Graduiertenkolleg 1103

Embedded Microsystems

Albert-Ludwigs-Universität Freiburg

Effiziente Klassifikationsverfahren für eingebettete Systeme

Statusbericht

Axel Rottmann

Betreuer: Prof. Dr. Wolfram Burgard
Lehrstuhl: Autonome Intelligente Systeme

Freiburg, im September 2008

Institut für Informatik

Institut für Mikrosystemtechnik

1 Aktueller Stand der Promotion

Meine Promotion befindet sich im dritten Jahr. Die Demonstrationsplattform Blimp (Prallluftschiff) ist fertig gestellt und derzeit evaluiere ich Lernstrategien zur autonomen Steuerung des Blimps. Danach erfolgt das Schreiben der Dissertation.

2 Zusammenfassung der Dissertation

Eingebettete Systeme unterliegen in der Regel bestimmten Anforderungen wie geringes Gewicht, beschränkte Leistungsaufnahme und sie müssen mit anderen Systemen kommunizieren. Damit diese Systeme ihre Aufgabe effizient erfüllen, ist das Verhalten meist zuvor bestimmt worden und dem System fest vorgegeben. In machen Situationen kann es jedoch von Vorteil sein aktiv auf äußere Einflüsse oder Änderungen in der Umwelt reagieren zu können. Daher beschäftige ich mich im Rahmen meiner Doktorarbeit mit der Entwicklung von Verfahren, die es eingebetteten Systemen erlauben, selbstständig Entscheidungen zu treffen und somit das Verhalten den momentanen Gegebenheiten anzupassen.

2.1 Einleitung

Damit eingebettete Systeme intelligent autonom operieren können, werden Informationen über die Umwelt und Auswirkungen der Aktionen benötigt. Als Anwendungsbeispiel untersuchen wir die Aufgabe die Höhe eines Luftschiffs ohne jegliches Vorwissen zu kontrollieren. Dem System sind dabei keine Informationen über die Auswirkungen der Aktionen oder Gegebenheiten der Umwelt bekannt. Das System soll selbständig seine Aktionen bewerten, um seine Flughöhe zu stabilisieren. Die Idee besteht darin, gegeben einem Zielzustand, einzelne Aktionen auszuprobieren, diese zu bewerten, um schließlich eine bestmögliche Strategie zu entwickeln. Dafür verwenden wir *Lernen durch Verstärkung* (Reinforcement Learning), das dem Lernen durch Versuch und Irrtum nachempfunden wurde.

2.2 Das autonome Luftschiff

Um das Potential von eingebetteten Mikrosystemen in praktischen Anwendungen demonstrieren zu können, haben wir ein autonomes Prallluftschiff (engl. Blimp) entwickelt, das als Demonstrationsplattform dienen soll. Ein Blimp für den Innenbereich unterliegt den gleichen Bedingungen wie eingebettete Mikrosysteme. Durch den geringen Auftrieb bestehen Einschränkungen beim Gewicht, bei der Leistungsaufnahme sowie bei der Größe der Systemkomponenten. Ebenso sollte ein Blimp, wie ein eingebettetes System, möglichst unabhängig von der Außenwelt operieren.

Als Gemeinschaftsprojekt (vgl. Abschnitt 3) entstand das in Abbildung 1(a) dargestellte System. Als Basis diente eine handelsübliche Hülle mit einer Länge von 1,8 m und einem Durchmesser von 0,9 m. Gesteuert wird der Blimp durch drei Motoren. Einer befindet sich in der Heckflosse sowie jeweils einer links und rechts der Gondel (siehe Abbildung 1(b)). Durch die Abmessungen der Hülle erhalten wir einen Auftrieb von etwa 500 g, was abzüglich der Hülle, der Gondel und der Motoren zu einem verbleibenden Gewicht von etwa 120 g für die Systemkomponenten führt. Eine genauere Erläuterung des gesamten Systems ist in unserer Arbeit [4] zu finden.

(a)

(b)

Abbildung 1: (a) Das Luftschiff mit einer Länge von 1,8 m, einem Durchmesser von 0,9 m und einem Gesamtgewicht von 500 g. Die gesamte Hardware ist in der Gondel untergebracht. (b) Die Gondel mit den beiden Motor um Nickwinkel und Schub zu kontrollieren. Die Motoren können dazu um 180 Grad gedreht werden.

2.3 Reinforcement Learning

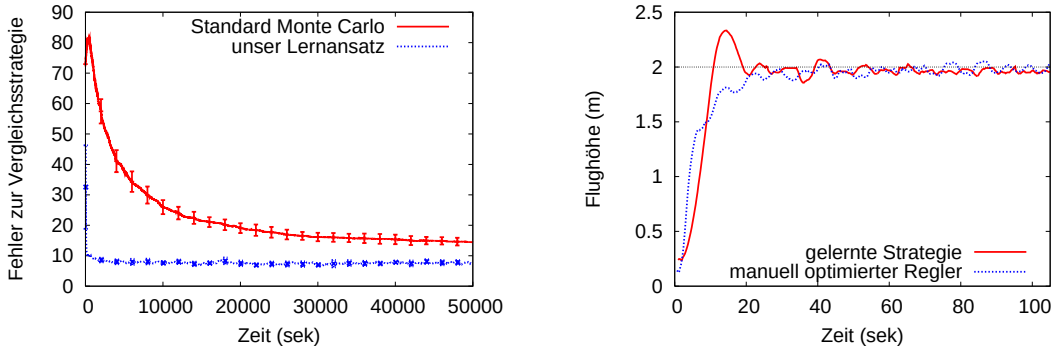
Reinforcement Learning [8] basiert auf der Idee, dass ein Agent mit einer ihm komplett unbekannten Umwelt interagiert und sein Verhalten basierend auf Belohnung und Bestrafung bewertet. Im Allgemeinen erhält er für Aktionen, die ihn seinem Ziel näher bringen, eine höhere Belohnung als für solche, die ihn davon entfernen. Letztlich ist es das Ziel des Agenten sich so zu verhalten, dass er auf lange Sicht eine möglichst hohe Belohnung erhält. Formal kann ein Reinforcement Learning Prozess durch ein Tupel $\{S, A, \delta, r\}$, bestehend aus einer Menge von Zuständen S, Aktionen A, einer Bewegungsfunktion $\delta : S \times A \rightarrow S$ und einer Belohnungsfunktion $r : S \times A \rightarrow \mathbb{R}$, beschrieben werden. Die Belohnungsfunktion gibt dabei an, welche Belohnung erhalten wird, wenn Aktion a im Zustand s ausgeführt wird. Das Ziel ist es nun eine Strategie $\pi : S \rightarrow A$ zu bestimmen, unter der eine möglichst hohe Belohnung erreicht wird.

In unserem Verfahren wird die sogenannte Monte Carlo Methode des Reinforcement Learnings angewandt. Der Vorteil dieser Methode liegt darin, dass direkt auf den Erfahrungen des Agenten gelernt wird, ohne Wissen über die Bewegungsfunktion δ zu haben. Ziel ist es die Q-Funktion $Q(s, a) : S \times A \rightarrow \mathbb{R}$ zu bestimmen, die die zu erwartende Belohnung repräsentiert, wenn Aktion a in Zustand s ausgeführt wird. Am Ende des Lernverfahrens kann die beste Aktion $\pi(s)$ für einen gegebenen Zustand s anhand der Q-Funktion wie folgt bestimmt werden

$$\pi(s) = \arg\max_{a} Q(s, a)\,. \tag{1}$$

2.4 Lernen der Q-Funktion

Um Reinforcement Learning auf dem echten Blimp effizient anwenden zu können, bedarf es weiterer Überlegungen. Im Allgemeinen muss jeder Zustand unendlich oft besucht werden, um die optimale Strategie zu erlernen. In unserem Fall ist jedoch bereits ein Besuch aller Zustände in akzeptabler Zeit nicht möglich. Daher repräsentieren wir die Q-Funktion durch einen *Gauß'schen Prozess* [2]. Mit einem Gauß'schen Prozess können wir, basierend auf Datenpunkten $\mathcal{D} = \{(\boldsymbol{x}_i, q_i)\}_{i=1}^{D}$ von Zustands-Aktions Paaren $\boldsymbol{x}_i = (s_i, a_i) \in S \times A$ und erwarteter Belohnung q_i, die wahren Q-Funktion approximieren. Dabei wird angenommen, dass die Zielwerte gaußverteilt sind $(q_1, \ldots, q_D) \sim \mathcal{N}(0, K)$, gegeben einer Kovarianzmatrix K. Ein neuer Zielwert q_{D+1} lässt sich schließlich durch eine eindimensionale Gaußverteilung $p(q_{D+1} \mid \boldsymbol{x}_{D+1}, \mathcal{D}, \boldsymbol{\theta})$ vorhersagen.



(a) (b)

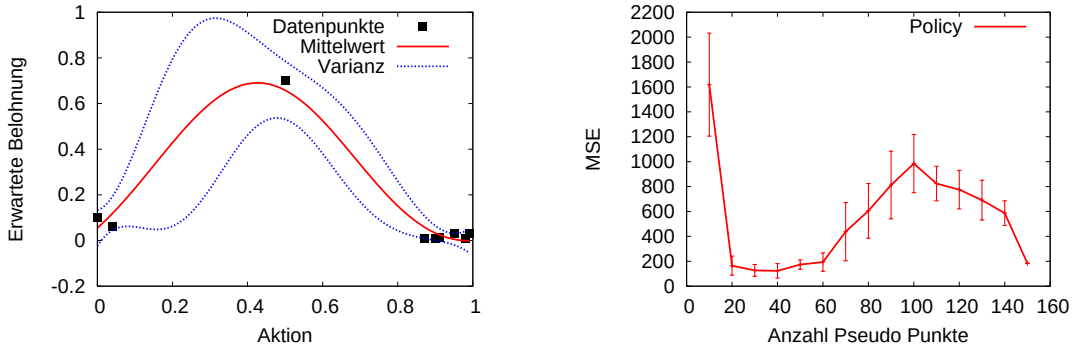
Abbildung 2: (a) Der Vergleich zwischen unserem Lernansatz und dem Standard Monte Carlo Ansatz. (b) Flugbahn des Blimps basierend auf einer durch unser Verfahren zuvor gelernten Strategie und auf einem manuell optimierten $\mathrm{PD}^2 - \mathrm{T}_2$ Regler.

Entscheidende Kriterien für die Güte der Regression sind die Wahl der Hyperparameter $\boldsymbol{\theta}$ der Kovarianzmatrix K sowie die Auswahl der Datenpunkte $\mathcal{D}$. Die optimalen Hyperparameter können während des Lernens durch Maximierung der Likelihood der Daten $p(q_1, \ldots, q_D \mid \boldsymbol{x}_1, \ldots, \boldsymbol{x}_D, \boldsymbol{\theta})$ bestimmt werden.

In unserer Arbeit [3] haben wir bereits gezeigt, dass die Q-Funktion durch Gauß'sche Prozesse sehr gut vorhergesagt werden kann. Eine gute Approximation der wahren Q-Funktion kann bereits durch eine geringe Anzahl an Datenpunkten $\mathcal{D}$ erzielt werden. In Abbildung 2(a) ist der Lernfortschritt unseres Ansatzes im Vergleich zum Standard Reinforcement Learning Ansatz zu sehen. Es wird der quadratische Fehler zu einer Vergleichsstrategie angegeben, die basierend auf der wahren Dynamik des Systems gelernt wurde. Wie man sieht, können wir mit unserem Ansatz wesentlich schneller eine bessere Strategie erlernen. Die Durchschnittliche Lerndauer für eine gute Strategie beträgt etwa 200 Sekunden. Dabei ist zu beachten, dass das System keinerlei Vorwissen über die Auswirkungen seiner Aktionen wie auch über die Parameter der Umwelt hat. Ebenso haben wir gezeigt (siehe Abbildung 2(b)), dass die Strategie, die nach unserem Ansatz zuvor gelernt wurde, mit einem manuell optimierten $\mathrm{PD}^2 - \mathrm{T}_2$ Regler vergleichbar ist und ein ähnlich gutes Verhalten erzielt.

Weitere wichtige Punkte bezüglich Lernkomplexität und Lerneffizienz ist zum einen die optimale Wahl der Aktionen a sowie die Anzahl und Auswahl der Datenpunkte $\mathcal{D}$. Eine weit verbreitete Strategie zur Wahl der nächsten Aktion ist die ϵ-greedy Strategie. Bei dieser Strategie wird mit Wahrscheinlichkeit $\epsilon \in [0, 1)$ eine Zufallsaktion ausgeführt, andernfalls die momentan beste Aktion nach Gleichung (1). Wünschenswert wäre es jedoch Aktionen gezielt auszuwählen, die am meisten Information für das Lernverfahren versprechen. Durch Ausnutzung der Varianz in Gauß'schen Prozessen und dem in [5] verwendeten Ansatz, ist die Auswahl solcher Aktionen möglich.

Abbildung 3(a) zeigt eine exemplarische Q-Funktion. Da die beste Aktion durch Maximierung der erwarteten Belohnung definiert ist (vgl. Gleichung 1), läge in diesem Beispiel die derzeit beste Aktion im Bereich von 0,42. Unter Berücksichtigung der Unsicherheit, wäre jedoch eine Aktion im Bereich von 0,3 zu bevorzugen.



(a) (b)

Abbildung 3: (a) Exemplarische Q-Funktion zur Wahl der nächsten Aktion für einen gegebenen Zustand. (b) Fehler zur optimalen Strategie durch Approximation mit einem Gauß'schen Prozess mit reduzierter Anzahl an Datenpunkte.

2.5 Ausblick

Die Laufzeit zur Berechnung Gauß'scher Prozesse hängt stark von der Anzahl der Datenpunkte $\mathcal{D}$ ab. Es gibt verschiedene Arbeiten [7, 1] in denen Ansätze zu Datenreduktion vorgestellte werden. Zu Berücksichtigen ist jedoch, dass die Punkte iterative während des Lernens gesammelt und in den Gauß'schen Prozess integriert werden. Ebenso ist das Bewegungsmodell eines Blimps nicht fix und kann sich während des Flugs, gegeben äußerer Einflüsse wie Batteriespannung oder Auftrieb der Hülle, verändern. Die optimale Auswahl der Datenpunkte $\mathcal{D}$ sowie effiziente Verfahren sind jedoch Gegenstand aktueller Untersuchungen. Bisherige Ergebnisse haben gezeigt, dass die Approximationsgüte durch wenige Punkte bereits hinreichend gut ist. Besonders bemerkenswert ist dabei, dass eine bessere Vorhersage der optimalen Policy durch eine Reduktion der Datenpunkte erreicht werden kann (siehe Abbildung 3(b)). Gründe hierfür und effiziente Verfahren zur Auswahl dieser Punkte werden in der verbleibenden Zeit meine Promotion untersucht.

3 Zusammenarbeit

Das in Abbildung 1 gezeigt Luftschiff ist in einer Kooperation innerhalb des Graduiertenkollegs mit den Stipendiaten Matthias Sippel aus der Arbeitsgruppe „Elektrische Mess- und Prüfverfahren“ und Thorsten Zitterell aus der Arbeitsgruppe „Betriebsysteme“ entstanden. Ziel der Kooperation war die Entwicklung einer Plattform zur Evaluation von Sensoren und Algorithmen für eingebettete Systeme. Daher lagen die Herausforderungen im beschränkten Gewicht und der geringen Leistungsaufnahme der Systemkomponenten. Aus dieser Kooperation entstand ein miniaturisiertes Luftschiff mit einem onboard Linux-System, generischen Schnittstellen zu Sensoren und Aktoren sowie einer Standard IEEE 802.11g WLAN Verbindung. Das System wurde bisher in zwei gemeinsamen Arbeiten [6, 4] veröffentlicht. In Zusammenarbeit mit Peter Hilger von der Arbeitsgruppe „Systemtheorie“ wurde ein $\mathrm{PD}^2 - \mathrm{T}_2$ Regler zur Höhenkontrolle entworfen, welcher zur Evaluation meines Lernverfahren diente [3]. Des Weiteren arbeitet der Stipendiat Ali Cubukcu aus der Arbeitsgruppe „Sensoren“ derzeit an einem Flusssensor der im Laufe des nächsten Jahres auf dem Blimp integriert werden soll. Dieser Sensor könnte beispielsweise bei der Positionsschätzung wichtige Daten über die aktuelle Geschwindigkeit liefern.

Neben all diesen Kooperationen fand ebenfalls eine enge Zusammenarbeit mit Martin Cornils aus der Arbeitsgruppe „Materialien der Mikrosystemtechnik“ statt. Darin wurden Algorithmen zur Nullstellensuche bei Funktion mit vielen Variablen mit Randbedingungen untersucht. Anwendung fand dies unter anderem in der Bestimmung der optimalen Hyperparameter θ der Kovarianzfunktion.

Literatur

[1] Y. Engel, S. Mannor, and R. Meir. “Reinforcement learning with gaussian processes”. In *Proceedings of the 22nd international conference on Machine learning (ICML)*. 2005, pp. 201–208.

[2] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press, 2006.

[3] A. Rottmann, C. Plagemann, P. Hilgers, and W. Burgard. “Autonomous blimp control using model-free reinforcement learning in a continuous state and action space”. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. San Diego, CA, USA, 2007, pp. 1895–1900.

[4] A. Rottmann, M. Sippel, T. Zitterell, W. Burgard, L. Reindl, and C. Scholl. “Towards an experimental autonomous blimp platform”. In *Proceedings of the 3rd European Conference on Mobile Robots (ECMR)*. Freiburg, Germany, 2007, pp. 19–24.

[5] R. Ruben Martinez-Cantin, N. de Freitas, A. Doucet, and J. Castellanos. “Active policy learning for robot planning and exploration under uncertainty”. In *Processing of Robotics: Science and Systems (RSS)*. 2007.

[6] M. Sippel, A. Rottmann, T. Zitterell, B. Steder, C. Scholl, W. Burgard, and L. Reindl. “Multisensor-Navigation für autonome Flugroboter”. In *Sensoren und Messsysteme 2008*. ser. VDI-Berichte 2011, Ludwigsburg, Germany, 2008, pp. 667–676.

[7] E. Snelson and Z. Ghahramani. “Sparse gaussian processes using pseudo-inputs”. In *Advances in Neural Information Processing Systems 18*. Y. Weiss, B. Schölkopf, and J. Platt, Eds. Cambridge, MA: MIT Press, 2006, pp. 1257–1264.

[8] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.