

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A
BIOMECHANIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND
BIOMECHANICS

NÁVRH ŘÍDICÍHO ČLENU S POUŽITÍM METODY CARLA

CONTROLLER DESIGN WITH USING CARLA METHOD

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MICHAL JASANSKÝ

VEDOUCÍ PRÁCE

SUPERVISOR

doc. RNDr. Ing. TOMÁŠ BŘEZINA,
CSc.

BRNO 2008

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav mechaniky těles, mechatroniky a biomechaniky

Akademický rok: 2007/08

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Jasanský Michal

který/která studuje v **bakalářském studijním programu**

obor: **Mechatronika (3906R001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Návrh řídicího členu s použitím metody CARLA

v anglickém jazyce:

Controller design with using CARLA method

Stručná charakteristika problematiky úkolu:

Navrhnout, implementovat a simulačně ověřit 2 řídicí členy, s nelineárním kompenzátorem používajícím metody CARLA (Continuos Action Reinforcement Learning Automaton).

Cíle bakalářské práce:

Získat simulačně ověřenou implementaci řídicího členu používajícího metodu CARLA, funkční pod Matlab/Simulink.

Seznam odborné literatury:

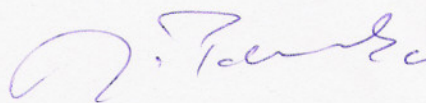
1. Frost, G.P., Gordon, T.J., Howell, M.N., Wu, Q.H., 1996. Reinforcement learning of active and semi-active vehicle suspension control laws. Proceedings of the Institution of Mechanical Engineers Part I 210, 249–257.
2. Howell, M.N., Frost, G.P., Gordon, T.J., Wu, Q.H., 1997. Continuous action reinforcement learning applied to vehicle suspension control. Mechatronics 7 (3), 263–276.
3. Howell, M.N., Gordon, T.J., 2001. Continuous action reinforcement learning automata and their application to adaptive digital filter design. Engineering Applications of Artificial Intelligence 14, 549–561.
4. Narendra, K.S., Thathachar, M.A.L., 1989. Learning Automata: an Introduction. Prentice-Hall, London.

Vedoucí bakalářské práce: doc. RNDr. Ing. Tomáš Březina, CSc.

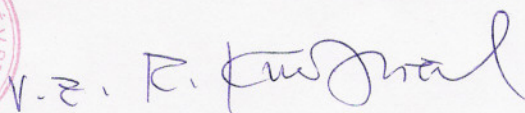
Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2007/08.

V Brně, dne 28.11.2007

L.S.



prof. Ing. Jindřich Petruška, CSc.
Ředitel ústavu



doc. RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

Abstrakt

Tato bakalářská práce je zaměřená na aplikaci metody opakovaně posilovaného učení známou pod názvem „Continuous Action Reinforcement Learning Automata“ (CARLA). V práci je shrnut základní popis metod umělé inteligence, ale podrobně se zabývá pouze metodou CARLA. Dále jsou v práci uvedeny její výhody, nevýhody a popsáno, na jakém matematickém základě je založena úprava zjednodušené verze metody CARLA. Hlavní část práce se zabývá testování a laděním metody s ukázkou vlivu nastavení jednotlivých parametrů. Nakonec je metoda úspěšně aplikována na dvou soustavách: DC motor a fyzikální kyvadlo.

Abstract

This bachelor thesis is focused on the application repeatedly reinforcement learning method named „Continuous Action Reinforcement Learning Automata“(CARLA). There is recapitulated a basic describe of the artificial intelligence method, however it deals mainly with the CARLA method. In the thesis are described the advantages, disadvantages and the type of mathematical basis for simplified version of the CARLA method. The main part of the thesis is concerned with the method testing and debugging. The influence of the setting the parameters is also demonstrated in the thesis. The method is successfully applied on two systems: DC motor and physical pendulum.

Klíčová slova

Umělá inteligence, opakovaně posilované učení, řízení, CARLA.

Key words

Artificial intelligence, reinforcement learning, control, CARLA.

Bibliografická citace práce

JASANSKÝ, M. *Návrh řídicího členu s použitím metody CARLA*. Brno: VUT v Brně, Fakulta strojního inženýrství, 2008. 52s. Vedoucí bakalářské práce doc. RNDr. Ing. Tomáš Březina, CSc.

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně 23. 5. 2008

podpis

Poděkování

Chtěl bych poděkovat doc. RNDr. Ing. Tomáši Březinovi, CSc. za pečlivé vedení mojí bakalářské práce, jeho cenné konzultace, četné rady a připomínky. Dále bych rád poděkoval všem, kteří mě podporovali a provázeli celým obdobím bakalářského studia.

OBSAH

1	ÚVOD	10
2	VÝVOJ V OBLASTI AUTOMATICKÉHO ŘÍZENÍ.....	11
2.1	Dnešní možnosti řízení	12
2.1.1	Konvenční metody nastavení zisků regulátoru	12
2.1.2	Nekonvenční metody nastavení zisků regulátoru	13
3	PRINCIP METODY CARLA	16
3.1	Metoda CARLA se člení na sedm základních částí.....	17
3.2	Inicializace rozložení pravděpodobnosti.....	18
3.3	Výběr akce	18
3.4	Aplikace vybrané akce na systém a čekání na její odezvu	18
3.5	Výpočet ceny akce	18
3.6	Výpočet posílení	19
3.7	Úprava rozložení pravděpodobnosti	19
3.8	Výhody.....	20
3.9	Nevýhody	20
4	ÚPRAVA VZORCŮ A IMPLEMENTACE DO S-FUNKCE	21
4.1	Implementace pod prostředí Matlab/Simulink.....	23
4.2	Zapojení bloku v Simulinku.....	23
4.3	Ladící parametry	24
4.3.1	Vliv parametru učení na proces učení.....	25
4.3.2	Vliv rozsahu přestavení na proces učení.....	26
4.3.3	Vliv periody přestavení na proces učení	27
4.3.4	Vliv počáteční četnosti na proces učení.....	28
4.3.5	Vliv intervalu hledaného řešení na proces učení.....	29
4.3.6	Vliv vzorkovací periody na proces učení.....	31
4.3.7	Zhodnocení kapitoly:	32

4.4	Výhody.....	32
4.5	Nevýhody.....	32
5	TESTOVÁNÍ CHOVÁNÍ NA TECHNICKÝCH SOUSTAVÁCH.....	33
5.1	Stejnoseměrný motor (DC).....	33
5.1.1	Model DC motoru v Simulinku	34
5.1.2	Implementace soustavy s regulátorem CARLA.....	35
5.1.3	Konvergence skutečné rychlosti DC motoru k referenční	36
5.1.4	Průběh řídicího napětí – střední hodnota	37
5.1.5	Cena vybraných akcí.....	38
5.1.6	Rozsah akčních zásahů.....	39
5.1.7	Směrodatná odchylka.....	40
5.1.8	Zhodnocení kapitoly.....	40
5.2	Fyzikální kyvadlo.....	40
5.2.1	Model kyvadla v Simulinku.....	41
5.2.2	Implementace soustavy s regulátorem CARLA.....	41
5.2.3	Konvergence skutečné polohy kyvadla k referenční	42
5.2.4	Průběh řídicího momentu – střední hodnota	43
5.2.5	Cena vybraných akcí.....	44
5.2.6	Směrodatná odchylka.....	45
5.2.7	Zhodnocení kapitoly.....	45
6	ZÁVĚR.....	46
	SEZNAM POUŽITÉ LITERATURY	47
	SEZNAM POUŽITÝCH SYMBOLŮ.....	49
	SEZNAM GRAFŮ, TABULEK A OBRÁZKŮ.....	51
	SEZNAM PŘÍLOH NA CD	52

1 ÚVOD

V dnešní době velkého průmyslového rozvoje, automatizace, a zefektivňování výroby, je kladen požadavek na dokonalejší řídicí systémy. Přes veškerý pokrok dosažený ve vývoji nových regulátorů, zůstává PID regulátor stále mezi nejběžněji používanými regulátory. Jeho rozšířenost je dána jednoduchostí, všestranností a vysokou spolehlivostí. Zákon řízení spojitého PID regulátoru je určen jednoduchým vztahem:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, \quad (1)$$

kde $e(t)$ je regulační odchylka (rozdíl požadovaný výstup – skutečný výstup ze soustavy), $u(t)$ je akční veličina, jež je závislá pouze na této regulační odchylce a třech pevných ziscích (K_p, K_i, K_d). Zisky regulátoru se nastavují podle požadavků na chování řízené soustavy, typicky na maximální velikost překmitu, dobu ustálení, velikost regulační odchylky po ustálení a řídicí rozsah. Oblíbenou metodou jejich nalezení je „Ziegler-Nicholsonova metoda“.[6, 8]

Protože je zákon řízení PID regulátoru lineární vzhledem k regulační odchylce, je PID regulátor vhodný nástroj především pro řízení lineárních soustav. Většina technických soustav je ale ze své podstaty nelineární. V případě, že je možné jejich nelinearity zanedbat, tj. že linearizované modely dostatečně věrně aproximují chování svých nelineárních předloh, lze PID použít i pro řízení takových soustav. To lze provést v dostatečně malém okolí ustáleného stavu soustavy vždy. Takto navržený regulátor však může být schopen řídit pouze nepoužitelně malé okolí ustáleného stavu.

Klasický návrh regulátoru lze rozdělit do dvou úloh, které jsou v rámci některých metod prováděny simultánně:

- návrh struktury regulátoru (zákona řízení regulátoru),
- nastavení zisků regulátoru (identifikace hodnot parametrů zákona řízení).

Klasická regulace vyžaduje matematický popis (model) soustavy, ten ale není vždy možné sestavit, anebo efektivně řešit. Pro soustavy s komplikovanou dynamikou nebo soustavy s neznámým (částečně známým) modelem je obtížné navrhnout vhodný regulátor, který by byl schopen systém efektivně řídit. Proto se zabýváme jinými přístupy k jejich řešení, např. inteligentními adaptivními učícími se regulátory. Jednou z těchto metod, které připadají v úvahu pro použití v takových regulátorech, je i

algoritmus CARLA (Continuous Action Reinforcement Learning Automata), který již byl úspěšně prakticky použit (viz.[1, 2, 3]) k řešení úlohy nastavení zisků regulátoru při zvolené struktuře regulátoru. V této práci je publikována původní modifikace této metody, která ji podstatně zjednodušuje a, jak se zdá, i zlepšuje její chování. Metoda je připravena pro použití pod simulačním prostředím Matlab/Simulink.

2 VÝVOJ V OBLASTI AUTOMATICKÉHO ŘÍZENÍ

Oblast adaptivního řízení a adaptivních systémů přitahuje již čtyři desetiletí řadu výzkumných a aplikačních inženýrů. I přes velké pokroky ve vývoji inteligentních adaptivních regulátorů, již v druhé polovině 80. let, nebylo zaznamenáno žádné velké aplikační použití, protože u mnoha výrobců panovala a dodnes panuje jistá nedůvěra v netradiční a složitější řízení. Požadavek na vysokou spolehlivost a kvalitu se u nich neslučoval s důvěrou v nové pojetí regulace. [7]

V současné době díky automobilové, letecké a přesné obráběcí technice se do popředí zájmu dostaly a pevně usídlily tak zvané inteligentní řídicí systémy. Tyto systémy využívají algoritmy umělé inteligence - genetické algoritmy, neuronové sítě, fuzzy množiny, Q-učení, CARLA, atd.

Nesporná výhoda, kterou tedy do regulace přinesly inteligentní regulátory, spočívá v automatickém ladění zisků regulátoru. Metoda umělé inteligence, kterou se v bakalářské práci zabývám, využívá strojového učení v reálném čase, konkrétně opakovaně posilovaného učení („Reinforcement learning“). Algoritmus je formulován jako optimalizační úloha, kde na základě náhodného výběru nastavení zisku regulátoru a následného ocenění vhodnosti nastavení (úspěšnosti), je voleno další nastavení zisku. Během mnohonásobného opakování tento proces může konvergovat k nastavení regulátoru, které maximalizuje úspěšnost nastavení zisku.

Při změně pracovního režimu (např. jiné ustálené otáčky motoru) nebo změně parametrů soustavy (např. opotřebení ložisek – větší tření) se regulátor může přizpůsobit a doladit na nejlepší možné chování řízené soustavy. K nastavení regulátorů není nutné znát matematický model soustavy a odpadá i požadavek znalosti lineárního nebo linerizovaného modelu. Pro rychlejší naučení je výhodné znát přibližné rozsahy vstupních parametrů a pro bezpečný proces učení i technicky realizovatelný rozsah výstupních parametrů. Pokud tato reálná omezení nezahrneme do nastavení učení, hrozí

velké riziko havárie a poškození soustavy nepřiměřenými akčními zásahy, které vyplnou z nevhodného nastavení zisků regulátoru.

Opakovaně posilované učení je založeno na paradigmatu, že optimální výběr akce vyvolá nejlepší ohodnocení akce. Zároveň bere oblast pravděpodobnosti, statistiky a jejich analýzy za rovnocenný aparát k deterministickým metodám popisu soustav. [15] Z principu nedokáže objevit něco nového, ale dokáže ve velkém množství dat najít nejlepší řešení. Metody umělé inteligence by však měly být schopny ze známých zkušeností vyvodit i nové znalosti a využívat je. Tohoto rysu zatím dnešní úroveň algoritmu nedosahuje, avšak je jen otázkou času a peněz, kdy počítače budou inteligentnější než lidé. [13]

2.1 Dnešní možnosti řízení

V současné době je k dispozici mnoho možných přístupů a metod pro návrh regulátorů. Tyto metody se liší v náročnosti návrhu, použití, rychlosti, cenou a přesností regulátoru. Omezíme-li se na nejoblíbenější metody nastavení zisků regulátoru, lze hovořit o skupině metod:

- konvenčních a
- nekonvenčních.

2.1.1 Konvenční metody nastavení zisků regulátoru

Do této skupiny lze zařadit nastavení zisků PID a stavového regulátoru. Přibližně 90% regulátorů používaných v průmyslu je určitou modifikací PID regulátoru. Pro návrh regulátoru potřebujeme lineární nebo linearizovaný model soustavy. Soustava může být nelineární, ale s ohledem na to, že celá klasická teorie automatizovaného řízení je vystavěna na lineárních rovnicích, je soustavu nutné v okolí pracovního bodu (nebo pracovní trajektorie) linearizovat. Metod linearizace je velmi mnoho. Mezi patrně nejpoužívanější se řadí použití prvních dvou členů Taylorova rozvoje. Dále se zpravidla použije některá z následujících metod nastavení zisků regulátoru:

- LQ design,
- metoda volby pólů a
- Ziegler - Nicholsova metoda.

2.1.2 Nekonvenční metody nastavení zisků regulátoru

Umělá inteligence bývá vymezována dosažením tří cílů - schopností učit se, usuzovat a dělat nové závěry z dosažených zkušeností. Učení je v literatuře definováno mnoha autory, výstižně je definován pojem učení v [10] „*jako schopnost systému zlepšovat svoji výkonnost na dané třídě úloh se získáváním zkušeností.*“ Ne všechny inteligentní učící systémy jsou také adaptivní, např. klasický algoritmus neuronové sítě. Zde proběhne jednorázové naučení synoptických vah, a ty se už dále nemění. Neuronová síť může být vhodná pro optimální nastavení zisků (naučení) regulátoru, ale nemusí už být schopna se při změně vstupních či jiných parametrů přeučit na nové parametry. Proto se neuronová síť neřadí mezi adaptivní systémy. Za určitých specifických podmínek je ale i neuronová síť adaptivní systém (viz. odst. Neuronové sítě).[14] V dnešní době se adaptivitou vyznačují zejména algoritmy založené na opakovaně posilovaném učení (Reinforcement Learning). Nejznámější a nejužívanější metoda je podle mého názoru Q-učení, dále pak CARLA.

Inteligentní regulační systémy jsou pro návrh mnohem složitější a dražší. Používají se u soustav s komplikovanou dynamikou nebo u soustav s neznámým (nebo částečně známým) modelem a u soustav, kde má řízení dosažené konvenčními metodami neakceptovatelnou chybu a kde cenová náročnost regulace neohroží konkurenční schopnost zařízení. Pro návrh se používá několik metod umělé inteligence:

- genetické algoritmy,
- neuronové sítě,
- Q-učení,
- CARLA, atd.

Genetické algoritmy

Genetické algoritmy jsou založeny na využití základních rysů genetiky. Při dostatečně dlouhé době učení algoritmus najde zisky regulátoru, které jsou velmi blízké optimálním ziskům.

Genetický algoritmus vygeneruje náhodné členy populace (regulátory s náhodnými zisky) se stejnosměrným rozložením po celém intervalu hledaného řešení. Ke každému jedinci populace vypočte koeficient, který ohodnocuje jeho úspěšnost (odpovídající např. dosažené kvalitě řízení). Následně vytvoří novou populaci, a to křížením dvou náhodně vybraných rodičů (např. použitím konvexní kombinace zisků

obou rodičů). Pravděpodobnost výběru rodičů je tím větší, čím mají vyšší úspěšnost. Aby byla zachována evoluce, je potřeba uskutečňovat (např. každých 10 generací) mutace potomků (náhodnou změnou jejich zisků). Ke konci učení dochází při nalezení hodnot s předepsanou úspěšností. Předepsanou úspěšnost musíme volit s uvážením požadované přesnosti, rychlosti a způsobu výpočtu úspěšnosti. Více v literatuře [12].

Neuronové sítě

Neuronové sítě byly znovu objeveny počátkem 80. let a přinesly velké oživení zájmu o umělou inteligenci. Neuronové sítě implementují konekcionistický model lidského mozku. Neuronová síť je tvořena množinou propojených neuronů (jednoduchých nelinearit). Výstup neuronu je dán váženou sumou jeho vstupů transformovanou přes aktivační matici. Podle typu aktivační matice a struktury propojení neuronu se neuronové sítě dělí na dopředné sítě, Kohonenovy samoorganizující sítě, rekurentní sítě a další. [11]

Velkou výhodou neuronové sítě je, že dokáže nalézt závislosti ve složitých a neúplných datech, která jsou pro komplexní zpracování člověkem příliš rozsáhlá. Mezi další výhody lze zařadit:

- **Adaptivní učení** – ve smyslu schopnosti přizpůsobovat se na základě vstupních dat
- **Automatická organizace** – neuronová síť si vytvoří vlastní strukturu informací, které jí jsou během učení předkládány,
- **Práce v reálném čase** – při použití speciálního HW a SW lze provádět výpočty paralelně na více počítačích a dosáhnout tak online řízení, algoritmus se stává přizpůsobivý na jakékoli podmínky z okolí,
- **Tolerance vůči chybám** – podobně jako v lidském mozku, při částečném poškození neuronové sítě přebírají úlohy poškozené části jiné struktury, snižuje se částečně její výkon, ale schopnosti zůstávají zachovány. Zajímavé ovšem je, že i přes velké destrukce některé schopnosti zůstávají.

Q-učení

Q-učení patří do skupiny metod opakovaně posilovaného učení. Pracuje na jednoduchém principu iterativního zpřesňování hodnocení tzv. Q hodnoty a provedení akce. Velkou výhodou je možnost online učení pouze na základě odměny/trestu za

provedenou akci. Zvolená metoda nepotřebuje znát model systému. Nevýhodu metody však spatřuji v požadavku, pamatovat si po celou dobu běhu programu všechny naučené Q hodnoty, a tedy velká paměťová náročnost. Q hodnoty jsou uspořádány v mřížkách o rozměrech:

počet stavů systému x počet kroků metody

U nestabilních soustav jsou iniciační Q hodnoty nepoužitelné pro řízení, a proto se přechází k tzv. offline učení, kde se Q hodnoty částečně předučí. Předučení může probíhat podle jednoduchého modelu soustavy, nebo pomocí stochastických metod, genetického algoritmu a mnoha jiných. Po naučení následuje online učení, které Q hodnoty zpřesňuje dle odezvy reálné soustavy. Během procesu učení se postupně snižuje rychlost změny Q hodnot až na rychlost blízkou nule. Pak můžeme tyto hodnoty považovat za konečné. Pokud ovšem vlivem vnějších vlivů dojde ke změně soustavy (opotřebení, porucha některé části), Q hodnoty se přeučí opět na nejvýhodnější hodnoty.

Diskrétní náhodný učící se automat

Diskrétní náhodný učící se automat pracuje na neznámém stavovém prostoru, a to pomocí náhodného výběru akcí z předdefinovaných diskrétních hodnot a jejich aplikaci na neznámou technickou soustavu. Podle odezvy soustavy se vyhodnotí úspěšnost volby akce, a to skalární veličinou, kterou nazveme cena. Úspěšnost akce je promítnuta do vztahů prostřednictvím posílení β , který se získá z ceny akce a z charakteristik, vyjadřujících historický průběh cen. Následně dochází k úpravě hustoty pravděpodobnosti

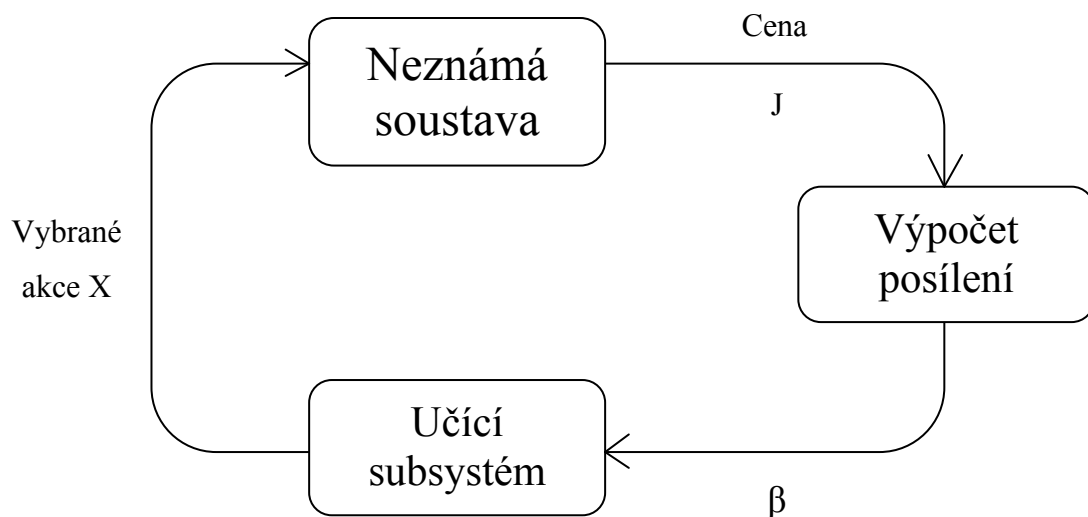
$$p_i(n+1) = p_i(n) + \theta\beta(n)(1 - p_i(n)) \quad (2)$$

$$p_j(n+1) = p_j(n) + \theta\beta(n)p_j(n), \quad \text{když } i \neq j, \quad (3)$$

kde θ je parametr učení, který se pohybuje v rozmezí intervalu $\langle 0;1 \rangle$. Posílení β nabývá hodnot z intervalu $\langle 0;1 \rangle$. Hodnota 0 vyjadřuje, že zvolená akce není úspěšná, a tedy nebude mít žádný vliv na úpravu pravděpodobnosti výběru dalších akcí. Pro hodnotu 1 se hustota pravděpodobnosti nejvíc přizpůsobí úspěšně zvolené akci.

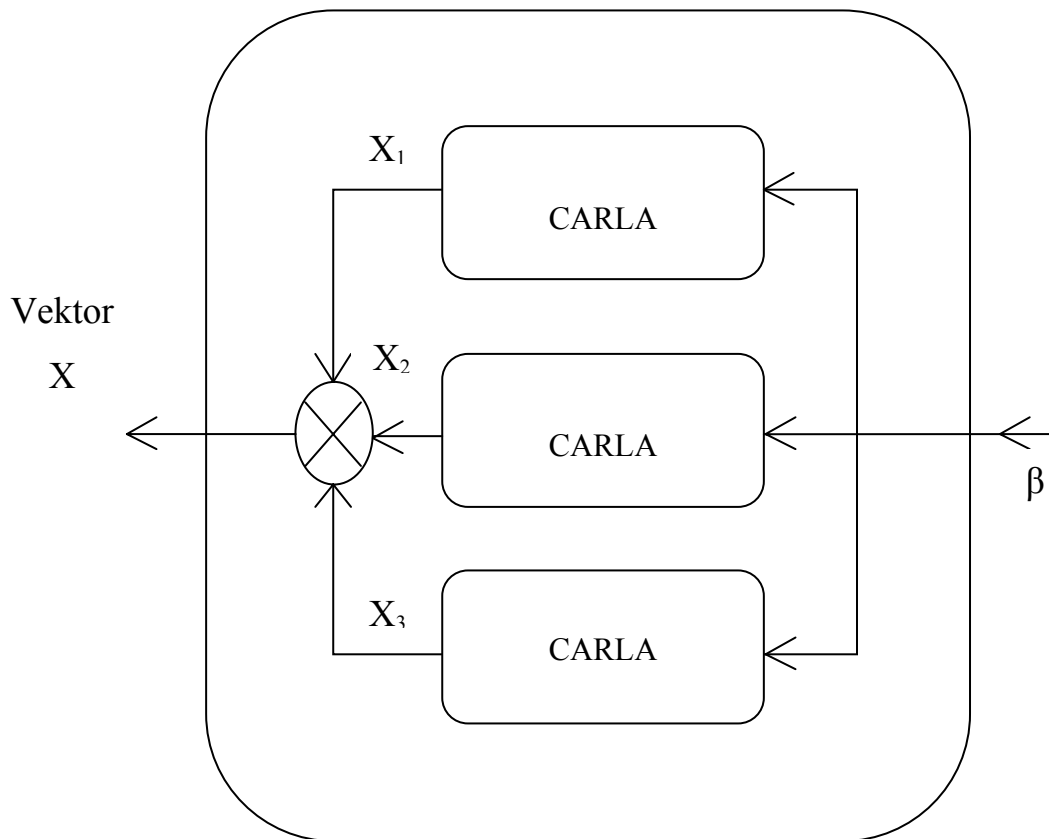
3 PRINCIP METODY CARLA

Metoda CARLA (Continuous Action Reinforcement learning automata) byla vyvinuta jako rozšíření diskretního náhodného učícího se automatu (discrete stochastic learning automata). Metoda CARLA nahrazuje diskretní prostor výběru akcí spojitým prostorem, a tím se více přibližuje inženýrským aplikacím, které jsou ze své podstaty spojité. Prakticky se vychází z diskretního učícího automatu. V diskretních uzlových bodech se vyčíslí hodnota distribuční funkce a mezi těmito body se použije metoda lineární interpolace. Metoda CARLA pracuje v interakci s obecně neznámým systémem pomocí náhodného výběru akcí. Učení spočívá ve zvýšení pravděpodobnosti výběru úspěšných akcí.



Obrázek 1: Typické uspořádání metody CARLA [1]

Metoda může mít mnoho paralelně zapojených instancí algoritmu, které pracují samostatně na nezávislých parametrech soustavy. Jediné propojení mezi instancemi je umožněno přes řízený výpočet ceny akce a následného posílení, které je pro všechny instance algoritmu společné. Uvnitř každé odpovídající instance je obsaženo rozložení pravděpodobnosti, které slouží pro náhodný výběr dalších akcí. Akce, které způsobí zlepšení chování systému, vyvolají vysoké posílení, a tedy i zvýšení pravděpodobnosti opakovaného výběru úspěšné akce. Toho je dosaženo úpravou rozložení pravděpodobnosti, konkrétně přičtením gaussovské funkce, centrované na úspěšnou akci. Během učení konverguje rozložení pravděpodobnosti k jednoduchému gaussovskému rozložení centrovanému na optimální hodnotu parametru.



Obrázek 2: Uspořádání učícího subsystému [1]

3.1 Metoda CARLA se člení na sedm základních částí

1. Inicializace rozložení pravděpodobnosti
2. Výběr akce
3. Aplikace vybrané akce na systém a čekání na její odezvu
4. Vypočet ceny akce
5. Výpočet posílení
6. Úprava rozložení pravděpodobnosti
7. Návrat ke kroku 2

3.2 Inicializace rozložení pravděpodobnosti

Rozložení pravděpodobnosti $f(x, 1)$ je inicializováno jako rovnoměrné v celém intervalu povolených akcí $\langle x_{min}; x_{max} \rangle$. Mimo interval povolených akcí je pravděpodobnost výběru akce nulová.

$$f(x, 1) = \begin{cases} \frac{1}{x_{max} - x_{min}} & \text{pokud } x \in \langle x_{min}; x_{max} \rangle \\ 0 & \text{jinak} \end{cases}, \quad (4)$$

kde $f(x, k)$ značí hustotu pravděpodobnosti v kroku k na vybrané akci x .

3.3 Výběr akce

Akce x_k je vybírána na základě náhodné hodnoty z_k z stejnoměrného rozložení z intervalu $\langle 0; 1 \rangle$ a aktuálního rozložení pravděpodobnosti $f(x, k)$. Integrál na levé straně rovnice vyjadřuje plochu pod křivkou hustoty pravděpodobnosti (distribuční funkci), a ten se musí rovnat náhodně vybrané akci z_k .

$$\int_{x_{min}}^{x_k} f(x, k) dx = z_k \quad (5)$$

3.4 Aplikace vybrané akce na systém a čekání na její odezvu

Vybraná akce se aplikuje na řízenou soustavu. Nyní je třeba počkat dostatečně dlouhou dobu na změnu chování soustavy. Je nutné, aby doba čekání byla větší, než je mechanická časová konstanta soustavy. Kámen úrazu může spočívat ve vhodném výběru způsobu ohodnocování chování.

3.5 Výpočet ceny akce

Cena akce J_k je skalární veličina, která ohodnocuje úspěšnost projeveného vlivu na soustavu. Čím je vliv akce na soustavu úspěšnější, tím je cena akce nižší. V praxi můžeme použít velké množství kriterií hodnocení. Nejčastěji se cena akce určuje jako integrál (suma) druhé mocniny rozdílu referenčního a dosaženého průběhu odezvy soustavy.

3.6 Výpočet posílení

Posílení β_k určuje, o kolik se zvýší pravděpodobnost opětovného výběru použité akce. Nula vyjadřuje, že aktuální akce je rovna nebo horší než je průměrná akce. Jednička znamená, že aktuální akce je nejlepší dosaženou akcí. Výpočet se provádí na základě aktuální ceny a charakteristik, získaných z historických cen podle vztahu.

$$\beta_k = \max \left\{ 0; \frac{J_k^{med} - J_k}{J_k^{med} - J_k^{min}} \right\}, \quad (6)$$

kde J_k^{med} je průměrná cena R posledních akcí.

$$J_k^{med} = \frac{\sum_{i=k-R}^k J_i}{R}, \quad (7)$$

kde J_k^{min} je minimální cena R posledních akcí

$$J_k^{min} = \min_{i=k-R}^k \{J_i\} \quad (8)$$

Ve výše uvedených vztazích (7), (8) vystupuje proměnná R, která určuje počet historických cen akcí, z nichž se uvedené charakteristiky zjišťují. Velikost proměnné R má vliv na rychlost konvergence metody. Obvyklá hodnota parametru R je 500 historických cen.

3.7 Úprava rozložení pravděpodobnosti

Na základě posílení se upraví rozložení pravděpodobnosti tak, aby byly častěji vybírány akce vedoucí ke zlepšení chování soustavy. Požadované úpravy dosáhneme přičtením gaussovské funkce centrované na poslední akci násobenou posílením. Tímto krokem je zajištěno, že na rozložení pravděpodobnosti mají vliv pouze úspěšné akce. Úpravu získáme při užití vztahu

$$f(x, k + 1) = \begin{cases} \alpha_k [f(x, k) + \beta_k H(x_k)] & \text{pokud } x \in \langle x_{min}; x_{max} \rangle \\ 0 & \text{jinak} \end{cases}, \quad (9)$$

kde hodnota α_k je volena tak, aby normalizovala rozložení pravděpodobnosti

$$\alpha_k = \frac{1}{\int_{x_{min}}^{x_{max}} (f(x, k) + \beta_k H(x_k)) dx}. \quad (10)$$

$H(x_k)$ je gaussovská funkce centrovaná na poslední akci x_k

$$H(x_k) = \frac{g_h}{x_{max} - x_{min}} \exp\left(-\frac{(x - x_k)^2}{2(g_w(x_{max} - x_{min}))^2}\right). \quad (11)$$

Konstanty g_h, g_w udávají šířku a výšku gaussovské funkce a zároveň mají vliv na rychlost a rozlišení učení. Jejich doporučené hodnoty jsou $g_h = 0,3$ a $g_w = 0,02$ (viz. [1]).

3.8 Výhody

- ✓ Metoda dokáže nalézt optimální řízení pro neznámé modely soustav.
- ✓ Velká robustnost metody. Při naučení je metoda dostatečně odolná proti aditivnímu šumu.

3.9 Nevýhody

- ✗ V každém kroku metody se musí provádět normalizace hustoty pravděpodobnosti, která je časově velice náročná.
- ✗ Při řízení nestabilních soustav je nutné vhodně zvolit interval hledaného řešení, přesto však není zaručena úspěšnost nalezení optimální akce.
- ✗ Vyčíslování hodnot hustoty pravděpodobnosti v mnoha uzlových bodech.
- ✗ Nepřesné doladění závislé na dané velikosti kroků mezi uzlovými body.
- ✗ Aby bylo nalezeno co nejpřesnější řešení, je třeba volit proměnnou vzdálenost uzlových bodů aproximace hustoty pravděpodobnosti podle rov. (9), čímž dochází k výraznému zpomalení řešení.
- ✗ Jakmile je nalezeno požadované řešení, opětovné přeučení probíhá pomalu. – Jiný pohled na tuto vlastnost je odolnost proti aditivnímu šumu (viz. výhody).

4 ÚPRAVA VZORCŮ A IMPLEMENTACE DO S-FUNKCE

Klasický postup metody, který byl popsán v kapitole 3 a byl úspěšně aplikován (viz. [1, 2, 3]) je zbytečně komplikovaný a není dostatečně efektivní. Existuje hned několik důvodů (viz. nevýhody).

Tyto nevýhody odstraňuje dále prezentovaná verze metody CARLA, která je přínosem této práce. Zlepšení je dosaženo pomocí úprav a zjednodušení vztahů. Jak již bylo řečeno dříve, metoda je založena na mnohonásobném opakování výběru akcí, oceňováním jejich úspěšnosti a zvyšování pravděpodobnosti výběru úspěšných zásahů na soustavu. K tomu CARLA využívá matematického aparátu teorie pravděpodobnosti. Pravděpodobnost můžeme popsat více způsoby, my ji budeme aproximovat relativní četností podle

$$n_{rel} = \frac{n_i + a}{n + a}, \quad (12)$$

kde n_i je četnost v i -tém uzlovém bodě, a představuje přírůstek četnosti a n značí celkovou četnost.

Pokud provedeme součet relativní četnosti přes celý definiční obor, v našem případě v definovaném intervalu $\langle x_{min}, x_{max} \rangle$, dostaneme absolutní četnost. Tato úprava je vyjádřena vzorcem (13), jako pro moment nultého řádu.

$$m_0 = \sum_{i=1}^m n_i \quad (13)$$

Výpočet hodnot momentů prvního a druhého řádů m_1 a m_2 se skrývá pod následujícími vztahy:

$$m_1 = \sum_{i=1}^m n_i * x_i, \quad (14)$$

$$m_2 = \sum_{i=1}^m n_i * x_i^2. \quad (15)$$

Normovaný moment prvního řádu m_{1n} vyjadřuje střední hodnotu a normovaný moment druhého řádu m_{2n} představuje rozptyl. Tyto momenty jsou určeny vztahy:

$$m_{1n} = m_1/m_0, \quad (16)$$

$$m_{2n} = m_2/m_0. \quad (17)$$

Další důležitou charakteristikou je směrodatná odchylka, která se určí jako druhá odmocnina z rozdílu rozptylu a druhé mocniny střední hodnoty podle následného vztahu

$$s_{2n} = \sqrt{m_{2n} - m_{1n}^2}. \quad (18)$$

Vliv přírůstku četnosti a na přírůstek momentů nultého až druhého řádu, vyjadřují vztahy:

$$m_{0,diff} = a, \quad (19)$$

$$m_{1,diff} = x_i * a, \quad (20)$$

$$m_{2,diff} = x_i^2 * a. \quad (21)$$

Z těchto jednoduchých vzorců vyplývá velmi důležitá a užitečná vlastnost. Jestliže chceme zjistit následující hodnotu distribuční funkce, přičteme k ní přírůstek četnosti, pro střední hodnotu přičteme přírůstek četnosti vynásobenou akčním zásahem a konečně pro rozptyl přírůstek četnosti hodnotu násobenou druhou mocninou akčního zásahu.

Nyní je nutné podrobněji analyzovat, jakým způsobem se určí přírůstek četnosti a . Přírůstek a v sobě musí zahrnovat hned několik vlastností, a to:

- Musí zohledňovat úspěšnost aplikované akce.
- Musí umožňovat volbu rychlosti učení.
- Musí zohledňovat velikost přírůstku k distribuční funkci.

S ohledem na tyto požadavky jsme zvolili výpočet přírůstku četnosti následovně

$$a = m_0 * \theta * \beta, \quad (22)$$

kde θ je parametr učení, β je posílení a m_0 je velikost distribuční funkce.

Nyní můžeme přistoupit k výběru následující akce. Akci zvolíme podle vztahu (23), a to náhodným výběrem se stejnoměrným rozložením z intervalu $\langle m_{1n} - s_{2n}; m_{1n} + s_{2n} \rangle$.

$$x_{i+1} = randn(m_{1n} - s_{2n}; m_{1n} + s_{2n}). \quad (23)$$

Cena je určena jako kvadrát rozdílu skutečné a referenční hodnoty dle vztahu

$$J = (y - y_{ref})^2, \quad (24)$$

kde y je skutečný výstup ze soustavy, y_{ref} je referenční výstup ze soustavy.

Následně je vypočten medián, a to pomocí klouzavého průměru. Konkrétně se jedná o průměr aktuální a minulé ceny akce

$$J_{med} = \frac{J + J_{i-1}}{2}. \quad (25)$$

Výpočet koeficientu posílení zůstává zachován dle vztahu (6).

Aby bylo možno zajistit přeučení, jsou v algoritmu zakotveny dva volitelné parametry - rozsah přestavení s_{2n} (spread rate) a perioda přestavení s_{2n} (spread period). Tyto parametry mají jediný účel, a to v pravidelných intervalech určených periodou přestavení zvětšit směrodatnou odchylku na hodnotu danou rozsahem přestavení a umožnit tím přeladění.

Větší podrobnosti a závislosti se dají vyčíst ze zdrojového souboru „carla.m“, který je uveden v přílohách na CD.

4.1 Implementace pod prostředí Matlab/Simulink

V zadání bakalářské byl popsán požadavek na funkčnost algoritmu pod prostředím Matlab/Simulink. Nadstavba Simulink je velmi vhodná pro simulování dynamických soustav. Stavební bloky, ze kterých se celý rozsáhlý systém skládá, jsou napsány v tzv. S-funkcích. Proto byl algoritmus implementován s využitím S-funkce. Větší náročnost na nastudování struktury a pochodů v S-funkci se následně mnohonásobně vrátí v rychlosti, jednoduchosti a snadné implementaci.

4.2 Zapojení bloku v Simulinku

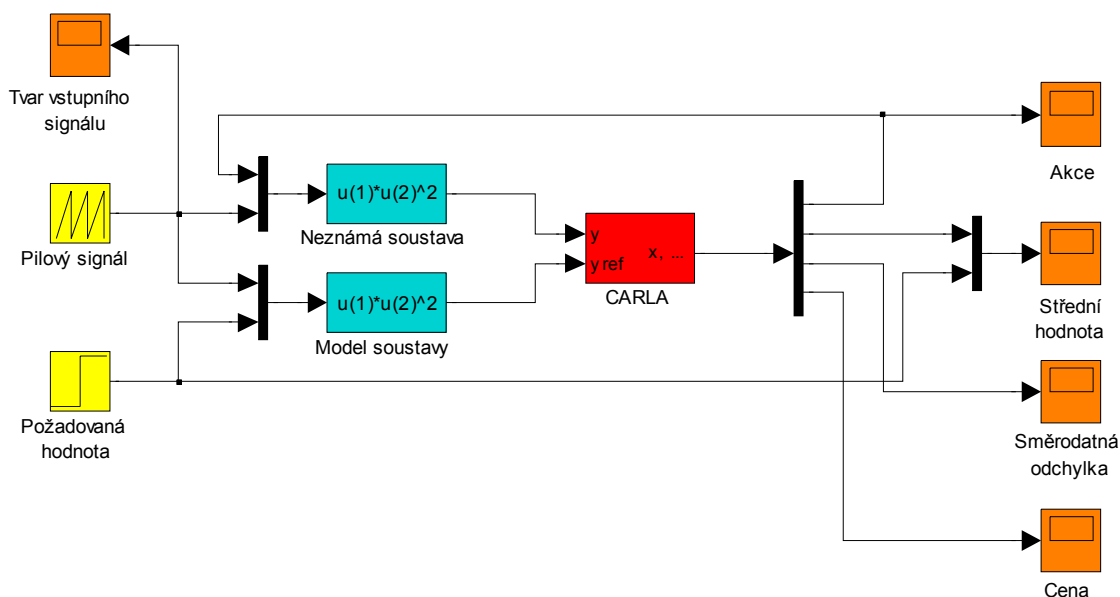
Celá aplikace CARLA je obsažena v jednom simulinkovském bloku se dvěma vstupy:

- y_{ref} referenční výstup,
- y skutečný výstup ze soustavy,

a čtyřmi výstupy:

- akce,
- střední hodnota,
- směrodatná odchylka,
- cena.

Model pro testování metody CARLA

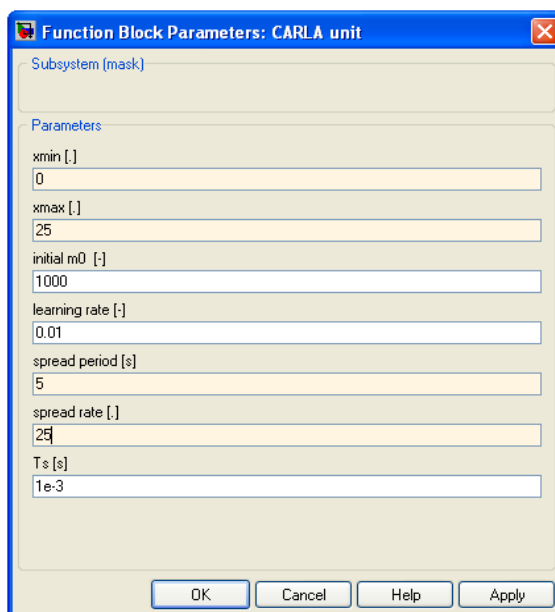


Obrázek 3: Zapojení CARLA pro zjištění vlivu volby ladících parametrů

Blok CARLA nastavuje hodnotu u_1 , což je parametr referenčního modelu soustavy.

4.3 Ladící parametry

Chování metody je možno ovlivňovat nastavením sedmi parametrů. Hodnoty parametrů lze nastavovat prostřednictvím masky subsystému v modelu CARLA.



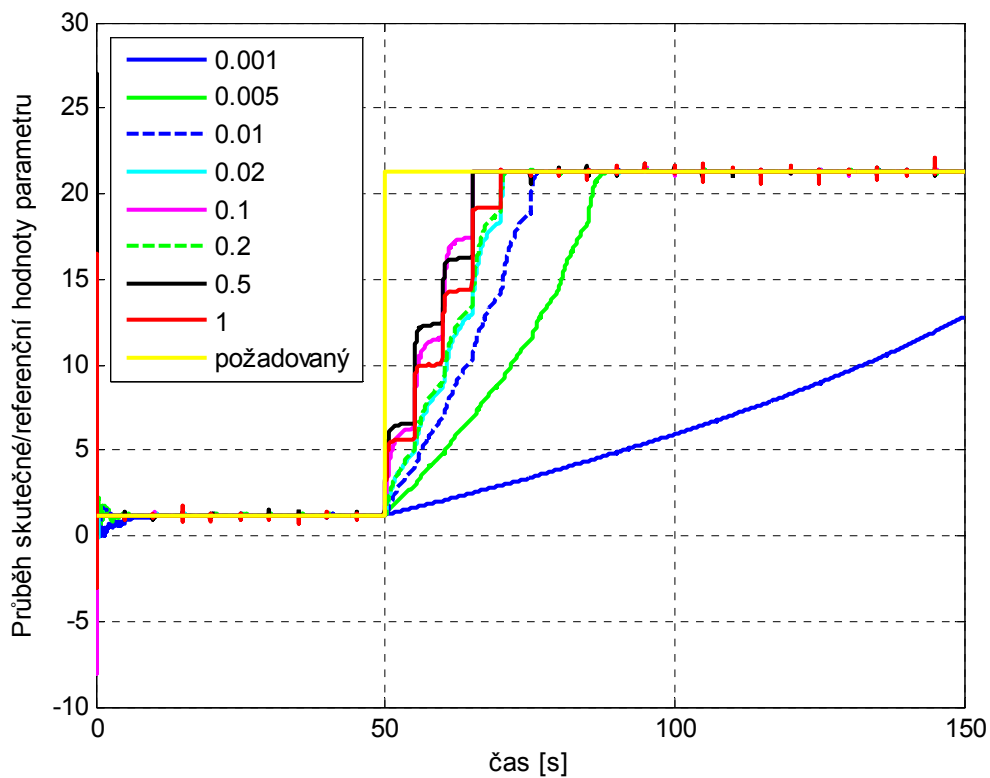
Obrázek 4: Maska bloku CARLA: volba ladících parametrů

Parametr	Funkce	Omezení
x_{min}	Dolní mez nastavovaného zisku	Omezení z řešené úlohy
x_{max}	Horní mez nastavovaného zisku	Omezení z řešené úlohy
Initial m_0	Počáteční četnost (m_0)	Velikost 1 – 100 000
Learning rate	Výrazně ovlivňuje rychlost nalezení hledaného řešení (θ)	Nabývá pouze hodnot z intervalu $\langle 0, 1 \rangle$
Spread period	Perioda přestavení s_{2n}	Větší než T_s Respektovat chování soustavy
Spread rate	Rozsah přestavení s_{2n}	-
T_s	Vzorkovací perioda	Rychlost a přesnost výpočtu

Tabulka 1: Funkce a omezení ladících parametrů

Algoritmus CARLA byl během vývoje testován na jednoduchém simulinkovském modelu (obr. 3). Z tohoto zapojení byla získána data, která jsou vykreslena na následujících grafech a ukazují závislost změny chování na volbě konkrétních velikostí parametrů.

4.3.1 Vliv parametru učení na proces učení

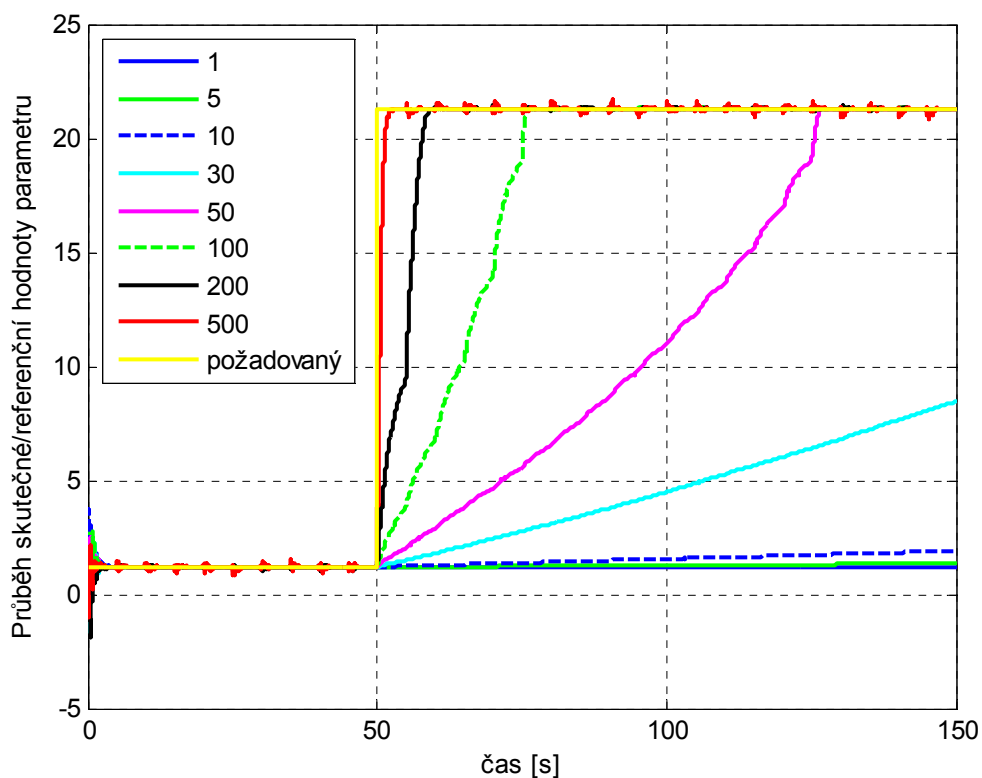


Graf 1: Vliv parametru učení na proces učení

Zhodnocení:

Parametr učení (learning rate) výrazně ovlivňuje rychlost změny hledaného řešení. Z matematického hlediska by měl nabývat hodnot z intervalu $\langle 0;1 \rangle$. Z grafu je patrné, že volba parametru vyšší než 0.005 dovoluje dostatečně rychlé přeladění. Dále je však patrné, že hodnoty parametry 0.5 a vyšší způsobují pravidelné rozkmitání od ustálené hodnoty. Na základě těchto výsledků se jako nejvhodnější velikost parametru jeví hodnoty v rozmezí 0.01 až 0.2. Tyto parametry rychle dosahují změny přeučení a zároveň mají malé odchylky od ustálené referenční hodnoty.

4.3.2 Vliv rozsahu přestavení na proces učení



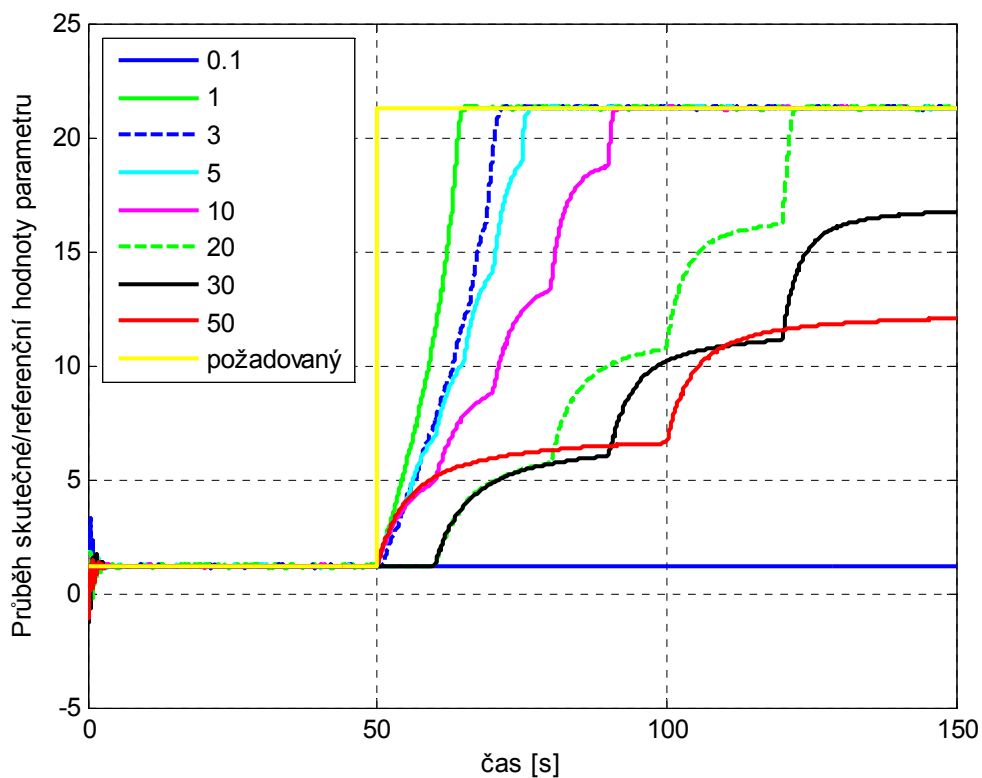
Graf 2: Vliv rozsahu přestavení na proces učení

Zhodnocení:

Další důležitý parametr „rozsah přestavení“ (spread rate), který výrazně ovlivňuje rychlost změny hledaného řešení a zároveň však rozkmitávání okolo ustálené hodnoty. Volba tohoto parametru je velmi provázaná s rozsahem hledávaného řešení. Výše uvedené závislosti byly určovány z intervalu $\langle -100, 100 \rangle$.

Z grafu vyplývá, že velmi dobrá volba parametru se nachází na hodnotách 200 a 100, které velikost rozsahu respektive polovičního rozsahu hledávaného řešení. Volba zbytečně velké hodnoty parametru způsobuje pravidelné výrazné rozlaďování od nalezeného řešení, což výrazně prodlužuje nalezení řešení u kmitavých a nestabilních soustav. Naproti tomu velmi malé nastavené hodnoty parametru zaručují konvergenci, ovšem až za velmi dlouhou dobu, která může být v praxi technicky i ekonomicky nepřijatelná. Za počáteční, hodnotu parametru, lze vzít hodnotu rovnou rozsahu hledávaného řešení.

4.3.3 Vliv periody přestavení na proces učení



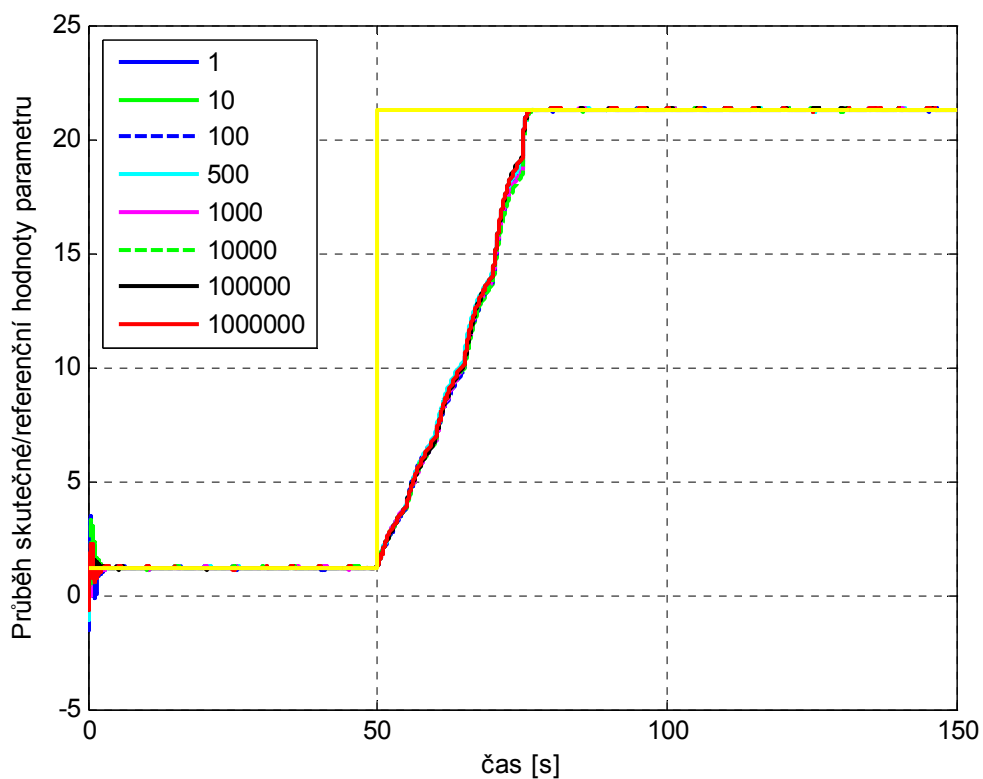
Graf 3: Vliv periody přestavení na proces učení

Zhodnocení:

Perioda přestavení (spread period) určuje, v jak velkých časových periodách bude zvětšena směrodatná odchylka na velikost, která je určena prostřednictvím rozsahu přestavení (spread rate). Velikost volby tohoto parametru výrazně ovlivňuje jak rychlost změny akce, tak také četnost rozlaďených akcí. Z grafu je vidět, že nejvhodnější volby z hlediska rychlosti leží mezi hodnotami 1s až 10s.

U průběhů s periodami 20s až 50s je patrné zpomalení učení. Takto velké periody budeme volit při učení stabilních kmitavých soustav, jako je například dvojitě kyvadlo. Z grafu je dále patrné, že volba velmi krátké periody 0.1s také není správná. Při výpočtu dochází tak často k pravidelnému rozladění, že metoda nestihne nalézt vhodnou akci s co nejmenší cenou. Na základě této analýzy lze za optimální volbu parametru považovat hodnotu 5 s. Tento průběh má velmi vysokou rychlost učení a přesto výrazně nereaguje na pravidelné rozladování.

4.3.4 Vliv počáteční četnosti na proces učení



Graf 4: Vliv počáteční četnosti na proces učení

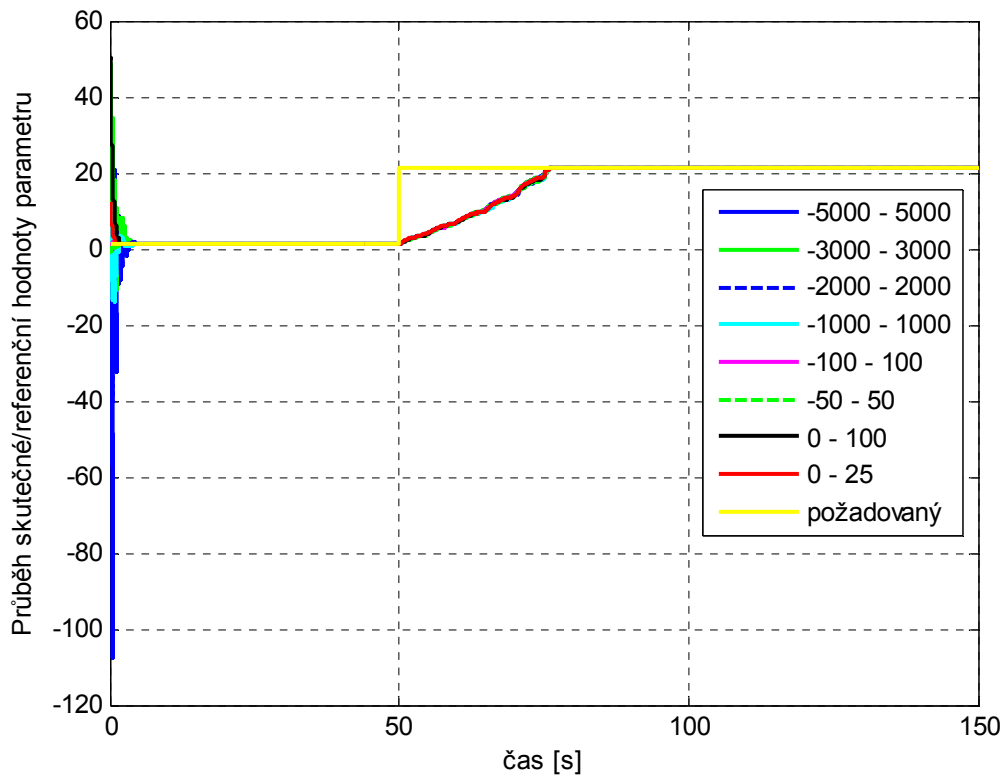
Zhodnocení:

Volba počáteční četnosti má pouze jedinou funkci, a to zajistit normalizaci výpočtu všech statistických charakteristik. Jak je znázorněno na grafu, volba parametru nijak výrazně neovlivňuje ani rychlost ani další jiné charakteristiky. Jako vhodná volba bude doporučena počáteční hodnota nastavení $m_0 = 1000$. K tomuto doporučení existují hned dva důvody.

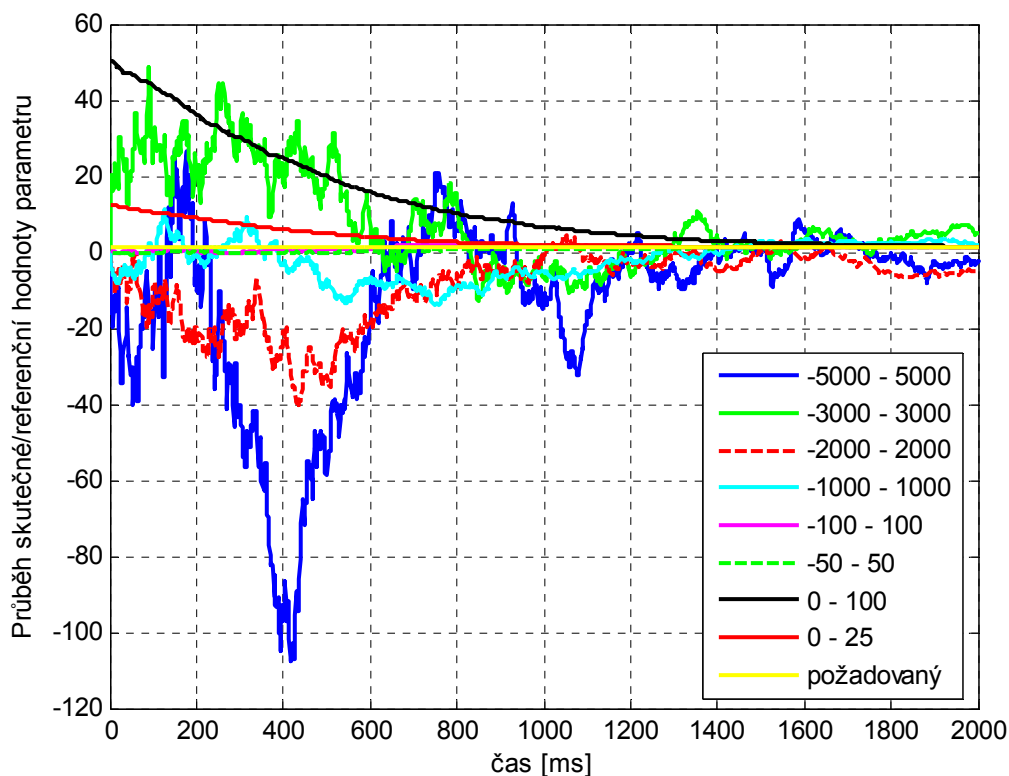
- při hodnotě $m_0 > 1e6$ je provedena výpočtová normalizace dělením číslem $1e6$, takže volba velmi vysokého m_0 zbytečně zpomaluje výpočet.
- v počátečních hodnotách učení je volbou dostatečně vysokého parametru m_0 zajištěn vhodný výběr další akce.

Z grafu 4 je zřejmé, že v prvních 5s nastal veliký rozkmit zeleného, modrého a červeného průběhu. Červený průběh pro $m_0 = 1e6$ je ale v programu dělen právě touto hodnotou na výslednou hodnotu 1. Z toho vyplývá, že nízká hodnota negativně ovlivňuje volbu akcí v prvních několika sekundách.

4.3.5 Vliv intervalu hledaného řešení na proces učení



Graf 5: Vliv intervalu hledaného řešení na proces učení



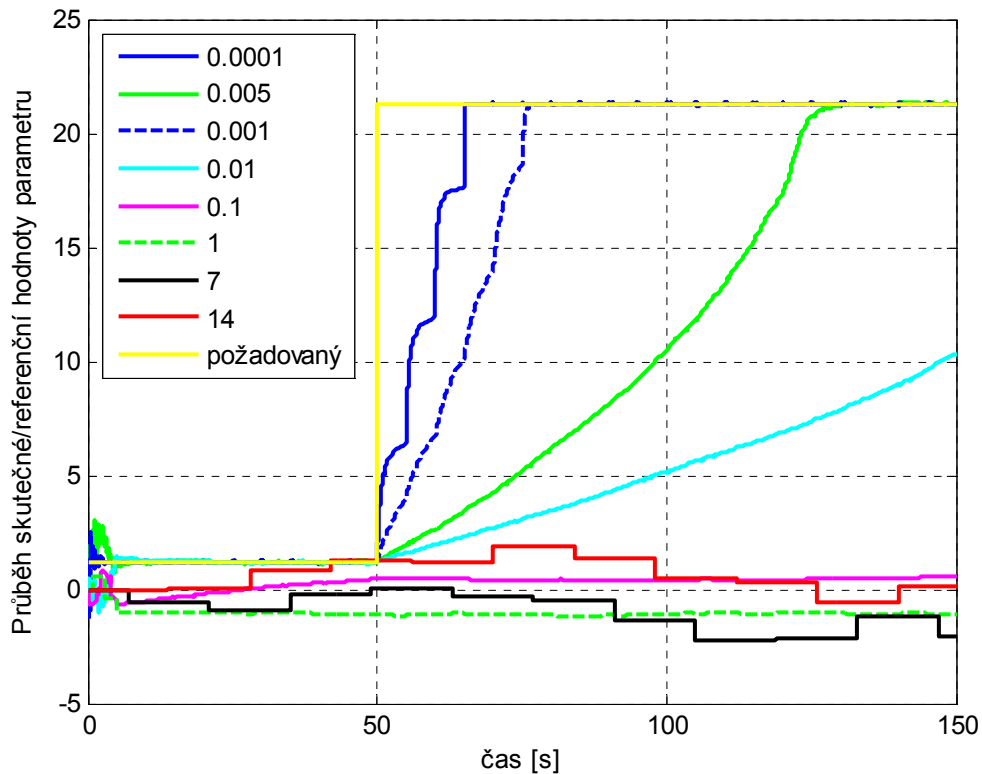
Graf 6: Detail vlivu intervalu hledaného řešení na proces učení (první 2s)

Zhodnocení:

Parametry vymezující rozsah hledaného řešení musíme volit velmi obezřetně a uváženě. Na grafu 5 je vidět, že pro všechny velikosti rozsahů je rychlost přeladění téměř totožná. Z výsledků tohoto grafu bychom se mohli mylně domnívat, že na volbě rozsahu nezáleží. Ale při volbě velkého, fyzicky nerealizovatelného rozsahu a aplikaci na neznámé fyzické soustavě může dojít k jejímu poškození. Naopak při volbě malého rozsahu, který neobsahuje hodnoty hledaného řešení, nebude optimální řešení nikdy nalezeno.

Na grafu 6 je znázorněn počáteční průběh nalezení správné hodnoty pro průběhy s velkými intervaly. Dále je třeba upozornit na fakt, že počáteční hodnota je vždy zvolena na střed zvoleného intervalu. Proto je vhodné, volit velikost intervalu centrovaného na střed očekávaného řešení. Nejvhodnější volbu intervalu představuje podle mého názoru interval $\langle -100, 100 \rangle$.

4.3.6 Vliv vzorkovací periody na proces učení



Graf 7: Vliv vzorkovací periody na proces učení

Zhodnocení:

Vzorkovací perioda T_s ovlivňuje, jak je patrné z grafu, rychlost přeladění a zároveň i výpočtovou náročnost. Z grafu 7 se dá dále vyčíst, že rozdíl v rychlosti se vzorkováním $1e-4$ a $1e-3$ je zanedbatelný a lze vylepšit jinými parametry. Naproti tomu doba výpočtu je 10krát větší. Při volbě malé vzorkovací periody konverguje algoritmus pomalu a při volbě hodnot větších než $1e-2$ přestává pracovat správně. Za vhodnou velikost vzorkovací frekvence lze považovat hodnotu 0.001. Nejlépe splňuje požadavky na přesnost a má minimální nároky na hardware.

4.3.7 Zhodnocení kapitoly:

Všechny nastavitelné parametry metody CARLA jsou navzájem provázány a ovlivňují se. Nalezení ideálního nastavení vyžaduje určitou dobu testování a zkušeností. Ukázané závislosti byly simulovány na velmi jednoduché soustavě. Při aplikaci na reálných soustavách, je nutné provádět volbu parametru velmi obezřetně s ohledem na charakter soustavy. Simulované soustavy je možné řešit jak v diskrétním, tak ve spojitém prostředí, ovšem nalezení řešení na diskretizovaném modelu je mnohem rychlejší.

4.4 Výhody

- ✓ Implementace pod prostředí Matlab/Simulink.
- ✓ Jednoduché nastavení parametrů učení.
- ✓ Velmi rychlý výpočet.
- ✓ Odstraněna náročná normalizace původní metody.

4.5 Nevýhody

- ✗ Nefunguje se zpožděním zapojeným mimo blok jednotky regulátoru CARLA.
- ✗ Nefunguje dobře na nestabilní soustavy.
- ✗ Závislost mezi rychlostí přeučení a stabilitou nalezené hodnoty – návrh na odstranění je popsán v závěru.

5 TESTOVÁNÍ CHOVÁNÍ NA TECHNICKÝCH SOUSTAVÁCH

5.1 Stejnosměrný motor (DC)

Výběr soustavy

Pro testování učícího regulátoru CARLA jsem zvolil stejnosměrný motor zcela záměrně. V technické praxi se s řízením motorů všech typů setkáváme každý den. Z hlediska teorie řízení se jedná o soustavu prvního řádu, a proto předpokládám, že by mělo řízení být dostatečně rychlé a přesné.

Sestavení dynamického matematického modelu DC motoru

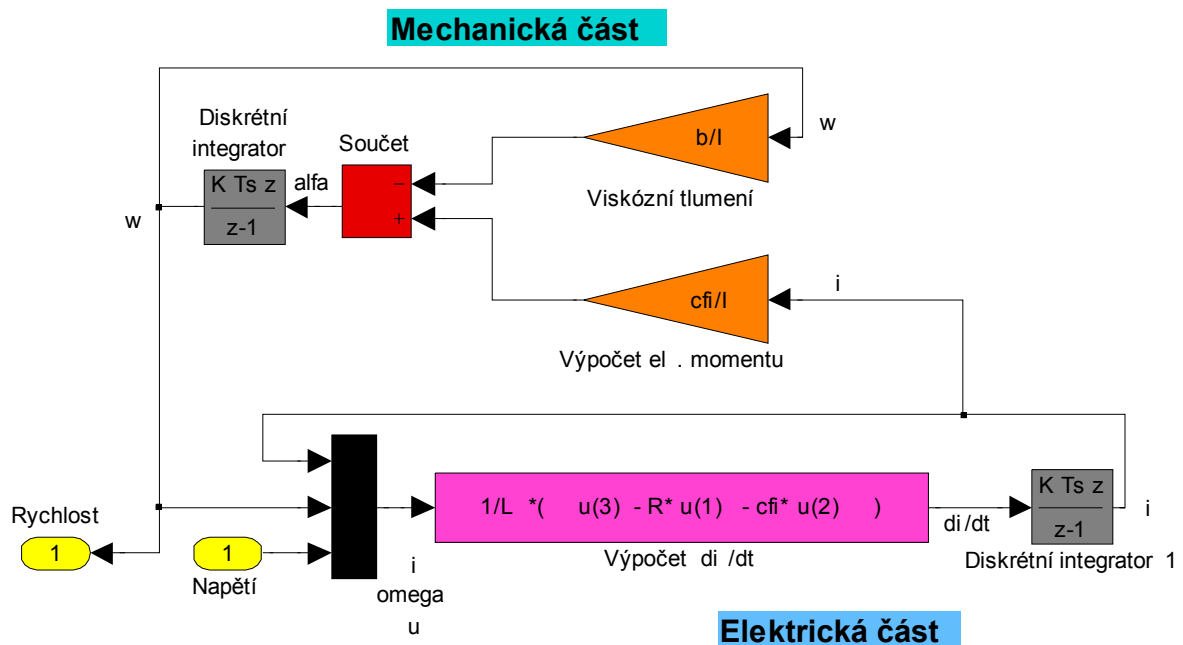
Poněvadž nemám k dispozici žádný reálný stejnosměrný motor, ani externí kartu pro řízení reálné soustavy, vytvořím si simulační model DC motoru v prostředí Simulink, na kterém řízení otestuju. DC motor je popsán dvěma rovnicemi, a to elektrickou a mechanickou. Elektrická rovnice popisuje závislost mezi změnou proudu ve vinutí a otáčkami rotoru. Mechanická rovnice pak určuje závislost mezi změnou rychlosti výstupní hřídele na elektrickém a mechanickém momentu motoru.

$$\text{Elektrická rovnice:} \quad \frac{di}{dt} = \frac{1}{L}(U - R \cdot i - c \cdot \Phi \cdot \omega) \quad (26)$$

$$\text{Mechanická rovnice:} \quad \frac{d\varpi}{dt} = \frac{c \cdot \Phi \cdot i}{I} - \frac{b}{I} \cdot \omega \quad (27)$$

Výsledná dynamická rovnice DC motoru je zobrazena pomocí zapojení bloku v Simulinku na následujícím obrázku 5.

5.1.1 Model DC motoru v Simulinku



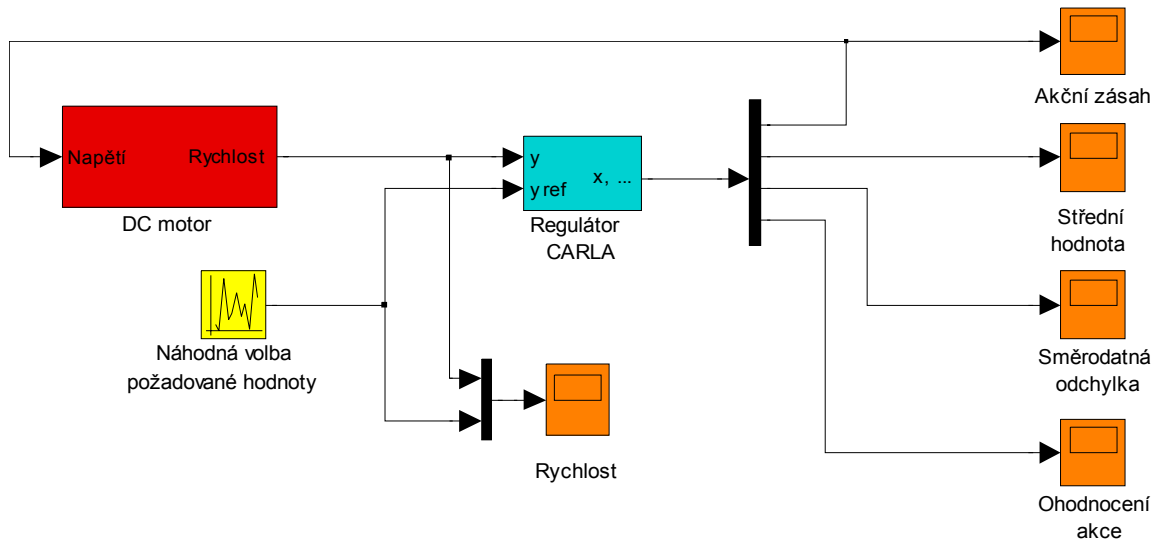
Obrázek 5: Model DC motoru

Diskretizace mechanické soustavy

Většina reálných dynamických soustav, které se v přírodě vyskytují, mají spojitou povahu (mají hladkou křivku závislosti na čase). Algoritmus metody CARLA je řešen jako diskrétní s možným nastavením velikosti konstantního kroku. Poněvadž řešení výpočtu v prostředí Matlab řešičem „VariableStepDiscrete“ je výrazně rychlejší a numericky stabilnější než řešení spojitých systémů řešičem „ode45“, je vhodné soustavy diskretizovat a následně řešit jako diskrétní. Při řešení velkých a rozsáhlých soustav to znamená velkou časovou úsporou. Přitom jsou dosažené výsledky dostatečně přesné. Pokud ovšem použijeme regulátor na reálnou fyzicky připojenou soustavu, bude učení řešičem „ode45“ probíhat také správně, musíme však počítat s vyššími časovými nároky na řešení.

5.1.2 Implementace soustavy s regulátorem CARLA

Zapojení stejnosměrného motoru s regulátorem CARLA



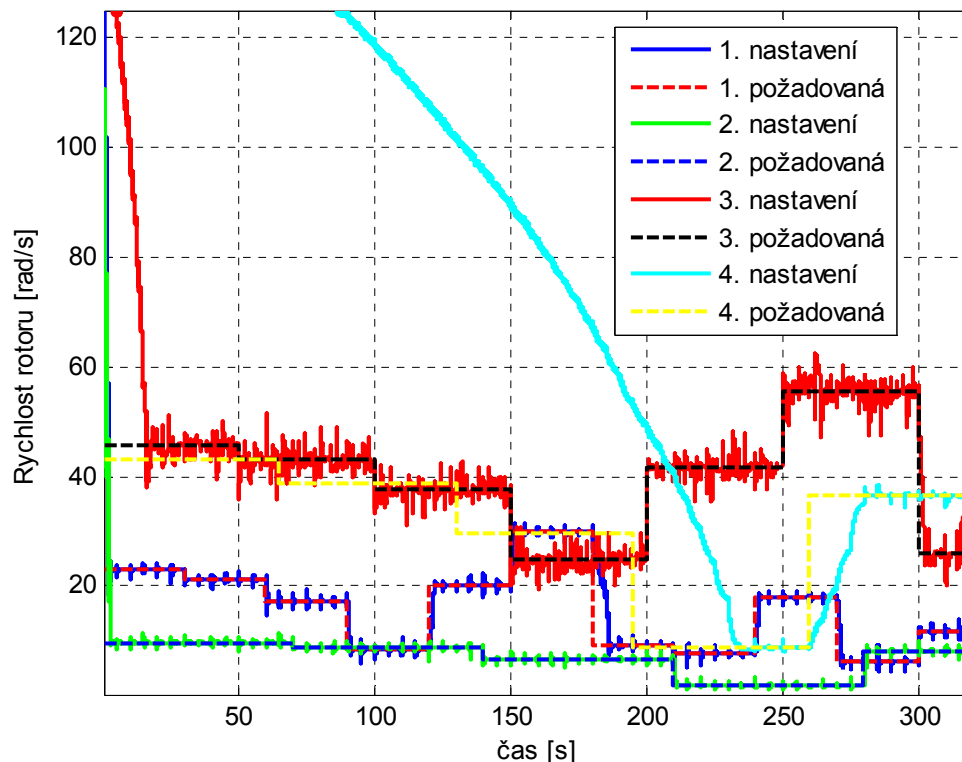
Obrázek 6: Zapojení DC motoru k regulátoru CARLA

Takto jednoduchá je implementace a použití metody pod prostředím Simulink. Výslednou závislost vstupní neznámé veličiny na požadované výstupní veličině nalezneme po rozkliknutí zobrazovacího bloku „Střední hodnota“. Tuto naměřenou závislost můžeme následně použít pro další zpracování a návrh jiného typu regulátoru.

	1. nastavení	2. nastavení	3. nastavení	4. nastavení
Rozsah	<0 - 150>	<0 - 100>	<0 - 150>	<0 - 100>
Learning rate	0.1	0.1	1	1
Spread period	5	5	2	5
Spread rate	150	100	150	50
Initial m0	1000	1000	1000	1000
Ts	0.001	0.001	0.001	0.001

Tabulka 2: Nastavení parametrů CARLA pro řízení DC motoru

5.1.3 Konvergence skutečné rychlosti DC motoru k referenční



Graf 8: Konvergence skutečné rychlosti DC motoru k referenční

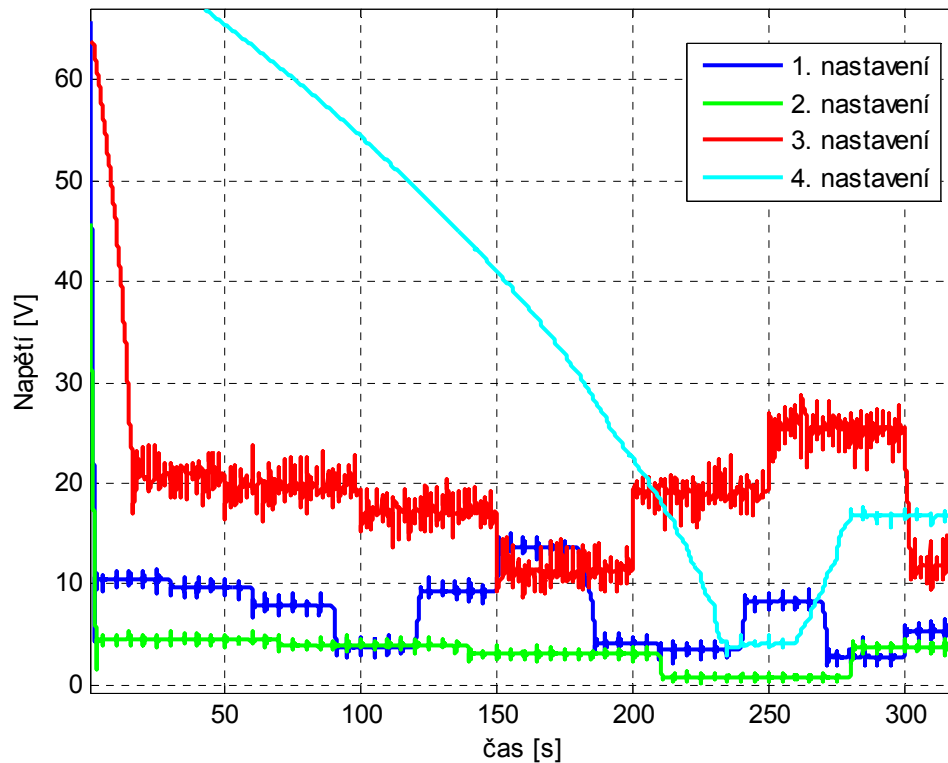
Zhodnocení:

Z grafu jasně vyplývá, že 1. – 3. nastavení výborně reaguje na požadované změny s dostatečně vysokou rychlostí, naproti tomu nastavení 4. dokáže sice také reagovat na změny, ale pomalá reakce mu znemožňuje použití na soustavách s rychlou a častou změnou otáček motoru. Další důležité kritérium je rozkmitávání okolo referenční hodnoty. Podle této podmínky musíme 3. nastavení vyřadit jako nevyhovující. Zbývá dvě nastavení jsou velmi podobná a pro určení vhodnějšího záleží na konkrétních požadavcích. Požadavky jsou:

- co nejmenší možný rozkmit,
- rychlost změny do 5 sekund.

Těmto kritériím lépe vyhovuje 2. nastavení metody. (na grafu zelený průběh)

5.1.4 Průběh řídicího napětí – střední hodnota

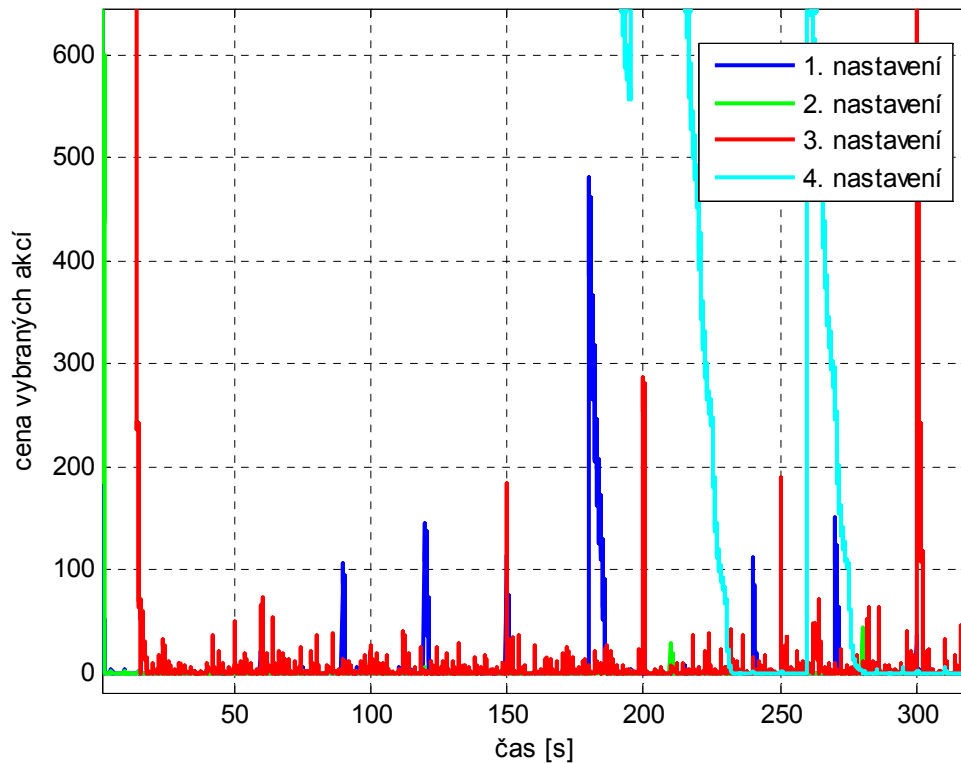


Graf 9: Průběh řídicího napětí – střední hodnota

Zhodnocení:

Graf s názvem „Průběh řídicího napětí - střední hodnota“ je pro nás ten nejdůležitější. Představuje charakteristickou závislost řídicí veličiny na požadovaném výstupu v čase. Srovnáním těchto dvou závislostí získáme matematickou závislost mezi vstupní a výstupní veličinou. Graf ukazuje průběh napětí v čase, tedy neznáme řídicí veličiny. Nastavení 3 a 4 jsou nevyhovující, z důvodů nestabilní nalezené hodnoty respektive pomalé konvergence. Nastavení 1 je vyhovující, avšak nastavení 2 je výhodnější a to z toho důvodu, že má menší odchylku okolo ustálené hodnoty a rychlejší reakci na změnu.

5.1.5 Cena vybraných akcií

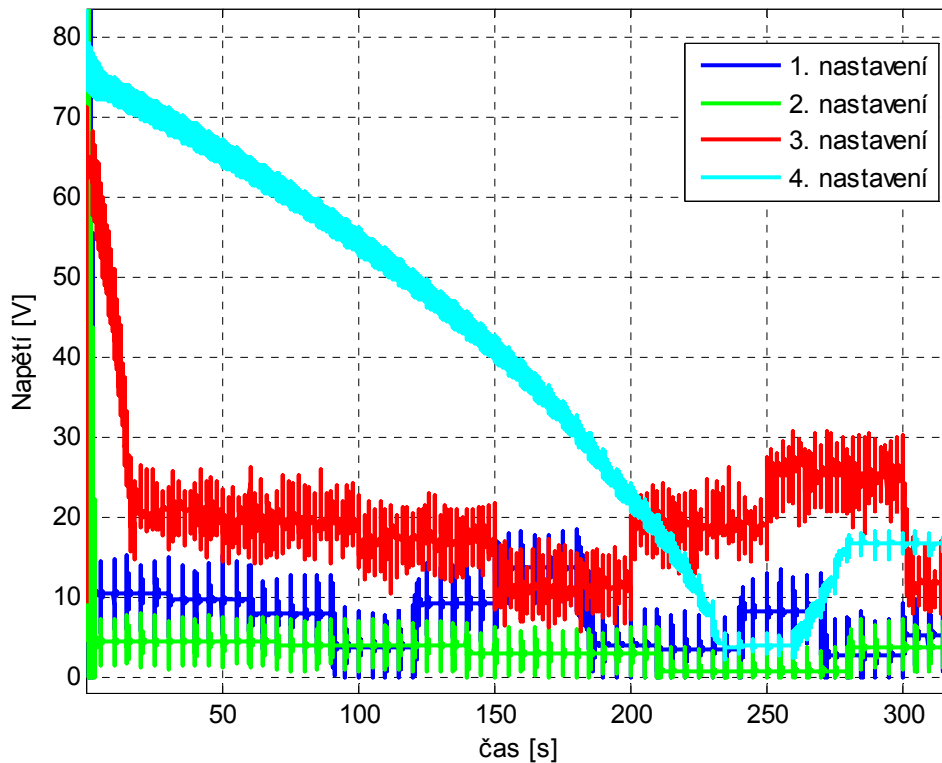


Graf 10: Cena vybraných akcií

Zhodnocení:

Graf vývoje ceny vyjadřuje matematické ohodnocení úspěšnosti provedené akce na soustavě. Z grafu č. 9 je zřejmé, že 4. nastavení je nevhodné, protože rychlost konvergence je velmi nízká. Další tři nastavení jsou si už více blízká. První nastavení má pomalejší konvergenci při změně, na grafu pozorovatelné časovou prodlevou k nalezení optimální akce. Třetí nastavení zase výrazněji zakmitává a znemožňuje použití pro velmi přesné stabilní řízení rychlosti. Jako nejlepší se tedy opět jeví druhé nastavení, které v sobě kombinuje rychlou změnu, výbornou konvergenci a stabilní požadovanou hodnotu.

5.1.6 Rozsah akčních zásahů



Graf 11: Rozsah akčních zásahů

Zhodnocení:

Z tohoto grafu je patrné, že velikost výběru akčních zásahů, má velký vliv na výsledné vlastnosti celého regulátoru CARLA. U 4. nastavení byla zvolena malá hodnota rozsahu přestavení (spread rate), což způsobilo pomalou konvergenci hodnot. Na druhou stranu, po nalezení řešení, je výsledná hodnota velmi stabilní a odolná proti aditivnímu šumu.

Pro 3. nastavení je rozsah přestavení nastaven až příliš velký a s kombinací s malou periodou přestavení (spread period) způsobuje, že nikdy není nalezena stabilní hledaná hodnota, a proto výstupní rychlost motoru výrazně osciluje okolo požadované hodnoty. Nastavení je výhodné pro rychle dynamicky se měnící děje, kde nepožadujeme velkou přesnost řízení. Nastavení 1. a 2. jsou v tomto grafu rovnocenná a nedokážeme určit lepší nastavení.

5.1.7 Směrodatná odchylka

V simulačním modelu je zobrazen blok pro snímání hodnoty směrodatné odchylky (rov. 18). Její graf zde není uveden, pro jeho nepřehlednost a malou vypovídací schopnost. Graf zobrazuje, jak se během učení mění interval hodnot, ze kterých je náhodně generovaná akční veličina.

5.1.8 Zhodnocení kapitoly

Z vyhodnocení jednotlivých grafů a uvážení požadavků na řízení stejnosměrného motoru docházím k závěru, že nejlepší vlastnosti má druhé nastavení. K tomuto výběru mě přesvědčily především:

- velmi rychlá reakce na změnu požadované rychlosti,
- minimální rozladění při nalezení správné řídicí veličiny,
- odchylka od ideálního výstupu je menší než 2% z rozsahu prohledávaného řešení.

5.2 Fyzikální kyvadlo

Výběr druhé soustavy

Jako druhou soustavu jsem zvolil model fyzikálního kyvadla. Z hlediska řízení se jedná o soustavu druhého řádu. Po vychýlení z rovnovážné polohy bude soustava okolo této pozice kmitat a působením tření se za určitou dobu opět ustálí na rovnovážné pozici. Pokud požadujeme výchylku kyvadla jinou, než v rovnovážné poloze, musíme kyvadlo řídit. Mezním případem je řízení inverzního kyvadla v horní nestabilní poloze. Taková soustava by se touto metodou nedala řídit anebo jen velice špatně.

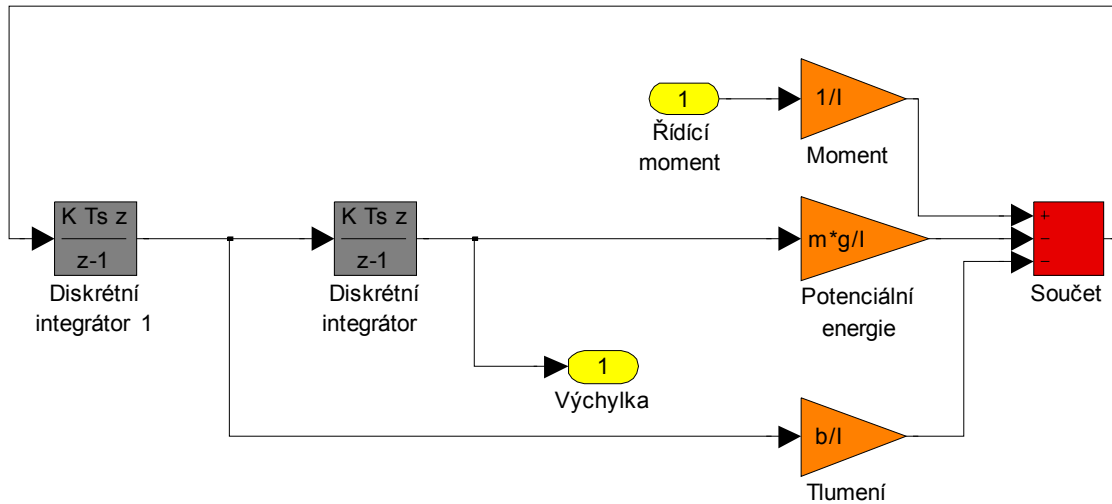
Sestavení matematického modelu kyvadla

Matematické kyvadlo je popsáno jednoduchou obyčejnou diferenciální rovnicí druhého řádu. Rovnici můžeme nalézt v literatuře, nebo si ji odvodit pomocí známých metod dynamiky tuhých těles. Rovnici jsem určil pomocí energického přístupu Euler-Lagrangeových rovnic. Výsledná rovnice vypadá následovně

$$\ddot{\varphi} = \frac{M}{I} - \frac{b}{I} \cdot \dot{\varphi} - \frac{m \cdot g}{I} \cdot \varphi. \quad (28)$$

5.2.1 Model kyvadla v Simulinku

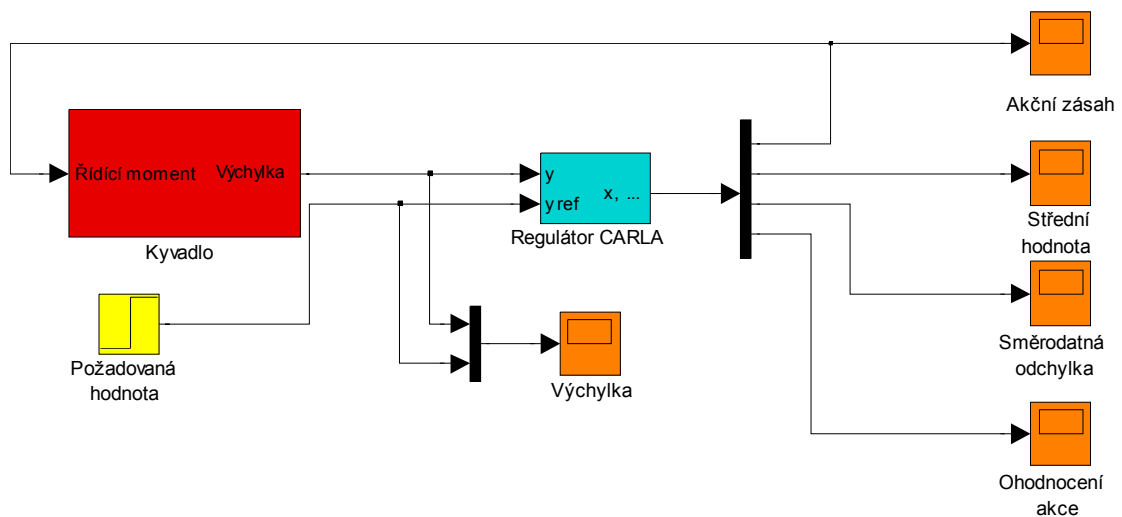
Mechanické kyvadlo



Obrázek 7: Model fyzikálního kyvadla

5.2.2 Implementace soustavy s regulátorem CARLA

Zapojení kyvadla s regulátorem CARLA

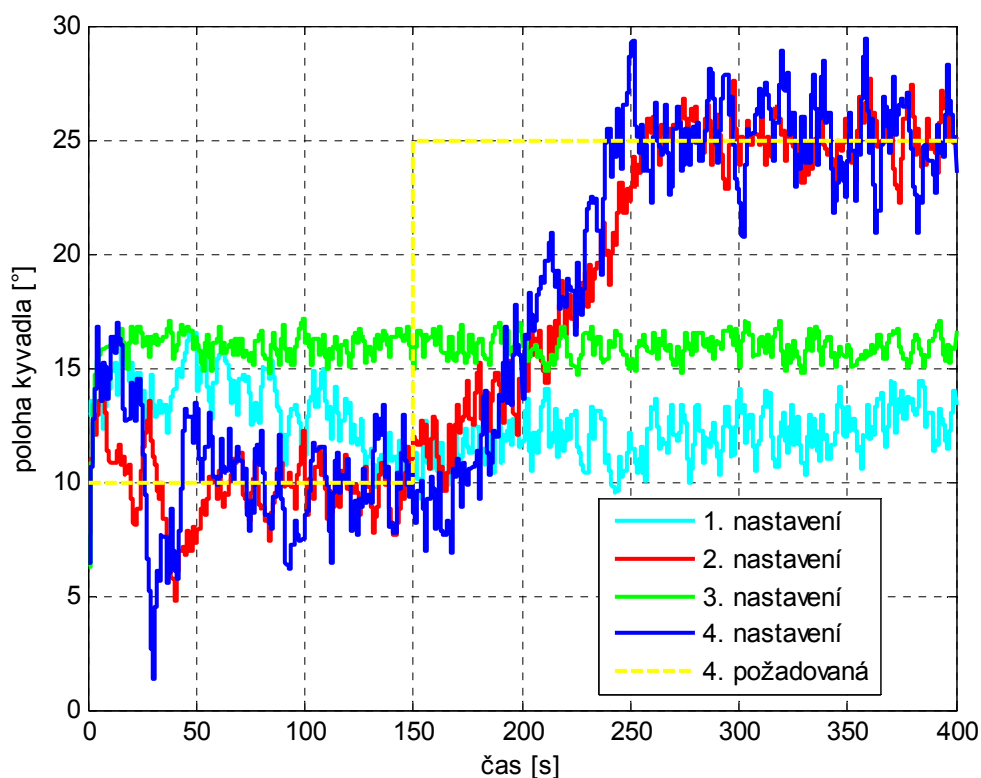


Obrázek 8: Zapojení kyvadla k regulátoru CARLA

	1. nastavení	2. nastavení	3. nastavení	4. nastavení
Rozsah	< -100 - 350 >	< -100 - 350 >	< 0 - 300 >	< 0 - 300 >
Learning rate	0.1	1	0.01	1
Spread period	5	2	10	2
Spread rate	900	1000	500	1500
Initial m0	1000	1000	1000	1000
Ts	0.001	0.001	0.001	0.001

Tabulka 3: Nastavení parametrů CARLA pro řízení kyvadla

5.2.3 Konvergence skutečné polohy kyvadla k referenční



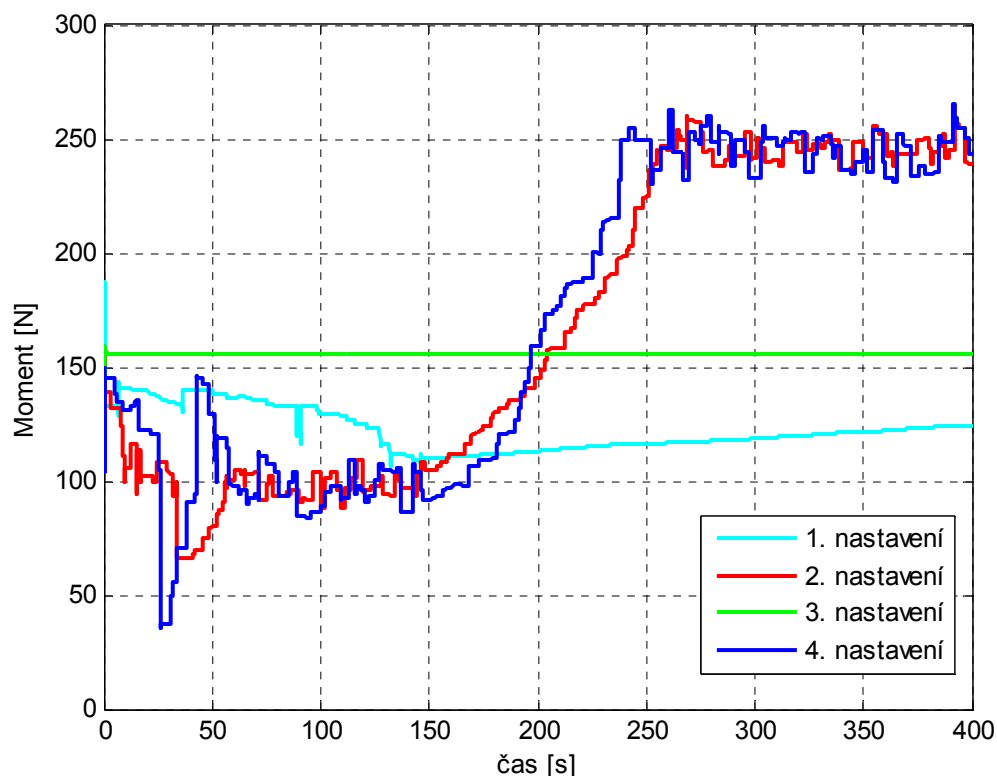
Graf 12: Konvergence skutečné polohy kyvadla k referenční

Zhodnocení:

Na grafu 12 je znázorněna rychlost konvergence na požadovanou polohu kyvadla pro různá nastavení parametrů metody CARLA. Nastavení 1. a 3. jsou zcela nevyhovující. U třetího nastavení k učení téměř vůbec nedochází, což je způsobeno velkou volbou parametru spread period a malou volbou parametru spread rate. Nastavení 2. a 4. už následují požadovanou hodnotu, a to i s dostatečně velkou

rychlostí. Nejlepší vlastnosti má druhé nastavení, a to především díky menší odchylce od požadované polohy.

5.2.4 Průběh řídicího momentu – střední hodnota

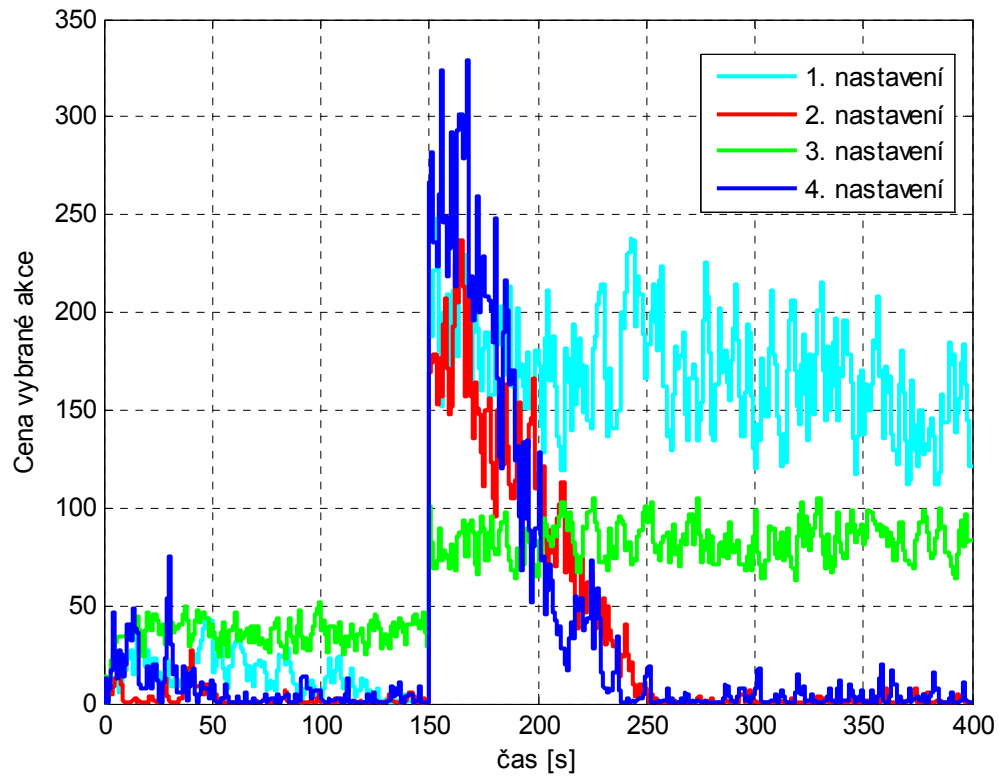


Graf 13: Průběh řídicího momentu – střední hodnota

Zhodnocení:

Tento graf zobrazuje hledanou závislost neznámé řídicí veličiny. Nastavení 1. a 3. opět jsou zcela nevyhovující (viz. graf 12). Nastavení 2. a 4. jsou si velmi podobná a dají se úspěšně použít. Z tohoto grafu usuzujeme na nejlepší chování druhého nastavení, které nalézá skutečnou hodnotu momentu sice pomaleji, ale zákmity okolo ustálené hodnoty jsou menší než pro nastavení čtvrté. Přesto jsou však velmi výrazné, což způsobuje problémy v přesném řízení na udržení požadované polohy kyvadla. Odstranění tohoto nedostatku vyžaduje určitý kompromis mezi rychlostí přechodu a udržení konstantní nalezené hodnoty. Se snižováním rozsahu přestavení mnohonásobně roste doba potřebná k naučení regulátoru. Tato nepříjemná vlastnost je společná pro všechny oscilující soustavy.

5.2.5 Cena vybraných akcií

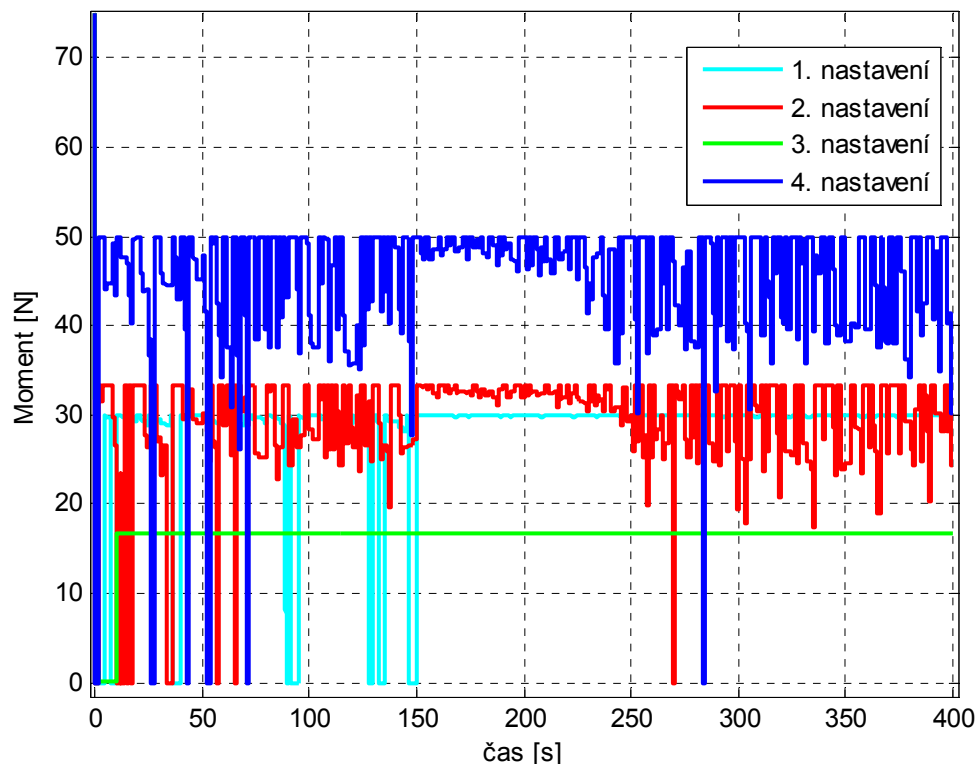


Graf 14: Cena vybraných akcií

Zhodnocení:

Z grafu 14 můžeme dojít opět ke stejným závěrům, jako na předešlých grafech. Je zde vidět, že čtvrté nastavení má rychlejší nalezení správné akce, ovšem za cenu malé stability řešení. Požadavkům proto opět nejlépe vyhovuje druhé nastavení.

5.2.6 Směrodatná odchylka



Graf 15: Velikost směrodatné odchylky

Zhodnocení:

Na grafu 15 je znatelný rozdíl mezi modrým a červeným průběhem. Modrý graf volí akční zásahy s 1,5krát větší odchylkou. Toto je také hlavní důvod, proč modrý graf hůře udržuje referenční polohu kyvadla.

5.2.7 Zhodnocení kapitoly

Nalezení optimálního nastavení metody pro úlohu mechanického kyvadla je problematictější, než pro DC motor. Nejúspěšnější nastavení metody mělo rozsah přestavení (spread rate) přibližně dvakrát větší, než byl rozsah hledaného řešení. Takto vysoká hodnota byla volena záměrně s požadavkem na co nejrychlejší změnu. Protože je soustava svým charakterem kmitavá, musíme opatrně volit periodu přestavení. Pokud ji zvolíme na složité soustavě moc malou, nemusí být řešení nikdy nalezeno.

6 ZÁVĚR

Cílem bakalářské práce bylo navrhnout a simulačně ověřit dva řídicí členy pomocí metody CARLA. Tohoto cíle bylo dosaženo.

Během testování jsem došel jak k nesporným výhodám, tak k nevýhodám metody. Tato metoda je velmi jednoduchá a snadná na použití. Implementace pod průmyslově rozšířené prostředí Matlab/Simulink ji umožňuje široké uplatnění v průmyslu. Metoda nevyžaduje téměř žádnou vnitřní znalost soustavy. Parametry, které regulátoru musíme zadat, jsou snadno zjistitelné anebo se volí podle pozorovaného chování.

Výpočetní, časové a ekonomické náklady na nalezené řešení neznámé soustavy jsou přímo závislé na množství informací o soustavě a na její složitosti. Pro velké soustavy typu podvozek automobilu, CNC stroje, systémy letadel, atd. může trvat nalezení řešení i několik hodin až dní. Přesto je to zanedbatelný čas s porovnáním doby trvání při použití jiných přístupů. Metoda nalezení ideálního nastavení je omezena dvěma protichůdnými požadavky, a to:

- velká rychlost nalezeného řešení,
- vysoká přesnost a stabilita řešení.

Tento nedostatek půjde odstranit úpravou algoritmu a zavedením přeladování pouze, pokud se cena akce zvyšuje, anebo pokud je větší než nastavitelná konstanta. Tímto nejspíš dojde k odstranění největší nevýhody algoritmu, kterou jsem při testování zjistil.

Další nevýhodou je, že metoda posilovaného učení není vhodná pro řízení nestabilních soustav. Tato nevýhoda je však také řešitelná, ale již vyžaduje jisté speciální přístupy, jako je například předučení na modelu soustavy a následné doučení na reálné soustavě. [9]

SEZNAM POUŽITÉ LITERATURY

- [1] HOWELL, M. N., GORDON, T. J.: *Continuous action reinforcement learning automata and their application to adaptive digital filter design*. Engineering Applications of Artificial Intelligence, 14, 2001, 549-561.
- [2] HOWELL, M. N., FROST, G. P., GORDON, T. J., WU, Q. H.: *Continuous action reinforcement learning applied to vehicle suspension control*. Mechatronics, 7, 1997, 263-276.
- [3] HOWELL, M. N., BEST, M. C.: *On-line PID tuning for engine idle-speed control using continuous action reinforcement learning automata*. Control Engineering Practice 8, 2000, 147-154.
- [4] NARENDRA, K.S., THATHACHAR, M.A.L.: *Learning Automata*. An Introduction, Prentice Hall, London, 1989.
- [5] FROST, G.P., GORDON, T.J., HOWELL, M.N., WU, Q.H. *Reinforcement learning of active and semi-active vehicle suspension control laws*. Proceedings of the Institution of Mechanical Engineers Part I 210, 1996, 249–257.
- [6] BALÁTĚ, J. Automatické řízení, 2. přepracované vydání. Brno: BEN, 2004. 664 s. ISBN 80-7300-148-9.
- [7] BOBÁL, V., BÖHM, J., PROKOP, R., FESSL, J. *Praktické aspekty samočinně se nastavujících regulátorů: algoritmy a implementace*. Brno: Nakladatelství VUTIUM, 1999. 242 s. ISBN 80-214-1299-2.
- [8] ŠVARC, I., ŠEDA, M., VÍTEČKOVÁ, M.: *Automatické řízení*. Brno: Akademické nakladatelství CERM, 2007. 324 s. ISBN 978-80-314-3491-2.
- [9] BŘEZINA, T.: *Efektivní metoda Q učení: simulační posouzení použitelnosti pro řízení aktivního magnetického ložiska - zkrácená verze*. Brno: VUTIUM, 2003. 35 s. ISBN 80-214-2414-1.
- [10] MITCHELL, T.: *Machine Learning*. Morgan Kaufmann, 1997.
- [11] *Artificial neural network*. [online]. Dostupné z http://en.wikipedia.org/wiki/Artificial_neural_network. Převzato 10. 4. 2008.
- [12] *Genetic algorithm*. [online]. Dostupné z http://en.wikipedia.org/wiki/Genetic_algorithm. Převzato 10. 4. 2008.

- [13] Sutton, R., S., Barto, A., G., “*Reinforcement Learning: An Introduction,*” MIT Press (Bradford Book), Cambridge, Mass., 1998, [online]. Dostupné z <http://www.cs.ualberta.ca/~sutton/book/the-book>. Převzato 10. 4. 2008.
- [14] KLÉMA, J.: *Strojové učení v diagnostice a při predikci časových řad: Praha: ČVUT, 2002. 8 s.*
- [15] SANTHARAM, G., SASTRY, P. S., THATHACHAR, M. A. L.: *Stochastic optimization with continuous action set learning automata*, Technical Report, Dept. of Electrical Engng., Indian Institute of Science, Bangalore 560012, India, 1995.

SEZNAM POUŽITÝCH SYMBOLŮ

$u(t)$	Akční veličina PID regulátoru v čase t
K_p	Koeficient proporcionální složky PID regulátoru
K_i	Koeficient integrační složky PID regulátoru
K_d	Koeficient derivační složky PID regulátoru
$e(t)$	Regulační odchylka PID regulátoru v čase t
$p_i(n)$	Pravděpodobnost v i -tém uzlu v současné iteraci výpočtu u diskrétního náhodného učícího automatu
$p_i(n + 1)$	Pravděpodobnost v i -tém uzlu v následující iteraci výpočtu u diskrétního náhodného učícího automatu
θ	Parametr učení
$\beta(n)$	Parametr posílení v n -té iteraci u diskrétního náhodného učícího automatu
$f(x, 1)$	Výchozí hodnota funkce hustoty pravděpodobnosti v první iteraci metody
$f(x, k)$	Funkce hustoty pravděpodobnosti v x -tém bodě a v k -té iteraci metody
x_{min}	Dolní mez povolených akcí metody CARLA
x_{max}	Horní mez povolených akcí metody CARLA
x_k	Náhodně vybraná akce v k -té iteraci metody
z_k	Náhodně vygenerovaná akce ze stejnoměrného rozložení $U(0;1)$ v k -té iteraci metody
J_k	Cena akce v k -té iteraci metody
β_k	Posílení v k -té iteraci metody
J_k^{med}	Průměrná cena akcí v k -té iteraci metody
J_k^{min}	Minimální cena akcí v k -té iteraci metody
R	Počet hodnot pro výpočet průměrné a minimální ceny
α_k	Koeficient normalizující hustotu pravděpodobnosti v k -té iteraci metody
$H(x_k)$	Gaussovská funkce centrovaná na k -tou akci
g_h	Koeficient udávající výšku gaussovské funkce
g_w	Koeficient udávající šířku gaussovské funkce
$f(x, k + 1)$	Funkce hustoty pravděpodobnosti v následující iteraci metody
n_{rel}	Relativní četnost

n_i	Četnost v i -tém uzlu
n	Absolutní četnost
a	Přírůstek četnosti
m_0	Absolutní četnost, distribuční funkce
m_1	Moment prvního řádu
m_2	Moment druhého řádu
m_{1n}	Střední hodnota
m_{2n}	Rozptyl okolo střední hodnoty
x_i	Akce působící na systém v i -té iteraci
s_{2n}	Směrodatná odchylka od střední hodnoty
$m_{0,diff}$	Přírůstek k distribuční funkci
$m_{1,diff}$	Přírůstek k momentu prvního řádu
$m_{2,diff}$	Přírůstek k momentu druhého řádu
β	Posílení
x_{i+1}	Výběr akce působící v další iteraci na soustavu
J	Cena
y	Skutečný výstupní hodnota z řízené soustavy
y_{ref}	Referenční hodnota řízené soustavy
J_{med}	Průměrná cena akcí
J_{i-1}	Cena minulé akce

SEZNAM GRAFŮ, TABULEK A OBRÁZKŮ

Graf 1: Vliv parametru učení na proces učení	25
Graf 2: Vliv rozsahu přestavení na proces učení	26
Graf 3: Vliv periody přestavení na proces učení	27
Graf 4: Vliv počáteční četnosti na proces učení	28
Graf 5: Vliv intervalu hledaného řešení na proces učení	29
Graf 6: Detail vlivu intervalu hledaného řešení na proces učení (první 2s)	30
Graf 7: Vliv vzorkovací periody na proces učení	31
Graf 8: Konvergence skutečné rychlosti DC motoru k referenční	36
Graf 9: Průběh řídicího napětí – střední hodnota	37
Graf 10: Cena vybraných akcí	38
Graf 11: Rozsah akčních zásahů	39
Graf 12: Konvergence skutečné polohy kyvadla k referenční	42
Graf 13: Průběh řídicího momentu – střední hodnota	43
Graf 14: Cena vybraných akcí	44
Graf 15: Velikost směrodatné odchylky	45
Tabulka 1: Funkce a omezení ladících parametrů	25
Tabulka 2: Nastavení parametrů CARLA pro řízení DC motoru	35
Tabulka 3: Nastavení parametrů CARLA pro řízení kyvadla	42
Obrázek 1: Typické uspořádání metody CARLA	16
Obrázek 2: Uspořádání učícího subsystému	17
Obrázek 3: Zapojení CARLA pro zjištění vlivu volby ladících parametrů	24
Obrázek 4: Maska bloku CARLA: volba ladících parametrů	24
Obrázek 5: Model DC motoru	34
Obrázek 6: Zapojení DC motoru k regulátoru CARLA	35
Obrázek 7: Model fyzikálního kyvadla	41
Obrázek 8: Zapojení kyvadla k regulátoru CARLA	41

SEZNAM PŘÍLOH NA CD

Příloha 1: Algoritmus CARLA

Příloha 2: Model DC motoru

Příloha 3: Model kyvadla