



(10) **DE 10 2012 112 900 A1** 2013.06.27

(12)

## Offenlegungsschrift

(21) Aktenzeichen: **10 2012 112 900.2**

(22) Anmeldetag: **21.12.2012**

(43) Offenlegungstag: **27.06.2013**

(51) Int Cl.: **B25J 9/18 (2013.01)**

(30) Unionspriorität:

**61/579,288**                      **22.12.2011**      **US**

(71) Anmelder:

**Fanuc Robotics America Corp., Rochester Hills,  
Mich., US**

(74) Vertreter:

**Meissner, Bolte & Partner GbR, 80538, München,  
DE**

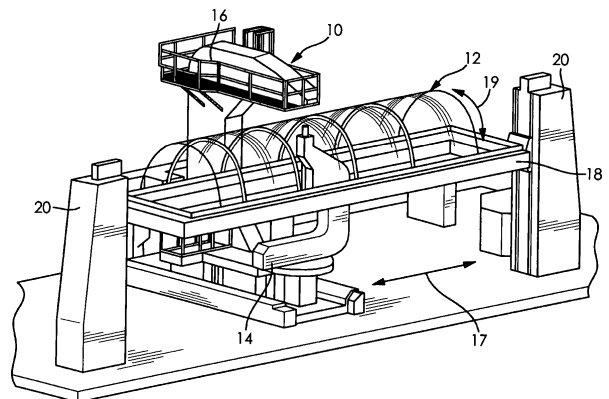
(72) Erfinder:

**Yanagita, Akihiro, Orion Township, Mich., US;  
Lee, Eric, Bloomfield Hills, Mich., US; McGee, H.  
Dean, Rochester Hills, Mich., US; Tsai, Jason,  
Bloomfield Hills, Mich., US**

**Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen**

(54) Bezeichnung: **Numerische Steuerungsprogrammausrichtung durch Roboter**

(57) Zusammenfassung: Ein System und Verfahren zur Umwandlung eines Werkzeugmaschinenprogramms in NC-Programmiersprache, die es zulassen, dass eine Robotersteuerung das Programm ausführt. Eine Robotersteuerung wandelt das NC-Programm entsprechend einer Umwandlungskonfigurationstabelle in eine Robotersprache um und nutzt die umgewandelte Sprache als Daten eines Pseudoprogramms, die intern in einem Datenspeicher innerhalb der Robotersteuerung gespeichert werden. Jeder M-Code (Mischcode) in dem NC-Programm wird als Aufruf eines Unterprogramms ausgeführt, das die Robotersprache verwendet. Der Inhalt der Unterprogramme kann durch den Anwender frei definiert und programmiert werden, und kann deshalb für die spezielle Anwendung kundenspezifisch gestaltet werden.



**Beschreibung**

## QUERVERWEIS AUF VERWANDTE ANMELDUNG

**[0001]** Diese Anmeldung erhebt den Anspruch auf den Nutzen der am 22. Dezember 2011 eingereichten vorläufigen US-Patentanmeldung, Serien-Nr. 61/579, 288.

## GEBIET DER ERFINDUNG

**[0002]** Diese Erfindung betrifft den Betrieb von Werkzeugmaschinen durch Roboter, die Programme von numerischen Steuerungen ausführen.

## HINTERGRUND DER ERFINDUNG

**[0003]** Bekannt ist, Werkzeugmaschinen durch ein „numerisches Steuerungsverfahren (NC) automatisch zu bedienen, indem ein programmierter Digitalrechner oder eine speicherprogrammierbare Steuerung (PLC) genutzt werden, die mit der Werkzeugmaschine verbunden sind, um eine Werkzeugmaschine mit CNC-Steuerung zu bilden. Ein Anwenderprogramm (Teileprogramm) wird in einer NC-Programmiersprache (G-Code-Programm) zum Betreiben der CNC-Werkzeugmaschine bereitgestellt, um einen Herstellungsprozess durchzuführen. Die Programmierung wird typischerweise in auf Standards basierenden Programmiersprachen wie FBD (Funktionsbausteinsprache), LD (Kontaktplan), ST (Strukturierter Text, der Pascal-Programmiersprache ähnlich), IL (Anweisungslisten-Sprache, der Assemblersprache ähnlich) und SFC (Sequentielle Funktionsgrafik) vorgenommen. Unterschiede bei der I/O Adressierung, Speicherorganisation und Befehlssätzen bedeuten jedoch, dass die Programme zwischen unterschiedlichen Werkzeugmaschinenherstellern durchaus nicht perfekt austauschbar sind.

**[0004]** Es ist außerdem bekannt, Werkzeugmaschinen durch einen Roboter zu bedienen. Typisch ist, dass Roboter durch Computer gesteuert werden, die für den Hersteller von Robotern gesetzlich geschützte Softwareprogramme ausführen. Die Unterschiede zwischen den NC-Programmiersprachen und den Roboter-Programmiersprachen verhindern, dass die Robotersteuerungen eine NC-Programmiersprache ausführen, die für eine CNC-Werkzeugmaschine geschrieben ist.

**[0005]** Typischerweise erfordert eine CNC-Werkzeugmaschine eine kostspieligere mechanische Konstruktion zur Erlangung der gewünschten Funktionsfähigkeit als eine Roboter-Werkzeugmaschine. Seitdem es üblich geworden ist, die NC-Programmiersprachen zu verwenden, fordern Benutzer für viele Anwendungen von Werkzeugmaschinen eine NC-Programmiersprache, um die Werkzeugmaschinen zu programmieren. Leider können Roboter bei sol-

chen Nachfragen nicht genutzt werden, weil Robotersteuerungen die Befehle in diesen NC-Programmen nicht übersetzen und ausführen können. Andererseits sind NC-Steuerungen wegen der speziellen Beschaffenheit einer Robotersteuerung (wie beispielsweise komplexe Arithmetik der Kinematik, usw.) nicht ohne weiteres imstande, Roboter zu steuern.

**[0006]** Es gibt einige rechnerunabhängig arbeitende Umwandlungssysteme, die aus einem NC-Programm ein Roboterprogramm erzeugen können. Jedoch wandeln solche Umwandlungssysteme im Allgemeinen nur die Bewegungen der mechanischen Konstruktion um, sind aber nicht imstande, den „M-Code“ (Mischcode) umzuwandeln, um für die Verarbeitung einer maschinenspezifischen Anwendung zu sorgen. Weil M-Codes durch die Anwender (Werkzeugmaschinenbauer) für die spezielle Werkzeugmaschine, die sie bauen, definiert werden, wird die Robotersteuerung nicht imstande sein, einen M-Code ohne Kenntnis davon, wie er verarbeitet werden sollte, auszuführen.

## ABRISS DER ERFINDUNG

**[0007]** Gemäß der vorliegenden Erfindung kann durch die Verwendung von Roboterarmen eine Maschine mit einer weniger kostspieligen mechanischen Konstruktion hergestellt werden. Unter Verwendung des folgenden Verfahrens kann die Robotersteuerung ein in der NC-Programmiersprache geschriebenes Teileprogramm ausführen.

Schritt 1) Die Robotersteuerung wandelt das NC-Programm um in eine Robotersprache und nutzt sie als die intern im Datenspeichersystem in der Robotersteuerung gespeicherten Pseudoprogrammdateien.

Schritt 2) Jeder M-Code (Mischcode) wird ausgeführt als Aufruf eines Unterprogramms, das die Robotersprache verwendet. Der Inhalt der Unterprogramme kann durch den Anwender (Systemintegrator) frei definiert und programmiert und deshalb für die spezielle Anwendung kundenspezifisch gestaltet werden.

Schritt 3) Einige M-Codes sind mit einer Bewegung verknüpft, wobei der M-Code zu dem Zeitpunkt ausgeführt wird, wo die Bewegung beendet ist. In diesen Fällen ruft die Robotersteuerung das entsprechende Unterprogramm in der Robotersprache auf, wenn die Bewegung beendet ist.

Schritt 4) Eine I/O Schnittstelle muss ein Programm auswählen und abarbeiten sowie den Zustand des Programmablaufs überwachen können. Weil die Zeilennummer eine Eins-zu-eins Entsprechung zwischen dem originalen NC-Programm und dem Pseudoroboterprogramm aufweist, wird das Erzeugen einer Benutzerschnittstelle erleichtert, um die gegenwärtig ausgeführte Programmzeile darzustellen.

**[0008]** Das Verfahren zur Steuerung eines Roboters umfasst die Schritte:

Speichern einer NC-Programmiersprache in einem Massenspeichergerät, das mit einer Zentraleinheit verbunden ist, wobei die NC-Programmiersprache angepasst ist, um durch eine Werkzeugmaschine mit CNC-Steuerung zur Durchführung eines Herstellungsprozesses ausgeführt zu werden; Bedienen der Zentraleinheit zum Umwandeln der NC-Programmiersprache in eine Roboter-Programmiersprache basierend auf einer vorher festgelegten Tabelle für Umwandlungskonfigurationen, die diskrete Konfigurationsanweisungen aufweist, um jeden Positionsbeehl der NC-Sprache in einen Positionsbeehl der Robotersprache umzuwandeln, und jeden Mischcodebeehl der NC-Sprache in ein Unterprogramm der Robotersprache zu konvertieren; Speichern der Robotersprachen-Positionsbeehle und der Robotersprachen-Unterprogramme als Roboter-Programmiersprache in dem Massenspeichergerät; und Ausführen der Roboter-Programmiersprache mit einer mit dem Massenspeichergerät und dem Roboter verbundenen Robotersteuereinheit bei einer Laufzeit des Roboters zur Durchführung des Herstellungsprozesses.

#### BESCHREIBUNG DER ZEICHNUNGEN

**[0009]** Die oben erwähnten sowie andere Vorteile der vorliegenden Erfindung erschließen sich dem Fachmann ohne weiteres aus der folgenden ausführlichen Beschreibung einer bevorzugten Ausführungsform bei Betrachtung angesichts der begleitenden Zeichnungen, in denen:

**[0010]** [Fig. 1](#) ist die perspektivische Darstellung einer CNC-Werkzeugmaschine im Stand der Technik, die zur Ausführung von Schweißungen an einer Sektion eines Flugzeugrumpfes ein NC-Programm verwendet;

**[0011]** [Fig. 2](#) ist die perspektivische Darstellung einer robotergesteuerten Werkzeugmaschine zur Ausführung von Schweißungen an der Sektion des in [Fig. 1](#) gezeigten Flugzeugrumpfes;

**[0012]** [Fig. 3](#) ist ein Blockdiagramm der Steuereinheit des in [Fig. 2](#) gezeigten Roboters;

**[0013]** [Fig. 4](#) ist das Ablaufdiagramm einer ersten Stufe einer Programmumwandlung von einer NC-Programmiersprache in die Roboter-Programmiersprache bei Einschaltung.

**[0014]** [Fig. 5](#) bis [Fig. 7](#) sind Ablaufdiagramme einer zweiten Stufe der Programmumwandlung bei jeder Programmumwandlung zum Umwandeln des originalen NC-Programms in das interne Roboterprogramm; und

**[0015]** [Fig. 8](#) ist ein Blockdiagramm der Steuereinheit für die in [Fig. 1](#) dargestellte CNC-Werkzeugmaschine, die zum Bedienen der in [Fig. 2](#) gezeigten, robotergesteuerten Werkzeugmaschine modifiziert wird.

#### AUSFÜHRLICHE BESCHREIBUNG DER ERFINDUNG

**[0016]** Die folgende ausführliche Beschreibung und angefügte Zeichnungen beschreiben und veranschaulichen verschiedene beispielhafte Ausführungsformen der Erfindung. Beschreibung und Zeichnungen dienen dazu, es einem Fachmann zu ermöglichen, die Erfindung herzustellen und zu verwenden, und sind nicht dazu beabsichtigt, den Umfang der Erfindung in jeglicher Art und Weise zu beschränken. Hinsichtlich der offenbarten Verfahren sind die dargestellten Schritte beispielhafter Natur, und somit ist die Reihenfolge der Schritte nicht notwendig oder entscheidend.

**[0017]** Die am 22. Dezember 2011 eingereichte vorläufige US-Patentanmeldung, Seriennummer 61/579,288, ist hier durch Verweis einbezogen.

**[0018]** CNC-Werkzeugmaschinen werden genutzt, um eine breite Vielfalt von Herstellungsprozessen einschließlich Montagevorgängen durchzuführen. Zum Beispiel ist in [Fig. 1](#) eine CNC-Werkzeugmaschine **10** zum Montieren eines Teils **12** eines Flugzeugrumpfes dargestellt. Das Flugzeugrumpfteil **12** hat einen halbkreisförmigen Querschnitt und wird aus einer Vielzahl von Außenhautblechen oder Platten gebildet, die an einem Trägerrost aus bogenförmigen Spanten und geraden Längsversteifungen typischerweise durch Niete, Schweißen oder Adhäsionskleben befestigt werden. Die Werkzeugmaschine **10** besitzt ein Paar von Armen **14**, **16**, die auf der Innenseite beziehungsweise Außenseite des Flugzeugrumpfteils **12** angeordnet sind. Die Werkzeugmaschine **10** ist entlang einer X-Achse **17** parallel zu einer Längsachse des Flugzeugrumpfteils **12** bewegbar, um die Außenhautbleche an jeder beliebigen ausgewählten Position entlang einer der Längsversteifungen zu befestigen, die sich zwischen gegenüberliegenden Enden des Flugzeugrumpfteils **12** erstrecken. Das Flugzeugrumpfteil **12** wird durch einen Rahmen **18** unterstützt, der in einer Richtung **19** um eine horizontale Achse drehbar ist. Der Rahmen **18** ist zwischen einem Paar von Stützen **20** eingebaut. Folglich können sich Flugzeugrumpfteil **12** und Rahmen **18** um die horizontale Achse drehen, damit sich die Werkzeugmaschine **10** zwischen Längsversteifungen bewegen kann und die Außenhautplatten längs jedes beliebigen der Rahmen befestigt werden können. Jedoch erfordert die CNC-Werkzeugmaschine **10** eine große und kostspielige Konstruktion, die den Flugzeugrumpfteil **12** tragen und drehen kann,

um die notwendigen Befestigungsvorgänge der Außenhautbleche durchzuführen.

**[0019]** In **Fig. 2** ist die CNC-Werkzeugmaschine **10** von **Fig. 1** durch einen Roboter **30** ersetzt, um den gleichen Befestigungsvorgang der Außenhautbleche durchzuführen. Das Flugzeugrumpfteil **12** ist an einer Spannvorrichtung **32** in einer im Allgemeinen vertikalen Ausrichtung dem Roboter **30** zugewandt befestigt. Der Roboter **30** kann sich entlang der Spannvorrichtung in Bewegungsrichtung **17** parallel zur Längsachse des Flugzeugrumpfteils **12** bewegen. Des Weiteren ist ein Arm **34** des Roboters **30** gelenkig befestigt, so dass er sich entlang des Flugzeugrumpfteils **12** in einer bogenförmigen Strecke längs der bogenförmigen Rahmen und zwischen den Längsversteifungen bewegt. Somit kann die robotergesteuerte Werkzeugmaschine **30** als eine kleinere und weniger teure Konstruktion als Alternative zu der CNC-Werkzeugmaschine **10** realisiert werden.

**[0020]** Wie oben erläutert ist, schalten Benutzer der CNC-Werkzeugmaschine **10** jedoch ungern auf die robotergesteuerte Werkzeugmaschine **30** um, weil eine Robotersteuerung **36** die von der CNC-Werkzeugmaschine **10** genutzte NC-Programmiersprache nicht ausführen kann. In **Fig. 3** ist im Format eines Blockdiagramms ein System **40** zum Umwandeln der NC-Programmiersprache in eine Roboter-Programmiersprache gemäß der vorliegenden Erfindung dargestellt. Eine Zentraleinheit (CPU) **42** hat über eine Busleitung **44** Zugriff auf jedes Gerät des Systems. Die Zentraleinheit **42** arbeitet nach einem Steuerungsprogramm, das in einem Steuerprogrammsspeicher (ROM) **46**, der ein mit der Busleitung **44** verbundener, nichtflüchtiger Speicher ist, gespeichert wird.

**[0021]** Ein mit der Busleitung **44** verbundener Arbeitsspeicher (RAM) **48** ist ein Speichersystem, das den Arbeitsbereich zum Ausführen der Verarbeitung für das System **40** bereitstellt. Mit der Busleitung **44** verbunden ist ein Display und Tastatur **50**, um zu ermöglichen, dass eine Bedienungsperson mit dem System **40** in Wechselwirkung tritt. Das System ist über einen Kommunikationsport **52** und die Busleitung **44** mit einem an den Port **52** angeschlossenen, rechnerunabhängig arbeitenden Programmiersystem **54** verbunden. Der Benutzer oder die Bedienungsperson erzeugen auf dem rechnerunabhängig arbeitenden Programmiersystem **54** das „Teileprogramm“ in NC-Programmiersprache. Das NC-Programm wird dann über den Kommunikationsport **52** auf das System **40** herunter geladen und auf ein Massenspeichergerät **56** kopiert, indem ein eingereguliertes Verfahren (wie beispielsweise (Dateiübertragungsprotokoll, usw.) verwendet wird. Das Massenspeichergerät **56** ist an die Busleitung **44** angeschlossen und stellt einen Platz zum Speichern eines originalen Teileprogramms (in NC-Sprache) **58**, einer XML-Datei **60** (erweiterbare Auszeichnungsspra-

che) des Teileprogramms, eines konvertierten Teileprogramms **62** (in Robotersprache) sowie von Unterprogrammen für M-Codes **64** zur Verfügung. Der Roboterarm **34** wird durch einen mit der Busleitung **44** verbundenen Hilfsverstärker **66** auf der Basis von Bewegungsbefehlen in dem Roboterprogramm **62** angetrieben.

**[0022]** Programmumwandlung von der NC-Programmiersprache in die Roboter-Programmiersprache basiert auf einer Tabelle für Umwandlungskonfigurationsdaten, die aus der XML-Datei **60** abgeleitet ist. Die Tabelle definiert die Sprachenelemente des NC-Programms **58**. Diese Tabelle kann auch die Sprache neu bestimmen, die von der Sprache abweicht, die typischerweise verwendet wird. Einige Teile der Tabelle sind:

Zwei-Buchstaben-Befehl – ob Zwei-Buchstaben-Befehle erlaubt sind

(Voreinstellung: Ja)

X-Vorspann – Das Vorspannzeichen für die X-Positionsdaten

(Voreinstellung: X)

X-Nachspann – Das Nachspannzeichen für X-Positionsdaten

(Voreinstellung: <Keine>

Config-Vorspann – Das Vorspannzeichen für die Konfigurationszeichenfolge

(config string) (Voreinstellung: 0)

F-Vorspann – Das Vorspannzeichen für Vorschubgeschwindigkeitswert

(Voreinstellung: F)

Misch-Vorspann – Liste von Mischcodes und ihre Handhabung

(Voreinstellung: M)

Verbindungsmodifikationstyp – M-Codewert zur Festlegung des Verbindungsmodifikationstyp (Voreinstellung: <Keine>)

Uframe num – M-Codewert zur Festlegung der Anzahl des Userframe

(Voreinstellung: <Keine>), usw..

**[0023]** Die Programmumwandlung findet in zwei Verarbeitungsstufen wie folgt statt: 1) Beim Einschalten liest das System **40** die XML-Datei **60** und stellt die Tabelle für Umwandlungskonfigurationen in dem Arbeitsspeicher RAM **48** ein; und 2) Bei jeder Programmumwandlung wandelt das System **40** das originale NC-Programm **58** in das interne Roboterprogramm **62** gemäß der Umwandlungskonfigurationstabelle um.

**[0024]** Jedes Element des NC-Teileprogramms **58** ist ein Buchstabe des Alphabets. Dieser Schritt wird die Bedeutung jedes Programmelements im NC-Programm **58** auf der Basis der XML-Datei **60** definieren. Diese Umwandlungstabelle ist nötig, weil die Bedeutung des Elements unterschiedlich zu der im Allgemeinen verwendeten typischen Definition sein kann. Mit anderen Worten, ein C-Code kann zum Bei-

spiel eine andere Bedeutung als das haben, was C-Code in der NC-Programmiersprache **58** typischerweise bedeutet.

**[0025]** In diesem Schritt liest das System **40** die XML-Datei **60** und erzeugt die Tabelle für Umwandlungskonfigurationen entsprechend den XML-Attributen in der Datei. Die Umwandlungskonfigurationstabelle enthält die Vorschrift zur Umwandlung. Zum Beispiel ist in dieser Tabelle der für die X-Komponente der Positionsdaten zu verwendende Buchstabe festgelegt. Die Tabelle wird in dem Arbeitsspeicher RAM **48** gespeichert und verbleibt darin, bis das System **40** heruntergefahren wird oder bis die neue XML-Datei geladen ist.

**[0026]** Daher wird dieser Prozess typischerweise einmal nach dem Einschalten erledigt. Das Ablaufdiagramm **70** dieses Bearbeitungsverfahrens ist in [Fig. 4](#) gezeigt und umfasst die folgenden Schritte:

Schritt **71** – Starten

Schritt **72** – Öffnen der Daten der XML-Datei **60**

Schritt **73** – Zuordnen eines Bereiches für die Umwandlungskonfigurationstabelle im RAM **48**

Schritt **74** – Lesen einer Zeile von XML-Daten

Schritt **75** – Identifizieren des XML-Attributs für die Zeile

Entscheidungspunkt **76** – „Gültig?“ Wenn es ein gültiges XML-Attribut ist, Springen bei „Ja“ zum Schritt **77**; und wenn nicht, Springen bei „Nein“ zum Schritt **78**

Schritt **77** – Speichern von XML-Attribut in der Umwandlungskonfigurationstabelle

Schritt **78** – Versenden eines Fehlers und Abbrechen der Bearbeitung durch Schritt **80** bei Schritt **81**

Entscheidungspunkt **79** – Falls es eine weitere Zeile gibt, Springen bei „Ja“ und Zurückkehren zum Schritt **74**; und falls es keine weiteren Zeilen gibt, Springen bei „Nein“ zum Schritt **80**

Schritt **80** – Schließen der XML-Datei

Schritt **81** – Beenden

**[0027]** Das System **40** wandelt das originale NC-Programm **58** entsprechend der Tabelle für Umwandlungskonfigurationen in das interne Roboterprogramm **62** um. Die eigentliche Verarbeitung zum Umwandeln des vorgegebenen NC-Teileprogramms (Anwenderprogramm) **58** in das Roboterprogramm **62** ist in [Fig. 5](#) bis [Fig. 7](#) als ein Ablaufdiagramm **90** dargestellt, das die folgenden Schritte aufweist:

Schritt **91** – Starten

Schritt **92** – Öffnen der originalen NC-Programmdatei **58**

Entscheidungspunkt **93** – Falls Öffnen nicht erfolgreich ist, Springen bei „Nein“ zum Schritt **94**; falls erfolgreich, Springen bei „Ja“ zum Schritt **96**

Schritt **94** – Versenden eines Fehlers

Schritt **95** – Verlassen

Schritt **96** – Zuordnen des Ausgabebereiches für logische Aussagen

Entscheidungspunkt **97** – Falls Zuordnung nicht erfolgreich ist, Springen bei „Nein“; Versenden eines Fehlers im Schritt **94** und Verlassen bei Schritt **95**; falls erfolgreich, Springen bei „Ja“ zum Schritt **98**

Schritt **98** – Zuordnen des Ausgabebereiches für Positionen

Entscheidungspunkt **99** – Falls Zuordnung nicht erfolgreich ist, Springen bei „Nein“; Versenden eines Fehlers im Schritt **94** und Verlassen bei Schritt **95**; falls erfolgreich, Springen bei Schritt „Ja“ zum Schritt **100**

Schritt **100** – Lesen einer Befehlszeile aus der NC-Programmdatei **58**

Schritt **101** – Erzeugen einer Eingabe in den logischen Ausgabebereich

Schritt **102** – Aus der Befehlszeile, Abrufen eines codierten Befehlswortes und des Arguments (numerisch oder Zeichenfolge), die sich an das codierte Befehlswort anschließt

Schritt **103** – Aufsuchen des Buchstabens des codierten Befehlswortes in der Umwandlungskonfigurationstabelle und Bestimmen des Befehlstyps auf der Basis der Tabellendaten

Entscheidungspunkt **104** – Wenn der Buchstaben des codierten Befehlswortes in der Tabelle nicht gefunden wird, Springen bei „Nein“;

**[0028]** Versenden eines Fehlers im Schritt **94** und Verlassen bei Schritt **95**; wenn der Buchstabe des codierten Befehlswortes gefunden wird, Springen bei „Ja“ zum Entscheidungspunkt **105**

Entscheidungspunkt **105** – Wenn der Befehl vom Typ des Vorbereitungsbefehls ist (typischerweise ist es ein „G“ Code), Springen zum Schritt **106**; wenn der Befehl einer vom Typ Positionsbefehle (X, Y, Z, W, P, R, E oder C) ist, dann Springen zum Entscheidungspunkt **107**; wenn der Befehl ein anderer gültiger Befehl ist, der auf der Tabelle für Umwandlungskonfigurationen basiert, Springen zum Schritt **111**

Schritt **106** – Festhalten der Bewegungscharakteristiken basierend auf dem Befehlsargument und Gehen zum Entscheidungspunkt **112**

Entscheidungspunkt **107** – Falls in dem Positionsausgabebereich keine Eingabe vorhanden ist, Springen bei „Nein“ zum Schritt **108**; falls es die Eingabe gibt, Springen bei „Ja“ zum Schritt **109**

Schritt **108** – Zuordnen einer Eingabe in den Positionsausgabebereich

Schritt **109** – Speichern des Positionswertes in der Positionseingabe im Positionsausgabebereich

Schritt **110** – Speichern des Bewegungsbefehls in der logischen Eingabe im logischen Ausgabebereich

Schritt **111** – Speichern des „anderen“ Befehls in der logischen Eingabe im logischen Ausgabebereich

Entscheidungspunkt **112** – Wenn es mehrere codierte Befehlsörter in der Befehlszeile gibt, dann Springen bei „Ja“ zum Schritt **102**; falls nicht, Springen bei „Nein“ zum Entscheidungspunkt **113**

Entscheidungspunkt **113** – Wenn es mehrere Befehlszeilen in der Programmdatei gibt, dann Springen

gen bei „Ja“ zum Schritt **100**; falls nicht, Springen bei „Nein“ zum Schritt **114**

Schritt **114** – Schließen der originalen NC-Programmdatei **58**

Schritt **115** – Erzeugen der Ausgangsdatei des Roboterprogramms **62** und Öffnen derselben

Schritt **116** – Schreiben der Kopfzeilendirektive für die logischen Aussagen

Schritt **117** – Schreiben aller logischen Ausgabeeinträge in die Datei

Schritt **118** – Schreiben der Kopfzeilendirektive für die Positionsausgabe

Schritt **119** – Schreiben aller Positionsausgabeeinträge in die Datei

Schritt **120** – Schließen der Ausgabedatei

Schritt **121** – Aufheben der Zuordnung aller logischen und Positionsausgabeeinträge

Schritt **122** – Beenden

**[0029]** Hinsichtlich der Positionsdarstellungen werden solche NC-Programme typischerweise auf einem rechnerunabhängig arbeitenden Programmiersystem erzeugt. Üblich ist es, dass die Positionen in den NC-Programmen in Form eines absoluten Positionswertes in jedem beweglichen Gelenk/Achse vorgegeben sind. Auf einer typischen Werkzeugmaschine sind die beweglichen Gelenke nach kartesischen Koordinaten ausgerichtet, und der Wert von jedem dieser Gelenke ist in X, Y, Z, usw. gegeben. Aufgrund der komplexen Beschaffenheit der Achsenkonfiguration ist das Berechnen der Gelenkwinkel der Roboterachsen für das rechnerunabhängig arbeitende Programmiersystem manchmal schwieriger als das Berechnen des Positionswertes im Rahmen kartesischer Koordinaten.

**[0030]** Das vorgeschlagene Verfahren kann im NC-Programm **58** zusätzlich zur Gelenkdarstellung die Positionsdarstellung XYZWPR verwenden. Mit der Positionsdarstellung XYZWPR ist es notwendig, den Konfigurationswert „Config“ zusätzlich zu dem Positionswert festzulegen. Es ist notwendig, einen einzigen Gelenkwert für die gegebene XYZWPR Position auszuweisen. Im vorgeschlagenen Verfahren wird der Wert „Config“ als eine durch „C“-Code (config) angegebene Kommentarzeichenfolge dargestellt. Zum Beispiel kann in der Darstellung XYZWPR ein Positionswert in dem folgenden Format gegeben sein:

```
X 123.00Y-234.562 345.03W179. 34P-178.34R0.234C(N,L,0,1,-1)
```

**[0031]** Weil die Position mit der Positionsdarstellung XYZWPR als die Position des Werkzeugmaschinenmittelpunktes (TCP) des Roboters **30** definiert ist, ist es notwendig, das Bezugssystem für jede Position festzulegen. Im vorgeschlagenen Verfahren wird dies durch den speziellen M-Code im Programm vorgenommen, der das Anwenderkoordinatensystem oder das Anwenderkoordinatensystem der Werkzeugmaschine auswählt.

Beispiele:

M23T01 – Zum Setzen des Anwenderkoordinatensystems auf #1

M24T02 – Zum Setzen des Anwenderkoordinatensystems der Werkzeugmaschine auf #2

**[0032]** Einige der M-Codes (Mischcodes) sind für die spezielle Werkzeugmaschine zur spezifischen Anwenderanforderung definiert. Oft wird er von einem durch einen T-Code festgelegten Argument begleitet. Um die Flexibilität einer solchen speziellen Verarbeitung zu erfüllen, generiert das vorgeschlagene Verfahren einen Aufruf an das Unterprogramm **64** für jeden M-Code, der festgelegt ist. Zum Beispiel weist das NC-Programm **58** einen Code „M08T02“ auf, und das umgewandelte Roboterprogramm **62** weist den Befehl „CALL“ wie beispielsweise „CALL M08(2)“ auf. Es ist außerdem anzumerken, dass das Argument T02 als der Parameter der ganzen Zahl an das Unterprogramm „M08“ weitergegeben wird.

**[0033]** Ein Mischcode kann einer Bewegung zugeordnet festgelegt werden. In solchen Fällen wird ein „CALL Befehl“ an eine Bewegung in dem umgewandelten Roboterprogramm **62** angefügt, wobei der Befehl zu dem Zeitpunkt stattfindet, an dem der Roboter den festgelegten Zielpunkt mit einer Abweichung erreicht, dass die zeitliche Einstellung entweder früher oder später eingestellt werden kann, indem die Zeitabweichung (in Millisekunden) festgelegt wird.

**[0034]** In [Fig. 8](#) ist das Blockdiagramm einer Steuereinheit **130** für die in [Fig. 1](#) gezeigte CNC-Werkzeugmaschine **10** dargestellt, die modifiziert ist, um die in [Fig. 2](#) gezeigte, robotergesteuerte Werkzeugmaschine **30** zu bedienen. Die CNC-Werkzeugmaschine **10** wird durch eine zweckbestimmt ausgelegte PLC **131** (frei programmierbare Steuerung) gesteuert, die mit einem Displaygerät **132** verbunden ist, um der Bedienungsperson Programm- und Statusinformationen anzuzeigen. Die PLC **131** stellt jedes beliebige Gerät dar, das imstande ist, das NC-Programm abzuarbeiten, um die CNC-Werkzeugmaschine **10** zu bedienen. Des Weiteren ist die PLC **131** optional, wenn die CNC-Maschine das NC-Programm abarbeiten kann.

**[0035]** Die robotergesteuerte Werkzeugmaschine **30** ist zur Steuerung nach dem umgewandelten Anwenderprogramm **62** ([Fig. 3](#)) mit der Robotersteuerung **36** verbunden. Die PLC **131** ist an eine PLC-Schnittstelle **133** in der Robotersteuerung **36** durch einen Feldbus **134** zum Austausch von Signalen angeschlossen, um eine Programmsteuerung und Zustandsüberwachung bereitzustellen. Das NC-Programm wird von der PLC **131** auf das MD-Laufwerk **56** ([Fig. 3](#)) in der Robotersteuerung **36** herunter geladen. Die Robotersteuerung **36** wandelt das NC-Programm in das in TP Sprache umgewandelte Anwen-

derprogramm **62** um und erzeugt die TP Unterprogramme für die M-Codes **64** (Fig. 3). Anschließend ist die Robotersteuerung **36** bereit, die robotergesteuerte Werkzeugmaschine **30** zur Ausführung des in dem NC-Programm verkörperten Herstellungsprozesses zu betreiben. Ein Programmierhandgerät **135** ist mit der Robotersteuerung **36** verbunden, damit die Bedienungsperson alle diejenigen Funktionen ausführen kann, die normalerweise mit der Steuerung eines Roboters verbunden sind.

**[0036]** Die vorliegende Erfindung wurde gemäß den Bestimmungen der patentrechtlichen Gesetzesvorschriften dahingehend beschrieben, was als ihre bevorzugte Ausführungsform darzustellen angesehen wird. Es soll jedoch angemerkt werden, dass die Erfindung praktisch, anders als speziell dargestellt und beschrieben, durchgeführt werden kann, ohne von ihrem Geist oder Umfang abzuweichen.

### Patentansprüche

1. Verfahren zur Steuerung eines Roboters, umfassend die Schritte:

Speichern einer NC-Programmiersprache in einem Massenspeichergerät, das mit einer Zentraleinheit verbunden ist, wobei die NC-Programmiersprache angepasst ist, um durch eine CNC-Werkzeugmaschine zur Durchführung eines Herstellungsprozesses ausgeführt zu werden;

Bedienen der Zentraleinheit zum Umwandeln der NC-Programmiersprache in eine Roboter-Programmiersprache auf der Basis einer vorher festgelegten Tabelle für Umwandlungskonfigurationen, die diskrete Konfigurationsanweisungen aufweist, um jeden Positionsbefehl der NC-Sprache in einen Positionsbefehl der Robotersprache umzuwandeln und jeden Mischcode-Befehl der NC-Sprache in ein Robotersprachen-Unterprogramm umzuwandeln;

Speichern der Robotersprachen-Positionsbefehle und der Robotersprachen-Unterprogramme als die Roboter-Programmiersprache in dem Massenspeichergerät; und

Ausführen der Roboter-Programmiersprache mit einer an das Massenspeichergerät und den Roboter angeschlossenen Robotersteuereinheit bei einer Laufzeit des Roboters, um den Herstellungsprozess durchzuführen.

2. Verfahren nach Anspruch 1, umfassend das Speichern eines Steuerungsprogramms in einem Steuerungsprogramm Speicher der Robotersteuereinheit und Durchführen des Umwandlungsschrittes, indem das Steuerungsprogramm genutzt wird.

3. Verfahren nach Anspruch 1, umfassend das Bereitstellen einer XML-Datei von Umwandlungsvorschriften in dem Massenspeichergerät und das Bedienen der Zentraleinheit, um aus der XML-Datei die Umwandlungskonfigurationstabelle zu erzeugen, wo-

bei die Tabelle für Umwandlungskonfigurationen Vorschriften zum Umwandeln der NC-Programmiersprache in die Roboter-Programmiersprache enthält.

4. Verfahren nach Anspruch 1, umfassend das Durchführen des Umwandlungsschrittes, um eine Eins-zu-eins-Übereinstimmung zwischen Instruktionszeilen der Roboter-Programmiersprache und Instruktionszeilen der NC-Programmiersprache zur Verfügung zu stellen.

5. Verfahren nach Anspruch 1, bei dem der Bedienschnitt umfasst:

Öffnen der im Massenspeichergerät gespeicherten NC-Programmiersprache;

Zuordnen eines Ausgabebereichs des Massenspeichergerätes für logische Aussagen;

Zuordnen eines Ausgabebereichs des Massenspeichergerätes für Positionen;

Lesen einer Befehlszeile aus der NC-Programmiersprache;

Erzeugen einer Eingabe in dem logischen Ausgabebereich für die Befehlszeile;

Abrufen aus der Befehlszeile eines codierten Befehlswortes und einer Argumentzeichenfolge, die sich an das codierte Befehlswort anschließt;

Aufsuchen eines codierten Befehlswortes, das dem codierten Befehlswort in der Tabelle für Umwandlungskonfigurationen zugeordnet ist; und

Bestimmen eines Typs für die Befehlszeile auf der Basis von Daten in der Tabelle für Umwandlungskonfigurationen.

6. Verfahren nach Anspruch 5, bei dem Bewegungseigenschaften, die auf der Argumentzeichenfolge der Befehlszeile basieren, gespeichert werden, wenn die Befehlszeile ein Befehl vom Vorbereitungstyp ist.

7. Verfahren nach Anspruch 5, bei dem eine Eingabe im Positionsausgabebereich zugeordnet wird, wenn die Befehlszeile ein Befehl vom Positionstyp ist.

8. Verfahren nach Anspruch 5, bei dem, wenn die Befehlszeile ein Befehl eines gültigen anderen Typs außer einem Befehl vom Vorbereitungstyp oder einem Befehl vom Positionstyp ist, der Befehl des anderen Typs als eine Eingabe in dem logischen Ausgabebereich gespeichert wird.

9. Verfahren nach Anspruch 1, bei dem der Bedienschnitt umfasst:

Öffnen der im Massenspeichergerät gespeicherten NC-Programmiersprache;

Bestimmen eines Typs für jede Befehlszeile in der NC-Programmiersprache;

Zuordnen einer Eingabe in einem logischen Ausgabebereich oder einem Positionsausgabebereich basierend auf dem Befehlszeilentyp; und

Schließen der NC-Programmiersprache.

10. Verfahren nach Anspruch 9, umfassend:  
 Erzeugen einer Roboterprogrammdatei;  
 Schreiben einer Übersetzungsanweisung der Kopfangaben für logische Eingaben in die Roboterprogrammdatei;  
 Schreiben aller logischen Eingaben aus dem logischen Ausgabebereich in die Roboterprogrammdatei;  
 Schreiben einer Übersetzungsanweisung der Kopfangaben für die Positioneingaben in die Roboterprogrammdatei;  
 Schreiben aller Positioneingaben aus dem Positionsausgabebereich in die Roboterprogrammdatei;  
 Schließen der Roboterprogrammdatei; und Aufheben der Zuordnung aller logischen und Positioneingaben.

11. Verfahren nach Anspruch 1, umfassend das Hinzufügen eines eindeutig bestimmten Positionswertes zu einer in der NC-Programmiersprache beschriebenen Positionsdarstellung XYZWPR zum Erzeugen eines entsprechenden Positionsbefehls der Robotersprache.

12. Verfahren nach Anspruch 11, bei dem ein eindeutig bestimmter Konfigurationswert in der Positionsdarstellung als eine Kommentarzeichenfolge dargestellt wird.

13. Verfahren nach Anspruch 1, umfassend eine Zeitabweichung, die einem der Mischcodes abhängig vom Eintreffen des Roboters an einem Bestimmungspunkt zugeordnet ist.

14. Verfahren nach Anspruch 1, umfassend das Einsetzen eines speziellen Mischcodes vor einem der Positionsbefehle der Robotersprache, um eines von Anwenderkoordinaten und Anwender-Werkzeugkoordinaten auszuwählen.

15. Verfahren nach Anspruch 1, Einsetzen eines speziellen Mischcodes vor einem der Robotersprachen-Positionsbefehle, um einen Typ der Roboterbewegung auszuwählen.

16. System zur Steuerung eines Roboters basierend auf einer NC-Programmiersprache, umfassend:  
 einen Roboter;  
 eine mit dem Roboter verbundene Robotersteuereinheit zum Steuern des Roboters entsprechend einer Roboter-Programmiersprache;  
 ein Massenspeichergerät, das mit der Robotersteuereinheit verbunden ist, um die NC-Programmiersprache zu speichern; und  
 ein Steuerungsprogramm, das durch die Robotersteuereinheit zum Umwandeln der NC-Programmiersprache in die Roboter-Programmiersprache ausgeführt wird basierend auf einer vorher festgelegten Umwandlungskonfigurationstabelle, die diskrete Konfigurationsanweisungen aufweist, um jeden Posi-

tionsbefehl der NC-Sprache in einen Positionsbefehl der Robotersprache umzuwandeln, und jeden Mischcode-Befehl der NC-Sprache in ein Unterprogramm der Robotersprache umzuwandeln.

17. System nach Anspruch 16, bei dem das Massenspeichergerät die Roboter-Programmiersprache in einer Roboter-Programmdatei und die Unterprogramme der Robotersprache getrennt von der Roboter-Programmiersprache speichert.

18. System nach Anspruch 16, bei dem das Massenspeichergerät eine XML-Datei von Umwandlungsvorschriften speichert, um eine Umwandlungskonfigurationstabelle zu erzeugen, die Vorschriften zum Umwandeln der NC-Programmiersprache in die Roboter-Programmiersprache enthält.

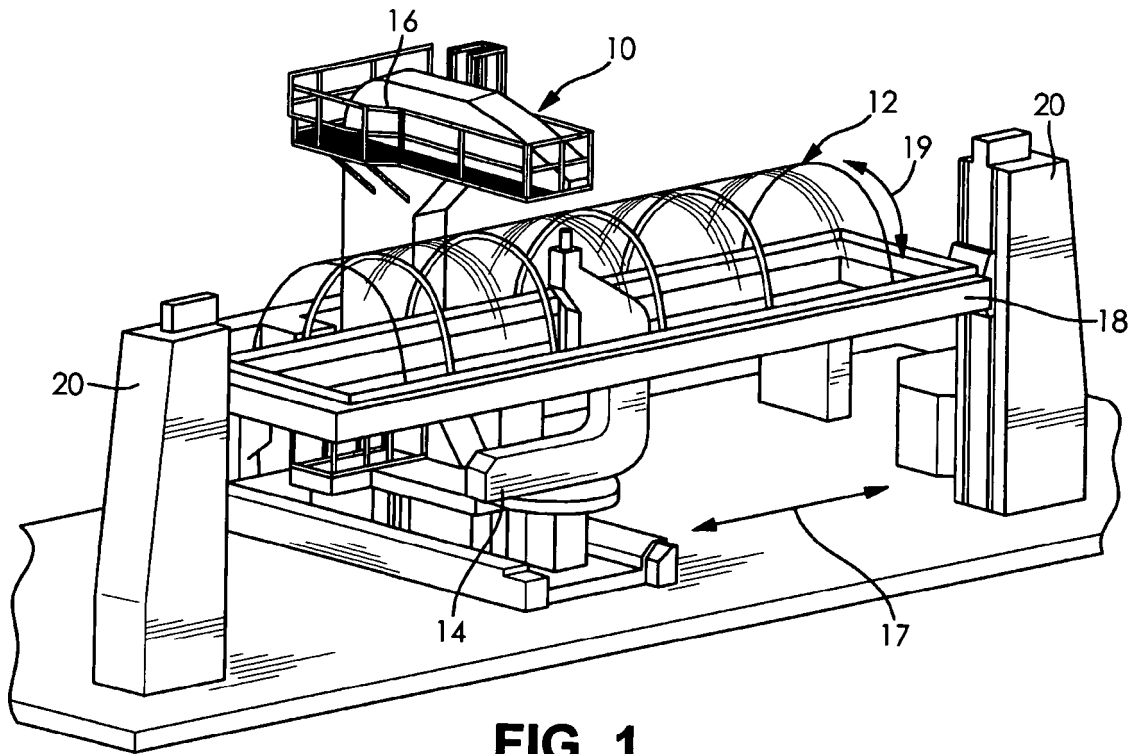
19. System nach Anspruch 16, umfassend ein mit der Robotersteuereinheit verbundenes Displaygerät zum Anzeigen von Instruktionszeilen der NC-Programmiersprache und der Roboter-Programmiersprache.

20. System nach Anspruch 16, bei dem die Robotersteuereinheit eine PLC-Schnittstelle zur Kommunikation mit einer speicherprogrammierbaren Steuerung umfasst.

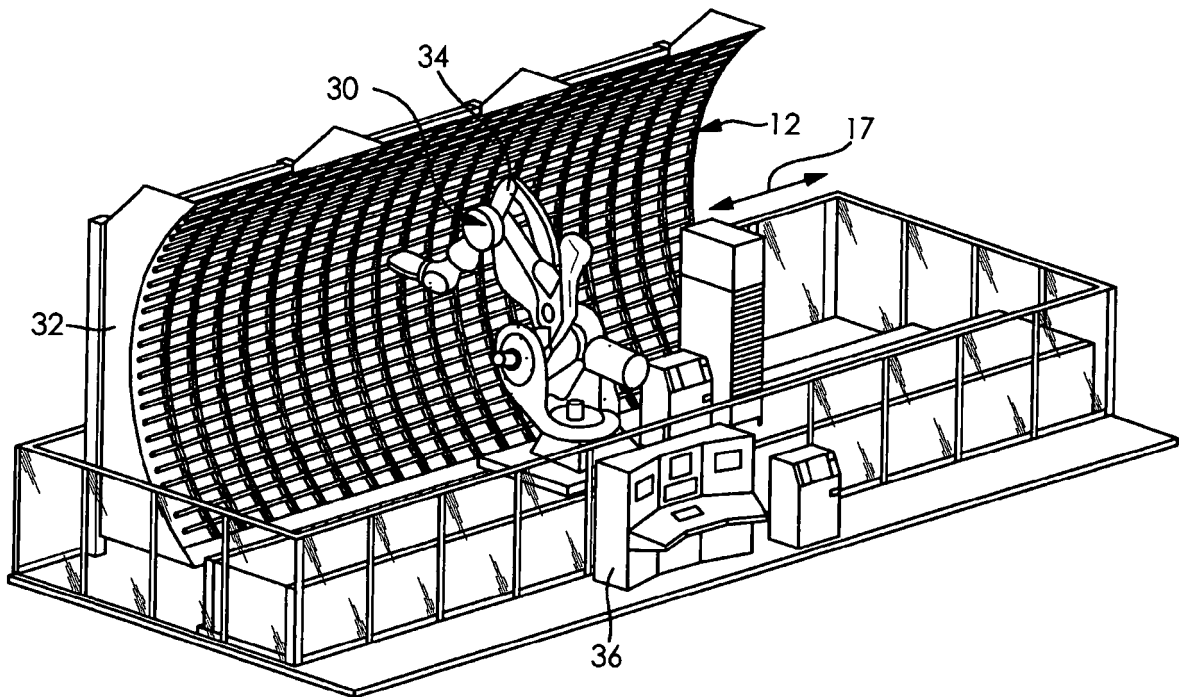
Es folgen 7 Blatt Zeichnungen



Anhängende Zeichnungen



**FIG. 1**



**FIG. 2**

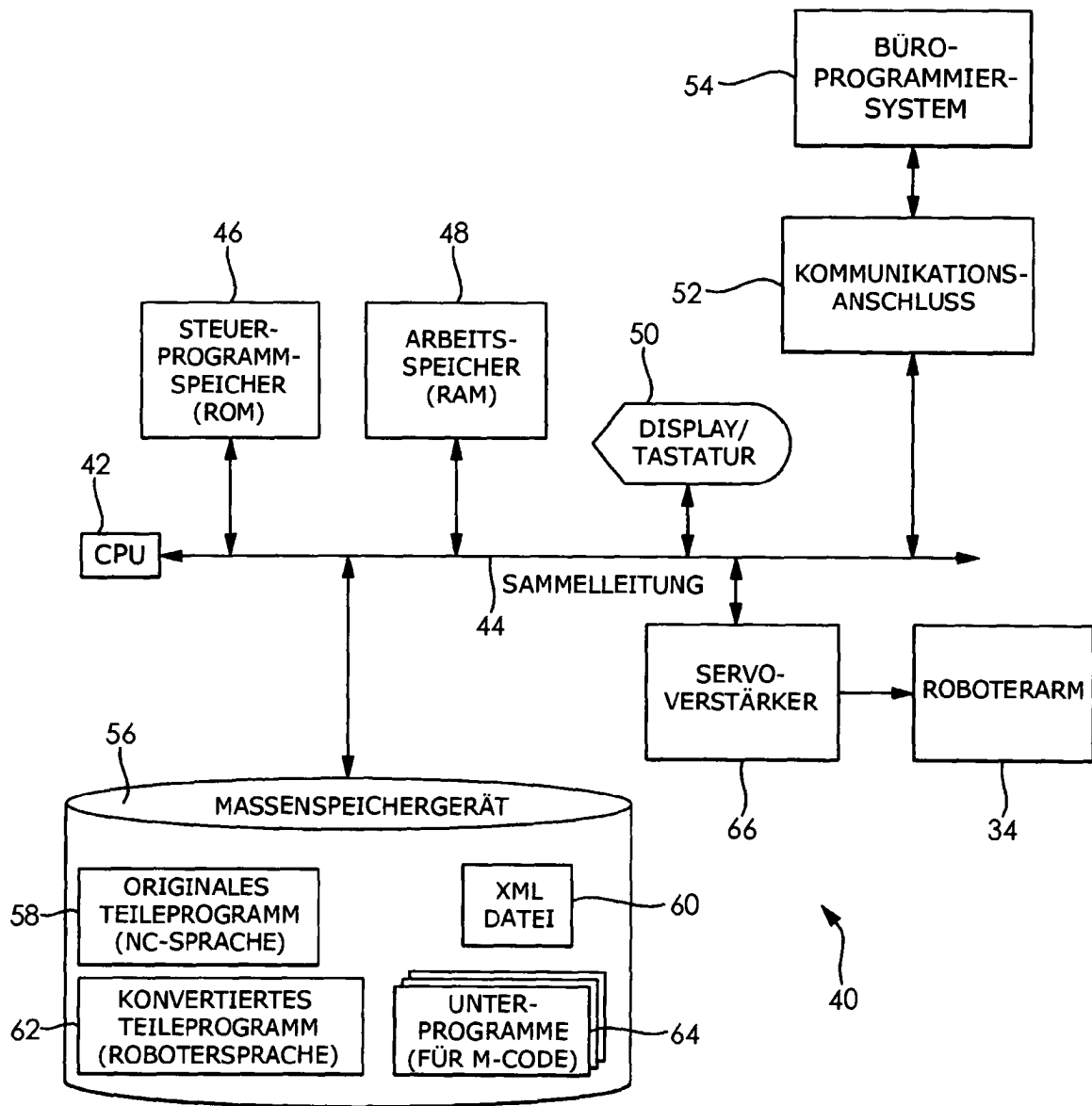
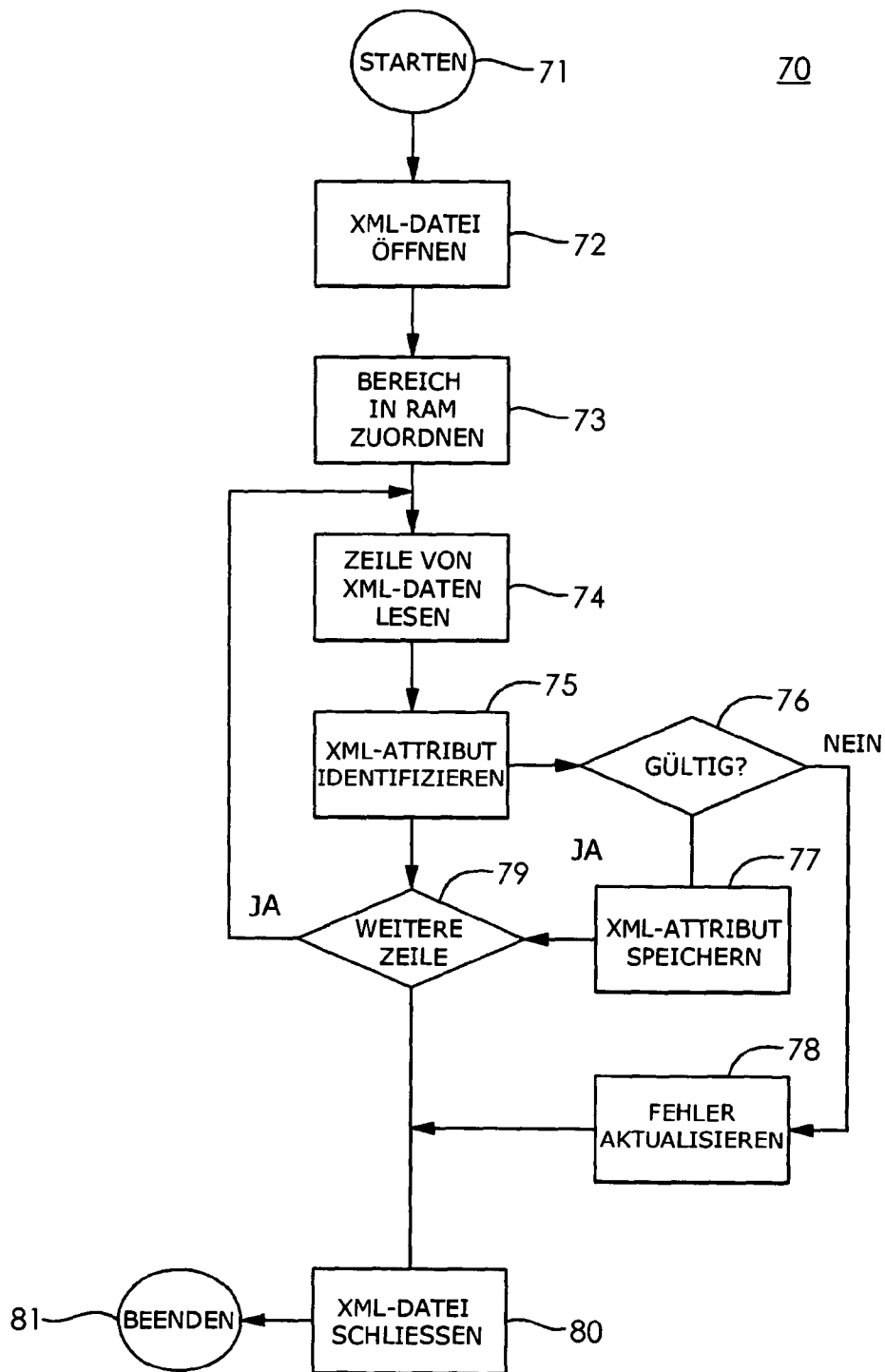


FIG. 3



**FIG. 4**

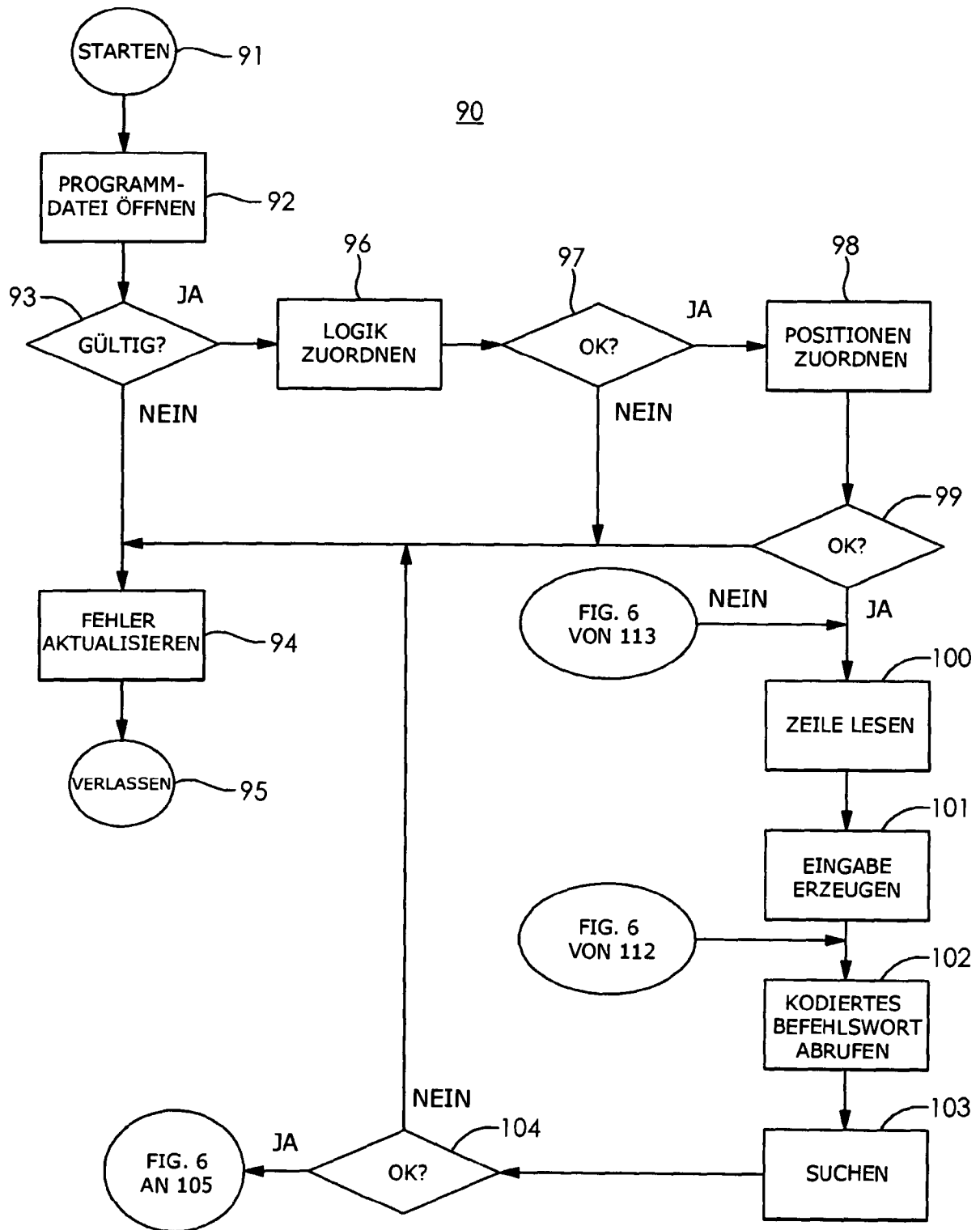
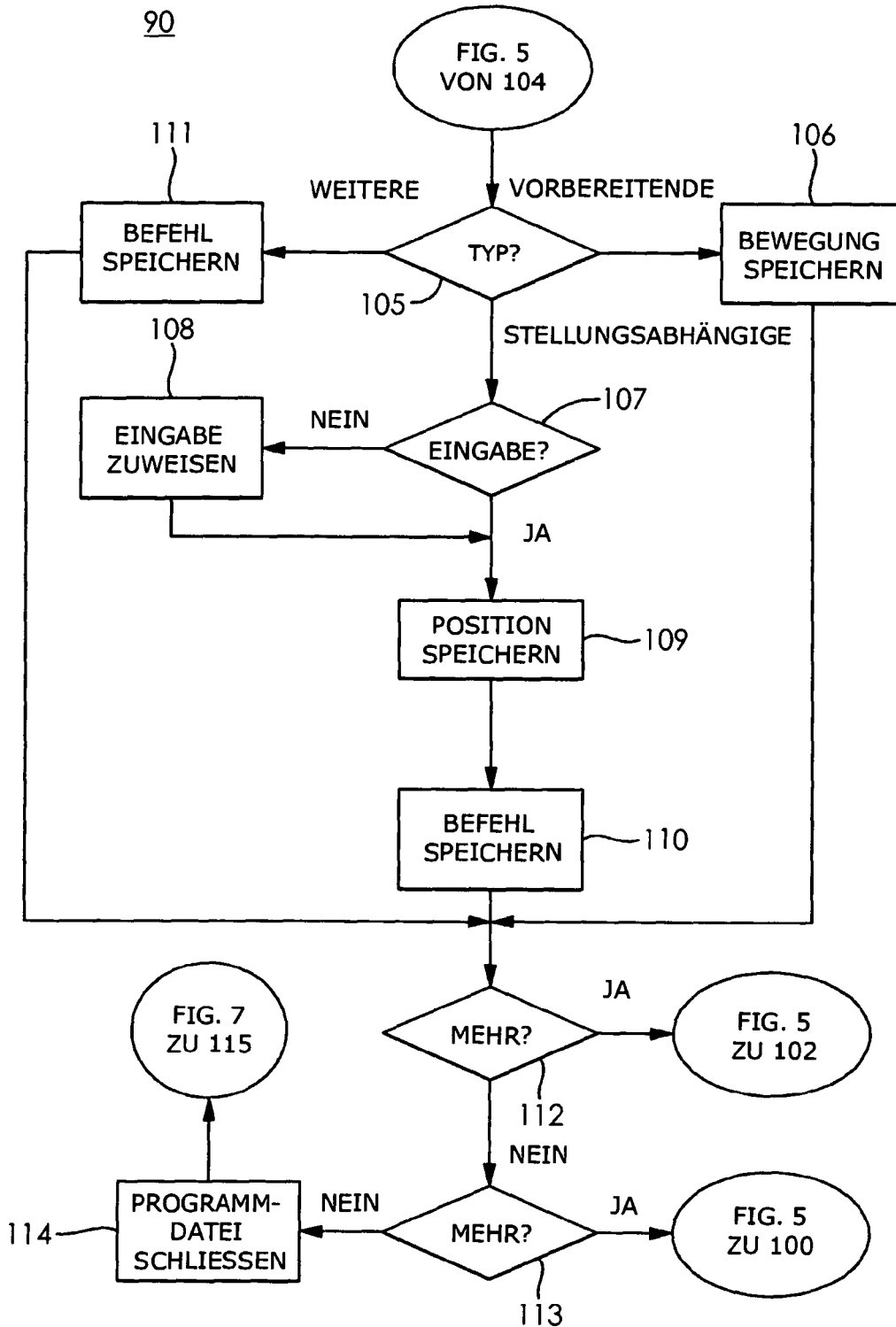
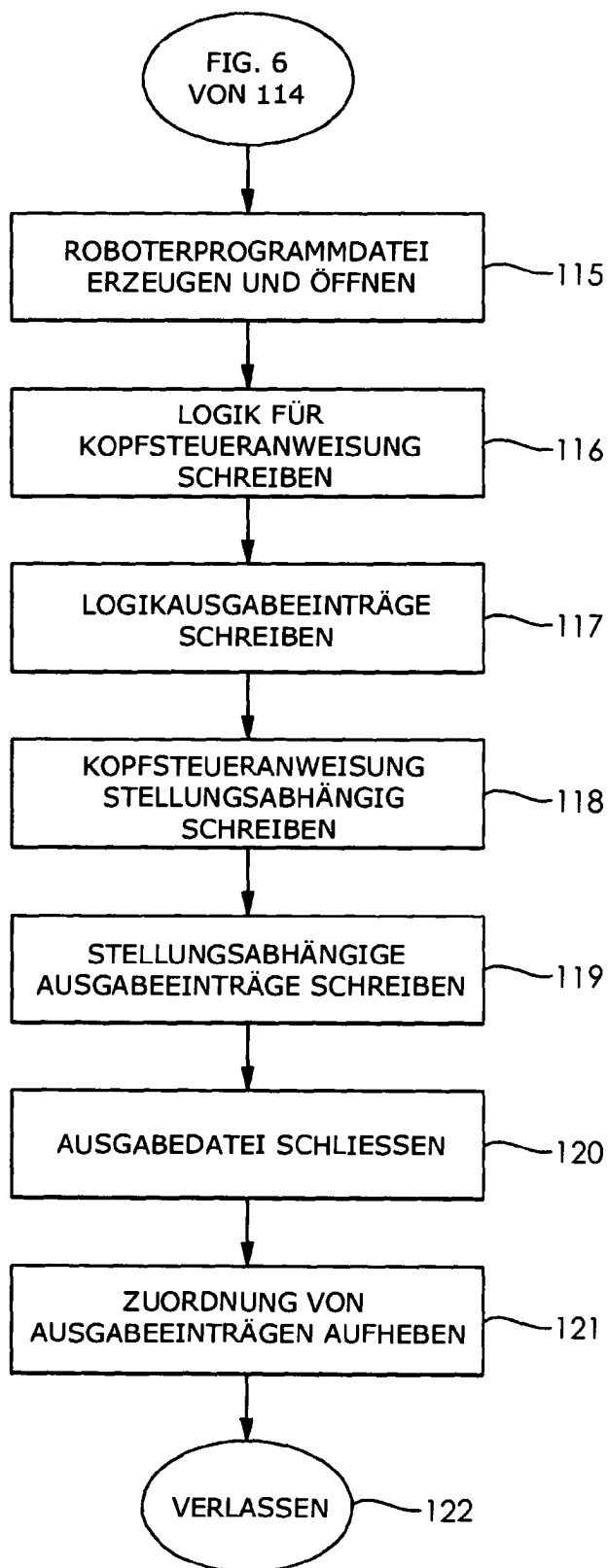


FIG. 5



**FIG. 6**

90



**FIG. 7**

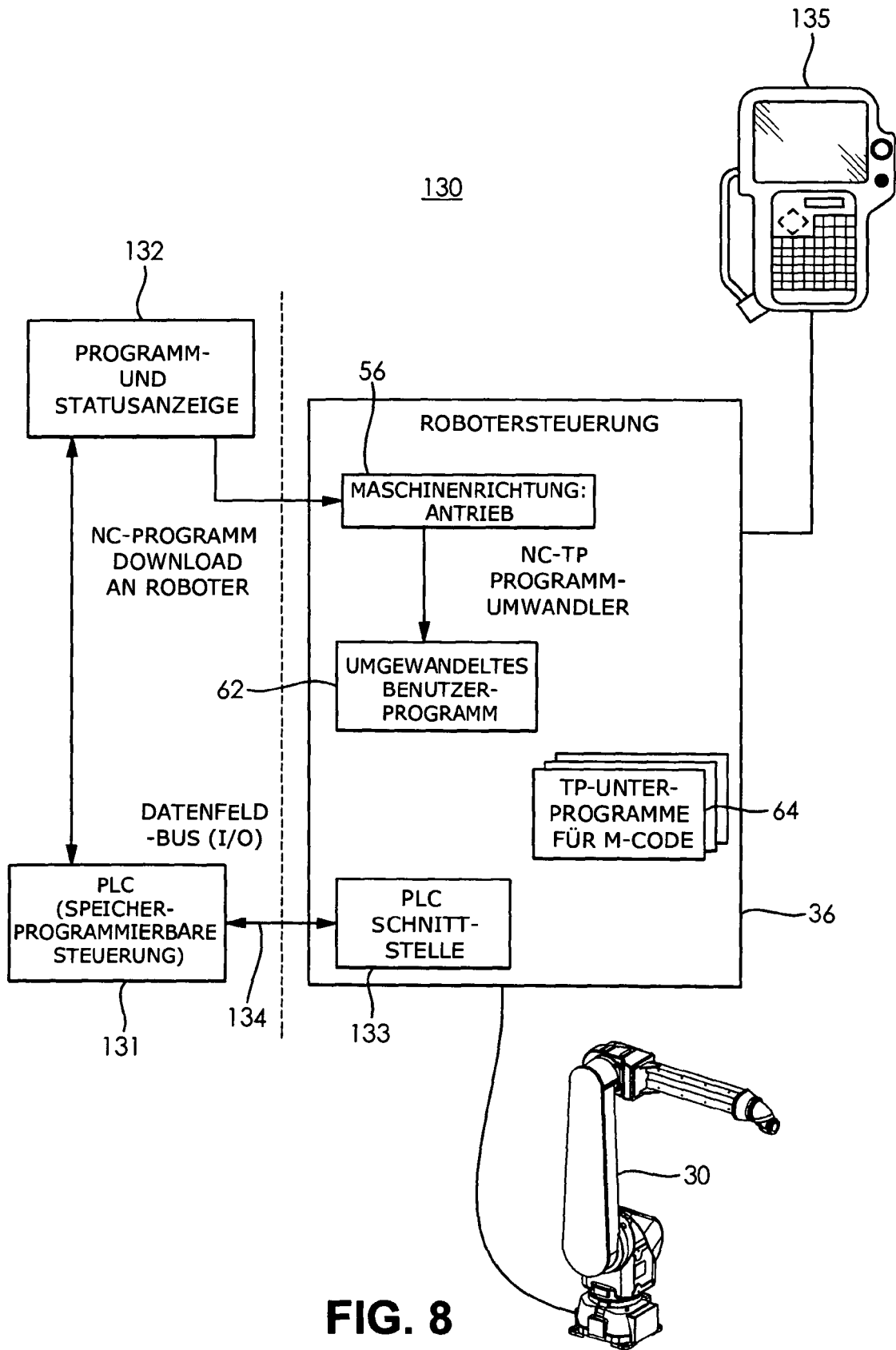


FIG. 8