Národní konference s mezinárodní účastí

INŽENÝRSKÁ MECHANIKA 2002

13. – 16. 5. 2002, Svratka, Česká republika

Q-LEARNING: CONTROL OF ASYNCHRONOUS ELECTRIC MOTOR

Tomáš Marada[1]

Abstrakt: *Tento článek obsahuje numerické experimenty, popisující nastavení parametrů Q-učení pro řízení asynchronního elektromotoru. Cílem je stanovit velikost a strukturu stavového prostoru, vhodně zvolit optimální akce pro řízení a dále určit odměnu pro posilovací funkci. V závěru je provedeno porovnání s klasickým PID regulátorem.*

Klíčová slova
Asynchronní elektromotor, Q-učení, řízení

1. Úvod

Jednou z možností jak zlepšit kvalitu řízení asynchronního elektromotoru, je využití metod s netradičním řízením. Například se může jednat o implementaci metod umělé inteligence, speciálně těch metod, které používají strojové učení v reálném čase.

Metody opakovaně posilovaného učení (Reinforcement Learning) pracují na principu naučení agentova chování v dynamickém prostředí metodou „pokus-omyl“. To se totiž jeví jako přirozený způsob učení nejen pro zvířata, ale i pro člověka. Agent stejně jako živý organismus vnímá prostředí, ve kterém se právě nachází, provádí v něm různé akce a na základě jejich výsledků dostává číselné ohodnocení (Reinforcement) od vhodné posilovací funkce. Tento způsob učení má tu výhodu, že není nutné dopředu znát funkci, kterou je potřeba se naučit, protože nalezení vhodné funkce je neočekávaně těžkým úkolem, mnohdy neřešitelným.

Q-učení je jednou z nejpopulárnějších a nejpoužívanějších metod patřících do oblasti opakovaně posilovaného učení. Naším cílem je optimalizovat vztah mezi jakýmsi řídícím členem a dynamickým prostředím. V našem případě je agentem QL-regulátor a prostředím je asynchronní elektromotor chápaný jako 3-D svět se spojitými souřadnicemi. Úhel natočení elektromotoru, úhlová rychlost a úhlové zrychlení představuje jednu z možných konstrukcí stavů prostředí. Úkolem QL-regulátoru je postupně se naučit, prostřednictvím simulačního modelu asynchronního elektromotoru, minimalizovat hodnotu integrálního kriteria kvality regulace.

[1] Tomáš Marada, Ing.,VUT v Brně, FSI ÚMT, Technická 2, 616 69 Brno, tomas.marada@seznam.cz

V algoritmu Q-učení se vyskytují následující pojmy:

Agent – v našem případě je agentem QL-regulátor. Úkolem agenta je, aby svým chováním (generováním akcí) dostal aktuální otáčky asynchronního elektromotoru do požadovaného intervalu otáček a v tomto intervalu je udržel. Agent tak představuje řídící člen.

Model – je v našem případě výše zmíněný asynchronní elektromotor.

Akce – pomocí sady akcí agent řídí model a tím mění jeho aktuální stav v prostředí.

Stavový prostor – popisuje stavy prostředí ve kterých se může model nacházet. Na vhodné volbě stavového prostoru závisí nejen kvalita naučení, ale i rychlost naučení. Pro stavový prostor je velmi důležitá jeho velikost, ale neméně významná je i jeho organizace. Z hlediska organizace stavového prostoru je významné, které veličiny jsou pro naučení podstatné. V naší úloze jsou podstatné tyto veličiny: Řídící frekvence a aktuální otáčky. Samotná volba toho, co je dobré znát, však nestačí. Jelikož se jedná o spojité veličiny, je třeba je vhodně rozdělit na určité intervaly. Čím více jich bude, tím přesněji bude systém fungovat, ale učení bude trvat déle. Stavový prostor lze rozdělit několika způsoby. Můžeme ho rozdělit na stejně velké intervaly, nebo na intervaly mající různou velikost. První jmenovaná možnost se nazývá symetrické lineární rozdělení stavového prostoru. Z hlediska rozlehlosti stavového prostoru není příliš vhodné a je proto vhodné použít druhé možnosti nazvané symetrické nelineární rozdělení stavového prostoru. Toto rozdělení vychází z toho, že není nutné mít stavový prostor rozdělen na velký počet částí v oblasti, které pro nás nejsou příliš důležité. V našem případě v blízkosti námi požadovaných otáček je stavový prostor rozdělen na malé úseky, a čím více se vzdaluje od těchto otáček, tím také velikost těchto úseků roste. Tímto se dá stavový prostor dostatečně minimalizovat.

2. Model učení

Algoritmus Q-učení (Q-Learning) popsal Watkins. Tento algoritmus je velmi jednoduché implementovat. Q-učení je pravděpodobně nejpopulárnější a nejefektivnější RL procedura opakovaně posilovaného učení bez modelu.

Odhad celkového výkonu agenta (Q-funkce)

$$\begin{aligned} &Q_\pi(s_i,u) = r(s_i,u) + \sum_{k=i+1}^{\infty} \gamma^{k-i} r(s_k, \pi(s_k)) \\ &\pi(s) = \arg\max_u Q(s,u) \end{aligned} \quad , \; 0 < \gamma < 1, \; \forall s_i, s \in S, \; \forall u \in U, \qquad (1)$$

kde

S		konečná množina stavů prostředí,
U		konečná množina akcí,
$\pi : S \to U$		strategie agenta,
$r : S \times U \to R$		ohodnocení okamžitého výkonu agenta
γ		srážkový faktor.

Odhad optimálního celkového výkonu agenta je odhad celkového výkonu agenta při jeho optimální strategii (optimální Q-funkce).

$$\begin{aligned} Q^*(s_i,u) &= \max_\pi Q_\pi(s_i,u) \\ \pi^*(s) &= \arg\max_u Q^*(s,u) \end{aligned} \quad , \ \forall s_i, s \in S, \ \forall u \in U \tag{2}$$

Vhodnost dané akce *u* ve stavu *s* je

$$\Delta e(s,u) = \begin{cases} (\gamma\lambda - 1)e(s,u) + 1, & jestliže \ (s = s_k) \wedge (u = u_k) \\ (\gamma\lambda - 1)e(s,u), & jinak \end{cases} \forall s, s_k \in S, \forall u, u_k \in U. \tag{3}$$

Rovnice pro Q-učení tedy je

$$\Delta Q(s,u) = \alpha\left[r(s_k,u_k) + \gamma \max_{u_{k+1}} Q(s_{k+1},u_{k+1}) - Q(s_k,u_k) \right] e(s,u), \ \forall s \in S, \forall u \in U. \tag{4}$$

Toto pravidlo zaručuje s pravděpodobností 1 konvergenci Q-hodnot k optimální hodnotě, jestliže je každá akce v každém stavu prostředí vykonána nekonečně mnohokrát během nekonečného počtu kroků a pokud hodnota učícího poměru α vhodně klesá. Pokud Q-hodnoty konvergují k optimální hodnotě, potom je vhodné, aby agent v každém stavu vykonal vždy akci s nejvyšší Q-hodnotou. Q-učení je necitlivé na nastavení, to znamená, že Q-hodnoty budou vždy konvergovat k optimální hodnotě, budou-li všechny dvojice stav-akce vyzkoušeny v dostatečném počtu. Bohužel tato konvergence může být velmi pomalá. Q-učení nepracuje příliš dobře při zobecněných problémech, nebo při stavovém prostoru, jenž je rozsáhlý.

3. Model asynchronního elektromotoru

Jako model byl použit model asynchronního elektromotoru. Řízení otáček je prováděno změnou kmitočtu napájecího proudu. Implementace modelu je provedena následujícími iteračními rovnicemi.

Ud = U·sin(2π·f·T)
Uq = -U·cos(2π·f·T)
MZ = 5
D[0] = Id
D[1] = Iq
Id = Id+(K0·T(Ud-Rs·Id+K1·ID+Lsr·WR·IQ+K2·Iq·Wr))
Iq = Iq+(K0·T(Uq-Rs·Iq+K1·IQ-Lsr·WR·ID-K2·Id·Wr))
ID = ID+(-K4(Id-D0)+T(-K3·ID-WR·IQ-K4·WR·Iq))
IQ = IQ+(-K4(Iq-D1)+T(-K3·IQ+WR·ID+K4·WR·Id))
MI = 1.5·Lsr(Iq·ID-IQ·Id)
WR = WR+T(MI-MZ)/J
T = T + krok

Kde konstanty K0 až K4 jsou vypočteny následujícími vztahy:

K[0] = Lr/(Ls·Lr-Lsr·Lsr)
K[1] = Lsr·Rr/Lr
K[2] = Lsr·Lsr/Lr
K[3] = Rr/Lr
K[4] = Lsr/Lr

Proměnné v těchto rovnicích značí a byly nastaveny následovně:

Rs	Odpor statoru	4.37 Ω
Rr	Odpor rotoru	2.95 Ω
Ls	Indukčnost statoru	0.471 H
Lr	Indukčnost rotoru	0.476 H
Lsr	Indukčnost mezi statorem a rotorem	0.459 H
f	Frekvence	50 Hz
U	Napětí	311.13 V
J	Moment setrvačnosti	0.0015 $Kg \cdot m^2$
MZ	Zátěžný moment	5 N·m
Krok	Časový krok	0.00001
Ud	Napětí v ose d	
Uq	Napětí v ose q	
Id	Proud v ose d	
Iq	Proud v ose q	
ID	Proud v ose D	
IQ	Proud v ose Q	
MI	Interní moment stroje	
WR	Aktuální otáčky	
T	Čas modelu	

4. Parametry experimentů

Pro všechny testy platí, není-li řečeno jinak, tato pravidla:

V každém experimentu bylo provedeno 5000 pokusů. Úspěšnost byla spočtena každých 250 pokusů jako percentuální poměr úspěšných pokusů k neúspěšným. Za úspěšný pokus byl považován pokus, ve kterém bylo dosaženo požadovaných otáček.

Byly použity tyto akce:

- „přidej frekvenci o 10Hz“
- „přidej frekvenci o 1Hz“
- „udržuj stejnou frekvenci“
- „uber frekvenci o 1Hz“
- „uber frekvenci o 10Hz“.

Počáteční otáčky byly nastaveny náhodně.

Agentova ohodnocovací funkce *r* byla definována následovně:
Pokud byla akce vhodná, to znamená, že se otáčky modelu přiblížily požadovaným otáčkám, nebo pokud těchto otáček model dosáhl, bylo ohodnocení *r* funkcí aktuálních otáček v intervalu (0,1). V případě, že akce neměla význam na otáčky modelu, byla hodnota *r* nastavena na hodnotu 0. Pokud akce nebyla vhodná, to znamená, že se otáčky modelu vzdálily od požadovaných otáček, nebo pokud model dosáhl nulové rychlosti, bylo ohodnocení *r* funkcí aktuálních otáček v intervalu (0,-1).

Stavový prostor byl nelineární a měl velikost 350. Hodnota 350 udává velikost stavového prostoru vypočtenou jako velikost gridu frekvence*otáčky*akce, což je 7*10*5 = 350.

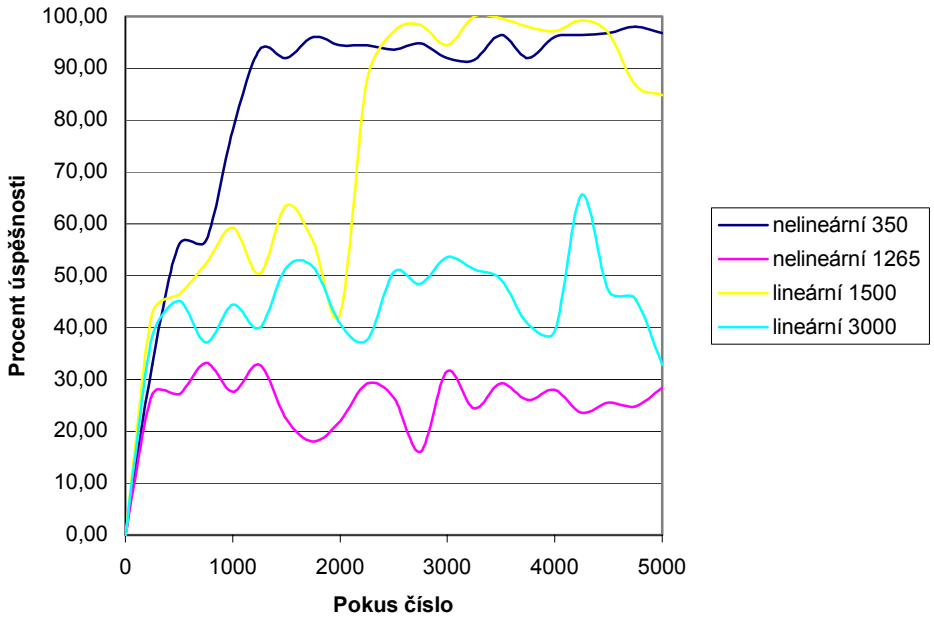
Koeficienty pro Q-učení byly nastaveny v následujícím rozsahu:

Učící poměr	$\alpha \in (0, 1)$
Faktor slevy	$\gamma \in (0.7, 1)$
Míra zapomínání	$\lambda \in (0.7, 1)$

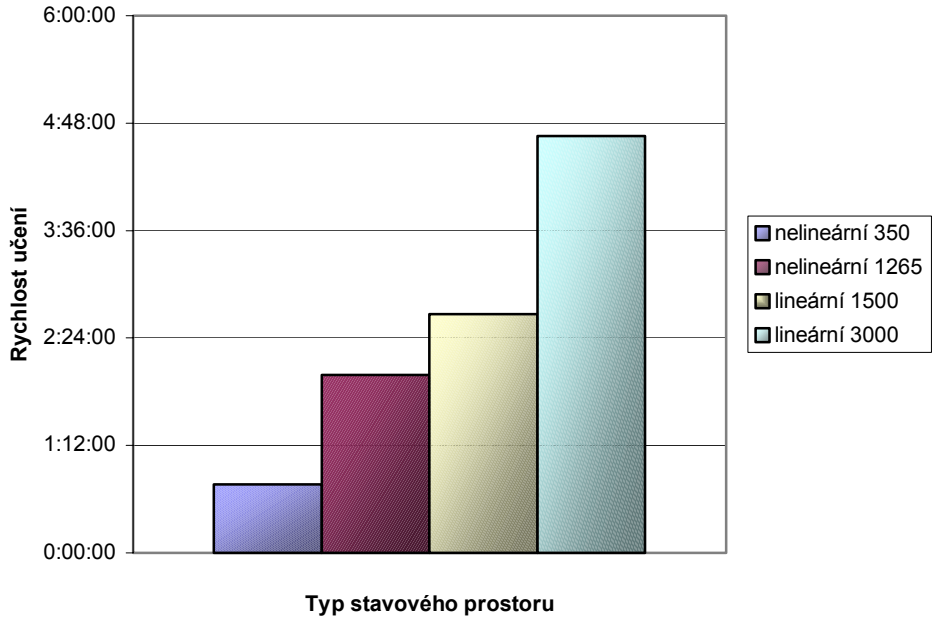
5. Provedené experimenty

5.1. Rozdělení stavového prostoru

Vliv rozdělení stavového prostoru na průběh kvality naučení je zobrazen na obrázku 1. Velikost gridu byla [7x10x5]=350, [11x23x5]=1265, [10x30x5]=1500 a [15x40x5]=3000. Z obrázku je patrné, že po 5000 pokusech se systém nejlépe naučil s nelineárním stavovým prostorem o velikosti 350. Dá se předpokládat, že systém s rozlehlejšími stavovými prostory by se naučil lépe, ale na úkor prodloužení doby výpočtu. Rychlost učení v závislosti na volbě stavového prostoru je na obrázku 2.



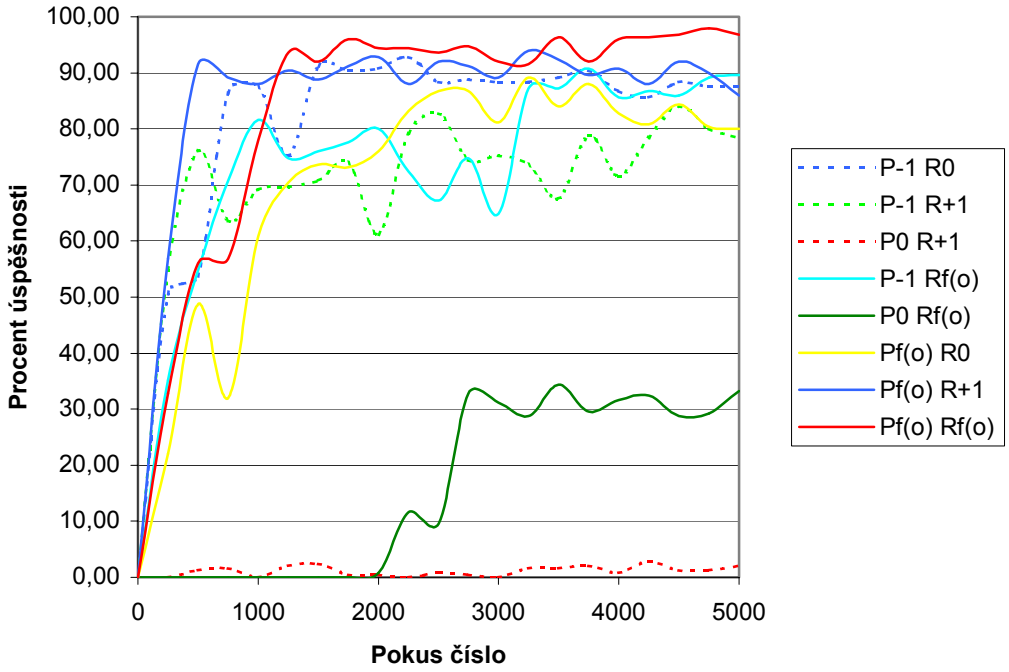
Obr. 1: Vliv velikosti a rozdělení stavového prostoru na průběh kvality naučení



Obr. 2: Rychlost učení v závislosti na volbě stavového prostoru

5.2. Volba odměny

Z obrázku 3 plyne, že ke správnému naučení systému je třeba systém trestat. Různé kombinace odměn a trestů, ať už poměrných či absolutních, přináší podobné výsledky. Ty se pak mohou lišit dle typu řešené úlohy. Problém poměrného trestu a odměny u řídící úlohy je shodný s problémem poměrného trestu v trestním právu. Každý konkrétní případ je jiný a i když si určíme nějaké měřítko, nikdy správně nepostihneme všechny aspekty. Proto je poměrná odměna i trest nejvýhodnější.

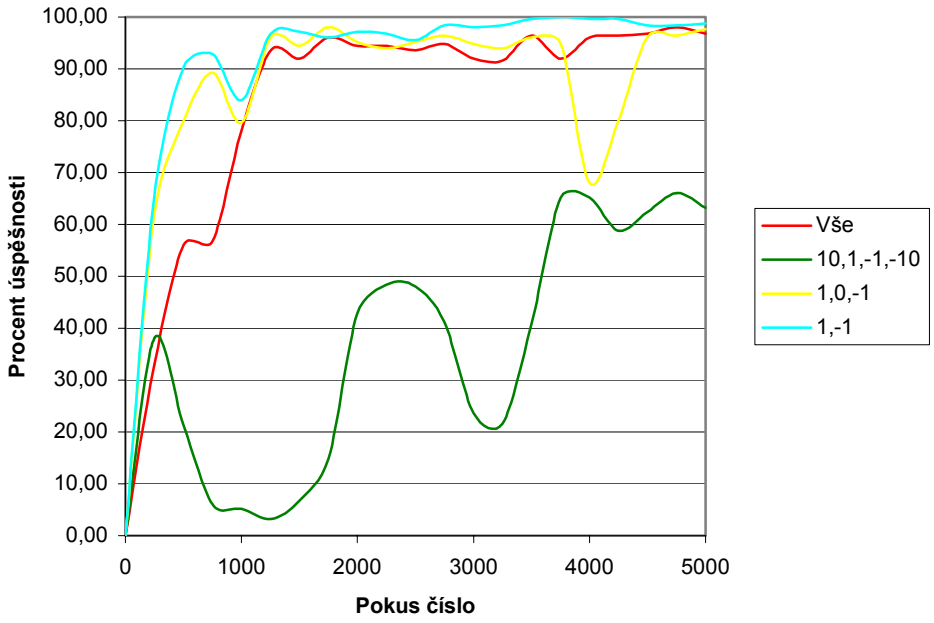


Obr. 3: Graf vlivu různé odměny na průběh kvality naučení

V grafu P udává trest a R odměnu. P-1 a R+1 tudíž znamená, že při špatné akci byl agent potrestán hodnotou $r = -1$ a při vhodné akci byl naopak odměněn hodnotou $r = 1$. Pf(o) udává, že míra trestu byla stanovena na základě aktuálních otáček modelu. Za špatnou akci považujeme tu akci, na základě které se otáčky modelu vzdálí od požadovaných otáček, případně pokud bude dosaženo nulových otáček. Vhodná akce je zase ta, na základě které se aktuální otáčky modelu přiblíží k námi požadovaným otáčkám, případně těchto požadovaných otáček dosáhne.

5.3. Volba akce

Z obrázku 4 je patrné, že nejlepšího naučení se docílilo při použití akcí „přidej frekvenci o 1Hz“ a „uber frekvenci o 1Hz“. V praktických testech se ale ukázalo, že volba těchto akcí není příliš vhodná. Byly proto rozšířeny o akce „přidej frekvenci o 10Hz“, „uber frekvenci o 10Hz“ a „udržuj stejnou frekvenci“, které se osvědčily více.

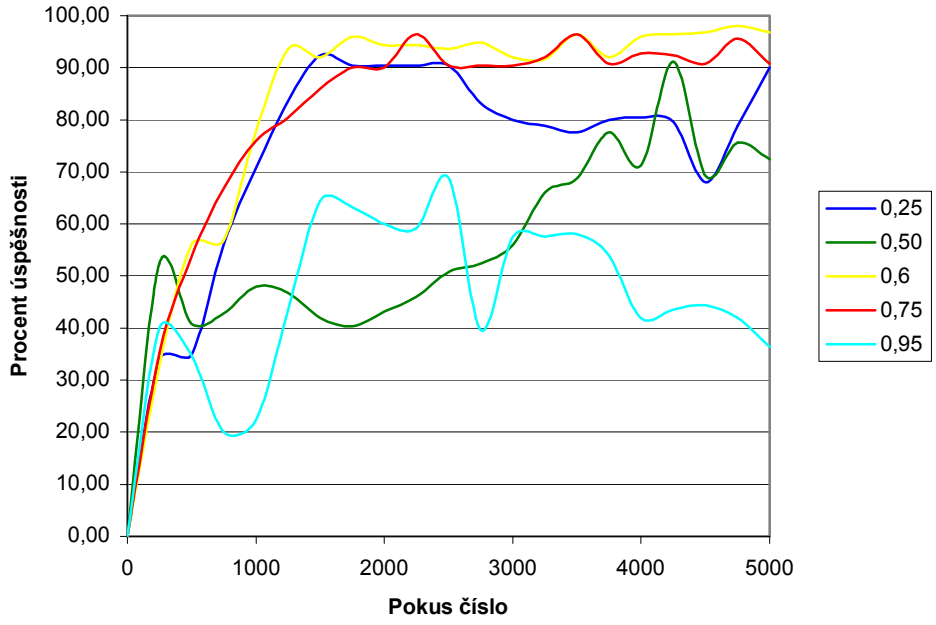


Obr. 4: Graf vlivu různých akcí na průběh kvality naučení

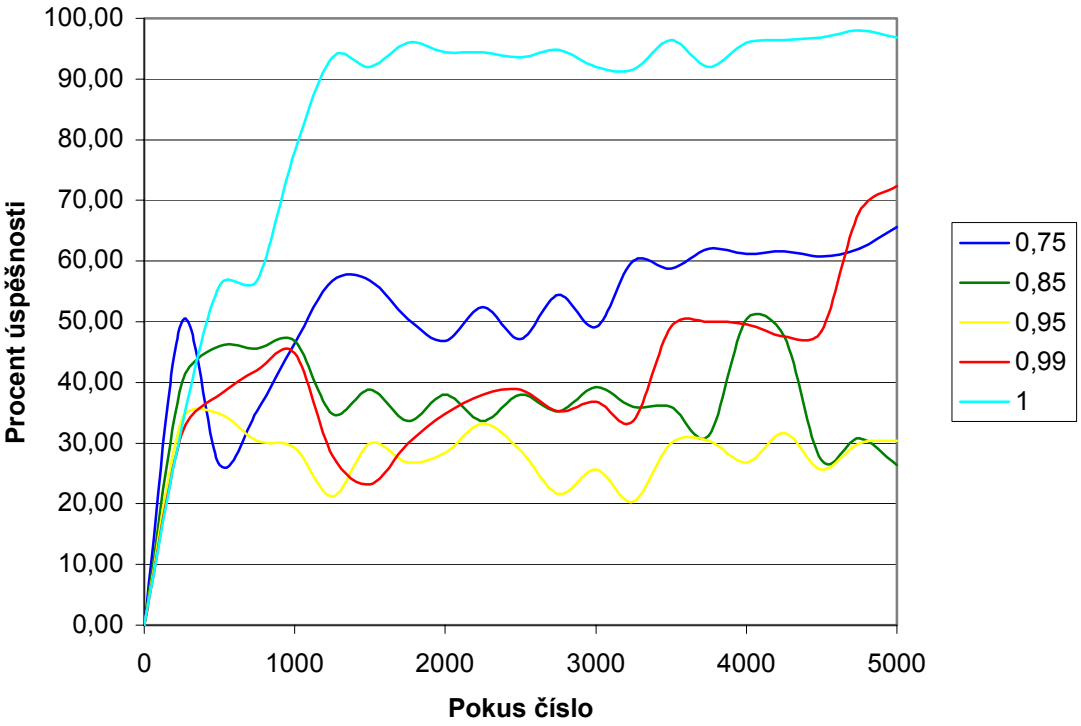
5.4. Volba koeficientů α, γ, λ

Z obrázku 5, 6, 7 je patrné, že nejlepší kvality naučení bylo dosaženo pro následující hodnoty koeficientů Q-učení.

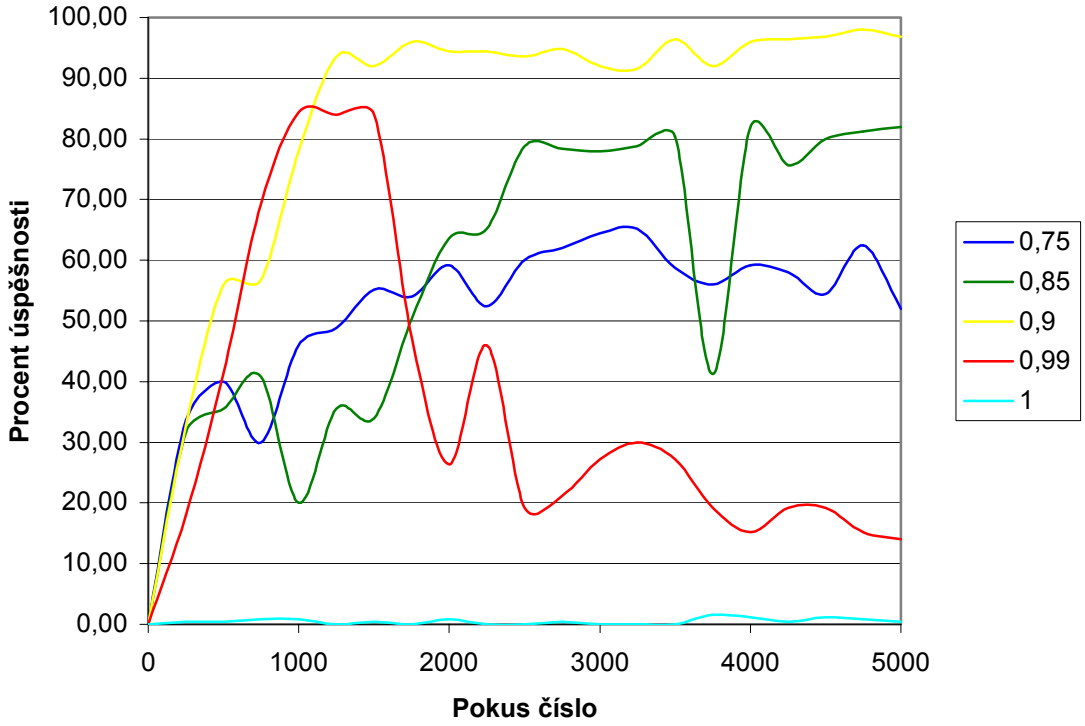
Učící poměr $\alpha = 0.6$
Srážkový faktor $\gamma = 1$
Míra zapomínání $\lambda = 0.9$



Obr. 5: Graf vlivu parametru ALPHA na průběh kvality naučení



Obr. 6: Graf vlivu parametru GAMMA na průběh kvality naučení

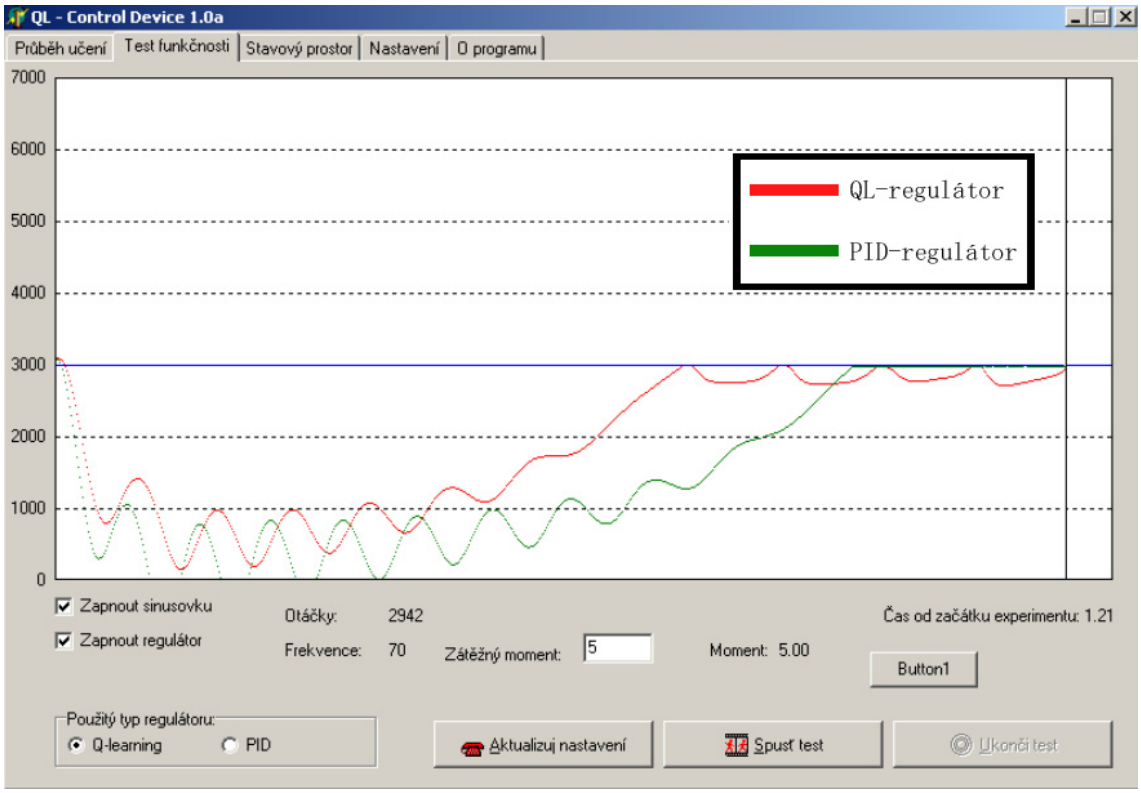


Obr. 7: Graf vlivu parametru LAMBDA na průběh kvality naučení

5.5. Porovnání QL-Regulátoru s PID-Regulátorem

Učení QL-regulátoru se skládalo z učení na dvě úlohy. První úlohou bylo naučit regulátor dosáhnout vhodným způsobem požadovaných otáček. Druhou úlohou bylo tyto požadované otáčky udržet. Systém byl naučen na dosažení požadovaných otáček. PID regulátor byl nastaven pomocí Ziegler Nicholsova pravidla po mírných úpravách na tyto hodnoty.

Zesílení	0.4
Derivační časová konstanta	2
Integrační časová konstanta	50



Obr. 8: Porovnání QL-Regulátoru s PID-Regulátorem

Na obrázku 8 je červenou barvou vyobrazen průběh QL-regulátoru a zelenou barvou průběh PID-regulátoru. Jestliže si rozebereme průběh experimentu zobrazeného na obrázku vidíme, že v první fázi učení dosáhl QL-regulátor lepšího výsledku. Můžeme si také povšimnout toho, že QL-regulátor se choval podobně jako PID-regulátor. Ve druhé fázi učení vidíme, že PID-regulátor dosáhl požadovaných otáček a tuto hodnotu dále udržel. Na rozdíl od něj QL-regulátor tuto hodnotu dosáhl, ale docházelo k periodickým oscilacím. Z toho vyplývá, že ve druhé fázi QL-regulátor nedosáhl stejné kvality jako jeho konkurent PID-regulátor. Tento výsledek byl zapříčiněn tím, že systém nebyl na úlohu udržení otáček naučen.

6. Závěr

Metoda využívá schopnost učení postupně zlepšovat chování asynchronního elektromotoru. Experimentálně bylo vyvozeno, že optimální hodnoty pro nastavení Q-učení jsou tyto:

- učící poměr $\alpha = 0.6$
- srážkový poměr $\gamma = 1$
- faktor zapomínání $\lambda = 0.9$

Nejvýhodnější je poměrná odměna a trest. Jako nejvhodnější akce byly zvoleny akce:

- Dekrementuj řídící frekvenci o 10Hz
- Dekrementuj řídící frekvenci o 1Hz

- Udržuj stejnou řídící frekvenci
- Inkrementuj řídící frekvenci o 1Hz
- Inkrementuj řídící frekvenci o 10Hz

Zavedení nelineárního rastru pro všechny veličiny výrazně zlepšuje úspěšnost učení a proto je výhodnější než lineární rastr. Při naučení systému na dosažení požadovaných otáček se systém chová lépe než PID regulátor.

Doporučená literatura

[1] Richard S. Sutton, Andrew G. Barto. Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA, 1998.

[2] Leslie P. Kaebling, Michael L. Littmann, Andrew W. Moore. Reinforcement learning: A survey. Journal of Artificial Intelligence Research, 4, 1996.

[3] Christopher J.C.H. Watkins and Peter Dayan. Q-learning. Machine Learning, 8(3):279-292, 1992.

[4] Doc.Ing. Jaroslav Šubrt CSc.: Elektrické stroje, Brno, 1999.

[5] Doc.Ing. Petr Vavřín CSc.: Teorie automatického řízení 1, SNTL.