

# 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-Roboter systems, such as autonomous robots collaborating in a swarm, offer applications in unknown environments like mapping, exploration, or rescue missions [1]. To ensure collision-free paths, this work addresses collective navigation under limited sensor and communication ranges. To enhance system robustness, a nature-inspired swarm behavior is applied, based on three simple rules: cohesion, separation, and alignment [4]. While global motion planning assumes an initial map with static obstacles, this work focuses on local motion planning that can dynamically respond to environmental changes. Instead of Simultaneous Localization And Mapping (SLAM), a decentralized, potential field-based navigation approach (virtual force field in which robots are attracted or repelled) is used, relying solely on local sensor data and communication with neighboring robots. This enables the robots to make early decisions to avoid collisions while ensuring the system's scalability and flexibility.

## II. THEORETICAL BACKGROUND

Im betrachteten Multi-Roboter-System werden die Roboter als Punktmassen (d.h. ohne räumliche Ausdehnung, nur durch Position und Massepunkt) in einer zweidimensionalen Umgebung modelliert. Die Systemdynamik basiert auf potenzialfeldbasierten Kräften, wobei nur die Position und Geschwindigkeit jedes Roboters  $i$  betrachtet werden,  $(p_i, v_i) \in \mathbb{R}^2 \times \mathbb{R}^2$ . Die Bewegungsdynamik jedes Roboters  $i$  wird durch folgende Differentialgleichungen beschrieben:

$$\dot{p}_i = v_i \quad (1)$$

$$\dot{v}_i = u_i \quad (2)$$

Die Position des Roboters ändert sich also mit seiner Geschwindigkeit (1) und diese wiederum durch eine Steuerkraft (2). Jeder Roboter kann nur mit benachbarten Robotern innerhalb eines festen Kommunikationsradius  $r_c$  bidirektional interagieren. Die zeitabhängige und ungerichtete Nachbarschaft ist definiert durch

$$\mathcal{N}_i^\alpha = \{j \in \mathcal{V} \mid \|p_j - p_i\| < r_c\} \quad (3)$$

mit  $\mathcal{V} = \{1, \dots, N\}$ ,  $N \in \mathbb{N}$ , die Menge aller Roboter. Um Schwarmverhalten zu erzeugen, werden zwischen zwei Robotern Distanzwerte berechnet, welche Abweichungsenergie, den Grad der Interaktion und Verbundenheit bestimmen ((4) bis (6)).

Die Steuerkraft  $u_i$  eines Roboters  $i$  wird über die Gleichung

$$u_i = u_i^\alpha + u_i^\beta + u_i^\gamma \quad (4)$$

berechnet, wobei  $u_i^\alpha$  für Schwarmverhalten,  $u_i^\beta$  für Hindernisvermeidung und  $u_i^\gamma$  für Navigation sorgt. Das Schwarmverhalten  $u_i^\alpha$  setzt sich aus Abstands- und Geschwindigkeitsanpassungen zu Nachbarn (8) sowie deren Wirkstärke in Abhängigkeit vom Abstand ((9) bis (10)) zusammen.

Die Hindernisvermeidung  $u_i^\beta$  wird definiert durch die Hindernispunkte die 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  $u_i^\gamma$  sorgt für eine Anziehung des Roboters zum Ziel.

## 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. Verfahren für einzelne Roboter, wie kamera- oder eliminationsbasierte Ansätze stoßen bei schwierigen Umweltbedingungen oder bei Hinderniskonfigurationen an ihre Grenzen. Weiterhin ist die Herausforderung für Multi-Roboter-Systeme, kollisionsfrei und effizient in unbekannten und komplexen Umgebungen zu navigieren, trotz begrenzter Sensor- und Kommunikationsreichweiten. Aus diesem Grund, befasst sich diese Studie mit Kommunikation und Bewegungsplanung mehrerer Roboter, wobei Rausch-Effekte, Zeitverzögerung oder Paketverlust nicht berücksichtigt werden.

## 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 ein Roboter zu einem festgelegtem Ziel bewegen und achtet währenddessen in einem bestimmten Sensorradius  $r_{\text{tan}}$  auf Hindernisse. Nimmt er dabei ein Hindernis (repräsentiert durch den Punkt  $\hat{p}_{i,k}$ ) wahr, weicht er parallel zur Hinderniskante, also tangential aus. Dies geschieht unter Berücksichtigung der bewegungsrichtungsabhängigen Winkel  $\alpha_i$  und  $\beta_i$  und sich daraus ergebenden Rotationswinkel  $\gamma_i$ . Diese Winkel bestimmen die Ausweichrichtung ((15) bis (17)) und ein virtuelles Ziel (18). Über eine angepasste Steuerkraft ((19) ähnlich zu (14)) wird der Roboter dabei zum virtuellen Ziel  $p_i^v$  beschleunigt.

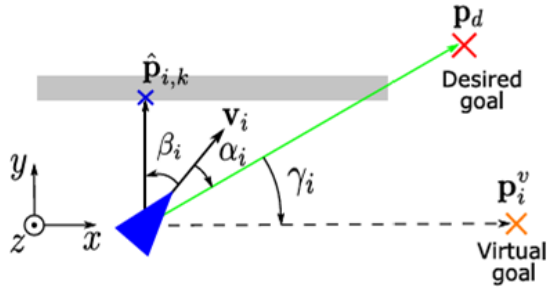


Fig. 1. tangential navigation

Nimmt der Roboter zwei Hindernisse (repräsentiert durch zwei Punkte  $\hat{p}_{i,n}$  und  $\hat{p}_{i,n90}$ ) wahr, die eine Ecke bilden, wendet er das *Corner Avoidance* Manöver an. Dabei wird der Rotationswinkel  $\gamma_i$  um einen zusätzlichen Winkel  $\varepsilon_i$  ergänzt, der sich aus den Tangentenrichtungen  $n$  und  $n_{90}$  zu beiden Hindernispunkten relativ zum Roboter berechnet ((20) bis (21)). Anschließend wird wieder ein virtuelles Ziel  $p_i^v$  bestimmt, das den Roboter aus der Ecke herausführt (22).

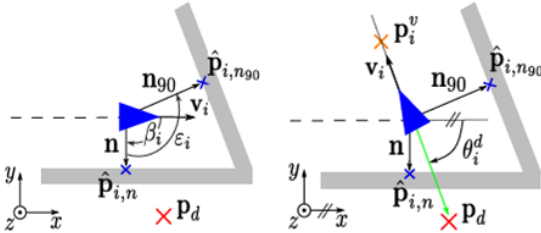


Fig. 2. corner avoidance maneuver

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  $\hat{p}_{i,e}$ , der in seinem Sensorradius registriert wurde. Anhand Gleichung (23) und dem Winkel  $\beta_i$ , welcher sich durch die tangentiale Navigation ergab, berechnet der Roboter in gleichen Abständen virtuelle Ziele (berechnet durch Gleichung (24) und repräsentiert durch die Punkte  $p_i^{v1}$ ,  $p_i^{v2}$  und  $p_d$ ) und rotiert um einen festen Winkel  $\delta$ , bis der Winkel zwischen Bewegungsvektor und virtuelles Ziel kleiner oder gleich  $\delta$  ist (25).

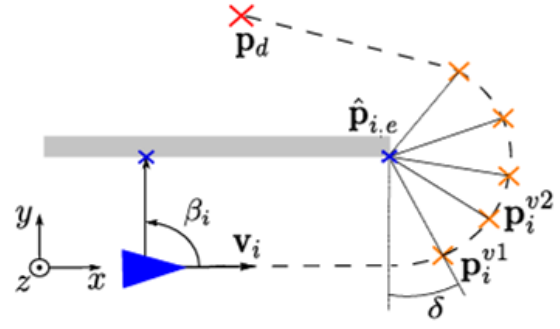


Fig. 3. circular motion at obstacle ending

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

Der nun 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.

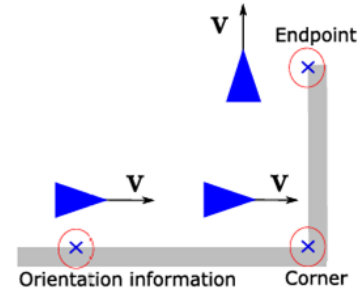


Fig. 4. circular motion at obstacle ending

**Orientierung (Fig. 5):** Enthält den Winkel  $\theta_i$  zum aktuellen Ziel, den erkannten Punkt  $\hat{p}_{i,k}$  auf dem Hindernis, den aktuellen Status des Roboters (alle Status in Table 1 erklärt) sowie den Zeitpunkt der Hinderniserkennung.

**Endpunkte (Fig. 6(a)):** Besteht aus dem zuletzt erkannten Punkt  $\hat{p}_{i,e}$ , dem Winkel  $\omega_{i,e}$  zwischen dem Normalenvektor vom Roboter zum Hindernis und der x-Achse des inertialen Koordinatensystems<sup>1</sup> sowie dem Zielwinkel  $\theta_{i,e}$  im Moment der Endpunkterkennung.

**Ecken (Fig. 6(b)):** Umfasst den Zielwinkel beim Eintritt ( $\theta_{i,ent}$ ) und Austritt ( $\theta_{i,ex}$ ) aus der Ecke sowie den Eckpunkt

<sup>1</sup>unbeschleunigtes und dem Trägheitsgesetz unterliegendes Koordinatensystem

$\hat{p}_{i,c}$ , der als Schnittpunkt aus den erkannten Hindernislinien berechnet wird.

Bevor ein Roboter eine Aktion anhand der gegebenen Orientierungsinformationen aus dem Kommunikationsnetzwerk ausführt, berechnet dessen Relevanz über die Funktion  $\text{rel}$ , welche im Intervall  $] -\infty, 10]$  liegt. Dabei steht  $\text{rel} = 10$  für eine sehr relevante und  $\text{rel} \leq 0$  für eine zu ignorierende Information. Unterschieden wird dabei wie folgt:

- **Alter der Information:** Die Gleichung

$$\text{rel}_t = 10 - \frac{10 \cdot (t_k - \hat{t}^c)}{d_t} \quad (5)$$

bestimmt: je älter die Information, desto unwichtiger ist sie, mit der aktuellen Zeit  $t_k$  und der Registrierungszeit  $\hat{t}^c$  der Information von einem anderen Roboter und  $d_t \in \mathbb{R}$  eine Konstante, die  $\text{rel}_t$  ab einer bestimmten Zeit negativ werden lässt.

- **Distanz vom Hindernis:** Sollte das Hindernis in Bewegungsrichtung liegen, so ermittelt sich die Relevanz der Distanz mit

$$\text{rel}_{\text{dist}} = 10 - \frac{10 \cdot \|\hat{p}^c - p_i\|}{d_x} \quad (6)$$

mit  $d_x \in \mathbb{R}$  die maximale Distanz und  $\|\hat{p}^c - p_i\|$  der Abstand des Roboters ( $p_i$ ) zum Hindernis ( $\hat{p}^c$ ). Liegt es dahinter, dann mit

$$\text{rel}_{\text{dist}} = -\frac{10 \cdot \|\hat{p}^c - p_i\|}{d_x} \quad (7)$$

und sollten keine Orientierungsinformationen gegeben sein, so ist  $\text{rel}_{\text{dist}} = 0$ .

- **Relevanz der Orientierung basierend auf dem vorherigen Zeitschritt:** Jeder Roboter erwartet für jeden Zeitschritt nur kleine Änderungen in der tangentialen Navigation. Die Gleichung

$$\text{rel}_{\text{exp}} = 10 - \frac{10 \cdot |\theta_i(t_k) - \theta^c|}{d_\theta} \quad (8)$$

beschreibt die Relevanz der erwarteten Orientierung, wobei  $|\theta_i(t_k) - \theta^c|$  der Vergleich mit der aktuellen Zielorientierung ( $\theta_i(t_k)$ ) und der neuen Orientierung ( $\theta^c$ ) ist und  $d_\theta$  die maximal erlaubte Winkeldifferenz, sodass  $\text{rel}_{\text{exp}} > 0$ .

- **Auswertung des Senders:** Empfangenen Informationen sind relevanter, wenn der Sender der Besitzer dieser Informationen ist.

$$\text{rel}_o = \begin{cases} 10, & \text{wenn direkt gesendet} \\ 0, & \text{wenn nur weitergeleitet} \end{cases} \quad (9)$$

- **Auswertung anhand des Status:** Aktionen die den Roboter neu ausrichten, werden mit höchster Relevanz bewertet. Alle *status* sind in Tabelle 1 nachsehbar.

$$\text{rel}_{\text{type}} = \begin{cases} 10, & \text{if } \text{status}^c = 4 \vee \text{status}^c = 3 \\ 5, & \text{if } \text{status}^c = 1 \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

Die endgültige Relevanz ergibt sich aus der Gleichung

$$\overline{\text{rel}}_n = \frac{c_{\text{type}} \cdot \text{rel}_{\text{type}} + c_o \cdot \text{rel}_o + c_{\text{exp}} \cdot \text{rel}_{\text{exp}} + c_{\text{dist}} \cdot \text{rel}_{\text{dist}} + c_t \cdot \text{rel}_t}{c_{\text{type}} + c_o + c_{\text{exp}} + c_{\text{dist}} + c_t} \quad (11)$$

wobei die maximale Relevanz durch

$$\text{rel}_{\text{max}} = \arg\max_{\text{rel}_n \in \mathcal{R}_i}(\text{rel}_n) \quad (12)$$

berechnet wird, mit  $n \in \mathcal{S}_i$ , die Menge aller aus dem Netzwerk erhaltene Informationspakete und  $\mathcal{R}_i$  die Menge aller aus den Informationspaketen entnommenen Relevanzwerte für Roboter  $i$ .

Ein Informationspaket ist für einen Roboter schließlich relevant, wenn die Bedingung

$$\text{rel}_n \geq 0.95 \cdot \text{rel}_{\text{max}} \quad (13)$$

erfüllt ist. Er passt daraufhin seine Bewegungsrichtung für den nächsten Zeitschritt an. Sollten mehrere Informationspakete diese Bedingung erfüllen, wird der Mittelwert daraus bestimmt.

Zu übermittelten Ecken oder Endpunkten, prüft der Roboter seine Distanz  $d_s$  mit

$$d_s \leq \|p_i - \hat{p}_c^c\| \leq R_{\text{rel}}, \quad (14)$$

$$d_s \leq \|p_i - \hat{p}_e^c\| \leq R_{\text{rel}}, \quad (15)$$

und sieht diese Punkte dann als seine eigenen Hindernispunkte an, sollten sie innerhalb der maximalen Distanz  $R_{\text{rel}} \in \mathbb{R}$  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).

### Collective tangential navigation

Die kollektive tangential Navigation umfasst die Status 1, 4 und 5. Erkennt ein Roboter ein Hindernis innerhalb seiner Wahrnehmungsreichweite  $r_{\text{tan}}$ , so leitet er, wie beim Einzelroboter, die tangential Navigation ein. Diese Aktion wird als Status 1 bezeichnet. Die dabei ermittelten Informationen werden an benachbarte Roboter weitergeleitet (Algorithmus 2). Da die Steuergröße  $u_i^\alpha$  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 Gleichung 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.

#### *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.

#### *Collective motion at obstacle extremities*

Erkennt ein Roboter den Endpunkt eines Hindernisses, folgt er virtuellen Zielpositionen auf einem kreisförmigen Pfad (Status 2), wie in Abschnitt 3.3 beschrieben. Jeder Roboter legt individuelle Ziele auf diesem Pfad fest und teilt Informationen zum Endpunkt in seinem Netzwerk. Nach Erhalt der Endpunktdaten prüft ein Roboter seine Position durch orthogonale Projektion auf eine virtuelle Startlinie (Fig. 11(a)), bevor er die Kreisbewegung beginnt. Sollte der Roboter diese Startlinie noch nicht erreicht haben (Bedingung 42), setzt er weitere tangentielle Ziele und passt seine Geschwindigkeit an die Mindestgeschwindigkeit für die Kreisbewegung an (Gleichung 43 und 44). Erreicht er die Startlinie, beginnt die Kreisbewegung entsprechend des in Abschnitt 3.3 vorgestellten Schemas. Während der Bewegung stellt eine Komponente aus Gleichung 8 sicher, dass der Roboter sein Ziel auf dem kreisförmigen Pfad erreicht. Überschreitet ein Roboter sein virtuelles Ziel (geprüft über eine Winkelbedingung), wird eine neue Zielposition berechnet. Die Roboter bewegen sich auf individuellen Kreisbahnen mit unterschiedlichen Radien, wodurch innere Kreise kürzere Wege bedeuten. Der Geschwindigkeitskonsens basiert bei der Kreisbewegung auf gleichen Winkelgeschwindigkeiten. Dafür wird eine minimale Winkelgeschwindigkeit (45) festgelegt und die Steuergröße aus Gleichung 8 für eine angepasste Steuergröße aus Gleichung 46 und 47 ersetzt. Dadurch können die Roboter gleichförmig das Hindernisende umkreisen.

#### *Conditional Braking*

Wenn mehrere Roboter einer Gruppe eine kreisförmige Bewegung um ein Hindernis ausführen, können durch unterschiedliche Zeitpunkte beim Verlassen der Kreisbahn Geschwindigkeitsunterschiede entstehen, die zum Auseinanderreißen der Gruppe führen. Um dies zu verhindern, werden Roboter, die das Hindernis bereits passiert haben, abgebremst. Dieses Abbremsen bleibt solange aktiv, bis der letzte Roboter der Gruppe die Kreisbahn verlassen hat.

Zur Bestimmung dieses letzten Roboters wird ein neues Koordinatensystem definiert, dessen Ursprung im Endpunkt des Hindernisses liegt. In diesem System wird die Position jedes Roboters auf die x-Achse projiziert. Anhand dieser projizierten Positionen und über Gleichung 53 betrachten die Roboter ihre Nachbarschaft und identifizieren den Roboter, der am weitesten hinten liegt.

Über eine Gossip-ähnliche<sup>2</sup> Kommunikation wird die gesamte Gruppe über diesen Kandidaten informiert, bis alle Roboter den globalen hintersten Roboter ermittelt haben. Sobald dieser Roboter seine Kreisbewegung beendet hat, sendet er ein Reset-Signal, um das Bremsen der restlichen Roboter aufzuheben und die Gruppenbewegung wieder zu synchronisieren.

#### *Watchdog timer*

Fällt die Geschwindigkeit eines Roboters unter einen definierten Schwellenwert, aktiviert er einen selbstüberwachenden Mechanismus, den *Watchdog Timer*. Dabei speichert der Roboter seine aktuelle Position und den aktuellen Zeitpunkt. Der Timer wird wieder deaktiviert, sobald sich der Roboter innerhalb der folgenden Zeitschritte um mehr als einen festgelegten Schwellenwert bewegt (Roboter nimmt an einen möglichen Deadlock (ausweglose Situation) verlassen zu haben). Bewegt sich der Roboter jedoch über einen Zeitraum von 15 Sekunden kaum und wurde der Timer nicht deaktiviert, wechselt er automatisch in Status 0 und bewegt sich in Richtung *desired goal*. Diese Maßnahme kann zwar zu einer Fragmentierung der Gruppe führen, stellt jedoch sicher, dass jeder Roboter das *desired goal* erreicht.

## VI. SIMULATION RESULTS

Die Simulationsergebnisse werden anhand von zwei Szenarien dargestellt: einem mit einem Zickzack-Hindernis und einem mit einem Korridor mit Hindernissen. In beiden Fällen werden zwölf Roboter zufällig innerhalb eines Startbereichs positioniert, jeweils ohne Anfangsgeschwindigkeit. Zusätzlich wird ein gewünschtes Ziel (als rotes Kreuz markiert) definiert.

Die Roboter werden in den Abbildungen als schwarze Dreiecke dargestellt. Ihre Bewegung wird durch eine bunte Linie hinter den Dreiecken visualisiert, die Bewegungsrichtung entspricht der Ausrichtung der Dreiecksspitze. Die schwarzen Linien zwischen den Robotern zeigen das aktive Kommunikationsnetzwerk an. Die Abbildungen geben Momentaufnahmen

<sup>2</sup>lokaler, dezentraler und iterativer Informationsaustausch

zu bestimmten Zeitpunkten bzw. bei relevanten Ereignissen während der Simulation wieder.

Im ersten Szenario (Abbildung 13) zeigen sich folgende Abläufe:

- (a): Erkennung eines Hindernisses und Beginn der tangentialen Navigation
- (b): Erkennung eines Eckpunktes und Ausführung des *corner avoidance maneuvers*
- (c) und (d): Erkennung eines Endpunktes und Einleitung der kreisförmigen Bewegung um das Hindernis
- (e): Annäherung der Roboter an das Hindernis zur Optimierung der Navigation mithilfe von Gleichung 39
- (f): Erreichen des gewünschten Zielpunkts

Im zweiten Szenario (Abbildung 15) kommen dieselben Verfahren zum Einsatz. Der Unterschied liegt lediglich darin, dass die Roboter hier durch einen Korridor mit Hindernissen navigieren müssen.

#### *Guideline for parameter choice*

Die Parameter der Relevanzfunktion (Gleichung 32) sollten in Abhängigkeit des jeweiligen Szenarios gewählt werden. Ein globales Verhalten des Multi-Roboter-Systems, bei dem alle Roboter nahezu gleichzeitig identische Aktionen ausführen, ist dabei für kleinere Systeme mit sechs bis zwölf Robotern geeignet, wenn deren Verteilung im Verhältnis zu den Hindernissen gering ist. In diesem Fall sollten die Relevanz der Zeit des Informationspakets (*relt*) und der Relevanz des Aktionsstatus (*reltype*) höher gewichtet werden als die übrigen Werte, damit Roboter gleichartig und möglichst synchron auf aktuelle Ereignisse im Kommunikationsnetzwerk reagieren.

Bei steigender Anzahl von Robotern oder einer dichteren Hinderniskonstellations kann ein solches globales Verhalten jedoch problematisch sein, da einzelne Roboter gleichzeitig mehrere relevante Informationspakete erhalten könnten. In diesen Situationen sollten Roboter stärker auf ihre eigene Position und lokale Informationen achten. Dazu werden die Relevanz des Abstands zum Hindernis (*reldist*), die Relevanz der erwarteten Orientierung für die nächste Aktion (*relexp*) sowie die Relevanz des Senders der Information (*relo*) höher gewichtet als *relt* und *reltype*.

Die Effekte dieser unterschiedlichen Gewichtungen zeigen sich in den folgenden Abbildungen:

- Abbildung 16 zeigt ein *squeezing maneuver*, bei dem sich die Robotergruppe zwischen zwei kreisförmigen Hindernissen hindurch bewegt.
- In Abbildung 17 ist zu erkennen, wie ein halbkreisförmiges Hindernis die Gruppe aufteilt. Die Roboter korrigieren daraufhin ihre Bewegungsrichtung, indem sie ihren aktuellen Aktionsstatus überprüfen und anpassen.
- Abbildung 18 veranschaulicht, dass durch die priorisierte Hindernisvermeidung eine Fragmentierung der Gruppe auftritt. Dennoch erreichen letztlich alle Roboter das gewünschte Ziel.

## VII. CONCLUSION

In dieser Arbeit wurde ein kollektives Navigationskonzept für mehrere autonome Roboter ohne Vorwissen über die Umgebung vorgestellt. Es basiert auf einer tangentialen Fluchtstrategie und dem Austausch lokaler Informationen über ein Kommunikationsnetzwerk. Künstliche Kräfte aus Potentialfeldern ermöglichen kollisionsfreie, kooperative Manöver und ein schwarmähnliches Verhalten. Durch Optimierung der Parameter der Relevanzfunktion und Gewichtungsfaktoren kann der Fragmentierung von Robotergruppen entgegengewirkt werden. Sollten Roboter ihre Verbindung zu Anderen verlieren, können sie auch allein ihr Ziel erreichen. Für holonomische<sup>3</sup> Roboter ist dabei ein ausreichend großer Sicherheitsabstand und niedrige Geschwindigkeit notwendig. Künftige Arbeiten sollen sich auf eine Echtzeit-Umsetzung, verbesserte drahtlose Kommunikation und alternative Netzwerktopologien für Roboter konzentrieren.

## 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>

<sup>3</sup>Position kann unabhängig der Ausrichtung geändert werden