci je na následujícím obrázku.



Obrázek 3.3: Simulinkový model kuličky v obruci

3.3 Identifikace laboratorního modelu

Před samotnou identifikací systému bylo potřeba převést výstupní veličiny z bezrozměrných čísel na fyzikální jednotky. Abych na výstupu měl otáčky v ot/min , bylo třeba přenásobit výstup hodnotou 900. Pro úhel natočení obruče je převodní konstanta na stupně 140 u polohy kuličky 40.5. Abych viděl, jak systém reaguje, určil jsem si před identifikací statickou charakteristiku mezi vstupním napětím u a otáčkami obruče ω .



Obrázek 3.4: Statická převodní charakteristika u na ω

Dynamika motoru budu identifikovat v pracovním bodě $u_0 = 2\text{V}$. Sestavil jsem si simulinkové schéma podle fyzikálních rovnic (3.2), (3.3) a (3.12).

3.3.0.1 Otáčky obruče

I když podle fyzikálních rovnic měl být přenos druhého řádu, experimentálně jsem zjistil, že jej lze approximovat přenosem prvního řádu. Pomocí přechodové charakteristiky se mi podařilo určit přenos

$$G_1 = \frac{\Omega(s)}{U(s)} = \frac{9,63}{s + 1,07} \quad (3.13)$$

3.3.0.2 Úhel natočení obruče

Podle rovnice (3.3) je zřejmé, že přenos pro úhel natočení obruče je

$$G_2(s) = \frac{\varphi(s)}{U(s)} = \frac{1}{s} \frac{9,63}{(s + 1,07)} \quad (3.14)$$

3.3.0.3 Poloha kuličky

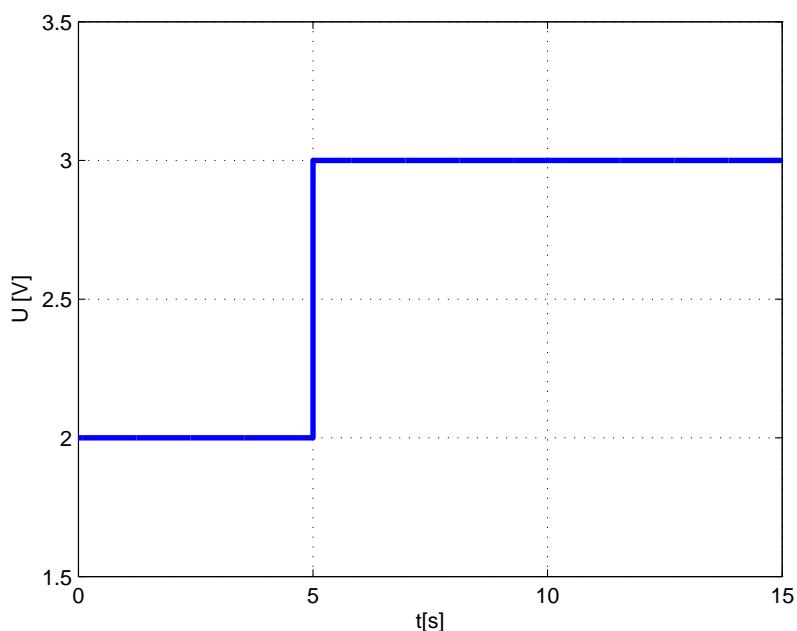
V rovnici pro popis polohy kuličky máme několik konstant, které můžeme změřit. Plomér oboru je 8,7cm, poloměr kuličky 9,5mm a hmotnost kuličky 28,3g. Zbylé konstanty bylo třeba určit experimentálně a sice tření působící na kuličku a moment setrvačnosti motoru. Výsledná rovnice pro polohu kuličky je

$$\ddot{\psi}(t) = -3,8586\dot{\psi}(t) - 90,4147 \sin(\psi(t)) + 0,7802 \frac{d\omega(t)}{dt}. \quad (3.15)$$

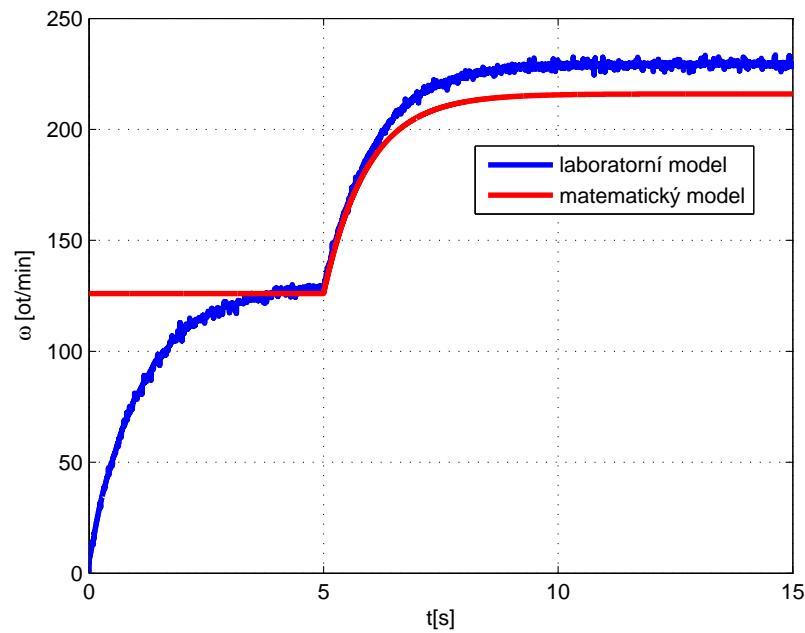
3.3.1 Měření

Na modelu jsem provedl řadu měření a výsledné průběhy porovnal s matematickým modelem. Během prvních 5 sekund se systém ustaluje do pracovního bodu, poté se skutečné průběhy moc neliší od průběhů na matematickém modelu.

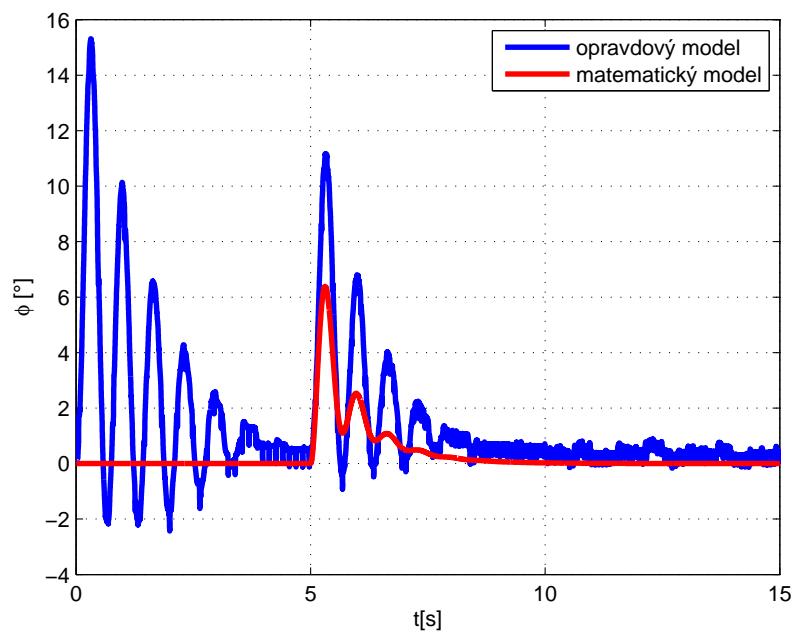
3.3.1.1 skok + 1V



Obrázek 3.5: Vstupní napětí - skok

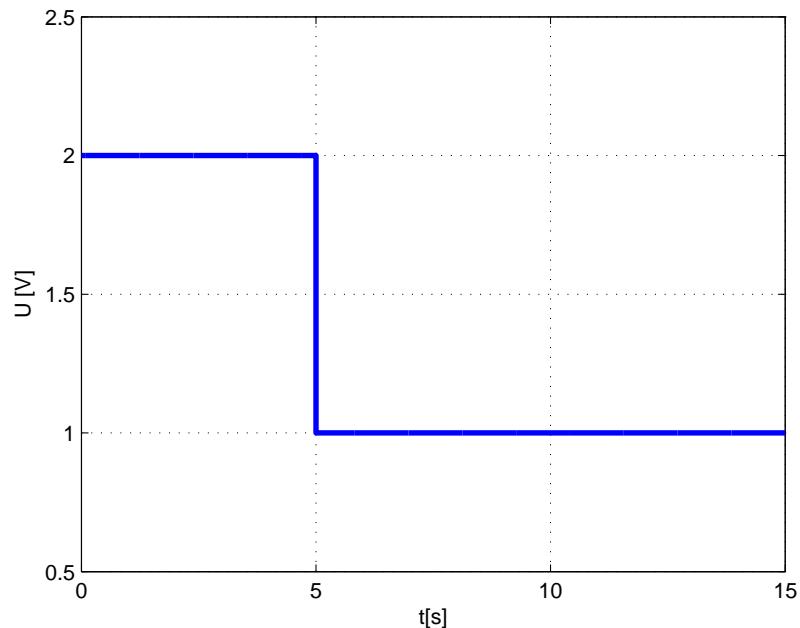


Obrázek 3.6: Otáčky obruče

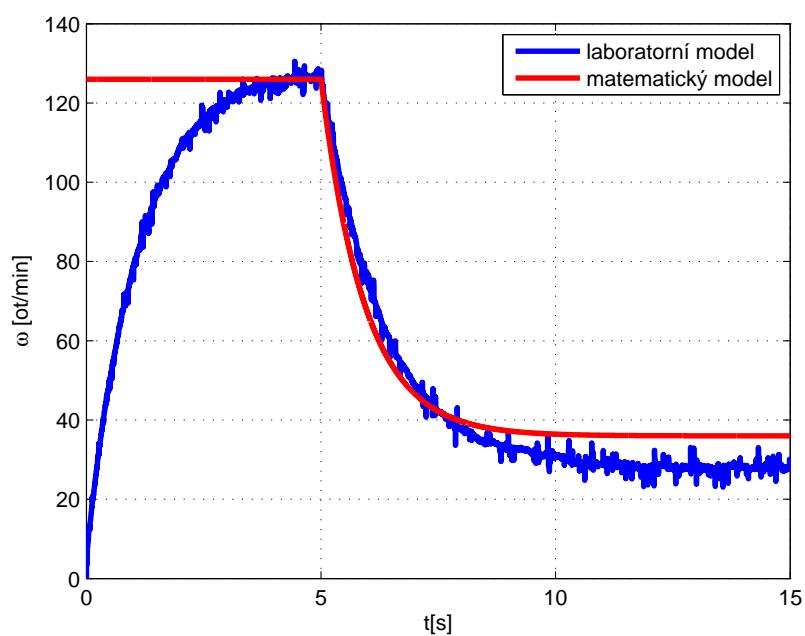


Obrázek 3.7: Poloha kuličky

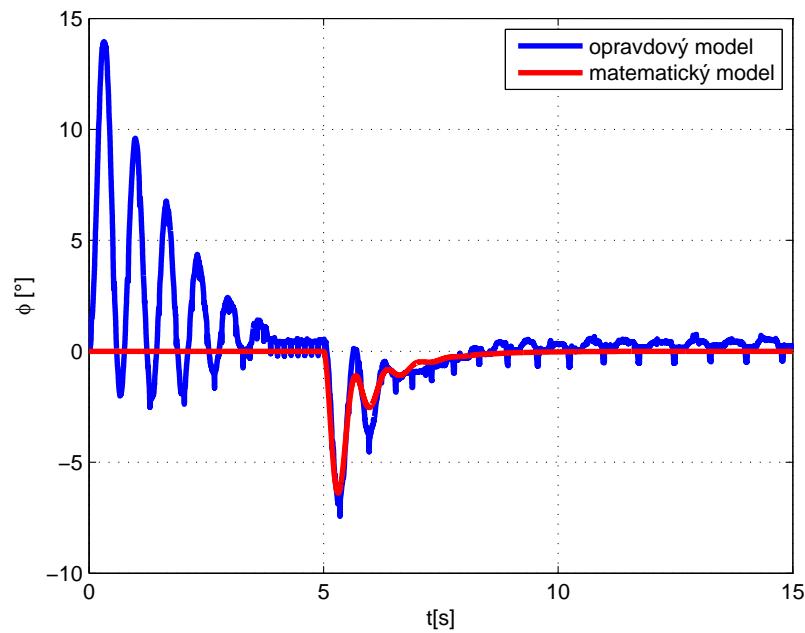
3.3.1.2 skok - 1V



Obrázek 3.8: Vstupní napětí - skok

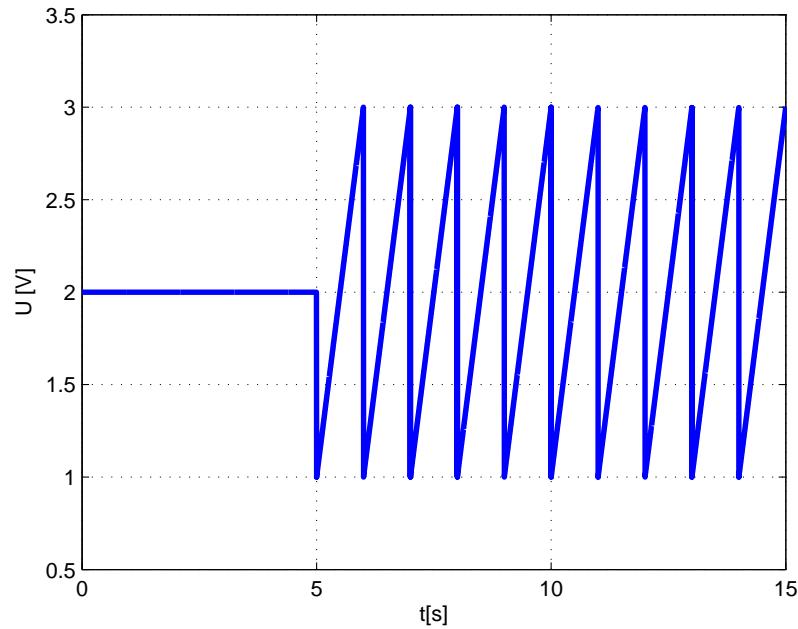


Obrázek 3.9: Otáčky obruče

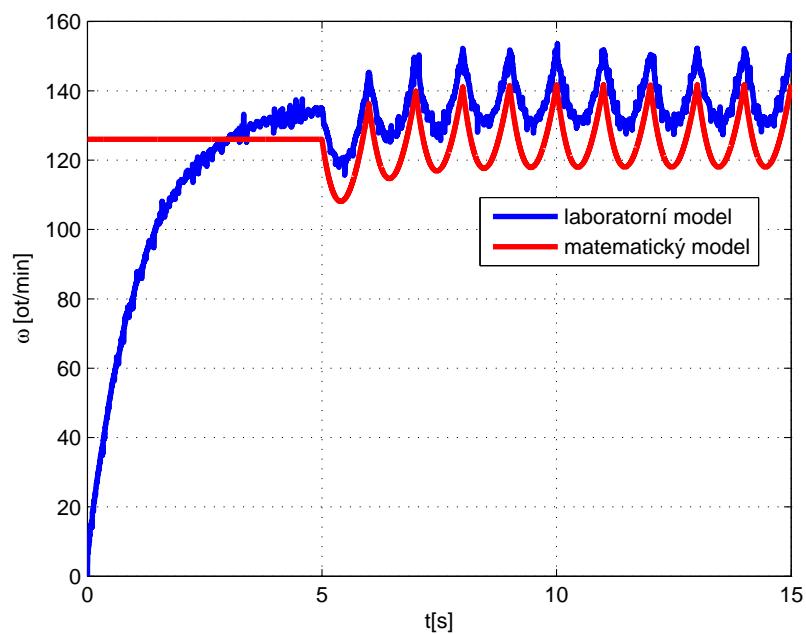


Obrázek 3.10: Poloha kuličky

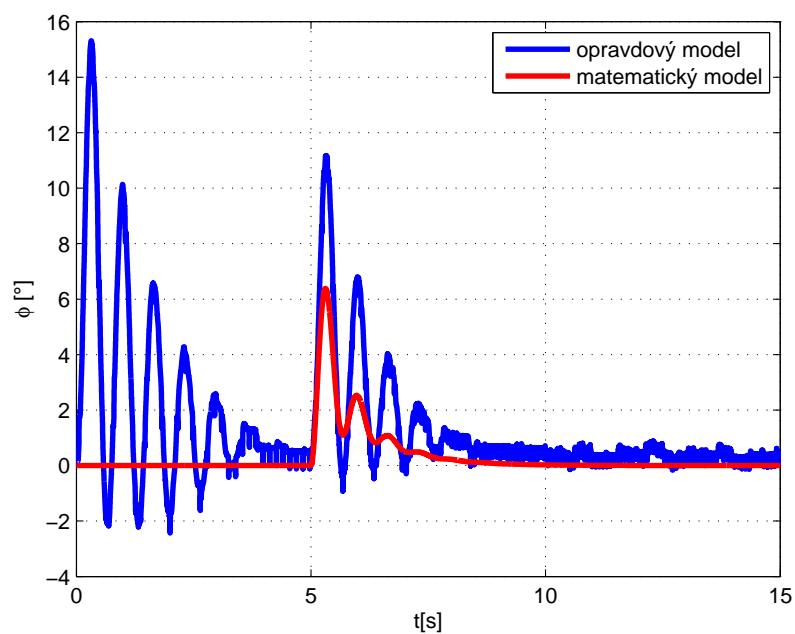
3.3.1.3 trojúhelníkový vstup



Obrázek 3.11: Vstupní napětí - trojúhelník

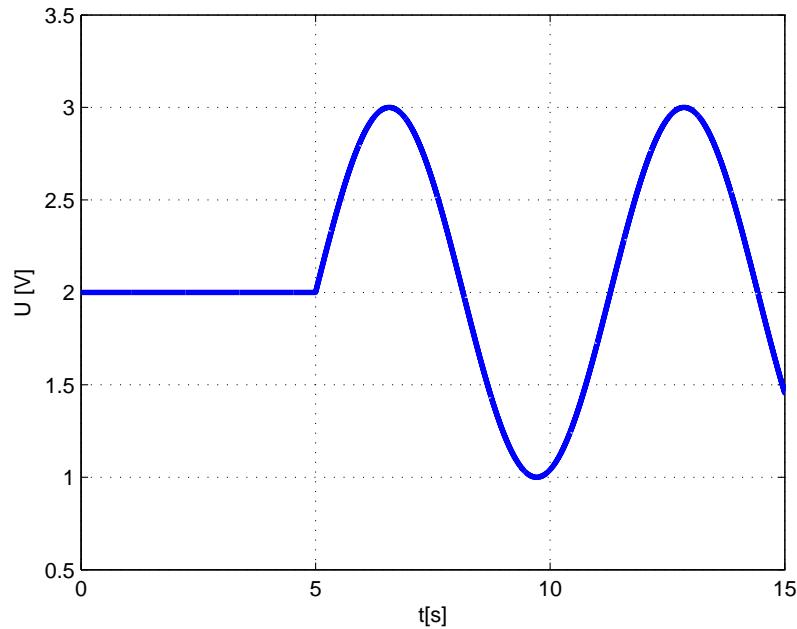


Obrázek 3.12: Otáčky obruče

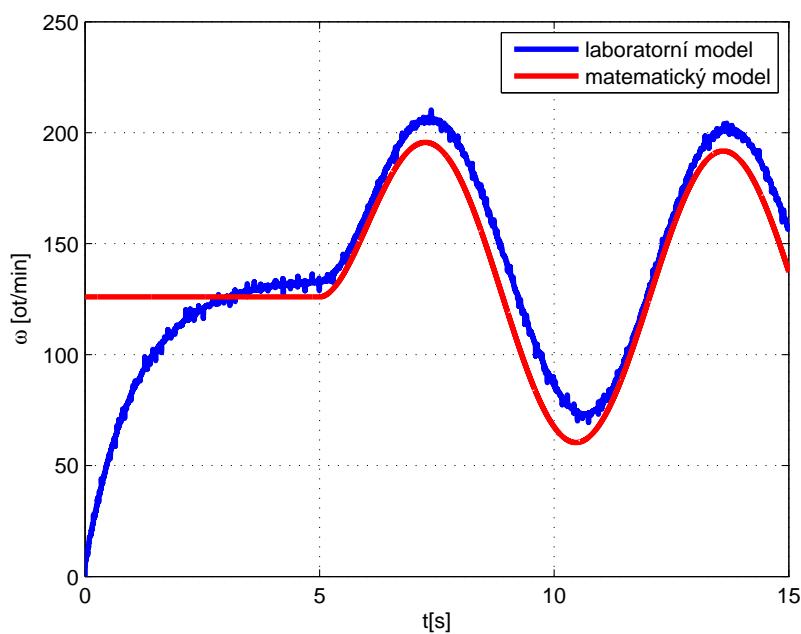


Obrázek 3.13: Poloha kuličky

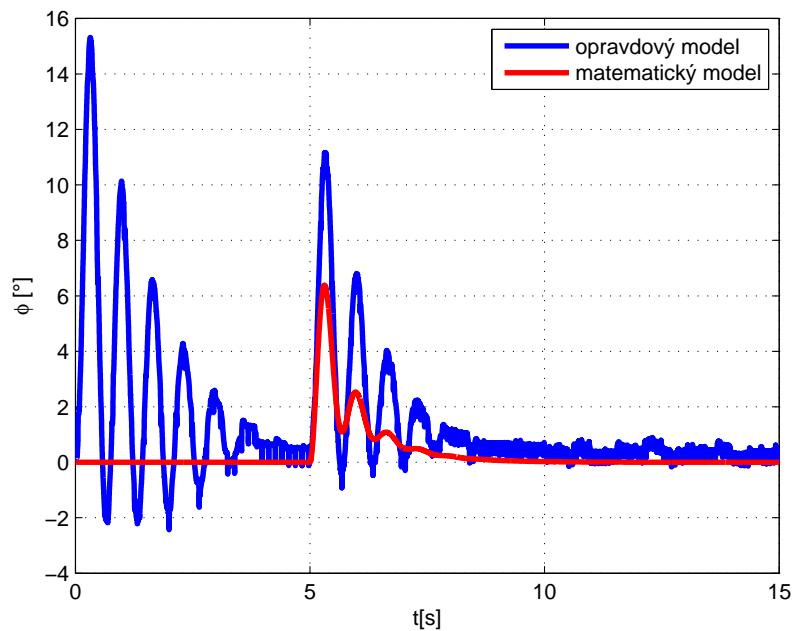
3.3.1.4 sinusový vstup



Obrázek 3.14: Vstupní napětí - sinus

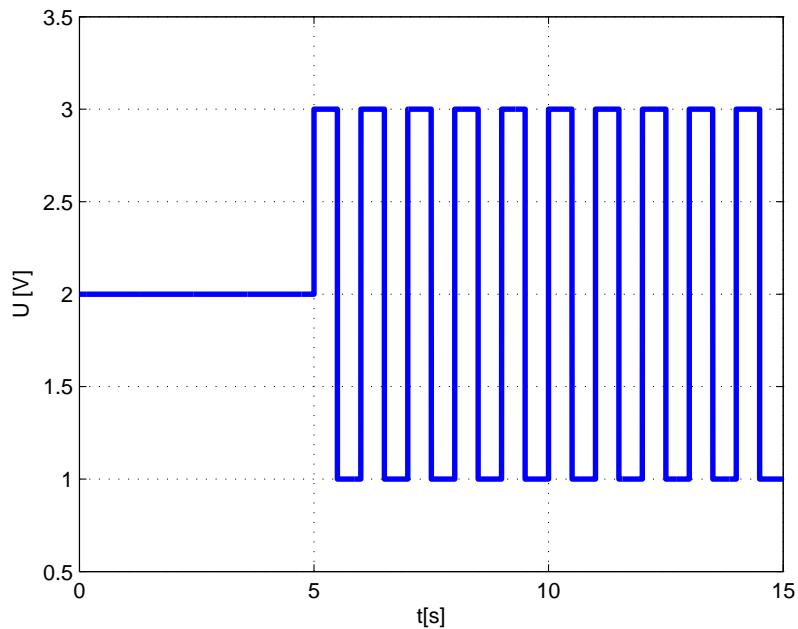


Obrázek 3.15: Otáčky obruče

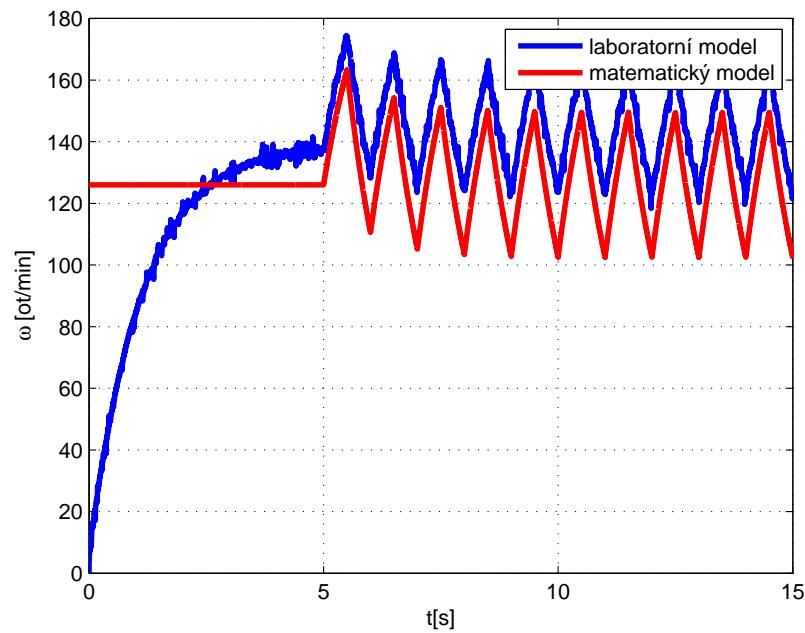


Obrázek 3.16: Poloha kuličky

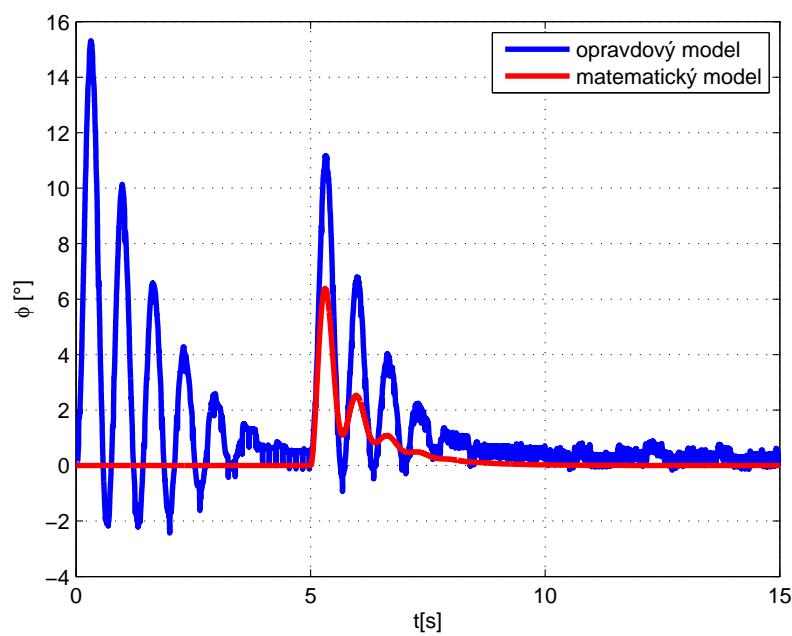
3.3.1.5 obdélník vstup



Obrázek 3.17: Vstupní napětí - obdélník



Obrázek 3.18: Otáčky obruče



Obrázek 3.19: Poloha kuličky

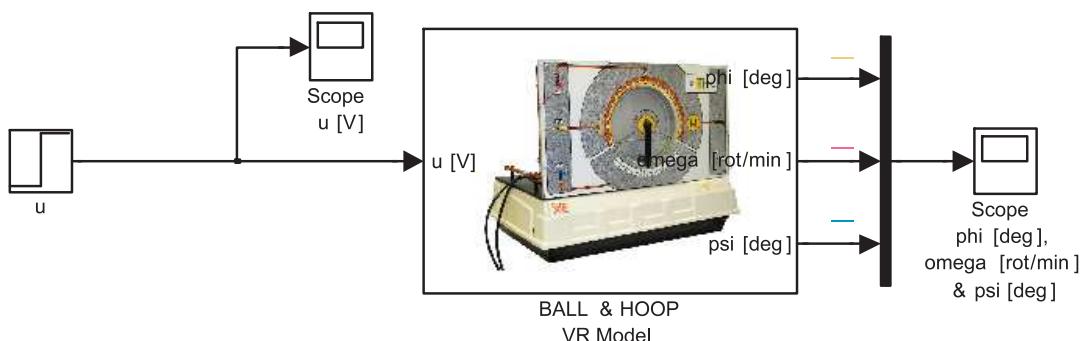
3.3.2 Závěr

Virtuální model měl principiálně odpovídat skutečnému, proto nevadí, že průběhy virtuálního a skutečného modelu nejsou úplně shodné. Odchylky mohly vzniknout z důvodu nepřesné identifikace, nebo v případě otáček nahrazením systému druhého řádu řádem prvním. Na změnu vstupního signálu reaguje virtuální i skutečný model přibližně stejně. Virtuální model je tedy použitelný pro naše účely (trénink identifikace pro studenty).

3.4 Virtuální realita

Matematický model použiji pro vytvoření virtuální reality. Simulinkové schéma jsem uzavřel do masky. Rozkliknutím bloku BALL and HOOP VR Model můžem uživatel nastavit počáteční podmínky kuličky.

Ball and Hoop (BH)

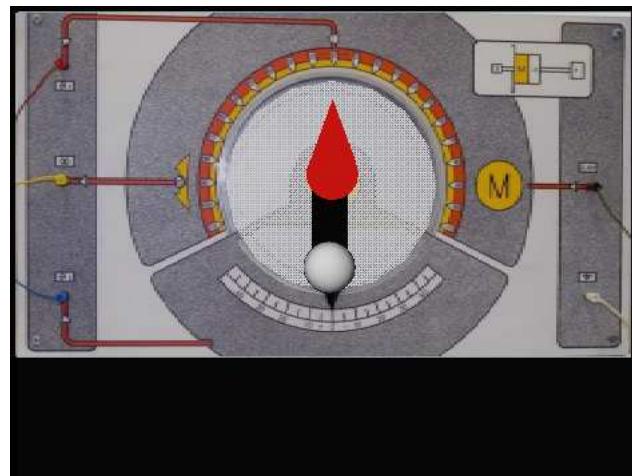


Obrázek 3.20: Virtuální model

Virtuální realitu jsem vytvořil pomocí programu V-Realm Builder. Tento program umožňuje vytvoření 3D modelu. Uživatel má k dispozici náhled ze všech stran a možnost vložení geometrických tvarů, kvádru, koule, válce a zvláštních tvarů. U jednotlivých geometrických tvarů je možné měnit velikost, natočení a polohu. Výsledný model je možné propojit se simulinkovým obvodem a zajistit tak dynamiku některých částí modelu.

Základním blokem modelu kuličky v obruci je kvádr, na který jsem vložil obrázek modelu. Dalším použitým blokem je válec, který představuje obruc. Černá šipka, která ukazuje pozici kuličky a kulička jsou spojeny do jednoho bloku. Pomocí Virtual tool-

boxu jsem pak tento blok propojil s výstupem ze Simulinku. Díky tomu se šipka natáčí stejně, jako u skutečného modelu, tedy o úhel $\psi[\circ]$. Druhým dynamickým prvkem ve virtuální realitě je červená šipky, která se otáčí stejnou rychlostí jako obrouč $\omega[\text{ot}/\text{min}]$. Díky virtuálnímu modelu si může uživatel vyzkoušet nejen jak systém reaguje, ale i jak systém vypadá.



Obrázek 3.21: Čelní pohled ve virtuální realitě

Kapitola 4

GUI pro řešení lineárních spojitých a diskrétních dynamických modelů.

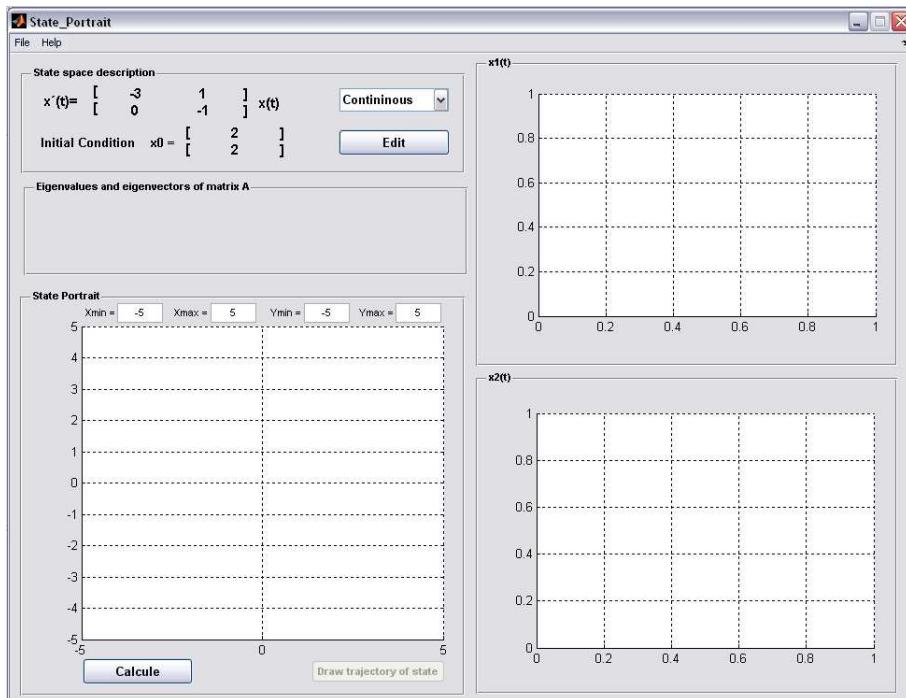
Grafické uživatelské rozhraní (dále jen GUI) slouží k usnadnění komunikace mezi uživatelem a počítačem. Má dvě části, grafickou a programovou. Grafická část zajišťuje vzhled, který se uživateli po spuštění zobrazí. Programová část zajišťuje komunikaci mezi jednotlivými grafickými prvky a funkcemi, které počítají jednotlivé parametry, kreslí grafy atd. GUI, které jsem vytvořil, kreslí stavový portrét a průběh obou vnitřních stavů. Ten reprezentuje vývoj stavů v prostoru $x_1 - x_2$.

4.1 Vytvoření GUI

Program MATLAB mimo jiné umožňuje uživateli vytvořit GUI. Editor pro vytváření GUI se spouští příkazem `guide`. Samotné GUI se skládá ze dvou souborů. Jedná se o soubory `jmenoGUI.fig`, který slouží ke grafickému vytvoření GUI a `jmenoGUI.m`, kam můžeme napsat funkce, které se budou v GUI provádět. Komunikace mezi `jmenoGUI.fig` a `jmenoGUI.m` se provádí pomocí speciálních funkcí zvaných `callback`. `jmenoGUI.m` je standartní M-file, takže vytváření metod je stejné, jako při "standartním" použití MATLABu. (*The MathWorld [online]*, 2007)

Základní funkcí celého GUI je funkce `OpeningFcn`, která se provede při spuštění programu. Podle dat, která jsou vložena do této funkce, bude vypadat úvodní obrazovka. V mém případě dojde pouze k načtení grafických prvků a za matice A a \mathbf{x}_0 se dosadí předdefinované hodnoty. Vykreslování grafů a počítání hodnot se spustí až na žádost

uživatele. Úvodní obrazovka mého programu je na následujícím obrázku



Obrázek 4.1: Úvodní obrazovka mého GUI

4.1.1 Nastavení GUI

GUI se nastavuje pomocí tlačítka, jednoho pop-up menu a čtyřech textových polí. Díky menu může uživatel vybrat, jedná-li se o systém spojity nebo diskrétní. Jak pop-up menu funguje, ukazuje následující zdrojový kód. V případě spojitého systému se provede vše, pod sekcí `case 1`, tedy označení budou $x'(t)$ a $x(t)$. V případě diskrétního systému se provedou příkazy v sekci `case 2`. Číslo `case`, která byla vybrána se uloží do hodnoty `type`. Funkce, které zajišťují výpočty a vykreslení grafů, pak podle hodnoty `type` použijí vztah pro diskrétní nebo spojity systém.

```
type = get(hObject,'Value');
switch type
    case 1
        set(handles.textDerivace,'String','x'(t)=');
        set(handles.textXt,'String','x(t)');
    case 2
        set(handles.textDerivace,'String','x(k+1)=');
        set(handles.textXt,'String','x(k)');
end
```

Tlačítkem `edit`, může uživatel změnit vstupní parametry, jmenovitě matici A a počáteční podmínky. Pro tuto editaci slouží dialogové okno, jeho vytvoření popisuje následující zdrojový kód. V dialogovém okně jsou kolonky již dopředu vyplněny. Hodnoty, které se na řádcích objeví odpovídají poslednímu nastavení programu. Okno, které se objeví, má v záhlaví název `State-space Description`. Nad každým editačním polem je uvedeno co do řádku má uživatel vyplnit, tedy A a X_0 . Na obr. 4.2 je zobrazeno, jak toto dialogové okno vypadá. Uživatel může zadat nové hodnoty. V případě, že klikne na tlačítko `Cancel`, dialogové okno se uzavře a GUI zůstane nastavené stejně, jako před otevřením dialogu. V případě, že klikne na `OK`, nové hodnoty projdou testem, který ověřuje, že políčka byla vyplněna správně, tzn. nebyla použita písmena (kromě e) a matice mají správný rozměr, matice A 2×2 a počáteční podmínky 2×1 . V případě, že je test úspěšný, staré grafy se smažou a uživateli se zobrazí nově navolené parametry. Čísla se zaokrouhlí na dvě desetinná místa a budou ve tvaru `cislo e cislo`. Po stisknutí tlačítka `calcule` dojde k vypočítání stavů a vykreslení grafů. Není-li test úspěšný, je uživatel upozorněn na chybu.

```
%nactu si z textoveho pole matici A
%tim se uživateli vzdy nabidne jako preddefinovana odpoved podledni pouzita matice
A = editMatice(handles.textA1,handles.textA2);
Ax=mat2str(A);

%nacteni pocatecni podminky
X = editMatice(handles.textIC);
Xx = mat2str(X);

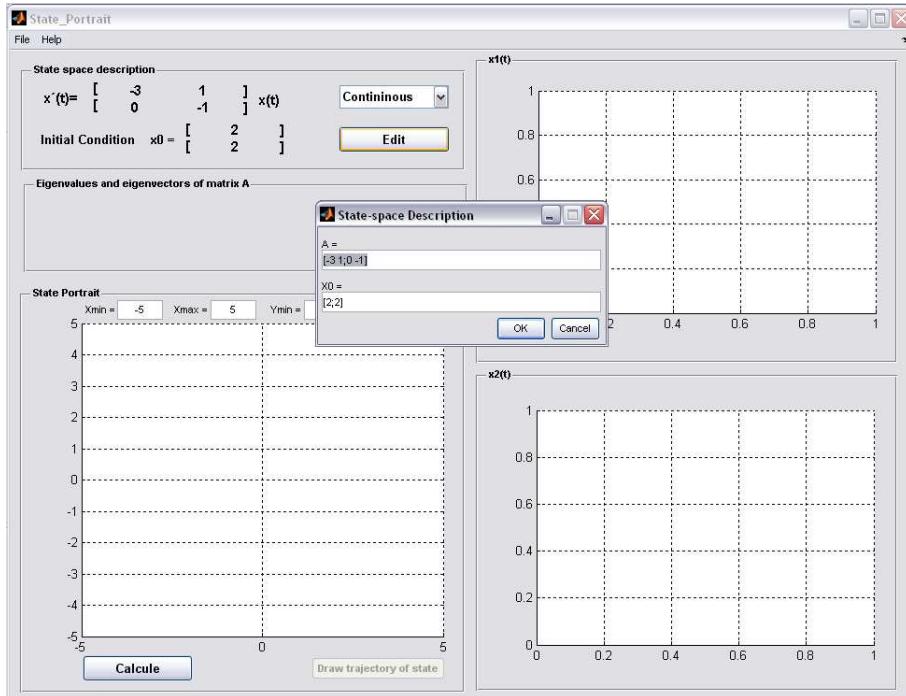
% nazev editacnihc poli
prompt = {'A =', 'X0 ='};

% nadpis dialogoveho okna
dlg_title = 'State-space Description';

% delka editacnich poli
length_basic = 70;
length_edit=max([length(Ax) length(Xx) length_basic]);
num_lines = [1 round(length_edit*0.7); 1 round(length_edit*0.7) ];

%preddefinovana odpoved
defAns = {Ax,Xx};

% otevre dialog
answer = inputdlg(prompt,dlg_title,num_lines,defAns);
```



Obrázek 4.2: Zadávání vstupních parametrů

Čtyři textová pole, která jsou umístěna nad grafem stavového portrétu, slouží k nastavení rozsahu právě grafu stavového portréu. Vložená čísla procházejí testem, jestli se opravdu jedná o čísla a jestli maximální hodnota je větší, než minimální.

4.1.2 Kreslení grafů

Grafy se vykreslí až po stisknutí tlačítka `calcule`. V callbacku tohoto tlačítka dojde ke získání matice A z textového pole a zavolání metod pro kreslení průběhu funkcí.

Pro kreslení používám pomocnou funkci `calculeEq`, která vypočítá hodnoty stavů a uloží je do vektoru \mathbf{x} . V případě, že se jedná o spojitý systém, průběhy stavů se počítají podle vzorce $\mathbf{x}(t) = e^{\mathbf{A}t} \mathbf{x}_0$. Diskrétní pak podle vztahu $x(k) = A^k x_0$.

```
switch typeSys
    %spojity system
    case 1
        for i = 1 : 1 : max(size(t));
            x(:,i) = expm(A*t(i))*x0;
        end
    Iout = x(j,:);

```

```
%diskretni system
case 2
    for i = 1 : 1 : max(size(t));
        x(:,i) = A^t(i)*x0;
    end
    Iout = x(j,:);
end
```

Program kreslí celkem tři grafy. První dva zobrazují průběhy stavů. Pro vykreslení používám funkci `axesTX_Plot`, ta si z funkce `calculeEq` vezme vypočítané hodnoty a vykreslí je do příslušných grafů. Vstupními parametry funkce `axesTX_Plot` jsou počáteční podmínky, matice A a koeficient j , který určuje jedná-li se o první nebo druhý stav.

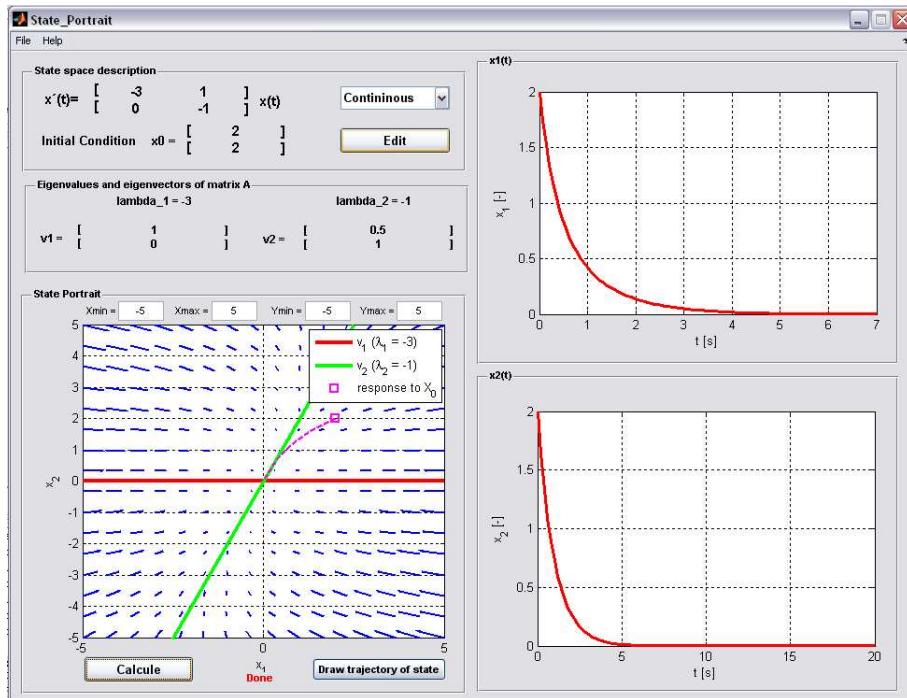
```
%rozhodne jestli se jedna o diskretni nebo spojity system
switch typeSys
    case 1
        %casova baze se pocita podle velikosti vlastniho vektrou
        t = linspace(0,ceil(1/abs(lambda(j))*20));
        %zavola metodu pro vypocitani stavovych rovnic
        eq = calculeEq(A, x0, j, t, typeSys);
        %vykresleni prubehu
        plot(t,eq,'Color','LineColor','LineStyle','LineWidth',2.5);
        grid on;
        xlabel('t [s]'); ylabel(sprintf('x_%d [-]',j));
    case 2
        k = [0:1:10];
        eq = calculeEq(A, x0, j, k, typeSys);
        stairs(k,eq,'Color','LineColor','LineStyle','LineWidth',2.5);
        grid on;
        xlabel('k [-]'); ylabel(sprintf('x_%d [-]',j));
end
```

Posledním grafem je stavový portrét. Tento graf představuje vývoj stavu. Jedná se o trajektorie v prostoru $x_1 - x_2$. Aby bylo vidět, jak se stavy vyvíjejí bylo třeba nahradit plné čáry šipkami. K vykreslení šipek můžeme použít funkci `quiver(a1,b1,a2,b2)`. Pro vykreslení šipky je tedy nutné znát počáteční a koncový bod podle obou os. Pro ukázku přikládám kus zdrojového kódu, který kreslí stavový portrét. Přes graf pro stavový portrét položím mrázku s různými počátečními podmínkami. Z každé počáteční podmínky pak určíme směr šipky. Délku šipky určí čas t . K vykreslení šipky pak použiji již zmíněnou funkci. Jako počáteční parametry použiji vygenerovanou počáteční podmínku, tedy $X0$. Koncové souřadnice šipky pak zajistí rozdíl mezi souřadnicemi $X0$ a $X1$. Na obr. 4.3 je pak vidět, jak program vykreslil všechny tři grafy

```

for x1 = Xmin : dx : Xmax
    for x2 = Ymin : dy : Ymax
        %urcení počáteční podmínky
        X0=[x1;x2];
        %vypočet hodnoty v nové počáteční podmínce
        X1 = expm(A*t)*X0;
        %vykreslení pomocí předchozí a následující počáteční podmínky
        quiver(X0(1), X0(2), X1(1)-X0(1), X1(2)-X0(2), 0.4, 'b','LineWidth',2);
    end
end

```



Obrázek 4.3: Výsledné vykreslené průběhy

4.1.3 Vykreslení konkrétní počáteční podmínky

Pod stavovým portrétem se nachází ještě jedno tlačítko. Po kliknutí na něj se zavolá funkce `ginput(1)`. V okamžiku, kdy uživatel stiskne levé tlačítko na myši, tato funkce vrátí souřadnice. Pomocí podmínek je omezeno kliknutí pouze v rozsahu stavového portrétu. Uživatel může tedy kliknout kamkoliv do stavového portrétu a určí tak hodnoty nové počáteční podmínky. Podle ní dojde k dopočítání stavové rovnice a vykreslení obou stavů a do stavového portrétu se přikreslí trajektorie z této počáteční podmínky. Protože funkce na vykreslování průběhů mají jako jeden vstupní parametr i počáteční podmínku, zavolám tyto funkce s počáteční podmínkou, kterou mi vrátila funkce `ginput`. Aby grafy byly přehledné, chtěl jsem vykreslit každý průběh jinou barvou. Protože MATLAB používá

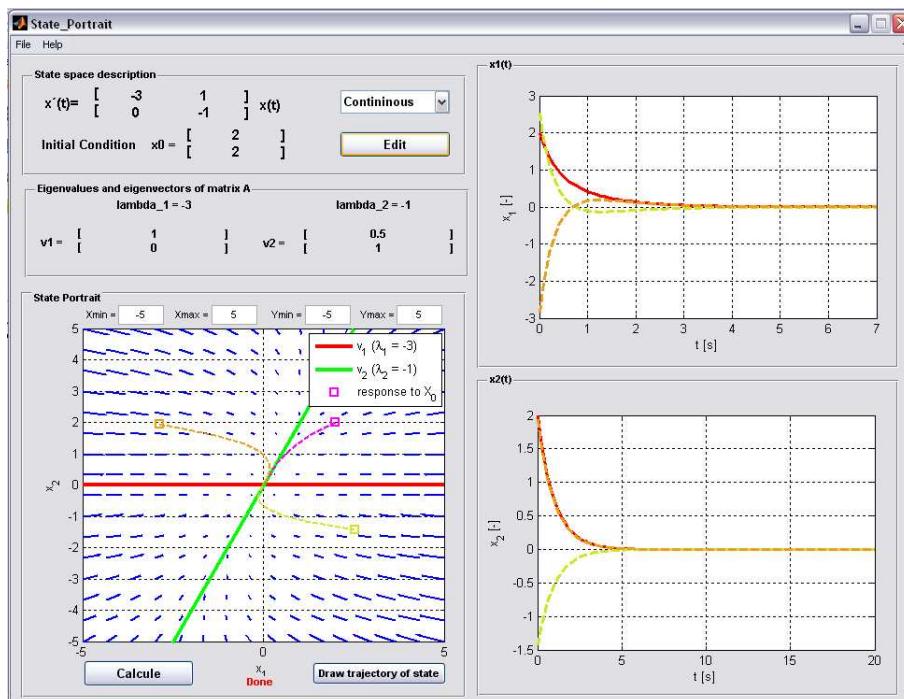
RGB paletu, tak použiji třikrát funkci `rand(1)`, která vrátí náhodnou hodnotu v rozsahu $\langle 0, 1 \rangle$ a vygeneruje tak náhodnou hodnotu pro červenou, zelenou a modrou. Jejich smícháním vznikne výsledná barva. Do grafů jsem přikreslil odezvu na dvě počáteční podmínky, to je vidět na obr. 4.4.

```
%podle toho kam uživatel kline se nactou pocatecni podminky
x0=ginput(1)';
%uživatel musi kliknout na graf s portretem
CursorPosition = get(handles.figure1,'CurrentPoint');
if (CursorPosition(1) >= 14 && CursorPosition(2) >= 5) ...
&&(CursorPosition(1) <= 93 && CursorPosition(2) <= 32)
    %zavola metodu ktera napocita hodnoty stavu
    eq1 = calculeEq( Anew, x0, 1, t, type);
    eq2 = calculeEq( Anew, x0, 2, t, type);
    %vygeneruje se automaticka barva
    r=rand(1);
    g=rand(1);
    b=rand(1);
    %ctvercem se oznaci misot pocatecni podminky a z nej carkovane
    %prubeh
    j=plot(x0(1),x0(2),'s',eq1,eq2,'--','LineWidth',1.5);
    set(j,'Color',[r g b]);
    %do grafu stavu 1 se prikresli prubeh
    axes(handles.axesX1);
    axesTX_Plot(hObject, eventdata, handles,type,Anew,x0,1,'--',[r g b]);
    %do grafu stavu 2 se prikresli prubeh
    axes(handles.axesX2);
    axesTX_Plot(hObject, eventdata, handles,type,Anew,x0,2,'--',[r g b]);

else
    errordlg(sprintf('You must click on the portrait graf!!!'), 'Error position');
end
```

4.1.4 Export grafu do figure

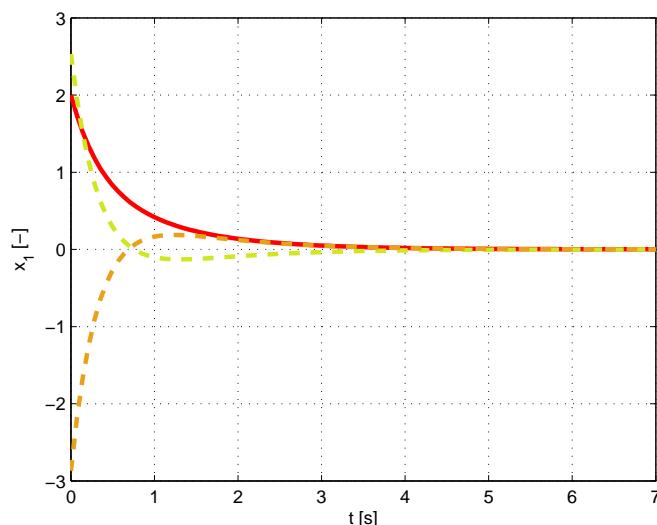
Aby uživatel mohl výsledné průběhy dále používat, je možné je vyexportovat z GUI do `figure`. Pro vyexportování je třeba kliknout na panel, ve kterém je graf zobrazen. `callback` tohoto kliknutí zavolá metodu pro exportaci. Použil jsem MATLABovskou funkci `copyobj`. Tato funkce umožňuje překopírovat obsah z jednoho objektu do druhého. Na následujícím zdrojovém kódu je vidět jak tuto funkci používám. Nejprve dojde k vytvoření `figure` pomocí příkazu `newFig=figure(j)`. Do ní se pak pomocí funkce `copyobj` překopíruje obrázek z GUI. Tato funkce má dva vstupy. V mém případě je první objekt graf v GUI a druhým `figure`, do které se vykreslí. Parametr `j` určuje co se má vykreslit pro stav x_1 je $j = 1$, pro stav x_2 je $j = 2$ a pro stavivý portrét je $j = 3$. Příkaz `set` nastaví rozměry nově vzniklého obrázku, aby odpovídal standartnímu `figure`, který MATLAB vykresluje. Na obr. 4.4 vidíte, jak v rámci GUI vypadá průběh prvního stavu.



Obrázek 4.4: Byly přikresleny odezvy na dvě počáteční podmínky

Na obr. 4.5 je pak tento stav vyexportován do figure.

```
%vytvori figure
newFig = figure(j);
%prekopiruje z axes do figure
fig = copyobj(axesObject,newFig);
%nastavi stjeny rozemr jako ma standartni okno figure v MATLABu
set(get(j, 'CurrentAxes'), 'Position', [15.1300    4.3077    86.6700    26.3846])
```



Obrázek 4.5: První stav vyexportovaný do figure

Kapitola 5

Závěr

První část mé bakalářské práce mi pomohla k rozšíření znalostí získaných v předmětu X35SAM – Systémy a modely. Stavový popis je důležitým nástrojem pro popisování a řízení systémů. Znalost průběhu jednotlivých stavů a výstupů nám umožňuje navrhnut optimální regulátor pro daný systém. Popsaná teorie je ukázána na sérii řešených příkladů. Pro další studium jsou připraveny neřešené příklady, čtenář má k dispozici pouze správné výsledky.

Dále jsem vytvořil virtuální model systému kulička v obruči. Virtuální modely jsou velmi praktické, protože umožňují studium systémů mimo laboratoř. Práce na modelu kulička v obruči byla velice zajímavá. U tohoto modelu se mi podařilo velmi dobře vytvořit matematický model podle sil, které na soustavu působí. Protože většina proměnných se dala změřit, výsledný model je velmi přesný a odpovídá požadavkům ze zadání. Vytvoření virtuální reality bylo dosti pracné. Při umístění nového dílku do virtuálního světa bylo nutné sledovat jeho umístění ze všech směrů, nestačil pouze čelní pohled. Podařilo se mi sestavit funkční model, který je ke stažení na stránkách Laboratoře teorie automatického řízení (ROUBAL, J. a HOLEČEK, J., 2008).

Program MATLAB má v technice velké uplatnění. Zatím jsem ho používal pouze pro kreslení grafů a různé výpočty. Ale díky práci na GUI pro zobrazování stavového portrétu jsem se naučil používat MATLAB i jiným způsobem. Samotné GUI je pak využitelné pro výuku na Katedře řídicí techniky. Z tvaru stavového portrétu můžeme snadno zjistit, jestli je zadaný systém stabilní, nebo nestabilní, jak se vyvíjejí jednotlivé stavy v čase atd.

Literatura

- Roubal, J. (n.d.), Jirkovy stránky [online]. [cit. 2008-05-11] [\(http://support.dce.felk.cvut.cz/pub/roubalj/\)](http://support.dce.felk.cvut.cz/pub/roubalj/).
- ANTSAKLIS, P. J. a MICHEL, A. N. (2007), *A Linear Systems Primer*, 1st edn, Springer. ISBN 978-0-8176-4460-4.
- BABUŠKA, I., PRÁGER, M. a VITÁSEK, E. (1966), *Numerical Processes in Differential Equations*, London – New York – Sydney, Interscience Publishers.
- BRABEC, J. a HRŮZA, B. (1986), *Matematická analýza II*, Praha: Státní nakladatelství technické literatury.
- CHEN, C. T. (1998), *Linear System Theory and Design*, 3rd edn, Oxford University Press. ISBN 0-195-11777-8.
- DORF, R. C. a BISHOP, R. H. (2007), *Modern Control Systems*, 11th edn, Prentice-Hall. ISBN 0132270285.
- FRANKLIN, G. F., POWELL, J. D. a EMAMI-NAEINI, A. (2005), *Feedback Control of Dynamic Systems*, 5th edn, Prentice-Hall. ISBN 0-13-149930-0.
- KAILATH, T. (1980), *Linear Systems*, Prentice Hall, New Jersey. ISBN 0-135-36961-4.
- REKTORYS, K. (1969), *Survey of Applicable Mathematics*, 1st edn, MIT Press. ISBN 0592039277.
- ROUBAL, J. a HOLEČEK, J. (2008), Laboratoř teorie automatickeho řízení [online]. [cit. 2008-05-11] [\(http://support.dce.felk.cvut.cz/lab26/\)](http://support.dce.felk.cvut.cz/lab26/).
- ROUBAL, J., HUŠEK, P. a SPOL. (200x), *Základy regulační techniky v příkladech*, Pracovní verze. [\(http://support.dce.felk.cvut.cz/pub/roubalj/\)](http://support.dce.felk.cvut.cz/pub/roubalj/).

ŠTECHA, J.; HAVLENA, V. (1999), *Teorie dynamických systémů*, Praha: Vydavatelství ČVUT. ISBN 80-01-01971-3.

VOLTERRA, V. (1931), ‘Variations and Fluctuations of the Number of Individuals in Animal Species Living Together’, *Animal Ecology*, pp. 409–448. Translated from 1928 edition by R. N. Chapman.

The MathWorld [online] (2007). [cit. 2005-10-25] <<http://mathworld.wolfram.com/>>.

Wikipedia – Open encyclopedia [online] (2008). [cit. 2008-05-11] <<http://wikipedia.org/>>.

Příloha A

Obsah CD

Na přiloženém CD je

- bakalářská práce ve formátu pdf,
- virtuální model kuličky v obruči,
- GUI pro vykreslení stavového portrétu.