

(12) **Österreichische Patentanmeldung**

(21) Anmeldenummer: A 412/2010
(22) Anmeldetag: 12.03.2010
(43) Veröffentlicht am: 15.09.2011

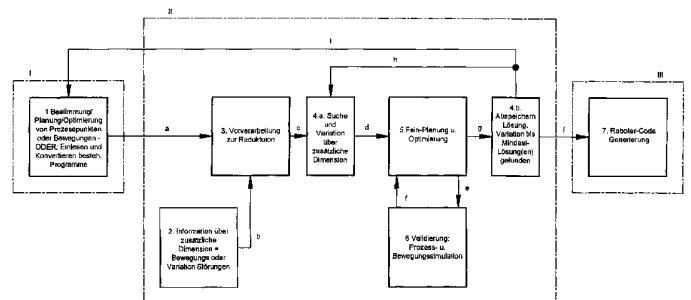
(51) Int. Cl. : **G05B 19/04** (2006.01)
G06N 7/08 (2006.01)
G06F 17/17 (2006.01)

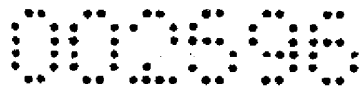
(73) Patentanmelder:
CONVERGENT INFORMATION TECHNO-
LOGIES GMBH
A-4407 STEYR-GLEINK (AT)

(54) **VERFAHREN ZUR AUTOMATISIERTEN PROGRAMMIERUNG UND OPTIMIERUNG VON ROBOTISCHEN ARBEITSABLÄUFEN**

(57) Es wird Verfahren zur automatisierten Programmierung und Optimierung von robotischen Arbeitsabläufen mit folgenden Verfahrensschritten vorgestellt: Programmieren des Arbeitsablaufes für einen virtuell vereinfachten Fall, insbesondere an einem lagefesten Werkstück, Aufspannen und Einschränken eines Suchraumes unter Berücksichtigung von Informationen über eine zusätzliche Komplexität/Dimension, insbesondere Berücksichtigung von Bewegungen von Werkstück bzw. Aktuatoren und Festlegen eines zeitlichen Rahmens für Start und Endzeitpunkt des Programmes bzw. festlegen von erlaubten und Verbotenen Bereichen für Werkstück und Aktuatoren, Suchen nach Startplänen/Lösungen für eine Feinplanung in einer Grobplanung und festlegen der Reihenfolge, in der die Feinplanungen in einer Schleife abgearbeitet werden, optimieren, validieren oder verwerfen von in der Grobplanung gefundenen Lösungen durch eine Feinplanung, abspeichern gefundener Lösungen und übersetzen ausreichend guter Lösungen in ein oder mehrere Roboterprogramme.

Fig. 1





Patentanwälte
Dipl.-Ing. Helmut Hübscher
Dipl.-Ing. Karl Winfried Hellmich
Dipl.-Ing. Friedrich Jell
Spittelwiese 7, A 4020 Linz

(37 006)

Z u s a m m e n f a s s u n g :

Es wird Verfahren zur automatisierten Programmierung und Optimierung von robotischen Arbeitsabläufen mit folgenden Verfahrensschritten vorgestellt:

Programmieren des Arbeitsablaufes für einen virtuell vereinfachten Fall, insbesondere an einem lagefesten Werkstück, Aufspannen und Einschränken eines Suchraumes unter Berücksichtigung von Informationen über eine zusätzliche Komplexität/Dimension, insbesondere Berücksichtigung von Bewegungen von Werkstück bzw. Aktuatoren und Festlegen eines zeitlichen Rahmens für Start und Endzeitpunkt des Programmes bzw. festlegen von erlaubten und Verbotenen Bereichen für Werkstück und Aktuatoren, Suchen nach Startplänen/Lösungen für eine Feinplanung in einer Grobplanung und festlegen der Reihenfolge, in der die Feinplanungen in einer Schleife abgearbeitet werden, optimieren, validieren oder verwerfen von in der Grobplanung gefundenen Lösungen durch eine Feinplanung, abspeichern gefundener Lösungen und übersetzen ausreichend guter Lösungen in ein oder mehrere Roboterprogramme.

(Fig. 1)



(37 006) hel

Die Erfindung betrifft ein Verfahren zur automatisierten Programmierung und Optimierung von robotischen Arbeitsabläufen.

Die Programmierung von Anwendungen im Bereich der automatisierten Bearbeitung bewegter Werkstücke erfolgt bislang in Off-Line-Programmierung, welche nur ungenügend Anwendungen unterstützt die zusätzliche Komplexitäten in der Form von weiteren Freiheitsgraden oder Variationen aufweisen und die bei der Programmierung berücksichtigt werden müssen. Dies ist z.B. der Fall wenn berücksichtigt werden muss, dass das Werkstück auf einem (nicht frei programmierbaren) Förderer bewegt wird, während der Roboter beispielsweise fest montiert ist, dass der Roboter mit einem weiteren Roboter oder einem frei programmierbaren Aktuator kooperiert welcher das Werkstück positioniert und/oder dass Störungen oder Abweichungen, beispielsweise Schwankungen der Werkstückelage oder der Prozessparameter, auftreten können. Diesen Fällen ist gemein, dass zusätzliche Freiheitsgrade oder Varianten eingeführt werden die mit bekannten Methoden, wenn überhaupt, nur sehr schwer berücksichtigt werden können.

In der Produktion werden Werkstücke mit automatischen Fördereinrichtungen oft von Arbeitszelle zu Arbeitszelle befördert. Es ist dabei als ideal anzusehen, wenn das zu bearbeitende Werkstück für eine Ausführung einer programmierten Anwendung (Bearbeitung) nicht gestoppt bzw. aus der Fördereinrichtung ausgeschleust werden muss, sondern sich mit der vorgegebenen Geschwindigkeit weiterbewegt. Die Programmierung von nicht trivialen Anwendungen, bei denen sich ein Werkstück bewegt während die Roboterbasis selbst z.B. fest montiert ist (Line-Tracking), beinhaltet jedoch eine zusätzliche in der Programmierung zu bewältigende Komplexität.

Bei bekannten Lösungen für einfache Anwendungen gleicht der Roboter-Controller die Bewegung des Werkstücks aus, indem er sein Zielkoordinatensystem mit der Vorschubinformation des Förderers abgleicht. Da die reine Verschiebung des Koordinatensystems keine Prüfung auf Kollisionen beinhaltet, erfordert dieser Ansatz, dass beim Aufbau der Zelle an sich schon die Kollisionsfreiheit gewährleistet wird. Erreicht kann dies beispielsweise dadurch werden, dass der Roboter so montiert ist, dass er beim Ausführen der Anwendung aus einem freien Bereich, beispielsweise von oben her, gegen das bewegte Werkstück herangeführt wird. Die Anwendung wird also ungeprüft auf etwaige Kollisionen ausgeführt. Dieses Verfahren eignet sich nicht für Anwendungen, bei denen der Roboter längeren Kontakt zwischen einem Werkzeug und einem Werkstück herstellen soll, insbesondere wenn das Werkzeug über die Werkstückoberfläche bewegt werden soll, oder wenn es sich um ein komplexes Werkstück handelt, da nicht über längere Zeit und weitere Bewegung des Werkstücks hinweg vorab sichergestellt werden kann, dass nicht, beispielsweise der Ellbogen des Roboters mit dem Werkstück kollidiert.

Eine andere Möglichkeit zur Bearbeitung bewegter Werkstücke besteht darin, den Roboter für komplexere Aufgaben auf eine Zusatzachse zu stellen, und den Roboter über weite Bereiche oder an kritischen Bereichen der Anwendung synchron mit dem Werkstück zu bewegen. Damit gleicht dieses Vorgehen einer Anwendung mit quasi stehendem Werkstück, mit dem Nachteil zusätzlich erforderlicher Achsen, Kosten, Platzkosten und Wartung.

In einem zweiten mit der Erfindung zu verbessernden Fall werden in der Produktion Werkstücke mittels eines programmierbaren Aktuators (z.B. einem weiteren Roboter, einem programmierbaren Dreh-Schwenk-Neigetisch oder einem Linearpositionierer) in einer für den die Anwendung ausführenden Prozessroboter günstigen Weise positioniert, wobei das Werkstück verschiedene, sich während des Prozesses gegebenenfalls verändernde Lagen einnimmt. Dies erhöht die Erreichbarkeit zu bearbeitender Punkte durch den Prozess-Roboter und erlaubt komplexe und verschiedenste Teile in einer Zelle zu bearbeiten. Die Programmierung nicht trivialer Anwendungen, bei denen zusätzlich zum Prozess-Roboter weitere Roboter oder frei

programmierbare Aktuatoren eingesetzt werden, beinhaltet eine zusätzliche Komplexität in der Programmierung, die mit bekannten Mitteln kaum bewältigt werden können. Der Unterschied zum oben erwähnten Line-Tracking ist, dass die zusätzlichen Freiheitsgrade frei programmierbar sind, während beim Line-Tracking die Geschwindigkeit des Förderers vorgegeben ist und sich das Roboterprogramm daran anpassen muss.

Derartige Kooperative Roboter oder Aktuatoren sind derzeit noch wenig verbreitet. In den Fällen in denen Sie eingesetzt werden nimmt man für die Vorteile eine sehr aufwändige konventionelle Programmierung in Kauf, wobei zumindest die Synchronisation der Programmausführung für alle Aktuatoren durch den Robotercontroller sehr gut unterstützt werden. Dazu führen die Roboter über eine Transformation zueinander definierter Werkzeugpunkte gleichförmige Bewegungen aus wodurch die Roboter so bewegt werden können als wären sie eine gekoppelte Kinematik. Jedoch gibt es weder eine Unterstützung zur Überprüfung, ob diese gekoppelte Bewegung auch kollisionsfrei ausführbar ist, noch für das automatische Anpassen bzw. Umplanen von Anwendungen. Daher werden Prozessschritte in denen sich Roboter und kooperierender Aktuator während eines Prozessschrittes gleichzeitig bewegt werden üblicherweise entweder vermieden, vergleichsweise kurz gehalten oder derart optimiert, dass einer der beiden Roboter/Aktuatoren nur möglichst einfache Bewegungen ausführt.

In einem dritten mit der Erfindung zu verbessernden Fall treten in der Produktion in relevanten Parametern (z.B. Teilelage, Prozessparametern) Schwankungen, Variationen und/oder Störungen und somit Abweichungen vom Ideal auf, die, sofern sie jenseits eines tolerierbaren Bereichs liegen, die die Ausführbarkeit der Anwendung oder die Qualität des Ergebnisses beeinträchtigen können. Die Programmierung von Anwendungen die derartige, zusätzliche Komplexitäten aufweisen, wird, weil unvorhergesehene statistische Prozessschwankungen in der Programmierung berücksichtigt werden müssen, von bekannten Methoden nicht ausreichend unterstützt.

In der Regel wird erst nach der Programmierung, d.h. während eines Betriebs der Anlage durch Tests oder durch Simulation des fertigen Programms, ermittelt bis zu welchen Schwankungen die Anwendung gewünschte Ergebnisse liefert. Diese Toleranzen müssen dann eingehalten werden und sind beispielsweise mit entsprechenden Sensoren zu validieren. Liegen die Werte außerhalb der Toleranzen wird die Anwendung nicht gestartet. Da konventionell nur weniger robuste Programme programmiert werden können, und die Programmierung für verschiedene Störungen zeitaufwendig ist, müssen die Toleranzen klein gehalten werden und erfolgt öfter als nötig ein Abbruch der Anwendung und somit der Produktion.

Der Erfindung liegt ausgehend von einem Stand der Technik der vorgeschilderten Art die Aufgabe zugrunde ein Verfahren zur voll- oder teilautomatischen Programmierung und Optimierung von robotischen Anwendungen anzugeben, mit dem eine vorteilhafte Programmierung von kollisionsfreien, nicht trivialen Anwendungen möglich ist, bei denen ein Werkstück bewegt wird bzw. bei denen in relevanten Parametern Schwankungen, Variationen und/oder Störungen und somit Abweichungen vom Ideal auftreten.

Die Erfindung löst diese Aufgabe durch folgenden Verfahrensablauf

- Programmieren des Arbeitsablaufes für einen virtuell vereinfachten Fall, insbesondere an einem lagefesten Werkstück,
- Aufspannen und Einschränken eines Suchraumes unter Berücksichtigung von Informationen über eine zusätzliche Komplexität/Dimension, insbesondere Berücksichtigung von Bewegungen von Werkstück bzw. Aktuatoren und Festlegen eines zeitlichen Rahmens für Start und Endzeitpunkt des Programmes bzw. festlegen von erlaubten und Verbotenen Bereichen für Werkstück und Aktuatoren,
- Suchen nach Startplänen/Lösungen für eine Feinplanung in einer Grobplanung und festlegen der Reihenfolge, in der die Feinplanungen in einer Schleife abgearbeitet werden,
- optimieren, validieren oder verwerfen von in der Grobplanung gefundenen Lösungen durch eine Feinplanung,

- abspeichern gefundener Lösungen und
- übersetzen ausreichend guter Lösungen in ein oder mehrere Roboterprogramme.

Den hier behandelten drei Fällen ist dieser eine Erfindungsgedanke gemein, wobei der Bediener oder eine automatische Planung die Anwendung für einen virtuell vereinfachten Fall in herkömmlicher Weise programmieren kann ohne vorgegebene Werkstück-Bewegungen, ohne frei programmierbare Werkstück-Bewegungen in Folge von Kooperation von Aktuatoren, oder ohne Variationen und Störungen berücksichtigen zu müssen. Deren Berücksichtigung erfolgt im Anschluss voll- oder teilautomatisch nach dem im Folgenden dargestellten Verfahren. Dazu wird in einer Vorverarbeitung anhand quantitativer, insbesondere metrischer, Zusatzinformation über die zusätzlich zu betrachtenden Komplexität ein Suchraum aufgespannt und eingeschränkt und so für eine schnellere anschließende Suche aufbereitet. Anschließend sucht eine Grobplanung nach Startlösungen, beispielsweise für einzelne Flächen, und legt die Reihenfolge fest in der die Startlösungen in einer Schleife abzuarbeiten sind. Innerhalb dieser Schleife optimiert und validiert oder verwirft eine Feinplanung diese Startlösungen. Gefundene Pläne/Metaprogramme werden gespeichert. Nach dem Finden einer ausreichend guten Lösung oder von Lösungen für alle Fälle werden ein oder mehrere Metaprogramme gespeichert und in ein oder mehrere ausführbare Roboterprogramme übersetzt. Der Benutzer bzw. ein Software-Tool muss sich nur um die Planung für den vereinfachten Fall kümmern. Die zusätzliche Komplexität wird mit dem erfindungsgemäßen Verfahren berücksichtigt, das gleichzeitig Bewegungen des Roboters und die Anwendung optimiert.

Fall A: Für den Fall, dass berücksichtigt werden muss, dass sich das Werkstück mit einem nicht frei programmierbaren Förderer bewegt, wird die robotische Anwendung in einem ersten Schritt in üblicher Weise geplant als würde das Werkstück feststehen. Dies geschieht vorteilhaft mit einer voll- oder teilautomatischen Planung, Simulation und Optimierung des Prozesses und der Werkzeugbewegung. Alternativ können konventionell generierte Programme für vereinfachte Anwendungen einge-

lesen werden und in Prozesspunkte oder Pfade im Werkstückkoordinatensystem transformiert werden.

In Folge werden je Prozesspfad oder –Punkt der programmierten Anwendung zur Einschränkung des Suchraumes erlaubte Bereiche für einen vom Ideal abweichenden Versatz des zu bearbeitenden Werkstücks ermittelt. In diesen Bereichen sind die Bewegungsabläufe zur Bearbeitung des Werkstücks auszuführen.

Anschließend werden in einer Grobplanung die prinzipiell ausführbaren Reihenfolgen und Richtungen der Pfade/Bewegungen bestimmt und die gefundenen Reihenfolgen nacheinander in einer Schleife geprüft. Diese Reihenfolgeplanung schätzt die Ausführungsdauer für jeden Prozesspfad oder –punkt. Aus diesen Zeiten ist auch die Bewegung die das Werkstück auf dem Förderer (mit definierter Geschwindigkeit) gleichzeitig ausführt bestimmt. Die Grobplanung muss jetzt die Anordnungen der Pfade suchen für bei der jeder Pfad in einen Zeitraum und damit resultierendem Versatz zwischen Roboterbasis und stetig bewegtem Bauteil und fällt in dem der Pfad kollisionsfrei ausführbar ist und die Zeit zwischen den Pfaden hinreichend dimensioniert ist. Diese Zeit wird als Versatz des Bauteils zur Roboterbasis zwischen Zwei Prozesspfaden oder punkten berücksichtigt.

Innerhalb der Schleife werden die Pläne aus der Grobplanung weiter verfeinert und validiert, d.h. genaue Start-End-Werte (in Bezug auf den Vorschub des Förderers zur Roboterbasis) und Timing (Pausen, Geschwindigkeiten) und der genauen Pfadverlauf selbst optimiert bzw. fein-geplant. Zur Validierung werden die geplanten zeitgleichen Werkstück- und Roboterbewegungen (während der Prozesspfade und der Verbindungsbewegungen zwischen diesen) genau simuliert. Das Ergebnis „nicht ausführbar“ oder das aus dieser genauen Bewegung resultierende Prozessergebnis wird als Optimierungskriterium für die Feinplanung /Optimierung verwendet. Die Feinplanung und Optimierung (innerhalb der Schleife über Reihenfolgen und Startpunkte) verwertet dabei das Prozessergebnis wie folgt: Das Ergebnis der Prozesssimulation (z.B. Schichtdicke auf der Werkstückoberfläche beim Lackieren (in Werten oder „OK, zu dick oder zu dünn“), Reinigungswirkung beim Reinigen



(gereinigt, nicht gereinigt, unsicher)) wird auf die Oberfläche des Werkstücks „projiziert“ (auf den Mesh-Daten (Netzbeschreibung der Oberfläche des Bauteils) abgespeichert) und kann für jeden Punkt der Oberfläche abgefragt werden. Die Planung oder Optimierungsalgorithmen werden nun die Positionen des Werkzeuges und damit des Roboters in Bezug zur Oberfläche so verändern, dass der Anteil der relevanten Werkstückoberfläche bei denen die Qualitätskriterien eingehalten werden maximiert wird. „Lücken“ oder „Fehlstellen“ werden vorzugsweise stärker gewichtet so dass diese bevorzugt geschlossen werden und zusammenhängende Regionen erreicht werden die die Kriterien erfüllen. Bewegungen die nicht ausführbar sind werden verworfen oder mit einem hohen Maluswert bewertet. Optional kann beim ersten hinreichend gutem Ergebnis die Schleife abgebrochen werden. Durch die genaue Planung von Prozesspfaden und Zwischenbewegung explizit auf Bewegungsebene wird aus dem Plan ein Metaprogramm welches Zwischengespeichert wird. Wenn es für keine der geplanten Reihenfolgen (Varianten) eine ausreichende Lösung (-svariante) gibt, wird zurückgegangen auf die „quasistatische Erstbestimmung“. In einer vorteilhaften Realisierung wird dabei gleich angezeigt „wo (bei früheren Versuchen) das Problem lag“ also welche Pfade oder Prozesspunkte oder welche Bereiche des Werkstücks kritisch oder unkritisch sind.

Das Endergebnis ist ein Meta-Programm das danach in ein Programm konvertiert wird welches sicher synchron zu den Förderbewegungen ausgeführt werden kann. Optional können Angaben enthalten sein wie weit die Synchronisation Vorschubschwankungen des Förderers sicher ausgleichen darf und ab wann ein Not-Aus nötig ist.

Fall B: In einem ersten Schritt wird die Anwendung (für den Bediener vereinfacht) programmiert/geplant als würde es keine Kooperation zwischen Werkzeug und Werkstück geben und als wäre das Werkstück für den „Prozess-Roboter“ immer perfekt zugänglich. Die Planung erfolgt über Prozess-Pfade oder Punkte und nicht über die Roboterbewegungen, da letztere wegen der Freiheitsgrade von Roboter und Aktuator noch nicht feststehen können. In einer vorteilhaften Realisierung erfolgt die Planung der virtuell vereinfachten Anwendung beschleunigt durch eingebet-

tete voll- oder teilautomatische Planung, Simulation und Optimierung des Prozesses und der Werkzeug-Bewegungen.

Anschließend wird der Bereich über den optimiert werden soll vorverarbeitet, um den Suchraum zu reduzieren. Dazu können a) Bereiche erlaubter Werkstückpositionen ermittelt werden in denen jeweils ein einzelner Pfad des (virtuell statisch geplanten) Plans mit dem Prozess-Roboter ausgeführt werden kann (und das Prozessergebnis gemäß Simulation in Toleranzen erreicht wird) und an denen das Werkstück auch durch den kooperierenden Aktuator positioniert werden kann, und/oder b) über Vorgaben z.B. an die Lage des Werkstücks wird der Zustandsraum des kooperierenden (das Teil bewegendes) Aktuators verkleinert (Anm: z.B. nur Gelenkwinkel bei denen das Werkstück nicht auf dem Kopf steht).

Anschließend wird durch Kombinationen von Werkstückpositionen und Reihenfolgen für die einzelnen Prozesspfade oder Punkte nach möglichen Lösungen gesucht die dann in einer Schleife abgearbeitet werden. Diese Suche nach günstigen Reihenfolgen und Werkstückpositionen für jeden einzelnen Pfad verwendet in einer vorteilhaften Realisierung als Optimierungskriterium die Minimierung der Umrorientierungen, d.h. Änderungen der aufeinanderfolgenden Werkstückpositionen (da diese schnell ermittelt werden können) oder Werkzeugausrichtungen (alternativ: Summe über alle Bewegungen oder Dauer der Ausführung etc). Innerhalb der Schleife werden dann die jeweils ausgewählte Kombination von Reihenfolgen und Werkstückepositionen je Prozess-Pfad oder Punkt genauer untersucht, wobei dann anders als in der Vorprüfung die Werkstückeposition variiert werden darf während ein Prozesspfad oder -Punkt abgearbeitet wird. Dabei werden die synchronen Bewegungen von Prozessroboter und Aktuator für die Prozesspfade (oder Punkte) und die Verbindungsbewegungen zwischen diesen und das daraus resultierende Prozessergebnis optimiert (Feinplanung) und durch Simulation der gleichzeitigen Bewegung validiert. Das Simulationsergebnis wird identisch wie im Fall A, Line-Tracking, ermittelt und als Optimierungskriterium für die Optimierung/Feinplanung verwendet. Die sequentielle Untersuchung der Kombinationen (d.h. die Schleife) kann beendet werden sobald eine (ausreichend gute) Lösung gefunden wurde.

Durch die genaue Feinplanung von Prozesspfaden und Zwischenbewegung explizit auf Bewegungsebene wird aus dem Plan ein Metaprogramm welches Zwischengespeichert wird.

Wenn in all den untersuchten Variationen keine Lösung gefunden wird (oder keine robuste Lösung bzgl. Sicherheit, Mindestabstand u. dgl.), wird zurückgegangen auf die „quasiideale Erstbestimmung“, wobei gleich angezeigt wird „wo das Problem ist“ also welcher Pfad oder welcher Bereich der Werkstücks (einfach oder mehrfach) kritisch oder unkritisch war. Wird eine Lösung gefunden wird das entsprechende Meta-Programm in ein ausführbares Roboterprogramm für einen spezifischen Roboterdialekt übersetzt.

Das Endergebnis ist ein geprüftes Programm das sicher synchron durch Prozessroboter und kooperierende Aktuatoren ausgeführt werden kann. Die Aktuatoren sind weit präziser als ein Förderband und in Echtzeit kontrollierbar. Tolerierte Abweichungen von der Sollposition könnten zwar errechnet werden, sind aber nicht so sinnvoll wie bei einem Förderband das auch mal stehen bleiben muss.

Fall C: In einem ersten Schritt wird die (ideale) robotische Anwendung programmiert/geplant als würde es keine Schwankungen oder Störungen geben – d.h. idealisiert und virtuell frei von variierenden Einflüssen – und somit „wie gewohnt“ vereinfacht für den Bediener. Dies geschieht z.B. beschleunigt durch eingebettete voll- oder teilautomatische Planung und Simulation von Prozess und Bewegung. Um die folgende Planung zu beschleunigen werden die erwarteten Schwankungen vorab analysiert in wie weit sie unabhängig, korreliert oder gegenseitig ausschließend auftreten, wonach über alle sinnvoll möglichen Zustände ein minimaler Zustandsraum aufgespannt werden kann, aus dem alle nicht einnehmbaren Zustände bereits eliminiert sind.

In weiterer Folge wird in einer Schleife über Störungen und Schwankungen (diskret) variiert und für diese Fälle nach Lösungen gesucht, wobei die Abfolge und die (Dichte der) Abtastung der in der Schleife untersuchten Variationen/Störungen

anhand der Ergebnisse vorheriger Durchläufe gesteuert sein kann. Beispielsweise ein „Region-Growing“ ähnlicher Ansatz, oder eine gesteuerte Abtastung (z.B. um Stetigkeiten auszunutzen und dort weniger Variationen untersuchen zu müssen wo die Ergebnisse vorheriger Feinplanungen eine Lösung gefunden haben die sehr robust ist (z.B. keine Lösung in der Nähe zu Gelenkwinkelrestriktionen oder zu Kollisionen, nur gering beeinflusstes Prozessergebnis)).

Innerhalb der Schleife wird (vorteilhaft) zuerst evaluiert ob ein bereits vorher geplanter Plan oder ein Metaprogramm für die aktuelle Variation ausführbar ist und ein Ergebnis liefert, welches innerhalb der erlaubten Toleranzen liegt. Dazu wird das Ergebnis der Prozesssimulation (für die gemäß Bewegungssimulation kollisionsfrei ausführbaren Prozesspfade) wie z.B. die resultierende Schichtdicke auf der Werkstückoberfläche beim Lackieren (in Werten oder „OK zu dick, zu dünn“), oder z.B. die Reinigungswirkung beim Reinigen (gereinigt, nicht gereinigt, unsicher,...) auf die Oberfläche des Werkstücks „projiziert“ (auf den Mesh-Daten abgespeichert) und anschließend für jeden Punkt der Oberfläche abgefragt. Ist der Anteil der relevanten Werkstückoberfläche bei denen die Qualitätskriterien eingehalten werden zur Gesamtoberfläche über der geforderten Schranke (wobei verschiedene Regionen (z.B. Designkanten) verschieden gewichtet werden können), so ist der getestete Plan für diese aktuelle Variation freizugeben. Dann wird diese Variation mit den anderen erlaubten Variationen des passenden Plans zusammengefasst (geclustert). Wenn nicht, so wird der Planungsprozess für die Variation neu angestoßen und durch Adaptierung der Pfadverläufe selbst (z.B. unter Verwendung des bisherigen Plans mit dem besten Ergebnis oder des Plans für den „idealen Fall“) ein neuer Plan erstellt und optimiert. Zur Validierung werden die geplanten Roboterbewegungen (Prozesspfade und Verbindungsbewegungen zwischen diesen) simuliert und deren Ausführbarkeit ggf. –Zeit und das aus dieser genauen Bewegung resultierende Prozessergebnis als Optimierungskriterium für die Planung/Optimierung verwendet. Wie in den vorherigen Fällen wertet die Optimierung (innerhalb der Schleife über Reihenfolgen und Startpunkte) dabei das Prozessergebnis wie folgt: Das Ergebnis der Prozesssimulation wird auf die Oberfläche des Werkstücks „projiziert“ (auf dem Mesh-Daten abgespeichert) und für jeden Punkt der Oberfläche abgefragt. Die

Planung oder Optimierungsalgorithmen werden nun die Positionen des Tools und damit des Roboters in Bezug zur Oberfläche so verändern, dass der Anteil der relevanten Werkstückoberfläche bei denen die Qualitätskriterien eingehalten werden maximiert wird. Vorzugsweise werden „Lücken“ oder „Fehlstellen“ stärker gewichtet so dass diese bevorzugt geschlossen werden und zusammenhängende Regionen erreicht werden, die die Kriterien erfüllen. Bewegungen die nicht ausführbar sind werden verworfen oder mit einem hohen Maluswert bewertet. Durch die genaue Planung von Prozesspfaden und Zwischenbewegung explizit auf Bewegungsebene wird aus dem Plan ein Metaprogramm welches zusammen mit dem Wert der untersuchten Variation gespeichert wird. D.h. es wird ein neuer Cluster angelegt und später untersuchte Störungen, die mit demselben Metaprogramm bzw. Plan abgedeckt werden können, werden dazugefasst (geclustert).

Die Schleife wird weitergeführt bis für jede Variation/Störung ein Plan existiert, d.h. es sind keine Störungen mehr unabgedeckt. Wichtig: die Randbereiche eines Clusters sollen in einem benachbarten Cluster auch vorkommen, damit nicht „Lücken“ zwischen den Parametern entstehen die de-facto nicht zugeordnet und evtl. nicht ausführbar sind. Wenn der gesamte Bereich der Schwankungen mit einem einzigen Plan abgedeckt werden kann, ist das Ergebnis ein einziger (robuster) Plan oder Metaprogramm der/das in ein Programm umgewandelt wird, ansonsten werden mehrere Pläne oder Metaprogramme erstellt Anm: Die Schleife kann abgebrochen werden wenn für einen Fall keine Lösung gefunden wurde (zurück zur initialen Bestimmung) oder alle Fälle werden abgehandelt und nur für die (wenigen) Fälle für die keine Lösung gefunden werden komplett neu gesucht während die gefundenen Lösungen beibehalten werden. Idealerweise können in einer nachgeschalteten Optimierung die Pläne/Metaprogramme nochmal über die Breite der im Cluster abgedeckten Variationen/Störungen optimiert werden. Metaprogramme werden abschließend in ausführbare Roboterprogramme für einen spezifischen Roboterdiaklekt übersetzt.

Bei der on-line Ausführung der Automation wird die Störung (z.B. Lageabweichung) gemessen und der entsprechende Cluster ausgewählt und das entsprechende

Programm aufgerufen. Optional kann mit der genau gemessenen Abweichung das gespeicherte (im Mittel optimale) Programm aufgerufen werden und anschließend für die genau gemessene Störung / Parameter optimiert werden.

In den Zeichnungen ist die Erfindung an Hand eines Ausführungsbeispiels schematisch dargestellt. Es zeigen:

- Fig. 1 Ein allen vorgestellten Varianten gemeinsamer Verfahrensablauf eines erfindungsgemäßen Verfahrens,
- Fig. 2 Details der inneren Planung und Optimierung des Verfahrens aus Fig. 1,
- Fig. 3 Schematischer Verfahrensablauf für ein mit einem nicht frei programmierbaren Förderer bewegtes Werkstück (Fall A),
- Fig. 4 einen Verfahrensablauf bei einer Kooperation zwischen Werkzeug und Werkstück, also frei programmierbaren Werkzeug- und Werkstückbewegungen,
- Fig. 5 einen Verfahrensablauf zur Berücksichtigung variierender Einflüsse und
- Fig. 6 Details eines Verfahrensablaufes aus Fig. 5.

Nach dem der Erfindung zugrunde liegenden Verfahren (Fig. 1) wird erst eine idealisierte Anwendung geplant (bzw. programmiert). Idealisiert heißt hierbei z.B. quasi-statisch oder quasi- ohne Störungen oder wie dem Bediener von einfacheren Anwendungen vertraut. In einer vorteilhaften Ausführung wird der Bediener dabei bereits durch Planung und/oder Simulation und/oder Optimierung des Prozesses und der Roboterbewegungen unterstützt. Alternativ können auch existierende Programme eingelesen und konvertiert werden. Erst danach wird (voll- oder teil-) automatisiert die zusätzliche, durch Dynamik oder Störungen bedingte Komplexität/Dimension berücksichtigt und werden die Pläne generiert die für diese komplexen Fälle einsetzbar sind.

Dies geschieht wie folgt: In einer Vorverarbeitung wird der Suchraum aufgespannt und eingeschränkt. Dann sucht eine Grobplanung nach Startlösungen oder Fällen. Diese werden in einer Schleife abgearbeitet und gefundene Lösungen werden optimiert und validiert und ggf. zusammengefasst. Aus gefunden Plänen wird ein oder werden mehrere Programme zusammengefügt. I und III zeigen den bekannten

Verfahrensablauf und II die zwischen I und III eingeschobene (voll- oder teil-) automatische erfindungsgemäße Bearbeitung. Fig. 1 zeigt insbesondere

1. Initiale Bestimmung / Planung und ggf. Optimierung der Prozessbahnen und/oder -Punkte für den virtuell vereinfachten Fall, d.h. ohne Berücksichtigung der zusätzlichen Dimension(en) (Bewegung, Variation). Anm.: In einer vorteilhaften Realisierung sind diese auf Kollisionsfreiheit des Werkzeugs geprüft. Bei der Prozessabsicherung ist auch die Kollisionsfreiheit des Roboters vorab prüfbar. Alternativ können auch existierende oder klassisch programmierte Programme eingelesen und (vorzugsweise in Werkstückkoordinaten) konvertiert werden.
2. Eingabe oder Einlesen der Information über die zusätzliche Komplexität/Dimension. (Im Falle des „Line-Tracking“ ist dies die Bewegungsgeschwindigkeit eines Förderers und damit des Werkstücks, und frühester Start- späterster Endzeitpunkt des Programms. Im Falle der kooperativen Robotik sind dies Angaben über den Bewegungsraum der kooperierenden Aktuatoren oder Angaben über Bereiche/Lagen die das Werkstück nicht einnehmen darf. Im Falle der Prozessabsicherung sind dies Information über Einzelstörungen und ggf. deren Abhängigkeiten oder Ausschluss.)
3. Optionale Vorverarbeitung der Information aus (2) zum Aufspannen und Reduzieren des Suchraums (in 4):
 - a. Im Falle des Line-Tracking werden „Fenster“ oder „Korridore“ bei denen die Prozessbahnen aus (1) durch den Roboter (ohne Konfigurationswechsel) erreichbar sind ermittelt.
 - b. Bei kooperativer Robotik (Roboter und Aktuator) analog und/oder es werden die Zustände- die nicht eingenommen werden dürfen weil z.B. das Werkstück auf dem Dach stünde aus der Zustandskarte der erlaubten Bereiche gelöscht oder als künstliche Hindernisse abgebildet.
 - c. Im Falle der Prozessabsicherung werden z.B. zusammengesetzte Störungsmuster auf ihre Zusammensetzung aus Einzelstörungen hin überprüft um die Anzahl der zu prüfenden Fälle zu reduzieren.
4. Grobplanung (Suche) nach den Startlösungen oder Reihenfolgen welche die in (3) spezifizierten Bedingungen oder Zustände abbilden. Abarbeitung dieser Start-

lösungen als Schleife. Für die Fälle a) und b), bei denen schon bei einer ersten (im folgenden ausreichend gute bewerteten) Lösung die Schleife abgebrochen werden kann beschleunigt die Grobplanung das Finden einer solchen Lösung durch Optimierung. Als Kriterien verwendet die Optimierung das Einhalten von Randbedingungen für die einzelnen Pfade (Grobprüfung das z.B. Pfad innerhalb des erlaubten Zeit/Positionsfensters) und/oder die geschätzte (unproduktive) Zeit (Summe der Umorientierungen oder Luft-Bewegungen oder auch akkumulierte Gelenkwinkel etc...). Für den Fall c) bei dem die Summe der (N) gefundenen Lösungen den Suchraum vollständig abdecken muss und daher alle Fälle abgearbeitet werden müssen, kann die Optimierung der Reihenfolge der Abarbeitung (z.B. gemäß Ähnlichkeitsmaß) in Verbindung mit einer gesteuerten Abtastung bei der die „Abtastdichte“ abhängig sind von vorherigen Ergebnissen (der Optimierung/Bewertung innerhalb der Schleife) oder Randregionen, die Anzahl der Durchläufe reduzieren.

5. Feinplanung bzw. Optimierung für den aktuellen, zuvor gefundenen „Fall“. Kriterium sind die Ausführbarkeit (ggf. Zeit) und die erzielte Prozessqualität (Anteil der relevanten Werkstückoberfläche bei der das Prozessergebnis innerhalb der Toleranzen liegt) die in der Simulation (6) ermittelt werden. Sofern eine Lösung gefunden wird, werden die Abläufe im Plan um genaue Bewegungen und Schaltpunkte etc. ergänzt d.h. zu einem Metaprogramm zusammengesetzt.

6. Validierung über die genaue Simulation der Bewegung auf Ausführbarkeit und Simulation des aus der genauen Bewegung resultierenden Prozessergebnisses (Prozessergebnis (OK, bzw. Über-/Unterschreitung) je Segment oder Teil der Oberfläche des Werkstückes).

5b Rücksprung wenn noch nicht alle Fälle abgearbeitet sind oder noch kein hinreichend gute Lösung gefunden wurde. Wenn alle Fälle abgearbeitet sind und beim Line-Tracking oder der kooperativen Robotik überhaupt keine Lösung (Variation) gefunden wurde, bzw. bei der Prozessabsicherung nicht für alle Fälle (Variationen) eine Lösung gefunden wurde erfolgt ein Rücksprung zu 1.

7. Umwandlung der Metaprogramme in ausführbare Roboterprogramme.

In den Zeichnungen beziehen sich die Buchstaben auf:

a: einen Plan oder eine Liste von Prozessbahnen bzw. -punkten,

- b: Information über die zusätzliche Dimension, z.B Bewegungsgeschwindigkeit des Werkstücks / Förderers und/oder von Störungen/Schwankungen bei der Prozessabsicherung,
- c: einen Plan oder eine Liste von Prozessbahnen oder -Punkten (vgl. b) zusammen mit Informationen über die zu prüfenden Zustände oder Reihenfolgen je Pfad,
- d: einen Initialer Plan (Prozesspfade oder -punkte mit Reihenfolge) zusammen mit Information über die aktuell zu prüfenden Zustände,
- e: einen durch die Optimierung/Feinplanung variierten Plan für die Simulation/Validierung,
- f: das Ergebnis der Validierung: verworfene und ausführbare Prozessbahnen. Letztere inkl. Genau ausführbarer Roboterbewegung und dem diesem entsprechenden Prozessergebnis und
- g: ein optimiertes/ adaptiertes Plan/Metaprogram oder Information darüber, dass keine gültige Lösung gefunden wurde.
- H: Einen Rücksprung, wenn noch nicht alle Fälle abgearbeitet wurden oder noch kein ausreichendes Ergebnis vorliegt (Bisherige Ergebnisse stehen für die Grobplanung zur Verfügung),
- i: einen Rücksprung zu (1) „quasi- idealisierten Planung“ wenn alle Fälle / Optionen der Schleife abgearbeitet wurden und keine Lösung oder nicht für alle Fälle eine Lösung gefunden wurde. Vorteilhaft ist dabei die Übergabe von Ergebnissen, wie (un-)kritische Stellen und/oder Pfaden und Variationen.
- J: Wenn alle Fälle abgearbeitet und ausreichende Lösungen bzw. Lösungen für alle Fälle gefunden wurden erfolgt eine Übergabe der Metaprogramme für eine Generierung ausführbarer Roboterprogramme.

In Fig. 2 wird die innere Schleife aus Fig. 1 beispielsweise im Detail dargestellt. Rechts Details der inneren Optimierung für die in der äußeren Schleife (4a) vorgegebene Variation. Wichtig: Die Reihenfolge von Bewegungssimulation/ -bewertung und Prozesssimulation/-Bewertung kann auch vertauscht werden, so dass zuerst der Prozess simuliert wird. Sofern nur Prozessparameter geändert werden die das Tool betreffen, aber nicht den Roboter (d.h. z.B. Luftparameter oder Lackmenge beim Lackieren aber nicht z.B. Tool-Geschwindigkeit über der Oberfläche die sich ja

auf die Roboterbewegung auswirken) kann eine neue Bewegungssimulation übersprungen werden. III kann dabei mit Ia zusammengefasst sein, oder in einer anderen Realisierung gar nicht vorkommen. Im Block Ia könnte anstelle oder in Zusammenarbeit mit der automatischen Lösung ein Mensch/Bediener hoch effizient eingreifen um auf Basis seiner Erfahrung oder Intuition Optimierungsvorschläge zu machen die dann in Modulen II-VII automatisch sehr schnell geprüft und bewertet werden.

- I. (a) Optimierung: Je nach Realisierung kann Modul I beim ersten Durchlauf gar nicht aufgerufen werden, da noch keine Simulation vorliegt. Dieser Fall gleicht einer Version in der I-a erst nach I-b durchlaufen wird und am Ende der Schleife steht.
- II. Modelle: beinhaltet Modell der Zelle, der Roboter, der Werkstücke, des Prozess-Tools und des Prozesses.
- III. Schnelle Vorab Test und Optimierung auf Basis Tool (Kollisionsvermeidung Tool, Überwachung prinzipieller Prozessparameter, Glättungen,...). Anm: III kann auch mit Ia zusammengefasst sein oder nicht verwendet.
- IV. Simulation der Bewegung (Genaue mit den Roboter(n) ausführbare Bahnen und Geschwindigkeiten) und Test auf Kollisionsfreiheit/Erreichbarkeit.
- V. Bewertung der Bewegung (Alle oder ausreichender Anteil der Bahnen erreichbar? Ausführungszeit?) Entscheidung ob Überspringen oder Prüfung der Lösung in der Prozesssimulation VI.
- VI. Simulation des Prozesses für die in IV genau bestimmte/simulierte Bewegung. Anreichern der Werkstückbeschreibung bzw. seiner Oberfläche mit Information aus der Prozesssimulation.
- VII. Bewertung des Prozessergebnisses aus der Simulation. Z.B. Anteil der Oberfläche des Werkstücks mit Prozessergebnis innerhalb der Parameter. Regionen mit Verletzung der Gütevorgaben etc.
- VIII. I. (b) Ende der Optimierungsschleife. Abspeichern. Beenden der inneren Optimierungsschleife und Weitergabe des besten Ergebnisses sofern ein ausreichend gutes Ergebnis erzielt wurde oder ein Zeitlimit oder die max. Anzahl der Iterationen erreicht ist. Andernfalls Rücksprung zu Ia) zum Fortsetzen der Optimierung.

A) Zugriffe auf Modelle zur Abfrage von Tool- und Werkstückmodellen

- B) Vorgeschlagene Prozessbahnen oder –Punkte
- C) Zugriff auf Modell zur Abfrage von Tool- und Werkstückmodellen
- D) Variierte und optimierte Bahnen (Tool kollisionsfrei, ggf. geglättet, lokale Überschreitung (einfacher geometriescher) Parameter oder Anpressdruck etc. vermeiden)
- E) Bahnen zur Verwendung/Validierung in der Bewegungssimulation
- F) Zugriff auf Modelle des Werkstücks, der Zelle, der Roboter und der Tools
- G) Rückgabewert „Kollision“ oder „nicht erreichbar“ (inkl. Wo) ODER Rückgabe genauer Bahn/Bewegung (wie vom Roboter gemäß Simulation ausführbar, mit Geschwindigkeiten der Bahn) und Ausführungsdauer
- H) Überspringen der Prozesssimulation wenn G kein oder kein hinreichend gut bewertetes Ergebnis ist (z.B. zu hohe Ausführungsdauer, Anteil der zu verworfenen Bahnen zu hoch)
- I) Übergabe der hinreichend gut bewerteten genau ermittelten/simulierten Bahn(en) an die Prozesssimulation.
- J) Zugriff auf Modelle des Werkstücks, Tools und Prozess (keine Zelle, kein Roboter)
- K) Simulationsergebnis des Prozesses: Beschreibung des Werkstücks bzw. seiner Oberfläche angereichert mit der Information über lokale Prozessgüte/Merkmale (z.B. gereinigt/nicht gereinigt, Schichtdicke,... je Oberflächenteil)
- L) Ergebnis und Bewertung
- M) Rücksprung zur Optimierung oder
- g) Abbruch bei Erreichen eines ausreichend guten Ergebnisses oder dem Überschreiten von Zeitlimits oder der maximalen Anzahl an Iterationen.

Fig. 3 zeigt eine Beispielhafte Realisierung des Schemas für die Version „Line Tracking“, bei der die Bearbeitung eines mit einem Förderer bewegten Werkstückes erfolgen soll.

1. Initiale „quasistatische“ Bestimmung / Planung (und ggf. Optimierung) der Prozessbahnen und/oder –Punkte ohne Berücksichtigung der zusätzlichen Dimension (Bewegung des Werkstückes). Vorteilhaft wenn für das Werkzeug bereits kollisions-

frei. Alternativ können auch existierende oder klassisch programmierte Programme eingelesen und (in gereichte Pfade und Punkte in Werkstückkoordinaten) konvertiert werden.

2. Eingabe oder Einlesen der Information über die zusätzliche Komplexität/Dimension (Bewegungsgeschwindigkeit des Förderers / Werkstücks und frühester Start und spätester Endzeitpunkt des Ablaufs).

3. Vorverarbeitung. Ermittlung der „Fenster“ oder „Korridore“ über welche die (einzelnen) Prozessbahnen oder Punkte (aus 1) durch den Roboter (ohne Konfigurationswechsel) erreichbar sind.

4a. Grobplanung: Suche nach möglichen Pfad-Reihenfolgen und groben Startwerten die die Vorgaben aus (2) erfüllen und deren Abarbeitung in einer Schleife. In einer Vorteilhaften Realisierung verwendet die Grobplanung die Bewertung der Ergebnisse der Feinplanung aus vorhergehenden Durchläufen der Schleife.

5. (Fein-)planung und Optimierung von Prozess und Bahnen/Punkten (Prozess- und Zwischenbewegungen) und deren genaues Timing innerhalb der (in 4a) vorgegeben Reihenfolge. Mögliche Kriterien für die Optimierung kommen aus der Simulation (6) und können sein: Prozessergebnis (Qualität, d.h. Anteil der relevanten Werkstückoberfläche bei der das Prozessergebnis innerhalb der Toleranzen liegt), Stabilität (d.h. Maximierung der verbleibenden Toleranzen bei den Zeitfenstern), Taktzeit. Sofern eine Lösung erzielt wird, werden die Abläufe im Plan um genaue Bewegungen und Schaltpunkte etc. ergänzt d.h. werden zu einem Metaprogramm.

6. Validierung der Planungsergebnisse aus 5: (Echtzeit-)Simulation der genauen gleichzeitigen Bewegung des Roboters und des Werkstücks auf Ausführbarkeit (und Zeit) und Simulation des aus der gleichzeitigen Bewegung resultierenden Prozessergebnis als (individuelles und/oder kumuliertes) Ergebnis (gereinigt, inspiziert, Schichtdicke etc.) je Segment der Oberfläche des Werkstücks.

4b Abspeichern des Ergebnisses. Abspeichern des Meta-Programms wenn eine Lösung gefunden wurde. Ende der Schleife: Rücksprung nach 4a wenn noch nicht alle Fälle abgearbeitet und keine ausreichend gute Lösung gefunden wurde. Sofern bereits alle Fälle (aus 4a) abgearbeitet sind und überhaupt keine Lösung (Variation) gefunden wurde, erfolgt ein Rücksprung zu 1. Dabei wird gleich akkumulierte Information (aus der Statistik) über die kritischen und unkritischen Pfade oder Regionen

(einzeln oder wechselseitig) mit übergeben und können bei der „quasistatischen Generierung“ (1) zur Analyse und Problemlösung verwendet (oder visualisiert) werden.

7. Umwandeln des Metaprogramms in auf den spezifischen Robotern ausführbare Programme.

Fig. 4 zeigt ein Beispiel für eine Realisierung einer kooperativen Bearbeitung mit einem Roboter der das Prozesswerkzeug führt und einem (oder mehreren) programmierbaren Aktuatoren die das Werkstück führen (z.B. Roboter oder Dreh-Schwenk-Neigetisch).

1. Bestimmung / Planung (und ggf. Optimierung) der Prozessbahnen und/oder -Punkte ohne Berücksichtigung der zusätzlichen Dimension (Bewegung des Werkstückes durch einen anderen Aktuator). Vorteilhaft ist wenn die Prozessbahnen bereits kollisionsfrei sind. Auf Kollisionsfreiheit ist zu diesem Zeitpunkt nur das Werkzeug geprüft, da die Relativposition von Roboter und Werkstück ja durch den zweiten Aktuator bestimmt werden der bis zu diesem Zeitpunkt hier noch ignoriert ist. Alternativ können auch klassisch programmierte Programme eingelesen und konvertiert werden.

2. Einlesen oder Eingabe relevanter Information über die zusätzliche Komplexität/Dimension. Dies sind insbesondere Einschränkung der Lage/Orientierung des Werkstücks welches über den frei programmierbaren Aktuator bewegt oder positioniert werden kann durch weitere Bedingungen (z.B. „gerade halten“).

3. Vorverarbeitung: Ermittlung der möglichen Werkstückpositionen bei denen die einzelnen Prozessbahnen und Prozesspunkte (aus 1) durch den Roboter erreichbar sind und die Bedingungen von 2 erfüllt sind. Letzteres erlaubt den Suchbereich zu reduzieren, d.h. Konfigurationen zu verwerfen die zu unerlaubten Positionen führen.

4. a) Grobplanung der möglichen Werkstücklagen (aus 3) und Reihenfolgen der Pfade (aus 1) mit dem Ziel bereits vorteilhafte Reihenfolgen an die nachfolgende Feinplanung und Optimierung zu übergeben. Kriterium dafür ist die Minimierung der Änderungen in aufeinanderfolgenden Werkstückpositionen. Die Abarbeitung erfolgt in einer Schleife. In einer vorteilhaften Realisierung verwendet die Grobplanung

früher bewerteter Grobpläne, d.h. die Ergebnisse der Feinplanung aus vorhergehenden Durchläufen der Schleife.

5. (Fein-)Planung, und Optimierung von Prozess und Pfaden (Prozess- und verbindende Zwischenbewegungen) . Diese Feinplanung generiert Bewegungsabläufe für den Aktuator der das Werkstück bewegt und den der das Werkzeug führt unter Berücksichtigung aller Freiheitsgrade und übergibt diese an die Validierung (6). Wichtiger Unterschied zwischen Feinplanung (5) und Grobplanung (4a): Die Lage des über den Aktuator bewegten Werkstücks kann auch während der Ausführung einer Prozessbahn kontinuierlich bewegt werden (um die Erreichbarkeit zu optimieren oder Umorientierungen zwischen Prozessbewegungen zu minimieren). Kriterien für die Optimierung kommen aus der Simulation (6) und können sein: Prozessergebnis (Qualität), Taktzeit, Minimierung der Umorientierung, etc. Sofern eine Lösung erzielt wird, sind die Abläufe im Plan um genaue Bewegungen und Schaltpunkte etc. ergänzt d.h. sind bereits ein Metaprogramm.

6. Validierung der Feinplanung aus 5: (Echtzeit-) Simulation der genauen zeitgleichen Bewegung v. Prozessroboter und Aktuators der das Werkstück bewegt und des daraus resultierenden Prozessergebnisses (Prozessergebnis (OK, bzw. Über-/Unterschreitung) je Segment oder Teil der Oberfläche des Werkstückes).

4b Schleifenende, Abspeichern des Ergebnisses. Rücksprung nach 4a wenn noch nicht alle Fälle abgearbeitet wurden und keine hinreichend gute Lösung gefunden wurde. (mit Fall, Bewertung und Angabe vom Grund für das Scheitern). Wegen der vielen Kombinationen ist die Beendigung des Loops beim ersten hinreichendem Ergebnis vorteilhaft. Sofern bereits alle Fälle (aus 4a) abgearbeitet sind und keine Lösung (Variation) gefunden wurde erfolgt ein Rücksprung zu 1. Dabei können akkumulierte Information über die (lt. Statistik) kritischen und unkritischen Pfade, Regionen und Lagen (einzeln oder wechselseitig) mit übergeben und bei der „quasistatischen Generierung“ (1) zur Analyse und Problemlösung verwendet (oder visualisiert) werden.

7. Umwandeln des Metaprogramms in ein auf dem spezifischen Roboter(typ) und kooperierenden Aktuator ausführbares Programm.

Fig. 5 zeigt beispielhaft eine Realisierung des Schemas für die Version „Prozessabsicherung“, also die Berücksichtigung hinsichtlich Störgrößen, statistischer Schwankungen u. dgl.

1. Bestimmung / Planung und ggf. Optimierung der Prozessbahnen und/oder – Punkte ohne Berücksichtigung der zusätzlichen Dimension (Variation / Störungen). Alternativ können auch klassisch programmierte Programme eingelesen und konvertiert werden
2. Eingabe oder Einlesen der Information über die zusätzliche Varianzen / Störungen wie z.B. der Schwankungsbreite der Lage des Werkstücks oder der Prozessparameter.
3. Vorverarbeitung Analyse der Störungen und Aufbau eines Verteilungsraums der Störungen welcher Abhängigkeiten, gegenseitigen Ausschluss etc. nutzt um nicht einnehmbare Zustände zu eliminieren.
4. a) Schleife über die in 3 spezifizierten Störungen/Variationen. In einer vorteilhaften Realisierung ist die Abtastung des Zustandsraumes nicht gleichmäßig sondern anhand der Ergebnisse vorheriger Durchläufe gesteuert, d.h. eine Suche die z.B. einen Region-Growing ähnlichen Ansatz verwendet, oder eine gesteuerte Abtastung die z.B. Stetigkeiten zwischen getesteten Variationen ausnutzt um weniger Variationen untersuchen zu müssen.
5. Test bereits gebildeter Pläne/Metaprogramme: In einer vorteilhaften Realisierung werden zuerst die bereits für vorher geprüfte Fälle optimierte Pläne unverändert (!) auf Ihre Eignung für den aktuellen Fall getestet, d.h. ob die / welche der geplanten Bewegungen auch in dem Fall mit dem Roboter ausführbar sind und b) ob ein ausreichendes Ergebnis erreicht wird. Mögliche Kriterien für die Bewertung kommen aus der Simulation (5b) und können sein: Ausführbarkeit, und Ausführungszeit, Prozessergebnis (Qualität), Stabilität. Wenn das existierende getestete Metaprogramm passt, d.h. wenn der Anteil an der relevanten Oberfläche der innerhalb der Parameter liegt bzw. die Qualitätskriterien erfüllt über der definierten Schwelle oder Schranke liegt, wird der aktuelle Zustand (die aktuelle Variation) dem Cluster zugeordnet und weiter gesprungen zu 4b.
- 5b) Validierung des bestehenden Metaprogramms aus 5a: Simulation der Bewegung des Roboters (genaue Bahn und Geschwindigkeit, Ausführbarkeit, Erreichbar-

keit, ggf. Zeitbedarf) gemäß Metaprogramm aus 5. Simulation des resultierenden Prozessergebnisses für die aktuell zu untersuchende Variation (d.h. das Ergebnis der Wechselwirkung zwischen Prozess und Werkstückoberfläche wird simuliert und je Oberflächensegment (z.B. Mesh-Triangle oder Textur) abgespeichert.

6. Feinplanung und Optimierung von Prozess und Pfaden durch Variation und Optimierung des initialen Plans (aus 1) (oder den eines dem aktuellen ähnlichen Zustands (aus 5)). Mögliche Kriterien für die Optimierung kommen aus der Simulation (7) und können sein: Ausführbarkeit, und Ausführungszeit, Prozessergebnis (Qualität), Stabilität. Speichern eines in 7 bestätigten neuen Plans/Metaprograms in einem neu angelegten Zustand-Cluster.

7. Validierung der Planungsergebnisse aus 6: Simulation der Bewegung des Roboters (genaue Bahn und Geschwindigkeit, Erreichbarkeit, ggf. Zeitbedarf) gemäß Plan. Simulation des resultierenden Prozessergebnisses (je Oberfläche)

8. 4b: Abspeichern des Ergebnisses (Meta-Programm) wenn eine Lösung gefunden wurde. Rücksprung zu (4a) wenn noch nicht alle Fälle abgearbeitet wurden. Ende und weiter zu (8) wenn alle Zustände abgearbeitet sind und für jeden Fall eine Lösung gefunden wurde. Sofern es Fälle / Variationen (an Störungen) gibt für die keine Lösung gefunden werden konnte, erfolgt ein Rücksprung zu (1).

9. Optional Feinplanung und Optimierung über alle Varianten je Cluster wie in (6).

10. Optionale Validierung wie in (7) (ggf. Optional Worst-case simulation)

11. Umwandlung der Metaprogramme in ausführbare Roboterprogramme. Abspeichern zusammen mit dem Zustandsbereich in dem anzuwenden sind.

a3: Plan oder Liste von Prozessbahnen / -Punkten

b3: Angaben über Schwankungsbreiten einzelner Parameter und Abhängigkeiten

c3: (Zu überprüfender) Zustandsraum der Störungen/Variationen

d3: Plan zusammen mit Information über die aktuell zu untersuchende Variation (Störung)

e3: zu validierender bestehender Plan (unverändert)

f3: Ergebnis der Validierung: verworfene (inkl. Ursachen) und ausführbare Prozessbahnen inkl. ausführbarer Roboterbewegung sowie das aus diesen resultierende Prozessergebnis

g3: Sofern (ein) bereits für eine früher geprüfte Variation/Störung optimierte(r) Plan(e) zufriedenstellende Ergebnisse liefert: passende(s) Plan/Meta-Programm(e) oder Verweis darauf

h3: Sofern keines der bisherigen Pläne/Metaprogramme passt: Plan (aus 1) zusammen mit Information über die aktuell zu untersuchende Variation (Störung)

i3: zu validierender optimierter Plan

j3: Ergebnis der Validierung: verworfene (inkl. Ursachen) und ausführbare Prozessbahnen inkl. ausführbarer Roboterbewegung sowie das aus diesen resultierende Prozessergebnis

k3: optimiertes/adaptiertes Metaprogramm oder Information das keine gültige Lösung

l3: Rücksprung zu (4a) solange nicht alle Variationen der Schleife abgearbeitet sind: Information über Erfolg/nicht Erfolg für die aktuell getestete Variation.

m3: Rücksprung zum quasi-idealen Bestimmung (1) im Fall dass nicht für alle Fälle eine Lösung gefunden wurde, inklusive Information über Fortschritt der bisherigen Lösungssuche: z.B. Variationen für die eine/keine Lösung gefunden wurde. Erreichte oder kritische Zustände, Pfade,...

n3: Information über geclusterte (zusammengefasste) Meta-Programme und Zustände

o3: optimiertes Metaprogramm (für die Simulation)

p3: Ergebnis der Validierung: verworfene (inkl. Ursachen) und ausführbare Prozessbahnen inkl. ausführbarer Roboterbewegung und Prozessergebnis

q3: Für definierte Störungen optimierte Meta-Programm(e), ggf. geclustert.

Für den Fall das nicht alle Störungen mit einem Programm abgedeckt werden können ist folgende Vorgangsweise vorteilhaft. Fig. 6 zeigt eine Programmauswahl bei der Prozessabsicherung für den Fall das die Bandbreite an Störungen nicht mit einem einzigen Programm abgedeckt werden kann.

1: Messung der Störung,

2. Auswahl eines Programmes, welches die Ausprägung der Störung abbildet,

3. Optionale Nachoptimierung des (Meta-) Programs aus der Auswahl für den genau gemessen Wert der Störung. Unter Meta-Programm wird ein Ablauf, d.h. eine

Abfolge präzise definierter und vom Roboter ausführbarer Bewegungen in einer Controller-unabhängigen Sprache verstanden. Das Metaprogramm lässt sich in ein auf einem Robotersteuergerät ausführbares Programm umwandeln.

4. Umwandlung in den spezifischen Roboter-Dialekt Falls nicht schon im Dialekt gespeichert.

a1 Gemessene Störung

b1 Ausgewähltes Metaprogramm

c1 Optional: Metaprogramm neu optimiert für aktuell gemessene Störung

Stiborlin

002595

Patentanwälte
Dipl.-Ing. Helmut Hübscher
Dipl.-Ing. Karl Winfried Hellmich
Dipl.-Ing. Friedrich Jell
Spittelwiese 7, A 4020 Linz

(37 006) hel

Patentansprüche:

1. Verfahren zur automatisierten Programmierung und Optimierung von robotischen Arbeitsabläufen mit folgenden Verfahrensschritten
 - Programmieren des Arbeitsablaufes für einen virtuell vereinfachten Fall, insbesondere an einem lagefesten Werkstück,
 - Aufspannen und Einschränken eines Suchraumes unter Berücksichtigung von Informationen über eine zusätzliche Komplexität/Dimension, insbesondere Berücksichtigung von Bewegungen von Werkstück bzw. Aktuatoren und Festlegen eines zeitlichen Rahmens für Start und Endzeitpunkt des Programmes bzw. festlegen von erlaubten und Verbotenen Bereichen für Werkstück und Aktuatoren,
 - Suchen nach Startplänen/Lösungen für eine Feinplanung in einer Grobplanung und festlegen der Reihenfolge, in der die Feinplanungen in einer Schleife abgearbeitet werden,
 - optimieren, validieren oder verwerfen von in der Grobplanung gefundenen Lösungen durch eine Feinplanung,
 - abspeichern gefundener Lösungen und
 - übersetzen ausreichend guter Lösungen in ein oder mehrere Roboterprogramme.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass je Prozesspfad oder -Punkt des programmierten Arbeitsablaufes zur Einschränkung des Suchraumes erlaubte Bereiche für einen vom Ideal abweichenden Versatz des zu bearbeitenden Werkstücks ermittelt werden.
3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass in der Grobplanung prinzipiell ausführbare Reihenfolgen und Richtungen der Pfade/Bewegungen ermittelt und die ermittelten Reihenfolgen nacheinander in einer Schleife auf Ausführbarkeit geprüft werden, wobei die Reihenfolgeplanung vorzugsweise die Ausführungsdauer für jeden Prozesspfad oder -punkt errechnet.
4. Verfahren nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, dass in der Feinplanung die in der Grobplanung gefundenen Lösungen verfeinert und hinsichtlich zeitlicher und ergebnisorientierter Vorgaben validiert werden, wobei die geplanten zeitgleichen Werkstück- und Roboterbewegungen zur Validierung vorzugsweise simuliert und nicht ausführbare Ergebnisse als Optimierungskriterium für die Feinplanung verwendet werden.
5. Verfahren nach einem der Ansprüche 1 bis 4, dadurch gekennzeichnet, dass das Ergebnis der Prozesssimulation Oberflächenpunkten des Werkstücks zugeordnet wird und die Optimierung der Positionen des Werkzeuges und damit des Roboters in Bezug zur Oberfläche derart erfolgt, dass der Anteil vorgegebene Qualitätsmerkmale einhaltender Oberflächenpunkte maximiert wird.
6. Verfahren nach einem der Ansprüche 1 bis 5, dadurch gekennzeichnet, dass die Planung über Prozess-Pfade oder -Punkte erfolgt.
7. Verfahren nach einem der Ansprüche 1 bis 6, dadurch gekennzeichnet, dass zur Reduzierung des Suchraumes Bereiche erlaubter Werkstückpositionen ermittelt werden in denen jeweils ein einzelner Pfad mit dem Prozess-Roboter ausgeführt wird an den das Werkstück auch durch einen kooperierenden Aktuator positioniert

werden kann, und/oder dass über Vorgaben, insbesondere über die Lage des Werkstücks der Zustandsraum des kooperierenden Aktuators verkleinert wird.

8. Verfahren nach einem der Ansprüche 1 bis 7, dadurch gekennzeichnet, dass durch Kombinationen von Werkstückpositionen und Reihenfolgen für die einzelnen Prozesspfade oder Punkte nach möglichen Lösungen gesucht wird, wobei als Optimierungskriterium die Minimierung der Anzahl an Umoorientierungen von Werkzeug und/oder Werkstück, verwendet wird.

9. Verfahren nach einem der Ansprüche 1 bis 8, dadurch gekennzeichnet, dass zur Berücksichtigung von Schwankungen oder Störungen im Arbeitsablauf zunächst erwartete Schwankungen vorab analysiert werden, wonach über alle sinnvoll möglichen Zustände ein minimaler Zustandsraum aufgespannt wird.

10. Verfahren nach einem der Ansprüche 1 bis 9, dadurch gekennzeichnet, dass für verschiedene Störungen und Schwankungen Lösungen ermittelt und diese Lösungen in einem Speicher abgelegt werden, wobei die Abfolge und die Abtastung der in der Schleife untersuchten Variationen/Störungen vorzugsweise anhand der Ergebnisse bereits erfolgter Berechnungen gesteuert wird.

11. Verfahren nach Anspruch 10, dadurch gekennzeichnet, dass solange Lösungen ermittelt werden, bis für jede Variation/Störung ein Lösung existiert, wobei die Randlösungen aneinander angrenzender Bereiche angeglichen werden.

12. Verfahren nach einem der Ansprüche 9 bis 11, dadurch gekennzeichnet, dass bei einer Echtzeitausführung des Verfahrens an einer Vorrichtung zunächst die Variation/Störung, beispielsweise Lageabweichung, gemessen und die entsprechende und bereits vorab ermittelte Lösung ausgewählt und das entsprechende Programm aus einem Speicher aufgerufen wird.

13. Verfahren nach Anspruch 12, dadurch gekennzeichnet, dass in Abhängigkeit der gemessenen Variation/Störung ein gespeichertes und im Mittel bestgeeignetes

002598

Programm aufgerufen und anschließend gegebenenfalls für die genau gemessene Störung / Parameter optimiert wird.

14. Datenträger, dadurch gekennzeichnet, dass auf ihm ein Programm zur Ausführung des in den Ansprüchen 1 bis 13 beanspruchten Verfahrens gespeichert ist.

Linz, am 12. März 2010

Convergent Information Technologies GmbH

durch:

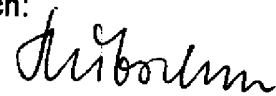
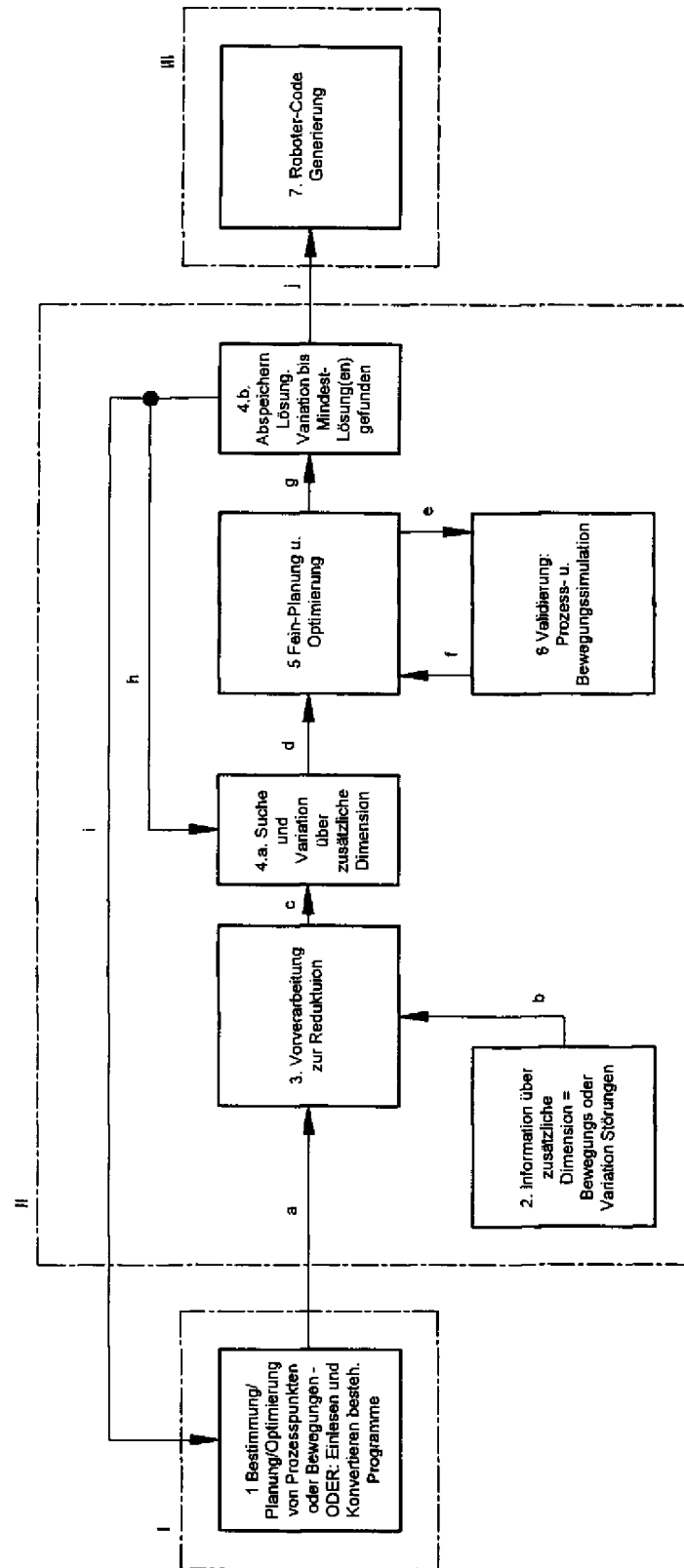


Fig. 1



000000

Fig. 2

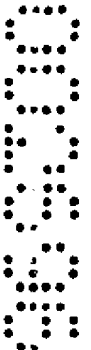
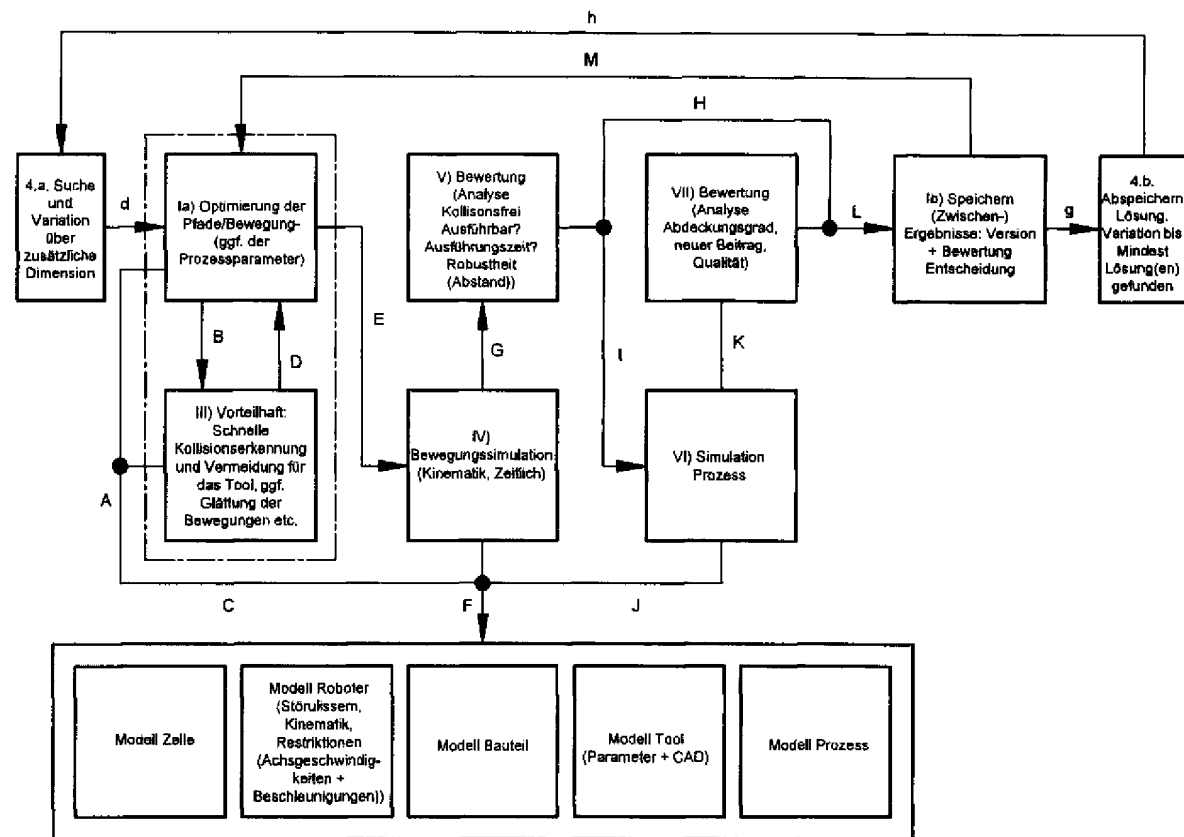
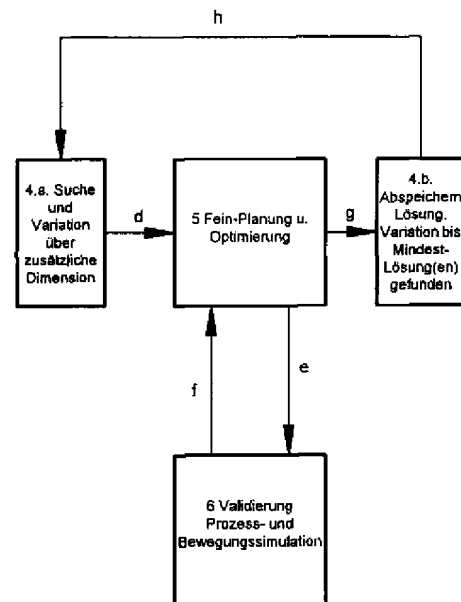
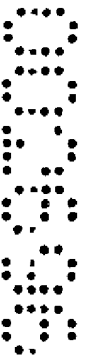
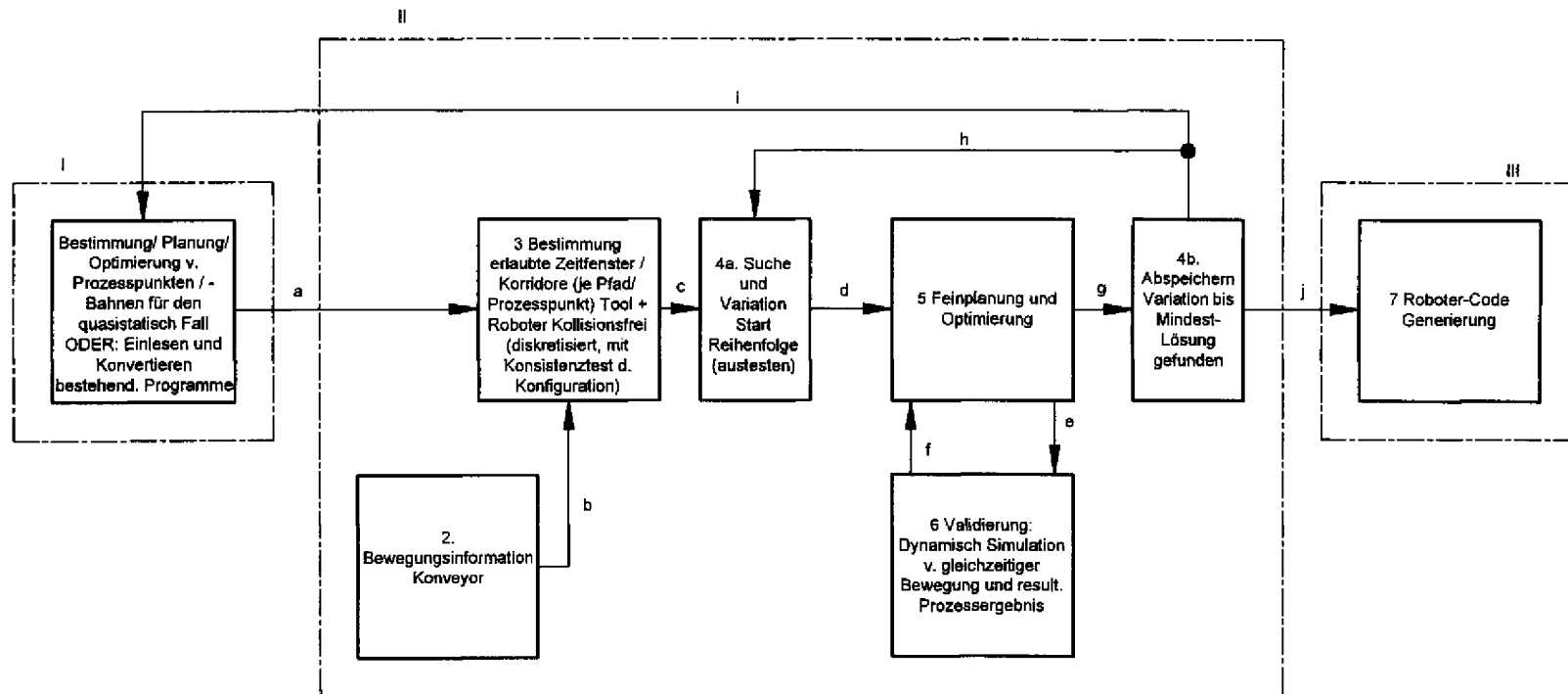


Fig. 3



A 5x5 grid of dots forming the number 8. The dots are arranged in a pattern that resembles a digital display of the digit 8.

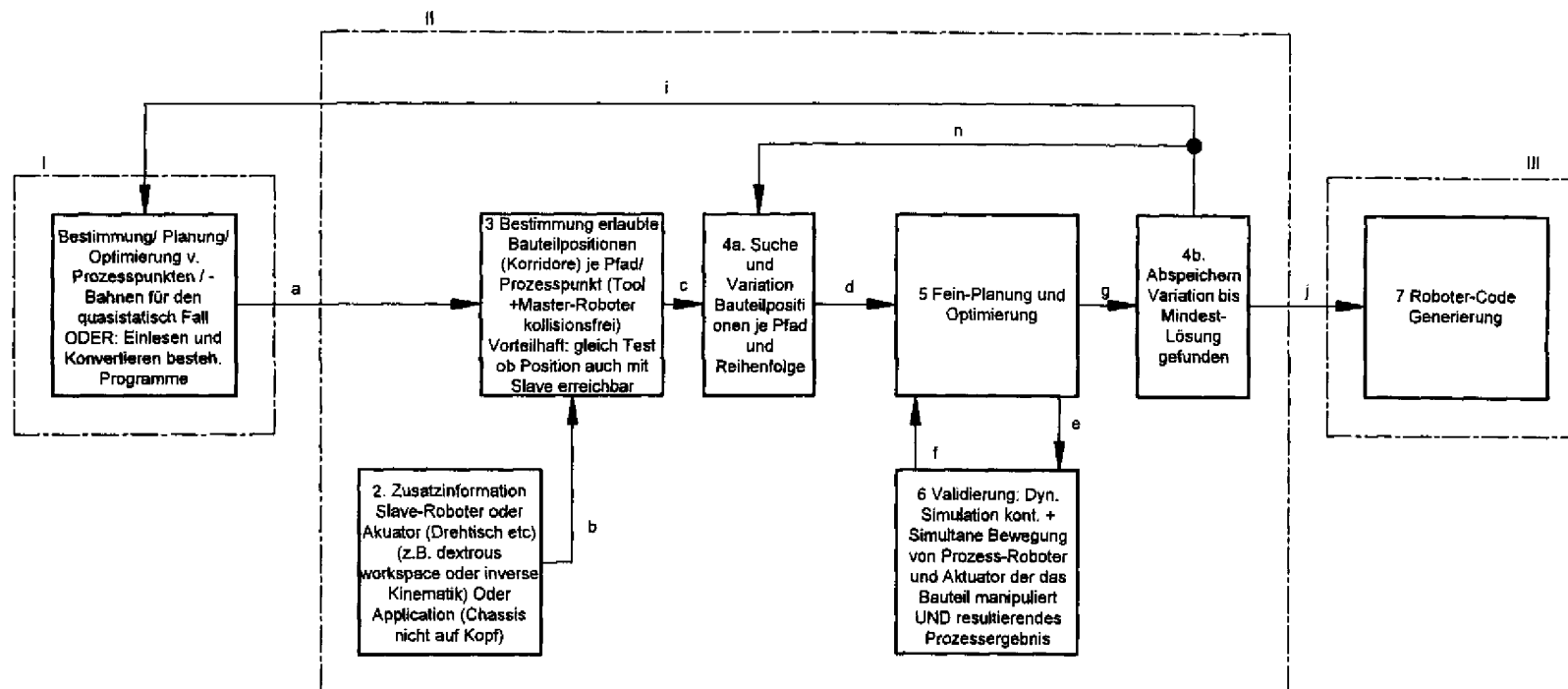


Fig. 5

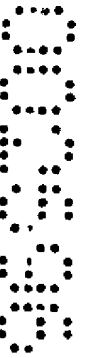
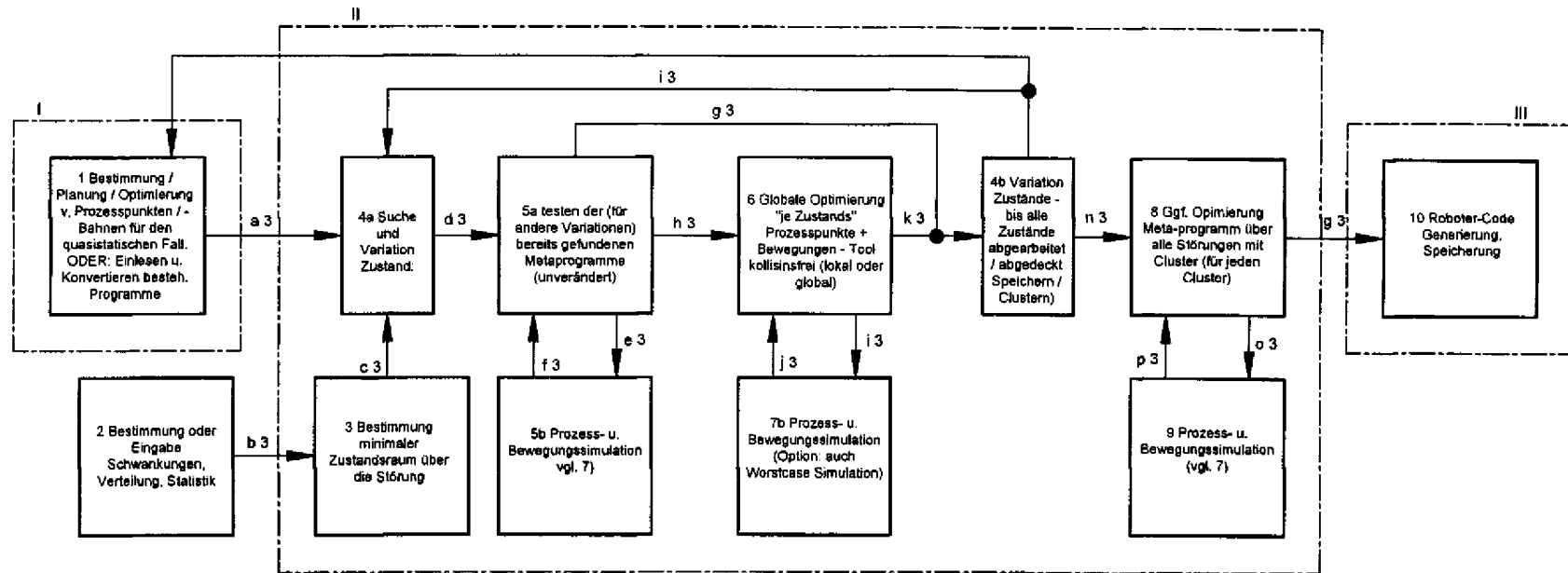


Fig. 6

