

Seminar - Collective navigation of a multi-robot system in an unknown environment

Gregor Germerodt

Hochschule Trier - Trier University of Applied Sciences

E-Mail: grgr3403@hochschule-trier.de

Themensteller und Betreuer: Prof. Dr. Jürgen Graf

Cognitive Systems and Robotics

Abstract—Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. Test [1]

I. INTRODUCTION

Multi-Agenten-Systeme wie autonome und im Schwarm zusammenarbeitende Roboter, bringen in unbekannten Umgebungen Möglichkeiten wie Kartierung, Erkundung oder Rettungsmissionen [1]. Um den Robotern dabei kollisionsfreie Wege zu gewährleisten, behandelt diese Arbeit kollektive Navigation mit limitierter Sensorik und Kommunikationsfähigkeiten. Zur Robustheit des Systems, trägt das von der Natur abgesehene Schwarmverhalten einen Beitrag, welches hier in drei Regeln definiert wurde: Cohesion, separation and alignment [4]. Neben der globalen Bewegungsplanung, welche dem Roboter eine anfängliche Karte der Umgebung mit allen statischen Objekten bereitstellt, wird hier die lokale Bewegungsplanung behandelt, welche auf eine sich ändernde Umgebung automatisch reagieren kann. Obwohl Simultaneous Localization And Mapping (SLAM) Methoden als herkömmliche lokale Navigationsstrategie erwähnt werden, wird hier stattdessen eine dezentrale, potenzialfeldbasierte Navigation mit lokaler Sensorik und Nachbarschaftskommunikation genutzt. Das hilft Robotern frühzeitig Entscheidungen zu treffen, zur Kollisionsvermeidung und um die Systemgröße flexibel und skalierbar zu halten.

II. THEORETICAL BACKGROUND

Im betrachteten Multi-Roboter-System werden die Roboter als Punktmassen in einer zweidimensionalen Umgebung modelliert. Die Systemdynamik basiert auf potenzialfeldbasierten Kräften, wobei nur die Position und Geschwindigkeit jedes Roboters betrachtet werden. Die Bewegungsdynamik jedes Roboters i wird durch folgende Differentialgleichungen beschrieben:

- Formel (Die Ableitung der Position entspricht der Geschwindigkeit) mit $(p_i, v_i) \in \mathbb{R}^2 \times \mathbb{R}^2$
- Formel (Die Ableitung der Geschwindigkeit entspricht der Steuereingabe) mit $u_i \in \mathbb{R}^2$

In der zeitlich diskreten Umsetzung folgt die Aktualisierung schrittweise über: Formel 2,5 Jeder Roboter kann nur mit

benachbarten Robotern innerhalb eines festen Kommunikationsradius r_c bidirektional interagieren. Formel 3 Um Schwarmverhalten zu erzeugen, werden zwischen zwei Robotern Distanzwerte berechnet, welche Abweichungsenergie, den Grad der Interaktion und Verbundenheit bestimmen ((4) bis (6)). Die Steuereingabe u_i eines Roboters i wird über die Formel Formel 7 berechnet, wobei u_{ia} für Schwarmverhalten, u_{ib} für Hindernisvermeidung und u_{iy} für Navigation sorgt. Das Schwarmverhalten u_{ia} setzt sich aus den Formeln (8) bis (10) zusammen, indem sie Abstands- und Geschwindigkeitsanpassungen zu Nachbarn (8), sowie deren Wirkstärke in Abhängigkeit vom Abstand (9, 10) festlegen. Die Hindernisvermeidung u_{ib} setzt sich aus den Formeln (11) bis (13) zusammen, die definieren, welche Hindernispunkte ein Roboter erkennt (11), wie stark er von ihnen abgestoßen wird (12) und wie diese Abstoßungskraft in Abhängigkeit vom Abstand berechnet wird (13). Die Steuerkraft für die Navigation u_{iy} berechnet sich mit Formel (14) um den Roboter zum Ziel zu führen.

A. The problem statement

Viele bestehende Navigationsansätze für Multi-Roboter-Systeme basieren auf künstlichen Potenzialfeldern, bei denen Roboter von Zielpunkten angezogen und von Hindernissen abgestoßen werden. Ein zentrales Problem dieser Methoden sind jedoch lokale Minima, in denen sich Anziehungs- und Abstoßungskräfte ausgleichen und Roboter feststecken. Besonders in Umgebungen mit konkaven oder labyrinthartigen Strukturen können herkömmliche Verfahren nicht zuverlässig entkommen und alternative Ansätze wie kamerabasierte Hinderniserkennung stoßen unter schwierigen Umweltbedingungen an ihre Grenzen. Für Einzelroboter existieren bereits verschiedene Strategien, um lokale Minima zu überwinden, deren Erfolg hängt jedoch stark von der Hinderniskonfiguration ab. Die Herausforderung für Multi-Roboter-Systeme liegt darin, kollisionsfrei und effizient in unbekannten und komplexen Umgebungen zu navigieren, trotz begrenzter Sensor- und Kommunikationsreichweiten. Hierzu sind kooperative und dezentrale Entscheidungsmechanismen erforderlich, um spontane und sichere Gruppenmanöver zu ermöglichen. Für diese Studie gingen die Autoren von keinem Rauschen (noise) aus.

III. SINGLE ROBOT NAVIGATION

Bevor ein Roboter im Schwarm handeln kann, muss er allein zurechtkommen. Dazu werden in diesem Abschnitt drei Methoden vorgestellt:

- Tangential Navigation, inspiriert von [32],
- Corner avoidance und
- Motion planning at obstacle extremities.

Grundsätzlich möchte sich der Roboter zum desired goal bewegen und achtet währenddessen in einem bestimmten Sensorradius r_{tan} auf Hindernisse. Nimmt er dabei ein Hindernis (repräsentiert durch einen Punkt $p_{i,k}$) wahr, weicht er unter Berücksichtigung der bewegungsrichtungsabhängigen Winkel α_{phai} und β_{tai} und sich daraus ergebenden Rotationswinkel γ_i , die die Ausweichrichtung (Formeln 15 bis 17) und ein virtuelles Ziel (18) bestimmen, parallel zur Hinderniskante, also tangential aus. Über eine angepasste Steuerkraft (19 ähnlich zu 14) wird der Roboter dabei zum virtual goal beschleunigt. Nimmt der Roboter zwei Hindernisse (repräsentiert durch zwei Punkte p_{in} und p_{in90}) wahr, die eine Ecke bilden, wendet er das Corner Avoidance Manöver an. Dabei wird der Rotationswinkel γ_i um einen zusätzlichen Winkel ϵ_{phai} ergänzt, der sich aus den Tangentenrichtungen zu beiden Hindernispunkten relativ zum Roboter berechnet (20, 21). Über Formel 22 wird anschließend ein virtuelles Ziel bestimmt, das den Roboter aus der Ecke herausführt. Wenn der Roboter das Ende eines Hindernisses erreicht hat, führt er eine kreisförmige Bewegung um das Ende herum aus. Dazu merkt er sich den letzten Hindernispunkt, der in seinem Sensorradius registriert wurde. Anhand der Formel (23) und dem Winkel β_i , welcher sich durch die tangential Navigation ergab, berechnet der Roboter in gleichen Abständen virtuelle Ziele (berechnet durch 24 und repräsentiert durch die Punkte P_{iv1} bis p_d) und rotiert um einen festen Winkel δ , bis der Winkel zwischen Bewegungs- und Zielvektor (desired goal) kleiner gleich δ ist (25).

IV. THE PROPOSED NAVIGATION APPROACH FOR A MULTI-ROBOT SYSTEM

Der vorgestellte Ansatz erweitert den Navigationsalgorithmus für einzelne Roboter um eine Kommunikationsschnittstelle für die kollektive Bewegungsplanung mehrerer Roboter. Der Zusammenhalt der Gruppe könnte gefährdet werden, wenn jeder Roboter sein eigenes virtuelles Ziel verfolgt. Daher werden Informationen über virtuelle Ziele und kritische Punkte (wie in Fig. 4 dargestellt) untereinander ausgetauscht. Diese Informationen werden nach Relevanz, Aktualität und dem Prinzip "Erkennung vor Kommunikation" priorisiert (erkannte Hindernisse haben dabei Vorrang vor übermittelten Informationen aus dem Kommunikationsnetzwerk). Das Kommunikationsnetzwerk besteht aus Informationspaketen, die sich in drei Typen gliedern: Orientierung, Endpunkte und Ecken. Orientierung (Fig. 5): Enthält den Winkel Θ_{tai} zum aktuellen Ziel, den erkannten Punkt $p_{i,k}$ auf dem Hindernis, den aktuellen Status des Agenten (Table 1) sowie den Zeitpunkt der Hinderniserkennung. Endpunkte (Fig. 6(a)):

Besteht aus dem zuletzt erkannten Punkt $p_{i,e}$, dem Winkel $w_{i,e}$ zwischen dem Normalenvektor vom Agenten zum Hindernis und der x-Achse des inertialen Koordinatensystems sowie dem Zielwinkel $\Theta_{tai,e}$ im Moment der Endpunkterkennung. Ecken (Fig. 6(b)): Umfasst den Zielwinkel beim Eintritt ($\Theta_{tai,ent}$) und Austritt ($\Theta_{tai,ex}$) aus der Ecke sowie den Eckpunkt $p_{i,c}$, der als Schnittpunkt zweier Linien berechnet wird. Bevor ein Roboter eine Aktion anhand der gegebenen Orientierungsinformationen aus dem Kommunikationsnetzwerk ausführt, berechnet er eine Relevanz-Funktion rel , welche im Intervall $[-\infty, 10]$ liegt. Dabei steht rel gleich 10 für eine sehr relevante und rel kleiner gleich 0 für eine zu ignorierende Information. Unterschieden wird dabei wie folgt:

- Alter der Information: Die Formel kleinerFormel 26 einfügen größer bestimmt je älter die Information ist, desto unwichtiger ist sie, mit t_k gleich aktuelle Zeit, t_c gleich Registrierungszeit der Information von einem anderen Roboter und dt element R gleich eine Konstante, die die Dauer beschreibt, worin Information positive Relevanz haben kann
- Distanz vom Hindernis: Sollte das Hindernis in Bewegungsrichtung liegen, so ermittelt sich $reldist$ mit kleinerFormel 27 einfügen größer, mit dx element R die maximale Distanz, die durch $p_{i,pipe}$ minus p_i $pipe$ für eine positive Relevanz erforderlich ist. Liegt es dahinter dann mit kleinerFormel 28 einfügen größer und sollten keine Orientierungsinformationen gegeben sein so ist $reldist$ gleich 0
- Relevanz der Orientierung basierend auf dem vorherigen Zeitschritt: Jeder Roboter erwartet für jeden Zeitschritt nur kleine Änderungen in der tangentialen Navigation. Die Formel kleinerFormel 29 einfügen größer beschreibt die Relevanz der erwarteten Orientierung, wobei $pipe\Theta_{tai}(t_k)$ minus $\Theta_{tai,pipe}$ der Vergleich mit der Zielorientierung ($\Theta_{tai}(t_k)$) mit der neuen Orientierung (Θ_{tai}) und $d\Theta_{tai}$ die maximal erlaubte Winkeldifferenz für eine positive Relevanz ist.
- Auswertung des Senders: rel weist empfangenen Informationen eine hohe Relevanz zu, wenn sie vom Sender stammen und nicht nur von anderen sendenden Robotern weitergereicht wurden. kleinerFormel 30 einfügen größer
- Auswertung anhand des Status: In Table 1 wurden verschiedene Status vorgestellt, kleinerFormel 31 einfügen größer weist ihnen eine Relevanz zu.

Die endgültige Relevanz ergibt sich aus der Formel kleinerFormel 32 einfügen größer, wobei die maximale Relevanz durch kleinerFormel 33 einfügen größer berechnet wird. Ein Informationspaket ist für einen Roboter schließlich relevant, wenn die Bedingung kleinerFormel 34 einfügen größer herrscht und passt daraufhin seine Bewegungsrichtung für den nächsten Zeitschritt an. Sollten mehrere Informationspakete die Bedingung 34 erfüllen, wird der Mittelwert daraus bestimmt. Weiterhin prüft ein Roboter seine Distanz zu den übermittelten Hindernispunkten mit Formel 35 und sieht diese Punkte dann als seine eigenen Hindernispunkte an, sollte sie

in Rahmen der maximalen Distanz R_{rel} element R liegen.

V. COLLECTIVE NAVIGATION USING SHARED INFORMATION

In diesem Abschnitt wird erklärt, wie die Verfahren eines einzelnen Roboters auf ein Multi-Roboter-System adaptiert werden können. Dazu sind 7 Status definiert, in dem sich ein Roboter befinden kann, wobei sich Jeder zum Start im Status 0 befindet (Erklärungen und Abhängigkeiten in Appendix A).

A. Collective tangential navigation

Die kollektive tangentielle Navigation umfasst die Status 1, 4 und 5. Erkennt ein Roboter ein Hindernis innerhalb seiner Wahrnehmungsreichweite r_{tan} , so leitet er, wie beim Einzelroboter, die tangentielle Navigation ein. Diese Aktion wird als Status 1 bezeichnet. Die dabei ermittelten Informationen werden an benachbarte Roboter weitergeleitet (Algorithmus 2). Da die Steuergröße u_{alpha} gemäß Gleichung (8) dazu führen kann, dass sich Roboter durch Abstoßung senkrecht zum Hindernis voneinander entfernen und so unbeabsichtigt ein Wechsel von Status 0 auf 1 ausgelöst werden könnte, wird eine Ignoranz-Bedingung (Bedingung 36) eingeführt. Dadurch können Roboter Hindernisse ignorieren. Wird ein Roboter durch neue Informationen aus dem Kommunikationsnetzwerk in eine ungünstige Ausrichtung gebracht und erfüllt dabei Bedingung (37), so wechselt er in den Status 4 (Algorithmus 5), um seine Ausrichtung zu korrigieren. Berechnet ein Roboter aufgrund empfangener Informationen ein virtuelles Ziel mittels Formel (38), befindet er sich im Status 5 (Algorithmus 6). Um zu verhindern, dass Roboter Hindernisendpunkte zu spät erkennen, wenn diese nur von anderen benachbarten Robotern erfasst wurden, wird eine Distanzabschätzung mit einer Fallunterscheidung (Bedingung 39) vorgenommen. Damit kann der Roboter anhand der Entfernung zum Hindernis oder zu einer projizierten Position seine Zielausrichtung rechtzeitig anpassen und so Verzögerungen im Bewegungsablauf vermeiden.

B. Collective corner avoidance

Nähert sich ein Roboter einer Ecke, kann er sie, wie in Abschnitt 3.2 beschrieben, umfahren und dabei zusätzlich Informationen aus dem Kommunikationsnetzwerk nutzen. Diese Aktion wird als Status 3 bezeichnet. Damit dabei keine möglichen Durchgänge (z. B. zwischen zwei Hindernissen) übersehen werden, prüft der Roboter zuvor die Bedingung 40. Nur wenn diese erfüllt ist, wird die Ecke als solche erkannt. Anschließend definiert der Roboter ein neues virtuelles Ziel und wechselt in Status 4, um sich neu auszurichten. Gleichzeitig sendet er Informationen über die erkannte Ecke an die benachbarten Roboter. Werden die Bedingungen (40) jedoch nicht erfüllt, verbleibt der Roboter zunächst im Wartezustand (Status 6). In diesem Zustand überwacht er das Kommunikationsnetzwerk seiner Nachbarn. Wird dabei eine Information mit Status 3 oder 4 als relevant eingestuft, übernimmt der Roboter sie und verlässt den Wartezustand. Gibt es stattdessen relevante Informationen mit Status 1 oder 2, passt er sich entsprechend an und folgt entweder der Tangentialnavigation oder dem Manöver an einem Hindernisendpunkt.

REFERENCES

- [1] E. Olcay, F. Schuhmann, and B. Lohmann, "Collective navigation of a multi-robot system in an unknown environment," *Robotics and Autonomous Systems*, vol. 132, p. 103604, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889020304449>