

Technik Autonomer Systeme: Multi Agent Systems

Dirk Wollherr

*Lehrstuhl für Steuerungs- und Regelungstechnik
Technische Universität München*

Multi-Agenten / Multi-Roboter-Systeme Anwendungen

- Gefährliche Umgebungen:
 - Bergbau
 - Weltraum
 - Urban Search And Rescue (USAR)
 - Giftmüll Reinigung
- Überwachung
- Sensoren-Netze
- Unterhaltung (RoboCup)
- Service-Roboter (Reinigung, Objekt-Transport)
- UAVs, UGVs ...
- Militärische Anwendungen
- Industrie
- Landwirtschaft



3

Heterogenität



Rocky 8

VxWorks x86

JPL



K9

Linux x86

Ames



Rocky 7

VxWorks ppc

JPL



FIDO

VxWorks x86

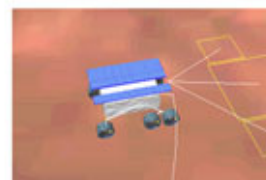
JPL



ATRV

Linux x86

CMU



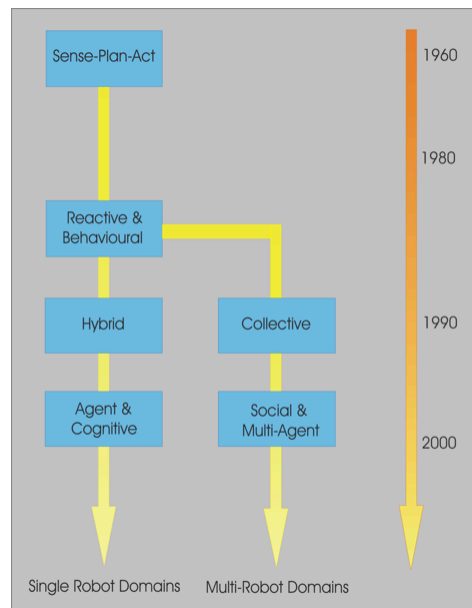
ROAMS

Linux

JPL

4

Multiroboter Architekturen



5

Überblick

- Collective Robotics (gemeinschaftliche Robotik)
 - Schwarm Intelligenz
 - Beispiele: Schwärmen und Futtersuche
- Spieltheorie
- Koordination
- Social Robotics (soziale Robotik)
 - Markt-Basierte Ansätze
 - Multi-Robot Exploration

6

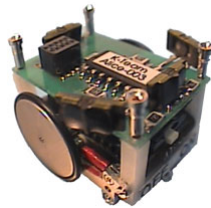
Überblick

- **Collective Robotics (gemeinschaftliche Robotik)**
 - **Schwarm Intelligenz**
 - Beispiele: Schwärmen und Futtersuche
- Spieltheorie
- Koordination
- Social Robotics (soziale Robotik)
 - Markt-Basierte Ansätze
 - Multi-Robot Exploration

7

Collective Robotics

- Idee: Viele einfache reaktive Roboter führen komplexe Aufgaben aus
- Vorteile:
 - Gleichzeitige Wahrnehmungen und Aktionen an mehreren Orten
 - Bessere Leistung durch Aufgaben-Verteilung
 - Kooperation für die Ausführung komplexerer Aufgaben
- Möglich weil Hardware-Komponenten immer
 - kleiner
 - billiger
 - und besser werden

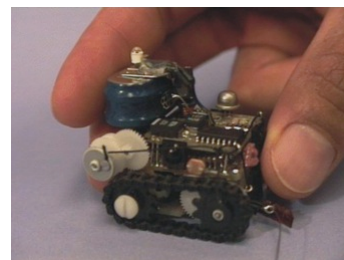


8

Alice (length 2.2 cm)



Khepera (length 7 cm)



Ant MIT (length 3.6 cm)

Schwarm Intelligenz

- Als Vorbilder dienen Insektenkollonien und Schwarmverhalten von Wirbeltieren
- Minimalistische aber voll-autonome Roboter
- Verteilte Steuerung
- Nutzung von Unit-Unit und Unit-Umgebung Interaktionsmöglichkeiten
- Skalierbarkeit (bis zu tausende von Robotern)
- Robustheit (Redundanz)
- Kosten-Effektivität (minimalistisches Design, Massenproduktion von Teilen)



9

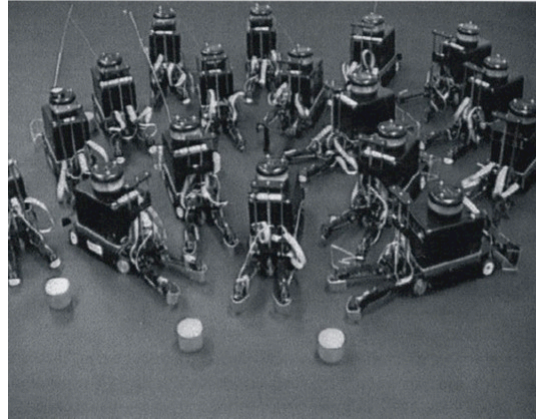
Überblick

- Collective Robotics (gemeinschaftliche Robotik)
 - Schwarm Intelligenz
 - **Beispiele: Schwärmen und Futtersuche**
- Spieltheorie
- Koordination
- Social Robotics (soziale Robotik)
 - Markt-Basierte Ansätze
 - Multi-Robot Exploration

10

Schwärmen und Futtersuche

- Grund-Prinzip
Basis-Verhaltensweisen als
allgemeine Bausteine für den
Aufbau des Gruppenverhaltens
- Basis-Verhaltensweisen:
minimale Menge von
Verhaltensweisen mit
entsprechenden Struktur-
Eigenschaften



TAS - SLAM

Folie 11

Auswahl von Basis-Verhaltensweisen

- Schwierig konkrete Optimalitätskriterien oder „proof of correctness“
Kriterien zu definieren
- Kriterien für die Auswahl:
 - Notwendigkeit: Jedes Verhalten erfüllt ein relevantes Ziel, dass
von keinem anderen Verhalten erfüllt werden kann
 - Sufficiency: Kein weiteres Verhalten wird gebraucht, um die Ziele
zu erreichen
 - Einfachheit
 - Lokalität
 - Stabilität
 - Robustheit
 - Skalierbarkeit

12

Basis-Verhalten für Schwärmen und Futtersuche

- Sicheres wandern: Die Roboter wandern herum, ohne miteinander zu kollidieren
- Folgen: Der Roboter folgt einem anderen Roboter
- Kondensation: Die Fähigkeit der Gruppe sich zu sammeln, so dass ein minimaler Inter-Roboter Abstand erreicht wird
- Ausbreitung: Die Fähigkeit der Gruppe sich auszubreiten, so dass ein maximaler Inter-Roboter Abstand erreicht wird
- Zielsuche: Die Fähigkeit der Gruppe einen bestimmten Ort oder ein Ziel zu finden

13

Sicheres wandern: Algorithmus

- **Agent-Vermeidung:**
 - Whenever an agent is within d_{avoid}
 - ♦ If the nearest agent is on the left
 - Turn right
 - ♦ Else
 - Turn left
- **Hindernis-Vermeidung:**
 - Whenever an obstacle is within d_{avoid}
 - ♦ If obstacle is on right only, turn left
 - ♦ If obstacle is left-only, turn right
 - ♦ After 3 consecutive identical turns, backup and turn
 - ♦ If an obstacle is on both sides, stop and wait
 - ♦ If an obstacle persists on both sides, turn randomly and back up
- **Wandern:**
 - Otherwise move forward by d_{forward} , turn randomly

14

Folgen: Algorithmus

➤ Folgen:

- Whenever an agent is within d_{follow}
 - ♦ If an agent is on the right only, turn right
 - ♦ If an agent is on the left only, turn left



15

Kondensation: Algorithmus

➤ Kondensation:

- Whenever nearest agent is outside $d_{\text{aggregate}}$
 - ♦ Turn towards the local centroid_aggregate, go
- Otherwise, stop



16

Ausbreitung: Algorithmus

➤ Ausbreitung:

- Whenever one or more agents are within $d_{disperse}$
 - ♦ Move away from Centroid_disperse



17

Zielsuche: Algorithmus

➤ Zielsuche:

- Whenever at home
 - ♦ Stop
- Otherwise, turn toward home, go

18

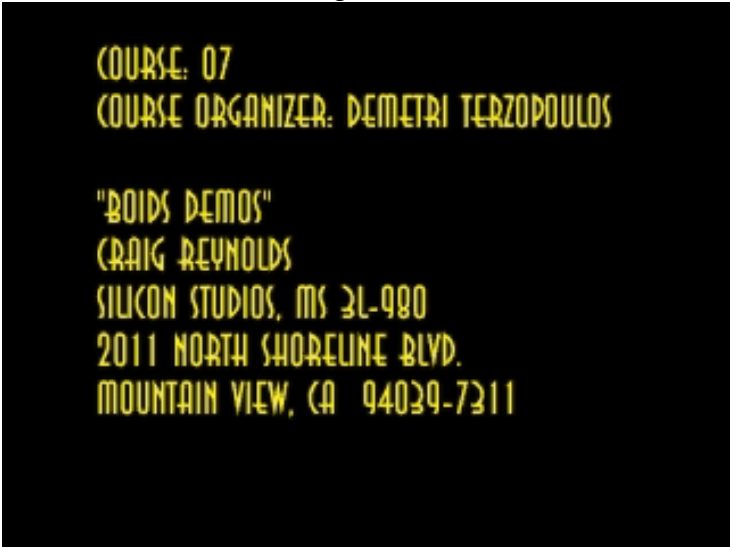
Wahrnehmungsinformationen, Verhaltenswahl

- Wahrnehmungsinformationen sind zu einer Reihe von Eigenschaften kodiert
 - At-home?
 - Have-puck?
 - Crowded?
 - Behind-Agent?
 - Sense-puck?

19

Schwärmen (1)

- **Flock:**
 - Summieren gewichteter Ausgänge von Sicher-Wandern, Kondensation, Ausbreitung, Zielsuche



COURSE: 07
COURSE ORGANIZER: DEMETRI TERZOPOULOS

"BOIDS DEMOS"
CRAIG REYNOLDS
SILICON STUDIOS, MS 3L-980
2011 NORTH SHORELINE BLVD.
MOUNTAIN VIEW, CA 94039-7311

20

Schwärmen (2)

- Welche Roboterfähigkeiten erfordert das Schwärmen?
 - Wahrnehmung benachbarter Roboter
 - ♦ Position, Bewegungsrichtung, Geschwindigkeit
 - Wahrnehmung benachbarter Hindernisse
 - Unterscheidung zwischen Robotern und Hindernissen
- Wie?
 - Bilderkennung
 - Beacons auf Roboter (z.B. Infrarot, usw.)
 - Globaler Lokalisierungsalgorithmus + Funkübertragung von globalen Roboterpositionen

21

Futtersuche (1)

- Futtersuche:
 - Der Roboter sucht nach attraktiven Objekten (Puck)
 - Wenn ein Puck gefunden wurde, bewegt sich der Roboter zum Puck, holt ihn ab und kehrt zu einer Basis zurück
 - Die Roboter wiederholen das Ganze bis alle Pucks zur Basis gebracht wurden
- Benötigte Basis-Verhalten:
 - Ausbreitung
 - Folgen
 - Zielsuche

22

Futtersuche (2)

➤ Futtersuche:

- Whenever crowded, Ausbreitung
- Whenever at-home,
 - ♦ If have-puck, drop-puck
 - ♦ Else Ausbreitung
- Whenever sense-puck
 - ♦ If not have-puck, Zielsuche Puck, pickup-puck
- Whenever behind-agent, Folgen.

23

Nerd Herd (Mataric)



24

Überblick

- Collective Robotics (gemeinschaftliche Robotik)
 - Schwarm Intelligenz
 - Beispiele: Schwärmen und Futtersuche
- **Spieltheorie**
- Koordination
- Social Robotics (soziale Robotik)
 - Markt-Basierte Ansätze
 - Multi-Robot Exploration

25

Spieltheorie – Crash Course

- Die Spieltheorie beschäftigt sich mit dem Multi-Agenten-Entscheidungstreffen unter Unsicherheit und ist basiert auf zwei Voraussetzungen:
 - Die beteiligten Agenten agieren rationell
 - Agenten entscheiden strategisch → unter Berücksichtigung anderer Agenten-Entscheidungen
- Viele Kategorien von Spielen

26

Strategische Spiele

- Ein strategisches Spiel $G = \{N, A_1, \dots, A_n, u^1, \dots, u^n\}$ beinhaltet
 - eine endliche Spielermenge N
 - für jeden Spieler $i \in N$ eine nichtleere Menge A_i von Aktionen/Strategien
 - für jeden Spieler $i \in N$ eine Utility-Funktion oder Auszahlungsfunktion $U_i : A \rightarrow \mathbb{R}$
- Strategische Spiele werden oft in Matrixform angegeben

		Spieler 2	
		L	R
Spieler 1	T	w_1, w_2	x_1, x_2
	B	y_1, y_2	z_1, z_2

- Wählt Spieler 1 T und Spieler 2 wählt L, dann erhält Spieler 1 w_1 und Spieler 2 w_2
- Beispiel: Gefangenendilemma

27

Iterative Eliminierung von streng dominierten Strategien

- Strikt dominierte Strategie: Eine Aktion $\alpha_j^* \in A_j$ im Spiel $\{N, A_1, \dots, A_n, u^1, \dots, u^n\}$ heißt **strikt dominant** gegenüber einer Aktion $\alpha'_j \in A_j$, wenn für alle $\alpha_{-j} \in A$ der anderen Agenten gilt

$$u_j(\alpha_{-j}, \alpha_j^*) > u_j(\alpha_{-j}, \alpha'_j)$$

Ein rationaler Agent wird niemals eine streng dominierte Aktion spielen!

- Iterative Elimination (Iterative Eliminierung)
 - Strikt dominierte Strategien streichen, so lange welche da sind
 - Bleibt nur ein Profil übrig, ist das die Lösung
- Bemerkungen:
 - Nur selten existiert strikte Dominanz
 - Das Ergebnis der iterativen Eliminierung ist bei strikter Dominanz eindeutig (unabhängig von der Reihenfolge der Eliminierung)

28

Nash-Gleichgewichte (1)

- Das meist benutzte Lösungskonzept der Spieltheorie:
Strategiekombination, in der kein Spieler durch Abweichung einen Vorteil erlangen kann

- Definition 1 (Nash-Gleichgewichte):

Ein Nash-Gleichgewicht eines strategischen Spieles $\{N, A_1, \dots, A_n, u^1, \dots, u^n\}$ ist ein Profil $\alpha_j^* \in A_j$ von Aktionen mit der Eigenschaft, dass für alle Spieler $j \in N$ gilt:

$$u_j(\alpha_j^*) = u_j(\alpha_{-j}^*, \alpha_j^*) > u_j(\alpha_{-j}^*, \alpha_j) \quad \forall \alpha_j \in A_j$$

- Definition 2 (Alternative: Nash-Gleichgewicht):

Sei $B_j(\alpha_{-j})$ die Menge von Aktionen $\alpha_i \in A_i$, die die Beste Reaktion auf α_{-i} sind:

$$B_i(\alpha_{-i}) = \{\alpha_i \in A_i \mid u_i(\alpha_{-i}, \alpha_i) \geq u_i(\alpha_{-i}, \alpha_i')\} \quad \forall \alpha_i' \in A_i$$

Ein Nash-Gleichgewicht α^* ist ein Profil mit der Eigenschaft

$$\alpha \in B_i(\alpha_{-i}^*) \quad \forall i \in N$$

29

Nash-Gleichgewichte (2)

- Eine gemischte Strategie eines Agents i ist eine Wahrscheinlichkeits-Verteilung $p_i(\alpha_i)$ auf Aktionen $\alpha_i \in A_i$
- Interpretation von Nash-Gleichgewichten
 1. Resultiert die iterative Eliminierung strikt dominierter Strategien in einer eindeutigen Strategiekombination, so ist dies ein Nash-Gleichgewicht des ursprünglichen Spiels und zudem das Einzige
 2. Ansonsten:
 - Deduktive/rationale Deutung: künstliche Agenten sind hier ideal!
 - Erfahrung/Lernen: Spieler tendieren auf Grund gemachter Erfahrung hin zu Nash-Gleichgewichten
- Probleme:
 - Existiert immer ein Nash-Gleichgewicht? Nein! Ja, wenn Strategien zufällig sind
 - Sind die Nash-Gleichgewichte eindeutig? Nein
 - Sind Nash-Gleichgewichte einfach zu berechnen? Ja, im Matrixfall. Vermutlich nein für zufällige Spiele

30

Überblick

- Collective Robotics (gemeinschaftliche Robotik)
 - Schwarm Intelligenz
 - Beispiele: Schwärmen und Futtersuche
- Spieltheorie
- **Koordination**
- Social Robotics (soziale Robotik)
 - Markt-Basierte Ansätze
 - Multi-Robot Exploration

31

Koordination

- Mehrere Nash-Gleichgewichte existieren: Welche sollen die Agenten auswählen?
- Definition: Koordination ist der Prozess, wobei eine Gruppe von Agenten ein einziges Nash-Gleichgewicht in einem strategischen Spiel auswählt
- Koordinations-Mechanismen für kooperative Agenten
 - Soziale Konventionen
 - Rollen
 - Verhandlungen
- Koordinations-Mechanismen für nicht-kooperative Agenten
 - Voting
 - Auktionen
 - Markt-Mechanismen
 - Contract Nets

32

Überblick

- Collective Robotics (gemeinschaftliche Robotik)
 - Schwarm Intelligenz
 - Beispiele: Schwärmen und Futtersuche
- Spieltheorie
- Koordination
- Social Robotics (soziale Robotik)
 - Markt-Basierte Ansätze
 - Multi-Robot Exploration

33

Social Robotics

- Social Robotics beschäftigt sich mit Roboter-Teams die auf hybriden Architekturen basieren
- Einzelne Roboter sind fähig, andere Roboter zu modellieren und mittels dieser Modellierung Ziele zu erreichen
- Hier analysieren wir Markt-Basierte Ansätze (Market-based approaches)

34

Markt-Basierte Ansätze – Grundidee

- Basiert auf einem ökonomischen Modell
- Jeder Roboter versucht, seinen eigenen Nutzen (Utility) zu maximieren
- Roboter können miteinander Verhandeln und Angebote für Aufgaben abgeben
- Der eigene Nutzen dient dem Allgemeinwohl
- Entscheidungen werden dezentral getroffen, aber die Ergebnisse sind fast optimal

35

Ähnlichkeit mit der Ökonomie

- Roboter sind rationale Agenten
- Manchmal kooperieren die Roboter miteinander, manchmal konkurrieren sie
- Die einzelnen Roboter erhalten für eine Belohnung für richtige Entscheidungen und eine Bestrafung für falsche
- Das Ergebnis muss global effizient sein

36

Markt-Mechanismen

- Wir haben:
 - Ein Roboterteam gebaut, um eine bestimmte Menge von Aufgaben zu erledigen
 - Das Team ist ökonom
 - Das Ziel ist die Aufgaben zu erledigen, so daß die gesamten Kosten minimiert werden
- $Utility = \text{Gewinn} - \text{Kosten}$
- Der Teamgewinn ist die Summe der einzelnen Gewinnen und Teamkosten ist die Summe der einzelnen Kosten
- Kosten und Gewinne sind von der Anwendung abhängig
- Roboter die viel Gewinn mit wenig Kosten erreichen, bekommen einen größeren Anteil des gesamten Nutzens

37

Beispiele

- Die Kostenfunktionen können kompliziert sein
 - basiert auf der zurückgelegten Distanz
 - basiert auf der benötigten Zeit
 - oder eine Funktion der verbrauchten Energie, CPU Perioden usw.
- Der Gewinn ist abhängig von der Aufgabenerfüllung
 - Ein Ziel erreichen
 - Ein Objekt verschieben
 - usw.

38

Wie werden die Preise beschlossen?

- Abgabe von Geboten
 - Roboter verhandeln und drücken gegenseitig die Preise
 - Bemerkung: So erzielen wir bessere Lösungen (vielleicht optimale Lösungen)
- Roboter verhandeln gleichzeitig mehrere Angebote
- Angebote können Multi-Partie sein
- Preise werden durch Angebot und Nachfrage bestimmt

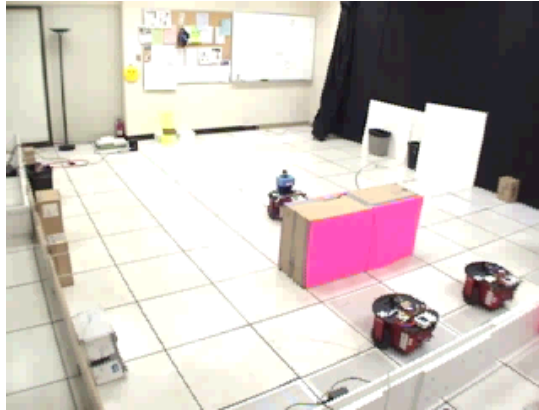
39

Vorteile

- Robust in dynamische Umgebungen
 - keine Hierarchie
 - Wenn ein Roboter ausfällt kann die Aufgabe neu zugeordnet werden
- Verteilte Intelligenz ermöglicht schnelle Aktion
- Nur lokale Kommunikation benötigt
- Effiziente Ressourcenverwendung und Rollen Anpassung

40

Video (1)



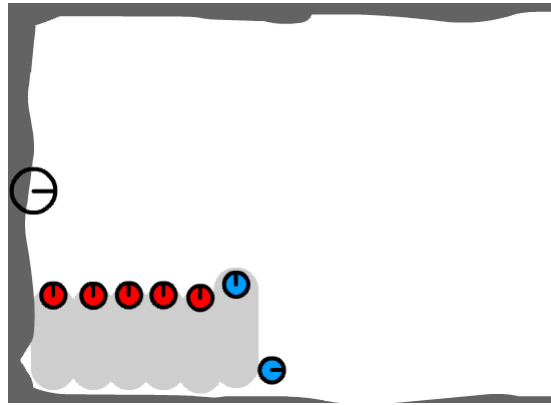
41

Video (2)



42

Video (3)



43

Überblick

- Collective Robotics (gemeinschaftliche Robotik)
 - Schwarm Intelligenz
 - Beispiele: Schwärmen und Futtersuche
- Spieltheorie
- Koordination
- Social Robotics (soziale Robotik)
 - Markt-Basierte Ansätze
 - **Multi-Robot Exploration**

44

Multi-Robot Exploration

- Ziel: Erforschung und Kartenerstellung einer unbekannten Umgebung
- Kommunikation kann unsicher sein
- Mehrere Roboter
 - große Fläche überdecken, so schnell wie möglich
 - Robustheit gegen Roboter ausfall
- Mehrfache Abdeckung vermeiden
- Lösung: Koordination
 - Informationsgewinne maximieren, Kosten minimieren

45

Markt-Ansatz

- Die Welt wird als ein Gitter repräsentiert
 - Die Quadrate sind: unbekannt (0), besetzt (+) oder leer (-)
- Ziel ist es, die Quadrate in dem Gitter durch die Roboter zu erforschen
- Für Zielquadrate in dem Gitter
 - Kosten sind vom Absatz abhängig
 - Gewinn ist von den Informationsgewinnen abhängig
 - $R = (\# \text{ von unbekannten Zellen in der Nähe vom Ziel}) * (\text{Gewichtungsfaktor})$
- Teamgewinn = Summe der einzelnen Gewinne

46

Erforschungsalgorithmus (1)

- Für jeden Roboter:
 1. Ziele bekommen
 2. Ziele „gierig“ einschätzen bezogen auf erwarteten Nutzen (Utility)
 3. Versuch, Ziele an allen erreichbaren Robotern zu versteigern
 - Wenn das Angebot günstig ist
 4. Sobald alle Auktionen abgeschlossen sind, Erforschung des Zieles mit dem höchsten Nutzen
 5. Beim Erreichen des Zieles wird ein neues Ziel zugewiesen
 6. Algorithmus wiederholen bis die Karte komplett ist

47

Erforschungsalgorithmus (2)

- Tatsächlicher Gewinn kann anders sein als erwartet
 - z.B. unvorhergesehene Hindernisse
- Wenn tatsächliche Kosten eine Begrenzung übersteigen, Ziel aufgeben
 - nicht stecken bleiben

48

Beispiel

DARPA SDR Final Experiments
Fort AP Hill, Feb 2004
Autonomous exploration and mapping
Run 5B: Single Breach Point

Andrew Howard, USC RRL