



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung Seminar

Ivo Nikolov

Maschinelles Lernen zur Steuerungsoptimierung eines
autonomen Fahrzeugs

Inhaltsverzeichnis

1	Einleitung	2
1.1	Motivation	2
1.2	Problemstellung	2
1.3	Zielsetzung	2
2	Lernverfahren zur autonomen Fahrzeugquerführung	3
2.1	Q-Learning und Neuronale Netze	3
2.2	Neural Fitted Q Iteration	4
2.3	Der RL-Lenkungsregler	5
2.4	Ergebnisse	5
3	Entwurf eines Lenkungsreglers	6
3.1	Fahrzeug-Zustandsbeschreibung	6
3.2	Explorationsstrategie	6
3.3	Funktionsapproximator	7
4	Testumgebung	7
4.1	Onyx Fahrzeug	7
4.2	TORCS Simulator	8
5	Zusammenfassung	9
6	Aktueller Stand	9

1 Einleitung

1.1 Motivation

Im Forschungsprojekt FAUST aus dem Department für Informatik der Hochschule für Angewandte Wissenschaften Hamburg werden Technologien für Fahrerassistenz- und Autonome Systeme entwickelt und entworfen [FAUST 2010]. Autonomes Fahren wurde auf Autobahnen und Landstraßen seit Anfang der neunziger Jahre in verschiedenen Projekten erforscht. Am Markt verfügbare Fahrerassistenzsysteme greifen bereits heute aktiv in die Fahrzeuglängs- und Querverführung ein. Für die Zukunft ist der Automatisierung in der Fahrzeugführung von der reinen Assistenz hin zu automatischen Fahrfunktionen denkbar [Weiser u. a. 2009].

1.2 Problemstellung

Der Hochschulwettbewerb Carolo-Cup [Carolo-Cup], einer der Schwerpunkte innerhalb des Projektes FAUST bietet Studententeams die Möglichkeit, sich mit der Entwicklung und Umsetzung von autonomen Modellfahrzeugen auseinander zu setzen. Beim Wettbewerb müssen bestimmte Fahraufgaben autonom, möglichst schnell und fehlerfrei bewältigt werden und die erarbeiteten Konzepte in Präsentationen erläutert werden. Laut [Carolo-Cup 2010] sollen die Fahrzeuge bei der dynamischen Disziplin Rundstrecke autonom drei Minuten lang auf einem unbekanntem Rundkurs so weit wie möglich fahren. Bei der Straße handelt es sich um die Nachbildung einer Landstraße, bestehend aus langen Geraden, schnellen Kurven, engen Serpentin und Kreuzungen. Die Straße ist konstant 820 mm breit und der minimale Kurvenradius beträgt 1000 mm. Die zwei Fahrspuren werden durch eine gestrichelte Mittellinie geteilt.

1.3 Zielsetzung

Folgende Zielvereinbarungen werden für das Konzept der Lenkungsregelung getroffen:

- Der Lenkungsregler soll zusammen mit dem Fahrspurerkennungssystem Polaris [Jenning 2009] eine autonome Fahrbahnverfolgung realisieren.
- Eine möglichst hohe Geschwindigkeit ist bei der Fahrbahnverfolgung anzustreben.
- Der Lenkungsregler soll den kinematischen und dynamischen Fahrzeugszustand betrachten.
- Der Lenkungsregler soll mittels maschineller Lernverfahren erstellt werden.

2 Lernverfahren zur autonomen Fahrzeugquerführung

In [Montemerlo u. a. 2007] wird vorgestellt wie ein Fahrzeug innerhalb von 20 Minuten lenken "lernt". Die autonome Fahrzeugquerführung erfolgt anhand eines Lenkungsreglers, der die Abweichung zur Sollspur minimiert. Das Verfahren basiert auf der Neural Fitted Q Iteration (NFQ) [Riedmiller 2005] und ist komplett datengetrieben d.h. keine Simulation oder Fahrzeugmodell sind erforderlich.

2.1 Q-Learning und Neuronale Netze

Q-Learning ist eine modellfreie Methode der Reinforcement Learning, die iterativ Zustands-Aktionspaare bewertet, um eine optimale Strategie zu erreichen. Nach jeder Systembeobachtung wird der entsprechende Q-Wert wie folgt aktualisiert:

$$Q_{k+1}(s, a) := (1 - \alpha)Q(s, a) + \alpha(c(s, a) + \gamma \min_b Q_k(s', b)) \quad (1)$$

wobei s den aktuellen Zustand, a die ausgeführte Aktion und s' den Folgezustand repräsentieren. α stellt dabei den Lernfaktor dar und $0 \leq \gamma \leq 1$ die Diskontrate [Sutton und Barto 1998].

Wenn Zustände und Aktionen diskret sind, werden die Q-Werte tabellarisch verwaltet. Bei kontinuierlichen Zustands- und Aktionsräumen kann eine Diskretisierung angewandt werden. Diskretisierung ist aber für hochdimensionale Probleme nicht geeignet [Ng 2008]. Eine andere Alternative ist das Ganze als Regressionsproblem zu betrachten. Hier geht es um eine Aufgabenstellung des überwachten Lernens, wobei ein Regressor $Q(s, a | \theta)$ definiert wird, der s und a als Eingabe nutzt und durch einen Vektor von Parametern θ parameterisiert wird, um die Q-Werte zu approximieren [Alpaydin 2004]. Hierzu können künstliche neuronale Netze eingesetzt werden. Das Problem bei der on-line Implementierung dieses Ansatzes (beim Training werden die Parameter nach jeder Aktion aktualisiert) ist, dass jede einzige Aktualisierung der Parameter die gesamte Approximationsleistung beeinträchtigen kann. Aus diesem Grund kann es sehr lange dauern bis die Q-Funktion erfolgreich approximiert wird [Riedmiller 2005].

Es wurde bewiesen, dass unter gewissen Voraussetzungen Q-Learning zum Optimum konvergiert. Obwohl dieser Beweis für Konvergenz bei kontinuierlichen Zustände und Aktionen nicht mehr gilt, es existieren viele Anwendungen, die neuronale Q-Funktionen erfolgreich nutzen.

2.2 Neural Fitted Q Iteration

Im Gegensatz zu der on-line Implementierung, approximiert der NFQ-Algorithmus die Q-Funktion off-line, d.h. um einen Trainingsdatensatz zu bilden werden zuerst Trainingsdaten gesammelt und dann insgesamt erlernt. Weiterer Vorteil generell für off-line Methoden ist, dass auch fortgeschrittene Techniken des überwachten Lernens anwendbar sind.

Die Trainingsdaten werden in Tripeln der Form (s, u, s') gespeichert. Hier repräsentiert s den ursprünglichen Zustand, u die gewählte Aktion und s' der Folgezustand. Der Trainingsdatensatz wird mit D bezeichnet.

NFQ ist eine Instanz der Fitted Q Iteration Familie von Algorithmen [Ernst u. a. 2005], wobei der Regression-Algorithmus durch ein neuronales Netz realisiert wird. Der Algorithmus ist in Abbildung 1 dargestellt. Er besteht aus zwei großen Schritten: Die Generation des Trainingssets P und das Trainieren des neuronalen Netzes mit s (Zustand) und u (Aktion) als Eingabe und die Kosten dieser Transition summiert mit den minimalen geschätzten Kosten für den Folgezustand als Ausgabe.

```

NFQ_main() {
  input: a set of transition samples  $D$ ; output: Q-value function  $Q_N$ 
  k=0
  init_MLP()  $\rightarrow Q_0$ ;
  Do {
    generate_pattern_set  $P = \{(input^l, target^l), l = 1, \dots, \#D\}$  where:
       $input^l = s^l, u^l$ ,
       $target^l = c(s^l, u^l, s'^l) + \gamma \min_b Q_k(s'^l, b)$ 
    Rprop_training( $P$ )  $\rightarrow Q_{k+1}$ 
    k:= k+1
  } WHILE ( $k < N$ )

```

Abbildung 1: Neural Fitted Q Iteration

In der ersten Iteration des NFQ-Algorithmus werden die Kosten $c(s, u, s')$ für die Zustands-Aktionspaare approximiert. Aus diesem Grund benutzt der Regression-Algorithmus die Zustands-Aktionspaare als Eingaben und die beim Training ermittelte Kosten als Ausgaben. In jeder weiteren Iteration bleiben die Eingaben unverändert und die Ausgaben werden in Bezug auf die im vorherigen Schritt approximierte Q-Funktion aktualisiert.

Um den Algorithmus zu unterbrechen kann der Anzahl der Iterationen fest definiert werden. Andernfalls iteriert der Algorithmus so lange bis die Q-Funktion sich nicht mehr stark ändert [Ernst u. a. 2005]. Dieses Abbruchkriterium ist aber nur dann anwendbar, wenn der Folge Q_N konvergiert.

2.3 Der RL-Lenkungsregler

Um autonomes Fahren zu gewährleisten minimiert der RL-Regler die Abweichung zur Sollspur indem er das Lenkungssystem des Fahrzeugs regelt. Der Schwerpunkt ist diese Abweichung oder anders ausgedrückt cross-track-error (cte) kleiner als 0.5 m zu halten. Der RL-Lenkungsregler wird mittels der NFQ-Methode gelernt. Die Aufgabe des entstehenden Reglers besteht darin, die Kosten zu minimieren. Die Kostenfunktion ist wie folgt definiert:

$$c(s, u) = \begin{cases} 0 & , \text{ wenn } |cte| < 0.05m \text{ (Erfolg)} \\ +1 & , \text{ wenn } |cte| < 0.5m \\ 0,01 & , \text{ sonst} \end{cases} \quad (2)$$

Basierend auf [Hoffmann u. a. 2007] wird der kinematische und dynamische Fahrzeugszustand durch folgende 5 kontinuierlichen Variablen beschrieben:

- *cte*: Abweichung zur Sollspur
- *cte'*: erste Ableitung der cte
- *v*: Geschwindigkeit
- *heading error*: Differenz zwischen Gierwinkel und Krümmung der Fahrbahn
- *yaw – rate – matching*: Differenz zwischen Beschleunigung des Gierwinkels und Beschleunigung der Fahrbahnkrümmung

Anhand dieser Zustandsinformation kann der gelernte Regler den besten Stellwert für die Lenkung bestimmen, indem er die Kosten für unterschiedliche Stellwerte vergleicht.

2.4 Ergebnisse

In diesem Kapitel beschriebenes Verfahren wurde auf einem echten Fahrzeug erfolgreich getestet [Montemerlo u. a. 2007]. Bei einer Geschwindigkeit bis zu 9 m/s wurde schon in 16 Minuten ein RL-Lenkungsregler gelernt, der die Abweichung zur Sollspur kleiner als 0.5 m innerhalb einer vollen Runde der Teststrecke halten konnte.

3 Entwurf eines Lenkungsreglers

Ähnlich dem NFQ-Algorithmus basiert der eigene Lenkungsregler-Ansatz auf der Fitted Q Iteration.

3.1 Fahrzeug-Zustandsbeschreibung

Ziel der Fahrzeug-Zustandsbeschreibung ist die lenkungsrelevante Information vollständig zu formulieren. In [Montemerlo u. a. 2007] werden die Zustandsvariablen mit der aktuellen Fahrzeugposition bezüglich der Fahrspurgeometrie und deren Änderung gebunden. Die Bewertungsfunktion ermittelt die Kosten für die Zustandsübergänge indem die Abweichung zur Sollspur kontinuierlich berechnet wird. [Rajamani u. a. 2000; Rossetter 2003; Gerdes und Rossetter 2001] motivieren eine lookahead Regelung für die automatische Spurführung. Um eine solche Regelung zu ermöglichen, soll der Fahrzeugzustand vollständiges Wissen über Sollspurgeometrie, Fahrzeugposition und -bewegung erhalten. Die genaue Auswahl der Zustandsvariablen ist noch nicht getroffen worden.

3.2 Explorationsstrategie

Um einen Trainingsdatensatz für den Funktionsapproximator zu bilden, verwendet der Fitted Q Iteration Algorithmus Trainingsdaten, die während einer Trainingsphase gesammelt werden. In Methoden von Reinforcement Learning werden üblicherweise die Agentaktionen in dieser Phase von einer Explorationsstrategie bestimmt. Q-Learning Methode sind generell wenig empfindlich gegen Explorationsstrategien solange alle Zustands-Aktionspaare oft genug bewertet werden [Kaelbling u. a. 1996]. Jedoch führen effiziente Strategien zur Minimierung der Trainingszeit. Deren Vor- und Nachteile sind noch nicht detailliert untersucht worden.

Um eine stabile spurerkennungbasierte Fahrzeugführung zu ermöglichen, soll das Fahrzeug während der Trainingsphase dauerhaft auf der zu verfolgenden Spur bleiben. Zu diesem Zweck ist eine Begrenzung des Aktionsraums erforderlich. Die Grenze des Aktionsraums werden mittels eines Verfahrens zur Fahrbahnverfolgung festgelegt. Dieses Verfahren lokalisiert kontinuierlich ein Ziel, dass das Fahrzeug verfolgen soll. Um den Lenkwinkel in Bezug auf das Ziel zu bestimmen wird der Pure Pursuit Algorithmus verwendet [Nikolov 2009]. Dieser Ansatz führt gleichzeitig zu einer präziseren Funktionsapproximation, da der Aktionsraum deutlich reduziert wird.

3.3 Funktionsapproximator

Der Auswahl der Approximationsmethode in der Fitted Q Iteration bleibt dem Anwender überlassen. Hinsichtlich des Lenkungsreglers soll diese Methode auch für hochdimensionale nichtlineare Funktionen eine genaue Approximation leisten können, weil der Fahrzeugzustand durch mehrere kontinuierlichen Variablen beschrieben wird. Zu diesem Zweck eignet sich eine Regression mittels Support Vector Machines (SVM) [Drucker u. a. 1996]. SVM sind ein Teilgebiet des überwachten Lernens und wurden ursprünglich nur zur Klassifizierung eingesetzt [Cortes und Vapnik 1995]. Diese versuchen die Klassengrenzen optimal zu bilden, indem die ihnen am nächsten liegenden Objekte stärker die Trennebene beeinflussen. Zur Trennung von nicht linear separierbaren Daten wird der Kerneltrick angewandt. Um eine Regression zu leisten werden die Support Vector Machines um eine Verlustfunktion erweitert [Drucker u. a. 1996]. [Martin 2002; Shin u. a. 2009] nutzen erfolgreich diese Methode.

4 Testumgebung

4.1 Onyx Fahrzeug

Zur Teilnahme am Carolo-Cup 2009 wurde an der HAW Hamburg das Fahrzeug Onyx entwickelt. Dieses Fahrzeug basiert auf einem Ford F-350 Pickup-Modell im Maßstab 1:10. Die Auswahl von Sensoren wurde anhand der Erfahrungen aus dem Carolo-Cup 2008 und den dort zu bewältigenden Aufgaben getroffen [Hensel 2008]. Zur Umfelderkennung verfügt das Fahrzeug über Inkrementalgeber, Ultraschallsensoren, Infrarotsensoren, Kompass und Kamera mit einem Weitwinkelobjektiv.



Abbildung 2: Onyx Fahrzeug

Als Steuerungselektronik werden drei ARM 7 Prozessoren mit entsprechenden IO-Plattinen

für die Sensordatenverarbeitung verwendet. Die eigentliche Rechenleistung stellt ein Acer Aspire One Subnotebook zur Verfügung, welches mit einem Intel Atom Prozessor mit 1.6 GHz ausgestattet ist. Die Kamera und die ARM-Prozessoren sind jeweils über USB mit dem Notebook verbunden. Auf dem Acer Aspire One Subnotebook ist ein Debian Linux mit dem RT-Preempt Realtime-Kernel eingerichtet.

Die C++ Systemsoftware FAUSTcore stellt abstrakte Basiskomponenten zur Implementierung von Treibern und Tasks zur Verfügung [Jenning 2009]. Zudem ermöglicht der FAUSTcore die bequeme Steuerung der Anwendung durch Parametereinstellungen über eine Web-Oberfläche. Der Task-Scheduler basiert auf der Subsumption Architektur [Brooks 1986]. In der softwaretechnischen Umsetzung von Algorithmen für die Bildverarbeitung und das maschinelle Lernen wird die OpenCV Bibliothek [opencv 2006] verwendet.

4.2 TORCS Simulator

TORCS (The Open Racing Car Simulator) ist eine freie Autorennsimulation. Die Visualisierung basiert auf OpenGL; die Implementierung benutzt dabei plattformunabhängige Bibliotheken wie Mesa 3D und GLUT (OpenGL Utility Toolkit) (bzw. auch FreeGLUT) und ist auf verschiedene Plattformen (Linux, FreeBSD, MacOS X und Windows) portiert. Die von der Software kontrollierten Fahrer werden innerhalb des Projekts als Robots bezeichnet. Die Eigenschaften und das Verhalten eines Robots können über einen Teil des Quellcodes in C oder C++ programmiert werden. Es existiert auch ein Streckeneditor zum Erstellen eigener Strecken [wiki].



Abbildung 3: TORCS

5 Zusammenfassung

In dieser Ausarbeitung wurde ein Verfahren zum Entwurf eines Reglers mittels Reinforcement Learning vorgestellt. Das Verfahren basiert auf dem NFQ-Algorithmus und ist komplett datengetrieben d.h. keine Simulation oder Fahrzeugmodell ist erforderlich. NFQ ist eine Instanz der Fitted Q Iteration Familie von Algorithmen, wobei der Regression-Algorithmus durch ein künstliches neuronales Netz realisiert wird (vgl. Kapitel 2).

Der im Kapitel 3 beschriebene Reglerentwurf basiert ebenfalls auf der Fitted Q Iteration. Im Gegenteil zu NFQ setzt der eigene Ansatz Pure Pursuit in der Trainingsphase ein. Dadurch wird ein relativ stabiles Lenkungsverhalten gewährleistet d.h. das Fahrzeug bleibt dauerhaft auf der zu verfolgenden Spur. Außerdem minimiert sich gleichzeitig die Trainingszeit, weil der Aktionsraum begrenzt wird. Eine effiziente Explorationsstrategie soll das Lernverfahren optimieren. Weiterhin wird als Regression-Algorithmus für die Approximation der Q-Funktion eine Support Vector Regression Machine anstatt eines neuronalen Netzes verwendet.

6 Aktueller Stand

Im Projekt 1 [Nikolov 2010] wurde das Verfahren zur Fahrbahnverfolgung [Nikolov 2009] mittels der Neural Fitted Q Iteration Methode optimiert und auf einem Modellfahrzeug des VW Touareg im Maßstab 1:10 getestet. Innerhalb von nur 3 Minuten bei einer Geschwindigkeit zwischen 0.8 m/s und 1.3 m/s wurde ein Lenkungsregler gelernt, der erfolgreich die Abweichung zur Sollspur minimiert. Dieser Regler ermöglicht eine autonome Fahrbahnverfolgung auf einer Teststrecke, die mit dem Carolo-Cup-Regelwerk konform ist. Die vom Fahrzeug erreichte mittlere Geschwindigkeit beträgt 2,2 m/s.

Um eine Simulationsumgebung zu schaffen und weitere Stabilitätsanalyse durchzuführen wurde dieses Verfahren im TORCS Simulator implementiert [Nikolov 2011]. Ein optimales Lenkungsverhalten wurde mit einer Frequenz von 20 Hz erreicht. Lenkungsverhalten und Verlauf der Abweichung zur Sollspur innerhalb einer Runde der Teststrecke bei dieser Reglerfrequenz sind in Abbildung 4 dargestellt, wobei auf der x-Achse die Zeit im 20 Millisekundentakt eingetragen ist. Daraus ist zu erkennen, dass der NFQ-Regler im Vergleich zur Pure Pursuit basierten Spurführung erfolgreich die Abweichung zur Sollspur minimiert. Weiterhin wurde am besten geeignete Zielentfernung für das Pure Pursuit basierende Spurführung [Nikolov 2009] dynamisch festgelegt, indem die Zielentfernung als Aktion für den NFQ-Regler angewandt wurde. Dieser Ansatz führte auch erfolgreich zu einem optimierten Lenkungsverhalten.

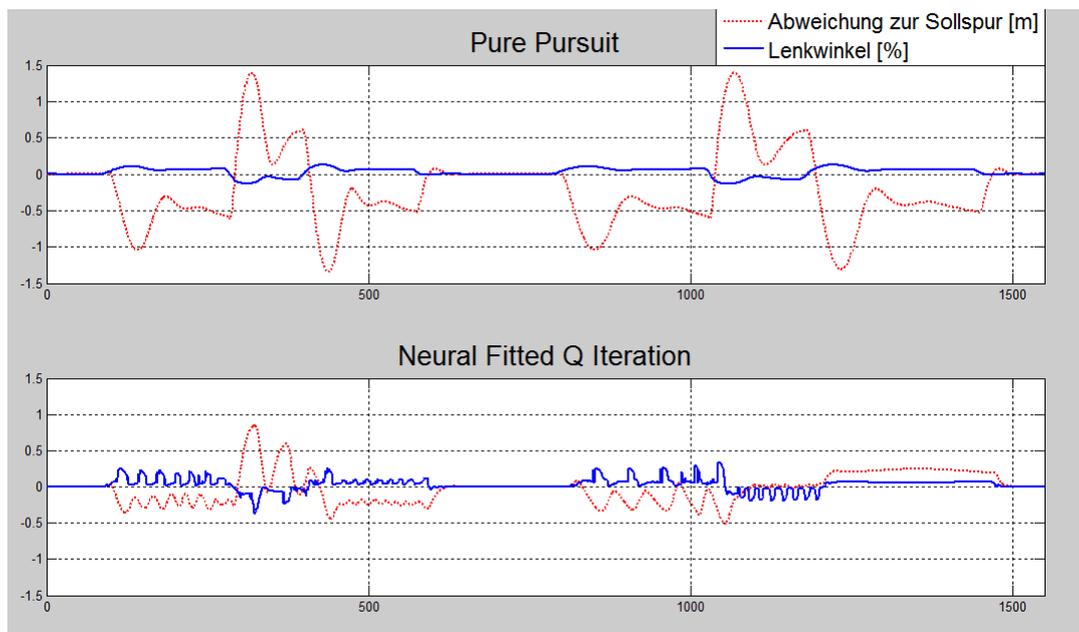


Abbildung 4: Lenkungsverhalten in einer vollen Runde der Teststrecke

Abbildungsverzeichnis

1	Neural Fitted Q Iteration	4
2	Onyx Fahrzeug	7
3	TORCS	8
4	Lenkungsverhalten in einer vollen Runde der Teststrecke	10

Literatur

- [Alpaydin 2004] ALPAYDIN, Ethem: *Introduction to Machine Learning*. The MIT Press Cambridge, Massachusetts London, England, 2004
- [Brooks 1986] BROOKS, R. A.: A Robust Layered Control System for a Mobile Robot. In: *IEEE Journal of Robotics and Automation* 2. 1986
- [Carolo-Cup] CAROLO-CUP: *Homepage des Carolo-Cup Wettbewerbs*. – URL <http://www.carolocup.de>
- [Carolo-Cup 2010] CAROLO-CUP: *Carolo-Cup Regelwerk*. 2010. – URL <http://www.carolo-cup.de/uploads/media/Regelwerk2011.pdf>
- [Cortes und Vapnik 1995] CORTES, Corinna ; VAPNIK, Vladimir: Support-vector networks. In: *Machine Learning* 20 (1995), S. 273–297. – URL <http://dx.doi.org/10.1007/BF00994018>. – ISSN 0885-6125
- [Drucker u. a. 1996] DRUCKER, Harris ; BURGESS, Chris J. ; KAUFMAN, Linda ; SMOLA, Alex ; VAPNIK, Vladimir: *Support Vector Regression Machines*. 1996
- [Ernst u.a. 2005] ERNST, D. ; ; WEHENKEL, L. ; GEURTS, P.: *Tree-based batch mode reinforcement learning*. *Journal of Machine Learning Research*. 2005. – URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.63.7705&rep=rep1&type=pdf>
- [FAUST 2010] FAUST: *FAUST Homepage*. 2010. – URL <http://www.informatik.haw-hamburg.de/faust.html>
- [Gerdes und Rosseter 2001] GERDES, J. C. ; ROSSETTER, Eric J.: A Unified Approach to Driver Assistance Systems Based on Artificial Potential Fields. In: *Journal of Dynamic Systems, Measurement, and Control, Vol. 123*, URL <http://www-cdr.stanford.edu/dynamic/PF/papers/pfieldsJDSMC.pdf>, 2001
- [Hensel 2008] HENSEL, Enrico: *Führungskonzept eines autonomen Fahrzeuges*, Hochschule für Angewandte Wissenschaften Hamburg, Bericht Anwendung 1, 2008
- [Hoffmann u. a. 2007] HOFFMANN, Gabriel M. ; TOMLIN, Claire J. ; MONTEMERLO, Michael ; THRUN, Sebastian: *Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing* / Stanford University. 2007. – Forschungsbericht

- [Jenning 2009] JENNING, Eike: *Systemidentifikation eines autonomen Fahrzeugs mit einer robusten, kamerabasierten Fahrspurerkennung in Echtzeit*, Hochschule für Angewandte Wissenschaften Hamburg, Masterarbeit, 2009
- [Kaelbling u. a. 1996] KAEHLING, Leslie P. ; LITTMAN, Michael L. ; MOORE, Andrew W.: Reinforcement Learning: A Survey. In: *Journal of Artificial Intelligence Research* 4, URL <http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume4/kaelbling96a.pdf>, 1996
- [Martin 2002] MARTIN, Mario: *On-line Support Vector Machines for Function Approximation*. 2002. – URL <http://www.lsi.upc.edu/~mmartin/repsvm.pdf>
- [Montemerlo u. a. 2007] MONTEMERLO, Mike ; DAHLKAMP, Hendrik ; RIEDMILLER, Martin: Learning to Drive a Real Car in 20 Minutes. In: *FBIT conference, Jeju, Korea. Special Track on autonomous robots.*, URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.70.3532&rep=rep1&type=pdf>, 2007
- [Ng 2008] NG, Andrew: *Machine Learning (Stanford)*. 2008. – URL http://www.youtube.com/watch?v=-ff615D8-j8&feature=Playlist&p=A89DCFA6ADACE599&playnext_from=PL&index=18
- [Nikolov 2009] NIKOLOV, Ivo: *Verfahren zur Fahrbahnverfolgung eines autonomen Fahrzeugs mittels Pure Pursuit und Follow-the-carrot*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2009
- [Nikolov 2010] NIKOLOV, Ivo: *NFQ zur Optimierung eines Lenkungsreglers*, Hochschule für Angewandte Wissenschaften Hamburg, Projekt 1 Ausarbeitung, 2010
- [Nikolov 2011] NIKOLOV, Ivo: *Stabilitätsanalyse eines Lenkungsreglers mittels TORCS*, Hochschule für Angewandte Wissenschaften Hamburg, Projekt 2 Ausarbeitung, 2011
- [opencv 2006] OPENCV: *OpenCV Library. Version 1.0*. 2006. – URL <http://opencv.willowgarage.com/wiki/>
- [Rajamani u. a. 2000] RAJAMANI, Rajesh ; TAN, R. Han-Shue ; LAW, Boon K. ; ZHANG, Wei-Bin: Demonstration of Integrated Longitudinal and Lateral Control for the Operation of Automated Vehicles in Platoons. In: *European Symposium on Artificial Neural Networks, Bruges*, 2000
- [Riedmiller 2005] RIEDMILLER, Martin: Neural Fitted Q Iteration - First experiences with a data efficient neural Reinforcement Learning Method. In: *European Conference on Machine Learning, Porto, Portugal*, URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.72.1193&rep=rep1&type=pdf>, 2005

- [Rossetter 2003] ROSSETTER, Eric J.: *A Potential Field Framework for Active Vehicle Lanekeeping Assistance*, Stanford University, PhD Thesis, 2003. – URL ddl.stanford.edu/files/eric_diss.pdf
- [Shin u. a. 2009] SHIN, Jongho ; KIM, H. J. ; PARK, Sewook ; KIM, Youdan: *Model predictive flight control using adaptive support vector regression*. 2009. – URL <http://www.ksiam.org/conference/annual091/upfile/jhshin.pdf>
- [Sutton und Barto 1998] SUTTON, R. S. ; BARTO, A. G.: *Reinforcement Learning*. MIT Press, Cambridge, MA, 1998
- [Weiser u. a. 2009] WEISER, Andreas ; BARTELS, Dr. A. ; STEINMEYER, Simon ; SCHULTZE, Dipl.-Ing K. ; MUSIAL, Dr. M. ; WEISS, Dr. K.: *Intelligent Car - Teilautomatisches Fahren auf der Autobahn*. In: *AAET*, 2009
- [wiki] WIKI: *TORCS*. – URL <http://de.wikipedia.org/wiki/TORCS>