1 Zusammenfassung und Ausblick

1.1 Zusammenfassung

Aufgabe:

Der Roboter erfüllt die Aufgabe der Hindernisvermeidung durch Reinforcement Lernens mittels dem Lernverfahren Q-Lernen.

Sowohl im Simulator als auch auf dem realen Roboter wird der Kontakt mit Hindernissen nach einiger Laufzeit gemieden. Dabei wird Online mittels eines trial-and-error-Verfahrens gelernt. Die Kurve der Lernergebnisse zeigt bei der Auswahl der passenden Parameter stetig nach oben. Das gelernte Verhalten kann anschließend in der Umgebung eingesetzt werden. Das Q-Lernen kann als Algorithmus gut im Rahmen einer Online-Navigation zur Darstellung von Lernverfahren eingesetzt werden.

Ergebnisse:

Das Auswerten der Testergebnisse erbrachte viele Erkenntnisse bezüglich der Wahl der Wertebereiche für die Parameter.

Bei allen drei Welten erzeugten hohe Werte des Discountfaktors γ (s. Kapitel **??**) gute Werte der Zustands-Aktions-Paare. Damit erweist sich ein Aufnehmen von neuen Einflüssen und Erkenntnissen über die Umgebung für den Algorithmus als besonders positiv.

Ein weiteres wesentliches Ergebnis der Tests war die Verwendung des Reinforcements bis zu einem Wertebereich von minus eins. Der für die Entwicklung des Algorithmus negative Wandkontakt wurde mit einem negativen Reinforcement beantwortet, entsprechend der Aufgabe der Diplomarbeit. Alle Auswertungsergebnisse deuten darauf hin, dass dieses Vorgehen auch die beste Lösung zur Hindernisvermeidung im Rahmen des Reinforcements bedeutet.

Entscheidenen Einfluss auf die Auswertungsergebnisse nahm auch der Befehl *wait* im Activity reinforcement, s. Kapitel **??**. Er vollführt eine Wartepause zwischen Ausführung der Aktion und Feststellung des Nachfolgezustandes. Die Wartepause ist ein Vielfaches eines Saphira-Zyklus. Eine niedrige bzw. keine Wartezeit, wie zunächst angedacht, behindert die Entwicklung der Aktionen und Zustände. Es kann nicht sinnvoll navigiert werden. Ein Hochsetzen der Wartezeit ermöglichte dem Roboter die Ausführung seiner Aktionen und die Überführung in neue Zustände, die dann angemessen durch den Algorithmus ausgewertet werden konnten.

Die Temperatur erfüllt ihren Zweck der zufälligen Aktionsauswahl durch Absenken eines Zahlenwertes. Dieses Absenken kann durch die Parameter des Startwertes $start_{tp}$ und der Absenkungsrate τ_{tp} eingestellt werden. Je nach Aufgabe und Umgebung sind andere Werte für die Parameter sinnvoll. Die gleiche Aussage gilt für die Lernrate mit dem Parameter τ_{α}, der Lernratenabsenkung. Allgemein gilt, dass sowohl auf die Explorierung der Zustände als auch auf die Entwicklung der Aktionen wert gelegt werden muss.

Die Parameter Geschwindigkeit und Winkeländerung stellen durch unterschiedliche Auswahl die Aktionen bereit. Ihr Einfluss auf den Algorithmus erwies sich als gering. Bei der Geschwindigkeit muss auf das Vorhandensein einer geringen Stufe geachtet werden, um das Navigieren in engen Umgebungen zu ermöglichen. Die Winkeländerungen müssen der Umwelt angepasst sein. In einer Umgebung, in der selektiv und fein gesteuert werden muss, setzt man auch kleine Winkel mit geringem Niveauunterschied ein.

Einschränkungen:

Die Ergebnisse und die Auswertung hängen auch von einigen Einschränkungen ab.

Die Wahl der Welten war nicht optimal. Das Testen in drei verschiedenen Welten deckt nicht alle möglichen Navigationsfälle ab. Eine Lösung ohne viel Mehraufwand ist aufgrund der Komplexität der Parameter aber nur schwer möglich. Das Testen der verschiedenen Parameterkonstellationen für die nur drei Welten nahm einundachtzig(!) Tests in Anspruch. Ein Automatisierung der Auswertung käme dem Anwender sehr zugute.

Die Strategie zur Auswahl der Nachfolgeaktion ist sehr einfach gehalten. Durch umfangreiche und aufwendigere Verfahren für die Wahl der Nachfolgeaktion (s. Kapitel **??**) wäre die Qualität der Endergebnisse sicher höher gewesen.

Andere Lernverfahren:

Durch Einsatz anderer Lernverfahren können die erzielten Ergebnisse des Q-Lernens u.U. noch verbessert werden.

Neben dem Q-Lernen gibt es eine Reihe weiterer modellfreie Lernalgorithmen, s. Kapitel **??**.
Beim TD-Lernen bietet sich die Möglichkeit, mehrere Schritte als nur den vorhergehenden in die Bewertungen einfliessen zu lassen. Ein Befähigungspfad (engl. eligibility-trace) sorgt für die zeitliche Einordnung der neuen Lernergebnisse.
Ein Vorteil des C-Lernens ist die Einstellung der Explorationsrate. Sie wird lokal nach Erhalt und Auswertung der Sensorinformationen neu bestimmt. Weitere Vorzüge sind vereinfachte Formeln und die vergleichsweise geringe Komplexität durch Wegfall der Zustands-Aktions-Paare gegenüber dem Q-Lernen.
In wie weit das C-Lernen einfacher zu implementieren ist, kann in weiterführenden Arbeiten untersucht werden.

Modellbasierte Lernalgorithmen bieten einen anderen Ansatz, s. Kapitel **??**.
Der Dyna-Algorithmus, ein Vertreter dieser Klasse von Algorithmen, vollzieht einen Schritt in der Realität durch Interaktion mit der Umwelt. Fast zeitgleich wird eine bestimmte Anzahl von virtuellen Schritten durch Interaktion mit dem Modell ausgeführt. Das spart eine große Anzahl von Durchläufen bis zur Zielerreichung durch den Roboter. Nachteilig ist der Einsatz enormer Hardwareressourcen, da die Berechnung der Modelldaten bis zum nächsten Durchlauf abgeschlossen sein muss.
Eine mögliche Lösung stellt der Einsatz in parallelen Rechensystemen dar.

Fazit:

Das Q-Lernen, als Reinforcement Lernalgorithmus, ist in der Sahira-Umgebung mit Anbindung an den Pioneer 2 CE für die Navigation gut geeignet. Die Wahl der Parameter stellt ein wichtiges Kriterium dar, um für die Umgebung auch die passende Navigation bereitzustellen. Durch Einsatz anderer Lernverfahren können die erzielten Lernergebnisse u.U. noch verbessert und die Ansätze der Diplomarbeit weiterverfolgt werden.

1.2 Ausblick

Wie können die erzielten Ergebnisse für die Zukunft genutzt werden?

Der Ansatz der flexiblen Benutzung, der in der Diplomarbeit verfolgt wird, lässt den Raum für Erweiterungen offen.

Durch Erweiterung des Colbert-Activity ist der Weg frei für den Vergleich verschiedener Lernverfahren. Die Saphira-Oberfläche ermöglicht das zeitgleiche Laden von mehreren Bibiotheken. Diese Module mit den verschiedenen Lernverfahren könnten so leicht gewechselt werden und in gleicher Umgebung in Konkurrenz zueinander treten. Jedes Lernverfahren hat seine Vor- und Nachteile. Das eigene Austesten und Anwenden macht Algorithmen begreifbar und weckt Verständnis.

Falls der Studiengang Intelligente Systeme plant den Pioneer mit der Saphira-Software in Online-Navigationen einzusetzen, könnten die Ergebnisse im Rahmen der Diplomarbeit bzw. etwaige Erweiterungen den Teil der Hindernisvermeidung durch Online-Lernen (Reinforcement Lernen) übernehmen.

Anstatt der bloßen Anzeige der Zwischen- bzw. Endwerte, können die Ergebnisse des Algorithmus grafisch aufbereitet werden. Die Tabellenkalkulation MS Excel 2000 leistete sehr gute Dienste im nachträglichen Auswerten und Aufbereiten der Daten. Eine visuelle Online-Auswertung kann diese jedoch nicht ersetzen. Eine Online-Auswertung setzt den Benutzer in die Lage, den Algorithmus schon zur Laufzeit zu beurteilen und ggf. bei schlechter Performance abzubrechen. Dann ist das Absolvieren von Tests mit verschiedenen Parametern ist in einer ansprechenden Zeit möglich. Wie schon erwähnt, können verschiedene Lernverfahren eingesetzt werden. Dazu muss dann das GUI entsprechend ausgebaut werden.

Nicht zuletzt wird mit dieser Arbeit auch der Gedanke verknüpft, dass sich in der Zukunft eine Möglichkeit zum Einsatz in Schule und Lehre ergibt. Der Umfang der Anwendungsgebiete für den Pioneer-Roboter vergrößert sich dadurch. Die Verbindung von theoretischen KI-Lernverfahren und praktischen Einsatz auf Robotern lässt hoffen, dass das Interesse für die KI geweckt wird. Studenten bekommen durch eigenes Erleben und Anwenden einen praktischen Bezug zur Künstlichen Intelligenz.

Dank gilt meinem Betreuer, Prof. Dr. J. Heinsohn, und dem wissenschaftlichen Mitarbeiter Dipl.-Inform. I. Boersch für die immer freundliche und hilfsbereite Unterstützung. Ein herzliches Dankeschön geht besonders an Doreen.

Rene Eggert