



Zentrale vs. Dezentrale Adaptierungsplanung in Multidrohne-Szenarien

Bachelorarbeit

NAME

Matr.-Nr. NUMMER

DATUM

Betreuer:

Prof. Dr. Sabine Glesner

- WiMi -

- ggf. WEITERER WiMi -

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, den DATUM

Zusammenfassung

DEUTSCHES ABSTRACT

Abstract

ENGLISCHES ABSTRACT

Inhaltsverzeichnis

1 Einleitung

HIER BEGINNT DIE ARBEIT

1.1 Motivation

Koordination bedeutet das aufeinander Abstimmen der verschiedenen Komponenten, die ein Kollektiv ausmachen. Als entscheidendes Ziel eines Selbst-adaptiven Systems kann auch die Koordination der einzelnen Komponenten als Teil der Anpassung betrachtet werden. Um mit dem unsicheren und statisch unvorhergesehenen Umgebungsverhalten komplexer Systeme fertig zu werden, hat die Selbst-anpassungsfähigkeit breite Akzeptanz gefunden. Als Beispiel betrachten wir einen Lieferservice, der aus selbst-adaptiven Drohnen besteht.

Selbst-adaptive Systeme verfügen über die Fähigkeit, sich an Änderungen in ihrer Ausführungsumgebung und inneren Dynamik zu adaptieren, wie beispielsweise Reaktion auf Ausfall, Variabilität der verfügbaren Ressourcen oder sich ändernde Benutzerprioritäten, um weiterhin ihr Ziel zu erreichen [1]. Das ermöglicht den Drohnen, z.B. ihre Flughöhe anzupassen, wenn sie einen neuen Bereich betreten, in dem andere Regelungen gelten oder zum Akkusparmodus zu wechseln, wenn ihr Akkustand einen bestimmten Grenzwert erreicht. Rückkopplungsregelkreise wurden als entscheidende Elemente in der Realisierung der Adaptierung von Softwaresystemen identifiziert [1]. Zu diesem Zweck benutzen wir IBMs MAPE-K Schleife. Diese Schleife besteht aus 4 Phasen, nämlich “Monitor”, “Analyse”, “Plan” und “Execution”, die zusammen eine statische Adaption unseres Systems ausführen.

1.2 Problemstellung

Die Ausgangslage ist ein Szenario, in dem eine Menge von selbst-adaptiven Drohnen einen Lieferservice in einer sich ändernden Umgebung vornehmen muss, wobei jede Drohne über ein voneinander unabhängiges System verfügt. Hierbei entsteht das Problem, dass Adaptionen mehrere Drohnen betreffen können oder die Mitarbeit einer anderen Drohne erfordert. Bei einem Ausweichmanöver zur Kollisionsvermeidung sollte beispielsweise die Flughöhe so angepasst werden, dass die Anpassung beider beteiligter Drohnen nicht in die selbe Richtung erfolgt. Die Mitarbeit einer anderen Drohne ist zum Beispiel erforderlich, wenn eine Drohne aufgrund niedrigen Batteriestands ihr Transportgewicht reduzieren muss um sicher bis zur Ladestation zu kommen. Hier sollte eine andere Drohne Pakete übernehmen können. Lokale Adaptionen stoßen hier an ihre Grenzen.

Im nächsten Absatz stellen wir unsere Lösungen dar, um diese Probleme zu behe-

ben.

1.3 Ziel und Lösungsansatz

Wenn die Systeme groß, komplex und heterogen sind, reicht eine einzelne MAPE-K Schleife möglicherweise nicht aus, um die Adaptierung auszuführen. In solchen Fällen können mehrere MAPE-Schleifen verwendet werden, die verschiedene Teile des Systems verwalten [1] und Informationen austauschen. Im Bereich der Koordination von mehreren voneinander unabhängigen Systemen stehen viele Design-Möglichkeiten zur Verfügung. Diese sind grundsätzlich zwei: der (a) zentrale und der (b) dezentrale (oder verteilte) Ansatz. (a) könnte zum Beispiel umgesetzt werden durch eine hierarchische Anordnung der MAPE-K Loops, wobei jede Ebene der Hierarchie Instanzen von allen vier MAPE-Komponenten enthält [1]. Für (b) liegen mehrere Schemata vor, die verschiedene Ansätze durch Zentralisierung und Dezentralisierung der Funktionen der Selbstadaptation auf unterschiedliche Weisen [1] darstellen.

Nun werden die Ziele dieser Arbeit definiert. Erstens möchten wir eine Anwendung zur “Multi-Drohnen Kollisionsvermeidung” entwickeln. Dazu erstellen wir einen Konsens über Flugkorridore in verschiedenen Höhen. Die Korridore müssen sich an die individuellen Einschränkungen jeder Drohne halten. Zweitens, wird auch eine Anwendung zur “Multi-Drohnen Paket Optimierung” entwickelt. Diese Anwendung ermöglicht, dass die Drohnen Pakete von anderen Drohnen übernehmen können, wenn deren Akkustand niedrig ist.

Zu diesem Zweck ist ein Informationsaustausch zwischen den Drohnen erforderlich. Wie oben erklärt, gibt es hierfür zwei Vorgehensweisen: der zentrale und der dezentrale Ansatz. Um den ersten Ansatz auszuführen, benutzen wir eine hierarchische Anordnung der MAPE-K Schleifen. Um den zweiten Ansatz vorzunehmen, benutzen wir Dezentralisierungsvorlagen in [1] präsentiert. Schließlich nehmen wir eine Vergleichung beider Ansätzen miteinander vor.

Darüber hinaus realisieren wir eine Implementation der zentralen und dezentralen Systemen für die Anwendung zur “Multi-Drohnen Kollisionsvermeidung”, wobei C++ Programmiersprache benutzt wird und SystemC Kanäle für die Kommunikation verantwortlich sind. Als ein wünschenswertes Ziel, implementieren wir auch die Anwendung zur “Multi-Drohnen Paket Optimierung”. Falls es die Zeit nicht erlaubt, erklären wir die Vorgehensweise, wie man das implementieren könnte.

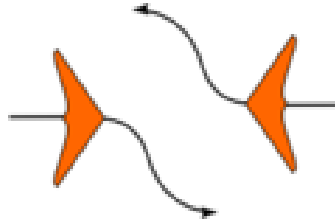


Abbildung 1: Einander entgegenkommende Luftfahrzeuge haben stets nach rechts auszuweichen.

1.4 Hintergrund des MAPE-Ks

1.5 Verwandte Arbeiten

2 Drohnenanordnungsmethode

Zuallererst stellen wir unsere Drohnenanordnungsmethode vor. Es handelt sich um eine Methode, deren Aufgabe es ist, einen Plan zur Anpassung der Flughöhe der Drohnen durchzuführen und dabei enge Flugbahnen und ggf. Kollisionen zu vermeiden. Im Gegensatz zur Annäherung zweier Luftfahrzeuge mit Kollisionsgefahr, wobei meistens Ausweichregeln zwischen Luftfahrzeugen zum Tragen kommen, stellt unsere Drohnenanordnungsmethode eine vorbeugende Maßnahme dar, die den Luftraum aufteilt.

Beispiele von Ausweichmanövern¹ bei Kollisionsgefahr können wir in den Abbildungen 1, 2 und 3 sehen. Eine Annäherung mit Kollisionsgefahr ist tatsächlich eine Situation, die für Verkehrsflugzeuge extrem selten stattfindet. Um dies zu verhindern, werden TCAS's (Traffic Alert and Collision Avoidance System) benutzt, die spezifische „Schutzzonen“ für die eigenen Flugzeuge definieren und dazu beitragen, zu nahe beieinander liegende Flugbahnen und damit das Risiko einer Annäherung mit Kollisionsgefahr zu vermeiden.

Die Drohnenanordnungsmethode wird innerhalb einer Gruppe von Drohnen aufgerufen und ausgeführt, die zu nahe beieinander in der XY-Ebene liegen. Späterhin bei den Abschnitten 3 und ?? betrachten wir zwei verschiedene Wege, durch die wir Gruppen von Drohnen bilden, die einander nahe stehen. Im Moment erklären wir jedoch nur, wie die Methode innerhalb einer Gruppe funktioniert.

¹<https://de.wikipedia.org>

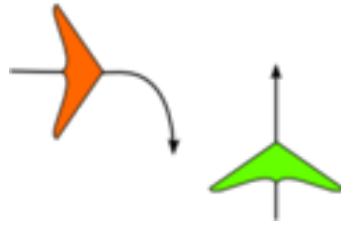


Abbildung 2: Von links kommende Luftfahrzeuge sind ausweichpflichtig.

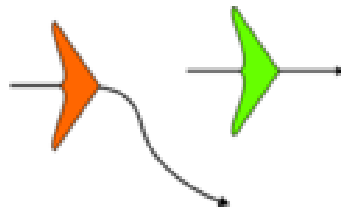


Abbildung 3: Überholende Luftfahrzeuge haben rechts am Überholten vorbeizufiegen.

2.1 Funktionsweise

Nun erklären wir die Funktionsweise unserer Methode. Eine Schleife durchläuft alle Korridore von oben nach unten. Bei jedem Korridor überprüfen wir zunächst, ob dieser überbelegt ist bzw., ob er mehr als eine Drohne hat. Soll es voll sein, so sollen wir nach einem freien Flugkorridor für die Drohne mit dem leichtesten Paket suchen. Eine vollständige Erklärung zur Methode finden wir in [Abbildung 4](#).

Wenn es mehr als eine Drohne in einem Flugkorridor gibt, müssen die überschüssigen Drohnen in andere Korridore fliegen, bis die Überbelegung gelöst ist. Um zu entscheiden, welche Drohne den Korridor zuerst verlassen soll, verwenden wir ein Gewichtskriterium. Dieses lautet wie folgt: „Die leichteste Drohne im überbelegtem Korridor muss sich an der ersten Stelle bewegen“. Da die Drohnen einen Lieferservice ausmachen, trägt jede ein unterschiedliches Gewicht. Die Verwendung des Gewichts als Kriterium bietet uns eine hohe Zeit- und Energieeffizienz, denn eine leichtere Drohne erreicht ihr neues Flugziel schneller, wobei sie auch weniger Energie verbraucht.

Zur Veranschaulichung unserer Methode verwenden wir ein Beispiel. In der [Abbildung 6](#) sehen wir, wie die Methode Schritt für Schritt funktioniert.

2.1 Funktionssweise

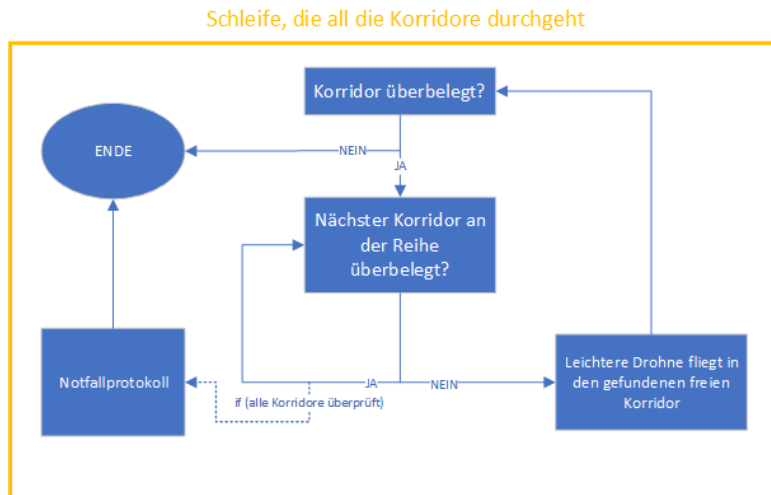


Abbildung 4: Vorgehensweise der Drohnenanordnungsmethode

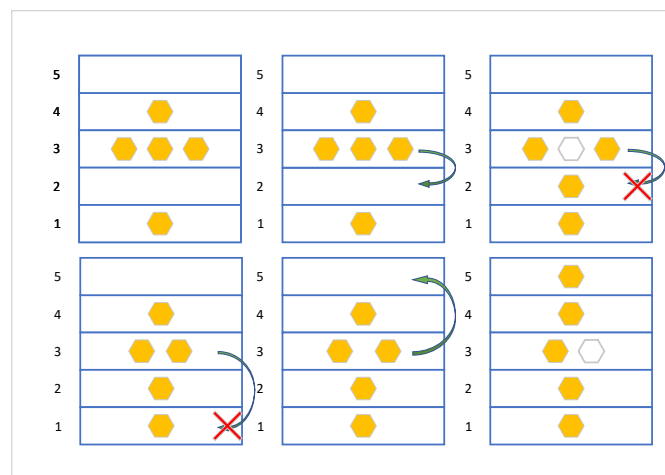


Abbildung 5: Illustratives Beispiel der Drohnenanordnungsmethode

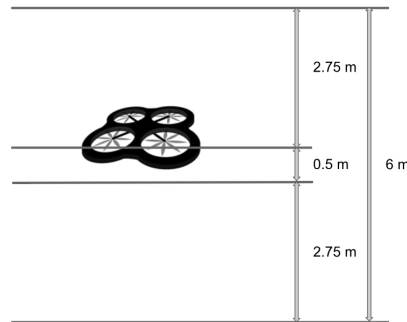


Abbildung 6: Dimensionen des Flugkorridors

2.2 Dimensionen

Im Folgenden stellen wir eine erste Annäherung an die Dimensionen dieser Flugkorridore in der realen Welt vor. Unter Berücksichtigung einer Dicke der Drohnen von ungefähr 0,5 Metern (einschließlich des Paketes) schlagen wir vor, Korridore von 6 Metern einzurichten. Dies bietet den Drohnen einen angemessenen Sicherheitsbereich und ermöglicht Kreuzungen ohne Störungen aufgrund Luftströmungen. In der Regel stellt sich eine Drohne in der Mitte ihres Korridors.

In diesem Abschnitt haben wir sowohl die Flugkorridore als auch unsere Dronen-anordnungsmethode vorgestellt. Als nächstes sprechen wir über den ersten Ansatz bei der Adaptierungsplanung: der zentrale Ansatz.

3 Kollisionsvermeidung: Zentraler Ansatz

Wie in Abschnitt 1.4 erklärt, besteht ein prominenter Ansatz zum Organisieren einer Regelschleife in selbstadaptiven Systemen aus vier Komponenten, die für die Hauptfunktionen der Selbstanpassung verantwortlich sind. Diese sind Überwachen, Analysieren, Planen und Ausführen, häufig als MAPE-Schleife bezeichnet [?]. Bei einem zentralen Ansatz verwenden wir ausschließlich eine MAPE-K Schleife für die Koordination unseres ganzen Systems. Das heißt, all die verschiedenen Gruppen von Drohnen, in denen eine Drohnenanordnungsmethode (siehe Abschnitt 2) abgerufen wird, müssen von einer einzigen, allgemeinen Schleife geregelt werden. Der erste Schritt zu diesem Zweck ist die Festlegung, nach welchem Kriterien wir lokale Gruppe von Drohnen definieren, bei denen Drohnen zu nahe beieinander stehen.

3.1 Gruppenerkennung

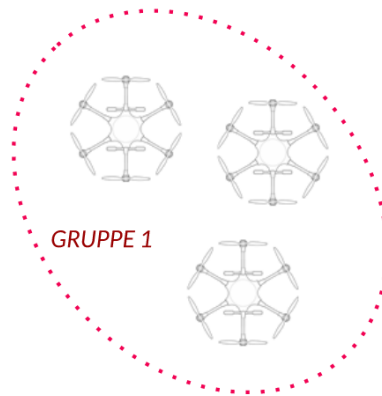


Abbildung 7: Jede Drohne steht zu nahe von all den Rest

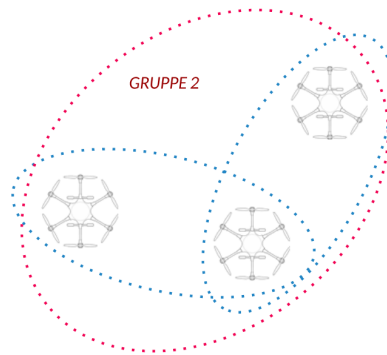


Abbildung 8: Kettenanordnung

3.1 Gruppenerkennung

3.1.1 Grundlagen

Ein einfaches Beispiel für eine Gruppe von zu nahe beieinander stehenden Drohnen ist eins, bei dem jede Drohne zu nahe von dem Rest ist. Die Abbildung 7 stellt ein Beispiel für diesen Fall dar. In der Praxis gibt es aber kompliziertere Fälle. Abbildung 8 veranschaulicht einen Fall, wo drei Drohnen eine „Kettenanordnung“ erzeugen, d.h. jede Drohne befindet sich ausschließlich zu nahe an der nächsten Drohne in der Kette. Solche Ketten, die auch über vielfache Zweige verfügen können, werden auch als einzelne Gruppe betrachtet.

Nun, wo wir wissen, was eine Gruppe bei dem zentralen Ansatz ist, können wir eine Methode zu ihrer Erkennung einführen.

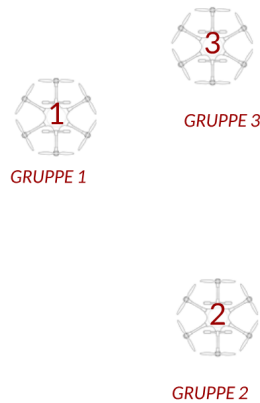


Abbildung 9: Jede Drohne verfügt über ihre eigene Gruppe

3.1.2 Methode

In der Monitoring-Phase unseres MAPE-K prüfen wir zunächst auf mögliche Gruppen von Drohnen, d.H. Stellen, wo eine Drohnenanordnung ausgeführt werden muss. Wie genau dieses Verfahren in der Monitoring-Phase integriert ist, wird in einem späterem Abschnitt besprochen. Im Moment führen wir jedoch nur die Funktionsweise der Methode ein.

Nun können wir das Konzept der Parent- und Child-Drohne einführen. Jede Drohne verfügt über ihre eigene Gruppe, wo wir zu nahe stehende Drohnen speichern können (siehe Abbildung 9). Diese Drohne ist die Parent-Drohne ihrer Gruppe. Die Child-Drohnen sind also diejenigen, die wir wegen Überschreitung des Grenzwertes der Gruppe hinzufügen.

Die Methode zur Gruppenerkennung besteht aus einer Schleife, die über alle Drohnen geht. Der Prozess für jede Drohne bzw. Gruppe ist wie folgt:

1. Prüfen, ob Drohnen der **Parent-Drohne** zu nah stehen. Wenn eine Drohne(n) in der Nähe der Parent-Drohne gefunden wird, fügen wir diese der Gruppe als Child-Drohne hinzu.
2. Prüfen, ob Drohnen den **Child-Drohnen der Gruppe** zu nah stehen. Wenn eine Drohne(n) in der Nähe der Child-Drohnen gefunden wird, fügen wir diese als Child-Drohne der Gruppe hinzu.
3. Schritt 2 Wiederholen, bis keine Child-Drohne mehr hinzugefügt wird

In den Abbildungen 10, 11 und ?? können wir ein Beispiel dieses Verfahren sehen.

3.1 Gruppenerkennung

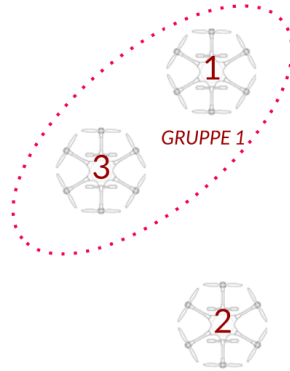


Abbildung 10: Im verwendeten Beispiel betrachten wir die Erkennung der Gruppe 1 und deswegen ist die Drohne 1 die Parent-Drohne. $\text{GRUPPE 1} = \{1, 3\}$

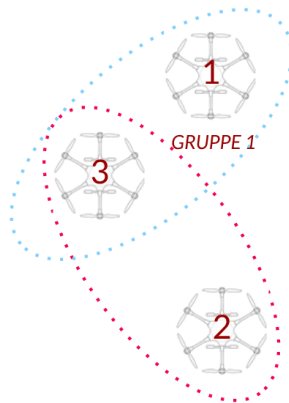


Abbildung 11: Nun sind sowohl die Drohne 3 als auch die Drohne 2 die Child-Drohnen von der Gruppe 1. $\text{GRUPPE 1} = \{1, 3, 2\}$

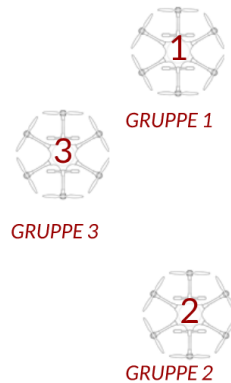


Abbildung 12: GRUPPE 1 = {1, 3, 2}, GRUPPE 2 = {2, 1, 3}, GRUPPE 3 = {3, 1, 2}

Am Ende des Prozesses haben wir also noch genauso viele Gruppen wie es Drohnen gibt. Das heißt aber nicht, dass wir in Wirklichkeit immer genauso viele Gruppen wie es Drohnen gibt haben, aber jetzt müssen wir die Gruppen filtern, die nicht repräsentativ sind. Damit ist der Prozess fertig.

Zur Optimierung des Verfahrens können wir folgende Weiterentwicklung betrachten: wir haben nicht eine Gruppe für jede Drohne, sondern erstellen sie nach Bedarf bzw. jedes Mal, zwei Drohnen **die noch keiner Gruppe angehören** zu nahe beieinander stehen. Auf diese Weise gewinnt der Prozess spürbar an Effizienz.

3.2 Zentrales MAPE-K

Nun erklären wir die Arbeitsweise des zentralen MAPE-Ks und seiner Phasen:

- **Monitoring-Phase:** Bei dieser Phase wollen wir Informationen über die Drohnen sammeln. Zuerst löschen wir die Gruppen vom letzten Zyklus. Anschließend führen wir die Gruppenerkennungsmethode aus, um diese zu bestimmen.
- **Analysis-Phase:** Im Wesentlichen wird es bei dieser Phase überprüft, ob das Ziel verletzt ist, bzw., ob Drohnen in einer gleichen Gruppe auch auf einem gleichen Korridor fliegen. Sollte dies der Fall sein, stellen wir das Ziel als „verletzt“ ein (Goal Violated) und triggern die Planning-Phase.
- **Planning-Phase:** Diese Phase wird erst aufgerufen, wenn die Variable „triggerPlan“ auf „TRUE“ gesetzt wird. Ist dies der Fall, sammeln wir Informa-

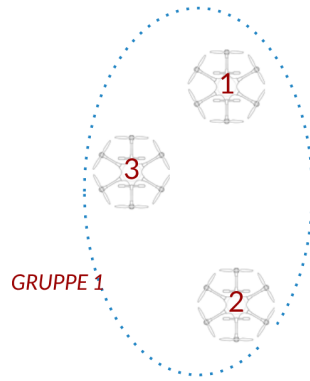


Abbildung 13: Wir filtern die Gruppen und behalten nur die, die repräsentativ sind

tionen darüber, welche Flugkorridor jede Drohne in der Gruppe belegt und anschließend führen wir die in Abschnitt 2 erklärte Drohnenanordnungsmethode.

- **Execution-Phase:** Bei der Execution-Phase handelt es sich um einen redundanten Schritt, denn wir haben keine Steuerungsparameter bei unserer Simulation. Die Existenz dieser Phase ermöglicht es uns jedoch, die Planning- und Execution-Phase separat auszuführen und somit die Steuerungsparameter in der Zukunft direkt einbeziehen zu können.

4 Kollisionsvermeidung: Dezentraler Ansatz

5 Evaluierung

6 Zukünftige Arbeit

7 Fazit

Literatur

- [1] Kephart. The vision of autonomic computing. computer 36(1),41–50. 2003.