



(10) **DE 100 48 360 B4** 2016.09.15

(12)

## Patentschrift

(21) Aktenzeichen: **100 48 360.7**  
(22) Anmeldetag: **29.09.2000**  
(43) Offenlegungstag: **10.05.2001**  
(45) Veröffentlichungstag  
der Patenterteilung: **15.09.2016**

(51) Int Cl.: **G05B 13/04 (2006.01)**  
**G05B 17/02 (2006.01)**  
**G05B 19/042 (2006.01)**

Innerhalb von neun Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

(30) Unionspriorität:  
**09/412,078 04.10.1999 US**

(73) Patentinhaber:  
**Fisher-Rosemount Systems, Inc., Austin, Tex., US**

(74) Vertreter:  
**Meissner Bolte Patentanwälte Rechtsanwälte  
Partnerschaft mbB, 80538 München, DE**

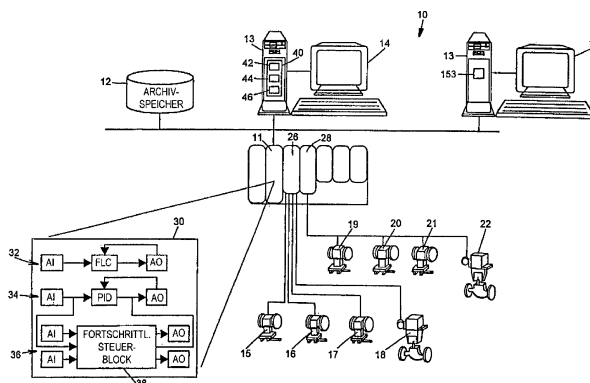
(72) Erfinder:  
**Blevins, Terrence L., Round Rock, Tex., US;**  
**Wojsznis, Wilhelm K., Round Rock, Tex., US;**  
**Thiele, Dirk, Austin, Tex., US; Tzovla, Vasiliki,**  
**Austin, Tex., US**

(56) Ermittelter Stand der Technik:  
**siehe Folgeseiten**

(54) Bezeichnung: **Integrierte, fortschrittliche Steuerblöcke in Prozeßsteuersystemen**

(57) Hauptanspruch: Prozess-Steuerung zur Steuerung eines Prozesses (58), umfassend:  
eine Vielzahl von Steuereingängen (IN1–IN3), die mit einer Vielzahl von Prozessausgängen verbunden sind, wobei jeder Steuereingang (IN1–IN3) ein Prozessausgangssignal von einem von der Vielzahl von Prozessausgängen empfängt;  
eine Vielzahl von Steuerausgängen (OUT1–OUT3), die mit einer Vielzahl von Prozesseingängen verbunden sind, wobei jeder Steuerausgang (OUT1–OUT3) ein Steuersignal an einen von der Vielzahl von Prozesseingängen (X1–X5) abgibt;  
einen Wellenformgenerator zum Erzeugen von Anregungswellenformen an jedem der Vielzahl von Steuerausgängen (IN1–IN3);  
eine Datensammeleinheit zum Sammeln von Daten, die Signalwerte an jedem der Vielzahl von Steuereingängen (IN1–IN3) darstellen, wenn der Wellenformgenerator die Anregungswellenformen an jedem der Vielzahl von Steuerausgängen (OUT1–OUT3) erzeugt;  
einen Steuerlogikparametergenerator, der die gesammelten Daten nutzt, um ein Prozessmodell für den Prozess (58) zu erzeugen, basierend darauf, wie jede der Steuereingänge auf die bei jeder der Vielzahl von Steuerausgängen erzeugten Anregungswellenformen antwortet, und Steuerlogikparameter gemäß des erzeugten Prozessmodells erzeugt; und  
eine Mehrfacheingabe-/Mehrfachausgabe-Steuerlogik, derart ausgebildet, vor der Erzeugung des Prozessmodells nicht konfiguriert zu sein und keine Steuerlogikparameter als modellvorhersagende Steuerung gemäß den Steuerlogikparameter konfiguriert zu sein, wobei die Prozess-Steuerung derart ausgebildet ist, ein Steuersignal an jedem der Vielzahl von Steuerausgängen zu erzeugen, basierend auf (i) der konfigurierten Mehrfacheingabe-/Mehrfachausgabe-Steuerlogik, und (ii) der Vielzahl von Prozessausgangssignalen, die an der Vielzahl von Steuereingängen empfangen wurden.

parameter als modellvorhersagende Steuerung gemäß den Steuerlogikparameter konfiguriert zu sein, wobei die Prozess-Steuerung derart ausgebildet ist, ein Steuersignal an jedem der Vielzahl von Steuerausgängen zu erzeugen, basierend auf (i) der konfigurierten Mehrfacheingabe-/Mehrfachausgabe-Steuerlogik, und (ii) der Vielzahl von Prozessausgangssignalen, die an der Vielzahl von Steuereingängen empfangen wurden.



(19)



Deutsches  
Patent- und Markenamt

(10) **DE 100 48 360 B4** 2016.09.15

(56) Ermittelter Stand der Technik:

DE	694 09 250	T2
US	7 257 523	B1
US	5 838 656	A
US	5 594 858	A
US	5 920 478	A
US	5 477 449	A
WO	96/ 34 324	A1
WO	97/ 42 553	A1
JP	H04- 33 102	A
JP	H06- 95 706	A

JP 2954660 B2 (Maschinenübersetzung), AIPN  
JPO (abgerufen am 30.11.2012)

## Beschreibung

**[0001]** Die vorliegende Erfindung betrifft allgemein Prozeßsteuersysteme und insbesondere die Verwendung von fortschrittlichen Steuerblöcken, wie etwa modellvorhersagenden und neuronalen Netzwerk-Steuerblöcken, in Prozeßsteuersystemen.

**[0002]** Prozeßsteuersysteme, wie verteilte oder skalarisierbare Prozeßsteuersysteme, die beispielsweise in chemischen Prozessen, in Erdölverarbeitungsprozessen oder anderen Prozessen verwendet werden, enthalten allgemein eine oder mehrere Prozeßsteuereinrichtungen, die miteinander und mit wenigstens einer Host- oder Bedienungsworkstation und mit einer oder mehreren Anlageneinrichtungen über analoge, digitale oder kombiniert analog-digitale Busverbindungen in Kommunikationsverbindung stehen. Die Anlageneinrichtungen, die beispielsweise Ventile, Ventilpositioniereinrichtungen, Schalter, Sensoren (wie zum Beispiel Temperatur-, Druck- und Durchflußmengensensoren) sein können, führen Steuerfunktionen im Prozeß aus, wie zum Beispiel Öffnen oder Schließen von Ventilen und das Messen von Prozeßparametern. Die Prozeßsteuereinrichtung empfängt Signale, die die Größen des Prozeßablaufes anzeigen, die von den Anlageneinrichtungen stammen, oder andere Informationen der Anlageneinrichtungen darstellen, verwendet diese Informationen zur Umsetzung eines Steuerprogramms und empfängt Steuersignale, die über die Busverbindungen zu den Anlageneinrichtungen geleitet werden, um den Betriebsablauf des Prozesses zu steuern bzw. zu regeln. Die Informationen von den Anlageneinrichtungen und der Steuereinrichtung sind üblicherweise einer oder mehreren Anwendungen, die von der Bedienungsworkstation ausgeführt werden, zugänglich, um es dem Bediener zu ermöglichen, jede gewünschte Funktion in Bezug auf den Prozeß auszuführen, so z. B. die Überwachung des Ist-Zustandes des Prozesses, Modifizierung des Prozeßablaufes, etc.

**[0003]** In der Vergangenheit wurden herkömmliche Anlageneinrichtungen verwendet, um analoge (beispielsweise 4–20 Milliampere) Signale zu und von der Prozeßsteuereinrichtung über eine analoge Busverbindung oder analoge Kanäle zu leiten. Diese 4–20 Milliampere Signale waren in ihrer Art derart begrenzt, daß sie die von der Einrichtung gemachten Messungen anzeigten oder Steuersignale waren, die von der Steuereinrichtung erzeugt wurden, um den Betrieb der Einrichtung zu steuern. Jedoch wurden ungefähr in der letzten Dekade intelligente Anlageneinrichtungen mit einem Mikroprozessor und einem Speicher in der Prozeßsteuerungsindustrie vorherrschend. Zusätzlich zur Ausführung einer Primärfunktion im Prozeß speichern die intelligenten Anlageneinrichtungen zu der Einrichtung gehörende Daten, kommunizieren mit der Steuereinrichtung und/oder

anderen Einrichtungen in einem digitalen oder kombinierten digitalen und analogen Format und führen weitere Aufgaben aus, wie beispielsweise Selbstkalibrierung, Identifikation, Diagnose und dergleichen. Eine Anzahl standardisierter und offener intelligenter Einrichtungskommunikationsprotokolle, wie zum Beispiel das HART<sup>®</sup>-, PROFIBUS<sup>®</sup>-, WORLDFIP<sup>®</sup>-, Device-Net<sup>®</sup>- und CAN-Protokoll, wurden entwickelt, um es intelligenten Anlageneinrichtungen, die von verschiedenen Herstellern erzeugt wurden, zu ermöglichen, im gleichen Prozeßsteuernetz miteinander verwendet zu werden.

**[0004]** Desweiteren gab es in der Prozeßsteuerungsindustrie die Tendenz, die Prozeßsteuerfunktionen zu dezentralisieren. Zum Beispiel verwendet das voll digitalisierte Zweileitungs-Busverbindungsprotokoll, das von der Fieldbus Foundation propagiert wurde und als FOUNDATION<sup>™</sup> Fieldbus bekannt ist (nachfolgend "Fieldbus" genannt), Protokoll-Funktionsblöcke, die in unterschiedlichen Anlageneinrichtungen angeordnet sind, um Steueraufgaben auszuführen, die vorher in einer zentralisierten Steuereinrichtung ausgeführt wurden. Im Besonderen ist jede Fieldbus-Anlageneinrichtung in der Lage, einen oder mehrere Funktionsblöcke einzubeziehen und anzusteuern, von denen jeder Eingangssignale von anderen Funktionsblöcken empfängt und/oder Ausgangssignale liefert (sei es in der gleichen Einrichtung oder in anderen Einrichtungen). Jeder Funktionsblock führt Prozeßsteuerbefehle, wie zum Beispiel Messung oder Erfassung eines Prozeßparameters, Steuerung einer Einrichtung oder Durchführung eines Steuerbefehls, wie zum Beispiel Umsetzung einer proportional derivativ integralen Steueroutine (PID), aus. Die verschiedenen Funktionsblöcke in einem Prozeßsteuersystem sind derart ausgelegt, daß sie miteinander kommunizieren (z. B. über eine Busverbindung), um eine oder mehrere Prozeßsteuerschleifen bzw. Regelkreise zu bilden, deren individuelle Operationen über den ganzen Prozeß verteilt und somit dezentralisiert sind.

**[0005]** Prozeßsteuereinrichtungen sind typischerweise dazu programmiert, verschiedene Algorithmen, Subroutinen oder Steuerschleifen (die alle Steuerrountinen sind) für jede einer Anzahl von verschiedenen Schleifen, die zu dem Prozeß definiert oder in ihm enthalten sind, wie zum Beispiel Durchfluß-Steuerschleifen, Temperatur-Steuerschleifen, Druck-Steuerschleifen etc. auszuführen. Im allgemeinen enthält jede derartige Steuerschleife einen oder mehrere Eingangssignalblöcke, wie zum Beispiel einen analogen Eingabefunktionsblock (AI), einen Einzelausgabesteuerblock, wie zum Beispiel einen proportional integral derivativen Steuerfunktionsblock (PID), oder einen Fuzzy-Logik-Steuerfunktionsblock, und einen Einzelausgabeblock, wie zum Beispiel einen analogen Ausgabefunktionsblock (AO). Diese Steuerschleifen führen typischerweise

Einzeleingabe-/Einzelausgabesteuerfunktionen aus, da der Steuerblock ein einzelnes Ausgangssignal erzeugt, um ein einzelnes Prozeßeingangssignal zu steuern, wie zum Beispiel eine Ventilposition oder dergleichen. Jedoch ist in bestimmten Fällen die Verwendung einer Anzahl von unabhängig arbeitenden Einzeleingabe-/Einzelausgabesteuerschleifen nicht sehr effektiv, da die zu steuernden Prozeßgrößen von mehr als einem einzelnen Prozeßeingangssignal beeinflusst sind und jedes Prozeßeingangssignal den Zustand vieler Prozeßausgangssignale tatsächlich beeinflusst. Als Beispiel hierfür kann ein Prozeß gewählt werden, bei dem ein Tank über zwei Einläufe gefüllt und über einen einzelnen Auslauf geleert wird, wobei die Zu- und Abläufe von verschiedenen Ventilen gesteuert werden und wobei die Temperatur, der Druck und der Durchsatz im Tank gesteuert werden, um die gewünschten Werte zu erreichen. Wie oben beschrieben kann die Steuerung des Durchsatzes, der Temperatur und des Drucks im Tank mit einer getrennten Durchsatzsteuerschleife, einer getrennten Temperatursteuerschleife und einer getrennten Drucksteuerschleife ausgeführt werden. Jedoch kann bei dieser Anordnung die Bedienung der Temperatursteuerschleife mit dem Ändern der Einstellung eines der Einlaßventile zur Steuerung der Temperatur im Tank verursachen, daß der Druck im Tank ansteigt, was zum Beispiel die Drucksteuerschleife veranlaßt, das Auslaßventil zu öffnen und den Druck zu erniedrigen. Dies kann die Durchsatzsteuerschleife dazu veranlassen, eines der Einlaßventile zu schließen und dadurch die Temperatur zu beeinflussen und zu verursachen, daß die Temperatursteuerschleife eine weitere Maßnahme vornimmt. Wie aus diesem Beispiel ersichtlich, verursachen die Einzeleingabe-/Einzelausgabesteuerschleifen, daß die Prozeßausgangssignale (in diesem Fall Durchsatz, Temperatur und Druck) schwanken, ohne jemals einen Gleichgewichtszustand zu erreichen, was unerwünscht ist.

**[0006]** Modellvorhersagende Steuerungen oder andere Arten von fortschrittlichen Steuerungen wurden in der Vergangenheit verwendet, um derartige Vorgänge zu steuern. Im allgemeinen ist eine modellvorhersagende Steuerung eine Mehrfacheingabe-/Mehrfachausgabesteuerstrategie, bei der die Auswirkung der Änderung von jeweils einer Anzahl von Prozeßeingangssignalen auf jeweils eine Anzahl von Prozeßausgangssignalen gemessen wird und diese gemessenen Reaktionen verwendet werden, um ein Modell des Prozesses zu erzeugen. Das Modell des Prozesses wird mathematisch invertiert und als Mehrfacheingabe-/Mehrfachausgabesteuereinrichtung verwendet, um die Prozeßausgangssignale auf der Grundlage der Änderungen der Prozeßeingangssignale zu steuern. In einigen Fällen umfaßt das Prozeßmodell eine Prozeßausgangssignalreaktionskurve für jedes Prozeßeingangssignal und diese Kurven können auf der Grundlage einer Rei-

he von beispielsweise pseudo-zufälligen Schritttänderungen erzeugt werden, die für jedes der Prozeßeingangssignale erzeugt werden. Diese Reaktionskurven können in bekannter Weise dazu verwendet werden, den Prozeß zu modellieren. Die modellvorhersagende Steuerung ist im Stand der Technik bekannt und daher werden ihre Eigenheiten hier nicht beschrieben. Die modellvorhersagende Steuerung ist allgemein in Qin, S. Joe und Thomas A. Badgwell, "An Overview of Industrial Model Predictive Control Technology", AIChE Conference, 1996, beschrieben.

**[0007]** In der Vergangenheit erforderte die Schaffung einer modellvorhersagenden Steuerung und die Einbindung dieser Steuerung in einem Prozeßsteuernetz eine erhebliche Zeitdauer und Anstrengungen und konnte extrem teuer sein. Normalerweise wurde zur Schaffung einer modellvorhersagenden Steuerung für einen bestimmten Prozeß ein Prozeßexperte, typischerweise ein externer Berater, beschäftigt, der zu der Anlage kam und sie oder die Prozeßbedienung überwachte. Nach der Auswahl der geeigneten Prozeßeingangs- und Ausgangssignale für die modellvorhersagende Steuerung saß der Experte im Kontrollraum und wies den Bediener an, eine Reihe von gestuften Eingangssignalwellenformen an jeden der ausgewählten Prozeßeingänge zu leiten und die Auswirkungen jedes dieser Eingangssignale auf jedes der ausgewählten Prozeßausgangssignale zu messen. Nach Sammlung all dieser Prozeßdaten übermittelte der Experte im allgemeinen die gesammelten Daten an ein nicht angeschlossenes System. Dort führte der Experte eine erste Routine durch, um die gesammelten Daten durchzusehen, um schlechte Daten auszusondern, wie beispielsweise Daten, die gesammelt wurden, als der Prozeß nicht normal ablief, beendet wurde oder bei welchen ein anderer Fehler vorhanden war, der verhinderte, daß die gesammelten Daten den normalen Ablauf des Prozesses darstellen. Das nicht angeschlossene System führte dann eine zweite Routine durch, die die kontrollierten Daten verwendete, um ein Modell des Prozesses zu erzeugen. Danach wurde das Modell des Prozesses invertiert oder in anderer bekannter Weise bearbeitet, um eine modellvorhersagende Steuereinrichtung für den Prozeß zu erhalten. Nachdem die modellvorhersagende Steuereinrichtung erzeugt war, mußte sie in das Prozeßsteuersystem eingefügt werden, was im allgemeinen bedeutete, daß ein Prozeßingenieur das Steuerprogramm bereits in das Steuersystem zu programmieren hatte, um jedes der bestimmten Steuereingangssowie -ausgangssignale der modellvorhersagenden Steuereinrichtung zuzuführen, so daß die modellvorhersagende Steuereinrichtung jedes der Steuerausgangssignale sowie Steuereingangssignale an den geeigneten Ort im Steuersystem leiten konnte, um die Steuerung durchzuführen. Obwohl einige Händler die gleichen Bezeichnungen für die modellvorhersagenden Steuereingangs- und -ausgangssigna-

le wie in der Prozeßsteueroutine oder im Prozeßsteuersystem verwendeten, war es in einigen Fällen notwendig, die Eingangs- und Ausgangssignale der modellvorhersagenden Steuerung an die Prozeßeingänge und -ausgänge gemäß der Definition im Prozeßsteuersystem anzupassen. Auf jeden Fall konnte der Schritt des Einbeziehens einer modellvorhersagenden Steuerung in ein Prozeßsteuersystem eine erhebliche Programmierarbeit erfordern.

**[0008]** Obwohl nach dem Stand der Technik allgemein bekannt, ist daher das Erzeugen eines Prozeßmodells aus den gesammelten Daten, das Erzeugen einer modellvorhersagenden Steuerung und das Einbeziehen dieser Steuerung in einen Prozeß langwierig und erfordert im allgemeinen das Hinzuziehen eines Experten und kann sehr teuer sein. Tatsächlich kann es mehrere Monate dauern und hunderttausende Dollar kosten, eine einzelne modellvorhersagende Steuerung für einen Prozeß zu erzeugen. Zum Nachteil des Prozeßbedieners können Änderungen im Prozeß, wie etwa die durch Alterung der Prozeßeinrichtungen verursachten, dafür verantwortlich sein, daß die vorhandene modellvorhersagende Steuerung obsolet wird oder nicht mehr an den Prozeß angepaßt ist, was bedeutet, daß der gesamte Vorgang wiederholt werden muß, um eine neue modellvorhersagende Steuerung zu erzeugen.

**[0009]** Da ferner die modellvorhersagende Steuerung typischerweise von einem nicht angeschlossenen System erzeugt wurde, war diese Steuerung im allgemeinen nicht in der gleichen Weise wie Einzelschleifen- oder andere Steuerrouinen, die von dem Steuersystem ausgeführt werden, in das Prozeßsteuersystem integriert und erforderte daher das Erstellen von speziellen grafischen Darstellungen, damit der Anwender oder Bediener den Zustand und Betrieb der modellvorhersagenden Steuerung betrachten konnte. Daher war es schwierig, modellvorhersagende Steuerungen in Prozeßsteuersysteme, wie zum Beispiel das von Fisher-Rosemount Systems Inc. vertriebene DeltaV™-Steuersystem, einzubeziehen, die eine Prozeßsteuerungsanzeigeeinrichtung haben, die in den Betriebsablauf der Steuerblöcke oder Steuerschleifen in der Steuereinrichtung integriert ist. Tatsächlich liefert das DeltaV™-System verschiedene Ansichten, wie für einen Ingenieur, einen Bediener oder dergleichen, die den Betrieb des Prozesses einem Anwender darstellen. Einmal installiert, werden diese Ansichten durch den Betriebsablauf der Funktionsblöcke, die beispielsweise in der Prozeßsteuerung ausgeführt werden, automatisch auf den neuesten Stand gebracht. Um jedoch eine Ansicht oder eine andere Informationsbildschirmdarstellung für eine modellvorhersagende Steuerung hinzuzufügen, die von einem anderen nicht angeschlossenen System entworfen wurde, war es notwendig, spezielle grafische Anzeigen zu erzeugen, und zwar übli-

cherweise in einem anderen Format als dem von dem DeltaV™-System verwendeten.

**[0010]** Die Druckschrift WO 97/42553 A1 beschreibt ein Verfahren zum Modellieren von dynamischen und stationären Prozessen für die Vorhersage, Steuerung und Optimierung. Hierzu werden ein unabhängiges statisches Modell und ein unabhängiges dynamisches Modell verwendet.

**[0011]** Die Druckschrift WO 96/34324 A1 erläutert ein Feedback Verfahren zum Steuern eines nichtlinearen Prozesses. Hierbei verwendet ein Prozessor Daten zum Beschreiben eines Prozessmodells, wobei vorhergesagte Werte für eine Steuerungsvariable bereitgestellt werden.

**[0012]** Die Druckschrift DE 69409250 T2 beschreibt einen Störungsprädiktor auf der Basis eines neuronalen Netzes zur modellprädiktiven Regelung.

**[0013]** Die Druckschrift US 5477449 A bezieht sich auf ein hierarchisches modellvorhersagendes Steuerungssystem, das ein dezentralisiertes Steuersystem mit zahlreichen PID Reglern und einem modellvorhersagendem Steuermittel umfasst.

**[0014]** Die Druckschrift US 5920478 A beschreibt eine Multi-Input Multi-Output Steuerung, die als multi-variable Steuerung mit adaptiven Eigenschaften konfiguriert ist.

**[0015]** Die Druckschrift JP 2954660 B2 bezieht sich auf eine modellvorhersagende Steuerung, die ein dynamisches Modelleinschätzungsmittel bereitstellt.

**[0016]** Während die zuvor genannten Probleme für modellvorhersagende Steuerungen bestehen, bestehen die gleichen oder ähnliche Probleme in der Entwicklung und beim Gebrauch anderer fortschrittlicher Mehrfacheingabe-/Mehrfachausgabesteuerblöcke oder -systeme, wie etwa modellbildende Systeme oder Steuersysteme mit neuronalen Netzwerken oder Fuzzy-Logik-Steuerungen mit mehreren Variablen, Echtzeitoptimierungssysteme und dergleichen.

**[0017]** Es ist Aufgabe der Erfindung, ein Prozeßsteuersystem mit fortschrittlichen Steuerblöcken zu schaffen, bei dem die nach dem Stand der Technik auftretenden Probleme vermieden werden.

**[0018]** Die Lösung der Aufgabe ergibt sich aus den Patentansprüchen. Unteransprüche beziehen sich auf bevorzugte Ausführungsformen der Erfindung.

**[0019]** Gemäß einem Aspekt der Erfindung setzt ein fortschrittlicher Steuerblock Mehrfacheingabe-/Mehrfachausgabesteuerfunktionen, wie zum Beispiel modellvorhersagende Steuerung, Modellbildung oder Steuerung in einem neuronalen Netzwerk, und der-

gleichen, in einem Prozeßsteuersystem in einer Weise um, bei der die Integration in die implementierten Steuerblöcke unter Verwendung eines Steuerparadigmas, wie zum Beispiel eines Fieldbus-Paradigmas, erfolgt. Der fortschrittliche Steuerblock kann durch Erstellen eines Steuerblocks begonnen werden, der die gewünschten Eingänge und Ausgänge hat, die mit den jeweiligen Prozeßeingängen und -ausgängen verbunden werden um einen Prozeß zu steuern. Der Steuerblock kann so ausgelegt sein, daß er bestimmungsgemäß schließlich beispielsweise eine vollständige modellvorhersagende Steuereinrichtung beinhaltet, hat jedoch anfangs eine Datensammelroutine und einen Wellenformgenerator, die ihm zugeordnet sind. Wenn gewünscht, kann der Steuerblock eine Steuerlogik haben, die nicht abgestimmt ist oder anderweitig nicht entwickelt ist, da dieser Logik Abstimmparameter, Matrix-Koeffizienten oder andere Steuerparameter fehlen, die zur Implementierung erforderlich sind. Der Steuerblock ist in dem Prozeßsteuersystem angeordnet, wobei die definierten Ein- und Ausgänge in dem Steuersystem in der Weise in Kommunikationsverbindung verbunden sind, wie die Ein- und Ausgänge verbunden wären, wenn der fortschrittliche Steuerblock verwendet würde, um den Prozeß zu steuern. Während eines Testlaufs regt der Steuerblock systematisch jeden der Prozeßeingänge über die Ausgangssignale des Steuerblocks unter Verwendung von Wellenformen an, die vom Wellenformgenerator erzeugt werden, der speziell für die Verwendung bei der Entwicklung eines Prozeßmodells gestaltet ist. Dann koordiniert der Steuerblock über die Steuerblockeingänge die Sammlung der Daten, die die Reaktion jedes der Prozeßausgangssignale auf jede der erzeugten Wellenformen, die an jeden der Prozeßeingänge geleitet wurden, darstellen. Diese Daten können zum Beispiel zur Speicherung an einen Archivdatenspeicher geleitet werden.

**[0020]** Nachdem ausreichend Daten gesammelt wurden, wird ein Ablauf zur Prozeßmodellierung durchgeführt, bei dem ein Prozeßmodell beispielsweise unter Verwendung einer Routine zur Bildung eines Prozeßmodells für eine modellvorhersagende Steuereinrichtung aus den gesammelten Daten gebildet wird. Danach wird eine Routine zur Bestimmung der Logikparameter der fortschrittlichen Steuerblöcke, zum Bilden und Entwickeln der von der Steuerlogik benötigten Parameter angewendet, um den Prozeß zu steuern. Die Steuerlogikparameter und, wenn benötigt, das Prozeßmodell werden dann in den Steuerblock heruntergeladen, um das Erstellen des fortschrittlichen Steuerblocks zu vervollständigen, so daß der fortschrittliche Steuerblock, mit den Parametern der fortschrittlichen Steuerlogik und dem Prozeßmodell darin, zur Steuerung des Prozesses verwendet werden kann.

**[0021]** Der fortschrittliche Steuerblock kann im gleichen Format wie oder den gleichen Programmierpa-

radigmen wie andere Steuerblöcke in dem Prozeßsteuersystem entsprechend entworfen werden und kann daher die gleichen grafischen Abbildungen unterstützen, die auch von anderen Blöcken (oder Elementen) in dem Prozeßsteuerprogramm unterstützt werden. Daher kann der fortschrittliche Steuerblock eine oder mehrere grafische Abbildungen haben, die einem oder mehreren Nutzern angezeigt werden, und er kann während des Betriebs des fortschrittlichen Steuerblocks Daten an diese Abbildungen leiten.

**[0022]** Desweiteren kann das Prozeßmodell, das durch den Prozeßmodellierungsvorgang erzeugt wurde, dazu verwendet werden, den Ablauf des Prozesses zu simulieren und/oder den Dialog des Prozesses mit dem fortschrittlichen Steuerblock zu simulieren. In einem Fall kann ein Prozeßsimulationsblock aus dem bestimmten Prozeßmodell gebildet und in Kommunikationsverbindung mit dem erzeugten fortschrittlichen Steuerblock gebracht werden, um den Betrieb des fortschrittlichen Steuerblocks zu testen, bevor der fortschrittliche Steuerblock verwendet wird, um einen tatsächlichen Prozeß zu steuern. In einem anderen Fall kann ein Prozeßsimulationsblock, der unter Verwendung einer geänderten Version des bestimmten Prozeßmodells erzeugt wurde, dazu verwendet werden, die Alterung oder andere Änderungen innerhalb des Prozesses wiederzugeben. Dieser Simulationsblock kann in Kommunikationsverbindung mit dem fortschrittlichen Steuerblock gebracht werden, um den Betrieb des fortschrittlichen Steuerblocks bei der Anwesenheit von Änderungen im Prozeß zu simulieren, um damit die Leistung des fortschrittlichen Steuerblocks in der Anwesenheit von Abweichungen vom Prozeßmodell zu bestimmen. In einem weiteren Fall kann ein Simulationsblock, der aus dem Prozeßmodell entwickelt ist, in Verbindung mit dem Prozeß betrieben und dazu verwendet werden, virtuelle Prozeßausgangssignale zu erzeugen, die als Eingangssignale für den fortschrittlichen Steuerblock verwendet werden, wenn zum Beispiel ein Sensor versagt, der die Messwerte eines tatsächlichen Prozeßausgangssignals erfaßt. Die simulierten Prozeßausgangssignale können auch mit den tatsächlichen Prozeßausgangssignalen verglichen werden, um den Grad der Abweichung zwischen dem Prozeß und dem Prozeßmodell, das zur Erzeugung des fortschrittlichen Steuerblocks verwendet wurde, zu bestimmen, das heißt die Abweichung zwischen Prozeß und Prozeßmodell.

**[0023]** Nachfolgend wird eine Ausführungsform der Erfindung anhand der Zeichnung näher beschrieben.

**[0024]** Fig. 1 zeigt ein Blockdiagramm/schematisches Diagramm eines Prozeßsteuersystems, in dem ein fortschrittlicher Steuerblock erzeugt und verwendet werden kann;

**[0025]** Fig. 2 ist ein Flußdiagramm, das den Betrieb und die Erzeugung eines fortschrittlichen Steuerblocks im Prozeßsteuersystem von Fig. 1 zeigt;

**[0026]** Fig. 3 ist ein Blockdiagramm eines modellvorhersagenden Steuerblocks, der zur Steuerung eines Prozesses in eine Prozeßsteueroutine eingebunden ist;

**[0027]** Fig. 4A und Fig. 4B sind Blockdiagramme eines modellvorhersagenden Steuerfunktionsblocks, der zur Koordinierung einer bestehenden Steuerstrategie mit Funktionsblöcken in einer Prozeßsteueroutine verbunden ist;

**[0028]** Fig. 5 ist ein Blockdiagramm eines modellvorhersagenden Steuerfunktionsblocks, der zur Koordinierung von Einzelschleifen-Steueroutinen mit Funktionsblöcken in einer Prozeßsteueroutineverbunden ist;

**[0029]** Fig. 5 zeigt ein Beispiel für eine Bildschirmanzeige, die von einem prozeßmodellierenden Tool zur Entwicklung eines fortschrittlichen Steuerblocks erzeugt wird;

**[0030]** Fig. 7 ist ein Blockdiagramm/schematisches Diagramm eines Teils des Prozeßsteuersystems aus Fig. 1, wobei die mit einem fortschrittlichen Steuerblock verbundenen grafischen Ansichten verdeutlicht sind;

**[0031]** Fig. 8 zeigt ein Blockdiagramm eines modellvorhersagenden Steuerblocks, der mit einem Prozeßsimulationsblock verbunden ist;

**[0032]** Fig. 9 zeigt ein Beispiel für eine Bildschirmanzeige, die von einem prozeßmodellierenden Tool erzeugt wird, das zur Entwicklung eines fortschrittlichen Steuerblocks oder zur Entwicklung eines Prozeßsimulationsblocks verwendet wird;

**[0033]** Fig. 10 ist eine beispielhafte Bildschirmanzeige, die von einem prozeßmodellierenden Tool erzeugt wird, das zur Entwicklung eines fortschrittlichen Steuerblocks oder zur Entwicklung eines Prozeßsimulationsblocks verwendet wird; und

**[0034]** Fig. 11 zeigt ein Blockdiagramm eines modellvorhersagenden Steuerblocks, der sowohl mit einem Prozeßblock als auch einem Prozeßsimulationsblock verbunden ist, der den Betrieb des Prozesses simuliert, um virtuelle Prozeßausgangssignale zu erzeugen.

**[0035]** Fig. 1 zeigt ein Prozeßsteuersystem 10, das eine Prozeßsteuereinrichtung 11 enthält, die mit einem Archivspeicher 12 und mit einer oder mehreren Host-Workstations oder Computern 13 (die jede Art von Personal Computer, Workstation etc. sein kön-

nen) verbunden ist, von welchen jeder einen Anzeigebildschirm 14 hat. Die Steuereinrichtung 11 ist ferner mit Anlageneinrichtungen 15 bis 22 über Eingabe-/Ausgabe-(I/O)-Karten 26 und 28 verbunden. Der Archivspeicher 12 kann jede gewünschte Art von Datensammeleinheit sein mit jeder gewünschten Art eines Speichers oder jeder gewünschten oder bekannten Software, Hardware oder Firmware zur Speicherung von Daten, und er kann, wie in Fig. 1 dargestellt, getrennt oder als Teil einer der Workstations 13 vorhanden sein. Die Steuereinrichtung 11, die zum Beispiel die von Fisher-Rosemount Systems Inc. vertriebene DeltaV™-Steuereinrichtung sein kann, steht beispielsweise über eine Ethernet-Verbindung oder jedes andere gewünschte Kommunikationsnetzwerk in Kommunikationsverbindung mit den Hostcomputern 13 und dem Archivspeicher 12. Die Steuereinrichtung 11 steht auch in Kommunikationsverbindung mit den Anlageneinrichtungen 15 bis 22 unter Verwendung jeder gewünschten Hardware und Software, die beispielsweise den standardisierten 4–20 Milliampere-Einrichtungen oder einem beliebigen intelligenten Verbindungsprotokoll, wie z. B. dem Fieldbus-Protokoll, dem HART-Protokoll etc. zugehörig ist.

**[0036]** Die Anlageneinrichtungen 15 bis 22 können jede Art von Einrichtungen sein, wie zum Beispiel Sensoren, Ventile, Übertragungsmittel, Stalleinrichtungen etc., während die I/O-Karten 26 und 28 jede Art von I/O-Einrichtungen sein können, die zu den gewünschten Kommunikations- oder Steuerprotokollen passen. In der in Fig. 1 dargestellten Ausführungsform sind die Anlageneinrichtungen 15 bis 18 Standard- 4–20 Milliampere-Einrichtungen, die über analoge Kanäle mit der I/O-Karte 26 kommunizieren, während die Anlageneinrichtungen 19 bis 22 intelligente Einrichtungen sind, wie zum Beispiel Fieldbus-Anlageneinrichtungen, die über digitale Busverbindungen mit der I/O-Karte 28 unter Verwendung von Fieldbus-Protokollverbindungen kommunizieren. Allgemein ausgedrückt ist das Fieldbus-Protokoll ein volldigitales, serielles, Zwei-Wege-Kommunikationsprotokoll, das eine standardisierte physikalische Schnittstelle für eine Zweidrahtschleife oder Busverbindung, die mit den Anlageneinrichtungen verbunden ist, zur Verfügung stellt. Das Fieldbus-Protokoll bildet tatsächlich ein lokales Netzwerk für die Anlageneinrichtungen in einem Prozeß, das es diesen Anlageneinrichtungen ermöglicht, Prozeßsteuerfunktionen (unter Verwendung von Funktionsblöcken, die entsprechend dem Fieldbus-Protokoll definiert sind) an über die Prozeßanlage verteilten Orten auszuführen, und vor und nach der Ausführung der Prozeßsteuerfunktionen miteinander zu kommunizieren, um eine generelle Steuerstrategie umzusetzen. Es ist klar, daß, obwohl das Fieldbus-Protokoll ein relativ neues volldigitales Verbindungsprotokoll ist, das zur Verwendung in Prozeßsteuernetzwerken entwickelt wurde, dieses Protokoll im Stand der Technik

bekannt ist und in zahlreichen Artikeln, Broschüren und Beschreibungen ausführlich beschrieben ist, die unter anderem von der Fieldbus Foundation, einer gemeinnützigen Organisation mit Sitz in Austin, Texas, veröffentlicht, verteilt und zugänglich gemacht wurden. Daher wird das Fieldbus Kommunikationsprotokoll hier nicht im Detail beschrieben. Natürlich können die Anlageneinrichtungen **15** bis **22** auch jedem anderen gewünschten Standard oder Protokoll entsprechen, einschließlich jeglicher Standards oder Protokolle, die in Zukunft entwickelt werden.

**[0037]** Die Steuereinrichtung **11** implementiert ein oder mehrere Prozeßsteuerroutinen, die darin gespeicherte oder in anderer Weise damit verbundene Steuerschleifen bzw. Regelkreise enthalten, und kommuniziert zur Steuerung bzw. Regelung eines Prozesses in jeder gewünschten Weise mit den Einrichtungen **15** bis **22**, dem Hostcomputer **13** und dem Archivspeicher **12**. Es soll festgehalten werden, daß jede Steueroutine oder jedes Steuerelement, das hier beschrieben wird, Teile haben kann, die von unterschiedlichen Steuereinrichtungen oder anderen Einrichtungen umgesetzt oder ausgeführt werden, wenn es gewünscht ist. Daher können die im Prozeßsteuersystem **10** umzusetzenden Steuer Routinen oder -elemente, die hier beschrieben sind, jede Form, einschließlich Software, Firmware, Hardware etc. annehmen. Für die Zwecke dieser Erfindung kann ein Prozeßsteuerelement jeder Teil oder Abschnitt eines Prozeßsteuersystems sein, einschließlich zum Beispiel einer Routine, eines Blocks oder eines Moduls, die auf jedem computerlesbaren Medium gespeichert sind. Steuer Routinen, die ein Modul oder irgendein Teil eines Steuervorgangs sein können, wie zum Beispiel eine Subroutine, Teile einer Subroutine (wie Zeilen des Codes) etc., können in jedem gewünschten Softwareformat umgesetzt werden, wie beispielsweise unter Verwendung von Leiterlogik, sequentieller Funktionsdiagramme, Funktionsblockdiagramme oder jeder anderen Programmiersprache oder Entwurfsparadigmen. Dementsprechend können die Steuer Routinen auf zum Beispiel einem oder mehreren EPROMs, EEPROMs, anwendungsspezifischen integrierten Schaltkreisen (ASICs) oder beliebigen anderen Hardware- oder Firmware-Elementen fest codiert sein. Desweiteren können die Steuer Routinen unter Verwendung beliebiger Entwurfstools gebildet sein, einschließlich grafischer Entwurfstools oder jeder anderen Art von Software/Hardware/Firmware-Programmier- oder -gestaltungstools. Daher kann die Steuereinrichtung **11** so konfiguriert sein, daß sie eine Steuerstrategie oder eine Steueroutine in jeder gewünschten Weise umsetzt.

**[0038]** In einer Ausführungsform setzt die Steuereinrichtung **11** eine Steuerstrategie unter Verwendung von allgemein als Funktionsblöcke bezeichneten Elementen um, wobei jeder Funktionsblock ein Teil (zum Beispiel eine Subroutine) einer Gesamtsteuerrou-

ne ist und zur Umsetzung von Prozeßsteuerschleifen im Prozeßsteuersystem **10** in Verbindung mit anderen Funktionsblöcken (über Kommunikationsverbindungen genannte Verbindungen) arbeitet. Funktionsblöcke führen typischerweise entweder eine Eingabefunktion aus, wie zum Beispiel eine, die mit einem Geber, einem Sensor oder einer anderen Einrichtung zur Messung eines Prozeßparameters verbunden ist, eine Steuerfunktion, wie zum Beispiel die, die mit einer Steueroutine verbunden ist, die eine PID-, Fuzzy-Logik-Steuerung oder dergleichen ausführt, oder eine Ausgabefunktion, die den Betrieb einer Einrichtung steuert, wie zum Beispiel eines Ventils, um eine physikalische Funktion im Prozeßsteuersystem **10** auszuführen.

**[0039]** Selbstverständlich existieren Hybrid- und andere Arten von Funktionsblöcken. Funktionsblöcke können in der Steuereinrichtung **11** gespeichert und ausgeführt werden, was typischerweise der Fall ist, wenn diese Funktionsblöcke mit Standard 4–20 Milliampere-Einrichtungen und einigen Arten von intelligenten Anlageneinrichtungen, wie zum Beispiel HART-Einrichtungen, verwendet werden oder diesen zugehörig sind. Sie können auch in den Anlagen-einrichtungen selbst gespeichert und umgesetzt werden, was bei Fieldbus-Einrichtungen der Fall sein kann. Während die Beschreibung des Steuersystems hier unter Verwendung einer Funktionsblocksteuerstrategie erfolgt, können die Steuerstrategie oder die Steuerschleifen oder die Module auch unter Verwendung von anderen Normen, wie zum Beispiel der Leiter-Logik, sequentiellen Funktionsdiagrammen etc., oder unter Verwendung jeder anderen gewünschten Programmiersprache oder -paradigmen entworfen und umgesetzt werden.

**[0040]** Wie im vergrößerten Block **30** in **Fig. 1** gezeigt, kann die Steuereinrichtung **11** eine Anzahl von Einzelschleifen-Steuer Routinen enthalten, die als Routinen **32** und **34** bezeichnet sind, und kann auf Wunsch eine oder mehrere fortschrittliche Steuerschleifen umsetzen, die als Steuerschleife **36** dargestellt ist. Jede der Schleifen wird üblicherweise als ein Steuermodul bezeichnet. Die Einzelschleifen-Steuer Routinen **32** und **34** sind so dargestellt, daß sie eine Signalschleifensteuerung ausführen unter Verwendung eines Einzeleingabe-/Einzelausgabe-Fuzzy-Logik-Steuerblocks und eines Einzeleingabe-/Einzelausgabe-PID-Steuerblocks, die mit geeigneten analogen Eingangs-(AI)- und analogen Ausgangs-(AO)-Funktionsblöcken verbunden sind, die mit Prozeßsteuereinrichtungen, wie zum Beispiel Ventilen, Meßeinrichtungen, wie zum Beispiel Temperatur- und Druckgebern, oder mit jeder anderen Einrichtung im Prozeßsteuersystem **10** verbunden sein können. Die fortschrittliche Steuerschleife **36** ist so dargestellt, daß sie einen fortschrittlichen Steuerblock **38** enthält, dessen Eingänge in Kommunikationsverbindung mit zahlreichen AI-Funktionsblö-



cken stehen und dessen Ausgänge in Kommunikationsverbindung mit zahlreichen AO-Funktionsblöcken stehen. Die Ein- und Ausgänge des Steuerblocks **38** können aber auch mit beliebigen gewünschten Funktionsblöcken oder Steuerelementen verbunden sein und andere Arten von Eingangssignalen erhalten und andere Arten von Steuerausgangssignalen liefern. Der fortschrittliche Steuerblock **38** kann jede Art von Mehrfacheingabe-/Mehrfachausgabesteuerblock sein, der zur Steuerung zweier oder mehrerer Prozeßausgangssignale verwendet wird, indem Steuersignale an zwei oder mehr Prozeßeingänge abgegeben werden. Während der fortschrittliche Steuerblock **38** im folgenden als ein modellvorhersagender Steuerblock (MPC) beschrieben wird, kann der fortschrittliche Steuerblock **38** auch jede andere Art von Mehrfacheingabe-/Mehrfachausgabeblock sein, wie zum Beispiel ein modellbildender Block oder Steuerblock in einem neuronalen Netzwerk, ein Fuzzy-Logik-Steuerblock mit mehreren Variablen, ein Echtzeit-Optimierungsblock etc. Es ist klar, daß die in **Fig. 1** dargestellten Funktionsblöcke einschließlich des fortschrittlichen Steuerblocks **38** von der Steuereinrichtung **11** ausführbar sein oder alternativ dazu in jeder anderen Prozeßeinrichtung angeordnet und ausgeführt werden können, wie zum Beispiel einer der Workstations **13** oder sogar einer der Anlageneinrichtungen **19** bis **22**.

**[0041]** Wie in **Fig. 1** dargestellt, enthält eine der Workstations **13** ein Routine **40** zur Erzeugung eines fortschrittlichen Steuerblocks, die dazu verwendet wird, den fortschrittlichen Steuerblock **38** in einer im folgenden näher beschriebenen Weise zu erzeugen, zu laden und umzusetzen. Während die Routine **40** zur Erzeugung des fortschrittlichen Steuerblocks in einem Speicher in der Workstation gespeichert sein kann und von einem Prozessor darin ausführbar ist, kann die Routine (oder jeder Teil davon) zusätzlich oder alternativ dazu in jeder anderen Einrichtung im Prozeßsteuersystem **10** gespeichert und ausgeführt werden, wenn es erwünscht ist. Im allgemeinen beinhaltet die Routine **40** zur Erzeugung eines fortschrittlichen Steuerblocks eine Routine **42** zur Erzeugung eines Steuerblocks, die einen fortschrittlichen Steuerblock erzeugt und die diesen fortschrittlichen Steuerblock in das Prozeßsteuersystem einbindet, eine Routine **44** zur Prozeßmodellbildung, die ein Prozeßmodell für den Prozeß oder einen Teil davon auf der Grundlage der von dem fortschrittlichen Steuerblock erfaßten Daten erzeugt, und eine Routine **46** zur Erzeugung eines Steuerlogikparameters, die einen Steuerlogikparameter für den fortschrittlichen Steuerblock aus dem Prozeßmodell erzeugt und die diesen Steuerlogikparameter in den fortschrittlichen Steuerblock zur Steuerung des Prozesses speichert oder lädt. Es ist klar, daß die Routinen **42**, **44** und **46** aus einer Reihe verschiedener Routinen bestehen können, wie zum Beispiel einer ersten Routine, die ein fortschrittliches Steuerelement mit Steuerein-

gängen erzeugt, die dazu geeignet sind, Prozeßausgangssignale zu empfangen, und mit Steuerausgängen, die dazu geeignet sind, Steuersignale an Prozeßeingänge abzugeben, eine zweite Routine, die es einem Benutzer ermöglicht, das fortschrittliche Steuerelement mit der Prozeßsteueroutine (die jede gewünschte Konfigurationsroutine sein kann) in Kommunikationsverbindung zu bringen, eine dritte Routine, die das fortschrittliche Steuerelement dazu verwendet, Anregungswellenformen an jeden der Prozeßeingänge abzugeben, eine vierte Routine, die das fortschrittliche Steuerelement dazu verwendet, Daten zu erfassen, die die Reaktion jedes Prozeßausgangs auf die wellenförmigen Anregungen wiederzugeben, eine fünfte Routine, die ein Prozeßmodell aus den gesammelten Daten erzeugt, eine sechste Routine, die aus dem Prozeßmodell fortschrittliche Steuerlogikparameter entwickelt, und eine siebte Routine, die die fortschrittliche Steuerlogik und, wenn notwendig, das Prozeßmodell in das fortschrittliche Steuerelement einbindet, um es dem fortschrittlichen Steuerelement zu ermöglichen, den Prozeß zu steuern.

**[0042]** In **Fig. 2** stellt ein Flußdiagramm **50** die Schritte des Erzeugens und der Anwendung eines fortschrittlichen Steuerblocks, insbesondere eines MPC-Steuerblocks, in einem Prozeßsteuersystem, wie etwa dem Prozeßsteuersystem **10** aus **Fig. 1**, dar. Während das Flußdiagramm **50** aus **Fig. 2** die Erzeugung eines MPC-Blocks oder -Moduls zeigt, können die gleichen oder ähnliche Schritte ausgeführt werden, um einen beliebigen anderen fortschrittlichen Steuerblock, wie zum Beispiel einen beliebigen Mehrfacheingabe-/Mehrfachausgabesteuerblock, wie etwa einen modellbildenden Block oder einen Steuerblock in einem neuronalen Netzwerk, einen Fuzzy-Logik-Steuerblock mit mehreren Variablen etc., zu erstellen und zu verwenden.

**[0043]** Zunächst wird zu einem bestimmten Zeitpunkt **52** eine Entscheidung getroffen, durch Umsetzung eines MPC-Ablaufes die Steuerung im Prozeßsteuersystem **10** vorzusehen oder zu verbessern. Die Entscheidung kann zu dem Zeitpunkt getroffen werden, zu dem das Prozeßsteuersystem erstmalig installiert wird, oder eine gewisse Zeit später, zum Beispiel wenn festgestellt wird, daß andere Steuerroutinen, wie etwa Einzelschleifensteuerroutinen, mangelhaft steuern. Zum Zeitpunkt **52** führt ein Bediener oder ein anderer Benutzer die Routine **40** zur Erzeugung eines MPC-Blocks aus, um damit zu beginnen, ein MPC-Modul oder eine Steuerschleife im Prozeßsteuersystem zu erzeugen. Als Teil dieses Prozesses wählt der Bediener die Prozeßeingänge, an die die Ausgänge des im Entstehen befindlichen MPC-Blocks angeschlossen werden sollen, und wählt die Prozeßausgänge, an die die Eingänge des im Entstehen befindlichen MPC-Blocks angeschlossen werden sollen. Während der MPC-Block jede Anzahl von Ein- und Ausgängen haben

kann, hat jeder MPC-Block im wesentlichen drei Arten von Eingängen, darunter Eingänge für die geregelten Parameter, die die Prozeßvariablen oder -parameter sind, die auf einem Sollwert (oder innerhalb eines eingestellten Bereiches) zu halten sind, beschränkte Eingänge, die die Prozeßvariablen sind, die auf der Grundlage von beispielsweise physikalischen Beschränkungen des Prozesses auf einen bestimmten Grenzwert oder Wertebereich beschränkt sind und die vom MPC-Block nicht zwangsweise aus dem beschränkten Bereich oder Grenzwert gebracht werden dürfen, und Prozeßstörameingänge, die andere Prozeßvariable sind, von denen bekannt ist, daß, wenn sie geändert werden, Änderungen in den geregelten Parametern verursacht werden. Der MPC-Block verwendet die Prozeßstörameingänge zur Vorhersage von Änderungen der gesteuerten Parameter, das heißt der geregelten Prozeßausgänge, und zur Begrenzung der Auswirkungen dieser Änderungen, bevor sie auftreten. Weitere Eingänge können an dem MPC-Block vorgesehen sein, wie zum Beispiel Rückkopplungen von einer Einrichtung oder einem anderen Prozeßelement, das gesteuert wird, die es dem MPC-Steuerblock ermöglichen, eine effektivere Steuerung dieser Elemente auszuführen. In ähnlicher Weise können die Ausgänge des MPC-Blocks derart angeschlossen sein, daß jede gewünschte Prozeßvariable oder andere Prozeßeingangssignale, einschließlich Steuerschleifeneingangssignale, Einstellungssteuerungseingaben etc. gesteuert werden können. Die Routine, die durch den Anschluß des MPC-Blocks an andere Steuerelemente entwickelt wurde, wird hier als MPC-Modul bezeichnet. Während der Benutzer einen MPC-Funktionsblock erzeugen kann, kann er auch einen ursprünglichen Funktionsblock aus einem Speicher, wie zum Beispiel einer Bibliothek von Funktionsblöcken, entnehmen und diesen Funktionsblock verwenden oder eine Instanz dieses Funktionsblocks zur Verwendung in dem Prozeßsteuersystem erzeugen. Entsprechend kann der Nutzer oder ein anderer Anwender einen Funktionsblock oder ein anderes Steuerelement in jeder gewünschten Weise vorsehen.

**[0044]** In Schritt 54 erzeugt der Bediener ein MPC-Modul mit einem MPC-Block (der noch nicht alle notwendigen Informationen hat, um eine modellvorhersagende Steuerung zu liefern) mit den angegebenen Ein- und Ausgängen, die in Kommunikationsverbindung mit dem Prozeßsteuersystem verbunden sind, und lädt den Block oder das Modul auf die geeignete Steuereinrichtung oder eine andere Einrichtung, die das MPC-Modul umsetzen wird. Als Teil dieses Vorgangs konfiguriert der Bediener das Prozeßsteuersystem 10 zur Umsetzung des MPC-Blocks, indem die Ausgänge des MPC-Blocks mit den geeigneten Prozeßeingängen in Kommunikationsverbindung gesetzt werden und indem die Eingänge des

MPC-Blocks mit den geeigneten Prozeßausgängen in Kommunikationsverbindung gesetzt werden.

**[0045]** In Fig. 3 ist ein MPC-Block 56 dargestellt, der an einen Prozeß 58 angeschlossen ist. Der MPC-Block 56 ist ein  $3 \times 3$  Steuerblock mit drei Eingängen  $IN_1$ – $IN_3$  und drei Ausgängen  $OUT_1$ – $OUT_3$ , während der Prozeß 58 Eingänge  $X_1$ – $X_5$  und Ausgänge  $Y_1$ – $Y_6$  hat. Der MPC-Block 56 und der Prozeß 58 können auch jede andere Anzahl von Ein- und Ausgängen haben. Während der MPC-Block 56 im allgemeinen ein quadratischer Block sein kann, das heißt mit der gleichen Anzahl von Ein- und Ausgängen, ist diese Konfiguration nicht unbedingt notwendig und der MPC-Block 56 kann auch eine unterschiedliche Anzahl von Ein- und Ausgängen haben. Wie in Fig. 3 dargestellt schließt der Bediener die Prozeßausgänge  $Y_1$ – $Y_3$  jeweils an die MPC-Block-Eingänge  $IN_1$ – $IN_3$  in Kommunikationsverbindung an und schließt die MPC-Block-Ausgänge  $OUT_1$ – $OUT_3$  jeweils an die Prozeßeingänge  $X_1$ – $X_3$  in Kommunikationsverbindung an. Jeder der Ein- und Ausgänge des Prozesses 58 kann selbstverständlich an andere Steuerschleifen oder an andere Elemente in anderen Steuerprogrammen, die mit dem Steuerprozeßsystem 10 verbunden sind, angeschlossen werden, wie durch die gepunkteten Linien an den Prozeßein- und -ausgängen in Fig. 3 dargestellt. Im allgemeinen werden der MPC-Block 56 und andere Blöcke, die Steuereingaben an den Prozeß 58 abgeben (wie durch die gepunkteten Linien an den Prozeßeingängen  $X_1$ – $X_3$  angedeutet), über Schalter einer beliebigen Art angeschlossen. Die Schalter sind durch Kästchen 59 in Fig. 3 dargestellt. Die Schalter 59 können Hardware- oder Softwareschalter sein. Wenn gewünscht können sie derart vorgesehen sein, daß unterschiedliche Steuereingangssignale an verschiedene Eingänge eines Funktionsblocks, wie etwa eines Fieldbus-Funktionsblocks geleitet werden, der dann zwischen dem Steuersignal des MPC-Blocks 56 und einem Steuersignal von einem anderen Funktionsblock, wie zum Beispiel einem PID-Funktionsblock, auf der Grundlage des Modus des Funktionsblocks, der die beiden Signale empfängt, auswählt.

**[0046]** Selbstverständlich kann der Bediener den MPC-Block 56 in jeder gewünschten Weise mit dem Prozeß 58 verbinden, und verwendet allgemein ausgedrückt das gleiche Steuerkonfigurations- oder Entwicklungsprogramm, das der Bediener verwendet, um andere Steuerschleifen, wie etwa Einzelschleifensteuerprogrammen, im Prozeßsteuersystem 10 zu erstellen. Zum Beispiel kann der Benutzer jede gewünschte Grafikprogrammerroutine verwenden, um die Verbindungen zwischen dem MPC-Block 56 und den Prozeßein- und -ausgängen anzugeben. Auf diese Weise wird der MPC-Block 56 in gleicher Art wie andere Steuerblöcke, -elemente oder -routinen unterstützt, wodurch die Konfiguration und der Anschluß des MPC-Blocks 56 und die Unterstützung dieses

Blocks im Steuersystem **10** sich nicht von der Konfiguration, Steuerung und Unterstützung anderer Blöcke im System unterscheiden. In einer Ausführungsform sind der MPC-Block **56** sowie die anderen Blöcke im Steuersystem **10** Funktionsblöcke, die gleich oder ähnlich wie Fieldbus-Funktionsblöcke gebildet sind. Bei dieser Ausführung kann der MPC-Block **56** die gleichen Arten von Ein- und Ausgängen etc. haben wie im Fieldbus-Protokoll angegeben oder vorgesehen, und ist in der Lage, von beispielsweise der Steuereinrichtung **11** unter Verwendung von Kommunikationsverbindungen, die gleich oder ähnlich den im Fieldbus-Protokoll angegebenen sind, umgesetzt zu werden. Ein Verfahren zur grafischen Erzeugung von Prozeßsteuerungen und Elementen davon ist in Dove et al., US-Patent Nr. 5,838,563 mit dem Titel "System for Configuring a Process Control Environment" beschrieben. Es können auch andere Strategien zum Entwurf von Steuerschleifen oder Steuermodulen verwendet werden, einschließlich solcher, die andere Arten von Funktionsblöcken verwenden oder andere Routinen, Subroutinen oder Steuerelemente in einem Prozeßsteuerkonfigurationsparadigma verwenden.

**[0047]** Wenn ein Steuersystem auf der Grundlage der Verbindung von Funktionsblöcken verwendet wird, wie etwa solche, die durch das Fieldbus-Funktionsblockparadigma vorgesehen sind, kann der MPC-Block **56** direkt mit anderen Funktionsblöcken in der Prozeßsteueroutine verbunden werden. Zum Beispiel kann der MPC-Block **56** mit Steuereinrichtungen, wie zum Beispiel Ventilen etc., direkt verbunden werden, indem ein Steuerausgang des MPC-Blocks **56** mit einem Ausgangsblock (wie zum Beispiel einem AO-Block) verbunden wird, der zu der zu steuernden Einrichtung gehört. Entsprechend kann der MPC-Block **56** Steuersignale an Funktionsblöcke in anderen Steuerschleifen, wie etwa an die Eingänge anderer Steuerfunktionsblöcke, abgeben, um den Betrieb dieser Steuerschleifen zu überwachen oder zu korrigieren.

**[0048]** Fig. 4A und Fig. 4B zeigen als Beispiel einen MPC-Funktionsblock **60** des Fieldbus-Typs, der mit anderen Funktionsblöcken des Fieldbus-Typs in einem Prozeßsteuersystem verbunden ist, um bestehende Strategien mit mehreren Variablen, die von Einzelschleifensteuerungen umgesetzt werden, zu koordinieren. Insbesondere hat der MPC-Funktionsblock **60** einen ersten Ausgang OUT1, der an den Eingang RCAS\_IN (ferngesteuerte Kaskade) eines AO-Blocks **62** angeschlossen ist, der zu einem zu betätigenden Ventil gehört, und einen zweiten Ausgang OUT2, der an den Eingang RCAS\_IN (ferngesteuerte Kaskade) eines PID-Funktionsblocks **64** angeschlossen ist. Desweiteren hat der MPC-Block **60** einen ersten Eingang I1 (der ein gesteuerter Parametereingang ist), der Signale von einem AI-Funktionsblock **66** erhält, einen zweiten Eingang I2 (der auch

ein gesteuerter Parametereingang ist), der Signale von einem AI-Funktionsblock **68** in Fig. 4B erhält, und einen dritten Eingang (der ein Störparametereingang ist), der Signale von einem AI-Funktionsblock **70** erhält. Die AI-Funktionsblöcke **66**, **68** und **70** können zu Anlageneinrichtungen, wie zum Beispiel Sensoren, gehören und von diesen gemessene Signale abgeben, die an die Steueroutine über einen Geber oder andere Einrichtungen übertragen werden. Der MPC-Block **60** empfängt ferner Rückkopplungen an Eingängen BKCAL\_IN1 und BKCAL\_IN2 (Rückkalibrierungseingänge) von den ferngesteuerten Kaskadenausgängen (RCAS\_OUT) des AO-Funktionsblocks **62** und des PID-Funktionsblocks **64** zur Verwendung bei der Bestimmung der Auswirkungen der Steuersignale, die vom MPC-Block **60** an die Funktionsblöcke **62** und **64** geleitet werden. Der Ausgang des AI-Funktionsblocks **66** wird auch an einen Eingang eines PID-Funktionsblocks **72** abgegeben, der ein Steuersignal an den Kaskadeneingang (CAS\_IN) des AO-Funktionsblocks **62** abgibt und ein Rückkopplungssignal vom Ausgang OUT des AO-Funktionsblocks **62** am BKCAL\_IN-Eingang des PID-Funktionsblocks **72** empfängt, um damit das betätigte Ventil während des normalen Prozeßablaufs zu steuern, das heißt ohne den MPC-Betrieb. Entsprechend gibt der AO-Funktionsblock **68** sein Ausgangssignal, das ein Prozeßausgangssignal ist, an den Auto-Eingang eines PID-Funktionsblocks **74** ab, der ein Steuersignal an den Kaskadeneingang des PID-Funktionsblocks **64** leitet. Der PID-Funktionsblock **74** empfängt auch ein Rückkopplungssignal des PID-Funktionsblocks **64** am BKCAL\_IN-Eingang des Funktionsblocks **74**. Die Ein- und Ausgänge der Funktionsblöcke in Fig. 4A und Fig. 4B sowie in Fig. 5 sind dem Fieldbus-Protokoll entsprechend definiert und arbeiten entsprechend den Definitionen und Bestimmungen des Fieldbus-Protokolls.

**[0049]** Die Funktionsblöcke **66**, **72** und **62** bilden eine erste Einzelschleifensteueroutine, während die Funktionsblöcke **68**, **74** und **64** eine zweite Einzelschleifensteueroutine bilden, die beide während des regulären oder automatischen Ablaufs des Prozesses arbeiten können, um eine Einzelschleifensteuerung zu erhalten. Der MPC-Block **60** kann jedoch die Steuerung des AO-Funktionsblocks **62** (und der zugehörigen Einrichtung) übernehmen sowie die Steuerung der Schleife, die zu dem PID-Funktionsblock **64** gehört, indem Steuereingangssignale an die ferngesteuerten Kaskadeneingänge des AO-Funktionsblocks **62** und des PID-Funktionsblocks **64** abgegeben werden. Dadurch arbeiten diese Blöcke im ferngesteuerten Modus (anstatt im automatischen Modus), und daher verwenden sie im Betrieb die Steuereingangssignale an den ferngesteuerten Eingängen anstatt an den Auto-Eingängen. Während des Betriebs im ferngesteuerten Modus ignorieren die Funktionsblöcke **62** und **64** die Eingangssignale der jeweiligen PID-Funktionsblöcke **72** und **74**. Auf diese Wei-

se kann der MPC-Block **60** an die Blöcke **62** und **64** angeschlossen werden und steuert sie, wobei dies an- und abschaltbar ist. Wenn die Blöcke **62** und **64** nicht durch den MPC-Funktionsblock **60** gesteuert werden, werden sie immer noch jeweils von den Blöcken **72** und **74** gesteuert, das heißt entsprechend einer Einzelschleifensteuerstrategie.

**[0050]** In gleicher Weise stellt **Fig. 5** einen MPC-Funktionsblock **80** dar, der mit einer Prozeßsteuer-routine zur Koordinierung von Einzelschleifenroutinen verbunden ist. Insbesondere empfängt der MPC-Funktionsblock **80** gesteuerte Parametereingangssignale von den AI-Funktionsblöcken **82** und **84** und ein Störparametereingangssignal vom AI-Funktionsblock **86**. Der MPC-Funktionsblock **80** liefert ein Steuerausgangssignal an einen AO-Funktionsblock **90**, der zu einem betätigten Ventil gehört, und leitet ein Sollwertausgangssignal an den Kaskadeneingang (CAS\_IN) eines PID-Funktionsblocks **92** in einer Steuerschleife **94**. Der AO-Funktionsblock **90** und der PID-Funktionsblock **92** liefern Rückkalibrierungsausgangssignale an die Rückkalibrierungseingänge des MPC-Funktionsblocks **80**. Die Steuerschleife **94** enthält ferner einen AI-Funktionsblock **96**, der ein Steuerparametereingangssignal (das heißt ein Prozeßausgangssignal) an den Auto-Eingang des PID-Funktionsblocks **92** abgibt, der dann ein Steuerausgangssignal an den AO-Funktionsblock **98** leitet, der zum Beispiel zu einem anderen Ventil oder einer anderen Einrichtung gehört. Der AO-Funktionsblock **98** gibt eine Rückkopplung an den Rückkalibrierungseingang des PID-Funktionsblocks **92** ab. In der Anordnung in **Fig. 5** steuert der MPC-Funktionsblock **80** das Ventil, das zu dem AO-Funktionsblock **90** gehört, direkt und steuert den Betrieb der Schleife **94** durch Beeinflussung des Sollwerts dieser Schleife. Jedoch arbeitet die Schleife **94** weiter, wenn der MPC-Funktionsblock **80** in Betrieb ist. Daher steuert der MPC-Funktionsblock **80** die Einrichtung, die zu dem AO-Funktionsblock **98** gehört, indirekt, steuert aber die Steuerschleife **94** direkt. Die MPC-Blöcke können selbstverständlich in einer Prozeßsteuer-routine in jeder anderen gewünschten Weise zur direkten oder indirekten Steuerung von Einrichtungen oder anderen Steuerelementen verbunden sein. Desweiteren können die Steuer-routinen oder -module unter Verwendung jeder Technik, einschließlich grafischer oder nichtgrafischer Programmier-techniken, entwickelt werden.

**[0051]** Es versteht sich daher, daß die Prozeßeingaben  $X_1$  bis  $X_3$ , an die die Ausgänge des MPC-Steuerblocks **56** in **Fig. 3** angeschlossen sind, jede Art von gewünschten Prozeßeingaben sein können, einschließlich Eingangssignalen für Steuerschleifen, die in bestehenden Steuerstrategien festgelegt sind, oder Eingangssignalen für Ventile oder andere Einrichtungen des Prozesses. Ebenso können die Ausgangssignale  $Y_1$  bis  $Y_3$ , die an die Eingänge des

MPC-Blocks **56** angeschlossen sind, jede Art von gewünschten Prozeßausgaben sein, einschließlich Ausgangssignalen von Ventilen oder anderen Sensoren, Ausgaben von AO- oder AI-Funktionsblöcken oder Ausgaben von anderen Steuerelementen oder Routinen.

**[0052]** In Schritt **54** in **Fig. 2** wird, nachdem der Bediener ein Steuermodul erzeugt hat, das einen ursprünglichen MPC-Block enthält, der Aus- und Eingänge hat, die jeweils mit den gewünschten Prozeßein- und -ausgängen verbunden sind, das Steuermodul mit dem ursprünglichen MPC-Block darin zur Ausführung auf eine entsprechende Einrichtung geladen, wie auf die Steuereinrichtung **11** oder eine der Workstations **13**. Dann weist bei einem Schritt **99** der Bediener den ursprünglichen MPC-Block an, den Prozeß in bekannter Weise anzuregen und Prozeßeingangs- und -ausgangsdaten zu sammeln, während der Prozeß angeregt wird.

**[0053]** Wie **Fig. 3** zeigt, umfaßt der ursprüngliche MPC-Block **56** eine Datensammelroutine **100**, einen Wellenformgenerator **101**, eine auswählbare Steuerlogik **102** und einen Speicher zur Speicherung von Steuerparametern **103** und eines Prozeßmodells **104**. Die auswählbare Logik **102** kann zum Beispiel eine auswählbare MPC-Routine sein, die Koeffizienten oder andere Steuerparameter benötigt, um zum Betrieb in der Lage zu sein und die Steuerung in einem bestimmten Fall ausführen zu können. In einigen Fällen kann die auswählbare Logik **102** auch ein Prozeßmodell für den zu steuernden Prozeß benötigen, um den Prozeß zu steuern. Nachdem er zum Beispiel auf die Steuereinrichtung **11** geladen wurde, wird der ursprüngliche MPC-Block **56** über die MPC-Erzeugungsroutine **42** angewiesen, die nächste Phase der Entwicklung eines MPC-Blocks **56** zu beginnen, in der Daten jedes Prozeßausgangssignals für die Verwendung in der Erzeugung eines Prozeßmodells gesammelt werden. Insbesondere beginnt der Wellenformgenerator **101** des MPC-Blocks **56**, wenn er entsprechend vom Bediener angewiesen wird oder zu jeder anderen gewünschten Zeit, eine Reihe von Wellenformen an den Ausgängen  $OUT_1$ – $OUT_3$  zu erzeugen, um Anregungswellenformen an jeden der Prozeßeingänge  $X_1$ – $X_3$  zu leiten. Wenn gewünscht können diese Wellenformen dem Generator **101** über Software in der Workstation **13** des Nutzers zu Verfügung gestellt werden, vorzugsweise werden sie jedoch vom Generator **101** erzeugt. Die Wellenformen, die vom Wellenformgenerator **101** erzeugt werden, sind vorzugsweise derart ausgebildet, daß sie verursachen, daß der Prozeß über unterschiedliche Wertebereiche der erwarteten Eingangssignale während des normalen Ablaufs des Prozesses arbeitet. Zur Entwicklung eines Prozeßmodells für eine MPC-Steuer-routine kann der Wellenformgenerator **101** an jeden der Prozeßeingänge  $X_1$ – $X_3$  eine Reihe von unterschiedlichen Impulssätzen leiten, wobei die Impul-

se in jedem dieser Impulssätze die gleiche Amplitude haben, aber pseudo-zufallsverteilte Längen, wobei die Impulse in den verschiedenen Impulssätzen unterschiedliche Amplituden haben. Eine derartige Reihe von Impulssätzen kann für jeden der verschiedenen Prozeßeingänge  $X_1$ – $X_3$  erzeugt und dann aufeinanderfolgend jeweils einzeln an diese geleitet werden. Während dieser Zeit sammelt die Datensammeleinheit **100** in dem MPC-Block **56** Daten oder koordiniert auf andere Weise das Sammeln der Daten, die die Reaktoren der Prozeßausgänge  $Y_1$ – $Y_3$  auf jede der vom Wellenformgenerator **101** erzeugten Wellenformen anzeigen, und sie kann die Daten sammeln oder das Sammeln der Daten koordinieren, die zu den erzeugten Anregungswellenformen gehören. Diese Daten können im MPC-Block **56** gespeichert werden, vorzugsweise werden sie aber automatisch an den Archivspeicher **12** zur Speicherung und/oder an die Workstation **13** gesendet, wo diese Daten auf dem Anzeigebildschirm **14** dargestellt werden können.

**[0054]** Anstatt zu versuchen, den Prozeß **58** unter Verwendung einer fortschrittlichen Steuerlogik zu steuern (die bis jetzt noch nicht vollständig entwickelt ist), leitet so der MPC-Block **56** zuerst eine Reihe von Anregungswellenformen an den Prozeß **58** und mißt die Reaktion des Prozesses **58** auf diese Anregungswellenformen. Selbstverständlich können die Anregungswellenformen, die vom Wellenformgenerator **101** erzeugt werden, jede gewünschte Wellenform sein, die entwickelt werden, um ein Prozeßmodell zu erzeugen, das für die Bildung von Steuerlogikparametern für jede fortschrittliche Steueroutine nützlich ist. In diesem Beispiel erzeugt der Wellenformgenerator **101** jeden Satz von Wellenformen, von welchen bekannt ist, daß sie bei der Entwicklung eines Prozeßmodells für eine modellvorhersagende Steuereinrichtung nützlich sind. Diese Wellenformen können jede bisher bekannte oder noch in Zukunft für diesen Zweck zu entwickelnde Form annehmen. Da Wellenformen, die dazu verwendet werden, einen Prozeß zum Zwecke des Sammelns von Daten anzuregen, um ein Prozeßmodell zur modellvorhersagenden Steuerung zu entwickeln, bekannt sind, werden diese Wellenformen hier nicht weiter beschrieben. Entsprechend kann jede andere oder jede gewünschte Art von Wellenformen mit dem Wellenformgenerator **101** erzeugt werden, um damit Prozeßmodelle für andere fortschrittliche Steueroutinen, die das Modellieren beinhalten, zu entwickeln, wie zum Beispiel Steueroutinen für neuronale Netze, für Fuzzy-Logik mit mehreren Variablen und dergleichen.

**[0055]** Es ist zu beachten, daß der Wellenformgenerator **101** jede gewünschte Ausbildung haben kann und zum Beispiel in Hardware, Software oder einer Kombination daraus umgesetzt sein kann. Wenn er in Software umgesetzt ist, kann der Wellenformgenerator **101** einen Algorithmus speichern, der dazu verwendet wird, die gewünschten Wellenformen zu er-

zeugen, kann er eine digitale Darstellung der zu erzeugenden Wellenform speichern oder kann er jedes andere Programm oder gespeicherte Daten zur Erzeugung derartiger Wellen verwenden. Wenn er in Hardware umgesetzt ist, kann der Wellenformgenerator **101** beispielsweise die Form eines Oszillators oder eines Rechteckwellengenerators haben. Wenn es gewünscht ist, kann der Bediener aufgefordert werden, bestimmte Parameter einzugeben, die zur Festlegung der Wellenform notwendig sind, wie die ungefähre Reaktionszeit des Prozesses, die Schrittgröße der Amplitude der an die Prozeßeingänge zu leitenden Wellenformen und dergleichen. Der Bediener kann nach diesen Informationen abgefragt werden, wenn der MPC-Block **56** das erste Mal erzeugt wird oder wenn der Bediener den MPC-Block **56** anweist, den Prozeß zu beginnen oder anzuregen und Prozeßdaten zu sammeln. In einer bevorzugten Ausführungsform sammelt die Datensammeleinrichtung **100** Daten (oder stellt deren Erfassung anderweitig sicher), die über das Drei- bis Fünffache der vom Bediener eingegebenen Ansprechzeit in Reaktion auf jede der Anregungswellenformen erfaßt wurden, um sicherzustellen, daß ein vollständiges und genaues Prozeßmodell entwickelt werden kann. Jedoch können die Daten auch für jede andere Zeitspanne gesammelt werden.

**[0056]** Auf jeden Fall arbeitet der MPC-Block **56** vorzugsweise bis der Wellenformgenerator **101** jede der notwendigen Anregungswellenformen an jeden der Prozeßeingänge  $X_1$ – $X_3$  geleitet hat und die Datensammeleinrichtung **100** Daten für die Prozeßausgänge  $Y_1$ – $Y_3$  gesammelt hat. Selbstverständlich kann der Betrieb des MPC-Blocks **56**, wenn es gewünscht wird oder während des Datensammelvorgangs notwendig ist, unterbrochen werden.

**[0057]** Fig. 6 zeigt eine Bildschirmanzeige **118**, die dem Bediener auf einer der Anzeigen **14** durch die Routine **40** zur Erzeugung einer Steuerlogik angezeigt werden kann. Es ermöglicht dem Bediener, die unterschiedlichen Schritte der Erzeugung eines fortschrittlichen Steuerblocks umzusetzen. Insbesondere beinhaltet die Bildschirmanzeige **118** ein Datenanzeigefeld **120** und drei Schaltflächen **122**, **123** und **124**, die dazu verwendet werden können, verschiedene Teile der Routine **40** zur Erzeugung eines fortschrittlichen Steuerblocks zu starten. Die Testschaltfläche **122** ermöglicht es dem Bediener, den ursprünglichen MPC-Block **56** zu veranlassen, Anregungssignale an den Prozeß **58** zu senden und Eingangs- und Ausgangsdaten zur Übermittlung an den Archivspeicher **12** zu sammeln. Die Schaltfläche **122** kann zum Beispiel die Zeit anzeigen, die zur Durchführung der Anregungsroutine verbleibt, das heißt die Zeit, die der MPC-Steuerblock **56** braucht, um alle Anregungswellenformen zu erzeugen und die Prozeßdaten, die ansprechend auf diese Wellenformen erzeugt werden, zu sammeln. Vor dem Drücken der

Schaltfläche **122** kann der Benutzer eine Ansprechzeit eingeben, die einer typischen Zeit entspricht, die der Prozeß benötigt, um auf ein Eingangssignal anzusprechen, und kann die Schrittgröße angeben oder festlegen, die vom MPC-Block **56** zur Erzeugung von Anregungswellenformen verwendet werden soll, wobei diese Daten an den Wellenformgenerator **101** des MPC-Blocks **56** geleitet werden können. Nach dem Drücken der Schaltfläche **122** können die von dem MPC-Block **56** gesammelten Daten auch auf dem Datenanzeigefeld **120** angezeigt werden, und, wenn gewünscht, kann der Nutzer die Daten, die nicht zur Erzeugung eines Prozeßmodells verwendet werden sollen, markieren. Es ist selbstverständlich, daß die Datensammeleinheit **100** diese Daten derart sammeln kann, daß sichergestellt ist, daß die Daten an den Archivspeicher **12** oder eine andere Speichereinrichtung zur Speicherung geleitet werden.

**[0058]** Danach, wie in **Fig. 2** bei Schritt **125** dargestellt, kann der Bediener an einem bestimmten Punkt entscheiden, daß die nächste Phase der Entwicklung des MPC-Blocks durch Ausführung des Prozeßmodellierungsroutine **44** umgesetzt wird, die auf die gesammelten Daten in dem Archivspeicher **12** zugreift und eine beliebige bekannte Routine zur Erzeugung eines Prozeßmodells verwendet, um ein Prozeßmodell aus den gesammelten Daten zu erzeugen. Im allgemeinen kann der Bediener diese Phase durch Betätigung der Schaltfläche "Steuerung erstellen" **123** auf der Bildschirmanzeige aus **Fig. 6** beginnen.

**[0059]** Wenn gewünscht, kann die Prozeßmodellierungsroutine **44** einen Datenprüfvorgang der gesammelten Daten durchführen. Dieser Datenprüfvorgang kann die gesammelten Daten auf Ausreißer und andere offensichtlich fehlerhafte Daten kontrollieren und kann andere mit den gesammelten Daten verknüpfte Werte überprüfen, wie zum Beispiel Zustands- und Grenzwerte, die zu den gesammelten Daten gehören, um zu bestimmen, ob die Daten von einem Funktionsblock in schlechtem oder ungeeignetem Zustand erzeugt wurden, ob die Daten an einem Grenzwert liegen, ob die Daten erzeugt wurden, als ein Funktionsblock oder ein anderes Element in einem ungeeigneten Modus waren oder ob die Daten auf andere Weise unter abnormalen oder unerwünschten Prozeßbedingungen erzeugt wurden. Zum Beispiel im Feldbus-Kommunikationsprotokoll beinhalten die von Funktionsblöcken erzeugten Daten auch eine Status-, eine Grenzwert- und eine Modusangabe, die zusammen mit den Daten im Archivspeicher **12** speicherbar und zur Kontrolle der Daten verwendbar sind. Wenn gewünscht, kann die Datenprüfroutine die gesammelten Daten dem Bediener auf dem Datenanzeigefeld **120** aus **Fig. 6** darstellen und es dem Bediener ermöglichen, die zu kontrollierenden oder auszusondernden Daten zu markieren. Dies erfolgt beispielsweise durch Hervorhebung oder andere Kenntlichmachung die-

ser Daten auf der Grundlage des Wissens des Bedieners über die Prozeßbedingungen. Auf diese Weise können Daten, die vom MPC-Block **56** gesammelt wurden, wenn der Prozeß **58** nicht angeschlossen war, wenn der Prozeß **58** nicht korrekt gesteuert war, wenn der Prozeß **58** gewartet wurde, wenn ein Sensor oder eine andere Einrichtung im Prozeß **58** fehlerhaft war oder ausgetauscht wurde etc., ausgewählt und aus den Daten entfernt werden, aus denen ein Prozeßmodell erzeugt wird.

**[0060]** Wie in **Fig. 6** dargestellt, kann eine Tendenz auf dem Anzeigefeld **120** dargestellt werden, die die MPC-Eingangs- und Ausgangssignale als ein Trenddiagramm enthält. Das Diagramm kann auf der Grundlage der Werte der Eingangs- und Ausgangssignale selbsttätig skaliert sein. Auch wird das Zeitfenster des dargestellten Abschnitts des Diagramms vorzugsweise der zweifachen festgelegten Ansprechzeit entsprechen. Mit Hilfe eines Schiebebalkens **126** kann das Zeitfenster verschoben werden, um Werte einer vorhergegangenen Zeit anzuzeigen, wie zum Beispiel aus den letzten zwei Tagen. Zur Sicherstellung, daß gute Daten beim Betrieb der Anlage gesammelt werden, kann eine automatisierte Testfunktion verwendet werden. Durch Betätigung der Testschaltfläche **122** werden die Prozeßeingangssignale, die von dem MPC-Block bearbeitet werden, mit der festgelegten Schrittgröße in eine pseudo-zufallsverteilte Folge über die festgelegte Ansprechzeit verteilt. Desweiteren wird, wenn die Testschaltfläche **122** betätigt wird, der Start- und der Endteilerbalken automatisch auf die Datenanzeige gesetzt, um den Start und das Ende des automatisierten Tests anzuzeigen, und der MPC-Block **56** übernimmt die Steuerung der bearbeiteten Ausgangssignale durch Weiterleitung der pseudo-zufallsverteilten Folge von Ausgangssignalen als Anregungswellenformen an den Prozeß **58**.

**[0061]** Die Zeitbalken oder das Datenfenster in dem Anzeigefeld **120** können auch dazu verwendet werden, Daten auszuwählen, mit denen das Prozeßmodell entwickelt wird. Ein Bediener kann einen der Teilerbalken auswählen und ihn an die gewünschte Start- oder Endzeit ziehen, um das Zeitfenster zu ändern, das für die Prozeßmodellidentifizierung in Betracht gezogen wird. Wenn ein Teil der Zeit zwischen dem Start- und dem Endbalken nicht repräsentativ für den normalen Betrieb der Anlage ist, kann der Nutzer oder Bediener diesen Zeitabschnitt, zu dem Datenwerte gesammelt werden, markieren, so daß er während des Vorgangs zur Prozeßmodellidentifizierung ignoriert wird. Ansprechend darauf kann das ausgewählte Gebiet mit einer dunkleren Hintergrundfarbe dargestellt und automatisch ausgenommen werden, wenn das Prozeßmodell erzeugt wird.

**[0062]** Nach Kontrolle der Daten erzeugt die Prozeßmodellierungsroutine **44** ein Prozeßmodell aus den ausgewählten Daten. Wie oben ausgeführt, kann

die Prozeßmodellierungsroutine **44** jede gewünschte oder bekannte Art von Prozeßmodellierungsanalyse durchführen, um aus den gesammelten und geprüften Daten ein Prozeßmodell zu entwickeln. Das entwickelte Prozeßmodell kann jede Form annehmen, wie zum Beispiel einen mathematischen Algorithmus, eine Reihe von Reaktionskurven und dergleichen.

**[0063]** Wenn die Prozeßmodellierungsroutine **44** Schwierigkeiten hat, ein Prozeßmodell zu bestimmen, kann ein Hinweis auf das Problem in einem Statusfeld auf der Benutzeranzeige wie der in **Fig. 6** gezeigten angezeigt werden. Ein Problem, das angezeigt werden kann ist, daß nicht ausreichend Proben vorhanden sind zur Festlegung oder Erzeugung eines Prozeßmodells. Eine Nachricht wie zum Beispiel "Für die festgelegte Konfiguration ist eine Mindestanzahl von XXX Proben notwendig. Der Datensatz enthält nur XXX Proben" kann erzeugt werden, um den Benutzer auf dieses Problem aufmerksam zu machen. Ein anderes Problem, das festgestellt werden kann, ist, daß nicht genügend Anregungen an den Prozeßeingängen erfolgten. Ein entsprechender Hinweis mit Identifizierung der Signalmarkierungsnamen, wie zum Beispiel Markierung-X, Markierung-Y etc., und die Mindestveränderung des Ausmaßes der Anregungen kann dem Bediener übermittelt werden, wenn ein derartiges Problem auftritt.

**[0064]** Wenn gewünscht und auf der Grundlage, daß die Bedingungen identifiziert wurden, die das erfolgreiche Bestimmen eines Modells verhinderten, kann der Nutzer den Zeitrahmen, über den die Prozeßmodellierung durchgeführt wird, ändern, oder er kann die Prozeßeingangssignale derart ändern, daß die von der Prozeßmodellierungsroutine **44** verwendeten Daten gültig sind. Das Prozeßmodell, das bestimmt wird, kann automatisch in jeder gewünschten Datenbank gespeichert werden, um für späteren Gebrauch zugänglich zu sein. Erfahrenere Nutzer könnten es wünschen, das Prozeßmodell, das bestimmt wurde, zu untersuchen oder zu überarbeiten. Durch Auswahl der Schaltfläche **124** "Fortgeschritten" auf dem Bildschirm aus **Fig. 6** kann dem Nutzer eine Auswahl gegeben werden, eine MPC-Steuereinrichtung aus einem ausgewählten Modell und der derzeitigen MPC-Funktionsblockkonfiguration zu erzeugen oder ein bestimmtes Modell zu bearbeiten und das erhaltene Modell als ein neues Modell zur Erzeugung einer MPC-Steuerlogik zu speichern. Wenn die Option zur Erzeugung einer Steuereinrichtung gewählt wird, kann dem Nutzer ein Dialog vorgelegt werden, aus dem er ein Modell auswählt, das vorher für den MPC-Block im MPC-Modul gespeichert wurde, das überarbeitet wird. Durch Auswahl der Überarbeitungsoption kann dem Nutzer eine Liste von Modellen dargestellt werden, die für das in Frage stehende MPC-Modul entwickelt wurden. Nach Auswahl eines Modells können dem Nutzer eine Anzeige mit einem Überblick über die Prozeßschrittreaktionen und an-

dere Bildschirmanzeigen gezeigt werden, wie im folgenden beschrieben, um Prozeßschrittreaktionen zu überarbeiten um ein neues oder geändertes Modul zu erzeugen.

**[0065]** An einem bestimmten Punkt im Prozeß kann die Routine **46** zur Erzeugung eines Logikparameters ausgeführt werden, um die Parameter zu erzeugen (die in den Variablen im MPC-Block **56** gespeichert werden), welche von der auswählbaren Logik **102** des ursprünglichen MPC-Blocks **56** benötigt werden, um eine modellvorhersagende Steuerung durchzuführen. Diese Steuerparameter, die zum Beispiel Matrix- oder andere MPC-Koeffizienten für eine MPC-Logik, Abstimmparameter, neuronale Netzparameter (für ein neuronales Netz), Skalierungsfaktoren (für Fuzzy-Logik mit mehreren Variablen) oder jede andere gewünschte Art von Parametern sein können, werden üblicherweise auf der Grundlage des erzeugten Prozeßmodells festgelegt. Die Routine **46** zur Erzeugung eines Logikparameters kann jede gewünschte oder bekannte Prozedur zur Erzeugung der Parameter aus dem Prozeßmodell ausführen. Im allgemeinen bringt dieser Prozeß das Invertieren des Prozeßmodells in ein Matrixformat mit sich. Jedoch können andere gewünschte Routinen zur Erzeugung von Logikparametern verwendet werden. Da die Eigenheiten der Erzeugung eines Prozeßmodells aus gesammelten Daten eines Prozesses und die Erzeugung von MPC- oder anderen Steuerlogikparametern aus diesem Prozeßmodell im Stand der Technik bekannt sind, werden diese Vorgehensweisen hier nicht weiter beschrieben. Es ist jedoch festzustellen, daß der Bediener einigen Einfluß auf die Erzeugung der Steuerlogikparameter für den MPC-Block **56** hat. Tatsächlich kann der Bediener aufgefordert oder auf andere Weise ermächtigt werden, die Werte bestimmter Variablen anzugeben, die üblicherweise zur Erzeugung einer MPC-Steuereinrichtung verwendet werden. Zum Beispiel kann der Benutzer die Sollwerte und Grenzwerte jeder der beschränkten Eingänge des MPC-Blocks festlegen, den Zeitrahmen, über welchen Änderungen in der Steuerung erfolgen sollen, das heißt den Sollwert-Trajektorien-Filter und die mit diesem Filter verknüpften Zeitkonstanten, die maximale oder minimale Bewegung (Verhältnissgrenze) eines MPC-Ausgangssignals oder eines Prozeßausgangssignals, ob einer der gesteuerten Parameter in integrierter Weise anspricht, MPC-Optimierungsfaktoren, Variable oder Abstimmungsparameter, den Horizont des MPC-Steuerblocks, das heißt wie viele Schritte zur Steuerung eines gewünschten Zustandes im voraus berechnet werden, die berechneten Einheitenbereiche für jeden der Ein- und Ausgänge des MPC-Blocks **56**, welche der bearbeiteten variablen Ziele gelockert oder nicht verwirklicht werden, wenn eine der Beschränkungen verletzt wird, eine Beschreibung und/oder ein Name jedes der MPC-Block-Eingangs- und Ausgangssignale, den Wert von Optimierungsvariablen, die gewählt werden können,

den Wert von Variablen bezogen auf das Ansprechverhalten oder die Widerstandsfähigkeit des MPC-Blocks und dergleichen. Wenn gewünscht, kann die Routine **46** zur Erzeugung einer Steuerlogik Standardwerte für einige oder alle dieser Variablen oder Einstellungen speichern und diese Standardwerte zur Erzeugung der MPC-Logik verwenden. Jedoch kann der Bediener oder jeder andere Nutzer in der Lage sein, die Einstellungen über die Benutzeranzeige **14** zu ändern.

**[0066]** Auf jeden Fall verwendet die Routine **46** zur Erzeugung von MPC-Logikparametern diese Informationen und alle anderen erforderlichen Informationen zur Erzeugung von MPC- oder anderen Steuerlogikparametern, wie zum Beispiel MPC-Koeffizienten. Die Schaltfläche "Steuerung erzeugen" **123** auf der Bildschirmanzeige **118** kann anzeigen, ob die Erzeugung eines Prozeßmodells und von Steuerlogikparametern erfolgreich war oder nicht.

**[0067]** Nachdem in Schritt **128** in **Fig. 2** die MPC-Steuerlogikparameter erzeugt sind, können die MPC-Steuerlogikparameter oder Koeffizienten mit einem Prozeßsimulationsblock getestet werden. Dieser Simulationsblock kann im allgemeinen aus dem für den Prozeß erzeugten Prozeßmodell entwickelt werden und an einen MPC-Block in einer Testumgebung angeschlossen werden, wie nachfolgend beschrieben wird, um zu testen, ob die erzeugte MPC-Steuerlogik über den normalen Betriebsbereich des Prozesses zufriedenstellend arbeitet. Wenn die MPC-Logik nicht zufriedenstellend ist, können einige oder alle der Schritte **54**, **99** und **125** wiederholt werden, um eine andere MPC-Steuerlogik zu entwickeln. Wenn jedoch die MPC-Steuerlogik zufriedenstellend ist, können die MPC-Steuerlogikparameter und das Prozeßmodell in einem Schritt **130** in den MPC-Block **56** zur Speicherung im Parameterspeicher **130** und in den Prozeßmodellspeicher **104** zur Verwendung bei der Steuerung des Prozesses **58** geladen werden. Auf diese Weise sind die von der MPC-Steuerlogik benötigten Parameter in dem MPC-Block **56** bereitgestellt und enthalten, und der MPC-Block **56** kann für den Betrieb kommissioniert werden oder entsprechend der MPC-Steuerlogik **102** die tatsächliche Steuerung des Prozesses ausführen. Wenn gewünscht, kann die tatsächliche MPC-Logik **102** zusammen mit den dafür benötigten Parametern in der Workstation **13** erzeugt und auf den MPC-Block **16** geladen werden.

**[0068]** Wenn das MPC-Modul oder die Schleife mit dem MPC-Block **56** darin einmal auf die Steuereinrichtung **11** geladen und ausgeführt ist, kann sie in der gleichen Weise wie andere Blöcke oder Elemente in der Steueroutine Berichtsfunktionen ausführen, da – wie oben ausgeführt – der MPC-Block **56** und das diesen Block enthaltende Steuermodul mit denselben Programmparadigmen wie die anderen Steuerblöcke in dem Prozeßsteuersystem **10** gebildet sind. In einer

Ausführungsform kann der MPC-Block oder das -Modul damit verknüpfte grafische Ansichten haben, die einem Nutzer oder Bediener beispielsweise über einen der Anzeigeschirme **14** einer oder mehrerer der Workstations **13** angezeigt werden, wobei diese Ansichten Daten abonnieren, die zu den Blöcken in dem MPC-Steuermodul gehören, und wobei diese Daten in einer vorherbestimmten oder festgelegten Weise angezeigt werden.

**[0069]** In **Fig. 7** sind als Beispiel Abschnitte des Prozeßsteuersystems **10** aus **Fig. 1** dargestellt, einschließlich der Steuereinrichtung **11**, die über eine nicht gesondert dargestellte Kommunikationsverbindung an die Nutzeranzeigen **14A** und **14B** und an die Einrichtungen **15** bis **22** angeschlossen ist. In der Steuereinrichtung **11** ist ein MPC-Modul **132** dargestellt, mit einem MPC-Funktionsblock, der Eingangssignale von einer Reihe von AI-Funktionsblöcken empfängt und jeweils Ausgangssignale an eine Reihe von AO-Funktionsblöcken leitet, während eine Einzelschleifensteueroutine **134** derart dargestellt ist, daß sie einen PID-Block hat, der ein Eingangssignal zur Steuerung eines AO-Funktionsblock von einem AI-Funktionsblock empfängt.

**[0070]** Unterschiedliche Ansichten des Betriebs dieser beiden Steuermodule, wie zum Beispiel eine Bedieneransicht und eine Technikeransicht sind auf den Anzeigebildschirmen **14A** und **14B** grafisch abgebildet. Insbesondere umfaßt eine Technikeransicht auf der Anzeige **14A** eine grafische Darstellung des Betriebs der Schleife **132** sowie eine grafische Darstellung der Schleife **134**, die erstellt wurden, um es dem Techniker zu ermöglichen, auf Informationen über diese Schleifen zuzugreifen und diese Schleifen zu bearbeiten. Dementsprechend ist eine Bedieneransicht mit einer grafischen Abbildung des Betriebs der Schleife **132** sowie einer grafischen Abbildung der Schleife **134** auf der Anzeige **14B** vorhanden, um dem Bediener zu ermöglichen, auf Informationen über diese Schleifen zuzugreifen und diese Schleifen zu bearbeiten. Jedoch können die zur Verfügung gestellten Informationen in der Bedieneransicht sich von den Informationen in der Technikeransicht unterscheiden, und die Möglichkeiten zur Beeinflussung der Schleifen **132** und **134** durch diese Ansichten können unterschiedlich sein. Zum Beispiel kann die Bedieneransicht es dem Bediener lediglich ermöglichen, die Sollwerte zu ändern und beschränkte Funktionen durchzuführen, während die Technikeransicht es dem Nutzer ermöglicht, Änderungen in der Einstellung einer Schleife vorzunehmen, die Programmierung der Funktionsblöcke zu ändern und dergleichen. Diese unterschiedlichen Ansichten können in Zusammenhang mit den Funktionsblöcken in einer Weise erzeugt werden, die ähnlich der Weise ist, die in Bezug auf die in dem US-Patent Nr. 5,594, 858 von Blevins "Uniform Control Template Generating System and Method for Process Control Pro-



gramming" erörterten Vorlagen beschrieben wurde. Es ist jedoch klar, daß die MPC-Blöcke und Module, die unter Verwendung dieser Blöcke erzeugt wurden, die gleiche Art der grafischen oder berichtenden Unterstützung wie andere Blöcke, Programme oder Elemente im Prozeßsteuersystem **10** liefern können, da der MPC-Block im Online-Zustand unter Verwendung der gleichen Programmstrategie wie bei den anderen Steuerblöcken erzeugt wurde. Dieses Merkmal beseitigt die Notwendigkeit der Bereitstellung spezieller Programmierungen und ermöglicht es dem Bediener, Techniker, Ingenieur etc. in einfacher Weise anzusehen, was im MPC-Steuermodul oder -block vor sich geht.

**[0071]** Wenn gewünscht, kann das MPC-Modul **132** jede gewünschte Information an einen Benutzer über eine vorherbestimmte Ansicht oder Anzeige berichten und es dem Nutzer oder Bediener ermöglichen, jede gewünschte Handlung vorzunehmen. Zum Beispiel kann dem Nutzer eine Bildschirmanzeige gezeigt werden, die Alarmmeldungen erläutert, die vom MPC-Modul **132** erzeugt wurden oder zu ihm gehören, die eine grafische Darstellung der gesteuerten, beschränkten oder der Störparameter (die auch zukünftige vorausberechnete Werte dieser gesteuerten und beschränkten Parameter sein können) anzeigt, die es einem Nutzer oder Bediener ermöglicht, die Ausführung des MPC-Moduls **132** zu steuern, beispielsweise unter Verwendung des Modusparameters (wenn das MPC-Modul **132** entwickelt ist, das zum Beispiel ein Feldbus-Protokoll nutzt), die numerisch oder mit einem Balkendiagramm die Werte der Sollwerte, Beschränkungen und der gesteuerten und beschränkten Eingangs- und Ausgangssignale darstellt, die es zu ermöglicht, die MPC-Sollwerte oder Ziele zu ändern, die den Zustand der MPC-Eingangssignale anzeigen, um zum Beispiel anzuzeigen, ob die Eingangssignale schlecht, unbestimmt oder beschränkt sind, oder die jede andere Art von gewünschten Daten erläutert oder jede andere gewünschte Funktion ausführt.

**[0072]** Zusätzlich zum Herunterladen der MPC-Steuerlogik auf den MPC-Block im MPC-Steuermodul kann bei Schritt **135** in **Fig. 2** die MPC-Logik oder ein MPC-Block, der die entwickelte Logik enthält, an eine Workstation geleitet werden, um in einer oder mehreren Simulationsumgebungen verwendet zu werden, um zum Beispiel den Gebrauch eines MPC-Steuerblocks Nutzern beizubringen, den MPC-Block zu testen, etc. Eine derartige Simulationsumgebung kann unter Verwendung des Systems bereitgestellt werden, das im einzelnen in der Druckschrift US 7257523 B1 "Integrating Distributed Process Control System Functionality on a Single Computer", angemeldet am 6. Mai 1999, beschrieben ist.

**[0073]** **Fig. 8** zeigt eine Simulationsanordnung **149** mit einem MPC-Block **150**, der erzeugt wurde und mit

einem Prozeßsimulationsblock **152** in einer Simulationsumgebung verbunden ist. Die Simulationskonfiguration **149** kann bei Schritt **128** in **Fig. 2** verwendet werden, um zum Beispiel einen vervollständigten MPC-Block zu testen, um zu bestimmen, ob er den Prozeß, für den er entwickelt wurde, angemessen steuert. Oder sie kann bei Schritt **135** in **Fig. 2** verwendet werden, um zum Beispiel eine Übungs- oder andere Testumgebungen mit MPC-Blöcken zur Verfügung zu stellen. Der MPC-Block **150** in **Fig. 8**, der mit drei Eingängen  $IN_1$ – $IN_3$  und drei Ausgängen  $OUT_1$ – $OUT_3$  dargestellt ist, ist mit dem Simulationsblock **152** mit drei Eingängen  $X_1$ – $X_3$  und drei Ausgängen  $Y_1$ – $Y_3$  verbunden, wobei die Ausgänge  $Y_1$ – $Y_3$  an die jeweiligen Eingänge  $IN_1$ – $IN_3$  des MPC-Blocks **150** angeschlossen sind. Der Simulationsblock **152** kann den Prozeß simulieren, für den der MPC-Funktionsblock **150** mit dem Prozeßmodell erzeugt wurde, das bei Schritt **125** in **Fig. 2** erzeugt wurde, wie durch den Block **154** in **Fig. 8** dargestellt. In diesem Fall kann das Prozeßmodell, das bei Schritt **125** in **Fig. 2** erzeugt wurde, im Simulationsblock **152** gespeichert werden und mit ihm kann das Ansprechverhalten des Prozesses auf der Grundlage dieses Prozeßmodells und der Eingangssignale, die vom MPC-Block **150** erhalten werden, simuliert werden. Alternativ dazu kann der Simulationsblock **152** aus einem Prozeßmodell erzeugt werden, das bezüglich des Prozeßmodells, das bei Schritt **125** erzeugt wurde, geändert wurde, wie durch den Block **156** in **Fig. 8** dargestellt. In diesem Fall kann das bei Schritt **125** erzeugte Prozeßmodell derart geändert werden, daß es zum Beispiel Änderungen im Prozeß simuliert, die zum Beispiel durch physikalische Änderungen im Prozeß, Alterung der Prozeßausrüstung etc. verursacht werden. Wenn gewünscht, kann das im Block **125** in **Fig. 2** erzeugte Prozeßmodell auf verschiedene Arten geändert werden, um den Betrieb des MPC-Blocks **150** zu testen, wenn dieser Block zur Steuerung eines Prozesses verwendet wird, der sich von dem Prozeß unterscheidet, der zuerst verwendet wurde, um den MPC-Block **150** zu erzeugen. Daher kann, wenn gewünscht, das geänderte Prozeßmodell von Block **156** im Simulationsblock **152** verwendet werden, um den Steuerbereich, den ein MPC-Block bietet, wenn sich der Prozeß ändert oder nicht mit der MPC-Steuerlogik übereinstimmt, zu bestimmen, um es dem Nutzer zu ermöglichen, MPC-Blöcke zu entwerfen, die über längere Zeitdauern arbeiten oder die besser zur Steuerung von Prozessen in der Anwesenheit von Prozeßänderungen in der Lage sind.

**[0074]** Um einen geänderten Prozeß zu erzeugen, führt der Nutzer oder Bediener eine Routine aus, um das bei Schritt **125** erzeugte Prozeßmodell oder die Reihe der dazu gehörenden Prozeßeingangssignal-/Ausgangssignal-Reaktionskurven anzusehen, wie beispielsweise die in der Bildschirmanzeige in **Fig. 9** für einen  $5 \times 5$  MPC-Block dargestellten, um eine oder mehrere dieser Reaktionskurven, die ge-

ändert werden sollen, auszuwählen. Die ausgewählte Reaktionskurve (dargestellt als Überhang %C<sub>3</sub>H<sub>6</sub> gegen Rückkoherölfluß) kann dann in einem weiteren Bildschirm angezeigt und bearbeitet werden, wie etwa dem in **Fig. 10** dargestellten. Wie in **Fig. 10** dargestellt, kann der Nutzer oder Bediener eine Reaktionskurve importieren oder streichen, eine FIR-Reaktion hinzufügen, die Reaktionsparameter, wie zum Beispiel Totzeit und Verstärkung, ändern, neue Start- oder Endpunkte auswählen, den Wert jedes der Punkte in der Kurve ändern, die Steigung der Kurven ändern, die Kurven skalieren etc., um geänderte Reaktionskurven und folglich ein geändertes Prozeßmodell zu erzeugen. Natürlich kann der Nutzer oder Bediener das Prozeßmodell in jeder anderen Weise ändern. Der Nutzer kann ein Prozeßmodell **150** erzeugen oder überarbeiten, kann einen Prozeßsimulationsblock oder -element **152** aus einem solchen Modell erzeugen, kann den Prozeßsimulationsblock **152** mit dem MPC-Block **150** in Kommunikationsverbindung setzen und die angeschlossene Schleife unter Verwendung einer Routine, wie etwa der Routine **153**, dargestellt in **Fig. 1** im linken Prozessor **13**, oder in jeder anderen gewünschten Simulationsumgebung oder in der Prozeßumgebung betreiben.

**[0075]** In **Fig. 11** ist eine weitere Steuerschleife **170** mit einem MPC-Block **172** dargestellt, der an einen tatsächlichen Prozeß **174** angeschlossen ist. In diesem Fall sind die drei Ausgänge OUT<sub>1</sub>–OUT<sub>3</sub> des MPC-Blocks **172** mit den drei Prozeßeingängen X<sub>1</sub>–X<sub>3</sub> des Prozesses **174** verbunden sowie mit den drei Eingängen X<sub>1SIM</sub>–X<sub>3SIM</sub> eines simulierten Prozeßblocks **176**, der auf einem Prozeßmodell **178** basiert, das für den Prozeß **174** erzeugt wurde. In dieser Anordnung steuert der MPC-Block **172** sowohl den Prozeß **174** während der Laufzeit als auch den simulierten Prozeßblock **176**, der zum Beispiel in einer der Workstations **13** in **Fig. 1** oder in jeder anderen Steuereinrichtung oder Einrichtung umgesetzt wird, und der simulierte Ausgangssignale Y<sub>1SIM</sub>–Y<sub>3SIM</sub> erzeugt. In einem derartigen System können die Ausgangssignale Y<sub>1SIM</sub>–Y<sub>3SIM</sub> des simulierten Prozeßblocks **176** und die Ausgangssignale Y<sub>1</sub>–Y<sub>3</sub> des tatsächlichen Prozesses **174** miteinander verglichen werden, um zu bestimmen, ob die Prozeßausgangssignale Y<sub>1</sub>–Y<sub>3</sub> merklich von den simulierten Ausgangssignalen Y<sub>1SIM</sub>–Y<sub>3SIM</sub> abweichen und ob es somit eine Abweichung zwischen dem tatsächlichen Prozeß **174** und dem Prozeßmodell **178** gibt, aus dem der MPC-Block **172** erzeugt wurde. Wenn dem so ist, kann es notwendig sein, einen anderen MPC-Block **172** zu erzeugen oder die Logikparameter oder das Modell, das vom MPC-Block **172** bei der Steuerung des Prozesses **174** verwendet wird, zu erneuern.

**[0076]** Auch wenn aus einem beliebigen Grund eines der Ausgangssignale Y<sub>1</sub>–Y<sub>3</sub> des Prozesses **174** aufgrund von zum Beispiel einer Fehlfunktion des Sensors, der dieses Ausgangssignal mißt, fehlerhaft

ist, kann das entsprechende simulierte Ausgangssignal des simulierten Prozeßblocks **174** dem entsprechenden Eingang des MPC-Blocks **172** zugeleitet werden, wie durch die gepunktete Linie in **Fig. 11** dargestellt, um es dem MPC-Block **172** zu ermöglichen, den tatsächlichen Prozeß **174** besser zu steuern, bis der fehlerhafte Sensor oder die Einrichtung ausgetauscht oder repariert ist. Auf diese Weise kann ein virtuelles Prozeßausgangssignal für jedes der tatsächlichen Prozeßausgangssignale durch den simulierten Prozeßblock **176** entwickelt werden und eines oder mehrere dieser virtuellen Ausgangssignale kann als Eingangssignal für den MPC-Block **172** verwendet werden, wenn das entsprechende tatsächliche Prozeßausgangssignal fehlerhaft oder in anderer Weise nicht verwendbar ist. Zum Beispiel kann, wenn ein Sensor zur Messung eines der Prozeßausgangssignale Y<sub>1</sub>–Y<sub>3</sub> nachts eine Fehlfunktion hat, der Nutzer oder Bediener einfach das entsprechende virtuelle Ausgangssignal auf den Eingang des MPC-Block **172** legen, so daß der MPC-Block **172** den Prozeß **174** angemessen steuert, bis am nächsten Tag ein Monteur den fehlerhaften Sensor auswechselt oder repariert. Es ist klar, daß der simulierte Prozeßblock **176** zu allen Zeiten, zu denen der tatsächliche Prozeß **174** abläuft, betrieben werden kann und mit den gleichen Eingangssignalen versorgt wird, so daß der simulierte Prozeßblock **176** realistische virtuelle Ausgangssignale erzeugen kann. Natürlich können andere Simulationsszenarien umgesetzt werden und ein Prozeßmodell verwenden, das in Verbindung mit der Erzeugung des MPC-Blocks **150** oder **172** erzeugt wurde, oder sie können ein Prozeßmodell verwenden, das eine Abänderung der Prozeßmodelle ist, die in Verbindung mit der Erzeugung des MPC-Blocks **150** oder **172** erzeugt wurden.

**[0077]** Die Erzeugung eines MPC-Steuerblocks ohne die dafür notwendigen Steuerlogikparameter und Prozeßmodelle und das Verbinden dieses Blockes mit dem Prozeßsteuersystem in einer Weise, die gleich der ist, in der andere Steuerblöcke oder -elemente mit dem System verbunden werden, der Betrieb des MPC-Steuerblocks zur Sammlung von Prozeßdaten, das Erzeugen eines Prozeßmodells aus diesen Daten, die Erzeugung von Logikparametern für den MPC-Block aus diesem Prozeßmodell und das Laden der Logikparameter und, wenn notwendig, des Prozeßmodells in einen MPC-Steuerblock ermöglicht es einem Nutzer, einen MPC-Steuerblock oder ein -Modul in einer Prozeßsteueroutine zu erzeugen, ohne in den nicht angeschlossenen bzw. Offline-Zustand gehen zu müssen, ohne daß es notwendig ist, großes Wissen über die Erzeugung einer MPC-Steueroutine zu haben, ohne daß viel Ingenieursarbeit zur Erzeugung von Wellenformen zur Bildung eines Prozeßmodells geleistet werden muß und ohne daß eine Steueroutine zur Einführung modellvorhersagender oder anderer fortschrittlicher Steuerung neu programmiert werden muß. Im Ergebnis

spart dieses Verfahren Zeit und Kosten und ermöglicht die Verwendung der erzeugten Prozeßmodelle für andere Zwecke, wie zum Beispiel zur Simulation und zur Erzeugung virtueller Prozeßausgangssignale in der Prozeßsteuerumgebung.

**[0078]** Die Routinen und Verfahren zur Erzeugung von MPC- oder fortschrittlichen Steuerlogiken, die hier beschrieben sind, ermöglichen es Benutzern, fortschrittliche Steuerblöcke wie z. B. MPC-Steuerblöcke, Blöcke zur Modellierung oder Steuerblöcke in neuronalen Netzen etc. zu erzeugen, ohne daß sie viel Expertenwissen haben, wie diese Blöcke erzeugt werden. Sie ermöglichen es dem Bediener, einen fortschrittlichen Steuerblock zu erzeugen und zu nutzen, ohne daß die Prozesse zur Einführung der fortschrittlichen Steuerung mit großem Aufwand umprogrammiert werden müssten. Auch können, da die fortschrittlichen Steuerblöcke mit den gleichen Programmparadigmen wie die anderen Steuerelemente im System erzeugt wurden, übereinstimmende Ansichten des Prozesses oder grafische Darstellungen des Prozesses mit dem fortschrittlichen Steuerblock dargestellt werden. Desweiteren kann, da das Prozeßmodell zum Beispiel zur Erzeugung eines MPC-Funktionsblocks erstellt werden muß, dieses Prozeßmodell dazu verwendet werden, Simulationsfunktionsblöcke zu erzeugen, die zur Simulation des Prozesses für andere Zwecke verwendet werden können, wie zum Beispiel Testen, Schulung, Auffinden von Abweichungen zwischen Prozeß und Prozeßmodell, oder zur Erzeugung virtueller Ausgangssignale des Prozesses zur Verwendung bei der Steuerung eines Prozesses.

**[0079]** Während die fortschrittlichen Steuerblöcke, die Prozeßsimulationsblöcke und die damit verknüpften Routinen zur Erzeugung und zum Testen hier derart beschrieben wurden, daß sie in Verbindung mit Fieldbus- und Standard 4–20 Milliampere-Einrichtungen genutzt werden, können sie natürlich auch unter Verwendung jedes anderen Prozeßsteuerkommunikationsprotokolls oder jeder anderen Programmierungsumgebung implementiert werden, und sie können mit jeder anderen Art von Einrichtungen, Funktionsblöcken oder Steuereinrichtungen verwendet werden. Desweiteren ist festzuhalten, daß die Verwendung des Ausdrucks "Funktionsblock" hier nicht darauf beschränkt ist, was das Fieldbus-Protokoll oder das DeltaV-Steuereinrichtungs-Protokoll unter Funktionsblock versteht, sondern daß auch jede andere Art von Block, Programm, Hardware, Firmware, etc. eingeschlossen ist, die zu jeder Art von Steuersystem und/oder Kommunikationsprotokoll gehören, mit denen eine Prozeßsteuerfunktion implementiert werden kann. Auch ist es nicht notwendig, daß die Funktionsblöcke die übliche Form von Objekten in einer objektorientierten Programmierungsumgebung haben.

**[0080]** Obwohl die fortschrittlichen Steuerblöcke, die Prozeßsimulationsblöcke und die hier beschriebenen, damit verbundenen Erzeugungs- und Testroutinen vorzugsweise in Software implementiert sind, können sie auch in Hardware, Firmware, etc. implementiert werden und können mit jedem anderen mit einem Prozeßsteuersystem verknüpften Prozessor umgesetzt werden. Folglich kann die hier beschriebene Routine **40** in einer standardisierten Mehrzweck-CPU implementiert werden oder, wenn es gewünscht ist, in speziell hierfür entworfener Hardware oder Firmware, wie zum Beispiel in ASICs. Wenn sie in Software implementiert ist, kann die Software auf jedem computerlesbaren Speicher wie einer Magnetplatte, einer Laserplatte, einer optischen Platte oder anderen Speichermedien, in einem RAM oder ROM eines Computers oder Prozessors etc. gespeichert werden. Dementsprechend kann die Software an den Nutzer oder ein Prozeßsteuersystem über jede bekannte oder gewünschte Lieferart geliefert werden, einschließlich zum Beispiel auf einer computerlesbaren Platte oder anderen transportablen Computerspeichereinrichtungen, oder über einen Kommunikationskanal, wie eine Telefonverbindung, das Internet etc. übermittelt werden (was als gleichartig oder austauschbar zur Bereitstellung der Software über ein transportables Speichermedium angesehen wird).

### Patentansprüche

1. Prozess-Steuerung zur Steuerung eines Prozesses (**58**), umfassend:  
 eine Vielzahl von Steuereingängen (IN1–IN3), die mit einer Vielzahl von Prozessausgängen verbunden sind, wobei jeder Steuereingang (IN1–IN3) ein Prozessausgangssignal von einem von der Vielzahl von Prozessausgängen empfängt;  
 eine Vielzahl von Steuerausgängen (OUT1–OUT3), die mit einer Vielzahl von Prozesseingängen verbunden sind, wobei jeder Steuerausgang (OUT1–OUT3) ein Steuersignal an einen von der Vielzahl von Prozesseingängen (X1–X5) abgibt;  
 einen Wellenformgenerator zum Erzeugen von Anregungswellenformen an jedem der Vielzahl von Steuerausgängen (IN1–IN3);  
 eine Datensammeleinheit zum Sammeln von Daten, die Signalwerte an jedem der Vielzahl von Steuereingängen (IN1–IN3) darstellen, wenn der Wellenformgenerator die Anregungswellenformen an jedem der Vielzahl von Steuerausgängen (OUT1–OUT3) erzeugt;  
 einen Steuerlogikparametergenerator, der die gesammelten Daten nutzt, um ein Prozessmodell für den Prozess (**58**) zu erzeugen, basierend darauf, wie jede der Steuereingänge auf die bei jeder der Vielzahl von Steuerausgängen erzeugten Anregungswellenformen antwortet, und Steuerlogikparameter gemäß des erzeugten Prozessmodells erzeugt; und  
 eine Mehrfacheingabe-/Mehrfachausgabe-Steuerlogik, derart ausgebildet, vor der Erzeugung des Pro-

zessmodells nicht konfiguriert zu sein und keine Steuerlogikparameter zu beinhalten und nach der Erzeugung der Steuerlogikparameter als modellvorhersagende Steuerung gemäß den Steuerlogikparameter konfiguriert zu sein,

wobei die Prozess-Steuerung derart ausgebildet ist, ein Steuersignal an jedem der Vielzahl von Steuer-  
ausgängen zu erzeugen, basierend auf (i) der konfigurierten Mehrfacheingabe-/Mehrfachausgabe-Steuerlogik, und (ii) der Vielzahl von Prozessausgangssignalen, die an der Vielzahl von Steuereingängen empfangen wurden.

2. Prozess-Steuerung nach Anspruch 1, **dadurch gekennzeichnet**, dass die Steuerlogik eine neuronale Netzwerksteuerlogik ist.

3. Prozess-Steuerung nach Anspruch 1, **dadurch gekennzeichnet**, dass die Steuerlogik ein auf einer Steuerlogik basierendes Prozessmodell ist.

4. Prozess-Steuerung nach Anspruch 1, **dadurch gekennzeichnet**, dass der Wellenformgenerator eine Reihe von Impulsen mit pseudo-zufallsverteilter Länge als Anregungswellenformen erzeugt.

5. Prozess-Steuerung nach Anspruch 1, **dadurch gekennzeichnet**, dass der Wellenformgenerator Anregungswellenformen erzeugt, die zur Entwicklung eines Prozessmodells für den Prozess zur Verwendung bei der Erzeugung einer modellvorhersagenden Steuereinrichtung ausgelegt sind.

6. Prozess-Steuerung nach Anspruch 1, **dadurch gekennzeichnet**, dass Prozess-Steuerung des Weiteren eine grafische Ansicht enthält, um Informationen, die den Betriebsablauf der Prozess-Steuerung betreffen, über eine Benutzerschnittstelle anzeigt.

7. Prozess-Steuerung nach Anspruch 1, ferner enthaltend einen Prozesssimulationsblock, der kommunikativ mit den Steuereingängen (IN1–IN3) und Steuerausgängen (OUT1–OUT3) verbunden ist.

8. Prozess-Steuerung nach Anspruch 7, **dadurch gekennzeichnet**, dass der Prozesssimulationsblock ein Prozessmodell enthält, das den Betrieb des Prozesses simuliert.

9. Verfahren zur Entwicklung eines fortschrittlichen Steuerelements (38) zur Verwendung in einer Prozesssteueroutine, die einen Prozess (58) steuert, wobei das Verfahren die folgenden Schritte enthält:  
a) Erzeugen eines fortschrittlichen Steuerelements (38) mit einer ersten Vielzahl von Steuereingängen (IN1–IN3) zum Empfangen von einer Vielzahl von Prozessausgangssignalen aus Prozessausgängen (Y1–Y6), einer zweiten Vielzahl von Steuerausgängen (OUT1–OUT3) zum Abgeben von Steuersignalen an eine

Vielzahl von Prozesseingängen (X1–X5) des Prozesses (58) und

nicht-konfigurierte keine Steuerlogikparameter beinhaltende Mehrfacheingabe-/Mehrfachausgabe-Steuerlogik (102)

durch eine Workstation oder Steuereinrichtung (11);

b) Herstellen einer Kommunikationsverbindung für das fortschrittliche Steuerelement (38), wobei die Steuerausgänge (OUT1–OUT3) mit den Prozesseingängen (X1–X5) und die Steuereingänge (IN1–IN3) mit den Prozessausgängen (Y1–Y6) in Kommunikationsverbindung gesetzt werden;

c) Ausführen des fortschrittlichen Steuerelements (38) auf der Steuereinrichtung (11) oder der Workstation (13), um Anregungswellenformen an jeden der Prozesseingänge (X1–X5) abzugeben;

d) Sammeln von Daten durch das fortschrittliche Steuerelement (38), wobei die Daten das Ansprechen jedes der Prozessausgänge (Y1–Y6) auf die Anregungswellenformen wiedergeben;

e) Erzeugen eines Prozessmodells aus den gesammelten Daten;

f) Entwickeln von fortschrittlichen Steuerlogikparametern aus dem Prozessmodell, wobei Koeffizienten zur modellvorhersagenden Steuerung entwickelt werden; und

g) Konfigurieren der Mehrfacheingabe-/Mehrfachausgabe-Steuerlogik des fortschrittlichen Steuerelements (38) gemäß den fortschrittlichen Steuerlogikparametern, um dem fortschrittlichen Steuerelement (38) das Steuern des Prozesses (58) zu ermöglichen wobei das fortschrittliche Steuerelement ein Steuersignal an jeden der Vielzahl von Steuerausgängen abgibt, basierend auf (i) der konfigurierten Mehrfacheingabe-/Mehrfachausgabe-Steuerlogik, und (ii) der Vielzahl von Prozessausgangssignalen, die an der Vielzahl von Steuereingängen empfangen wurden.

10. Verfahren nach Anspruch 9, **dadurch gekennzeichnet**, dass das fortschrittliche Steuerelement ein Funktionsblock oder Feldbus-Funktionsblock ist.

11. Verfahren nach Anspruch 9, **dadurch gekennzeichnet**, dass der Schritt b) die Schritte umfasst:

b1) Herstellen einer Kommunikationsverbindung eines der Steuereingänge (OUT1–OUT3) mit einem ersten Funktionsblock und

b2) Herstellen einer Kommunikationsverbindung eines der Steuerausgänge (OUT1–OUT3) mit einem zweiten Funktionsblock.

12. Verfahren nach Anspruch 11, **dadurch gekennzeichnet**, dass der Schritt b2) den Schritt des Herstellens einer Kommunikationsverbindung des einen der Steuerausgänge (OUT1–OUT3) mit einem Steuerfunktionsblock beinhaltet, um eine Steuer-schleife in der Prozesssteueroutine zu steuern.

13. Verfahren nach Anspruch 11, **dadurch gekennzeichnet**, dass der Schritt b2) den Schritt des Herstellens einer Kommunikationsverbindung des einen der Steuerausgänge (OUT1–OUT3) mit einem Ausgangsfunktionsblock beinhaltet, der zu einer Anlageneinrichtung (15–22) gehört, um die Anlageneinrichtung (15–22) zu steuern.

14. Verfahren nach Anspruch 9, **dadurch gekennzeichnet**, dass der Schritt c) den Schritt des Erzeugens einer Reihe von Impulsen mit pseudo-zufallsverteilter Länge als die Anregungswellenformen enthält.

15. Verfahren nach Anspruch 9, **dadurch gekennzeichnet**, dass der Schritt c) den Schritt des Erzeugens einer Reihe von Anregungswellenformen beinhaltet, die dazu ausgelegt sind, ein Prozessmodell für den Prozess (58) zu entwickeln, das zur Erzeugung einer modellvorhersagenden Steuereinrichtung verwendet wird.

16. Verfahren nach Anspruch 9, **dadurch gekennzeichnet**, dass der Schritt e) den Schritt des Erzeugens eines Satzes von Ansprechkurven aus den gesammelten Daten als das Prozessmodell beinhaltet.

17. Verfahren nach Anspruch 16, **dadurch gekennzeichnet**, dass der Schritt e) den Schritt des Änderns einer oder mehrerer Kurven des Satzes von Ansprechkurven beinhaltet, die aus den gesammelten Daten erzeugt wurden, um einen geänderten Satz von Ansprechkurven zu erzeugen und den geänderten Satz von Ansprechkurven als Prozessmodell zu verwenden.

18. Verfahren nach Anspruch 9, gekennzeichnet durch den Schritt h) Verwenden des fortschrittlichen Steuerelements in einer Simulationsumgebung.

19. Verfahren nach Anspruch 18, **dadurch gekennzeichnet**, dass der Schritt h) den Schritt des Verbindens des fortschrittlichen Steuerelements mit einem simulierten Prozesselement beinhaltet.

20. Verfahren nach Anspruch 19, **dadurch gekennzeichnet**, dass der Schritt h) den Schritt des Entwickelns des simulierten Prozesselements aus dem Prozessmodell beinhaltet.

21. Verfahren nach Anspruch 19, **dadurch gekennzeichnet**, dass der Schritt h) den Schritt der Änderung des Prozessmodells sowie der Entwicklung des simulierten Prozesselements aus dem geänderten Prozessmodell beinhaltet.

22. Verfahren nach Anspruch 9, **dadurch gekennzeichnet**, dass der Schritt a) den Schritt der Einbindung einer Steuerlogik in das fortschrittliche Steuerelement vor dem Schritt b) beinhaltet.

23. Verfahren nach Anspruch 9, ferner enthaltend den Schritt:

i) Vorsehen des Prozessmodells für den fortschrittlichen Steuerblock zur Verwendung bei der Steuerung des Prozesses (58).

24. Verfahren nach Anspruch 9, **dadurch gekennzeichnet**, dass der Schritt e) den Schritt des Überprüfens der gesammelten Daten und des Erzeugens des Prozessmodells aus den überprüften Daten beinhaltet.

25. System zur Entwicklung eines fortschrittlichen Steuerelements zur Verwendung in einer Prozesssteueroutine, wobei das System zur Durchführung des Verfahrens nach einem der Ansprüche 9–24 ausgebildet ist.

Es folgen 10 Seiten Zeichnungen

## Anhängende Zeichnungen

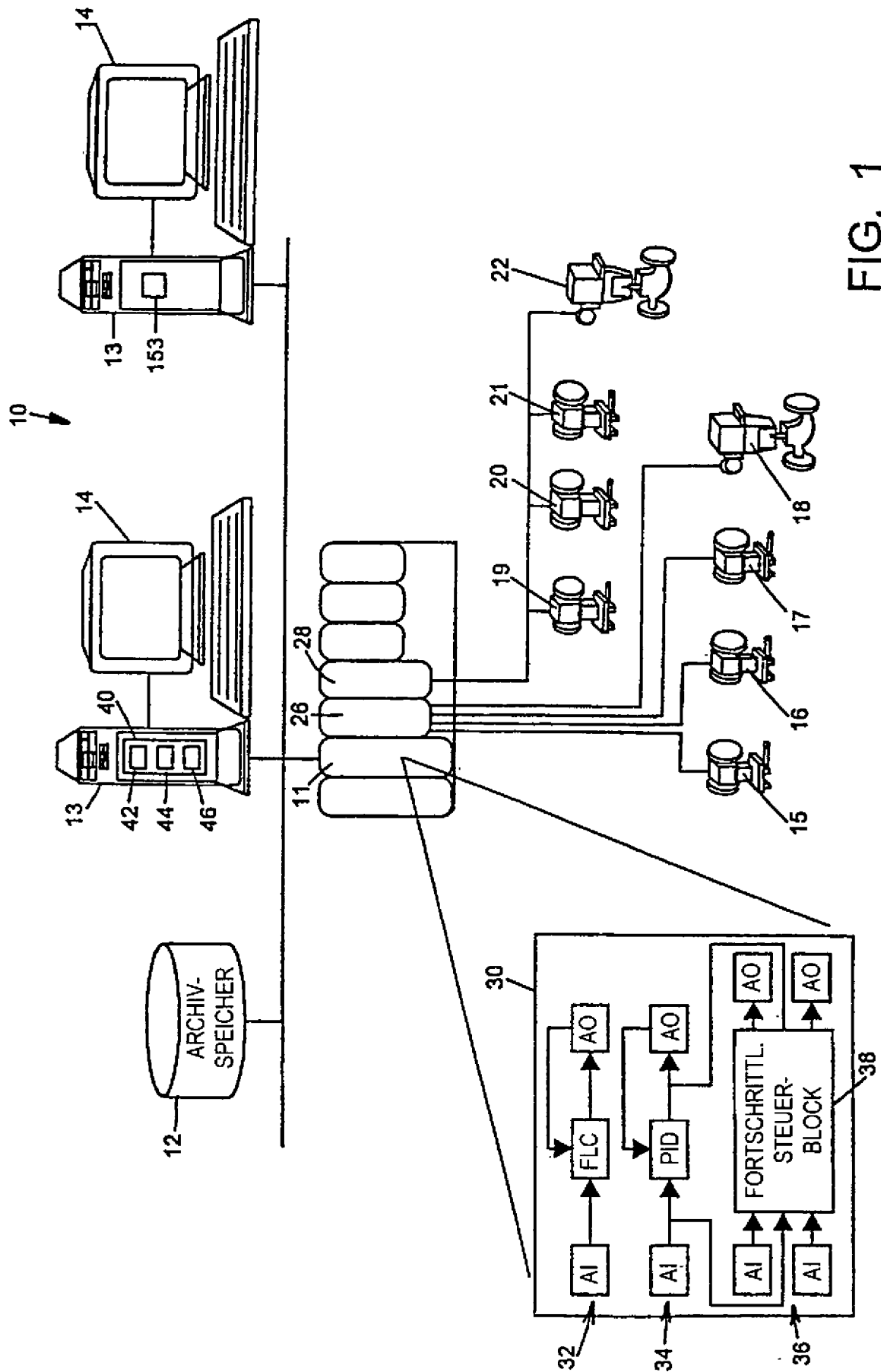


FIG. 1

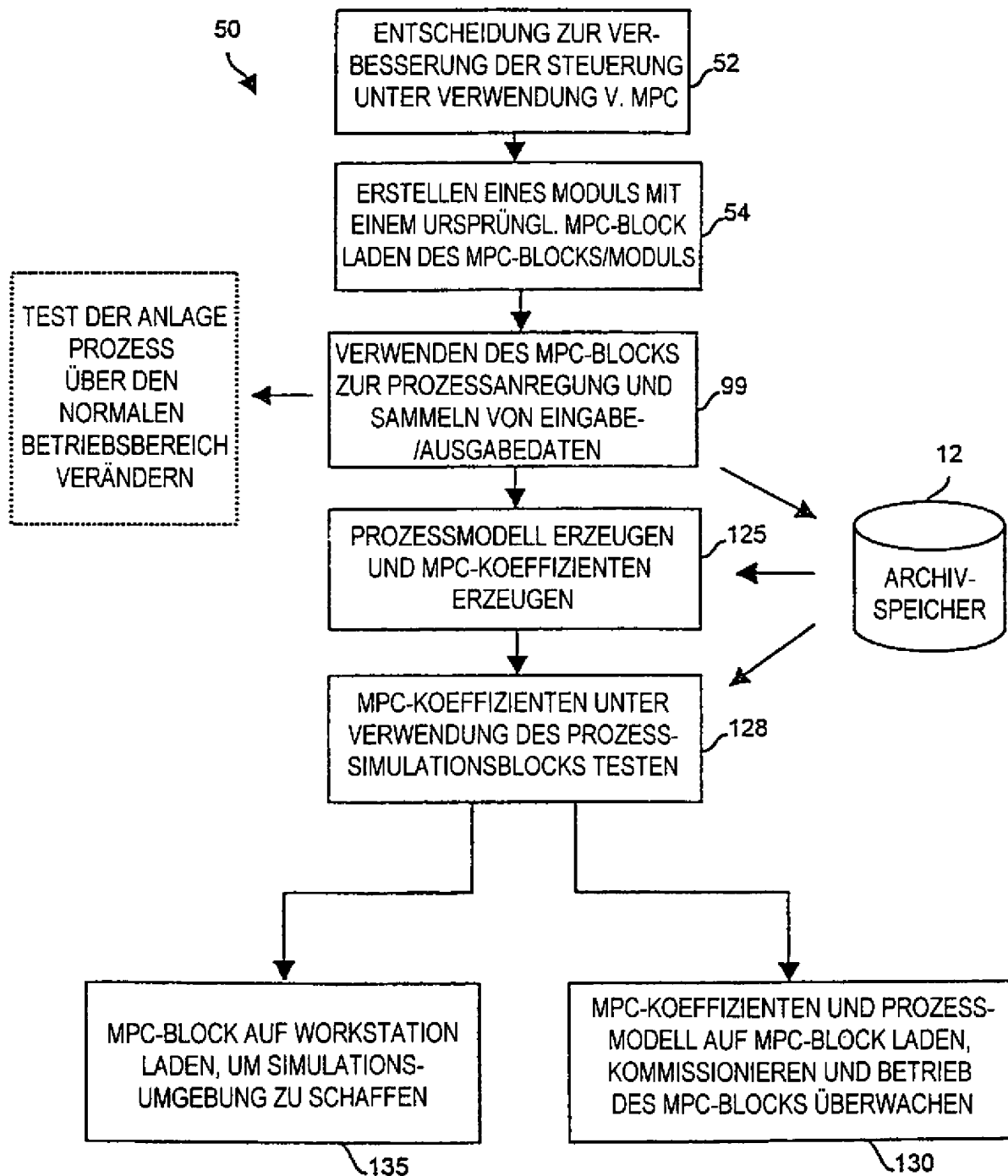
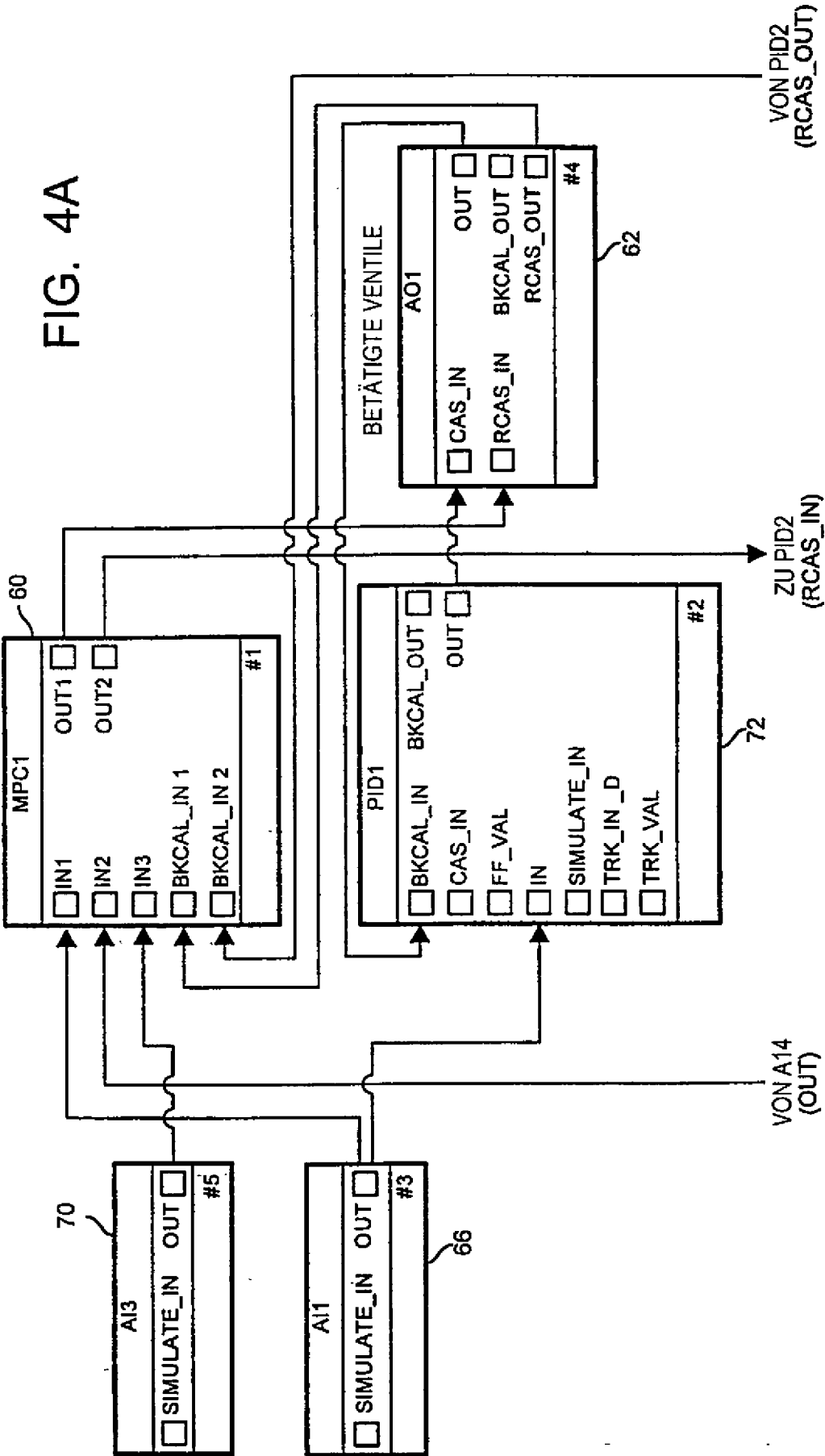


FIG. 2





FIG. 4A



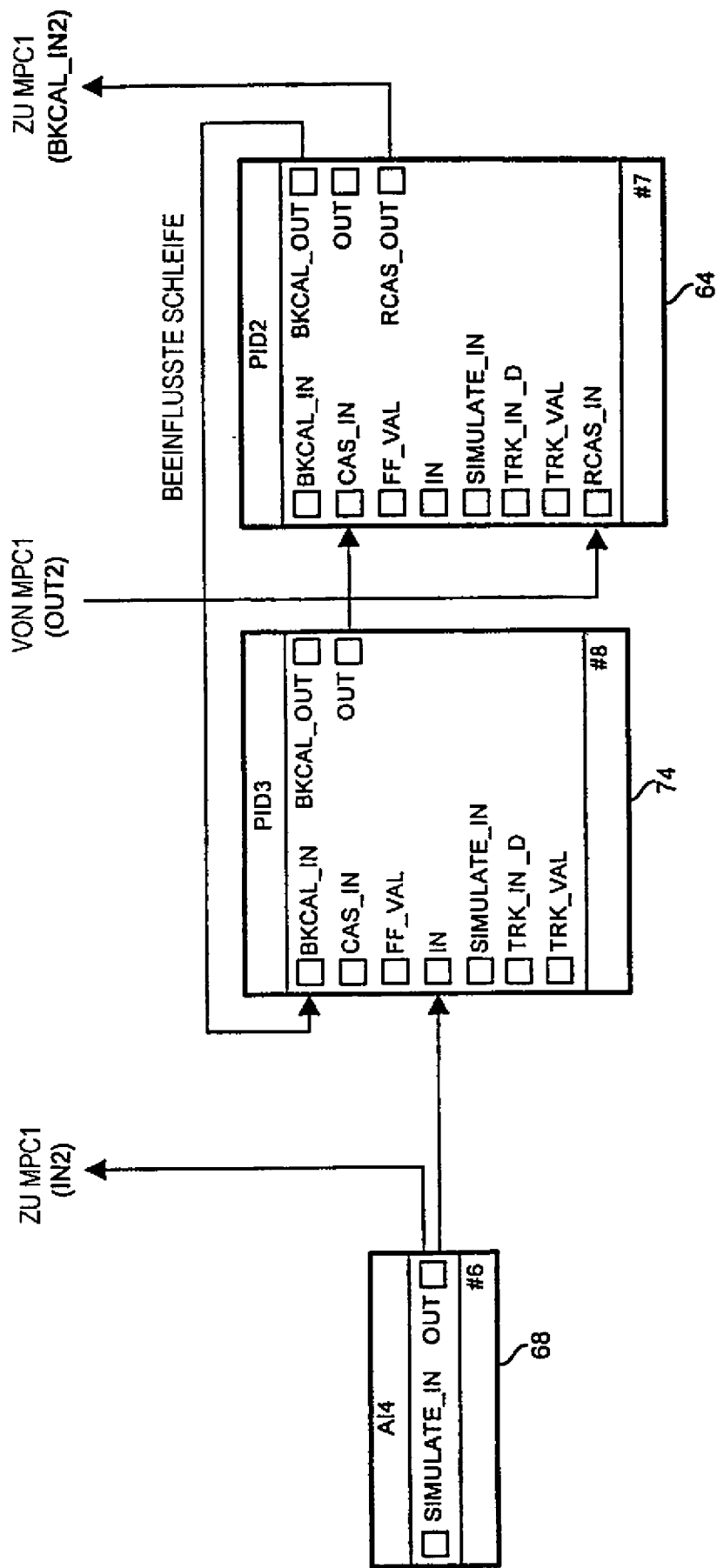


FIG. 4B

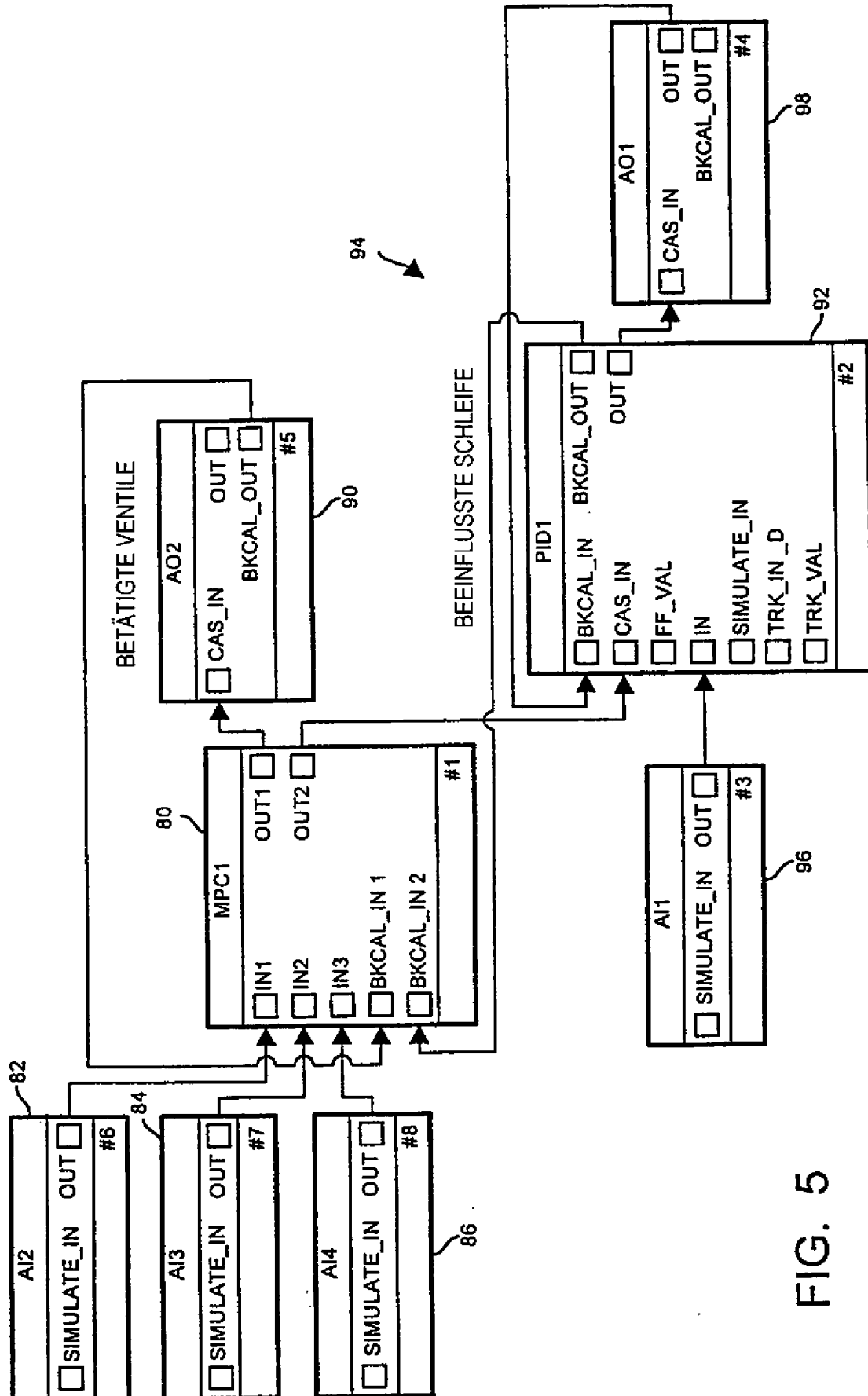


FIG. 5

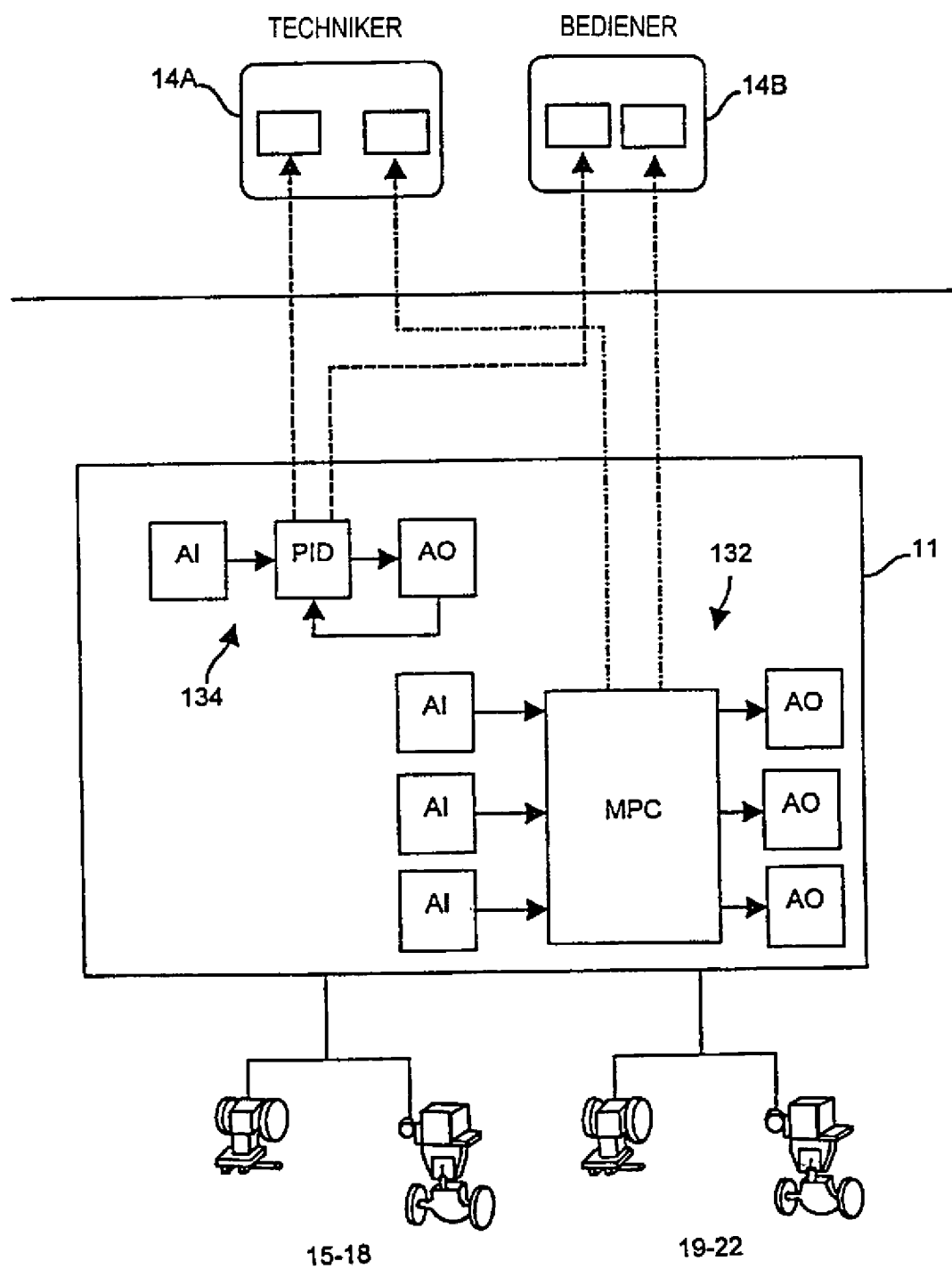
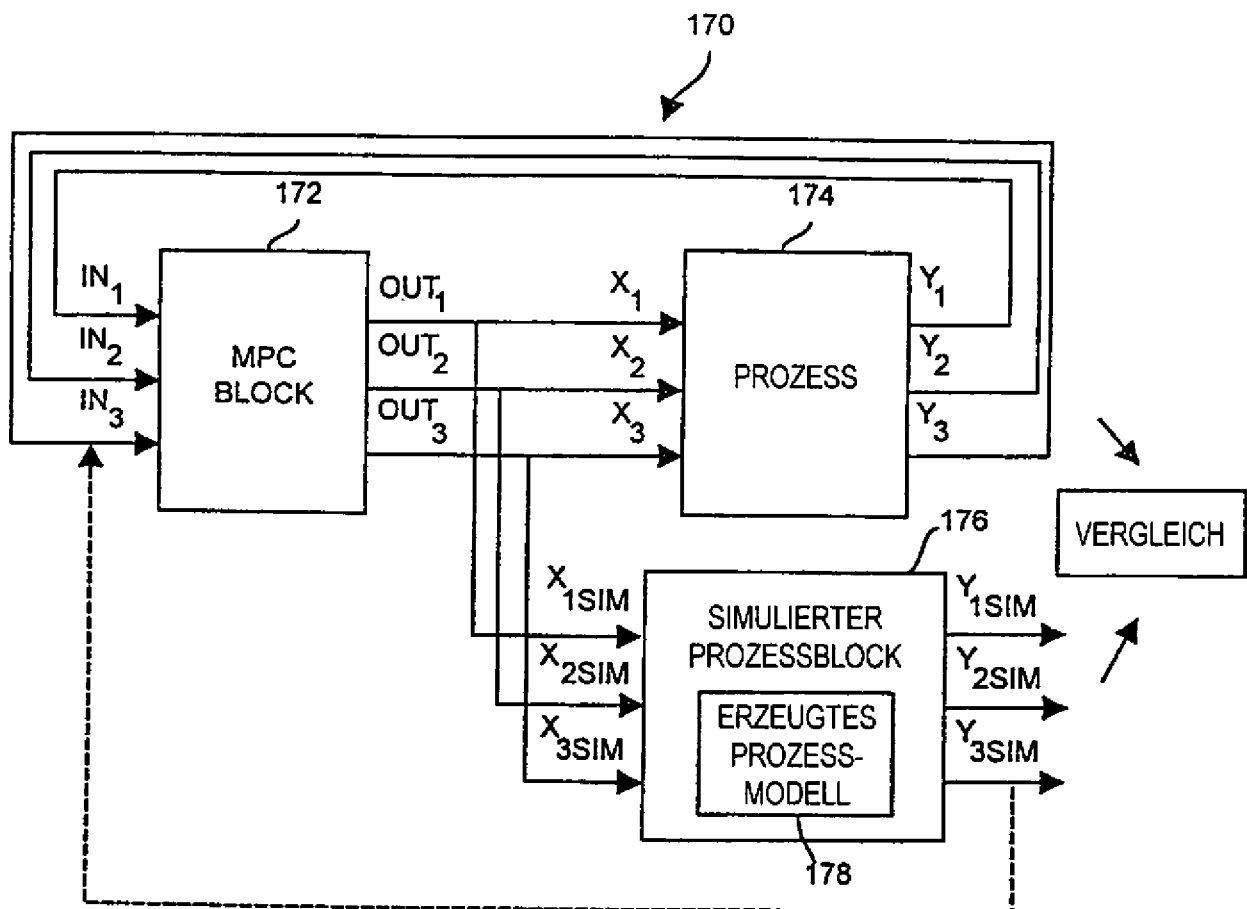
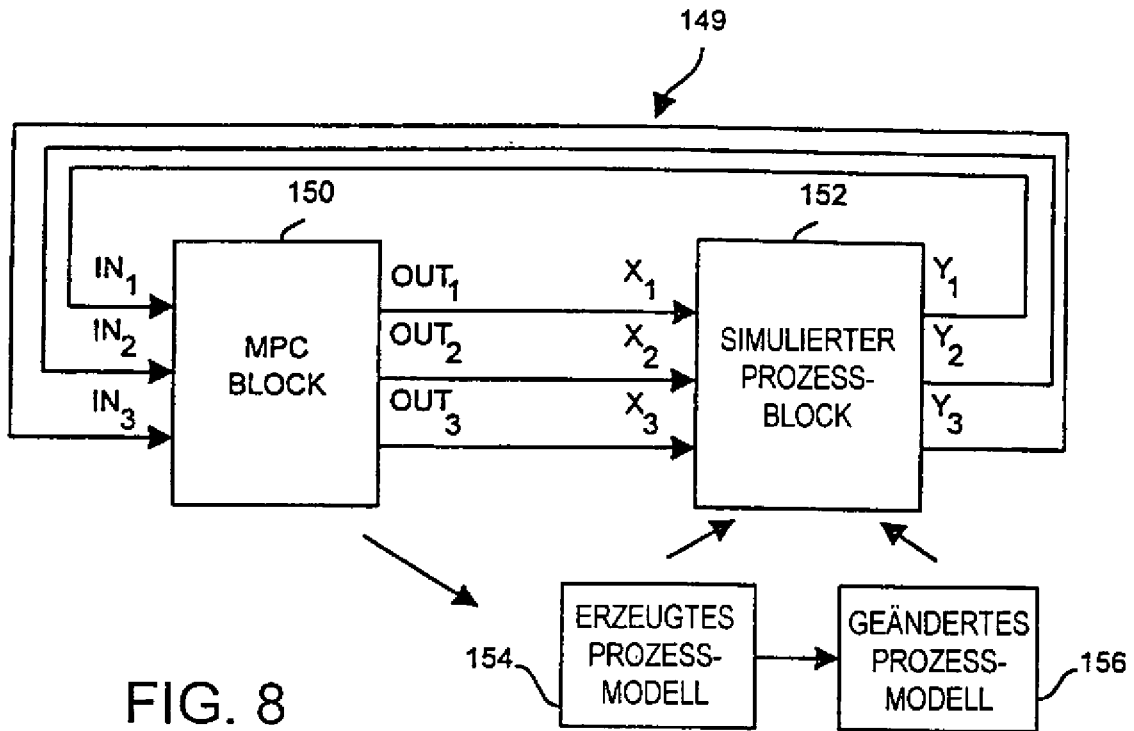


FIG. 7



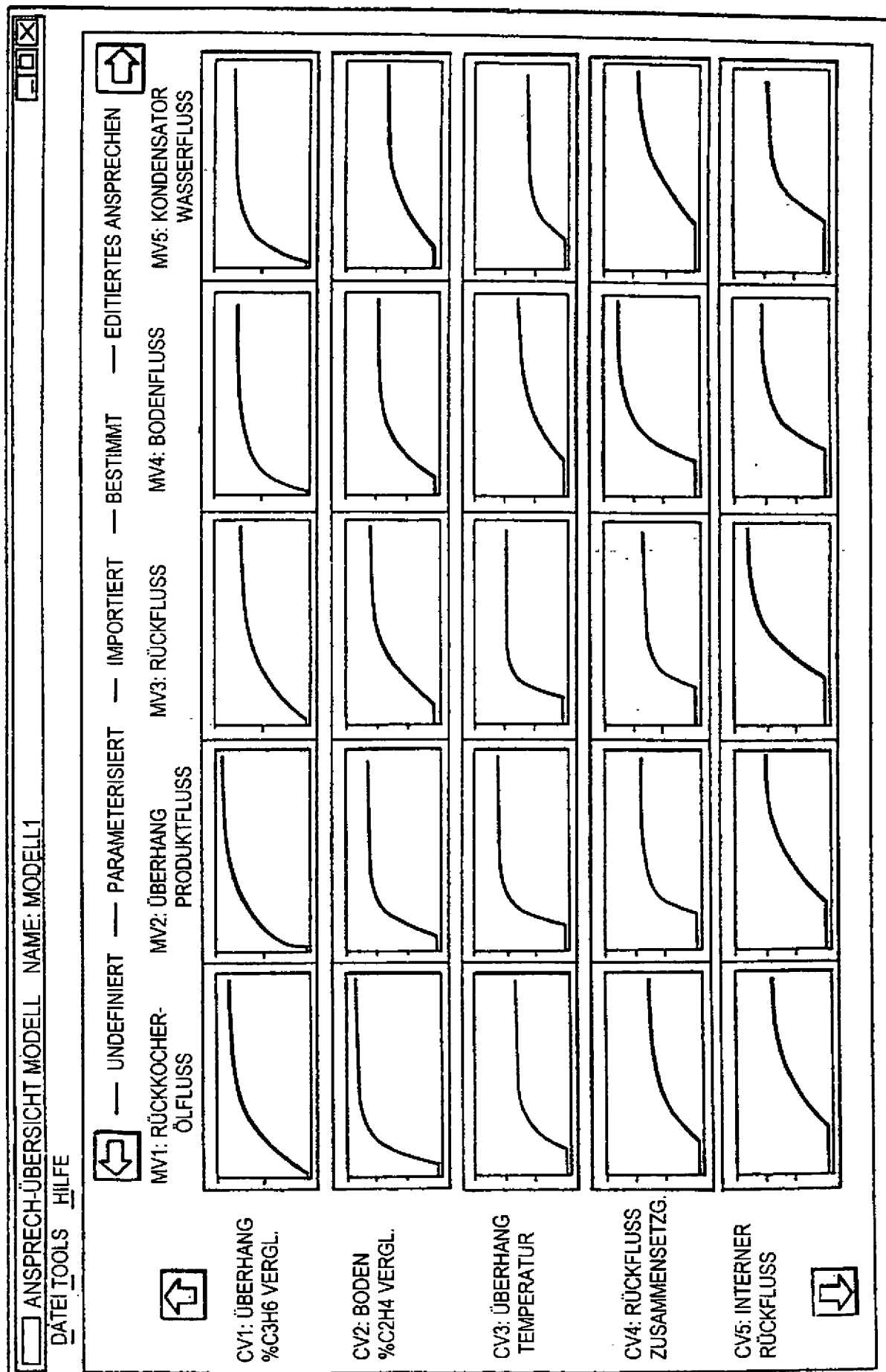


FIG. 9

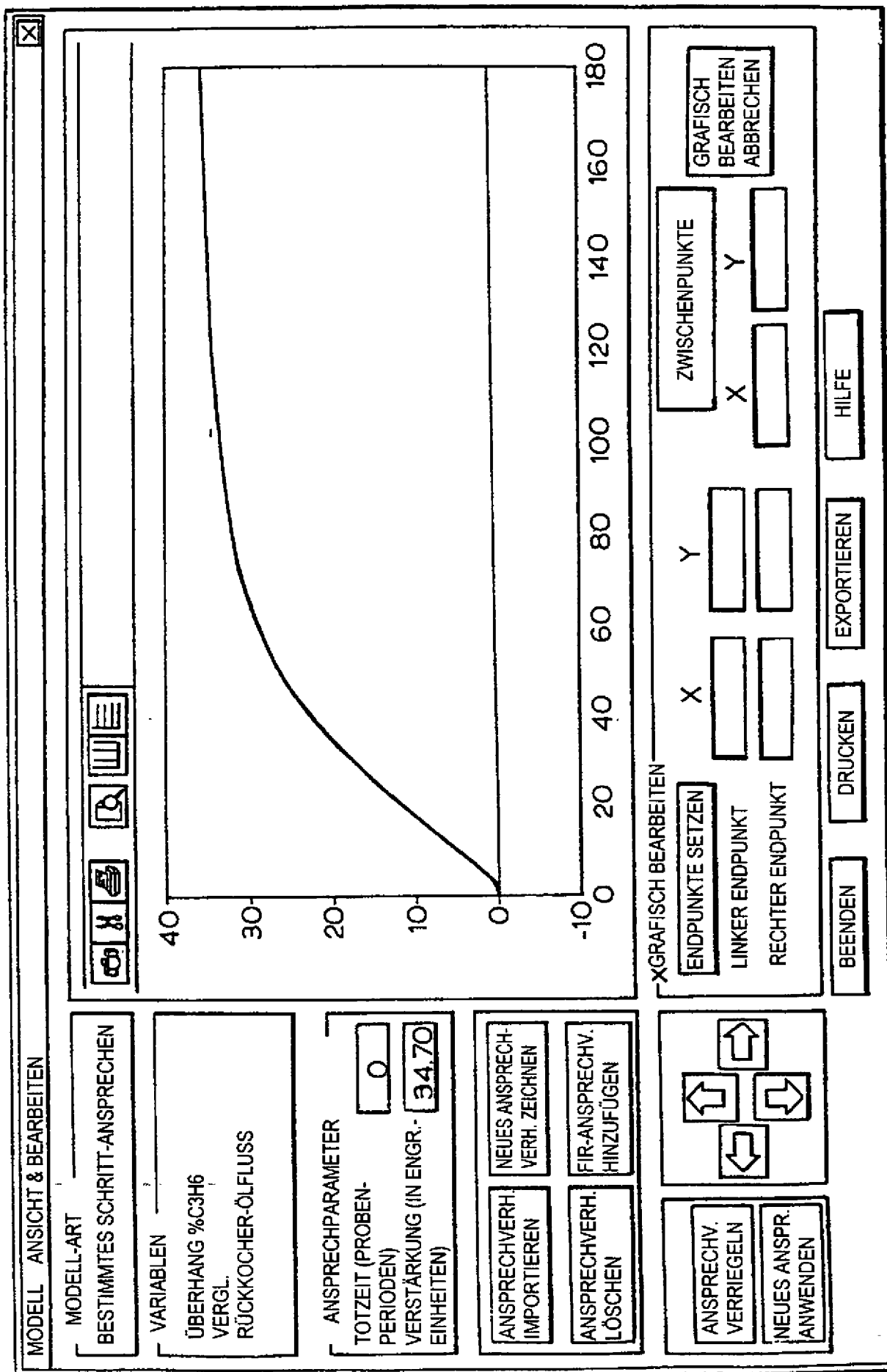


FIG. 10