



(10) **DE 10 2020 207 085 A1** 2021.12.09

(12) **Offenlegungsschrift**

(21) Aktenzeichen: **10 2020 207 085.7**

(22) Anmeldetag: **05.06.2020**

(43) Offenlegungstag: **09.12.2021**

(51) Int Cl.: **B25J 9/16 (2006.01)**

B25J 13/00 (2006.01)

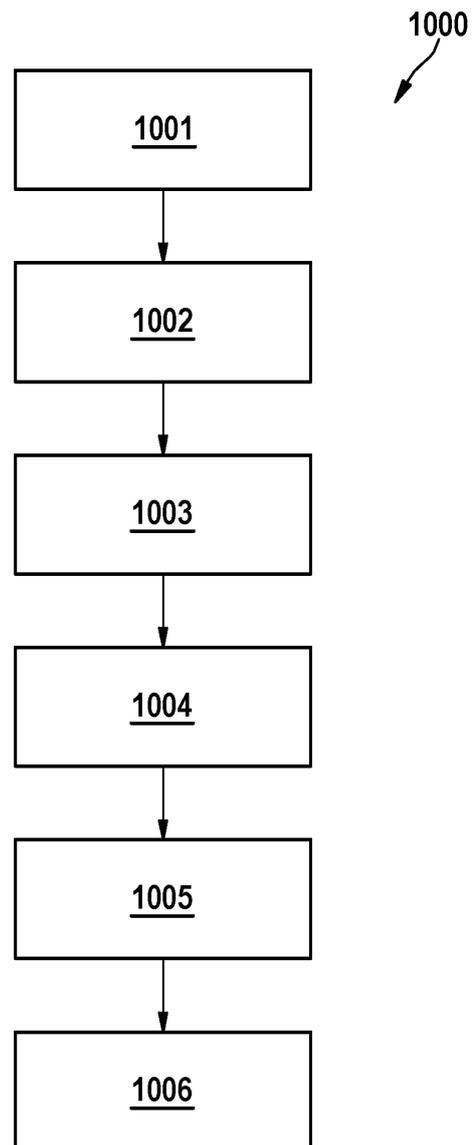
(71) Anmelder:
**Robert Bosch Gesellschaft mit beschränkter
Haftung, 70469 Stuttgart, DE**

(72) Erfinder:
**Buegger, Mathias, 70563 Stuttgart, DE; Guo,
Meng, 71272 Renningen, DE**

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **VERFAHREN ZUM STEUERN EINES ROBOTERS UND ROBOTERSTEUEREINHEIT**

(57) Zusammenfassung: Gemäß verschiedenen Ausführungsformen wird ein Verfahren zum Steuern eines Roboters geschaffen, das das Bereitstellen von Demonstrationen zum Durchführen von jeder von mehreren Fähigkeiten, das Trainieren eines Robotertrajektorienmodells für jede Fähigkeit aus den Demonstrationen, wobei jedes Trajektorienmodell ein verborgenes Semi-Markov-Modell mit einem oder mehreren anfänglichen Zuständen und einem oder mehreren Endzuständen ist, das Trainieren eines Vorbedingungsmodells für jede Fähigkeit aus den Demonstrationen, das für jeden anfänglichen Zustand des Robotertrajektorienmodells der Fähigkeit eine Wahrscheinlichkeitsverteilung von Roboterkonfigurationen vor dem Ausführen der Fähigkeit umfasst, und eines Endbedingungsmodells für jede Fähigkeit, das für jeden Endzustand des Robotertrajektorienmodells der Fähigkeit eine Wahrscheinlichkeitsverteilung von Roboterkonfigurationen nach dem Ausführen der Fähigkeit umfasst, das Empfangen einer Beschreibung einer Aufgabe, wobei die Aufgabe das Durchführen der Fähigkeiten der mehreren Fähigkeiten in einer Sequenz und/oder in Zweigen umfasst, das Erzeugen eines zusammengesetzten Robotertrajektorienmodells und das Steuern des Roboters gemäß dem zusammengesetzten Robotertrajektorienmodell, um die Aufgabe auszuführen, umfasst.



Beschreibung

[0001] Die vorliegende Offenbarung bezieht sich auf Verfahren zum Steuern eines Roboters und Robotersteuereinheiten.

[0002] Roboterfähigkeiten können durch Methoden des Lernens aus Demonstrationen programmiert werden, wobei ein nominaler Plan einer Fähigkeit durch einen Roboter aus Demonstrationen gelernt wird. Die anzuwendende Fähigkeit, um eine bestimmte Aufgabe auszuführen, kann jedoch in Abhängigkeit von dem anfänglichen Systemzustand variieren. Wenn beispielsweise die Aufgabe das Aufgreifen eines Objekts umfasst, kann die anzuwendende Fähigkeit „greife Objekt von der Seite auf“ oder „greife das Objekt von oben auf“ in Abhängigkeit davon, wie das Objekt orientiert ist, sein. Wenn das Objekt von der Seite aufgegriffen wird, kann es ferner erforderlich sein, das Objekt umzuorientieren, um in der Aufgabe weiter vorzugehen. Daher sind Methoden, die ermöglichen, dass Roboter mehrere Fähigkeiten sequenzieren und sie alternativ in sich ändernden Situationen anwenden, erwünscht.

[0003] In L. Schwenkel, M. Guo, und M. Bürger, „Optimizing sequences of probabilistic manipulation skills learned from demonstration,“ in Conference on Robot Learning, 2019, (die im Folgenden als Referenz [1] bezeichnet wird), ist eine auf Fähigkeiten fixierte Methode beschrieben, wobei jede Fähigkeit unabhängig unter verschiedenen Szenarios gelernt wird, aber nicht an irgendeine spezielle Aufgabe gebunden ist.

[0004] Gemäß verschiedenen Ausführungsformen wird ein Verfahren zum Steuern eines Roboters geschaffen, das das Bereitstellen von Demonstrationen zum Durchführen von jeder von mehreren Fähigkeiten, das Trainieren eines Robotertrajektorienmodells für jede Fähigkeit aus den Demonstrationen, wobei jedes Trajektorienmodell ein verborgenes Semi-Markov-Modell mit einem oder mehreren anfänglichen Zuständen und einem oder mehreren Endzuständen ist, das Trainieren eines Vorbedingungsmodells für jede Fähigkeit aus den Demonstrationen, das für jeden anfänglichen Zustand des Robotertrajektorienmodells der Fähigkeit eine Wahrscheinlichkeitsverteilung von Roboterkonfigurationen vor dem Ausführen der Fähigkeit umfasst, und eines Endbedingungsmodells für jede Fähigkeit, das für jeden Endzustand des Robotertrajektorienmodells der Fähigkeit eine Wahrscheinlichkeitsverteilung von Roboterkonfigurationen nach dem Ausführen der Fähigkeit umfasst, das Empfangen einer Beschreibung einer Aufgabe, wobei die Aufgabe das Durchführen der Fähigkeiten der mehreren Fähigkeiten in einer Sequenz und/oder in Zweigen umfasst, das Erzeugen eines zusammengesetzten Robotertrajektorienmodells durch

- wenn zwei Fähigkeiten in einer Sequenz in der Aufgabe durchgeführt werden sollen, das Kaskadieren der Robotertrajektorienmodelle der Fähigkeiten durch

- Aufnehmen der Zustände der Trajektorienmodelle der zwei Fähigkeiten in das zusammengesetzte Robotertrajektorienmodell und

- Berechnen einer Übergangswahrscheinlichkeit zwischen jedem Endzustand des Trajektorienmodells der ersten Fähigkeit der zwei Fähigkeiten und jedem anfänglichen Zustand des Trajektorienmodells der zweiten Fähigkeit der zwei Fähigkeiten als Funktion der Ähnlichkeit zwischen der Wahrscheinlichkeitsverteilung des Endbedingungsmodells der ersten Fähigkeit für den Endzustand der ersten Fähigkeit und der Wahrscheinlichkeitsverteilung des anfänglichen Modells der zweiten Fähigkeit für den anfänglichen Zustand für die zweite Fähigkeit

- wenn zwei Fähigkeiten in Zweigen in der Aufgabe durchgeführt werden, das Kombinieren der Robotertrajektorienmodelle der Fähigkeiten durch

- Aufnehmen der Zustände der Trajektorienmodelle der zwei Fähigkeiten in das zusammengesetzte Robotertrajektorienmodell und

- Setzen der Übergangswahrscheinlichkeit zwischen Zuständen der ersten Fähigkeit und Zuständen der zweiten Fähigkeit auf null; und

das Steuern des Roboters gemäß dem zusammengesetzten Robotertrajektorienmodell, um die Aufgabe durchzuführen, umfasst.

[0005] Gemäß einer weiteren Ausführungsform wird eine Robotersteuereinheit, die dazu konfiguriert ist, das vorstehend beschriebene Verfahren durchzuführen, geschaffen.

[0006] Dieses Verfahren zum Steuern eines Roboters und die Robotersteuereinheit, die vorstehend beschrieben sind, ermöglichen eine automatische Zusammensetzung von Manipulationsfähigkeiten, um eine festge-

legte Manipulationsaufgabe durchzuführen. Folglich muss ein Benutzer oder Bediener nicht manuell Verzweigungsbedingungen zum Wählen unter verschiedenen Fähigkeiten definieren, um die Aufgabe durchzuführen. Mit anderen Worten, nach dem Festlegen des Aufgabendiagramms kann der Bediener es direkt abarbeiten, selbst wenn das Aufgabendiagramm Fähigkeiten in Zweigen, d. h. den Bedarf, zwischen verschiedenen Fähigkeiten in Abhängigkeit vom aktuellen Systemzustand (z. B. Roboter- und/oder Objektkonfiguration) auszuwählen, umfasst.

[0007] Dies spart eine signifikante Anstrengung beim Durchführen einer Manipulationsaufgabe, da die Bewertung von Verzweigungsbedingungen einen direkten Zugriff auf verschiedene Sensoren wie z. B. Roboter gelenkzustände, Lage von Objekten, die manipuliert werden, erfordert. Das Lesen und Aufzeichnen der korrekten Daten können ziemlich zeitraubend sein. Ferner müssen solche Daten direkt in den Quellencode geschrieben werden, was einen direkten Zugriff auf den Quellencode erfordern würde. Überdies erhöhen das geschaffene Steuerverfahren und die geschaffene Robotersteuereinheit die Robustheit der Aufgabenausführung, da es häufig der Fall ist, dass es der Kombination von Verzweigungsbedingungen misslingt, den ganzen Zustandsraum abzudecken, und folglich Situationen bestehen, in denen keine Bedingungen erfüllt sind, was zu einem Ausfall führt.

[0008] Im Folgenden werden verschiedene Beispiele gegeben.

[0009] Beispiel 1 ist ein Verfahren zum Steuern eines Roboters, wie vorstehend beschrieben.

[0010] Beispiel 2 ist das Verfahren von Beispiel 1, wobei jede Fähigkeit eine Manipulation von einem oder mehreren Objekten durch den Roboter umfasst und das verborgene Semi-Markov-Modell von jeder Fähigkeit ein aufgabenparametrisiertes verborgenes Semi-Markov-Modell mit Aufgabenparametern ist, die einer Roboter- und Objektkonfiguration entsprechen, in der die Fähigkeit angewendet wird.

[0011] Insbesondere ermöglichen verschiedene Ausführungsformen ein effizientes Training und eine effiziente Steuerung eines Roboters für die Handhabung von einem oder mehreren Objekten.

[0012] Beispiel 3 ist das Verfahren von Beispiel 2, das ferner das Trainieren eines Effektmodells für jede Fähigkeit umfasst, das für jeden Endzustand des Robotertrajektorienmodells der Fähigkeit eine Wahrscheinlichkeitsverteilung dessen, wie die Roboter- und/oder Objektkonfiguration nach dem Ausführen der Fähigkeit relativ zu einer anfänglichen Roboter- und/oder Objektkonfiguration verändert ist, auf die die Fähigkeit angewendet wird, umfasst, wobei, wenn zwei Fähigkeiten in einer Sequenz in der Aufgabe durchgeführt werden sollen, das Kaskadieren der Robotertrajektorienmodelle der Fähigkeiten das Transformieren des aufgabenparametrisierten verborgenen Semi-Markov-Modells der zweiten Fähigkeit umfasst, so dass seine Aufgabenparameter Aufgabenparameter sind, die der Roboter- und/oder Objektkonfiguration entsprechen, die durch das Effektmodell der ersten Fähigkeit gegeben ist.

[0013] Folglich kann sichergestellt werden, dass eine Steuertrajektorie über eine Sequenz von Fähigkeiten korrekt berechnet wird. Die Transformation des Modells stellt beispielsweise sicher, dass die zugehörige globale Gauß-Glocke der zweiten Fähigkeit auf der Basis des anfänglichen Zustandes der ersten Fähigkeit direkt berechnet werden kann. Die globale Gauß-Glocke kann dann verwendet werden, um Beobachtungswahrscheinlichkeiten zu berechnen, z. B. für die Anwendung eines Viterbi-Algorithmus.

[0014] Beispiel 4 ist das Verfahren von Beispiel 3, das ferner das affine Transformieren des Effektmodells der zweiten Fähigkeit umfasst, so dass seine Aufgabenparameter die Aufgabenparameter sind, die der Roboter- und/oder Objektkonfiguration entsprechen, die durch das Effektmodell der ersten Fähigkeit gegeben ist.

[0015] Dies ermöglicht eine korrekte Berechnung einer Steuertrajektorie über die Sequenz von Fähigkeiten in einem Szenario mit einem oder mehreren Objekten, die beispielsweise durch den Roboter manipuliert werden sollen.

[0016] Beispiel 5 ist das Verfahren von irgendeinem der Beispiele 1 bis 4, das ferner, wenn zwei Fähigkeiten in einer Sequenz in der Aufgabe durchgeführt werden sollen, die Verwendung des Vorbedingungsmodells der ersten Fähigkeit als Vorbedingungsmodell für die kaskadierten Robotertrajektorienmodelle der Fähigkeiten der zwei Fähigkeiten und die Verwendung des Endbedingungsmodells der zweiten Fähigkeit als Endbedingungsmodell für die kaskadierten Robotertrajektorienmodelle der Fähigkeiten umfasst.

[0017] Durch Berechnen eines Vorbedingungsmodells für eine Kaskade von Fähigkeiten können weitere Fähigkeiten hinzugefügt werden, z. B. kann ein zusammengesetztes Modell für mehr als zwei Fähigkeiten in einer Sequenz oder in Zweigen bestimmt werden. Dies ermöglicht das Training und die Steuerung für komplexe Aufgaben.

[0018] Beispiel 6 ist das Verfahren von irgendeinem der Beispiele 1 bis 5, das ferner, wenn zwei Fähigkeiten in einem Zweig in der Aufgabe durchgeführt werden sollen, das Aufnehmen sowohl des Vorbedingungsmodells für die erste Fähigkeit als auch des Vorbedingungsmodells für die zweite Fähigkeit in das Vorbedingungsmodell der kombinierten Robotertrajektorienmodelle der Fähigkeiten und das Aufnehmen sowohl des Endbedingungsmodells für die erste Fähigkeit als auch des Endbedingungsmodells für die zweite Fähigkeit in das Vorbedingungsmodell der kombinierten Robotertrajektorienmodelle der Fähigkeiten umfasst.

[0019] Durch Berechnen eines Vorbedingungsmodells für Zweige von Fähigkeiten können weitere Fähigkeiten hinzugefügt werden, z. B. kann ein zusammengesetztes Modell für mehr als zwei Fähigkeiten in einer Sequenz von Zweigen bestimmt werden. Dies ermöglicht das Training und die Steuerung für komplexe Aufgaben.

[0020] Beispiel 7 ist das Verfahren von irgendeinem der Beispiele 1 bis 6, wobei die Ähnlichkeit zwischen der Wahrscheinlichkeitsverteilung des Endbedingungsmodells der ersten Fähigkeit für den Endzustand der ersten Fähigkeit und der Wahrscheinlichkeitsverteilung des anfänglichen Modells der zweiten Fähigkeit für den anfänglichen Zustand für die zweite Fähigkeit eine KL-Divergenz der Wahrscheinlichkeitsverteilung des Endbedingungsmodells der ersten Fähigkeit für den Endzustand der ersten Fähigkeit und der Wahrscheinlichkeitsverteilung des anfänglichen Modells der zweiten Fähigkeit für den anfänglichen Zustand für die zweite Fähigkeit ist.

[0021] Die Verwendung der KL-Divergenz ermöglicht eine effiziente Verkettung von Trajektorienmodellen für Sequenzen von Fähigkeiten.

[0022] Beispiel 8 ist das Verfahren von irgendeinem der Beispiele 1 bis 7, wobei das Vorbedingungsmodell und das Endbedingungsmodell aufgabenparametrisierte Gauß-Mischmodelle sind.

[0023] Eine Gauß-Glocke wird beispielsweise für jeden Aufgabenparameterwert bestimmt. Die Aufgabenparametrisierung ermöglicht die Anwendung von trainierten Modellen in verschiedenen Szenarios (d. h. variierende anfängliche Systemzustände (Konfigurationen)).

[0024] Beispiel 9 ist das Verfahren von irgendeinem der Beispiele 1 bis 8, wobei das Erzeugen des zusammengesetzten Robotertrajektorienmodells das wiederholte Kaskadieren von Robotertrajektorienmodellen, kaskadierten Robotertrajektorienmodellen und kombinierten Robotertrajektorienmodellen und das Kombinieren von Robotertrajektorienmodellen, kaskadierten Robotertrajektorienmodellen und einer kombinierten Robotertrajektorie gemäß der Aufgabe umfasst, so dass das Robotertrajektorienmodell ein Trajektorienmodell für die ganze Aufgabe ist.

[0025] Folglich kann ein zusammengesetztes Modell für eine komplexe Aufgabe iterativ bestimmt werden, was Robustheit für eine komplexe Aufgabe sicherstellt und einen Bediener von dem Bedarf befreit, eine komplexe Hierarchie von Verzweigungsbedingungen zu definieren.

[0026] Beispiel 10 ist eine Robotersteuereinheit, die dazu konfiguriert ist, das Verfahren von irgendeinem der Beispiele 1 bis 9 durchzuführen.

[0027] Beispiel 11 ist ein Computerprogramm mit Befehlen, die, wenn sie durch einen Prozessor ausgeführt werden, bewirken, dass der Prozessor ein Verfahren gemäß irgendeinem der Beispiele 1 bis 9 durchführt.

[0028] Beispiel 12 ist ein computerlesbares Medium, das Befehle speichert, die, wenn sie durch einen Prozessor ausgeführt werden, bewirken, dass der Prozessor ein Verfahren gemäß irgendeinem der Beispiele 1 bis 9 durchführt.

[0029] In den Zeichnungen beziehen sich ähnliche Bezugszeichen im Allgemeinen auf dieselben Teile in den ganzen verschiedenen Ansichten. Die Zeichnungen sind nicht notwendigerweise maßstäblich, wobei die Betonung stattdessen im Allgemeinen auf die Darstellung der Prinzipien der Erfindung gelegt wird. In der folgenden Beschreibung werden verschiedene Aspekte mit Bezug auf die folgenden Zeichnungen beschrieben, in denen:

Fig. 1 einen Roboter zeigt.

Fig. 2 ein Ablaufdiagramm zeigt, das eine Manipulationsaufgabe mit Fähigkeiten in einer Sequenz und in Zweigen darstellt.

Fig. 3 ein Ablaufdiagramm zeigt, das ein Verfahren zum Steuern eines Roboters gemäß einer Ausführungsform darstellt.

Fig. 4 eine Kaskadierungsoperation zum Kaskadieren von Robotertrajektorienmodellen von Fähigkeiten, die in einer Sequenz ausgeführt werden sollen, zu einem zusammengesetzten Robotertrajektorienmodell darstellt.

Fig. 5 eine Kombinationsoperation zum Kombinieren von Robotertrajektorienmodellen von Fähigkeiten, die in Zweigen (d. h. alternativ) ausgeführt werden sollen, zu einem zusammengesetzten Robotertrajektorienmodell darstellt.

Fig. 6 das resultierende Aufgabendiagramm nach einer Anwendung einer Kaskadierungsoperation auf das Aufgabendiagramm von **Fig. 2** zeigt.

Fig. 7 das resultierende Aufgabendiagramm nach einer Anwendung einer Kombinationsoperation auf das Aufgabendiagramm von **Fig. 6** zeigt.

Fig. 8 das resultierende Aufgabendiagramm nach einer Anwendung einer Kaskadierungsoperation auf das Aufgabendiagramm von **Fig. 7** zeigt.

Fig. 9 das resultierende Aufgabendiagramm nach einer Anwendung einer Kombinationsoperation auf das Aufgabendiagramm von **Fig. 8** zeigt.

Fig. 10 ein Ablaufdiagramm zeigt, das ein Verfahren zum Steuern eines Roboters gemäß einer Ausführungsform darstellt.

[0030] Die folgende ausführliche Beschreibung bezieht sich auf die begleitenden Zeichnungen, die zur Erläuterung spezielle Details und Aspekte dieser Offenbarung zeigen, in denen die Erfindung ausgeführt werden kann. Andere Aspekte können verwendet werden und strukturelle, logische und elektrische Änderungen können durchgeführt werden, ohne vom Schutzbereich der Erfindung abzuweichen. Die verschiedenen Aspekte dieser Offenbarung schließen sich nicht notwendigerweise gegenseitig aus, da einige Aspekte dieser Offenbarung mit einem oder mehreren anderen Aspekten dieser Offenbarung kombiniert werden können, um neue Aspekte zu bilden.

[0031] Im Folgenden werden verschiedene Beispiele genauer beschrieben.

[0032] **Fig. 1** zeigt einen Roboter **100**.

[0033] Der Roboter **100** umfasst einen Roboterarm **101**, beispielsweise einen Industrieroboterarm zum Handhaben oder Montieren eines Werkstücks (oder eines oder mehrerer anderer Objekte). Der Roboterarm **101** umfasst Manipulatoren **102, 103, 104** und eine Basis (oder Stütze) **105**, durch die die Manipulatoren **102, 103, 104** abgestützt sind. Der Begriff „Manipulator“ bezieht sich auf die beweglichen Elemente des Roboterarms **101**, deren Betätigung eine physikalische Wechselwirkung mit der Umgebung ermöglicht, z. B. um eine Aufgabe auszuführen. Für die Steuerung umfasst der Roboter **100** eine (Roboter-) Steuereinheit **106**, die dazu konfiguriert ist, die Wechselwirkung mit der Umgebung gemäß einem Steuerprogramm zu implementieren. Das letzte Element **104** (am weitesten von der Stütze **105** entfernt) der Manipulatoren **102, 103, 104** wird auch als Endeffektor **104** bezeichnet und kann ein oder mehrere Werkzeuge wie z. B. einen Schweißbrenner, ein Greifinstrument, eine Anstreichausrüstung oder dergleichen umfassen.

[0034] Die anderen Manipulatoren **102, 103** (näher an der Stütze **105**) können eine Positionierungsvorrichtung bilden, so dass zusammen mit dem Endeffektor **104** der Roboterarm **101** mit dem Endeffektor **104** an seinem Ende vorgesehen ist. Der Roboterarm **101** ist ein mechanischer Arm, der ähnliche Funktionen wie ein menschlicher Arm (möglicherweise mit einem Werkzeug an seinem Ende) vorsehen kann.

[0035] Der Roboterarm **101** kann Gelenkelemente **107, 108, 109** umfassen, die die Manipulatoren **102, 103, 104** miteinander und mit der Stütze **105** verbinden. Ein Gelenkelement **107, 108, 109** kann ein oder mehrere Gelenke aufweisen, von denen jedes eine drehbare Bewegung (d. h. Drehbewegung) und/oder Translationsbewegung (d. h. Verlagerung) zu zugehörigen Manipulatoren relativ zueinander bereitstellen kann. Die Bewegung der Manipulatoren **102, 103, 104** kann mittels Aktuatoren eingeleitet werden, die durch die Steuereinheit **106** gesteuert werden.

[0036] Der Begriff „Aktuator“ kann als Komponente verstanden werden, die dazu ausgelegt ist, einen Mechanismus oder Prozess in Reaktion darauf, dass sie angetrieben wird, zu beeinflussen. Der Aktuator kann Befehle, die durch die Steuereinheit **106** ausgegeben werden (die sogenannte Aktivierung), in mechanische Bewegungen implementieren. Der Aktuator, z. B. ein elektromechanischer Wandler, kann dazu konfiguriert sein, elektrische Energie in mechanische Energie in Reaktion auf den Antrieb umzuwandeln.

[0037] Der Begriff „Steuereinheit“ kann als irgendein Typ von Logik, die eine Entität implementiert, verstanden werden, die beispielsweise eine Schaltung und/oder einen Prozessor, der in der Lage ist, eine Software auszuführen, die in einem Speichermedium gespeichert ist, Firmware oder eine Kombination davon umfassen kann, und die Befehle, z. B. an einen Aktuator im vorliegenden Beispiel, ausgeben kann. Die Steuereinheit kann beispielsweise durch einen Programmcode (z. B. Software) konfiguriert sein, um den Betrieb eines Systems, eines Roboters im vorliegenden Beispiel, zu steuern.

[0038] Im vorliegenden Beispiel umfasst die Steuereinheit **106** einen oder mehrere Prozessoren **110** und einen Speicher **111**, der einen Code und Daten speichert, auf deren Basis der Prozessor **110** den Roboterarm **101** steuert. Gemäß verschiedenen Ausführungsformen steuert die Steuereinheit **106** den Roboterarm **101** auf der Basis eines statistischen Modells **112**, das im Speicher **111** gespeichert ist.

[0039] Ein Roboter **100** kann Methoden des Lernens aus Demonstration (LfD) ausnutzen, um zu lernen, eine Aufgabe auszuführen oder mit einem menschlichen Partner zusammenzuarbeiten. Menschliche Demonstrationen können durch ein probabilistisches Modell (auch als statistisches Modell bezeichnet) codiert werden, das den nominalen Plan der Aufgabe für den Roboter darstellt. Die Steuereinheit **106** kann anschließend das statistische Modell verwenden, das auch als Robotertrajektorienmodell bezeichnet wird, um gewünschte Roboterbewegungen zu erzeugen, möglicherweise als Funktion des Zustandes sowohl des menschlichen Partners als auch der Umgebung.

[0040] Die Grundidee von LfD besteht darin, ein vorgeschriebenes Fähigkeitsmodell wie z. B. GMMs an eine Handvoll Demonstrationen anzupassen. M Demonstrationen sollen vorhanden sein, von denen jede T_m

Datenpunkte für einen Datensatz von $N = \sum_m T_m$ Gesamtbeobachtungen $\xi = \{\xi_t\}_{t=1}^N$ enthält, wobei $\xi_t \in \mathbb{R}^d$. Es wird auch angenommen, dass dieselben Demonstrationen aus der Perspektive von P verschiedenen Koordinatensystemen aufgezeichnet werden (durch die Aufgabenparameter gegeben, wie z. B. lokale Koordinatensysteme oder Rahmen von interessierenden Objekten). Eine übliche Weise zum Erhalten solcher Daten besteht darin, die Demonstrationen von einem statischen globalen Rahmen auf einen Rahmen p durch

$\xi_t^{(p)} = \mathbf{A}^{(p)-1} (\xi_t - \mathbf{b}^{(p)})$ zu transformieren. Hier ist $\left\{ \left\{ \mathbf{b}^{(p)}, \mathbf{A}^{(p)} \right\} \right\}_{p=1}^P$ die Translation und Rotation des (lokalen) Rahmens p in Bezug auf den Weltraum (d. h. globalen Rahmen). Dann wird ein TP-GMM durch die Modell-

parameter $\left\{ \pi_k, \left\{ \mu_k^{(p)}, \Sigma_k^{(p)} \right\}_{p=1}^P \right\}_{k=1}^K$ beschrieben, wobei K die Anzahl von Gauß-Komponenten im Mischmodell

darstellt, π_k die vorherige Wahrscheinlichkeit jeder Komponente ist und $\left\{ \mu_k^{(p)}, \Sigma_k^{(p)} \right\}_{p=1}^P$ die Parameter der k -ten Gauß-Komponente innerhalb des Rahmens p sind.

[0041] Im Unterschied zum Standard-GMM kann das obige Mischmodell nicht unabhängig für jeden Rahmen gelernt werden. Tatsächlich werden die Mischkoeffizienten π_k durch alle Rahmen geteilt und die k -te Komponente im Rahmen p muss auf die entsprechende k -te Komponente im globalen Rahmen abbilden. Erwartungs-Maximierung (EM) ist ein fest etabliertes Verfahren, um solche Modelle zu lernen.

[0042] Sobald es gelernt ist, kann das TP-GMM während der Ausführung verwendet werden, um eine Trajek-

torie für die gelernte Fähigkeit zu reproduzieren. In Anbetracht der beobachteten Rahmen $\left\{ \mathbf{b}^{(p)}, \mathbf{A}^{(p)} \right\}_{p=1}^P$ wird

nämlich das gelernte TP-GMM in ein einzelnes GMM mit Parametern $\left\{ \pi_k, \left\{ \hat{\mu}_k, \hat{\Sigma}_k \right\} \right\}_{k=1}^K$ durch Multiplizieren der affin transformierten Gauß-Komponenten über verschiedene Rahmen umgewandelt, wie folgt

$$\hat{\Sigma}_k = \left[\sum_{p=1}^P \left(\hat{\Sigma}_k^{(p)} \right)^{-1} \right]^{-1}, \quad \hat{\mu}_k = \hat{\Sigma}_k \left[\sum_{p=1}^P \left(\hat{\Sigma}_k^{(p)} \right)^{-1} \hat{\mu}_k^{(p)} \right], \quad (1)$$

wobei die Parameter der aktualisierten Gauß-Glocke an jedem Rahmen p berechnet werden als $\hat{\mu}_k^{(p)} = \mathbf{A}^{(p)} \mu_k^{(p)} + \mathbf{b}^{(p)}$ und $\hat{\Sigma}_k^{(p)} = \mathbf{A}^{(p)} \Sigma_k^{(p)} \mathbf{A}^{(p)\top}$. Obwohl die Aufgabenparameter über die Zeit variieren können, wird der Zeitindex wegen der Schreibweise weggelassen.

[0043] Verborgene Semi-Markov-Modelle (HSMMs) erweitern verborgene Standard-Markov-Modelle (HMMs) durch Einbetten von Zeitinformationen des zugrundeliegenden stochastischen Prozesses. Das heißt, während in HMM der zugrundeliegende verborgene Prozess als Markov angenommen wird, d. h. die Wahrscheinlichkeit des Übergangs zum nächsten Zustand nur vom aktuellen Zustand abhängt, wird in HSMM der Zustandsprozess als Semi-Markov angenommen. Dies bedeutet, dass ein Übergang zum nächsten Zustand vom aktuellen Zustand sowie von der abgelaufenen Zeit, seitdem in den Zustand eingetreten wurde, abhängt. Sie können in Kombination mit TP-GMMs für Roboterfähigkeitscodierung angewendet werden, um räumlich-zeitliche Merkmale der Demonstrationen zu lernen. Insbesondere ist ein aufgabenparametrisiertes HSMM-Modell (TP-HSMM-Modell) definiert als:

$$\Theta = \left\{ \left\{ a_{hk} \right\}_{h=1}^K, \left(\mu_k^D, \sigma_k^D \right), \pi_k, \left\{ \left\{ \mu_k^{(p)}, \Sigma_k^{(p)} \right\}_{p=1}^P \right\}_{k=1}^K \right\},$$

wobei a_{hk} die Übergangswahrscheinlichkeit vom Zustand h zu k ist; (μ_k^D, σ_k^D) die Gauß-Verteilungen für die Dauer des Zustandes k beschreiben, d. h. die Wahrscheinlichkeit, dass im Zustand k für eine bestimmte Anzahl

von aufeinander folgenden Schritten geblieben wird; $\left\{ \pi_k, \left\{ \mu_k^{(p)}, \Sigma_k^{(p)} \right\}_{p=1}^P \right\}_{k=1}^K$ gleich dem früher eingeführten TP-GMM ist, das die Beobachtungswahrscheinlichkeit darstellt, die dem Zustand k entspricht. Hier ist zu beachten, dass die Anzahl von Zuständen der Anzahl von Gauß-Komponenten im „angebundenen“ TP-GMM darstellt.

[0044] In Anbetracht einer bestimmten (teilweisen) Sequenz von beobachteten Datenpunkten $\{\xi_\ell\}_{\ell=1}^t$, soll angenommen werden, dass die zugehörige Sequenz von Zuständen in Θ durch $s_t = s_1 s_2 \dots s_t$ gegeben ist. Die Wahrscheinlichkeit, dass der Datenpunkt ξ_t zum Zustand k gehört (d. h. $s_t = k$), ist durch die Vorwärtsvariable $\alpha_t(k) = p\left(s_t = k, \{\xi_\ell\}_{\ell=1}^t\right)$ gegeben:

$$\alpha_t(k) = \sum_{\tau=1}^{t-1} \sum_{h=1}^K \alpha_{t-\tau}(h) a_{hk} \mathcal{N}\left(\tau \mid \mu_k^D, \sigma_k^D\right) o_\tau^t, \quad (2)$$

wobei $o_\tau^t = \prod_{\ell=t-\tau+1}^t \mathcal{N}\left(\xi_\ell \mid \hat{\mu}_k, \hat{\Sigma}_k\right)$ die Emissionswahrscheinlichkeit ist und $(\hat{\mu}_k, \hat{\Sigma}_k)$ von (1) in Anbetracht der Aufgabenparameter abgeleitet sind. Ferner kann dieselbe Vorwärtsvariable auch während der Reproduktion verwendet werden, um zukünftige Schritte bis T_m vorherzusagen.

[0045] Da jedoch in diesem Fall zukünftige Beobachtungen nicht verfügbar sind, werden nur Übergangs- und Dauerinformationen verwendet, d. h. durch Setzen von $\mathcal{N}\left(\xi_\ell \mid \hat{\mu}_k, \hat{\Sigma}_k\right) = 1$ für alle k und $\ell > t$ in (2). Schließlich wird die Sequenz der wahrscheinlichsten Zustände $\mathbf{s}_{T_m}^* = \mathbf{s}_1^* \mathbf{s}_2^* \dots \mathbf{s}_{T_m}^*$ durch Wählen von $\mathbf{s}_t^* = \operatorname{argmax}_k \alpha_t(k), \forall 1 \leq t \leq T_m$ bestimmt.

[0046] Nun soll eine gewünschte Endbeobachtung des Roboterzustandes als ξ_T gegeben sein, wobei T der Fähigkeitszeithorizont (z. B. die mittlere Länge über die Demonstrationen) ist. Überdies wird der anfängliche Roboterzustand als ξ_1 beobachtet. Für die Ausführung der Fähigkeit (d. h. Fähigkeitsreproduktion) in Anbe-

tracht des gelernten Modells Θ_a wird nur die wahrscheinlichste Zustandssequenz \mathbf{s}_T^* in Anbetracht von nur ξ_1 und ξ_T konstruiert.

[0047] Die Reproduktion unter Verwendung der Vorwärtsvariable kann in diesem Fall nicht direkt sein, da die Vorwärtsvariable in Gleichung (2) die Sequenz von marginal wahrscheinlichsten Zuständen berechnet, während das, was erwünscht ist, die gemeinsam wahrscheinlichste Sequenz von Zuständen in Anbetracht von ξ_1 und ξ_T ist. Folglich besteht, wenn (2) verwendet wird, keine Garantie, dass die zurückgegebene Sequenz

\mathbf{s}_T^* sowohl den räumlich-zeitlichen Mustern der Demonstrationen als auch der Endbeobachtung entspricht. Hinsichtlich eines Beispiels zum Aufgreifen eines Objekts kann es eine wahrscheinlichste Sequenz zurückgeben, die „Aufnehmen von der Seite“ entspricht, selbst wenn die gewünschte Endkonfiguration darin besteht, dass der Endeffektor sich an der Oberseite des Objekts befindet.

[0048] Um dieses Problem zu überwinden, wird gemäß einer Ausführungsform eine Modifikation des Viterbi-Algorithmus verwendet. Der klassische Viterbi-Algorithmus kann verwendet werden, um die wahrscheinlichste Sequenz von Zuständen (auch Viterbi-Pfad genannt) in HMMs zu finden, die zu einem gegebenen Strom von beobachteten Ereignissen führen. Gemäß einer Ausführungsform wird eine Methode verwendet, die sich von jener in zwei Hauptaspekten unterscheidet: (a) sie arbeitet an HSMM anstelle von HMM; und bedeutender (b) die meisten Beobachtungen abgesehen von der ersten und der letzten fehlen. Insbesondere wird bei Abwesenheit von Beobachtungen der Viterbi-Algorithmus zu

$$\begin{aligned} \delta_t(j) &= \max_{d \in \mathcal{D}} \max_{i \neq j} \delta_{t-d}(i) a_{ij} p_j(d) \prod_{t'=t-d+1}^t \tilde{b}_j(\xi_{t'}), \\ \delta_1(j) &= \pi_j(\xi_1) \pi_j p_j(1), \end{aligned} \quad (3)$$

wobei $p_j(d) = \mathcal{N}(d | \mu_j^D, \sigma_j^D)$ die Dauerwahrscheinlichkeit des Zustandes j ist, $\delta_t(j)$ die Wahrscheinlichkeit, dass das System sich im Zustand j zur Zeit t und nicht im Zustand j bei $t + 1$ befindet; und

$$\tilde{b}_j(\xi_t) = \begin{cases} \mathcal{N}(\xi_t | \hat{\mu}_j, \hat{\Sigma}_j), & t=1 \vee t=T; \\ 1, & 1 < t < T. \end{cases}$$

wobei $(\hat{\mu}_j, \hat{\Sigma}_j)$ die globale Gauß-Komponente j in Θ_a von (1) in Anbetracht von ξ_t ist. Zu jeder Zeit t und für jeden Zustand j werden nämlich die zwei Argumente, die die Gleichung $\delta_t(j)$ maximieren, aufgezeichnet und eine einfache Rückverfolgungsprozedur wird verwendet, um die wahrscheinlichste Zustandssequenz \mathbf{s}_T^* zu finden.

Mit anderen Worten, der obige Algorithmus leitet die wahrscheinlichste Sequenz \mathbf{s}_T^* für die Fähigkeit a , die die Endbeobachtung ξ_T ergibt, ausgehend von ξ_1 ab.

[0049] Wenn der Roboteraufgabenraum durch zeitlich variierende Lagen (mit Position und Orientierung) des Endeffektors dargestellt wird, sind klassische Verfahren auf euklidischer Basis für die Verarbeitung solcher Daten ungeeignet. Daher wird gemäß verschiedenen Ausführungsformen der Roboteraufgabenraum mit einer Riemannschen Mannigfaltigkeit M ausgestattet. Kurz existiert für jeden Punkt x in der Mannigfaltigkeit M ein Tangentenraum $T_x M$. Dies ermöglicht uns, euklidische Operationen lokal auszuführen, während sie geometrisch mit Mannigfaltigkeitseinschränkungen konsistent sind.

[0050] Exponentielle und logarithmische Abbildungen können verwendet werden, um Punkte zwischen $T_x M$ und M abzubilden. Die exponentielle Abbildung $\text{Exp}_x: T_x M \rightarrow M$ bildet einen Punkt im Tangentenraum des Punkts x auf einen Punkt auf der Mannigfaltigkeit ab, während der geodätische Abstand aufrechterhalten wird. Die inverse Operation wird logarithmische Abbildung $\text{Log}_x: M \rightarrow T_x M$ genannt. Eine andere nützliche Operation ist der

parallele Transport \parallel_x der Elemente zwischen Tangentenräumen bewegt, ohne eine Verzerrung einzuführen. Die exakte Form der vorstehend erwähnten Operationen hängt von der Riemannschen Metrik ab, die der Mannigfaltigkeit zugeordnet ist. Gemäß verschiedenen Ausführungsformen werden Riemannsche Mannigfaltigkeiten verwendet, um Statistiken über M unter Verwendung von Riemannschen Normalver-

teilungen korrekt zu berechnen, die die beobachteten Bewegungsmuster codieren und die Steuerhandlungen, die dem Aufgabenplan (z. B. sequenzierte Fähigkeiten) entsprechen, unter Verwendung einer optimalen Riemannschen Steuereinheit abrufen.

[0051] Für die folgenden Erläuterungen wird ein Roboterarm **101** mit mehreren Freiheitsgraden als Beispiel betrachtet, dessen Endeffektor **104** einen Zustand $x_e \in \mathbb{R}^3 \times S^3 \times \mathbb{R}^1$ aufweist (der die kartesische Position, das Orientierungsquaternion und den Greiferzustand beschreibt), der innerhalb eines statischen und bekannten Arbeitsraums arbeitet. Innerhalb der Reichweite des Arms **101** gibt es auch interessierende Objekte, die mit $O = \{o_1, o_2, \dots, o_j\}$ bezeichnet sind, von denen jedes einen Zustand $x_{o_j} \in \mathbb{R}^3 \times S^3$ aufweist. Der Einfachheit halber wird der gesamte Systemzustand mit $x = \{x_e, \{x_{o_j}, \forall o_j \in O\}\}$ bezeichnet

[0052] Innerhalb dieser Einrichtung führt ein menschlicher Benutzer mehrere kinästhetische Demonstrationen an dem Arm durch, um ein oder mehrere Objekte für bestimmte Manipulationsfähigkeiten zu manipulieren. Der Satz von demonstrierten Fähigkeiten soll mit $A = \{a_1, a_2, \dots, a_H\}$ bezeichnet werden. Überdies ist für die Fähigkeit $a \in A$ der Satz von Objekten, die beteiligt sind, durch O_a gegeben und der Satz von verfügbaren Demonstrationen wird mit D_a bezeichnet. Es sollte beachtet werden, dass alle Demonstrationen der auf Objekte fixierten Struktur folgen, die vorstehend eingeführt wurde, d. h. sie werden von mehreren Rahmen aufgezeichnet, die normalerweise den Objekten in O_a zugeordnet sind, die häufig die Objektlage im Arbeitsraum darstellen. Die Fähigkeit „Einsetzen des Zapfens in den Zylinder“ beinhaltet beispielsweise die Objekte „Zapfen“ und „Zylinder“ und die zugehörigen Demonstrationen werden sowohl von den Roboter-, den „Zapfen“- als auch den „Zylinder“-Rahmen aufgezeichnet.

[0053] Die (Manipulations-) Aufgabe, die im Folgenden betrachtet wird, besteht aus einer Sequenz von Fähigkeiten a^* , die aus den demonstrierten Fähigkeiten A ausgewählt sind. Eine Einsetzaufgabe beinhaltet beispielsweise „Aufgreifen der Kappe, Umorientieren der Kappe, erneutes Aufgreifen der Kappe und Einsetzen der Kappe“. Am Ende der Aufgabe ist eine Zielkonfiguration G als gewünschter Endzustand des Systems erreicht, einschließlich des Roboters und der Objekte.

[0054] Die übliche Weise zum Organisieren von Manipulationsaufgaben im Werk ist über ein Diagramm oder einen Ablaufplan. Sie werden üblicherweise über Ziehen und Ablegen in einer GUI (graphischen Benutzerschnittstelle) definiert. Eine solche Methode ist ausreichend, wenn zwei Bedingungen gelten: (1) Die Aufgabe ist einfach und als lineare Sequenz von Fähigkeiten festgelegt; (2) jede Fähigkeit ist einfach ohne Zweige. In dieser Weise kann jede Fähigkeit in der Sequenz ausgelöst und ausgeführt werden, wie festgelegt.

[0055] In vielen Fällen gilt jedoch eine der obigen Bedingungen nicht. Die gewünschte Aufgabe weist beispielsweise mehrere Wahlen der Ausführung in verschiedenen Arbeitsraumsituationen auf oder einige Fähigkeiten darin weisen mehrere Wahlen der Ausführung in verschiedenen Situationen auf.

[0056] Fig. 2 zeigt ein Ablaufdiagramm (oder Aufgabendiagramm) 200, das eine Manipulationsaufgabe mit Fähigkeiten in einer Sequenz und in Zweigen darstellt.

[0057] Die Manipulationsaufgabe umfasst beispielsweise als erste Tätigkeit **201**, ein Objekt aufzugreifen. Dies kann bedeuten, dass der Roboter die Fähigkeit „Aufgreifen eines Objekts von oben“ bei **202**, „Aufgreifen eines Objekts von links“ bei **203** oder „Aufgreifen eines Objekts von rechts“ bei **204** in Abhängigkeit von der anfänglichen Konfiguration (d. h. Zustand) des Objekts ausführen muss. Somit umfasst die Aufgabe die Ausführung dieser Fähigkeiten in Zweigen, d. h. sie sollen alternativ ausgeführt werden, d. h. nur eine von ihnen soll ausgeführt werden. Der ersten Operation **201**, d. h. der Ausführung von einer der Fähigkeiten bei **202**, bei **203** oder bei **204**, folgt eine oder mehrere Fähigkeiten in der Sequenz. Wenn beispielsweise die Fähigkeit „Aufgreifen eines Objekts von oben“ bei **202** ausgeführt wurde, folgt dieser (in der Sequenz) die Fähigkeit „Befestigen des Objekts“ bei **205**.

[0058] Wenn die jeweilige Fähigkeit bei **203** oder bei **204** ausgeführt wurde, muss diesem eine Umorientierungstätigkeit **206** folgen, d. h. für jeden Fall durch Ausführung einer Umorientierungsfähigkeit bei **207** oder bei **208**. Die Umorientierungsfähigkeiten können sich in der Umorientierungsrichtung unterscheiden. Der Umorientierungsoperation **206** folgt dann die Ausführung von „Aufgreifen des Objekts von oben“ bei der Fähigkeit **209** und schließlich die Ausführung der Fähigkeit „Befestigen des Objekts“ bei **210**.

[0059] Die Verzweigung kann durch manuelles Festlegen der Verzweigungsbedingungen **211**, **212**, **213** adressiert werden, üblicherweise als „falls“-Bedingungen, z. B. „falls das Objekt steht“ **211**, „falls das Objekt nach

links liegt“ **212** und „falls das Objekt nach rechts liegt“ **213**. Um solche Bedingungen zu entwerfen, kann ein Bereich von Systemzuständen manuell als Bereich von Gültigkeit gemessen werden, damit diese Bedingung gilt.

[0060] Dies bedeutet, dass der Roboter einen Satz von Manipulationsfähigkeiten vorinstalliert (vorprogrammiert vom Wert oder über Demonstration gelehrt) haben kann und für eine spezielle Montageaufgabe konstruiert der Bediener manuell ein Diagramm, das diese Aufgabe angibt (z. B. wie in **Fig. 2**), wobei die Bausteine der Satz von gelernten Fähigkeiten sind. Aufgrund der Zweige (möglicherweise sowohl auf der Aufgabenebene als auch auf der Fähigkeitsebene) muss der Bediener die Verzweigungsbedingungen für jeden Zweig manuell definieren.

[0061] Gemäß verschiedenen Ausführungsformen werden Methoden geschaffen, die insbesondere das Vermeiden der Notwendigkeit zum manuellen Definieren von Verzweigungsbedingungen ermöglichen.

[0062] **Fig. 3** zeigt ein Ablaufdiagramm, das ein Verfahren zum Steuern eines Roboters gemäß einer Ausführungsform darstellt.

[0063] Bei **301** werden Demonstrationen von Fähigkeiten durchgeführt.

[0064] Die Fähigkeiten umfassen zumindest diejenigen Fähigkeiten, die für die Ausführung einer Aufgabe erforderlich sind, die durch ein Aufgabendiagramm **303** gegeben ist.

[0065] Für eine demonstrierte Fähigkeit $a \in A$, wie vorstehend beschrieben, ist der Satz von verfügbaren Demonstrationen gegeben durch $D_a = \{\xi_t\}_{t=1}^N$, die in P Rahmen aufgezeichnet sind. Es sollte beachtet werden, dass solche Rahmen direkt an die Objekte in O_a gebunden sind.

[0066] Bei **302** wird ein Robotertrajektorienmodell (auch als „Roboterhaltensmodell“ bezeichnet) für jede Fähigkeit gelernt.

[0067] Wie vorstehend beschrieben, kann beispielsweise in Anbetracht einer korrekt gewählten Anzahl von Komponenten K das TP-HSMM-Modell Θ_a , das die räumlich-zeitlichen Merkmale von Trajektorien in Bezug auf eine Fähigkeit a abstrahiert, unter Verwendung eines Algorithmus nach Art von EM (Erwartung-Maximierung) gelernt werden.

[0068] Bei **304** wird ein zusammengesetztes Robotertrajektorienmodell aus den bei **302** gelernten Robotertrajektorienmodellen erzeugt.

[0069] Dafür umfasst das Lernen der Fähigkeitsmodi ferner das Lernen eines Vorbedingungsmodells, eines Endbedingungs- und eines Effektmodells für jede Fähigkeit. Bei **304** wird unter Verwendung dieser Modelle ein Zusammensetzungsmodell der festgelegten Aufgabe konstruiert und die Wahlen auf der Aufgabenebene und der Fähigkeitsebene können dann automatisch in Abhängigkeit von der Arbeitsraumsituation durchgeführt werden. Einfach gesagt kapselt das Vorbedingungsmodell ein, wie das System vor dem Ausführen der Fähigkeit sein sollte, während das Effektmodell und das Endbedingungsmodell einkapseln, wie das System nach dem Ausführen der Fähigkeit verändert sein sollte. Diese Modelle sind ein wichtiger Teil zum Berechnen des Zusammensetzungsmodells, da sie die Kompatibilität zwischen Fähigkeiten messen und die Entwicklung des Systemzustandes verfolgen. Es sollte beachtet werden, dass der Begriff „Fähigkeitsmodell“ als alle des Robotertrajektorienmodells, des Vorbedingungsmodells, des Endbedingungsmodells und des Effektmodells für die Fähigkeit umfassend verstanden werden kann.

[0070] Wie mit Bezug auf **Fig. 2** beschrieben, kann eine Aufgabe die Ausführung von Fähigkeiten in Zweigen (d. h. als Alternativen) und in einer Sequenz umfassen.

[0071] Folglich umfasst die Erzeugung des zusammengesetzten Modells die rekursive Anwendung von Kombinationsoperationen zum Kombinieren von Fähigkeiten in der Sequenz und einer Operation zum parallelen Kombinieren der Fähigkeiten.

[0072] **Fig. 4** stellt eine Kaskadierungsoperation zum Kaskadieren von Robotertrajektorienmodellen von Fähigkeiten **401**, **402**, **403**, die in einer Sequenz ausgeführt werden sollen, zu einem zusammengesetzten Robotertrajektorienmodell **404** dar.

[0073] Fig. 5 stellt eine Kombinationsoperation zum Kombinieren von Robotertrajektorienmodellen von Fähigkeiten **501**, **502**, **503**, **504**, die in Zweigen (d. h. alternativ) ausgeführt werden sollen, zu einem zusammengesetzten Robotertrajektorienmodell **505** dar.

[0074] Die Kombinationsoperation zum Kaskadieren einer Sequenz von Fähigkeiten, wie in Fig. 4 dargestellt, umfasst eine oder mehrere Anwendungen einer Operation zum Kaskadieren von zwei Fähigkeiten, die in einer Sequenz ausgeführt werden sollen. Ebenso umfasst eine Kombinationsoperation zum Kombinieren einer Sequenz von Fähigkeiten, wie in Fig. 5 dargestellt, eine oder mehrere Anwendungen einer Operation zum Kombinieren von zwei Fähigkeiten, die in Zweigen ausgeführt werden sollen.

[0075] Für die Kombination von zwei Fähigkeiten, die in einer Sequenz ausgeführt werden sollen, werden die Trajektorienmodelle der zwei Fähigkeiten folgendermaßen zu einem zusammengesetzten Trajektorienmodell kaskadiert.

[0076] In Anbetracht von zwei TP-HSMMs Θ_{a_1} und Θ_{a_2} von zwei Fähigkeiten in einer Sequenz ist die Operation zum Kaskadieren derselben zu Θ im Algorithmus 1 zusammengefasst 1.

Algorithmus 1: Kaskadieren eines Paares von TP-HSMMs

Eingabe: $(\Theta_{a_1}, \Gamma_{a_1})$ und $(\Theta_{a_2}, \Gamma_{a_2})$.

Ausgabe: $(\hat{\Theta}, \hat{\Gamma})$

1 **für alle** Endkomponente $k_f \in \Theta_{a_1}$ **führe durch**

2 Erzeuge Kopie von Θ_{a_2} als $\Theta_{a_2}^{k_f}$.

3 Berechne $\{a_{k_f, k_i}\}$ für alle anfänglichen $k_i \in \Theta_{a_2}^{k_f}$

4 Aktualisiere $\Theta_{a_2}^{k_f}$ und $\Gamma_{1T, a_2}^{k_f}$

5 Kaskadiere Θ_{a_1} und $\Theta_{a_2}^{k_f}$. Füge hinzu zu $\hat{\Theta}$.

6 Setze zusätzliche Parameter von $\hat{\Theta}$.

7 $\hat{\Gamma} = \{\hat{\Gamma}_1, \hat{\Gamma}_T, \hat{\Gamma}_{1T}\} = \{\Gamma_{1, a_1}, \Gamma_{T, a_2}, \{\Gamma_{1T, a_2}^{k_f}, \forall k_f\}\}$.

[0077] Es sollte beachtet werden, dass die Berechnung und die Aktualisierung der Zeilen 3 und 4 des Algorithmus 1 gemäß den Gleichungen (4) bzw. (5) durchgeführt werden können, die nachstehend gegeben sind.

[0078] Eine Schlüsselerkenntnis kann darin gesehen werden, dass dasselbe Modell Θ_{a_2} in Abhängigkeit von der Endkomponente (d. h. HSMM-Zustand) von Θ_{a_1} , mit der Θ_{a_2} kaskadiert wird, unterschiedlich aktualisiert wird. Dies liegt daran, dass jede Endkomponente unterschiedliche Transformationen der Aufgabenparameter von Θ_{a_1} nach dem Ausführen von a_1 codiert, was wiederum zu verschiedenen Weisen führt, die Komponenten in Θ_{a_2} zu aktualisieren. Folglich weist das zusammengesetzte Modell Θ die Größe $K_1 + K_{1,f} \cdot K_2$ auf, wobei K_1 und K_2 die Anzahl von Komponenten von Θ_{a_1} bzw. Θ_{a_2} sind, während $K_{1,f}$ die Anzahl von Endkomponenten in Θ_{a_1} ist. Insbesondere besteht der Algorithmus 2 aus zwei Hauptoperationen: (a) Berechnen der Übergangswahrscheinlichkeit von jeder Endkomponente in Θ_{a_1} zu jeder anfänglichen Komponente in Θ_{a_2} ; (b) Modifizieren aller Komponenten von Θ_{a_2} für jede Endkomponente in Θ_{a_1} , mit der Θ_{a_2} kaskadiert wird.

[0079] Gemäß einer Ausführungsform werden ein Vorbedingungsmodell und ein Effektmodell, wie in Referenz [1] beschrieben, verwendet. Insbesondere enthält das gelernte Vorbedingungsmodell, das mit $\Gamma_{1,a}$ bezeichnet ist, TP-GMMs für den anfänglichen Roboterzustand (d. h. die anfängliche Konfiguration (z. B. Position und/oder Lage) des Roboters), d. h.

$\Gamma_{1,a} = \left\{ \left(\hat{\mu}_1^{(p)}, \hat{\Sigma}_1^{(p)} \right), \forall p \in P_{1,a} \right\}$, wobei $P_{1,a}$ der gewählte Satz von Aufgabenparametern ist, die vom anfänglichen Systemzustand (d. h. der anfänglichen Konfiguration (z. B. Position und/oder Lage) des Roboters und/oder der Objekte) abgeleitet sind. Außerdem wird hier ein Endbedingungsmodell eingeführt, das mit $\Gamma_{T,a}$ bezeichnet ist, das in einer ähnlichen Weise wie $\Gamma_{1,a}$ gelernt wird, aber

für den Endroboterzustand, d. h. $\Gamma_{T,a} = \left\{ \left(\hat{\mu}_T^{(p)}, \hat{\Sigma}_T^{(p)} \right), \forall p \in P_{T,a} \right\}$, wobei $P_{T,a}$ der gewählte Satz von Rahmen ist, die vom Endsystemzustand abgeleitet sind. Einfach gesagt modelliert $\Gamma_{1,a}$ die anfängliche Konfiguration vor dem Ausführen der Fähigkeit a , während $\Gamma_{T,a}$ die Endkonfiguration danach modelliert. Ferner enthält das gelernte Effektmodell, das mit $\Gamma_{1T,a}$ bezeichnet ist, TP-GMMs für den vorhergesagten Endsystemzustand, d. h.

$\Gamma_{1T,a} = \left\{ \left\{ \left(\hat{\mu}_{1,o}^{(p)}, \hat{\Sigma}_{1,o}^{(p)} \right), \forall p \in P_{1,a} \right\}, \forall o \in O_a \cup e \right\}$, wobei $P_{1,a}$ in $\Gamma_{1,a}$ definiert ist. Die Unterschiede unter diesen drei Modellen sind erwähnenswert: die Aufgabenparameter für $\Gamma_{T,a}$ werden aus dem Endsystemzustand (nach dem Durchführen von a) berechnet, während jene für $\Gamma_{1,a}$ und $\Gamma_{1T,a}$ aus dem anfänglichen Systemzustand extrahiert werden (vor dem Durchführen von a). Wegen der Schreibweise $\Gamma_a \triangleq \{ \Gamma_{1,a}, \Gamma_{T,a}, \Gamma_{1T,a} \}$.

[0080] Dann ist die Übergangswahrscheinlichkeit von einer Endkomponente k_f von Θ_{a_1} zu einer anfänglichen Komponente k_i von Θ_{a_2} :

$$a_{k_f, k_i} \propto \exp \left(- \sum_{p \in P_c} K L \left(\Gamma_{T,a_1}^{(p)}(k_f) \parallel \Gamma_{1,a_2}^{(p)}(k_i) \right) \right), \quad (4)$$

wobei $KL(\cdot \parallel \cdot)$ die KL - Divergenz (Kullback-Leibler-Divergenz) ist, $\Gamma_{T,a_1}^{(p)}(k_f)$ das GMM ist, das der Komponente k_f für den Rahmen p zugeordnet ist, $\Gamma_{1,a_2}^{(p)}(k_i)$ das GMM ist, das der Komponente k_i für den Rahmen p zugeordnet ist; $P_c = P_{T,a_1} \cap P_{1,a_2}$ der Satz von gemeinsamen Rahmen ist, die durch diese zwei Modelle geteilt werden, die auf nicht leer gezwungen werden können, indem immer der globale Rahmen hinzugefügt wird. Dieser Prozess wird für alle Paare von Endkomponenten in Θ_{a_1} und anfänglichen Komponenten in Θ_{a_2} wiederholt. Es sollte beachtet werden, dass die Ausgangswahrscheinlichkeit von irgendeiner Endkomponente in 19" normiert werden sollte.

[0081] In Anbetracht einer Endkomponente k_f von Θ_{a_1} sollte zweitens jede Komponente k von Θ_{a_2} affin transformiert werden, wie folgt:

$$\left(\hat{\mu}_k^{(\hat{p})}, \hat{\Sigma}_k^{(\hat{p})} \right) \triangleq \left(\mu_k^{(p)}, \Sigma_k^{(p)} \right) \otimes \left(\mathbf{b}_{k_f}^{(\hat{p})}, \mathbf{A}_{k_f}^{(\hat{p})} \right), \quad (5)$$

wobei die Operation \otimes als gleiche Operation von (1) definiert ist; $\left(\mathbf{b}_{k_f}^{(\hat{p})}, \mathbf{A}_{k_f}^{(\hat{p})} \right)$ der Aufgabenparameter ist, der aus dem Mittelwert von $\Gamma_{1T,a_1}^{(\hat{p}),o}(k_f)$ berechnet wird, wobei o das Objekt ist, das dem alten Rahmen \hat{p} in Θ_{a_1} zugeordnet ist, und \hat{p} der neue Rahmen in $\Gamma_{1T,a_1}^o(k_f)$ ist. Es sollte beachtet werden, dass die Änderung von Rahmen wichtig ist, um direkt alle Komponenten von Θ_{a_2} in Anbetracht eines anfänglichen Systemzustandes von Θ_{a_1} zu berechnen. Derselbe Prozess wird auch auf jede Komponente von Γ_{1T,a_2} , durch Ändern ihrer Rahmen auf der Basis von $\Gamma_{1T,a_1}^o(k_f)$ angewendet.

[0082] Schließlich werden, wie im Algorithmus 1 angegeben, andere Modell-Parameter von Θ wie z. B. Dauerwahrscheinlichkeiten, anfängliche und Endverteilungen trivial mit geringfügigen Änderungen von Θ_{a_1} und Θ_{a_2} festgelegt. Die Dauerwahrscheinlichkeit von Θ_{a_2} wird beispielsweise auf k_f mehrfache Kopien dupliziert; die anfänglichen Verteilungen Θ_{a_2} werden auf null gesetzt, da die anfänglichen Zustände von Θ jenen des ersten Modells Θ_{a_1} entsprechen; die Endkomponenten von Θ_{a_1} werden entfernt, da die Endzustände von Θ nun die Endkomponenten von Θ_{a_2} sind, die auf ihre mehreren Instanzen aktualisiert sind.

[0083] Für die Kombination von zwei Fähigkeiten, die parallel ausgeführt werden sollen, werden die Trajektorienmodelle der zwei Fähigkeiten zu einem zusammengesetzten Trajektorienmodell wie folgt kombiniert.

[0084] Zwei TP-HSMMs Θ_{a_1} und Θ_{a_2} von zwei Fähigkeiten sollen parallel betrachtet werden, die Operation für das Kombinieren derselben zu Θ ist im Algorithmus 2 zusammengefasst.

Algorithmus 2: Paralleles Zusammensetzen eines Paares von TP-HSMMs

Eingabe: $(\Theta_{a_1}, \Gamma_{a_1})$ und $(\Theta_{a_2}, \Gamma_{a_2})$.

Ausgabe: $(\hat{\Theta}, \hat{\Gamma})$

- 1 Indiziere neu alle Komponenten von Θ_2 durch Inkrementieren von K_1 .
/* Berechne TPHSMM $\hat{\Theta}$ */
 - 2 Kopiere alle Komponenten von Θ_2 und ihre Dauerverteilungen in $\hat{\Theta}$.
 - 3 Füge $\{a_{kh}\}_1$ von Θ_1 und $\{a_{kh}\}_2$ von Θ_2 diagonal zu $\{a_{kh}\}$ von $\hat{\Theta}$ hinzu.
 - 4 Fülle andere Einträge in $\{a_{kh}\}$ von $\hat{\Theta}$ mit 0.
/* Berechne die Vorbedingung und Effekte TPGMM $\hat{\Gamma}$ */
 - 5 Kopiere Γ_1 in $\hat{\Gamma}$.
 - 6 Kopiere Γ_2 (mit dem Namen der aktualisierten Komponenten) in $\hat{\Gamma}$.
-

[0085] Der Algorithmus 2 besteht aus zwei Teilen: einem Teil, um das zusammengesetzte TP-HSMM-Modell Θ zu berechnen und einem anderen Teil, um das zusammengesetzte TPGMM-Modell $\hat{\Gamma}$ zu berechnen. Der erste und wichtigste Schritt besteht darin, Komponentenindizes von Θ_2 durch die Gesamtzahl von Komponenten in Θ_1 zu aktualisieren. Dies soll mehrere Komponenten von verschiedenen Fähigkeiten mit demselben Index vermeiden. Danach müssen alle des zugehörigen TPGMM-Modells, Dauermodells, Vorbedingungs- und Effektmotells dementsprechend aktualisiert werden. Zu guter Letzt müssen wir, wenn die zusammengesetzte Übergangsmatrix $\{a_{kh}\}$ berechnet wird, $\{a_{kh}\}_1$ von Θ_1 und $\{a_{kh}\}_2$ von Θ_2 diagonal zu $\{a_{kh}\}$ hinzufügen, während die restlichen Einträge mit null gefüllt werden. Dies bedeutet, dass keine zusätzlichen Übergänge von Θ_1 zu Θ_2 hinzugefügt werden, da sie parallel zusammengesetzt werden (d. h. nicht in einer Sequenz).

[0086] Zusammengefasst umfasst das Erzeugen des zusammengesetzten Modells bei **304** die wiederholte Anwendung der folgenden Operationen von zwei Fähigkeiten:

- 1) Operation 1 (Kaskadieren von Fähigkeiten in einer Sequenz): falls „Fähigkeit#1“ und „Fähigkeit#2“ in einer Sequenz verbunden sind (wie in **Fig. 4** dargestellt), Berechnen eines zusammengesetzten Fähigkeitsmodells gemäß dem Algorithmus 1
- 2) Operation 2 (Kombinieren von Fähigkeiten in Zweigen): falls „Fähigkeit#1“ und „Fähigkeit#2“ parallel verbunden sind (wie in **Fig. 5** dargestellt), Berechnen des zusammengesetzten Fähigkeitsmodells gemäß dem Algorithmus 2.

[0087] Insbesondere werden diese zwei Operationen wiederholt wie folgt durchgeführt:

- A) Für jeden Zweig oder Unterzweig innerhalb des Aufgabendiagramms **303** rekursives Anwenden der Operation 1, um die zusammengesetzte Fähigkeit für jeden Zweig abzuleiten.
- B) Für alle parallelen Zweige rekursives Anwenden der Operation 2, um die zusammengesetzte Fähigkeit für alle Zweige abzuleiten. Es ist zu beachten, dass nach A jeder Zweig nur eine zusammengesetzte Fähigkeit aufweisen sollte.
- C) Rekursives Anwenden von A auf alle Sequenzen von Fähigkeiten und von B auf alle Parallelen von Fähigkeiten.

[0088] Für die in **Fig. 4** dargestellte Aufgabe umfasst beispielsweise der Prozess für die Erzeugung des zusammengesetzten Modells eine Anwendung von A, dann B, dann erneut A und dann erneut B, wie in **Fig. 6** bis **Fig. 9** dargestellt.

[0089] **Fig. 6** zeigt das Ergebnis nach einer ersten Anwendung von A auf die Aufgabe, wie durch das Aufgabendiagramm von **Fig. 2** gegeben.

[0090] Die Fähigkeitsmodelle **201** und **205** müssen zu **601** kombiniert werden, die Fähigkeitsmodelle **203**, **207** müssen zu **602** kombiniert werden und die Fähigkeitsmodelle **204**, **208** müssen zum Fähigkeitsmodell **603** kombiniert werden.

[0091] **Fig. 7** zeigt das Ergebnis nach einer ersten Anwendung von B. Die Fähigkeitsmodelle **602** und **603** müssen zum Fähigkeitsmodell **701** kombiniert werden.

[0092] **Fig. 8** zeigt das Ergebnis nach einer zweiten Anwendung von B. Die Fähigkeitsmodelle **701**, **209** und **210** wurden zum Fähigkeitsmodell **801** kombiniert.

[0093] **Fig. 9** zeigt das Ergebnis nach einer zweiten Anwendung von B. Die Fähigkeitsmodelle **601** und **801** wurden zum Fähigkeitsmodell **901** kombiniert.

[0094] Wie in **Fig. 9** zu sehen ist, ist die ganze Aufgabe durch ein einzelnes zusammengesetztes Fähigkeitsmodell **901** dargestellt. Keine „falls“-Bedingungen sind erforderlich.

[0095] Wenn das zusammengesetzte Robotertrajektorienmodell bei 304 erzeugt wurde, kann eine Aufgabe in einer gegebenen Situation ausgeführt werden.

[0096] Dafür wird der anfängliche Systemzustand (Konfiguration) in der gegebenen Situation bei **305** beobachtet und z. B. durch Anwenden von Gleichung (3) auf das zusammengesetzte Robotertrajektorienmodell wird die wahrscheinlichste Sequenz von Komponenten innerhalb des zusammengesetzten Robotertrajektorienmodells bestimmt, die das System (einschließlich des Roboters und der Objekte) auf den Zielzustand mit der höchsten Wahrscheinlichkeit treibt.

[0097] Die Bestimmung der Sequenz von Komponenten gibt auch die aktuelle Sequenz von Fähigkeiten aus, die unter der gegebenen Situation ausgeführt werden müssen. Dies ist von Bedeutung, da die Sequenz von Fähigkeiten unterschiedlich ist, wenn verschiedene Zweige durch den Algorithmus gewählt werden.

[0098] Bei **306** wird während der Ausführung die optimale Sequenz von Fähigkeiten, die in dieser Ausgabe enthalten sind, ausgeführt, indem der optimalen Sequenz von Komponenten gefolgt wird. In Anbetracht der Zustandssequenz kann beispielsweise eine lineare quadratische Verfolgung (LQT) verwendet werden, um die optimale Trajektorie abzurufen.

[0099] Zusammengefasst wird gemäß verschiedenen Ausführungsformen ein Verfahren geschaffen, wie in **Fig. 10** dargestellt.

[0100] **Fig. 10** zeigt ein Ablaufdiagramm **1000**, das ein Verfahren zum Steuern eines Roboters gemäß einer Ausführungsform darstellt.

[0101] Bei **1001** werden Demonstrationen zum Durchführen von jeder von mehreren Fähigkeiten bereitgestellt.

[0102] Bei **1002** wird ein Robotertrajektorienmodell für jede Fähigkeit aus den Demonstrationen trainiert, wobei jedes Trajektorienmodell ein verborgenes Semi-Markov-Modell mit einem oder mehreren anfänglichen Zuständen und einem oder mehreren Endzuständen ist.

[0103] Bei **1003** werden ein Vorbedingungsmodell mit einer Wahrscheinlichkeitsverteilung von Roboterkonfigurationen vor dem Ausführen der Fähigkeit für jeden anfänglichen Zustand des Robotertrajektorienmodells der Fähigkeit und ein Endbedingungsmodell mit einer Wahrscheinlichkeitsverteilung von Roboterkonfigurationen nach dem Ausführen der Fähigkeit für jeden Endzustand des Robotertrajektorienmodells der Fähigkeit aus den Demonstrationen für jede Fähigkeit trainiert.

[0104] Bei **1004** wird eine Beschreibung einer Aufgabe empfangen, wobei die Aufgabe das Durchführen der Fähigkeiten der mehreren Fähigkeiten in einer Sequenz und/oder in Zweigen umfasst.

[0105] Bei **1005** wird ein zusammengesetztes Trajektorienmodell erzeugt durch

- wenn zwei Fähigkeiten in einer Sequenz in der Aufgabe durchgeführt werden sollen, Kaskadieren der Robotertrajektorienmodelle der Fähigkeiten durch

- Aufnehmen der Zustände der Trajektorienmodelle der zwei Fähigkeiten in das zusammengesetzte Robotertrajektorienmodell und

- Berechnen einer Übergangswahrscheinlichkeit zwischen jedem Endzustand des Trajektorienmodells der ersten Fähigkeit der zwei Fähigkeiten und jedem anfänglichen Zustand des Trajektorienmodells der zweiten Fähigkeit der zwei Fähigkeiten als Funktion der Ähnlichkeit zwischen der Wahrscheinlichkeitsverteilung des Endbedingungsmodells der ersten Fähigkeit für den Endzustand der ersten Fähigkeit und der Wahrscheinlichkeitsverteilung des anfänglichen Modells der zweiten Fähigkeit für den anfänglichen Zustand für die zweite Fähigkeit

- wenn zwei Fähigkeiten in Zweigen in der Aufgabe durchgeführt werden, Kombinieren der Robotertrajektorienmodelle der Fähigkeiten durch

- Aufnehmen der Zustände der Trajektorienmodelle der zwei Fähigkeiten im zusammengesetzten Robotertrajektorienmodell und

- Setzen der Übergangswahrscheinlichkeit zwischen Zuständen der ersten Fähigkeit und Zuständen der zweiten Fähigkeit auf null.

[0106] Bei **1006** wird der Roboter gemäß dem zusammengesetzten Robotertrajektorienmodell gesteuert, um die Aufgabe auszuführen.

[0107] Gemäß verschiedenen Ausführungsformen werden mit anderen Worten Modelle für einen Roboter für mehrere Fähigkeiten trainiert, und wenn eine Aufgabe ausgeführt werden soll, die mehrere Ausführungen dieser Fähigkeiten in Zweigen oder in einer Sequenz beinhaltet, werden die Modelle zu einem zusammengesetzten Modell kaskadiert und/oder kombiniert. Das zusammengesetzte Modell kann dann zum Steuern des Roboters verwendet werden, als ob es ein Modell für eine einzelne Fähigkeit wäre, d. h. beispielsweise durch Bestimmen einer optimalen Zustandssequenz für die Aufgabe (und der anfänglichen Konfiguration von Roboter und Objekten, an denen die Aufgabe ausgeführt werden soll) und dementsprechendes Steuern des Roboters.

[0108] Das Verfahren von **Fig. 10** kann durch einen oder mehrere Computer mit einer oder mehreren Datenverarbeitungseinheiten durchgeführt werden. Der Begriff „Datenverarbeitungseinheit“ kann als irgendein Typ von Entität verstanden werden, die die Verarbeitung von Daten oder Signalen ermöglicht. Die Daten oder Signale können beispielsweise gemäß mindestens einer (d. h. einer oder mehr als einer) speziellen Funktion behandelt werden, die durch die Datenverarbeitungseinheit durchgeführt wird. Eine Datenverarbeitungseinheit kann eine analoge Schaltung, eine digitale Schaltung, eine Verbundsignalschaltung, eine Logikschaltung, einen Mikroprozessor, einen Mikrocontroller, eine Zentraleinheit (CPU), eine Graphikverarbeitungseinheit (GPU), einen Digitalsignalprozessor (DSP), eine integrierte Schaltung einer programmierbaren Gatteranordnung (FPGA) oder irgendeine Kombination davon umfassen oder aus dieser ausgebildet sein. Irgendeine andere Weise zum Implementieren der jeweiligen Funktionen, die nachstehend genauer beschrieben werden, kann auch als Datenverarbeitungseinheit oder Logikschaltungsanordnung verstanden werden. Selbstverständlich können ein oder mehrere der im Einzelnen hier beschriebenen Verfahrensschritte durch eine Datenverarbeitungseinheit durch eine oder mehrere spezielle Funktionen ausgeführt (z. B. implementiert) werden, die durch die Datenverarbeitungseinheit durchgeführt werden.

[0109] Der Begriff „Roboter“ kann als sich auf irgendein physikalisches System (mit einem mechanischen Teil, dessen Bewegung gesteuert wird) beziehend verstanden werden, wie z. B. eine computergesteuerte Maschine, ein Fahrzeug, ein Haushaltsgerät, ein Elektrowerkzeug, eine Fertigungsmaschine, einen persönlichen Assistenten oder ein Zugangssteuersystem.

[0110] Verschiedene Ausführungsformen können Sensorsignale von verschiedenen Sensoren wie z. B. Video, Radar, LiDAR, Ultraschall, Bewegung, Wärmeabbildung usw. empfangen und verwenden, beispielsweise um Sensordaten hinsichtlich Demonstrationen oder Zuständen des Systems (Roboter und Objekt) und Konfigurationen und Szenarios zu erhalten. Die Sensordaten können verarbeitet werden. Dies kann die Klassifikation der Sensordaten oder das Durchführen einer semantischen Segmentierung an den Sensordaten umfassen, beispielsweise um die Anwesenheit von Objekten (in der Umgebung, in der die Sensordaten erhalten wurden) zu detektieren. Ausführungsformen können zum Trainieren eines Maschinenlernsystems und Steuern eines Roboters, z. B. autonom von Roboter-Manipulatoren, um verschiedene Manipulationsaufgaben unter verschiedenen Szenarios zu erreichen, verwendet werden. Insbesondere sind Ausführungsformen auf die Steuerung

und Überwachung der Ausführung von Manipulationsaufgaben anwendbar, z. B. in Montagelinien. Sie können beispielsweise nahtlos mit einer herkömmlichen GUI für einen Steuerprozess integriert werden.

[0111] Obwohl spezielle Ausführungsformen hier dargestellt und beschrieben wurden, wird vom Fachmann auf dem Gebiet erkannt, dass die speziellen Ausführungsformen, die gezeigt und beschrieben sind, gegen eine Vielfalt von alternativen und/oder äquivalenten Implementierungen ausgetauscht werden können, ohne vom Schutzbereich der vorliegenden Erfindung abzuweichen. Diese Anmeldung soll irgendwelche Anpassungen oder Variationen der speziellen Ausführungsformen abdecken, die hier erörtert sind. Daher ist beabsichtigt, dass diese Erfindung nur durch die Ansprüche und die Äquivalente davon begrenzt ist.

Patentansprüche

1. Verfahren zum Steuern eines Roboters, das Folgendes umfasst:

Bereitstellen von Demonstrationen zum Durchführen von jeder von mehreren Fähigkeiten;

Trainieren eines Robotertrajektorienmodells für jede Fähigkeit aus den Demonstrationen, wobei jedes Trajektorienmodell ein verborgenes Semi-Markov-Modell mit einem oder mehreren anfänglichen Zuständen und einem oder mehreren Endzuständen ist;

Trainieren eines Vorbedingungsmodells für jede Fähigkeit aus den Demonstrationen, das für jeden anfänglichen Zustand des Robotertrajektorienmodells der Fähigkeit eine Wahrscheinlichkeitsverteilung von Roboterkonfigurationen vor dem Ausführen der Fähigkeit umfasst, und

eines Endbedingungsmodells für jede Fähigkeit, das für jeden Endzustand des Robotertrajektorienmodells der Fähigkeit eine Wahrscheinlichkeitsverteilung von Roboterkonfigurationen nach dem Ausführen der Fähigkeit umfasst;

Empfangen einer Beschreibung einer Aufgabe, wobei die Aufgabe das Durchführen der Fähigkeiten der mehreren Fähigkeiten in einer Sequenz und/oder in Zweigen umfasst;

Erzeugen eines zusammengesetzten Robotertrajektorienmodells durch

- wenn zwei Fähigkeiten in einer Sequenz in der Aufgabe durchgeführt werden sollen, Kaskadieren der Robotertrajektorienmodelle der Fähigkeiten durch

○Aufnehmen der Zustände der Trajektorienmodelle der zwei Fähigkeiten in das zusammengesetzte Robotertrajektorienmodell und

○Berechnen einer Übergangswahrscheinlichkeit zwischen jedem Endzustand des Trajektorienmodells der ersten Fähigkeit der zwei Fähigkeiten und jedem anfänglichen Zustand des Trajektorienmodells der zweiten Fähigkeit der zwei Fähigkeiten als Funktion der Ähnlichkeit zwischen der Wahrscheinlichkeitsverteilung des Endbedingungsmodells der ersten Fähigkeit für den ersten Zustand der ersten Fähigkeit und der Wahrscheinlichkeitsverteilung des anfänglichen Modells der zweiten Fähigkeit für den anfänglichen Zustand für die zweite Fähigkeit

- wenn zwei Fähigkeiten in Zweigen in der Aufgabe durchgeführt werden, Kombinieren der Robotertrajektorienmodelle der Fähigkeiten durch

○ Aufnehmen der Zustände der Trajektorienmodelle der zwei Fähigkeiten in das zusammengesetzte Robotertrajektorienmodell und

○ Setzen der Übergangswahrscheinlichkeit zwischen Zuständen der ersten Fähigkeit und Zuständen der zweiten Fähigkeit auf null; und Steuern des Roboters gemäß dem zusammengesetzten Robotertrajektorienmodell, um die Aufgabe auszuführen.

2. Verfahren nach Anspruch 1, wobei jede Fähigkeit eine Manipulation von einem oder mehreren Objekten durch den Roboter umfasst und das verborgene Semi-Markov-Modell jeder Fähigkeit aufgabenparametrisierte verborgene Semi-Markov-Modell mit Aufgabenparametern ist, die einer Roboter- und Objektkonfiguration entsprechen, in der die Fähigkeit angewendet wird.

3. Verfahren nach Anspruch 2, das ferner das Trainieren eines Effektmodells für jede Fähigkeit umfasst, das für jeden Endzustand des Robotertrajektorienmodells der Fähigkeit eine Wahrscheinlichkeitsverteilung dessen, wie die Roboter- und/oder Objektkonfiguration nach dem Ausführen der Fähigkeit relativ zu einer anfänglichen Roboter- und/oder Objektkonfiguration verändert ist, auf die die Fähigkeit angewendet wird, umfasst, wobei, wenn zwei Fähigkeiten in einer Sequenz in der Aufgabe durchgeführt werden sollen, das Kaskadieren der Robotertrajektorienmodelle der Fähigkeiten das Transformieren des aufgabenparametrisierten verborgenen Semi-Markov-Modells der zweiten Fähigkeit umfasst, so dass seine Aufgabenparameter Aufgabenparameter sind, die der Roboter- und/oder Objektkonfiguration entsprechen, die durch das Effektmodell der ersten Fähigkeit gegeben ist.

4. Verfahren nach Anspruch 3, das ferner das affine Transformieren des Effektmodells der zweiten Fähigkeit umfasst, so dass seine Aufgabenparameter die Aufgabenparameter sind, die der Roboter- und/oder Objektkonfiguration entsprechen, die durch das Effektmodell der ersten Fähigkeit gegeben ist.

5. Verfahren nach einem der Ansprüche 1 bis 4, das ferner, wenn zwei Fähigkeiten in einer Sequenz in der Aufgabe durchgeführt werden sollen, die Verwendung des Vorbedingungsmodells der ersten Fähigkeit als Vorbedingungsmodell für die kaskadierten Robotertrajektorienmodelle der Fähigkeiten der zwei Fähigkeiten und die Verwendung des Endbedingungsmodells der zweiten Fähigkeit als Endbedingungsmodell für die kaskadierten Robotertrajektorienmodelle der Fähigkeiten umfasst.

6. Verfahren nach einem der Ansprüche 1 bis 5, das ferner, wenn zwei Fähigkeiten in einem Zweig in der Aufgabe durchgeführt werden sollen, das Aufnehmen sowohl des Vorbedingungsmodells für die erste Fähigkeit als auch des Vorbedingungsmodells für die zweite Fähigkeit in das Vorbedingungsmodell der kombinierten Robotertrajektorienmodelle der Fähigkeiten und das Aufnehmen sowohl des Endbedingungsmodells für die erste Fähigkeit als auch des Endbedingungsmodells für die zweite Fähigkeit in das Vorbedingungsmodell der kombinierten Robotertrajektorienmodelle der Fähigkeiten umfasst.

7. Verfahren nach einem der Ansprüche 1 bis 6, wobei die Ähnlichkeit zwischen der Wahrscheinlichkeitsverteilung des Endbedingungsmodells der ersten Fähigkeit für den Endzustand der ersten Fähigkeit und der Wahrscheinlichkeitsverteilung des anfänglichen Modells der zweiten Fähigkeit für den anfänglichen Zustand für die zweite Fähigkeit eine KL-Divergenz der Wahrscheinlichkeitsverteilung des Endbedingungsmodells der ersten Fähigkeit für den Endzustand der ersten Fähigkeit und der Wahrscheinlichkeitsverteilung des anfänglichen Modells der zweiten Fähigkeit für den anfänglichen Zustand für die zweite Fähigkeit ist.

8. Verfahren nach einem der Ansprüche 1 bis 7, wobei das Vorbedingungsmodell und das Endbedingungsmodell aufgabenparametrisierte Gauß-Mischmodelle sind.

9. Verfahren nach einem der Ansprüche 1 bis 8, wobei das Erzeugen des zusammengesetzten Robotertrajektorienmodells das wiederholte Kaskadieren von Robotertrajektorienmodellen, kaskadierten Robotertrajektorienmodellen und kombinierten Robotertrajektorienmodellen und das Kombinieren von Robotertrajektorienmodellen, kaskadierten Robotertrajektorienmodellen und einer kombinierten Robotertrajektorie gemäß der Aufgabe umfasst, so dass das Robotertrajektorienmodell ein Trajektorienmodell für die ganze Aufgabe ist.

10. Robotersteuereinheit, die dazu konfiguriert ist, das Verfahren nach einem der Ansprüche 1 bis 9 durchzuführen.

11. Computerprogramm mit Befehlen, die, wenn sie durch einen Prozessor ausgeführt werden, bewirken, dass der Prozessor ein Verfahren nach einem der Ansprüche 1 bis 9 durchführt.

12. Computerlesbares Medium, das Befehle speichert, die, wenn sie durch einen Prozessor ausgeführt werden, bewirken, dass der Prozessor ein Verfahren nach einem der Ansprüche 1 bis 9 durchführt.

Es folgen 6 Seiten Zeichnungen

Anhängende Zeichnungen

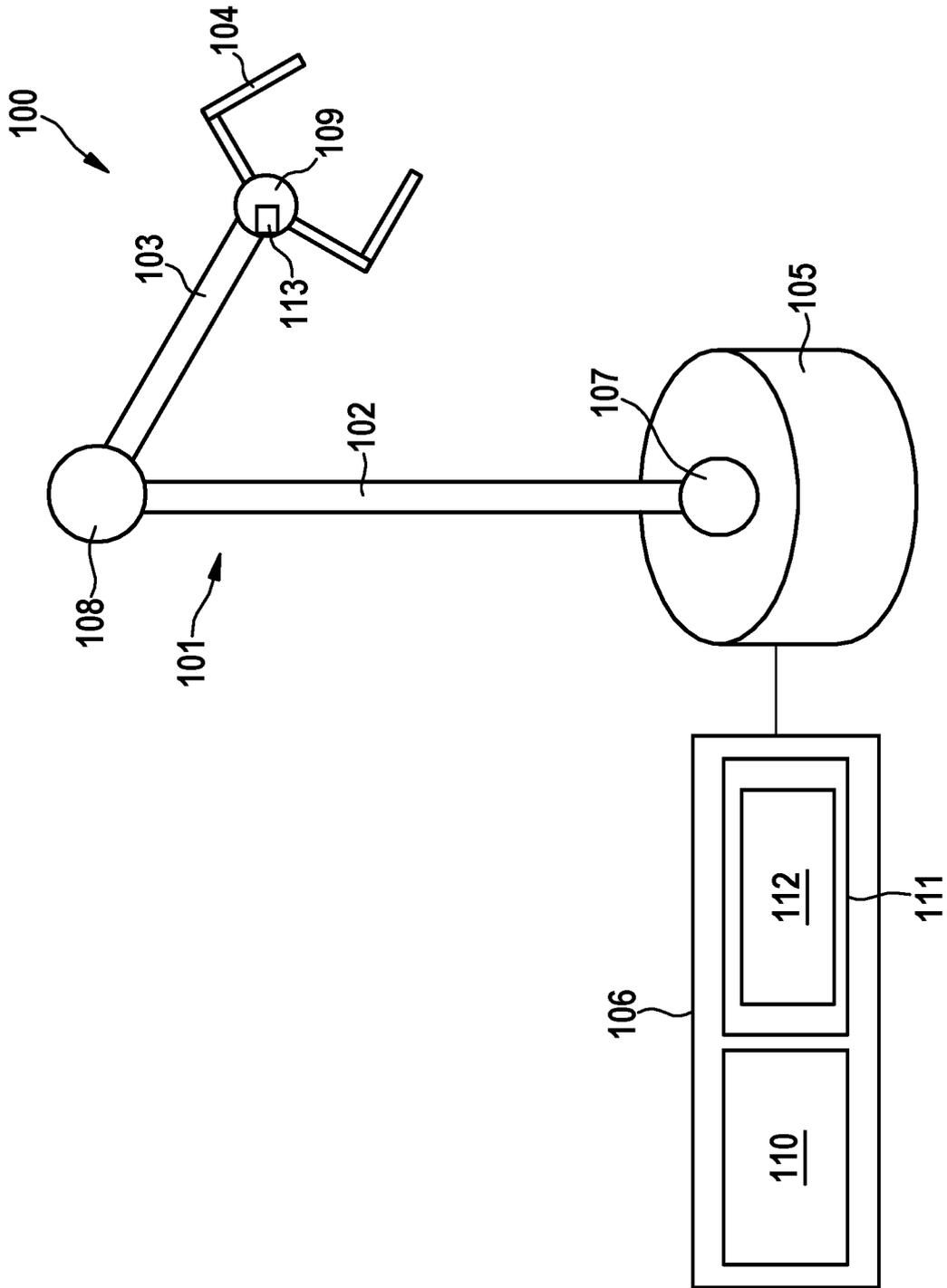


Fig. 1

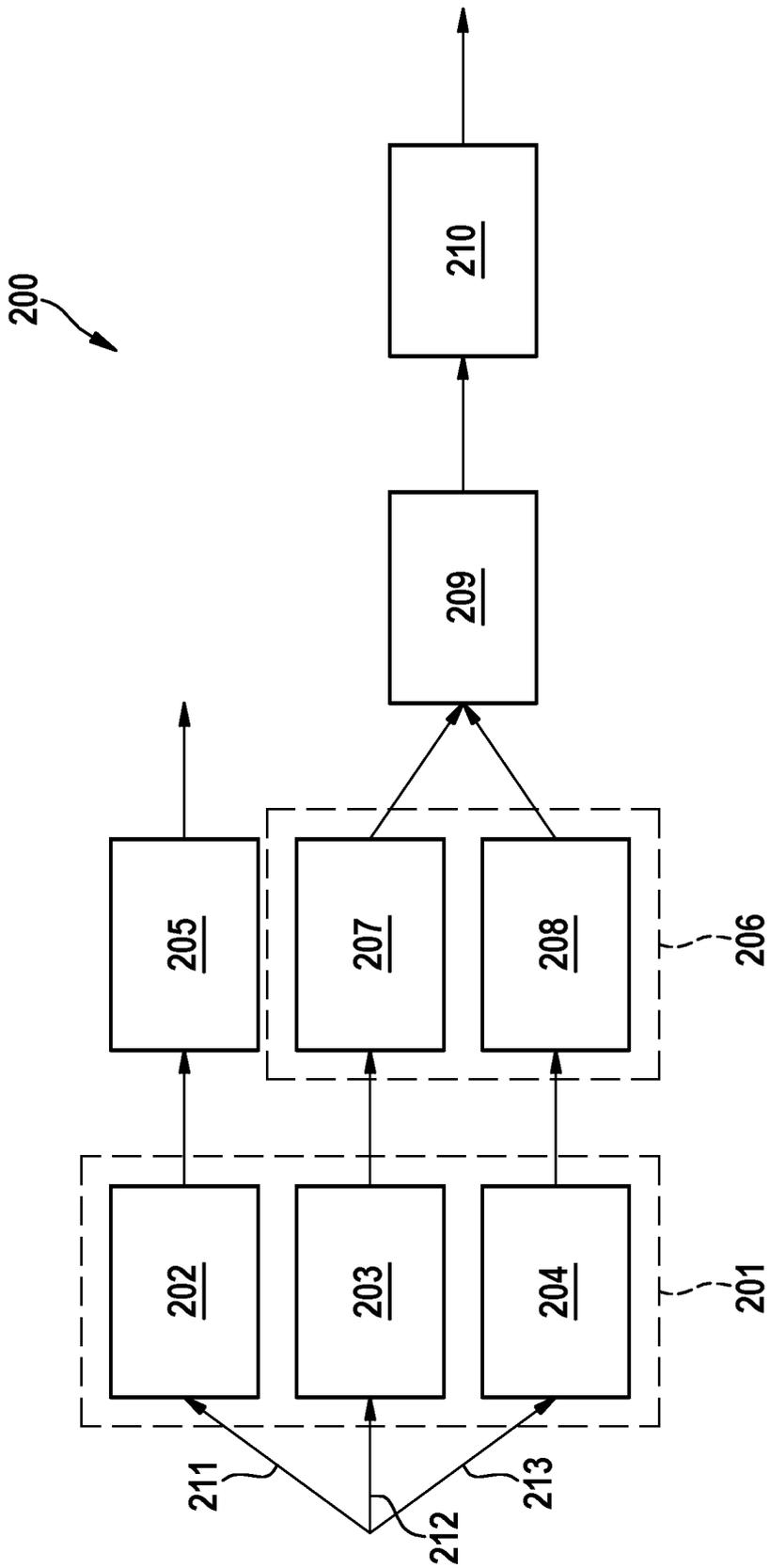


Fig. 2

300

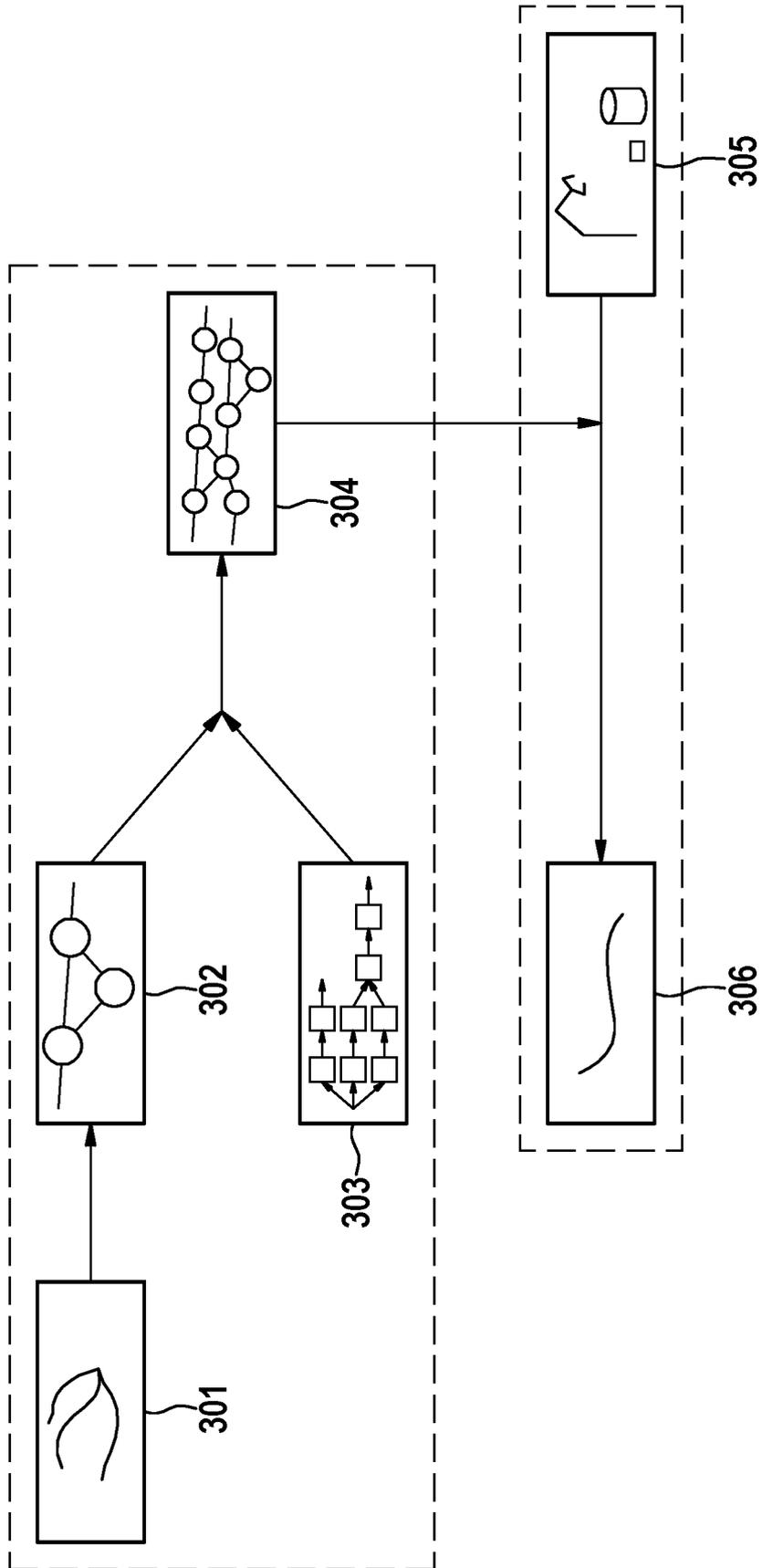


Fig. 3

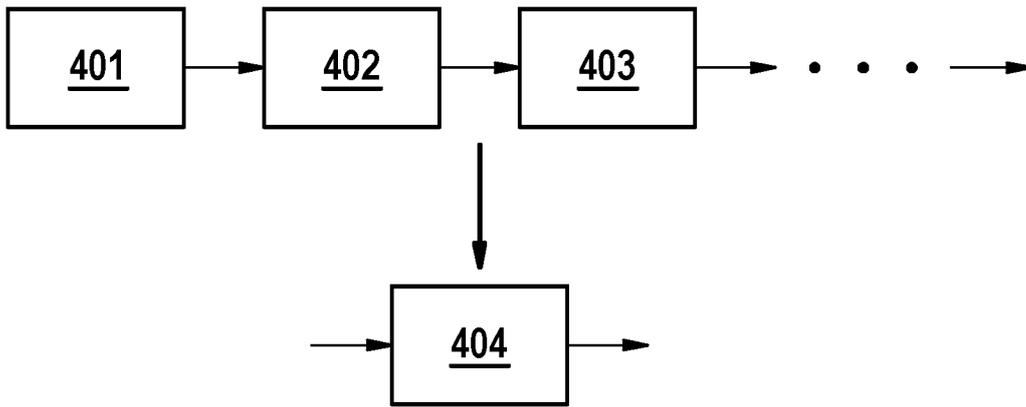


Fig. 4

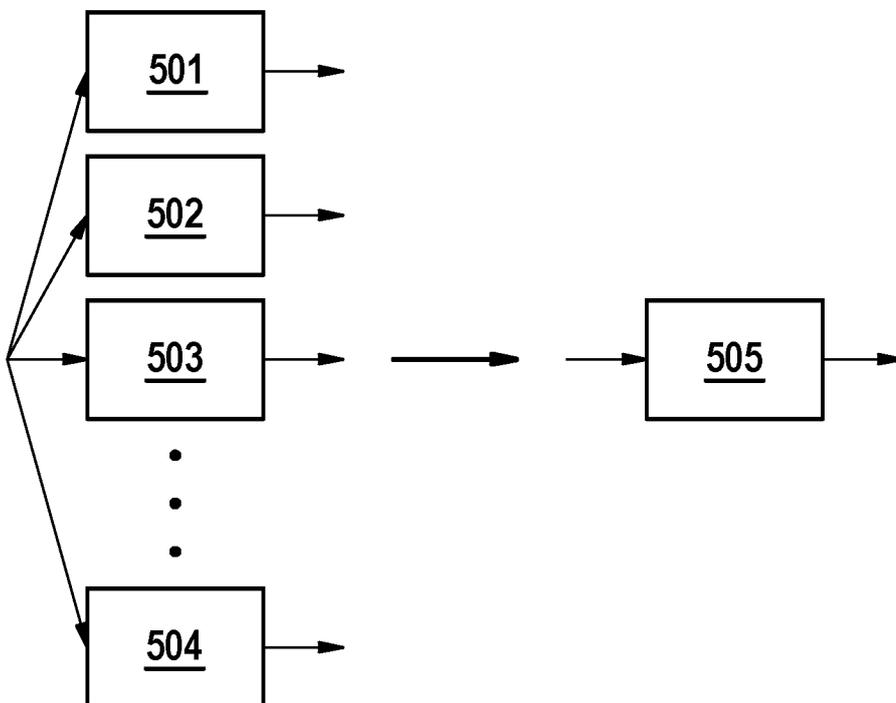


Fig. 5

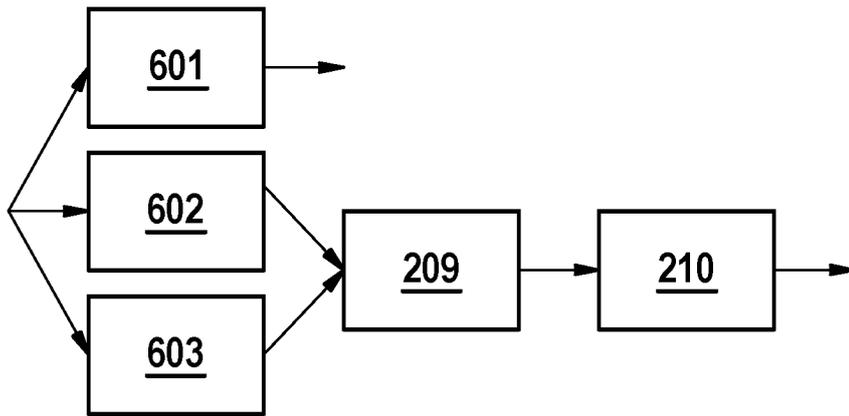


Fig. 6

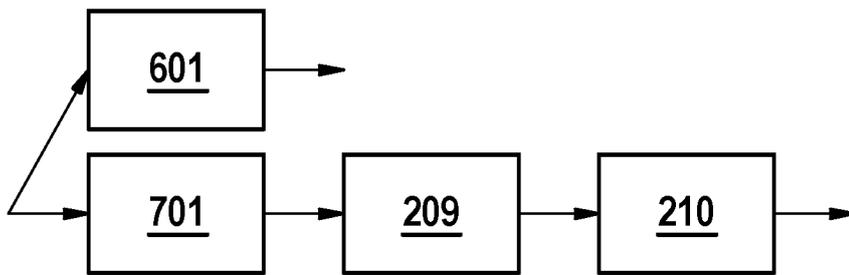


Fig. 7

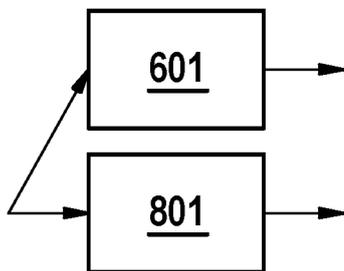


Fig. 8

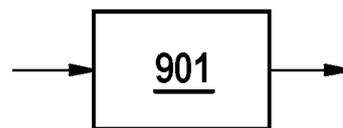


Fig. 9

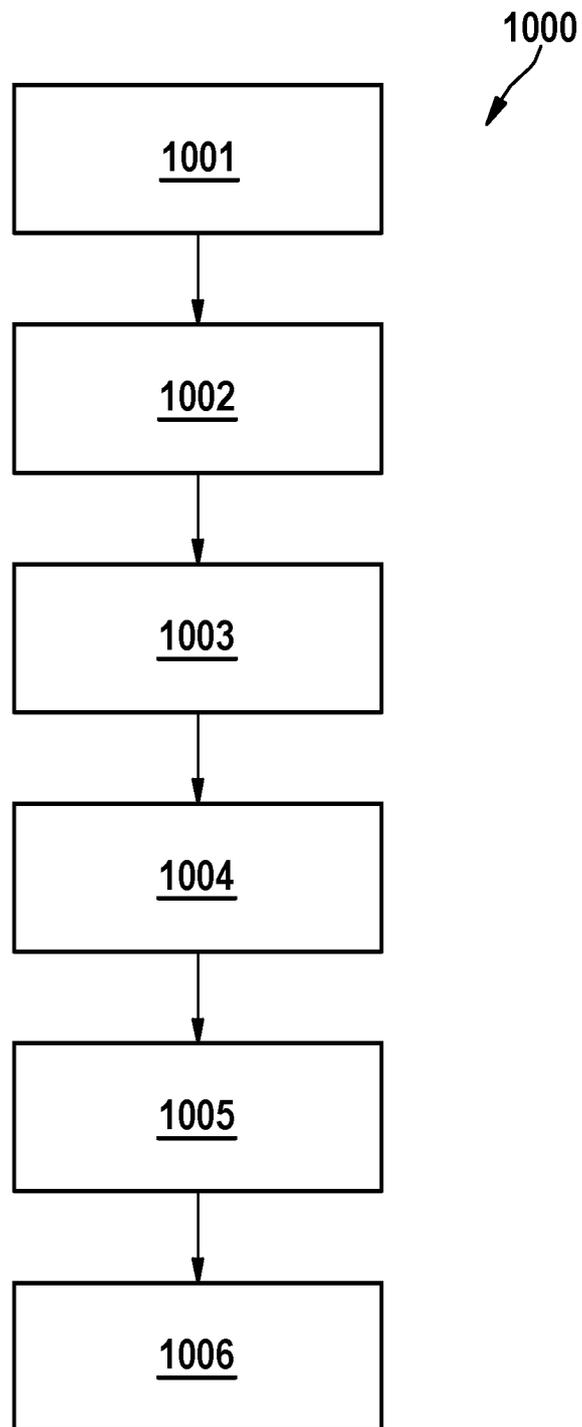


Fig. 10