

Bewegungsalgorithmen in der Robotik

Seminar über Algorithmen (SoSe 2004)
Tobias Hermes, Raphaela Wrede

Einleitung:

Roboter ...

- Erfüllen Aufgaben, indem sie Bewegungen ausführen, im physikalischen Raum, der sie umgibt
- Sind programmierbar, Software enthält Roboter-Algorithmen

Roboter-Algorithmen (allgemein)

- Unterscheiden sich von traditionellen Algorithmen
- Prozesse finden häufig nicht in streng begrenzten und definierten Umgebungen statt
- Aufgaben/mögliche Eingaben (Daten) sind unterspezifiziert

Grundlegende Prinzipien in der Robotik

- Erstes Ziel eines Roboter-Algorithmus ist es, eine Prozedur zu beschreiben, um eine Teilmenge der realen Welt (Arbeitsumgebung (workspace)) zu kontrollieren, um ein bestimmtes Ziel zu erreichen

Grundlegende Prinzipien in der Robotik

- Roboter-Algorithmen muss so entworfen sein, dass er eine Menge von relevanten Zuständen identifizieren kann, um Abfolge von Operationen auswählen zu können, um die Arbeitsumgebung durch eine Folge von Zuständen zu führen bis Zielzustand erreicht ist
- 3 Probleme: Zustandwahl, Operationsauswahl, Konstruktion von Zustandserkennungsfunktionen

Algorithmen-Planer (planner)

- Algor. muss Sensing, Planning und Acting miteinander verweben
- z.B. muss zur eindeutigen Zustandserkennung „on the fly“ Sensing durchgeführt werden
- Es ist deshalb ein sog. Planer nötig, der einen Steuerungs-Algorithmen generiert und als ein separater Algorithmus. gesehen werden kann

Roboter-Algorithmus

=

**Planungs-Algorithmus
+ Steuerungs-Algorithmus**

Hohe Laufzeiten

- treten auf, weil sehr große Räume betrachtet werden müssen
- deshalb wiegt man „Vollständigkeit“ gegen Laufzeit ab und verzichtet häufig auf Vollständigkeit

Hohe Kosten

- für das Planen, das „on-line“ geschehen muss
- Algorithmus muss sehr vorsichtig Zeit für das Planen und andererseits für das Steuern einteilen
- Zeiteinteilungsproblem bis heute sehr wenig verstanden

Roboter-Algorithmen bedenken Geometrie

- Alle Roboter, egal für welchen Zweck müssen die Geometrie ihrer Umgebung berücksichtigen
- Zusätzlich auch noch Kräfte wie Gravitation, Trägheit, Reibung (bei Kontakten)
- D.h. Konzepte der Mechanik müssen betrachtet werden

Roboter-Algorithmen haben physikalische Komplexität

- Charakterisiert durch Anzahl benötigter Hände, Anzahl Bewegungen oder Blinklichter
- Dies beeinflusst Laufzeit sowie Baukosten/Herstellungskosten
- Falls physikalische Komplexität zu groß ist, kann dies dazu führen, dass Aufgabe redefiniert wird

Aufgabenbereiche für Roboter-Algorithmen

Bearbeitung von Teilen (Part Manipulation)

- Wird sehr oft in der industriellen Robotik benötigt

Das Greifen

- Hauptziel ist es, ein sicheres Greifen von einem Objekt, dessen Modell als Eingabe dient, zu berechnen

Anbringen, Befestigen, Fixieren (Fixturing)

- Die meisten Operationen in der industriellen Fertigung verlangen das Fixieren von Teilen
- Oft werden einstellbare Halterungssystem verwendet (z.B. Lochgitter)

Teilzuführung (Part Feeding)

- Muss sehr schnell erfolgen und mit hoher Zuverlässigkeit
- Basiert meistens auf „nicht-greifende“ Methoden

Montageplanung (Assembly-Sequencing)

- Ziel: sowohl eine Reihenfolge, in der die Teile zusammengefügt werden können, als auch die dafür benötigten Bewegungen müssen berechnet werden
- Damit kann geprüft werden, ob ein best. Produkt leicht herzustellen ist

Wegplanung (Basic Path Planning)

- Bewegungsplanung: Roboter werden mit der Fähigkeit ausgestattet, automatisch zu entscheiden, welche Bewegungen ausgeführt werden müssen
- Bei der grundlegenden Wegplanung geht es darum einen kollisionsfreien Weg für einen Roboter in einem statischen Umfeld zu finden
- Andere Bewegungsplanungsprobleme beschäftigen sich mit sich bewegenden Hindernissen, mehreren Robotern, bewegbaren Objekten und Unsicherheit...

Begriffe:

Konfiguration (Configuration)

- Konfiguration eines Roboters A = Position und Orientierung jedes Teils von A

Konfigurationsraum (Configuration Space) C

- Konfigurationsraum C = Menge aller Konfigurationen von A

Freier Weg (Free Path)

- Weg, der vollständig im freien Raum, d.h., ohne Hindernisse, liegt

Halbfreier Weg (Semi-free Path)

- Weg im Verschluss von freiem Raum

Komplette Algorithmen

- Finden immer Lösung, falls eine existiert und melden es, falls keine existiert
- Benötigen exponentielle Zeit in der Anzahl der Freiheitsgrade

wahrscheinlichkeitstheoretische Algorithmen (Probabilistic Algorithms)

- Vollständigkeit wird gegen Zeit abgewägt
- Falls eine Lösung existiert, wird diese nur mit hoher Wahrscheinlichkeit gefunden
- Benutzen häufig „Randomization“
- Bei einfachen geometrischen Umgebungen kann logarithmische Laufzeit erreicht werden

Wegplanung für nichtsteuerbare Roboter (Path Planning for nonholonomic Robots)

- Anzahl der Parameter, die gesteuert werden, ist kleiner als die, die eine Konfiguration definieren
- Beispiele:
 - car-like Roboter
 - enthalten in Lastzügen, Flugzeugen, Sateliten

Wegplanung für nichtsteuerbare Roboter

- Es ist wenig bekannt über die Wegplanung für nichtsteuerbare Roboter, die nicht lokal steuerbar sind
- Der Algorithmus entscheidet ob ein Weg zwischen zwei Konfigurationen existiert oder nicht

Bewegungsplanung mit Ungewissheit

- Problemformulierung
- Urbild (Preimage)
- schrittweise Planung (One-Step Planning)
- Mehrstufige Planung (Multistep Planning)
- grenzsteinbasierte Planung (Landmark-Based Planning)

Bewegungsplanung mit Ungewissheit

Problemformulierung

- Eingabe:
Startort, an dem sich der Roboter schon bewegt hat
Zielort, an dem der Roboter aufhören soll sich zu bewegen

Bewegungsplanung mit Ungewissheit

Problemformulierung

- **Output:**
Abfolge von Bewegungsbefehlen, falls eine existiert, mit der der Roboter das Ziel vom Start aus erreichen kann.
Jeder Befehl ist beschrieben durch einen Bewegungsvektor und einer Abbruchbedingung.

Bewegungsplanung mit Ungewissheit

Problemformulierung

Der Bewegungsvektor legt das erwünschte Verhalten der Roboter über die Zeit fest. Die Abbruchbedingung ist eine Bool'sche Funktion, abhängig von den Sensormessungen und der Zeit. Ein Plan kann mehrere Verzweigungen haben.

Bewegungsplanung mit Ungewissheit

Urbild (Preimage)

- Ein Urbild, in Abhängigkeit von einem Ziel und einem Befehl, ist ein Ort, von dem aus nach Ausführung des Befehls das Ziel erreicht werden kann
- Ein Ort heißt „backprojection“, wenn von dieser aus der Roboter das Ziel mit Hilfe der Bewegungsabfolge erreichen kann

Bewegungsplanung mit Ungewissheit schrittweise Planung (One-Step Planning)

- eine schrittweise Planung kann erreicht werden, indem man den Startort durch die „backprojection“ eingrenzt

Bewegungsplanung mit Ungewissheit

Mehrstufige Planung (Multistep Planning)

- in einer mehrstufigen Planung setzt sich jeder Schritt aus folgenden drei Quantoren zusammen: Bewegungsrichtung, Zeit, Sensormessungen

Bewegungsplanung mit Ungewissheit

grenzsteinbasierte Planung (Landmark-Based Planning)

- sog. Grenzsteine ermöglichen es, dass ein Roboter präzise seinen Ort bestimmen kann
- Durch diese Grenzsteine ist es für einen Roboter möglich, sich von Grenzstein zu Grenzstein zu bewegen und so sein Ziel zu erreichen
- Ein Weg wird berechnet, indem man vom Ziel rückwärts zum Start sich einen Weg über bekannte Grenzsteine errechnet und diesen dann vom Start aus abgeht

Andere Bewegungsplanungsprobleme

- Dynamischer Arbeitsraum (Dynamic Workspace)
- Einordnung von multiplen Robotern (Coordination of Multiple Robots)
- Handhabungsplanung (Manipulation Planning)
- Optimale Planung (Optimal Planning)

Andere Bewegungsplanungsprobleme

Dynamischer Arbeitsraum (Dynamic Workspace)

- in der Präsenz von Bewegungshindernissen ist es nicht mehr möglich einen statischen Weg zu planen. Ein Weg muss anhängig von der Zeit geplant werden und wird dann Bewegungsablauf genannt

Andere Bewegungsplanungsprobleme

Einordnung von multiplen Robotern (Coordination of Multiple Robots)

- multiple Roboter können trivialerweise adressiert werden, indem man die Bestandteile als einzelne Roboter sieht (Fehler im Handout)

Andere Bewegungsplanungsprobleme

Handhabungsplanung (Manipulation Planning)

- Viele Roboter-Aufgaben beinhalten das Ausführen von Anordnungen von physikalischen Objekten. Diese Objekte werden „bewegliche Objekte“ genannt und können sich nicht selbständig bewegen. Die Planung von diesen beweglichen Objekten wird Handhabungsplanung genannt

Andere Bewegungsplanungsprobleme

Optimale Planung (Optimal Planning)

- **Optimal-Zeit Steuerungsplanung (Optimal-Time Control Planning):**
Die Eingabe ist ein geometrischer freier Weg mit dem Parameter des Weges, der schon seit der Startkonfiguration gegangen worden ist. Das Problem ist es eine Zeitparametrisierung zu finden, die die Reisezeit für den Weg minimiert

Andere Bewegungsplanungsprobleme

Optimale Planung (Optimal Planning)

- Minimal-Zeit Bewegungsablauf Planung (Minimal-Time Trajectory Planning):
Der zuerst geplante geometrisch freie Weg wird iterativ korrigiert um die Reisezeit zu verringern. Jede Iteration benötigt eine Überprüfung des neuen Weges nach Kollisionen und Neuberechnung der Optimal-Zeit Steuerung

Abtastung (Sensing)

- Vorlagenerstellung (Model Building)
- Roboterlokalisierung (Robot Localization)
- Ergänzende Fragen in Bezug auf Abtastung

Abtastung (Sensing)

Vorlagenerstellung (Model Building)

- Die erste Aufgabe eines Roboters in einer unbekanntem Arbeitsumgebung ist die Erstellung eines geometrischen Modells dieser Umgebung. Hierzu benötigt der Roboter eine Reihe von Abtastungsoperationen an unterschiedlichen Stellen. Jede Operation liefert ein Teilmodell. Der Roboter muss versuchen diese Teilmodelle zu einer kompletten Karte seiner Arbeitsumgebung zusammenstellen

Abtastung (Sensing)

Roboterlokalisierung (Robot Localization)

- Ein Roboter muss oft sich selbst in Relation zur Arbeitsumgebung lokalisieren. Hierzu vergleicht er die Abtastungsergebnisse mit der Arbeitsumgebung.
- Eine weitere Methode hierzu ist GPS (Global Positioning System), die sich schon weit verbreitet hat, jedoch nicht in jeder Arbeitsumgebung anwendbar ist.

Abtastung (Sensing)

Roboterlokalisierung (Robot Localization)

Es gibt zwei Probleme bei sensorbasierter Lokalisation:

- statisch: Der Roboter befindet sich in einer beliebigen unbekanntem Konfiguration; das Problem ist nun die Berechnung dieser Konfiguration. Das Problem selbst ist sehr komplex, jedoch ist die Rechenzeit zu vernachlässigen.
- dynamisch: Der Roboter bewegt sich kontinuierlich und muss regelmäßig seine Konfiguration aktualisieren. Hier besteht i.d.R. ein Zeitproblem, da dies in Real-Zeit geschehen muss.

Abtastung (Sensing)

Ergänzende Fragen in Bezug auf Abtastung

- Einerseits ist das Problem, der Platzierung des Sensors von dem aus der Sensor die Region aufzeichnen soll, andererseits besteht das Problem, die Minimalzahl von Sensoren und deren Platzierung zu finden, so dass sie eine Region vollständig erfassen können. Es besteht immer mehr das Interesse, die Form so zu verändern, dass einfachste Sensoren verwendet werden können

Zusammenfassung

Roboter Algorithmen senden Befehle an Bedienteile und Sensoren, um damit einen Teil der Realen Welt, also der Arbeitsumgebung des Roboters, zu steuern



So long ...