

Deterministický chaos v prostředí Mathematica

Bc. František Můčka

Diplomová práce
2006

 Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. František MŮČKA**
Studijní program: **N 2807 Chemické a procesní inženýrství**
Studijní obor: **Automatizace a řídicí technika**

Téma práce: **Deterministický chaos v prostředí Mathematica**

Zásady pro vypracování:

Práce bude zaměřena na principy a využití deterministického chaosu. Náplní práce bude vytvoření programového balíku demonstrujícího chaos a umožňujícího ovládní reálného modelu :

- 1. vypracovat přehled problematiky řízení deterministického chaosu**
- 2. vybrat vhodné již řešené příklady**
- 3. vypracovat alternativní řešení pomocí sw Mathematica**
- 4. Provést závěr**

Rozsah práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

Arrowsmith D.K., Place C.M., An Introduction to Dynamical Systems, Cambridge University Press, UK, ISBN 0-521-316-50, 1990

Gleick J.1996, Chaos, Ando publishing, ISBN 80-86047-04-0, 1996

Hilborn R.C.1994, Chaos and Nonlinear Dynamics, Oxford University Press, ISBN 0-19-508816-8, 1994

Horák J., Krlín L., Deterministický chaos, Academia, ISBN 80-200-0416-5, 1996

Marek M., Schreiber I., Chaotic Behaviour of Deterministic Dissipative Systems, Academia, ISBN 80-200-0186-7

Nicolis G., Introduction to Nonlinear Science, Cambridge University Press, ISBN 0-521-46782-9, 1982

Palis J., Takens F.1993, Hyberbolicity & sensitive chaotic dynamics at homoclinic bifurcations, Cambridge University Press, ISBN 0-521-47572-4, 1993


Peitgen H.O., Jurgens H., Saupe D. 1992, Chaos and Fractals, New Frontiers of Science, Springer-Verlag 1992, ISBN 3-540-97903-4

Vedoucí diplomové práce: **doc. Ing. Ivan Zelinka, Ph.D.**
Ústav aplikované informatiky

Datum zadání diplomové práce: **14. února 2006**

Termín odevzdání diplomové práce: **26. května 2006**

Ve Zlíně dne 25. února 2006


prof. Ing. Vladimír Vašek, CSc.
pověřený děkan




prof. Ing. Petr Dostál, CSc.
ředitel ústavu

ABSTRAKT

Tato diplomová práce je zaměřena na realizaci softwarového rozhraní mezi reálným modelem a softwarovým prostředím Mathematica. Vybraný reálný model – spřažené elektromotory - byl spojen se softwarem Mathematica a byla na něm provedena měření otáček a protažení pásu s cílem prokázat výskyt deterministického chaosu.

Klíčová slova: deterministický chaos, reálný model, měření, Lyapunovy exponenty

ABSTRACT

This master thesis is concerned to realization of software interface between real model and software environment Mathematica. Chosen real model – coupling electro motors – was connected with software Mathematica and measurements of revolutions and belt elongation were carried out to prove occurrence of deterministic chaos.

Keywords: deterministic chaos, real model, measurement, Lyapun exponents

Tímto bych chtěl poděkovat doc. Ing. Ivanovi Zelinkovi, PhD. za cenné rady a nápady při vypracovávání diplomové práce a Ing. Zuzaně Oplatkové za konzultace týkající se prostředí Mathematica . V neposlední řadě děkuji své rodině za podporu při studiu a za finanční zabezpečení.

Ve Zlíně

.....

Jméno diplomanta

OBSAH

ÚVOD	7
I TEORETICKÁ ČÁST	8
1 OSOBNOSTI V HISTORII CHAOSU	9
2 HAMILTONIÁNSKÉ SYSTÉMY	17
3 DISIPATIVNÍ SYSTÉMY	20
3.1 ŘÍZENÍ CHAOSU – PRINCIPY A TRENDY	20
3.2 NEJEDNODUŠŠÍ MODEL DETERMINISTICKÉHO CHAOSU - LOGISTICKÁ ROVNICE.....	24
3.3 DETERMINISTICKÝ CHAOS – VÝSKYT.....	27
II PRAKTICKÁ ČÁST	29
4 ROZHRANÍ REÁLNÉHO MODELU A SOFTWARE MATHEMATICA	30
4.1 POPIS REÁLNÉ SOUSTAVY	30
4.1.1 Části soustavy.....	32
4.1.2 Model	34
4.2 PROPOJENÍ REÁLNÉHO MODELU S PC	35
4.2.1 Popis technologické karty	35
4.2.2 Instalace technologické karty.....	37
5 REALIZACE SOFTWARE ROZHRANÍ	39
5.1 POPIS OVLADAČE KARTY ADVANTECH PCL-812PG.....	39
5.2 POPIS ROZHRANÍ C++ A MATHEMATICA (MATHLINK).....	39
5.3 POPIS PROGRAMU CE108	40
6 POUŽITÍ PROGRAMU CE108 V MATHEMATICI	42
7 ZHODNOCENÍ VÝSLEDKU	44
7.1 LYAPUNOVY EXPONENTY	45
7.2 BIFURKAČNÍ DIAGRAMY	48
7.3 ZÁVISLOST PŘEDCHOZÍHO PRVKU NA NÁSLEDUJÍCÍM	51
7.4 VYHODNOCENÍ METOD	53
ZÁVĚR	54
SEZNAM POUŽITÉ LITERATURY	55
SEZNAM OBRÁZKŮ	57
SEZNAM PŘÍLOH	59

ÚVOD

Když se vysloví slovo „chaos“, obvykle se u lidí, kteří se touto disciplínou nezabývají, vybaví proces který je čistě nahodilý a bez vnitřních zákonitostí. Málokdo si uvědomuje že „být chaotický“, znamená striktně se řídit přesně danými pravidly, v nichž mnohdy není pro náhodu místo. Chaos je disciplínou, která dostala své jméno až ve dvacátém století, nicméně jeho kořeny sahají až do 19 století, kdy bylo zjištěno, že už jednoduché problémy generují velmi složité a nepředpověditelné chování [1,2].

V poslední době vznikl v souvislosti s deterministickým chaosem směr zvaný řízení deterministického chaosu. Je to relativně nový vědecký interdisciplinární směr, v němž dochází k symbióze více oborů jako je např. fyzika, chemie, biologie či elektronika. To vše pak zastřešuje kybernetika, která využívá poznatků přispívajících oborů k syntéze toho, čemu se říká "řízení deterministického chaosu" [3 - 6]. Navzdory faktu, že v samotném názvu se vyskytují dvě vzájemně odporující si pojmy (řízení a chaos), je pravdou, že se jedná o rigorózní vědu řídící se striktními pravidly. Hlavním úkolem této práce je vytvoření softwarového rozhraní mezi reálným modelem a softwarem Mathematica od firmy Wolfram (www.wolframresearch.com). Následující měření otáček a protažení pásu a vyhodnocení získaných dat by mělo prokázat přítomnost chaosu. Budoucí práce by pak mohla být zaměřená na řízení deterministického chaosu na reálném modelu.

I. TEORETICKÁ ČÁST

1 OSOBNOSTI V HISTORII CHAOSU

Dvacáté století je století v němž prošlo lidstvo mnoha revolučními změnami a to jak v oblasti sociálním, tak vědecké. Ve vědě došlo k mnoha dramatickým zvrátům, které se promítly i do každodenního života běžného občana. Jedním z těchto zvrátů je i vznik „nové vědy“ populárně zvané „Chaos“. Na jejím vzniku se podílelo mnoho lidí a postupně vznikající a sílící podhoubí této disciplíny lze vysledovat až do 19. století. V tomto období vznikaly základní myšlenky, jež se později ukázaly být velmi přínosné pro chaos jako takový.

Jedním z těchto lidí byl i ruský matematik A. M. Ljapunov, jehož jméno je nejvíce známo z teorie dynamických systémů, konkrétně z oblasti jejich stability. Jeho jméno je spojeno s chaosem přes tzv. Ljapunovovy exponenty. Tyto exponenty, obvykle označované jako „ λ “, vyjadřují u chaotických systémů divergenci nebo-li rozbíhavost blízkých trajektorií. Tato rozbíhavost je jedním z charakteristických rysů chaosu .



Obr. 1 A.M.Ljapunov



Obr. 2 G.Cantor

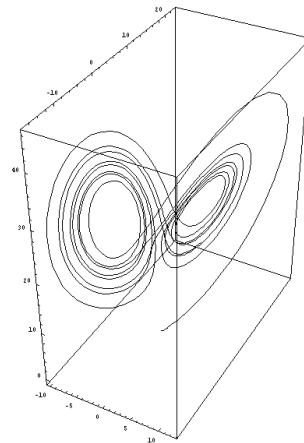
Dalším z řady významných lidí byl George Cantor, německý matematik, pracující na univerzitě v Halle. I když jeho objevy patří spíše do oblasti fraktální geometrie, je velmi často spojován s chaosem. G. Cantor je objevitel tzv. Cantorovy množiny, která byla poprvé publikována v roce 1883. Tento geometrický útvar patří do fraktální geometrie, kde náleží mezi jedny ze základních tvarů. Geometrická struktura chaotických atraktorů se mnohdy projevuje jako Cantorova množina. Konstrukce Cantorovy množiny je v podstatě triviální záležitost. Celá leží v intervalu $\langle 0,1 \rangle$ a její konstrukce spočívá ve vynechávání druhé třetiny tohoto intervalu a všech, které tímto vzniknou. To znamená, že v prvním kroku po vy-

nechání zůstanou dva menší intervaly a to $0-1/3$ a $2/3-1$. Každý z nich podrobíme stejné operaci a dostaneme čtyři menší intervaly $0-1/9$, $2/9-3/9$, $6/9-7/9$, $8/9-9/9$, atd.. Cantorovu množinu reprezentují ne koncové body úseků vzniklých během iterací, ale množina bodů, která zůstane po „nekonečně mnoha“ iteracích [7].

Za otce chaosu se považuje až E. N. Lorenz, který se proslavil objevem tzv. "Motýlího efektu", který prezentoval v 60. letech. Motýlí efekt poukazuje na nemožnost dlouhodobé předpovědi počasí (max. 2-3 dny) a tím na jeho chaotičnost. Základní "myšlenkou" motýlího efektu je, zda nepatrné mávnutí motýlích křídel nad Tokiem může způsobit uragán nad New Yorkem [7]. Lorenzův atraktor je spolu s Ljapunovovým exponentem symbolem pro chaos. Své revoluční výpočty prováděl Lorenz na počítači tehdy střední třídy, který dokázal provést úžasných 17 výpočtů za sekundu. V dnešní době superpočítače používané pro předpověď počasí provedou stovky milionů výpočtů za sekundu.



Obr. 3 Edward N. Lorenz –
Objevitel chaosu



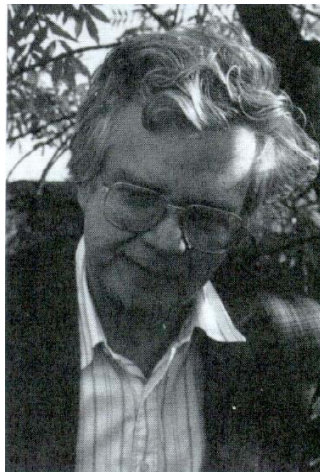
Obr. 4 Lorenzův atraktor

Lorenz měl jistý model počasí, který produkoval časovou řadu jeho vývoje. Jednou v zimě roku 1961 chtěl zkoumat podrobněji jednu z takovýchto řad. Aby si zkrátil výpočet, rozhodl se začít ne od začátku, ale od středu řady. Počáteční podmínky zadal podle staršího výpisu a šel na kávu. Po návratu na něj čekal výsledek, který dal základy nové vědě - chaosu. Obě řady se od sebe po určité době začaly rozcházet bez ohledu na to, jak přesně Lorenz zadal počáteční podmínky. Tak se zrodil termín "citlivost na počáteční podmínky".

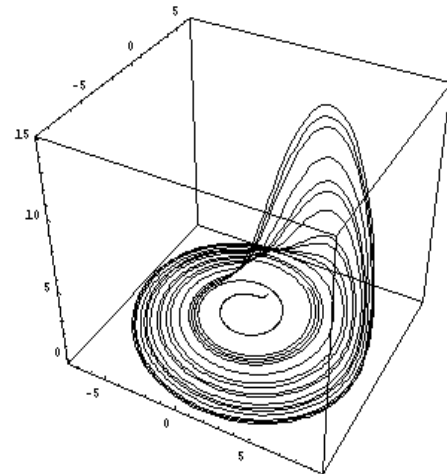
Lorenz tak poukázal na to, že mnohé systémy jsou citlivé na počáteční podmínky a že mohou díky tomu generovat to, čemu se dnes říká chaos.

Röslerův atraktor je umělý systém generující podivný atraktor založený na nejjednodušších principech generace chaosu, a to transformací "rozprostření a ohyb" (stretch and fold). Röslerův atraktor (1976) vychází z atraktoru Lorenzova, který byl publikován 13 roků před Röslerovým. Rösler byl původním povoláním lékař, který se k chaosu dostal přes chemii a teoretickou biologii. Röslerovo jméno začalo být spojováno s jednoduchým atraktorem ve tvaru stuhy, svinuté do věnečku se záhybem. Rösler si ale také představoval atraktory z vyšších dimenzí jako ohýbání a stlačování stavového prostoru, které se skutečně staly klíčem ke konstrukci podivných atraktorů. Rösler zastával názor, že tyto tvary představují princip samoorganizace v přírodě (vznik disipativních struktur díky složitým a vzájemně souvisejícím procesům). Představoval si punčochu k měření větru, do které se chytil vítr. Vítr je v pasti a musí "proti své vůli" konat něco užitečného. Podle něj princip samoorganizace spočíval v tom, že příroda dělá něco proti své vůli a díky tomu se zaplétá do sebe a dává tak vzniknout kráse. Ať už však byly jeho filozofické úvahy jakékoliv, neodmyslitelně patří do historie fraktálů a chaosu.

Balthasar Van der Pol (1889 – 1959) se narodil v Holandsku. Studoval experimentální fyziku s takovými lidmi jako byli J. A. Fleming a sir J. J. Thompson. Doktorát z fyziky obdržel v Utrechtu v roce 1920. Van der Pola lze považovat za jednoho z otců moderní experimentální dynamiky, kterou inicializoval v období 1920 – 1930 svými pracemi na elektrických obvodech, jenž obsahovaly elektronky. Během těchto prací objevil stabilní oscilace, které jsou dnes nazývány „limitní cykly“. Van der Pol zjistil při experimentech, že pokud je takovýto obvod „řízen“ signálem, jehož frekvence je blízká frekvenci limitního cyklu, pak frekvence periodické odezvy tohoto dynamického systému je „strhována“ k frekvenci řídicího signálu. V roce 1927 Van der Pol spolu se svým kolegou Van den Markem v časopise Nature publikoval článek o jevu, který se zabýval šumem objevujícím se mezi jistými jasně definovanými frekvencemi. Jak z popisu jevu tak z rekonstrukce jeho obvodu plyne, že za objevitele deterministického chaosu lze považovat už Van der Pola a ne E. N. Lorenze, jak bývá často uváděno. Jejich článek je samozřejmě jeden z prvních, který pojednává o deterministickém chaosu.



Obr. 5 O. Rösler



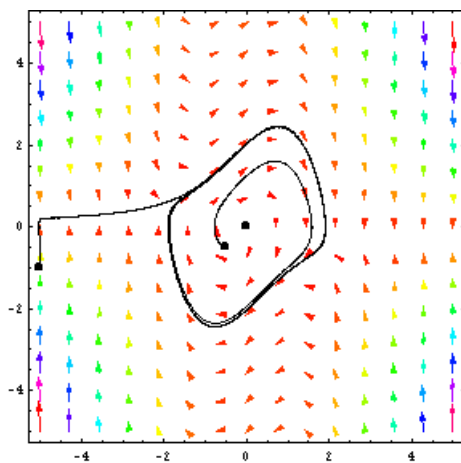
Obr. 6 Röslerův atraktor

Van der Pol postavil mnoho elektronických modelů lidského srdce za účelem studia srdeční dynamiky a stability. Jeho výzkum byl zaměřen na „řízení“ srdeční činnosti externím signálem, což je dnes záležitost kardio-stimulátorů. Jeho cílem bylo najít metodu jak stabilizovat srdeční arytmie. Van der Pol byl oceněn medailí IRE v roce 1935 za významný přínos v oblasti elektronických obvodů a fenoménu šíření elektro-magnetických vln.

Lev Davidovič Landau se narodil v Baku jako syn inženýra a doktorky. Po vystudování fyziky na Leningradské Univerzitě nastoupil na Leningradský technický institut. V letech 1929-1931 byl dočasně v Německu, Švýcarsku, Anglii a v Kodani, kde pracoval spolu s Nielsem Bohrem. V letech 1932-1937 byl vedoucím katedry fyziky na Ukrajinském technickém institutu a od roku 1937 byl vedoucím na katedře fyziky Akademie Věd SSSR v Moskvě. Současně s touto pozicí vyučoval teoretickou fyziku v Charkově a na Moskevské státní univerzitě. Jeho práce pokrývaly všechny oblasti tehdejší teoretické fyziky od mechaniky kapalin až po kvantovou teorii pole. L. D. Landau rovněž přispěl ke studiu chaosu, kde se zabýval studii turbulence. V roce 1946 by zvolen členem Akademie Věd SSSR. Mnohokrát dostal Státní cenu SSSR a v roce 1962 dostal Leninovu cenu za přínos ve fyzice. Dále byl členem mnoha dalších zahraničních společností.

Jules Henri Poincare (1854-1912) byl jednou z prvních osobností, která přispěla k teorii chaosu. I když žil ještě před oficiálním objevením deterministického chaosu, přispěl k jeho současnému studiu některými myšlenkami. Snad nejužívanějším výrazem je tzv. Poincarého plocha (rovina, řez, sekce), která slouží k vizualizaci chování mnohorozměrových dynamických systémů. J. H. Poincare byl francouzský matematik který velmi významně při-

spěl originálním způsobem k nebeské mechanice. Je zakladatelem tzv. „kvalitativní dynamiky“ - matematické teorie dynamických systémů. J. H. Poincare se intenzivně zabýval studiem pohybu tří těles ve vzájemné gravitační interakci. Své poznatky získané v tomto směru zveřejnil v práci „On The Problem of Three Bodies and the Equations of Equilibrium“. Při práci na tomto problému objevil Poincare komplexitu chování na první pohled jednoduchého systému. Problém tří těles je jedním ze základních demonstrací deterministického chaosu tzv. Hamiltoniánských systémů. Pozoruhodné je, že i v té době, kdy neexistovaly počítače, dokázal díky svým matematickým schopnostem a intuici popsat mnoho základních vlastností deterministického chaosu.



Obr. 7 Chování Van der Polova systému



S teorií chaosu dost úzce souvisí i tzv. teorie katastrof, která byla vyvinuta francouzským matematikem Rene Thomem v šedesátých letech 20. století. Tato matematická disciplína byla poprvé publikována v knize „Stabilité Structurelle et Morphogenese“ (Strukturální stabilita a morfogeneze) s podtitulem „Essai d'une theorie générale des modes“ (Esej o obecné teorii modelů). Kniha v té době vzbudila velmi silný ohlas a poměrně rozsáhlé diskuze, které byly na klasické matematické poměry dosti bouřlivé. Samotný autor tuto knihu pojal jako esej, což matematiky trochu vyvedlo z míry. Ti jsou totiž zvyklí na jiný způsob vyjadřování, čemuž se tato kniha vymykala.

Teorie katastrof je speciálním odvětvím teorie dynamických systémů zabývající se prudkými změnami chování dynamických systémů při kontinuálních a malých změnách tzv. řídicích parametrů. Vlastní teorie katastrof tedy pracuje s modely reálného světa a hledá tzv. katastrofickou množinu, která určuje, za jakých podmínek a kdy dojde v daném systému ke katastrofě. Tato ryze matematická teorie pracuje se speciální množinou topologic-

kých objektů, které popisují to, čemu se v hovorovém jazyce říká katastrofa. Lapidárně řečeno, teorie katastrof popisuje prudké změny chování systémů v závislosti na plynulých a malých změnách některých veličin, které na daný systém působí [8 – 15]. S katastrofami se lze setkat každý den. Jako příklad lze uvést roztržení papíru (plynulá změna – tah na papír, prudká změna – roztržení), zlomení pravítka, optické jevy pozorovatelné v každém šálku s tekutinou (tzv. Kaustika), pevnost a stabilita inženýrských konstrukcí atd.



Obr. 9 J. H. Poincare



Obr. 10 R. Thom

Michal Feigenbaum je další osobnost, která přispěla významnou měrou k teorii chaosu. Původem je ze smíšeného manželství, kde otec pocházel z Polska a matka ze SSSR. Oba rodiče emigrovali do USA. M. Feigenbaum měl poměrně pestrý studijní život. Svá vysokoškolská studia začal v New Yorku (elektroinženýrství), pokračoval na MIT v doktorátu opět z elektroinženýrství. Toto zaměření však záhy změnil a začal studovat fyziku se zaměřením na obecnou relativitu. Jeho největší přínos v teorii chaosu spočívá ve faktu, že objevil konstanty (nesoucí jeho jméno), které jsou nezávislé na generujícím systému a ukazují na jistou univerzálnost chaosu. Tyto konstanty jsou velmi dobře patrné z tzv. logistické rovnice, která je často používána k demonstraci koexistence druhů typu predátor-kořist. V říjnu 1975 Feigenbaum zjistil, že tato konstanta je stejná pro širokou třídu systémů, v jejichž chování se objevuje tzv. zdvojení periody. Podle jeho vzpomínek hned volal rodičům a sdělil jim, že našel něco, co by z něj mohlo udělat slavného muže. A také udělalo. V dubnu 1976 Feigenbaum poslal první příspěvek k časopiseckému zveřejnění, ten však byl zamítnut. Během roku 1977 však byl požádán více jak tisíci vědci o jeho kopii, což asi vedlo k tomu, že v roce 1979 byl tento příspěvek konečně zveřejněn. V roce 1986 přijal Feigenbaum místo profesora na Rockefellerově univerzitě. Dopad Feigenbaumova

objevu je fenomenální. Nejen, že ukázal, že chaos je de facto univerzální jev, ale také způsobil posun v „teoretické a experimentální matematice“.

Andrei Nikolajevič Kolmogorov je další z dlouhé řady významných osobností, které přispěly nejen k teorii chaosu. Už v raném dětství byl Kolmogorov bez rodičů (oba tragicky zahynuli) a byl vychován u tety Věry Jakovleny. V roce 1920 Kolmogorov nastoupil do Moskevské státní univerzity, kde nestudoval matematiku (ta se stala „jeho pozdějším osudem“), ale takové obory jako metalurgie, ruskou historii apod. Napsal například i vědeckou práci na téma vlastnictví nemovitostí v Novgorodu v období 15. a 16. století. V oblasti matematiky byl Kolmogorov ovlivněn řadou matematiků (Alexandrov, Lužin, Jedorov, Šušlin ...) ze svého okolí, kteří se shodou okolností s Kolmogorovem více či méně potkali. Dle Kolmogorova však měl z matematické stránky na něj největší vliv jistý Stepanov, který Kolmogorova učil trigonometrické řady. Už jako student dosáhl Kolmogorov pozoruhodných výsledků, když napsal práci o pracích na množinách, jež byla významným zobecněním Šušlinových výsledků (publikoval celkem 8 prací ještě než dokončil studia!!!). Kolmogorov ukončil svá studia na Moskevské státní univerzitě v roce 1925 a začal pracovat ve výzkumné Lužinově skupině. Za významný milník v jeho kariéře je považován přínos v oblasti teorie pravděpodobnosti (1925), kde spolu s Chinčinem publikoval výsledky, jež vedly k vytvoření tzv. stochastického kalkulu. A později k stochastickým diferenciálním rovnicím.



Obr. 11 M. Feigenbaum



Obr. 12 A. N. Kolmogorov

V roce 1929 Kolmogorov úspěšně ukončil doktorát a v roce 1931 přijal místo profesora na Moskevské univerzitě. Jeho přínos v oblasti matematiky včetně aplikací byl velmi široký. Zabýval se problémy jako je např. pohyb planet, turbulence, dynamické systémy apod. V

roce 1953-54 publikoval dva příspěvky z oblasti dynamických systémů, které se staly neoficiálně zakládajícími příspěvky tzv. KAM (Kolmogorov-Arnold-Moser) teorie z oblasti tzv. Hamiltoniánských systémů a samozřejmě i jejich chaotického chování. Kolmogorovův přínos samozřejmě nezůstal neoceněn. Byl členem akademie věd SSSR, oceněn cenou Lenina a Lobačevského. V zahraničí byl členem mnoha vědeckých společností, jako byla rumunská Akademie věd, americká Akademie věd a umění, Matematická společnost v Londýně, nizozemská Akademie věd, indický Institut statistiky apod.

2 HAMILTONIÁNSKÉ SYSTÉMY

Z historických důvodů byly jako první studovány tzv. Hamiltoniánské systémy, jejichž typickými reprezentanty jsou problémy nebeské mechaniky. Jejich studium vedlo k objevení mnoha zákonitostí, které jsou platné pro širokou třídu Hamiltoniánských systémů, jež generují chaos. Dokonce byla zjištěna jejich platnost i pro některé disipativní chaotické systémy. Při zjednodušeném pohledu na věc se dá říci, že systémy generující deterministický chaos se dělí na systémy Hamiltoniánské a disipativní. Studium Hamiltoniánských systémů má své kořeny v devatenáctém století a je úzce svázáno s ergodickou teorií. Svě jméno mají tyto systémy podle skotského matematika Williama Hamiltona.

Typickým rysem pro Hamiltoniánské systémy je skutečnost, že u nich „nedochází“ k disipaci energie. To se musí brát s určitou rezervou, protože každý systém ztrácí časem energii a je tudíž disipativní. Pokud se mluví o faktu, že systém je Hamiltoniánský a tudíž neztrácí energii (je konzervativní), pak je to míněno jen v rámci krátkého časového úseku, během kterého je prováděno pozorování a disipace je tudíž zanedbatelně malá. Na vzniku teorie chaosu těchto systémů se podíleli lidé jako Boltzman (položil základy ergodické teorie a objevil kontradikci mezi vratností systému a nevratností jeho chování) a Poincare. Výhodou Hamiltoniánských systémů bylo to, že je možné je řešit bez pomoci výpočetní techniky, což je v dnešní době u mnoha problémů prakticky nemyslitelné. Výrazem konzervativní se rozumí takový systém, v němž jisté fyzikální veličiny, jako např. mechanická energie apod., zůstávají konstantní. Jestliže pozorovaný systém započne svou aktivitu, pak je množství energie s rostoucím časem konstantní. Jak už bylo naznačeno, tak ideální konzervativní systémy neexistují. Nicméně z jistého časového hlediska lze některé fyzikální systémy považovat za konzervativní. Typickým představitelem je naše sluneční soustava. Pokud by Slunce nezaniklo dle schématu hvězdného vývoje, pak by jistě jednoho krásného dne planety dopadly na jeho povrch. Nicméně z hlediska nás, coby krátkodobě žijících pozorovatelů, lze takový systém považovat za konzervativní. Matematický aparát a tím pádem i filozofie Hamiltoniánských systémů nachází v dnešní době uplatnění v mnoha fyzikálních směrech, jako je fyzika plazmatu, kvantová mechanika apod.

Disipativní systémy se na rozdíl od Hamiltoniánských systémů vyznačují tím, že u nich dochází k disipaci neboli ztrátě energie. Takové dynamické systémy jsou pak popsány soustavami diferenciálních rovnic, které umožňují syntézu jejich řízení. Vlastní výraz „deterministický chaos“ plyne z faktu, že systémy, které jej produkují, lze modelovat a tím, že se

v těchto modelech nevyskytuje náhodnost. Typickým příkladem je například tzv. logistická rovnice (Obr. 16) a Lorenzův atraktor (Obr. 4), který je generován třemi diferenciálními rovnicemi (1). Struktura těchto rovnic je přesně známa, a i přesto tato sestava jednoduchých diferenciálních rovnic generuje chaotické chování bez ohledu na počáteční podmínky.

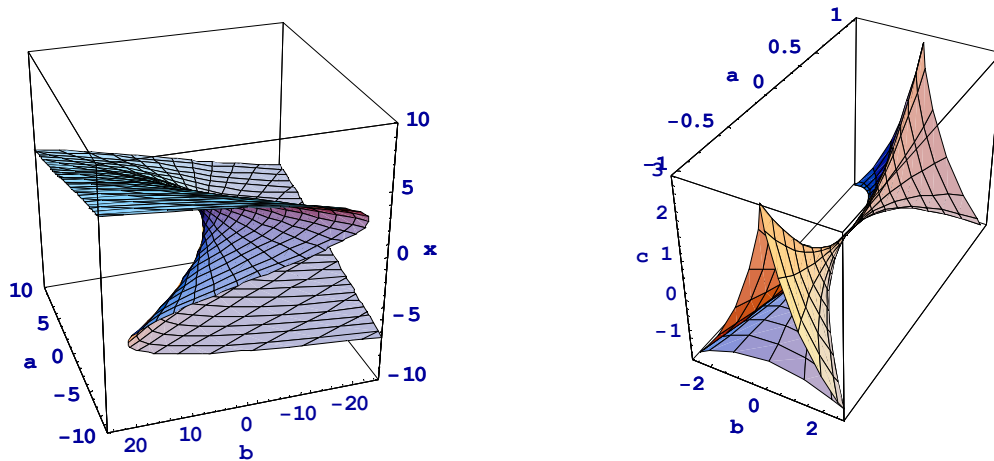
$$\begin{aligned}\dot{x}(t) &= -3(x(t) - y(t)) \\ \dot{y}(t) &= -x(t)z(t) + 26.5x(t) - y(t) \\ \dot{z}(t) &= x(t)y(t) - z(t)\end{aligned}\tag{1}$$

Vznik deterministického chaosu lze pochopit při pohledu na stavový prostor dynamického systému. Jsou-li v něm vhodně uspořádány singulární body typu zřídlo a propad (nor) [9].

Deterministický chaos souvisí rovněž s tzv. teorií katastrof [1 - 3, 5 - 6, 9 - 12], pomocí které lze např. vysvětlit i to, jak při plynulé změně vhodných parametrů systému může ryze deterministický systém přejít přes sérii bifurkací až do chaotického režimu. Vlastní teorie katastrof tedy pracuje s modely reálného světa a hledá tzv. katastrofickou množinu, která pak říká, za jakých podmínek a kdy dojde v daném systému ke katastrofě. V případě deterministického chaosu se katastrofou rozumí tzv. bifurkace (změna počtu a charakteru singulárních bodů ve stavovém prostoru systému), při níž dochází ke kvalitativním změnám ve struktuře stavového prostoru daného systému. Na Obr. 13 – Thomovy katastrofy jsou znázorněny nejjednodušší katastrofy typu záhyb a pyramida. Katastrofa záhyb umožňuje jednoduše demonstrovat ideu teorie katastrof [1]. Každý bod této plochy je určen třemi čísly – parametry „a“, „b“ a hodnotou „x“. Matematik by řekl, že „a“ a „b“ jsou parametry modelu (teplota, tlak, napětí apod.) a „x“ je závisle proměnná (např. stav systému, mechanická pevnost, apod.).

Jestliže se mění hodnoty parametrů (např. $a = -10$ do $+10$ a „b“ se mění od -20 do $+20$), pak stav daného systému lze chápat jako bod, který se pohybuje po ploše v závislosti na parametrech. Při překročení parametru „b“ přes hodnotu 20 při $a < 0$ se stane podivná věc. Pro tuto a větší hodnotu je „x“ definováno mnohem výše, než tomu bylo doposud, což znamená, že systém přejde skokem z jednoho stavu do druhého (charakter a tím i struktura stavového prostoru se skokem změní). Lidově (a kupodivu i matematicky) řečeno, se stane katastrofa.

U složitějších systémů, které mají více řídicích parametrů je interpretace podobná. Na Obr. 13 je katastrofa typu pyramida. Plocha tohoto tvaru je tvořena množinou bodů, které představují katastrofu. Jinými slovy, pokud se v daném dynamickém systému mění tři parametry, pak tzv. parametrická trajektorie v tomto parametrickém prostoru dynamického systému při průchodu plochou indikuje vznik katastrofy, která může znamenat např. právě změnu charakteru singulárního bodu systému – bifurkaci [3].



Obr. 13 Thomovy katastrofy

Vlastní popis principů a podmínek, které vedou ke vzniku deterministického chaosu je značně rozsáhlé téma a proto lze případné zájemce odkázat na [9 - 10].

3 DISIPATIVNÍ SYSTÉMY

3.1 Řízení chaosu – principy a trendy

Deterministický chaos lze pozorovat v mnoha přírodních a technických systémech. Lze jej najít v biologii (predátor - kořist), chemii (Bélousov – Žabotinského reakce), elektronice (viz Obr. 19) apod. Řízením deterministického chaosu se rozumí takové působení na daný chaotický systém, aby se z režimu chaotického dostal do režimu periodického či neperiodického ustáleného stavu. To je základní myšlenka řízení deterministického chaosu. Jsou zde samozřejmě ještě i opačné trendy, tzn. takové řízení, jenž by daný systém přivedlo právě do chaotického režimu. Ačkoliv to zní absurdně, je faktem, že některé děje mají optimální průběh právě při chaotickém chování (např. chemické reakce). Možnost řídit deterministický chaos dává rovněž možnost dopravit systém do jiného stavu s minimálními nároky na energii, pokud se využije principu deterministického chaosu „in natura“.

S myšlenkou řízení deterministického chaosu přišli jako první Ott, Greboki a York (OGY) v publikaci [5]. Principem řízení chaosu je ovlivňovat aktuální stav systému vykazujícího chaos malými perturbacemi za účelem jeho stabilizace do periodického chování či ustáleného stavu. Tato perturbace může ovlivnit chaotické chování jen tehdy, pokud je aktuální stav systému dostatečně „blízko“ režimu, do kterého má být systém převeden. Díky tomu, že daný systém je v chaotickém režimu a díky tzv. ergodicitě [9] navštíví stavová trajektorie dříve či později „dostatečně hustě“ všechna místa stavového prostoru. V takovém případě stačí počkat, dokud se stavová trajektorie nedostane dostatečně blízko stavu, v němž má být systém udržen a po té aktivovat již zmíněnou perturbaci.

Základní princip řízení chaosu snad nejlépe objasní následující příklad [5 – 6], jenž je založen na logistické rovnici (2), [9]. Z bifurkačního diagramu (Obr. 16 c) je vidět, že při různých hodnotách řídicího parametru vykazuje rovnice různé typy chování deterministickým počínaje a chaotickým konče. Chaos nastává přibližně při hodnotě $A > 3.57$ a v případě, že $A = 3.8$, je chování chaotické. Necht' je cílem řízení periodická trajektorie o periodě „ m “, tzn. $x(m+1) = x(1)$ a zároveň existuje možnost jemně ladit parametr A v intervalu

$\langle A-\delta, A+\delta \rangle$. Díky již zmíněným vlastnostem chaotického atraktoru se s pravděpodobností jedna dostane stav systému do blízkosti periodické trajektorie v pozdějším čase [3, 6]. Bez zásahu by samozřejmě stav systému pokračoval na své chaotické pouti stavovým prostro-

rem. Proto je nutno v okamžiku, kdy je systém ve vhodném stavu, provést řídicí zásah. V tomto případě změnit parametr A o ΔA , neboli perturbovat jeho hodnotu. Nutno podotknout, že míra perturbace se musí neustále přepočítávat.

Výpočet ΔA vychází z následujícího postupu. Necht' je cílem řízení trajektorie $x(i)$, $i=1, 2 \dots m$, s periodou „ m “. Pokud se v iteraci (čase) „ n “ dostane stavová trajektorie do blízkosti i -té části cílové trajektorie, pak je linearizovaná dynamika v iteraci $i+1$ dána vztahem (2).

$$\begin{aligned} x_{n+1} - x(i+1) &= \frac{\partial f}{\partial x}(x_n - x(i)) + \frac{\partial f}{\partial A} \Delta A_n \\ &= A_0(1 - 2x(i))(x_n - x(i)) + x(i)(1 - x(i))\Delta A_n \end{aligned} \quad (2)$$

Jestliže je cílem řízení periodický stav, pak musí platit, že levá strana (2) se musí rovnat 0. Elementárními úpravami lze pak obdržet rovnici (3), která dává velikost perturbace řídicího parametru.

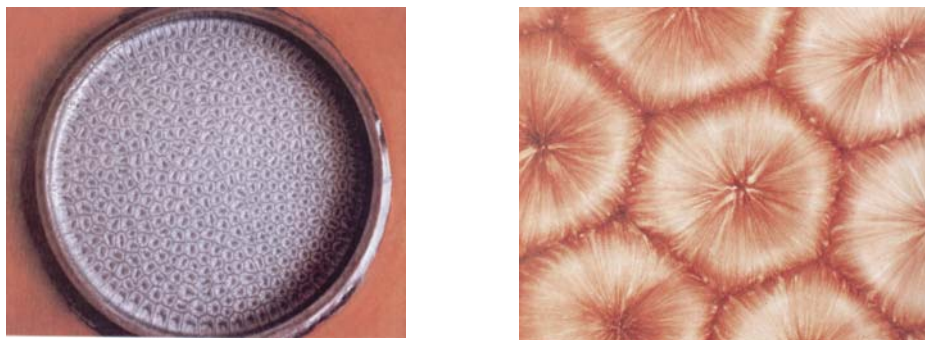
$$\Delta A_n = \frac{A_0(2x(i) - 1)(x_n - x(i))}{x(i)(1 - x(i))} \quad (3)$$

Aplikováním této perturbace se systém dostane do režimu m -periodické trajektorie a zůstane v ní, dokud bude existovat toto řízení. Pokud bude ΔA nastaveno na 0, systém opět přejde do chaotického režimu a bude pokračovat ve své „bludné“ pouti stavovým prostorem. Tohoto faktu lze využít např. také k tomu, že pokud se během řízení ukáže, že momentální řízení již není z nějakého hlediska optimální, lze ΔA nastavit na 0 a nechat bez jakéhokoliv vnějšího působení běžet chaotický systém „in natura“, až do okamžiku, kdy se jeho stav dostane blízko ke stavu, v němž by měla být příslušná hlediska optimality splněna. Pak stačí opět aktivovat perturbaci daného řídicího parametru.

Nutno poznamenat, že čas potřebný k tomu, aby stav systému dospěl do dostatečné blízkosti je relativně dlouhý a souvisí s definováním okolí „ ε “ okolo m -periodické trajektorie. Perturbace „ δ “ je tedy aplikována, pokud je momentálně stav systému v okolí $u < \varepsilon$. Jinými slovy musí platit $x_n - x(i) \ll 1$, což znamená, že ΔA je rovněž velmi malé.

Řízení deterministického chaosu, které zde bylo naznačeno, je zaměřeno na řízení s ohledem na časový vývoj daného systému. V současné době teorie řízení deterministického chaosu nezahrnuje jen tento základní pohled, ale také řízení vícerozměrných systémů [6] nebo systémů s dopravním zpožděním, což jsou v podstatě klasické oblasti řízení. Co je však velmi zajímavé, že se začínají v teorii řízení deterministického chaosu objevovat i studie, které se zaměřují na řízení časoprostorového chaosu. Systémy, které vykazují časoprostorový chaos jsou obvykle systémy, které lze popsat parciálními diferenciálními rovnicemi a jejich chování je mnohem složitější. Chování těchto systémů už lze mnohdy definovat jako samoorganizaci, což je další speciální třída chování, při níž se v chování systému objevuje nová kvalita (Obr. 14, obvykle geometrické uspořádání apod.), která nebyla systému importována.

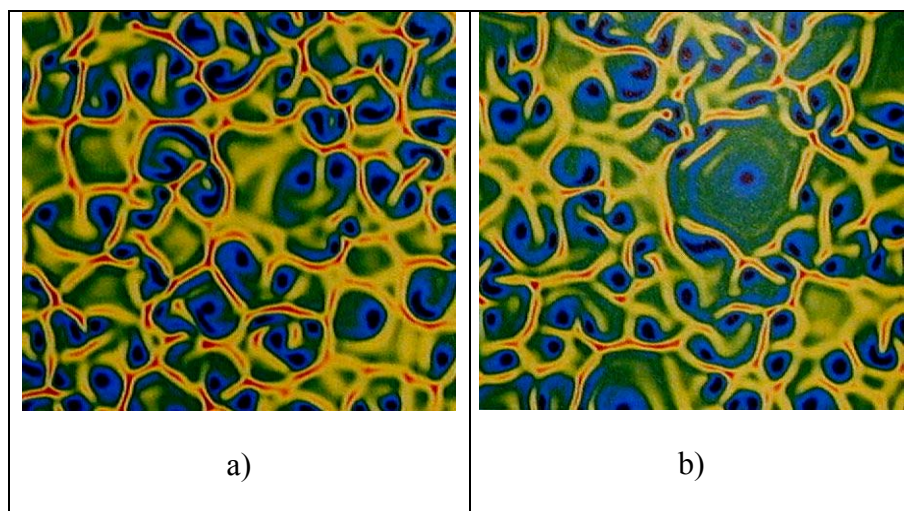
Samoorganizace [4, 15 – 18] je jev, který je intenzivně studován teprve až v minulém století. Je velmi dobře propracován po teoretické stránce v chemii, kde má na „svědomí“ například to, že se při vhodných podmínkách začnou milióny molekul chovat jako inteligentní jedinci a např. v Bělousov – Žabotinského reakci začnou cyklicky vytvářet okem viditelné barevné vzory v tekutině s překrásným geometrickým motivem. Samoorganizace lze pochopit díky entropii, která vyjadřuje míru neuspořádanosti daného systému. Čím více je daný systém neuspořádaný, tím větší je entropie a naopak. Samoorganizace pak vzniká v systému, který se skládá z velikého množství samostatných jednotek (molekuly, lidé...), které tvoří menší systémy, mezi nimiž dochází k „tokům“ entropie. Jinými slovy, vzniká mezi částmi tvořícími celek, které si navzájem snaží ovlivnit své entropie. Dochází tak ke vzniku tzv. disipativních struktur (struktur, kde dochází k tokům energie „z“ a „do“ okolí systému), které mohou za určitých podmínek vykazovat chaotické chování [15].

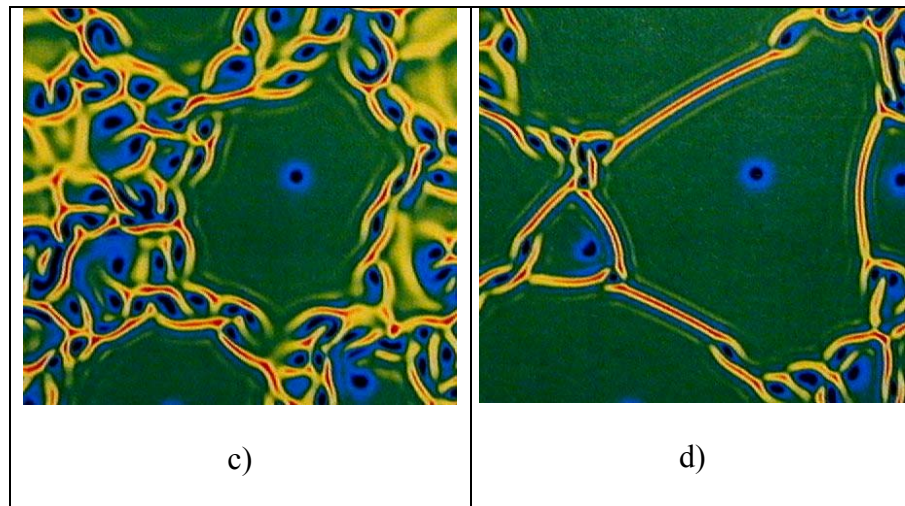


Obr. 14 Samoorganizace – samovolně se tvořící šestihrany v zahříváné kapalině, celkový pohled (vlevo) a detail (vpravo)

Pro řízení systémů generujících časoprostorový chaos existují metody, které lze chápat jako „rozšíření“ základní myšlenky o řízení chaosu. Samotné principy OGY algoritmu jsou vhodné pro řízení chaotických systémů s nízkou dimenzí (obvykle méně jak 5), pro systémy s vyšší dimenzí je obvykle nevhodný. Pro příklad takovýchto systémů se nemusí chodit daleko. Lze je nalézt v oblasti chemie (chemické oscilace), či problematice laseru, supravodivosti či hydrodynamice. Řízení časoprostorového chaosu je vzhledem ke své podstatě značně významný úkol, vezmou-li se v úvahu takové jevy jako je turbulence apod. Tento typ chování je nazýván jako „časoprostorový chaos“ na zdůraznění odlišnosti vzhledem k nízkodimenzionálnímu časovému chaosu, jenž je reprezentován dnes již klasickými příklady, jako je logistická rovnice či Lorenzův atraktor. Při jeho řízení, vzhledem k velkému počtu stupňů volnosti musí být příslušné řízení vhodně distribuováno. Jeho princip je založen na aplikaci řídicího zásahu v jednom místě a ten je pak díky fyzikálním vlastnostem systému „distribuován“ do ostatních míst.

Ukázalo se však, že tento způsob řízení není vhodný pro jinou třídu systémů jako jsou chemické reakce, optické systémy, biologické systémy typu „srdce“ apod. Proto byla vypracována metoda využívající topologických defektů k řízení daného systému. Jako ukázkou řízení tohoto typu lze uvést řízení časoprostorového chaos, který lze modelovat komplexní Ginzburg-Landauovou rovnicí. Na Obr. 15 a-d) je postupně vidět jak samotný časoprostorový chaos, tak i jeho postupné ustálení.





Obr. 15 Řízení časoprostorového chaosu

3.2 Nejjednodušší model deterministického chaosu - logistická rovnice

Jeden z velmi jednoduchých modelů chaotického chování je tzv. logistická rovnice (4). Tato rovnice [13] vznikla jako odezva na potřebu simulace biologických systémů, v jejichž rámci popisuje chování druhu v jeho přirozeném prostředí. Tento popis byl založen na existenci nějakého druhu v "uzavřeném" prostředí (většina druhů má omezené migrační schopnosti), které mu poskytovalo obživu. Nejjednodušším příkladem jsou např. kapři a štiky v rybníku. Pokud se do rybníku nasadí určité množství kaprů a štik, pak samozřejmě dojde k jejich množení a růstu. Ten bude mít nároky na kapacitu potravy (kaprů) v rybníku. Při rostoucím počtu jedinců bude klesat množství potravy v rybníku, což se projeví zpomalením růstu počtu kaprů. Od jisté hranice pro nedostatek potravy začnou štiky vymírat hlady, zatímco díky jejich poklesu se množství kaprů začne zvyšovat. To od určité hranice množství kaprů způsobí jejich opětovný nárůst. Prostým citem tedy lze očekávat, že populace bude asi "periodicky" oscilovat, nebo se ustálí na nějaké hodnotě. Jak již jednoduché simulace ukázaly, může systém popsáný touto rovnicí vykazovat velmi komplikované chování od ustáleného přes periodické až po chaotické.

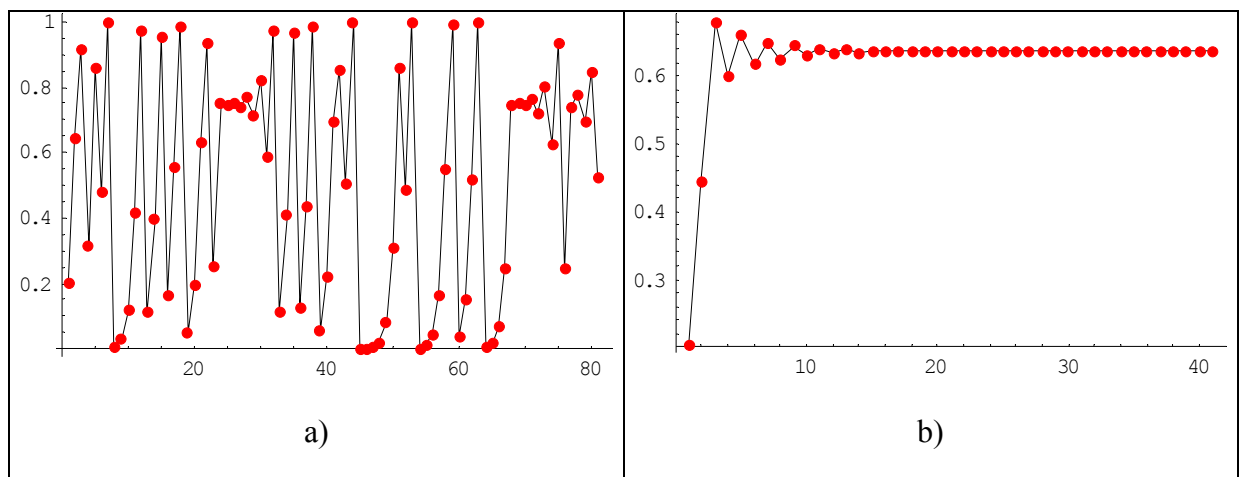
$$x_{n+1} = Ax_n(1 - x_n) \quad (4)$$

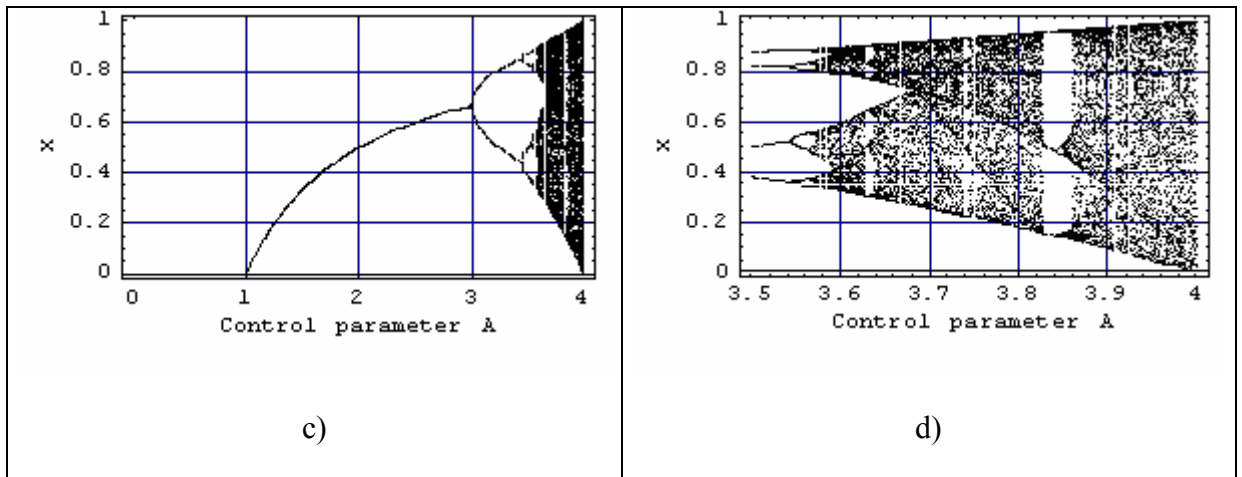
Toto chování se někdy znázorňuje pomocí tzv. bifurkačního diagramu (Obr. 16, Obr. 18), což je graf, který ukazuje závislost chování modelu systému na řídicím parametru. Bifurkační diagram lze "číst" tak, že plná křivka představuje ustálený stav daného systému na různých hodnotách řídicího parametru (Obr. 16, $A = 2.75$). To je dobře vidět na Obr. 16 c)

v intervalu $\langle 1, 3 \rangle$, kde každý bod křivky představuje ustálený stav systému s různou hodnotou. Soubor těchto hodnot pak v závislosti na parametru A vytváří onu hladkou křivku. Při dalším nárůstu však dochází k zajímavému jevu, a to je tzv. zdvojení periody ($x_{\text{start}} = 0.54278$, $A = 3.13$), což znamená, že se stav populace vrací do původního stavu až po dalším stavu (0.5, 0.3, 0.5, 0.3,...). Při dalším nárůstu pak dochází dalšímu zdvojení již zdvojené periody ($x_{\text{start}} = 0.381$, $A = 3.51$) až vývoj přejde do chaotického průběhu. Zajímavostí je, že daný průběh vykazuje fraktální charakter (Feigenbaumova konstanta) - daná zdvojení se „de facto“ v přesném měřítku opakují. A to i u jiných rovnic, než je tato.

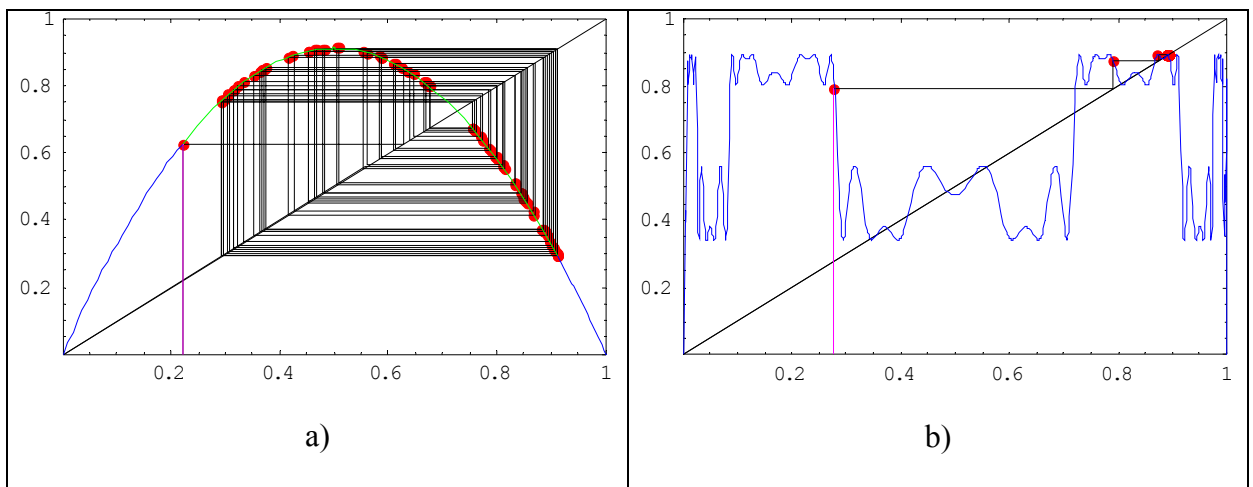
Z Obr. 16 c) a d) je rovněž vidět, že v bifurkačních diagramech jsou oblasti, které ukazují, kdy se systém chová ustáleně $\langle 1, 3 \rangle$, kdy dochází ke zdvojení periody $\langle 3, 3.5 \rangle$, ale také, kdy o ustáleném stavu či periodě nemůže být ani řeči. To je dobře vidět např. v místech (Obr. 16 d) cca $A = 3.7, 3.8$, či 3.9 a více. Toto „zrnění“, můžeme-li to tak nazvat, je deterministický chaos, který se v časovém rozvoji projeví jako křivka, která se nikdy přesně neopakuje.

Dalším způsobem, jak znázornit chování takového systému je tzv. WEB diagram (Obr. 17) [9, 14]. V tomto grafu představuje přímka jdoucí z počátku množinu tzv. pevných bodů, které mají charakter odpuzování či přitahování. Pokud by takový např. přitahující bod byl dosazen do logistické rovnice, pak by funkční hodnota byla stejná jako argument. Vlastní křivka (Obr. 17 a) parabola, b) křivka připomínající hradní cimbuří) znázorňuje samotnou logistickou rovnici s tím, že směrnice jejího průniku s přímkou (množinou pevných bodů) udává charakter příslušného bodu, ale také i chování systému - rovnice v jeho okolí.

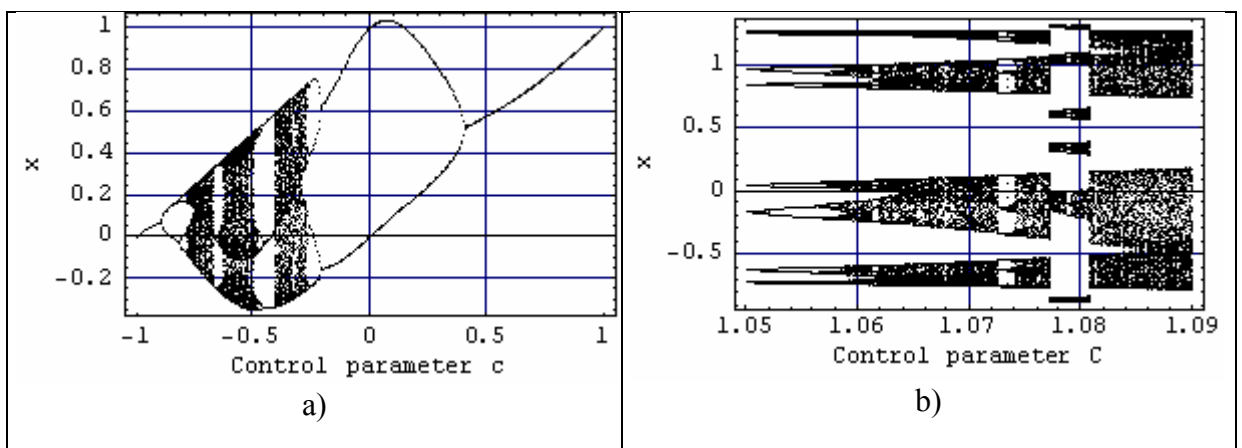


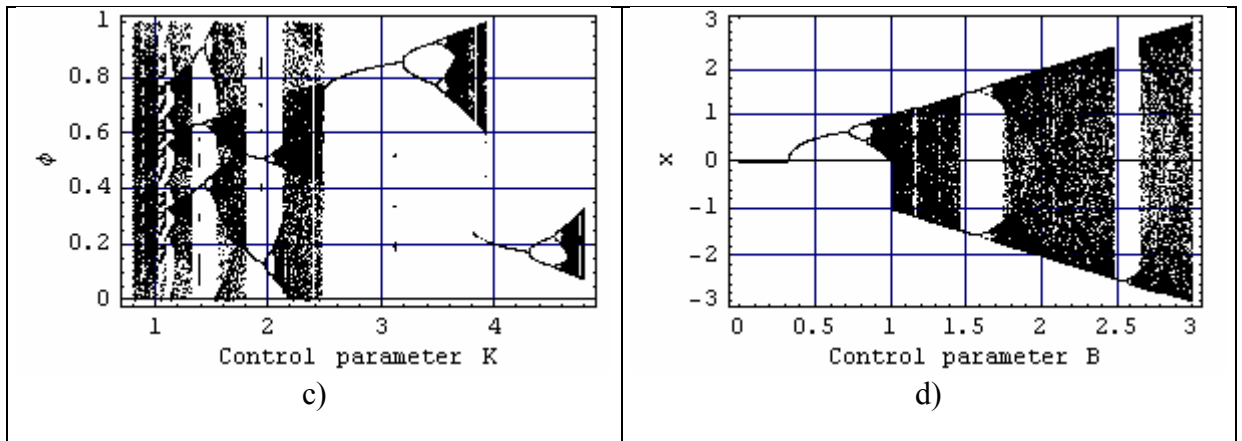


Obr. 16 Logistická rovnice; a) chaotické chování rovnice při vhodné hodnotě řídicího parametru, b) deterministické chování téže rovnice pro jiné nastavení, c) bifurkační diagram, d) detail. Všimněte si v části a) v kroku (osa x) 25 – 31 a 70 – 75 tvaru, který křivka vytváří. Jde o tzv. „trojúhelník“, jednu z mnoha geometrických formací (též technické ukazatele), které jsou zcela běžně využívány jako ukazatel blízkého budoucího chování pro obchodování na burzách.



Obr. 17 WEB diagram



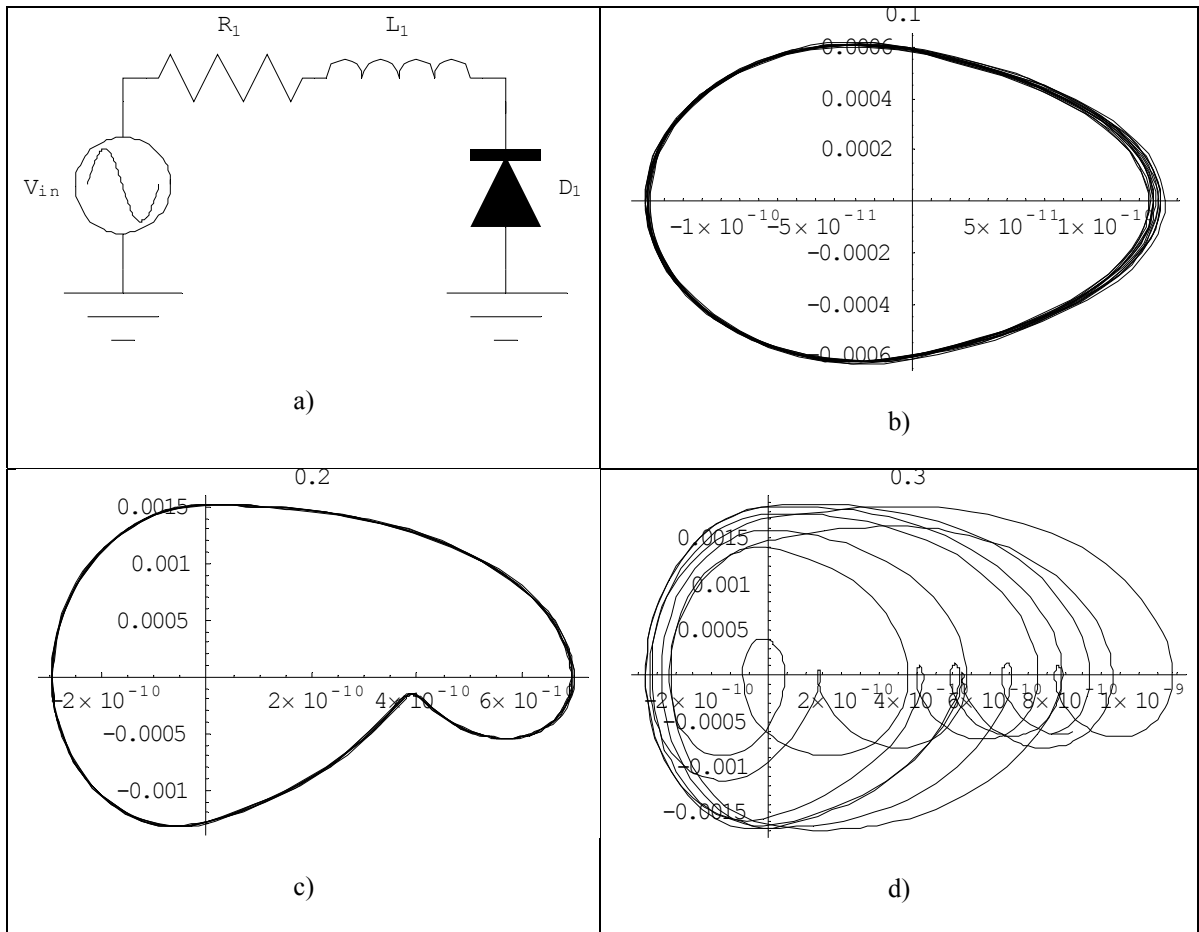


Obr. 18 Různé bifurkační diagramy

Z výše uvedených bifurkačních diagramů plyne rovněž ještě další zajímavý fakt. Systém, který přešel do chaotického režimu, v něm nemusí zůstat navždy a to i navzdory rostoucímu řídicímu parametru, díky jehož nárůstu se do něj dostal. To je nejlépe patrné z Obr. 18 a), na němž je vidět, že při nárůstu řídicího parametru daný systém prošel z ustáleného režimu přes zdvojování period až do chaotických režimů a nakonec se dostal do oblasti stabilních stavů, kde zaujímal různé ustálené hodnoty. Také je vidět, že v oblasti, kde „pánuje chaos“, jsou oblasti klidu, v nichž vymizí jakákoliv stopa po chaosu a existuje zde pouze periodický či ustálený stav (Obr. 16 a) -0.4, c) 2), (Obr. 16 c) 3 a 4, d) 2.5). Pokud tedy vykazuje daný systém režim chaotického chování, pak je šance, že změnou řídicího parametru přejde systém opět do stabilního režimu.

3.3 Deterministický chaos – výskyt

S deterministickým chaosem se lze setkat téměř na každém kroku. Typickým příkladem je počasí. Jeho základním typickým modelem je soustava tří diferenciálních rovnic (1), která generuje známý Lorenzův atraktor (Obr.4). Velmi často se s chaosem lze setkat v elektronických obvodech [13, 15], kde ke vzniku chaosu stačí v podstatě jen pár součástek. Jeden z velmi jednoduchých chaotických obvodů je zobrazen na Obr. 19. V části a) je vlastní schéma a v části b)-d) je změna chování daného obvodu v závislosti na změnách řídicího napětí.



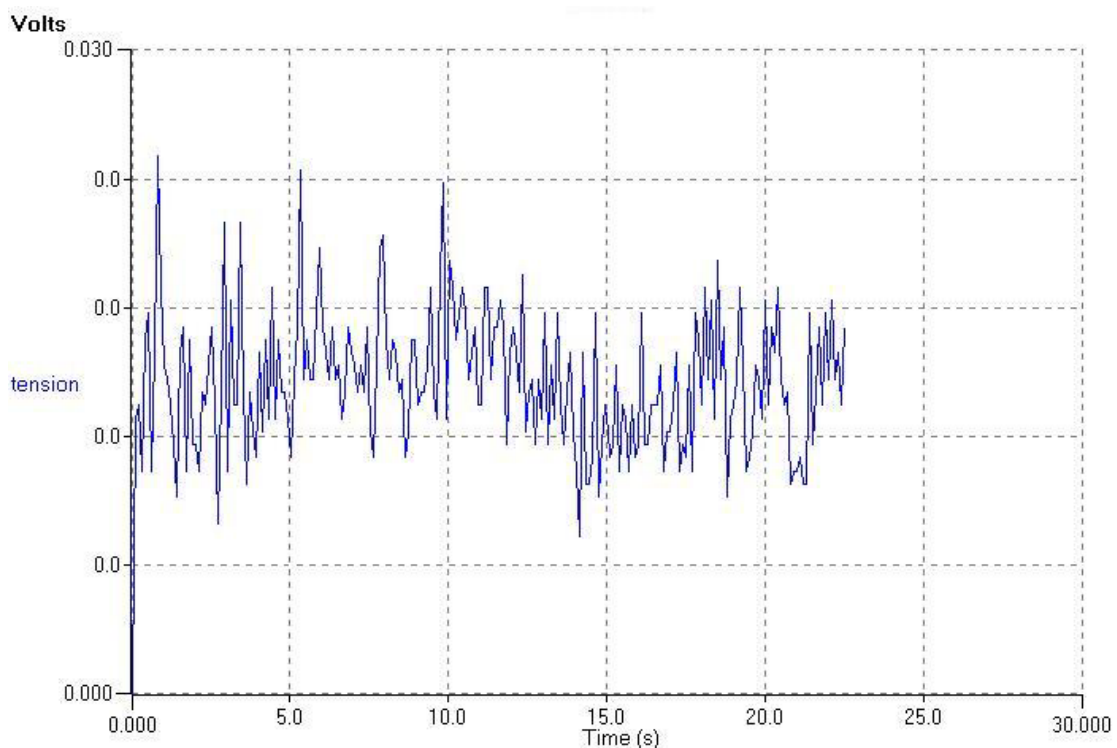
Obr. 19 Jednoduchý elektronický obvod generující chaos

Ad absurdum vzato, lze výskyt chaosu očekávat a pozorovat u systémů, které obsahují vhodnou nelinearitu, nebo pokud mezi spolupracujícími systémy existuje nelineární vazba [6]. Samotný výskyt chaosu ještě nemusí znamenat, že se v daném systému děje něco špatného. Mnohdy to pouze znamená, že je příslušný systém „na cestě“ ke kvalitnějšímu uspořádání .

PRAKTICKÁ ČÁST

4 ROZHRANÍ REÁLNÉHO MODELU A SOFTWARE MATHEMATICA

Cílem praktické části diplomové práce bylo vyvinutí softwarového propojení mezi prostředím Mathematica a reálným modelem soustavy spřažených elektromotorů pod označením CE108, který je určen pro výuku mnohorozměrových řídicích systémů. K úvahám, že by tento reálný model mohl generovat chaotické chování vedl graf (Obr. 20) z manuálu [19] pro řídicí software pro tento model.

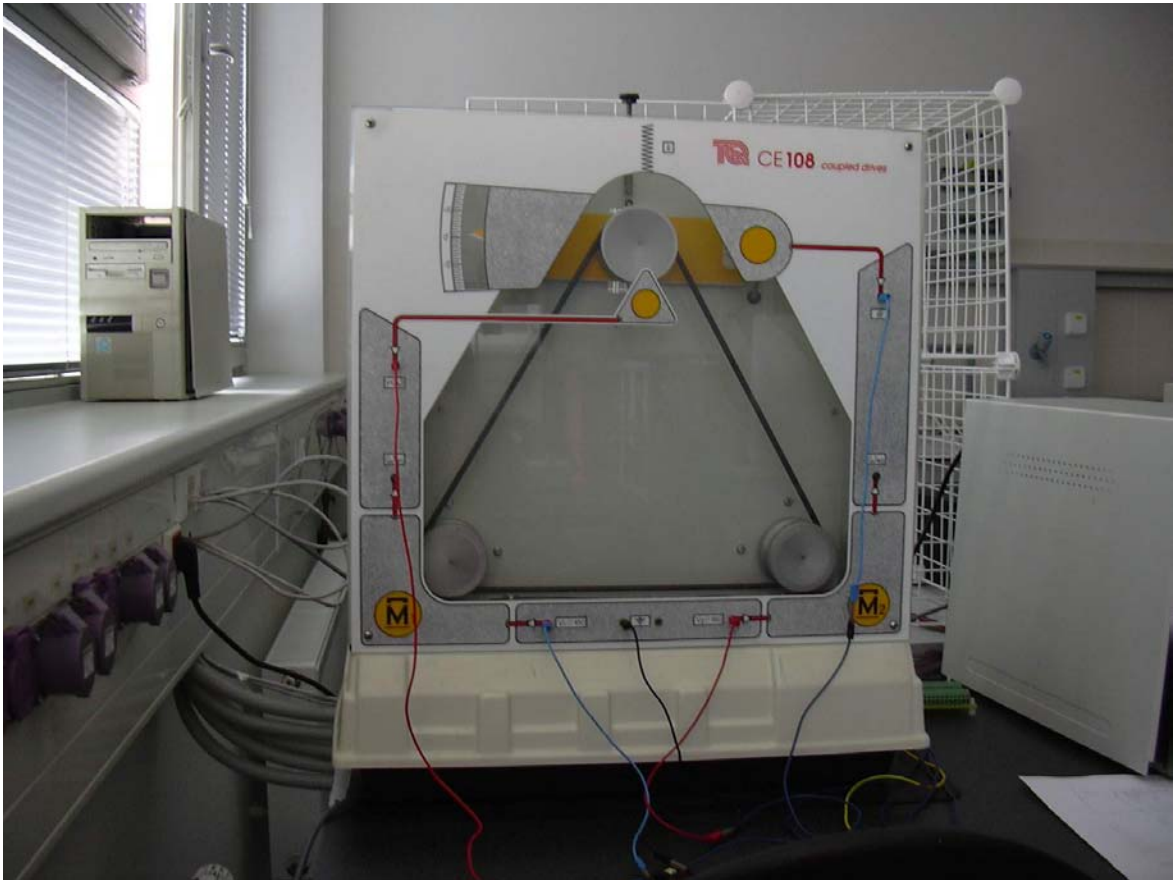


Obr. 20 Ukázka napnutí pásu

4.1 Popis reálné soustavy

CE108 soustava (Obr. 21) spřažených elektromotorů je určena pro výuku mnohorozměrových řídicích systémů. Přístroj byl vyvíjen ve spolupráci s Centrem řídicích systémů Manchesterké university, Institutem pro vědu a technologii (UMIST). Přístroj umožňuje řešit praktické úlohy napínání a rychlosti materiálu u spojitých procesů. Příkladem může být rychlost a napínání nitě z jedné cívky na druhou, které je třeba řídit. Další příklady lze najít například při výrobě papíru, výrobě kabelů a všude tam, kde dochází k napínání spojitého pásu materiálu. Při těchto procesech prochází materiál pracovní stanicí, kde se měří rych-

lost a napínání, které jsou na sobě závislé a tyto veličiny se upravují rychlostí motorů umístěných před a za měřicí stanicí.

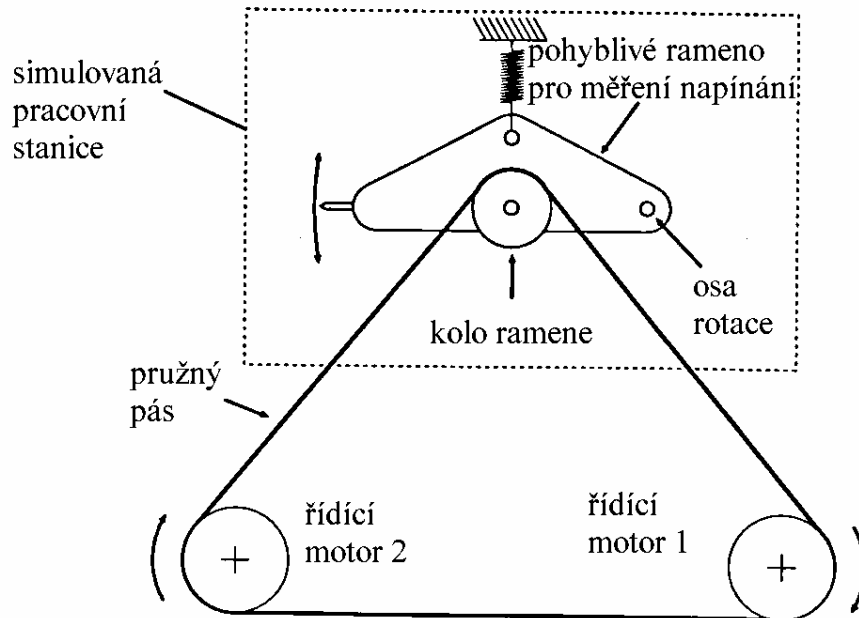


Obr. 21 Fotka soustavy CE108

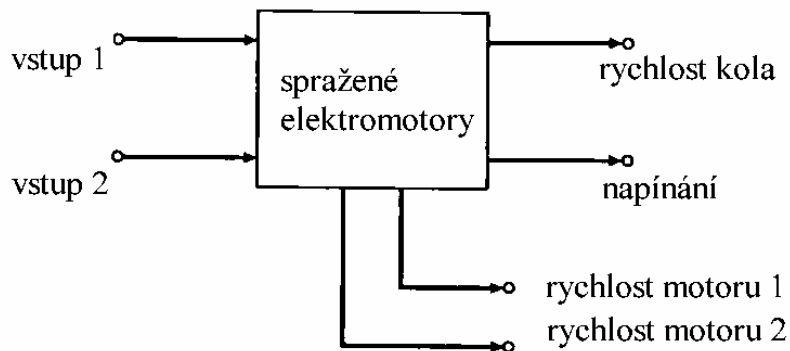
Tato situace je upravena pro laboratorní měření tak, že pružný pás je upevněn na 3 kolech. Rychlost dvou kol je přímo úměrná otáčkám servomotorů. Tato kola jsou napevno umístěná na servomotorech. Třetí kolo se může pohybovat (umístěno na pohyblivém rameni zavěšeném na pružině) a simuluje tak pracovní stanici spolu s měřením napínání a rychlosti. Servomotory řídí napínání pásu a rychlost pohybu, která odpovídá 0 až 10 V, respektive 0 až 3000 ot/min.

Měření napínání je nepřímé přes úhel pohyblivého ramene, ± 10 V respektive 10° . Je-li umístěn pojistný šroub na pohyblivém rameni, pak je minimalizováno napínání. Vstupy do přístroje jsou řídicí napětí k zesilovačům u servomotorů, které se realizují přes 2 mm zdířky (0 ... 10 V) umístěné na předním panelu CE108. Výstupy ze systému jsou také přes 2 mm zdířky umístěné na předním panelu. Výstupy jsou rychlosti servomotorů (ω_1, ω_2), výstup snímače napínání (θ) a rychlost kola na pohyblivém rameni (ω_3). Servozesilovače

motorů jsou obousměrné a dovolují tak řídit každý motor v obou směrech. Schématické znázornění reálného modelu je na obrázcích 22 a 23.



Obr. 22 Princip činnosti modelu CE108



Obr. 23 Vstupy a výstupy modelu CE108

4.1.1 Části soustavy

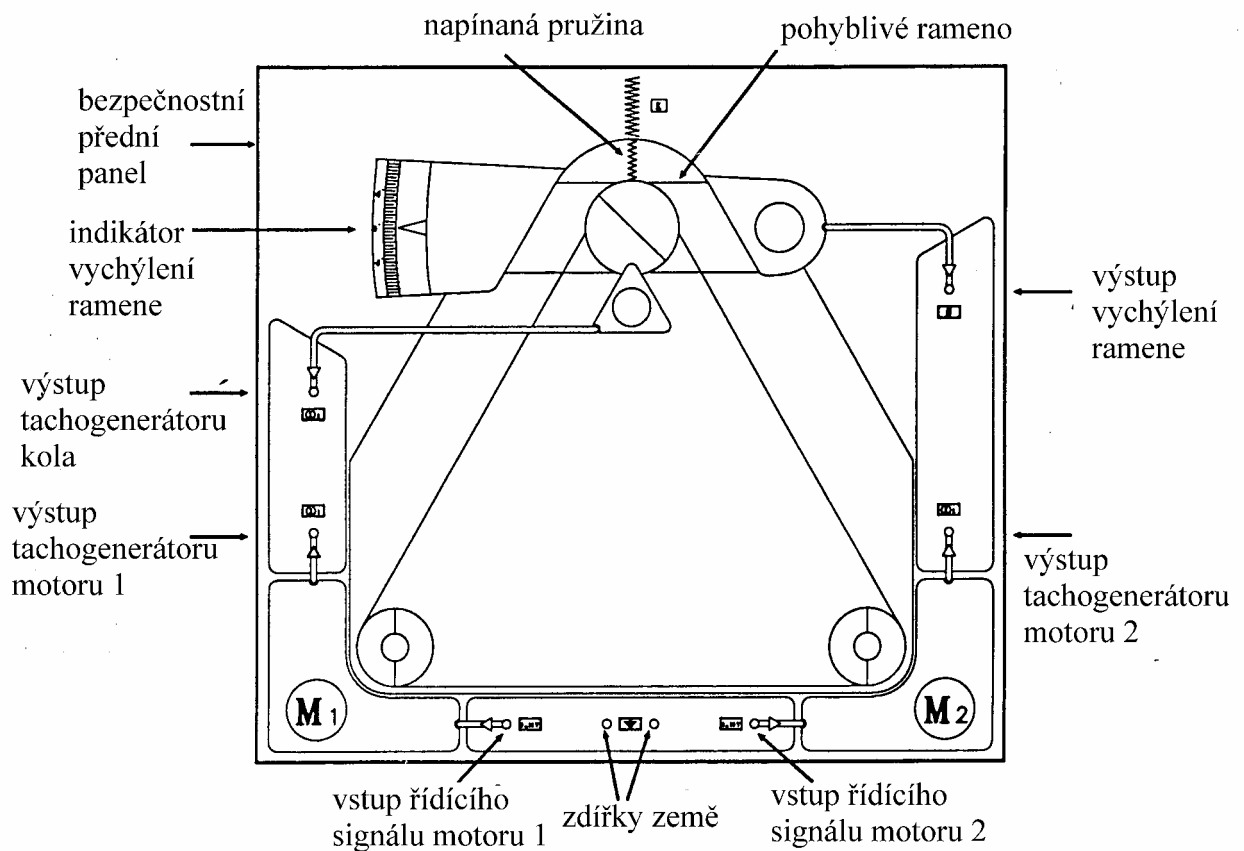
Hlavní funkční části CE108 lze rozdělit na: dvojici stejných servomotorů, tachogenerátor a přesný servopotenciometr. Pro tuto diplomovou práci nebylo využito výstupů pro měření rychlosti motorů.

Servomotory – stejné servomotory jsou řízeny pomocí zesilovačů napájení umístěných v hlavní části přístroje. Motory mají malou setrvačnost a dobrou charakteristiku kroutícího momentu. Motory generují kroutící moment (Γ_1, Γ_2), který je přímo úměrný napětí (U_1, U_2). Na Obr. 24 je napětí označeno jako vstup řídicího signálu motoru

Servomotory jsou spojeny s tachogenerátory, které vytváří elektrický signál V_{ω_1} a V_{ω_2} , který je přímo úměrný úhlovým rychlostem ω_1, ω_2 kol 1 a 2.

Tachogenerátor – tachogenerátor počítající pulsy s výstupem 0 až 10 V pro rychlost 0 až 3000 ot/min je připojen ke kolu na pohyblivém rameni. Zajišťuje výstupní napětí y_1 , které je úměrné rychlosti kola, na Obr. 24 jako výstup tachogenerátoru kola.

Přesný servopotenciometr – tento potenciometr je umístěn axiálně k ose rotace ramene. S vychýlením ramene potenciometr dává odpovídající výstupní napětí y_2 , které je zesíleno zesilovači v hlavní části přístroje. Napětí y_2 lze vztáhnout ke změně úhlu θ ramene použitím označení na krytu přední části CE108, na Obr. 24 jako výstup vychýlení ramene.



Obr. 24 Popis prvku modelu

4.1.2 Model

Přístroj je ovládán napětími u_1, u_2 , které vstupují do zesilovačů motorů a výstupy systému ω a x_k jsou získány na základě výstupu tachogenerátoru y_1 a snímače napínání y_2 . Pro tyto signály platí:

$$\Gamma_1 = g_a u_1,$$

$$\Gamma_2 = g_b u_2,$$

$$y_1 = g_1 \omega,$$

$$y_2 = g_2 x_k,$$

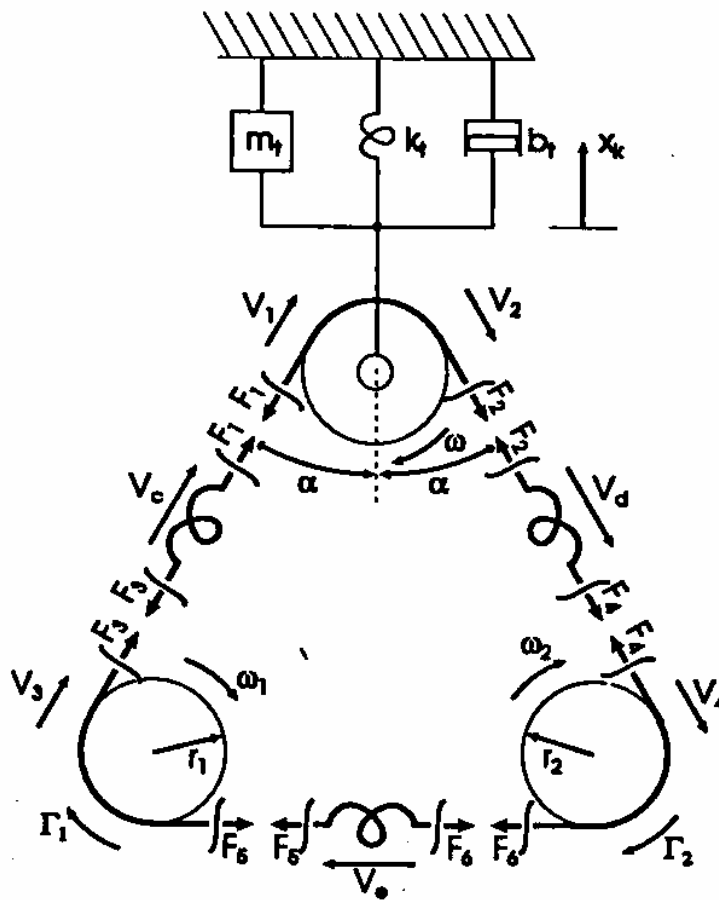
takže platí následující Obr. 25. Konstanty g_a a g_b jsou získány z charakteristik motoru a zesilovače a g_1 a g_2 jsou funkcí konstrukce tachogenerátoru a snímače napínání.

Jestliže použijeme stejný signál u_1 na oba motory, lze předpokládat, pokud jsou oba motory stejné, že rychlost $\omega(s)$ poroste, zatímco napínání bude nezměněné. Analogická formulace platí pro signál u_2 . Potom obdržíme rovnice pro přenosové funkce [19].

$$\omega(s) = \frac{g_m u_1(s)}{Is + b} + \frac{g_m (2 \cos \alpha)^2 k s u_2(s)}{(Is^2 + bs + 2k^1(s)) (2(s^2 m_t + sb_t + k_t) - (2 \cos \alpha)^2 k)} \quad (5)$$

$$x_k(s) = \frac{g^4 m r k \cos \alpha u_2(s)}{(Is^2 + bs + 2k^1(s)) (2(s^2 m_t + sb_t + k_t) - (2 \cos \alpha)^2 k)} \quad (6)$$

kde charakteristiky motoru a zesilovače se předpokládají stejné, tedy $g_a = g_b = g_m$.



Obr. 25. Schéma motorů - převzato z [19]

4.2 Propojení reálného modelu s PC

4.2.1 Popis technologické karty

Propojení reálného modelu s počítačem je realizováno pomocí karty Advantech PCL-812PG (Obr. 26). Je to multifunkční analogově digitální karta, která umožňuje pět nejčastěji žádaných měřících a řídicích funkcí: A/D převodník, D/A převodník, digitální vstup, digitální výstup a časovač.



Obr. 26 Karta Advantech PCL-812PG

Základní specifikace:*Analogový vstup:*

Počet kanálů	16
Rozlišení	12bit
Max. vzorkovací kmitočet	30kS/s
Vstupní rozsah	$\pm 10V, \pm 5V, \pm 2.5V, \pm 1.25V$ (softwarově nastavitelný)
Přesnost	0,01%

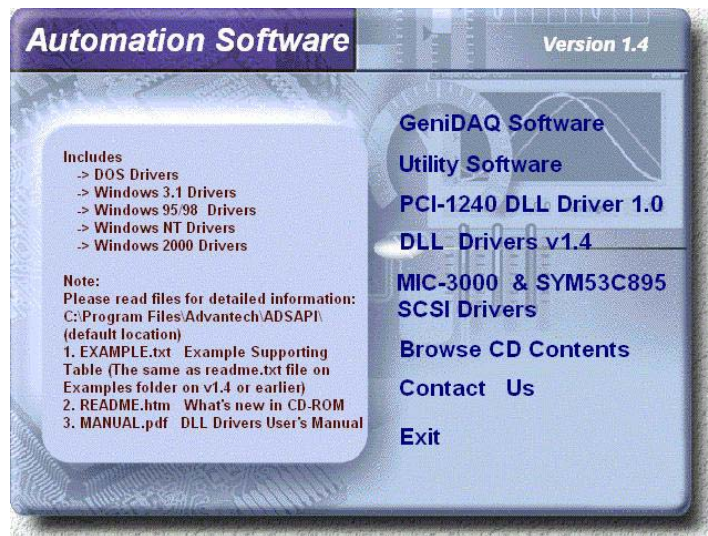
Analogový výstup:

Počet kanálů	2
Rozlišení	12bit
Výstupní rozsah	$\pm 5V, \pm 10V$ (softwarově nastavitelný)
Výstupní proud	10mA max

Jak bylo popsáno v kapitole popis reálného modelu, servozesilovače jsou obousměrné, ale díky výstupu technologické karty 0 - 10V, není v tomto případě možná regulace motoru opačným směrem, jelikož karta neumí pracovat na výstupu se záporným napětím.

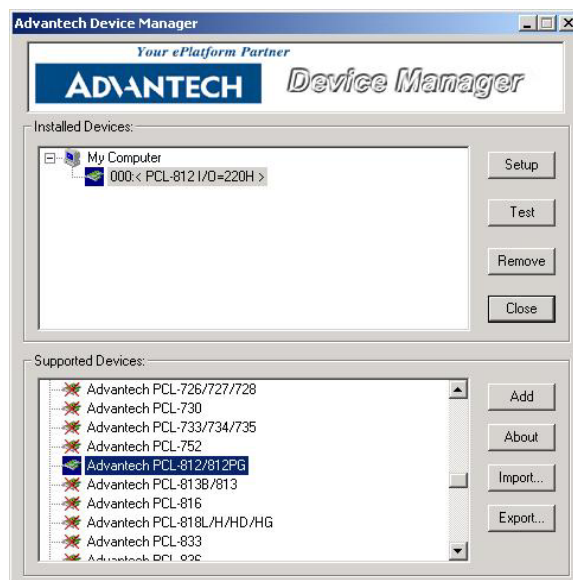
4.2.2 Instalace technologické karty

Pro používání je nutné nainstalovat výrobcem dodaný ovladač. Obr. 27 ukazuje instalační okno ovladače karty.



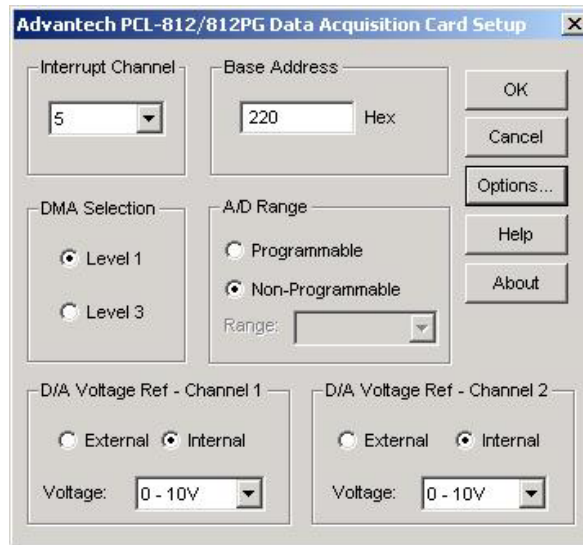
Obr. 27 Instalační okno ovladače

Po instalaci je nutné nastavit správný typ karty v Advantech device manageru - Advantech PCL-812PG (Obr. 28).



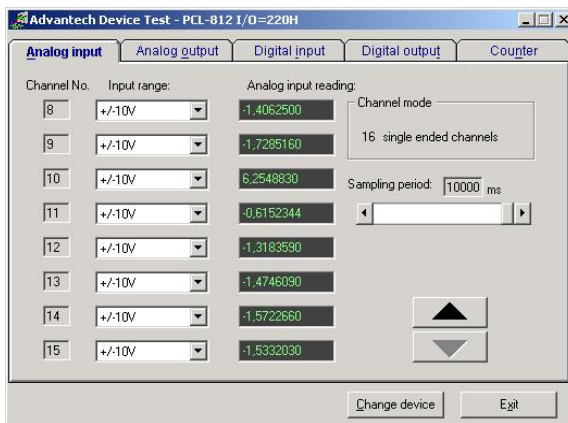
Obr. 28 Okno Advantech device manageru

Po výběru správné karty je třeba zkontrolovat nastavení v položce setup v device manageru, jestli výstupní rozsah napětí je na hodnotě 0-10V u obou kanálů (Obr. 29).

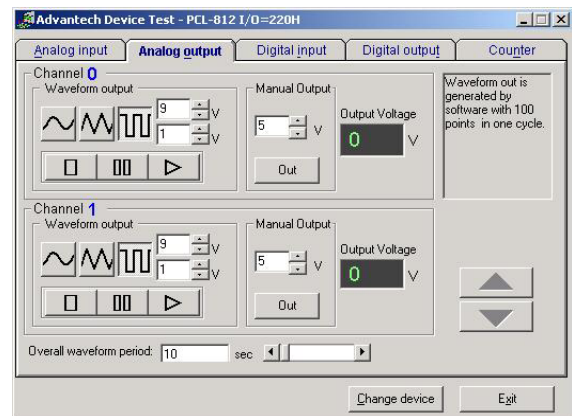


Obr. 29 Nastavení rozsahu kanálů

Pro otestování správného nastavení karty a její funkčnosti slouží tlačítko test (Obr. 28) v device manageru. Po jeho stisknutí se objeví okno na Obr. 30. Existují zde záložky analogického vstupu (Obr. 30) a výstupu (Obr. 31), digitálního vstupu a výstupu a počítadlo. Všechny záložky umožňují testování karty.



Obr. 30 Testovací okno vstupních kanálů



Obr. 31 Okno pro test výstupů

5 REALIZACE SOFTWAREVÉHO ROZHRAŇÍ

Použité vývojové prostředky:

Microsoft Visual C++ 6.0

Mathematica 4.0, 5.2

5.1 Popis ovladače karty Advantech PCL-812PG

Pro ovládání technologické karty v jazyce C jsem použil a dále modifikoval vzorové příklady od výrobce technologické karty firmy Advantech dodané na instalačním cd s ovladači pro danou kartu. Jednalo se o části DAsoft.c určené pro ovládání výstupů karty a ADsoft.c pro ovládání vstupů karty. Tyto ukázky zdrojových kódů jsou automaticky vytvořeny při instalaci ovladačů pro technologickou kartu v cestě C:\Program Files\ADVANTECH\ADSAPI\Examples\Console\

Pro vytvoření spustitelných programů z ukázkových příkladů je nutné do projektu přidat knihovnu ADSAPI32.DLL, která je umístěna v adresáři C:\Program Files\ADVANTECH\ADSAPI\Lib\

5.2 Popis rozhraní C++ a Mathematica (mathlink)

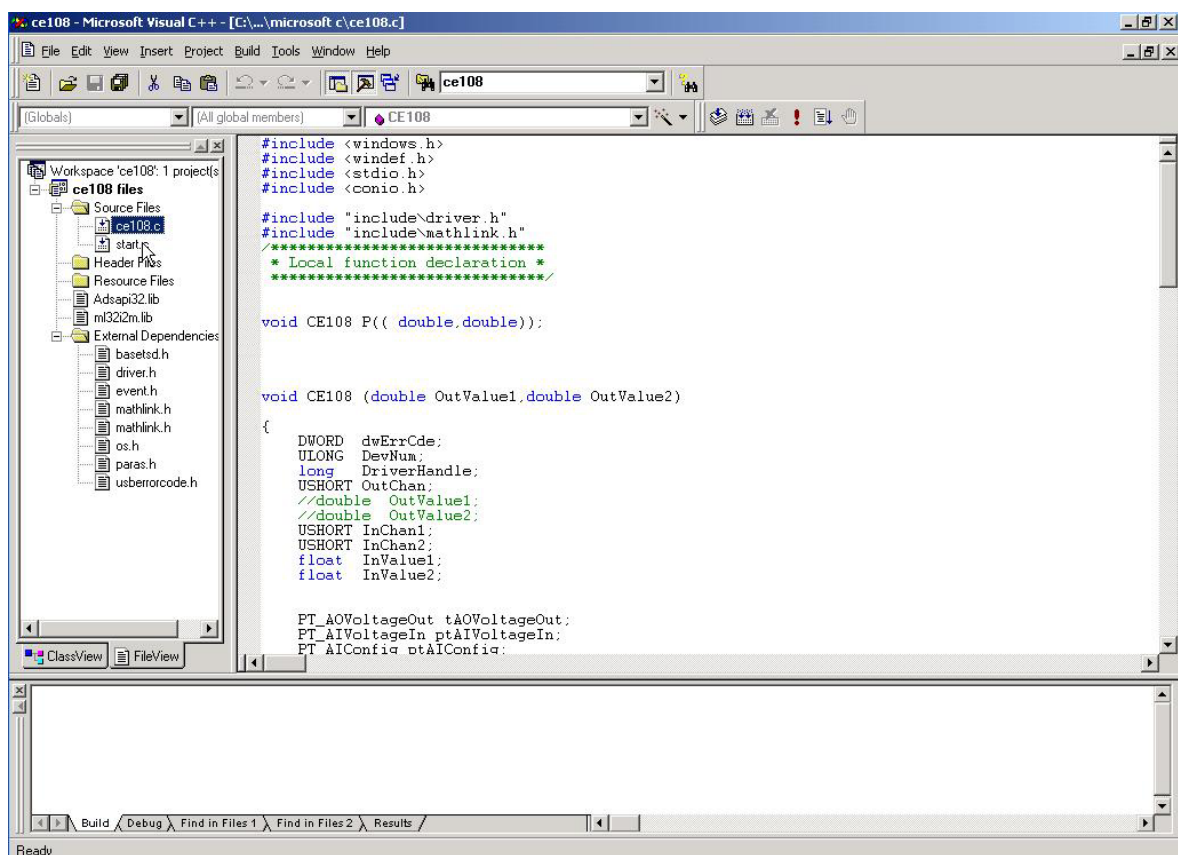
K vytváření programů v jazyce C pro Mathematicu je nutné použít mathlink preprocesor Mprep.exe dodávaný firmou Wolfram Research. Mprep.exe vytvoří ze souboru s příponou .tm zdrojový kód v jazyce C pro překladač. Preprocesor se používá přes příkazový řádek následujícím příkazem:

```
C:\.....\MPREP start.tm -o start.c
```

Tečky stojí místo cesty, kde je uložen soubor Start.tm. Výpis toho souboru je v příloze PI, kde je důležitá tučně tištěná část. Popisuje název funkce, v tomto případě CE108. Způsob, jakým se bude funkce volat z prostředí Mathematica, obsluhuje řádek Pattern. Řádek Arguments popisuje, jaké má funkce vstupní proměnné a jejich typy lze najít na řádce Argumentstype. Soubor start.tm také určuje, jaký typ proměnné se bude vracet do Mathematicy. Hodnota Manual říká, že se typ definuje přímo ve funkci mathlinku, která bude proměnnou vracet.

5.3 Popis programu CE108

Projekt finálního programu CE108.exe (Obr. 32) je složen z hlavní části programu CE108.C, ve kterém je tělo programu, výše uvedené knihovny pro ovládání technologické karty ADSAPI32.DLL, na kterou se odkazuje hlavičkový soubor driver.h. Dále byla přidána knihovna potřebná pro funkci rozhraní mathlink ML32I2M.LIB, na níž odkazuje hlavičkový soubor mathlink.h, a soubor START.C, který byl generován pomocí preprocesoru mprep. Pro úspěšné vytvoření je třeba mít v adresáři include soubory, na které odkazuje hlavičkový soubor driver.h. Jsou to tyto: OS.H, EVENT.H, PARAS.H a USBErrorCode.h.



Obr. 32 Okno s projektem v Microsoft Visual C++ 6.0

Proměnné v CE108.C OutValue1 a OutValue2, obě formátu float, slouží jako výstupní proměnné pro motory M1 a M2 s rozsahem od 0 do 10. Tyto hodnoty se předávají pomocí mathlinku přímo z prostředí Mathematica. Další důležitá proměnná je DevNum, určuje číslo zařízení. Pokud je technologická karta v PC jen jedna, bude toto číslo vždy nula. V jiném případě se toto číslo nalézá v Device Manageru pro karty Adventech, číslo před

názvem karty v okně Instadled device. Pro nastavení výstupního kanálu slouží proměnná OutChan, pro výstupní kanál jedna má hodnotu 0 a pro druhý kanál hodnotu 1.

Pro nadefinování vstupních kanálů slouží proměnné InChan1 a InChan2, v tomto případě byly použity kanály 11 a 12. Naměřená data z těchto kanálů jsou předávána v proměnných InValue1 a InValue2. První hodnota znamená otáčky, druhá protažení pásu. Jako poslední důležitá proměnná se používá DwErrCde, obsahuje kód chyby ovladače technologické karty. Pokud nějaká při jeho používání nastane, vypíše kód, jinak vrací vždy nulu.

Naměřené údaje z proměnných InValue1, InValue2 a kód chyby z DwErrCde se vrací pomocí mathlinku do Mathematicy pomocí následující funkce:

```
MLPutFunction(stdlink, "List", 3);  
  
    MLPutReal(stdlink, InValue1);  
  
    MLPutReal(stdlink, InValue2);  
  
    MLPutInteger(stdlink, dwErrCde);
```

Výpis celého programu CE108.C je v příloze P2.

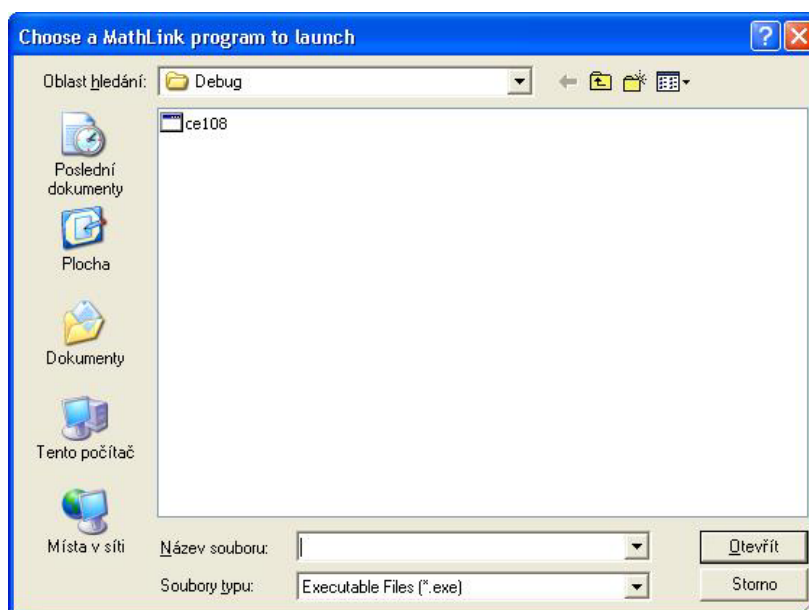
6 POUŽITÍ PROGRAMU CE108 V MATHEMATICCE

Před použitím programu CE108.exe je nutné zkontrolovat sdílenou knihovnu ML32I2.DLL, zda-li se nachází v systémovém adresáři. Pro Windows 9x je to C:\WINDOWS\SYSTEM, pro Windows NT/2000 C:\WINNT\SYSTEM32 a pro Windows XP C:\WINDOWS\SYSTEM32. Pokud se zde nenalézá, je nutné ji zkopírovat z c:\ProgramFiles\WolframResearch\Mathematica\5.1\AddOns\MathLink\DeveloperKit\Windows\SystemAdditions\ do daného umístění. Tato knihovna je univerzální pro všechny verze systému Windows. Dále musí být nainstalována knihovna ADSAPI32.DLL pro ovládání technologické karty Advantech ve stejných systémových adresářích.

Přilinkování programu v prostředí Mathematica se volá příkazem

```
link = Install[LinkLaunch[]]
```

Uživatel je po spuštění buňky vyzván k vyhledání programu (Obr.33).



Obr. 33 Výběr umístění CE108.exe

Po úspěšném přilinkování programu CE108.exe se funkce použije následovně, viz Obr. 34:

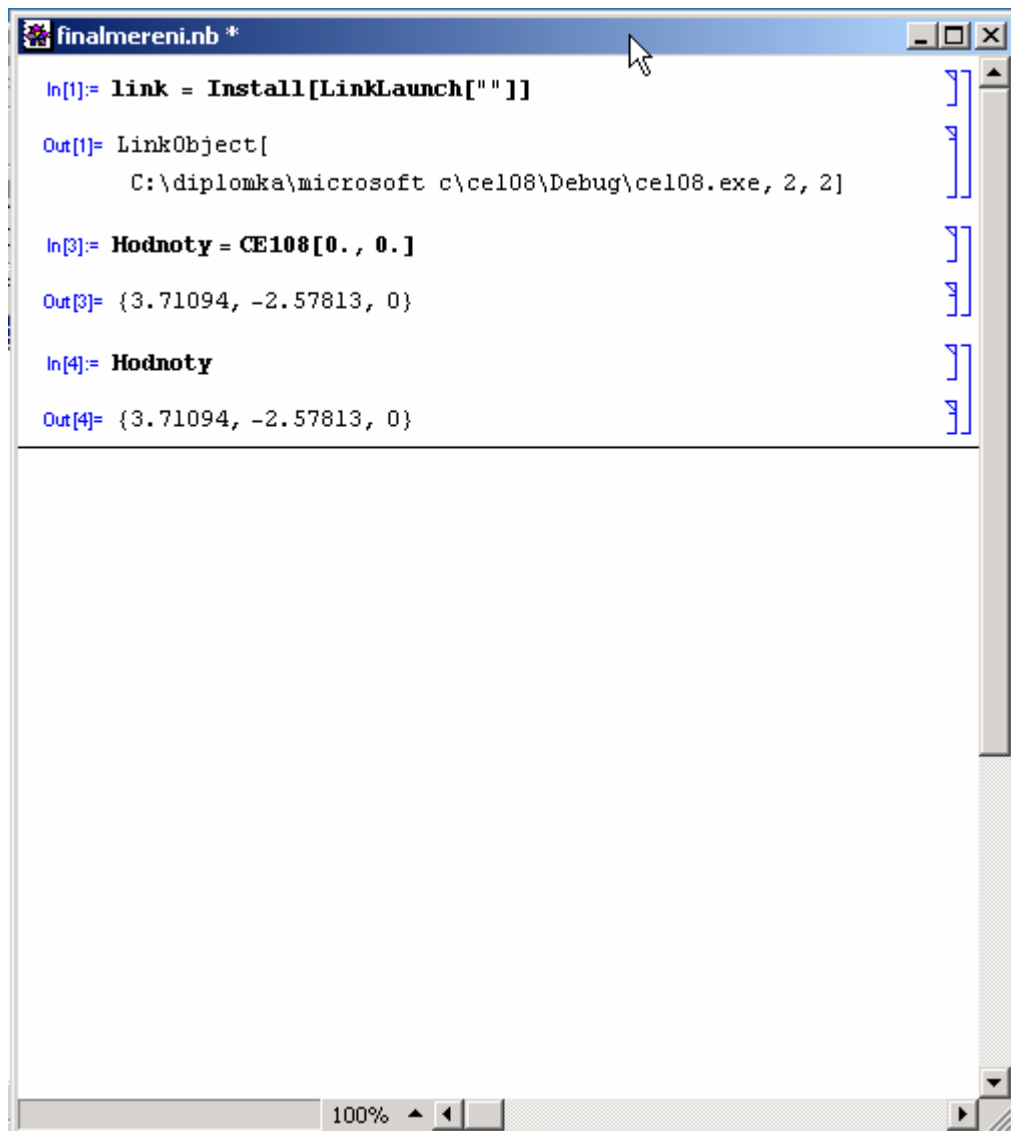
```
CE108[3.,3.]
```

Za čísla jsou důležité tečky! Znamenají, že se proměnná bude do programu předávat jako reálné číslo, kvůli syntaxi softwaru Mathematica. Možný rozsah ovládání je 0 až 10. Čísla symbolizují řídicí napětí motorů, což odpovídá otáčkám 0-3000ot/min.

Po spuštění program vrátí trojici čísel, např.

{3.71094,-2.57813,0}

První dvě hodnoty jsou naměřená data, otáčky pásu a vychýlení ramene (protažení pásu). Třetí hodnota je kód chyby, pokud nějaká při provádění programu nastala. Chybu lze identifikovat dle jejího kódu v manuálu k technologické kartě.



```
finalmereni.nb *  
  
In[1]:= link = Install[LinkLaunch[" "]]  
Out[1]= LinkObject[  
    C:\diplomka\microsoft c\ce108\Debug\ce108.exe, 2, 2]  
  
In[3]:= Hodnoty = CE108[0., 0.]  
Out[3]= {3.71094, -2.57813, 0}  
  
In[4]:= Hodnoty  
Out[4]= {3.71094, -2.57813, 0}
```

Obr. 34 Ukázka použití programu CE108 v Mathematice

7 ZHODNOCENÍ VÝSLEDKU

V průběhu diplomové práce bylo provedeno měření pomocí výše popsaného softwarového rozhraní. Postupně se na oba řízené motory nastavovalo řídicí napětí od 1V do 10V po skoku 0,1V. Kombinací těchto hodnot bylo získáno 8100 souborů se vzorky o délce 10 000 hodnot. Údaje pro napětí od 0V do 1V nebyly měřeny kvůli saturaci měřené soustavy v tomto řídicím rozsahu.

Jelikož před dokončením diplomové práce byl vyměněn starý poškozený pás na reálné soustavě za nový a původní data tak nebyla věrohodná, bylo provedeno nové měření. Získaná data obsahovala pouze 500 vzorků pro každou kombinaci. Z časových důvodů nebylo možno znovu měřit všech 10 000 vzorků. Nové měření pak bylo podrobena analýze, zda-li soustava vykazuje chaotické chování.

Pro automatické měření výše zmíněných dat byl vytvořen soubor „finalmereni.nb“ (Obr. 35), i přes toto řešení trvalo první měření po 10 000 vzorcích asi 5 dnů. Výsledná data měla velikost 3GB. Stejný soubor byl použit i pro další měření po 500 vzorcích.



```
SetDirectory["C:\\diplomka\\SimulacniData"]
C:\\diplomka\\SimulacniData

link = Install[LinkLaunch[" "]]
LinkObject[C:\\diplomka\\microsoft c\\ce108\\Debug\\ce108.exe, 2, 2]

pocetvzorku = 10000
odkud = 1
kam = 10
pres = 0.1

50
1
10
3

Table[Table[vzorek = Table[CE108[N[k], N[j]], {i, pocetvzorku}];
  OutputForm["\\n\\nHodnoty vstupu " <> ToString[k] <> " " <>
    ToString[j] <> "\\n" >>>
  "vzorek_" <> ToString[k] <> "_" <> ToString[j] <> ".txt";
  vzorek >>> "vzorek_" <> ToString[k] <> "_" <> ToString[j] <> ".txt",
  {j, odkud, kam, pres}], {k, odkud, kam, pres}];

CE108[0., 0.]
{1.69922, 0.722656, 0}
```

Obr. 35 Výpis programu pro sběr dat

7.1 Lyapunovy exponenty

První metoda, kterou jsme použili pro potvrzení chaosu v souboru naměřených dat, byla metoda výpočtu Lyapunových exponentů. Vzorce byly použity z knihy [3] Chaos and Nonlinear Dynamics.

V prvním kroku se spočítají hodnoty d podle (7) jako rozdíl hodnot x_j a x_i , kdy odstup mezi těmito hodnotami v číselné řadě je 30 hodnot, tj. $j-i = 30$.

$$\begin{aligned}
 d_0 &= x_j - x_i \\
 d_1 &= x_{j+1} - x_{i+1} \\
 d_2 &= x_{j+2} - x_{i+2} \\
 &\cdot \\
 &\cdot \\
 d_n &= x_{j+n} - x_{i+n}
 \end{aligned}
 \tag{7}$$

Výsledky by měly vykazovat exponenciální charakter. U semilogaritmického grafu by hodnoty měly spadat do přímky, jejíž směrnice určuje hodnotu Lyapunova exponentu [3].

Jako druhý krok následuje výpočet Lyapunova exponentu dle vzorce (8).

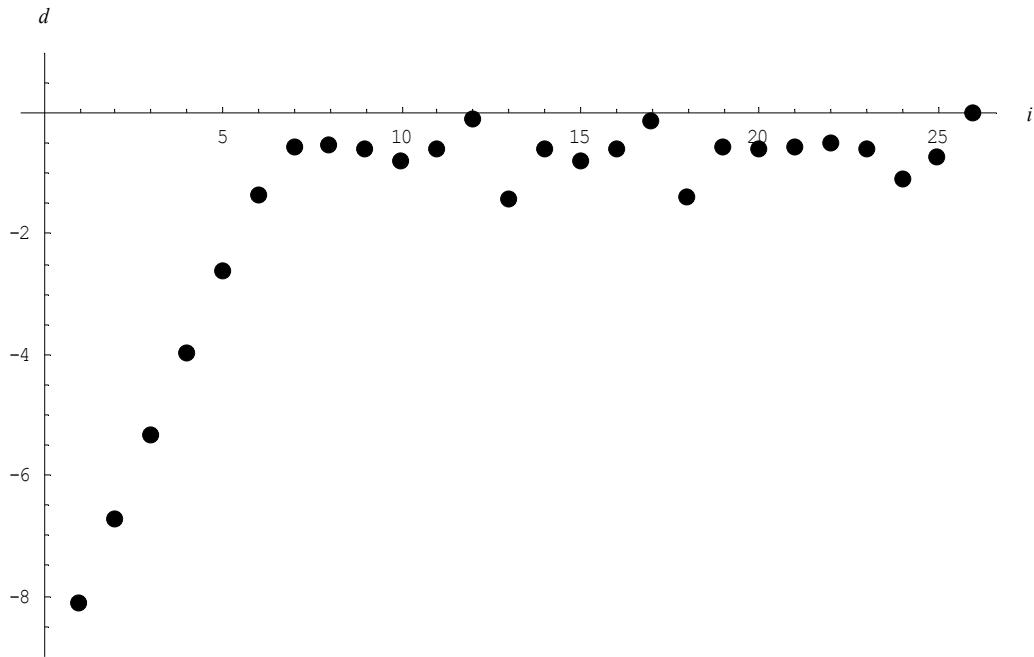
$$\lambda = \frac{1}{n} \ln \frac{d_n}{d_0}
 \tag{8}$$

Jestliže hodnota λ je kladná, data vykazují chaotické chování. Jsou směrodatná pouze za předpokladu, že je splněna podmínka exponenciálního růstu u d .

Správnost naprogramovaných vzorců byla ověřena na logistické rovnici (9) generující chaotické chování.

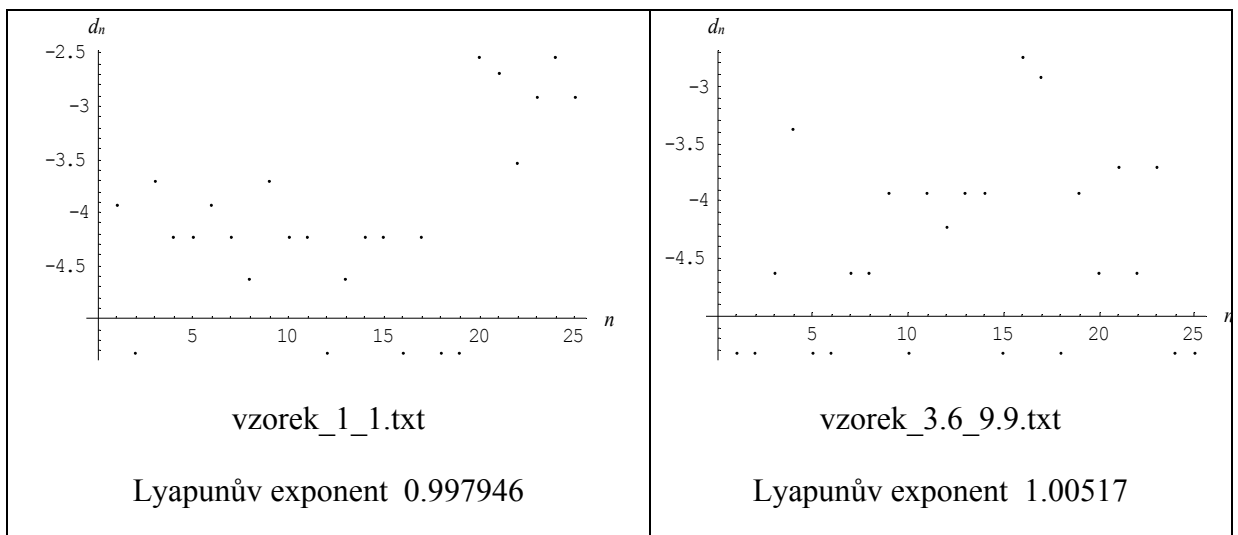
$$x_{n+1} = Ax_n(1 - x_n)
 \tag{9}$$

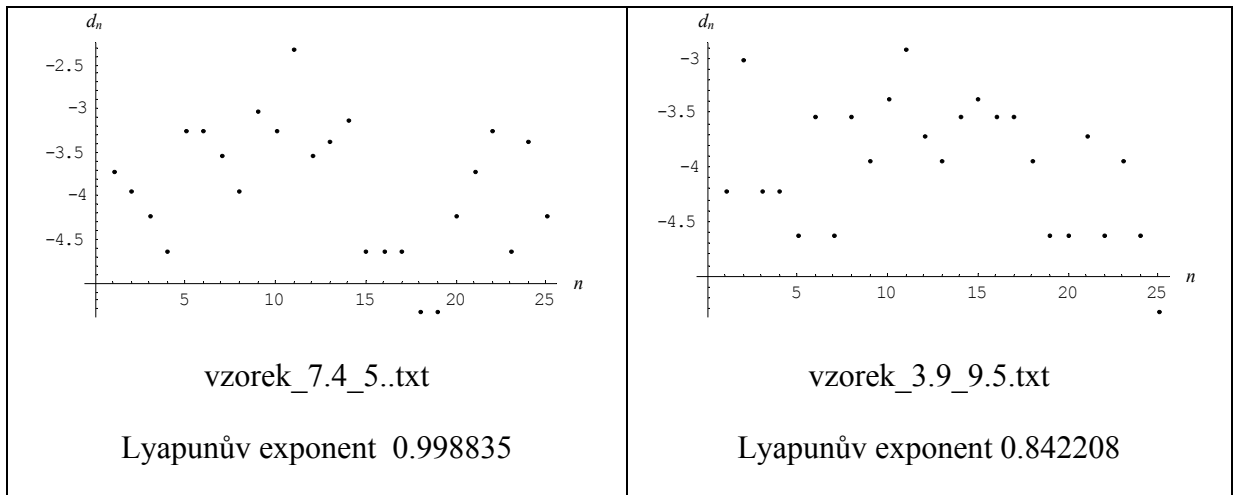
Exponenciální průběh hodnot d , resp. hodnoty v přímce v semilogaritmickém grafu, a správnou funkci výpočtu lze vidět na následujícím obrázku (Obr. 36).



Obr. 36 Semilogaritmičeský graf hodnot d u logistické rovnice

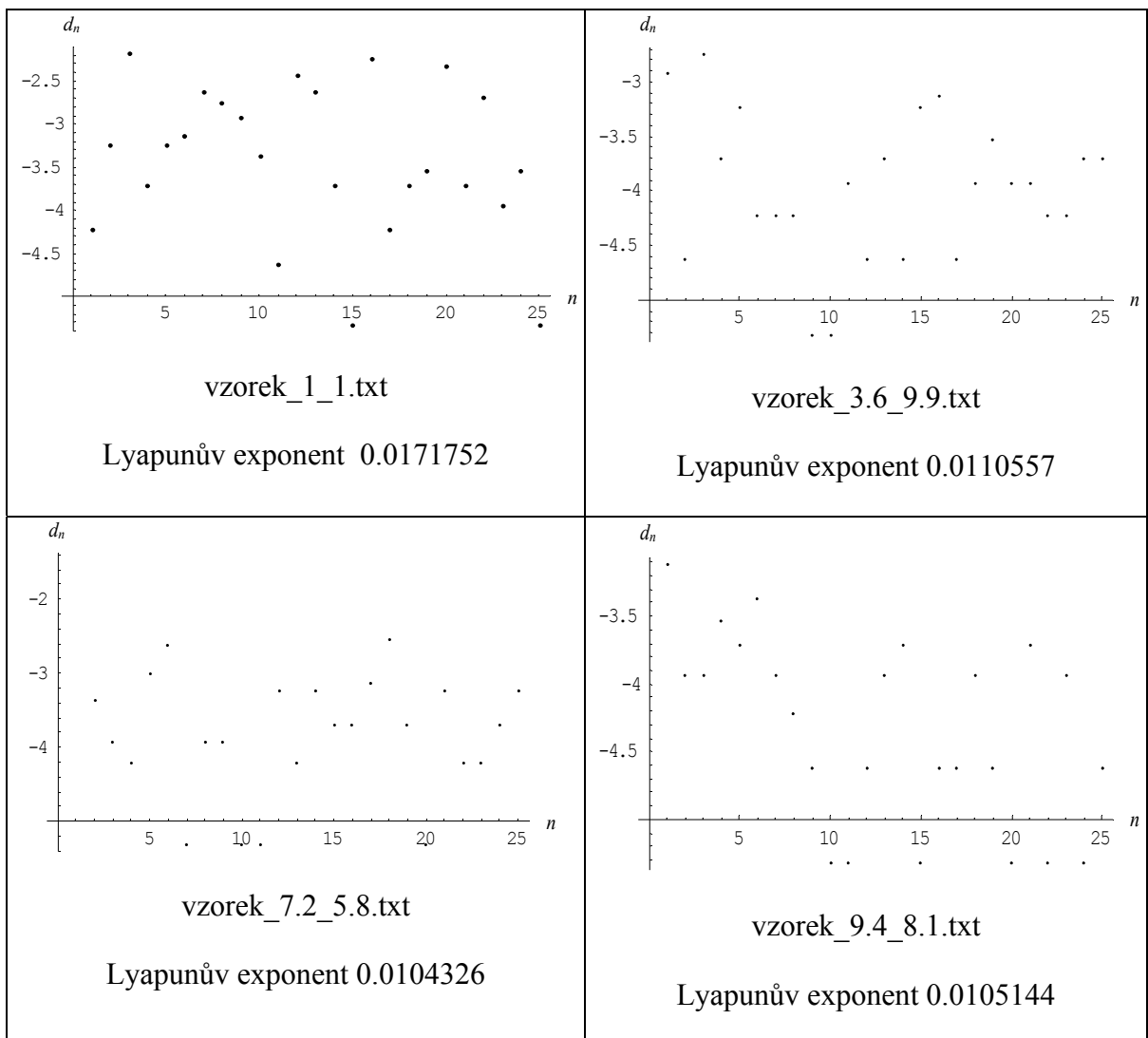
Výpočty provedené u změřených dat a následné vykreslení všech 8100 grafů neprokázalo chaotické chování. Graf, který by na svém počátku měl požadovaný exponenciální růst, se neobjevil. Pro názornost je zde uvedeno několik ukázkových grafů pro otáčky pásu (Obr. 37) a pro protažení pásu (Obr. 38). Lyapunův exponent byl počítán, ale z důvodu nesplnění základní podmínky o exponenciálním růstu nemají hodnoty Lyapunova exponentu význam.





Obr. 37 výsledky pro vybrané vzorky otáček pásu

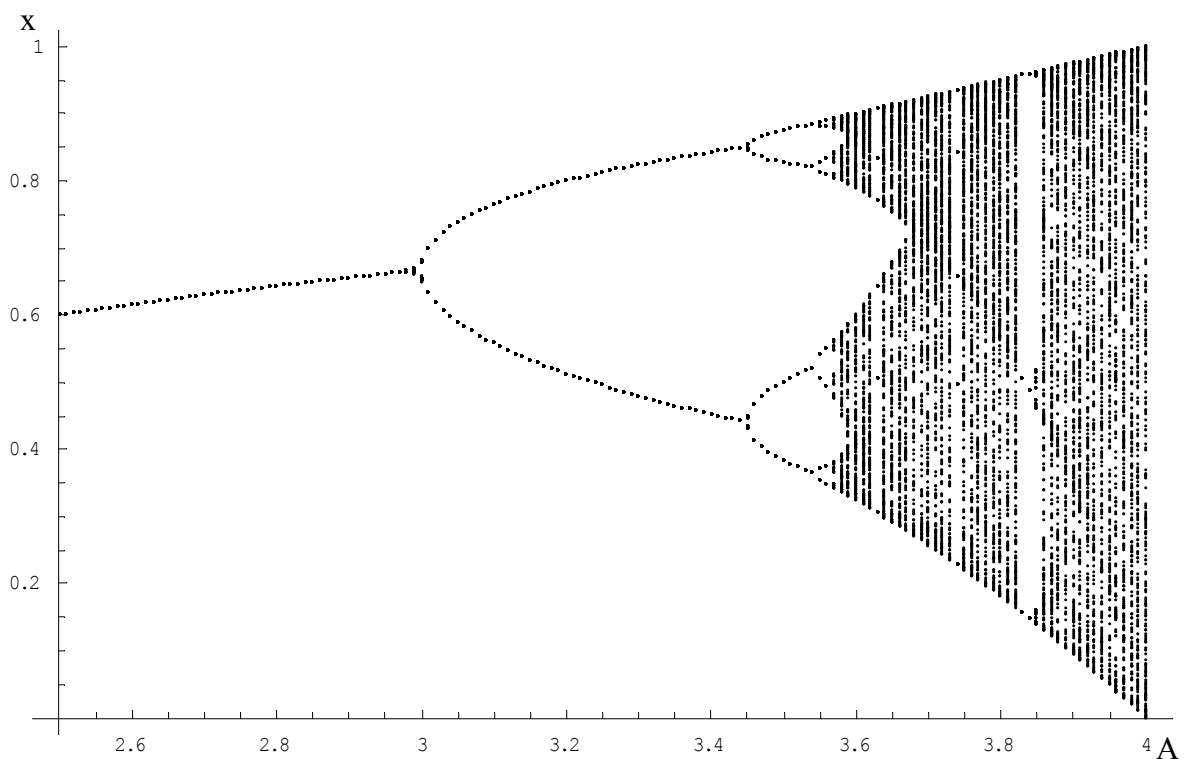
Čísla obsažená v názvu textového souboru s daty odpovídají nastavenému napětí na motorech reálné soustavy.



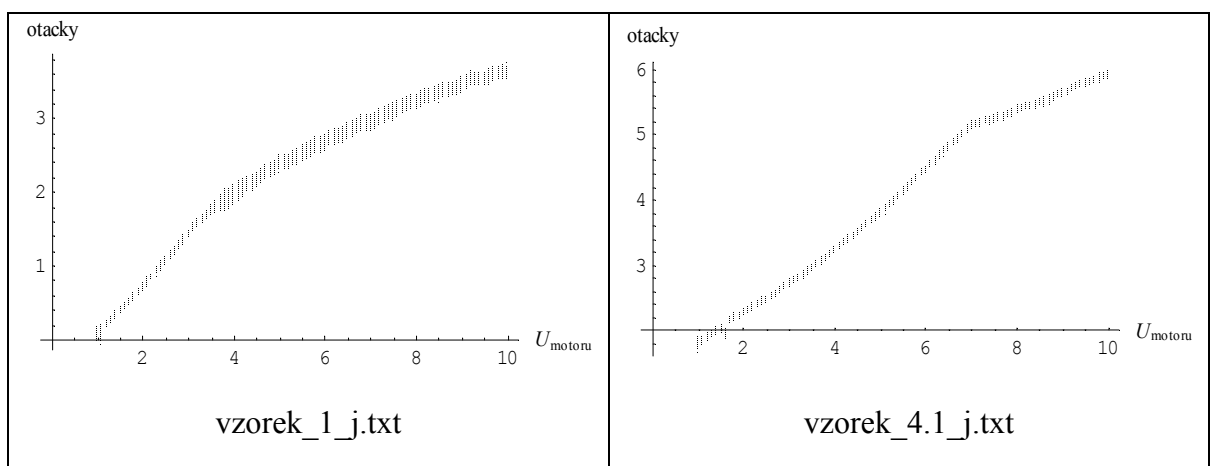
Obr. 38 výsledky pro vybrané vzorky protažení pásu

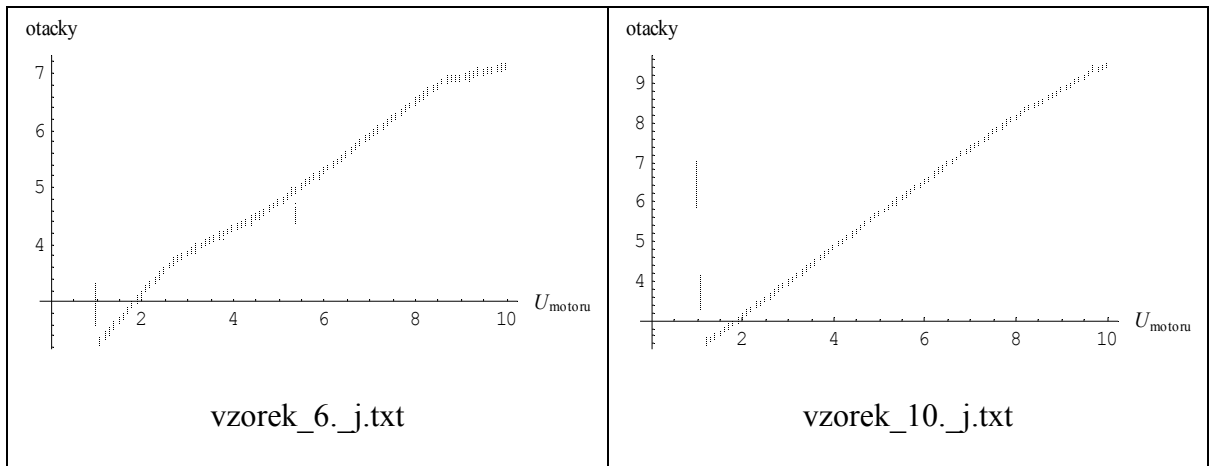
7.2 Bifurkační diagramy

Dalším krokem jsme se snažili prokázat chaos za pomoci vykreslení bifurkačních diagramů. Na Obr. 39 je zobrazen bifurkační diagram logistické rovnice generující chaotické chování, které je zde zřetelné. Proto byly vykresleny bifurkační diagramy pro měřenou soustavu. Jeden z řízených motorů měl konstantní hodnotu otáček a druhý je postupně zvyšoval. Získaná data byla vynesena do bifurkačních grafů pro otáčky Obr. 40 i pro protažení Obr. 42 pásu. Pak se zaměnily vstupní hodnoty řízených motorů, zafixoval se druhý motor a první postupně zvyšoval otáčky. Data byla opět vykreslena do grafů, pro otáčky Obr. 41 a protažení pásu Obr. 43.

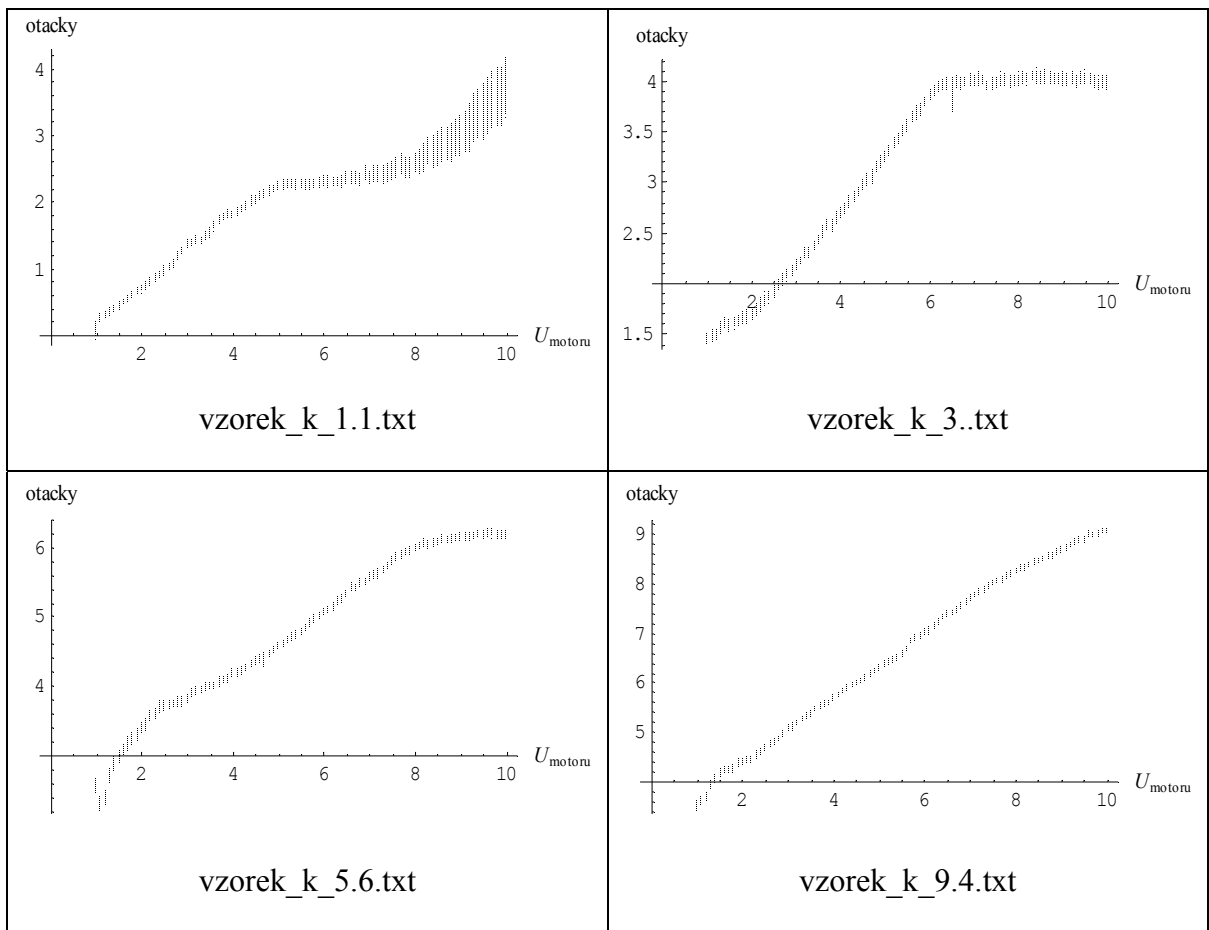


Obr. 39 Bifurkační diagram logistické rovnice

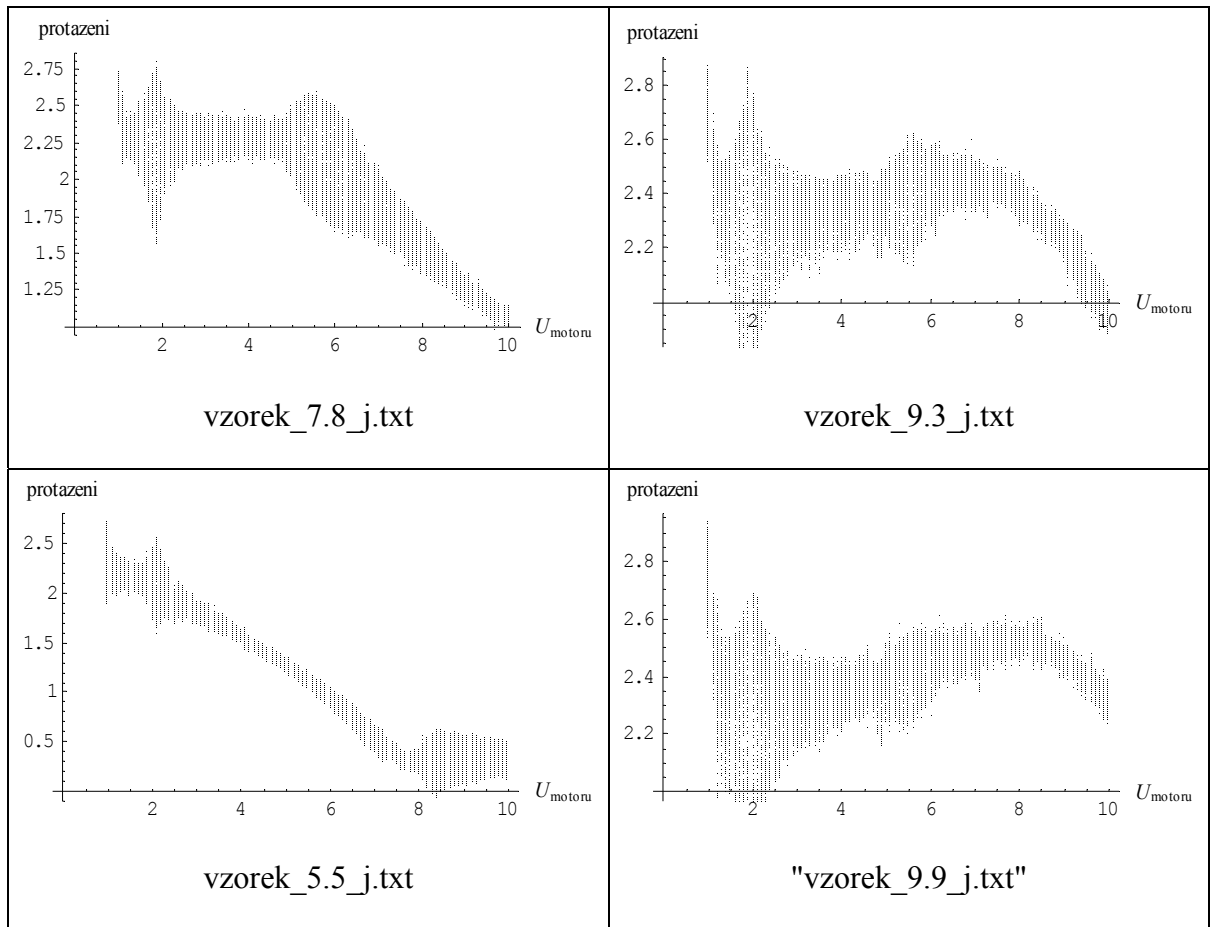




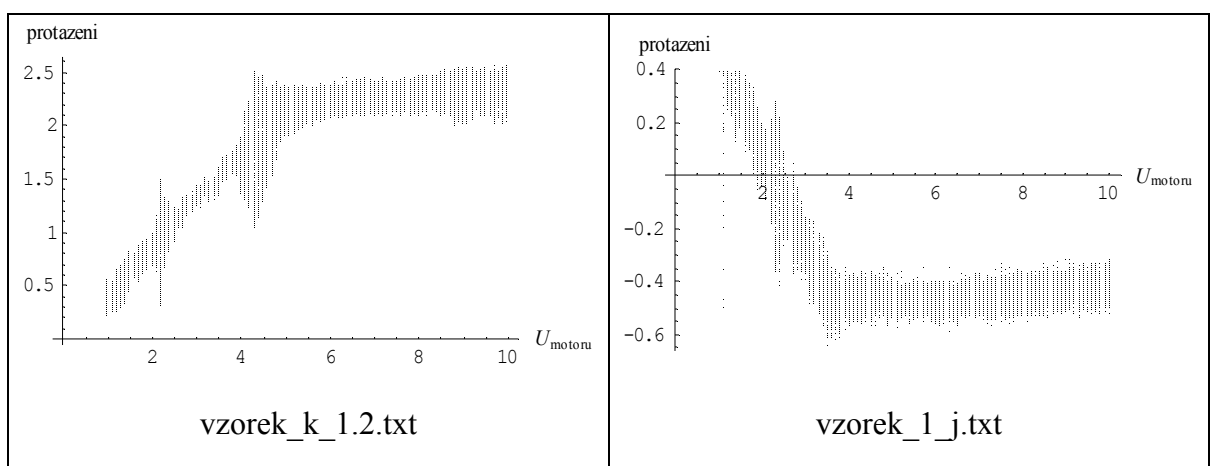
Obr. 40 Bifurkační diagramy pro otáčky pásu

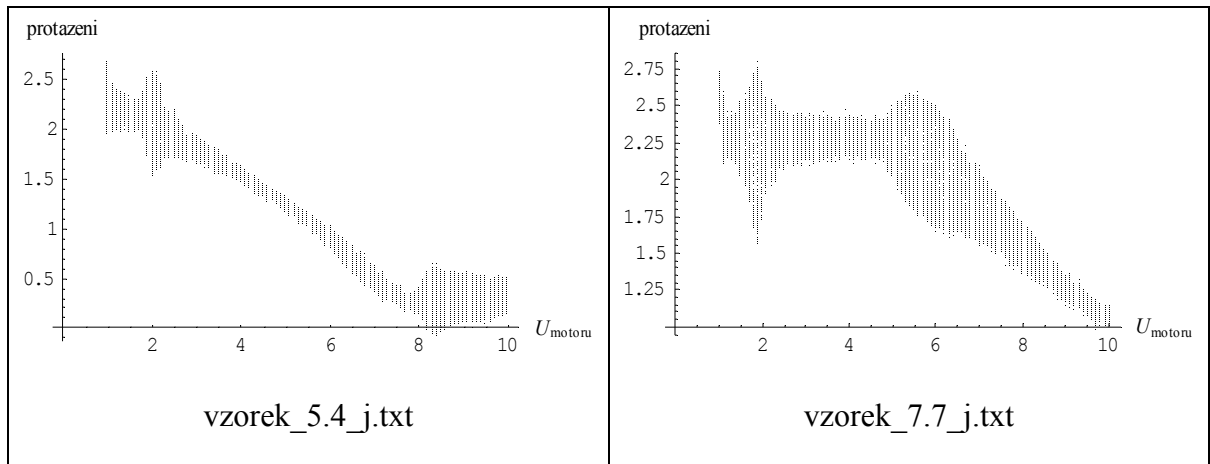


Obr. 41 Bifurkační diagramy pro otáčky motoru po záměně řídicích parametrů



Obr. 42 Bifurkační diagramy pro protažení pásu



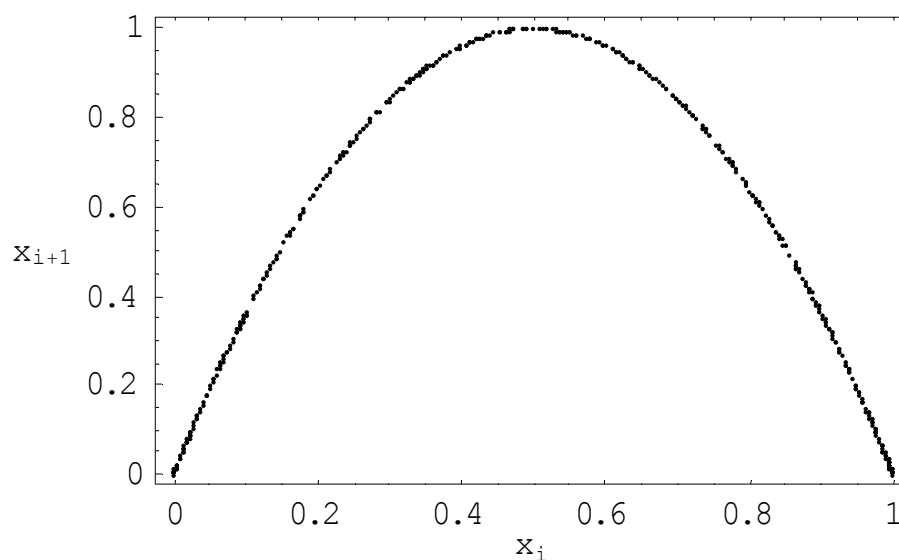


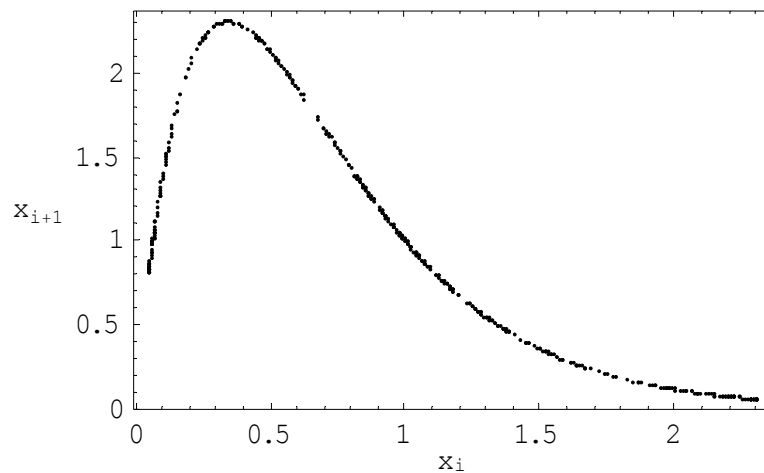
Obr. 43 Bifurkační diagramy pro protažení pásu po záměně parametrů

Jak je z předchozích obrázků vidět, žádný neprokazuje chaotické chování reálné soustavy.

7.3 Závislost předchozího prvku na následujícím

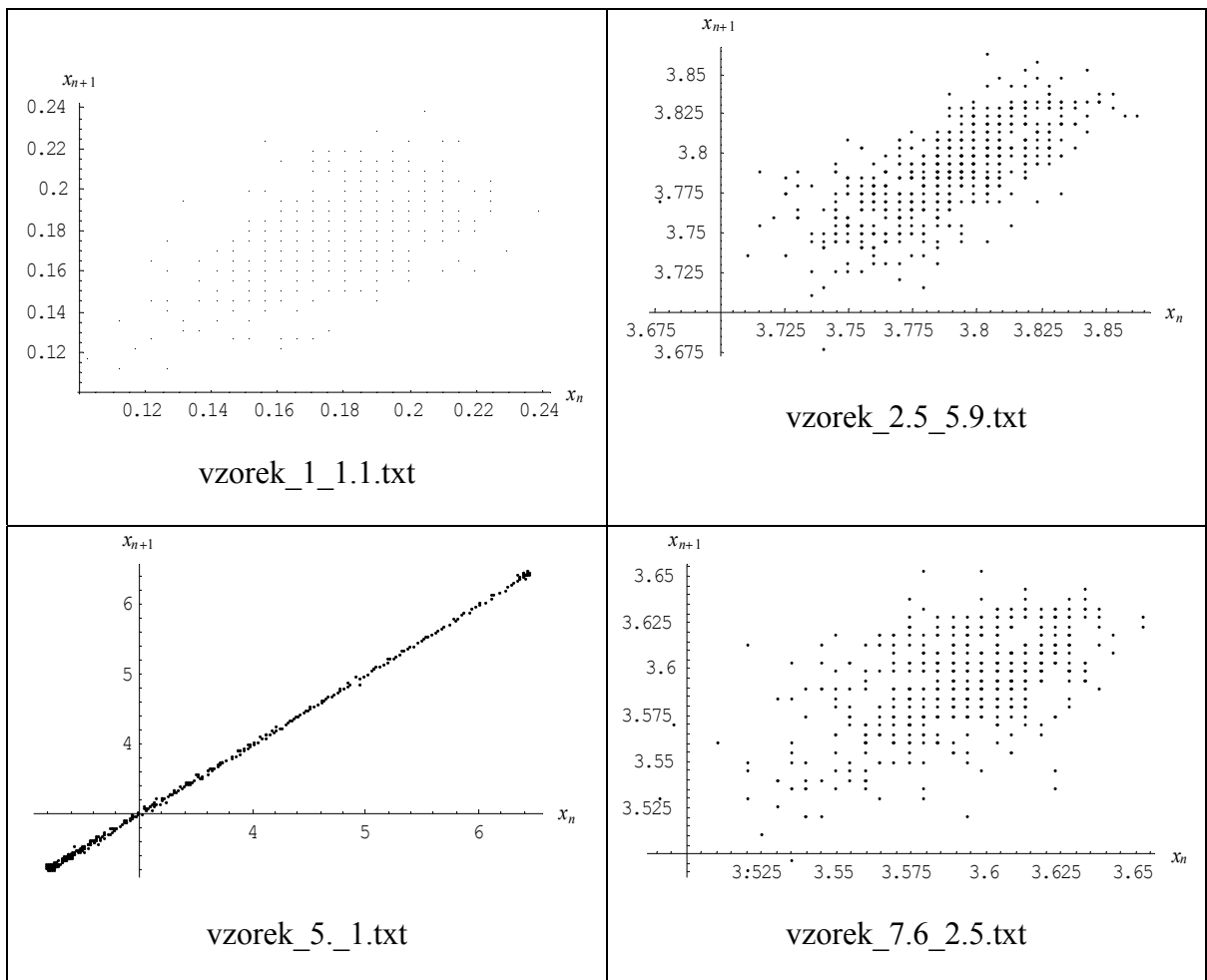
Pro další zobrazení možného výskytu chaotického chování je možné použít metodu grafické závislosti x_{i+1} prvku na x_i prvku z datové řady. Pro logistickou rovnici se tato závislost chová jako parabola, Obr. 44 pro $A=4$ nebo jiná, Obr. 45 pro $A=2.9$.

Obr. 44 Graf závislosti x_{i+1} na x_i pro logistickou rovnici s parametrem $A=4$

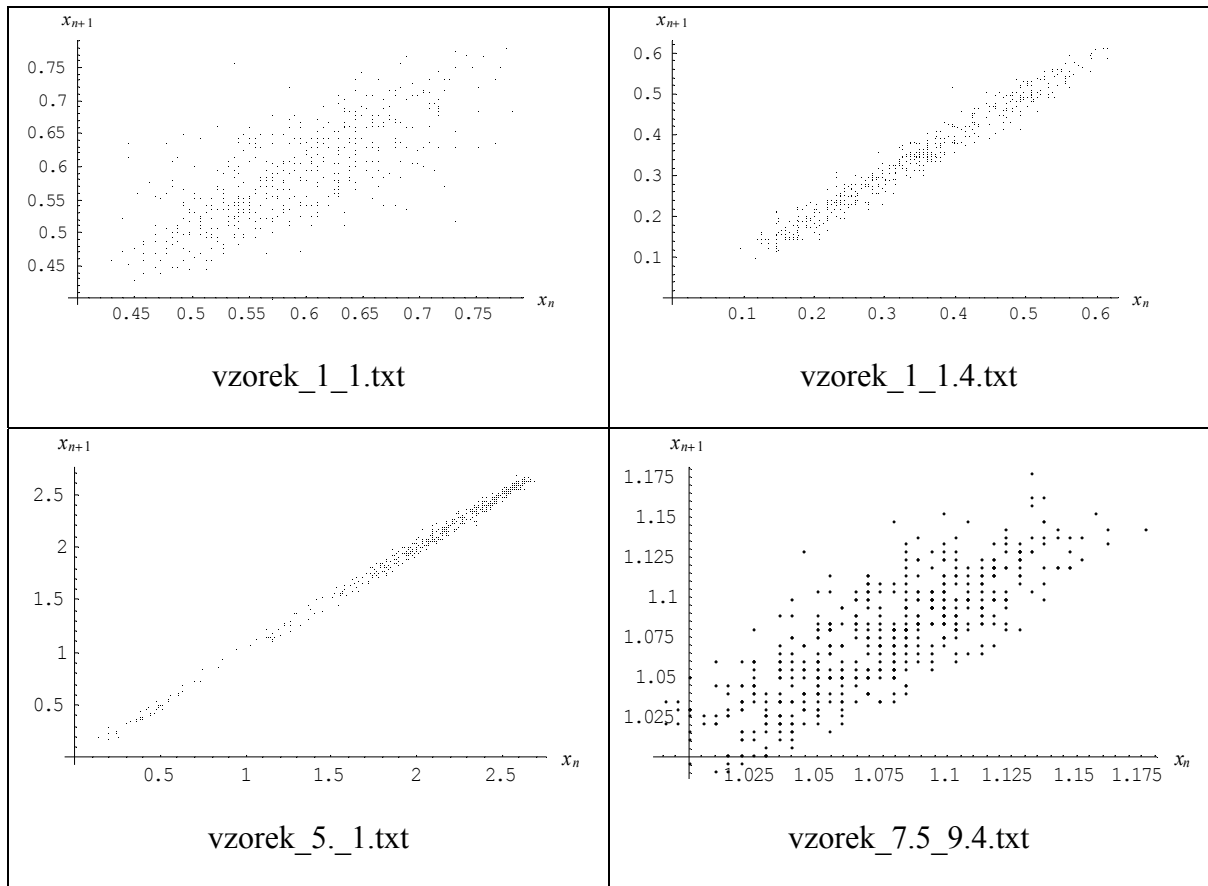


Obr. 45 Graf závislosti x_{i+1} na x_i pro logistickou rovnicí s parametrem $A=2.9$

Pro naměřená data bylo opět vygenerováno 8100 grafů pro otáčky pásu a stejný počet grafů pro protažení pásu. Ukázkové grafy jsou uvedeny na Obr. 46 pro otáčky a na Obr. 47 pro protažení pásu.



Obr. 46 Grafy pro závislost x_{i+1} prvku na x_i pro otáčky pásu

Obr. 47 Grafy pro závislost x_{i+1} prvku na x_i pro protažení pásu

V grafech není vidět žádná závislost. Data se shlukují v mraku, ale není to podobné ani parabole ani žádné jiné funkci.

7.4 Vyhodnocení metod

Pro prokázání chaosu jsme použili tři metody – výpočet Lyapunových exponentů, bifurkační diagramy a závislosti x_{i+1} prvku na x_i . Ani jedna z výše uvedených metod neprokázala u měřených dat chaotické chování. Jelikož se v literatuře neuvádí, jak jsou metody robustní pro měřená data, nemůžeme s jistotou tvrdit, že chaotické chování se v reálné soustavě nevyskytuje. Je možné, že jiné metody by tuto vlastnost mohly potvrdit.

ZÁVĚR

Tato diplomová práce byla zaměřena na prokázání předpokladu chaotického chování reálného modelu soustavy dvou spřažených motorů. V rámci diplomové práce bylo vytvořeno funkční rozhraní mezi softwarovým prostředím Mathematica a reálnou soustavou. Toto rozhraní má univerzální charakter. S kartou Advantech PG812 je schopné ovládat jakékoli jiné reálné modely, i s možností snadného rozšíření vstupních i výstupních kanálů až do maximálních možností technologické karty. Za použití vytvořeného rozhraní bylo naměřeno několik balíčků dat, kdy byly oba řízené vstupy na soustavu postupně inkrementovány, aby se pokryl rozsah všech možných vstupů. Byly měřeny výstupní hodnoty soustavy, v tomto případě otáčky a protažení gumového pásu spojujícího motory. Po naměření uvedených dat byla provedena analýza, zda-li soustava vykazuje chaotické chování či ne.

Jednou z metod bylo vykreslení bifurkačního diagramu pro tuto soustavu. Neukázal na možný výskyt chaosu. Dále byly zvoleny numerické výpočty, výpočet Lyapunova exponentu. Pro důkaz chaotického chování byl spočítán Lyapunův exponent pro všech 8100 souborů s daty, které byly naměřeny jako odezva soustavy, zvláště pro otáčky a pro protažení pásu. Už na počátku se prokázalo, že požadovaný exponenciální růst pro výpočet rozdílů hodnot d není splněn, tudíž spočítaný Lyapunův exponent nemá žádnou vypovídací schopnost o chování dané soustavy. Může být otázkou, je-li tato aplikace Lyapunova exponentu správná. Vzorce podle knihy [3] používáme pro měřená data a ne logistickou rovnici. Podle této metody plyne výsledek, že daná soustava nevykazuje chaotické chování.

Poslední použitou metodou k určení výsledků bylo vynesení grafu pro závislosti x_{i+1} prvku na x_i , což již dle předběžných odhadů přineslo tentýž výsledek, že soustava nevykazuje chaotické chování.

Závěrem provedených měření a vyhodnocení dat tedy je, že soustava spřažených elektromotorů nevykazuje chaotické chování. Tímto není vyloučeno, že za použití jiných metod měření nebo vyhodnocování výsledků, nemůže být na dané soustavě chaotické chování nalezeno.

SEZNAM POUŽITÉ LITERATURY

- [1] Kolář I., Úvod do Thomovy teorie katastrof, Academia, ISBN 21-056-88, 1988
- [2] Gleick J., Chaos, Ando, ISBN 80-86047-04-0, 1996
- [3] Hilborn R.C., Chaos and Nonlinear Dynamics, Oxford University Press, ISBN 0-19-508816-8, 1994
- [4] Haken H., Synergetics: Introduction and Advanced Topics, Springer-Verlag, ISBN 354040824X, 2004
- [5] Ott E., Greboki C., Zorke J.A., Controlling Chaos, Phys. Rev. Lett. 64, 1196, 1990
- [6] Schuster H.G., Handbook of Chaos Control, Wiley-Vch, ISBN 3-527-29436-8, 1999
- [7] Barnsley M.F., Fractals Everywhere, Academic Press Professional, 1993, ISBN 0-12-079061-0
- [8] R. Gilmore, Catastrophe Theory for Scientists and Engineers, John Wiley and Sons, ISBN 0-486-67-539-4, 1993
- [9] Arnold V.I., Teória katastrof, Alfa, ISBN 63-204-86, 1986
- [10] Arnold V.I., The Theory of Singularities and Its Applications, Accademia Nazionale Dei Lincei, Pisa, Italy, ISBN 0-521-42280-9, 1991
- [11] Poston T., Stewart I., Catastrophe Theory and its Applications, Pitman, ISBN, 1977
- [12] May R., Simple Mathematical Modelwith Very Complicated Dynamics, Nature, 261, 45-67, 1976
- [13] Wyk M.A., Steeb W.H., Chaos in electronics, Kluwer academic publisher, 1997, ISBN 0-7923-4576-2
- [14] Guanrong Chen, Controlling Chaos and Bifurcations in Engineering Systems, CRC Press, ISBN 0-8493-0579-9, 2000
- [15] Lam L., Langevin D., Spatio-Temporal Pattern Formation, Springer, ISBN 0-387-94857-0
- [16] Haken H., Information and Self-Organization, Springer-Verlag, ISBN 0-387-18639-5

- [17] Zelinka, Ivan, Elliott's Waves and its Using for Prediction. In Nostradamus'98. Volume 1. Zlín: Knihovna F. Batroše, Zlín, 1998, p. 63;72, ISBN 80-214-1222-4
- [18] Krempaský J. 1994, Synergetika, Vydavatelství STU Bratislava, ISBN 80-227-0707-4, 1994
- [19] Advantech DLL Drivers, User's Manual and Programmer's Reference, 3rd Edition

SEZNAM OBRÁZKŮ

Obr. 1 A.M.Ljapunov	9
Obr. 2 G.Cantor	9
Obr. 3 Edward N. Lorenz –Objevitel chaosu.....	10
Obr. 4 Lorenzův atraktor	10
Obr. 5 O. Rösler.....	12
Obr. 6 Röslerův atraktor	12
Obr. 7 Chování Van der Polova systému.....	13
Obr. 8 L. D. Landau.....	13
Obr. 9 J. H. Poincare	14
Obr. 10 R. Thom	14
Obr. 11 M. Feigenbaum.....	15
Obr. 12 A. N. Kolmogorov	15
Obr. 13 Thomovy katastrofy.....	19
Obr. 14 Samoorganizace – samovolně se tvořící šestihrany v zahříváné kapalině, celkový pohled (vlevo) a detail (vpravo)	22
Obr. 15 Řízení časoprostorového chaosu	24
Obr. 16 Logistická rovnice; a) chaotické chování rovnice při vhodné hodnotě řídícího parametru, b) deterministické chování téže rovnice pro jiné nastavení, c) bifurkační diagram, d) detail. Všimněte si v části a) v kroku (osa x) 25 – 31 a 70 – 75 tvaru, který křivka vytváří. Jde o tzv. „trojúhelník“, jednu z mnoha geometrických formací (též technické ukazatele), které jsou zcela běžně využívány jako ukazatel blízkého budoucího chování pro obchodování na burzách.	26
Obr. 17 WEB diagram	26
Obr. 18 Různé bifurkační diagramy	27
Obr. 19 Jednoduchý elektronický obvod generující chaos.....	28
Obr. 20 Ukázka napnutí pásu.....	30
Obr. 21 Fotka soustavy CE108	31
Obr. 22 Princip činnosti modelu CE108.....	32
Obr. 23 Vstupy a výstupy modelu CE108	32
Obr. 24 Popis prvku modelu	33

Obr. 25. Schéma motorů - převzato z [19]	35
Obr. 26 Karta Adventech PCL-812PG	36
Obr. 27 Instalační okno ovladače	37
Obr. 28 Okno Advantech device manageru.....	37
Obr. 29 Nastavení rozsahu kanálů	38
Obr. 30 Testovací okno vstupních kanálů	38
Obr. 31 Okno pro test výstupů.....	38
Obr. 32 Okno s projektem v Microsoft Visual C++ 6.0	40
Obr. 33 Výběr umístění CE108.exe.....	42
Obr. 34 Ukázka použití programu CE108 v Mathematice	43
Obr. 35 Výpis programu pro sběr dat	44
Obr. 36 Semilogaritmický graf hodnot d u logistické rovnice	46
Obr. 37 výsledky pro vybrané vzorky otáček pásu.....	47
Obr. 38 výsledky pro vybrané vzorky protažení pásu	47
Obr. 39 Bifurkační diagram logistické rovnice	48
Obr. 40 Bifurkační diagramy pro otáčky pásu.....	49
Obr. 41 Bifurkační diagramy pro otáčky motoru po záměně řídicích parametrů.....	49
Obr. 42 Bifurkační diagramy pro protažení pásu	50
Obr. 43 Bifurkační diagramy pro protažení pásu po záměně parametrů	51
Obr. 44 Graf závislosti x_{i+1} na x_i pro logistickou rovnici s parametrem $A=4$	51
Obr. 45 Graf závislosti x_{i+1} na x_i pro logistickou rovnici s parametrem $A=2.9$	52
Obr. 46 Grafy pro závislost x_{i+1} prvku na x_i pro otáčky pásu.....	52
Obr. 47 Grafy pro závislost x_{i+1} prvku na x_i pro protažení pásu	53

SEZNAM PŘÍLOH

Příloha P I: Výpis Zdrojového kódu pro start.tm

Příloha P II: Výpis Zdrojového kódu pro CE108.exe

PŘÍLOHA P I: VÝPIS ZDROJOVÉHO KÓDU PRO START.TM

/* To launch this program from within Mathematica use:

* In[1]:= link = Install["sumalist"]

*

* Or, launch this program from a shell and establish a

* peer-to-peer connection. When given the prompt Create Link:

* type a port name. (On Unix platforms, a port name is a

* number less than 65536. On Mac or Windows platforms,

* it's an arbitrary word.)

* Then, from within Mathematica use:

* In[1]:= link = Install["portname", LinkMode->Connect]

*/

:Begin:

:Function: CE108

:Pattern: CE108[OutValue1_Real,OutValue2_Real]

:Arguments: { OutValue1,OutValue2 }

:ArgumentTypes: { Real,Real }

:ReturnType: Manual

:End:

:Evaluate: "CE108[OutValue1,OutValue2]

```
int PASCAL WinMain( HANDLE hinstCurrent, HANDLE hinstPrevious, LPSTR lpsz-
CmdLine, int nCmdShow)
{
    char buff[512];
    char FAR * buff_start = buff;
    char FAR * argv[32];
    char FAR * FAR * argv_end = argv + 32;

    if( !MLInitializeIcon( hinstCurrent, nCmdShow)) return 1;
    MLScanString( argv, &argv_end, &lpszCmdLine, &buff_start);
    return MLMain( argv_end - argv, argv);
}
```

```
int main(argc, argv)
    int argc; char* argv[];
{
    return MLMain(argc, argv);
}
```

PŘÍLOHA P II: VÝPIS ZDROJOVÉHO KÓDU PRO CE108.EXE

```
#include <windows.h>
#include <windef.h>
#include <stdio.h>
#include <conio.h>
#include "include\driver.h"           //hlavičkový soubor pro ovladač technologické karty
#include "include\mathlink.h"       //hlavičkový soubor pro rozhraní Matematiky a C

void CE108 P(( double,double));
void CE108 (double OutValue1,double OutValue2) //definice funkce se vstupními proměnnými
{
    //definice dalších proměnných
    DWORD dwErrCde;
    ULONG DevNum;
    long DriverHandle;
    USHORT OutChan;
    USHORT InChan1;
    USHORT InChan2;
    float InValue1;
    float InValue2;

    PT_AOVoltageOut tAOVoltageOut;
    PT_AIVoltageIn ptAIVoltageIn;
    PT_AIConfig ptAIConfig;

    //Step1: Input parameters;
    // nDevice Number (check the device installation utility): ");
    DevNum=0;           // pokud je karta v pc jedna vždy 0
    // input parameters for output;
        //Output Channel1;
    OutChan=0;
    //Output Value;
    //OutValue1=2;     //hodnota výstupu při testování programu

    //Step 3: Open device
```

```
dwErrCde = DRV_DeviceOpen(DevNum, &DriverHandle); //otevára přístup k zařízení
```

```
//poslání dat
```

```
// Step 4: Output value to the specified channel
```

```
tAOVoltageOut.chan = OutChan;
```

```
tAOVoltageOut.OutputValue = OutValue1;
```

```
dwErrCde = DRV_AOVoltageOut(DriverHandle, &tAOVoltageOut);
```

```
//Output Channel 2;
```

```
OutChan=1;
```

```
//Output Value 2;
```

```
//OutValue2=2;
```

```
//poslání dat
```

```
// Step 4: Output value to the specified channel
```

```
tAOVoltageOut.chan = OutChan;
```

```
tAOVoltageOut.OutputValue = OutValue2;
```

```
dwErrCde = DRV_AOVoltageOut(DriverHandle, &tAOVoltageOut);
```

```
//načítání dat
```

```
//input parameters for input;
```

```
//input channel;
```

```
InChan1=11;
```

```
InChan2=12;
```

```
//otáčky
```

```
//Step 5: Config device
```

```
ptAIConfig.DasChan = InChan1;
```

```
ptAIConfig.DasGain = 0;
```

```
dwErrCde = DRV_AIConfig(DriverHandle, &ptAIConfig);
```

```
// Step 6: Read one data
```

```
ptAIVoltageIn.chan = InChan1; // input channel
```

```
ptAIVoltageIn.gain = 0; // gain code: refer to manual for voltage range
```

```
ptAIVoltageIn.TrigMode = 0; // 0: internal trigger, 1: external trigger
```

```

ptAIVoltageIn.voltage = &InValue1;    // Voltage retrieved

dwErrCde = DRV_AIVoltageIn(DriverHandle, &ptAIVoltageIn);

    // Step 7: Display reading data
//printf("Reading data = %10.6f\n", InValue1);

//vychyleni
    //Step 5: Config device
ptAIConfig.DasChan = InChan2;
ptAIConfig.DasGain = 0;
dwErrCde = DRV_AIConfig(DriverHandle, &ptAIConfig);

    // Step 6: Read one data
ptAIVoltageIn.chan = InChan2;        // input channel
ptAIVoltageIn.gain = 0;              // gain code: refer to manual for voltage range
ptAIVoltageIn.TrigMode = 0;         // 0: internal trigger, 1: external trigger
ptAIVoltageIn.voltage = &InValue2;  // Voltage retrieved

dwErrCde = DRV_AIVoltageIn(DriverHandle, &ptAIVoltageIn);

    // Step 7: Display reading data
//printf("Reading data = %10.6f\n", InValue2);

    // Step 8: Close device
dwErrCde = DRV_DeviceClose(&DriverHandle);
//vraci kod chyby
    //printf("\nOutput data = %f\n", dwErrCde);

MLPutFunction(stdlink, "List", 3);    //rozhraní mathlink pro vrácení proměnných do Mathematicy
MLPutReal(stdlink, InValue1);
    MLPutReal(stdlink, InValue2);
    MLPutInteger(stdlink, dwErrCde);

return;
} //main

```