Graduiertenkolleg 1103

Embedded Microsystems

Albert-Ludwigs-Universität Freiburg

Lernverfahren zur Landmarkenselektion

Statusbericht

Hauke Strasdat

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

Freiburg, im September 2008

Institut für Informatik

Institut für Mikrosystemtechnik

1 Aktueller Stand der Dissertation

Meine Dissertation befindet sich im ersten Jahr.

2 Zusammenfassung

Intelligente autonome Robotersysteme benötigen für Navigationsaufgaben im Allgemeinen hohe Anforderungen an Rechenleistung und Speicherbedarf. Ein wichtiger Aspekt im Kontext von eingebetteten Systemen ist die Reduktion dieser Beschränkung. Dies lässt sich erreichen, indem aus der Menge der verfügbaren Landmarken, die der Roboter zur Positionsbestimmung nutzt, nur eine kleine Auswahl für die Navigation verwendet wird. In meiner Arbeit werden Lernverfahren für Landmarkenselektion vorgestellt. An Hand von Experimenten wird gezeigt, dass die gelernten Strategien signifikant besser als naive und manuell erstellte Strategien sind. Darüberhinaus wird demonstriert, dass das Lernverfahren ein hohes Maß an Generalisierbarkeit besitzt: In einem Szenario gelernte Strategien können erfolgreich auf andere Szenarien angewendet werden.

3 Einleitung

In der Robotik gibt es einen Trend hin zu eingebetteten Systemen. Zu nennen sind u.a. Arbeiten mit Helikoptern [6, 1], Luftschiffen [3, 7] und Amphibienfahrzeugen [9]. Wie jedes andere Computersystem haben eingebettete Systeme eine begrenzte Rechenleistung und eine begrenzte Speicherkapazität. Jedoch sind diese Grenzen meist viel enger gesteckt als bei einem modernen Arbeitsplatzsystem. Um so wichtiger ist es, effiziente Algorithmen zu entwickeln, die mit der Rechenleistung skalieren.

Die Navigation von autonomen intelligenten Systemen ist eine Kernanwendung in der Robotik. Bevor übergeordnete Aufgaben wie z.B. die Exploration, Botendienste usw. angegangen werden können, muss der Roboter die Fähigkeit besitzen sich in seiner Umgebung autonom zu bewegen. Für intelligente Roboternavigation wiederum ist das sogenannte *Simultaneous Localization and Mapping* (SLAM) essenziell: Um in der Welt navigieren zu können, braucht der Roboter eine Repräsentation seiner Umwelt; d.h. er muss eine Karte seiner Umgebung erstellen. Gleichzeitig muss geschätzt werden, wo sich der Roboter innerhalb der Karte befindet. Meist besteht diese Karte aus einer Menge von Landmarken - ausgewiesenen Objekte in der Umgebung. Das Standardverfahren für SLAM ist der sogenannte *Extended Kalman Filter* (EKF) [4] und seine Varianten wie z.B. der *Unscented Kalman Filter* (UKF) [2]. Bei diesen Verfahren wächst sowohl der Rechenaufwand als auch der Speicherbedarf quadratisch mit der Anzahl der Landmarken, da die gesamte Korrelation zwischen der Position des Roboters und aller Landmarken untereinander berücksichtigt wird. Deshalb gibt es für SLAM diverse Approximationen, wie z.B. der *Rao-Blackellized Particle Filter* (RBPF) [5] oder *Sparse Extended Information Filter* (SEIF) [11]. Diese Approximationen haben alle gemeinsam, dass sie nicht die gesamte Korrelation zwischen den Landmarken berücksichtigen und somit eine schnellere Laufzeit haben. Ihr Speicherbedarf hängt jedoch mindestens linear von der Anzahl der Landmarken ab.

Eine andere Möglichkeit um Rechenleistung und Speicherplatz zu sparen ist es, aus der Menge der zur Verfügung stehenden Landmarken eine Auswahl zu treffen. Zhang u.a. [12] stellen eine Landmarkenselektionsstrategie vor, die auf einer Heuristik - der Entropiereduktion - basiert. Es ist jedoch unklar, wie sich diese Heuristik auf Navigationsaufgaben mit einer harten Speicherplatzbeschränkung anwenden lässt. In meiner Doktorarbeit untersuche ich Strategien zur Landmarken-

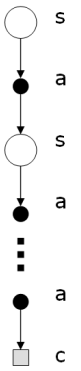

Abbildung 1: **Episode.**

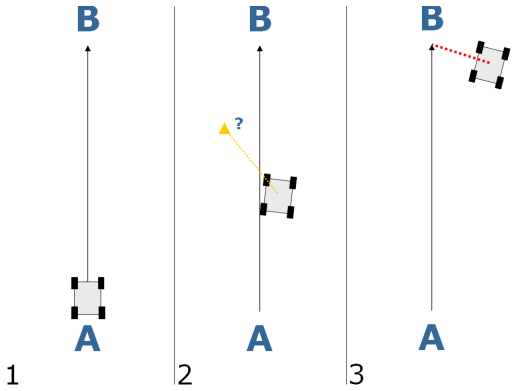

Abbildung 2: **Navigationsaufgabe.**

selektion mit Hilfe eines Lernverfahrens - dem sogenannten *Reinforcement Learning* [8]. Dabei betrachte ich exemplarisch die Landmarkenselektion beim UKF-SLAM. Allerdings ist es genauso gut möglich die Lernverfahren zur Landmarkenselektion auf andere SLAM-Verfahren wie EKF, RBPF oder SEIF anzuwenden.

4 Grundlagen

Beim Simultaneous Localization and Mapping wird die gemeinsame Wahrscheinlichkeitverteilung

$$p(x_t, l_{1:N}|u_{1:t}, z_{1:t})$$

über die Pose des Roboters x_t zum Zeitpunkt t und die Position der Landmarken $l_{1:N}$ gegeben aller vorherigen Bewegungen $u_{1:t}$ und Beobachtungen $z_{1:t}$ geschätzt [10]. Im Falle von UKF-SLAM wird diese Verteilung mit einer multivariaten Normalverteilung mit Mittelwert μ und Kovarianz Σ approximiert. Dabei werden Nichtlinearitäten im Sensor- und Bewegungsmodell mit Hilfe einer Menge von sogenannten Sigmapunkten modelliert [2].

Die Grundidee des Reinforcement Learnings [8] ist es, durch Interaktion mit seiner Umgebung zu lernen. Der Lernprozess besteht in der Regel aus einer Anzahl von Episoden. Jede Episode wiederum besteht aus einer Serien von Zuständen $s \in \mathcal{S}$ und Aktionen $a \in \mathcal{A}$ gefolgt von den finalen Kosten c (siehe Abbildung 1). Beim Lernen wird die *Q-Funktion* approximiert, die jedem Zustands/Aktionspaar einen Kostenwert zuordnet. Gleichzeit wird für diese Q-Funktion die beste Strategie π gelernt, die die geschützten Kosten minimiert:

$$\forall s \in \mathcal{S} : \pi(s) = \arg\min_{a \in \mathcal{A}} Q(s, a)$$

5 Lernverfahren zur Landmarkenselektion

Betrachten wir folgende Navigationsaufgabe (siehe Abbildung 2). Der Roboter befindet sich auf der Position A. Nun soll er möglichst genau zum Ziel B fahren. Das Bewegungsmodell des Roboters unterliegt jedoch einem Fehler, so dass seine Bewegung eine gewisse Abdrift wiederfährt. Darüber hinaus befinden sich in der Umgebung M Landmarken. Sobald der Roboter eine neue Landmark sieht, muss entschieden werden, ob diese neue Landmark in den UKF aufgenommen

werden soll, um seine Positionsschätzung zu verbessern. Der UKF besitzt eine Landmarkenkapazität von N Landmarken mit $N \ll M$. Ziel ist es, den Abstand zwischen der Endposition des Roboters und dem Ziel B zu minimieren.

Will man mit Reinforcment Learning eine Landmarkenselektionstrategie erlernen, so entspricht der Abstand des Roboters zum Ziel B den finalen Kosten c. Nun muss noch der Zustandsraum $\mathcal{S}$ und der Aktionsraum $\mathcal{A}$ definiert werden. Die verfügbare Information für den Zustandsraum besteht aus dem UKF Zustand $\langle \mu, \Sigma \rangle$ und der Position der potentiell neuen Landmark. Ein aus der vollständigen Information bestehender Zustandsraum ist jedoch zu hochdimensional für eine erfolgreiches Lernen. Wünschenswert ist es, diesen Raum so zu reduzieren, dass möglichst viel relevante Information erhalten bleibt. Dies geschieht anhand von Merkmalen, die die essentielle Information zusammenfassen. Folgende vier Merkmale haben sich als besonders relevant erwiesen: **dist2goal** (geschätzte Distanz zum Ziel B), **number of landmarks** (Anzahl der bereits in den UKF integrierten Landmarken), **phi** (relativer Winkel zur potentiell neuen Landmarke) und **dist2closest** (Abstand der potentiell neuen Landmarke zur am nächsten liegenden bereits integrierten Landmarke). Weitere Merkmale, wie z.B. die Entropie der Roboterpose, sind denkbar. Im folgenden werden zwei Lernstrategien verglichen. Eine, die auf den ersten beiden Merkmalen beruht und eine, die alle vier verwendet. Bei unserer Lernaufgabe ist die Aktion eine binäre Entscheidung: Entweder wird die potentiell neue Landmarke ausgewählt oder nicht.

6 Generalisierung

Bis jetzt haben wir ein Verfahren betrachtet, um eine Strategie für ein spezifisches Szenario zu lernen. Wünschenswert ist es jedoch eine Strategie in einem Szenario zu lernen und diese Strategie dann in einem anderen Szenario anzuwenden. Wichtige Parameter eines Lernszenarios sind einerseits M - die Anzahl der Landmarkenverteilung in der Umgebung und andererseits N - die Landmarkenkapazität des UKFs. Um eine Generalisierung zu ermöglichen ist es wichtig, dass wir eine Szenario-unabhängige Darstellung des Zustandsraums haben. Zum Beispiel, müssen wir anstelle von der Anzahl der integrierten Landmarken (number of landmarks) jetzt von dem Prozentsatz der bereits integrierten Landmarken sprechen.

7 Experimente und Ergebnisse

Die Performanz des Lernverfahrens wird in einer Simulationsumgebung evaluiert. Dabei wird die zu evaluierende Lernstrategie über 2000 Episoden trainiert. In jeder Episode werden die Landmarken zufällig neuverteilt in der Umgebung. Wir wollen unser Lernverfahren mit zwei naiven Strategien vergleichen: Die sogenannte *gierige Strategie* besteht darin, die ersten N sichtbaren Landmarken zu integrieren. Eine anscheinend bessere Strategie ist die *Äquidistanten Strategie*. Der Roboter fährt jeweils eine gewisse Distanz, bevor er eine neue Landmarke integriert.

Abbildung 3 zeigt die Entwicklung des Fehlers beim Training für zwei Lernstrategien: Die erste berücksichtigt die zwei Merkmale **dist2goal** und **number of landmark**, die zweite alle vier. Man sieht, dass letztere schneller konvergiert und nach 2000 Lernepisoden in einen kleineren Fehler resultiert. Daraus kann man ablesen, dass die Merkmale **phi** und **dist2closest** tatsächlich relevante Zusatzinformationen liefern. Darüberhinaus sieht man, dass die Performanz der gelernten Strategien deutlich besser als die der naiven Strategien ist.

Um die Generalisierbarkeit unseres Lernverfahrens zu evaluieren, trainieren und testen wir unsere Lernverfahren in Umgebungen mit sowohl 50 als auch 100 zufällig positionierten Landmar-

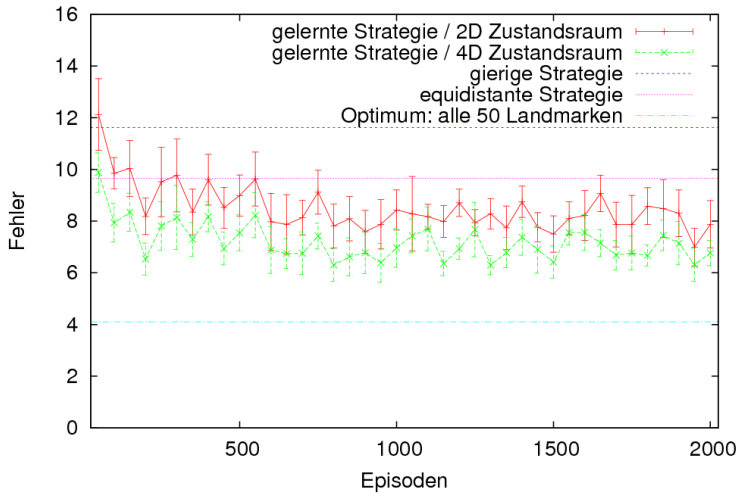

Abbildung 3: **Entwicklung des Fehlers beim Training.** Für beide Lernstrategien wurden jeweils zehn Traingsdurchläufe durchgefährt. Zu sehen ist der gemittelte Fehler – Abstand des Roboters zum Ziel B – und die entsprechende Standardabweichung.

Testszenario	**Gelernte Strategie im Trainingsszenario**						**Äquidistante**
N / M	5 / 50	5 / 100	10 / 50	10 / 100	15 / 50	15 / 100	**Strategie**
5 / 50	**10, 5**	10, 7	12, 2	11, 6	13, 5	12, 7	13, 6
5 / 100	12, 2	**11, 5**	15, 6	14, 9	17, 4	16, 6	13, 2
10 / 50	7, 2	8, 2	**6, 8**	7, 1	7, 2	7, 1	9, 6
10 / 100	6, 3	6, 9	7, 6	**6, 8**	9, 7	8, 4	8, 5
15 / 50	6, 9	8, 0	5, 9	6, 5	**5, 6**	6, 0	7, 5
15 / 100	5, 1	5, 9	5, 1	4, 9	6, 0	**5, 1**	6, 6

Tabelle 1: **Kreuzaufstellung.** Gezeigt sind die gemittelten Fehler über zehn Traingsläufe und 1000 Testepisoden. Alle mit rot gekennzeichneten Strategien sind signifikant besser als die äquidistante Strategie. Dies wurde mit einem t-Test ($\alpha = 2,5\%$) validiert.

ken. Außerdem, verwenden wir UKFs mit einer Kapazität von fünf, zehn und 15 Landmarken. Das hohe Maß an Generalisierbarkeit lässt sich aus Tabelle 1 ablesen. Wird zum Beispiel in einem Szenario mit $M = 50$ und $N = 5$ trainiert, dann liefert die erlernte Strategie in allen sechs Testszenarien signifikant bessere Ergebnisse als die äquidistante Strategie.

8 Ausblick

Momentan untersuche ich Selektionsstrategien für Navigationsaufgaben die komplexer sind als die vorgestellte (z.B. Rundfahrten). Dabei möchte ich mich u.a. auf das Wiederentfernen von Landmarken aus den UKF konzentrieren. In Zukunft möchte ich das Lernverfahren zur Landmarkenselektion auf Anwendungen mit echten Roboter übertragen. Im Gegensatz zur Simulation ist es jedoch bei echten Robotern meist nicht praktikabel, über eine große Anzahl von Episoden zu trainieren. Dieses Problem ließe sich lösen, indem man eine Strategie in einer hinreichend realistischen Simulationsumgebung lernt und die gelernte Strategie dann auf einen echten Roboter

anwendet. Alternativ könnte man auch versuchen, allgemeine Regeln aus den in der Simulation gelernten Strategien zu extrahieren. Ziel meiner Arbeit ist es, das Verfahren auf fliegende Roboter zu erweitern, um Navigationsaufgaben auf einem autonomen Luftschiff durchzuführen.

Meine Arbeit ist mit der Arbeit von Herrn Rottmann vernetzt. Dabei bilden Lernverfahren für eingebettete Systeme die Schnittmenge.

Literatur

[1] R. He, S. Prentice, and N. Roy. "Planning in information space for a quadrotor helicopter in a gps-denied environments". In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2008.

[2] S. J. Julier and U. J. K. "A new extension of the kalman filter to nonlinear systems". In *Proceedings of the International Symposium on Aerospace/Defense Sensing, Simulation and Controls*. 1997.

[3] J. Ko, D. J. Klein, D. Fox, and D. Hähnel. "Gaussian processes and reinforcement learning for identification and control of an autonomous blimp". In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2007.

[4] P. S. Maybeck. "The kalman filter: An introduction to concepts". In *Autonomous Robot Vehicle*. I. J. Cox and G. T. Wilfog, Eds. Springer Verlag, 1990.

[5] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. "Fastslam: A factored solution to the simulaneous localization and mapping problem". In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. Edmonton, Canada, 2002.

[6] A. Y. Ng. "Inverted autonomous helicopter flight via reinforcement learning". In *Proceedings of the International Symposium on Experimental Robotics*. 2004.

[7] 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, October 2007.

[8] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. Cambridge, MA, USA: MIT Press, 1998.

[9] M. Theberge and G. Dudek. "Gone swimmin"'. *IEEE Spectrum*, June 2006.

[10] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.

[11] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. "Simultaneous localization and mapping with sparse extended information filters". Vol. 23, 2006.

[12] S. Zhang, L. Xie, and M. D. Adams. "Entropy based feature selection scheme for real time simultaneous localization and map building". In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. august 2005.