Hochschule für Angewandte Wissenschaften Hamburg
*Hamburg University of Applied Sciences*

# Ausarbeitung Anwendungen 2 - SoSe 2009

Manuel Trittel

## Reinforcement Learning in der autonomen Fahrzeugnavigation

*Fakultät Technik und Informatik*
*Department Informatik*

*Faculty of Engineering and Computer Science*
*Department of Computer Science*

# Inhaltsverzeichnis

**Kurzzusammenfassung**

Diese Ausarbeitung setzt sich mit dem wissenschaftlichen und technischen Stand von RL Methoden auseinander, die für den konkreten Anwendungsfall einer Geschwindigkeitsregelung auf statischen Rundkursen von Relevanz sein könnten. Nach einer Betrachtung von Anwendungs- und Beschleunigungsmöglichkeiten der Verfahren, sowie Herausforderungen möglicher Problempunkte, wird mit Hilfe der gewonnenen Eindrücke das weitere Vorgehen am Anwendungsfall bestimmt.

# 1 Einführung in das Themengebiet

„Die Entwicklung intelligenter Systeme, die selbständig komplexe Aufgaben lösen, ist ein zentrales Forschungsgebiet in der Informatik. Im Vordergrund steht hierbei stets die Frage, wie das System lernen kann, sich korrekt zu verhalten, so dass die vorgegebenen Ziele verwirklicht werden“ (Wolter, 2008, S.1).

Hierbei stellt Reinforcement Learning (RL) eine maschinelle Lernmethode dar, welche im Gegensatz zu überwachten Methoden ohne Trainingsdatensätze und ohne einen Lehrer auskommt, aber weitreichendere und komplexere Möglichkeiten bietet als die unüberwachten Lernmethoden mit z. B. der automatischen Klassifikation oder der Komprimierung von Daten. Neben der Relevanz als theoretisches Werkzeug zur Untersuchung von Verhaltensweisen lernender Agenten wird RL bereits seit mehreren Jahrzehnten unterstützend in der Praxis eingesetzt, um autonome Systeme zu konstruieren, die in der Lage sind ihr Verhalten selbst anhand der gemachten Erfahrungen bei ihren Tätigkeiten zu optimieren (vgl. Kaelbling u. a., 1996, S.270).

Diese Ausarbeitung entspringt dem Kontext des FAUST-Projekts (Fahrerassistenz- und Autonome Systeme) der HAW Hamburg (FAUST, 2009) und beschäftigt sich mit der Anwendung von Reinforcement Learning in der Modellfahrzeugnavigation. In einem konkreten ersten Anwendungsfall wird versucht RL Konzepte auf eine Geschwindigkeitsregelung in festen Rundkursen anzuwenden.

Im weiteren Verlauf dieser Ausarbeitung werden geeigenete Methoden und Algorithmen mit verschiedenen Anwendungsfällen in verwandten Arbeiten betrachtet. Ferner werden typische Probleme, Einschränkungen und Herausforderungen erörtert, die bei der praktischen Verwendung dieser Verfahren eine Rolle spielen und hierzu mögliche Lösungsansätze vorgeschlagen. Abschließend wird die Relevanz der gemachten Erfahrungen für die Geschwindigkeitsregelung diskutiert und die Verwendung der vorgestellter Methoden für das weitere Vorgehen in Betracht gezogen.

# 2 Verwandte Arbeiten und relevante RL Methoden

Dieses Kapitel befasst sich mit zum Kontext passenden, klassischen und populären RL Vorgehensweisen, sowie Verfahren zur Funktionsapproximation. Entsprechend ihrer Eignung für die Geschwindigkeitsregelung fällt das Augenmerk an dieser Stelle auf die Dynamische Programmierung zu wie Temporal Differenz Methen.

Es wird folgende Nomenklatur verwendet:

- $S$: eine Menge diskreter Umweltzustände; $s$ ein Zustand aus $S$ ($s'$ Folgezustand)
- $A$: eine Menge diskreter Aktionen; $a$ eine Aktion aus $A$ ($a'$ Folgeaktion)
- $R(s, a)$: ein numerischer Wert als Belohnung/Bestrafung beim Ausführen von $a$ in $s$
- $r_t$: Wertigkeit sich im Zustand $s_t$ zu befinden
- $T(s, a, s')$: Wahrscheinlichkeit eines Zustandsübergangs $s \rightarrow s'$ bei Aktion $a$
- $\pi(s)$: eine Strategie bzw. Zustandstrajektorie ab Zustand $s$
- $\gamma$: Diskontierungsfaktor
- $V^{\pi}(s)$: diskontierte Summe aller $R(s, a)$ bei Ausführung von $\pi(s)$
- $Q^{\pi}(s, a)$: Belohnung eines Zustandsübergangs + $V^{\pi}$ des Folgezustands

## 2.1 Relevante Lernverfahren und -ansätze

Die folgenden Beschreibungen beziehen sich vereinfachend nur auf diskrete Zustands- und Aktionsräume. In Kapitel 3.1 wird näher auf kontinuierliche Fälle und damit auftretende Probleme eingegangen.

**Dynamische Programmierung (DP)**
Die Methoden der Dynamischen Programmierung gehen zurück in die fünfziger Jahre und wurden von Bellman (Bellman, 1957) entwickelt. Auf der Grundlage von Markov'schen Entscheidungsprozessen (MDP) (für Details zu MDPs sei auf (Alpaydin, 2008) verwiesen) und der *value function*

$$V^{\pi}(s_0) = \sum_{i=0}^{\infty} \gamma^i R(s_i, \pi(s_i))$$

kann die sogenannte *value iteration* durchlaufen werden, um eine optimale *value function* zu finden. Damit wiederum kann dann die *policy iteration* durchgeführt werden, um die beste Zustandstrajektorie zum angestrebten Zielzustand zu finden. Coulom beschreibt diese Schritte ausführlich in seiner Doktorarbeit, siehe (Coulom, 2002, S.35).

Die Methoden der DP verwenden keine echten Belohnungen, denn der Agent führt keine echten Aktionen aus. Es werden Lösungen auf bekannten Zustands- und Aktionsräumen ermittelt. Die DP hat heute eher eine theoretische Bedeutung und wird in der Praxis kaum verwendet. Sie ist einfach, auf ein vollständiges MDP Modell angewiesen und durch zahlreiche Iterationen mit sehr hohem Rechenaufwand verbunden. Es gibt jedoch erweiterte Ansätze wie das *Real-Time Dynamic Programming (RTDP)*, welches größere, praktische Bedeutung erlangt hat.

**Temporal Difference Learning (TD)**
Im Gegensatz zur DP wird bei den TD Methoden nicht die vollständige *value function V* (für alle Zustände) aktualisiert, sondern nur $V(s)$ des verlassenen Zustands nach einer Aktion. Der neue Wert ergibt sich anteilig aus dem vorherigen Wert und der erhaltenen Aktualisierung. Die Gewichtung erfolgt über die Lernrate $\eta$.

$$V_{k+1}(s_t) = (1 - \eta) \cdot V_k(s_t) + \eta(r_t + \gamma V_k(s_{t+1}))$$

Die resultierende Änderung $\Delta V(s)$ wird *temporal difference* genannt.

$$\Delta V(s_t) = \eta \cdot td = \eta(r_t + \gamma V(s_{t+1}) - V(s_t))$$

Hierzu analog wird auch die Q-Funktion $Q(s, a)$ aktualisiert

$$td = R(s, a) + \gamma Q^{\pi}(s', a') - Q^{\pi}(s, a)$$

wobei die Strategie $\pi$ als *estimation policy* bezeichnet wird. Die estimation policy muss beim Lernen der Q-Funktion nicht zwangsläufig der Strategie entsprechen, die vom Agenten praktisch verwendet wird. Genau an dieser Stelle unterscheiden sich die wichtigsten TD Verfahren: State Action Reward State Action learning (**SARSA**) und **Q-learning**. Während SARSA eine Strategie lernt und gleichzeitig benutzt (on-policy Verfahren), bedient sich Q-learning unabhängig von der aktuell ausgeführten Strategie immer der möglichst optimalen Aktionsauswahl als estimation policy (off-policy). Welches der beiden Verfahren praktisch die besseren Ergebnisse liefert ist strittig. Während Q-learning wahrscheinlich schneller konvergiert, hat SARSA Vorteile

bei der Umgebungserkundung. Günstigere Zustände können eher aufgefunden und ungünstige eher vermieden werden. Ausführliche Details zu den TD Lernalgorithmen beinhaltet Kapitel 6 in (Sutton und Barto, 1998).

Die bis zu diesem Punkt beschriebene Vorgehensweise entspricht dem einfachsten Fall des TD Lernens. Bei 1-Schritt weiten Aktualisierungen spricht man von $TD(0)$ Lernen. Eine Möglichkeit TD Lernverfahren mit Hilfe sogenannter *eligibility traces* zu beschleunigen wird in Kapitel 3.2 beschrieben. Mit ihnen kann der Agent „weitsichtiger“ gemacht werden und man spricht von $TD(\lambda)$ Lernen.

Die TD Lernmethoden werden auch in erweiterten oder abgewandelte Formen eingesetzt. So setzt z. B. (Shibata und Yoshinaka, 2008) ein spezielles Q-learning im Rahmen von kontextfreien Grammatiken ein.

## 2.2 Generalisierung und Funktionsapproximation

Wird die Anzahl der Zustands-Aktions-Paare zu groß oder steht nicht ausreichend Speicherplatz zur Verfügung um für jedes Tupel einen Wert $Q(s, a)$ aufzunehmen, müssen diese mit Hilfe von Generalisierungstechniken kompakter dargestellt werden. An dieser Stelle müssen keine neuen Erfindungen gemacht werden, sondern es kann auf bereits existierende Methoden zurück gegriffen werden. Die benötige Generalisierung wird auch *Funktionsapproximation* genannt, (vgl. Sutton und Barto, 1998, Kap.8). Funktionsapproximationen werde im RL häufig mit Gradienten-Abstiegs-Methoden realisiert. Hierbei werden die Zustände mit einem Parametervektor $\vec{\theta}_t$ abgebildet. Die einzelnen Parameter können durch Minimierung des *mean-squared-error (MSE)* optimiert werden. Dies geschieht durch die partielle Ableitung des Vektors und eine schrittweise Parameteranpassung in Richtung des resultierenden Gradienten, der gerade in die Richtung zeigt in die der Fehler am stärksten abnimmt.

Außerdem gibt es verschiedene lineare Methoden, wie Grob- und Teilkodierungen *(Coarse Coding and Tile Coding)*, Kubische Splines, Radiale Basisfunktionen, die Kanerva Kodierung oder die Darstellung der Approximationsfunktion durch lineare Funktionen des Feature-Vektors. Als nicht-lineare Methoden kommen Mustererkennung, neuronale Netze oder z. B. Regressionsmethoden in Frage.

Laut (Irodova und Sloan, 2005) gibt es nur wenige Arbeiten mit Funktionsapproximationen bei modellfreien RL Methoden, insbesondere nicht beim Q-learning. In ihrer Arbeit wurde das klassische *Blocks World* Problem in großen Dimensionen mit einer Approximation der Zustände durch Linearkombinationen von Features gelöst.

# 3 Herausforderungen und Problemstellungen

In diesem Kapitel werden existierende Problemstellungen, besondere Herausforderungen und zur Verfügung stehende Hilfsmittel betrachtet und verschiedene, bekannte Lösungsansätze vorgeschlagen.

## 3.1 Diskretisierung kontinuierlicher Zustands- und Aktionsräume

Die bisher beschriebenen Verfahrensweisen haben sich immer auf diskrete Zustands- und Aktionsräume gestützt. In der Praxis treten jedoch häufig kontinuierliche Zustände, wie Geschwindigkeiten und kontinuierliche Aktionsmöglichkeiten wie auszuübende Kräfte auf. Üblicherweise wird in solchen Fällen versucht eine Diskretisierung vorzunehmen, um mit möglichst wenigen Anpassung auf die Theorie der diskreten Verfahren zurück greifen zu können.

Coulom beschreibt im Rahmen der DP eine Diskretisierungsmethode mit finiten Differenzen unter Anderem am Beispiel eines 2-gliedrigen Pendels. Winkel und Winkelgeschwindigkeit des Gelenks werden in je 1600 diskrete Zustände gerastert. Problem bei diesem einfachen Gitteransatz ist ein exponentielles Anwachsen des Zustandsraums mit der Größe seiner Dimension. Als alternativen Ansatz beschreibt Coulom die Wahl eines neuronalen Netzes, (vgl. Coulom, 2002, S.53).

## 3.2 Beschleunigung der Lernverfahren

Ein großes Problem der RL Verfahren ist die Performance. Zum Einen tritt die Frage auf, wie lange es dauert, bis die Strategie zu einer bestimmten Qualität konvergiert, zum Anderen kann der Rechenaufwand der gewählten Algorithmen schnell zu einem K.O.-Kriterium werden. Dies betrifft nicht nur die zuvor genannten Diskretisierungsverfahren, sondern auch die Algorithmen selbst - je nachdem, wie zeitkritisch das umgebende System ausgelegt sein muss. An dieser Stelle sollen einige allgemeine und speziellere Beschleunigungsverfahren für die zuvor behandelten RL Methoden beschrieben werden.

- Vorwissen über die Systemumgebung
- Hybride Ansätze
- Diskretisierungsverfahren
- Eligibility traces für TD Lernverfahren
- Balance von Exploration und Exploitation

**Vorwissen über die Systemumgebung**
In den meisten Anwendungsfällen lassen sich Lernverfahren durch Einbeziehung von Vorwissen über die Systemumgebung beschleunigen. Dies ist in vielen Fällen sogar notwendig um brauchbare Ergebnisse zu erzielen. Nachteil hierbei ist, dass der Agent durch ein systemabhängiges, zuvor einprogrammiertes menschliches Wissen nicht mehr vollständig autonom agiert. Unterschiedlichste Beispiele hierfür werden in (Kaelbling u. a., 1996) aufgezählt:

- Erwartungen über z. B. Linearität oder lokal stetige Abschnitte einer Strategie
- Passende, manuelle Diskretisierungen des Zustandsraums
- Vorkenntnisse zur effizienteren Erkundung ähnlichartiger Zustandsräume

**Hybride Ansätze**
Auch Mischformen unterschiedlicher Lernverfahren sind denkbar. So hat (Thrasher u. a., 1997) bei seinen Experimenten Hüft- und Beinprothesen zu trainieren zuerst mit kurze Phasen überwachten Lernens Vorwissen gesammelt, um damit die Konvergenz des RL zu beschleunigen. Weiterhin wurde versucht mit dem antrainierte Wissen von einem 55Kg schweren Körper auf einen 75Kg schweren Körper umzulernen. Die benötigten Zyklen bis zur Konvergenz haben sich hierbei deutlich verringert. Möchte man solche hybriden Ansätze auf andere Problemstellungen anwenden stellt sich die Frage, ob der zu betreibende Aufwand im Verhältnis zu den Ergebnissen steht.

**Diskretisierungsverfahren**
Sind Diskretisierungen wie in Kapitel 3.1 erforderlich können insbesondere bei mehrdimensionalen Zuständen sehr schnell sehr große Zustandsräume entstehen, die viel Rechenaufwand beim Lernen der Q- und V-Funktion nach sich ziehen. Der einfache Gitteransatz aus Kapitel 3.1 lässt sich z. B. durch eine intelligentere Diskretisierung verbessern. Ist das Gitter in unwichtigen Bereichen grob und nur in wichtigen Bereichen fein strukturiert kann die Anzahl der Zustände deutlich verkleinert werden. Weiterhin können auch Mehrgitter-, Dreicksgitter- oder adaptive Verfahren die Diskretisierung optimieren, siehe (Pareigis, 1997b) und (Pareigis, 1997a).

**Eligibility traces für TD Lernverfahren**
Die in Kapitel 2.1 beschriebenen TD Lernverfahren werden häufig mit *eligibility traces (e-traces)* beschleunigt. Diese kann man sich wie eine im langsam verschwindende Spur vorstellen, die der Agent nach sich zieht. So kann nun eine gesamte Trajektorie ausfindig gemacht werden, die zum aktuellen Zustand geführt hat. Für alle betroffenen Zustände (statt wie bisher nur dem letzten) können nun je nach Stärke der Spur V- und Q-Funktion aktualisiert werden. Der Agent wird weitsichtiger. Der Dämpfungsfaktor $\lambda$ bestimmt wie schnell die Spur wieder verschwindet.

$$e_{t+1}(s) = \begin{cases} \lambda \cdot \gamma \cdot e_t(s), & \text{falls } s \neq s_t \\ 1, & \text{sonst} \end{cases}$$

Diese Formel entspricht den ersetzenden e-traces. Alte, evtl. bereits vorhandene Spuren auf den Zuständen werden überschrieben. Durch den gewonnenen Informationsgehalt über die Historie werden bei jedem Schritt des Agenten die vorher besuchten Zustände gezielt aktualisiert und so eine schnellere Konvergenz angestrebt. Werden e-traces berücksichtigt spricht man von TD($\lambda$) Verfahren.

## 3.3 Exploration vs Exploitation

Schlussendlich gehört auch das Problem zwischen Exploration und Exploitation in die Thematik der Beschleunigung von Lernverfahren. Grundlegend stellt sich die Frage wie wichtig überhaupt eine optimale Erkundung des Zustandsraums ist und ob Lernabschnitte klar von denen getrennt werden können, in denen das gesammelte Wissen optimal genutzt wird. Wie bereits zuvor angedeutet kann auch (menschlich vorprogrammiertes) Vorwissen über den Zustandsraum genutzt werden um ein effizientes Auskundschaften zu vollziehen. Es kann jedoch auch zwischen mehreren allgemeinen Verfahren entschieden werden.

Einer der einfachsten und bekanntesten Ansätze ist die $\epsilon$-*greedy* Suche. Hierbei wird mit einer Wahrscheinlichkeit von $(1-\epsilon)$ wobei $(\epsilon \in [0,1))$ die derzeit optimale Aktion gewählt. Andernfalls wird die Aktion zufällig bestimmt. Nun wählt man beispielsweise zu Anfang ein hohes $\epsilon$, das kontinuierlich verringert wird. Auf diese Weise nutzt der Agent mit der Zeit zunehmend das gesammelte Wissen aus, um optimale Aktionen zu wählen.

Alternative Ansätze die ebenfalls auf zufälliger Aktionswahl basieren sind die *Boltzmann exploration (z. B. Softmax Methode)* (Sutton und Barto, 1998) und das *Simulated Annealing (SA)* Verfahren (Guo u. a., 2004). Bei diesen Verfahren gibt es einen positiven Parameter $\tau$ mit dessen Hilfe und einer Wahrscheinlichkeit proportional zu $e^{\frac{Q(s,a)}{\tau}}$ die beste Aktionen gewählt werden. Wie auch mit dem $\epsilon$ kann über das $\tau$ ein Verhältnis von Exploration und Exploitation eingestellt werden. Für alle genannten Verfahren ist es sehr schwierig gute Werte für die Parameter zu finden, um ein ausbalanciertes System zu erhalten. Zusätzlich ist es problematisch die Parameter dynamisch zu ändern, sollte eine Neuerkundung im Lebenszyklus des Agenten erforderlich sein.

Einen neuen Ansatz verfolgt (Chen und Dong, 2008) bei einer neuen RL Methode - dem *Superposition-inspired reinforcement learning (SIRL)*. Ausgegangen wird hierbei von Quanten Charakteristiken, wobei besonders großer Wert Problematik auf die Exploration gelegt wurde.

Erste Experimente liefern vielversprechende Ergebnisse und weitere theoretische Nachforschungen stehen aus.

# 4 Relevanz für eigene Arbeitsziele und Methoden

Die Zustände im konkreten Anwendungsfall setzen sich aus einer odometrischen Streckenposition in cm und der seitlichen Beschleunigungskraft auf das Fahrzeug zusammen. Eine Diskretisierung wird für beide Parameter nicht erforderlich sein. Dennoch kann über eine Funktionsapproximation für die gelernten Aktionen nachgedacht werden, um auch mit wenigen angelernten Stützstellen ohne einen vollständig erkundeten Zustandsraum brauchbare Aktionen für unbekannte Zustände ermitteln zu können. Zu erwarten ist eine zumindest abschnittweise stetige Funktion (Kurven/Geraden), so dass sich unter Umständen Kubische Splines oder Radiale Basisfunktionen anbieten.

Die Beschleunigung eines Fahrzeugs ist ein träger, stetiger Vorgang und es wird davon ausgegangen, dass sich die Geschwindigkeit nicht von einem Zustand zum nächsten unstetig verändern kann (z. B. durch einen Aufprall). Erfährt das Fahrzeug in einem Zustand eine deutlich überhöhte seitliche Beschleunigungkraft, fährt es offenbar zu schnell für den aktuellen Streckenverlauf. Abhilfe dagegen schafft aber nicht allein die Wahl einer geringen Geschwindigkeit im vorherigen Zustand, sondern die ganze Trajektorie muss dementsprechend aktualisiert werden. Zum Lernen der maximalen Geschwindigkeiten scheinen also TD($\lambda$) Lernverfahren mit e-traces besonders gut geeignet zu sein.

Im Hinblick auf Exploration vs Exploitation ist zu sagen, dass es einen relevanten Verlauf durch den Zustandsraum geben wird. Diese Information kann verwendet werden, um den Lernvorgang zu beschleunigen. Da ein statischer Kurs abgefahren wird kann die Erforschung des Zustandsraums allerdings klar von der Ausnutzung der gesammelten Informationen getrennt werden. Die einzige harte Zeitanforderung ist, dass der Algorithmus beim Lernen keine elementaren Steuerungstasks des Fahrzeugs behindern darf. Der Lernprozess sollte lediglich mit möglichst wenigen Zyklen konvergieren.

Beim Abfahren des Kurses werden sich (bedingt durch ungenaue Odometriedaten) Abweichungen zur tatsächlichen Streckenposition ergeben und aufsummieren, siehe (Rull, 2009). Regelmäßige Orientierungspunkte wie z. B. eine Startlinie, Scale Invariant Feature Transform (SIFT) Features (siehe (Wagner, 2009)) oder eine kartenbasierte Positionsbestimmung (siehe (Rull, 2008)) können diesem Problem entgegenwirken, indem mit ihnen der Fehler regelmäßig ausgeglichen wird.

Für den konkreten Anwendungsfall stehen einige Hilfsmittel zur Verfügung. Besonders erwähnenswert ist *The Reinforcement Learning Toolbox*, die Gerhard Neumann in seiner Diplomarbeit entwickelt hat, siehe (Neumann, 2005). Neben zahlreichen, ausführlichen, theoretischen

Erklärungen zu Zustandsrepräsentationen, Algorithmen und Funktionsapproximationen stehen diese mit umfangreichen Implementationshinweise (C++) und Beispielexperimenten zur Verfügung. Sie wird kontinuierlich weiter entwickelt und aktualisiert.

# 5 Zusammenfassung

Auf dem Gebiet des Reinforcement Learning gibt es eine breit gefächerte Anzahl verschiedenster Anwendungsfälle. Vom Spielen von Backgammon über die Analyse kontext-freier Grammatiken, intelligenter Prothesen bis hin zu Versuchen Verhaltensweisen und Strukturen des menschlichen Gehirns nachzubilden. Jede Anwendung beinhaltet ihre eigenen Finessen. Die verbreitedsten RL Methoden bilden einen Rahmen zur Lösung von solch unterschiedlichen Aufgabenstellungen. Ein richtiges Vorgehen kann nicht verallgemeinert werden, sondern es muss individuell auf die Gegebenheiten der Umgebung reagiert werden.

In dieser Ausarbeitung wurden neben einigen essentiellen Grundlagen wichtige Teilgebiete des RL angerissen und grundlegende Fragen zu ihnen geklärt. Die Handhabung von kontinuierlichen und diskreten Zustands- und Aktionsräume und unterschiedliche Wege der Funktionsapproximation wurden verdeutlicht und aufgezählt. Verschiedene allgemeine und spezielle Möglichkeiten Lernverfahren (insbesondere DP und TD) zu beschleunigen wurden erklärt und es wurde daraufhin auf das Explorations-Problem eingegangen.

Mit den recherchierten Fakten fand schließlich eine Einordnung des Anwendungsfalls der Geschwindigkeitsregelung statt und relevante Alternativen zur Beantwortung der unterschiedlichen Fragestellungen konnten ermittelt werden.

# Literatur

[Alpaydin 2008] ALPAYDIN, Ethem: *Maschinelles Lernen*. München : Oldenbourg, 2008. – ISBN 978-3-486-58114-0

[Bellman 1957] BELLMAN, Richard E.: *Dynamic Programming*. New York: Courier Dover Publications, 1957. – ISBN 978-0486428093

[Chen und Dong 2008] CHEN, Chun-Lin ; DONG, Dao-Yi: Superposition-Inspired Reinforcement Learning and Quantum Reinforcement Learning. In: *Reinforcement Learning: Theory and Applications*. Wien, Österreich : I-Tech Education and Publishing, 2008, S. 59–84. – ISBN 978-3-902613-14-1

[Coulom 2002] COULOM, M. R.: *Apprentissage par renforcement utilisant des réseaux de neurones, avec des applications au contrôle moteur*, Nationales Institut für Polytechnik Grenoble, Doktorarbeit, 2002

[FAUST 2009] HAMBURG, HAW: *FAUST Fahrerassistenz- und Autonome Systeme*. 2009. – URL http://www.informatik.haw-hamburg.de/faust.html. – Abruf: 2009-06-21

[Guo u. a. 2004] GUO, M.Z. ; LIU, Y. ; MALEC, J.: A new Q-learning algorithm based on the metropolis criterion. In: *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, Volume 34*, 2004, S. 2140–2143

[Irodova und Sloan 2005] IRODOVA, Marina ; SLOAN, Robert H.: Reinforcement Learning and Function Approximation. In: *Proceedings of the Eighteenth International Florida Artificial Intelligence Research Society Conference*. Clearwater Beach, Florida, USA : AAAI Press, 2005, S. 455–460. – ISBN 1-57735-234-3

[Kaelbling u. a. 1996] KAELBLING, Leslie P. ; LITTMAN, Michael L. ; MOORE, Andrew W.: Reinforcement Learning: A Survey. 4 (1996), S. 237–285

[Neumann 2005] NEUMANN, Gerhard: *The Reinforcement Learning Toolbox, Reinforcement Learning for Optimal Control Tasks*, University of Technology Graz, Diplomarbeit, 2005

[Pareigis 1997a] PAREIGIS, Stephan: Adaptive choice of grid and time in reinforcement learning. In: *Proceedings of the International Conference on Neural Information Processing Systems*, 1997

[Pareigis 1997b] PAREIGIS, Stephan: Multi-grid methods for reinforcement learning in controlled diffusion processes. In: *Advances in Neural Information Processing Systems, Volume 9*. Cambridge, UK : The MIT Press, 1997, S. –

[Rull 2008] RULL, Andrej: *Sensorbasierte Umgebungskartierung mit lokaler Positionskorrektur für autonome Fahrzeuge*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2008

[Rull 2009] RULL, Andrej: *Fahrspur- und Odometriebasierte Selbstlokalisierung und Kartierung (SLAM) - Problemstellung*, Hochschule für Angewandte Wissenschaften Hamburg, Ausarbeitung, 2009. – URL http://users.informatik.haw-hamburg.de/~rull_a/download/Ausarbeitung_AW1_Andrej_Rull.pdf

[Shibata und Yoshinaka 2008] SHIBATA, Takeshi ; YOSHINAKA, Ryo: An Extension of Finite-state Markov Decision Process and an Application of Grammatical Inference. In: *Reinforcement Learning: Theory and Applications*. Wien, Österreich : I-Tech Education and Publishing, 2008, S. 85–104. – ISBN 978-3-902613-14-1

[Sutton und Barto 1998] SUTTON, Richard S. ; BARTO, Andrew G.: *Reinforcement Learning - An Introduction*. Cambridge : MIT Press, 1998. – ISBN 978-0262193986

[Thrasher u. a. 1997] THRASHER, Adam ; ANDREWS, Brian ; WANG, Feng: Control of FES using Reinforcement Learning: Accelerating the learning rate. In: *Proceedings - 19th International Conference - IEEE/EMBS*. Chicago, IL., USA : IEEE Computer Society, 1997, S. 1774–1776. – ISBN 0-7803-4262-3

[Wagner 2009] WAGNER, Benjamin: *3D-Objekterkennung im Kontext eines Assistenzroboters*, Hochschule für Angewandte Wissenschaften Hamburg, Ausarbeitung, 2009. – URL http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master08-09-aw1/wagner/bericht.pdf

[Wolter 2008] WOLTER, Anne: *Reinforcement Learning in der Roboter-Navigation*. Saarbrücken : Verlag Dr. Müller, 2008. – ISBN 978-3-639-04702-8