

# Lokalisierung eines Miniaturschiffs durch Distanzsensoren

Thorben Schnirpel

Hochschule für Angewandte Wissenschaften Hamburg

Hamburg, Deutschland

thorben.schnirpel@haw-hamburg.de

**Abstract**—Diese Arbeit beschäftigt sich mit der Lokalisierung eines Miniaturschiffs. Die Lokalisierung basiert hierbei lediglich auf den Messungen von Distanzsensoren und einer vorher bekannten Karte der Umgebung. Hierfür wird ein Partikelfilter mit unterschiedlichen Ansätzen des Resamplings eingesetzt. Die umgesetzten Lösungen werden anhand von Referenzdaten analysiert. Hierbei wird auf die Genauigkeit und den Rechenaufwand dieser Lösungen eingegangen. Die Stärken und Schwächen der Ansätze werden analysiert und veranschaulicht.

**Index Terms**—Mapping, Localization, Partikelfilter

## I. EINLEITUNG

Für diese Arbeit wird ein Miniaturschiff verwendet, welches im Rahmen unterschiedlicher Projekte an der Hochschule für Angewandte Wissenschaften Hamburg entstanden ist. Das Miniaturschiff wurde im Maßstab 1:80 entwickelt und vereint unterschiedliche Sensorik. Zu dieser Sensorik des Schiffs zählt eine Kamera, ein Lidar, eine inertielle Messeinheit (IMU) und elf Distanzsensoren. Für die angestrebte Lokalisierung des Miniaturschiffs, sollen nur die elf Distanzsensoren verwendet werden. Hierfür wurden diese rund um das Schiff in unterschiedlichen Ausrichtungen befestigt. Die verwendeten Distanzsensoren können jeweils nur in eine Richtung eine Entfernung messen und weisen in ihren Messungen teilweise starkes Rauschen auf. Vorteile dieser Sensoren sind der kleine Formfaktor und die geringen Anschaffungskosten.

Das Problem, welches diese Arbeit versucht anzugehen, besteht darin, eine ausreichend gute Lokalisierung des Miniaturschiffs in der gegebenen Karte zu realisieren und das basierend auf wenigen und teils verrauschten Daten der Umgebung. Das Ziel dieser Arbeit besteht also in der Lokalisierung des Miniaturschiffs. Oft wird die Herausforderung der Lokalisierung durch das Bestimmen der Position und der Ausrichtung des Systems beschrieben. Diese Herausforderung wird umso größer, je weniger und ungenauer die verfügbaren Daten für diese Aufgabe sind. Aus diesem Grund ist die Lokalisierung des Miniaturschiffs mit Hilfe der verwendeten Distanzsensoren eine besonders große Herausforderung.

Ein Ansatz, um sich auch mit verrauschten Sensordaten lokalisieren zu können, wird von [1] durch das Verwenden von Partikelfiltern beschrieben. In [2] wird beschrieben, dass Partikelfilter im Kontext der Lokalisierung oft als

Monte Carlo Localization (MCL) bekannt sind. Für die Lokalisierung durch Partikelfilter wird in den meisten Fällen eine Karte der Umgebung vorausgesetzt. Diese kann parallel zur Lokalisierung durch Simultaneous Localization and Mapping (SLAM)-Ansätze konstruiert werden oder bereits vorher bekannt sein. Mit Hilfe dieser Karte, den Steuerbefehlen und der gemessenen Distanzen soll der Partikelfilter in der Lage sein die Position und die Ausrichtung des Miniaturschiffs zu bestimmen.

Der in dieser Arbeit implementierte MCL-Ansatz basiert auf einer Karte, welche vor der Lokalisierung bekannt ist. Diese Karte wurde mit Hilfe der Referenzdaten und dem 2D Lidar des Miniaturschiffs erzeugt. Die erzeugte Karte ist eine 3D Occupancy Grid Map, welche auf einem Octree basiert.

Insgesamt soll dieser Ansatz das Problem der globalen Lokalisierung und das Kidnapped-Robot-Problem lösen. Diese Probleme werden in [2] kurz beschrieben. Das Miniaturschiff soll also irgendwo innerhalb der Karte ausgesetzt werden können und sich anschließend darin lokalisieren können.

## II. MAPPING

In diesem Abschnitt der Arbeit wird darauf eingegangen, wie die Karte für die Lokalisierung erzeugt wird, was sie darstellt und wie sie es darstellt.

Das Mapping beschäftigt sich mit der Erstellung einer Karte der Umwelt des Fahrzeugs. Die erzeugten Karten unterscheiden sich indem was sie von ihrer Umgebung aufnehmen. Es gibt z.B. Karten, welche nur die besonderen Merkmale der Umgebungen aufnehmen, wie geometrische Formen, andere Fahrzeuge oder die Fahrbahn. Andere Karten versuchen ihre Umgebung möglichst detailliert darzustellen. Hierzu gehören die Occupancy Grid Maps, welche die gesamte Umgebung in gleichgroße Vierecke bzw. Würfel teilt und diesen Teilen jeweils eine Wahrscheinlichkeit zuordnet, wie wahrscheinlich es ist, dass diese Teile belegt sind. Occupancy Grid Maps können hierbei unterschiedliche Dimensionen abdecken, so gibt es 2D, 3D aber auch 2.5D Occupancy Grid Maps. Bei den 2.5D Occupancy Grid Maps wird die vertikale Achse nur zum Teil, oft nur einige Zentimeter, abgedeckt. Für diese Arbeit soll genau so eine Occupancy Grid Map erstellt

werden. Für den Anwendungsfall unseres Miniaturschiffs würde eine 2.5D Occupancy Grid Map ausreichen. Eine Abdeckung der vertikalen Achse sollte durch das Rollen und Stampfen des Miniaturschiffs gegeben sein. Außerdem weist das Becken, in dem das Miniaturschiff fahren soll, einen Tidenhub von circa 5 Zentimetern auf.

Um eine solche Occupancy Grid Map zu erstellen, wurde ein 3D Mapping Framework, welches auf Octrees basiert, genutzt. Dieses Framework wird in [3] näher beschrieben. In diesem Framework wird die Umgebung als Octree verwaltet. Hierfür wird der gesamte zu überwachende Bereich in acht gleichgroße Würfel geteilt. Diese Würfel enthalten jeweils wieder acht gleichgroße Würfel. Dies wird bis zu einer bestimmten Tiefe weitergeführt. Die Standardtiefe ist mit 16 festgelegt, kann aber auch bis auf 32 erweitert werden. Die Auflösung der Umgebung lässt sich über das Framework festlegen, so lässt sich z.B. durch eine Auflösung von 1 cm und einer Tiefe von 16 schon eine Fläche von ca. 430.000 m<sup>2</sup> abdecken. Die einzelnen Würfel bzw. Zellen der Karte können unterschiedliche Zustände repräsentieren. Diese Zustände umfassen belegt (occupied), frei (free) und unbekannt (unknown).

Damit man einer Karte die Entfernungsdaten eines Lidars hinzufügen kann, werden bestimmte Funktionen von dem Framework gestellt, so werden z.B. Funktionen für das Hinzufügen von 3D-Punktwolken gegeben. Da das Miniaturschiff nur ein 2D-Lidar verwendet, werden die 2D-Scans des Lidar mit Hilfe der inertialen Messeinheit in eine 3D-Punktwolke transformiert. Bei der Erstellung der 3D-Punktwolke ist außerdem die Positionsveränderung der Plattform während des Laserscans und die Position der Plattform hilfreich. Die Positionsveränderung der Plattform während eines Laserscans sollte bei der Erstellung der 3D-Punktwolke berücksichtigt werden, da das Erstellen eines Laserscans eine gewisse Zeit in Anspruch nimmt, in der sich die Plattform weiterbewegt und somit die einzelnen Entfernungsmessungen eines Laserscans verfälschen kann. Die Position der Plattform ist für die Erstellung der 3D-Punktwolke nötig, da die Laserscans nur einzelne Entfernungsmessungen sind und am Ende einzelne 3D-Punkte in der 3D-Punktwolke darstellen sollen. Hierfür wird der Startpunkt der Messung, also die Position der Plattform bzw. des Lidars benötigt.

Eine 3D-Punktwolke wird in die Karte integriert, indem ein Strahl von der Position des Lidars zu jedem Punkt der Punktwolke gebildet wird. Die Zellen, in die die Endpunkte fallen, werden als occupied markiert. Die Zellen, durch die ein Strahl führt, werden als free markiert. Alle Zellen die nicht free oder occupied sind, sind unknown. Zu diesen Zellen sind also noch keine Zustandsinformationen bekannt. Um den Zustand einer Zelle festlegen zu können, ist es wünschenswert möglichst viele Messungen zu erfassen, welche einen Zustand bestätigten können. Belegte Zellen können im Verlauf des Mappings auch wieder zu nicht belegten Zellen werden und andersrum. Hierfür werden untere und obere Grenzen,

sowie ein Threshold für die Wahrscheinlichkeit der Belegung festgelegt.



(a)



(b)

Fig. 1: Testaufbau im Trackingsystem

Um mit Hilfe dieses Mapping Frameworks zum Ende dieser Arbeit die Ansätze der Lokalisierung bewerten zu können, wurde sich eine statische Umgebung mit Referenzdaten zur Hilfe genommen. Diese Umgebung besteht aus einem Bereich der durch ein Trackingsystem überwacht wird. Dieses Trackingsystem liefert genaue Positionen und Ausrichtungen von bestimmten Markern. Diese Marker wurden für diesen Versuch fest an der Miniaturschiffsplattform angebracht. In dem Bereich des Trackingsystems wurden außerdem einige Konstruktionen aufgebaut. Diese sollen in die Karte aufgenommen werden und später bei der Lokalisierung helfen. In Abbildung 1 ist der Trackingbereich mit dem Testaufbau zu sehen. In 1b ist außerdem ein Wagen (oben links) zu sehen, auf dem das Miniaturschiff durch diesen Testaufbau geschoben wurde. Hier ist zu erwähnen, dass das Schieben des Miniaturschiffs durch den Testaufbau nur geringe Auswirkungen auf das normale Verhalten des Rollen und Stampfen des Miniaturschiffs hat. Die Erfahrung aus den Fahrten im Wasser zeigten, dass das Rollen und Stampfen in den meisten Fällen zu vernachlässigen war, weshalb für das Testen der Lokalisierung auf das Nach-

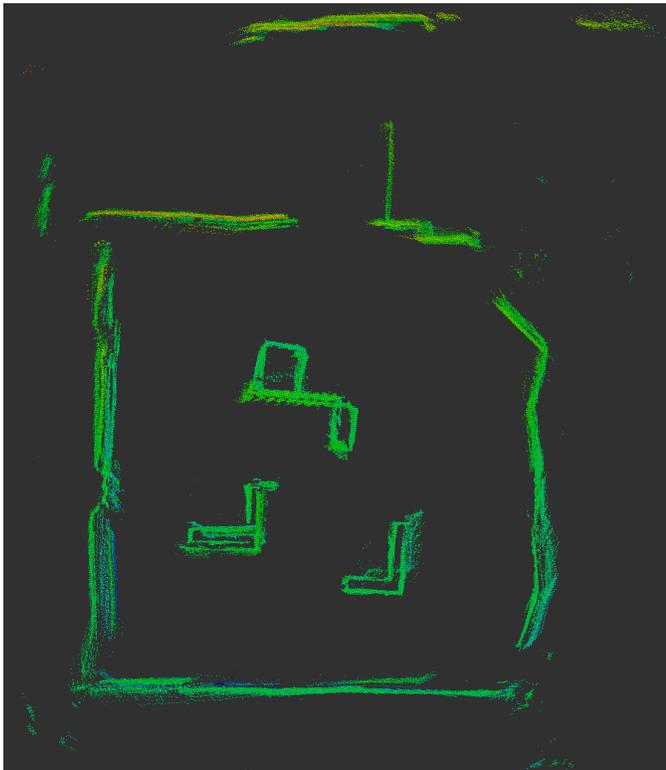


Fig. 2: 3D Occupancy Grid Map des Testaufbaus

bilden der Bewegung im Wasser, etwa durch Tragen des Miniaturschiffs, verzichtet wurde.

In Abbildung 2 ist die, durch das Mapping Framework erstellte, 3D Occupancy Grid Map zu sehen. Die Karte wurde mit einer Auflösung von 1 cm erzeugt. Die für die Erstellung nötigen Positions- und Ausrichtungsdaten wurden vom Trackingsystem generiert. Dargestellt werden nur die Zellen mit dem occupied-Zustand. Die Farbunterschiede der dargestellten Zellen kodieren die Position auf der vertikalen Achse. Es ist zu sehen, dass nur ein geringer Teil der vertikalen Achse des Aufbaus abgedeckt wird. Außerdem ist bemerkbar, dass alle Konstruktionen aus 1a in die Karte übernommen wurden. So sind z.B. zwei L-Formen zu erkennen.

Die Karte aus Abbildung 2 wird im Folgenden als Basis für die Lokalisierung dienen.

### III. LOCALIZATION

Wie bereits beschrieben ist das Ziel dieser Arbeit die Lokalisierung des Miniaturschiffs innerhalb einer Karte mit Hilfe der elf Distanzsensoren. Das Ergebnis dieser Lokalisierung soll aus einer Position innerhalb dieser Karte und einer Ausrichtung bestehen. Die Position besteht hierbei aus einer X-, Y- und Z-Koordinate (X, Y, Z). Die Ausrichtung besteht aus der Drehung um die Vertikalachse ( $\theta$ ). Um dieses Ziel zu erreichen wird ein MCL-Ansatz verfolgt. Hierfür wird die Karte aus Abschnitt II und ein Partikelfilter genutzt. Dieser Partikelfilter wird im Folgenden näher beschrieben.

Nach [4] ist das Ziel eines Partikelfilters das Verfolgen einer Wahrscheinlichkeitsverteilung, welche Information über den Zustand eines dynamischen System gibt. Hierbei kann der Zustandsraum durch zufällig generierte Proben, oder auch Partikel, erkundet werden. Die Wahrscheinlichkeitsverteilung wird durch diese Partikel und ihre zugehörigen Gewichte approximiert. Ein Partikel stellt also einen möglichen Zustand des Systems dar. In Partikelfiltern werden oft drei unterschiedliche Schritte durchgeführt. In [4] und [5] werden diese Schritte als Partikelausbreitung, Gewichtsrechnung und Resampling beschrieben.

Bei der Partikelausbreitung geht es darum, neue Partikel aus den bisherigen Partikeln zu bilden, diese aber anhand eines Modells in der Zeit fortschreiten zu lassen. Angewendet auf ein Lokalisierungsproblem, werden hierbei z.B. alle Partikel anhand eines Bewegungsmodells und weiteren Informationen, wie Steuersignalen und Odometriedaten, angepasst. In [2] wird beschrieben, dass dieser Schritt oft als Ansteuerung des Bewegungsmodells (Motion Model) betrachtet wird.

Bei der Gewichtsrechnung wird jedem Partikel eine Wahrscheinlichkeit zugewiesen. Hierfür werden aktuelle Sensordaten verwendet. Die zugewiesene Wahrscheinlichkeit entspricht dann der Wahrscheinlichkeit, dass diese Sensormessung auftritt, wenn der Zustand dem des Partikels entspricht. Dieser Schritt wird von [2] als Messmodell (Measurement Model) beschrieben. In [6] wird hierfür der Begriff Sensor Update verwendet.

Beim Resampling werden aus einer Menge von Partikeln eine andere Menge von Partikeln gebildet. Hierbei werden aus der alten Menge Partikel proportional zu ihrem Gewicht in die neue Menge übernommen. Die neue Menge von Partikeln kann also mehrere identische Partikel aus der alten Menge enthalten.

#### A. Partikelfilter

In diesem Abschnitt soll der für diese Arbeit verwendete Partikelfilter beschrieben werden. Es wird auf den Zustandsraum, das Motion Model sowie das Measurement Model eingegangen. Außerdem werden die verwendeten Resampling-Ansätze beschrieben.

1) *Zustandsraum*: Bei Partikelfiltern ist es besonders wichtig sich Gedanken über den Zustandsraum zu machen, denn dieser soll am Ende Informationen über den Zustand des Systems liefern. Der Zustandsraum sollte hierbei möglichst realistisch den Zustand des Systems beschreiben können. Hier liegt aber auch ein großes Problem des Partikelfilters, denn je größer der Zustandsraum wird, desto größer wird auch der Rechenaufwand. In [2] wird hierfür beschrieben, dass die Anzahl der nötigen Partikel, um eine Schätzung für den Zustand abgeben zu können, exponentiell zur Dimension des Zustandsraumes steigt. Mit steigender Dimension des Zustandsraumes steigt also auch der Rechenaufwand der Ansteuerung des Motion Models und des Measurement Models exponentiell. Dies wird auch von [4] als großes Problem des Partikelfilters angeführt. Oft werden Partikelfilter zur Lokalisierung in einer Ebene eingesetzt, da sich der

Zustandsraum hier auf drei Dimensionen eingrenzen lässt, die Position als X- und Y-Koordinate und die Ausrichtung als Gierwinkel. Von [2] wird aber auch angeführt, dass es bereits Ansätze gibt die mit mehr als 100.000 Dimensionen klar kommen, wie der FastSLAM-Ansatz aus [7]. Bei diesem Ansatz besitzt jedes Partikel mehrere eigene Kalman-Filter.

Für diese Arbeit ist jedoch kein großer Zustandsraum nötig. Der erste Gedanke würde auf einen Zustandsraum mit X-, Y- und Z-Koordinate, sowie einem Roll-, Nick- und Gierwinkel deuten. In [8] wird für die Oberflächenpositionierung eines Schiffs jedoch nur ein Zustandsraum, welcher eine X-Koordinate, eine Y-Koordinate sowie den Gierwinkel umfasst, genutzt. Wie bereits im Abschnitt II erwähnt, wurde festgestellt, dass das Miniaturschiff nur wenig rollt und stampft. Aus diesem Grund wird auch hier auf den Roll- und Nickwinkel verzichtet. Auf die Z-Koordinate wird jedoch nicht verzichtet, da im Fahrbecken ein gewisser Tidenhub vorhanden ist. Insgesamt wird also ein Zustandsraum verwendet, welcher die X-, Y- und Z-Achse, sowie den Gierwinkel umfasst. Der Zustand eines Partikels ( $m$ ) wird im Folgenden als  $x_m$  notiert und ist in (1) dargestellt. Neben dem Zustand ist hier auch noch die Gewichtung ( $\omega$ ) und die normalisierte Gewichtung ( $\omega_n$ ) aus dem zweiten Schritt des Partikelfilters enthalten.

$$x^m = \{X^m, Y^m, Z^m, \theta^m, \omega^m, \omega_n^m\} \quad (1)$$

2) *Initialisierung*: Bevor ein Schritt des Partikelfilters ausgeführt werden kann, muss erstmal eine Menge an Partikeln bestehen. Diese Menge wird im Initialisierungsschritt erzeugt. Die Werte für die Zustandsvariablen der Partikel werden für jedes Partikel zufällig gewählt. Als Grenzen für die Zustandsvariablen der X-, Y- und Z-Achsen werden jeweils die zugehörigen Dimensionen der Karte gewählt.  $\theta_t$  wird nicht eingegrenzt.

Wie bereits erwähnt, braucht es eine gewisse Anzahl an Partikeln, um einen gegebenen Zustandsraum ausreichend abzudecken. Durch den für diese Arbeit gewählten Zustandsraum und die große Karte werden also sehr viele Partikel benötigt. Erste Tests haben gezeigt, dass die zur Verfügung stehende Rechenleistung für die Abdeckung der kompletten Karte nicht ausreicht. Aus diesem Grund wurde sich dafür entschieden, den Startbereich der Lokalisierung vorzugeben. Somit brauchen die Partikel nur in diesem Bereich initialisiert werden. Um eine Lokalisierung zu ermöglichen reicht nun nur noch ein Bruchteil der ursprünglichen Partikelanzahl. Eine Einschränkung für unseren Anwendungsfall entsteht nicht, da das Miniaturschiff ohnehin nur in bestimmten Bereichen ins Wasser gelassen werden kann. Ein Startbereich von 3m \* 3m wurde getestet und mit 5000 Partikeln bevölkert.

3) *Partikelausbreitung - Motion Model*: In diesem Schritt werden alle Partikel durch ein Motion Model von ihrem Zustand  $x_{t-1}$  in einen neuen Zustand  $x_t$  geleitet. Hierfür werden neben dem alten Zustand im Normalfall Steuersignale und

auch Odometriedaten verwendet. Diese Daten werden zum Zeitpunkt  $t$  mit  $u_t$  beschrieben. Die Partikelausbreitung bzw. die Ansteuerung des Motion Models kann also durch  $p(x_t | u_t, x_{t-1})$  beschrieben werden.

Das Miniaturschiff hat bisher keine sichere Quelle für Odometriedaten. Auch lassen sich die Steuerbefehle in der Testumgebung nicht in eine Zustandsveränderung übertragen, da das Miniaturschiff hier nicht wirklich gesteuert wurde. Aus diesen Gründen wurde sich dafür entschieden, in diesem Schritt die Referenzdaten zur Hilfe zu nehmen. Es entsteht also ein einfaches Motion Model, wie es in (2) gezeigt wird.  $T$  steht hierbei für die Dauer zwischen dem letzten Motion Update und dem aktuellen Zeitpunkt.  $\dot{X}_t$  beschreibt die Geschwindigkeit auf der X-Achse zum Zeitpunkt  $t$ . Gleichmaßen steht  $\dot{Y}_t$  für die Geschwindigkeit auf der Y-Achse und  $\dot{Z}_t$  für die Geschwindigkeit auf der Z-Achse.

$$\begin{aligned} X_{t+1} &= X_t + T\dot{X}_t \\ Y_{t+1} &= Y_t + T\dot{Y}_t \\ Z_{t+1} &= Z_t + T\dot{Z}_t \\ \theta_{t+1} &= \theta_t + T\dot{\theta}_t \end{aligned} \quad (2)$$

Sollte dies in Zukunft ausgetauscht werden, kann das Motion Model aus [8] verwendet werden. Dieses wird z.B. für die Oberflächenpositionierung eines Schiffs genutzt und basiert auf einem Zustandsraum, welcher nur die X- und Y-Achse sowie den Gierwinkel umfasst. Sollte sich doch für ein Zustandsraum von X-, Y- und Z-Achse sowie Roll-, Nick- und Gierwinkel entschieden werden, kann das Motion Model aus [9] genutzt werden. Dies ist allerdings für ein Unterwasserfahrzeug gedacht. Außerdem ist zu den Motion Models des Partikelfilters zu sagen, dass diese Motion Models, nach [8], nicht sehr genau die wirkliche Bewegung des Fahrzeuges bestimmen müssen, da der Partikelfilter nicht auf reiner Koppelnavigation aufbaut und empfiehlt selbst das Verwenden von einfachen Motion Models.

4) *Gewichtsberechnung - Measurement Model*: In diesem Schritt wird jedem Partikel ein Gewicht zugewiesen. Dieses Gewicht spiegelt die Wahrscheinlichkeit wieder eine bestimmte Sensormessung im Zustand des Partikels aufgenommen zu haben. Hierfür sind Sensordaten erforderlich. Diese Sensordaten zum Zeitpunkt  $t$  werden im Folgenden mit  $z_t$  notiert. Die Gewichtsberechnung bzw. das Measurement Model lässt sich also durch  $p(z_t | x_t)$  beschreiben.

Im Abschnitt I wurde bereits beschrieben, dass das Miniaturschiff elf Distanzsensoren in unterschiedlichen Ausrichtungen auf einer Plattform vereint. Bei diesen Distanzsensoren handelt es sich um die GP2Y0E03. Der Erfassungsbereich dieses Sensors reicht ungefähr von 4-50 cm. In Abbildung 3 ist zu sehen, wo die elf Distanzsensoren positioniert und wie sie ausgerichtet sind. Die Kegel sollen die Messrichtungen der Sensoren verdeutlichen, hierbei zeigt die Spitze auf die Position des Sensors. Die Höhe der Kegel hat in dieser Abbildung keine Bedeutung.

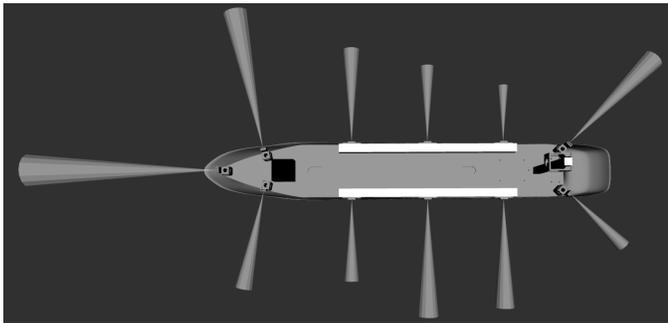


Fig. 3: Positionierung und Ausrichtung der Distanzsensoren auf dem Miniaturschiff

Diese Sensoren werden durch das Miniaturschiff mit 100 Hz ausgelesen. Hierbei ist aufgefallen, dass die Sensoren teilweise Distanzen messen, obwohl sich kein Objekt im Messbereich befindet. Diese Messungen treten unregelmäßig und mit unterschiedlichen Messwerten auf. Die Fehlmessungen erstrecken sich meist nur über einzelne bzw. sehr wenige Samples. Aus diesem Grund wurde sich für die Implementierung eines Medianfilters entschieden. Hierfür werden jeweils die letzten zehn Messungen der Sensoren in einer Liste sortiert. Der Messwert in der Mitte der Liste wird dann als Messwert für den aktuellen Zeitpunkt verwendet.

Für diese Arbeit besteht  $z_t$  nun also aus elf unterschiedlichen Entfernungsmessungen. Um nun das Gewicht eines Partikels bestimmen zu können, wird für jedes Partikel, das Miniaturschiff an die Position und Ausrichtung des Partikels in die Karte projiziert. In dieser Pose wird dann für jeden Sensor, der eine valide Messung liefert, geschaut, ob die Karte eine belegte Zelle auf der Messachse des Sensors aufweist. Eine valide Messung ist eine Messung die in den Messbereich des Sensors fällt. Wenn eine belegte Zelle gefunden wurde, liefert die Distanz vom Sensor zur belegten Zelle ( $d_k$ ) in einer Normalverteilung, welche durch den erfassten Sensorwert ( $d_s$ ) als Erwartungswert und der Varianz des Sensors ( $\sigma_s$ ) ausgezeichnet wird, die Wahrscheinlichkeit für dieses Ereignis. Die Gewichtung eines Partikels besteht nun also aus der Summe aller Wahrscheinlichkeiten der Sensorendaten ( $\omega^m$ ). Dieses Modell ist in (3) dargestellt. Hier steht das  $i$  für den Messwert des  $i$ -ten Sensors der elf Sensoren.

$$\omega^m = \sum_{i=1}^{11} \frac{1}{\sqrt{2\pi\sigma_s^2}} e^{-\frac{(d_{k,i} - d_{s,i})^2}{2\sigma_s^2}} \quad (3)$$

Die Berechnung der normalisierten Gewichtung eines Partikels ( $\omega_n^m$ ) wird in 4 beschrieben. Die Anzahl der Partikel wird mit  $M$  beschrieben.

$$\omega_n^m = \frac{\omega^m}{\sum_{i=1}^M \omega^i} \quad (4)$$

5) *Resampling*: Beim Resampling wird aus einer Menge von Partikeln eine neue Menge von Partikeln gebildet. Um

eine neue Menge von Partikeln zu bilden, werden aus der alten Menge eine gewisse Anzahl von Partikeln auf Basis ihrer Gewichtung gewählt. Hierbei verhält sich die Wahrscheinlichkeit zur Übernahme in die neue Menge proportional zur Gewichtung des Partikels ( $\omega_n$ ). Die Kardinalitäten dieser beiden Mengen müssen hierbei jedoch nicht identisch sein. Die neue Menge enthält nach dem Resampling oft viele gleiche Partikel mit einer hohen Gewichtung. Die Partikel mit einer geringen Gewichtung sterben so aus. Der Resampling-Schritt sorgt somit dafür, dass Partikel die den Zustand des Systems nicht korrekt approximieren wegfallen und reduziert damit die Varianz der Schätzung. Der Effekt, dass die Gewichte der Partikel nur aus sehr hohen und sehr niedrigen Werten bestehen, der beim Weglassen des Resamplings entsteht, wird durch [4] als Entartung beschrieben.

Der Resampling-Schritt bringt jedoch nicht nur Vorteile, denn durch das Resampling kann es laut [4] und [8] zur Probenverarmung (Sample Impoverishment) kommen. Sample Impoverishment bezeichnet den Effekt, dass die Vielfalt in den Partikeln verloren geht. Der Verlust der Vielfalt im Zustand der Partikel kann dazu führen, dass die Partikel den korrekten Zustand des Systems nicht mehr approximieren können. Um diesen Effekt zu verhindern, muss darauf geachtet werden, dass der Resampling-Schritt nicht zu häufig durchgeführt wird. In [8] wird beschrieben, dass dieser Effekt verhindert werden kann, indem der Resampling-Schritt nur durchgeführt wird, sollte sich die zu lokalisierende Plattform eine signifikante Distanz fortbewegt haben. Andere Methoden werden in [4] durch das Verwenden von deterministischen und adaptiven Resampling-Strategien genannt. Bei den deterministischen Ansätzen wird der Resampling-Schritt zu festen Zeitpunkten durchgeführt. Die adaptiven Ansätze führen den Resampling-Schritt auf Basis eines Kriteriums aus, welches die Qualität der Partikelgewichte auswertet. Oft wird hierfür die Varianz der Partikelgewichte genutzt, da diese den Grad der Entartung zeigt. Als ein Beispiel eines solchen Kriteriums wird die effektive Stichprobengröße (Effective Sample Size / ESS) genannt. In [4] wird die ESS, wie in (5) dargestellt, berechnet. Dies ist jedoch nicht die einzige Möglichkeit die ESS zu berechnen oder anzunähern.

$$N_{eff,t} = \left( \sum_{m=1}^M (w_t^m)^2 \right)^{-1} \quad (5)$$

Laut [4] liefern die adaptiven Ansätze bessere Ergebnisse als die deterministischen Ansätze.

Für diese Arbeit wurden zwei unterschiedliche Resampling-Ansätze implementiert. Beide Ansätze setzen auf ein traditionelles sequentielles Resampling. Hierbei werden nur die Gewichte des letzten Zyklus berücksichtigt. Es gibt auch andere Verfahren, die z.B. auf die Historie der Gewichte eingehen oder auch nur auf einem Teil der Partikel sampeln.

Beim ersten Resampling-Ansatz wird eine statische Anzahl an Partikeln verwendet. Die Resampling-Frequenz wird durch

einen adaptiven-Ansatz, durch Berechnung der ESS, realisiert. Die Berechnung der ESS wird, wie in (5) beschrieben, berechnet. Ist die ESS kleiner als ein Threshold, wird ein Resampling-Schritt durchgeführt. Erfahrungsgemäß ist ein Threshold, welcher der Hälfte der Anzahl von Partikeln entspricht, sinnvoll. Der Ablauf des Partikelfilters nach diesem Ansatz sieht, wie im Algorithmus 1 dargestellt, aus. Die Menge der Partikel aus dem vorherigen Zyklus wird mit  $S_{t-1}$  notiert. In Zeile 3 wird durch (5) bestimmt, ob ein Resampling stattfinden muss. Sollte dies der Fall sein, wird dies in Zeile 5 durchgeführt. Zeile 6 und 13 bestehen jeweils aus der Ansteuerung des Motion Models. Zeile 7 und 14 berechnen die Gewichte der Partikel anhand des Measurement Models. Auffällig ist, dass der Resampling-Schritt bereits als erstes durchgeführt wird. Hierbei ist zu beachten, dass dieser Schritt auf der Menge der Partikel aus dem letzten Schritt durchgeführt wird und somit als letzter Schritt des vorherigen Durchgangs zählen kann. Der Unterschied zwischen den Zeilen 6-9 und 13-16 besteht darin, dass die Zeilen 13-16 auf jedes einzelne Partikel der Menge angewendet werden. Die Zeilen 6-9 beziehen sich nur auf die Partikel die durch das Resampling auf Basis ihres Gewichts ausgesucht wurden. Durch die Zeilen 19 bis 21 werden die Partikelgewichte auf 1 normalisiert.

---

**Algorithm 1** Partikelfilter (Ansatz 1)

---

```

1: Inputs:  $S_{t-1}, u_t, z_t$ 
2: Initialization:  $S_t := \emptyset, \alpha = 0$ 
3: if  $0.5 * M > (N_{eff,t}$  from (5) using  $S_{t-1})$  then
4:   for 1 to M do
5:     Sample particle  $x_{t-1}^{(j)}$  from  $S_{t-1}$  based on particle weights
6:     Sample particle  $x_t^{(j)}$  from  $p(x_t|u_t, x_{t-1})$  using  $u_t$  and  $x_{t-1}^{(j)}$  from the previous step
7:      $w_t^{(j)} = p(z_t|x_t^{(j)})$ 
8:      $\alpha := \alpha + w_t^{(j)}$ 
9:      $S_t = S_t \cup \{x_t^{(j)}, w_t^{(j)}\}$ 
10:   end for
11: else
12:   for  $i := 1, \dots, M$  do
13:     Sample particle  $x_t^{(i)}$  from  $p(x_t|u_t, x_{t-1})$  using  $u_t$  and  $x_{t-1}^{(i)}$  from  $S_{t-1}$ 
14:      $w_t^{(i)} = p(z_t|x_t^{(i)})$ 
15:      $\alpha := \alpha + w_t^{(i)}$ 
16:      $S_t = S_t \cup \{x_t^{(i)}, w_t^{(i)}\}$ 
17:   end for
18: end if
19: for  $i := 1, \dots, M$  do
20:    $w_t^{(i)} := w_t^{(i)}/\alpha$ 
21: end for
22: return  $S_t$ 

```

---

Der zweite Resampling-Ansatz, welcher in dieser Arbeit implementiert wurde, ist ein Ansatz, der mit einer adaptiven

Anzahl an Partikeln arbeitet. Vorgestellt wird dieser Ansatz in [1]. Beim Resampling-Schritt wird hierbei nur die nötige Anzahl an Partikeln aus der alten Menge von Partikeln gesampelt. Hierfür wird angenommen, dass sich die Partikel in unterschiedlichen Behältern ( $k$ ) befinden. Der Zustandsraum der Partikel wird also in gleichgroße Behälter geteilt. Hierbei werden die X-, Y- und Z-Achse in 40 cm Schritten geteilt. Der Gierwinkel wird auf vier Bereiche aufgeteilt. Alle Partikel der alten Menge lassen sich so je einem Behälter zuweisen. Ein Behälter kann mehrere aber auch kein Partikel beinhalten.

$$n = \frac{k-1}{2\epsilon} \left( 1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)} z_{1-\delta}} \right)^3 \quad (6)$$

Die nötige Anzahl an Partikeln wird in [1], wie in (6) dargestellt, berechnet. Die Anzahl an Partikeln, welche nötig sind, um den aktuellen Zustand des Systems zu approximieren, wird durch  $n$  beschrieben. Mit  $z_{1-\delta}$  ist das obere  $1-\delta$  Quantil der Standardnormalverteilung  $N(0,1)$  gemeint. Das  $k$  steht für die Anzahl der gefüllten Behälter. Steigt also die Anzahl der unterschiedlichen gefüllten Behälter  $k$ , so steigt auch  $n$ , die nötige Anzahl an Partikeln, um diesen Zustand zu approximieren. Die Anzahl der gefüllten Behälter hängt davon ab, wie weit die Partikel im Zustandsraum verteilt sind. Wie in [1], wurden für  $\epsilon$  Werte von 0.015 bis 0.4 getestet, wobei für die folgende Arbeit 0.015 als Standardwert verwendet wird. Der Wert für  $\delta$  liegt bei 0.99. In [4] wird bei Ansätzen mit einer adaptiven Partikelanzahl das Einschränken dieser Anzahl empfohlen. Aus diesem Grund wird in dieser Arbeit eine obere und eine untere Grenze für die Anzahl der Partikel implementiert.

Um diesen Ansatz mit möglichst wenig Rechenaufwand zu implementieren, wird von [1] ein ähnlicher Ablauf des Partikelfilters, wie im Algorithmus 2 beschrieben.  $n$  steht für die verwendete Anzahl an Partikeln.  $a$  beschreibt den Normalisierungsfaktor der Gewichte.  $k$  gibt die Anzahl der, durch die gesampelten Partikel, belegten Behälter an. In Zeile 16 steht das  $w_t^{(i)}$ , der linken Seite, für das normalisierte Gewicht der Partikel. Auffällig an diesem Ansatz ist, dass das Motion Model sowie das Measurement Model nur bei den ausgewählten Partikeln Anwendung finden.

Beide Resampling-Ansätze wählen Partikel auf Basis ihrer Gewichtung aus einer Menge aus. Hierfür gibt es mehrere Ansätze. Aus Gründen der Vollständigkeit wird kurz auf das Verfahren, welches für diese Arbeit implementiert wird eingegangen. Dieses Verfahren ist im Algorithmus 3 dargestellt.  $w_{max}$  steht für das maximale vorkommende normalisierte Gewicht innerhalb der Partikelmenge. Als Startindex wird zufällig ein Index eines Partikels aus der Menge aller Partikel  $S_t$  ausgewählt. Die Schrittweite  $\beta$  wird durch eine reelle Zufallszahl im Bereich von 0 und dem doppelten maximal Gewicht beschrieben. Diese obere

---

**Algorithm 2** Partikelfilter (Ansatz 2)

---

```
1: Inputs:  $S_{t-1}, u_t, z_t, \epsilon, \delta$ 
2: Initialization:  $S_t := \emptyset, n = 0, k = 0, \alpha = 0$ 
3: repeat
4:   Sample particle  $x_{t-1}^{(i)}$  with index  $i$  from  $S_{t-1}$  based on
   particle weights
5:   Sample particle  $x_t^{(i)}$  from  $p(x_t|u_t, x_{t-1})$  using  $u_t$  and
    $x_{t-1}^{(i)}$  from the previous step
6:    $w_t^{(i)} = p(z_t|x_t^{(i)})$ 
7:    $S_t = S_t \cup \{x_t^{(i)}, w_t^{(i)}\}$ 
8:    $\alpha := \alpha + w_t^{(i)}$ 
9:   if  $x_t^{(i)}$  falls into empty bin  $b$  then
10:      $k := k + 1$ 
11:      $b :=$  not-empty
12:   end if
13:    $n := n + 1$ 
14: until  $n < (n \text{ from (6)})$ 
15: for  $i := 1, \dots, n$  do
16:    $w_t^{(i)} := w_t^{(i)} / \alpha$ 
17: end for
18: return  $S_t$ 
```

---

Grenze ist wichtig, damit es ab dem Partikel mit der höchsten Gewichtung weitergehen kann. Ansonsten würde der Algorithmus wegen der Zeile 5 nur noch das Partikel mit der maximalen Gewichtung auswählen. Vorstellen lässt sich die Funktionsweise dieses Algorithmus durch einen Strahl. Die Partikel nehmen auf diesem Strahl eine Strecke, die proportional zu ihrer Gewichtung steigt, ein. Es wird ein zufälliger Startpunkt auf diesem Strahl ausgesucht. Die zufällige Schrittweite  $\beta$  gibt dann an, welches Partikel ausgewählt wird. Partikel die eine große Strecke einnehmen werden dementsprechend auch oft ausgewählt. Das Ende des Strahls zeigt wieder zum Anfang des Strahls.

---

**Algorithm 3** Auswahl der Partikel

---

```
1: Inputs:  $S_t, w_{max}$ 
2: Initialization:  $idx = \text{random}(0, M), \beta = 0$ 
3: for 1 to  $M$  do
4:    $\beta = \text{random}(0.0, (2.0 * w_{max}))$ 
5:   while  $\beta > w_n^{idx}$  do
6:      $\beta = \beta - w_n^{idx}$ 
7:      $idx = (idx + 1) \text{ mod } M$ 
8:   end while
9:   pick particle with index  $idx$  from  $S_t$ 
10:  ...
11: end for
```

---

Für diese Arbeit wurden also zwei adaptive Strategien für die Frequenz des Resampling-Schritts implementiert, einer basierend auf der Varianz und einer basierend auf der zurückgelegten Entfernung.

Der Ansatz, welcher auf der zurückgelegten Distanz basiert, ist im Algorithmus 1 und 2 nicht dargestellt. Dieser Ansatz besteht lediglich aus einer Abfrage, ob eine gewisse Distanz zurückgelegt wurde. Die Distanz bezieht sich sowohl auf die Distanz im Raum als auch auf die Distanz der Gierwinkel. Überschreitet diese Distanz einen Threshold werden die Algorithmen überhaupt erst ausgeführt.

Im Resampling-Ansatz mit statischer Anzahl an Partikeln werden beide Strategien verwendet. Im Resampling-Ansatz mit adaptiver Partikelanzahl wird sich auf die Strategie basierend auf der zurückgelegten Distanz beschränkt. Diese Entscheidung wurde getroffen, da der Resampling-Ansatz mit adaptiver Anzahl an Partikeln seinen Vorteil der Performance möglichst behalten soll. Durch das Weglassen der Berechnung des Varianz-basierten Kriteriums für das Resampling konnte hier weiterer Rechenaufwand reduziert werden. Zur Sample Impoverishment sollte es, durch das Beibehalten der Entfernung-basierten Resampling-Strategie, nicht kommen. Außerdem könnte das Nutzen der Varianz-basierten Resampling-Strategie dazu führen, dass für eine lange Zeit mit zu vielen Partikeln gearbeitet wird. Dies kann passieren, wenn die Gewichte der Partikel eine geringe Varianz aufweisen, aber trotzdem sehr viele Partikel für die Approximation genutzt werden.

Beide Ansätze haben ihre Vor- und Nachteile. Würden beide Ansätze jeweils immer die gleiche Anzahl an Partikeln verwalten, so wäre der Ansatz 1 vom Rechen- und Speicheraufwand besser geeignet. Dies liegt daran, dass der Ansatz 2 die Behälter zur Berechnung der nötigen Partikelanzahl speichern und auffrischen muss. Außerdem kann der Resampling-Schritt bei Ansatz 1 in manchen Fällen weggelassen werden. Dieser stellt gegenüber den anderen beiden Schritten jedoch nur wenig Rechenaufwand dar. Sollten beide Ansätze selbst über die Partikelzahl und die Resampling-Frequenz entscheiden können, wäre der Ansatz 2 dem Ansatz 1 im Bezug auf den Rechenaufwand überlegen.

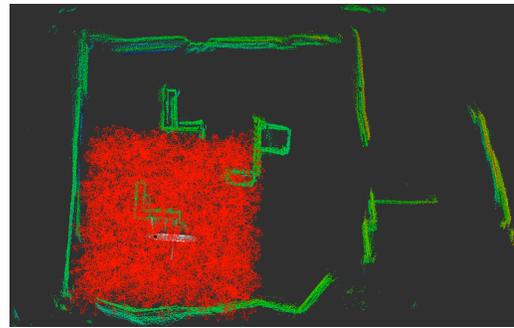
Insgesamt liefert ein Zyklus des Partikelfilters eine Approximation der gewünschte Wahrscheinlichkeitsfunktion  $p(x_t | z_t, u_t)$ , welche durch die Partikel mit ihrer normalisierten Gewichtung repräsentiert wird.

Neben den bisher vorgestellten Varianten, um den Effekt des Sample Impoverishment zu vermindern, werden von [8] noch weitere vorgestellt. Hierzu zählt das Hinzufügen von Rauschen zum Measurement und Motion Model. Dies wurde für diese Arbeit auch berücksichtigt. Das Rauschen wurde jeweils als normalverteilt modelliert. Die Anwendung des vorgestellten Partikelfilters hat gezeigt, dass das Hinzufügen eines Rauschens im Motion und Measurement Model dazu beiträgt, dass der korrekte Zustand des Systems gefunden bzw. approximiert werden kann. Im nächsten Abschnitt wird darauf eingegangen, wie dieses Rauschen auf die Schätzung des Zustands Einfluss nimmt.

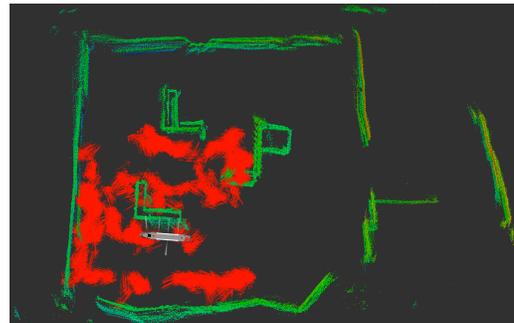
#### IV. ERGEBNISSE DER LOKALISIERUNG

In diesem Abschnitt der Arbeit wird auf das Verhalten des Partikelfilters und die Ergebnisse der Lokalisierung eingegangen. Es wird unter anderem darauf eingegangen, welche Auswirkungen bestimmte Parameter und das hinzugefügte Rauschen auf die Ergebnisse der Lokalisierung haben. Außerdem wird auf die Genauigkeit und den Rechenaufwand der Lösung eingegangen.

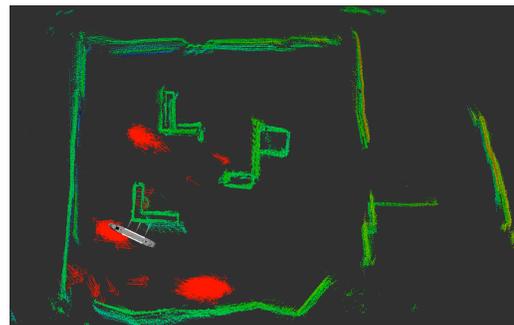
Um sich einen Eindruck verschaffen zu können, wie sich der entwickelte Partikelfilter verhält, wird ein Anlauf des Partikelfilters in Abbildung 4 dargestellt. Es wurde der Partikelfilter aus Algorithmus 2 genutzt. Dargestellt ist die Karte aus Abbildung 2 und das Modell des Miniaturschiffs mit visualisierten Distanzmessungen aus Abbildung 3. Die Länge der Kegel entsprechen in diesem Fall den gemessenen Distanzen. Die Position des Schiffsmodells stimmt mit der richtigen Position des Miniaturschiffs überein. Außerdem sind die einzelnen Partikel als rote Pfeile dargestellt. Die Pfeile repräsentieren durch ihre Position die X-, Y- und Z-Koordinate des Partikels und durch die Richtung des Pfeils den Gierwinkel. Die Abbildung 4a zeigt die Partikelmenge kurz nach der Initialisierungsphase. Alle Partikelzustände wurden zufällig innerhalb des Startbereichs initialisiert. Der Startbereich umfasst in diesem Beispiel einen Bereich von  $3\text{m} \times 3\text{m}$ . Als obere Grenze wurde eine Partikelanzahl von 5000 Partikeln gesetzt. Diese wurde zur Initialisierungsphase auch erreicht. Als untere Grenze wurden 100 Partikel konfiguriert. In Abbildung 4b ist die Menge der Partikel nach einem Durchlauf des Algorithmus zu sehen. Auffällig ist, dass sich viele kleine Gruppen von ähnlichen Partikeln gebildet haben. Dies liegt am Resampling-Schritt. Es wurden Partikel mit hohem Gewicht in die neue Menge übernommen. Durch das hinzugefügte Rauschen zu den gewählten Partikeln, ergibt sich ein Partikel mit ähnlichem Zustand. Zu sehen ist, dass die meisten Partikel so orientiert und positioniert sind, dass sie auf ihrer rechten Seite im gemessenen Abstand zur Karte liegen. In dieser Abbildung ist zudem eine Distanzmessung zu sehen, welche eigentlich nicht möglich sein sollte. Diese Messung ist mittig auf der linken Seite des Miniaturschiffs zu sehen. Dies ist ein Grund, weshalb der bereits vorgestellte Medianfilter implementiert wurde. In Abbildung 4c sind die Partikel nach dem siebten Zyklus des Partikelfilter-Algorithmus zu sehen. Aus den vielen kleinen Gruppen sind nur noch drei größere Gruppen entstanden. In dieser Sequenz von Abbildungen ist gut zu erkennen, dass der Partikelfilter bis zu diesem Zeitpunkt nicht genau eingrenzen kann, wo das Miniaturschiff liegt. Besonders bei gleichen Strukturen, wie den beiden L-Strukturen, ist dies schwer. Das liegt daran, dass das Miniaturschiff hier auch gleiche Entfernungen messen würde. Es kann also nicht bestimmt werden, ob sich das Miniaturschiff mit einer höheren Wahrscheinlichkeit bei der unteren oder der oberen L-Struktur befindet. Erst als sich das Miniaturschiff in die Richtung eines anderen Objekts bewegt hat, konnte dieses Problem gelöst werden. Diese Situation



(a) Partikelverteilung nach der Initialisierungsphase (Anzahl der Partikel: 5000)



(b) Partikelverteilung nach dem ersten Zyklus (Anzahl der Partikel: 5000)



(c) Partikelverteilung nach dem siebten Zyklus (Anzahl der Partikel: 1167)



(d) Partikelverteilung nach dem sechzehnten Zyklus (Anzahl der Partikel: 184)

Fig. 4: Testlauf des Partikelfilters nach Algorithmus 2

ist dann in Abbildung 4d eingetroffen. Es ist nur noch eine Gruppe von Partikeln zu sehen. Insgesamt fällt auf, dass sich die Anzahl der Partikel in kurzer Zeit relativ schnell reduzieren lässt. Zu Beginn waren es noch die maximale Anzahl von 5000 Partikeln. Bereits nach sieben Zyklen konnte die Anzahl auf 1176 Partikel reduziert werden. Nach sechzehn Zyklen waren es sogar nur noch 184 Partikel.

Der Rechenaufwand eines Zyklus steigt linear mit der Anzahl der Partikel. Da durch den Algorithmus 2 die Anzahl der Partikel sehr schnell auf eine kleine Anzahl von Partikeln begrenzt werden kann, wird der Unterschied im Rechenaufwand zwischen den beiden Algorithmen schnell sehr deutlich. Aus diesem Grund wurde die Analyse des Rechenaufwands in dieser Arbeit nicht weiter thematisiert und für weitere Arbeiten offen gelassen. Außerdem hätte die Zeit die ein Zyklus in Anspruch nimmt wenig Aussagekraft, da diese sehr von der verwendeten Berechnungshardware abhängt. Interessant in diesem Zusammenhang ist jedoch, ob es durch das Einsparen von Rechenaufwand zum Verlust der Genauigkeit kommt.

Um durch die Menge der Partikel eine Aussage über den Zustand treffen zu können, wird von [6] beschrieben, dass hierfür jeweils der Mittelwert der Zustandsvariablen gebildet werden kann. Die Unsicherheiten lassen sich durch die Varianzen beschreiben.

Die Genauigkeit der Partikelfilter-Ansätze hängt von unterschiedlichen Faktoren ab. Von [2] wird beschrieben, dass durch eine höhere Anzahl an Partikeln auch die Genauigkeit steigt. Zu den anderen beeinflussenden Faktoren gehören noch die Unsicherheiten im System, aber auch die Resampling-Frequenz. Im Folgenden soll auf diese Faktoren und ihre Auswirkung auf die Genauigkeit eingegangen werden.

TABLE I: Konfigurationen des Partikelfilters

Versuch	$n_{min}$	$n_{max}$	$\sigma_s$	$\sigma_{motion}$	$T_{dist}$	$T_{angle}$
1	100	800	0.02	$0.4 * T$	0.1	0.1
2	100	800	0.2	$0.4 * T$	0.1	0.1
3	100	800	0.02	$0.1 * T$	0.1	0.1
4	100	800	0.02	$0.8 * T$	0.1	0.1
5	100	800	0.02	$0.4 * T$	0.3	0.3
6	1000	5000	0.02	$0.4 * T$	0.1	0.1

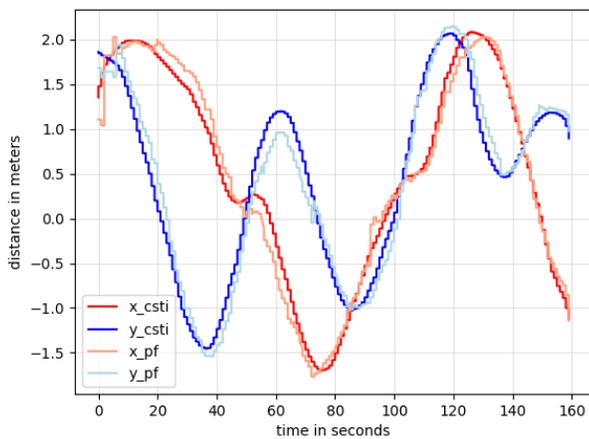
Um die Auswirkungen einzelner Parameter analysieren zu können, wird in jeder Versuchsdurchführung jeweils nur ein Parameter angepasst. Das Ergebnis dieser Versuchsdurchführungen kann dann mit einer Referenzdurchführung verglichen werden. Als Referenz wurde sich für eine Konfiguration, wie sie in Zeile 1 der Tabelle I dargestellt ist, entschieden.  $n_{min}$  und  $n_{max}$  beschreiben die untere und obere Grenze der Partikelanzahl.  $\sigma_{motion}$  gibt die Stärke des Rauschens im Motion Model an. Die Varianz des normalverteilten Rauschens im Motion

Model wird durch einen Anteil des Distanzthresholds  $T_{dist}$  bzw.  $T_{angle}$  bestimmt.  $T_{dist}$  wird in Metern und  $T_{angle}$  im Bogenmaß angegeben.  $\sigma_s$  gibt auch hier die Varianz der Distanzsensoren an.

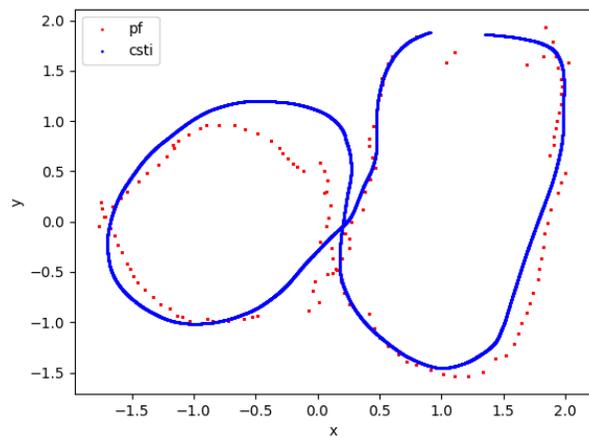
Die Auswirkungen der Parameter auf die vorgestellten Algorithmen werden im Folgenden beschrieben. In Tabelle I sind alle hierfür genutzten Konfigurationen dargestellt.

In Abbildung 5 sind die Ergebnisse aus einem Durchlauf des Partikelfilters mit der Konfiguration der ersten Version zu sehen. Auffällig ist, dass die Werte der Z-Koordinate in den Abbildungen fehlen. Der Grund hierfür ist, dass das Trackingsystem nicht korrekt konfiguriert war. Die XY-Ebene war an manchen Stellen des Trackingbereichs etwas rotiert, somit war es für die Auswertung nicht möglich die Geschwindigkeit auf der Z-Achsen im Motion Model zu berücksichtigen. Dies führte dazu, dass die Partikel die Z-Koordinate nicht richtig schätzen konnten. Aus diesem Grund wurde die Z-Achse in allen folgenden Darstellungen weggelassen. Auf die X- und Y-Koordinate sowie die Ausrichtung hatte diese Fehlkonfiguration keine Auswirkung. In Abbildung 5a sind die gemittelten XY-Positionen der Partikel sowie die XY-Positionen aus dem Trackingsystem über den Verlauf der Versuchszeit zu sehen. Es lässt sich beobachten, dass sich die Positionen in den ersten Sekunden noch sehr stark unterscheiden. Dieser Unterschied in den Positionen ist durch die große Streuung der Partikel im Anfangsbereich zu begründen. Insgesamt lässt sich sehen, dass die Positionsveränderungen des Miniaturschiffs durch den Partikelfilter erkannt werden. Dies ist auch in Abbildung 5b zu sehen. Es ist der Trackingbereich aus der Vogelperspektive dargestellt. Zum Start des Versuchs war das Miniaturschiff ungefähr auf der Position (1.5/1.75). Zum Ende befand sich das Miniaturschiff in der Nähe der Position (1.0/1.75). Es ist zu beobachten, dass der Partikelfilter zum Start des Versuchs, noch stark in der geschätzten Position springt. Dieses Verhalten legt sich jedoch im Verlauf des Versuchs. In Abbildung 5c sind die Gierwinkel des Partikelfilters und des Trackingsystems über den Zeitraum des Versuchs dargestellt. Auch hier ist sichtbar, dass zu Beginn der Gierwinkel des Partikelfilters stark schwankt, bis er sich ungefähr dem Gierwinkel des Trackingsystems nähert. Diese Phase erstreckt sich ungefähr auf die ersten 20 Sekunden.

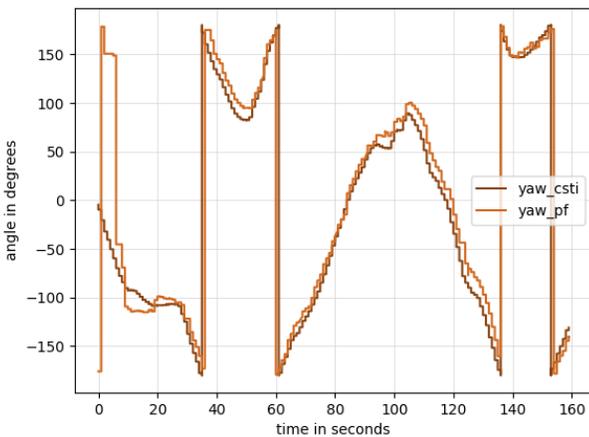
Damit die Genauigkeit des Partikelfilters bewertet werden kann, werden in Abbildung 6 die Abweichungen vom approximierten Zustand des Partikelfilters und Referenzwert aus dem Trackingsystem dargestellt. In Abbildung 6a sind jeweils die euklidischen Abstände der beiden Positionen zu sehen. In Abbildung 6b jeweils die Differenzen der Gierwinkel. In beiden Abbildungen ist das bereits angesprochene Verhalten des Partikelfilters zu Anfang des Versuchs zu sehen. Am Anfang ist die Abweichung relativ hoch, nimmt aber im Verlauf des Versuchs ab. Es ist zu sehen, dass sich die Abweichungen in einem Bereich einpendeln. Um die Genauigkeit eines Ansatzes nun beschreiben zu können, wird für diese



(a) Positionen aus dem Partikelfilter und dem Trackingsystem

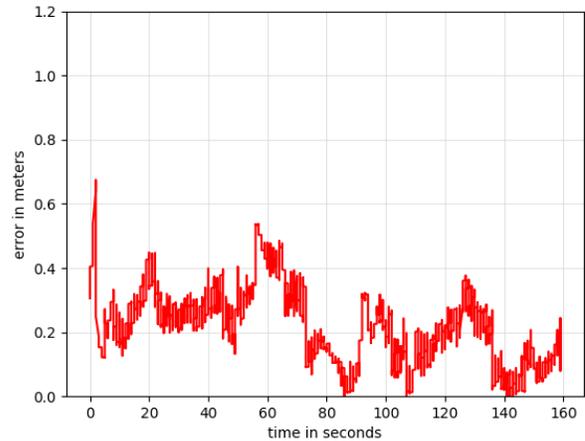


(b) Positionen aus dem Partikelfilter und dem Trackingsystem in der XY-Ebene

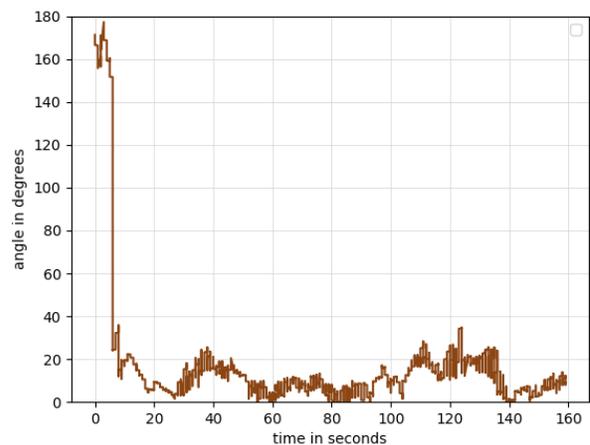


(c) Gierwinkel aus dem Partikelfilter und dem Trackingsystems

Fig. 5: Ergebnisse des 1.Versuchs



(a) Differenzen in den Positionen zwischen Partikelfilter und Trackingsystem

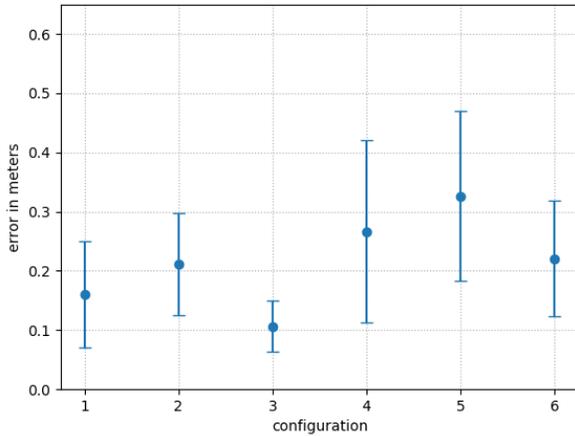


(b) Differenzen in den Gierwinkeln zwischen Partikelfilter und Trackingsystem

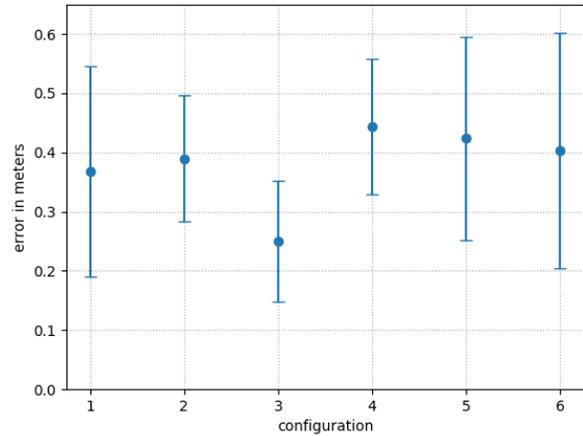
Fig. 6: Abweichungen des 1.Versuchs (Referenzversuch)

Arbeit der Mittelwert und die Standardabweichung auf dem eingedellten Intervall berechnet. Für den weiteren Verlauf der Arbeit wird angenommen, dass sich der Partikelfilter bis zur 80.Sekunde eingedellt hat. Für den Partikelfilter nach Ansatz 2 in der 1.Konfiguration aus Tabelle I ergibt sich so eine mittlere Abweichung von 0,16 Metern in der Position und 10,5 Grad im Gierwinkel.

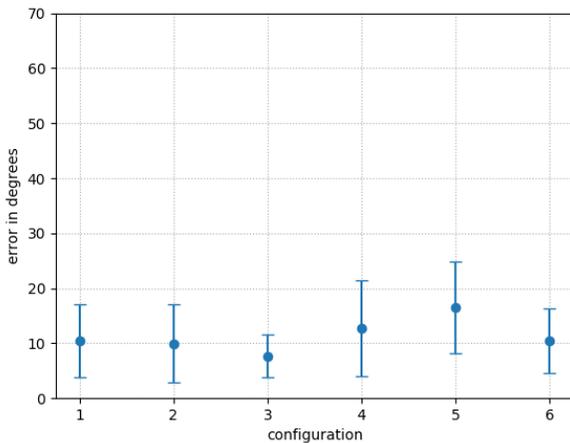
In Abbildung 7 sind nach diesem Prinzip alle Mittelwerte und Standardabweichungen der Positionen und Gierwinkel aller Konfigurationen aus Tabelle I, bei der Versuchsdurchführung des Partikelfilters aus Algorithmus 2, dargestellt. In Abbildung 8 analog dazu die Versuchsdurchführungen des Partikelfilters nach Algorithmus 1. Vergleicht man die Algorithmen mit jeweils gleicher Konfiguration, stellt man fest, dass die Mittelwerte und Standardabweichungen der Abweichungen von Algorithmus 2 nicht größer ausfallen, als die des



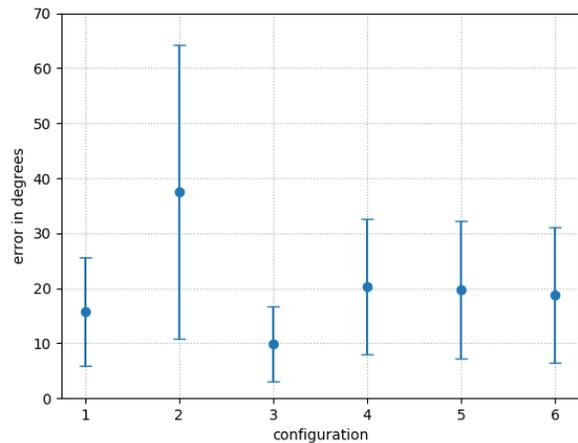
(a) Mittelwerte und Standardabweichungen der Positionsabweichung



(a) Mittelwerte und Standardabweichungen der Positionsabweichung



(b) Mittelwerte und Standardabweichungen der Gierwinkelabweichung



(b) Mittelwerte und Standardabweichungen der Gierwinkelabweichung

Fig. 7: Versuchsläufe des Partikelfilter nach Algorithmus 2 mit den Konfigurationen aus Tabelle I

Fig. 8: Versuchsläufe des Partikelfilter nach Algorithmus 1 mit den Konfigurationen aus Tabelle I

Algorithmus 1, obwohl Algorithmus 1 den Zustand mit deutlich mehr Partikeln approximiert. Dies kann an der zusätzlichen Resampling-Strategie von Algorithmus 1 liegen, welche die Resampling-Frequenz senkt und somit Ungenauigkeiten in den Zustand einführen bzw. den Zustand entarten lassen kann. Insgesamt ist auffällig, dass gleiche Änderungen in der Konfiguration der Partikelfilter auch ähnliche Auswirkungen auf den Mittelwert und die Standardabweichung des Fehlers haben.

Durch Konfiguration 2 bzw. Versuch 2 aus Tabelle I soll untersucht werden, zu welchen Auswirkungen es durch ein stark erhöhtes Rauschen im Measurement Model kommt. In Abbildung 7a und 8a ist zu sehen, dass die mittlere Abweichung der Position deutlich anstieg. Beim geschätzten Gierwinkel fällt der Effekt unterschiedlich aus, beim Partikelfilter nach Algorithmus 2 ist kein großer Unterschied

zur Referenzkonfiguration zu sehen, beim Partikelfilter nach Algorithmus 1 steigt die mittlere Abweichung und dessen Standardabweichung stark an. Durch ein stark erhöhtes Rauschen im Measurement Model kommt es also, wie erwartet, zu einer höheren Abweichung vom Referenzwert.

Die Konfigurationen 3 und 4 aus Tabelle I beschreiben gegenüber der Referenzkonfiguration eine Veränderung im Rauschen des Motion Models. Durch Konfiguration 3 wird ein geringes Rauschen simuliert. Konfiguration 4 hingegen erzeugt ein sehr starkes Rauschen. In der Position und im Gierwinkel ist für beide Ansätze des Partikelfilters zu beobachten, dass die mittlere Abweichung sowie die Standardabweichung bei starkem Rauschen steigt und bei geringerem Rauschen sinkt. Dieses Verhalten ist jedoch nicht linear. Verstärkt man das Rauschen aus Konfiguration 1 um das doppelte, hat es ungefähr die gleiche Auswirkung auf die

mittlere Abweichung, wie die Verstärkung des Rauschens aus Konfiguration 3 um das Vierfache. Dieses Verhalten ist den Positionen und Gierwinkeln beider Algorithmen zu sehen.

Durch Konfiguration 5 wird die Resampling-Frequenz verringert. Dies wird durch das Verdreifachen des Bewegungsthresholds realisiert. Für Algorithmus 2 entspricht die Resampling-Frequenz der Konfiguration 5 also nur noch einem Drittel der Resampling-Frequenz von Konfiguration 1. Bei Algorithmus 1 ist die Auswirkung der Änderung des Bewegungsthresholds nicht direkt abzusehen, da hier noch eine andere Resampling-Strategie angewendet wird. Insgesamt ist jedoch auffällig, dass durch die Erhöhung des Bewegungsthresholds auch die mittlere Abweichung sowie die Standardabweichung für die Positionen und Gierwinkel beider Algorithmen steigt. Eine geringere Resampling-Frequenz wirkt sich also auch negativ auf die Genauigkeit aus. Hier ist jedoch wieder anzumerken, dass eine Resampling-Frequenz wegen des Sample Impoverishment Effekts nicht zu hoch gewählt werden sollte.

Zuletzt soll durch Konfiguration 6 die Auswirkung der Partikelanzahl analysiert werden. Auch in den letzten Konfigurationen wurden teilweise unterschiedliche Anzahlen von Partikeln verwendet. Dies liegt daran, dass für Algorithmus 2 eine minimale und eine maximale Anzahl vorgegeben wurde. Algorithmus 1 hingegen hat nur eine statische Anzahl an Partikeln, welche auf die obere Grenze gesetzt wurde. Im Bereich, in dem die Partikel einen Zustand korrekt approximiert haben, wurden durch Algorithmus 2 also meist nur die untere Grenze als Anzahl der Partikel genutzt. Somit ergab sich in diesen Konfigurationen auch schon ein Unterschied in der Anzahl der Partikel. Dieser Unterschied hat sich, wie in Abbildungen 7 und 8 zu sehen ist, nicht wirklich gezeigt. Im Gegensatz, der Algorithmus mit weniger Partikeln hat den Zustand meist genauer approximiert. Auch durch Konfiguration 6, mit deutlich erhöhter Partikelanzahl, ist kein großer Unterschied in der Genauigkeit festzustellen. Auch hier ist eher ein kleiner Verlust in der Genauigkeit zu verzeichnen.

Diese Konfigurationen decken nicht den vollständigen Bereich des Parameterraums ab, sie sollen lediglich eine grobe Richtung angeben, in wie weit welche Parameter Auswirkungen auf die Ergebnisse haben können.

## V. FAZIT

Diese Arbeit hat gezeigt, dass es möglich ist, sich mit rauschenden und ungenauen Distanzsensoren innerhalb einer gegebenen Karte zu lokalisieren. Hierbei hat sich gezeigt, dass bei der Entwicklung eines solchen MCL-Ansatzes besonders wichtig ist, zu analysieren, welche Genauigkeit nötig ist und wie rechenaufwändig die Lösung sein darf. Es hat sich gezeigt, dass auch schon kleine Erweiterungen des Zustandsraums große Auswirkungen auf den Rechenaufwand haben. Der Rechenaufwand wurde auch sehr stark durch die Anzahl der Partikel geprägt. Dabei wurde festgestellt, dass durch eine erhöhte Anzahl der Partikel keine große Veränderung

der Genauigkeit bewirkt wurde. Aus diesem Grund wird der Partikelfilter nach Algorithmus 2 empfohlen. Zum Start kann eine hohe Anzahl an Partikeln gesetzt werden, damit der Zustand korrekt erfasst wird, anschließend wird die Partikelanzahl auf einen Bruchteil reduziert, sodass der Zustand weiterhin ausreichend genau approximiert werden kann. Eine weitere wichtige Erkenntnis dieser Arbeit ist, dass das zum Motion- und Measurement Model hinzugefügte Rauschen die Genauigkeit des approximierten Zustands zwar deutlich verringert, die Zuverlässigkeit, dass der Partikelfilter auch wirklich den korrekten Zustand approximiert aber deutlich steigert. Außerdem wird dadurch deutlich, dass die Partikelfilter auch mit sehr stark rauschenden bzw. sehr ungenauen Daten im Motion Modell umgehen können und immer noch ein ausreichend genaues Ergebnis liefern. Dies ist besonders für das Projekt des Miniaturschiffs von großer Bedeutung. Zuletzt ist auch noch anzumerken, dass die Resampling-Frequenz einen großen Einfluss auf das Ergebnis und den Erfolg des Partikelfilters hat. Die Auswirkungen auf das Ergebnis wurden bereits im vorherigen Abschnitt anhand von Beispielen gezeigt. Die Auswirkungen auf den Erfolg des Partikelfilters, also das erfolgreiche Approximieren des korrekten Zustandes des Systems ist hiervon aber auch betroffen. Dieses Problem hat sich besonders am Anfang dieser Arbeit als große Herausforderung dargestellt, da hier noch keine Resampling-Strategie angewendet wurde. Der Resampling-Schritt wurde so bei jeder neuen Messung der Distanzsensoren durchgeführt. Dies führte dazu, dass in den meisten Fällen der Effekt des Sample Impoverishments eintrat und somit nicht der korrekte Zustand gefunden wurde. Die richtige Resampling-Frequenz ist also ein entscheidender Faktor für den Erfolg des Partikelfilters. Insgesamt konnte in den getesteten Konfigurationen die Position im Mittel mit einer Abweichung von 10,6 cm genau bestimmt werden. Bei gleicher Konfiguration belief sich die mittlere Abweichung der Orientierung auf nur 7,7 Grad. Insgesamt sollte diese Möglichkeit der Lokalisierung des Miniaturschiffs also für unsere Zwecke ausreichend genau sein. Um diesen Ansatz jedoch wirklich umsetzen zu können, muss noch eine Lösung für das nicht vollständige Motion Model gefunden werden. Außerdem bleibt für die Zukunft noch offen, ob der Ansatz auch in einer realistischeren Umgebung, wie dem Miniatur Wunderland Hamburg, mit detaillierterer Umgebung und somit eventuell auch mit einer stärker veräuschten Karte klarkommt. Zudem kann in zukünftigen Projekten analysiert werden, welche Auswirkungen das Erweitern des Zustandsraums, um z.B. den Roll- und Nickwinkel, auf das Ergebnis der Lokalisierung hat.

## REFERENCES

- [1] D. Fox, "Kld-sampling: Adaptive particle filters," in *Advances in neural information processing systems*, 2002, pp. 713–720.
- [2] S. Thrun, "Particle filters in robotics," in *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2002, pp. 511–518.
- [3] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.

- [4] T. Li, M. Bolic, and P. M. Djuric, "Resampling methods for particle filtering: classification, implementation, and strategies," *IEEE Signal processing magazine*, vol. 32, no. 3, pp. 70–86, 2015.
- [5] T.-c. Li, G. Villarrubia, S.-d. Sun, J. M. Corchado, and J. Bajo, "Resampling methods for particle filtering: identical distribution, a new method, and comparable study," *Frontiers of Information Technology & Electronic Engineering*, vol. 16, no. 11, pp. 969–984, 2015.
- [6] S. Lenser and M. Veloso, "Sensor resetting localization for poorly modelled mobile robots," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2. IEEE, 2000, pp. 1225–1232.
- [7] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit *et al.*, "Fastslam: A factored solution to the simultaneous localization and mapping problem," *Aaai/iaai*, vol. 593598, 2002.
- [8] T. B. Schön, F. Gustafsson, and R. Karlsson, "The particle filter in practice," in *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press, 2011.
- [9] K. M. Fauske, F. Gustafsson, and O. Hegrehaes, "Estimation of auv dynamics for sensor fusion," in *2007 10th International Conference on Information Fusion*. IEEE, 2007, pp. 1–6.
- [10] Z. Pustai and L. Hajder, "Accurate calibration of lidar-camera systems using ordinary boxes," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 394–402.
- [11] Y. Park, S. Yun, C. Won, K. Cho, K. Um, and S. Sim, "Calibration between color camera and 3d lidar instruments with a polygonal planar board," *Sensors*, vol. 14, no. 3, pp. 5333–5353, 2014.

## APPENDIX

### A. Mapping

1) *3D-Mapping*: Um einen Eindruck zu verschaffen, wie eine Occupancy Grid Map aussehen könnte, wenn das Miniaturschiff getragen wird, ist dies in Abbildung 9 dargestellt.

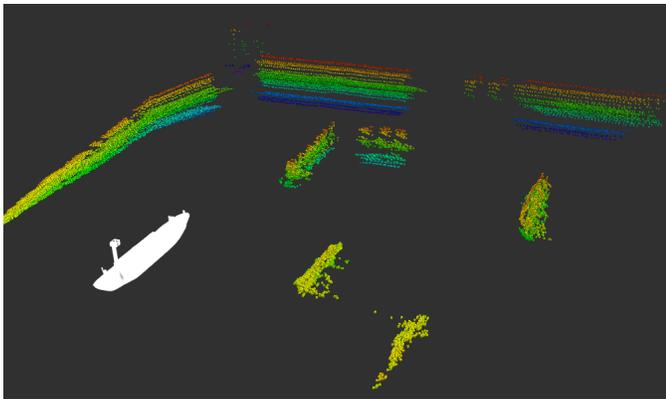
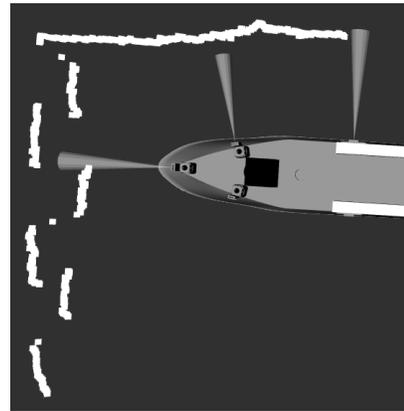
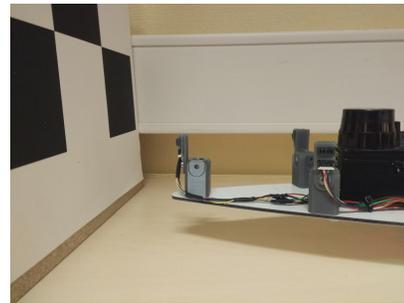


Fig. 9: Unvollständiger Mapping-Anlauf während des Tragens des Miniaturschiffs

2) *Lidar*: Während des Mappings ist ein Effekt aufgetreten, welcher hier kurz beschrieben wird. Wie in den Abbildungen 1a und 1b zu sehen ist, befinden sich innerhalb des Trackingsystems flache Bretter mit unterschiedlich großen Schachbrettmustern. Diese sollten für ein anderes Projekt in der leeren Umgebung des Trackingsystems für wiedererkennbare Merkmale in der Umgebung sorgen. Während des Mappings ist aufgefallen, dass diese ebenen Flächen vom Lidar nicht als Ebenen wahrgenommen werden. Dieser Effekt ist in Abbildung 10a zu sehen. Diese Daten sind durch den



(a)



(b)

Fig. 10: Testaufbau und Lidar-Daten eines Schachbrettmusters

Aufbau aus Abbildung 10b entstanden. Hierbei fällt auf, dass die schwarzen Rechtecke für das Lidar näher erscheinen, als die weißen Rechtecke. In [10] und [11] ist dieser Effekt auch aufgefallen. Beschrieben wird er durch [11] als systematic range-reflectivity-bias. Dieser Effekt hängt also mit dem unterschiedlichen Reflektionsverhalten der beiden Farben zusammen.

Interessant ist dabei auch, dass der Distanzsensord, welcher nach vorne zeigt, auch eine andere Entfernung misst, als das Lidar.

Dieser Effekt ist auch in der erstellten Karte des Trackingbereichs aus Abbildung 2 zu erahnen. Durch die unterschiedlichen Verhaltensweisen der beiden Sensoren und dem eventuell daraus entstehenden Fehler, wurde dieser Effekt kurz erwähnt.