



(10) **DE 10 2017 120 218 A1** 2019.03.07

(12)

Offenlegungsschrift

(21) Aktenzeichen: **10 2017 120 218.8**

(22) Anmeldetag: **01.09.2017**

(43) Offenlegungstag: **07.03.2019**

(51) Int Cl.: **G05D 1/02 (2006.01)**

(71) Anmelder:

Robart GmbH, Linz, AT

(74) Vertreter:

**Westphal, Mussgnug & Partner Patentanwälte mit
beschränkter Berufshaftung, 81541 München, DE**

(72) Erfinder:

**Vogel, Reinhard, Linz, AT; Artes, Harold, Linz, AT;
Freudenthaler, Christoph, Linz, AT; Knittel, Ivo,
Linz, AT**

(56) Ermittelter Stand der Technik:

DE	10 2008 050 206	A1
DE	10 2015 119 865	A1
US	2015 / 0 166 060	A1
EP	2 752 726	A1

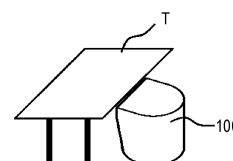
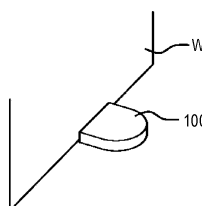
**FASOLA, Juan; RYBSKI, Paul E.; VELOSO,
Manuela M.: Fast goal navigation with obstacle
avoidance using a dynamic local visual model.
In: Proceedings of the VII. Simpósio Brasileiro de
Automação Inteligente, Setembro de 2005, São
Luis. 2005, S. 1-6**

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **BEWEGUNGSPLANUNG FÜR AUTONOME MOBILE ROBOTER**

(57) Zusammenfassung: Es wird ein Verfahren zur Steuerung eines autonomen mobilen Roboters beschrieben, der in einem ersten und mindestens einem zweiten Konturfolgemodus arbeiten kann, wobei in jedem der Konturfolgemodi der Roboter einen im Wesentlichen konstanten Abstand zu einer Kontur einhält, während er sich entlang der Kontur bewegt. Gemäß einem Ausführungsbeispiel weist das Verfahren folgendes auf: das Starten des ersten Konturfolgemodus, wobei der Roboter der Kontur in einer ersten Fahrtrichtung folgt; das Detektieren einer Sackgassen-Situation, in der ein fortgesetztes Folgen der Kontur im ersten Konturfolgemodus nicht kollisionsfrei möglich ist; das Starten eines zweiten Konturfolgemodus, wobei der Roboter der Kontur in einer zweiten Fahrtrichtung folgt; und das Festlegen eines Kriteriums, bei dessen Erfüllung der zweite Konturfolgemodus beendet wird, sowie laufendes Auswerten des Kriteriums während der Roboter im zweiten Konturfolgemodus arbeitet.



Beschreibung**TECHNISCHES GEBIET**

[0001] Die Beschreibung betrifft das Gebiet der autonomen mobilen Roboter, insbesondere die Planung und Durchführung von Bewegungen eines autonomen mobilen Roboters allgemeiner Form.

HINERGRUND

[0002] In den letzten Jahren finden autonome mobile Roboter, insbesondere Serviceroboter, zunehmend Verwendung im Haushaltsbereich, beispielsweise zur Reinigung oder zur Überwachung einer Wohnung. Häufig besitzen die Roboter eine runde Form und eine Antriebseinheit, die es ihnen erlaubt, sich um eine Hochachse zu drehen. Dies vereinfacht die Pfadplanung (Trajektorienplanung) und Steuerung dieser Roboter erheblich, da ihr Rotationsfreiheitsgrad niemals durch angrenzende Hindernisse eingeschränkt wird.

[0003] Aufgrund spezieller Anforderungen an die Funktion des Roboters, kann es wünschenswert sein, von der runden (scheibenförmigen) Form abzuweichen. Beispielsweise kann die an sonst runde Form des Roboters an einer Seite abgeflacht sein, so dass der Roboter mit der flachen Seite parallel zu einer Wand entlang dieser fahren kann. An der flachen Seite kann beispielsweise eine Einigungseinheit angeordnet sein (z.B. eine Bürste), sodass diese möglichst nahe an der Wand geführt werden kann. Auch aus anderen Gründen kann es notwendig oder wünschenswert sein, von einer runden Bauform des Roboters abzuweichen.

[0004] Eine in Bezug auf die Grundfläche des Roboters nicht runde Bauform kann zur Folge haben, dass der Roboter sich nicht in jeder Situation auf der Stelle drehen kann, auch wenn seine Antriebseinheit dies prinzipiell ermöglicht. Wenn wie im oben genannten Beispiel, der Roboter mit seiner flachen Seite sehr nahe an einem Hindernis (z.B. einer Wand) steht, kann der Roboter sich nicht mehr beliebig um seine Hochachse drehen, ohne mit dem Hindernis zu kollidieren. Für die Planung und Bewertung der Bewegungsmöglichkeiten des Roboters ist somit zusätzlich zur Position der Hindernisse und des Roboters im Robotereinsatzgebiet auch die Orientierung des Roboters zu berücksichtigen. Ein Ansatz, mit diesem Problem umzugehen, besteht in der Verwendung von Standardbewegungsmuster für vordefinierte Situationen. Dieser Ansatz ist jedoch unflexibel und fehleranfällig. Außerdem ist es schwierig, alle möglichen Situationen vorherzusehen, in die ein autonomer mobiler Roboter geraten kann. Ein weiterer Ansatz ist die exakte Planung der Bewegung des Roboters also der Änderung der Position und Orientierung (zusammen auch Pose genannt), um von einem Ausgangspunkt

zu einem Zielpunkt zu gelangen. Dies ist jedoch deutlich aufwändiger als für einen runden Roboter, wodurch zum einen die Fehleranfälligkeit bei der Implementierung, als auch der Ressourcenaufwand (Rechenzeit, Prozessorleistung, Speicherbedarf) für die notwendigen Berechnungen steigt.

[0005] Die Erfinder haben sich die Aufgabe gestellt, eine einfache aber robuste Planung für die Bewegung eines autonomen mobilen Roboters beliebiger Form zu ermöglichen.

ZUSAMMENFASSUNG

[0006] Die oben genannte Aufgabe wird durch die Verfahren gemäß Anspruch 1, 16, 28, 45 und 50 gelöst. Verschiedene Ausführungsbeispiele und Weiterentwicklungen sind Gegenstand der abhängigen Ansprüche.

[0007] Es wird ein Verfahren zur Steuerung eines autonomen mobilen Roboters beschrieben, der in einem ersten und mindestens einem zweiten Konturfolgemodus arbeiten kann, wobei in jedem der Konturfolgemodi der Roboter einen im Wesentlichen konstanten Abstand zu einer Kontur einhält, während er sich entlang der Kontur bewegt. Gemäß einem Ausführungsbeispiel weist das Verfahren folgendes auf: das Starten des ersten Konturfolgemodus, wobei der Roboter der Kontur in einer ersten Fahrtrichtung folgt; das Detektieren einer Sackgassen-Situation, in der ein fortgesetztes Folgen der Kontur im ersten Konturfolgemodus nicht kollisionsfrei möglich ist; das Starten eines zweiten Konturfolgemodus, wobei der Roboter der Kontur in einer zweiten Fahrtrichtung folgt; und das Festlegen eines Kriteriums, bei dessen Erfüllung der zweite Konturfolgemodus beendet wird, sowie laufendes Auswerten des Kriteriums während der Roboter im zweiten Konturfolgemodus arbeitet.

[0008] Des Weiteren wird ein Verfahren zur Steuerung eines autonomen mobilen Roboters in einem Konturfolgemodus beschrieben, in dem der Roboter in einem Konturfolgeabstand im Wesentlichen einer Kontur folgt. Gemäß einem Ausführungsbeispiel weist das Verfahren in dem Konturfolgemodus folgendes auf: das Bewerten von mindestens drei verschiedenen Elementarbewegungen anhand mindestens eines vorgebbaren Kriteriums, sowie das Ausführen einer der drei Elementarbewegungen basierend auf deren Bewertung. Die erste der drei Elementarbewegungen ist eine rein translatorische Bewegung des Roboters ist, die zweite der drei Elementarbewegungen beinhaltet eine Drehung des Roboters zu der Kontur hin beinhaltet, und die dritte der drei Elementarbewegungen beinhaltet eine Drehung des Roboters von der Kontur weg beinhaltet.

[0009] Darüber hinaus wird ein Verfahren zur Steuerung eines autonomen mobilen Roboters beschrie-

ben, der eine erste Karte eines Robotereinsatzgebietes hat, wobei diese zumindest Daten über die Position von Hindernissen enthält. Gemäß einem Ausführungsbeispiel umfasst das Verfahren das Planen eines Pfades zu einem Zielpunkt in der ersten Karte unter der Annahme einer vereinfachten virtuellen Form des Roboters. In manchen Ausführungsbeispielen kann das Verfahren weiter umfassen: das Bewegen des Roboters entlang des geplanten Pfades, das Erfassen von Hindernissen in der Umgebung des Roboters mittels einer Sensoreinheit des Roboters, während der der Roboter sich entlang des geplanten Pfades bewegt, und schließlich das Feststellen, dass der geplante Pfad aufgrund eines Hindernisses nicht kollisionsfrei befahren werden kann, unter Beachtung der tatsächlichen Roboterform, und das Fortsetzen der Bewegung des Roboters unter Beachtung der tatsächlichen Roboterform.

[0010] Des Weiteren wird ein Verfahren zur Steuerung eines autonomen mobilen Roboters mit Hilfe einer Karte des Robotereinsatzgebietes beschrieben, wobei die Karte zumindest Informationen über die Position realer, mittels einer Sensoreinheit erkannter Hindernisse sowie Informationen über virtuelle Hindernisse enthält. Gemäß einem Ausführungsbeispiel umfasst das Verfahren das Steuern des Roboters in der Nähe eines realen Hindernisses derart, dass eine Kollision mit dem realen Hindernis vermieden wird, wobei die tatsächliche Form des Roboters berücksichtigt wird, und Steuern des Roboters in der Nähe eines virtuellen Hindernisses derart, dass eine Kollision mit dem virtuellen Hindernis vermieden wird, wobei eine vereinfachte, virtuelle Form des Roboters berücksichtigt wird.

[0011] Weiter wird ein Verfahren zur Steuerung eines autonomen mobilen Roboters in einem Konturfolgemodus beschrieben, in dem der Roboter in einem Konturfolgeabstand im Wesentlichen einer Kontur folgt. Eine Karte des Roboters enthält zumindest Informationen über die Position realer, mittels einer Sensoreinheit erkannter Hindernisse sowie Informationen über virtuelle Hindernisse. Der Roboter bestimmt fortlaufend seine Position in dieser Karte, wobei in dem Konturfolgemodus sich der Roboter entlang einer Kontur bewegt; und die Kontur durch den Verlauf eines realen Hindernisses und den Verlauf einer virtuellen Grenze eines virtuellen Hindernisses gegeben ist.

Figurenliste

[0012] Verschiedene Ausführungsbeispiele werden nachfolgend anhand von Abbildungen näher erläutert. Die Darstellungen sind nicht zwangsläufig maßstabsgetreu und die Erfindung beschränkt sich nicht nur auf die dargestellten Aspekte. Vielmehr wird Wert darauf gelegt, die zugrunde liegenden Prinzipien darzustellen. In den Abbildungen zeigt:

Fig. 1 illustriert zwei Beispiele eines autonomen mobilen Roboters, bei dem jeweils eine Seite abgeflacht ist, sodass sich der Roboter mit seiner flachen Seite sehr nah entlang eines Hindernisses, z.B. einer Wand bewegen kann.

Fig. 2 illustriert anhand eines Blockdiagramms exemplarisch den Aufbau eines autonomen mobilen Roboters.

Fig. 3 zeigt verschiedene Varianten von Gehäuseformen für autonome mobile Roboter und illustriert die Auswirkung der Gehäuseform auf die Möglichkeiten des Roboters, sich zu bewegen.

Fig. 4 illustriert anhand eines Flussdiagramms ein Verfahren zur Steuerung eines autonomen mobilen Roboters in einer Sackgassen-Situation.

Fig. 5 illustriert anhand von vier Diagrammen (a) bis (d) eine exemplarische Vorgehensweise zur Steuerung eines autonomen mobilen Roboters in einer Sackgassen-Situation.

Fig. 6 illustriert exemplarisch ein Verfahren zur Steuerung eines autonomen, mobilen Roboters in einer Sackgassen-Situation.

Fig. 7 illustriert ein weiteres, komplexeres Beispiel eines Verfahrens zur Steuerung eines autonomen, mobilen Roboters in einer komplizierteren Sackgassen-Situation.

Fig. 8 illustriert exemplarisch verschiedene Elementarbewegungen.

Fig. 9 illustriert an einfachen Beispiel die Auswahl einer Elementarbewegung.

Fig. 10 illustriert die Bewegung von Elementarbewegungen basierend auf der Umgebung.

Fig. 11 illustriert eine Konturfolgefahrt mit virtuellem Hindernis.

Fig. 12 illustriert die Pfadplanung eines runden Roboters zwischen Hindernissen; diese ist äquivalent zu einer Pfadplanung für einen Punkt zwischen Hindernissen, die jeweils um den Roboterradius vergrößert wurden.

Fig. 13 illustriert ein Beispiel einer kostenbasierten Pfadplanung.

DETAILLIERTE BESCHREIBUNG

[0013] Aufgrund spezieller Anforderungen an die Funktion eines Roboters **100** kann es wünschenswert oder notwendig sein, von einer im Wesentlichen runden, scheibenförmigen Gehäuseform des Roboters abzuweichen. **Fig. 1** illustriert hierfür zwei Beispiele. Diagramm (a) in **Fig. 1** zeigt einen autonomen mobilen Roboter **100** zur Reinigung einer Bodenfläche (Reinigungsroboter). Eine Seite des Robotergehäuses ist abgeflacht, so dass der Roboter **100** sich mit

der flachen Seite parallel zu einer Wand **W** ausrichten kann. Diagramm (b) in **Fig. 1** zeigt ein weiteres Beispiel mit einem autonomen mobilen Roboter **100** zum Transport von Gegenständen (Serviceroboter) mit einer Plattform, welche bündig zur Kante eines Tisches **T** oder einer Arbeitsfläche bewegt werden kann. Die dargestellten Beispiele machen deutlich, dass derartige gestaltete Roboter sich nicht in jeder Situation auf der Stelle um ihre Hochachse drehen können, auch wenn ihre Antriebseinheiten dies prinzipiell ermöglichen würden. In den in **Fig. 1** dargestellten Situationen kann sich der Roboter **100** nicht drehen, ohne mit dem Hindernis (Wand **W**, Tisch **T**) zu kollidieren. Diese Tatsache hat Auswirkungen auf die Bewegungsplanung des Roboters **100**, da für die Planung einer Trajektorie von einem Ausgangspunkt zu einem Zielpunkt im Robotereinsatzgebiet zusätzlich die Orientierung des Roboters berücksichtigt werden muss.

[0014] Bevor auf die Bewegungsplanung für autonome mobile Roboter genauer eingegangen wird, soll zunächst der Aufbau eines autonomen mobilen Roboters kurz beschrieben werden. **Fig. 2** zeigt beispielhaft anhand eines Blockdiagramms verschiedene Einheiten (Module) eines autonomen mobilen Roboters **100**. Eine Einheit bzw. ein Modul kann dabei eine eigenständige Baugruppe oder ein Teil einer Software zur Steuerung des Roboters sein. Eine Einheit kann mehrere Untereinheiten aufweisen. Die für das Verhalten des Roboters **100** zuständige Software kann von der Steuereinheit **150** des Roboters **100** ausgeführt werden. In dem dargestellten Beispiel beinhaltet die Steuereinheit **150** einen Prozessor **155**, der dazu ausgebildet ist, in einem Speicher **156** enthaltene Software-Instruktionen auszuführen. Einige Funktionen der Steuereinheit **150** können zumindest teilweise auch mit Hilfe eines externen Rechners durchgeführt werden. Das heißt, die von der Steuereinheit **150** benötigte Rechenleistung kann zumindest teilweise auf einen externen Rechner ausgelagert sein, welcher beispielsweise über ein Heimnetzwerk oder über das Internet (Cloud) erreichbar sein kann.

[0015] Der autonome mobile Roboter **100** umfasst eine Antriebseinheit **170**, welche beispielsweise Elektromotoren, Getriebe und Räder aufweisen kann, wodurch der Roboter **100** - zumindest theoretisch - jeden Punkt eines Einsatzgebiets anfahren kann. Die Antriebseinheit **170** ist dazu ausgebildet, von der Steuereinheit **150** empfangene Kommandos oder Signale in eine Bewegung des Roboters **100** umzusetzen.

[0016] Der autonome mobile Roboter **100** umfasst des Weiteren eine Kommunikationseinheit **140**, um eine Kommunikationsverbindung **145** zu einer Mensch-Maschine-Schnittstelle (HMI) **200** und/oder anderen externen Geräten **300** herzustellen. Die Kommunikationsverbindung **145** ist beispielsweise eine direkte drahtlose Verbindung (z. B. Bluetooth),

eine lokale drahtlose Netzwerkverbindung (z. B. WLAN oder ZigBee) oder eine Internetverbindung (z. B. zu einem Cloud-Service). Die Mensch-Maschine-Schnittstelle **200** kann einem Nutzer Informationen über den autonomen mobilen Roboter **100** beispielsweise in visueller oder auch akustischer Form ausgeben (z. B. Batteriestatus, aktueller Arbeitsauftrag, Karteninformationen wie eine Reinigungskarte, etc.) und Nutzerkommandos für einen Arbeitsauftrag des autonomen mobilen Roboters **100** entgegennehmen. Beispiele für eine HMI **200** sind Tablet-PC, Smartphone, Smartwatch und andere Wearables, Computer, Smart-TV, oder Head-Mounted Displays, usw. Eine HMI **200** kann zusätzlich oder alternativ direkt in den Roboter integriert sein, wodurch der Roboter **100** beispielsweise über Tasten, Gesten und/oder Sprach-ein- und -ausgabe bedient werden kann.

[0017] Beispiele für externe Geräte **300** sind Computer und Server, auf denen Berechnungen und/oder Daten ausgelagert werden, externe Sensoren, die zusätzliche Informationen liefern, oder andere Haushaltsgeräte (z.B. andere autonome mobile Roboter), mit denen der autonome mobile Roboter **100** zusammenarbeiten und/oder Informationen austauschen kann.

[0018] Der autonome mobile Roboter **100** kann eine Arbeitseinheit **160** aufweisen, wie beispielsweise eine Bearbeitungseinheit zur Bearbeitung einer Bodenfläche und insbesondere zur Reinigung einer Bodenfläche (z. B. Bürste, Saugvorrichtung) oder ein Greifarm zum Fassen und Transportieren von Gegenständen.

[0019] In einigen Fällen wie beispielsweise bei einem Telepräsenz-Roboter oder einem Überwachungsroboter wird eine andere Baueinheit zur Erfüllung der vorgesehenen Aufgaben genutzt und es ist keine Arbeitseinheit **160** notwendig. So kann ein Telepräsenz-Roboter eine mit der HMI gekoppelte Kommunikationseinheit **140** aufweisen, die beispielsweise mit einer Multimediaeinheit ausgestattet sein kann, welche z.B. Mikrofon, Kamera und Bildschirm umfasst, um die Kommunikation zwischen mehreren räumlich weit entfernten Personen zu ermöglichen. Ein Überwachungsroboter ermittelt auf Kontrollfahrten mit Hilfe seiner Sensoren ungewöhnliche Ereignisse (z. B. Feuer, Licht, unautorisierte Personen, etc.) und informiert beispielsweise eine Kontrollstelle hierüber. In diesem Fall ist statt der Arbeitseinheit **160** eine Überwachungseinheit mit Sensoren zur Überwachung des Robotereinsatzgebietes vorgesehen.

[0020] Der autonome mobile Roboter **100** umfasst eine Sensoreinheit **120** mit verschiedenen Sensoren, beispielsweise einen oder mehrere Sensoren zur Erfassung von Informationen über die Umgebung (environment) des Roboters in seinem Einsatzgebiet, wie z.B. die Position und Ausdehnung von Hindernis-

sen oder anderen Landmarken (landmarks) im Einsatzgebiet. Sensoren zur Erfassung von Informationen über die Umgebung sind beispielsweise Sensoren zur Messung von Abständen zu Objekten (z.B. Wänden oder anderen Hindernissen, etc.) in der Umgebung des Roboters wie beispielsweise einen optischen und/oder akustischen Sensor, der mittels Triangulation oder Laufzeitmessung eines ausgesandten Signals Abstände messen kann (Triangulations-sensor, 3D-Kamera, Laserscanner, Ultraschallsensoren, etc.). Alternativ oder zusätzlich kann eine Kamera genutzt werden, um Informationen über die Umgebung zu sammeln. Insbesondere bei Betrachtung eines Objekts von zwei oder mehr Positionen aus, kann ebenfalls die Position und Ausdehnung eines Objekts bestimmt werden.

[0021] Zusätzlich kann der Roboter Sensoren besitzen, um einen (zumeist unbeabsichtigten) Kontakt (bzw. Kollision) mit einem Hindernis zu detektieren. Dies kann durch Beschleunigungsmesser (die z.B. die Geschwindigkeitsänderung des Roboters bei einer Kollision detektieren), Kontaktschalter, kapazitive Sensoren oder andere taktile bzw. berührungsempfindliche Sensoren realisiert werden. Zusätzlich kann der Roboter Bodensensoren besitzen, um eine Kante im Boden, beispielsweise einer Treppenstufe, zu erkennen. Weitere übliche Sensoren im Bereich autonomer mobiler Roboter sind Sensoren zur Bestimmung der Geschwindigkeit und/oder des zurückgelegten Weges des Roboters wie z.B. Odometer bzw. Inertialsensoren (Beschleunigungssensor, Drehratensensor) zur Bestimmung von Lage- und Bewegungsänderung des Roboters sowie Radkontaktschalter, um einen Kontakt zwischen Rad und Boden zu detektieren.

[0022] Der autonome mobile Roboter **100** kann einer Basisstation **110** zugeordnet sein, an welcher er beispielsweise seine Energiespeicher (Batterien) laden kann. Zu dieser Basisstation **110** kann der Roboter **100** nach Abschluss einer Aufgabe zurückkehren. Wenn der Roboter keine Aufgabe mehr zu bearbeiten hat, kann er in der Basisstation **110** auf einen neuen Einsatz warten.

[0023] Die Steuereinheit **150** kann dazu ausgebildet sein, alle Funktionen bereit zu stellen, die der Roboter benötigt, um sich selbstständig in seinem Einsatzgebiet zu bewegen und eine Aufgabe zu verrichten. Hierzu umfasst die Steuereinheit **150** beispielsweise den Prozessor **155** und das Speichermodul **156**, um eine Software auszuführen. Die Steuereinheit **150** kann basierend auf den von der Sensoreinheit **120** und der Kommunikationseinheit **140** erhaltenen Informationen Steuerkommandos (z. B. Steuersignale) für die Arbeitseinheit **160** und die Antriebseinheit **170** erzeugen. Die Antriebseinheit **170** kann wie bereits erwähnt diese Steuersignale bzw. Steuerkommandos in eine Bewegung des Roboters umsetzen.

Auch die in dem Speicher **156** enthaltene Software kann modular ausgebildet sein. Ein Navigationsmodul **152** stellt beispielsweise Funktionen zum automatischen Erstellen einer Karte des Robotereinsatzgebietes, sowie zur Bewegungsplanung des Roboters **100** bereit. Das Steuermodul **151** stellt z.B. allgemeine (globale) Steuerfunktionen bereit und kann eine Schnittstelle zwischen den einzelnen Modulen bilden.

[0024] Damit der Roboter autonom eine Aufgabe (task) verrichten kann, kann die Steuereinheit **150** Funktionen zur Navigation des Roboters in seinem Einsatzgebiet umfassen, die von dem oben erwähnten Navigationsmodul **152** bereitgestellt werden. Diese Funktionen sind an sich bekannt und können unter anderem eine der folgenden umfassen:

- das Erstellen von (elektronischen) Karten durch Sammeln von Informationen über die Umgebung mit Hilfe der Sensoreinheit **120**, beispielsweise aber nicht ausschließlich mittels SLAM-Verfahren (Simultaneous Localization and Mapping, simultane Lokalisierung und Kartenerstellung),
- das Management von einer oder mehreren Karten zu einem oder mehreren den Karten zugeordneten Einsatzgebieten des Roboters,
- das Bestimmen der Position und Orientierung (Pose) des Roboters in einer Karte basierend auf den mit den Sensoren der Sensoreinheit **120** ermittelten Informationen der Umgebung,
- eine kartenbasierte Pfadplanung (Trajektorienplanung) von einer aktuellen Pose des Roboters (Startpunkt) zu einem Zielpunkt,
- ein Konturfolgemodus, in dem der Roboter (**100**) sich entlang der Kontur eines oder mehrerer Hindernisse (z.B. einer Wand) in einem im Wesentlichen konstanten Abstand d zur dieser Kontur bewegt.

[0025] Die Steuereinheit **150** kann mit Hilfe des Navigationsmoduls **152** und basierend auf den Informationen der Sensoreinheit **120** beispielsweise während des Betriebs des Roboters laufend eine Karte des Robotereinsatzgebietes aktualisieren z.B. wenn die Umgebung des Roboters sich verändert (Hindernis wird verschoben, Tür wird geöffnet, etc.). Eine aktuelle Karte kann dann von der Steuereinheit **150** zur kurz- und/oder langfristigen Bewegungsplanung für den Roboter verwendet werden. Als Planungshorizont wird jener Weg bezeichnet, den die Steuereinheit **150** eine (Soll-) Bewegung des Roboters vorherberechnet, bevor diese tatsächlich ausgeführt wird. Die hier beschriebenen Ausführungsbeispiele betreffen unter anderem verschiedene Ansätze und Strategien für die Bewegungsplanung in bestimmten Situationen, z.B. in Situationen, in denen bestimmte Manö-

ver durch Hindernisse blockiert sind und daher nicht durchgeführt werden können.

[0026] Im Allgemeinen ist eine vom Roboter **100** verwendbare (elektronische) Karte eine Sammlung von Kartendaten (z.B. eine Datenbank) zur Speicherung von ortsbezogenen Informationen über ein Einsatzgebiet des Roboters und die für den Roboter relevante Umgebung in diesem Einsatzgebiet. In diesem Zusammenhang bedeutet „ortsbezogen“, dass die gespeicherten Informationen jeweils einer Position oder einer Pose in einer Karte zugeordnet sind. Eine Karte repräsentiert also eine Vielzahl von Datensätzen mit Kartendaten, und die Kartendaten können beliebige ortsbezogene Informationen enthalten. Hierbei können die ortsbezogenen Informationen in unterschiedlichen Detail- und Abstraktionsgrad gespeichert sein, wobei dieser an eine spezifische Funktion angepasst sein kann. Insbesondere können einzelne Informationen redundant gespeichert sein. Häufig wird eine Zusammenstellung mehrerer Karten, die das gleiche Gebiet betreffen, jedoch in unterschiedlicher Form (Datenstruktur) gespeichert sind, ebenfalls als „eine Karte“ bezeichnet.

[0027] Nicht-kreisrunde Roboter - Einleitung: **Fig. 3** zeigt, jeweils in einer Ansicht von unten, verschiedene an sich bekannte Beispiele von Gehäuseformen für autonome mobile Roboter **100**. In den dargestellten Beispielen weisen die Roboter **100** jeweils eine Arbeitseinheit **160** auf, beispielsweise zur Bearbeitung einer Bodenfläche wie insbesondere eine Bürste, eine Saugeinheit und/oder eine Wischeinheit.

[0028] Des Weiteren weisen die Roboter **100** jeweils eine Antriebseinheit **170** mit zwei unabhängig voneinander angetriebenen Rädern **170R**, **170L** auf. Die Roboter können eine bevorzugte Bewegungsrichtung (ohne Beschränkung der Allgemeinheit als Vorwärtsrichtung definiert) besitzen, die durch einen Pfeil gekennzeichnet ist. Diese bevorzugte Bewegungsrichtung bzw. Vorwärtsrichtung kann beispielsweise durch die Anordnung der Arbeitseinheit im oder am Gehäuse, aber auch durch die Anordnung von Sensoren (z.B. der Sensoreinheit **120**) vorgegeben sein. Beispielsweise kann eine Reinigungseinheit zur Aufnahme von Schmutz (z. B. Saugvorrichtung) vor der Antriebseinheit **170** angebracht sein, so dass weniger Schmutz in die Räder geraten kann. Des Weiteren kann beispielsweise eine Reinigungseinheit zum Auftragen einer Reinigungsflüssigkeit oder zum Polieren einer Bodenfläche hinter der Antriebseinheit **170** angebracht sein, so dass die Räder keine Verschmutzung auf der gereinigten Bodenfläche hinterlassen. Beispielsweise sind Sensoren so angeordnet, dass sie vorrangig die Umgebung in der bevorzugten Bewegungsrichtung des Roboters (also in vor dem Roboter **100**) detektieren. Der Roboter **100** kann sich allerdings auch (unter Umständen mit gewissen

Einschränkungen) entgegen der bevorzugten Bewegungsrichtung (d.h. rückwärts) bewegen.

[0029] Wenn die zwei angetriebenen Räder **170R**, **170L** sich in entgegengesetzte Richtung drehen, so dreht sich der Roboter auf der Stelle um den durch ein „x“ markierten zentralen Punkt (kinematischer Mittelpunkt, Drehmittelpunkt) um seine Hochachse und führt damit eine reine Rotationsbewegung aus (d.h. ohne translatorische Bewegungskomponente).

[0030] Diagramm (a) aus **Fig. 3** zeigt einen runden Roboter, dessen Räder **170R** und **170L** in einer der Symmetrieachsen angeordnet ist. Dies hat den Vorteil, dass der Roboter sich auf der Stelle um seinen Mittelpunkt drehen kann. Unabhängig von der Position von Hindernissen **H** wird diese Drehung niemals gestört, weshalb der runde Roboter immer nach einer geeigneten Drehung um seine Hochachse in seine bevorzugte Richtung (d.h. vorwärts) fahren kann.

[0031] Diagramm (b) aus **Fig. 3** zeigt einen D-förmigen Roboter. Die D-Form hat den Vorteil, dass eine Arbeitseinheit **160** genutzt werden kann, welche sich über die gesamte Breite des Roboters erstreckt. Zusätzlich kann die Arbeitseinheit **160** besonders nahe an Hindernisse **H** (wie z.B. eine Wand) heranbewegt werden. In dieser der dargestellten Pose kann sich der Roboter jedoch nicht mehr kollisionsfrei drehen; er muss vor einer Drehung um seine Hochachse erst zumindest ein Stück rückwärtsfahren (entgegen der bevorzugten Bewegungsrichtung).

[0032] Diagramm (c) aus **Fig. 3** zeigt einen runden Roboter, dessen Räder **170R** und **170L** nicht entlang einer der Symmetrieachsen der Roboterform angeordnet ist. Dies hat den Vorteil, dass die Arbeitseinheit **160** sich über die gesamte Breite des Roboters erstrecken kann. Allerdings stimmt der zentrale Punkt „x“ (kinematischer Mittelpunkt) des Roboters **100** nicht mehr mit dem geometrischen Mittelpunkt der kreisrunden Gehäusegrundfläche überein, weshalb es bei einer Drehung zu einer Kollision mit einem Hindernis **H** kommen kann. Zur Vermeidung solcher Kollisionen wie in Diagramm (c) aus **Fig. 3** muss der Roboter erst zumindest ein Stück rückwärtsfahren. Im vorliegenden Beispiel kann der Roboter nicht auf einfache Weise eine reine Rotationsbewegung durchführen. Eine Drehung um den kinematischen Mittelpunkt „x“ hat zur Folge, dass die Bewegung des geometrischen Mittelpunkts auch eine translatorische Bewegungskomponente aufweist (insbesondere Bewegung auf einer Kreisbahn um den kinematischen Mittelpunkt), die ggf. von einem Hindernis blockiert sein kann.

[0033] Diagramm (d) aus **Fig. 3** zeigt eine tropfenförmige Gehäuseform eines Roboters **100**, bei dem die Grundfläche des Gehäuses eine ausgeprägte Ecke aufweist, im Übrigen jedoch rund ist. Dies hat den

Vorteil, dass eine Arbeitseinheit **160** in der Ecke des Roboters angeordnet werden kann, und somit diese besonders nahe an Hindernisse herangeführt werden kann (z.B. in die Ecke eines Raumes). Die Bewegung des Roboters ist weniger stark eingeschränkt wie bei der D-Form. Aber auch hier gibt es Situationen, in denen der Roboter zumindest ein Stück rückwärtsfahren muss, bevor eine Drehung um die Hochachse ungehindert möglich ist.

[0034] Diagramm (e) aus **Fig. 3** zeigt einen langgestreckten im Wesentlichen D-förmigen Roboter. Dies hat den Vorteil, dass mehr Platz für eine Arbeitseinheit **160** vorhanden ist, die sich über die gesamte Breite des Roboters erstrecken kann. Zusätzlich kann die Arbeitseinheit **160** besonders nahe an Hindernisse H wie eine Wand herangeführt werden. In dieser Position kann der Roboter sich aber nicht mehr drehen, und muss erst zumindest ein Stück rückwärtsfahren.

[0035] Situation, aus denen der Roboter, wie in den Diagrammen (b) bis (e) der **Fig. 3** dargestellt, nur durch Fahren in eine Rückwärtsrichtung (entgegengesetzt zur bevorzugten Bewegungsrichtung) heraussteuern kann, wird im Folgenden als „Sackgassen-Situation“ bezeichnet. Es sei angemerkt, dass es sich bei den in **Fig. 3** dargestellten Robotern nur um Beispiele handelt. Selbstverständlich sind beliebige andere Formen möglich. Insbesondere kann die Form auch mit der Höhe des Roboters variieren (vgl. Diagramm (b) in **Fig. 1**). Andere Varianten des Antriebsmoduls **170** (wie z. B. Kettenantrieb, Beine) sind ebenfalls bekannt und möglich.

[0036] Konturfolgefahrt: Ein einfacher Ansatz zur lokalen Planung eines Pfades (einer Trajektorie) für einen autonomen mobilen Roboter **100** besteht darin, dass der Roboter schlicht in einem im Wesentlichen konstanten Konturfolgeabstand d einer Kontur eines oder mehrerer Hindernisse (Konturfolgefahrt) folgt. Ein Betriebsmodus, in dem der Roboter sich an einer Kontur eines Hindernisses orientierend entlang einer Kontur in einem im Wesentlichen konstanten Abstand bewegt wird im Folgenden als Konturfolgemodus (contour following mode) oder Hindernisfolgemodus (obstacle following mode) bezeichnet. Die vom Roboter im Konturfolgemodus ausgeführte Bewegung wird als Konturfolgefahrt (contour following run) bezeichnet, und der Abstand zur Kontur als Konturfolgeabstand (contour following distance). Die Verwendung eines Konturfolgemodus ist an sich bekannt und wird beispielsweise verwendet, um Hindernissen auszuweichen (siehe z.B. J. Fasola et al., „Fast Goal Navigation with Obstacle Avoidance using a Dynamic Local Visual Model“, in: Proc. VII. Simpósio Brasileiro de Automação Inteligente, São Luis, Sept. 2005). Methoden zur Implementierung von Konturfolgefahrten können unter anderem auf Konzepten der Behavior-Based-Robotics oder der Reactive-Robotics be-

ruhen, wobei aktuelle Sensormessungen (insbesondere zur Position eines Hindernisses relativ zum Roboter und/oder dem Abstand des Roboters von einem Hindernis) direkt in Steuerkommandos für die Antriebseinheit umgesetzt werden können

[0037] Die Kontur kann durch die Form einer Wand, eines großen Hindernisses, aber auch durch mehrere kleine, eng stehende Hindernisse gegeben sein. Eine Kante, über die ein Roboter abstürzen kann (Absturzkante), wie z.B. bei einer Treppe wird in diesem Zusammenhang auch als Hindernis mit einer Kontur betrachtet, welcher der Roboter folgen kann. Zudem kann es sich bei den die Kontur bildenden Hindernissen um Markierungen (beispielsweise in Form von Magnetbändern, Stromschleifen oder Leitstrahlender) handeln, die der Roboter mit einem entsprechenden Sensor detektieren kann. Aus diesen Sensordaten kann eine Grenze (z. B. der Verlauf des Magnetbandes bzw. der Stromschleife, Verlauf des ausgesandten Leitstrahls) abgeleitet werden, welche der Roboter nicht selbsttätig überfahren darf. Diese Grenze kann auch als Kontur genutzt werden, welcher der Roboter folgen kann. Des Weiteren können in den Kartendaten virtuelle Hindernisse verzeichnet sein, die Bereiche markieren, die der Roboter nicht selbstständig befahren darf (diese werden auch als Sperrbereiche, „Keep-Out-Areas“ oder „No-Go-Areas“ bezeichnet). Zusätzlich oder alternativ kann ein virtuelles Hindernis und insbesondere seine virtuelle Kontur vorübergehend benutzt werden, um den Roboter in einem zur Bearbeitung vorgesehenen Bereich „einzusperren“ oder zu führen, bis die Bearbeitung abgeschlossen ist. Die virtuellen Konturen eines solchen virtuellen Hindernisses können in einem Konturfolgemodus ebenfalls als Kontur genutzt werden, der der Roboter folgen kann..

[0038] Der Konturfolgeabstand d ist abhängig von der Größe und der Aufgabe des Roboters, kann jedoch in einem konkreten Konturfolgemodus im Wesentlichen konstant bleiben. Durch einen größeren Abstand können unbeabsichtigte Kollisionen beispielsweise durch Fahrfehler leichter (mit größerer Wahrscheinlichkeit) vermieden werden. Bei Robotern zur Bearbeitung (insbesondere Reinigung) einer Bodenfläche kann ein Konturfolgemodus für die Bearbeitung nahe an Wänden und anderen Hindernissen genutzt werden. Folglich können derartige Roboter sehr nahe an Hindernissen entlangfahren, um eine hohe Flächenabdeckung und insbesondere eine gründliche Reinigung in Ecken und Kanten zu erreichen. Exemplarische Werte für kleine Reinigungsroboter im Haushaltsbereich liegen hierbei zwischen 2, 5 mm und 20 mm. Es gibt auch Reinigungsroboter die während einer Konturfolgefahrt direkten Kontakt (d.h. mittels Berührung) zwischen einem Teil des Roboters und der zu folgenden Kontur herstellen und halten. Für große Roboter kann der Konturfolgeab-

stand d deutlich größer sein als bei vergleichsweise kleinen Robotern.

[0039] Zur Steuerung des Roboters während einer Konturfolgefahrt kann der Roboter Sensoren zur Erfassung der unmittelbaren Umgebung des Roboters besitzen (siehe **Fig. 2**, Sensoreinheit **120**). Diese Sensoren können beispielsweise Abstände zu Hindernissen und insbesondere zu der zu folgenden Kontur im Nahbereich zuverlässig bestimmen. Beispielsweise kann solch ein Sensor an jener Seite des Roboters angeordnet sein, welche der zu folgenden Kontur zugewandt ist.

[0040] Alternativ oder zusätzlich kann die Steuerung des Roboters während einer Konturfolgefahrt auf Kartendaten basieren, wobei Sensormessungen zur Bestimmung von Position und Orientierung (Pose) des Roboters und von Hindernissen gespeichert und weiterverarbeitet werden. Eine kartenbasierte Planung ermöglicht eine vorausschauende Trajektorienplanung und Robotersteuerung und berücksichtigt auch Informationen über Hindernisse die gerade nicht von einem der Sensoren erfasst werden können („blinder Fleck“ eines Sensors). Insbesondere können auch Informationen berücksichtigt werden, die nicht mit Sensoren erfasst werden können, wie beispielsweise in der Karte verzeichnete virtuelle Hindernisse (z.B. Sperrbereiche), die der Roboter nicht selbstständig befahren, überfahren und/oder bearbeiten darf. Bei den hier beschriebenen Ausführungsbeispielen können beispielsweise die Kriterien, die verwendet werden, um von einem Konturfolgemodus in einen anderen Konturfolgemodus zu wechseln (oder einen Konturfolgemodus abubrechen), kartenbasiert ausgewertet werden. Beispielsweise kann ein Kriterium zum Beenden eines Konturfolgemodus sein, dass der Roboter sich in Richtung eines Zielpunktes kollisionsfrei drehen kann. Dieses Kriterium „Roboter kann sich kollisionsfrei zum Zielpunkt hin drehen“ kann beispielsweise anhand der aktuellen Kartendaten des Roboters beurteilt werden.

[0041] Zur Navigation und Kartenerstellung werden zur Erkennung von Hindernissen üblicherweise Sensoren mit vergleichsweise großer Reichweite eingesetzt, die zwar weiter entfernte Hindernisse gut detektieren können, jedoch im Nahbereich häufig ungeeignet sind. Beispielsweise kann ein Triangulationssensor genutzt werden, welcher durch Aussenden von strukturiertem Licht (z. B. ein Laserstrahl oder ein aufgefächerter Laserstrahl) und dem Detektieren des von einem Hindernis H zurückgestreuten Lichts den Abstand zu diesem Hindernis H bestimmen kann. In der Regel ist die Messung des Abstandes zu dem Hindernis umso genauer, je kleiner der Abstand ist. Jedoch kann es einen minimalen Abstand geben, bei dem das zurückgestreute Licht nicht mehr vom Sensor empfangen werden kann, da es außerhalb seines Sichtfeldes liegt. Beispielsweise können Senso-

ren, welche die Laufzeitmessung eines abgestrahlten Signals (Licht, Schall) messen, genutzt werden; diese Sensoren besitzen in der Regel auch einen Mindestabstand zur Detektion eines Hindernisses. Bei Kameras können im Nahbereich ebenfalls Probleme aufgrund eines beschränkten Sichtfeldes als auch einer beschränkten Fokussierung auftreten.

[0042] Durch Nutzung der Kartendaten kann der Roboter trotz eingeschränkter Sensorik nahe an Hindernissen navigieren, ohne dass ein zusätzlicher Sensor für die Konturfolgefahrt benötigt wird. Zusätzlich wird eine Steuerung entgegen die bevorzugte Bewegungsrichtung (d.h. in Rückwärtsrichtung) einfach ermöglicht, ohne aufwändige zusätzliche Sensorik im rückwärtigen Teil des Roboters einzusetzen.

[0043] Handhabung von Sackgassen-Situationen - Rückwärtsfahren: Wie in **Fig. 3** exemplarisch dargestellt, können allgemeine, nichtrunde Roboterformen dazu führen, dass eine Bewegung des Roboters **100** in eine bevorzugte Richtung (Vorwärtsrichtung) nicht immer möglich ist, da eine Drehbewegung (insbesondere im Stand um den zentralen Punkt „x“) des Roboters in die gewünschte Richtung aufgrund eines Hindernisses in der Umgebung des Roboters blockiert sein kann. In diesem Fall befindet sich der Roboter in einer Sackgassen-Situation. Es sei angemerkt, dass insbesondere Roboter zur Bearbeitung einer Bodenfläche in solche Situationen hinein fahren sollen, um eine möglichst große Flächenabdeckung der Bearbeitung und eine effiziente Reinigung von Ecken und Kanten zu erreichen. Das heißt der Roboter wird im normalen Betrieb bei der Durchführung seiner Aufgaben zwangsläufig immer wieder in Sackgassen-Situationen geraten.

[0044] Eine einfache Möglichkeit, aus einer Sackgasse-Situation herauszusteuern, besteht darin, genau jenen Weg rückwärts zu fahren, auf dem der Roboter (vorwärts) in die Sackgasse hineingefahren ist. Das heißt, die zuletzt erzeugten Steuerungskommandos für die Antriebseinheit würden in umgekehrter Reihenfolge und in invertierter Form erneut ausgeführt, bis eine Abbruchbedingung (z. B. der Roboter kann sich im Stand drehen) erfüllt ist.

[0045] Für die oben erwähnte Rückwärtsfahrt zur Befreiung aus einer Sackgassen-Situation müssen zusätzliche Informationen über die Steuerkommandos und/oder über den gefahrenen Weg (z.B. Wegpunkte) des Roboters gespeichert werden, wodurch der Speicherbedarf steigt. Des Weiteren führt ein invertiertes Steuersignal nicht notwendigerweise zu einer invertierten Bewegung. So können z.B. fortlaufende Störungen der Bewegung (beispielsweise durch Schlupf und Drift der Antriebseinheit und insbesondere der Räder), die nicht direkt proportional zu der theoretischen ungestörten Bewegung sein müssen, dazu führen, dass eine inverse Ansteuerung des

Antriebs für die Rückwärtsfahrt nicht zu derselben Trajektorie führt, die zuvor in Vorwärtsrichtung gefahren wurde. Das Weiteren können Situationen auftreten, dass ein mobiles Hindernis, das die Sackgassen-Situation verursacht hat, seine Position ändert. In dieser und anderen Situationen führt ein fest vordefiniertes Fahrmanöver (Rückwärtsfahren für eine bestimmte Strecke) nicht immer zu einem „sinnvollen“ Verhalten des Roboters.

[0046] Um dieses Problem zu überwinden, können basierend auf den Karteninformationen neue Steuerkommandos erzeugt werden, um den Roboter rückwärts zu steuern. Hierbei kann insbesondere der Kontur gefolgt werden, die der Roboter in die Sackgassen-Situation hinein gefolgt ist. Dies geschieht so lange, bis festgestellt wird, dass die Sackgasse verlassen werden kann bzw. verlassen wurde.

[0047] Fig. 4 zeigt ein mögliches Vorgehen zur Steuerung eines autonomen mobilen Roboters **100**, um der Kontur eines Hindernisses zu folgen. Hierbei wird ein erster Konturfolgemodus gestartet und durchgeführt (Fig. 4, Schritt 10). Dieser ist beispielsweise durch jene Seite des Roboters, die der Kontur zugewandt ist, die Richtung, entlang der der Kontur gefolgt werden soll, und den Konturfolgeabstand d gekennzeichnet. Während der Roboter sich entlang der Kontur bewegt, kann es passieren, dass der Roboter feststellt, dass eine fortgesetzte Bewegung des Roboters in dem ersten Konturfolgemodus entlang der ersten gewählten Richtung der Kontur nicht möglich ist, weil er sich beispielsweise in einer Sackgassen-Situation befindet (Fig. 4, Schritt 11). Die Sackgassen-Situation detektiert der Roboter beispielsweise, indem er seine Bewegungsmöglichkeiten anhand seiner aktuellen Position in der Karte und der hierin verzeichneten Hindernisse ermittelt. Wenn keine Vorwärts- oder Drehbewegung möglich ist, weil diese zu einer Kollision mit einem Hindernis führen würde, so ist dies die besagte Sackgassen-Situation. Um aus der Sackgasse heraus zu navigieren, folgt der Roboter in einem zweiten Konturfolgemodus der Kontur entgegen der ersten Richtung (Fig. 4, Schritt 13). Dabei wird ein Kriterium festgelegt (Fig. 4, Schritt 12) bei dessen Erfüllung der zweite Konturfolgemodus gestoppt werden soll, um beispielsweise die Bewegung im ersten Konturfolgemodus entlang der ersten gewählten Richtung wieder aufzunehmen.

[0048] Die Diagramme (a) bis (d) in Fig. 5 sollen das Verfahren gemäß Fig. 4 an einem Beispiel verdeutlichen. Diagramm (a) der Fig. 5 zeigt einen Roboter **100** wie er einer Kontur einer Wand **W** (oder eines anderen Hindernisses) folgt, wobei er einen möglichst konstanten Abstand d zur Kontur der Wand **W** (Konturfolgeabstand) einhält. Der Roboter **100** folgt der Kontur so lange, bis sein Weg wie z.B. in Diagramm (b) der Fig. 5 dargestellt durch ein (z.B. vor dem Roboter **100** befindliches) Hindernis **H** blockiert

ist. Das Hindernis **H** kann hierbei auch ein Teil einer Wand sein, wie es z.B. in einer Ecke eines Zimmers der Fall ist. Im Folgenden wird der Einfachheit halber die Kontur mit dem Bezugszeichen **W** bezeichnet. Es versteht sich, dass diese Kontur **W** sowohl die Kontur einer Wand als auch ein oder mehrerer anderer Hindernisse repräsentieren kann. Exemplarisch kann man sich die Kontur **W** jedoch als Wand eines Raumes vorstellen.

[0049] Der Weg wird als blockiert angesehen, wenn das Hindernis **H** sich (nur mehr) in einem Sicherheitsabstand d_s vom Hindernis **H** entfernt befindet und eine Rotation des Roboters **100** nicht möglich ist. Es versteht sich, dass bei einer hinreichend großen Wahl des Sicherheitsabstands d_s der Rotationsfreiheitsgrad des Roboters nicht durch ein vor dem Roboter befindliches Hindernis eingeschränkt wird, jedoch wird vor allem bei Robotern zur Bodenbearbeitung der Sicherheitsabstand d_s möglichst klein gewählt (deutlich kleiner als die Außenabmessungen des Roboters selbst), um eine möglichst gut Flächenabdeckung bei der Bearbeitung der Bodenfläche zu erreichen. So kann der Sicherheitsabstand d_s beispielsweise so gewählt werden, dass der Roboter sich sicher kollisionsfrei drehen kann oder dass er sich nicht kollisionsfrei drehen kann. In vielen Anwendungen ist letzteres der Fall. Der Sicherheitsabstand d_s kann beispielsweise kleiner oder gleich dem Konturfolgeabstand d sein ($d_s \leq d$). Beispielsweise kann auf den Sicherheitsabstand d_s ganz verzichtet werden (d.h. $d_s = 0$ mm), so dass der Roboter so lange der Kontur **W** folgen würde, bis ein vor dem Roboter liegendes Hindernis **H** berührt wird. Die Berührung kann z.B. mittels eines taktilen Sensors (Sensor, der auf Berührung reagiert) detektiert werden.

[0050] Um aus dieser Position heraus zu fahren, wechselt die Robotersteuerung **150** in einen zweiten Konturfolgemodus, in dem der Roboter **100** der Kontur der Wand **W** in die entgegengesetzte Richtung (siehe Diagramm (b) aus Fig. 5) folgt, bis ein definiertes Kriterium erfüllt ist, nämlich beispielsweise bis der Roboter **100** so weit von dem Hindernis entfernt ist, dass er sich ohne Kollision drehen kann und der Kontur des neuen Hindernis **H** in der ursprünglichen Richtung (vorwärts) folgen kann. Der zweite Konturfolgemodus unterscheidet sich also von dem ersten Konturfolgemodus in einem Parameter, nämlich in der Richtung unterscheidet, in welcher der Roboter der Kontur folgen soll. Zudem wird ein Kriterium gesetzt (z.B. Drehung nicht mehr blockiert), bei welchem der zweite Konturfolgemodus beendet werden kann, um beispielsweise in den ersten Konturfolgemodus zurückzukehren bzw. ihn erneut zu starten. Weitere Konturfolgemodi können sich in anderen Parametern unterscheiden (z.B. den Konturfolgeabstand, Seite des Roboters (links oder rechts) an der die zu folgende Kontur liegt, etc.). In einfachen Beispielen ist ein bestimmter Konturfolgemodus durch die Parame-

ter Fahrtrichtung (vorwärts oder rückwärts) und dem Konturfolgeabstand festgelegt.

[0051] Das Kriterium zum Beenden des zweiten Konturfolgemodus kann gemäß den hier beschriebenen Beispielen sein, dass der Roboter sich wieder weitgehend frei bewegen und insbesondere die erste Konturfolgefahrt entlang der Kontur eines neuen Hindernisses fortsetzen kann. Dies bedeutet unter anderem, dass der Rotationsfreiheitsgrad des Roboters nicht mehr blockiert ist. Hierbei ist jedoch a priori nicht klar, wie weit der Roboter sich drehen muss, um die Konturfolgefahrt fortsetzen zu können. Dies ist exemplarisch in den Diagrammen (c) und (d) der **Fig. 5** visualisiert.

[0052] In Diagramm (c) der **Fig. 5** ist Fahrmanöver dargestellt, bei dem der Roboter **100** an einem Hindernis **H** vorbeifährt, welches zentral auf der Bahn des Roboters während der Konturfolgefahrt liegt. In diesem Fall muss der Roboter der Kontur der Wand **W** eine Strecke d_{w1} rückwärts folgen, bis sich der Roboter **100** wieder frei drehen kann. Der Platz, den der Roboter für eine Drehung um den zentralen Punkt „x“ benötigt, ist durch den Kreis **C** gekennzeichnet. Hierbei sei angemerkt, dass der Roboter **100**, sobald er sich ein kleines Stück rückwärts entlang der Kontur **W** bewegt hat, sich drehen kann. Hierbei kann er sich aber nicht soweit drehen, um an dem Hindernis **H** vorbei zu steuern.

[0053] Diagramm (d) der **Fig. 5** zeigt ein Fahrmanöver, um an einem Hindernis **H** vorbei zusteuern, welches nahe an der zu folgenden Kontur **W** liegt. Der Roboter muss hierzu der Kontur der Wand **W** entlang einer Strecke d_{w2} rückwärts folgen, um sich wieder drehen zu können. Die hierbei zurückzusetzende Strecke d_{w2} ist kleiner als die Strecke d_{w1} aus **Fig. 5 C**. Gleichzeitig ist der Rotationsfreiheitsgrad des Roboters weiter durch ein zweites Hindernis **H'** eingeschränkt, welches sich innerhalb des Drehkreises **C** befindet. Jedoch kann der Roboter trotz dieser Einschränkung zwischen den beiden Hindernissen **H**, **H'** hindurch fahren und anschließend den ersten Konturfolgemodus fortsetzen.

[0054] Aus den in Diagramm (c) und (d) der **Fig. 5** dargestellten Beispielen ist ersichtlich, dass die Tatsache, ob und wie weit sich der Roboter drehen kann, kein aussagekräftiges Kriterium ist, um die zweite Konturfolgefahrt zu beenden. Dies gilt insbesondere, wenn die erste Konturfolgefahrt fortgesetzt werden soll.

[0055] Ein mögliches Kriterium für die Beurteilung (durch den Roboter), ob der zweite Konturfolgemodus beendet und die vorherige Konturfolgefahrt (im ersten Konturfolgemodus) sinnvoll fortgesetzt werden kann, ist beispielsweise, dass der Roboter sich nach erfolgreicher Drehung geradlinig vorwärts (also

in die Bewegungsrichtung des ersten Konturfolgemodus) bewegen kann. Dies ist in **Fig. 5**, Diagramme (c) und (d) durch die Passage **P** gekennzeichnet, in der sich der Roboter die Länge **l** geradlinig bewegen kann. Die Länge **l** kann hierbei ein voreingestellter Wert sein oder kann zumindest teilweise basierend auf dem während der Rotation zurückgelegten Winkel oder der während der zweiten Konturfolgefahrt zurückgelegten Strecke d_{w1} bzw. d_{w2} bestimmt werden. Beispielsweise kann die Länge **l** so gewählt werden, dass die vordere Kontur des Roboters **100** den Drehkreis **C** verlässt. Die Länge **l** kann auch kürzer gewählt sein, als zum Verlassen des Drehkreises **C** notwendig ist. Dies führt dazu, dass der Roboter enger an Hindernissen navigieren kann. Jedoch kann dies dazu führen, dass nach Rückkehr zum ersten Konturfolgemodus dieser erneut unterbrochen werden muss, was zu einer Folge an Vor- und Zurückbewegungen führen kann. Das Kriterium, ob der zweite Konturfolgemodus abgebrochen werden soll, kann insbesondere kartenbasiert ausgewertet werden. Dabei wird davon ausgegangen, dass die Karte zumindest in der lokalen Umgebung des Roboters **100** hinreichend genau und aktuell ist.

[0056] In manchen Ausführungsbeispielen kann das Kriterium für das Beenden des zweiten Konturfolgemodus lediglich in der Möglichkeit einer geraden Vorwärtsbewegung bestehen. Beispielsweise kann das Kriterium sein, dass der Roboter sich um jene Strecke, die er im zweiten Konturfolgemodus rückwärts gefahren ist, plus eine weitere vorgebbare Strecke (z.B. Abstand **d**) in eine vorgebbare Richtung vorwärtsbewegen können muss. Um sich in diese vorgebbare Richtung ausrichten zu können, muss der Roboter sich in der Regel drehen. Die Möglichkeit der Drehung muss kein expliziter Bestandteil des Kriteriums für das Beenden des zweiten Konturfolgemodus sein. In manchen Situationen kann der Roboter ohne eine zusätzliche Drehung eine entsprechende Richtung erreichen, wenn der Roboter sich beispielsweise während des zweiten Konturfolgemodus entlang einer gekrümmten Kontur (rückwärts) bewegt. Ein weiteres Beispiel, bei dem eine Drehung nicht notwendig sein kann, ist eine dynamische Änderung der Umgebung. So kann beispielsweise ein Nutzer jenes Hindernis **H** entfernen, welches den zweiten Konturfolgemodus ausgelöst hat. Folglich ist die Vorwärtsbewegung des Roboters nicht mehr eingeschränkt und der zweite Konturfolgemodus kann mit einer geraden Bewegung ohne Drehung beendet werden.

[0057] Zusätzlich oder alternativ kann bei der Auswertung des Kriteriums, das zur Beendigung des zweiten Konturfolgemodus führt, die Lage des Hindernis **H**, welches zu einer Unterbrechung des ersten Konturfolgemodus führte, oder die Lage eines anderen Hindernisses **H'** nach einer möglichen Drehung geprüft werden. So sollte sich in einem vorgebbaren Abstand vor dem Roboter kein Hindernis

befinden. Gleichzeitig soll jenes Hindernis, das zuvor die Sackgassen-Situation herbeigeführt hat, nach der Drehung so relativ zum Roboter positioniert sein, dass dieser im ersten Konturfolgemodus der Kontur dieses Hindernisses **H** in dem vorgegebenen Konturfolgeabstand **d** folgen kann. Dies bedeutet insbesondere, dass nach einer Bewegung vorwärts um die Länge **l** sich ein Teil der Kontur des Hindernis in dem Konturfolgeabstand **d** zum Roboter befindet (vgl. Diagramm (c) aus Fig. 5).

[0058] Wie zuvor anhand von Diagramm (d) der Fig. 5 erläutert, kann der Winkel, um den sich der Roboter mindestens drehen können muss, um den zweiten Konturfolgemodus zu beenden, vergleichsweise klein sein, z.B. im Bereich von 1 - 5 Grad, oder es kann ganz auf eine Drehung verzichtet werden. Fig. 6, Diagramm (a), zeigt ein Beispiel, in dem neben dem Hindernis **H** vor dem Roboter ein zweites Hindernis **H'** die Rotation des Roboters unmittelbar einschränkt. Solche Hindernisse sind beispielsweise dadurch erkennbar, dass sie sich zumindest teilweise innerhalb des vorderen Bereichs **S** des Drehkreises **C** befinden (z.B. innerhalb des vorderen Halbkreises). In solch einer Konstellation ist immer eine vergleichsweise große Drehung notwendig, damit der Roboter den zweiten Konturfolgemodus beenden und den erste Konturfolgemodus fortsetzen kann. Beispielsweise, um die zu prüfenden Bewegungsoptionen einzuschränken, kann es hier vorteilhaft sein, als Kriterium einen großen Mindestwinkel zu setzen, der zwischen der Orientierung des Roboters vor und nach der Drehung liegen muss. Dieser Mindestwinkel kann ein Standardwert sein (z. B. 45°) oder abhängig von der Roboterform und/oder der Form und Größe des Hindernis **H'** gewählt werden.

[0059] Das Setzen des Kriteriums zum Beenden des zweiten Konturfolgemodus kann also abhängig von der (z.B. in einer Karte des Roboters gespeicherten) Position der Hindernisse in der Umgebung des Roboters sein. Insbesondere kann ein erstes Kriterium festgelegt und verwendet werden, wenn sich zumindest ein Punkt eines Hindernisses in einem vorgebbaren Bereich **S** insbesondere neben dem Roboter befindet, und andernfalls ein zweites Kriterium. Gemäß beiden Kriterien soll beispielsweise eine Drehung des Roboters in eine Position weg von der Kontur möglich sein, wobei zumindest bei dem ersten Kriterium der Winkel der Drehung größer als ein vorgebarer Mindestwinkel sein kann. Wenn beide Kriterien einen Mindestwinkel enthalten, so ist der Mindestwinkel gemäß dem ersten Kriterium größer, als der Mindestwinkel gemäß dem zweiten Kriterium.

[0060] Fig. 6, Diagramm (b), zeigt ein Beispiel, in dem das zweite Hindernis **H'** sich an der gleichen Position wie in Fig. 5, Diagramm (d) befindet. Gemäß dem in Fig. 5, Diagramm (d), dargestellten Beispiel liegt das erste Hindernis **H** nahe der Kontur **W**, so

dass der Roboter nach einer kleinen Drehung zwischen den beiden Hindernissen **H**, **H'** hindurch fahren kann. In Fig. 6, Diagramm (b), liegt das erste Hindernis **H** so, dass ein derartiges Fahrmanöver nicht möglich ist, da die beiden Hindernisse **H**, **H'** zu nahe beieinander liegen. Daher kann - wie in dem in Diagramm (a) der Fig. 6 dargestellten Beispiel - ebenfalls ein Kriterium zum Beenden des zweiten Konturfolgemodus mit einem großen Mindestwinkel festgelegt und verwendet werden.

[0061] Ob solch ein Mindestwinkel benötigt wird, kann beispielsweise basierend auf der Position, Form und Größe des ersten Hindernisses **H** bestimmt werden. Alternativ kann auf das Festlegen eines Mindestwinkels verzichtet werden (wie in dem in Diagramm (d) der Fig. 5 dargestellten Beispiel). Der Mindestwinkel kann beispielsweise nachträglich gesetzt werden, wenn in dem zweiten Konturfolgemodus festgestellt wird, dass sich ein Hindernis **H'** in dem Bereich **S** befindet, und somit eine Rotation des Roboters blockiert. Alternativ oder zusätzlich kann der große Mindestwinkel nachträglich festgelegt werden, wenn in dem zweiten Konturfolgemodus festgestellt wird, dass das Hindernis **H** aufgrund des Abstands zum Roboter dessen Rotation nicht mehr blockiert, jedoch aufgrund des Hindernis **H'** das Kriterium zum Beenden des zweiten Konturfolgemodus nicht erfüllbar ist. Das Kriterium kann also während einer Fahrt im zweiten Konturfolgemodus aktualisiert werden.

[0062] Das Kriterium zum Beenden des zweiten Konturfolgemodus kann zusätzlich zur Bewertung einer möglichen Bewegung aufgrund der Informationen über die Umgebung des Roboters (insbesondere Kartendaten) die kollisionsfreie Durchführung dieser geplanten Bewegung umfassen. Dies bedeutet, dass erst mit der erfolgreichen Durchführung der Bewegung der zweite Konturfolgemodus beendet würde. Wenn es während der Bewegung zu einer unerwarteten Kollision kommt, so würde unmittelbar der zweite Konturfolgemodus die Steuerung des Roboters **100** entlang der Kontur der Wand **W** fortsetzen (in Rückwärtsrichtung). Die Informationen über die Kollision würden in die Informationen über die Umgebung des Roboters und insbesondere in die Kartendaten übernommen werden, und somit für die Steuerung des Roboters im Folgenden zur Verfügung stehen. Es sei angemerkt, dass der bis zur Kollision durchgeführte Teil der Bewegung in der Regel in dem zweiten Konturfolgemodus wieder rückgängig gemacht werden kann, obwohl dies nicht explizit implementiert ist. Vielmehr ist dies eine Eigenschaft des Konturfolgemodus, der den Roboter **100** in eine weitgehend parallele Ausrichtung zu der zu folgenden Kontur steuert.

[0063] Es sei angemerkt, dass in den in den Fig. 5 und Fig. 6, Diagramm (a) und (b) dargestellten Beispielen die Kontur **W** immer als Gerade dargestellt ist, und somit der Roboter gerade zurücksetzt. Im All-

gemeinen ist die Kontur der Wand **W** (oder eines anderen Hindernisses) nicht notwendigerweise gerade, sondern kann Kurven und Ecken enthalten, denen der Roboter im zweiten Konturfolgmodus ebenfalls zurück folgen würde. Das Beispiel in den Diagrammen (c) und (d) der Fig. 6 illustriert einen Fall mit nichtgeradliniger Kontur **W**, der der Roboter in einem ersten Konturfolgmodus folgt, bis ein Hindernis **H** die weitere Ausführung der Konturfolgefahrt blockiert (siehe Diagramm (c) der Fig. 6). Im darauffolgenden zweiten Konturfolgmodus fährt der Roboter entlang der Kontur **W** zurück bis er sich soweit drehen kann, dass er an dem Hindernis **H** vorbeifahren kann (Kriterium zum Beenden des zweiten Konturfolgmodus). Danach kann der erste Konturfolgmodus fortgesetzt werden und der Roboter folgt der Kontur des Hindernisses **H**. Dieser Ansatz unterscheidet das hier dargestellte Verfahren von anderen Ansätzen, bei denen ein vordefiniertes Bewegungsmuster (Manöver) wie beispielsweise einfaches gerades Rückwärtsfahren verwendet wird. Dieser Fall ist in Diagramm (e) der Fig. 6 dargestellt; durch ein einfaches Zurücksetzen tritt in dem mit **Z** markierten Bereich eine Kollision auf. Diagramm (f) der Fig. 6 illustriert ein weiteres Beispiel einer Sackgassensituation, in der ein Entkommen aus der Sackgasse durch ein einfaches vordefiniertes Bewegungsmuster wie z.B. Zurücksetzen und Drehen nicht kollisionsfrei möglich ist. Zudem kann der Roboter auf dynamische Änderungen durch Bewegungen (z. B. eines Menschen oder Tieres) in seiner Umgebung (die er beispielsweise mit der Sensoreinheit 120 erfasst und zur Aktualisierung seiner Kartendaten nutzt) direkt reagieren. Hierdurch ist das hier dargestellte Verfahren deutlich flexibler und vielseitiger einsetzbar.

[0064] Während der Roboter der Kontur der Wand **W** (oder eines anderen Hindernisses) im zweiten Konturfolgmodus folgt, kann es passieren, dass der zweite Konturfolgmodus ebenfalls keine weitere Bewegung ermöglicht. Beispielsweise ist dies möglich, wenn auf drei Seiten des Roboters ein Hindernis liegt, also insbesondere die Wand **W**, deren Kontur gefolgt wird, ein Hindernis, welches die weitere Rückwärtsfahrt verhindert, als auch ein Hindernis **H'**, so dass das zur Beendigung des zweiten Konturfolgmodus notwendige Kriterium nicht erfüllt ist. In diesem Fall kann die Richtung erneut geändert werden, so dass der Roboter wieder in die ursprüngliche Richtung in einem dritten Konturfolgmodus fährt. Um eine weitgehend gleiche Wiederholung des vorangegangenen Fahrmusters, welches in die Sackgasse führte, zu vermeiden, kann beispielsweise die Seite, auf welcher der Roboter der Kontur folgt gewechselt werden. Hierdurch wird sich der Roboter von der Kontur **W** lösen, um beispielsweise der Kontur des Hindernis **H'** (welches die Erfüllung des zur Beendigung des zweiten Konturfolgmodus notwendigen Kriteriums blockiert) zu folgen, und eine Position erreichen, die beispielsweise eine Fortsetzung des ersten Kon-

turfolgmodus ermöglicht. Es kann ein neues Kriterium zum Beenden des dritten Konturfolgmodus gesetzt werden. Alternativ kann das zuvor gesetzte Kriterium zum Beenden des zweiten Konturfolgmodus beibehalten bzw. übernommen werden.

[0065] Das Vorgehen entspricht hierbei im Wesentlichen dem zuvor mit Blick auf Fig. 4 beschriebenen Verfahren, mit dem einzigen Unterschied, dass dem ersten Konturfolgmodus 10 schon ein weiterer voraus gegangen ist. Prinzipiell lässt sich dieses Vorgehen mit einem vierten, fünften, usw. Konturfolgmodus wiederholen, bis der Roboter einen Weg aus der Sackgassen-Situation gefunden hat. Im Allgemeinen unterscheiden sich die Kontur-Folge-Modi zumindest durch eines der folgenden Merkmale:

- die Richtung, in der einer Kontur gefolgt wird,
- der Seite des Roboters, die der Kontur zugewandt ist,
- Änderung von Parametern zur Navigation wie z. B. Konturfolgeabstand d , Sicherheitsabstand d_s zu Hindernissen, Geschwindigkeit,
- die Priorität, mit der die Berührung von Hindernissen (Kollisionen) vermieden wird,
- die Roboterform, die zur Bestimmung von Kollisionen berücksichtigt wird (z.B. kann ein Sicherheitsabstand bei kartenbasierten Auswertungen in Form einer virtuell vergrößerten Gehäuseform des Roboters berücksichtigt werden),
- die Regeln zur Generierung von Bewegungen entlang der Kontur, und
- die Interpretation und Bewertung von Kartendaten.

[0066] Durch Veränderung des Konturfolgeabstand d und/oder des Sicherheitsabstands d_s (verkleinern oder vergrößern) kann der Roboter mehr Bewegungsfreiraum erlangen. Analog kann durch Anpassung der Geschwindigkeit des Roboters die Genauigkeit der Navigation erhöht werden, wodurch der Roboter beispielsweise einfacher durch enge Stellen navigieren kann oder besser auf Fahrfehler beispielsweise aufgrund des Bodenbelags (z. B. Reibung und Drift) reagieren kann.

[0067] Die zu beachtende Roboterform muss bei einer geänderten Fahrtrichtung entsprechend gespiegelt beachtet werden. Beispielsweise kann bei einem D-förmigen Roboter während der Konturfolgefahrt der Rotationsfreiheitsgrad eingeschränkt sein (u. a. abhängig vom Konturfolgeabstand d). Wenn hierbei die flache Seite in die Fahrtrichtung zeigt, so ist eine Drehung zur Kontur hin nicht oder nur eingeschränkt möglich. Wenn andererseits die flache Seite entgegen die Fahrtrichtung zeigt, so ist eine Drehung (im Stand) von der Kontur weg eingeschränkt. Dies führt direkt dazu, dass auch die Regeln zur Er-

zeugung der Bewegung entlang der Kontur entsprechend geändert werden.

[0068] In einigen Ausführungsbeispielen kann es passieren, dass der Roboter **100** mit einer kollisionsvermeidenden Strategie keinen Weg aus der Sackgasse findet. Dies kann beispielsweise daran erkannt werden, dass der Roboter mehrmals erfolglos den Konturfolgemodus (insbesondere Richtung und/oder die Seite des Roboters, die der einer Kontur zugewandt ist) wechselt, ohne dass das Kriterium zum Beenden des jeweiligen Konturfolgemodus erfüllt werden konnte. Ursache hierfür können beispielsweise fehlerhafte Sensor- und/oder Kartendaten sein, wodurch der Roboter einen Punkt in der realen Umgebung als durch ein Hindernis blockiert ansieht, welcher jedoch frei befahrbar ist. In solch einem Fall kann die kollisionsvermeidende Strategie aufgegeben werden, und durch eine auf Kontakt fahrende Strategie ersetzt werden.

[0069] Die Punkte, an denen der Roboter hierbei ein Hindernis berührt, können ebenfalls in den Kartendaten gespeichert und für die weitere Steuerung des Roboters verwendet werden. In einer exemplarischen Implementierung des Verfahrens zur Steuerung des Roboters in einem Konturfolgemodus können der erste Konturfolgemodus und der zweite Konturfolgemodus (als auch weitere Konturfolgemodi) jeweils eigenständige Softwaremodule sein. Alternativ oder zusätzlich können mehrere Konturfolgemodi in einem Softwaremodul implementiert sein, welches mit unterschiedlich gesetzten Parametern gestartet werden kann.

[0070] Es sei angemerkt, dass der Roboter sich auch in einer Sackgassen-Situation befinden kann ohne zuvor eine Konturfolgefahrt durchgeführt zu haben. Auch hierbei ist es nützlich, einer Kontur rückwärts zu folgen, bis der Roboter feststellt, dass er aus der Sackgasse herausfahren kann bzw. heraus gefahren ist. Beispielsweise kann hierfür eine übergeordnete Steuerungs-Instanz zur Planung der Funktion des Roboters einen ersten Konturfolgemodus starten, der den Roboter in die bevorzugte Richtung (Vorwärtsrichtung) entlang der Kontur steuern soll. Bevor der Roboter eine Bewegung ausführt, kann es passieren, dass er feststellt, dass in diesem ersten Konturfolgemodus keine Bewegung ausgeführt werden kann, weshalb ein zweiter Konturfolgemodus in die entgegengesetzte Richtung gestartet und ein Kriterium zur Beendigung desselben festgelegt und verwendet wird. Alternativ oder zusätzlich kann die übergeordnete Steuerungs-Instanz zur Planung der Funktion des Roboters direkt den zweiten Konturfolgemodus starten und nach vorgebbaren Kriterien wieder beenden.

[0071] Fig. 7 illustriert anhand eines weiteren, etwas komplexeren Beispiels das Verfahren zur Steuerung

des autonomen, mobilen Roboters in einer Sackgassen-Situation, die geometrisch etwas komplizierter ist als in den vorherigen Beispielen. Dieses Beispiel macht auch deutlich, dass einfache Ansätze wie z.B. das Durchführen eines fest vordefinierten Bewegungsmusters nicht immer geeignet sind, um eine Sackgassen-Situation aufzulösen. Die Diagramme (a) bis (d) in Fig. 7 zeigen den Roboter **100** in aufeinanderfolgenden Positionen während er sich in einem ersten Konturfolgemodus entlang der Kontur **W** bewegt, wobei die Kontur **W** rechts vom Roboter liegt (d.h. die rechte Seite des Roboters **100** ist der Kontur **W** zugewandt). Im vorliegenden Beispiel hat die Kontur **W** einen „Knick“, und der Roboter folgt der Kontur über den Knick hinaus (vgl. Diagramme (b) und (c) Fig. 7). In der in Fig. 7, Diagramm (d), dargestellten Situation hat der Roboter **100** eine Position erreicht, in der eine weitere Bewegung in den ersten Konturfolgemodus nicht mehr möglich ist. Die Steuerung **150** des Roboters **100** wechselt folglich in den zweiten Konturfolgemodus, in dem die Fahrtrichtung rückwärts ist. Der Roboter folgt der Kontur **W** zurück und erreicht eine weitere Sackgassen-Situation an dem erwähnten Knick der Kontur **W** (siehe Diagramm (e) in Fig. 7); sowohl die Fortsetzung einer Rückwärtsfahrt als auch eine größere Drehung (z.B. um 45°) sind blockiert.

[0072] Als Reaktion auf diese zweite Sackgassen-Situation wird auch der zweite Konturfolgemodus beendet und die Steuerung **150** des Roboters **100** wechselt in einen dritten Konturfolgemodus, in dem sowohl die Bewegungsrichtung als auch die Seite des Roboters, auf der die Kontur liegt (der im Abstand d gefolgt werden soll) im Vergleich zum zweiten Konturfolgemodus „invertiert“ ist (Vorwärtsbewegung statt Rückwärtsbewegung, Kontur links statt rechts). Die Reaktion des Roboters ist in den Diagrammen (f) bis (g) in Fig. 7 dargestellt; der Roboter **100** dreht sich zu der Kontur zu seiner Linken hin und richtet sich in dem Konturfolgeabstand d an dieser aus, bis die Vorwärtsbewegung erneut blockiert ist (Diagramm (g) der Fig. 7). Als Reaktion auf diese dritte Sackgassen-Situation wird der dritte Konturfolgemodus beendet und die Steuerung **150** des Roboters **100** wechselt in einen vierten Konturfolgemodus, wobei erneut die Bewegungsrichtung gewechselt wird (Rückwärtsbewegung, Kontur links bleibt erhalten). Der Roboter kann sich hierbei rasch parallel zu der Kontur zu seiner Linken hin in dem Konturfolgeabstand d ausrichten. Ausgehend von der in Diagramm (h) in Fig. 7 dargestellten Situation kann der Roboter der Kontur zu seiner Linken rückwärts (im vierten Konturfolgemodus) folgen bis ein Kriterium zum Beenden des Konturfolgemodus erfüllt wird, was in der in Diagramm (i) in Fig. 7 gezeigten Situation der Fall ist. Der Roboter kann sich um einen (vordefinierbaren) Winkel drehen und kann einer weiteren Kontur (in Diagramm (j) der Fig. 7 senkrecht verlaufenden Kontur) im ersten Konturfolgemodus (Vorwärtsbewe-

gung, Kontur zur Rechten des Roboters) folgen. Die strichlierte Linie zeigt den zurückgelegten Weg des Mittelpunktes „x“.

[0073] Das Diagramm (k) der Fig. 7 zeigt eine gegenüber den Diagrammen (a) bis (j) geringfügig modifizierte Situation, welche der Roboter auf ähnliche Weise erreicht wie in den Diagrammen (a) bis (j) dargestellt. In dem dargestellten Beispiel kann der Roboter nach der Rückwärtsfahrt (im vierten Konturfolgemodus) sich im Uhrzeigersinn drehen (sodass die Kontur wieder rechts des Roboters liegt) und die Konturfolgefahrt im ersten Konturfolgemodus fortsetzen.

[0074] Elementarbewegungen: Im Folgenden wird eine mögliche Form der Steuerung eines autonomen mobilen Roboters **100** in einer Konturfolgefahrt dargestellt. Um die Komplexität der zahlreichen Möglichkeiten zur Bewegung des Roboters **100** zu reduzieren, werden zumindest drei Elementarbewegungen eingeführt, die geeignet sind, den Roboter in einer gewünschten Richtung in einem vorgebbaren Konturfolgeabstand entlang einer Kontur zu bewegen. Diese Elementarbewegungen werden basierend auf den Informationen zur Umgebung des Roboters und insbesondere basierend auf den Kartendaten bewertet. Die Elementarbewegung mit der besten Bewertung wird ausgewählt. Auf Basis der ausgewählten Elementarbewegung werden Steuerkommandos für die Antriebseinheit **170** erzeugt. Dieses Verfahren nutzt die Vorteile von geplanten Bewegungen und ermöglicht gleichzeitig eine schnelle Reaktion auf Änderungen in der Umgebung (z. B. Bewegung von Menschen oder Tieren) oder Fahrfehler beispielsweise aufgrund des Bodenbelags (Reibung, Drift), indem es nur einen kurzen Planungshorizont und schnelle Wiederholungen der Planung gibt.

[0075] Bei der Bewertung der Elementarbewegungen kann festgestellt werden, dass keine der Elementarbewegungen ausgeführt werden kann oder soll. Beispielsweise kann auf Basis der Kartendaten festgestellt werden, dass keine der Elementarbewegungen kollisionsfrei ausgeführt werden kann. Auch auf Basis weiterer Auswahlregeln kann festgestellt werden, dass keine der Elementarbewegungen sinnvoll ausgeführt werden kann. Ein Beispiel hierfür ist eine weiter unten noch genauer beschriebene Sackgassen-Situation, in der der erste Konturfolgemodus keine weitere Bewegung in die bevorzugte Richtung entlang der Kontur ermöglicht.

[0076] Um den Roboter aus einer Sackgassen-Situation heraus zu steuern, wird ein neuer Konturfolgemodus gestartet, wobei prinzipiell die gleichen oder ähnliche Elementarbewegungen genutzt werden können, jedoch die Richtung der Bewegung invertiert wird. Die Regeln zur Bewertung der Elementarbewegungen können neu festgelegt werden oder weitgehend unverändert beibehalten werden. Wenn

die Regeln zur Bewertung der Elementarbewegung unverändert bleiben, muss lediglich beachtet werden, dass die Kontur des Gehäuses des rückwärtsfahrenden Roboters genutzt wird (z.B. bei dem D-förmigen Roboter ist so die halbkreisförmige Seite in Bewegungsrichtung).

[0077] Fig. 8 zeigt mögliche Elementarbewegungen. Diese umfassen hierbei zumindest:

- eine erste Elementarbewegung: eine gerade Bewegung in die aktuelle Bewegungsrichtung,
- eine zweite Elementarbewegung: eine Drehung zu einer zu folgenden Kontur hin,
- eine dritte Elementarbewegung: eine Drehung von der zu folgenden Kontur weg.

[0078] Die Richtung der Drehung der dritten Elementarbewegung ist also entgegengesetzt zur Richtung der Drehung der zweiten Elementarbewegung. Welche Seite des Roboters **100** der zu folgenden Kontur zugewandt sein soll, kann durch eine übergeordnete Planungsinstanz festgelegt werden, durch welche die Konturfolgefahrt ausgelöst wird. Alternativ oder zusätzlich kann zu Beginn des Konturfolgemodus (beispielsweise aufgrund der Karteninformationen) festgestellt werden, auf welcher Seite des Roboters sich eine Kontur befindet, der gefolgt werden kann bzw. soll. Soll der Roboter einer Wand folgen, so ist die Seite die der Roboter der (Kontur) zuwenden soll in der Regel klar. Wenn der Roboter einem Hindernis ausweichen soll, kann der Roboter theoretisch das Hindernis im Uhrzeigersinn oder gegen den Uhrzeigersinn umfahren, wobei eine bevorzugte Richtung (z.B. im Uhrzeigersinn) vordefiniert sein kann, von der nur in Ausnahmefällen abgewichen wird.

[0079] Fig. 8, Diagramm (a), zeigt eine Geradeausbewegung als erste Elementarbewegung. Hierbei bewegen sich beide Räder **170L**, **170R** um die gleiche Distanz nach vorne. Die bei der ersten Elementarbewegung zurückzulegende Distanz kann eine feste Distanz sein. Für eine höhere Flexibilität kann die zurückzulegende Distanz während der Bewertung der Bewegungen bestimmt werden. Hierbei können beispielsweise eine minimale und/oder eine maximale Distanz für die gerade Bewegung berücksichtigt werden.

[0080] Fig. 8, Diagramm (b), zeigt eine mögliche Variante der zweiten bzw. dritten Elementarbewegung. Hierbei bewegen sich das Rad **170R** und das Rad **170L** in entgegengesetzte Richtung, was zur Folge hat, dass der Roboter sich um seinen zentralen Punkt dreht.

[0081] Fig. 8, Diagramm (c), zeigt eine weitere mögliche Variante der zweiten bzw. dritten Elementarbewegung. Hierbei bewegt sich nur eines der beiden Räder **170L** nach vorne, während das zweite Rad

170R stillsteht. Der gesamte Roboter dreht sich somit um das zweite Rad **170R**. Der zentrale Punkt „x“ bewegt sich auf einer Kreisbahn vorwärts.

[0082] Fig. 8, Diagramm (d), zeigt eine weitere mögliche Variante der zweiten bzw. dritten Elementarbewegung. Hierbei bewegt sich nur eines der beiden Räder **170R** nach hinten, während das zweite Rad **170L** stillsteht. Der gesamte Roboter dreht sich somit um das zweite Rad **170L**. Der zentrale Punkt „x“ bewegt sich auf einer Kreisbahn rückwärts. Der Drehsinn ist jedoch der gleiche wie in Fig. 8, Diagramm (b) und (c).

[0083] Durch passendes Ansteuern der Antriebsräder kann der Roboter auch um andere Punkte gedreht werden, wobei der zentrale Punkt „x“ sich immer auf einer Kreisbahn bewegt. Durch Wahl einer geeigneten Drehbewegung können insbesondere die gewünschten Eigenschaften der Bewegung der Arbeitseinheit **160** (nicht dargestellt) erzielt werden. Beispielsweise kann es im Normalfall wünschenswert sein, dass sich die Arbeitseinheit **160** des Roboters **100** immer vorwärtsbewegt, was beispielsweise durch eine in Diagramm (c) der Fig. 8 dargestellte Bewegung erreicht werden kann. Hierdurch könnte in machen Anwendungen beispielsweise auf Teppichen, auf denen z.B. eine Bürste des Roboters eine deutliche Reinigungsspur hinterlässt, ein schöneres Reinigungsmuster erzielt werden. Durch die in Diagramm (d) der Fig. 8 dargestellte Bewegung würde eine im vorderen Bereich des Roboters angeordnete Reinigungseinheit (vgl. Fig. 3, Diagramm (b)) ein wenig zurückbewegt werden, wodurch eine gründlichere Reinigung erreicht werden kann. Eine zu den in den Diagrammen (b) bis (d) der Fig. 8 dargestellten Drehbewegungen (zur entsprechenden Definition der zweiten bzw. dritten Elementarbewegung) entgegengesetzte Drehung kann durch vertauschen der Antriebsregel (vorwärts/rückwärts) für die beiden Räder **170L**, **170R** erzeugt werden.

[0084] Der bei der zweiten bzw. dritten Elementarbewegung zurückzulegende Drehwinkel kann ein fester Winkel von beispielsweise $0,5^\circ$... 5° sein. Für eine höhere Flexibilität insbesondere zur Ausrichtung des Roboters parallel zu der zu folgenden Kontur kann ein geeigneter Drehwinkel während der Bewertung der Bewegungen bestimmt werden. Hierbei können beispielsweise ein minimaler und/oder ein maximaler Drehwinkel für die Bewegung berücksichtigt werden. Die bei der zweiten und dritten Elementarbewegung genutzte Drehbewegung kann im Wesentlichen die Gleiche sein, wobei sich nur der Drehsinn unterscheidet. Beispielsweise kann bei beiden Elementarbewegungen die in Abbildung Fig. 8, Diagramm (b), dargestellte Drehung auf der Stelle genutzt werden.

[0085] Alternativ können die zweite und dritte Elementarbewegung verschieden gewählt werden (d.h.

es ist nicht nur der Drehsinn verschieden, sondern auch ein anderes Merkmal der Bewegung). Hierdurch kann die Charakteristik der Bewegung an die unterschiedlichen Anforderungen leichter angepasst werden. So kann die zweite Elementarbewegung entsprechend Fig. 8, Diagramm (d), eine kleine Bewegung nach hinten enthalten, um engere Umfahrung (und somit gründlichere Reinigung) von kleinen Hindernissen wie Stuhlbeinen zu erreichen und/oder die dritte Elementarbewegung kann entsprechend Fig. 8, Diagramm (c), eine kleine Bewegung nach vorne enthalten, um eine glatte Bewegung zur Ausrichtung parallel zu einer Wand zu erreichen. Die resultierende Bewegung des Roboters ist ja eine Abfolge einzelner Elementarbewegungen (also z.B. mehrfache Drehung um 1° nach rechts, Vorwärtsbewegung, mehrfache Drehung um 1° nach rechts, Vorwärtsbewegung, etc.) welche, wenn sie nacheinander ausgeführt würden zu einer ruckelnden Bewegung führen würden. Die Steuereinheit **150** kann dazu ausgebildet sein, diese Bewegung zu glätten (z.B. mit einem Moving-Average-Filter).

[0086] Es können auch weitere Elementarbewegungen berücksichtigt werden. Beispielsweise können auch Elementarbewegungen entgegen die aktuelle Bewegungsrichtung (rückwärts) berücksichtigt werden. Um eine glatte Bewegung zu erhalten, können die zu den geplanten Elementarbewegungen zu erstellenden Steuerkommandos zur Umsetzung der Elementarbewegungen zusätzlich geglättet werden.

[0087] Bewertung der Bewegungen: Für die Bewertung von Bewegungen zur Steuerung eines autonomen mobilen Roboters **100** sind verschiedene Methoden an sich bekannt. Beispielsweise können basierend auf in der Umgebung des Roboters erkannten Hindernissen „virtuelle Kräfte“, „virtuelle Potentiale“ oder „virtuelle Kosten“ bestimmt werden. Diese können zur Bewertung der Elementarbewegung genutzt werden, wobei dann jene Bewegung gewählt wird, die einem vorgebbaren Optimum (z.B. Bewegung entlang der virtuellen Kraft, Minimierung von virtuellen Potentialen oder virtuellen Kosten) folgt. Die Wahl der Methode zur Bewertung der Bewegung ist nicht wesentlich für die Ausführung der hier beschriebenen Ausführungsbeispiele.

[0088] Bei der Bewertung der Elementarbewegungen kann es passieren, dass zwei oder mehr Elementarbewegungen gleich bewertet werden. In diesem Fall wird jene Elementarbewegung bevorzugt, die an der Kontur entlangführt. Dies bedeutet für die hier dargestellten Bewegungen, dass bei gleicher Bewertung eine Drehung zur Kontur hin (zweite Elementarbewegung) bevorzugt ausgeführt wird. Bei gleicher Bewertung der geraden Bewegung (erste Elementarbewegung) und der Drehung von der Kontur weg (dritte Elementarbewegung) wird die gerade Bewe-

gung entlang der Kontur (erste Elementarbewegung) gewählt.

[0089] Bei der Bewertung der Elementarbewegungen können eine oder mehrere vorhergehende Elementarbewegungen berücksichtigt werden. So kann beispielsweise „verboten“ sein, die letzte Bewegung rückgängig zu machen. Insbesondere wenn die zweite und dritte Elementarbewegung (Drehung zur Kontur hin und von der Kontur weg) eine Drehung im Stand (vgl. Diagramm (b) in Fig. 8) ist, kann eine direkte Abfolge dieser Elementarbewegungen verboten werden. Weitere Regeln zur Abfolge der Elementarbewegung können aufgestellt werden, um ein glatteres Fahrverhalten des Roboters entlang der zu folgenden Kontur zu erreichen.

[0090] Ein wesentlicher Aspekt bei der Bewertung der Bewegung ist die Vermeidung von Kollisionen. Beispielsweise kann eine Bewegung, die zu einer Kollision mit zumindest einem Punkt eines Hindernisses führen würde, generell verboten sein, oder mit sehr hohen Kosten belegt werden. Zusätzlich kann es sinnvoll sein, bei der Bewertung der Bewegung die Position der Kontur des zu folgenden Hindernisses zu berücksichtigen. Fig. 8 zeigt in den Diagrammen (a) bis (d) vier vereinfachte Beispiele. Wenn wie in Diagramm (a) der Fig. 9 der Abstand zwischen der zu folgenden Kontur **W** und dem autonomen mobilen Roboter **100** größer als ein vorgebbare Abstand d (Konturfolgeabstand) ist, so soll sich der Roboter zu der Kontur hindrehen (zweite Elementarbewegung). Wenn - wie in Diagramm (b) der Fig. 9 gezeigt - der Abstand zwischen der zu folgenden Kontur **W** und dem autonomen mobilen Roboter **100** ungefähr gleich dem vorgebbare Konturfolgeabstand d ist (z.B. innerhalb eines bestimmten Toleranzbereichs $d \pm \epsilon$), so bewegt sich der Roboter im Wesentlichen gerade aus parallel zur Wand (erste Elementarbewegung). Wenn - wie in Diagramm (c) der Fig. 9 gezeigt - der Abstand d zwischen der zu folgenden Kontur **W** und dem autonomen mobilen Roboter **100** kleiner als der vorgebbare Konturfolgeabstand d ist, so soll sich der Roboter von der Kontur wegdrehen (dritte Elementarbewegung).

[0091] Wie in Diagramm (d) der Fig. 9 dargestellt, ist der Roboter **100** im Allgemeinen nicht parallel zu der Kontur **W** ausgerichtet. Entsprechend muss eine Steuerung (d.h. eine automatisierte Auswahl einer Sequenz von Elementarbewegungen) des Roboters **100** so erfolgen, dass der Roboter **100** weitgehend parallel zu der Kontur **W** ausgerichtet wird. Hierzu kann beispielsweise die Orientierung **O** der Kontur **W** bestimmt werden, und basierend auf der Orientierung **O** der Kontur und der Orientierung des Roboters die Elementarbewegung gewählt werden, so dass eine parallele Ausrichtung in dem vorgebbaren Konturfolgeabstand d erfolgt (Orientierung des Roboters **100** und der Kontur **W** sind dann gleich). An die-

ser Stelle sei noch einmal darauf hingewiesen, dass die Kontur **W** im Allgemeinen nicht geradlinig ist, auch wenn sie in den Abbildungen vereinfachend als Gerade dargestellt ist.

[0092] Die Orientierung **O** der Kontur **W** kann beispielsweise als Verbindungsvektor zweier Punkte der Kontur, als Regressionsgerade einer Auswahl an mehreren Punkten, als Tangente an die Kontur oder ähnlich bestimmt werden. Die Kartierung der Umgebung kann z.B. mittels eines Algorithmus zur Feature-Extraktion geschehen, bei der Teile der Kontur eines Hindernisses (insbesondere einer Wand) als Linie (oder Fläche) erfasst und gespeichert werden. Die Orientierung **O** der Kontur hat in der Regel eine natürliche Richtung, die sich beispielsweise aus der Richtung ergibt, aus der das Hindernis beobachtet wurde, und/oder aus der Richtung, der der Roboter entlang der Kontur folgen soll. Wenn die Kontur als ungerichtetes Objekt dargestellt wird (z. B. Linie), so ist trotzdem die Ausrichtung des Roboters parallel zu der Kontur eindeutig durch die Wahl der Seite des Roboters gegeben, die bei der Konturfolgefahrt der Kontur **W** zugewandt sein soll (und damit auch durch die Festlegung der Drehrichtung der zweiten Elementarbewegung).

[0093] Zur Bewertung einer Elementarbewegung kann die Umgebung des Roboters in einzelne Sektoren eingeteilt werden. Eine mögliche Aufteilung umfasst beispielsweise einen Sektor, in dem sich für eine kollisionsfreie Durchführung der Elementarbewegung kein Hindernis befinden darf, ein Sektor zur Analyse der zu folgenden Kontur und/oder einen Sektor, zur Analyse weitergehender Bewegungsmöglichkeiten. Die Unterteilung der Umgebung des Roboters in Sektoren ist exemplarisch in den Diagrammen (a) bis (c) der Fig. 10 dargestellt.

[0094] Diagramm (a) der Fig. 10 zeigt exemplarisch die Sektoren in der Umgebung des Roboters für die Bewertung einer geraden Bewegung (erste Elementarbewegung). Dargestellt ist beispielhaft eine Wand **W** mit einer Ecke als zu folgende Kontur.

[0095] Sektor I (in Diagramm (a) schattiert dargestellt) beschreibt jene Fläche, die der Roboter für eine geradlinige (Vorwärts-) Bewegung um die Mindestlänge $l_{\min}$ benötigt. Wenn in diesem Sektor zumindest ein Teil oder ein Punkt eines Hindernisses liegt, so kann die Bewegung nicht kollisionsfrei ausgeführt werden, und wäre somit verboten.

[0096] Sektor II ist ein Bereich neben dem Roboter auf der Seite der zu folgenden Kontur. Dieser Sektor ist beispielsweise ausgehend von der Roboterseite so breit, wie der Konturfolgeabstand d . Wenn sich in diesem Sektor II zumindest ein Teil oder ein Punkt eines Hindernisses befindet, so ist dies nicht notwendigerweise Ausschlusskriterium für die Durchführung

der Bewegung. Jedoch kann z.B. bei der Bewertung geprüft werden, ob der Roboter beispielsweise durch die dritte Elementarbewegung den Abstand zur zu folgenden Kontur **W** vergrößern sollte. Wenn beispielsweise die Kontur **W** deutlich in den Sektor **II** hineinragt, so kann die Bewertung z.B. dazu führen, dass sich der Roboter von der Kontur weg bewegt. Wenn jedoch nur eine kleine Ecke oder ein einzelner Punkt nahe am Rand des im Sektors **II** liegt, so sollte dies nicht zu einer Ausweichbewegung führen, um eine schlingernde Bewegung zu vermeiden. Hierzu können beispielsweise im Fall einer kostenbasierten Bewertung der dritten Elementarbewegung vorgebbare Grundkosten berücksichtigt werden, die den Kosten einer kleinen in Sektor **II** hineinragende Ecke entsprechen. Die Kosten können beispielsweise basierend auf der Länge und/oder des Flächenanteils des in Sektor **II** ragenden Teils der Kontur bestimmt werden. Wenn Teile der zu folgenden Kontur auf dem Rand von Sektor **II** liegen kann dies einen Bonus (z.B. negative Kosten) geben. Wenn keine Kontur in Sektor **II** und insbesondere im Randbereich der zu folgenden Kontur liegen, so kann dies ebenfalls mit Kosten belegt werden.

[0097] Sektor **III** ist ein Bereich, in dem eine weitere mögliche Bewegung des Roboters geprüft wird. Beispielsweise wird geprüft, ob und wie weit der Roboter sich weiter geradlinig vorwärts bewegen kann, ohne dass es zu einer Kollision kommt. Dies ist beispielsweise durch einen maximalen Planungshorizont $I_{\max}$ begrenzt. Innerhalb dieses Sektors **III** kann der Roboter beispielsweise eine Strecke $I_{\min} < l < I_{\max}$ bestimmen, die er kollisionsfrei befahren kann. Hierbei kann z.B. ein Sicherheitsabstand d_s zu einem vor dem Roboter liegenden Hindernis berücksichtigt werden.

[0098] Sektor **IV** ist ein Bereich neben dem Roboter **100** auf der der Kontur abgewandten Seite des Roboters. In der Regel wird sich hier kein Hindernis befinden. Falls sich hier zumindest ein Teil eines Hindernisses befinden sollte, kann diese Information genutzt werden, um den Roboter **100** durch den Korridor zwischen diesem Hindernis der Kontur **W** hindurch zu bewegen.

[0099] Diagramm (b) der Fig. 9 zeigt exemplarisch die Sektoren für die Durchführung einer Drehung (zweite/dritte Elementarbewegung). Es sei angemerkt, dass für eine bessere Darstellung eine vergleichsweise große Drehung gewählt wurde. Für die tatsächliche Robotersteuerung kann die Drehung deutlich kleiner gewählt werden.

[0100] Der in Diagramm (b) der Fig. 10 schattiert dargestellte Sektor **I** ist jener Bereich, der vom Roboter während einer Drehung im Stand (vgl. Diagramm (b), Fig. 8) überstrichen wird. Dieser hängt stark von der Form des Roboters ab. Für einen runden symmetrischen Roboter (vgl. Diagramm (a) aus Fig. 3)

ist dieser Sektor nicht vorhanden, da sich aufgrund der Symmetrie keine Einschränkung des Rotationsfreiheitsgrads durch nahe Hindernisse ergeben. Für den D-förmigen, in Diagramm (b) der Fig. 10 dargestellten Roboter **100** zerfällt der Sektor **I** in zwei unabhängig Teile, die durch die beiden Ecken (rechts und links vorne am Roboter) bestimmt sind. Der hinter Teil des Roboters ist rund gestaltet, so dass es keine Einschränkung des Rotationsfreiheitsgrads in diesem Bereich gibt. Wenn die Drehung nicht um den zentralen Punkt sondern um einen anderen Punkt (vgl. Diagramme (c) und (d), Fig. 8) durchgeführt wird, so wird Sektor **I** entsprechend größer und verschoben.

[0101] Zusätzlich kann mit Sektor **III** die Möglichkeit einer Bewegung wie eine gerade Bewegung nach Abschluss der Drehbewegung in die Bewertung mit einfließen. Hierbei können auch Hindernisse in den neben dem Roboter befindlichen Sektoren **II** und **IV** bewertet werden. Beispielsweise kann vorgesehen sein, eine Drehung hin zur Kontur **W** (zweite Elementarbewegung) nur dann als geeignet zu beurteilen, wenn eine anschließende gerade Bewegung um eine vorgebbare Distanz (z. B. $I_{\min}$) ausführbar ist. Der Winkel der Drehung und die Distanz $I_{\min}$ der auf die Drehung folgenden translatorischen Bewegung können aufeinander abgestimmt sein. Wenn beispielsweise der Roboter von der Kontur weiter entfernt ist als der Konturfolgeabstand, soll sich der Roboter zur Kontur hin drehen können, wenn der Abstand des Roboters von der Kontur kleiner oder gleich dem Konturfolgeabstand ist, soll eine Auswahl einer Elementarbewegung zur Kontur hin verhindert werden (weil nach einer Drehung keine Geradeausbewegung mehr möglich ist). Dieses Verhalten kann durch eine Abstimmung von $I_{\min}$ und Drehwinkel erreicht werden.

[0102] Es sei angemerkt, dass, wenn eine gerade Bewegung nach einer Drehung nicht möglich ist, die geplante Elementarbewegung (also die Drehung) im nächsten Schritt möglicher Weise rückgängig gemacht werden müsste, was in der Regel unerwünscht ist und vermieden werden sollte. Mit der erwähnten Bedingung, dass nach der Rotation eine gerade Bewegung mit einer vorgebbaren Distanz möglich sein soll, kann dies vermieden werden. Dies bedeutet nicht, dass die gerade Bewegung direkt im Anschluss auch tatsächlich ausgeführt werden muss. Alternativ oder zusätzlich kann der Roboter auch eine weitere zusätzliche Drehung prüfen, analog zu der geraden Bewegung um die Strecke $I_{\min}$ in dem in Fig. 10, Diagramm (a), dargestellten Beispiel.

[0103] In einigen Anwendungen autonomer mobiler Roboter ist es erwünscht, dass der Roboter möglichst wenig rückwärtsfahren muss. Die Häufigkeit von Rückwärtsfahrten kann reduziert werden, wenn der Roboter für jede (Elementar-) Bewegung, insbesondere jede gerade Bewegung (erste Elementarbe-

wegung), prüft, ob nach Durchführung der geradlinigen Bewegung durch Sektor I eine vollständige oder teilweise Drehung kollisionsfrei möglich ist. Sektor III zeigt beispielhaft den Sektor, in dem sich kein Hindernis befinden darf, damit der Roboter sich um den zentralen Punkt drehen kann. Sektor III' zeigt beispielhaft den Sektor, in dem sich kein Hindernis befinden darf, damit der Roboter eine Kreisbewegung um einen Punkt oberhalb des zentralen Punktes durchführen kann (vgl. den Fall aus Diagramm (c) der Fig. 8).

[0104] Zusätzlich kann z.B. bei Reinigungsrobotern die bearbeitete Fläche (also z.B. die von der Bearbeitungseinheit 160 überstrichene Fläche) als Karteninformation gespeichert und zur Bewertung der Bewegungen des Roboters genutzt werden. Gleichzeitig kann ein Bearbeitungsgewinn für eine zu bewertende Elementarbewegung ermittelt werden und zu ihrer Bewertung herangezogen werden. Beispielsweise kann auf diese Weise erkannt werden, wenn der Roboter vollständig entlang der Kontur gefahren ist, und einen zuvor bereits bearbeiteten Bereich (insbesondere aber nicht ausschließlich den Startpunkt der Konturfolgefahrt) erneut erreicht. Der einer (Elementar-) Bewegung zugeordnete Bearbeitungsgewinn kann z.B. jene (noch nicht bearbeitete) Bodenfläche sein, welche bei Durchführung der Bewegung zusätzlich bearbeitet werden würde. Diese Fläche kann auch (z.B. abhängig vom Bodenbelag oder vom Raum, in dem sich der Roboter befindet) gewichtet werden.

[0105] Beispielsweise kann der Roboter zu einer Kontur, die er zuvor schon gereinigt hatte, aber nochmals entlang fahren muss, einen größeren Abstand halten. Des Weiteren kann ein größerer Konturfolgeabstand eine geringere Genauigkeit der Navigation erfordern, um unbeabsichtigte Kollisionen zu vermeiden. So könnte beispielsweise der Planungshorizont und/oder die Geschwindigkeit des Roboters vergrößert werden. Beispielsweise kann ein größerer Abstand zu Hindernissen vor dem Roboter eingehalten werden, wodurch der Roboter nicht mehr so häufig und so weit in Ecken und andere enge Stellen (potentielle Sackgassen) hinein fährt.

[0106] Parameterwahl: Die hier beschriebenen Beispiele betreffend Verfahren zur Steuerung eines Roboters mittels drei oder mehr verschiedenen Elementarbewegungen und der Bewertung dieser Bewegungen nach einfachen vorgebbaren Regeln ist ein sehr mächtiges Werkzeug, mit dem prinzipiell zahlreiche Bewegungsprofile in einer Konturfolgefahrt für verschiedene Einsatzzwecke erzeugt werden können. Die Wahl der zahlreichen Parameter (Bewertungsregeln, Drehpunkt der Drehbewegung, zurückzulegende Distanzen und Drehwinkel) kann jedoch schnell unübersichtlich und komplex werden. Mit Hilfe von Simulationen kann das Verhalten eines Roboters mit einem gegebenen Parametersatz prinzipiell analysiert

und an ein gewünschtes Verhalten angepasst werden.

[0107] Zusätzlich ermöglicht der Einsatz von Optimierungsverfahren wie z.B. Methoden des maschinellen Lernens (machine learning) eine zumindest teilweise automatisierte Bestimmung der Parameter. Beispielsweise können bestimmte Szenarien (verschiedene Anordnungen von Hindernissen wie Wände und Stuhlbeine) vorgegeben werden und mit vorgebbaren Maßfunktionen optimiert werden. Beispielsweise kann die bearbeitete Fläche in der Nähe einer Wand maximiert oder die benötigte Zeit minimiert werden. Zusätzlich oder alternativ können von einem Menschen gewünschte Bewegungsmuster (z.B. basierend auf Marktstudien ermittelt) vorgegeben werden. Die Parameter können dahingehend optimiert werden, dass der Roboterpfad (simuliert und/oder im Test) möglichst nah an dem vorgegebenen Bewegungsmuster liegt.

[0108] Nicht-sichtbare Hindernisse: Die Konturfolgefahrt kann wie beschrieben weitgehend kollisionsfrei basierend auf den Informationen der Umgebung und auch basierend auf den Kartendaten geplant und ausgeführt werden. Insbesondere die Bewertung von Elementarbewegungen oder der Kriterien zum Beenden eines Konturfolgemodus kann kartenbasiert erfolgen. Zusätzlich kann der Roboter über geeignete Notfallroutinen (z.B. von der Steuereinheit 150 ausgeführtes Softwaremodul, vgl. Fig. 2) verfügen, die bei unvorhergesehenen Ereignissen gestartet werden können. Beispielsweise kann eine geplante Bewegung abgebrochen und der Roboter 100 hierdurch gestoppt werden, um einen Unfall zu vermeiden oder in seinen Auswirkungen zu begrenzen. Die Information über das unvorhergesehene Ereignis kann beispielsweise in die Kartendaten aufgenommen und für die weitere Steuerung des Roboters genutzt werden. Nach Abschluss der Notfallroutine kann der so unterbrochene Konturfolgemodus fortgesetzt werden oder die aktuelle Aufgabe des Roboters neu geplant werden.

[0109] Solch ein unvorhergesehenes Ereignis ist beispielsweise die Detektion einer Absturzkante wie bei einer Treppe, die nur erkannt wird, wenn der Roboter sich dieser nähert und/oder zumindest teilweise mit einem entsprechenden Sensor überfahren hat. Ein weiteres Beispiel für ein unvorhergesehenes Ereignis ist das Berühren eines Hindernisses (z.B. eine Kollision). Dies kann geschehen, weil das Hindernis vorher nicht mit dem Navigationssensor erkannt und/oder in die Kartendaten aufgenommen wurde. Dies kann bei niedrigen, transparenten oder spiegelnden Hindernissen vorkommen. In einigen Fällen, kann auch ein Fahrmanöver beispielsweise aufgrund eines schlechten Bodens nicht wie geplant ausgeführt worden sein, wodurch der Roboter mit einem zuvor detektierten Hindernis unbeabsichtigt kollidiert. Es kann

auch vorkommen, dass sich ein Hindernis bewegt (z.B. durch Einfluss eines Menschen oder eines Tieres) und dadurch eine Kollision verursacht wird.

[0110] Zusätzlich zum sofortigen Stopp des Roboters, kann im Rahmen der Notfallroutine eine weitere an das unerwartete Ereignis (das die Notfallroutine ausgelöst hat) angepasste standardisierte Bewegung ausgeführt werden. Beispielsweise kann die letzte Bewegung zumindest soweit invertiert (umgekehrt) werden, dass der Roboter einen sicheren Abstand zu einer detektierten Absturzkante einnimmt, und/oder dass ein taktiler Sensor zur Detektion einer Kollision oder einer Berührung eines Hindernisses wieder gelöst ist (also kein Hindernis mehr detektiert). Beispielsweise kann der Roboter einige Zentimeter rückwärtsfahren. Wenn das unerwartete Ereignis bei einer Drehung auftritt, kann der Roboter sich in die entgegengesetzte Richtung drehen.

[0111] Nach Abschluss der standardisierten Bewegung kann die normale Konturfolgefahrt wieder aufgenommen werden. Die Ursache des unerwarteten Ereignisses kann in die Karte eingetragen werden, damit sie für die weitere Bewertung der Bewegung des Roboters berücksichtigt werden kann. Dies ist beispielsweise der Ort, an dem die Absturzkante detektiert wurde. Dieser Ort kann basierend auf der Pose (Position und Orientierung) des Roboters und der Lage (im Roboter) des Sensors, der die Absturzkante detektiert hat, bestimmt werden. Im Allgemeinen sind dies ein oder mehrere Punkte, welche wie Punkte einer Kontur eines Hindernisses behandelt werden können.

[0112] Ein durch eine Berührung bzw. Kollision aufgetretenes unerwartetes Ereignis wird ebenfalls in die Karte eingetragen. Dabei kann es wünschenswert sein, dass der taktile Sensor zur Detektion einer Kollision oder einer Berührung eine vergleichsweise gute Ortsauflösung besitzt, so dass die Position, an der das Hindernis berührt wurde, mit hoher Genauigkeit in die Karte eingetragen werden kann. In der Praxis haben taktile Sensoren jedoch oft nur eine grobe Auflösung. In diesem Fall kann der gesamte Teil der äußeren Kontur des taktilen Sensors des Roboters, an dem ein Hindernis das gemessene Sensorsignal möglicherweise erzeugt haben kann, in die Karte eingetragen werden (als geometrische Figur oder in Form von Sample-Punkten). Beispielsweise ist dies je ein Kontaktschalter für verschiedene unabhängige Bereiche des Roboters. Es sei angemerkt, dass aus dem Auslösen zweier Kontaktschalter in naher zeitlicher Abfolge zusätzliche Informationen über den Ort der Kollision abgeleitet werden können, welche in geeigneter Weise in die Karte eingetragen werden können.

[0113] Da die hierdurch in die Karte aufgenommenen Informationen nicht direkt der Position von Hin-

dernissen entsprechen, kann es notwendig sein, sie anders als die zuvor beschriebenen Informationen über Hindernisse zu behandeln. Das heißt, bei der Bewertung der Elementarbewegung kann die Art eines Hindernisses berücksichtigt werden sowie auch mit welchem Sensor die Kontur des Hindernisses (oder ein Teil davon) detektiert wurde. Beispielsweise können die Informationen optimistisch interpretiert werden. Dies bedeutet, dass für die Bewertung einer Elementarbewegung das kleinstmögliche Hindernis mit der am wenigsten störenden Position, welches die Sensorinformation erzeugt haben kann, angenommen wird. Dies kann zu einem weiteren Kontakt mit dem Hindernis führen, wodurch die Anzahl der taktilen Informationen über das nicht detektierte Hindernis zunimmt. Hierdurch kann der Roboter sich mit einer tastenden Bewegung entlang des Hindernisses bewegen.

[0114] Wie bereits in Bezug auf die Handhabung von Sackgassen-Situationen erwähnt kann es notwendig sein, einen Konturfolgemodus zu starten, in dem absichtlich eine Kollision riskiert wird. Dies bedeutet, dass die mit dem Navigationssensor detektierten Informationen über die Umgebung des Roboters und/oder die Kartendaten nicht oder nur eingeschränkt genutzt werden. Die oben beschriebene Notfallroutine beinhaltet damit auch eine Methode zur taktilen Erkundung der Umgebung, auf deren Basis der Roboter sich in einem Konturfolgemodus entlang einer Kontur bewegen kann.

[0115] In einigen Ausführungsbeispielen kann der Roboter so konstruiert sein, dass die Gefahr einer Kollision mit einem detektierten sich bewegenden Hindernis oder aufgrund eines nicht planmäßig ausgeführten Fahrmanövers erkannt wird, bevor es zu einer tatsächlichen Kollision oder Berührung des Hindernisses kommt. Daraufhin kann der Roboter sofort gestoppt werden, wodurch eine Kollision verhindert werden kann.

[0116] Virtuelle Hindernisse: Weitere Beispiele für Hindernisse verschiedener Art, die bei der Bewertung der Elementarbewegung in besonderer Form berücksichtigt werden können, sind durch einen Nutzer in die Umgebung eingebrachte Markierungen mit dem Ziel der Abgrenzung von Bereichen des Robotereinsatzgebietes, die der Roboter nicht befahren darf. Solche Markierungen sind beispielsweise Magnetbänder und Stromschleifen, die ein vom Roboter detektierbares Magnetfeld aufbauen, oder Leitstrahlsender, die einen vom Roboter detektierbaren Lichtstrahl (z. B. Infrarotlaserstrahl) aussenden. Diese Markierungen können mit einem entsprechenden Sensor der Sensoreinheit **120** des Roboters **100** (vgl. **Fig. 2**) erkannt werden, und werden vom Roboter z.B. nicht überquert. Sie stellen somit eine Art Hindernis für den Roboter dar, welches bei der Navigation des Roboters berücksichtigt werden kann. Zusätzlich

kann der Kontur eines derartigen Hindernisses in einer Konturfolgefahrt gefolgt werden.

[0117] Da eine Kollision mit einer (z.B. magnetischen oder optischen) Markierung nicht möglich ist, können sie bei der Bewertung von Elementarbewegungen anders behandelt werden, als beispielsweise Hindernisse, die durch Abstandsmessung oder mit einer Kamera detektiert wurden. So reicht es die Markierung nicht zu überfahren, während eine Einschränkung des Rotationsfreiheitsgrads nicht nötig ist. So kann beispielsweise akzeptiert werden, dass eine Ecke eines D-förmigen Roboters (vgl. **Fig. 3**) während einer Drehung über die Markierung streicht.

[0118] Ein Vorteil der Nutzung von Kartendaten für die Steuerung des Roboters insbesondere in einem Konturfolgemodus ist die Verfügbarkeit von virtuellen Hindernissen, die Bereiche in der Karte markieren, die der Roboter nicht selbstständig befahren und/oder überqueren darf. Diese können beispielsweise von einem Nutzer über das HMI **200** eingegeben werden oder durch den Roboter selbstständig erstellt („erlernt“) werden. Der Roboter kann sich auf diese Weise Bereiche merken, die er nicht erneut befahren will, beispielsweise weil hier ein sicherer Betrieb nicht gewährleistet ist. Ein Nutzer kann hierdurch beispielsweise einen Bereich vorübergehend oder dauerhaft für den Roboter sperren, ohne physische Markierungen in die Umgebung einbringen zu müssen. Dies ist deutlich flexibler und weniger störend als reale Markierungen.

[0119] Für derartige rein virtuellen Hindernisse gilt das gleiche wie für durch Markierungen erzeugte Hindernisse. Da eine reale Kollision nicht möglich ist, genügt eine vereinfachte Behandlung. Diese kann nur darauf ausgelegt sein, ein Überfahren der virtuellen Grenze des virtuellen Hindernisses zu vermeiden, und insbesondere das Befahren des gesperrten Bereichs zu verhindern. Das Verfahren wird beispielhaft für die virtuellen Hindernisse erläutert.

[0120] **Fig. 10** illustriert exemplarisch die Konturfolgefahrt entlang einer Kontur **W** eines Hindernisses (z.B. einer Wand) und einer hierzu senkrecht stehenden Kontur **V** eines virtuellen Hindernisses, welches in der Karte enthalten ist, jedoch real nicht existiert. Für die Beurteilung der Elementarbewegungen entlang der Kontur **W** der Wand wird die vollständige D-förmige Form des Roboters beachtet. Hingegen wird zur Beurteilung der Elementarbewegungen bezüglich der Kontur **V** des virtuellen Hindernisses nur eine vereinfachte virtuelle Form **101** des Roboters **100** berücksichtigt. In dem dargestellten Beispiel von **Fig. 11** ist die vereinfachte virtuelle Form **101** ein Kreis, dessen Mittelpunkt der zentrale Punkt „x“ ist, und dessen Durchmesser der Breite des Roboters entspricht. Auf diese Weise wird eine reine Drehung im Stand durch virtuelle Hindernisse nicht eingeschränkt, wäh-

rend ein Überqueren der Kontur **V** des virtuellen Hindernisses vermieden wird, indem die normalen Regeln für die Kollisionsvermeidung auf die vereinfachte virtuelle Form des Roboters angewendet werden. Der Radius des Kreises kann demnach so gewählt werden, dass bei einer Drehung des Roboters **100** um dessen (kinematischen) Mittelpunkt „x“ sich zumindest zwei Punkte der Außenkontur des Gehäuses des Roboters **100** auf dem Kreis bewegen. In dem dargestellten Beispielen ist der Radius des Kreises gleich der halben Breite des Gehäuses des Roboters **100**. Ein Teil des Roboters **100** befindet sich also außerhalb des Kreises, der die virtuelle Form **101** repräsentiert.

[0121] In dem in Diagramm (a) der **Fig. 11** dargestellten Beispiel fährt der Roboter **100** entlang der Kontur **W** der Wand, bis er eine Position erreicht, in der der Roboter **100** nur mehr einen Sicherheitsabstand d_s von dem virtuellen Hindernis entfernt ist. Der Sicherheitsabstand d_s kann der gleiche Abstand wie für die anderen Arten von Hindernissen sein (vgl. **Fig. 5**). Alternativ kann er größer oder kleiner gewählt werden. Insbesondere kann der Sicherheitsabstand zu virtuellen Hindernissen vor der vereinfachten Form **101** des Roboters **100** gleich dem Konturfolgeabstand gewählt werden.

[0122] Aufgrund der Tatsache, dass bei der Behandlung von virtuellen Hindernissen nur eine vereinfachte Form **101** des Roboters **100** betrachtet wird, führt die durch die Konturen **W** und **V** gebildete Ecke nicht zu einer Sackgassen-Situation; der Roboter kann sich in der Ecke ohne Einschränkung von der Kontur **W** der Wand weg drehen (z.B. durch eine Abfolge der dritten Elementarbewegung als Drehung im Stand), wodurch es dem Roboter **100** möglich ist, sich parallel zur Kontur **V** des virtuellen Hindernisses auszurichten. Wie in Diagramm (b) der **Fig. 11** dargestellt ragt bei der Drehung (dritte Elementarbewegung) eine Ecke **A** des Roboters in das virtuelle Hindernis hinein. Anders als im Beispiel aus **Fig. 5** ist ein Verlassen des aktuellen Konturfolgemodus, um der Kontur **W** in die entgegengesetzte Richtung zu folgen, hier nicht notwendig.

[0123] In dem Beispiel in Diagramm (c) der **Fig. 11** ist der Roboter vollständig parallel zur Kontur **V** des virtuellen Hindernisses ausgerichtet, wobei die Kontur **V** einen Abstand d_s zum Roboter aufweist. Wenn der Sicherheitsabstand d_s gleich dem Konturfolgeabstand d ist, so kann der Roboter nun der Kontur **V** des virtuellen Hindernis weiter folgen. Falls der Sicherheitsabstand d_s kleiner oder größer als der Konturfolgeabstand d ist, so würde die Steuereinheit **150** im Konturfolgemodus den Roboter so steuern, dass der Abstand zwischen virtueller Kontur **V** und Roboter **100** (oder der vereinfachten Roboterform **101**) dem Konturfolgeabstand entspricht.

[0124] Es sei angemerkt, dass der Konturfolgeabstand und der Sicherheitsabstand insbesondere bei Robotern zur Bearbeitung einer Bodenfläche zur Vermeidung von unbeabsichtigten Kollisionen dient. Da solche Kollisionen mit einem virtuellen Hindernis nicht möglich sind, kann der Konturfolgeabstand und/oder der Sicherheitsabstand ebenfalls abhängig von der Art eines Hindernisses gesetzt werden. Speziell für virtuelle Hindernisse kann der Konturfolgeabstand und/oder der Sicherheitsabstand kleiner als für andere Hindernisse oder ganz auf Null gesetzt werden. Insbesondere ein Konturfolgeabstand und/oder der Sicherheitsabstand von Null für virtuelle Hindernisse kann einigen Aufwand für Berechnungen und Bewertungen ersparen.

[0125] Die größtmögliche Vereinfachung der virtuellen Form **101** des Roboters **100** besteht darin, den Roboter durch einen einzigen Punkt zu repräsentieren. Dieser Punkt ist vorzugsweise der zentrale Punkt „x“ (kinematischer Mittelpunkt, Drehmittenpunkt). Beispielsweise kann der Roboter in einem Konturfolgemodus so gesteuert werden, dass der zu einem Punkt vereinfachte (punktförmige) Roboter sich möglichst genau auf der Kontur **V** des virtuellen Hindernisses bewegt. Insbesondere können dabei der Konturfolgeabstand d und der Sicherheitsabstand d_s auf Null gesetzt (d.h. nicht beachtet) werden. Dies bedeutet, dass die Grenze des Hindernisses den Weg des Roboters (bzw. des kinematischen Mittelpunkts) beschreibt, der gerade noch erlaubt ist, so dass der für den Roboter gesperrte Bereich nicht befahren wird. Dies kann bei der Definition des gesperrten Bereichs berücksichtigt werden. Beispielsweise kann der Nutzer über das HMI **200** den Bereich, der nicht befahren werden soll eingeben. Basierend auf dieser Eingabe können dann die virtuellen Grenzen des virtuellen Hindernisses so bestimmt werden, dass der Roboter mit seinem zentralen Punkt „x“ dieser Grenze in einem Konturfolgemodus sicher folgen kann.

[0126] Im Rahmen einer z.B. kostenbasierten Bewertung von Elementarbewegungen für die vereinfachte virtuelle Form **101** des Roboters **100**, kann beispielsweise jede Bewegung weg von der virtuellen Kontur **V** mit Kosten belegt werden. Eine Bewegung in dem gesperrten Bereich kann hierbei verboten sein oder mit höheren Kosten belegt werden als eine Bewegung in dem frei befahrbaren Bereich. Dies ist im Wesentlichen analog zu einer Bewertung basierend auf dem Konturfolgeabstand.

[0127] Für Roboter mit sehr lang gestreckter Form (vgl. Diagramm (e) der Fig. 3) kann eine Vereinfachung zu einer Kreis-Form oder auf einen Punkt dazu führen, dass Teile des Roboters sehr weit in das virtuelle Hindernis (d.h. den gesperrten Bereich) hineinragen. In diesem Fall kann auch eine andere virtuelle Form verwendet werden, um die Navigation und die Bewertung der Bewegungen zu vereinfachen.

Die virtuelle Form ist hierfür möglichst einfach zu bestimmen, indem sie aus Kreisbögen, Geradenstücken oder einfachen Polygonstücken (z. B. Parabel) zusammengesetzt wird. Insbesondere können Kreisbögen gewählt werden, deren Mittelpunkt der zentrale Punkt „x“ (kinematischer Mittelpunkt) ist.

[0128] Beispielsweise kann die virtuelle Form **101** als konvexe Form gewählt werden (d.h. zwei beliebige Punkte können durch eine Strecke innerhalb der Form verbunden werden). Beispielsweise kann die virtuelle Form **101** so gewählt werden, dass sie vollständig in der realen Form enthalten ist. Die Bereiche des Roboters die außerhalb der virtuellen Form liegen, können somit die virtuelle Kontur des virtuellen Hindernisses zumindest temporär z.B. während einer Rotation überschreiten. Es werden also Bewegungen insbesondere Rotationen gegenüber virtuellen Hindernissen ermöglicht, die bei realen Hindernissen zu einer Kollision führen würden.

[0129] Beispielsweise kann die virtuelle Form **101** so gewählt werden, dass ein maximaler Abstand zwischen den Punkten der realen Form des Roboters **100** und der virtuellen Form **100** des Roboters nicht überschritten wird. Analog zu der Vereinfachung zu einem Punkt kann die virtuelle Form eine Linie sein, wobei ein Punkt der Linie der zentrale Punkt „x“ (kinematischer Mittelpunkt) ist. Beispielsweise ist dies einer der Endpunkte der Linie. Der zweite Endpunkt könnte dann die lang gestreckte Form eines Roboters berücksichtigen (vgl. Diagramm (e) der Fig. 3).

[0130] Es sei angemerkt, dass sich das hier beschriebene Konzept der vereinfachten virtuellen Form **101** eines Roboters **100** auf die volle dreidimensionale Form des Roboters verallgemeinern lässt. Beispielsweise ist eine vereinfachte dreidimensionale Form ein Kegel oder ein anderer Rotationskörper. Insbesondere kann durch geeignete Projektion in die Ebene, das dreidimensionale Problem auf den hier beschriebenen zweidimensionalen Fall zurückgeführt werden.

[0131] Vereinfachte Pfadplanung für nicht-kreisförmige Roboter: Fig. 11 zeigt zwei äquivalente Darstellungen zur Pfadplanung eines Roboters mit im Wesentlichen kreisförmiger Grundfläche von einem Startpunkt zu einem Zielpunkt. In der in Diagramm (a) links in Fig. 12 dargestellten Situation soll ein kollisionsfreier Pfad für den Roboter **100** durch eine Anzahl kleinerer Hindernisse **H** (z. B. Stuhlbeine) ermittelt werden. Diese Situation ist äquivalent mit der in Diagramm (b) rechts in Fig. 12 dargestellten Situation, in der ein Pfad für einen punktförmigen Roboter **100'** durch eine Anzahl Hindernisse **H** zu bestimmt wird, wobei die Hindernisse **H** (im Vergleich zu der in Diagramm (a) dargestellten Situation) um den Radius des Roboters **100** vergrößert wurden. Die durch Diagramm (b) veranschaulichte Problemstellung ist intuitiv.

tiv lösbar, da jeder nicht von einem Hindernis besetzter Punkt eine mögliche Position des Roboters ist.

[0132] Der in **Fig. 12** illustrierte Ansatz ist prinzipiell auch für allgemeine nicht-kreisförmige Formen eines Roboters **100** möglich. Jedoch ist die Regel zur Vergrößerung eines Hindernisses abhängig von der Orientierung des Roboters. Gleichzeitig muss die Zwangsbedingung (constraint) beachtet werden, dass die Bewegung nur parallel zur Orientierung des Roboters erfolgen kann. Hierdurch wird die mathematische Formulierung sehr komplex und aufwändig zu berechnen. Der Aufwand steigt weiter, wenn zusätzlich die dreidimensionale Form des Roboters und der Umgebung wie bei dem Beispiel aus **Fig. 1**, Diagramm **(b)**, beachtet werden muss. Folglich wird eine einfachere Methode benötigt.

[0133] Das Problem der Pfadplanung für Roboter mit komplexer Form kann beispielsweise dadurch vereinfacht werden, dass für eine großräumige Pfadplanung insbesondere in weitgehend freien Bereichen bekannte Methoden der Pfadplanung für eine vereinfachte „virtuelle“ Form des Roboters genutzt werden. An Engstellen wird eine lokale Planung unter Beachtung der exakten Roboterform genutzt. Beispielsweise können hierzu die hier beschriebenen Verfahren für die Durchführung einer Konturfolgefahrt genutzt werden, um z.B. einen Weg durch den in Diagramm **(a)** der **Fig. 12** dargestellten Bereich mit komplexer Umgebung zu ermitteln.

[0134] Eine exemplarische Situation, in der die Kombination der Pfadplanung mit einer vereinfachten virtuellen Form des Roboters und einer lokalen Berücksichtigung der vollständigen Form des Roboters genutzt werden kann, ist in **Fig. 1** dargestellt. In beiden Darstellungen (siehe Diagramme **(a)** und **(b)** in **Fig. 1**) steht der Roboter mit seiner bevorzugten Fahrtrichtung unmittelbar vor einer Wand. Hierdurch ist sowohl eine Bewegung in die bevorzugte Fahrtrichtung (Vorwärtsrichtung) als auch eine Drehung im Stand nicht möglich. Bei der Pfadplanung mit einer vereinfachten virtuellen Form des Roboters würde dieses Problem ignoriert, wodurch sich die nötigen Algorithmen und Berechnungen erheblich vereinfachen. Bei dem Versuch den so geplanten Pfad abzufahren, würde der Roboter detektiert, dass eine Ausrichtung auf den geplanten Pfad auf Grund eines Hindernisses (d.h. die Wand) nicht möglich ist, weshalb beispielsweise der Konturfolgemodus gestartet wird.

[0135] Wie weiter oben in Bezug auf die Behandlung von Sackgassen-Situationen beschrieben, würde dies dazu führen, dass sich der Roboter ein Stück entgegen die bevorzugte Fahrtrichtung (d.h. rückwärts) bewegt, bis er sich frei drehen kann. Eine übergeordnete Steuerungs-Instanz könnte nun beispielsweise feststellen, dass der Roboter sich nun entlang des geplanten Pfades ausrichten und diesem folgen

kann, weshalb der Konturfolgemodus beendet wird. Ein spezielles vordefiniertes Fahrmanöver, mit dem sich der Roboter rückwärtsfahrend von der Wand lösen kann, ist nicht notwendig. Ebenfalls wird eine komplexe, die vollständige Kontur des Roboters berücksichtigende Pfadplanung für dieses kleine Manöver nicht benötigt. Die hier beschriebenen Ansätze zur Pfadplanung sind somit flexibel, robust und ressourcenschonend.

[0136] Die vereinfachte virtuelle Form des Roboters entspricht hierbei insbesondere einer runden Form deren Mittelpunkt im zentralen Punkt „x“ (kinematischer Mittelpunkt) liegt. Hierdurch können die z.B. in **Fig. 12** skizzierten Ansätze zur Pfadplanung genutzt werden. Das Ergebnis der Pfadplanung ist ein Pfad **P**, der in entsprechende Steuerkommandos für die Antriebseinheit **170** des Roboters **100** umgesetzt werden kann. Die Steuerung des Roboters wird hierbei fortlaufend basierend auf den mit der Sensoreinheit **120** erfassten Informationen über die Umgebung des Roboters korrigiert. Beispielsweise kann hierbei eine gewünschte Genauigkeit vorgegeben werden, mit der der Roboter dem Pfad **P** folgt. Methoden zur Steuerung eines Roboters **100** entlang eines Pfades sind an sich bekannt. Beispielsweise kann eine Form der Konturfolgefahrt genutzt werden, um einem Pfad zu folgen, analog wie dies weiter oben für das Folgen einer Kontur **V** eines virtuellen Hindernisses mit einer zu einem virtuellen Punkt vereinfachten Form eines Roboters beschrieben wurde.

[0137] Die Pfadplanung wird in der Regel auf Kartendaten beruhen, die das Einsatzgebiet mehr oder weniger vollständig beschreiben. In diesen globalen Karten ist häufig die Genauigkeit der erfassten Details reduziert, um den Speicherbedarf und/oder die Komplexität von Berechnungen zu begrenzen. Beispiele für solche Karten sind:

- Feature-Karten, welche die Konturen von Hindernissen in Form von Punkten, Linien und/oder Flächen darstellen können,
- Rasterkarten (auch Grid-Karte genannt), in denen die Fläche des Einsatzgebiets in einzelne Zellen unterteilt ist und für jede Zelle markiert sein kann, ob sie durch ein Hindernis besetzt oder frei befahrbar ist,
- topologische Karten, die Informationen enthalten, welche charakteristischen Punkte und/oder Bereiche des Einsatzgebiets in durch den Roboter befahrbar in Verbindung stehen.

[0138] Für diese Karten sind Methoden zur Pfadplanung an sich bekannt und können beliebig kombiniert werden.

[0139] Für die Steuerung des Roboters entlang des Pfades kann der Roboter über eine zweite Karte bzw. zweite Form vom Kartendaten verfügen, die mehr De-

tails und aktuelle mit den Sensoren der Sensoreinheit **120** erfasste Informationen der Umgebung enthalten. Insbesondere können in dieser zweiten Karte aktuelle Informationen der Umgebung mit hoher Genauigkeit eingetragen werden, während der Roboter entlang des Pfades **P** steuert. Die in die zweite Karte eingetragenen Informationen können nach einiger Zeit wieder gelöscht werden, um den Speicherbedarf und den Verarbeitungsaufwand zu reduzieren. Alternativ kann nach einiger Zeit der Informationsgehalt der zweiten Karte durch Interpretation und/oder Vereinfachung reduziert werden, wodurch ebenfalls der Speicherbedarf und der Verarbeitungsaufwand reduziert wird.

[0140] Basierend auf der zweiten Karte und auf den mit den Sensoren der Sensoreinheit **120** erfasste Informationen der Umgebung kann in manchen Situationen z.B. festgestellt werden, dass ein weiteres Folgen des geplanten Pfades zu einer Kollision mit zumindest einem Hindernis **H** führen könnte. Hierfür wird insbesondere die vollständige Form des Roboters berücksichtigt. Das Feststellen, dass der geplante Pfad aufgrund eines Hindernisses nicht kollisionsfrei befahren werden kann, kann auch durch das Detektieren einer tatsächlichen Kollision erfolgen.

[0141] Ein Grund für diese möglicherweise bevorstehende Kollision kann insbesondere darin bestehen, dass die vereinfachte virtuelle Form **101** des Roboters **100** nicht mit dem Hindernis kollidieren würde und bei der Planung nur diese Vereinfachte Form berücksichtigt wurde. Weitere Gründe, die insbesondere auch bei runden Roboter zu berücksichtigen sein können, können sein: falsche oder ungenaue Kartendaten, limitierte Genauigkeit in der Planung großräumiger Bewegungen des Roboters, die Positionsänderung eines Hindernisses (z.B. durch den Gebrauch eines Stuhls) und/oder ein neues Hindernis.

[0142] Nachdem solch eine drohende Kollision erkannt wurde, kann der Roboter hierauf reagieren, um so die Kollision abzuwenden. Hierbei kann z.B. die vollständige Kontur der Form des Roboters berücksichtigt werden. Beispielsweise kann die Steuereinheit **150** in einem Konturfolgemodus so steuern, dass er der Kontur des Hindernisses folgt, bis der Roboter wieder auf den ursprünglich geplanten Pfad trifft, einen Zielpunkt erreichbar oder eine Abbruchbedingung erfüllt ist.

[0143] Beispielsweise kann vor Beginn des Konturfolgemodus ein zusätzlicher Zielpunkt gesetzt werden. Dieser kann Teil des ursprünglich geplanten Pfades sein, so dass von diesem Punkt aus der Roboter diesem weiter folgen kann. Der zusätzliche Zielpunkt wird (soweit möglich) hinter dem auszuweichenden Hindernis gesetzt. Der Zielpunkt ist beispielsweise dann erreichbar, wenn zwischen Roboter und Ziel-

punkt kein Hindernis liegt und der Roboter sich kollisionsfrei auf den Zielpunkt zu drehen kann.

[0144] Eine Abbruchbedingung ist beispielsweise, dass der Zielpunkt nicht erreichbar ist, da er in dem Hindernis liegt. Eine weitere Abbruchbedingung kann sein, dass der Abstand des Roboters zu dem Zielpunkt größer als ein vorgegebbarer Wert wird, und/oder dass der Abstand des Roboters zu dem ursprünglichen Pfad größer als ein vorgegebbarer Wert wird. Die Kontur des Hindernisses würde also den Roboter ungewöhnlich weit von seinem ursprünglichen Kurs wegführen. Der vorgebbare Wert des maximalen Abstands ist beispielsweise die Breite des Roboters oder die doppelte Breite des Roboters. Eine weitere Abbruchbedingung ist beispielsweise, die benötigte Zeit und/oder der zurückgelegte Weg während der Konturfolgefahrt.

[0145] Wenn eine Abbruchbedingung erfüllt ist, so stoppt der Roboter und prüft, ob es einen anderen Pfad von seiner aktuellen Position zu dem Zielpunkt gibt. Hierzu wird in die Kartendaten die Information aufgenommen, dass der zuvor geplante Pfad nicht erfolgreich befahren werden konnte, und an welcher Position bzw. in welchem Bereich es zu der Unterbrechung und dem Abbruch der Bewegung entlang des Pfades **P** kam. Insbesondere wird so für zukünftige Pfadplanungen mit der vereinfachten virtuellen Form des Roboters die Information gespeichert, wenn ein Pfad für die vereinfachte Form **101** aber nicht für die vollständige Form des Roboters passierbar ist.

[0146] Die Pfadplanung eines Pfades **P** kann „pessimistisch“ erfolgen, wobei der Roboter bei einer Planung basierend auf einer idealen Karte (ohne Fehler oder beschränkter Genauigkeit) immer das Ziel erreicht. Dies geschieht beispielsweise durch die Wahl der vereinfachten virtuellen Form des Roboters als Umkreis des Roboters. Dies bedeutet, dass alle Punkte des Roboters vollständig in dem Kreis liegen und der Mittelpunkt dem zentralen Punkt entspricht (vgl. auch **Fig. 13**, Umkreis **102**). Somit ist eine Drehung des Roboters im Stand an jedem Punkt des Pfades **P** möglich. Hierbei kann die virtuelle Form **101** des Roboters breiter als der tatsächliche Roboter **100** sein, wodurch enge Passagen zwischen zwei Hindernissen nicht befahren werden.

[0147] Alternativ oder zusätzlich kann die Planung eines Pfades **P** „optimistisch“ erfolgen. Hierbei wird beispielsweise für die vereinfachte virtuelle Kontur eine Kreisform angenommen, deren Durchmesser der Breite des Roboters entspricht. Hierdurch ist sichergestellt, dass der Roboter zumindest zwischen zwei Hindernissen hindurch passt, wenn der Pfad durch deren Mitte führt. Es sei angemerkt, dies gilt nur für ideale Kartendaten. In der Praxis kann es trotzdem passieren, dass bei Ankunft vor den zwei Hindernissen festgestellt wird, dass nicht genügend Platz vor-

handen ist, um dem Pfad zwischen den beiden Hindernissen hindurch zu folgen. Zusätzlich ist es bei einer komplexen Form des Roboters möglich, dass die nötigen Drehungen nicht ausgeführt werden können, um dem geplanten Pfad P zu folgen.

[0148] Der Nachteil des pessimistischen Ansatzes ist, dass in einigen Umgebungen bzw. hierzu gehörigen Karten kein Pfad von einem Startpunkt zu einem Zielpunkt gefunden wird, obwohl dies praktisch möglich wäre. Der Nachteil des optimistischen Ansatzes ist, dass hierbei Pfade gefunden werden, die in der Praxis durch den Roboter nicht oder nur schwierig befahren werden können. Durch die konkrete Wahl der vereinfachten virtuellen Kontur können beliebige Abstufungen zwischen dem optimistischen und dem pessimistischen Ansatz gewählt werden.

[0149] Die Pfadplanung kann hierbei durch eine geeignete Kombination der erwähnten Ansätze (optimistisch, pessimistisch) erfolgen. Beispielsweise kann zuerst eine pessimistische Planung erfolgen. Wenn diese nicht erfolgreich ist, wird eine optimistische Planung durchgeführt, um zu prüfen, ob es überhaupt einen möglichen Pfad gibt. Beispielsweise können eine pessimistische und eine optimistische Planung durchgeführt werden, um die Ergebnisse miteinander zu vergleichen. Die geplanten Pfade können z.B. nach vorgebbaren Kriterien bewertet werden und der Pfad mit der besten Bewertung (z.B. den geringsten „Kosten“) ausgewählt werden. Die vorgebbaren Kriterien können z.B. die Länge des Pfades und/oder dessen Abstand zu Hindernissen berücksichtigen. Wenn beispielsweise die pessimistische Planung zu einem Pfad führt, der nur „unwesentlich länger“ als der optimistisch geplante Pfad ist, so kann der pessimistische Pfad gewählt werden. „Unwesentlich länger“ kann hierbei ein fester möglicher Umweg von beispielsweise 0,1m - 10m und/oder ein fester Faktor von beispielsweise 1,5-3 Mal so lang bedeuten. Gegebenenfalls können in dem Vergleich weitere Planungen mit einer anderen Variante der virtuellen Form berücksichtigt werden.

[0150] Alternativ können der pessimistische (z.B. erste virtuelle Roboterform, die den Roboter vollständig einschließt) und der optimistische Ansatz (z.B. zweite virtuelle Roboterform, die den Roboter nicht vollständig einschließt) in einem Planungsansatz kombiniert werden. Ein einfaches Beispiel hierfür ist in **Fig. 13** dargestellt, wobei die dargestellte Situation sehr ähnlich ist wie die Situation in **Fig. 12**, Diagramm (a). Dabei werden unterschiedlichen Teilgebieten einer Karte des Roboters jeweils Kosten (z.B. ein skalarer Kostenwert) zugeordnet, wobei diese Kosten insbesondere die tatsächliche Form des Roboters **100** berücksichtigen und beispielsweise höher angesetzt werden, wenn der Roboter aufgrund eines in der Nähe befindlichen Hindernisses potentiell in seiner Rotation (um seinen kinematischen Mittel-

punkt) eingeschränkt ist. In dem Beispiel aus **Fig. 13**, sind die Kosten an Positionen mit einem Abstand kleiner gleich Δr gleich K_1 , in anderen Teilbereichen K_0 ($K_1 > K_0$), wobei beispielsweise der Wert Δr die Differenz des Radius des „große“ Umkreises **102** des Roboters **100** (virtuelle Form als Worst-Case-Betrachtung) ist und des Radius der vereinfachten Roboterform **101** ist. Die tatsächliche Pfadplanung kann dann wie in **Fig. 12** basierend auf der vereinfachten virtuellen Roboterform **101** (z.B. Kreis mit einem Radius, der der halben Breite des Roboters entspricht, vgl. **Fig. 11**), was die Reduktion auf die Pfadplanung eines Punktes erlaubt (vgl. **Fig. 12**). Beispielsweise können bei einer derartigen kostenbasierten Planung die Kosten eines Weges abhängig von der Entfernung zu Hindernissen erstellt werden (wie erwähnt ist ein Hindernis nahe, ist die Rotation potentiell eingeschränkt und folglich der Kostenwert in dem jeweiligen Teilgebiet der Karte höher). Hierdurch kann ein Pfad zwischen zwei eng stehenden Hindernissen (optimistisch) mit höheren Kosten belegt werden, als der mit einem Umweg verbundene Pfad um die Hindernisse herum (pessimistisch). Durch die Wahl der Kosten für die Bewegung des Roboters nahe von Hindernissen wird der akzeptable Umweg definiert und kann als Ergebnis einer Optimierungsaufgabe angesehen werden. Der Vorteil eines solchen Ansatzes besteht darin, dass er immer ein Ergebnis liefert, wenn der optimistische Ansatz zu einem Pfad führt. Gleichzeitig ist der so erhaltene Pfad immer eine Abwägung zwischen den zwischen Startpunkt und Zielpunkt liegenden engen Stellen und den nötigen Umwegen, um die engen Stellen zu vermeiden. In dem Beispiel gemäß **Fig. 13** können die Kosten K_0 und K_1 diskrete Werte sein (z.B. $K_0=0$, $K_1=1$), alternativ kann K_1 mit kleiner werdendem Abstand zu einem Hindernis H auch ansteigen. Hindernisse können in diesem Ansatz dadurch berücksichtigt werden, dass die Kosten in Bereichen, die durch ein Hindernis besetzt sind praktisch unendlich groß gewählt werden.

[0151] Wenn während der Fahrt des Roboters **100** entlang eines Pfades P eine mögliche Kollision mit einem Hindernis H detektiert wird, so kann der Roboter dem Pfad folgend bis an das Hindernis heranfahren (z. B. mit Berücksichtigung eines Sicherheitsabstandes) und dann direkt in den Konturfolgemodus wechseln. Alternativ kann der Roboter prüfen, ob ein Ausweichpfad an dem Hindernis vorbei existiert, der auf den ursprünglichen Pfad P zurückführt.

[0152] Insbesondere kann die Reaktion auf eine mögliche Kollision mit einem Hindernis H in Abhängigkeit von der aktuellen Aufgabe des Roboters erfolgen. Beispielsweise kann bei einem autonomen mobilen Roboter zur Bearbeitung einer Bodenfläche während einer Bearbeitung einer geplanten Bahn der Roboter möglichst nahe an das Hindernis heranfahren und dann entlang der Kontur des Hindernis die Fläche bearbeiten. Andererseits kann der gleiche Ro-

boter bei einer Fahrt zu einem zugewiesenen zu bearbeitenden Bereich oder zu einer Basisstation, Bereiche durchfahren, die nicht bearbeitet werden sollen. In diesem Fall kann ein Ausweichpfad um das Hindernis **H** bestimmt werden, so dass das Ziel (z. B. zugewiesener Bereich, Basisstation) schneller erreicht wird.

um Roboter mit Radius 0,5-2 m) eingeschränkt werden.

[0153] Zur Bestimmung des Ausweichpfades kann direkt die vollständige Form des Roboters **100** berücksichtigt werden. Alternativ oder zusätzlich kann eine Vorplanung basierend auf der vereinfachten virtuellen Form **101** des Roboters **100** gemacht werden, ob überhaupt ein Pfad um das Hindernis herum möglich ist. Dies ist insbesondere hilfreich, wenn der Pfad zwischen zwei eng stehenden Hindernissen hindurchführen würde. Hierbei kann festgestellt werden, dass die vereinfachte virtuelle Form **101** dem ursprünglich geplanten Pfad nicht folgen könnte, weshalb eine weiträumigere Umfahrung und hiermit verbundene Pfadplanung notwendig ist. Hierbei kann festgestellt werden, dass die vereinfachte virtuelle Form **101** dem ursprünglich geplanten Pfad zwischen den beiden Hindernissen hindurch folgen kann. In diesem Fall kann sich der Roboter beispielsweise mit dem Konturfolgemodus unter Berücksichtigung der vollständigen Form des Roboters zwischen den Hindernissen hindurchbewegen.

[0154] Das Ergebnis der Bestimmung des Ausweichpfades kann sein, dass das Hindernis in einem bestimmten Abstand sicher umfahren werden kann. Dies ist insbesondere der Fall, wenn es sich um ein einzelnes Hindernis in einem sonst weitgehend freien Bereich handelt. Ein weiteres mögliches Ergebnis ist, dass dem Hindernis in einem Konturfolgemodus ausgewichen werden kann (in dem die vollständige Form des Roboters berücksichtigt wird).

[0155] Insbesondere bei Nutzung eines auf Kartendaten basierenden Konturfolgemodus ist eine Variante zur Prüfung, ob der Roboter das Hindernis umfahren und zurück auf den ursprünglichen Pfad **P** finden kann, die Vorausberechnung (oder Simulation) des Verlaufs der Konturfolgefahrt. Dies kann unter anderem auch genutzt werden, wenn es verschiedene Möglichkeiten des Beginns des Konturfolgemodus gibt (insbesondere rechts oder links herum ausweichen), um den am schnellsten zum Ziel führenden Weg zu finden.

[0156] Zur Planung des Ausweichpfades werden Kartendaten verwendet, welche die Umgebung mit einer hohen Genauigkeit beschreiben. Dies geschieht beispielsweise anhand der in der zweiten Karte gesammelten Information über die Umgebung. Um den Aufwand an Ressourcen für Speicherbedarf und Rechenleistung zu beschränken, kann die Planung des Ausweichpfades auf einen kleinen Bereich (z. B. Kreis

ZITATE ENTHALTEN IN DER BESCHREIBUNG

Diese Liste der vom Anmelder aufgeführten Dokumente wurde automatisiert erzeugt und ist ausschließlich zur besseren Information des Lesers aufgenommen. Die Liste ist nicht Bestandteil der deutschen Patent- bzw. Gebrauchsmusteranmeldung. Das DPMA übernimmt keinerlei Haftung für etwaige Fehler oder Auslassungen.

Zitierte Nicht-Patentliteratur

- J. Fasola et al., „Fast Goal Navigation with Obstacle Avoidance using a Dynamic Local Visual Model“, in: Proc. VII. Simpósio Brasileiro de Automação Inteligente, São Luis, Sept. 2005 [0036]

Patentansprüche

1. Ein Verfahren zur Steuerung eines autonomen mobilen Roboters, der in einem ersten und mindestens einem zweiten Konturfolgemodus arbeiten kann, wobei in jedem der Konturfolgemodi der Roboter (100) einen im Wesentlichen konstanten Abstand zu einer Kontur (W, V) einhält, während er sich entlang der Kontur (W, V) bewegt; das Verfahren weist folgendes auf:

Starten des ersten Konturfolgemodus, wobei der Roboter der Kontur (W, V) in einer ersten Fahrtrichtung folgt;

Detektieren einer Sackgassen-Situation, in der ein fortgesetztes Folgen der Kontur (W, V) im ersten Konturfolgemodus nicht kollisionsfrei möglich ist;

Starten eines zweiten Konturfolgemodus, wobei der Roboter der Kontur (W, V) in einer zweiten Fahrtrichtung folgt; und

Festlegen eines Kriteriums, bei dessen Erfüllung der zweite Konturfolgemodus beendet wird, und laufendes Auswerten des Kriteriums während der Roboter im zweiten Konturfolgemodus arbeitet.

2. Das Verfahren gemäß Anspruch 1, wobei die Konturfolgemodi durch mindestens zwei Parameter gekennzeichnet sind, wobei die mindestens zwei Parameter die Fahrtrichtung, den Konturfolgeabstand (d) und optional eines der folgenden Parameter umfassen: die Seite des Roboters, die der Kontur zugewandt ist, ein Sicherheitsabstand, die Roboterform, die zur Erkennung einer drohenden Kollision berücksichtigt wird, die Regeln, gemäß denen die Bewegung entlang der Kontur erfolgt, und wobei sich zwei verschiedene Korrekturfolgemodi durch mindestens einen Parameter unterscheiden.

3. Das Verfahren gemäß Anspruch 1 oder 2, wobei das Detektieren einer Sackgassen-Situation aufweist:

Detektieren, dass eine Bewegung des Roboters (100) entlang der Kontur (W) sowie eine Drehung des Roboters (100) nicht kollisionsfrei möglich sind, wobei bei der Detektion insbesondere in einer Karte des Roboters gespeicherte ortsbezogene Informationen berücksichtigt werden.

4. Das Verfahren gemäß einem der Ansprüche 1 bis 3, wobei ein dritter Konturfolgemodus gestartet wird, sofern im zweiten Konturfolgemodus erneut eine Sackgassen-Situation detektiert wird und wobei ein Kriterium, bei dessen Erfüllung der dritte Konturfolgemodus beendet wird, festgelegt wird und dieses Kriterium laufend ausgewertet wird, während der Roboter im dritten Konturfolgemodus arbeitet.

5. Das Verfahren gemäß Anspruch 4, wobei der dritte Konturfolgemodus sich von dem zweiten Konturfolgemodus durch den folgenden Parameter unter-

scheidet: die Seite des Roboters, die der Kontur (W) zugewandt ist.

6. Das Verfahren gemäß einem der Ansprüche 1 bis 5, wobei der erste Konturfolgemodus fortgesetzt wird, wenn der zweite Konturfolgemodus aufgrund der Erfüllung des festgelegten Kriteriums beendet wird.

7. Das Verfahren gemäß einem der Ansprüche 1 bis 6, bei dem die Kontur (W) durch ein real nicht existierendes jedoch in einer Karte des Roboters enthaltenes virtuelles Hindernis gebildet wird.

8. Das Verfahren gemäß einem der Ansprüche 1 bis 7, wobei das Kriterium, bei dessen Erfüllung der zweite Konturfolgemodus beendet wird, die Möglichkeit, eine bestimmte Bewegung auszuführen, beinhaltet.

9. Das Verfahren gemäß Anspruch 8, wobei die bestimmte Bewegung zumindest eines der folgenden umfasst:

eine Drehung um einen bestimmten Winkel und eine translatorische Bewegung, insbesondere in Vorwärtsrichtung, über eine bestimmte Strecke.

10. Das Verfahren gemäß einem der Ansprüche 1 bis 9, wobei das Auswerten des Kriteriums folgendes umfasst:

automatisches Planen einer kollisionsfreien Roboterbewegung nach vorgebbaren Regeln; Durchführen der geplanten Roboterbewegung; Prüfen, ob die geplante Roboterbewegung kollisionsfrei ausgeführt werden konnte.

11. Das Verfahren gemäß Anspruch 10, wobei die Planung der kollisionsfreien Roboterbewegung in einer Karte des Roboters gespeicherte ortsbezogene Informationen betreffend Hindernisse berücksichtigt.

12. Das Verfahren gemäß Anspruch 10 oder 11, wobei das automatische Planen einer Roboterbewegung nach vorgebbaren Regeln folgendes umfasst:

Planen einer Drehung und einer anschließenden translatorischen Bewegung, derart, dass nach Durchführung der Bewegung ein Punkt eines Hindernisses in einem bestimmten Abstand, insbesondere im Konturfolgeabstand, zum Roboter liegt.

13. Das Verfahren gemäß Anspruch 12, wobei die Drehung um einen Winkel erfolgt, der größer ist als ein vorgegebbarer Mindestwinkel.

14. Das Verfahren gemäß einem der Ansprüche 1 bis 13, wobei das Festlegen des Kriteriums, bei dessen Erfüllung der zweite Konturfolgemodus beendet wird, oder das Auswerten dieses Kriteriums unter Be-

rücksichtigung von in einer Karte des Roboters gespeicherten ortsbezogenen Informationen erfolgt.

15. Das Verfahren gemäß einem der Ansprüche 1 bis 14, wobei des Kriteriums, bei dessen Erfüllung der zweite Konturfolgemodus beendet wird, während der Durchführung des zweiten Konturfolgemodus aktualisiert wird.

16. Ein Verfahren zur Steuerung eines autonomen mobilen Roboters in einem Konturfolgemodus, in dem der Roboter (100) in einem Konturfolgeabstand im Wesentlichen einer Kontur (W, V) folgt; das Verfahren weist in dem Konturfolgemodus auf:
Bewerten von mindestens drei verschiedenen Elementarbewegungen anhand mindestens eines vorgebbaren Kriteriums, und
Ausführen einer der drei Elementarbewegungen basierend auf deren Bewertung,
wobei eine erste der drei Elementarbewegungen eine rein translatorische Bewegung des Roboters (100) ist,
wobei eine zweite der drei Elementarbewegungen eine Drehung des Roboters (100) zu der Kontur (W) hin beinhaltet, und
wobei eine dritte der drei Elementarbewegungen eine Drehung des Roboters (100) von der Kontur (W) weg beinhaltet.

17. Das Verfahren gemäß Anspruch 16 wobei bei gleicher Bewertung von zumindest zwei der Elementarbewegungen, jene ausgewählt wird, welche den Roboter näher zu der Kontur (W) hin oder weniger weit von der Kontur (W) weg führt.

18. Das Verfahren gemäß Anspruch 16 oder 17, wobei bei der Bewertung einer Elementarbewegung die davor ausgeführte Elementarbewegung berücksichtigt wird.

19. Das Verfahren gemäß Anspruch 18, wobei bei der Bewertung berücksichtigt wird, dass nach Ausführung der zweiten Elementarbewegung nicht die dritte Elementarbewegung ausgewählt werden soll und umgekehrt.

20. Das Verfahren gemäß einem der Ansprüche 16 bis 19, wobei die Bewertung der Elementarbewegungen zumindest eines der folgenden Kriterien berücksichtigt:
die Elementarbewegung ist ohne eine Kollision mit einem Hindernis (H) möglich;
die Entfernung des Roboters zu Hindernissen (H) während und/oder nach der Bewegung; und
die kollisionsfreie Ausführbarkeit einer weiteren, beispielsweise translatorischen Bewegung nach Ausführung der jeweiligen Elementarbewegung.

21. Das Verfahren gemäß Anspruch 20, wobei die Hindernisse (H) unterschiedlicher Art sein können

und die Art eines Hindernisses (H) bei der Bewertung berücksichtigt wird.

22. Das Verfahren gemäß Anspruch 21, wobei eine erste Art Hindernisse (H) von einer Sensoreinheit (120) des Roboters detektierte Hindernisse (H) beinhalten und eine zweite Art Hindernisse real nicht existierende jedoch in einer Karte des Roboters enthaltene virtuelle Hindernisse sind.

23. Das Verfahren gemäß einem der Ansprüche 16 bis 22, wobei die zweite und die dritte Elementarbewegung Drehungen im Stand beinhalten.

24. Das Verfahren gemäß einem der Ansprüche 16 bis 23, das weiter aufweist:
Detektieren, dass gemäß einem vorgebbaren Kriterium keine der drei Elementarbewegungen ausführbar ist,
wobei - sofern detektiert wurde, dass keine der drei Elementarbewegungen ausführbar ist - der Roboter die Fahrtrichtung und/oder die der Kontur zugewandte Seite des Roboters und/oder die Bewertungskriterien ändert.

25. Das Verfahren gemäß Anspruch 24, wobei der Roboter (100) eine bevorzugte Fahrtrichtung hat, und bei Änderung der Fahrtrichtung in die der bevorzugten Fahrtrichtung entgegengesetzte Fahrtrichtung ein Bewertungskriterium festgelegt wird, bei dessen Erfüllung wieder in die bevorzugte Fahrtrichtung gewechselt wird.

26. Das Verfahren gemäß einem der Ansprüche 16 bis 25, das weiter aufweist:
wobei die mindestens drei Elementarbewegungen durch mehrere Parameter definiert sind, und die Parameter mit Hilfe eines Optimierungsverfahrens, insbesondere eines maschinellen Lernverfahrens bestimmt werden.

27. Das Verfahren gemäß Anspruch 26, wobei mittels maschineller Lernverfahren die Parameter zumindest teilweise automatisch so bestimmt werden, dass der Roboter in bestimmten vorgebbaren Situationen gewünschten vorgebbaren Bewegungsmuster ausführt.

28. Ein Verfahren zur Steuerung eines autonomen mobilen Roboters (100) mit einer ersten Karte eines Robotereinsatzgebietes, wobei die erste Karte zumindest Daten über die Position von Hindernissen (H) enthält; das Verfahren umfasst:
Planen eines Pfades (P) zu einem Zielpunkt in der ersten Karte unter der Annahme einer vereinfachten virtuellen Form (101) des Roboters (100);
Bewegen des Roboters entlang des geplanten Pfades (P);
Erfassen von Hindernissen (H) in der Umgebung des Roboters (100) mittels einer Sensoreinheit (120) des

Roboters, während der Bewegung entlang des geplanten Pfades (P),

Feststellen, dass der geplante Pfad (P) aufgrund eines Hindernisses (H) nicht kollisionsfrei befahren werden kann, unter Beachtung der tatsächlichen Roboterform

Fortsetzen der Bewegung des Roboters (100) unter Beachtung der tatsächlichen Roboterform.

29. Verfahren gemäß Anspruch 28, wobei die vereinfachte virtuelle Form (101) des Roboters (100) durch einen Kreis repräsentiert wird, um dessen Mittelpunkt sich der Roboter (100) drehen kann, ohne dass sich der Mittelpunkt bewegt.

30. Das Verfahren gemäß Anspruch 29, wobei der Radius des Kreises so gewählt ist, dass bei einer Drehung des Roboters (100) um dessen Mittelpunkt sich zumindest zwei Punkte der Außenkontur des Roboters (100) auf dem Kreis bewegen, beispielsweise dass der Radius des Kreises der halben Breite des Roboters entspricht.

31. Das Verfahren gemäß Anspruch 30, wobei zumindest ein Teil des Roboters sich außerhalb des Kreises befindet.

32. Das Verfahren gemäß einem der Ansprüche 28 bis 31, wobei während der Roboter (100) sich entlang des geplanten Pfades bewegt, die erfassten Hindernisse (H) in einer zweiten Karte eingetragen werden, wobei die Genauigkeit der Position und/oder Ausdehnung von Hindernissen in der zweiten Karte größer als in der ersten Karte ist..

33. Das Verfahren gemäß Anspruch 32, wobei das Feststellen, dass der geplante Pfad (P) aufgrund eines Hindernisses nicht kollisionsfrei befahren werden kann, sowie das Fortsetzen der Bewegung des Roboters (100) auf der zweiten Karte basiert.

34. Das Verfahren gemäß einem der Ansprüche 28 bis 33, bei dem das Planen eines Pfades (P) zu einem Zielpunkt in der ersten Karte unter der Annahme einer vereinfachten virtuellen Form (101) des Roboters (100) folgendes umfasst:
Planen eines kollisionsfreien Pfades (P) zu dem Zielpunkt unter der Annahme einer ersten virtuellen Form des Roboters (100), die den Roboter vollständig beinhaltet.

35. Das Verfahren gemäß Anspruch 34, bei dem das Planen eines Pfades (P) zu einem Zielpunkt in der ersten Karte unter der Annahme einer vereinfachten virtuellen Form (101) des Roboters (100) weiter umfasst:
Erneutes Planen eines Pfades (P) zu dem Zielpunkt unter der Annahme einer zweiten virtuellen Form des Roboters (100), falls bei der Pfadplanung unter der Annahme einer ersten virtuellen Form kein kollisions-

freier Pfad gefunden wird, wobei die zweite virtuelle Form des Roboters diesen nicht vollständig beinhaltet.

36. Das Verfahren gemäß Anspruch 34, wobei die erste virtuelle Form des Roboters (100) eine Kreisform ist.

37. Das Verfahren gemäß Anspruch 35, wobei die erste und die zweite virtuelle Form des Roboters (100) eine Kreisform haben und/oder die erste virtuelle Form dem kleinsten Kreis entspricht, der den Roboter gerade noch einschließt und um dessen Mittelpunkt sich der Roboter (im Stand) drehen kann.

38. Das Verfahren gemäß einem der Ansprüche 28 bis 33, bei dem das Planen eines Pfades (P) zu einem Zielpunkt in der ersten Karte unter der Annahme einer vereinfachten virtuellen Form (101) des Roboters (100) folgendes umfasst:

Planen eines ersten Pfades zu dem Zielpunkt unter der Annahme einer ersten virtuellen Form des Roboters (100),

Planen mindestens eines zweiten Pfades zu dem Zielpunkt unter der Annahme mindestens einer zweiten, von der ersten verschiedenen virtuellen Form des Roboters (100),

Bewerten des ersten Pfades und des mindestens einen zweiten Pfades nach einem vorgebbaren Kriterium,

Auswählen eines Pfades (P) für eine nachfolgende Roboterbewegung aus dem ersten und dem mindestens einen zweiten Pfad basierend auf der Bewertung.

39. Das Verfahren gemäß einem der Ansprüche 28 bis 38, bei dem das Planen eines Pfades (P) zu einem Zielpunkt in der ersten Karte unter der Annahme einer vereinfachten virtuellen Form (101) des Roboters (100) folgendes umfasst:

Zuordnen eines Kostenfunktionswertes zur Befahrung von Bereichen des Robotereinsatzgebiets; und Bestimmen des Pfades geringster Kosten für die virtuell vereinfachte Form (101) des Roboters (100), wobei in die Kostenfunktionswerte der Abstand des Pfades zu Hindernissen einfließen,

40. Das Verfahren gemäß Anspruch 39, wobei insbesondere die Kosten abhängig von der Einschränkung des Rotationsfreiheitsgrades des Roboters durch ein Hindernis sind.

41. Das Verfahren gemäß Anspruch 39 oder 40, wobei die Kostenfunktionswerte basierend auf in der ersten Karte enthaltenen Informationen berechnet werden und wobei insbesondere die virtuelle Form eine Kreisform ist, die den Roboter nicht vollständig umfasst.

42. Das Verfahren gemäß einem der Ansprüche 28 bis 41, wobei das Fortsetzen der Bewegung des Roboters (100) unter Beachtung seiner tatsächlichen Roboterform aufweist:

Festlegen eines Zwischenzielpunktes auf dem geplanten Pfad (P);

Folgen der Kontur des Hindernisses bis entweder der Weg zwischen dem Roboter und dem Zwischenzielpunkt frei ist oder eine Abbruchbedingung erfüllt ist.

43. Das Verfahren gemäß Anspruch 42, wobei die Abbruchbedingung den Abstand des Roboters von dem geplanten Pfad (P) und/oder dem Zwischenzielpunkt berücksichtigt, und nach Erfüllung der Abbruchbedingung der Roboter (100) ausgehend von seiner aktuellen Position hin zu dem Zielpunkt einen neuen Pfad plant.

44. Das Verfahren gemäß Anspruch 43, nach Erfüllung der Abbruchbedingung der Roboter (100) einen Zähler erhöht, der die Anzahl der Fehlversuche zählt.

45. Ein Verfahren zur Steuerung eines autonomen mobilen Roboters (100) mit Hilfe einer Karte des Robotereinsatzgebietes, wobei die Karte zumindest Informationen über die Position realer, mittels einer Sensoreinheit (120) erkannter Hindernisse sowie Informationen über virtuelle Hindernisse enthält; das Verfahren umfasst:

Steuern des Roboters (100) in der Nähe eines realen Hindernisses derart, dass eine Kollision mit dem realen Hindernis vermieden wird, wobei die tatsächliche Form des Roboters (100) berücksichtigt wird, und Steuern des Roboters in der Nähe eines virtuellen Hindernisses derart, dass eine Kollision mit dem virtuellen Hindernis vermieden wird, wobei eine vereinfachte, virtuelle Form (101) des Roboters (100) berücksichtigt wird.

46. Das Verfahren gemäß Anspruch 45, wobei Teile der tatsächlichen Form des Roboters (100) außerhalb der vereinfachten, virtuellen Form (101) des Roboters (100) liegen.

47. Das Verfahren gemäß Anspruch 45 oder 46, wobei die vereinfachte, virtuelle Form (101) des Roboters (100) ein Kreis ist, um dessen Mittelpunkt sich der Roboter drehen kann, ohne dass sich der Mittelpunkt bewegt.

48. Das Verfahren gemäß Anspruch 47, wobei der Radius des Kreises so gewählt ist, dass bei einer Drehung des Roboters (100) um dessen Mittelpunkt sich zumindest zwei Punkte der Außenkontur des Roboters (100) auf dem Kreis bewegen, beispielsweise dass der Radius des Kreises der halben Breite des Roboters entspricht.

49. Das Verfahren gemäß einem der Ansprüche 45 bis 48, wobei die vereinfachte, virtuelle Form (101) des Roboters (100) dergestalt ist, dass der Roboter in jeder möglichen kollisionsfreien Position relativ zu einem Hindernis sich kollisionsfrei drehen kann.

50. Ein Verfahren zur Steuerung eines autonomen mobilen Roboters in einem Konturfolgemodus, in dem der Roboter (100) in einem Konturfolgeabstand im Wesentlichen einer Kontur (W, V) folgt, wobei eine Karte des Roboters zumindest Informationen über die Position realer, mittels einer Sensoreinheit (120) erkannter Hindernisse sowie Informationen über virtuelle Hindernisse enthält und der Roboter fortlaufend seine Position in dieser Karte bestimmt; wobei in dem Konturfolgemodus sich der Roboter (100) entlang einer Kontur (W, V) bewegt; wobei die Kontur (W, V) durch den Verlauf eines realen Hindernisses und den Verlauf einer virtuellen Grenze eines virtuellen Hindernisses gegeben ist.

51. Das Verfahren gemäß Anspruch 50, wobei der Abstand zwischen Roboter und Kontur (W, V) basierend auf den in der Karte gespeicherten Informationen ermittelt wird.

52. Der Verfahren gemäß Anspruch 50 oder 51, wobei der Konturfolgeabstand davon abhängt, ob die Kontur, der gerade gefolgt wird, ein virtuelles Hindernis repräsentiert oder nicht.

53. Eine Steuereinheit für einen autonomen mobilen Roboter, die dazu ausgebildet ist, ein Verfahren gemäß einem der Ansprüche 1 bis 52 auszuführen.

Es folgen 8 Seiten Zeichnungen

Anhängende Zeichnungen

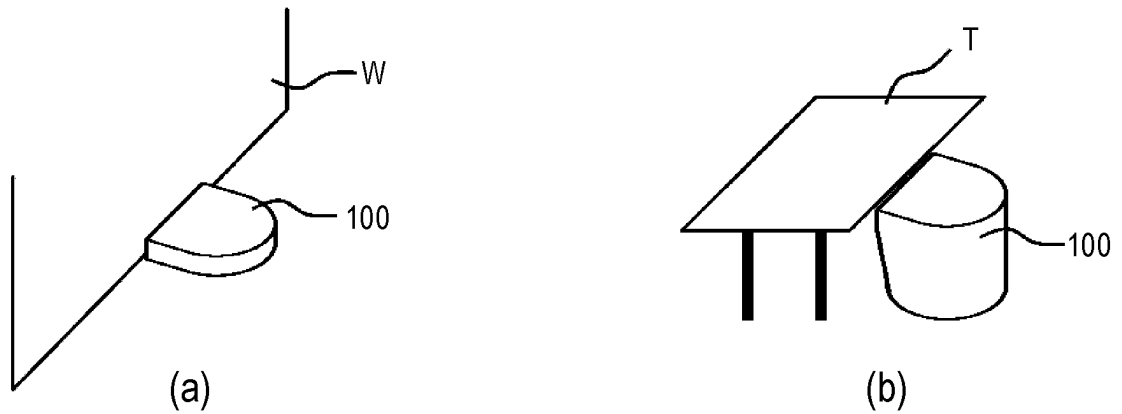


Fig. 1

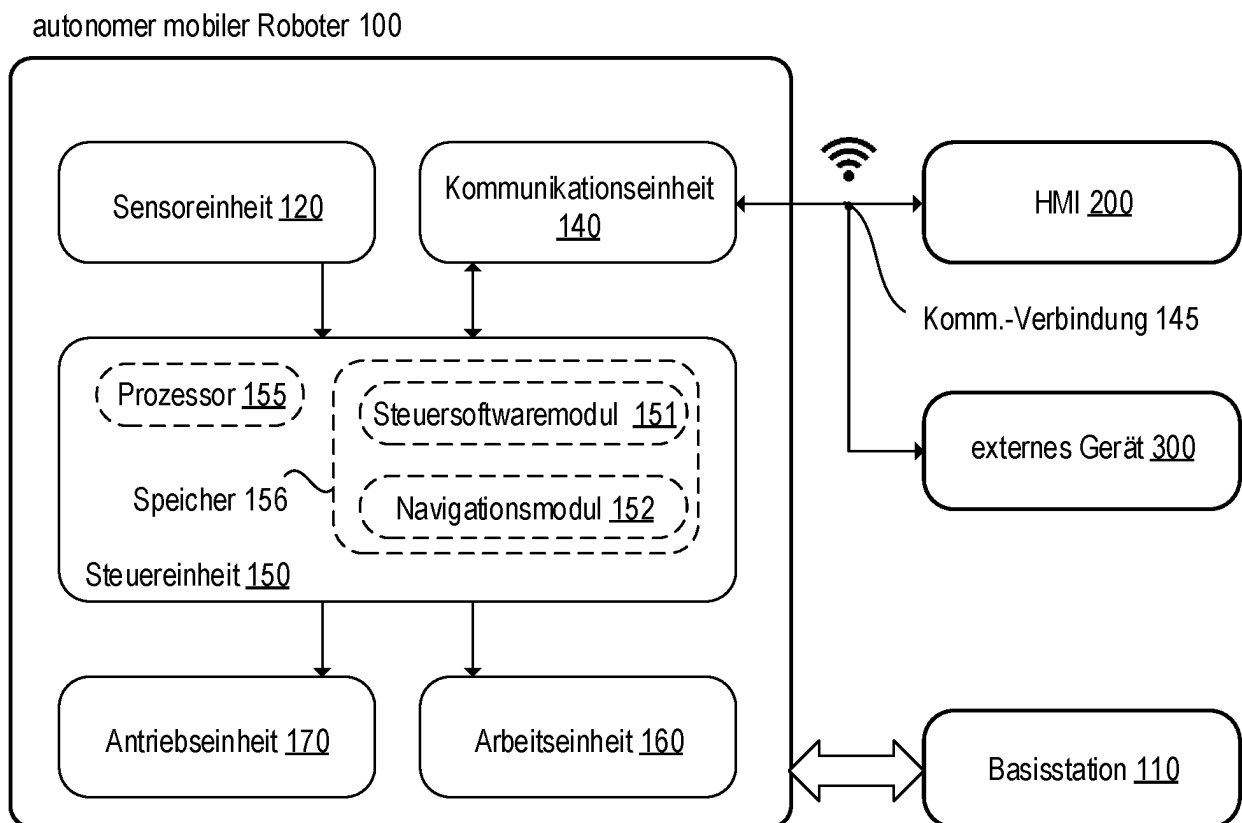


Fig. 2

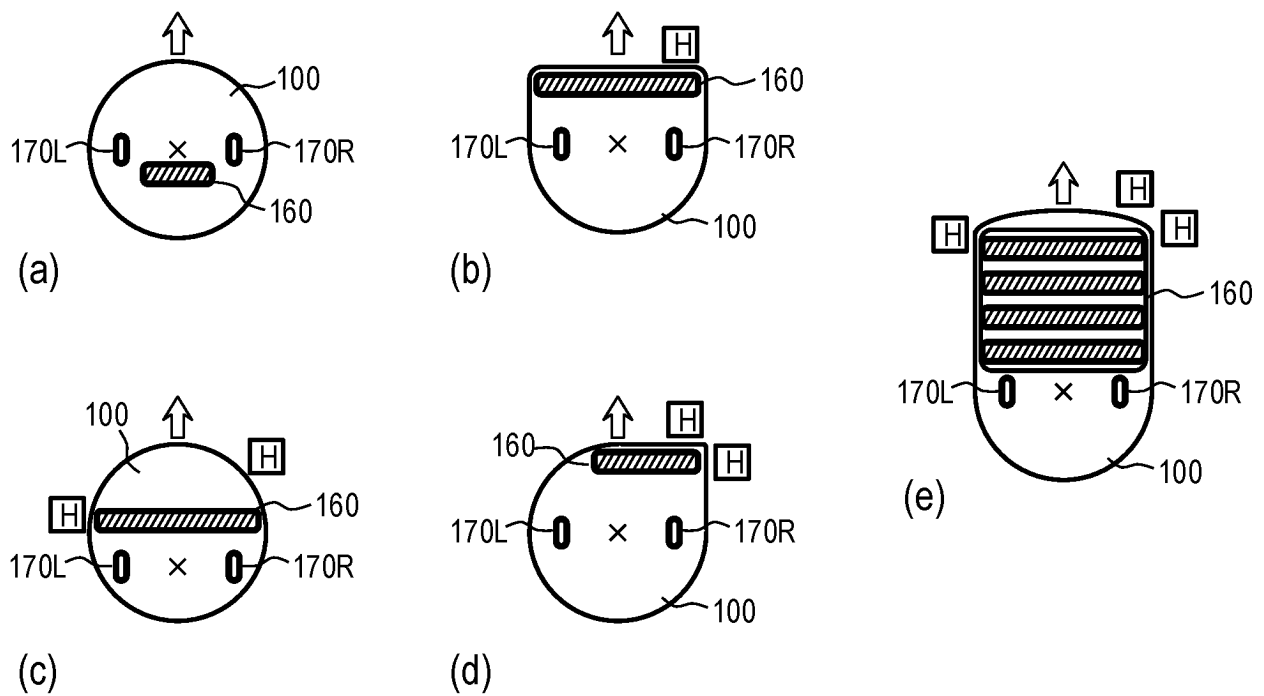


Fig. 3

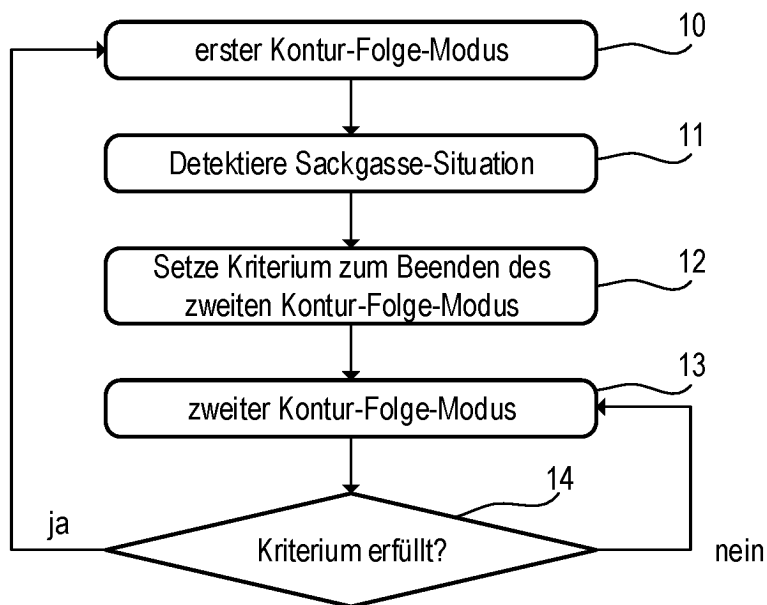


Fig. 4

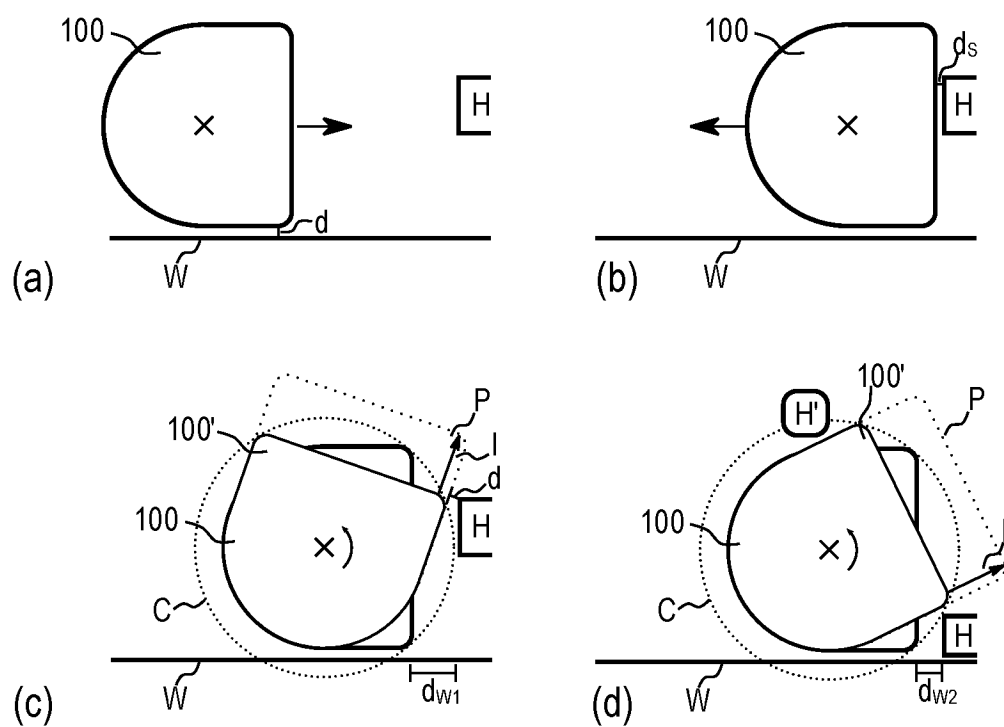


Fig. 5

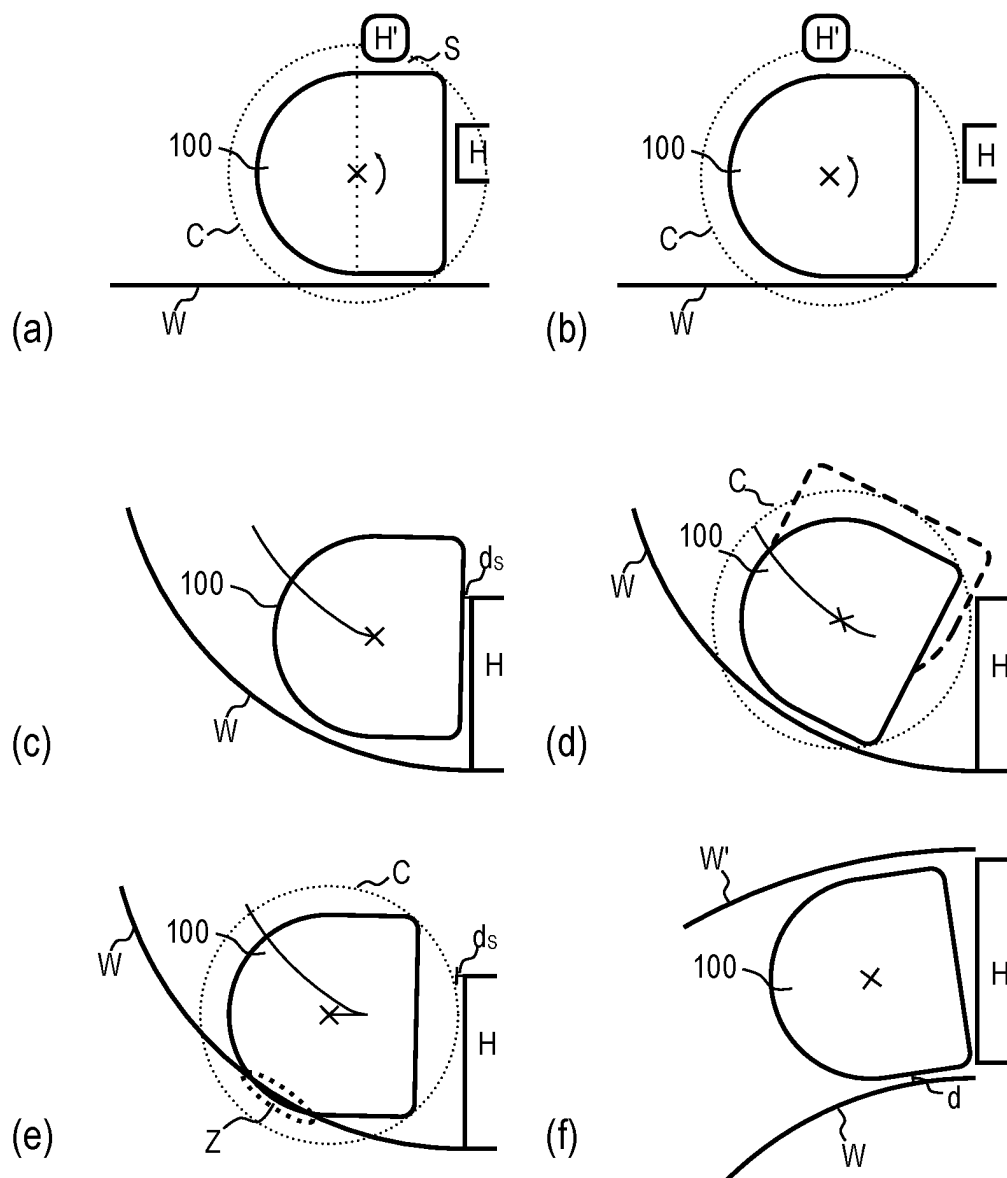


Fig. 6

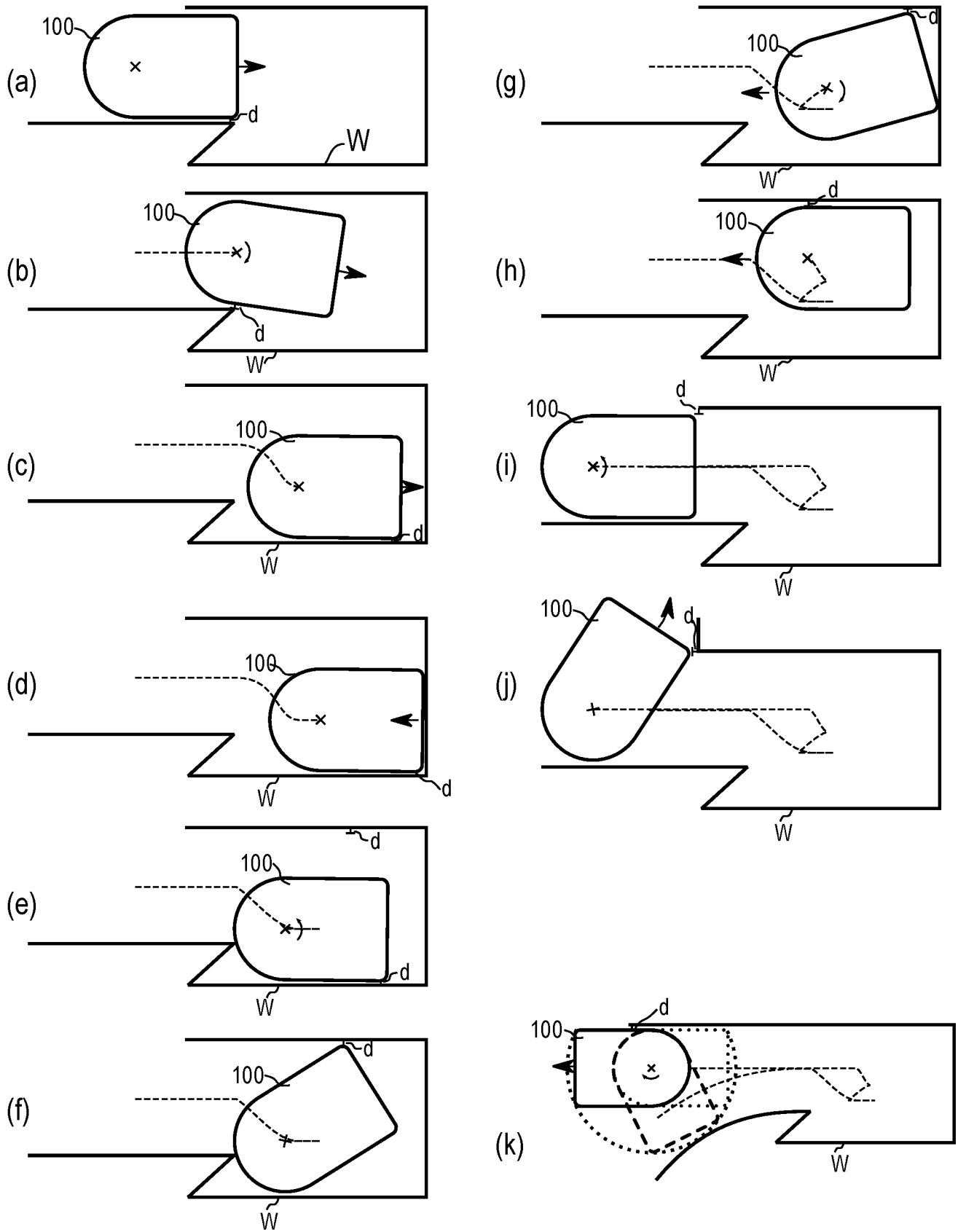


Fig. 7

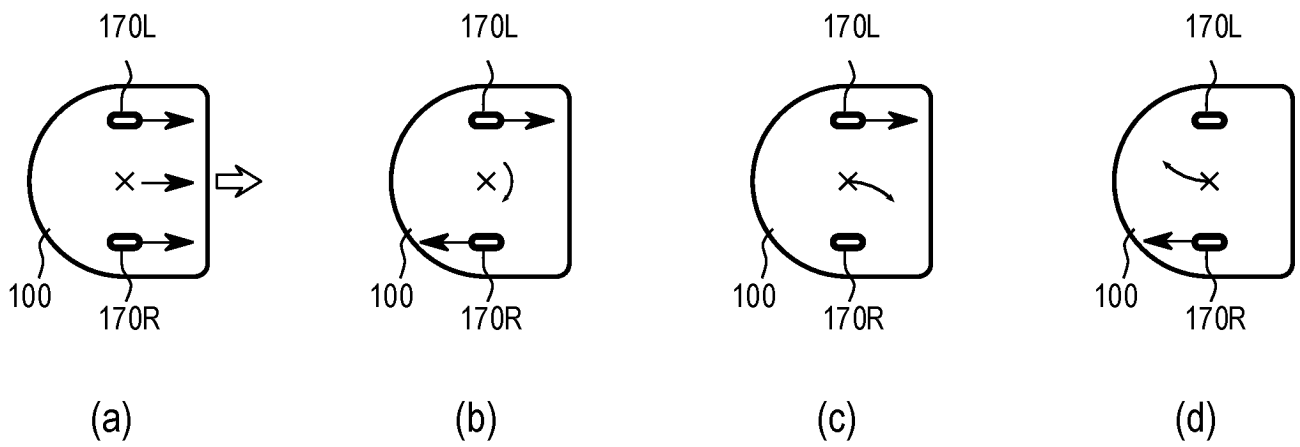


Fig. 8

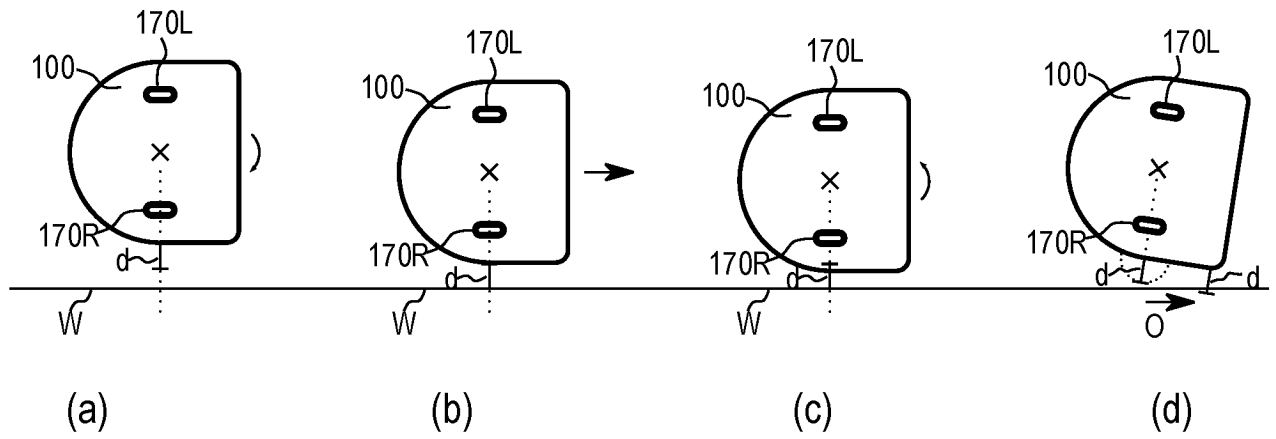


Fig. 9

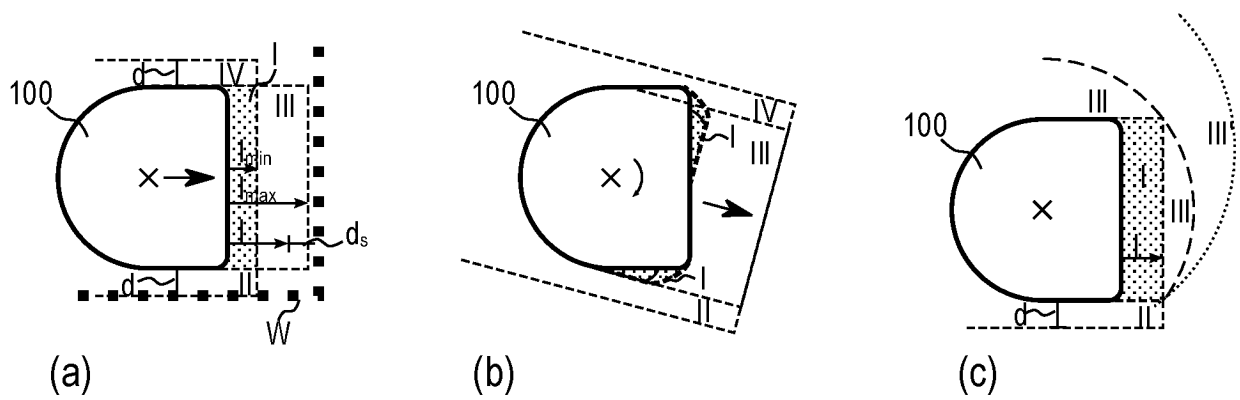


Fig. 10

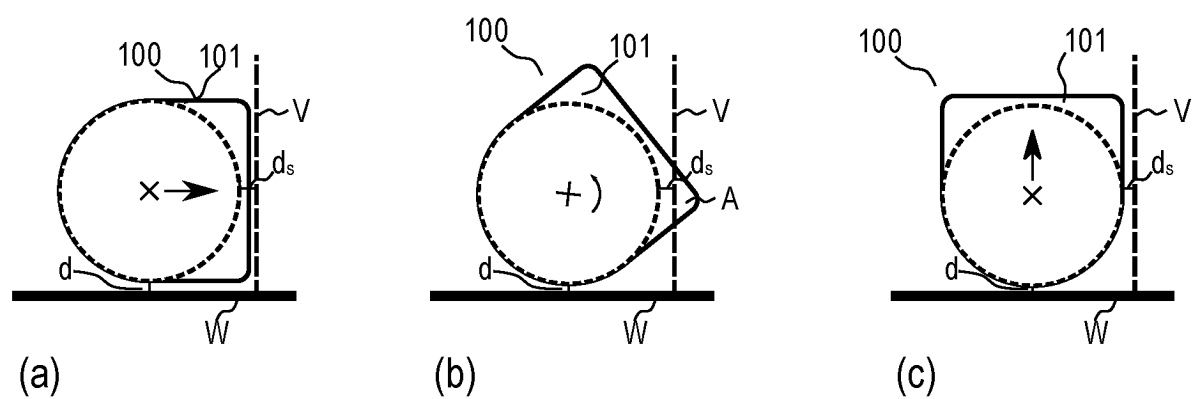


Fig. 11

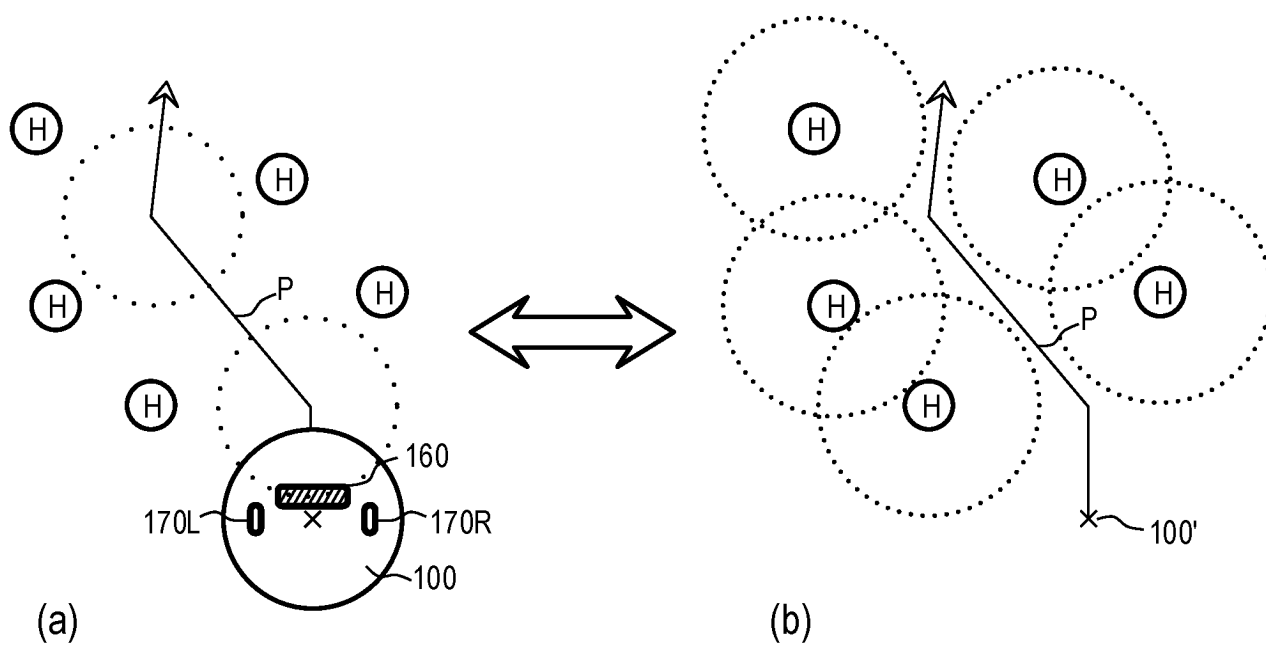


Fig. 12

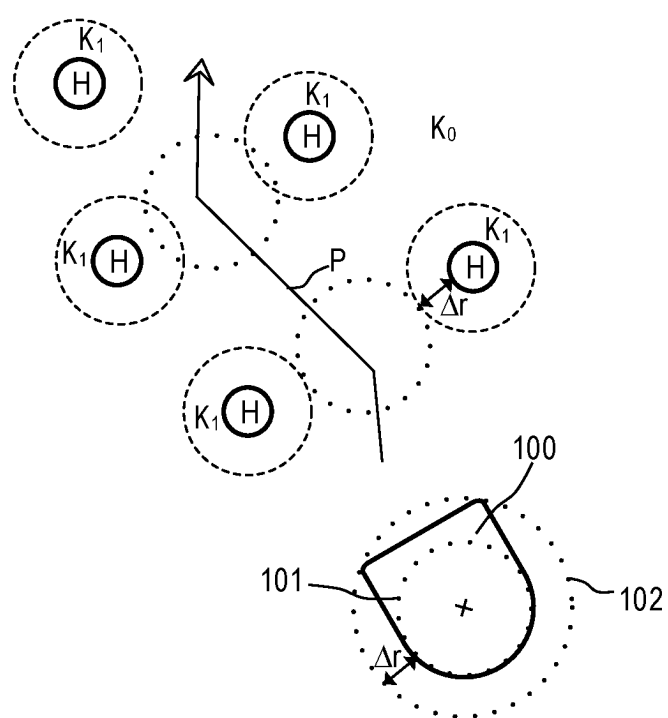


Fig. 13