VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta strojního inženýrství

RNDr. Ing. Tomáš Březina, CSc.

EFEKTIVNÍ METODA Q UČENÍ: SIMULAČNÍ POSOUZENÍ POUŽITELNOSTI PRO ŘÍZENÍ AKTIVNÍHO MAGNETICKÉHO LOŽISKA

EFFICIENT Q-LEARNING METHOD: SIMULATION ASSESSMENT OF ITS APPLICABILITY ON ACTIVE MAGNETIC BEARING CONTROL

ZKRÁCENÁ VERZE HABILITAČNÍ PRÁCE

VUTIUM

BRNO 2003

KLÍČOVÁ SLOVA
opakovaně posilované učení, Q-učení, aktivní magnetické ložisko, řízení

KEYWORDS
Reinforcement learning, Q-learning, Active Magnetic Bearing, Control

MÍSTO ULOŽENÍ PRÁCE
Oddělení pro vědu a výzkum Fakulty strojního inženýrství Vysokého učení technického v Brně

ISBN: 80-214-2414-1
ISSN: 1213-418X

OBSAH

PŘEDSTAVENÍ AUTORA HABILITAČNÍ PRÁCE

Tomáš Březina se narodil v roce 1954 v Brně. V roce 1978 absolvoval FE VUT v Brně, obor slaboproud, v roce 1985 absolvoval PřF UJEP Brno, obor matematická informatika a získal titul RNDr. V roce 1985 rovněž obhájil disertační práci v oboru radioelektronika a získal titul CSc.

Od roku 1979 pracuje na FS VUT v Brně. Nejdříve na katedře obráběcích a tvářecích strojů (výrobních strojů a průmyslových robotů) jako asistent, později samostatný asistent a odborný asistent. Od r. 1986 do r. 1991 zastával funkci vedoucího fakultního výpočetního střediska. Od r. 1991 odborného asistenta na katedře informatiky a od 1994 na Ústavu automatizace a informatiky. Od r. 1983 pedagogicky působí v předmětech zaměřených na výpočetní techniku.

Přednášky a cvičení vykonával na FSI dosud v těchto předmětech: Konstrukce a výpočetní technika, Teorie automatů, Základy numerické matematiky a programování, Algoritmy a programování, Základy výpočetní techniky, Databáze, Informatika I, Informatika II, Matematické základy informatiky, Programovací techniky, Algoritmy umělé inteligence, Fuzzy systémy a neuronové sítě.

Během působení na VUT spolupracoval na řešení několika úkolů rozvoje vědy a techniky a vedlejší hospodářské činnosti. Od r. 1993 působil jako řešitel nebo spoluřešitel projektů TEMPUS JEP 4143-92 „University Courses on Mechatronics“ (řešitel, 1993–1995), FR 0789/96 „Mechatronika“ (spoluřešitel, 1997), FR 0603/97 „Inovace studijního programu mechatronika“ (spoluřešitel, 1998), MŠMT VS 96 122 „Mechatronické soustavy“ (spoluřešitel, 1996–2000), výzk. záměru CEZ: J22/98: 261100009 „Netradiční metody studia komplexních a neurčitých systémů“ (spoluřešitel, 2000–2002), GA ČR 101/00/1471 „Stabilita řízení rotorů na magnetických ložiskách“ (řešitel, 2000–2002) a výzk. záměru MSM 262100024 „Výzkum a vývoj mechatronických soustav“ (spoluřešitel 2000–2004).

Od r. 1999 pracuje rovněž v Centru mechatroniky, které je společným pracovištěm FSI VUT v Brně a ÚT AV ČR (Ústav termomechaniky Akademie věd ČR). Zde vede skupinu umělé inteligence a robotiky.

Od r. 1997 je členem mezinárodních programových výborů konferencí Mechatronics and Robotics ’97, Mechatronics and Robotics ’99, Mechatronics, Robotics and Biomechanics 2001 a Mechatronics, Robotics and Biomechanice 2003.

Je autorem nebo spoluautorem 63 prací, z toho 12 zahraničních (Anglie, Bulharsko, Německo, Polsko, Rusko, Slovensko, Španělsko, USA). Jeho publikační činnost je orientována na inženýrské aplikace metod umělé inteligence.

1 ÚVOD

Aktivní magnetické ložisko (AML) je relativně nový a velmi perspektivní konstrukční prvek eliminující přímý mechanický dotyk levitací rotoru v řízeném magnetickém poli. Z mechanického hlediska se vyznačuje zanedbatelnými třecími odpory, opotřebením, nízkou energetickou náročností, nízkou hlučností a vysokou přesností i tuhostí. Z hlediska řízení je velmi výhodná možnost měnit jeho dynamické vlastnosti za chodu stroje. Z hlediska provozního je nezanedbatelná jeho použitelnost za extrémních nebo speciálních podmínek.

Samotné AML je nestabilní, a proto je vždy potřeba AML stabilizovat zpětnovazební regulační smyčkou. Signál zpětné vazby je obvykle získáván bezdotykovým snímačem polohy rotoru. Regulátor vyhodnocuje odchylku rotoru od geometrické osy ložiska, a tím se řídí hodnota napájecího proudu elektromagnetů ložiska. Na řídicí elektroniku jsou však kladeny značné nároky, což mimo jiné vede k vysoké ceně takových ložisek. Přesto se tato ložiska již řadu let komerčně vyrábějí a jsou používána u některých velkých strojů (turbíny, turbokompresory, velké elektrické stroje), kde je jejich cena přijatelná.

Metody umělé inteligence (metody UI), zejména ty, které používají strojového učení v reálném čase, mohou představovat východiska návrhu nových metod řízení („inteligentních řídicích členů“) zlepšujících řízení AML, případně vyžadujících méně složitou řídicí elektroniku. Učení se dokonce stane zásadním faktorem, jestliže se prostředí nebo cíle subjektu řízení mění v čase.

Zásahy do dynamiky AML prováděné řídicím členem mají podstatný vliv na dlouhodobou dynamiku AML. Bez úplného vyřešení požadovaného chování řídicího členu může být velmi obtížné získat i malou množinu vzorů jeho požadovaného chování. Disponovat dostatečně reprezentativní množinou vzorů je ale nutné, je-li problém formulován jako úloha učení s učitelem (Duda a Hart [29]).

Naproti tomu v úlohách opakovaně posilovaného učení je tato problematika formulována jako optimalizační úloha, která použitím jednoduchého okamžitého ohodnocování zásahu subjektu do prostředí postupně zlepšuje odhad výkonu tohoto subjektu. Na základě dosaženého odhadu celkového výkonu se potom vybírají optimální zásahy do prostředí tak, aby tyto zásahy celkový výkon extremalizovaly. Ohodnocování očekávaného výkonu subjektu může zahrnovat nejrůznější kritéria, jako např. minimální čas, minimální cenu, minimum kolizí. Probíhá-li proces učení v reálném čase, může být subjekt chápán jako řídicí člen a prostředí jako řízená soustava. Základním rysem opakovaně posilovaného učení je tak schopnost v průběhu procesu učení zlepšovat chování řízené soustavy.

Úlohu určení optimálního chování řídicích členů vložených do konečného, stacionárního Markovova prostředí lze chápat jako problém řešení soustavy nelineárních rekurzivních rovnic (Ross [53], Bertsekas [11]). Iteračními metodami určenými k jejich řešení disponuje oblast dynamického programování. Byly vyvinuty v rámci teorie operačního výzkumu (Bellman [9]). Metody opakovaně posilovaného učení tyto výsledky rozpracovávají s cílem dosáhnout co nejvyšší výpočetní efektivnosti, aby tyto metody byly použitelné pro úlohy řízení v reálném čase. Patrně nejpoužívanější variantou opakovaně posilovaného učení je tzv. Q-učení.

Proces učení konvenčních architektur Q-učení je ale stále příliš pomalý na to, aby bylo Q-učení prakticky použitelné pro řešení některých reálných úloh řízení. Aplikací Q-učení na řízení AML výrazně komplikuje skutečnost, že je AML nestabilní. Tím se zejména v počátečních etapách učení dramaticky zvyšuje počet výběrů řídicích zásahů generovaných řídicím členem, který je nutný k dosažení některého z cílových stavů AML.

V předložené práci je navrženo rozdělení procesu Q-učení do dvou fází: do fáze předučení a fáze doučování. Speciálně organizovaná fáze předučení je výrazně efektivní, ale vyžaduje výpočtový model. Vychází z Q-učení v reálném čase. Fáze doučování je určena k dalšímu zlepšování řízení soustavy. Konvenčně využívá Q-učení v reálném čase a předpokládá interakci se soustavou.

1.1 HEURISTICKÉ PROHLEDÁVÁNÍ A ŘÍZENÍ

V UI znamená řízení proces rozhodování o tom, jak manipulovat s modelem dotyčného problému ne nutně v reálném čase, a proto tento termín nemá stejný význam jako v teorii řízení. V UI je řízení procesem formálního prohledávání (zpravidla heuristického). Algoritmy *heuristického prohledávání*, které jsou speciálním případem algoritmů prohledávání stavového prostoru, používají modelu řešeného problému, který se skládá z

a) množiny stavů,
b) množiny operátorů, které zobrazují množinu stavů do množiny stavů,
c) počátečního stavu a
d) množiny cílových stavů.

Tyto algoritmy hledají posloupnost operátorů, která zobrazuje počáteční stav do množiny cílových stavů a eventuálně optimalizuje nějakou míru ceny nalezené cesty.

V předložené práci je pojmu řízení používáno pro řízení dynamických systémů, které explicitně pracují s časem, a nikoliv pro prohledávání stavového prostoru.

V teorii řízení je heuristickým prohledáváním obvykle off-line procedura návrhu řízení, která s využitím modelu soustavy navrhuje strategii řízení v otevřené smyčce pro daný počáteční stav.

Řízení v otevřené smyčce je použitelné pouze tehdy, jsou-li splněny následující podmínky:

a) model použitý k návrhu řídicí strategie je přesným modelem soustavy,
b) může být přesně určen počáteční stav soustavy,
c) soustava je deterministická a
d) neexistují žádné nemodelované poruchy.

Tyto podmínky jsou splněny pouze pro některé úlohy UI a neplatí pro reálné úlohy řízení. V případě stochastického systému nebo v případě existence nemodelovaných poruch nelze při návrhu řízení v otevřené smyčce předcházet důsledkům náhodných nebo nemodelovaných jevů a vždy se lépe bude chovat řízení v uzavřené smyčce.

U *řízení v uzavřené smyčce* závisí velikosti řídicích veličin na aktuálním pozorování soustavy. V úvahu mohou být brána nejen aktuální pozorování, ale i minulá pozorování soustavy, či eventuálně další vnitřní informace řídicího členu.

Teorie řízení se většinou zabývá úlohami off-line návrhu odpovídajících strategií uzavřené smyčky za předpokladu, že je k dispozici přesný model řízeného systému. Off-line procedura návrhu typicky dává výpočetně efektivní metody pro určení řídicích veličin systému.

1.2 OPTIMÁLNÍ ŘÍZENÍ

V úloze optimálního řízení je cílem řízení extremalizovat nějakou cenovou funkcí (kriteriální funkci) řízení. Protože je trajektorie součástí řešení úlohy optimálního řízení, souvisejí takové úlohy úzce s úlohami heuristického prohledávání.

Pro řešení úloh optimálního řízení existují specializované metody, které obvykle předpokládají lineární systémy a kvadratické cenové funkce. Numerické metody používají zejména gradientní metody a metody dynamického programování a umožňují řešit i úlohy s nelineárními systémy a nekvadratickými cenami. Podobně jako algoritmy heuristického prohledávání jsou metody dynamického programování off-line procedurami návrhu optimální řídicí strategie a vytvářejí (na rozdíl od algoritmů heuristického prohledávání) v nedeterministické případě optimální strategii uzavřené smyčky.

1.3 HEURISTICKÉ PROHLEDÁVÁNÍ V REÁLNÉM ČASE

K řešení úloh heuristického prohledávání stavového prostoru, ve kterých je model soustavy rozšířen tak, aby uvažoval čas, se používají algoritmy *heuristického prohledávání v reálném čase* (např. Korf [34]). Stavový prostor musí mít tyto vlastnosti:

a) v každém okamžiku existuje jediný stav řízeného systému, stav je řídicímu členu znám,
b) během každého z posloupnosti časových intervalů, které jsou konstantní a omezené délky, musí řídicí člen předložit jedinou akci, tj. předložit výběr operátoru a
c) systém může měnit stavy na konci každého časového intervalu pouze podle aktuálního stavu a poslední akce předložené řídicím členem.

Existuje proto pevná horní hranice doby, ve které řídicí člen musí rozhodnout, kterou akci povést, má-li být tato akce založena na nejnovější informaci o stavu. Proto může být algoritmus heuristického prohledávání v reálném čase vhodný pro řízení v uzavřené smyčce.

2 MARKOVOVY ROZHODOVACÍ PROBLÉMY

Přítomnost neurčitostí zvýhodňuje zpětnovazební nebo reaktivní řízení před řízením v otevřené smyčce. Proto jsou významné teoretické rámce, které zahrnují stochastické problémy.

V Markovových rozhodovacích problémech nabývají operátory tvaru akcí, tj. řídicí veličiny soustavy pravděpodobnostně určují následníky stavů. Ačkoliv z hlediska teorie jsou akce „primitiva“, v aplikacích mohou být i příkazy vyšší úrovně, které mohou provádět složitá chování.

Markovův rozhodovací problém je definován na základě stochastických dynamických systémů s konečnou množinou stavů $S = \{1, \ldots, n\}$. Čas je představován posloupností časových kroků $t = 0, 1, \ldots$. Řídicí člen pozoruje v každém časovém kroku aktuální stav systému a vybírá řídicí akci, která se používá jako vstup systému. Je-li *i* pozorovaný stav, je akce vybírána z konečné množiny $U(i)$ přípustných akcí. Provede-li řídicí člen akci $u \in U(i)$, přejde systém v dalším časovém kroku do stavu *j* s pravděpodobností $p_{ij}(u)$. Použití akce *u* ve stavu *i* vyvolá okamžitou cenu $c_i(u)$.

Strategie řízení v uzavřené smyčce určuje každou akci jako funkci pozorovaného stavu. Taková strategie je označena $\mu = \{\mu(1), \ldots, \mu(n)\}$. Řídicí člen provádí akci $\mu(i) \in \mu$, kdykoliv pozoruje stav *i*. Jedná se o stacionární strategii, protože se nemění v čase. Ke každé strategii existuje funkce f^{μ}, nazývaná ohodnocovací funkce nebo cenová funkce (cena), odpovídající strategii μ. Tato funkce přiřazuje každému stavu celkovou cenu $f^{\mu}(i)$, o které se očekává, že vznikne v průběhu času, bude-li řídicí člen používat (počínaje stavem *i*) strategii μ. V této práci je ohodnocovací funkce $f^{\mu}(i)$ pro libovolnou strategii μ a stav *i* definována jako očekávaná hodnota diskontní ceny nekonečného horizontu, jestliže bude řídicí člen počínaje stavem *i* používat strategii μ:

$$f^{\mu}(i) = \mathrm{E}_{\mu}\left[\sum_{t=0}^{\infty} \gamma^t c_t \,\middle|\, s_0 = i\right], \tag{2-1}$$

kde γ, $0 \le \gamma \le 1$ je diskontní faktor použitý ke snížení budoucích okamžitých cen a E_{μ} je očekávání (střední hodnota), bude-li řídicí člen používat strategie μ. O $f^{\mu}(i)$ se hovoří jako o ceně stavu *i* při strategii μ.

Cílem uvažovaného Markovova rozhodovacího problému je nalézt strategii $\mu^* = \{\mu^*(1), \ldots, \mu^*(n)\}$, která minimalizuje cenu podle (2-1) každého stavu. Tato strategie je optimální strategií, závisí na γ a nemusí být jediná. Všem optimálním strategiím ale odpovídá jediná ohodnocovací funkce f^*, která je *optimální ohodnocovací funkcí* (*optimální cenovou funkcí*); tj. je-li μ^* libovolná optimální strategie, pak $f^{\mu^*} = f^*$.

Diskontní verze Markovova rozhodovacího problému s nekonečným horizontem ($0 \le \gamma < 1$) je matematicky nejjednodušší a zajišťuje, že vždy existuje optimální strategie, která je stacionární.

Je-li $\gamma = 1$ (nediskontní případ), nemusí být cena stavu podle (2-1) konečná. Proto se zavádí absorbční množina stavů. Jde o množinu stavů, které nemohou být nikdy opuštěny, jakmile se do nich jednou vstoupí. Okamžitá cena použití akce v jakémkoliv stavu z absorbční množiny je 0. Ohodnocovací funkce nekonečného horizontu pro strategii, která převádí systém do absorbční množiny, je pak omezená, i když $\gamma = 1$. Jako v diskontním případě, existuje alespoň jedna optimální strategie, která je stacionární. Tyto úlohy nazývá Bertsekas a Tsitsiklis [12] úlohami stochastické nejkratší cesty. Absorbční množina stavů bude dále nazývána *cílová množina*. Dále se zavádí pojem *korektní strategie*. Strategie dosažení cílového stavu je korektní, vyplývá-li z jejího použití při libovolném počátečním stavu nenulová pravděpodobnost dosažení cílového stavu.

2.1 NENASYTNÁ STRATEGIE A ROVNICE OPTIMALITY

I když ohodnocovací funkce f^{μ} udává cenu každého stavu při strategii μ, nevybírá strategie μ nutně akce, které podle ohodnocení f^{μ} vedou na nejlepší následníky aktuálního stavu. To zajišťuje strategie, která je *nenasytná* vzhledem ke své ohodnocovací funkci. Pro definici nenasytné strategie je dále použito Watkinsovy „Q“ notace [68], která se používá v metodách Q-učení popsaných v odst. 5.3.

Nechť je *f* reálná funkce stavů. Může být buď ohodnocovací funkcí pro nějakou strategii nebo odhadem „dobré“ ohodnocovací funkce (např. heuristickou ohodnocovací funkcí v heuristickém prohledávání) nebo je to libovolná funkce. Pro každý stav *i* a akci $u \in U(i)$ nechť

$$Q^f(i,u) = c_i(u) + \gamma \sum_{j \in S} p_{ij}(u) f(j), \tag{2.1-1}$$

kde $Q^f(i,u)$ je cena akce *u* ve stavu *i* při ohodnocení *f*. Jde o součet okamžité ceny a očekávaných hodnot cen všech následníků při akci *u*.

Strategie μ je nenasytná vzhledem k *f*, jestliže pro všechny stavy *i* je $\mu(i)$ akce, která splňuje

$$Q^f(i,\mu(i)) = \min_{u \in U(i)} Q^f(i,u). \tag{2.1-2}$$

Označme jako μ^f jakoukoliv strategii, která je nenasytná vzhledem k *f*. Vzhledem k *f* může existovat více nenasytných strategií a opačně, libovolná strategie je nenasytná vzhledem k více různým ohodnocovacím funkcím, ale optimálními mohou být pouze ty strategie, které jsou nenasytné vzhledem ke svým ohodnocovacím funkcím. Jestliže je tedy μ^* optimální strategie, pak její ohodnocovací funkce je optimální ohodnocovací funkcí f^* a $\mu^* = \mu^{f^*}$, tj. pro každý stav *i*

$$Q^{f^*}(i,\mu^*(i)) = \min_{u \in U(i)} Q^{f^*}(i,u). \tag{2.1-3}$$

Je-li f^* známá, je možné definovat optimální strategii jednoduše tak, aby splňovala (2.1-3). Položíme-li $Q^*(i,u) = Q^{f^*}(i,u)$, je nejdůležitějším důsledkem, že f^* je optimální ohodnocovací funkcí právě když pro každý stav *i* platí

$$f^*(i) = \min_{u \in U(i)} Q^*(i,u)$$
$$= \min_{u \in U(i)} \left[c_i(u) + \gamma \sum_{j \in S} p_{ij}(u) f^*(j) \right]. \qquad (2.1\text{-}4)$$

Jedná se o jeden z tvarů Bellmanovy rovnice optimality, kterou lze řešit pro libovolné $f^*(i)$, $i \in S$. Jakmile je nalezena f^*, může být pro stav *i* podle (2.1-1) určena optimální akce. Výpočetní složitost nalezení optimální akce užitím této metody je určena výpočetní složitostí nalezení f^*.

3 DYNAMICKÉ PROGRAMOVÁNÍ

Je-li dán úplný a přesný *model Markovova rozhodovacího problému* v podobě známých pravděpodobností přechodů stavů $p_{ij}(u)$ a okamžitých cen $c_i(u)$ pro všechny stavy *i* a akce $u \in U(i)$, je možné off-line řešit rozhodovací problém metodami dynamického programování (DP). Dále bude popsána jedna z verzí základní metody dynamického programování nazývaného *iterace hodnot* (aproximace v prostoru funkcí, Bellman a Kalaba [7]). Existují i jiné metody dynamického programování, např. *iterace strategie* (aproximace v prostoru strategií).

Iterace hodnot je úspěšná aproximační procedura, která konverguje k optimální ohodnocovací funkci f^*. Existuje mnoho variant iterace hodnot, které se liší v konkrétní organizaci výpočtu.

3.1 OFF-LINE ASYNCHRONNÍ ITERACE HODNOT

Nechť f_k označuje odhad f^* v *k*-té etapě výpočtu. V každé etapě se přepočítávají ceny nějaké podmnožiny stavů, přičemž ceny ostatních stavů zůstávají nezměněny (Bertsekas [10], Bertsekas a Tsitsiklis [12]). Podmnožina stavů, jejichž ceny jsou zálohovány, se s etapami může měnit. Volba těchto podmnožin určuje konkrétní chování algoritmu. Asynchronní iteraci hodnot lze zapsat jako

$$f_{k+1}(i) = \begin{cases} \min_{u \in U(i)} Q^{f_k}(i,u) & \text{pro } i \in S_k \\ f_k(i) & \text{jinak} \end{cases}, \qquad (3.1\text{-}1)$$

kde $S_k \subseteq S$ je množina stavů, jejichž ceny budou v *k*-té etapě zálohovány a $k = 0, 1, \ldots$.

Asynchronní iterace hodnot zahrnuje jako zvláštní případ i synchronní iteraci hodnot. Synchronní případ nastane, je-li $S_k = S$, pro každé *k*.

Diskontní asynchronní iterace hodnot konverguje k f^* za podmínky, že je každý stav obsažen v nekonečně mnoha podmnožinách S_k, $k = 0, 1, \ldots$.

V nediskontním případě ($\gamma = 1$) konverguje asynchronní iterace hodnot (Bertsekas a Tsitsiklis [12]), jestliže:

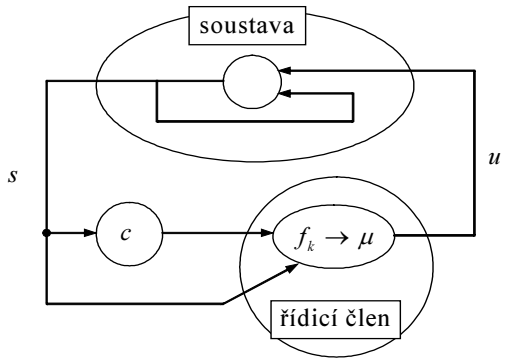
a) počáteční cena každého cílového stavu je nulová,
b) existuje nejméně jedna korektní strategie,
c) všechny strategie, které nejsou korektní, vyvolávají nekonečně velkou cenu nejméně jednoho stavu a
d) cena každého stavu je zálohována nekonečně mnohokrát.

Za splnění uvedených podmínek konverguje cena každého stavu opakovaným zálohováním k optimální ceně. Pořadí provádění záloh cen stavů (v závislosti na úloze) má vliv na poloměr konvergence (různé varianty viz např. Lin [43], Utgoff a Clouse [67] a Whitehead [70]).

4 UČENÍ JAKO ITERACE HODNOT V REÁLNÉM ČASE

Naznačený algoritmus iterace hodnot sice postupně aproximuje optimální ohodnocovací funkci, ale jeho etapy se nevztahují k časovým krokům, během nichž se odehrává řešený rozhodovací problém. Uvažujme algoritmus, ve kterém řídicí člen provádí asynchronní iterace hodnot *paralelně* s aktuálním procesem řízení (s procesem provádění akcí). Předpokládáme, že existuje úplný a přesný model rozhodovacího problému (Sutton [61]). Pro takové použití je vhodná asynchronní verze iterace hodnot. Podle toho, jak se řízená soustava skutečně chová, mohou být iterace hodnot soustředěny na ty množiny stavů, které jsou pro řízení soustavy nejvýznamnější. Třída takových algoritmů se nazývá iterace hodnot v reálném čase.

V dalším textu je použito indexu *k* pro etapy iterace hodnot a indexu *t* je použito k označení časového kroku, je-li řešena úloha řízení.



Obr. 4-1 Schéma iterace hodnot v reálném čase

Schéma iterace hodnot v reálném čase je znázorněno na obr. 4-1. Nechť s_t je poslední stav pozorovaný před časem t, a nechť k_t je celkový počet etap asynchronních iterací hodnot ukončených do času *t*. Pak f_{k_t} značí poslední odhad optimální ohodnocovací funkce f^*, který je k dispozici pro výběr akce $u_t \in U(s_t)$. Provedení akce u_t vyvolá okamžitou cenu $c_{s_t}(u_t)$ a stav soustavy se změní na s_{t+1}. Do okamžiku, než má být vybrána další akce u_{t+1}, jsou provedeny potřebné etapy asynchronní iterace hodnot a je určena ohodnocovací funkce $f_{k_{t+1}}$. Označme množinu stavů, jejichž ceny byly mezi časem t a $t+1$ zálohovány, jako B_t.

Různé varianty paralelního provádění iterace hodnot a řízení bývají mnoha autory nazývány *opakovaně posilovaným učením*. V těchto metodách se často nepracuje s okamžitou cenou stavu $c_{s_t}(u_t)$, ale s *okamžitým posílením* $R_{s_t}(u_t) = -c_{s_t}(u_t)$, které tak představuje cenu s opačným znaménkem. Je-li okamžité posílení kladné, hovoří se o *odměně*, je-li záporné, hovoří se o *pokutě*, a v rovnicích je pouze nahrazen operátor „min“ operátorem „max“.

4.1 ASYNCHRONNÍ ITERACE V REÁLNÉM ČASE

Při asynchronní iteraci hodnot v reálném čase paralelně probíhá asynchronní iterace hodnot a proces řízení. Interakce probíhá tak, že:

a) řídicí člen vždy používá strategii, která je nenasytná vzhledem k f_{k_t}; nejednoznačnosti výběru akcí jsou řešeny náhodně nebo tak, že je trvale zajištěn postupný výběr všech nenasytných akcí,
b) mezi provedením u_t a u_{t+1} je cena stavu s_t vždy zálohována, tj. $s_t \in B_t$ pro všechna *t*.

Nejjednodušší volbou je $B_t = \{s_t\}$, pro všechna *t*. Obecněji může B_t obsahovat i libovolné stavy, které byly generovány libovolnou metodou dopředného prohledávání. Řekneme, že asynchronní iterace hodnot v reálném čase konverguje, jestliže její asynchronní iterace hodnot konverguje k f^*. Protože řídicí člen vždy vybírá akce, které jsou nenasytné vzhledem k aktuálnímu odhadu f_{k_t}, je konvergentním procesem dosaženo optimálního řídicího chování. Existuje-li více než jedna optimální strategie, bude se řídicí člen přepínat mezi optimálními strategiemi, protože optimální akce bude vybírána z více nenasytných akcí.

Pro konvergenci platí podmínky uvedené v odstavci 3.1. V nediskontním případě je navíc požadováno, aby nebyl žádný stav úplně vyloučen ze zálohování své ceny. Protože asynchronní iterace hodnot v reálném čase vždy zálohuje cenu aktuálního stavu, je toho možné dosáhnout tak, že řídicí člen stále navštěvuje každý ze stavů, např. použitím opakovaných *pokusů*.

4.2 ASYNCHRONNÍ ITERACE V REÁLNÉM ČASE ZALOŽENÁ NA POKUSU

Pokus představuje provádění asynchronní iterace hodnot v reálném čase během časového intervalu nenulové omezené délky. Po uplynutí tohoto intervalu je soustava nastavena do nového počátečního stavu a začíná nový pokus. Jednoduchým způsobem, jak toho dosáhnout, je pokusy organizovat tak, že je každý stav s pravděpodobností 1 počátečním stavem nekonečně mnohokrát v nekonečné posloupnosti pokusů. Této metody nemůže být použito vždy, ale pro řadu úloh je tento přístup možný. Proto asynchronní iterace hodnot v reálném čase založená na pokusu konverguje s pravděpodobností 1 pro libovolný diskontní Markovův rozhodovací problém a libovolnou počáteční ohodnocovací funkci f_0.

V nediskontních úlohách nejkratší cesty konverguje asynchronní iterace hodnot v reálném čase založená na pokusu s pravděpodobností 1, jsou-li splněny podmínky konvergence uvedené v odstavci 3.1, s výjimkou bodu d).

Vhodná volba stavů může konvergenci zrychlit. Zajímavý přístup, který navrhl Peng a Williams [50] a Moore a Atkeson [49], nazývaný „upřednostňované procházení“, směruje provádění záloh na nejpravděpodobnější předchůdce stavů, jejichž ceny se významně mění.

5 ADAPTIVNÍ ZPŮSOB OPTIMÁLNÍHO ŘÍZENÍ

Uvedené verze iterace hodnot vyžadují jednak znalost pravděpodobností přechodů stavů $p_{ij}(u)$ pro všechny stavy *i*, *j* a všechny akce $u \in U(i)$ (přesný model soustavy), jednak vyžadují znalost okamžitých cen $c_i(u)$ pro všechny stavy *i* a akce $u \in U(i)$. Není-li tato informace k dispozici, jde o Markovův rozhodovací problém s neúplnou informací. Metody řešení těchto problémů jsou příklady metod *adaptivního optimálního řízení*.

Existují dvě hlavní třídy adaptivních metod pro Markovovy rozhodovací problémy s neúplnou informací (Kumar [36]). Bayesovské metody používají k odhadu pravděpodobností přechodů stavů Bayesových formulí, nebayesovské, většinou výpočetně méně komplikované, používají k odhadu pravděpodobností jiných způsobů jejich odhadů a jsou pro řadu úloh praktičtější.

Nepřímé metody explicitně modelují soustavu, která má být řízena. K aktualizaci modelu soustavy v libovolném čase během řízení používají algoritmů identifikace systémů. Při provádění řídicích rozhodnutí typicky předpokládají, že aktuální model je přesným modelem soustavy (Bertsekas [11]).

Naproti tomu *přímé metody* vytvářejí strategie bez použití explicitních modelů. Přímo odhadují strategii nebo informace jiné než model soustavy, např. ohodnocovací funkci.

Pro obě metody je zásadním problémem konflikt mezi řízením systému (využívání) a zkoumáním chování systému s cílem řízení zlepšit (prozkoumávání). Některé z mechanismů řešení tohoto

rozporu, pro které jsou k dispozici teoretické výsledky, porovnává Kurnar [36]. Další přístupy uvádí Barto a Singh [2], Kaelbling [31], Moore [47] Schmidhuber [56], Sutton [60], Watkins [68], Thrun [65] a Thrun a Möller [66].

V následujících odstavcích jsou uvedeny nebayessovské metody pro řešení Markovových rozhodovacích problémů s neúplnou informací.

5.1 ZÁKLADNÍ NEPŘÍMÁ METODA

Nepřímé adaptivní metody pro Markovovy rozhodovací problémy s neúplnou informací odhadují neznámé pravděpodobnosti přechodů stavů (a eventuálně okamžité ceny). Nejjednodušší možností, jak provést tuto identifikaci, je konstruovat model systému tak, aby se v každém časovém kroku skládal z maximálně věrohodných odhadů neznámých pravděpodobností přechodů stavů, označených $p_{ij}^{t}(u)$, pro všechny dvojice stavů *i*, *j* a akcí $u \in U(i)$. Nechť $n_{ij}^{u}(t)$ je pozorovaný počet provedení akce *u* před časovým krokem *t*, kdy systém přešel ze stavu *i* do stavu *j*. Pak $n_{i}^{u}(t) = \sum_{j \in S} n_{ij}^{u}(t)$ značí počet provedení akce *u* ve stavu *i* a maximálně věrohodný odhad $p_{ij}(u)$ v čase *t* je

$$p_{ij}^{t}(u) = \frac{n_{ij}^{u}(t)}{n_{i}^{u}(t)}. \qquad (5.1\text{-}1)$$

Jestliže bude v nekonečně mnoha časových krocích vybrána každá akce v každém stavu nekonečně mnohokrát, bude tento model systému konvergovat k přesnému modelu systému.

V každém časovém kroku *t* se používá k určení optimální ohodnocovací funkce f_t^* pro aktualizovaný model soustavy asynchronní iterace hodnot (ne v reálném čase). Pro přesný model by platilo $f_t^* = f^*$. Optimální strategie pro časový krok *t* je jakákoliv strategie $\mu_t^* = \{\mu_t^*(1), \ldots, \mu_t^*(n)\}$, která je nenasytná vzhledem k f_t^* a v časovém kroku *t* je $\mu_t^*(s_t)$ optimální akcí. Vhodné je algoritmus inicializovat v každém časovém kroku odhadem f^*, který byl vytvořen v předchozím časovém kroku. Protože během jednoho časového kroku dochází pouze k malým změnám modelu systému, nebudou se pravděpodobně optimální ohodnocovací funkce f_t^* a f_{t+1}^* významně lišit. Objem výpočtů nutný k provedení jedné asynchronní iterace hodnot ale může být tak vysoký, že znemožní řešení úlohy řízení. Řídicí člen by měl při sledování cílů řízení provádět posloupnosti akcí pouze podle μ_t^*. Protože ale aktuální model nutně není přesný, musí řídicí člen sledovat i identifikační cíle (např. Kumar [36]), musí občas vybrat akci jinou, než μ_t^*.

Jedním z nejjednodušších způsobů navození průzkumného chování je použití stochastických strategií, ve kterých jsou akce vybírány podle pravděpodobností, které závisejí na hodnotách aktuální ohodnocovací funkce. Každá akce má vždy nenulovou pravděpodobnost provedení, přičemž aktuální optimální akce podle μ_t^* má tuto pravděpodobnost nejvyšší. Často je používána metoda výběru akce založená na Boltzmanově rozložení (Watkins [68], Lin [42] a Sutton [60]).

5.2 ADAPTIVNÍ ASYNCHRONNÍ ITERACE HODNOT V REÁLNÉM ČASE

Metoda adaptivní asynchronní iterace hodnot v reálném čase je stejná, jako metoda uvedená v odstavci 3.1 s výjimkou toho, že

a) model systému je aktualizován vhodnou on-line metodou identifikace soustavy, např. podle (5.1-1)

b) během provádění etap je místo přesného modelu systému používáno aktuálního modelu soustavy a
c) akce je v každém časovém kroku určena náhodnou strategií, která vyvažuje cíle identifikace a řízení.

Adaptivní asynchronní iterace hodnot v reálném čase je blízká mnoha existujícím algoritmům, viz např. Sutton [61], Lin [41, 42] a Jalali a Ferguson [30]. Pro zefektivnění identifikace doporučuje např. Sutton [60] zálohovat ceny stavů, u kterých existuje „rozumná“ spolehlivost přesnosti odhadovaných pravděpodobností přechodů stavů. Taková strategie vyvolává prozkoumávání používající identifikaci, která ale může být v rozporu s řízením. Kaelbling [31], Thrun [65] a Thrun a Möller [66] uvádějí tyto i další možnosti.

5.3 Q-UČENÍ JAKO ADAPTIVNÍ PŘÍMÁ METODA

Na rozdíl od obou uvedených nepřímých adaptivních metod Q-učení (Watkins [68]) přímo odhaduje optimální Q-hodnoty dvojic stavů a přípustných akcí (nazývaných přípustné dvojice akce-stav). Připomeňme, že podle (2.1-4) je optimální Q-hodnota $Q^*(i,u)$ stavu i a akce $u \in U(i)$ cenou za generování akce u ve stavu i, která vyhovuje optimální strategii. Libovolná strategie vybírající vzhledem k optimálním Q-hodnotám nenasytné akce je optimální strategií. Jsou-li tedy k dispozici optimální Q-hodnoty, lze optimální strategii určit s relativně málo výpočty.

Výklad Q-učení v této práci se poněkud liší od Watkinsova výkladu [68] i dalších výkladů (např. Sutton [60], Barto a Singh [2]). Chápe metodu Q-učení jako metodu adaptivního on-line učení. K zdůraznění vztahu Q-učení a asynchronní iterace hodnot je nejprve uvedeno Q-učení jako off-line asynchronní metoda iterace hodnot, která nevyžaduje přímý přístup k pravděpodobnostem přechodů stavů rozhodovacího problému. Dále je uvedeno obvyklejší pojetí on-line Q-učení.

5.3.1 Off-line Q-učení

Zde Q-učení provádí odhady ohodnocovací funkce pro každou přípustnou dvojici stav-akce. Nechť pro libovolný stav i a akci $u \in U(i)$ je $Q_k(i,u)$ odhadem $Q^*(i,u)$ v k-té etapě výpočtu. S uvážením, že podle (2.1-4) je f^* v každém stavu minimem optimálních Q-hodnot, lze chápat Q-hodnoty v k-té etapě jako implicitně definované odhady f_k funkce f^*, které jsou pro libovolný stav i dány jako

$$f_k(i) = \min_{u \in U(i)} Q_k(i,u). \qquad (5.3.1\text{-}1)$$

Q-hodnoty obsahují více explicitní informace než ohodnocovací funkce. Výběr akcí nenasytné strategie lze například provést pouze použitím Q-hodnot.

Místo přímého používání pravděpodobností přechodů stavů používá off-line Q-učení pouze funkci, která s těmito pravděpodobnostmi generuje následníky stavu. Nezávisle proměnnou této funkce je stav i a akce $u \in U(i)$ a závisle proměnnou s pravděpodobností $p_{ij}(u)$ stav j. Označíme-li tuto funkci „succ“, lze psát $j = \mathrm{succ}(i,u)$.

V každé k-té etapě aktualizuje off-line Q-učení synchronně pouze Q-hodnoty podmnožiny přípustných dvojic stav-akce a podmnožina přípustných dvojic stav-akce, jejíž Q-hodnoty jsou aktualizovány, se etapu od etapy mění. Výběr těchto podmnožin určuje charakter algoritmu.

Nechť pro každé $k = 0, 1, \ldots,$ značí $S_k^Q \subseteq \{(i,u) | i \in S, u \in U(i)\}$ podmnožinu přípustných dvojic stav-akce, jejichž Q-hodnoty jsou aktualizovány v k-té etapě. Pro každou dvojici stav-akce z S_k^Q je třeba definovat parametr učení, který určuje, do jaké míry je nová Q-hodnota určena svou

původní hodnotou a do jaké míry zálohovanou hodnotou. Nechť $\alpha_k(i,u)$, $0<\alpha_k(i,u)<1$ označuje parametr učení aktualizace Q-hodnoty dvojice (i,u) v k-*té* etapě. Pak

$$Q_{k+1}(i,u)=\begin{cases}(1-\alpha_k(i,u))Q_k(i,u)+ \\ \quad \alpha_k(i,u)\left[c_i(u)+\gamma f_k(\mathrm{succ}(i,u))\right], & \text{pro } (i,u)\in S_k^Q, \\ Q_k(i,u), & \text{jinak,}\end{cases} \qquad (5.3.1\text{-}2)$$

kde f_k je definováno podle (5.3.1-1). Zálohování Q-učení (zálohování Q-hodnoty) znamená použití (5.3.1-2) pro jednu dvojici stav-akce (i,u).

Posloupnost $\{Q_k(i,u)\}$ generovaná off-line Q-učením konverguje pro $k\to\infty$ s pravděpodobností 1 k $Q^*(i,u)$ pro všechny přípustné dvojice (i,u) (viz Watkins [68], Watkins a Dayan [69]), je-li:

a) zálohování Q-hodnot prováděno pro každou přípustnou dvojici stav-akce (i,u) nekonečně mnohokrát a v nekonečně mnoha etapách a
b) snižuje-li se během etap vhodným způsobem parametr učení $\alpha_k(i,u)$.

Q-hodnoty v *k*-té etapě definují ohodnocovací funkci f_k podle (5.3.1-1), pro všechny přípustné dvojice stav-akce. Etapa off-line Q-učení definovaná podle (5.3.1-2) tak může být chápána jako aktualizace f_k na f_{k+1}, pro kterou je

$$f_{k+1}(i)=\min_{u\in U(i)} Q_{k+1}(i,u),$$

pro každý stav i. Tato aktualizace ohodnocovací funkce ale neodpovídá etapě obvyklého zálohování při iteraci hodnot, protože pro vybrané akce určené dvojicemi stav-akce v S_k^Q používá pouze stavů daných funkcí „succ“. Naproti tomu zálohování při iteraci hodnot používá skutečné očekávané ceny následníků pro všechny přípustné akce v daném stavu.

Off-line Q-učení je asynchronní na úrovni přípustných dvojic stav-akce. Každé zálohování Q-učení vyžaduje při určení $f_k(i)$ podle (5.3.1-1), kterého se používá ve výrazu $f_k(\mathrm{succ}(i,u))$ v (5.3.1-2), minimalizaci přes všechny přípustné akce v daném stavu. Nevyžaduje ale výpočet uvažující všechny možné následníky. Nevýhodou však je vyšší složitost prostoru Q-učení a dále to, že zálohování asynchronní iterace hodnot je srovnatelné s provedením více zálohování Q-učení.

5.3.2 Q-učení v reálném čase

Budeme-li provádět off-line Q-učení paralelně s řízením, získáme on-line algoritmus. Na rozdíl od odst. 5.2 Q-učení v reálném čase nepoužívá žádného modelu soustavy vymezujícího rozhodovací problém. Funkci následníka představuje přímo soustava. Metoda provádí zálohování Q-hodnoty pouze pro jedinou dvojici stav-akce v každém časovém kroku řízení. Dvojice stav-akce se skládá z pozorovaného aktuálního stavu a akce, která je právě prováděna.

Předpokládejme, že v každém časovém kroku *t* pozoruje řídicí člen stav s_t a má k dispozici odhad optimálních Q-hodnot vytvářených všemi předchozími etapami Q-učení v reálném čase. Označme tyto odhady $Q_t(i,u)$ pro všechny přípustné dvojice stav-akce (i,u). Řídicí člen vybírá

akci $u_t \in U(s_t)$, přičemž umožňuje prozkoumávání. Po provedení akce u_t získá řídicí člen během změny stavu systému do s_{t+1} okamžitou cenu $c_{s_t}(u_t)$. Pak se Q_{t+1} vypočte jako

$$Q_{t+1}(i,u) = \begin{cases} \big(1-\alpha_t(s_t,u_t)\big)Q_t(s_t,u_t) + \\ \quad \alpha_t(s_t,u_t)\big[c_{s_t}(u_t) + \gamma f_t(s_{t+1})\big], & \text{pro } (i,u) = (s_t,u_t), \\ Q_t(i,u), & \text{jinak,} \end{cases} \qquad (5.3.2\text{-}1)$$

kde $f_t(s_{t+1}) = \min_{u \in U(s_{t+1})} Q_t(s_{t+1},u)$ a $\alpha_t(s_t,u_t)$ je parametr učení v časovém kroku *t* pro aktuální dvojici stav-akce. Toto zálohování se opakuje pro každý časový krok.

K definici úplného adaptivního řídicího algoritmu využívajícího Q-učení v reálném čase je nezbytné určit, jak mají být na základě aktuálních Q-hodnot vybírány akce, aby bylo umožněno prozkoumávání. Konvergence k optimální strategii vyžaduje stejný způsob prozkoumávání, jako k identifikaci systému vyžadují nepřímé metody.

6 NÁVRH ORGANIZACE Q-UČENÍ

Definujeme-li cílové stavy AML jako stavy s malou výchylkou x a například malou rychlostí rotoru $\dot{x}$, ve kterých budeme rotor pokládat za stabilizovaný, lze úlohu řízení AML formulovat jako stochastickou úlohu nalezení nejkratší cesty z libovolného počátečního stavu do některého z cílových stavů (odst. 2, Bertsekas a Tsitsiklis [12]). Množina cílových stavů ale není absorpční množinou, protože navštívení některého z jejich stavů ještě neznamená, že AML nemůže uskutečnit přechod do stavu, který prvkem této množiny není (k tomu dojde například dostatečně velkou a náhlou změnou zátěžné síly rotoru). Důsledkem je, že úlohu nelze definovat jako Markovův rozhodovací problém s nediskontním nekonečným horizontem ($\gamma = 1$), ale je nutno uvažovat diskontní nekonečný horizont ($\gamma < 1$). Žádoucí je volit γ co nejblíže 1.

Samotné AML je nestabilní, a proto existuje pouze velmi malá pravděpodobnost, že zejména v počátečních etapách učení bude i sofistikovaným prozkoumáváním (odst. 5) dosaženo cílového stavu. Nestabilita AML vyhrotí konflikt mezi identifikací a řízením. To tak zpomalí konvergenci k optimálním cenám stavů, že je diskutabilní jejich použitelnost pro řešení úloh tohoto typu. Odhady optimálních cen stavů budou po mnoho etap s malou pravděpodobností blízké optimálním cenám stavů a z nich odvozené strategie budou s malou pravděpodobností blízké optimální strategii. Proto se jako vhodné jeví použití asynchronní iterace hodnot v reálném čase. K zajištění konvergence je nutné, aby každý stav byl navštíven nekonečně mnohokrát (odst. 4.1).

Toho lze dosáhnout použitím opakovaných pokusů organizovaných tak, že každý stav bude s pravděpodobností 1 počátečním stavem pokusu nekonečně mnohokrát v nekonečné posloupnosti pokusů (odst. 4.2). Bude-li navíc každý z pokusů probíhat během takového časového intervalu nenulové omezené délky, že během něj dojde právě k jednomu přechodu stavu, bude organizace takového *elementárního pokusu* stále vyhovovat podmínkám konvergence a přitom bude na nejvyšší možnou míru omezen vliv aktuální strategie na výběr stavů k zálohování. Bude tedy na nejvyšší možnou míru potlačen konflikt mezi identifikací a řízením (odst. 5). Fyzikální soustavu však nelze obecně nastavit do libovolného předepsaného stavu, který by odpovídal počátečnímu stavu pokusu. Proto je nutné chování soustavy simulovat s použitím výpočtového modelu. Protože není k dispozici model Markovova rozhodovacího problému, je nutno volit metodu, která je navíc adaptivní (odst. 5).

Zvoleno bylo Q-učení v reálném čase (odst. 5.3.2) jako efektivní reprezentant (přímé) adaptivní asynchronní iterace hodnot v reálném čase založené na pokusu. Proces Q-učení byl rozdělen do dvou fází:

a) fáze předučení a
b) fáze doučování.

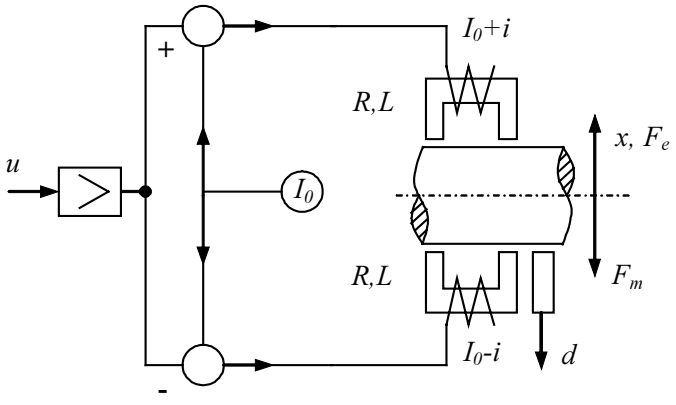
Fáze předučení představuje provádění elementárních pokusů, které začínají se zvolenými dvojicemi stav-akce (i,u), $i \in S$, $u \in U(i)$ a které realizují jediný přechod stavu, aby bylo možno provést zálohu Q-hodnoty pro tuto dvojici (i,u) podle (5.3.1-2). Funkci následníka stavu i při akci u poskytuje výpočtový model soustavy. Elementární pokusy jsou prováděny s počátečními dvojicemi (i,u), které systematicky a opakovaně procházejí celou množinu přípustných dvojic stav-akce. Tím je zaručeno provádění takových pokusů, které splňují podmínku a) konvergence Q-učení (viz odst. 5.3.1). Splnění i podmínky b) konvergence Q-učení má za důsledek to, že posloupnost Q-hodnot generovaná ve fází předučení podle (5.3.1-2) konverguje s pravděpodobností 1 k optimálním hodnotám Q^* a po provedení dostatečného počtu etap fáze předučení může být použito dosažených Q-hodnot k určení strategie velmi blízké optimální strategii. Tato strategie není citlivá na malé chyby v aproximaci Q-hodnot (Singh a Yee [57]) a lze proto použít přibližného výpočtového modelu soustavy.

Během fáze doučování je takto získaná strategie eventuálně dále zpřesňována, přizpůsobována skutečným provozním podmínkám soustavy atd. Jako počátečních Q-hodnot je použito Q-hodnot dosažených ve fázi předučení. Učení probíhá konvenčním způsobem, tj. provádí se zálohování paralelně s řízením podle (5.3.2-1). Zde již funkci následníka stavu s_t při použití akce $u_t \in S_t$ může poskytovat řízená soustava. Akce jsou stanovovány s využitím stochastické strategie (viz odst. 7.3), která umožňuje prozkoumávání. Parametry této strategie jsou ale nastaveny tak, aby pravděpodobnost prozkoumávání byla nízká.

7 IMPLEMENTAČNÍ PŘÍSTUPY

7.1 VÝPOČTOVÝ MODEL AKTIVNÍHO MAGNETICKÉHO LOŽISKA

Simulace byly prováděny s jednoduchým výpočtovým modelem AML s jedním stupněm volnosti (výchylkou v rovině kolmé na osu stroje), který sestavil Půst [51]. Modelována byla často publikovaná varianta soustavy se strukturou podle obr. 7.1-1 (např. Laier a Markert [39]).



Obr. 7.1-1 Struktura AML

Za předpokladu, že: symetrický tuhý rotor je nahrazen hmotným bodem, je zanedbána vazba mezi vodorovným a svislým kmitáním, je zanedbán vliv tíže a nelinearity jsou uvažovány pouze u elektromagnetického subsystému, lze psát

$$\begin{aligned} m\ddot{x}(t) + b\dot{x}(t) + F_m(x,i) &= F_e(t) \\ Ri(t) + L\dot{i}(t) &= u(t) \end{aligned}, \; x(t) = x_0, i(t) = i_0, \dot{x}(t) = \dot{x}_0 \text{ pro } t \le 0, \quad (7.1\text{-}1)$$

kde m a $x(t)$ jsou hmotnost a výchylka hmotného bodu nahrazujícího rotor od geometrické osy ložiska, $i(t)$ je odchylka proudu elektromagnetu ložiska od konstantní stejnosměrné složky I_0, R a L jsou odpor a indukčnost napájecích obvodů a elektromagnetu ložiska, $F_e(t)$ je zátěžná síla vázaná na nevyváženost rotoru (porucha v mechanické části ložiska, homogenity materiálu, klasická/rotující nevyváženost) a konečně $u(t)$ je budicí napětí magnetu ložiska. Koeficient b viskózního tlumení zavádí do modelu vliv okolního prostředí. $F_m(x,i)$ je síla, kterou magnetické ložisko působí na rotor. Hodnoty veličin jsou technologicky omezené, konkrétně musí být

$$|x| \le x_{\max}, \; |i(t)| \le i_{\max}, \; |u(t)| \le u_{\max}. \quad (7.1\text{-}2)$$

Pro $F_m(x,I)$ se pokládá za vyhovující aproximace

$$F_m(x,i;\, \mathbf{c}) = -c_1 x + c_2 i - c_3 x^3 + c_4 i x^2 - c_5 i^2 x - c_6 i^3, \quad c_1, \ldots, c_6 > 0. \quad (7.1\text{-}3)$$

Hodnoty parametrů modelu podle (7.1-1) až (7.1-3) byly rovněž převzaty z prací Půst a [51] a Kozánka a kol. [35]. Jsou uvedeny v tab. 7.1-1.

$R = 2$ [Ω]	$c_1 = 1289 \times 10^3$ [kgs^{-2}]	$x_{max} = 500 \times 10^{-6}$ [m]
$L = 5 \times 10^{-3}$ [H]	$c_2 = 454$ [kgms^{-2}A^{-1}]	$i_{max} = 2$ [A]
$m = 5.65$ [kg]	$c_3 = 3184 \times 10^9$ [kgm^{-2}s^{-2}]	$u_{max} = 100$ [V]
$b = 20$ [kg s^{-1}]	$c_4 = 841 \times 10^6$ [kgm^{-1}s^{-2}A^{-1}]	
	$c_5 = 38 \times 10^3$[kgs^{-2}A^{-2}]	
	$c_6 = 30$ [kgms^{-2}A^{-3}]	

Tab. 7.1-1 Hodnoty parametrů výpočtového modelu AML

Stavy soustavy byly pro potřeby Q-učení tvořeny hodnotami vybraných veličin soustavy pozorovanými v diskrétních časech *t*. Pro posouzení možností navržené organizace Q-učení byly stavy konstruovány ve dvou variantách:

a) třídimenzionální stav $\mathbf{y} = (x,\ \dot{x},\ \ddot{x})$, dále stručněji označovaný jako *3-D stav* a

b) dvoudimenzionální stav $\mathbf{y} = (x,\ \dot{x})$ označovaný jako *2-D stav*.

7.2 IMPLEMENTACE Q-FUNKCE

Nezávisle proměnné Q-funkce jsou tvořeny dvojicemi stav-akce (i,u) z diskrétních konečných množin S a $U(i)$. Stav soustavy $\mathbf{y} \in Y$ však představuje uspořádaná $\dim Y$-tice jistých (spojitých) veličin soustavy $\mathbf{y} = (y_1, \ldots, y_{\dim Y})$. Každé z veličin použitých při konstrukci stavu AML je proto třeba určit uzlové body *rastru*, které pak vymezují podmnožiny stavů soustavy (diskrétní stavy Markovova rozhodovacího problému). Pro řídicí veličiny, které jsou diskrétní, je vhodné

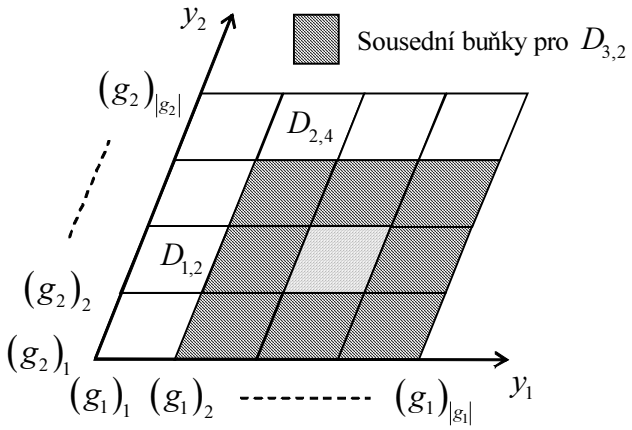
použít analogicky jejich přípustné hodnoty jako uzlové body rastrů těchto veličin. Rastry všech veličin (jak stavu AML, tak řídicích veličin) definují *mřížku* tabulky.

Nechť $\left\{\left(g_v\right)_w\right\}_{w=1}^{|g_v|}$ označuje vzestupně uspořádanou posloupnost uzlových bodů rastru v-té veličiny stavu AML, $v=1,\ldots,\dim Y$, kde $|g_v|$ znamená počet uzlových bodů příslušného rastru, který musí být lichý a $|g_v|\geq 3$.

Buňkou stavů soustavy nazveme množinu

$$D_{w(1),w(2),\ldots,w(\dim Y)}=\bigcap_{v=1}^{\dim Y} D'_{v,w(v)},$$

kde $D'_{v,w}=\left\{\mathbf{y};\ \mathbf{y}\in Y,\left(g_v\right)_w\leq y_v<\left(g_v\right)_{w+1},\ w=1,\ldots|g_v|\right\}$ představuje stavy soustavy vyhovující omezení w-tým intervalem rastru v-té veličiny.



Obr. 7.2-1 Rastr a buňky stavů soustavy pro $\dim Y=2$

O buňkách $D_{w_1(1)w_1(2)\ldots w_1(\dim Y)}$ a $D_{w_2(1),w_2(2),\ldots,w_2(\dim Y)}$ řekneme, že jsou sousední, právě když jsou různé a $|w_1(v)-w_2(v)|\leq 1$, $v=1,\ldots,\dim Y$ (viz obr. 7.2-1).

Bylo použito symetrických rastrů s uzlovými body definovanými pomocí *koeficientů nelinearity rastrů* r_v.

Nechť $w_0(v)=\left(|g_v|\div 2\right)+1$, $\left(h_v\right)_w=\dfrac{w-w_0(v)}{w_0(v)-1}$, $w=1,\ldots,|g_v|$, $v=1,\ldots,\dim Y$, pak

$$\left(g_v\right)_w=\left(y_v\right)_{max}\left(h_v\right)_w b_v^{1-|(h_v)_w|},\ b_v=r_v^{|w_0(v)-1|/(|w_0(v)-1|-1)} \qquad (7.2\text{-}1)$$

pro $0<r_v\leq 1$. Mřížku pak lze charakterizovat zápisem

$$\left(|g_1|/r_1,\ldots,|g_{\dim X}|/r_{\dim Y};|g_{\dim Y+1}|/r_{\dim Y+1},\ldots,|g_{\dim Y+\dim A}|/r_{\dim Y+\dim A}\right).$$

Pro lineární rastr v-té veličiny je dále použito místo symbolu $|g_v|/1$ jednoduššího symbolu $|g_v|$. Např. lineární rastr 2-D stavu s 11 uzlovými body rastru výchylky rotoru AML x, 7 uzlovými

body radiální rychlosti rotoru $\dot{x}$ a 3 možnými hodnotami jediné řídicí veličiny stejně od sebe vzdálenými je charakterizován zápisem $(11, 7; 3)$.

7.3 IMPLEMENTACE PROZKOUMÁVÁNÍ

K implementaci prozkoumávání bylo použito metody výběru akcí používající Boltzmanova rozložení (Watkins [68], Lin [42] a Sutton [60]). Metoda přiřazuje v aktuálním stavu každé přípustné akci pravděpodobnost jejího provedení

$$\mathrm{P}(\mathrm{u}) = \frac{e^{Q(s_t,u)/T}}{\sum_{v \in U(s_t)} e^{Q(s_t,u)/T}},\ u \in U(s_t), \tag{7.3-1}$$

kde $T > 0$ je parametr, který řídí, jak významně se tyto pravděpodobnosti podílejí na výběru akce $\mu_t(s_t)$ (viz odst. 5.1). Je-li T vysoké, je preferováno prozkoumávání, s klesajícím T je náhodný výběr akce potlačován a realizuje se akce $\mu_t(s_t)$. Hodnota parametru T se postupně snižuje podle

$$T_{k+1} = T_{min} + \beta(T_k - T_{min}), \tag{7.3-2}$$

kde k je etapa zálohování a $T_0 = T_{max}$.

Jako posloupnosti parametrů učení $\alpha_t(i,u)$ (odst. 5.3.2), která vyhovuje podmínkám konvergence Q-učení [68, 69], bylo použito posloupnosti podle Darkena a Moodyho [24])

$$\alpha_k(i,u) = \frac{\alpha_0 n_0}{n_0 + n_k(i,u)}, \tag{7.3-3}$$

kde $n_k(i,u)$ je počet záloh provedených na Q-hodnotě (i,u) do etapy k, α_0 je počáteční parametr učení a n_0 je parametr určený k řízení rychlosti snižování $\alpha_k(i,u)$.

8 SIMULAČNÍ PŘÍSTUPY

V dále popisovaných simulacích je pokus chápán poněkud šířeji, než jak je vymezen v odst. 4.2. Představuje simulaci procesu řízení, který trvá tak dlouho, dokud není

a) dosaženo stavu soustavy, ve kterém je splněna podmínka $|x| = x_{\max}$; pak hovoříme o *neúspěšném pokusu,* nebo
b) není vyčerpána *maximální délka pokusu* N, tedy předepsaný počet řídicích rozhodnutí (odvození akcí řídicím členem), a pak hovoříme o *úspěšném pokusu*.

Délkou pokusu je označen počet řídicích rozhodnutí provedených během pokusu a *úspěšnost pokusu* je délka pokusu vztažená na maximální délku pokusu.

Pokus může nebo nemusí být oddělen od Q-učení. Je-li pokus od Q-učení oddělen, pak

a) se nezálohují během něj prvky tabulky Q-funkce a
b) řídicí rozhodnutí jsou realizována pouze podle (2.1-2), tedy bez provádění prozkoumávání.

Pokusů oddělených od Q-učení bylo použito k *testování* dosaženého chování řídicího členu např. během fáze předučení. V této fázi bylo střídavě prováděno vždy několik průchodů tabulkou Q-funkce a poté několik desítek pokusů. Není-li pokus oddělen od Q-učení, pak

a) se zálohují během něj prvky tabulky Q-funkce a

b) řídicí rozhodnutí jsou realizována s prozkoumáváním podle (7.3-1 a 7.3-2). Takto byly prováděny pokusy ve fázi doučování.

Protože optimální strategie μ může být dosaženo daleko dříve, než je splněno konvergenční kritérium procesu Q-učení, nebyla při vyhodnocování simulací ve fázi předučení posuzována rychlost konvergence k optimální Q-funkci, ale rychlost předučení. Průměrná délka pokusu a procento úspěšných pokusů byly stanovovány vždy ze 100 pokusů, které se lišily pouze v počátečním stavu AML a průběhu zátěžné síly $F_e(t)$. *Délka předučení*, kterou se rozumí. maximální povolený počet průchodů tabulkou Q-funkce, byla omezena na 1 000 průchodů.

Počáteční stavy AML byly voleny náhodně z takových stavů soustavy, pro které AML neuskutečnilo během 100×10^{-6} [s] použitím libovolné akce z množiny $\{-u_{\max}, 0, u_{\max}\}$ přechod do stavu s $|x| = x_{max}$. Takto byly hrubě odhadnuty řiditelné stavy soustavy.

Zátěžná síla $F_e(t)$ byla modelována schodovitou funkcí s náhodnou velikostí konstantních částí z intervalu $\langle -200, 200\rangle$ [N]. Aby byla zajištěna srovnatelnost testů, byly pro zmíněných 100 pokusů vygenerovány jednak náhodné počáteční podmínky, jednak pro každý pokus 3 000 náhodných velikostí konstantních částí zátěžné síly (průběhů zátěžné síly s 3 000 možnými změnami její velikosti) a uloženy do souboru označeného *data A*. Pro podrobnější testování byl připraven i rozsáhlejší soubor označený *data B*, do kterého byly opět pro 100 pokusů uloženy jednak náhodné počáteční podmínky, jednak pro každý pokus 100×10^3 náhodných velikostí konstantních částí zátěžné síly (průběhů zátěžné síly s 100×10^3 možnými změnami její velikosti).

Dat A bylo použito pro veškeré testy, tj. pro:

a) stanovování průměrných délek pokusu během fáze předučení,
b) zjištění odolnosti dosažených strategií vůči chybám pozorování veličin soustavy,
c) zjištění odolnosti dosažených strategií vůči zpoždění akčního zásahu a
d) porovnávání chování AML řízeného dosaženou strategií s chováním AML řízeného referenčním PID regulátorem.

Dat B bylo použito pouze pro podrobnější porovnání dosažených strategií se strategií referenčního PID regulátoru podle bodu d).

Při změnách velikosti zátěžné síly $F_e(t)$ v datech A i B bylo generování hodnot organizováno tak, že po sobě mohlo následovat nejvýše 100 nulových změn velikostí $F_e(t)$ a změny velikosti $F_e(t)$ byly postupně používány pro stanovení skutečné velikosti zátěžné síly, a to v okamžiku zahájení přechodu stavu. Při délce trvání přechodu stavu τ_p se tak náhodná délka konstantní části zátěžné síly mohla pohybovat v intervalu $\langle \tau, 100\tau\rangle$ [s]. S výjimkou fáze předučení, ve které bylo použito jiných délek doby trvání přechodu (viz odst. 8.1), byla délka doby trvání přechodu pevná 10×10^{-6} [s], tj. délka konstantních částí zátěžné síly se pohybovala v intervalu $\langle 10\times10^{-6}, 1\times10^{-3}\rangle$ [s] při celkové době řízení 30×10^{-3} [s] pro data A, a 1 [s] pro data B.

Poznamenejme, že při použité organizaci simulací není dosažitelná průměrná úspěšnost pokusů 100 %. Je to dáno poměrně hrubým odhadem množiny řiditelných stavů, jejíž prvky byly používány jako počáteční stavy jednotlivých pokusů. Na druhé straně tento odhad umožňuje lépe porovnat úspěšnost získané strategie se strategií referenčního PID regulátoru.

8.1 STANDARDNÍ PODMÍNKY SIMULACÍ

Pokud není v textu uvedeno jinak, byly v jednotlivých sadách simulací brány jako výchozí standardní podmínky simulace podle tab. 8.1-1, vůči kterým byla měněna typicky hodnota jednoho parametru.

Množina řídicích akcí	$\{-u_{\max}, 0, u_{\max}\}$	
Délka trvání přechodu stavu	$\langle 1\times10^{-5}, 1\times10^{-3}\rangle$[s]	
Okamžité posílení	prostá omezená pokuta	
Parametry předučení	$\alpha_0 = 0.2$, $n_0 = 300$, $\gamma = 0.999$	viz (7.3-3) viz (5.3.1-2), (5.3.2-1)
Parametry doučování	$\alpha_0 = 0.1$, $n_0 = 100$,	viz (7.3-3)
	$\gamma = 0.999$,	viz (5.3.1-2), (5.3.2-1)
	$T_{min} = 5$, $T_{max} = 75$,	viz (7.3-2)
	$\beta = 0.999$	
Průchod tabulkou	Sekvenční dopředný	
Zátěžná síla	náhodná schodovitá, $(F_e)_{max} = 200$[N]	

Tab. 8.1-1 Standardní podmínky simulací

9 FÁZE PŘEDUČENÍ S LINEÁRNÍMI MŘÍŽKAMI Q-FUNKCE

9.1 PRŮBĚH PŘEDUČENÍ

Proces předučení za použití pevné doby trvání přechodu τ_p (odst. 7.2) nevykazuje během prvních 1 000 průchodů tabulkou znaky konvergence. Přitom nezáleží ani na délce doby trvání přechodu, ani na volbě mřížky tabulky. Konvergentního procesu nebylo dosaženo ani při simulacích s řádově vyšším počtem průchodů tabulkou.

Podrobnějšími simulacemi bylo zjištěno, že proces předučení vykazuje tím rychlejší zlepšování průměrné úspěšnosti pokusů, čím vyšší počet přechodů do sousedních buněk stavů AML je uskutečněn během jednoho průchodu tabulkou. Klesne-li počet přechodů do sousedních buněk pod přibližně 40 %, přestane proces předučení vykazovat znaky konvergence. Pro zvýšení počtu přechodů do sousedních buněk stavů bylo proto použito proměnné (adaptivní) délky doby trvání přechodu τ_p, která byla inicializována jako vhodně malá a byla prodlužována tak dlouho, dokud soustava nedosáhla stavu, který patří do sousední buňky. Adaptivní dobou trvání přechodu stavu již může být získán konvergentní proces předučení. Rychlost předučení ovlivňuje především horní hranice intervalu možného trvání doby přechodu τ_p: čím je hranice vyšší, tím je i rychlost předučení vyšší. Tato hranice však nebyla prodlužována za 1×10^{-3}[s], aby doba výpočtu zůstala v přijatelných mezích. Jako rozumný kompromis mezi délkou doby výpočtu a rychlostí předučení se jeví interval $\langle 1\times10^{-5}, 1\times10^{-3}\rangle$[s], kterého bylo použito ve všech dalších simulacích.

Pro lineární mřížky 3-D i 2-D stavu s nejméně 120 buňkami stavů soustavy, tj. pro mřížku $(13, 11;\ 3)$ a pro všechny mřížky 3-D stavu, dosahuje proces předučení již po 800 průchodech tabulkou nejméně 94% průměrné úspěšnosti pokusů (s výjimkou občasných fluktuací), přičemž při použitých podmínkách simulací bylo dosaženo nejvýše 98% průměrné úspěšnosti pokusů.

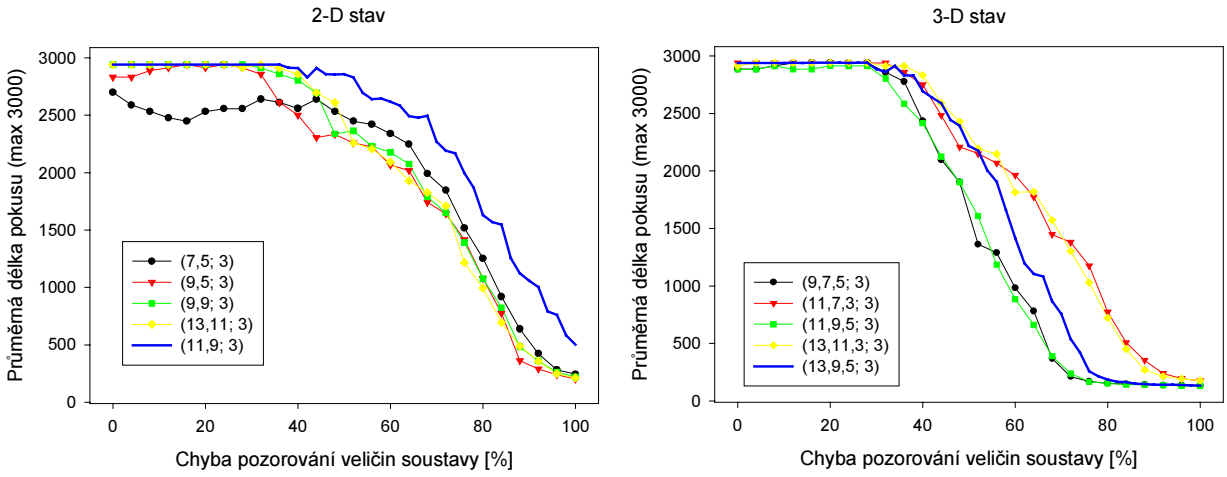
Dále je možno usuzovat, že čím nižší je horní hranice intervalu, tím více závisí rychlost předučení na volbě mřížky tabulky.

Použitelnost strategie získané ve fázi předučení byla dále ověřována simulacemi, během kterých byla testována jednak odolnost strategie vůči chybám pozorování veličin soustavy, jednak vůči zpoždění akčního zásahu.

9.2 ODOLNOST STRATEGIE VŮČI NÁHODNÝM CHYBÁM POZOROVÁNÍ VELIČIN SOUSTAVY

Výsledky testů jsou shrnuty na obr. 9.2-1. Chyby pozorování veličin soustavy byly zavedeny do všech veličin stavu soustavy. S výjimkou mřížek 2-D stavů, které definují méně nebo 32 buněk stavů soustavy, mají závislosti velmi podobný charakter: do přibližně 35 % úrovně chyb pozorování veličin soustavy je zachována průměrná úspěšnost pokusů dosažená při simulacích prováděných bez chyb pozorování. S dalším zvyšováním úrovně chyb pozorování veličin soustavy se průměrná délka pokusů snižuje.

Z mřížek 2-D stavu vykazuje nejméně strmý sestup mřížka $(11, 9; 3)$ se zlomovou hodnotou úrovně chyb pozorování asi 40 %. Pro 3-D stav vykazují prakticky stejný sestup mřížky $(11, 7, 3; 3)$ a $(13, 11, 3; 3)$, které mají zlomovou hladinu úrovně chyb pozorování asi na 35 %, a mřížka $(13, 9, 5; 3)$ vykazuje průměrně strmý sestup. Celkově lze usoudit, že z hlediska chyb pozorování veličin soustavy existují jak pro 2-D, tak pro 3-D stav AML optimální mřížky, které pouze nevýrazně ovlivňují strmost poklesu průměrné délky pokusů a téměř vůbec neovlivňují úroveň chyb pozorování veličin soustavy, při které sestup nastává. Obsahuje-li stav AML zrychlení rotoru $\ddot{x}$, je pokles tím méně strmý, čím méně je uzlových bodů rastru zrychlení.

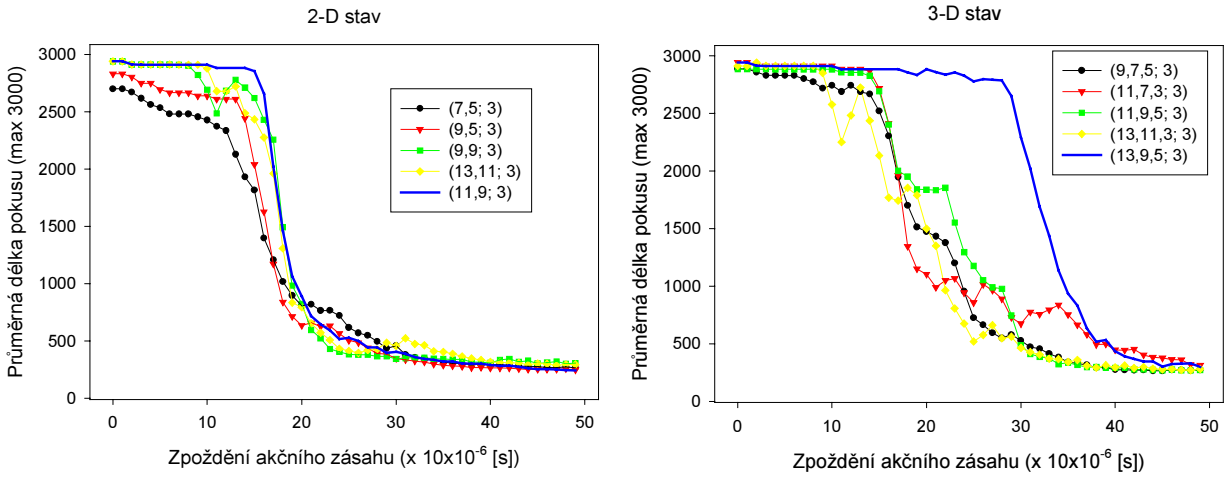


Obr. 9.2-1 Odolnost strategie vůči náhodné chybě pozorování veličin soustavy

Odolnost vůči chybám pozorování veličin soustavy testovaných lineárních mřížek je vysoká.

9.3 ODOLNOST STRATEGIE VŮČI ZPOŽDĚNÍ ŘÍDICÍHO ZÁSAHU

Výsledky jsou shrnuty na obr. 9.3-1. Nad hodnotou zpoždění přibližně $100\times10^{-6}\,[\mathrm{s}]$ začíná u 2-D stavu prudce klesat průměrná úspěšnost pokusů. S výjimkou mřížek, které definují 32 buněk stavů soustavy nebo méně, jsou závislosti opět velmi podobné. Strmost sestupu není volbou mřížek příliš ovlivněna. Nejlépe se chová mřížka $(11, 9; 3)$, u které se zlomová hodnota zpoždění posouvá o něco výše, asi na $130\times10^{-6}\,[\mathrm{s}]$. U 3-D stavu se zlomová hodnota zpoždění vyskytuje také na $130\times10^{-6}\,[\mathrm{s}]$, přičemž průběh poklesu je o něco méně strmý a více závisí na volbě mřížky. Vyniká mřížka $(13, 9, 5; 3)$, u které se posouvá zlomová hodnota zpoždění až na $280\times10^{-6}\,[\mathrm{s}]$. Mezi ostatními mřížkami příliš velký rozdíl není.



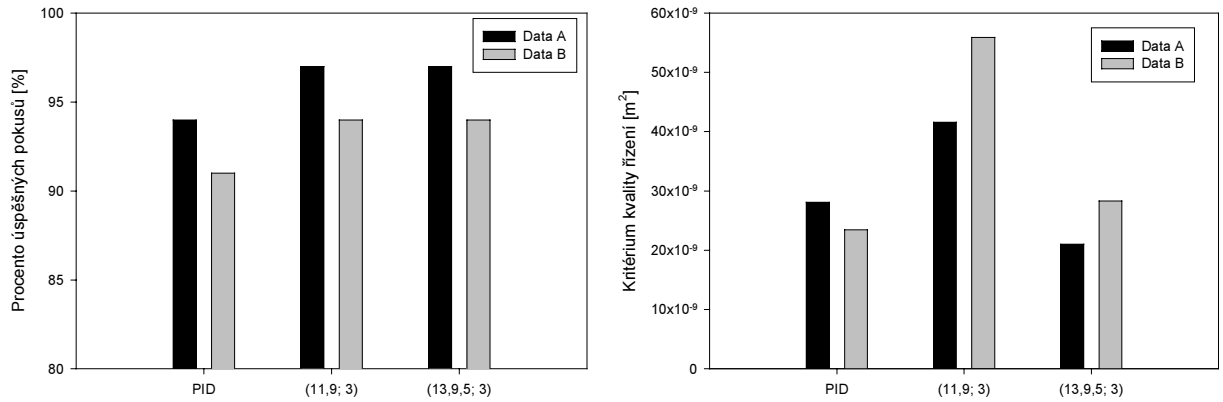
Obr. 9.3-1 Odolnost strategie vůči zpoždění akčního zásahu

9.4 VÝBĚR VHODNÉ MŘÍŽKY

Z posuzovaných mřížek 2-D stavu je jak z hlediska odolnosti vůči chybám pozorování veličin soustavy, tak vůči zpoždění provedení akčního zásahu, nejlepší mřížka $(11, 9; 3)$ a z mřížek 3-D stavu se chová nejlépe mřížka $(13, 9, 5; 3)$. Tato mřížka dosahuje výrazně nejlepší odolnosti vůči zpoždění provedení akčního zásahu a ještě nevykazuje nejhorší odolnost vůči chybám pozorování veličin soustavy. Dále bylo pro tyto dvě mřížky porovnáno procento úspěšných pokusů (odst. 8) strategie, která používala k aproximaci Q-funkce těchto mřížek, s AML řízeným referenčním PID regulátorem. Totéž bylo provedeno pro kritérium kvality řízení.

9.5 POROVNÁNÍ S REFERENČNÍM PID REGULÁTOREM

Výsledky testů strategie PID regulátoru a obou strategií s tabulkami vybranými v předchozím jsou uvedeny na obr. 9.5-1. PID regulátor dosáhl na datech A 94 % a na datech B 91 % úspěšných pokusů, na rozdíl od strategií obou tabulek, které shodně dosáhly 97 % úspěšných pokusů na datech A a 94 % na datech B.



Obr. 9.5-1 Porovnání získaných strategií s referenčním PID regulátorem

Celkově tedy dosáhly obě strategie získané v fázi předučení o něco vyššího procenta úspěšných pokusů než strategie referenčního PID. Strategie s mřížkou $(11, 9; 3)$ dosáhla horší hodnoty kritéria kvality řízení než strategie PID regulátoru, strategie s mřížkou $(13, 9, 5; 3)$ hodnoty zhruba srovnatelné.

Obou vybraných mřížek bylo použito jako výchozích pro všechny další simulace.

10 FÁZE PŘEDUČENÍ S NELINEÁRNÍMI MŘÍŽKAMI Q-FUNKCE

Simulace byly navrženy s cílem ověřit, zda zmenšením vzdáleností uzlových bodů nejbližších uzlovému bodu, který reprezentuje nulovou hodnotu veličiny (veličin), dojde ke zvýšení „rozlišovací schopnosti“ strategie získané předučením a tím i ke zlepšení kritéria kvality řízení $q(T)$. Dalším cílem bylo zjistit, jak se změní průběh procesu předučení, odolnost získané strategie vůči náhodným chybám pozorování veličin soustavy i vůči zpoždění provedení řídicího zásahu.

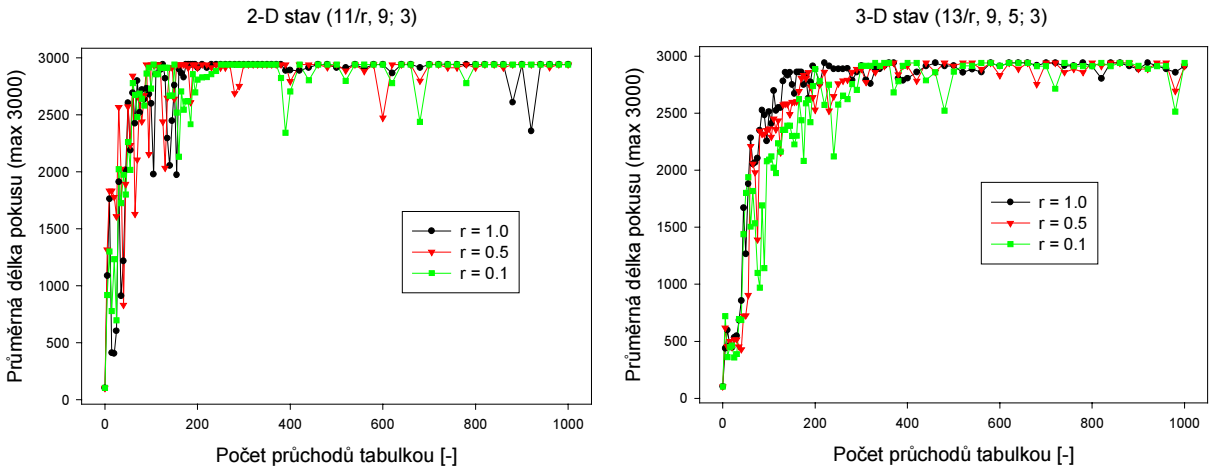
Použito bylo tabulek s lineárními mřížkami $(11, 9; 3)$ a $(13, 9, 5; 3)$, se kterými bylo dosaženo nejlepších strategií (odst. 9). Nelineárního rastru bylo použito pouze pro výchylku rotoru x, a to jak pro 2-D, tak pro 3-D stav AML.

10.1 PRŮBĚH PŘEDUČENÍ

Pro mřížku 2-D stavu byl proces předučení s prohlubující se nelinearitou o něco rychlejší, než pro mřížku 3-D stavu (viz obr. 10.1-1).

S výjimkou občasných fluktuací dosahoval proces předučení na mřížce 2-D stavu nejméně 94% průměrné úspěšnosti pokusů již po 300 průchodech tabulkou na rozdíl od mřížky 3-D stavu, kde bylo této hodnoty dosaženo až po 400 průchodech tabulkou. Na mřížce 3-D stavu docházelo sice k častějším, ale méně hlubokým fluktuacím, než na mřížce 2-D stavu.

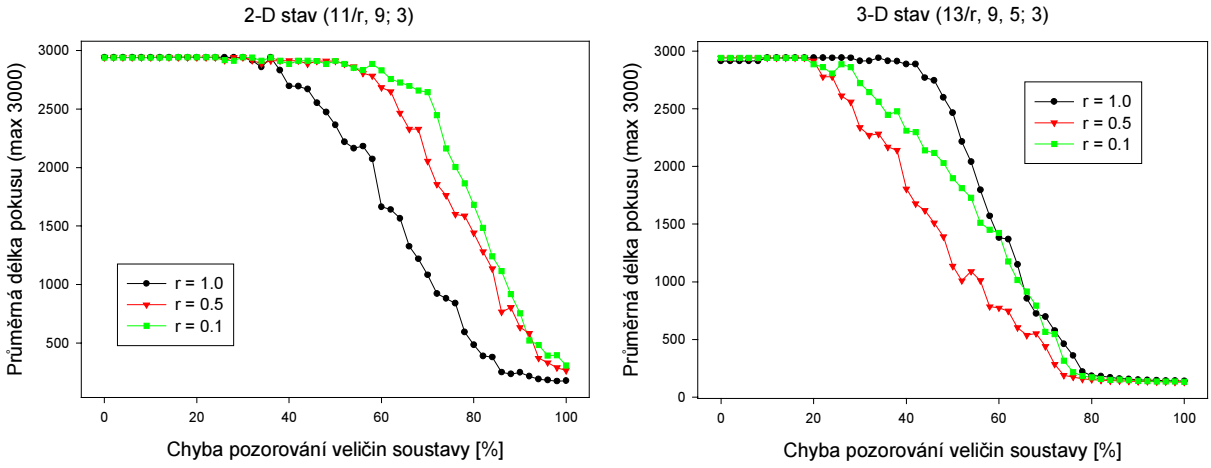
Proces předučení však pro obě nelineární mřížky probíhal prakticky stejně jako na obou lineárních mřížkách.



Obr. 10.1-1 Průběh předučení

10.2 ODOLNOST STRATEGIE VŮČI NÁHODNÝM CHYBÁM POZOROVÁNÍ VELIČIN SOUSTAVY

Odolnost dosažené strategie vůči chybám pozorování veličin soustavy se při 2-D stavu pro nelineární variantu vybrané mřížky oproti lineární variantě s prohlubováním nelinearity mřížky zvyšuje (viz obr. 10.2-1, 2-D stav). Zlomová úroveň chyb pozorování veličin soustavy se pro $r = 0.5$ i $r = 0.1$ posouvá z původních 40 % až na téměř 61 %, přičemž pro $r = 0.5$ je sestup poněkud méně strmý.



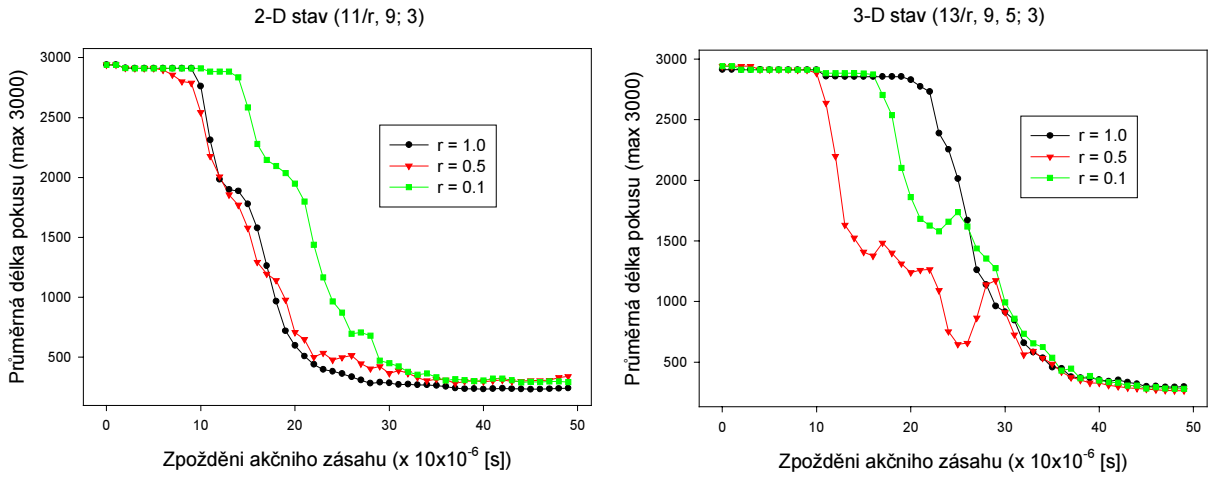
Obr. 10.2-1 Odolnost strategie vůči náhodné chybě pozorování veličin soustavy

Zcela odlišná je situace u vybrané mřížky 3-D stavu. Pro nelineární 3-D mřížku se odolnost dosažené strategie vůči chybám pozorování zhorší (viz obr. 10.2-1, 3-D stav). Největší zhoršení nastane při koeficientu nelinearity $r = 0.5$, kdy se zlomová hodnota úrovně chyb pozorování sníží až na 20 %. Pro $r = 0.1$ je sice zlomová hodnota úrovně chyb pozorování také přibližně 20 %, avšak sestup je méně strmý, než v případě $r = 0.5$.

10.3 ODOLNOST STRATEGIE VŮČI ZPOŽDĚNÍ ŘÍDICÍHO ZÁSAHU

U mřížky 2-D stavu dochází obdobně ke zlepšení odolnosti dosažené strategie vůči zpoždění řídicího zásahu (obr. 10.3-1). Zlomová hodnota se při koeficientu nelinearity $r = 0.1$ posouvá z původních 130×10^{-6} [s] na 170×10^{-6} [s] a snižuje se strmost poklesu. Pro $r = 0.5$ se zlomová hodnota zhoršuje, stejně jako pro lineární mřížku s přepočtenou množinou cílových stavů.

U mřížky 3-D stavu dochází ke zhoršení odolnosti strategie vůči zpoždění řídicího zásahu. Největší zhoršení nastane (jako u odolnosti strategie vůči náhodným chybám pozorování veličin soustavy) při koeficientu nelinearity $r = 0.5$, kdy se zlomová hodnota posouvá z 280×10^{-6} [s] na 110×10^{-6} [s]. Pro $r = 0.1$ není posun tak markantní; hodnota se posouvá na 170×10^{-6} [s].



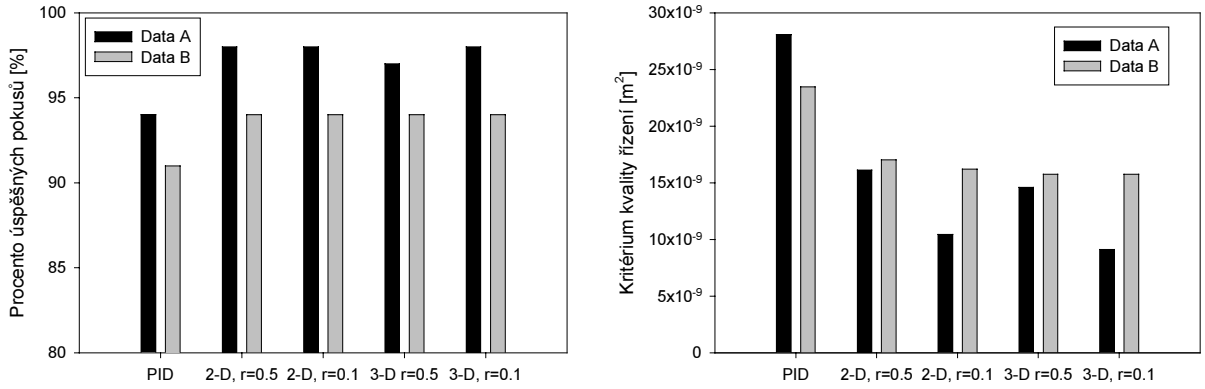
Obr. 10.3-1 Odolnost strategie vůči zpoždění akčního zásahu

10.4 VÝBĚR VHODNÝCH MŘÍŽEK

Z testovaných mřížek 2-D stavu se chová nejlépe nelineární mřížka $(11/0.1, 9; 3)$, která výrazně zlepšuje odolnost dosažené strategie vůči chybám pozorování, ale již méně zlepšuje odolnost vůči zpoždění řídicího zásahu. Z mřížek 3-D stavu je nejlepší lineární mřížka $(13, 9, 5; 3)$. U zbývajících mřížek dojde k zhoršení jak odolnosti strategie vůči náhodným chybám pozorování veličin soustavy, tak vůči zpoždění řídicího zásahu.

10.5 POROVNÁNÍ S REFERENČNÍM PID REGULÁTOREM

Výsledky testů na porovnání získaných strategií s referenčním PID regulátorem (procento úspěšných pokusů a kritérium kvality řízení) pro 2-D i 3-D stavy jsou uvedeny na obr. 10.5-1.



Obr. 10.5-1 Porovnání získaných strategií s referenčním PID regulátorem

Strategie s nelineárními mřížkami jak 2-D, tak 3-D stavu dosáhly na datech A 97 % až 98 % a shodně na datech B 94 % úspěšných pokusů. PID regulátor dosáhl na datech A 94 % a na datech B 91 % úspěšných pokusů, na rozdíl od obou strategií s vybranými mřížkami, které shodně dosáhly 98 % úspěšných pokusů na datech A a 94 % na datech B.

Strategie 2-D stavu s mřížkou $(11/0.5, 9; 3)$ vykazovala na datech A i B hodnoty kritéria kvality řízení $q(T)$ lepší než PID regulátor (data A 17×10^{-9} [m^2], data B 18×10^{-9} [m^2]), ještě o něco lepší byla mřížka $(11/0.1, 9; 3)$ (data A 11×10^{-9} [m^2], data B 17×10^{-9} [m^2]).

Strategie 3-D stavu jak s mřížkou $(13/0.5, 9, 5; 3)$, tak s mřížkou $(13/0.1, 9, 5; 3)$ vykazuje na datech B hodnotu asi 15×10^{-9} [m^2], která je opět lepší, než hodnota dosažená referenčním PID regulátorem. Na datech A je lepší mřížka $(13/0.1, 9, 5; 3)$ s hodnotou 9×10^{-9} [m^2].

Celkově dosáhly strategie s nelineárními mřížkami zhruba stejného procenta úspěšných pokusů jako obě strategie s lineárními mřížkami, přičemž hodnoty kritéria kvality řízení $q(T)$ jsou podstatně lepší než hodnoty, jakých dosáhl referenční PID regulátor. U strategií s nelineárními mřížkami došlo k výraznému zlepšení oproti oběma strategiím s lineárními mřížkami. Protože byla při hodnocení dosažených strategií dávána přednost robustnosti (odst. 9), byly jako nejlepší vyhodnocena nelineární mřížka 2-D stavu $(11/0.1, 9; 3)$ a lineární mřížka 3-D stavu $(13, 9, 5; 3)$. Těchto mřížek je používáno ve všech dalších testech. Pro porovnání je používáno i lineární mřížky 2-D stavu $(11, 9; 3)$ a v některých případech nelineární mřížky 3-D stavu $(13/0.1, 9, 5; 3)$.

11 DALŠÍ ZKUŠENOSTI S FÁZÍ PŘEDUČENÍ

Pro úplnost jsou uvedeny výsledky některých dalších simulací, které dokreslují vlastnosti strategií získaných ve fázi předučení.

11.1 POSILOVACÍ FUNKCE

V této skupině simulací byl testován vliv různých posilovacích funkcí jednak na rychlost předučení, jednak na strategii dosaženou ve fázi předučení.

.

11.1.1 Průběh předučení

Volba posilovací funkce ovlivnila průběh předučení pouze velmi málo. S lineární mřížkou 2-D stavu probíhalo předučení nejrychleji a velmi brzy vykazovalo nejvyrovnanější průměrné délky pokusů. Nelineární mřížka 2-D stavu vykazovala sice nejméně vyrovnané průměrné délky pokusů ze všech tří mřížek, a to pro všechny posilovací funkce. Naproti tomu lineární mřížka 3-D stavu vykazovala nejvyšší závislost na volbě posilovací funkce během prvních průchodů tabulkou. Nejlépe se choval proces používající prosté omezené pokuty.

11.1.2 Odolnost strategie vůči náhodným chybám pozorování veličin soustavy

Při lineární mřížce 2-D stavu vykazovala nejvyšší odolnost vůči náhodné chybě pozorování veličin strategie získaná použitím posilovací funkce ve tvaru prosté omezené pokuty se zlomovou hladinou chyby pozorování okolo 40 %. U strategií získaných použitím ostatních posilovacích funkcí se zlomová hladina chyby pozorování pohybovala okolo 20 %. U nelineární mřížky 2-D stavu bylo nejlepší strategie dosaženo opět prostou omezenou pokutou. Tato strategie vykazovala zlomovou hladinu chyby pozorování asi 50 %. Prakticky stejná zlomová hladina chyby pozorování byla získána kvadratickou omezenou pokutou. U nelineární mřížky 3-D stavu různé volby posilovacích funkcí prakticky neovlivnily odolnost získaných strategií. Pro kvadratickou omezenou pokutu bylo dosaženo mírného zlepšení z původních 35 % na asi 40 %.

11.1.3 Odolnost strategie vůči zpoždění řídicího zásahu

Při lineární mřížce 2-D stavu žádná z prezentovaných posilovacích funkcí výrazně neovlivnila odolnost vůči zpoždění akčního zásahu. Díky snižování odolnosti ještě před zlomovou hodnotou je možno za nejhorší označit strategii získanou použitím kvadratické pokuty.

Pro nelineární mřížku 2-D stavu jsou získané strategie ovlivněny volbou posilovací funkce ještě méně než pro lineární.

Naproti tomu strategie vyvinuté na lineární mřížce 3-D stavu na volbě posilovací funkce závisejí. Nejlepších výsledků bylo dosaženo použitím prosté omezené pokuty, která dosáhla zlomové hodnoty asi 280×10^{-6} [s], i když za mírného snížení průměrné délky pokusu.

Ostatní posilovací funkce dosáhly horších zlomových hodnot (kvadratická omezená pokuta asi 250×10^{-6} [s] a kvadratická pokuta pouhých 150×10^{-6} [s]). Strmost sestupu byla pro všechny volby posilovací funkce prakticky stejná.

Na procento úspěšných pokusů neměl způsob průchodu tabulkou vliv.

11.1.4 Porovnání s referenčním PID regulátorem

Z hlediska procenta úspěšných pokusů i hodnoty kritéria kvality řízení byla překvapivě nejhorší kvadratická pokuta ve spojení s lineární mřížkou 2-D stavu. S procentem úspěšných pokusů na datech B pouhých 88 % a kritériem kvality na datech B 39×10^{-9} [m^2] byla výrazně horší než referenční PID regulátor. Kvadratická omezená pokuta s lineární mřížkou 2-D stavu dosáhla na da-

tech A i B lepších hodnot jak v procentech úspěšnosti pokusů, tak v kritériu kvality než referenční PID regulátor. U kritéria kvality jsou hodnoty lepší než s prostou omezenou pokutou. Co se týče lineární mřížky 3-D stavu, byly pro obě kvadratické pokuty výsledky prakticky stejné, jako v případě lineární mřížky 2-D stavu a kvadratické omezené pokuty.

Na nelineární mřížce 2-D stavu se chová nejlépe prostá omezená pokuta, která na obou datových souborech vykazuje nejlepší procento úspěšnosti pokusů i nejvyšší kritérium kvality.

Celkově bylo dosaženo nejlepších výsledků s posilovací funkcí ve tvaru prosté omezené pokuty, která proto byla používána ve standardních podmínkách simulací.

11.2 ZPŮSOB PRŮCHODU TABULKOU

V této skupině simulací byl testován vliv různých způsobů průchodu tabulkou Q-funkce, navíc se synchronním průchodem, a to opět jednak na rychlost předučení, jednak na strategii dosaženou během 1 000 průchodů tabulkou Q-funkce.

11.2.1 Průběh předučení

Volba způsobu průchodu tabulkou nemá na průběh předučení podstatný vliv. S lineární mřížkou 2-D stavu byl získán nejrychlejší průběh předučení na začátku této fáze a nejvyrovnanější průměrné délky pokusů po asi 350 průchodech tabulkou. Naproti tomu nelineární mřížka 2-D stavu vykazuje ze všech tří mřížek nejvyšší závislost na volbě způsobu průchodu tabulkou. Konečně s lineární mřížkou 3-D stavu bylo na počátku fáze předučení dosaženo nejnižší rychlosti předučení s průměrnou závislostí na volbě způsobu průchodu tabulkou.

11.2.2 Odolnost strategie vůči náhodným chybám pozorování veličin soustavy

Při lineární mřížce 2-D stavu vykazuje nejvyšší odolnost vůči chybě pozorování strategie získaná sekvenčním dopředným průchodem tabulkou se zlomovou hladinou chyby pozorování okolo 40 %. U strategií získaných ostatními způsoby průchodů se zlomová hladina chyby pozorování pohybovala okolo 25 %. U nelineární mřížky 2-D stavu byla nejlepší strategie dosažena synchronním průchodem tabulkou. Tato strategie vykazovala zlomovou hladinou chyby pozorování až na 65 %. Ostatní způsoby průchodu tabulkou vedly na strategie se zlomovou hladinou chyby pozorování, která se pohybovala okolo 55 %. U nelineární mřížky 3-D stavu různé způsoby průchodů tabulkou odolnost získaných strategií vůči náhodným chybám pozorování soustavy prakticky neovlivnily.

11.2.3 Odolnost strategie vůči zpoždění řídicího zásahu

Při lineární mřížce 2-D stavu sekvenční dopředný, sekvenční zpětný ani synchronní způsob průchodu tabulkou odolnost vůči zpoždění akčního zásahu neovlivňuje. Náhodný způsob průchodu odolnost výrazně zhoršuje. Při zachování strmosti poklesu závislosti dochází k posunu zlomové hodnoty zpoždění ze 150×10^{-6} [s] až na 100×10^{-6} [s].

Při nelineární mřížce 2-D stavu zachovává zlomovou hodnotu poklesu závislostí sekvenční dopředný, sekvenční zpětný a náhodný způsob průchodu tabulkou. Synchronní způsob odolnost výrazně zhoršuje posunutím zlomové hodnoty opět až na 100×10^{-6} [s].

Při lineární mřížce 3-D stavu je nejlepší sekvenční dopředný způsob průchodu, který dosahuje zlomové hodnoty asi 280×10^{-6} [s]. Ostatní způsoby průchodu jsou s hodnotou asi 200×10^{-6} [s] zřetelně horší.

11.2.4 Porovnání s referenčním PID regulátorem

Nejlepších strategií bylo dosaženo při volbě sekvenčního dopředného způsobu, který byl z tohoto důvodu zařazen do standardních podmínek experimentu. Ostatní způsoby průchodu tabulkou negativně ovlivňují zejména odolnost vůči zpoždění akčního zásahu, a to jak u mřížky 2-D stavu, tak u mřížky 3-D stavu, a dále u mřížky 2-D stavu odolnost vůči chybám pozorování veličin soustavy.

Na procento úspěšných pokusů nemá způsob průchodu tabulkou vliv.

Co se týče hodnoty kritéria kvality řízení, dosahuje u mřížky 2-D stavu náhodný i synchronní průchod lepších hodnot než referenční PID regulátor. U mřížky 3-D stavu dosáhl lepších hodnot než PID regulátor sekvenční zpětný průchod a opět náhodný průchod.

11.3 MNOŽINA AKCÍ

Tato skupina simulací byla navržena s cílem zjistit, zda bude rychlost předučení, odolnost dosažené strategie vůči chybám pozorování veličin soustavy a vůči zpoždění akčního zásahu pozitivně ovlivněna rozšířením množiny akcí $A = \{-u_{max}, 0, u_{max}\}$, tj. $\{-100, 0, 100\}\,[\mathrm{V}]$ na množinu $\{-100, -50, 0, 50, 100\}\,[\mathrm{V}]$. Použito bylo opět obou vybraných lineárních mřížek tabulky Q-funkce $(11, 9; 3)$ a $(13, 9, 5; 3)$ i nelineární varianty $(11/0.1, 9; 3)$.

11.3.1 Průběh předučení

Vzhledem k předučení, které používalo původní tříprvkové množiny akcí, došlo ve všech případech ke zrychlení procesu předučení. Nejvyšší rychlosti předučení bylo dosaženo na lineární mřížce 2-D stavu. Na nelineární mřížce 2-D stavu a lineární mřížce 3-D stavu byla rychlost předučení mírně nižší.

11.3.2 Odolnost strategie vůči náhodným chybám pozorování veličin soustavy

Pro lineární mřížku 2-D stavu zůstala odolnost výsledné strategie vůči náhodným chybám pozorování veličin prakticky zachována. Zlomová hodnota úrovně činila 42 % (z původních 40 % v případě tříprvkové množiny akcí). Ke zhoršení na 24 % došlo jak pro nelineární mřížku 2-D stavu (z původních 61 %), tak pro lineární mřížku 3-D stavu (původně 35 %).

11.3.3 Odolnost strategie vůči zpoždění řídicího zásahu

Výrazný pokles proti použití původní tříprvkové množiny akcí zaznamenala také zlomová hodnota zpoždění akčního zásahu. U lineárních mřížek 2-D i 3-D stavu činila zlomová hodnota přibližně 140×10^{-6} [s] (původní hodnoty 150×10^{-6} [s] a 280×10^{-6} [s]), u nelineární mřížky 2-D stavu dokonce 80×10^{-6} [s] (z původních 150×10^{-6} [s]).

11.3.4 Porovnání s referenčním PID regulátorem

Co se týče procenta úspěšnosti pokusů, došlo u dat A k jeho mírnému zhoršení z 98 % u nelineární mřížky 2-D stavu a lineární mřížky 3-D stavu na z 97 %, pro data B je procento úspěšnosti zachováno.

Kritérium kvality řízení se zlepšilo u lineární mřížky 2-D stavu (z původních 41×10^{-9} [m^2] pro data A a 55×10^{-9} [m^2] pro data B na 21×10^{-9} [m^2] a 47×10^{-9} [m^2]) a nepatrně u lineární mřížky 3-D stavu (z 21×10^{-9} [m^2] a 28×10^{-9} [m^2] na 20×10^{-9} [m^2] a 2×10^{-9} [m^2]). U nelineární mřížky 2-D došlo ke zhoršení (z 11×10^{-9} [m^2] a 17×10^{-9} [m^2] na 15×10^{-9} [m^2] a 21×10^{-9} [m^2]).

Celkově je možno konstatovat, že rozšířením množiny akcí došlo ke snížení odolnosti strategie jak vůči chybám pozorování veličin soustavy, tak i vůči zpoždění akčního zásahu při zlepšení kritéria kvality $q(T)$. Protože byla při výběru mřížky i parametrů obou fází Q-učení preferována robustnost získané strategie a jednoduchost jeho budoucí technické realizace, nebylo rozšíření množiny řídicích akcí prozatím využito.

12 DOUČOVÁNÍ

Fáze doučování byla prováděna za standardních podmínek simulací (odst. 8.1) použitím konvenční organizace Q-učení. Pokusy byly generovány pomocí dat A, tj. maximální délka pokusu byla omezena provedením 3 000 řídicích zásahů a počáteční stavy pokusů byly nastavovány náhodně. Chování AML bylo opět simulováno pomocí výpočtového modelu, stejně jako ve fázi předučení (odst. 7.1). Výsledky některých testů jsou uvedeny v tab. 12-1.

Popis	Před doučováním		Po doučování		Zlepšení
	%	$q(T)$ [m^2]	%	$q(T)$ [m^2]	%
Lineární mřížka (11, 9; 3)	98	$41,0\times10^{-9}$	98	$25,9\times10^{-9}$	37
Lineární mřížka (13, 9, 5; 3)	98	$20,9\times10^{-9}$	98	$20,3\times10^{-9}$	3
Nelineární mřížka (11/0.1, 9; 3)	98	$10,6\times10^{-9}$	98	$10,2\times10^{-9}$	4
Nelineární mřížka (13/0.1, 9, 5; 3)	98	$9,4\times10^{-9}$	98	$9,0\times10^{-9}$	4

Tab. 12-1 Výsledky fáze doučování. % značí procento úspěšných pokusů

Během fáze doučování došlo k dalšímu zlepšení strategií získaných z procesu předučení. Zlepšení ale bylo často nevýrazné. Procento úspěšných pokusů zůstávalo beze změny na 98 %, ke kvantifikovatelnému zlepšení došlo u kritéria kvality $q(T)$. Zde dosahuje zlepšení typicky asi 4 %.

Fází doučování na lineárních variantách mřížek 2-D stavu i 3-D stavu bylo dosaženo hodnot kritéria kvality, které se pohybují okolo 22×10^{-9} [m^2], což může znamenat u lineární mřížky 2-D stavu zlepšení až o 30 %.

Strategie s nelineárními variantami mřížek 2-D stavu i 3-D stavu dosahují hodnoty kritéria kvality asi 11×10^{-9} [m^2].

Velké zlepšení u lineární mřížky 2-D stavu je možno vysvětlit pomalejším průběhem Q-učení členu s touto variantou mřížky ve fázi předučení. Během 1 000 průchodů tabulkou Q-hodnot je dosaženo relativně hrubého odhadu optimálních Q-hodnot. Odhad je zlepšen teprve ve fázi doučování.

Zajímavější je skutečnost, že se během fáze doučování vyrovnal rozdíl ve strategiích 2-D stavu a 3-D stavu používajících lineární variantu mřížek; obdobný výsledek byl zaznamenán i u mřížek nelineárních. V rámci dané úlohy tak byl získán důležitý závěr, že patrně není výsledná strategie ani tak ovlivňována tím, jestli pracuje s 2-D nebo 3-D stavem (i když strategie 3-D stavů vykazují mírně lepší výsledky), jako tím, jestli pracuje s nelineární mřížkou.

13 ZÁVĚR

V předložené práci byla navržena metoda Q-učení pro realizaci adaptivního optimálního řízení, která se skládá ze speciálně organizované fáze předučení a konvenčně organizované fáze doučování. Během fáze předučení jsou na výpočtovém modelu prováděny elementární pokusy, které

jsou zpracovávány prováděním zálohování Q-učení v reálném čase. Výpočtový model může být pouze přibližný. Zdůvodnění konvergence fáze předučení (odst. 6) je možno chápat i jako neformální důkaz konvergence této fáze. Protože je provádění elementárních pokusů speciálním případem provádění pokusů, je navržená metoda obecně použitelná nejen s Q-učením, ale také s asynchronní iterací hodnot v reálném čase založenou na pokusu.

Dále bylo v práci provedeno simulační ověření navržené metody na jednoduchém výpočtovém modelu AML s jedním stupněm volnosti (výchylka rotoru v rovině kolmé na osu stroje).

Překvapivě vysoké rychlosti předučení bylo dosaženo s adaptivní délkou doby trvání přechodu. Již po 1 000 průchodech tabulkou Q-funkce bylo jak pro 3-D stav AML (konstruovaný z výchylky, rychlosti a zrychlení rotoru), tak pro 2-D stav (konstruovaný pouze z výchylky a rychlosti rotoru) dosaženo použitelné strategie řízení. Navržená metoda umožnila překonat zásadní problém s použitím konvenčního Q-učení (nebo s iterací hodnot v reálném čase založené na pokusu), kterou je nestabilita samotného AML. Ta má při konvenčním přístupu za následek velmi pomalou konvergenci Q-učení, pokud vůbec ke konvergenci dojde. Získané strategie řízení byly posuzovány nejprve z hlediska dosažené průměrné délky pokusů a dále z hlediska odolnosti dosažených strategií vůči chybám pozorování soustavy a odolnosti vůči zpoždění akčního zásahu. Pro kontrolu byly strategie nakonec hodnoceny z hlediska dosaženého procenta úspěšných pokusů na kontrolních datech a dosažené hodnoty integrálního kritéria kvality řízení byly porovnány s referenčním PID regulátorem.

Tímto způsobem byly vybrány 2 mřížky tabulky Q-funkce, jedna pro 2-D stav AML a druhá pro 3-D stav AML (odst. 9, 10) s velmi malým počtem buněk stavů AML (konkrétně 80 a 96), a tím velmi jednoduchou architekturou řídicího členu již ve fázi předučení. Strategie vykazují vysokou odolnost vůči náhodným chybám pozorování veličin soustavy a dosahují lepších charakteristik chování než referenční PID regulátor, který byl převzat z literatury.

Strategii získanou ve fázi předučení dramaticky neovlivňuje ani volba posilovací funkce, ani způsob průchodu tabulkou, ani rozšíření množiny akcí. Lepších charakteristik dosahuje strategie 3-D stavu AML než 2-D stavu (odst. 11).

Během fáze doučování dochází k dalšímu zlepšení strategií získaných z procesu předučení (odst. 12). Zlepšení je ale často nevýrazné. Během této fáze se vyrovnává rozdíl mezi strategiemi 2-D a 3-D stavu, které používají lineární i nelineární variantu mřížek, i když strategie 3-D stavů vykazují mírně lepší výsledky.

Teoretický přínos představuje návrh nové metody Q-učení a zdůvodnění konvergence fáze předučení.

Praktický přínos této práce je možno spatřovat v simulačním ověření vysoké efektivnosti navržené nové metody Q-učení i v jejím úspěšném použití na obtížné úloze, jakou je z hlediska konvenčního Q-učení řízení AML. Simulace a učení byly prováděny na původním – pro tento účel vyvinutém programovém vybavení, které je dalším praktickým přínosem práce.

Aktuálnost problematiky je vysoká. Vyhovuje současnému trendu výzkumu nových metod řízení, které jsou založeny na využití metod UI, zejména učení. Základním rysem učení je rozvinutá schopnost adaptace, tj. schopnost automaticky zlepšovat chování řízené soustavy např. při změně provozních parametrů apod.

Ve spolupráci s fakultou elektrotechniky a komunikačních technologií VUT v Brně je v současné době dokončován fyzikální model aktivního magnetického ložiska a byl zahájen vývoj řídicí elektroniky s cílem praktického odzkoušení navržené metody. Zahájeny byly rovněž práce na simulačním ověření použitelnosti navržené metody při řízení asynchronního elektromotoru, kde dosavadní výsledky jsou rovněž slibné.

Uvedené výsledky vznikly při řešení grantu GAČR 101/00/1471 „Stabilita řízení rotorů na magnetických ložiskách“ a výzkumných záměrů MSM 262100024 „Výzkum a vývoj mechatronických soustav“ a CEZ: J22/98: 261100009 „Netradiční metody studia komplexních a neurčitých systémů“.

14 LITERATURA

[1] Anderson, C. W.: Strategy Learning with multilayer connectionist representations. Tech. Report TR87-509.3, GTE Laboratories, Incorporated, Waltham, MA, 1987

[2] Barto, A., Singh, S.: On the computational economics of reinforcement learning. In: D. S. Touretzky, J. L. Elman, T. J. Sejnowski and G. E. Hinton, eds., Connectionist Models: Proceedings of the 1990 Summer School, Morgan Kaufmann, San Mateo, CA, 1991, 35–44

[3] Barto, A. G., Sutton, R. S., Anderson, C. W.: Neuronlike elements that can solve difficult learning control problems. IEEE Trans. Syst. Man Cybern. 13, 1983, 835–846

[4] Barto, A. G., Sutton, R. S., Watkins, C.: Sequential decision problems and neural networks. In: D. S. Touretzky, ed., Advances in Neural Information Processing Systems 2, Morgan Kaufmann, San Mateo, CA, 1990, 686–693

[5] Barto, A. G., Sutton, R. S., Watkins, C. J. C. H.: Learning and sequential decision making. In: M. Gabriel and J. Moore. eds., Learning and Computational Neuroscience: Foundations of Adaptive Networks, MIT Press, Cambridge, MA, 1990, 539–612

[6] Bellman, R., Dreyfus, S. E.: Functional approximations and dynamic programming. Math Tables and Other Aides to Compu1aton 13, 1959, 247–251

[7] Bellman, J. R., Kalaba, R.: Dynamic programming and Modern Control Theory, Academic Press New York London, 1965

[8] Bellman, J. R., Kalaba, R., Kotkin, B.: Polynomial approximation - a new computational technique in dynamic programming: allocation processes, Math. Comp. 17, 1973, 155–161

[9] Bellman, J. R.: Dynamic Programming. Princeton University Press, Princeton, NJ, 1957

[10] Bertsekas, D. P.: Distributed dynamic programming, IEEE Trans. Autom. Control 27, 1982, 610–616

[11] Bertsekas, D. P.: Dynamic Programming: Deterministic and Stochastic Models, Prentice-Hall, Eng1ewood Cliffs, NJ, 1987

[12] Bertsekas, D. P., Tsitsiklis, J. N.: Parallel and Distributed Computation: Numerical Methods, Prentice-Hall, Englewood Cliffs, NJ, 1989

[13] Bradtke, S. J.: Reinforcernent learning applied to linear quadratic regulation. In: C. L. Giles, S. J. Hanson and J. D. Cowan, eds., Advances in Neural Information Processing 5, Morgan Kaufmann, San Mateo, CA, 1993, 295–302

[14] Březina T., Krejsa J.: The Control of Active Magnetic Bearing Using Reinforcement Learning, Diagnostika a Aktivní řízení 2000, Třešť, 2000, 7–8, CD ROM

[15] Březina T., Krejsa, J., Kratochvíl, C.: Reinforcement Learning: Control of the Magnetic Bearing, Eight Con. on Nonlinear Vibrations, Stability and Dynamics of Structures, July 2000, Blacksburg, Virginia 24061

[16] Březina, T.: Effect of Table Grid at Q-learning Process of Active Magnetic Bearing Controller. Mendel 2001, Int. Conf., Brno 2001, 215–220

[17] Březina, T., Ehrenberger, Z., Kratochvíl, C.: Reinforcement learning model: control of nonlinear and unstable processes. Engineering Mechanics 2001, Svratka, 2001, CD ROM

[18] Březina, T.: Learning in Mechatronic Conceptions. Engineering MECHANICS, Vol. 8, 2001, No. 6, 431–442

[19] Březina, T., Krejsa, J., Věchet, S.: Stochastic Policy in Q-learning used for Control of AMB, Inženýrská Mechanika 2002, 7–8, Svratka, 2002, CD ROM

[20] Březina, T., Krejsa, J.: Q-learning used for Control of AMB: Reduced State Definition, Mendel 2002, 347–352, Brno, 2002

[21] Březina, T., Krejsa, J.: Efficient Q-learning Modification Applied on Active Magnetic Bearing Control, Engineering Mechanics, Brno (v tisku)

[22] Chapman, D., Kaelbling, L. P.: Input generalization in delayed reinforcement learning: an algorithm and performance comparisons. In: Proceedings IJCAI-91, Sydney, NSW, 1991

[23] Danie1, J. W.: Splines and efficiency in dynamic programming. J. Math. Anal. Appl. 54, 1976, 402–407

[24] Darken, C., Moody, J.: Note on learning rate schedule for stochastic optimization. In: R. P. Lippmann, J. E. Moody and D. S. Touretzky, eds., Advances in Neural information Processing Systems 3, Morgan Kaufmann, San Mateo, CA, 1991, 832–838

[25] Dayan, P.: Navigating through temporal difference. In: R. P. Lippmann, J. E. Moody and D. S. Touretzky, eds., Advances in Neural Information Processing Systems 3, Morgan Kaufmann, San Mateo, CA, 1991, 464–470
[26] Dayan, P.: Reinforcing connectionism: learning the statistical way, Ph.D. Thesis, University of Edinburgh, Edinburgh, Scotland, 1991
[27] Dayan, P.: The convergence of TD(λ) for general λ. Mach. Learn. 8, 1992, 341–362
[28] Dean, T. L., Wellman, M. P.: Planning and Control, Morgan Kaufmann, San Mateo, CA, 1991
[29] Duda, R. O., Hart, P. E.: Pattern Classification and Scene Analysis. Wiley, New York, 1973
[30] Jalali, A., Ferguson, M.: Computationally efficient adaptive control algorithms for Markov chains. In: Proceedings 28th Conference on Decision and Control, Tampa. FL, 1989, 1283–1288
[31] Kaelbling, L. P.: Learning: in Embedded Systems, MIT Press, Cambridge, MA, 1991; revised version of: Teleos Research TR-90-04, 1990
[32] Klapka, J.: Numerical Stability of Dynamic Programming. Mendel 2001, International Conference, Brno, 2001, 228–231
[33] Klopf, A. H.: The Hedonistic Neuron: A Theory of Memory. Learning, and Intelligence, Hermishere, Washington, DC, 1982
[34] Korf, R. E.: Real-time heuristic search. Artif. Intell 42, 1990, 189–211
[35] Kozánek, J., Pesson, T., Kocanda, L.: Stabilita rotoru na magnetickém ložisku. Interakce a zpětné vazby ’98, ÚT AV ČR, Praha 1998, 23–29
[36] Kumar, P. R.: A survey of some results in stochastic adaptive control, SIAM J. Control Optimization 23, 1985, 329–380
[37] Kushner, H. J., Dupuis, P.: Numerical Methods for Stochastic Control Problems in Continuous time, Springer-Verlag, New York, 1992
[38] Kwon, W. H., Pearson, A. E.: A modified quadratic cost problem and feedback stabilization of a linear system, IEEE Trans. Autom. Control 22, 1977, 838–842
[39] Laier, D., Markert, R.: Nonlinear oscillations of magnetically suspended rotors. Proc. 2nd European Nonlinear Oscillations Conference, Prague 1996, 239–242
[40] Lemmon, M.: Real-time optimal path planning using a distributed computing paradigm. In: Proceedings American Control Conference, Boston, MA, 1991
[41] Lin, L. J.: Self-improvement based on reinforcement learning, planning and teaching. In: L. A. Birnbaum and G. C. Collins, eds., Machine Learning: Proceedings Eight International Workshop, Morgan Kaufmann, San Mateo, CA, 1991, 323–327
[42] Lin, L. J.: Self-improving reactive agents: case studies of reinforcement learning frameworks. In: From Animals to Animats: Proceeding First International Conference on Simulation of Adaptive Behavior, Cambridge, MA, 1991, 297–305
[43] Lin, L. J.: Self-improving reactive agents based on reinforcement learning, planning and teaching. Math. Learn. 8, 1992, 293–321
[44] Mahadevan, S., Connell, J.: Automatic programming of behavior-based robots using reinforcement learning, Artif. Intell. 55, 1992, 311–365
[45] Mayne, D. Q., Michalska, H.: Receding horizon control of nonlinear systems. IEEE Trans. Autom. Control 35, 1990, 814–824
[46] Michie, D., Chambers, R. A.: BOXES: an experiment in adaptive control. In: E. Dale and D. Michie, eds., Machine Intelligence 2, Oliver and Boyd, Edinburgh, 1968, 137–152
[47] Moore, A. W.: Efficient memory-based learning for robot control, Ph.D. Thesis, University of Cambridge, Cambridge, England, 1990
[48] Moore, A. W.: Variable resolution dynamic programming: efficiently learning action maps in multivariate real-valued state-spaces. In: L. A. Birnbaum and G. C. Col1ins, eds., Machine Learning: Proceedings Eighth International Workshop, Morgan Kaufmann, San Mateo, CA, 1991, 333–337
[49] Moore, A. W., Atkeson, C. G.: Memory-based reinforcement learning: efficient computation with prioritized sweeping. In: S. J. Hanson, J. D. Cowan and C. L. Giles, eds., Advances in Neural Information Proceeding 5, Morgan Kaufmann, San Mateo, CA, 1993
[50] Peng, J., Williams, R. J.: Efficient learning and planning within the dyna framework. Adaptive Behavior 2, 1993, 437–454
[51] Půst, L.: Dynamická stabilita magnetického ložiska. Engineering Mechanics ’97, Svratka 1997, 57–62

[52] Puterman, M. L., Shin, M. C.: Modified policy iteration algorithms for discounted Markov decision problems. Manage. Sci. 24, 1978, 1127–1137
[53] Ross, S.: Introduction to Stochastic Dynamic Programming. Academic Press, New York, 1983
[54] Samuel, A. L.: Some studies in machine learning using the game of checkers. IBM J. Res. Develop, 1959, 210–229; reprinted in: E. A. Feigenbaum and J. Feldman, eds., Computers and Thought, McGraw-Hill, New York, 1963
[55] Samuel, A. L.: Some studies in machine learning using the game of checkers. II Recent progress, IBM J. Res. Develop, 1967, 611–617
[56] Schmidhuber, J.: Adaptive confidence and adaptive curiosity. Tech. Report FKI-149-91 Institut für Informatik, Technische Universität München, 800 München 2, Germany, 1991
[57] Singh, S. P., Yee, R. C.: An upper bound on the loss from approximate optimal value functions, technical note. Mach. Learn. 16, 1994, 227–233
[58] Sutton, R. S.: Temporal credit assignment in reinforcement learning. Ph.D. Thesis, University of Massachusetts, Amherst, MA, 1984
[59] Sutton, R. S.: Learning to predict by the method of temporal differences. Mach. Learn. 3, 1988, 9–44
[60] Sutton, R. S.: Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In: Proceedings Seventh International Conference on Machine Learning, Morgan Kaufmann, San Mateo, CA, 1990, 216–224
[61] Sutton, R. S.: Planning by incremental dynamic programming. In: L. A. Birnbaum and G. C. Co1lins, eds., Machine Learning: Proceedings Eights International Workshop, Morgan Kaufmann, San Mateo. CA, 1991, 353–357
[62] Sutton, R. S., Barto, A. G.: Toward a modem theory of adaptive networks: expectation and prediction, Psychol. rev. 88, 1981, 135–170
[63] Tan, M.: Learning a cost-sensitive internal representation for reinforcement learning. In: L. A. Birnbaum and G. C. Collins, eds., Machine Learning: Proceedings Eighth International Workshop, Morgan Kaufmann, San Mateo. CA, 1991, 358–362
[64] Tesauro, G. J.: Practical issues in temporal difference learning. Mach. Learn. 8, 1992, 257–77
[65] Thrun, S.: The role of exploration in learning control. In: D. White and D. Sofge, eds., Handbook of intelligent control: Neural, Fuzzy and Adaptive Approaches, Van Nostrand Reinhold, New York, 1992, 527–559
[66] Thrun, S. B., Möller, K.: Active exploration in dynamic environments. In: J. E. Moody, S. J. Hanson and R. P. Lippmann, eds., Advances in Neural Information Processing Systems 4, Morgan Kaufmann, San Mateo, CA, 1992
[67] Utgoff, P. E., Clouse, J. A.: Two kinds of training information for evaluation function learning. In: Proceedings AAAI-91, Anaheim, CA, 1991, 596–610
[68] Watkins, C. J. C. H.: Learning from delayed rewards. Ph.D. Thesis, Cambridge University, Cambridge, England, 1989
[69] Watkins, C. J. C. H., Dayan, P.: Q-learning. Mach. Learn. 8, 1992, 279–292
[70] Whitehead, S. D.: Complexity and cooperation in Q-Learning. In: L. A. Birnbaum and G. C. Collins, eds., Machine Learning: Proceedings Eighth International Workshop, Morgan Kaufmann, San Mateo, CA, 1991, 363–367
[71] Williams, R. J., Baird, L. C.: A mathematic analysis of actor-critic architectures for learning optimal controls through incremental dynamic programming. In: Proceedings Sixth Yale Workshop on Adaptive and Learning Systems. New Haven, CT, 1990, 96–101
[72] Yee, R. C.: Abstraction in control learning. Tech. Report 92-16, Department of Computer Science, University of Massachusetts, Amherst, MA, 1992

ABSTRACT

The habilitation thesis is focused on application of Q-learning on active magnetic bearing (AMB) control. The bearing represents unstable continuous nonlinear dynamic system. The problems in the process of learning to control such systems are overcome by dividing the conventionally processed learning into two phases – prelearning stage and tutorage stage.

Prelearning stage uses convergence properties of asynchronous real time values iteration (particularly Q-learning) based on trial. It uses trial which processes only single state transition. This way the contradiction between exploring and exploiting is overridden, even if the use of simulation model becomes necessary. Prelearning stage shows high convergence speed. Consequential tutorage stage uses conventional Q-learning procedure applied on Q-function created during the prelearning stage. Tutorage is expected to be processed on real mechatronic system and its use for controller adaptation on possible system parameters changes is not excluded.

Using the space of two-dimensional and three-dimensional discrete states (2-D state and 3-D state) created from AMB state variables the convergence speed of prelearning stage for various variants of state space discretization (Q-functin table grid) is examined first. The applicability of found policies is further validated through extensive simulations. During those simulations the robustness of the policy against system variables observation error and action delay is tested. Based on the tests the best grids of Q-function table for 2-D state and 3-D state were selected and obtained policies behaviour was compared with the behaviour of AMB controlled by continuous linear PID controller, which was taken from the literature.

The results show that control policies obtained on 2-D state and 3-D state after the tutorage stage are not significantly different and also that controller found by Q-learning reaches better control quality than PID controller.