KOLLISIONSFREIE FLUGBAHNEN FÜR KIPPFLÜGELFLUGZEUGE AUF BASIS DES WAVEFRONT-ALGORITHMUS UNTER BERÜCKSICHTIGUNG FLUGMECHANISCHER BESCHRÄNKUNGEN

I. Barz, L. Danielmeier, S. Seitz, D. Moormann
Institut für Flugsystemdynamik, RWTH Aachen University, Deutschland

Zusammenfassung

In diesem Beitrag wird eine Bahnplanungsmethode für unbemannte Kippflügelflugzeuge vorgestellt. Diese ermöglicht es, während des Fluges eine neue Flugbahn so zu planen, dass Hindernisse umflogen werden und flugmechanische Beschränkungen eingehalten werden. Dafür wird ein Wavefront-Algorithmus verwendet. Dieser basiert auf der zufallsbasierten, wellenartigen Expansion von Kreisen, die zur Exploration des hindernisfreien Raums genutzt werden. Um mithilfe des Wavefront-Algorithmus während des Fluges neue Bahnen planen zu können, welche die flugemechanische Beschränkung des minimalen Kurvenradius berücksichtigen, werden Modifikationen des Algorithmus vorgenommen. Die so bestimmte hindernisfreie Bahn wird anschließend verbessert, indem sie gekürzt und geglättet wird. In einem Validierungsabschnitt werden durch mehrere Monte-Carlo-Simulationen die Funktionalität und die Leistungsfähigkeit der Bahnplanung nachgewiesen.

1. EINLEITUNG

In mehreren Forschungsprojekten des Instituts für Flugsystemdynamik wird der Einsatz von unbemannten Kippflügelflugzeugen (Bild 1) zur Unterstützung in Rettungsmissionen untersucht, um den Rettungskräften schnell ein Lagebild aus der Luft zu verschaffen.

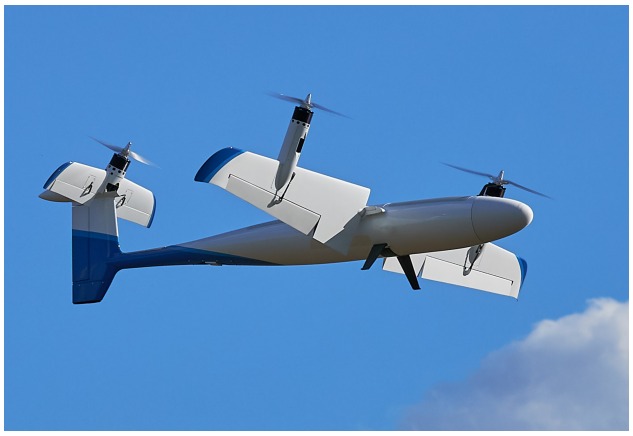


Bild 1. Kippflügelflugzeug im Flug

Kippflügelflugzeuge eignen sich hierbei aufgrund ihrer Fähigkeit sowohl einen effizienten Flächenflug, als auch einen Schwebeflug, der ein vertikales Starten- und landen ermöglicht, durchführen zu können. Somit kombiniert das Kippflügelflugzeug die Vorteile der Starr- und der Drehflügler-Konfiguration.
Da das Kippflügelflugzeug vor allem durch die Anwendung in Rettungseinsätzen oft nah am Boden operiert, muss bei der Planung von Flugbahnen eine Hindernisvermeidung berücksichtigt werden. Um in Rettungseinsätzen beispielsweise ein anderes Einsatzgebiet zu erkunden, soll das Flugsystem zudem auf Eingaben eines Anwenders reagieren können. Daraus ergeben sich zwei Anforderungen an die Planung von neuen, hindernisfreien Flugbahnen: Zum einen muss, um ein unnötiges Fliegen entlang der alten Flugbahn zu vermeiden, möglichst schnell eine neue Flugbahn bestimmt werden. Zum anderen ergeben sich aus dem aktuellen Flugzustand flugmechanische Beschränkungen an die Geometrie der Flugbahn wie beispielweise der minimal einzuhaltende Kurvenradius, die in keinen Fall verletzt werden dürfen.
In diesem Beitrag wird eine Planungsmethode für unbemannte Kippflügelflugzeuge vorgestellt, die während des Fluges schnell und unter Berücksichtigung flugmechanischer Beschränkungen eine kollisionsfreie Flugbahn bestimmt.
In der Literatur finden sich bereits unterschiedliche Methoden zur Planung von hindernisfreien Flugbahnen. Darunter befinden sich gitterbasierte Ansätze, die beispielsweise mit Hilfe von Dijkstra [1] oder A* [2] eine Lösung finden, samplebasierte Ansätze, wie beispielsweise RRT (rapidly exploring random tree) oder Methoden, die auf evolutionären Algorihmen basieren wie beispielsweise generische Algorithmen [3] [4].
Die gitterbasierten Ansätze sind zwar komplett, das heißt sie garantieren eine Lösung, sofern sie existiert. Für komplexe Probleme sind sie jedoch zu rechenintensiv. Zudem haben die Algorithmen Schwierigkeiten, Beschränkungen zu berücksichtigen. Die samplebasierten Algorithmen sind in der Regel effizienter und können somit auch komplexere Probleme lösen. Die Berücksichtigung von Beschränkungen wie beispielsweise den minimalen Kurvenradius führt hier jedoch oft zu nicht optimalen Lösungen. Die evolutionären Algorithmen können

CC BY 4.0

doi:10.25967/530192

auch hochdimensionale Probleme effizient lösen, können aber leicht in lokalen Optima gefangen werden.
Um die Anforderungen einer performanten Lösung, welche flugmechanischen Beschränkungen gerecht wird, zu erfüllen, wenden wir als Grundlage den in [5] vorgestellten, gitterbasierten Wavefront-Algorithmus an. Eine Anwendung im Bereich der Robotik wird bereits in [6] beschrieben, wobei sich wellenartig ausbreitende Kreise variabler Größe zur Exploration des hindernisfreien Raums genutzt werden. Durch die Anwendung von variablen Kreisgrößen können hierbei die Durchlaufzeiten auch für komplexe Probleme verringert werden.
Die Darstellung des hindernisfreien Raums über Kreise eignet sich hierbei besonders, um die Beschränkung des minimalen Kurvenradius zu berücksichtigen. Dafür modifizieren wir den Wavefront-Algorithmus so, dass bereits während der Exploration der minimale Kurvenradius berücksichtigt wird.
In diesem Beitrag wird zunächst die am Institut für Flugsystemdynamik verwendete Flugführungsarchitektur für Kippflügelflugzeuge vorgestellt, die es durch Vorgabe von flugmechanischen Beschränkungen ermöglicht, Flugbahnen dynamisch um- bzw. neu zu planen. Hieraus werden dann die Randbedingungen für die Bahnplanung hergeleitet. Anschließend wird der verwendete Wavefront-Algorithmus und dessen Modifikationen vorgestellt. Dieser wird in einem nächsten Schrittt bezüglich Laufzeit und Bahnlänge verbessert. In einem Validierungsabschnitt werden durch Monte-Carlo-Simulationen die Funktionalität und Laufzeit der Bahnplanungsmethode aufgezeigt.

2. BAHNPLANUNG MITTELS WAVEFRONT-ALGORITHMUS

2.1. Randbedingungen der Bahnplanung

Die Randbedingungen der Bahnplanung ergeben sich aus der Flugführungsarchitektur. Diese besteht aus verschiedenen Komponenten, die in einer Kaskadenstruktur angeordnet sind. In oberster Ebene befindet sich die Missionssteuerung. Diese ist als Zustandsautomat ausgeführt, der Eingaben von einem Bediener erhält, welcher jederzeit Missionsbefehle hinzufügen, löschen oder ändern kann. In Abhängigkeit von diesen Befehlen bestimmt die Missionssteuerung den aktuell durchzuführenden Flugauftrag. Um eine solche Aufgabe zu erfüllen, bestimmt der globale Bahnplaner eine Flugbahn, die dann an einen lokalen Bahnplaner weitergeleitet wird. In diesem wird die Bahn gegebenenfalls angepasst, um plötzlich aufkommende, dynamische Hindernisse zu vermeiden [7]. Als nächstes plant der Bahnregler die Geschwindigkeit entlang der Bahn nach dem Prinzip des zurückweichenden Horizonts [8]. Zudem berechnet der Bahnregler Bahnbeschränkungen, z.B. den minimalen Kurvenradius, die an die Bahnplanungskomponenten weitergegeben werden [9].
Bei der Planung von neuen Flugbahnen müssen diese Beschränkungen berücksichtigt werden, die sich aus dem aktuellen Flugzustand ergeben. Zudem ist zu berücksichtigen, dass die Planung einer neuen Flugbahn eine gewisse Zeit in Anspruch nimmt. Um unter Berücksichtigung dieser Zeit einen stetigen Übergang von der alten Flugbahn zur neuen zu gewährleisten, legen wir im Voraus eine feste Planungszeit Δt fest, innerhalb der eine neue Flugbahn geplant werden muss. Die neue Flugbahn beginnt dann an der Stelle der alten Flugbahn, an der sich das Flugsystem nach dem festgelegten Zeitpunkt befindet.
Für die Bestimmung der neuen Flugbahn werden also Informationen über die Geometrie der Flugbahn nach der Planungszeit Planungszeit Δt benötigt. Dies umfasst die Position $\vec{x}_1$, den Bahnwinkel χ_1 und den Steigwinkel γ_1. Diese Geometrieparameter müssen mit der neuen Flugbahn übereinstimmen, um einen stetigen Übergang von der alten Flugbahn zur neuen Flugbahn zu gewährleisten. Neben den Anfangsrandbedingungen muss die Bahnplanung auch Bedingungen am Endpunkt der zu planenden Bahn berücksichtigen. Hier wird entsprechend der Anwendung eine Position $\vec{x}_2$, ein Bahnwinkel χ_2 und ein Steigwinkel γ_2 vorgegeben.
Der sich aus dem Flugzustand nach der Zeit Δt ergebende minmale Kurvenradius wird vom Bahnregler unter Berücksichtigung der aktuellen Windsituation sowie flugzeugspezifischen Parametern wie der maximalen Querbeschleunigung oder der maximalen Heading-Änderungsrate bestimmt.
Die vorhandenden Hindernisse gehen ebenfalls als Eingangsparameter in die Bahnplanung ein. Hindernisse können beispielsweise hohe Gebäude oder Flugverbotszonen darstellen. Diese werden für die Bahnplanung als Kreise modelliert, mit deren Radius auch ein zusätzlichen Sicherheitsradius berücksichtigt werden kann.

2.2. Bahnplanungsalgorithmus

Um die in 2.1 beschriebenen Randbedinungen einzuhalten, wird der COWEX-Algorithmus (COnstrained Wavefront EXpansion) verwendet. Dieser greift die ursprüngliche Idee des Wavefront-Algorithmus auf. Die Idee der wellenartigen Ausbreitung wird im COWEX-Algorithmus so erweitert, dass bereits bei der Wellenausbreitung die flugmechanische Beschränkung des minimal einzuhaltenden Kurvenradius berücksichtigt wird. Die Funktionsweise des Algorithmus ist in Algorithmus 1 dargestellt.
Ausgehend vom Startpunkt wird ein Kreis aufgespannt, dessen Radius so groß ist wie der Abstand zum nächstgelegenen Hindernis in der Umgebung. Somit wird sichergestellt, dass der Baum mit allen Kreisen ein Volumen von freiem Raum beschreibt. Der Kreis wird jedoch maximal so groß wie ein vorgegebener maximaler Radius.
Anschließend werden die Kosten für diesen Kreis bestimmt.

CC BY 4.0

Algorithmus 1 COWEX

procedure COWEX($x_1, \chi_1, x_2, \chi_2, r_{min}, obs, n$)
  $S \leftarrow \varnothing$ ▷ Aktivliste
  $Q \leftarrow \varnothing$ ▷ Prioritätsliste
  $r_s \leftarrow$MINIMUM(r_{max},DIST(x_1, obs))
  $S \leftarrow (x_1, r_s, \varnothing)$
  $k \leftarrow$COST(Q, s_1, x_2)
  INSERT (Q, s, k)
  while $\|\boldsymbol{x_2} - \boldsymbol{c_{center}}\| > r_{min}$ **do**
    $\{c_{center}, r_s, x_{parent}\} \leftarrow$MIN_C($Q$)
    $S \leftarrow S \cup \{c_{center}, r_s, x_{parent}\}$
    $P \leftarrow$SAMPLE_CON(S, r_{min}, n)
    for each $p \in P$, **do**
      if $\neg$ IN_PARENT_CIRC(Q, p), **then**
        $r' \leftarrow$MINIMUM(r_{max}, DIST (p_i, obs))
        $s' \leftarrow (P_i, r', s)$
        $k \leftarrow$COST(Q, s_i, x_2)
        INSERT(Q, s', k)
  return S

Der erste Kreis und dessen Kosten werden nun in die Prioritätsliste Q verschoben. Dies ist eine Liste, in der alle generierten Kreise eingetragen werden und nach aufsteigenden Kosten sortiert werden.
Die nachfolgend beschriebenen Schritte werden solange wiederholt, bis der Zielpunkt x_2 in einem aufgespannten Kreis liegt. Zunächst wird der Kreis mit den geringsten Kosten als aktiv bezeichnet. Dieser wird dann aus der Prioritätsliste gewählt und in die Aktivliste S geschoben. Dies ist eine Liste, in der alle Kreise gespeichert werden, die einmal als aktiv bezeichnet wurden. Dabei muss der Kreis mit den geringsten Kosten, der aus der Prioritätsliste in die Aktivliste verschoben wird, nicht unbedingt aus dem vorher aktiven Kreis entstehen. Der neue aktive Kreis dient nun als Elternkreis. Auf seinem Rand werden n neue Kreise erzeugt und in die Prioritätsliste aufgenommen.

Die Besonderheit des COWEX-Algorithmus liegt in der Bestimmung der neuen Kreise. Diese werden so generiert, dass der minimale Kurvenradius berücksichtigt wird. Dazu ist in Bild 2 zunächst aufgezeigt, wie aus Wegpunkten eine Flugbahn mit einzuhaltendem minimalen Radius entsteht.

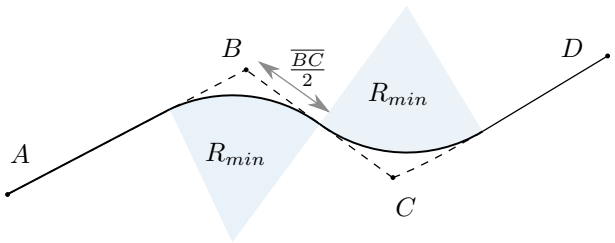


Bild 2. Verbinden von Wegpunkten

Anstatt dass sich die Geraden unter einem Winkel schneiden, werden sie durch ein Kreissegment mit dem Radius R_{min} verbunden. Dies ist in Bild 2 zwischen den Geraden $\overline{AB}$ und $\overline{BC}$ zu sehen. Der Mittelpunkt dieses Kreissegmentes liegt auf der Winkelhalbierenden und tangiert beide Geraden. Um nun auch den Kurvenradius im nachfoglenden Segment, also dem zwischen den Geraden $\overline{BC}$ und $\overline{CD}$, zu berücksichtigen, wird hier gefordert, dass das Kreissegement spätestens auf der Hälfte der Strecke $\overline{BC}$ tangiert. Um dies zu ermöglichen, muss der Winkel zwischen zwei Geraden betragsmäßig größer als ein Grenzwinkel χ_{grenz} sein, der mit (1) bestimmt werden kann.

$$\chi_{grenz} = 2 \cdot \arctan\left(\frac{2 \cdot R_{min}}{min(\overline{AB}, \overline{BC})}\right) \tag{1}$$

Übertragen auf den COWEX-Algorithmus wird dafür ein Elternkreis in zwei Bereiche aufgeteilt, die durch den Winkel χ_{grenz} getrennt sind (Bild 3).

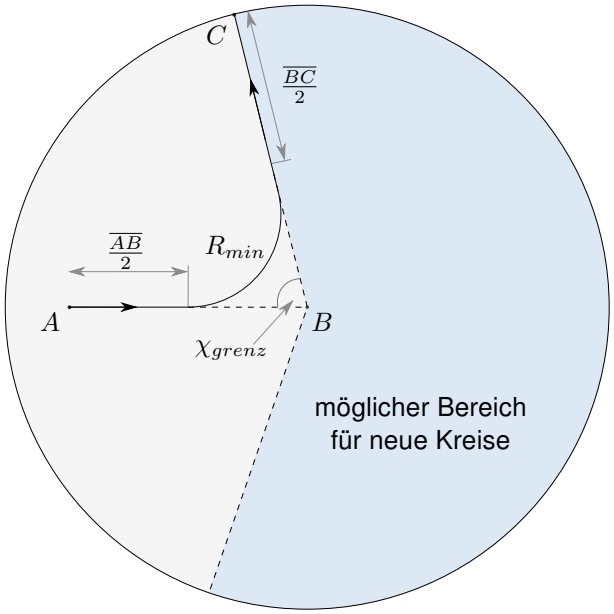


Bild 3. Unterteilung des Elternkreises in zwei Bereiche

Der Punkt A entspricht hierbei dem Mittelpunkt des Vorgängerkreises. Der Punkt B stellt den Mittelpunkt des aktuellen Elternkreises dar. Auf dem Rand dieses Elternkreises sollen nun neue Kreise mit dem Mittelpunkt C generiert werden. Dies ist in dem Bereich möglich, in dem der Winkel betragsmäßig größer als der Grenzwinkel χ_{grenz} ist. In diesem Beispiel ist der Radius des Vorgängerkreises, also die Strecke $\overline{AB}$, größer als der Radius des aktuellen Elternkreises, also die Strecke $\overline{BC}$. In Bild 4 ist dargestellt, wie auf dem Rand des Bereiches für den der Winkel größer ist als χ_{grenz} neue Kreise aufgespannt werden. Die neu aufgespannten Kreise können eigentlich einen größeren Radius haben, sind hier aber zur Übersichtlichkeit verkleinert dargestellt.
Um ein Verlaufen in einer Sackgasse zu vermeiden, werden zudem die neu aufgespannten Kreise verworfen, wenn ihr Mittelpunkt in einem Kreis liegt, der sich bereits in der Aktivliste befindet.
Die neu aufgespannten Kreise werden nun gewichtet

CC BY 4.0

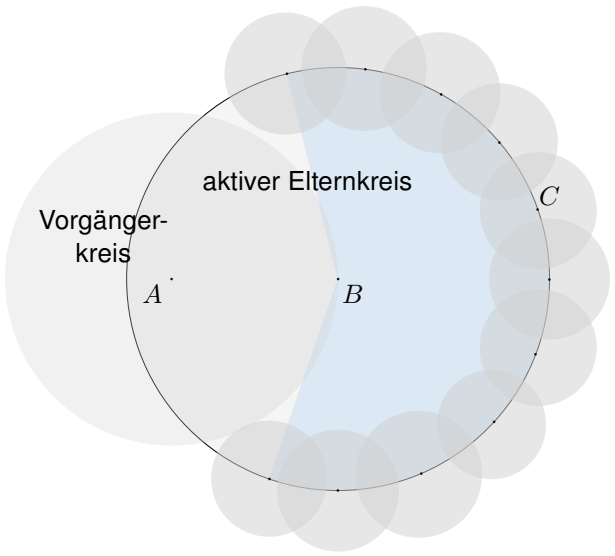
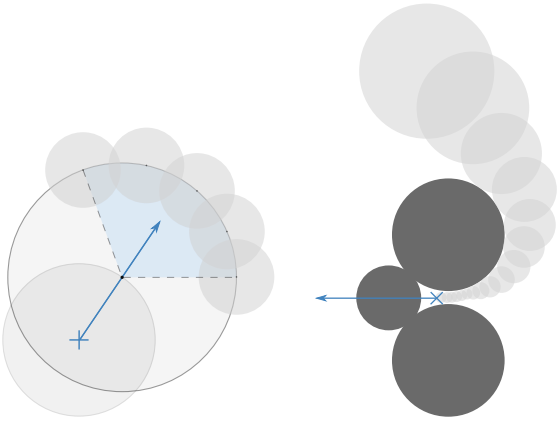


Bild 4. Neu aufgespannte Kreise für fliegbare Bahn

und der Prioritätsliste hinzugefügt. Anschließend wird aus der Prioritäsliste wieder der Kreis mit den geringsten Kosten gewählt. Auf dem Rand dieses Kreises werden wieder neue Kreise generiert. Dies wiederholt sich so lange, bis der Zielpunkt erreicht ist. Durch das Verbinden aller Kreismittelpunkte unter Berücksichtigung des minimalen Kurvenradius entsteht nun eine fliegbare und hindernisfreie Flugbahn.

Um mit dem COWEX-Algorithmus die Randbedingungen des vorgegebenen Start- und Zielazimuts zu berücksichtigen, darf der erste Kreis nur in Startrichtung aufgespannt werden (Bild 5a). Die Größe des aufgespannten ersten Kreises wird hierbei stark limitiert, sodass der nicht genutzte Raum gering bleibt.



(a) Startrichtung **(b) Zielrichtung**

Bild 5. Berücksichtigung der Start- und Endrichtungg

Da die Kreise nur vom Start zum Ziel generiert werden und somit der minimale Kurvenradius nur vorwärts berücksichtigt wird, ist ein solches Vorgehen für den Endazimut nicht möglich. Das Einhalten der Endrichtung wird deshalb durch drei am Zielpunkt platzierte Pseudohindernisse, gewährleistet (Bild 5b).
Die Radien der Pseudohindernisse sind hierbei vom minimalen Kurvenrradius abhängig. Sie sorgen dafür, dass die neu augespannten Kreise im COWEX-Algorithmus in die richtige Richtung geleitet werden, sodass am Endpunkt der Zielazimut χ_2 eingehalten wird.

2.3. Verbesserungen in der Bahnplanung

Nach dem in Kap. 2.2 vorgestellten COWEX-Algorihmus kann nun also eine fliegbare Flugbahn generiert werden, die alle Randbedinungen berücksichtigt. Um die Rechenzeit dieses Algorithmus gering zu halten und um die Bahn hinsichtlich Flugzeit und somit Länge zu optimieren, werden einige Verbesserungen vorgenommen, die nachfolgend beschrieben sind. In einem letzten Schritt wird die verwendete Kostenfunktion angepasst.

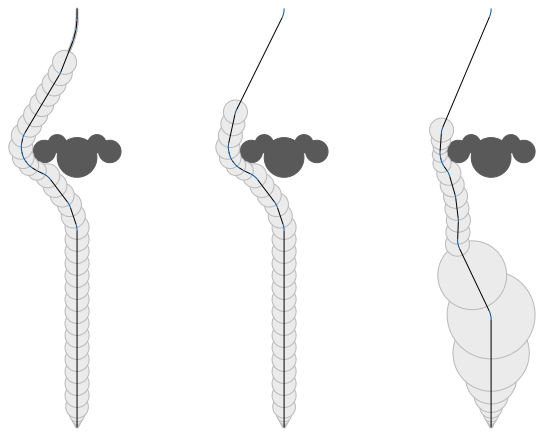
2.3.1. Laufzeitverbesserungen

Um die Durchlaufzeit der Bahngenerierung zu verringern werden zwei Verbesserungen vorgenommen. In der ersten Verbesserung wird jedes Mal, wenn ein Kreis in die Prioritätsliste hinzugefügt wird, geprüft, ob ein kollisionsfreier Dubinspfad vom Mittelpunkt des zuletzt hinzugefügten Kreises zum Zielpunkt generiert werden kann. Ist die der Fall, wird der COWEX-Algorithmus an dieser Stelle beendet.
In der zweiter Verbesserung wird die maximale Kreisgröße variiert, wenn das nächste Hindernis sehr weit entfernt ist. Das Hindernis wird dann als zu weit entfernt betrachtet, wenn die Distanz zwischen dem aktiven Kreis und dem nächsten Hindernis größer als vier mal der minimale Kurvenradius ist. Tritt dies ein, wird der nächste Kreis soweit aufgespannt, bis sein Radius so groß ist wie die halbe Distanz zu den Hindernissen. Somit wird die Distanz bis zum Erreichen des hindernisnahen Raums immer mit vier Kreisen überbrückt und die benötigte Rechenleistung sinkt. Wenn die Distanz zwischen den Kreisen und dem nächsten Hindernis kleiner wird als viermal der minimale Kurvenradius, werden die Kreise wieder bis zu einem fest definierten maximalen Radius aufgespannt. Ein Beispiel für die beiden Laufzeitverbesserungen ist in Bild 6 dargestellt. Bild 6a zeigt die aufgespannten Kreise ohne Verbesserung. In Bild 6b wird das frühzeitige Beenden des COWEX-Algorithmus gezeigt und in Bild 6c die zusätzliche variable Kreisgröße, wenn die Hindernisse weit genug entfernt sind.

2.3.2. Kürzung der Bahn

Wenn druch den COWEX-Algorithmus viele Kreise ausgespannt werden, kann die sich aus Kap. 2.2 ergebende Flugbahn durch Verbindung aller Kreismittelpunkte unter Berücksichtigung des minimalen Kurvenradius sehr lange werden. Hier findet ein ständiger Wechsel aus geraden und kreissegmentförmigen Flugbahnabschnitten statt. Um den Umweg zum Umfliegen der Hindernisse weiter zu minimieren,

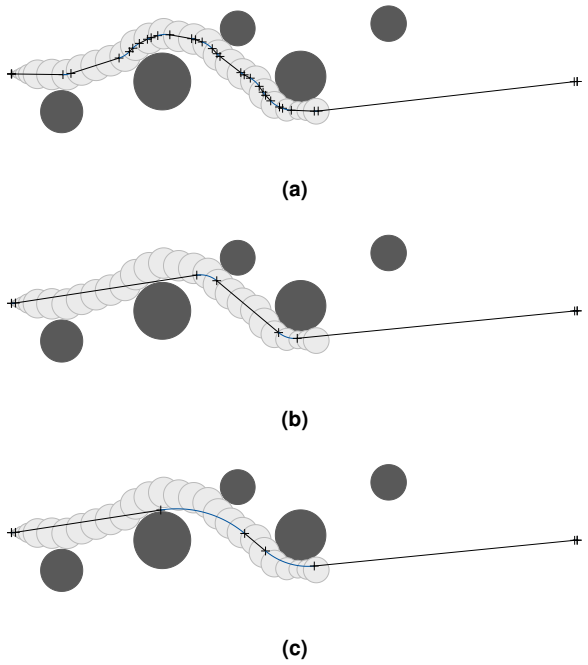
CC BY 4.0



(a) (b) (c)

Bild 6. Maßnahmen zur Laufzeitverbesserung

wird diese Bahn zunächst mittels A*-Algorithmus gekürzt [4] und anschließend durch das Vergrößern der Radien geglättet.
Beim A*-Algorithmus wird jede mögliche kollisionsfreie Verbindung von Wegpunkten gespeichert. Ausgehend von den kollisionsfreien Verbindungen der Mittelpunkte, wird ermittelt, welche Kombination von Wegpunkten die kürzeste Verbindung zwischen Start und Ziel darstellt. Da durch dieses Verfahren durch das Weglassen von Wegpunkten andere Winkel zwischen zwei Verbindungslinien von benachbarten Wegpunkten entstehen, muss im Anschluss sichergestellt werden, dass die gekürzte Bahn noch fliegbar ist, was mittels (1) überprüft wird.



(a)

(b)

(c)

Bild 7. Bahnkürzung und -glättung

Die Verbesserung der Flugbahn ist in Bild 7 zu sehen. In Bild 7a ist die Bahn ohne Kürzung zu sehen und in Bild 7b nach der Anwendung des A*-Algorithmus.
Um die Bahn noch weiter zu kürzen, wird für jede Kurve der Kurvenradius maximiert. Dieser ist entweder durch die kürzere der beiden Strecken zu den nächstgelegenen Wegpunkten, siehe Kap. 2.2, oder durch Hindernisse im Kurveninneren limitiert. Die weitere Kürzung ist in Bild 7c zu sehen.

2.3.3. Anpassung der Kostenfunktion

Abhängig vom Elternkreis und dem neu platzierten Kreis wird die Bahnrichtung durch Verbinden der beiden Kreismittelunkte bestimmt. Entlang dieser Richtung wird nach Hindernissen geschaut. Ist der Abstand zu einem Hindernis in dieser Richtung größer als viermal die aktuelle Kreisgröße und ist der aktuelle Kreis größer als sein Vorgänger bzw. größer als eine Mindestgröße, kann davon ausgegangen werden, dass momentan kein Hindernis eine Rolle spielt und der Fokus somit auf dem Erreichen des Zielpunktes liegt. In diesem Fall berechnen sich die Kosten nach (2).

$$k = 1,25 \cdot h + 0,25 \cdot g \tag{2}$$

Hierbei stellt h die Distanz zwischen dem Mittelpunkt des zu bewertenden Kreises und dem Zielpunkt dar. Die Variable g beschreibt die bisher zurückgelegte Distanz, die durch die Summe der Distanzen zwischen den Mittelpunkten aus allen sogenannten Elternkreisen bestimmt wird.
Ist dies nicht der Fall, kann davon ausgegangen werden, dass die Kreise sich schell einem Hindernis nähern beziehungsweise bereits nah an einem Hindernis sind. Daher liegt der Fokus hier auf Ausweichen. Die Kostenfunktion ergibt sich dann alternativ zu:

$$k = 2 \cdot \left(1 + \frac{R_{min}}{d_{nn}}\right) \cdot (h - R_{COWEX}) + 0,5 \cdot g \tag{3}$$

Hierbei beschreibt d_{nn} die Distanz des aktuellen Kreises zum nächstgelegenen Hindernis.
Zusätzlich werden die Kreise, die aus dem aktuell aktiven Kreis entstehen, für eine Bewertungsrunde deutlich günstiger bewertet, sodass der Algorithmus zunächst die aktuelle Expansion zuende plant. Sollte diese aktuelle Expansion in ein lokales Minimum laufen, kann dennoch zurückgesprungen werden.

3. VALIDIERUNG

Die Validerung des COWEX-Algorithmus wird mittels Monte-Carlo-Simulationen durchgeführt. Dazu wird eine sehr hohe Anzahl von Flugbahnen mit dem COWEX-Algorithmus generiert und anschließend bewertet.
Für jede der geplanten Bahnen wird ein definierter Satz an Eingangsparametern zufällig festgelegt, während ein anderer Satz von Eingangsparametern konstant gehalten wird. Somit kann der Einfluss verschiedener Eingangsparameter auf die Bahnplanung mit dem COWEX-Algorithmus festgestellt werden.

CC BY 4.0

Zusätzlich kann durch eine sehr hohe Anzahl an verschiedenen Bahnen validiert werden, dass der COWEX-Algorithmus immer eine kollisionsfreie fliegbare Bahn generiert, sofern diese existiert.
Neben den Kriterien der Kollisionsfreiheit und der Fliegbarkeit der Bahnen wird untersucht, wie sich die Durchlaufzeiten des COWEX-Algorithmus bei verschiedenen Eingangsparametern verhalten. Hierbei ist zu beachtetn, dass die in Kap. 2.1 beschriebene Planungszeit Δt in keinem Fall überschritten wird. In der vorhandenen Flugführungsumgebung ist $\Delta t = 2\ s$, weswegen alle Planungszeiten in der folgenden Validierung unter $t_{Planung} = 2\ s$ liegen müssen, um als erfolgreich zu gelten.

Als erstes wird der Einfluss der Start- und Endposition untersucht. Diese wirken sich indirekt über die Distanz zu den nächsten Hindernissen im Verhältnis zum minimalen Kurvenradius aus. Daher wird in den ersten Simulationen die Distanz zwischen Startposition und Hindernissen und zwischen Endposition und Hindernissen variiert. Zudem wird der minimale Kurvenradius für einen Wertebereich von 0 bis 200 m zufällig gewählt. Um bezüglich der Flugrichtungen den konservativsten Fall abzudecken, zeigt die Startrichtung immer auf das nächste Hindernis und die Endrichtung immer vom Hindernis weg. Bild 8 zeigt als Ergebnis der Simulationen und in Abhängigkeit des normierten Abstandes zum Hindernis, ob eine erfolgreiche Bahn geplant werden kann.

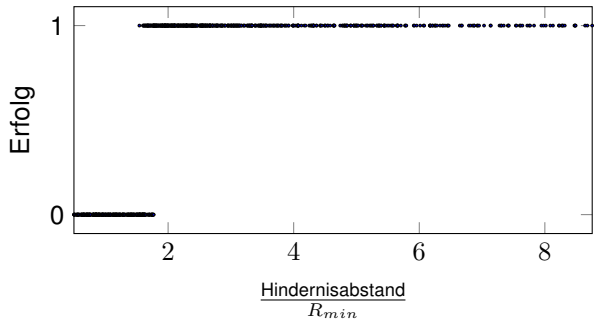


Bild 8. Bahnplanungserfolg in Abhängigkeit des Abstandes zu den Hindernissen

Es ist zu erkennen, dass nur Konstellationen, in denen der Startpunkt mehr als 1,8 Kurvenradien vom nächsten Hindernis entfernt liegt eine erfolgreiche Planung ermöglichen. Somit ist die minimale Distanz, die zu einer erfolgreichen Planung führt $1,8 \cdot R_{min}$.
Eine Bahn mit dieser Distanz ist in Bild 9b dargestellt.

Im Vergleich dazu ist in Bild 9a eine Bahn dargestellt, bei der der Startpunkt zu nah an den Hindernissen ist. In beiden Fällen ist zu erkennen, dass die neu augespannten Kreise vom Hindernis wegdrehen. In Bild 9a ist der Abstand jedoch nicht ausreichend.
Der Grund, weswegen die Grenze der fliegbaren Bahnen nicht bei einer normierten Distanz von 1 liegt, ist die Abschätzung, die in (1) getroffen wird. Diese besagt, dass der Kreisbogen zur Verbindung

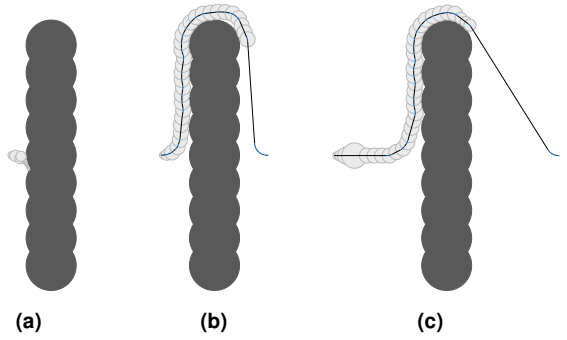


Bild 9. Variation Abstand Startpunkt zu Hindernissen

zweier Geraden beide Geraden spätestens bei der Hälfte tangiert haben muss. Diese Abschätzung ist für die Funktion des Algorithmus notwendig, führt jedoch zu einer konservativen Abschätzung, die in extremen Fällen wie in Bild 9a dazu führen, dass der Algorithmus keine Lösung findet.
Da im Anwendungsfall die Hindernisse im Voraus bekannt sind, kann es nicht zu einem Fall kommen, in dem so kurz vor einem Hindernis eine Ausweichbahn geplant werden muss. Somit ist das hier erzielte Ergebnis von 1,8 Kurvenradien mehr als ausreichend.
In Bild 9c ist ein typischer Anwendungsfall aus den ersten Simulationen gezeigt. Hier beträgt die Distanz zwischen Startpunkt und Hindernis sieben mal dem minimalen Kurvenradius. Der Algorithmus kann in diesem Fall deutlich früher beginnen, dem Hindernis auszuweichen und daher auch größere Kreise verwenden.

Da die Distanz zwischen den Hindernissen im Verhältnis zum minimalen Kreisdurchmesser $R_{min,COWEX}$ des COWEX-Algorithmus Einfluss auf die Durchlaufzeit hat, wird in weiteren Simulationen die Distanz zwischen den Hindernissen im Bereich von $0,5 \cdot R_{min,COWEX}$ bis $4 \cdot R_{min,COWEX}$ variiert.
Die Ergebnisse zeigen, dass ab einem Bereich, in dem die Distanz zwischen den Hindernissen größer als zweimal der minimale Radius $R_{min,COWEX}$ beträgt, die Durchlaufzeiten deutlich abfallen. Ab diesem Punkt können die Kreise durch den Spalt zwischen zwei Hindernissen navigiert werden, siehe Bild 10.

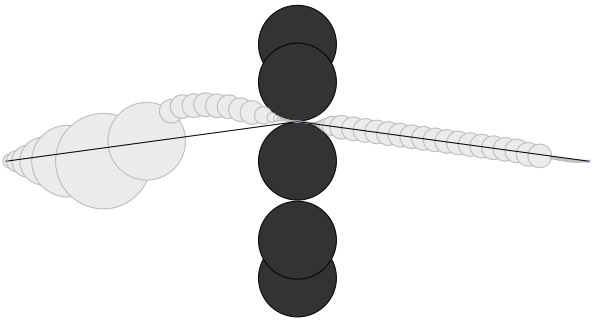


Bild 10. Distanz = 3 $R_{min,COWEX}$

CC BY 4.0

Bei einem kleineren Abstand der Hindernisse ist dies nicht mehr der Fall. Trifft der Algorithmus den Spalt nicht genau, oder ist er zu klein, kann er irgendwann keine Kreise mehr platzieren und muss an einen vorherigen Kreis zurückspringen bis schlussendlich die gesamte Hindernisgruppe vermieden wird. Ein solches Beispiel ist in Bild 11 dargestellt.

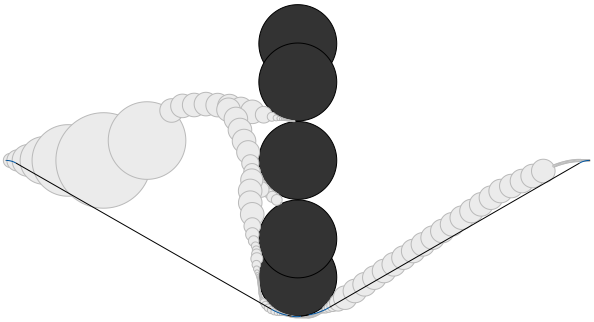


Bild 11. Distanz = 1,5 $R_{min,COWEX}$

In Bild 12 sind alle Durchlaufzeiten normiert auf die minimal benötigte Durchlaufzeit in schwarz dargestellt. Es ist ein großer Abfall der Durchlaufzeit des Algorithmus ab dem Bereich zu sehen, in dem die Distanz zwischen den Hindernissen größer als zweimal der minimale Radius $R_{min,COWEX}$ wird. Für die Fälle, wie in Bild 11 dargestellt, ist zu erkennen, dass die Durchlaufzeit bis zu viermal so hoch sind wie die schnellste Durchlaufzeit.

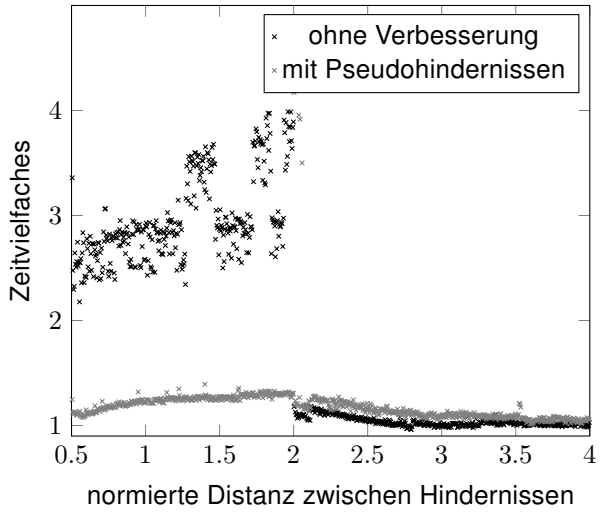


Bild 12. Durchlaufzeiten in Abhängigkeit der Distanz zwischen Hindernissen

Da in diesen Fällen im Vorhinein klar ist, dass der Spalt vom Algorithmus nicht passiert werden kann, werden in allen Spalten, die zu eng für den Algorithmus sind, Pseudohindernisse aufgespannt. Dies dient dazu, unnötige Zeit im Aufspannnen von Kreisen in den Spalten zu verlieren. Der Radius der Pseudohindernisse entspricht dem Radius des kleineren Hindernisses. Eine Verbesserung der Durchlaufzeiten bei Verwendung von Pseudohindernissen ist in Bild 12 zu erkennen. Die generierten Pseudohindernisse sind in Bild 13 in Grau gezeigt. Sie dienen nicht als Begrenzungen für die Kreisgröße, sondern wirken sich nur auf die Kostenfunktion aus.

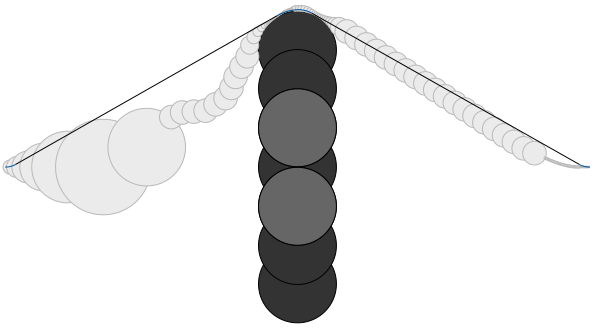


Bild 13. Pseudohindernisse, Distanz = 1,5 $R_{min,COWEX}$

Für den letzten Teil der Auswertung werden die Durchlaufzeit und die Zuverlässigkeit des COWEX-Algorithmus in Monte-Carlo-Simulationen untersucht, bei denen Hindernisse zufällig generiert werden. Um eine Aussage zur Durchlaufzeit zu erhalten, werden die Simulationen bereits auf der Zielhardware durchgeführt. Hierbei handelt es sich um einen phyCORE-AM5728-System-on-Module, von dem einer der 1,5 GHZ-taktenden Cortex-A15-Kerne für die Simulationen verwendet wird. Für die Simulationen werden drei unterschiedliche Konstellationen gewählt: Eine mit 150 kleinen Hindernissen, eine mit 30 mittelgroßen und eine mit vier kleinen. Für alle Konstellationen werden 100.000 Berechnungen mit zufällig generierten Hindernissen und zufällig gewähltem minimalen Kurvenradius durchgeführt. Anschließend wird die Durchlaufzeit und die Fehlschlagquote der Simulationen analysiert.
In Bild 14 sind die Ergebnisse der Simulation mit 150 sehr kleinen Hindernissen aufgetragen. In der Abbildung sind die Häufigkeit der auf der x-Achse aufgetragenen Durchlaufzeiten gezeigt.

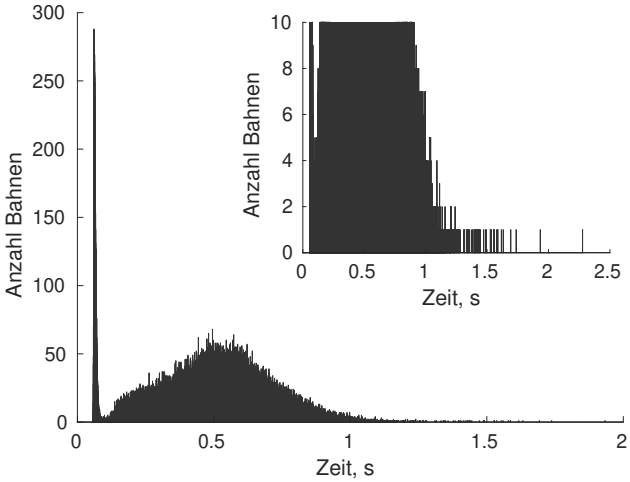


Bild 14. Durchlaufzeiten bei 150 Hindernissen

Es sind zwei Erhebungen zu beobachten. Die erste befindet sich bei sehr geringen Durchlaufzeiten von 0,05 bis 0,07 s. Die zweite bei Durchlaufzeiten von

CC BY 4.0

0,1 bis 1 s. Die erste dieser Erhebungen entsteht durch Bahnen, bei denen schon nach dem ersten Kreis eine kollisionsfreie Verbindung zwischen Start- und Zielpunkt existiert. Erst wenn die ersten Hindernisse den Algorithmus zwingen, die Kreise von der Verbindungslinie von Start- zu Zielpunkt wegzulegen, kommt es zu der zweiten Erhebung. Ab diesem Punkt ähnelt die Verteilung der Durchlaufzeiten einer Normalverteilung. Dies erklärt sich dadurch, dass die Durchlaufzeit nur noch davon abhängt, wie viele Kreise aufgespannt werden müssen, bevor eine kollisionsfreie Verbindung zwischen aktuellem Kreis und Zielpunkt existiert.

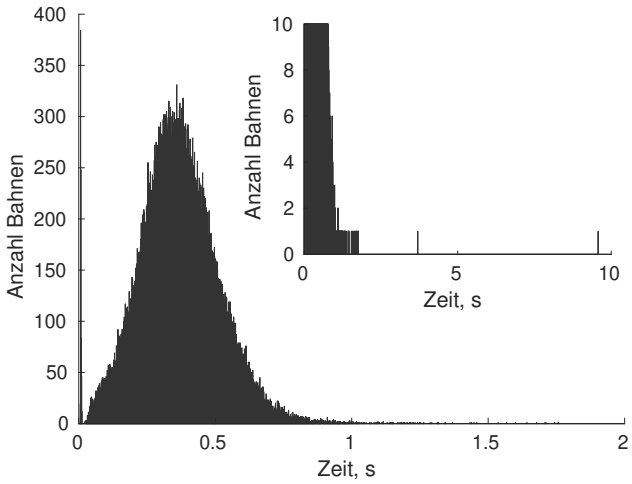


Bild 15. Durchlaufzeiten bei 30 Hindernissen

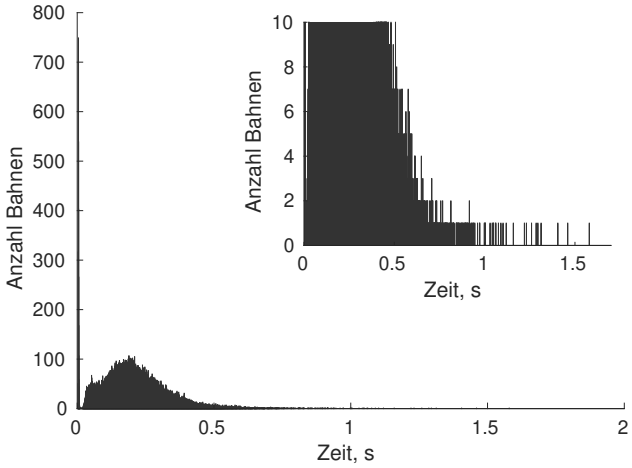


Bild 16. Durchlaufzeiten bei 4 Hindernissen

In Bild 15 und Bild 16 sind die Durchlaufzeiten für die anderen beiden Konstellationen abgebildet. Hier ist zu erkennen, dass der Verlauf der zweiten Erhebung ebenfalls einer Normalverteilung ähnelt. Diese ist mit abnehmender Hindernisanzahl zu geringeren Durchlaufzeiten hin verschoben.
Aus den Diagrammen wird ersichtlich, dass fast alle Bahnen unter der Zeit von zwei Sekunden geplant werden und somit die Planungszeit nicht verletzten. Bei der Konstellation mit 150 Hindernisen beträgt die Zeit bei einer Hindernisanordnung mehr als zwei Sekunden, bei 30 Hindernissen sind es zwei Anordnungen, bei denen die Zeit mit 3,5 s und 9,5 s deutlich über den zwei Sekunden liegt. Bei der Konstellation mit vier Hindernissen wird bei allen Fällen die maximale Planungszeit eingehalten. Die Überschreitungen kommen dann vor, wenn die Hindernisse besonders schlecht zueinander stehen, sodass der Algorithmus nicht schnell genug eine Lösung findet.

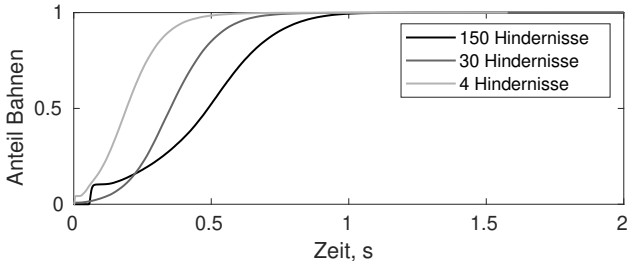


Bild 17. Zeiten

In Bild 17 ist der Anteil der generierten Bahnen in Abhängigkeit der Durchlaufzeit für alle Konstellationen dargestellt. Der Verlauf ähnelt der Verteilungsfunktion einer Normalverteilung. Es ist zu erkennen, dass der Großteil der Bahnen, vor allem in den Konstellationen mit weniger Hindernissen, bereits nach einer Sekunde generiert ist.

In Bild 18 sind zur Veranschaulichung die langsamste (Bild 18c), die schnellste (Bild 18a) und eine Bahn mit der durchschnittlichen Planungszeit (Bild 18b) für 30 Hindernisse dargestellt.

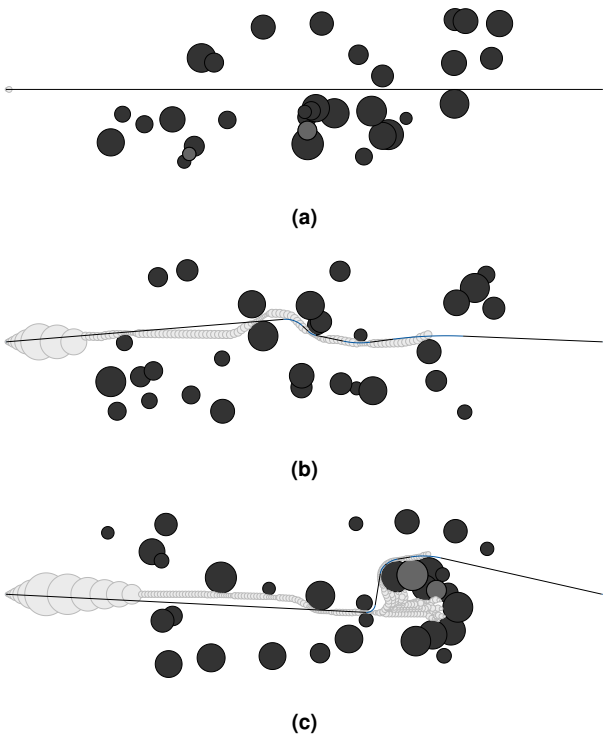


Bild 18. 30 Hindernisse

In der oben dargestellten und am schnellsten generierten Bahn ist zu erkennen, dass direkt vom

CC BY 4.0

Startpunkt aus eine hindernisfreie Verbindung zum Zielpunkt existiert, sodass der COWEX-Algorithmus keine weiteren Kreise aufspannen muss. Die Durchlaufzeit beträgt in diesem Fall 0,01 s. Bei der Bahn mit einer durchschnittlichen Planungszeit, die in der Mitte dargestellt ist, muss mehreren Hindernissen ausgewichen werden. Bereits vorm Ende des Hindernisgebietes wird eine kollisionsfreie Verbindung zum Zielpunkt gefunden. Die Planungszeit liegt in diesem Beispiel bei 0,36 s. In dem Beispiel mit der längsten Planungszeit, die 1,76 s beträgt, ist zu erkennen, dass der Algorithmus Schwierigkeiten mit konkaven Hinderniskonstellationen hat. Hier werden einige Pfade getestet, bis die konkave Konstellation umgangen werden kann.
Die Analyse der erstellten Bahnen ergab, dass für alle drei Konstellationen alle 100.000 Bahnen fliegbar sind. Hier wurden alle Elemente der Bahnen auf Stetigkeit, Richtungen, Längen und Einhaltung des minimalen Radius geprüft.
Lediglich für fünf von 300.000 Bahnen benötigte die Bahnplanungsmethode länger als zwei Sekunden. In diesem Fall würde der in der Flugführungsarchitektur implementierte Sicherheitsmechanismus, der in [10] näher beschrieben ist, eingreifen.

4. ZUSAMMENFASSUNG

In diesem Beitrag wurde eine Bahnplanungmethode für Kippflügelflugzeuge vorgestellt, die flugmechanische Beschränkungen berücksichtigt und Hindernisse umgeht. Dazu wird der COWEX-Algorithmus verwendet. Dieser basiert auf der zufallsbasierten, wellenartigen Expansion von Kreisen die zur Exploration des hindernisfreien Raums genutzt werden. Durch Modifikationen expandieren die neu aufgespannten Kreise im Algorithmus nicht mehr zufallsbasiert wellenartig. Stattdessen berücksichtigt der Algorithmus bei der wellenartigen Expansion die letzte bekannte Bahnrichtung sowie den minimalen Kurvenradius, sodass durch die Verbindung der Kreismittelpunkte immer eine fliegbare Bahn entsteht. Nach der Bestimmung der fliegbaren und hindernisfreien Bahn, wurde diese weiter optimiert, indem sie zum einen durch Anwendung eines A*-Algorithmus verkürzt wurde und zum anderen ein Algorithmus zum Glätten der Bahn angewendet wird. In einem Validierungsabschnitt konnten durch Monte-Carlo-Simulationen auf der Zielhardware die Funktionalität und Performanz der Bahnplanungmethode gezeigt werden.

5. DANKSAGUNG

Ein besonderer Dank gilt dem Bundesministerium für Bildung und Forschung (BMBF) für die Förderung sowie dem Projektträger VDI für die Betreuung und Koordination des Projekts FALKE.

Kontaktadresse:

barz@fsd.rwth-aachen.de

Literatur

[1] JT Economou, G Kladis, A Tsourdos, and BA White. Uav optimum energy assignment using dijkstra's algorithm. In *2007 European Control Conference (ECC)*, pages 287–292. IEEE, 2007.

[2] Ji Li and Xiu-xia Sun. A route planning's method for unmanned aerial vehicles based on improved a-star algorithm [j]. *Acta Armamentarii*, 7:788–792, 2008.

[3] Ozgur Koray Sahingoz. Generation of bezier curve-based flyable trajectories for multi-uav systems with parallel genetic algorithm. *Journal of Intelligent & Robotic Systems*, 74(1-2):499–511, 2014.

[4] Ioannis K Nikolos, Eleftherios S Zografos, and Athina N Brintaki. Uav path planning using evolutionary algorithms. In *Innovations in intelligent machines-1*, pages 77–111. Springer, 2007.

[5] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[6] Markus Rickert, Arne Sieverling, and Oliver Brock. Balancing exploration and exploitation in sampling-based motion planning. *IEEE Transactions on Robotics*, 30(6):1305–1317, 2014.

[7] I Barz, P Hartmann, and D Moormann. Near-term flight path adaption for tilt-wing aircraft obstacle avoidance. *EuroGNC. Milan, Italy*, 2019.

[8] Philipp Hartmann. *Vorausschauende Flugbahnregelung für Kippflügelflugzeuge*. PhD thesis, RWTH Aachen University, Aachen, Germany, 2017.

[9] P Hartmann, S Seitz, I Barz, and D Moormann. Berücksichtigung von flugmechanischen Beschränkungen bei der Bahnregelung und Bahnplanung von Kippflügelflugzeugen. 2020.

[10] S Seitz, P Hartmann, I Barz, and D Moormann. Fehlertolerante Architektur für die automatische Flugführung von Kippflugelflugzeugen. 2019.

CC BY 4.0