

(51) Int Cl.: **G05B 13/04** (2006.01)
G05B 19/04 (2006.01)
G05B 15/02 (2006.01)

(72) Erfinder: **Wojsznis, Wilhelm K., Austin, Tex., US; Blevins, Terrence L., Round Rock, Tex., US; Nixon, Mark, Round Rock, Tex., US**

(56) Ermittelter Stand der Technik:

US	5 301 101	A
US	4 616 308	A
US	4 349 869	A

QIN, S. Joe; BADGWELL, Thomas A.: An Overview of Industrial Modell Predictive Control Technology. In: AIChE Symposium Series. New York, 1997, 232-256.

[illegible]

Beschreibung

[0001] Die Erfindung betrifft Prozesssteuerungssysteme im Allgemeinen, insbesondere die Anwendung einer optimierten modellbasierten prädiktiven Steuerung in einem Prozesssteuerungssystem.

[0002] Prozesssteuerungssysteme, z. B. verteilte oder skalierbare Prozesssteuerungssysteme, wie sie in chemischen, erdölverarbeitenden oder anderen Prozessen verwendet werden, enthalten typischerweise eine oder mehrere Prozesssteuerungen, die kommunikativ miteinander, mit mindestens einem Host oder einer Operatorworkstation und zu einem oder mehreren Feldgeräten („field devices“) über analoge, digitale oder kombinierte analog/digitale Busse gekoppelt sind. Die Feldgeräte, die beispielsweise Ventile, Ventilpositionierer bzw. -steller, Schalter und Transmitter oder Messwertgeber (bspw. Temperatur-, Druck- und Durchflussraten-sensoren) sein können, führen Funktionen innerhalb des Prozesses aus, z. B. das Öffnen oder Schließen von Ventilen und das Messen von Prozessparametern. Die Prozesssteuerung empfängt Signale, die auf die durch die Feldgeräte ausgeführten Prozessmessungen und/oder andere, die Feldgeräte betreffende Informationen, angeben. Die Prozesssteuerung nutzt diese Informationen zum Implementieren einer Steuerungsroutine und erzeugt dann Steuerungssignale, die über die Busse an die Feldgeräte gesendet werden, um den Prozess zu steuern. Die Informationen von den Feldgeräten und der Steuerung werden typischerweise einer oder mehreren, auf der Operatorworkstation ausgeführten Anwenderweise einer oder mehreren, auf der Operatorworkstation ausgeführten Anwendung bzw. -en zur Verfügung gestellt, um es einem Operator zu ermöglichen, jede gewünschte, prozessbezogene Funktion auszuführen, z. B. das Ansehen des momentanen Prozesszustands, das Modifizieren des Prozessbetriebs usw.

[0003] Prozesssteuerungen sind typischerweise programmiert, um verschiedene Algorithmen, Subroutinen oder Steuerschleifen (die alle Steuerrouinen sind) für jede einer Anzahl unterschiedlicher Schleifen auszuführen, die für einen Prozess definiert sind oder in einem Prozess beinhaltet sind, wie etwa Flusssteuerungsschleifen, Temperatursteuerungsschleifen, Drucksteuerungsschleifen etc. Im allgemeinen enthält jede dieser Steuerungsschleifen einen oder mehrere Eingabeblocke, z. B. einen Funktionsblock für analoge Eingabe („analog input“, AI), einen Steuerungsblock mit einzelner Ausgabe, z. B. einen Proportional-Integral-Differential-(PID)-Steuerungsfunktionsblock oder einen Fuzzy-Logik-Steuerungsfunktionsblock, und einen Einzelausgabeblock, z. B. einen Funktionsblock zur analogen Ausgabe (AO). Diese Steuerschleifen führen typischerweise Einzeleingang/Einzelausgang-Steuerungen aus, weil der Steuerungsblock eine einzelne Steuerausgabe erzeugt, die zum Steuern einer einzelnen Prozesseingabe verwendet wird, z. B. zum Steuern einer Ventilposition etc. In gewissen Fällen ist allerdings die Verwendung einer Anzahl unabhängig voneinander arbeitender Einzeleingang/Einzelausgang-Steuerungsschleifen nicht sehr effektiv, weil die gesteuerten Prozessgrößen durch mehr als eine einzelne Prozesseingabe beeinflusst werden und faktisch jede Prozesseingabe den Zustand vieler Prozessausgaben beeinflussen kann. Ein Beispiel hierfür kann beispielsweise bei einem Prozess eintreten, der einen durch zwei Zuführleitungen gefüllten und durch eine einzelne Auslassleitung geleerten Tank beinhaltet, wobei jede Leitung durch ein anderes Ventil gesteuert wird und die Temperatur, der Druck und der Durchsatz des Tanks gesteuert werden, um bei oder nahe bei gewünschten Werten zu liegen. Wie oben angedeutet, kann die Steuerung des Durchsatzes, der Temperatur und des Druckes des Tanks unter Verwendung einer separaten Durchsatzsteuerschleife, einer separaten Temperatursteuerschleife und einer separaten Drucksteuerschleife ausgeführt werden. Allerdings kann in dieser Situation der Betrieb der Temperatursteuerschleife über ein Verändern der Einstellung von einem der Eingabeventile, um die Temperatur im Inneren des Tanks zu steuern, zu einer Druckerhöhung im Tankinneren führen. Das kann z. B. die Drucksteuerschleife zu einem Öffnen des Auslassventils veranlassen, um den Druck abzusinken. Diese Aktion kann dann die Durchsatzsteuerschleife veranlassen, eines der Einlassventile zu schließen, wodurch die Temperatur beeinflusst wird und die Temperatursteuerschleife veranlasst wird, eine andere Aktion auszuführen. Wie aus diesem Beispiel ersichtlich, veranlassen die Einzeleingang/Einzelausgang-Steuerungsschleifen die Prozessausgaben (in diesem Fall Durchsatz, Temperatur und Druck), sich in nicht hinzunehmender Weise zu verhalten, wobei die Ausgaben oszillieren, ohne jemals einen stabilen Zustand zu erreichen.

[0004] Die modellgestützte prädiktive Regelung bzw. Steuerung („modell predictive control“, MPC) oder andere Arten einer fortgeschrittenen Steuerung sind angewendet worden, um eine Prozesssteuerung in Situationen auszuführen, bei denen Veränderungen einer bestimmten gesteuerten Prozessgröße mehr als eine Prozessgröße oder mehr als eine Prozessausgabe beeinflussen. Seit den späten 1970er Jahren wurden von vielen erfolgreichen Implementierungen modellgestützter prädiktiver Regelungen bzw. Steuerungen berichtet und MPC wurde eine bevorzugte Form einer fortgeschrittenen Multigrößen-Steuerung in der Prozessindustrie. Weiterhin wurde die MPC-Steuerung in verteilten Steuerungssystemen als geschichtete Software („layered software“) im verteilten Steuersystem implementiert. Die US-Patente mit den Nummern 4,616,308 und 4,349,869 beschreiben allgemein innerhalb eines Prozesssteuerungssystems verwendbare MPC-Steuerungen.

[0005] Allgemein gesprochen, ist MPC eine Steuerungsstrategie für mehrfache Eingabe und mehrfache Ausgabe, bei der die Wirkungen einer Veränderung einer jeden von einer Anzahl von Prozesseingaben auf eine jede aus einer Anzahl von Prozessausgaben gemessen wird und diese gemessenen Antworten dann dazu verwendet werden, um eine Steuerungsmatrix oder ein Modell des Prozesses zu erstellen. Das Prozessmodell oder die Steuerungsmatrix (die allgemein den stabilen Zustand des Prozesses definiert), wird mathematisch invertiert und dann in einer oder als eine Steuerung mit mehrfacher Eingabe/mehrfacher Ausgabe zum Steuern der Prozessausgaben auf der Grundlage von durch die Prozesseingaben gemachten Veränderungen verwendet. In einigen Fällen wird das Prozessmodell durch eine Prozessausgabe-Antwortkurve (typischerweise eine Sprungantwort- bzw. Übergangsfunktionskurve) für jede der Prozesseingaben repräsentiert. Diese Kurven können z. B. auf Basis einer Reihe von pseudozufälligen Schrittländerungen erzeugt werden, die an jede der Prozesseingänge geliefert werden. Diese Antwortkurven können dazu benutzt werden, um den Prozess in bekannter Weise zu modellieren. Die modellgestützte prädiktive Regelung bzw. Steuerung ist bekannt, weshalb deren Einzelheiten im Folgenden nicht beschrieben werden. Die MPC wird jedoch in allgemeiner Weise in Qin, S. Joe und Thomas A. Badgwell, „An Overview of Industrial Model Predictive Control Technology“, AIChE Conference, 1996, beschrieben.

[0006] Die MPC hat sich als eine sehr effektive und nützliche Steuerungstechnik erwiesen und wurde in Verbindung mit der Prozessoptimierung benutzt. Um einen Prozess zu optimieren, der MPC benutzt, minimiert oder maximiert ein Optimierer eine oder mehrere der durch die MPC-Routine festgelegten Prozesseingabegrößen so, dass der Prozess zum an einem optimalen Punkt abläuft. Während diese Technik rechnerisch möglich ist, ist es notwendig, diejenigen Prozessgrößen auszuwählen, die z. B. eine signifikante Wirkung auf die Verbesserung der ökonomischen Arbeitsweise des Prozesses haben (z. B. auf Prozessdurchsatz oder Qualität), um den Prozess von einem ökonomischen Standpunkt aus zu optimieren. Der Betrieb des Prozesses an einem von einem finanziellen oder ökonomischen Gesichtspunkt aus optimalen Punkt erfordert typischerweise das Steuern vieler Prozessgrößen in gegenseitiger Abhängigkeit voneinander, nicht nur einer einzigen Prozessgrößen.

[0007] Eine Optimierung unter Verwendung quadratischer Programmiertechniken oder aktuelleren Techniken wie etwa Innere-Punkt-Methoden wurde als Lösung zum Verwirklichen einer dynamischen Optimierung mit MPC vorgeschlagen. Mit diesen Methoden wird eine Optimierungslösung bestimmt, und der Optimierer stellt der Steuerung Veränderungen in den Steuerungsausgaben (d. h. den Stellgrößen des Prozesses) bereit, wobei er die Prozessdynamik, momentanen Randbedingungen und Optimierungsziele berücksichtigt. Allerdings führt dieser Ansatz zu einem immensen Rechenaufwand und ist bei dem gegenwärtigen technologischen Stand nicht praktikabel anzuwenden.

[0008] In den meisten Fällen, in denen MPC verwendet wird, ist die Anzahl der innerhalb des Prozesses verfügbaren Stellgrößen (d. h. der Steuerausgaben der MPC-Routine) größer als die Anzahl der Steuergrößen des Prozesses (d. h. der Anzahl der Prozessgrößen, die zu steuern sind, damit sie sich an einem bestimmten Punkt befinden). Als Resultat hiervon gibt es für gewöhnlich mehrere Freiheitsgrade, die für die Optimierung und Behandlung der Randbedingungen verfügbar sind. Theoretisch sollten zum Ausführen einer solchen Optimierung Werte berechnet werden, die – ausgedrückt durch Prozessgrößen, Randbedingungen, Grenzen und ökonomischen Faktoren – einen optimalen Arbeitspunkt definieren. In vielen Fällen sind diese Prozessgrößen durch Randbedingungen eingeschränkte Größen, weil sie auf physikalische Eigenschaften des zugehörigen Prozesses bezogene Grenzen aufweisen, innerhalb derer diese Größen gehalten werden müssen. So ist beispielsweise eine einen Tankfüllstand repräsentierende Prozessgröße auf die maximalen und minimalen Füllstände des tatsächlich physikalisch verfügbaren Tankes begrenzt. Eine Optimierungsfunktion kann die Kosten und/oder die Gewinne berechnen, die mit jeder der eingeschränkten oder Hilfsgrößen verbunden sind, um einen Betrieb zu gewährleisten, bei dem der Gewinn maximiert wird, die Kosten minimiert werden, etc. Messungen dieser Hilfsgrößen können dann als Eingaben der MPC-Routine bereitgestellt und durch die MPC-Routine als Steuergrößen behandelt werden, die einen Sollwert hat, der gleich dem Betriebspunkt für die durch die Optimierungsroutine definierte Hilfsgröße hat.

[0009] MPC liefert die besten Ergebnisse, die häufig von der Anwendung angefordert wird, nur für eine quadratische Steuerung, bei der die Anzahl der Steuereingaben in den Prozess (d. h. der durch die Steueroutine gebildeten Stellgrößen) gleich der Anzahl der gesteuerten Prozessgrößen ist (das heißt der Eingaben an die Steuerung). In den meisten Fällen ist allerdings die Anzahl der beschränkten Hilfsgrößen plus der Anzahl der Prozesssteuergrößen größer als die Anzahl der Stellgrößen. Die Implementierung der MPC für derartige nicht-quadratische Konfigurationen führt zu einer unakzeptabel geringen Leistungsfähigkeit.

[0010] Vermutlich ist versucht worden, dieses Problem zu umgehen, und zwar durch dynamische Auswahl eines Satzes von Steuer- und Randbedingungsgrößen gleich zu der Anzahl der Stellgrößen und durch eine

Erzeugung der Steuerung on-line oder während des Prozessbetriebs, um die nächsten Veränderungen der Stellgrößen zu bestimmen. Eine derartige Technik ist allerdings rechnerisch aufwendig, denn sie erfordert eine Matrixinvertierung und kann in einigen Fällen nicht benutzt werden, etwa dann, wenn MPC als ein Funktionsblock in einer Prozesssteuerung implementiert ist. Von gleicher Bedeutung ist, dass sich aus einigen Eingaben/Ausgaben-Kombinationen der erzeugten Steuerung eine schlecht-konditionierte Steuerung ergibt, was zu einem nicht akzeptablen Betrieb führt. Während die Konditionierung der Steuerung überprüft und verbessert werden kann, wenn die Konfiguration der Steuerung offline aufgesetzt wird, bedeutet diese Aufgabe im Online-Betrieb eine beträchtliche Last und ist praktisch unmöglich auf der Steuerungsebene zu implementieren.

[0011] Einschlägiger Stand der Technik sind die Druckschriften US 4,616,308 A und US 5,301,101 A, in denen ein dynamisches Prozesssteuerungsverfahren unter Berücksichtigung einer Mehrzahl von unabhängig gesteuerten Variablen bzw. ein adaptiver und prädiktiver Controller zum Einsatz in einer Prozesssteuerung offenbart sind.

[0012] Eine integrierte Optimierungs- und Steuerungstechnik gemäß der Erfindung kombiniert eine Optimierungsprozedur mit einer erweiterten Steuerungsprozedur, wie einer modellbasierten prädiktiven Steuerung, bei der die Anzahl der Steuer- und Hilfsgrößen größer oder kleiner als die Anzahl der Stellgrößen innerhalb der Prozessanlage sein kann. Die Technik bestimmt zuerst eine Sprungantwortmatrix, die die Korrelation zwischen Änderungen der Stellgrößen und jeder der innerhalb eines Optimierers verwendeten Prozessgrößen (die Steuergrößen und Hilfsgrößen sein können) definiert, um den optimalen Betriebspunkt für den Prozess zu bestimmen. Ein Teilsatz der Prozesssteuergrößen und Hilfsgrößen, der eine Anzahl aufweist, die gleich zu oder kleiner als die Anzahl der Stellgrößen ist, wird zur Verwendung als Eingaben an die MPC-Routine während des Prozessbetriebes ausgewählt. Die auf diesem Teilsatz der Steuer- und Hilfsgrößen und Stellgrößen basierende Verstärkungsmatrix für die MPC-Steuerung wird dann zur Erzeugung der während des Betriebs der MPC-Routine verwendeten MPC-Steuerung genutzt. Danach berechnet die Optimierer-Routine während jedem Scan der Prozesssteuerung das optimale Betriebsziel von jeder Größe aus dem vollständigen Satz von Steuer- und Hilfsgrößen und stellt einen Satz von abgeleiteten Zielbetriebspunkten für jede Größe des gewählten Teilsatzes der Steuer- und Hilfsgrößen der MPC-Routine als Eingaben bereit. Die MPC-Routine bestimmt sodann Änderungen in den Stellgrößen zum Einsatz bei der Prozesssteuerung durch die vorgegebene MPC-Matrix unter Verwendung des Teilsatzes der Steuer- und Hilfsgrößen als Eingaben. Typischerweise werden für den Teilsatz der Steuer- und Hilfsgrößen diejenigen Steuer- und Hilfsgrößen ausgewählt, die Sprungantworten mit dem höchsten Verstärkungsfaktor und der schnellsten Antwortzeit für eine der Stellgrößen aufweisen. Diese Technik ermöglicht es, die MPC-Routine zu verwenden, die die gleiche Anzahl von Eingaben und Ausgaben hat, während es gleichzeitig ermöglicht wird, dass eine Optimierungsroutine den optimalen Betriebspunkt des Prozesses unter Verwendung einer Optimierungsroutine bestimmt, die mehr Steuer- und Hilfsgrößen einbezieht, als Stellgrößen innerhalb des Prozesses vorliegen. Diese Technik ist stabil, weil sie so arbeitet, dass sie in Reaktion auf Störungen und Rauschen innerhalb des Prozesses schnell auf einen gewünschten Betriebspunkt zu steuert, und ist weiterhin rechnerisch einfach, da sie keine Neuberechnung der MPC-Matrix während jedes Scans des Prozesssteuerungssystems erfordert.

[0013] Fig. 1 ist ein Blockdiagramm eines Prozesssteuerungssystems, enthaltend ein Steuerungsmodul mit einem erweiterten, einen Optimierer mit einer MPC-Steuerung integrierenden Steuerungsfunktionsblock;

[0014] Fig. 2 ist ein Blockdiagramm des erweiterten, einen integrierten Optimierer und eine MPC-Steuerung enthaltenden Steuerungsfunktionsblocks aus Fig. 1;

[0015] Fig. 3 ist ein die Art und Weise einer Erzeugung und Installation des integrierten Optimierers und des MPC-Steuerungsfunktionsblock aus Fig. 2 illustrierendes Flussdiagramm;

[0016] Fig. 4 ist ein den Betrieb des integrierten Optimierers und der MPC-Steuerung aus Fig. 2 während des Onlineprozessbetriebs illustrierendes Flussdiagramm;

[0017] Fig. 5 ist eine einen erweiterten Steuerblock innerhalb eines Steuermoduls verdeutlichende Bildschirmanzeige einer die Prozesssteuerung ausführenden Konfigurationsroutine;

[0018] Fig. 6 ist eine Dialogbox darstellende, auf die Eigenschaften des erweiterten Steuerblocks von Fig. 5 verweisende Bildschirmanzeige einer Konfigurationsroutine;

[0019] Fig. 7 ist eine die Art und Weise einer Auswahl und eine Spezifizierung von Eingaben zu und Ausgaben aus einem erweiterten Steuerungsfunktionsblock illustrierende Bildschirmdarstellung einer in der Darstellung von Fig. 5 abgebildeten Konfigurationsroutine;

[0020] Fig. 8 ist eine von einer Konfigurationsroutine bereitgestellte Bildschirmdarstellung, die es einem Nutzer oder Operator ermöglicht, eine Zielfunktion aus einem Satz von Zielfunktionen auszuwählen, die zum Erzeugen eines erweiterten Steuerungsblocks benötigt werden.

[0021] Fig. 9 ist eine Bildschirmdarstellung eines Testbildschirms, der dazu verwendet werden kann, um es einem Nutzer zu ermöglichen, ein Testen und ein Erzeugen eines Prozessmodells während des Erzeugens eines erweiterten Steuerungsblocks auszuführen;

[0022] Fig. 10 ist eine Bildschirmdarstellung einer eine Anzahl von Sprungantworten darstellenden Konfigurationsroutine, die die Antwort verschiedener Steuer- und Hilfsgrößen auf eine bestimmte Stellgröße angeben;

[0023] Fig. 11 ist eine Bildschirmdarstellung einer Konfigurationsroutine, die die Art und Weise der Auswahl einer als primär mit der Stellgröße verknüpften Steuer- und Hilfsgröße aus Fig. 9 illustriert;

[0024] Fig. 12 ist eine Bildschirmdarstellung einer eine Anzahl von Sprungantworten darstellenden Konfigurationsroutine, die die Antwort der gleichen Steuer- und Hilfsgrößen auf verschiedene Stellgrößen angeben;

[0025] Fig. 13 ist eine Bildschirmdarstellung die die Art und Weise eines Kopierens einer in ein verschiedenes Anwendungsmodell zu kopierenden Sprungantwort zeigt;

[0026] Fig. 14 ist eine die Art und Weise einer Ansicht und einer Änderung einer Sprungantwortkurve darstellende Bildschirmanzeige;

[0027] Fig. 15 ist eine einen Dialogbildschirm darstellende Bildschirmanzeige, der Informationen an den Operator während des Betriebs des erweiterten Steuerungsblocks bereitstellt; und

[0028] Fig. 16 ist eine einen Diagnosebildschirm darstellende Bildschirmanzeige, der einem Nutzer oder Operator zum Ausführen von Diagnosen auf dem erweiterten Steuerungsblock bereitgestellt wird.

[0029] Unter Bezugnahme auf Fig. 1 enthält ein Prozesssteuerungssystem **10** eine Prozesssteuerung **11**, die kommunikativ mit einem Datenhistorian **12** und einer oder mehreren Hostworkstations oder Computern **13** (die von jedem Typ von Personalcomputer, Workstations usw. sein können) verbunden sind, wobei diese jeweils einen Anzeigebildschirm **14** aufweisen. Die Steuerung **11** ist ebenfalls mit den Feldgeräten **15** bis **22** über Eingabe/Ausgabe-(I/O)-Karten **26** und **28** verbunden. Der Datenhistorian **12** kann von jedem gewünschten Typ einer Datensammeleinheit sein. Sie kann jeden gewünschten Speichertyp und jede gewünschte oder bekannte Software, Hardware oder Firmware zum Speichern von Daten aufweisen, und kann (wie in Fig. 1 gezeigt) von einer der Workstations **13** abgetrennt oder als ein Teil von diesen ausgeführt sein. Die Steuerung **11**, die z. B. eine von Fisher-Rosemount-Systems Inc. vertriebene Delta V™ Steuerung sein kann, ist kommunikativ mit den Hostcomputern **13** und dem Datenhistorian **12** bspw. über eine Ethernetverbindung oder jedes andere gewünschte Kommunikationsnetzwerk **29** verbunden. Das Kommunikationsnetzwerk **29** kann in der Form eines Local Area Network (LAN), eines Wide Area Network (WAN), eines Telekommunikationsnetzwerks und dgl. ausgeführt und unter Verwendung fest verdrahteter oder drahtloser Technologie verwirklicht sein. Die Steuerung **11** ist kommunikativ mit den Feldgeräten **15** bis **22** unter Verwendung jeglicher erforderlicher Hardware und Software verbunden, die bspw. mit 4–20 mA Standardgeräten und/oder jedem Smart-Kommunikationsprotokoll, d. h. einem intelligenten Kommunikationsprotokoll, z. B. dem FOUNDATION-Feldbusprotokoll (Fieldbus), dem HART-Protokoll usw. verknüpft ist.

[0030] Die Feldgeräte **15** bis **22** können von jedem Gerätetyp, z. B. Sensoren, Ventile, Transmitter, Stellglieder und so weiter sein, während die I/O-Karten **26** und **28** von jedem I/O-Gerätetyp sein können, der mit jedem erforderlichen Kommunikations- oder Steuerprotokoll konform ist. In der in Fig. 1 gezeigten Ausführungsform sind die Feldgeräte **15** bis **18** 4–20 mA-Standardgeräte, die über analoge Leitungen mit der IO-Karte **26** kommunizieren, während die Feldgeräte **19** bis **22** intelligente Einrichtungen, bzw. Smartgeräte, z. B. Fieldbus-Feldgeräte sind, die über einen digitalen Bus mit der IO-Karte **26** unter Verwendung von Fieldbusprotokoll-Kommunikationen kommunizieren. Natürlich können die Feldgeräte **15** bis **22** passend zu beliebigen anderen geforderten Standards oder Protokollen unter Einschluß beliebiger zukünftig sich entwickelnder Standards oder Protokollen ausgeführt sein.

[0031] Die Steuerung **11**, die eine von mehreren innerhalb der Anlage **10** verteilten Steuerungen sein kann und über mindestens einen eingefügten Prozessor verfügt, verwirklicht oder überwacht eine oder mehrere Prozesssteuerungsroutinen, welche in diese eingespeicherte oder ihnen auf andere Weise zugeordnete Steuererschleifen enthalten können. Die Steuerung **11** kommuniziert ebenfalls mit den Geräten **15** bis **22**, den Hostcomputern **13** und dem Datenhistorian **12**, um den Prozess auf jede gewünschte Weise zu steuern. Es ist zu betonen, dass jede der hier beschriebenen Steuerungen oder Elemente Bestandteile enthalten können, die wenn gewünscht, durch unterschiedliche Steuerungen oder von anderen Geräten ausgeführt oder auf diesen realisiert sein können. Ebenso können die hier beschriebenen und in das Prozesssteuerungssystem **10** implementierten Steuerungen oder Elemente jede Form, einschließlich Software, Firmware, Hardware usw. annehmen. Für den Zweck der hier gegebenen Darstellung kann ein Prozesssteuerelement jeder beliebige Teil oder Abschnitt eines Prozesssteuerungssystems, einschließlich z. B. eine Routine, ein Block oder ein auf jedem computerlesbaren Medium gespeichertes Modul sein. Die Steuerungen, die Module oder ein beliebiger Abschnitt einer Steuerprozedur, z. B. eine Subroutine, Teile einer Subroutine (z. B. Codezeilen) usw. sein können, können in jedem gewünschten Softwareformat, z. B. unter Verwendung von Ladder-Logic, sequenziellen Funktionslisten, Funktionsblockdiagrammen, objektorientierter Programmierung oder jeder anderen Softwareprogrammiersprache oder Designparadigmas ausgeführt sein. Ebenso können die Steuerungen in beispielsweise einem oder mehreren EPROM, EEPROM, anwendungsspezifischen integrierten Schaltkreisen (ASICs) oder jedem anderen Hardware- oder Firmwareelement fest encodiert sein. Weiterhin können die Steuerungen unter Verwendung jedes Designwerkzeugs, einschließlich graphischer Designwerkzeuge oder jedes anderen Typs einer Software/Hardware/Firmware-Programmierung oder Designwerkzeugs gestaltet sein. Die Steuerung **11** kann daher zum Verwirklichen einer Steuerstrategie oder Steueroutine in jeder gewünschten Weise konfiguriert sein.

[0032] In einer Ausführungsform verwirklicht die Steuerung **11** eine Steuerstrategie unter Verwendung von üblicherweise als Funktionsblöcke bezeichneten Objekten, wobei jeder Funktionsblock ein Bestandteil oder Objekt einer übergeordneten Steueroutine ist und in Verknüpfung mit anderen Funktionsblöcken (über Kommunikationsbeziehungen, die als Links oder Verbindungen bezeichnet werden) betrieben wird, um Prozesssteuerschleifen innerhalb des Prozesssteuerungssystems **10** zu verwirklichen. Die Funktionsblöcke führen typischerweise entweder eine Eingabefunktion, die bspw. mit einem Transmitter, einem Sensor oder anderen Messgeräten für Prozessparameter verknüpft ist, eine Steuerfunktion, die z. B. einer PID-, Fuzzylogik-, und dgl. Steuerung ausführenden Steueroutine zugeordnet ist oder eine Ausgabefunktion aus, die den Betrieb eines Gerätes, z. B. eines Ventils, steuert, um eine physikalische Funktion in dem Prozesssteuerungssystem **10** auszuführen. Natürlich gibt es auch hybride oder andere Funktionsblocktypen. Die Funktionsblöcke können in der Steuerung **11** gespeichert und durch die Steuerung **11** ausgeführt werden. Dies ist typischerweise dann der Fall, wenn diese Funktionsblöcke für 4–20 mA-Standardgeräte genutzt werden oder diesen und anderen Typen intelligenter Feldgeräte wie bspw. HART-Geräten, zugeordnet sind. Sie können auch in den Feldgeräten selbst gespeichert und durch die realisiert sein, wobei dies bei Feldbus-Geräten der Fall sein kann. Während die an dieser Stelle gegebene Beschreibung des Steuerungssystems eine Funktionsblock-Steuerungsstrategie anwendet, die ein objektorientiertes Programmparadigma nutzt, können die Steuerstrategie oder Steuererschleifen oder Module auch unter Verwendung anderer Konventionen, z. B. Ladder-Logic, sequenzieller Funktionslisten (function charts) usw. oder jeder anderen Programmiersprache oder jedes Paradigmas verwirklicht und gestaltet sein.

[0033] Wie durch den vergrößerten Block **30** aus **Fig. 1** illustriert ist, kann die Steuerung **11** eine Anzahl von Einzel-Schleifen-Steuerungen, dargestellt als die Routinen **32** und **34**, enthalten, und eine oder mehrere erweiterte Steuerschleifen, dargestellt als die Steuerschleife **36**, ausführen. Jede Schleife wird typischerweise als ein Steuermodul bezeichnet. Die Einzel-Schleifen-Steuerungen **32** und **34** sind als Einzel-Schleifen-Steuerungen ausführende Routinen dargestellt, die einen Einzeleingang/Einzelausgang-Fuzzylogik-Steuerblock bzw. einen Einzeleingang/Einzelausgang-PID-Steuerblock nutzen. Sie sind mit geeigneten Funktionsblöcken für einen Analogeingang (AI) und einen Analogausgang (AO) verbunden. Diese können Prozesssteuerungsgeräten, z. B. Ventilen, Messgeräten, bspw. Temperatur- und Messtransmittern oder jedem anderen Gerät innerhalb des Prozesssteuerungssystems **10** zugeordnet sein. Die erweiterte Steuerschleife **36** ist als einen erweiterten Steuerblock **38** enthaltend dargestellt. Dieser weist Eingänge auf, die kommunikativ mit verschiedenen AI-Funktionsblöcken verbunden sind und Ausgänge auf, die mit verschiedenen AO-Funktionsblöcken verbunden sind. Darüber hinaus können die Eingänge und Ausgänge des erweiterten Steuerblocks **38** kommunikativ mit jeden anderen gewünschten Funktionsblöcken oder Steuerelementen verbunden sein, um andere Eingabetypen zu empfangen und andere Steuerausgabetypen bereitzustellen. Wie im Folgenden beschrieben wird, kann der erweiterte Steuerblock **38** ein Steuerblock sein, der eine Modell prädiktive Steueroutine mit einem Optimierer integriert, um eine optimierte Steuerung des Prozesses oder eines Abschnittes des Prozesses auszuführen. Während der erweiterte Steuerblock **38** hier als ein ein Modell prädiktiver Steuerblock

(MPC) beschrieben wird, kann der erweiterte Steuerblock **38** jede andere Steueroutine mit multipler Eingabe/multipler Ausgabe oder Prozedur, z. B. als eine ein neuronales Netzwerk modellierende Steueroutine, eine multivariable Fuzzylogik-Steueroutine usw. enthalten. Es versteht sich, dass die in **Fig. 1** dargestellten Funktionsblöcke, einschließlich des erweiterten Steuerblocks **38**, durch die Steuerung **11** ausgeführt oder alternativ in jedem anderen verarbeitenden Gerät, z. B. eine der Workstations **13** oder sogar eines der Feldgeräte **19–22**, lokalisiert und von diesen ausgeführt werden können.

[0034] Wie in **Fig. 1** dargestellt, enthält eine der Workstations **13** eine Erzeugungsroutine **40** für den erweiterten Steuerblock, die zum Erzeugen, einem Laden und Implementieren des erweiterten Steuerblocks **38** verwendet wird. Während die Erzeugungsroutine **40** für den erweiterten Steuerblock in einem Speicher innerhalb der Workstation **13** und durch einen darin enthaltenen Prozessor gespeichert und ausgeführt werden kann, ist es möglich, dass diese Routine (oder jeder beliebige Teil derselben) zusätzlich oder alternativ dazu in jedem anderen Gerät innerhalb des Prozesssteuerungssystems **10** gespeichert und ausgeführt wird, falls dies so gewünscht ist. Allgemein gesprochen enthält die Erzeugungsroutine **40** für den erweiterten Steuerblock eine Steuerblockerzeugungsroutine **42**, die einen erweiterten Steuerblock wie im Folgenden beschrieben erzeugt und diesen erweiterten Steuerblock mit dem Prozesssteuersystem verbindet. Weiterhin enthält sie eine Prozessmodellierungsroutine **44**, die ein Prozessmodell für den Prozess oder einen Teil des Prozesses auf der Basis der durch den erweiterten Steuerblock gesammelten Daten erzeugt. Sie weist weiterhin eine Steuerlogikparameter-Erzeugungsroutine **46** auf, die Steuerlogikparameter für den erweiterten Steuerblock aus dem Prozessmodell erzeugt und diese Steuerlogikparameter herunter lädt oder in dem erweiterten Steuerblock zur Verwendung in einer Steuerung des Prozesses speichert, und eine Optimiereroutine **48**, die einen Optimierer zur Verwendung mit dem erweiterten Steuerblock erzeugt. Es ist selbstverständlich, dass die Routinen **42**, **44**, **46** und **48** aus einer Folge verschiedener Routinen bestehen können, bspw. einer ersten Routine, die ein erweitertes Steuerelement erzeugt, das Steuereingänge, die zum Empfangen von Prozessausgaben ausgestaltet sind und Steuerausgänge hat, die zum Bereitstellen von Steuersignalen an die Prozesseingänge angepaßt sind. Weiterhin einer zweiten Routine, die es einem Nutzer ermöglicht, das erweiterte Steuerelement innerhalb der Prozesssteueroutine herunterzuladen und kommunikativ zu verbinden (wobei dies jede gewünschte Konfigurationsroutine sein kann). Eine dritte Routine benutzt das erweiterte Steuerelement zum Bereitstellen von Anregungskurven an jeden der Prozesseingänge. Eine vierte Routine verwendet das erweiterte Steuerelement zum Sammeln von die Antwort auf jede der Prozessausgaben auf die Anregungskurven wiedergebende Daten. Eine fünfte Routine wählt einen Eingabensatz für den erweiterten Steuerblock entweder selbst aus oder ermöglicht es einem Nutzer, diese auszuwählen. Eine sechste Routine erzeugt ein Prozessmodell, eine siebente Routine entwickelt erweiterte Steuerlogikparameter aus dem Prozessmodell. Eine achte Routine platziert die erweiterte Steuerlogik und, falls gebraucht, das Prozessmodell innerhalb des erweiterten Steuerelementes, um dem erweiterten Steuerelement die Steuern des Prozesses zu ermöglichen. Und schließlich wählt eine neunte Routine einen Optimierer zur Verwendung in dem erweiterten Steuerblock **38** entweder selbst aus oder ermöglicht es einem Nutzer, diesen auszuwählen.

[0035] **Fig. 2** zeigt ein detaillierteres Blockdiagramm einer Ausführungsform des erweiterten Steuerblocks **38**, der kommunikativ an einen Prozess **50** gekoppelt ist, wobei, wie zu entnehmen ist, der erweiterte Steuerblock **38** einen Satz von Stellgrößen MV erzeugt, die an andere wiederum mit Steuereingängen des Prozesses **50** verbundene Funktionsblöcke bereitgestellt wird. Wie in **Fig. 2** dargestellt, enthält der erweiterte Steuerblock **38** einen MPC-Steuerblock **52**, einen Optimierer **54**, einen Zielkonversionsblock **55**, eine Sprungantwortmodell- oder Steuermatrix **56** und einen Eingabe-Verarbeitungs/Filter-Block **58**. Die MPC-Steuerung **52** kann jede übliche, quadratische M mal M (wobei M von jeder Zahl größer als 1 sein kann) MPC-Routine oder Prozedur sein, die die gleiche Anzahl von Eingängen und Ausgängen hat. Die MPC-Steuerung **52** empfängt als Eingaben einen Satz von innerhalb des Prozesses **50** gemessenen N Steuerungs- und Hilfsgrößen CV und AV (die Vektoren dieser Werte sind), einen Satz von Störgrößen DV, die bekannte oder erwartete, in dem Prozess **50** zu einem zukünftigen Zeitpunkt auftretende Veränderungen oder Störungen sind, und einen Satz von Stabilzustands-Zielsteuerungs- und -Hilfsgrößen CV_T und AV_T , die von dem Zielkonversionsblock **55** bereitgestellt werden. Die MPC-Steuerung **52** verwendet diese Eingaben, um einen Satz von M Stellgrößen MV (in Form von Steuersignalen) zu erzeugen und gibt die Stellgrößen MV-Signale zum Steuern an den Prozess **50** aus.

[0036] Weiterhin errechnet die MPC-Steuerung **52** einen Satz von vorausgesagten prädiktiven Beharrungszustandssteuergrößen CV_{SS} und -hilfsgrößen AV_{SS} zusammen mit einem Satz von vorausgesagten Beharrungstellgrößen MV_{SS} , die die vorausgesagten Werte der Steuergrößen CV, der Hilfsgrößen AV bzw. der Stellgrößen MV am Steuerhorizont repräsentieren und stellt diese an den Eingabe-Verarbeitungs/Filterblock **58** bereit. Der Eingabe-Verarbeitungs/Filterblock **58** verarbeitet die festgestellten vorausgesagten Beharrungswerte der Steuerungs-, Hilfs- und Stellgrößen CV_{SS} , AV_{SS} und MV_{SS} , um Rauscheffekte und Einflüsse nicht vorausgesagter Störungen auf diese Größen zu reduzieren. Es versteht sich, dass der Eingabe-Verarbeitungs/Filter-

block **58** einen Tiefpass-Filter oder eine beliebige andere Eingabeverarbeitung aufweisen kann, die die Effekte von Rauschen, Modellierungsfehlern und Störungen auf diese Werte reduziert und die gefilterten Steuer-, Hilfs- und Stellgrößen CV_{SSfil} , AV_{SSfil} und MV_{SSfil} an den Optimierer **54** bereitstellt.

[0037] Der Optimierer **54** ist, bei diesem Beispiel, ein Optimierer mit linearer Programmierung (LP), der eine Zielfunktion (OF) benutzt, die von einem Auswahlblock **62** für das Ausführen einer Prozessoptimierung bereitgestellt werden kann. Alternativ dazu kann der Optimierer **54** ein Optimierer mit quadratischer Programmierung sein. Dieser ist ein Optimierer mit einem linearen Modell und einer quadratischen Zielfunktion. Allgemein gesagt wird die Zielfunktion OF mit jeder aus der Zahl der Steuer-, Hilfs- und Stellgrößen verknüpfte Kosten und Gewinne bestimmen und der Optimierer **54** durch eine Maximierung oder Minimierung der Zielfunktion Zielwerte für diese Größen vorgeben. Der Auswahlblock **62** kann die Zielfunktion OF auswählen, die durch den Optimierer **54** als eine aus einem Satz von vorab gespeicherten Zielfunktionen **64** bereitgestellt wird. Diese repräsentieren jeweils verschiedene Arten einer Festlegung eines optimalen Betriebs des Prozesses **50**. Beispielsweise kann eine der vorab gespeicherten Zielfunktionen **64** zum Maximieren des Gewinns der Anlage eingerichtet sein. Eine andere der Zielfunktionen **64** kann daraufhin eingerichtet sein, die Verwendung eines bestimmten Rohmaterials mit begrenztem Vorrat zu minimieren, während schließlich eine andere der Zielfunktionen **64** dazu eingerichtet sein kann, um die Qualität des in dem Prozess **50** hergestellten Produktes zu maximieren. Generell nutzt die Zielfunktion einen mit jeder Bewegung einer Steuer-, Hilfsgröße und Stellgröße verbunden Kosten oder Gewinn, um den optimalsten Prozessbetriebspunkt innerhalb des Satzes akzeptabler Punkte festzulegen, die durch die Sollpunktwerte oder Bereiche der Steuergrößen CV und den Grenzen der Hilfs- und Stellgrößen AV und MV definiert sind. Natürlich kann jede gewünschte Zielfunktion anstelle oder zusätzlich zu den hier beschriebenen Funktionen verwendet werden, einschließlich von Zielfunktionen, die jede betreffende Zahl, z. B. die Verwendung von Rohmaterial, Gewinnabilität usw. in einem gewissen Maße optimieren.

[0038] Um aus den Zielfunktionen **64** eine Funktion auszuwählen, kann ein Nutzer oder Operator einen Verweis auf die zu verwendende Zielfunktion **64** durch eine Auswahl dieser Zielfunktion an einem Operator- oder Nutzerterminal (wie zum Beispiel eine der Workstations **13** aus **Fig. 1**) angeben, wobei diese Auswahl über eine Eingabe **66** an den Auswahlblock **62** geleitet wird. In Antwort auf die Eingabe **66** stellt der Auswahlblock **62** die ausgewählte Zielfunktion OF an den Optimierer **54** bereit. Natürlich kann der Nutzer oder Operator die während des Betriebs des Prozesses verwendete Zielfunktion ändern. Wenn gewünscht, kann eine vorgegebene Zielfunktion in Fällen verwendet werden, bei denen der Nutzer keine Zielfunktion anbietet oder auswählt. Eine mögliche vorgegebene Zielfunktion wird weiter unten genauer beschrieben. Während die Zielfunktionen als Bestandteil des erweiterten Steuerblocks **38** dargestellt sind, können die verschiedenen Zielfunktionen in dem Operatorterminal **13** aus **Fig. 1** gespeichert sein und eine dieser Zielfunktionen kann an den erweiterten Steuerblock **38** während der Erstellung oder Erzeugung dieses Blocks bereitgestellt werden.

[0039] Zusätzlich zu der Zielfunktion OF empfängt der Optimierer **54** als Eingaben einen Satz von Steuergrößen, Sollwerten (die typischerweise spezifizierte Sollwerte für die Steuergrößen CV des Prozesses **50** sind und durch den Operator oder einen anderen Nutzer verändert werden können) und einen Bereich und eine Gewichtung oder eine mit jeder der Steuergrößen CV verknüpfte Priorität. Der Optimierer **54** empfängt zusätzlich dazu einen Satz von Bereichen oder Randbedingungen und einen Satz von Gewichtungen oder Prioritäten für die Hilfsgrößen AV und einen Satz von Grenzen für die Stellgrößen MV, die zur Steuerung des Prozesses **50** verwendet werden. Allgemein gesagt legen die Bereiche der Hilfsgrößen und Stellgrößen die Grenzen (typischerweise auf der Grundlage von physikalischen Eigenschaften der Anlage) für die Hilfsgrößen und Stellgrößen fest, während die Bereiche für die Steuergrößen einen Bereich angeben, in dem die Steuergrößen eine befriedigende Steuerung des Prozesses betreiben. Die Gewichtungen für die Steuer- und Hilfsgrößen legen die relative Wichtigkeit der aufeinander bezogenen Steuergrößen und Hilfsgrößen während des Optimierungsprozesses fest und können gegebenenfalls dazu verwendet werden, um dem Optimierer eine Generierung einer Steuerungsziellösung zu ermöglichen, wenn einige der Randbedingungen verletzt werden.

[0040] Während des Betriebs kann der Optimierer **54** eine lineare Programmierungstechnik (LP) zum Ausführen der Optimierung verwenden. Wie bekannt, ist eine lineare Programmierung ein mathematisches Verfahren zum Lösen eines Satzes von linearen Gleichungen und Ungleichungen, die eine gewisse zusätzliche, als Zielfunktion bezeichnete Funktion maximieren oder minimieren. Wie oben dargestellt, kann die Zielfunktion einen ökonomischen Wert wie Kosten oder Gewinne, aber an deren Stelle auch andere Zielvorgaben ausdrücken. Ferner versteht sich, dass die Beharrungsverstärkungsmatrix die Beharrungszustandsverstärkung für jedes mögliche Paar der Stellgrößen und der Steuer- und Hilfsgrößen definiert. M. a. W. definiert die Beharrungsverstärkungsmatrix die Beharrungszustandsverstärkung in jeder Steuer- und Hilfsgröße für eine Einheitsänderung in jeder der Stellgrößen und Störgrößen. Diese Beharrungsverstärkungsmatrix ist im allgemeinen eine N

mal M-Matrix, wobei N die Anzahl der Steuer- und Hilfsgrößen und M die Anzahl der in der Optimieroutine verwendeten Stellgrößen ist. Im allgemeinen kann N größer, gleich oder kleiner als M sein, mit dem meist auftretenden Fall, dass N größer als M ist.

[0041] Unter Verwendung jedes bekannten oder üblichen LP-Algorithmus oder -Verfahrens, führt der Optimierer **54** Iterationen aus, um den Satz von Zielstellgrößen MV_T (wie durch die Beharrungszustandsverstärkungsmatrix festgelegt) zu finden, der die ausgewählte Zielfunktion OF maximiert oder minimiert. Daraus ergibt sich ein Prozessbetrieb, der die Sollwertbereichsgrenzen der Steuergrößen CV, die Randbedingungsgrenzen der Hilfsgrößen AV und die Grenzen der Stellgrößen MV trifft oder in diese hinein fällt. In einer Ausführungsform stellt der Optimierer **54** auch die Veränderungen in den Stellgrößen fest und wendet die Angabe auf die vorausgesagten Beharrungszustands-Steuergrößen, -Hilfsgrößen und -Stellgrößen CV_{ssfil} , AV_{ssfil} und MV_{ssfil} an, um die Veränderung in dem Prozessbetrieb aus dessen aktuellem Betrieb heraus festzustellen und somit den dynamischen Betrieb der MPC-Steueroutine während des Prozesses der Annäherung an das Ziel oder den optimalen Prozessbetriebspunkt festzustellen. Dieser dynamische Betrieb ist von großer Bedeutung, weil es notwendig ist, sicherzustellen, dass keine der Randbedingungsgrenzen während der Bewegung von dem aktuellen Betriebspunkt zu dem Zielbetriebspunkt verletzt werden.

[0042] In einer Ausführungsform kann der LP-Optimierer **54** ausgeführt sein, um eine Zielfunktion der folgenden Form zu minimieren:

$$Q = P^t \cdot A \cdot \Delta MV + C^t \Delta MV$$

wobei:

Q = Gesamtkosten/Gewinn

P = den AVs und CVs zugeordneter Gewinnvektor

C = den MVs zugeordneter Kostenvektor

A = Verstärkungsmatrix

ΔMV = Vektor für eine berechnete Veränderung in den MVs.

[0043] Die Gewinnwerte sind positive Zahlen und die Kostenwerte negative Zahlen, um deren Einfluss auf die Zielsetzung anzuzeigen. Unter Anwendung dieser Zielfunktion errechnet der LP-Optimierer **54** diejenigen Veränderungen in den Stellgrößen MV, die die Zielfunktionen minimieren. Dabei wird gesichert, dass die Steuergrößen CV innerhalb eines Bereiches um den Zielsollwert verbleiben, die Hilfsgrößen AV innerhalb ihrer oberen und unteren Randbedingungsgrenzen verbleiben und dass die Stellgrößen MV innerhalb ihrer oberen und unteren Grenzen sind.

[0044] In einer zu verwendenden Optimierungsprozedur werden inkrementelle Werte von Stellgrößen an der aktuellen Zeit (t) und eine Summe von Inkrementen der Stellgrößen über den Steuerhorizont mit inkrementellen Werten der am Ende des Prädiktionshorizonts festgestellten Steuer- und Hilfsgrößen anstelle von auf den Zustand bezogenen aktuellen Werten verwendet. Dies ist für LP-Anwendungen typisch. Natürlich kann der LP-Algorithmus in geeigneter Weise für diese Variation modifiziert werden. Auf jeden Fall kann der LP-Optimierer **54** ein Beharrungszustandsmodell verwenden und daher ist eine Beharrungszustandsbedingung für diese Anwendung erforderlich. Mit einem normalerweise in der MPC-Gestaltung verwendeten Prädiktionshorizont ist ein zukünftiger Beharrungszustand für einen selbstregulierenden Prozess garantiert. Eine mögliche vorausgesagte Prozessbeharrungszustandsgleichung für einen $m \times n$ Eingabe-Ausgabeprozess mit einem Prädiktionshorizont p, einem Steuerhorizont c, ausgedrückt in der inkrementellen Form ist:

$$\Delta CV(t + p) = A \cdot \Delta MV(t + c)$$

wobei

$$\Delta CV(t + p) = \begin{bmatrix} \Delta cv_1 \\ \dots \\ \Delta cv_n \end{bmatrix}$$

die vorausgesagten Veränderungen in den Ausgaben am Ende des Prädiktionshorizonts (t + p) bezeichnet,

$$A = \begin{bmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nm} \end{bmatrix}$$

die Prozessbeharrungszustands- $m \times n$ -Verstärkungsmatrix ist,

$$\Delta MV(t+c) = \begin{bmatrix} \Delta mv_1 \\ \dots \\ \Delta mv_m \end{bmatrix}$$

die Veränderungen in den Stellgrößen am Ende des Steuerungshorizontes ($p+c$) bezeichnet.

[0045] Der Vektor $\Delta MV(p+c)$ repräsentiert die Summe der Veränderungen über den Steuerhorizont, die durch jede Steuerungsausgabe mv_i gemacht wurden, sodass gilt:

$$\Delta mv_i = \sum_{j=1}^c mv_i(t+j) \quad i = 1, 2, \dots, m.$$

[0046] Die Veränderungen sollten die Grenzen sowohl der Stellgrößen MV als auch der Steuergrößen CV erfüllen (hier werden die Hilfsgrößen als Steuergrößen behandelt):

$$MV_{\min} \leq MV_{\text{current}} + \Delta MV(t+c) \leq MV_{\max}$$

$$CV_{\min} \leq CV_{\text{predicted}} + \Delta CV(t+p) \leq CV_{\max}$$

[0047] In diesem Fall kann die Zielfunktion zum Maximieren des Produktwertes und zum Minimieren der Rohmaterialkosten zusammenfassend so definiert werden:

$$Q_{\min} = -UCV^T * \Delta CV(t+p) + UMV^T * \Delta MV(t+c)$$

wobei:

UCV der Kostenvektor für eine Einheitsänderung in dem Steuergrößen CV-Prozesswert ist; und
UMV der Kostenvektor für eine Einheitsänderung in dem Stellgrößen MV-Prozesswert ist.

[0048] Unter Anwendung der oben genannten ersten Gleichung kann die Zielfunktion in Termen der Stellgrößen MV ausgedrückt werden als:

$$Q_{\min} = -UCV^T * A * \Delta MV(t+c) + UMV^T * \Delta MV(t+c)$$

[0049] Um eine optimale Lösung zu ermitteln, berechnet der LP-Algorithmus die Zielfunktionen für einen initialen Eckpunkt in dem durch diese Gleichung definierten Gebiet, und verbessert die Lösung an jedem nächst folgenden Schritt, bis der Algorithmus den Eckpunkt mit dem maximalen (oder minimalen) Wert der Zielfunktion als eine optimale Lösung ermittelt. Der festgestellte optimale Stellgrößenwert wird als die zu erhaltenden Zielstellgröße MV_T innerhalb des Steuerhorizontes verwendet.

[0050] Allgemein gesagt gibt der Ablauf des LP-Algorithmus auf der vorbereiteten Matrix drei mögliche Resultate zurück. In einem ersten Fall gibt es eine eindeutige Lösung für die Zielstellgrößen MV_T . In einem zweiten Fall ist die Lösung unbeschränkt, wobei dies nicht auftreten sollte, wenn jede Steuer- und Hilfsgröße eine oberen und eine unteren Grenze besitzen. In einem dritten Fall gibt es keine Lösung, wobei dies bedeutet, dass die Grenzen oder Abhängigkeiten der Hilfsgrößen zu eng sind. Um den dritten Fall zu behandeln können die allgemeinen Randbedingungen gelockert werden, um eine Lösung zu erhalten. Die grundlegende Prämisse dabei ist, dass die Grenzen der Stellgrößen (Hi/Lo-Grenzen) nicht durch den Optimierer geändert werden können. Das Gleiche trifft auch für die Grenzen oder Randbedingungen der Hilfsgröße zu (Hi/Lo-Grenzen). Allerdings kann der Optimierer von einem Ansteuern der Steuergröße CV an den spezifizierten Sollpunkt (CV-

Sollwertsteuerung) zu einem Ansteuern der Steuergrößen zu jedem der Werte innerhalb eines Bereiches von oder um den Sollwert herum wechseln (CV-Bereichssteuerung). In diesem Fall wird es den Werte der Steuergrößen erlaubt, sich innerhalb eines Bereiches anstatt an einem spezifischen Sollwert einzustellen. Gibt es mehrere Hilfsgrößen AV, die ihre Randbedingungen verletzen und ergibt das Umschalten von der CV-Sollwertsteuerung zur CV-Bereichssteuerung keine Lösung, ist es auch möglich, die Randbedingungen der Hilfsgrößen auf der Grundlage der vorliegenden Gewichtungen oder Prioritätsangaben freizugeben oder zu ignorieren. In einer Ausführungsform kann beispielsweise eine Lösung durch eine Minimierung des quadratischen Fehlers der Hilfsgrößen, die es jeder Größe erlaubt, ihre jeweiligen Randbedingungen zu verletzen oder durch ein Ungültigsetzen der Randbedingungen der Hilfsgröße mit niedrigster Priorität in einer fortschreitenden Weise ermittelt werden.

[0051] Wie oben bemerkt, kann die Zielfunktion OF ausgewählt werden oder als Voreinstellung durch das Steuerblockerzeugungsprogramm **40** gesetzt sein. Eine Methode zum Erstellen einer solchen Voreinstellung wird weiter unten bereitgestellt. Während es insbesondere erforderlich ist, das Optimierungsvermögen bereitzustellen, fordern viele Situationen möglicherweise nur, dass die Sollwerte für die Steuergrößen in einer Weise behandelt werden, die die Betriebs-Randbedingungen der Hilfsgrößen und Stellgrößen nach wie vor befolgt. Für diese Anwendungen kann der Block **38** so konfiguriert sein, dass dieser ausschließlich als MPC-Funktionsblock betrieben wird. Um diesen Bedienkomfort bereitzustellen, kann eine vorab eingestellte „Betriebs“zielfunktion automatisch mit voreingestellten Kosten erzeugt werden, die den verschiedenen Größen darin und einhergehend mit vorgegebenen Hilfsgrößen AV-Gewichtungen zugeordnet sind. Diese Voreinstellungen können alle Kosten für die Hilfsgrößen AV und die Stellgrößen MV gleichsetzen oder eine andere voreingestellte Kostenzuordnungen für die Hilfsgrößen und Stellgrößen AV und MV bereitstellen. Wenn eine Expertenoption ausgewählt worden ist, kann der Nutzer zusätzliche Optimierungsauswahlen erzeugen und deren damit zugeordnete Kosten für die verschiedenen Zielfunktionen **64** festlegen. Dem Expertennutzer wird es ebenfalls erlaubt, die vorgegebenen Hilfsgrößen- und die Steuergrößen-AV und CV-Gewichtungen für die vorgegebene Zielfunktion zu modifizieren.

[0052] Bei einer Ausführungsform, wenn z. B. Wirtschaftlichkeiten für die Prozesskonfigurationen nicht definiert sind, kann die Zielfunktion aus der MPC-Konfiguration automatisch konstruiert werden. Im Allgemeinen kann die Zielfunktion unter Anwendung folgender Formel konstruiert werden:

$$CD^T = C^T + P^T * A = [C_1, \dots, C_j, \dots, C_m] + \left[\sum_{i=1}^n p_i a_{i1}, \sum_{i=1}^n p_i a_{i2}, \dots, \sum_{i=1}^n p_i a_{im} \right] = [CD_1, \dots, CD_j, \dots, CD_m]$$

[0053] Die Größen C_j und p_j können aus den Konfigurationseinstellungen definiert werden. Insbesondere ist unter der Annahme, dass der Sollwert der Steuergröße CV nur an LL oder HL definiert werden kann, der p_j -Wert in folgender Weise definiert:

- $p_j = -1$, wenn der Sollwert an LL definiert ist oder Minimieren ausgewählt wurde; und
- $p_j = 1$ wenn der Sollwert an HL definiert ist oder Maximieren ausgewählt wurde.

[0054] Unter der Annahme, dass keine Konfigurationsinformation für die Hilfsgrößen AV eingegeben wurde, ist $p_j = 0$ für alle Hilfsgrößen AV. In vergleichbarer Weise hängt für die Stellgrößen MV der C_j -Wert davon ab, ob das bevorzugte Stellgrößenziel MV_T definiert ist oder nicht.

[0055] Wenn das bevorzugte Stellgrößenziel MV_T definiert sind, gilt:

- $C_j = 1$, wenn MV_T an HL (obere Grenze) ist oder Maximieren gewählt wurde,
- $C_j = -1$, wenn MV_T an LL (untere Grenze) ist oder Minimieren ausgewählt wurde, und
- $C_j = 0$, wenn MV_T nicht definiert ist.

[0056] Wenn gewünscht kann die Auswahl für die Benutzung des Optimierers **54** in Zusammenhang mit der MPC-Steuerung **52** einstellbar sein, um somit einen gewissen Optimierungsgrad zu ermöglichen. Um diese Funktion auszuführen, kann die Änderung in den von der Steuerung **52** verwendeten Stellgrößen MV durch ein Anlegen verschiedener Gewichtungen an die Änderungen in den Stellgrößen MV, die durch die MPC-Steuerung **52** und den Optimierer **54** festgelegt werden, abgeändert werden. Eine solche gewichtete Kombination der Stellgrößen MV wird hier als eine effektive MV (MV_{eff}) bezeichnet. Die effektive MV_{eff} kann wie folgt gewonnen werden:

$$\Delta MV_{eff} = \Delta MV_{mpc}(1 + \alpha/S) + \Delta MV_{opt}(1 - \alpha) \quad 0 < \alpha < 1$$

wobei S willkürlich oder heuristisch ausgewählt wird. Typischerweise wird S größer als 1 und in der Größenordnung von 10 sein.

[0057] Hier, mit $\alpha = 1$, trägt der Optimierer zu der effektiven Ausgabe bei, wie dies bei der Generierung vorgegeben worden ist. Mit $\alpha = 0$ stellt die Steuerung nur eine MPC-dynamische Steuerung bereit. Natürlich stellt der Bereich zwischen 0 und 1 verschiedene Beiträge des Optimierers und der MPC-Steuerung bereit.

[0058] Die oben beschriebene Standard-Zielfunktion kann dazu verwendet werden, um den Betrieb des Optimierers während unterschiedlicher möglicher Betriebsmodi desselben zu verwirklichen. Insbesondere dann, wenn die Anzahl der Steuergrößen CV der Anzahl der Stellgrößen MV entspricht, ist das erwartete Verhalten mit den Standard-Einstellwerten derart, dass die Sollgrößen der Steuergröße CV aufrecht erhalten werden, solange die Hilfsgrößen AV und die Stellgrößen MV innerhalb ihrer Grenzen zu erwarten sind. Wenn vorherzusehen ist, dass eine Hilfsgröße oder eine Stellgröße ihre Grenze verletzen wird, werden die Betriebssollgrößen der Steuergrößen innerhalb ihres Bereiches geändert, um diese Grenzen vor einer Verletzung zu bewahren. Wenn in diesem Fall der Optimierer **54** keine Lösung finden kann, die die Grenzen der Hilfsgrößen und Stellgrößen erfüllt und dabei die Steuergrößen innerhalb ihres Bereichs belässt, werden die Steuergrößen innerhalb ihres Bereiches aufrecht erhalten, während es den Hilfsgrößen erlaubt wird, sich von ihren Randbedingungen-Grenzen zu entfernen. Beim Auffinden der besten Lösung werden diejenigen Hilfsgrößen AV, bei denen mit einem Verletzen einer Grenze zu rechnen ist, gleichbehandelt und ihre durchschnittliche Grenzabweichung minimiert.

[0059] Um dieses Verhalten zu erreichen, werden die in der Zielfunktion verwendeten Standard-Kosten/Gewinne automatisch so eingestellt, dass den Steuergrößen CV ein Gewinn von 1 zugeordnet wird, wenn der Bereich definiert ist, dass dieser eine Abweichung unter den Sollwert erlaubt, und den Steuergrößen CV ein Gewinn von -1 zugeordnet wird, wenn der Bereich so definiert ist, dass dieser eine Abweichung über den Sollwert erlaubt. Den Hilfsgrößen AV innerhalb der Grenzen wird ein Gewinn von 0 zugeordnet und den Stellgrößen MV wird ein Kosten von 0 zugeordnet.

[0060] Wenn die Anzahl der Steuergrößen CV kleiner als die Anzahl der Stellgrößen MV ist, können dann die zusätzlichen Freiheitsgrade dazu verwendet werden, um die Bedingungen anzusteuern, die der finalen Ruhelage der konfigurierten Stellgrößen MV zugeordnet sind. Dann werden die Sollwerte der Steuergröße (sofern Steuergrößen CV definiert sind) aufrecht erhalten, solange die Hilfsgrößen und Stellgrößen innerhalb ihrer Grenzen zu erwarten sind. Die Durchschnittsabweichung der Stellgrößen von der konfigurierten finalen Ruheposition wird minimiert. Wenn zu erwarten ist, dass eine oder mehrere der Hilfsgrößen und Stellgrößen ihre Grenzen verletzen werden, werden die Betriebssollwerte der Steuergrößen innerhalb ihrer Bereiche geändert, um eine Verletzung dieser Grenzen zu verhindern. Falls Mehrfachlösungen existieren, wird unter dieser Bedingung die für die Steuerung verwendete Lösung die Durchschnittsabweichung der Stellgrößen von der konfigurierten finalen Ruhelage minimieren.

[0061] Wenn der Optimierer **54** keine Lösung finden kann (d. h. dass eine Lösung nicht existiert), die die Hilfsgrößen- und Stellgrößengrenzen nicht erfüllt, während die Steuergrößen innerhalb ihres Bereiches aufrecht erhalten werden, so werden dann die Steuergrößen innerhalb eines Bereiches gehalten, während es den Hilfsgrößen erlaubt wird, sich von deren Randbedingungen-grenzen zu lösen. Mit dem Auffinden der besten Lösung werden die Hilfsgrößen, von denen Verletzung einer Grenze vorauszusehen ist, gleich behandelt, und deren durchschnittliche Grenzabweichung wird minimiert. Um dieses Verhalten zu erreichen, werden die voreingestellten, durch die Zielfunktionen benutzten Kosten/Gewinne automatisch so eingestellt, dass den Steuergrößen ein Gewinn von 1 zugeordnet wird, wenn der Bereich so definiert ist, dass dieser eine Abweichung unter den Sollwert erlaubt, und ein Gewinn von -1 , wenn der Bereich so definiert ist, dass dieser eine Abweichung über den Sollwert erlaubt. Den Hilfsgrößen wird ein Gewinn von 1 oder -1 und den Stellgrößen wird ein Kosten von 0.1 zugeordnet.

[0062] In jedem Fall stellt nach dem Betrieb der Optimierer **54** den Satz von optimalen oder Zielstellgrößen MV_T an den Zielkonversionsblock **55** bereit, der die Beharrungszustands-Verstärkungsmatrix anwendet, um die Zielbeharrungszustands-Steuerung und die Stellgrößen festzulegen, die sich aus den Zielstellgrößen MV_T ergeben. Diese Umformung ist rechnerisch in einfacher Weise auszuführen, weil die Beharrungszustands-Verstärkungsmatrix die Wechselwirkung zwischen den Stellgrößen und den Steuer- und Hilfsgrößen definiert und aus diesem Grund dazu verwendet werden kann, um die Zielstellgrößen und Hilfsgrößen CV_T und AV_T eindeutig aus den definierten Ziel(Beharrungszustands)-Stellgrößen MV_T zu gewinnen.

[0063] Einmal festgelegt, wird mindestens ein Teilsatz aus N Ziel-Steuer- und Hilfsgrößen CV_T und AV_T als Eingaben an die MPC-Steuerung **52** bereitgestellt, die, wie vorhergehend erwähnt, diese Zielgrößen CV_T und AV_T verwendet, um einen neuen Satz von Beharrungswert-Stellgrößen (über den Steuerhorizont) MV_{SS} abzuleiten, die die aktuellen Steuergrößen und Stellgrößen CV und AV in Richtung der Zielgrößen CV_T und AV_T am Ende des Steuerhorizonts steuern. Natürlich ändert bekanntermassen die MPC-Steuerung die Stellgrößen in Schritten, mit dem Ziel, diejenigen Beharrungszustandswerte für diese Größen MV_{SS} zu erreichen. Diese werden theoretisch die durch den Optimierer **54** festgelegten Zielstellgrößen MV_T sein. Weil, wie oben erwähnt, der Optimierer **54** und die MPC-Steuerung **52** während jeder Prozessabfrage arbeiten, können sich die Zielgrößen der Stellgrößen MV_T von Abfrage zu Abfrage ändern. Als Resultat hiervon kann die MPC-Steuerung niemals tatsächlich jede einzelne Größe des Satzes der Zielstellgrößen MV_T , vor allem in der Gegenwart von Rauschen, unerwarteten Störungen, Veränderungen innerhalb des Prozesses, usw. erreichen. Allerdings steuert der Optimierer **54** die Steuerung **52** immer so an, dass die Stellgrößen MV in Richtung der optimalen Lösung bewegt werden.

[0064] Wie bekannt, enthält die MPC-Steuerung **52** ein Steuerungsvorhersage-Prozessmodell **70**, das eine N mal $M + D$ Sprungantwortmatrix sein kann (wobei N die Anzahl der Steuergrößen CV plus der Anzahl der Hilfsgrößen AV ist, M die Anzahl der Stellgrößen MV und D die Anzahl der Störgrößen DV ist). Das Steuerungsvorhersage-Prozessmodell **70** erzeugt an einem Ausgang **72** eine vorausberechnete Vorhersage für jede der Steuer- und Hilfsgrößen CV und AV und ein Vektoraddierer **74** subtrahiert für die laufende Zeit diese vorausgesagten Werte von den tatsächlich gemessenen Werten der Steuer- und Hilfsgrößen CV und AV , um einen Fehler oder Korrekptionsvektor an den Eingang **76** bereitzustellen.

[0065] Das Steuerungsvorhersage-Prozessmodell **70** verwendet dann die N mal $M + D$ Sprungantwortmatrix, um einen zukünftigen Steuerparameter für jede der Steuergrößen und Hilfsgrößen CV und AV über den Steuerhorizont auf der Grundlage der Störgrößen und Stellgrößen vorherzusagen, die an andere Eingänge des Steuerungsvorhersage-Prozessmodells **70** bereitgestellt werden. Das Steuerungsvorhersage-Prozessmodell **70** stellt ebenfalls die vorausgesagten Beharrungszustandswerte der Steuergrößen und Hilfsgrößen CV_{SS} und AV_{SS} an den Eingabe-Verarbeitungs/Filterblock **58** bereit.

[0066] Ein Steuerungszielblock **80** legt einen Steuerungszielvektor für jede der N Zielsteuerungs- und -hilfsgrößen CV_T und AV_T fest, die an diesen durch den Zielkonversionsblock **55** unter Anwendung eines vorab für den Block **38** erstellten Trajektorienfilters **82** bereitgestellt werden. Insbesondere stellt der Trajektorienfilter einen Einheitsvektor bereit, der die Vorgehensweise festlegt, in welcher die Steuer- und Hilfsgrößen in an deren Zielwerte im Laufe der Zeit zu steuern sind. Der Steuerungszielblock **80** nutzt diesen Einheitsvektor und die Zielgrößen CV_T und AV_T , um einen dynamischen Steuerungszielvektor für jede derjenigen Steuer- und Hilfsgrößen festzulegen, die die Veränderungen in den Zielgrößen CV_T und AV_T über eine durch die Steuerhorizontzeit festgelegte Zeitperiode definieren. Ein Vektoraddierer **84** subtrahiert dann den zukünftigen Steuerparametervektor für jede der Steuer- und Hilfsgrößen CV und AV von den dynamischen Steuervektoren, um einen Fehlervektor für jede der Steuer- und Hilfsgrößen CV und AV zu ermitteln. Der zukünftige Fehlervektor für jede der Steuer- und Hilfsgrößen CV und AV wird dann dem MPC-Algorithmus bereitgestellt, der so abläuft, dass dieser die Sprünge der Stellgrößen MV auswählt, die bspw. den Fehler der kleinsten Quadrate innerhalb des Steuerhorizonts minimieren. Natürlich verwendet der MPC-Algorithmus oder die Steuerung ein M mal M -Prozessmodell oder eine Steuermatrix, die aus den Verknüpfungen zwischen der Eingabe der N Steuer- und Hilfsgrößen an die MPC-Steuerung **52** und der Ausgabe der M Stellgrößen durch die MPC-Steuerung **52** erstellt wurde.

[0067] Insbesondere verfolgt der mit dem Optimierer zusammen arbeitende MPC-Algorithmus zwei Hauptzielstellungen. Erstens versucht der MPC-Algorithmus den CV -Steuerfehler mit minimalen MV -Bewegungen innerhalb der operativen Einschränkungen zu minimieren. Zweitens versucht der MPC-Algorithmus einen durch den Optimierer aufgestellten Satz von optimalen Beharrungszustands- MV -Werten und einen Zustand der Ziel- CV -Werte zu erreichen, die direkt aus den optimalen Beharrungszustands- MV -Werten errechnet worden sind.

[0068] Um diese Zielstellungen zu befriedigen, kann der anfänglich bedingungsfreie MPC-Algorithmus erweitert werden, um die MV -Ziele in die Lösung der kleinsten Quadrate einzufügen. Die Zielfunktion für diese MPC-Steuerung ist

$$\min_{\Delta MV(k)} \left\{ \left\| \Gamma^y [CV(k) - R(k)] \right\|^2 + \left\| \Gamma^u \Delta MV(k) \right\|^2 + \left\| \Gamma^0 \left[\sum \Delta MV(k) - \Delta MVT \right] \right\|^2 \right\}$$

wobei:

CV(k) der gesteuerte Ausgabe-p-Sprung in Richtung des Vorhersagevektors ist;
 R(k) der p-Sprung in Richtung des Referenztrajektorien(Sollwert)-Vektors ist;
 $\Delta MV(k)$ der c-Sprung in Richtung des inkrementellen Steuerbewegungsvektors ist;
 $\Gamma^y = \text{diag}\{\Gamma_1^y, \dots, \Gamma_p^y\}$ eine Strafmatrix des gesteuerten Ausgabefehlers ist;
 $\Gamma^u = \text{diag}\{\Gamma_1^u, \dots, \Gamma_p^u\}$ eine Strafmatrix der Steuerbewegungen ist;
 p der Prädiktionshorizont ist (Anzahl der Sprünge);
 c der Steuerhorizont ist (Anzahl der Sprünge); und
 Γ^0 eine Strafe auf einen Fehler der Summe der Steuerungsausgabebewegungen im Steuerhorizont bezüglich der durch den Optimierer definierten und angezielten optimalen Veränderung der MV ist.

[0069] Für die Einfachheit der Notation ist die Zielfunktion für eine Einzeleingang/Einzelausgang(SISO – single input/single output)-Steuerung gezeigt.

[0070] Es versteht sich, dass die ersten beiden Terme die Zielfunktion für die bedingungsfreie MPC-Steuerung sind, während der dritte Term eine zusätzliche Bedingung einführt, die die Summe der Steuerungsausgabebewegungen den optimalen Zielen gleich setzt. M. a. W. legen die ersten beiden Terme Zielstellungen für den dynamischen Betrieb der Steuerung fest, während der dritte Term Optimierungszielstellungen des Beharrungszustands festlegt.

[0071] Es ist zu beachten, dass die allgemeine Lösung für diese Steuerung in einer zu der Lösung für die bedingungsfreie MPC-Steuerung vergleichbaren Weise wie folgt ausgedrückt werden kann:

$$\Delta MV(k) = \left(S^{u^T} \Gamma^T \Gamma S^u + \Gamma^{u^T} \Gamma^u \right)^{-1} S^{u^T} \Gamma^T \Gamma E_{p+1}(k) = K_{mpc} E_{p+1}(k)$$

wobei:

$\Delta MV(k)$ die Veränderung in der MPC-Steuerungsausgabe zum Zeitpunkt k ist;
 K_{mpc} die optimierte MPC-Steuerungsverstärkung ist;
 S^u die Prozessdynamik-Matrix ist, die aus den Sprungantworten der Dimension p x c für ein SISO-Modell und p*n* x c*m für ein mehrfach-Eingangs/mehrfach-Ausgangs-(multiple-input/multiple-output)MIMO-Modell mit m Stelleingängen und n Steuerausgängen aufgebaut wurde.

[0072] Für die optimierte MPC wird die dynamische Matrix auf die Größe (p + l) x m für das SISO-Modell und (p + m)*n x c*m für das MIMO-Modell erweitert, um den MV-Fehler anzupassen. $E_{p+1}(k)$ ist der CV-Fehlervektor über den Prädiktionshorizont und der Fehler der Summe der Steuerungsausgabebewegungen über den Steuerungshorizont bezüglich der zieloptimalen Veränderung der MV. Die Matrix Γ verbindet die Matrizen Γ^y und Γ^0 und ist eine quadratische Matrix der Dimension (p + l) für eine SISO-Steuerung und [n(p + n)] für die Mehrgrößen-Steuerung. Das hochgestellte T bezeichnet eine transponierte Matrix.

[0073] Weil der Optimierer **54** auf der Grundlage aller Steuer- und Hilfsgrößen CV und AV optimiert, um einen Zielsatz von einen eindeutigen optimalen Betriebspunkt definierenden Stellgrößen MV_T festzulegen, ist es, wie festgestellt wurde, ohne Bedeutung, dass die MPC-Steuerung **52** nur unter Verwendung eines Teilsatzes der Steuer- und Hilfsgrößen CV und AV in deren Steuermatrix zu einem eigentlich aus dieser heraus erfolgenden Erzeugen der Stellgrößen MV-Ausgabe betrieben wird. Wenn die Steuerung **52** den ausgewählten Teilsatz der Steuer- und Hilfsgrößen CV und AV an die ihnen zugeordneten Ziele steuert, werden sich die anderen Größen des vollständigen Satzes der Steuer- und Hilfsgrößen ebenfalls an ihren Zielwerten befinden. Als Ergebnis wurde festgestellt, dass eine quadratische (MxM)-MPC-Steuerung mit einer M mal M-Steuerungsmatrix mit einem Optimierer verwendet werden kann, der ein rechteckiges (Nxm)-Prozessmodell zum Ausführen der Prozessoptimierung anwendet. Dies ermöglicht eine Anwendung von üblichen MPC-Steuerungsverfahren zusammen mit üblichen Optimierungsverfahren in einer Steuerung, ohne eine nichtquadratische Matrix mit den bei derartigen Umformungen einhergehenden Näherungen und Risiken invertieren zu müssen.

[0074] In einer Ausführungsform, bei der die MPC-Steuerung quadratisch ist, d. h. bei der die Anzahl der Stellgrößen MV gleich der Anzahl der Steuergrößen CV ist, kann das Stellgrößen MV-Ziel durch Veränderungen in den CV-Werten effektiv wie folgt erreicht werden:

$$\Delta CV = A \cdot \Delta MV_T$$

Δ MVT – optimale Zielveränderung der MV

Δ CV – CV-Veränderungen zum Erreichen der optimalen MV.

[0075] Die CV-Veränderung wird durch ein Betreiben der CV-Sollwerte verwirklicht.

[0076] Während des Betriebs setzt der Optimierer **54** die Beharrungszustandsziele für die bedingungsfreie MPC-Steuerung bei jedem Scan auf und aktualisiert diese. Die MPC-Steuerung **52** führt somit einen bedingungsfreien Algorithmus aus. Weil die Ziele CV_T und AV_T so vorgegeben werden, dass die Randbedingungen so lange berücksichtigt werden, wie eine geeignete Lösung existiert, arbeitet die Steuerung innerhalb der Randbedingungsgrenzen. Die Optimierung ist somit ein integraler Bestandteil der MPC-Steuerung.

[0077] Die **Fig. 3** und **Fig. 4** stellen ein Flussdiagramm **90** dar, das die Schritte verdeutlicht, die zum Ausführen einer integrierten modellbasierten prädiktiven Steuerung und Optimierung notwendig sind. Das Flussdiagramm **90** ist in zwei Abschnitte **90a** (**Fig. 3**) und **90b** (**Fig. 4**) unterteilt. Diese verdeutlichen Funktionen, die vor dem Prozessbetrieb (**90a**) und während des Prozessbetriebs (**90b**), d. h. während jeder Abfrage des Prozessbetriebs ablaufen. Vor dem Prozessbetrieb führt ein Operator oder Ingenieur eine Reihe von Schritten aus, um den erweiterten Steuerblock **38** einschließlich einer integrierten MPC-Steuerung und eines Optimierers zu erstellen. An einem Block **92** kann insbesondere eine erweiterte Steuerungsvorlage zur Verwendung als erweiterter Steuerblock **38** ausgewählt werden. Die Vorlage kann in einer Bibliothek in einem Anwendungsprogramm für eine Konfigurierung auf der Nutzerschnittstelle **13** gespeichert sein oder kopiert werden und kann die allgemeinen mathematischen und logischen Funktionen der MPC-Steerroutine **52** und des Optimierers **54** ohne die einzelnen MPC-, Prozessmodelle und Beharrungszustandsverstärkungs- oder Steuermatrizen und die jeweilige Zielfunktion enthalten. Diese erweiterte Steuerungsvorlage kann in einem Modul untergebracht sein, das andere Blöcke, bspw. Eingabe- und Ausgabeblocke aufweist, die zu einer Kommunikation mit Geräten in dem Prozess **50** eingerichtet sind sowie auch andere Funktionsblocktypen, z. B. Steuerblöcke, einschließlich PID-, neuronale Netzwerk- und Fuzzylogic-Steuerblöcke. Es versteht sich, dass in allen Ausführungsformen die Blöcke in einem Modul jeweils Objekte in einem objektorientierten Programmparadigma sind, die jeweils miteinander verbundene Ein- und Ausgänge zum Ausführen einer Kommunikation zwischen den Blöcken aufweisen. Während des Betriebs führt der die Module abarbeitende Prozessor jeden der Blöcke aufeinander folgend zu unterschiedlichen Zeiten aus und verwendet die Blockeingaben zum Erzeugen der Blockausgaben, um diese, durch die spezifizierten Kommunikationsverbindungen zwischen den Blöcken definiert, an die Eingänge anderer Blöcke bereitzustellen.

[0078] An einem Block **94** definiert der Operator die einzelnen Stellgrößen, Steuergrößen, Hilfsgrößen und Störgrößen für die Verwendung innerhalb des Blocks **38**. Wenn gewünscht, kann in einem Konfigurationsprogramm, z. B. in dem Programm **40** aus **Fig. 1**, der Nutzer die Steuerungsvorlage ansehen, zu benennende und zu konfigurierende Eingänge und Ausgänge auswählen, unter Verwendung jedes üblichen Browsers innerhalb der Konfigurationsumgebung blättern, um die aktuellen Eingaben und Ausgaben innerhalb des Steuersystems aufzufinden und sowohl die aktuellen Steuergrößen als auch die Eingabe- und Ausgabesteuergrößen für die Steuerungsvorlage auszuwählen. **Fig. 5** verdeutlicht eine von einer Konfigurationsroutine erzeugte Bildschirmdarstellung, die ein eine Reihe von miteinander verbundenen Funktionsblöcken aufweisendes Steuermodul DEB_MPC darstellt, das eine Reihe von AI (analogen Eingängen) und AO (analogen Ausgängen)-Funktionsblöcken, eine Reihe von PID-Steuerfunktionsblöcken und einen MPC-PRO-Funktionsblock, der ein erweiterter Funktionsblock ist, enthält. Die Baumstruktur auf der linken Seite der Darstellung aus **Fig. 5** verdeutlicht die Funktionsblöcke innerhalb des DIB_MPC-Moduls einschließlich bspw. Block 1, C4_AI, C4_DGEN usw.

[0079] Es versteht sich, dass der Nutzer die Eingaben zu und die Ausgaben von dem MPC-PRO-Funktionsblock durch ein Ziehen von Linien zwischen diesen Eingängen und Ausgängen und den Eingängen und Ausgängen der anderen Funktionsblöcke festlegt. Alternativ dazu kann der Nutzer den MPC-PRO-Block auswählen, um einen Zugriff auf die Eigenschaften des MPC-PRO-Blocks zu bekommen. Eine bspw. in **Fig. 6** gezeigte Dialogbox kann angezeigt werden, um es dem Nutzer zu ermöglichen, die Eigenschaften des MPC-PRO-Blocks anzusehen. Wie in **Fig. 6** dargestellt ist, kann für jede der Steuergrößen, der Stellgrößen, der Störgrößen und der Randbedingungs-(Hilfs-)Größen eine jeweils eigene Tabelle bereitgestellt werden, um eine Verwaltung dieser Größen zu ermöglichen. Diese ist insbesondere dann notwendig, wenn vielerlei Größen, bspw. 20 oder mehr mit dem erweiterten Steuerblock **38** verknüpft sind. Innerhalb der Tabelle können für eine jeweilige Größe eine Beschreibung, eine untere und obere Grenze (Randbedingung) und ein Pfadname bereitgestellt werden. Zusätzlich dazu kann der Nutzer oder der Operator festlegen, was der Block für den Fall eines fehlerhaften Zustands der Größe ausführen soll, wie z. B. keine Aktion ausführen, Verwenden des simulierten Wertes der Größe anstelle des gemessenen Wertes oder Akzeptieren einer manuellen Eingabe. Weiterhin kann der Operator festlegen, ob diese Größe zum Ausführen einer Optimierung entweder zu minimieren oder

zu maximieren ist, und die Priorität oder die Gewichtung und damit zusammen hängende Gewinnwerte der Größe festlegen. Diese Felder müssen ausgefüllt werden, wenn die voreingestellte Zielfunktion nicht angewendet wird. Natürlich kann der Nutzer unter Verwendung der entsprechenden Tasten auf der rechten Seite der Dialogbox Informationen über die Größen hinzufügen, verschieben, verändern oder löschen.

[0080] Der Nutzer kann die Information für eine oder mehrere Größen durch Auswahl der Größen festlegen oder ändern. In diesem Fall kann dem Nutzer eine Dialogbox wie z. B. eine in **Fig. 7** für die Stellgröße REFLUX FLOW gezeigte Dialogbox, präsentiert werden. Der Nutzer kann die Information in den jeweils unterschiedlichen Boxen ändern und Informationen, z. B. einen Pfadnamen der Größe (d. h. ihre Eingabe- und Ausgabe-Verbindung) mittels eines Durchblätterns festlegen. Unter Verwendung des Bildschirms aus **Fig. 7** kann der Nutzer einen internen oder externen Browserknopf für ein Blättern innerhalb des Moduls oder extern zu dem Modul, in dem der MPC-PRO-Block lokalisiert ist, auswählen. Natürlich kann, falls gewünscht, der Operator oder der Nutzer eine Adresse, einen Pfadnamen, einen Namen zur Markierung, d. h. einen Tag-Namen usw. manuell angeben, welche die Verbindungen zu und von den Eingängen und Ausgängen des erweiterten Steuerblocks definieren.

[0081] Nach Auswahl der Eingänge und Ausgänge des erweiterten Steuerfunktionsblocks kann der Nutzer die mit den Steuergrößen verbundenen Sollwerte, die Hilfsgrößen und die Stellgrößen und die Gewichtungen definieren, die mit jeder der Steuer-, Hilfsgrößen und Stellgrößen verbunden sind. Natürlich ein Teil dieser Information, z. B. Randbedingungsgrenzen oder Bereiche, bereits diesen Größen zugeordnet sein, weil diese Größen innerhalb der Konfigurationsumgebung des Prozesssteuerungssystems ausgewählt oder gefunden worden sind. Wenn gewünscht, kann an einem Block **96** aus **Fig. 3** der Operator die innerhalb des Optimierers zu verwendenden Zielfunktionen bzw. die Menge der Zielfunktionen durch Spezifizieren der Einheiten, Kosten oder Gewinne für jede der Stellgrößen, der Steuergrößen und der Hilfsgrößen konfigurieren. Natürlich kann der Operator an diesem Punkt wählen, die vorgegebene Zielfunktion zu einer wie oben beschriebenen Verwendung zu nutzen. **Fig. 8** ist eine von einer Konfigurationsroutine bereitgestellte Bildschirmdarstellung, die es einem Nutzer oder Operator ermöglicht, eine Auswahl aus einem Satz von Zielfunktionen vorzunehmen, um diese zum Erzeugen eines erweiterten Steuerblocks zu verwenden. Es ist verständlich, dass der Nutzer die wie die in **Fig. 8** gezeigte Bildschirmdarstellung anwenden kann, um aus einem Satz von vorab gespeicherten Zielfunktionen, hier dargestellt als eine Standardzielfunktion und Zielfunktionen 2–5, eine Auswahl zu treffen.

[0082] Nachdem die Eingaben (Steuer-, Hilfs- und Störgrößen) benannt, mit den erweiterten Steuervorgaben verknüpft und an Block **98** in **Fig. 3** die Gewichtungen, Beschränkungen und damit verbundene Sollwerte angefügt wurden, wird die erweiterte Steuerungsvorlage in eine ausgewählte Steuerung innerhalb eines Prozesses als ein für die Steuerung zu verwendender Funktionsblock geladen. Der allgemeine Aufbau des Steuerblocks und die Art und Weise einer Konfiguration dieses Steuerblocks ist in dem US-Patent Nummer 6,445, 963 mit dem Titel „Integrated Advanced Control Blocks in Prozess Control Systems“ beschrieben, das von dem Anmelder hier beschriebenen Erfindung angemeldet wurde und hier ausdrücklich als Referenz einbezogen wird. Während dieses Patent die Art und Weise einer Erzeugung einer MPC-Steuerung in einem Prozesssteuerungssystem beschreibt und nicht das Vorgehen behandelt, mit dem ein Optimierer mit dieser Steuerung verbunden werden kann, ist es selbstverständlich, dass die allgemeinen zum Verbinden und Konfigurieren der Steuerung ausgeführten Schritte für den hier beschriebenen Steuerblock **38** mit der Vorlage einschließlich aller hier behandelten logischen Elemente für den Steuerblock **38** anstelle der hier beschriebenen verwendet werden können.

[0083] In jedem Fall kann nun, nachdem die erweiterte Steuerungsvorlage in die Steuerung geladen wurde, der Operator an einem Block **100** die Entscheidung zum Ausführen einer Testphase der Steuerungsvorlage treffen, um die Sprungantwortmatrix und das innerhalb des MPC-Steuerungsalgorithmus zu verwendende Prozessmodell zu erzeugen. Wie in dem oben genannten Patent beschrieben, erzeugt die Steuerungslogik innerhalb des erweiterten Steuerblocks **38** an den Prozess während der Testphase eine Folge von Pseudozufalls-kurven als Stellgrößen und beobachtet die Veränderungen in den Steuer- und Hilfsgrößen (die durch die MPC-Steuerung hauptsächlich als Steuergrößen verhandelt werden). Wenn gewünscht, können die Stellgrößen und Störgrößen, aber auch die Steuer- und Hilfsgrößen durch den Historian **12** aus **Fig. 1** gesammelt werden, und der Operator kann das Konfigurationsprogramm **40** (**Fig. 1**) aktivieren, um diese Daten von dem Historian **12** zu erhalten und auf diese Art eine Hochrechnung über diese Daten auszuführen, um die Matrix der Sprungantworten zu erhalten oder abzuleiten. Jede Sprungantwort bezeichnet die zeitabhängige Antwort einer der Steuer- und Hilfsgrößen auf eine Einheitsveränderung in einer (und nur einer) der Stellgrößen und Steuergrößen. Diese Einheitsveränderung ist im Allgemeinen eine Sprungveränderung, kann aber auch von einem anderen Veränderungstyp, z. B. ein Impuls oder eine kontinuierliche Veränderung mit einem Anstieg sein. Falls gewünscht, kann andererseits der Steuerblock **38** die Sprungantwortmatrix in Antwort auf die ge-

sammelten Daten dann erzeugen, wenn die pseudozufälligen kurven an den Prozess **50** angelegt werden. Er kann anschließend diese kurven an die durch den Operator oder Nutzer genutzte Operatorschnittstelle **30** bereitstellen und dabei den erweiterten Steuerblock **38** installieren.

[0084] Fig. 9 erläutert die durch die Testroutine bereitgestellte Bildschirmanzeige, um dem Operator mit graphischen Darstellungen der gesammelten und hochgerechneten Daten zu versorgen. Diese ermöglichen es dem Operator, die Erzeugung der Sprungantwortkurve und damit das Prozessmodell oder der in der MPC-Steuerung verwendeten Steuermatrix des erweiterten Steuerblocks zu überwachen. Eine Plotregion **101** gibt insbesondere die Daten für eine Reihe von Eingaben oder Ausgaben oder andere (z. B. vorab durch den Operator festgelegte) Daten in Antwort auf die Test-kurven aus. Eine Balkengraphenregion **102** stellt einen Balkengraphen für jede der behandelten Datengrößen bereit, der für jede der hochgerechneten Größen den Namen der Größe, den aktuellen Wert der Größe in einer graphischen Balkenform, falls verfügbar, einen Sollwert (angezeigt durch ein größeres Dreieck über dem Balkengraphen) und, falls verfügbar, Grenzen (angezeigt durch ein kleineres Dreieck über dem Balkengraphen) anzeigt. Andere Bereiche der Anzeige zeigen andere Angaben über den erweiterten Steuerblock an, beispielsweise das Ziel und den aktuellen Modus des Blocks **104** und die eingestellte Zeit bis zum Beharrungszustand **106**. Dem Erzeugen eines Prozessmodells für den erweiterten Steuerblock vorausgehend, kann der Operator graphisch die für die Hochrechnungs-Plots **101** zu verwendenden Daten festlegen. Insbesondere kann der Operator die Beginn- und Endpunkte **108** und **110** des Plots **102** festlegen, sofern die Daten zum Erstellen der Sprungantwort zu verwenden sind. Die Daten in diesem Bereich können mit einer jeweils verschiedenen Farbe eingefärbt werden, z. B. in Grün, um die ausgewählten Daten visuell auszuzeichnen. Weiterhin kann der Operator Abschnitte innerhalb dieses eingefärbten Gebietes als auszuschließende Abschnitte (und als somit als nicht repräsentativ, als einen Rauscheffekt oder eine unerwünschte Störung usw.) festlegen. Dieses Gebiet ist zwischen den Linien **112** und **114** dargestellt und kann beispielsweise in Rot gefärbt sein, um darauf hinzuweisen, dass diese Daten nicht in die Erzeugung der Sprungantwort einzubeziehen sind. Natürlich kann der Nutzer jede gewünschte Daten ein- oder ausschließen und diese Funktion für jede der (Fig. 9 zeigt, dass in diesem Fall acht Hochrechnungsplots verfügbar sind) mit verschiedenen, z. B. einer Reihe von Stellgrößen, Steuergrößen, Hilfsgrößen usw. verknüpften Hochrechnungsplots ausführen.

[0085] Um einen Satz von Sprungantworten zu erzeugen, kann der Operator die Modellgenerierungstaste **116** auf der Bildschirmdarstellung von Fig. 9 auswählen. Die Erzeugungsroutine wird dann die ausgewählten Daten aus den Hochrechnungsplots zum Erzeugen eines Satzes von Sprungantworten verwenden, wobei jede Sprungantwort auf die Antwort einer der Steuer- und Hilfsgrößen auf eine der Stellgrößen oder Störgrößen verweist. Dieser Erzeugungsprozess ist bekannt und wird aus diesem Grund an dieser Stelle nicht im Detail dargestellt.

[0086] Nachdem die Sprungantwortmatrix für den Fall erzeugt wurde, bei welchem die Steuer- und Hilfsgrößen die Stellgrößen in ihrer Anzahl übertreffen, wird die Sprungantwortmatrix dazu verwendet, um den Teilsatz der Steuer- und Hilfsgrößen auszuwählen, die in dem MPC-Algorithmus verwendet werden sollen, bei dem die M mal M-Prozessmodell- oder Steuermatrix invertiert und innerhalb der MPC-Steuerung **52** verwendet wird. Dieser Auswahlprozess kann manuell durch den Operator oder automatisch durch eine z. B. innerhalb der Nutzerschnittstelle **13** abgelegte Routine erfolgen, die einen Zugriff auf die Sprungantwortmatrix hat. Generell wird eine der Steuer- und Hilfsgrößen als zu einer der Stellgrößen am nächsten in Beziehung stehende Größe gekennzeichnet. Somit wird eine einzelne und eindeutige (d. h. von allen anderen verschiedene) Steuer- und Hilfsgröße (die Eingaben an die Prozesssteuerung sind) mit jeder der unterschiedlichen Stellgrößen verknüpft (die die Ausgaben der Prozesssteuerung bilden), sodass der MPC-Algorithmus auf einem Prozessmodell aufgebaut werden kann, das von dem M mal M-Satz der Sprungantworten erzeugt wurde.

[0087] In einer Ausführungsform, die einen heuristischen Ansatz bei der Bereitstellung einer Paarung verwendet, wählt die automatische Routine oder der Operator den Satz von M (wobei M gleich der Anzahl der Stellgrößen ist) Steuer- und Hilfsgrößen mit dem Ziel aus, diejenigen Steuer- oder Hilfsgrößen auszuwählen, die eine gewisse Verknüpfung mit der größten Verstärkung und schnellsten Antwortzeit auf eine Einheitsänderung bezüglich einer der Stellgrößen haben und paart diese beiden Größen. Natürlich wird in einigen Fällen eine bestimmte Steuer- oder Hilfsgröße eine große Verstärkung und eine schnelle Antwortzeit mit mehreren Stellgrößen aufweisen. In diesem Fall kann diese Steuer- oder Hilfsgröße mit jeder der damit verknüpften Stellgrößen gepaart werden. Sie kann sogar mit einer Stellgröße gepaart werden, die nicht die größte Verstärkung und schnellste Antwortzeit erzeugt, weil im Ganzen diejenige Stellgröße, die die kleinere Verstärkung hat und die langsamere Antwortzeit verursacht, keine andere Steuer- oder Hilfsgröße bis zu einem zulässigen Grad beeinflusst. Somit werden die Paare der Stellgrößen auf der einen Seite und der Steuer- und Hilfsgrößen auf der anderen Seite so ausgewählt, um im Großen und Ganzen die Stellgrößen mit dem Teilsatz der Steuer-

und Hilfsgrößen zu paaren, die am meisten beeinflussbaren Steuergrößen bezüglich der Stellgrößen sind. Weiterhin macht es nichts aus, wenn nicht alle Steuergrößen aus dem Teilsatz der M-Steuer- und Hilfsgrößen ausgewählt wurden und damit die MPC-Steuerung nicht alle Steuergrößen als ihre Eingaben empfängt, weil der Satz der Steuer- und Hilfsgrößenziele durch den Optimierer zum Bezeichnen eines Betriebspunktes des Prozesses ausgewählt wurde, an dem die nicht ausgewählten Steuer-(und ebenso die nicht ausgewählten Hilfs-)Größen an ihrem Sollwert oder innerhalb eines vorbestimmten Betriebsbereiches sind.

[0088] Weil natürlich einige zehn oder hunderte Steuer- und Hilfsgrößen auf der einen Seite und einige zehn oder hunderte Stellgrößen auf der anderen Seite vorhanden sein können, kann es, nicht zuletzt auch in Hinblick auf die Bedürfnisse der Visualisierung, schwierig sein, einen Satz von Steuergrößen und Hilfsgrößen auszuwählen, der die beste Antwort auf jede der verschiedenen Stellgrößen hat. Um dieses Problem zu überwinden, kann die erweiterte Steuerblockerzeugungsroutine **40** innerhalb der Operatorschnittstelle **30** einen Satz von Bildschirmdarstellungen für den Nutzer Operator enthalten, um dem Operator zu helfen oder es diesem zu ermöglichen, geeignete Auswahlen aus denjenigen Steuer- und Hilfsgrößen zu machen, die als ein Teilsatz der in der MPC-Steuerung **52** während des Betriebs zu verwendenden Steuer- und Hilfsgrößen genutzt werden soll.

[0089] Somit wird bei einem in **Fig. 3** aufgeführten Block **120** dem Operator ein Bildschirm präsentiert, auf dem dieser die Antwort jeder der Steuer- und Hilfsgrößen auf eine einzelne oder ausgewählte Stellgröße ansehen kann. Ein derartiger Bildschirm ist in **Fig. 10** dargestellt und zeigt die Antwort jeder aus einer Anzahl von Steuer- und (als Randbedingung benannten) Hilfsgrößen auf eine hier als TOP_DRAW bezeichnete Stellgröße. Der Operator kann durch die Stellgrößen nacheinander hindurch scrollen, die Sprungantwort auf jede der Steuer- und Hilfsgrößen zu jeder der verschiedenen Stellgrößen anschauen und während des Prozesses die eine Steuer- oder Hilfsgröße auswählen, die am besten auf die Stellgröße anspricht. Typischerweise wird dann der Operator versuchen, diejenige Steuer- und/oder Stellgröße auszuwählen, die die beste Kombination aus der höchsten Beharrungszustandsverstärkung und der kürzesten Antwortzeit auf die Stellgröße aufweist. Wie in **Fig. 11** gezeigt ist, kann eine der Steuer- und Hilfsgrößen als die für diese Stellgröße am meisten signifikante Größe unter Verwendung einer Dialogbox ausgewählt werden. Wie in **Fig. 11** gezeigt, kann falls gewünscht, die aus den Steuer- und Hilfsgrößen ausgewählte Größe in einer anderen Farbe hervorgehoben werden, beispielsweise in Rot, während vorhergehend ausgewählte (d. h. Steuer- und Hilfsgrößen, die für andere Stellgrößen ausgewählt wurden) in einer davon abweichenden Farbe, wie z. B. in Gelb eingefärbt werden. In dieser Ausführungsform kann die Steueroutine **40**, die natürlich die vorausgehend gewählten Steuer- und Hilfsgrößen in einem Speicher speichert, einen Test ausführen, um sicherzustellen, dass der Operator nicht die gleiche Steuergröße oder Stellgröße als zwei verschiedenen Stellgrößen zugeordnete Größe auswählt. Wenn der Nutzer oder Operator eine Steuer- oder Hilfsgröße auswählt, die schon für eine andere Stellgröße gewählt wurde, kann die Routine **40** eine Fehlermeldung an den Nutzer oder Operator ausgeben, die den Nutzer oder Operator über die Auswahl der vorhergehend ausgewählten Steuer- und Hilfsgröße informiert. Auf diese Weise vermeidet die Routine **40** die Auswahl der selben Steuer- und Hilfsgrößen für zwei oder mehrere verschiedene Stellgrößen.

[0090] Wie in **Fig. 12** dargestellt, kann der Operator oder der Nutzer ebenfalls die Wahl treffen, sich die verschiedenen Sprungantworten für jede der verschiedenen Stellgrößen und Störgrößen anzusehen. **Fig. 11** zeigt die Sprungantwort von TOP_END_POINT auf jede der Stellgrößen und Störgrößen, die vorab für den erzeugten erweiterten Steuerblock festgelegt wurden. Natürlich kann der Operator die Anzeige von **Fig. 12** verwenden, um eine der Stellgrößen als mit der Steuergröße TOP_END_POINT verknüpfte Stellgröße auszuwählen.

[0091] Es versteht sich, dass es die Bildschirmanzeigen der **Fig. 10** bis **Fig. 12** dem Operator ermöglichen, die Untermenge der als Eingaben an den MPC-Steueralgorithmus verwendeten M Steuer- und Hilfsgrößen zu visualisieren und auszuwählen. Dies ist besonders dann nützlich, wenn eine Vielzahl dieser Größen vorhanden ist. Natürlich kann der Satz der in dem Block **74** festgelegten Steuer- und Randbedingungs-Größen automatisch oder elektronisch auf der Grundlage einiger vorab aufgestellter Kriterien oder einer Selektionsroutine ausgewählt werden. Diese wählt die zu verwendenden Eingabegrößen auf der Grundlage einer Kombination von aus den Sprungantworten der gesteuerten Randbedingungs-Größen und Steuergrößen festgestellten Verstärkungsantworten und Zeitverzögerungen aus.

[0092] In einer anderen Ausführungsform kann ein automatischer Auswahlprozess zunächst eine Steuermatrix durch Auswahl einer Eingabe/Ausgabe-Matrix auf der Grundlage der Konditions- oder Bedingungsanzahl der Matrix, d. h. durch ein Minimieren der Bedingungsanzahl auf einen gewissen Betrag und durch Erstellen einer Steuerungskonfiguration aus der Steuermatrix definieren.

[0093] In diesem Beispiel kann für eine Prozessverstärkungsmatrix A die Bedingungsanzahl der Matrix $A^T A$ festgestellt werden, um die Steuerbarkeit der Matrix zu testen. Eine kleinere Bedingungsanzahl bedeutet im Allgemeinen eine bessere Steuerbarkeit, während eine höhere Bedingungsanzahl eine geringere Steuerbarkeit anzeigt und mehr Steuersprünge oder Bewegungen während des dynamischen Steuerbetriebs erfordert. Es gibt keine strengen Kriterien zum Festlegen eines vertretbaren Grades einer Steuerbarkeit und somit kann die Bedingungsanzahl als eine relative Vergleichsmöglichkeit zwischen verschiedenen möglichen Steuermatrizen und als ein Test für schlecht eingestellte Matrizen verwendet werden. Wie bekannt ist, erreicht die Bedingungsanzahl für eine schlecht eingestellte Matrix einen unendlichen Wert. Mathematisch bedeutet dies, dass eine schlechte Einstellung in dem Fall co-linearer Prozessgrößen auftritt – d. h. auf Grund von co-linearen, insbesondere linear abhängigen, Zeilen oder Spalten in der Steuermatrix. Ein Hauptfaktor, der die Bedingungsanzahl und die Steuerbarkeit beeinflusst, ist somit eine Kreuzkorrelation zwischen Zeilen und Spalten in der Matrix. Eine sorgfältige Auswahl der Eingabe-Ausgabegrößen und der Steuermatrix kann die Konditionierungsprobleme reduzieren. Praktisch sollten dann Bedenken bestehen, wenn die Bedingungsanzahl einer Steuermatrix bei einem Wert in der Größenordnung von einigen hundert (zum Beispiel 500) oder höher liegt. Mit einer derartigen Matrix sind steuerungsgeführte Größenbewegungen in hohem Maße aufwendig.

[0094] Wie oben dargestellt, löst die Steuermatrix das dynamische Steuerproblem, während der LP-Optimierer das Beharrungszustands-Optimierungsproblem löst und die Steuermatrix eine quadratische Eingabe-Ausgabematrix sogar in dem Fall sein muss, wenn der MPC-Steuerungsblock eine ungleiche Anzahl von MV (einschließlich AV) und CV hat. Zu Beginn der Auswahl der Eingaben und Ausgaben für die zur Erzeugung der Steuerung zu nutzenden Steuermatrix werden alle verfügbaren MVs typischerweise als Ausgaben der Steuerung zusammengefasst oder ausgewählt. Nach dem Auswählen der Ausgaben (die MV) müssen die Prozessausgabegrößen (das heißt die CV und AV), die zu einem Bestandteil der dynamischen Steuermatrix gemacht wurden, so ausgewählt werden, dass eine quadratische Steuermatrix erzeugt wird, die nicht schlecht eingestellt ist.

[0095] Ein Verfahren einer automatischen oder manuellen Auswahl der CV und AV als Eingaben innerhalb der Steuermatrix wird nun dargestellt, wobei es sich versteht, dass andere Verfahren ebenso angewendet werden können.

[0096] Schritt 1 – die CVs werden ausgewählt bis, falls möglich, die Anzahl der CVs gleich der Anzahl der MVs (das heißt der Anzahl der Ausgaben Steuerung) ist. In dem Fall, bei dem mehr CVs als MVs vorhanden sind, können die CVs in jeder Reihenfolge auf der Basis jedes gewünschten Kriteriums, beispielsweise der Priorität, der Verstärkungs- oder Phasenantworten, von Nutzereingaben usw. gewählt werden. Wenn die mögliche Gesamtzahl der CVs gleich der Anzahl der MVs ist, dann laufe zu Schritt 4 und überprüfe, ob die sich daraus ergebende quadratische Bedingungsanzahl der Steuermatrix vertretbar ist. Wenn die Anzahl der CVs kleiner als die Anzahl der MVs ist, können AVs wie in Schritt 2 beschrieben gewählt werden. Sind keine CVs definiert, wähle die AVs mit einer maximalen Verstärkungsbeziehung zu einem MVs aus und gehe zu Schritt 2.

[0097] Schritt 2 – errechne nacheinander die Bedingungsanzahl für jede mögliche AV, die zu einer bereits ausgewählten, durch die vorhergehend gewählten CVs und AVs definierten Steuermatrix hinzugezählt wurde. Es versteht sich, dass die durch die ausgewählten CV's definierte Matrix eine Zeile für jedes ausgewählte CV und AV enthält, die die Beharrungszustandsverstärkung für diese CV oder AV bezüglich der ausgewählten MV's definieren.

[0098] Schritt 3 – wähle die in Schritt 2 festgelegten AVs aus, die zu einer minimalen Bedingungsanzahl für die sich ergebende Matrix führen und definiere die Matrix als die für das Hinzufügen der ausgewählten AV vorab gegebene Matrix. Wenn die Anzahl der MV an diesem Punkt gleich der Anzahl der ausgewählten CV plus der Anzahl der ausgewählten AVs ist (d. h., wenn die Matrix nun quadratisch ist) gehe zu Schritt 4. Wenn nicht, laufe zu Schritt 2 zurück.

[0099] Schritt 4 – Errechne die Bedingungsanzahl für die erzeugte quadratische Steuermatrix A_c . Wenn gewünscht, kann die Berechnung der Bedingungsanzahl für die Matrix A_c anstelle der Matrix $A_c^T A_c$ verwendet werden, weil sich die Bedingungsanzahlen für diese verschiedenen Matrizen über die Quadratwurzel der jeweils anderen beziehen.

[0100] Schritt 5 – Wenn die in Schritt 4 berechnete Bedingungsanzahl vertretbar ist, ordne jedes CV und ausgewähltes AV einem MV durch eine Auswahl des CV oder AV zu, das eine Maximumverstärkung in Bezug auf ein spezifisches MV hat, bis die Paarung vollständig ist. An diesem Punkt ist der Auswahlprozess abgeschlossen.

Wenn andererseits die Bedingungsanzahl größer als die minimal vertretbare Bedingungsanzahl ist, entferne das letzte zur Steuermatrix hinzugefügte AV/CV und führe die Wrap-Around-Prozedur von Schritt 6 aus.

[0101] Schritt 6 – Führe eine als Wrap-Around-Prozedur bezeichnete Ersetzungsprozedur für eine der ausgewählten MV's nach der anderen aus und berechne die Bedingungsanzahl der Matrix, die sich aus jedem Wrap-Around ergibt. Im Wesentlichen wird eine Schleifenprozedur durch ein der Reihe nach erfolgreiches, wechseln-des Anordnen einer Antwort in Form einer Eins, d. h. eine Eins-Antwort für jede der verschiedenen MVs anstelle der entfernten AV (oder CV) ausgeführt. Die Eins-Antwort ist jeweils eine Eins an einer Stelle in der Zeile der Matrix und Null überall sonst. In diesem Fall wird im Wesentlichen jedes der einzelnen MVs als eine Eingabe und eine Ausgabe anstatt der AV benutzt, um eine gut eingestellte quadratische Steuermatrix zu bilden. Als Beispiel werden für eine 4×4-Matrix die Kombinationen 1000, 0100, 0010 und 0001 in die Zeile der entfernten AV-Aufstellung in die Verstärkungsmatrix A_c eingefügt.

[0102] Schritt 7 – Nach Ausführen der Wrap-Around-Prozedur für jede der MVs, wähle die die zu einer minimalen Bedingungsanzahl führende Kombination aus. Tritt keine Verbesserung ein, behalte die ursprüngliche Matrix bei. Ordne dann jedes ausgewählte CV und jedes ausgewählte AV einem MV durch Auswählen des CV oder AV mit einer maximalen Verstärkung bezüglich eines spezifischen MV zu, aber unter Ausschluss der MV, die für die Steuerung selbst verwendet wird (das heißt der MV, die in der Wrap-Around-Prozedur nach Schritt 6 ersetzt wurde).

[0103] Natürlich kann die durch diese Prozedur definiert Steuermatrix wie auch die sich ergebende Bedingungsanzahl an den Nutzer übermittelt werden und der Nutzer kann die definierte Steuerungsmatrix für eine Verwendung in der Steuerung zulassen oder zurückweisen.

[0104] Es ist darauf hinzuweisen, dass in der oben beschriebenen automatischen Prozedur nur höchstens ein MV für die Selbststeuerung zum Zwecke einer Verbesserung der Steuerbarkeit ausgewählt (d. h., mittels der Wrap-Around-Prozedur eingesetzt) wurde. In einer manuellen Prozedur kann die Anzahl der Wrap-Around-MVs willkürlich sein. Die für die Selbststeuerung ausgewählten MVs stechen durch eine nicht vorhandene mit ihnen verknüpften Auswahl einer Ausgabengröße in der Steuerungskonfiguration hervor. Ebenso kann man mehrere MVs als Wrap-Around-Größen zur Steuerung verwenden, wenn die Anzahl der MVs größer als die Anzahl der gesamten CVs plus die der AVs ist. Auf diese Weise wird schließlich immer eine quadratische Steuerungsmatrix für die Steuerung bereitgestellt, die jede der MVs als Ausgaben hat. Es versteht sich, dass der Prozess des Ausführens und des Anwendens der Wrap-Around-Prozedur bedeutet, dass die Anzahl der für die Steuermatrix ausgewählten CVs und AVs kleiner als die Anzahl der die durch die Steuerung gesteuerten MVs sein kann, wobei deren Differenz die Anzahl der Wrap-Around-MVs als Eingänge für die Steuermatrix ist. Ferner kann diese Wrap-Around-Prozedur in einem Prozesssteuerungssystem verwendet werden, das weniger CVs plus AVs als MVs aufweist.

[0105] Natürlich wurde vorhergehend die Bedingungsanzahl unter Verwendung der Beharrungszustandsverstärkung berechnet und damit definiert die Steuerungsmatrix im Wesentlichen eine Steuerbarkeit für den Beharrungszustand. Die Prozessdynamik (Totzeiten, Verzögerungen usw.) und die Modellunsicherheiten haben ebenfalls einen Einfluss auf die dynamische Steuerbarkeit, und diese Effekte können durch eine Veränderung der Priorität der Prozessgrößen (d. h. der Steuer- und Hilfsgrößen) in Betracht gezogen werden. Diese können ihre Einbeziehung in die Steuermatrix auf Grund der durch sie auf die dynamische Steuerung ausgeübten Effekte vorschreiben.

[0106] Es ist ebenfalls möglich, weitere heuristische Prozeduren zu verwenden, die zur Verbesserung sowohl des Beharrungszustandes als auch der dynamischen Steuerbarkeit vorgesehen sind. Eine derartige Prozedur weist typischerweise eine Menge heuristischer Kriterien auf, von denen einige sich möglicherweise einander widersprechen, und die in verschiedenen Phasen der Entwicklung an eine Steuermatrix angelegt werden und dadurch zur Auswahl eines geeigneten Satzes von Steuerungseingaben führen, der gewisse Verbesserungen der Steuermatrix bewirkt. In einer solchen heuristischen Prozedur werden die CVs und die AVs durch eine MV auf der Basis der höchsten Verstärkungsbeziehung gruppiert. Dann wird für jede MV-Gruppierung diejenige Prozessausgabe mit der schnellsten Dynamik und der signifikanten Verstärkung ausgewählt. Dieser Auswahlprozess kann ein Toleranzintervall berücksichtigen und Präferenzen an die CVs über die AVs (die ansonsten alle gleichwertig sind) vergeben. Die Prozessmodellerzeugungsroutine wird dann die aus jeder Gruppe während der MPC-Steuerungserzeugung ausgewählten Parameter anwenden. Weil nur ein Parameter für jedes MV ausgewählt wurde, ist die Antwortmatrix quadratisch und kann invertiert werden.

[0107] In jedem Fall generiert ein Block **124** aus **Fig. 3** nach Auswahl des Teilsatzes von M (oder weniger) Steuer- und Hilfsgrößen-Eingaben für die MPC-Steuerung das in dem MPC-Steueralgorithmus **86** aus **Fig. 2** zu verwendende Prozessmodell oder die Steuerung aus der festgelegten quadratischen Steuermatrix. Wie bekannt, ist dieser Steuerungserzeugungsschritt eine rechenintensive Prozedur. Ein Block **126** lädt dann dieses MPC-Prozessmodell (das die Steuermatrix von vornherein enthält) oder die Steuerung und, falls gebraucht, die Sprungantworten und die Beharrungszustandsprungantwortverstärkungsmatrix in den Steuerblock **38**. Diese Daten werden in den Steuerblock **38** für den Betrieb eingebaut. Dann ist der Steuerblock **38** für einen Online-Betrieb innerhalb des Prozesses **50** bereit.

[0108] Wenn gewünscht, können die Prozesssprungantworten auf eine Weise rekonfiguriert oder bereitgestellt werden, sich von der Weise der Erzeugung dieser Sprungantworten unterscheidet. So kann bspw. eine der Sprungantworten aus verschiedenen Modellen kopiert und auf den Bildschirmen bspw. der **Fig. 10** bis **Fig. 12** bereitgestellt werden, um die Sprungantwort einer bestimmten Steuerungs- oder Hilfsgröße auf eine Stellgröße oder Störgröße festzulegen. **Fig. 13** zeigt einen Bildschirmanzeige, in der der Nutzer eine der Sprungantworten eines bestimmten Prozesses oder Modells auswählen und kopieren und dann die gleiche Antwort mittels eines Drag-and-Drop-Vorgangs in ein sich davon unterscheidendes Modell verschieben kann und somit diese Sprungantwort in das neue Modell einfügt. Es wird somit dem Nutzer ermöglicht, auf manuelle Weise ein Sprungantwortmodell festzulegen. Natürlich kann als Teil dieses Ablaufes der Nutzer eine oder mehrere der die wie oben beschrieben automatisch generierten Sprungantwortmodelle löschen.

[0109] **Fig. 14** zeigt eine Bildschirmanzeige, in welcher der Nutzer eine besondere Sprungantwort ansehen kann (hier die Sprungantwort auf TOP_END_POINT vs. TOP_DRAW). Die Parameter für diesen Sprungantwort, wie z. B. die Beharrungszustandsverstärkung, die Antwortzeit, die Zeitkonstante erster Ordnung und der quadratische Fehler, werden auf der Anzeige in einfacher, einen Bezug leicht herstellenden Weise dem Nutzer oder Operator angezeigt. Wenn gewünscht, kann der Nutzer die Eigenschaften dieser Sprungantwort einsehen und durch ein Festlegen einer Reihe von Parametern z. B. einer unterschiedlichen Verstärkung oder Zeitkonstante, falls erforderlich, verändern. Wenn der Nutzer eine andere Verstärkung oder einen anderen Parameter festlegt, kann das Sprungantwortmodell mathematisch regeneriert werden, um diese neuen Parameter oder den Satz der Parameter zu enthalten. Diese Vorgehensweise ist nützlich, wenn der Nutzer die Parameter der Sprungantwort kennt und die erzeugte Sprungantwort ändern muss, um dieser Parameter anzupassen oder zu befriedigen.

[0110] Unter Bezugnahme auf **Fig. 4** werden nun die allgemeinen Schritte erläutert, die während jedes Betriebszyklus oder einer Abfrage des unter Verwendung des Flussdiagramms **90a** aus **Fig. 3** erzeugten erweiterten Steuerblocks **38** ausgeführt werden, wenn der Prozess **50** online betrieben wird. An einem Block **150** empfängt die MPC-Steuerung **52** (**Fig. 2**) die gemessenen Werte der Steuer- und Hilfsgrößen CV und AV und verarbeitet diese. Insbesondere verarbeitet das Steuerungsvorhersage-Prozessmodell die CV-, AV- und DV-Messungen oder Eingaben, um sowohl den Zukunfts-Steuerparametervektor als auch die vorausgesagten Beharrungszustandssteuer- und Hilfsgrößen CV_{SS} und AV_{SS} zu erzeugen.

[0111] Als Nächstes verarbeitet oder filtert an einem Block **152** der Eingabe-Verarbeitungs/Filterblock **58** (**Fig. 2**) die vorausgesagten Steuer- und Hilfsgrößen und Stellgrößen CV_{SS} , AV_{SS} und MV_{SS} , die durch die MPC-Steuerung **52** erstellt wurden, und stellt diese gefilterten Werte an den Optimierer **54** bereit. An einem Block **154** führt der Optimierer **54** übliche LP-Verfahren aus, um den Satz von M Stellgrößenzielen MV_T festzulegen, die die gewählte oder voreingestellte Zielfunktion maximieren oder minimieren und dabei keine der Beschränkungen der Hilfsgrößen und Stellgrößen verletzt, während die Steuergrößen an ihrem festgelegten Sollwert oder innerhalb der festgelegten Bereiche für diese Größen verbleiben. Allgemein gesagt errechnet der Optimierer **54** durch ein Ansteuern jeder der Steuergrößen und der Hilfsgrößen an ihre Grenzen eine Lösung für die Zielstellgröße MV_T . Wie oben gezeigt, wird in vielen Fällen eine Lösung existieren, bei der sich jede der Steuergrößen an ihrem Sollwert befinden (die anfänglich als eine obere Grenze der Steuergröße behandelt werden können), während jede der Hilfsgrößen in ihren jeweiligen Randbedingungsgrenzen verbleibt. Wenn dies der Fall ist, muss der Optimierer **54** nur die festgestellten Stellgrößenziele MV_T ausgeben, die ein optimales Resultat für die Zielfunktion erzeugen.

[0112] In einigen Fällen kann es allerdings aufgrund von engen Randbedingungen bei einigen oder allen der Hilfsgrößen oder Stellgrößen unmöglich sein, einen Betriebspunkt zu finden, an dem sich alle der Steuergrößen an ihrem Sollwert befinden und alle der Hilfsgrößen innerhalb ihrer jeweiligen Randbedingungsgrenzen sind, weil es eine solche Lösung nicht gibt. In diesen Fällen kann es, wie oben bemerkt, der Optimierer **54** den Steuergrößen ermöglichen, sich innerhalb ihrer festgelegten Sollwertbereiche mit dem Ziel frei zu bewegen, um damit einen Betriebspunkt zu finden, bei dem die Hilfsgrößen innerhalb ihrer jeweiligen Randbedingun-

gen betrieben werden. Wenn in diesem Fall keine Lösung existiert, kann der Optimierer eine der Randbedingungsgrenzen der Hilfsgrößen als Grenze der Lösung verwerfen und stattdessen den optimalen Prozessbetriebspunkt bei einem Ignorieren der Randbedingungsgrenzen der verworfenen Hilfsgröße festlegen. Hierzu wählt der Optimierer aus, welche Hilfs- oder Steuergröße als eine Randbedingungsgrenze auf der Grundlage ihrer jeweiligen für jede der Steuer- und Hilfsgrößen bereitgestellten Gewichtungen zu verwerfen ist (wobei z. B. das niedrigsten Gewicht oder der höchsten Priorität als erstes verworfen wird). Der Optimierer **54** fährt fort, Hilfs- oder Steuergrößen auf der Basis ihrer bereitgestellten Gewichtungen oder Prioritäten zu verwerfen, bis dieser eine Zielstellgröße MV_T -Lösung findet, bei der alle der Sollwertbereiche für die Steuergrößen und die Grenzen für die verbleibenden Hilfsgrößen mit höherer Priorität befriedigt werden.

[0113] Als Nächstes wendet an einem Block **156** der Zielkonversionsblock **55** (Fig. 2) die Beharrungszustands-Sprungantwort-Verstärkungsmatrix an, um die Zielwerte der Steuer- und Hilfsgrößen CV_T und AV_T aus den Zielwerten der Stellgrößen MV_T zu bestimmen und stellt die ausgewählten N Werte (wobei N gleich oder kleiner als M ist) des Teilsatzes dieser Werte an die MPC-Steuerung **52** als Zieleingaben bereit. An Block **158** nutzt die MPC-Steuerung **52** die Steuermatrix oder die von dieser abgeleitete Logik, um eine wie oben beschriebene bedingungsfreie MPC-Steuerung zu betreiben, um die zukünftigen CV - und AV -Vektoren für diese Zielwerte zu bestimmen, und führt eine Vektorsubtraktion mit den zukünftigen Steuerungsparametervektor aus, um den Zukunftsfehlervektor zu ermitteln. Der MPC-Algorithmus arbeitet in einer bekannten Weise, um die Beharrungszustands-Stellgrößen MV_{SS} auf der Grundlage des von den M mal M -Sprungantworten erstellten Prozessmodells festzulegen und stellt diese MV_{SS} -Werte an den Eingabe-Verarbeitungs/Filterblock **58** (Fig. 2) bereit. An Block **160** legt der MPC-Algorithmus auch die MV -Sprünge fest, die an den Prozess **50** auszugeben sind und gibt die ersten dieser Schritte an den Prozess **50** in jeder geeigneten Weise aus.

[0114] Während des Betriebs können eine oder mehrere Überwachungsanwendungen, die z. B. in einer der Nutzerschnittstellen **13** ablaufen, zu einer Information aus dem erweiterten Steuerblock oder anderen mit diesem kommunikativ verbundenen Funktionsblöcken entweder direkt oder über den Historian **12** beitragen und eine oder mehrere Ansichten oder Diagnostikanzeigen an den Nutzer oder Operator zum Ansehen des Betriebszustandes des erweiterten Steuerblocks ausgeben. Das Funktionsblockverfahren verwendet sowohl Kaskadeneingaben (CAS_IN) und entfernte Kaskadeneingaben ($RCAS_IN$) als auch korrespondierende Rückkalkulationsausgaben ($BKCAL_OUT$ und $RCAS_OUT$) sowohl an den Steuer- als auch an den Ausgabefunktionsblöcken. Es ist möglich, unter Verwendung dieser Verbinder eine supervisor-optimierte MPC-Steuerungsstrategie auf die gegebene Steuerungsstrategie aufzusetzen. Diese Supervisor-Steuerungsstrategie kann unter Verwendung von einem oder mehreren Sichtbildschirmen oder Anzeigen verfolgt werden. Wenn gewünscht, können weiterhin Ziele für die optimierte MPC-Steuerung ebenso aus einer Strategie heraus modifiziert werden.

[0115] Fig. 15 ist eine beispielhafte Bildschirmanzeige, die durch eine oder mehrere derartige Anzeigeanwendungen erzeugt wird, und die einen Dialogbildschirm für einen Optimierer darstellt, der den Betrieb des erweiterten Steuerblocks betreffende Information während dessen Betriebs an den Operator bereitstellt. Insbesondere sind die Eingaben an den Prozess (die Stellgrößen MV) und die Ausgaben (die Steuer- und Hilfsgrößen CV und AV) voneinander getrennt dargestellt. Für jede dieser Größen zeigt die Bildschirmanzeige den Namen (die Bezeichnung) der Größe, deren aktuellen Messwert, einen verfügbaren Sollwert, den durch den Optimierer berechneten Zielwert, die Einheiten und die Einheitenwerte der Größenveränderungen und einen Verweis auf die laufenden Größenwerte. Für die Ausgangsgrößen wird ebenfalls eine Angabe darüber angezeigt, ob diese Größe eine der ausgewählten und durch die MPC-Steuerung verwendeten Größen ist. Der vorhergesehene, durch die MPC-Steuerung festgestellte Wert dieser Größe und die voreingestellte Priorität dieser Größe wird ebenfalls angezeigt. Dieser Bildschirm ermöglicht es dem Operator, den aktuellen Betriebszustand des erweiterten Steuerblocks einzusehen, um zu beobachten, wie der erweiterte Steuerblock die Steuerung ausführt. Weiterhin kann der Nutzer einige gesteuerte Parameter für die Fähigkeit einer fernen Sollwertsetzung einrichten, sodass von äußere Anwendungen Bedienungsziele für eine von außen durchgreifende Abstimmung gesetzt werden können.

[0116] Fig. 16 ist eine Bildschirmanzeige, die durch eine Anwendung für eine Diagnose erzeugt werden kann, und die einen Diagnosebildschirm zeigt, der einem Nutzer oder Operator zum Ausführen von Diagnosen in dem erweiterten Steuerblock bereitgestellt wird. Insbesondere stellt der Diagnosebildschirm aus Fig. 16 die Steuer- und Randbedingungs-(die Hilfs-)Größen, die Stell- und die Störgrößen getrennt von einander dar. Für jede wird der Name oder die Bezeichnung der Größe mit (in der ersten Spalte) einem Verweis angezeigt, ob eine Fehler- oder Alarmbedingung für diese Größe existiert. Ein solche Fehler oder Alarm kann graphisch unter Verwendung beispielsweise eines grünen Merkzeichens oder eines roten „x“ oder in jeder anderen gewünschten Weise graphisch dargestellt werden. Ein Wert und ein Zustand wird ebenfalls für jede dieser Größen angezeigt. Für die Stellgrößen wird der Wert und der Zustand der $Back_Cal$ (der rückgerechneten oder Feedback-)Größe

für diese Signale angezeigt. Es versteht sich, dass dieser Bildschirm zum Ausführen von Diagnosen in dem erweiterten Steuerblock ausgeführt werden kann, wobei dieser den Operator mit den Information versorgt, die zum Feststellen von Problemen innerhalb des Steuersystems notwendig sind. Natürlich können andere Typen von Bildschirmen und Informationen dem Operator bereitgestellt werden, um es diesem zu ermöglichen, den Betrieb des erweiterten Steuerblocks und die Ausführung von Diagnosen auf dem erweiterten Steuerblock und den Modulen, in welchen er verwirklicht ist, einzusehen und auszuführen.

[0117] Während der erweiterte Funktionsblock in dieser Beschreibung so dargestellt worden ist, dass dieser einen in dem gleichen Funktionsblock lokalisierten Optimierer hat, und aus diesem Grund in dem selben Gerät wie die MPC-Steuerung ausgeführt wird, ist es ebenfalls möglich, den Optimierer in einem getrennten Gerät zu verwirklichen. Insbesondere kann der Optimierer in einem anderen Gerät lokalisiert sein, z. B. in einer der Nutzerworkstations **13**, und mit der MPC-Steuerung wie in **Fig. 2** beschrieben während jeder Ausführung oder Systemabfrage der Steuerung zum Berechnen und Bereitstellen von Zielstellgrößen (MV_T) oder zum Bereitstellen des Teilsatzes der im Ergebnis an die MPC-Steuerung festgelegten Steuer- und Hilfsgrößen (CV und AV) kommunizieren. Natürlich kann eine spezielle Schnittstelle, wie zum Beispiel die bekannte OPC-Schnittstelle verwendet werden, um die Kommunikationsschnittstelle zwischen der Steuerung und dem die MPC-Steuerung enthaltenden Funktionsblock und der Workstation oder dem anderen, den Optimierer realisierenden oder ausführenden Computer bereitzustellen. Wie bei der unter Bezugnahme auf **Fig. 2** beschriebenen Ausführungsform müssen der Optimierer und die MPC-Steuerung miteinander während jedes Abfragezyklus kommunizieren, um die integrierte optimierte MPC-Steuerung auszuführen. Allerdings können in diesem Fall andere gewünschte Typen von Optimierern genutzt werden, z. B. bekannte oder üblichen Echtzeioptimierer, die bereits innerhalb einer Prozesssteuerungsumgebung vorhanden sein können. Dieses Merkmal kann auch vorteilhaft genutzt werden, wenn das Optimierungsproblem nicht linear ist und deren Lösung nicht lineare Programmierungstechniken erfordert.

[0118] Während der erweiterte Steuerblock und andere hier behandelte Blöcke und Routinen so beschrieben wurden, dass diese in Verbindung mit Fieldbus- und Standard 4–20 mA Geräten verwendet werden, können diese natürlich unter Verwendung jeder anderer Prozesssteuerungs-Kommunikationsprotokolle oder Programmierungsumgebungen verwirklicht sein und zusammen mit jedem anderen Typ von Geräten, Funktionsblöcken oder Steuerungen verwendet werden. Obwohl die erweiterten Steuerblöcke und die hier beschriebenen zugeordneten Erzeugungs- und Testroutinen vorzugsweise in Form von Software verwirklicht sind, können diese ebenfalls in Form von Hardware, Firmware und so weiter realisiert sein und durch jeden anderen, einem Prozesssteuerungssystem zugeordneten Prozessor ausgeführt werden. So kann die hier beschriebene Routine **40** in einer üblichen Mehrzweck-CPU oder auf einer spezifisch dafür entworfenen Hardware oder Firmware, beispielsweise einem ASIC, wenn gewünscht, verwirklicht sein. Ist diese in Softwareform gegeben, kann diese Software in jedem computerlesbaren Speicher, z. B. einer Magnetscheibe, insbesondere einer Diskette oder Festplatte, einer Laserdisk, einer optischen Disk oder jedem anderen Speichermedium, in einem RAM oder ROM eines Computers oder Prozessors und so weiter gespeichert werden. Ebenfalls kann diese Software an einen Nutzer oder ein Prozesssteuerungssystem über jede bekannte oder erforderliche Übertragungsmethode einschließlich z. B. einer Computerlesbaren Scheibe oder jedem anderen transportablen Computerspeichermechanismus oder über einen Kommunikationskanal, bspw. eine Telefonleitung, das Internet (was als das gleiche oder austauschbar in Bezug auf das Bereitstellen derartiger Software über ein transportierbares Speichermedium angesehen wird) übertragen werden.

Patentansprüche

1. Prozesssteuerungssystem (**10**) zum Steuern eines Prozesses (**50**), das folgendes aufweist:
einen Optimierer (**54**), der ausgebildet ist, um einen optimalen Betriebspunkt des Prozesses (**50**) auf der Grundlage einer ersten Anzahl prädiktiver Werte der Steuerungs- und Hilfsgrößen des Prozesses (**50**) und auf der Grundlage einer zweiten Anzahl momentaner Werte von Stellgrößen des Prozesses (**50**) zu bestimmen; und eine Steuerung (**11**) mit mehrfacher Eingabe/mehrfacher Ausgabe, die ausgebildet ist, um die zweite Anzahl von Stellgrößen auf der Grundlage eines vorgegebenen Teilsatzes der ersten Anzahl prädiktiver Werte der Steuerungs- und Hilfsgrößen des Prozesses (**50**) zu erzeugen, wobei der vorgegebene Teilsatz sich zahlenmäßig von der ersten Anzahl der Steuerungs- und Hilfsgrößen des Prozesses (**50**) unterscheidet.
2. Prozesssteuerungssystem (**10**) nach Anspruch 1, wobei der vorgegebene Teilsatz kleiner oder gleich der zweiten Anzahl der Stellgrößen ist.

3. Prozesssteuerungssystem (10) nach Anspruch 1 oder 2, wobei die Steuerung (11) mit mehrfacher Eingabe/mehrfacher Ausgabe eine modellbasierte prädiktive Steuerung ist, die eine quadratische Steuerungsmatrix einschließt.

4. Prozesssteuerungssystem (10) nach Anspruch 3, wobei die quadratische Steuermatrix Steuer- und Hilfsgrößen des Prozesses (50) als Eingaben und Stellgrößen des Prozesses (50) als Ausgaben beinhaltet und eine Anzahl von Steuer- und Hilfsgrößen beinhaltet, die gleich der Anzahl der Stellgrößen ist.

5. Prozesssteuerungssystem (10) nach Anspruch 3, wobei die quadratische Steuermatrix Steuer- und Hilfsgrößen des Prozesses (50) als Eingaben und Stellgrößen des Prozesssteuerungssystems (10) als Ausgaben enthält und eine Anzahl von Steuer- und Hilfsgrößen enthält, die kleiner als die Anzahl der Stellgrößen ist und mindestens eine der Stellgrößen sowohl als Stellgrößenausgabe und Steuergrößeneingabe hat.

6. Prozesssteuerungssystem (10) nach Anspruch 5, wobei die Anzahl der mindestens einen der Stellgrößen, denen zugewiesen ist, dass sie sowohl Stellgrößenausgaben als auch Steuergrößeneingaben sind, gleich der Differenz zwischen der Gesamtzahl der Stellgrößen und der Anzahl der als Steuergrößeneingabe der Steuermatrix verwendeten Steuer- und Hilfsgrößen ist.

7. Prozesssteuerungssystem (10) nach Anspruch 5 oder 6, wobei die mindestens eine der sowohl als Stellgrößenausgabe als auch als Steuergrößeneingabe verwendeten Stellgrößen eine Sprungantwort mit dem Verstärkungsfaktor Eins enthält.

8. Prozesssteuerungssystem (10) nach einem der Ansprüche 3 bis 7, wobei die modellbasierte prädiktive Steuerung zum Minimieren des quadratischen Steuerungsfehlers über einen Prädiktionshorizont und zum Minimieren von Stellgrößen-Bewegungen über einen Steuerungshorizont erzeugt wird.

9. Prozesssteuerungssystem (10) nach Anspruch 8, wobei die modellbasierte prädiktive Steuerung erzeugt wird, um die Summe der Stellgrößenbewegungen über den Steuerungshorizont gleich zu einer optimalen Zielveränderung der Stellgrößen zu erhalten.

10. Prozesssteuerungssystem (10) nach Anspruch 8 oder 9, wobei ein Satisfaktionsgrad der Minimierung des quadratischen Fehlers über den Prädiktionshorizont und der Minimierung der Stellgrößenbewegungen willkürlich eingestellt werden kann.

11. Prozesssteuerungssystem (10) nach Anspruch 10, wobei der Satisfaktionsgrad des Erhaltens optimaler Zielveränderungen der Stellgrößen willkürlich eingestellt werden kann.

12. Prozesssteuerungssystem (10) nach einem der Ansprüche 1 bis 11, wobei der Optimierer (54) ein Optimierer (54) mit linearer oder quadratischer Programmierung ist.

13. Prozesssteuerungssystem (10) nach Anspruch 12, wobei der Optimierer (54) einen Satz von Zielwerten für den vorgegebenen Teilsatz der ersten Anzahl der Steuer- und Hilfsgrößen erzeugt und die Steuerung (11) mit mehrfacher Eingabe/mehrfacher Ausgabe die Zielwerte für den vorgegebenen Teilsatz der ersten Anzahl der Steuer- und Hilfsgrößen mit gemessenen Werten des vorgegebenen Teilsatzes der ersten Anzahl von Steuer- und Hilfsgrößen kombiniert, um die Stell-Steuersignale zu erzeugen.

14. Prozesssteuerungssystem (10) nach Anspruch 13, wobei die Steuerung (11) mit mehrfacher Eingabe/mehrfacher Ausgabe einen Satz prädiktiver Steuer- und Hilfsgrößen und einen Satz prädiktiver Stellgrößen erzeugt und wobei der Optimierer (54) den Satz der vorgegebenen Steuer- und Hilfsgrößen und den Satz der vorgegebenen Stellgrößen verwendet, um den Satz von Zielwerten für den vorgegebenen Teilsatz der ersten Anzahl der Steuer- und Hilfsgrößen zu erzeugen.

15. Prozesssteuerungssystem (10) nach Anspruch 14, welches weiterhin eine Antwortmatrix aufweist, die die Reaktion von jeder aus der ersten Anzahl der Steuer- und Hilfsgrößen auf eine Veränderung von jeder der zweiten Anzahl der Stellgrößen definiert, wobei der Optimierer (54) einen Satz von Ziel-Stellgrößenwerten erzeugt, die einen optimalen Betriebspunkt definieren, und der Optimierer (54) die Antwortmatrix verwendet, um die Zielwerte des vorgegebenen Teilsatzes der ersten Anzahl der Steuer- und Hilfsgrößenwerte aus dem Satz der Ziel-Stellgrößenwerte zu bestimmen.

16. Prozesssteuerungssystem **(10)** nach Anspruch 15, bei dem der Optimierer **(54)** ausgebildet ist, um den Satz der Ziel-Stellgrößenwerte, die eine Zielfunktion maximieren oder minimieren, zu erzeugen, während jede der Steuergrößen an vorgegebenen Sollwerten und jede der Hilfsgrößen und Stellgrößen innerhalb vorgegebener Randbedingungsgrenzen gehalten wird.

17. Prozesssteuerungssystem **(10)** nach Anspruch 16, bei dem der Optimierer **(54)** ausgebildet ist, um einen Satz von Ziel-Stellgrößenwerten zu erzeugen, die die Zielfunktion maximieren oder minimieren, während jede der Steuergrößen innerhalb vorgegebener Sollwertgrenzen und jede der Hilfsgrößen und Stellgrößen innerhalb von Randbedingungsgrenzen gehalten wird, wenn eine Lösung nicht existiert, bei der jede der Steuergrößen an vorbestimmten Sollwerten und jede der Hilfsgrößen und Stellgrößen innerhalb vorgegebener Randbedingungsgrenzen gehalten wird.

18. Prozesssteuerungssystem **(10)** nach Anspruch 17, bei der der Optimierer **(54)** ausgebildet ist, um den Satz von Ziel-Stellgrößenwerten, der die Zielfunktion maximiert oder minimiert, zu erzeugen, während jede der Steuergrößen innerhalb vorgegebener Sollwertgrenzen und die Stellgrößen innerhalb vorgegebener Randbedingungsgrenzen gehalten werden, und während es einer oder mehrerer der Hilfsgrößen erlaubt ist, vorgegebene Randbedingungsgrenzen zu verletzen, und zwar auf der Grundlage von mit den Hilfsgrößen verknüpften Prioritäten, wenn eine Lösung nicht existiert, bei der jede der Steuergrößen innerhalb vorgegebener Sollwertgrenzen und jede der Hilfsgrößen und Stellgrößen innerhalb vorgegebener Randbedingungsgrenzen gehalten wird.

19. Verfahren zum Steuern eines Prozesses **(50)** mit einer Vielzahl von Stellgrößen und einer Mehrzahl von Steuer- und Hilfsgrößen, die durch Änderungen in den Stellgrößen beeinflussbar sind, wobei die Vielzahl der Stellgrößen zahlenmäßig von der Mehrzahl der Steuer- und Hilfsgrößen verschieden ist, und wobei das Verfahren folgende Schritte umfasst:

Auswählen eines Teilsatzes aus der Mehrzahl der Steuer- und Hilfsgrößen zur Verwendung beim Durchführen der Prozesssteuerung;

Erzeugen einer Steuermatrix unter Verwendung des ausgewählten Teilsatzes der Mehrzahl von Steuer- und Hilfsgrößen und der Vielzahl der Stellgrößen;

Erzeugen einer Steuerung aus der Steuermatrix, die den ausgewählten Teilsatz aus der Mehrzahl von Steuer- und Hilfsgrößen als Eingaben und die Vielzahl der Stellgrößen als Ausgaben aufweist;

Durchführen einer Prozessoptimierung durch Auswählen eines Prozessbetriebspunktes, um eine Zielfunktion, abhängig von der Vielzahl der Stellgrößen und der Mehrzahl der Steuer- und Hilfsgrößen, zu minimieren oder zu maximieren, wobei der Prozessbetriebspunkt durch einen Satz von Zielwerten für den ausgewählten Teilsatz der Mehrzahl der Steuer- und Hilfsgrößen definiert wird;

Ausführen einer Steuerung mit mehrfacher Eingabe/mehrfacher Ausgabe unter Verwendung einer aus der Steuermatrix erzeugten Steuerung, um einen Satz von Stellgrößenwerten aus den Zielwerten für den ausgewählten Teilsatz der Mehrzahl der Steuer- und Hilfsgrößen und den gemessenen Werten des ausgewählten Teilsatzes der Mehrzahl von Steuer- und Hilfsgrößen zu entwickeln; und

Verwenden des entwickelten Satzes von Stellgrößenwerten, um den Prozess zu steuern.

20. Verfahren nach Anspruch 19, wobei die Schritte des Durchführens der Prozessoptimierung und des Durchführens der Steuerung mit mehrfacher Eingabe/mehrfacher Ausgabe während aufeinanderfolgender Scanperioden des Prozesses **(50)** ausgeführt werden.

21. Verfahren nach Anspruch 19 oder 20, wobei die Schritte der Auswahl des Teilsatzes aus der Mehrzahl der Steuer- und Hilfsgrößen und der Erzeugung einer Steuermatrix vor dem Onlinebetrieb des Prozesses **(50)** erfolgen.

22. Verfahren nach einem der Ansprüche 19 bis 21, wobei die Schritte der Auswahl des Teilsatzes aus der Mehrzahl der Steuer- und Hilfsgrößen beinhaltet, dass jeder der Steuer- oder Hilfsgrößen innerhalb des Teilsatzes der Mehrzahl der Steuer- oder Hilfsgrößen auf der Grundlage einer minimalen Bedingungsanzahl für die Steuermatrix ausgewählt wird.

23. Verfahren nach einem der Ansprüche 19 bis 22, wobei der Schritt der Auswahl des Teilsatzes der Mehrzahl der Steuer- und Hilfsgrößen beinhaltet, dass eine der Steuer- oder Hilfsgrößen auf der Grundlage einer Verstärkungsantwort von einer der Steuer- oder Hilfsgrößen auf eine der Stellgrößen als am meisten auf eine der Stellgrößen ansprechend ausgewählt wird.

24. Verfahren nach einem der Ansprüche 19 bis 23, wobei der Schritt der Auswahl des Teilsatzes aus der Mehrzahl der Steuer- und Hilfsgrößen beinhaltet, dass eine der Steuer- oder Hilfsgrößen auf der Grundlage einer Antwortzeit einer der Steuer- oder Hilfsgrößen auf eine der Stellgrößen als am meisten auf eine der Stellgrößen ansprechend ausgewählt wird.

25. Verfahren nach einem der Ansprüche 19 bis 24, wobei der Schritt der Auswahl des Teilsatzes aus der Mehrzahl der Steuer- und Hilfsgrößen beinhaltet, dass eine der Steuer- oder Hilfsgrößen als am meisten auf eine der Stellgrößen ansprechend ausgewählt wird, und zwar auf der Grundlage einer Kombination einer Verstärkungsantwort und einer Antwortzeit der einen der Steuer- oder Hilfsgrößen auf die eine der Stellgrößen.

26. Verfahren nach einem der Ansprüche 19 bis 25, wobei der Schritt der Durchführung der Prozesssteuerung mit mehrfacher Eingabe/mehrfacher Ausgabe den Schritt einer modellbasierten prädiktiven Steuerung beinhaltet.

27. Verfahren nach Anspruch 26, wobei der Schritt der Durchführung der Prozessoptimierung den Schritt einer Durchführung einer linearen oder quadratischen Programmierung beinhaltet.

28. Verfahren nach einem der Ansprüche 19 bis 27, wobei der Schritt der Auswahl des Teilsatzes aus der Mehrzahl der Steuer- und Hilfsgrößen den Schritt des Auswählens einer einzelnen und unterschiedlichen der Steuer- und Hilfsgrößen für jede der Mehrzahl von Stellgrößen beinhaltet.

29. Prozesssteuerungselement, ausgebildet zur Verwendung als ein Bereich einer auf einem Prozessor implementierten Steuerungsroutine, um eine Vielzahl von Steuer- und Hilfsparametern eines Prozesses (**50**) unter Verwendung einer Mehrzahl von Stellparametern zu steuern, wobei das Prozesssteuerungselement folgendes enthält:

einen computerlesbaren Datenträger;

einen auf dem computerlesbaren Datenträger gespeicherten und für die Ausführung auf dem Prozessor ausgebildeten Funktionsblock, um eine Steuerung (**11**) mit mehrfacher Eingabe/mehrfacher Ausgabe des Prozesses (**50**) während jeder Steuerungs-Scanperiode zu implementieren, wobei der Funktionsblock enthält:

eine Zielfunktion, die ein Optimierungskriterium auf der Grundlage der Vielzahl von Steuer- und Hilfsparametern definiert;

eine Optimierer-Routine, die die Zielfunktion verwendet, um einen Satz optimaler Zielwerte für die Steuer- und Hilfsparameter während jeder Steuerungs-Scanperiode zu erzeugen;

eine Steuermatrix, die einen vorgegebenen Teilsatz der Vielzahl der Steuer- und Hilfsparameter zu der Mehrzahl der Stellparameter in Beziehung setzt; und

– eine Steuerungsroutine für mehrfache Eingabe/mehrfache Ausgabe, die ein Steuersignal für jeden der Mehrzahl der Stellparameter während jeder Steuer-Scanperiode unter Verwendung der Steuermatrix und der Zielwerte für den Teilsatz der Vielzahl der Steuer- und Hilfsgrößen erzeugt, wobei die Steuersignale so bestimmt werden, dass sie den Teilsatz der Vielzahl der Steuer- und Hilfsparameter zu den optimalen Zielwerten für den Teilsatz der Steuer- und Hilfsparameter treiben.

30. Prozesssteuerungselement nach Anspruch 29, wobei die Optimierer-Routine eine Routine zur linearen oder quadratischen Programmieroutine Programmierung beinhaltet.

31. Prozesssteuerungselement nach Anspruch 30, wobei die Steueroutine für die mehrfache Eingabe/mehrfache Ausgabe eine modellbasierte prädiktive Steuerungsroutine enthält.

32. Prozesssteuerungselement nach Anspruch 31, wobei die Steuermatrix quadratisch ist und eine erste Anzahl von Steuer- und Hilfsparametern und die erste Anzahl der Stellparameter verwendet und wobei die Zielfunktion ein Optimierungskriterium auf der Grundlage einer zweiten Anzahl von Steuer- und Hilfsparametern definiert, wobei die zweite Anzahl verschieden von der ersten Anzahl ist.

33. Prozesssteuerungselement nach einem der Ansprüche 29 bis 32, wobei die Zielfunktion für den Benutzer während des Prozessbetriebs wählbar ist.

34. Prozesssteuerungselement nach einem der Ansprüche 29 bis 33, das weiterhin eine Vielzahl unterschiedlicher potentieller Zielfunktionen beinhaltet, die als Zielfunktion, die durch die Optimierer-Routine zu verwenden ist, auswählbar sind.

35. Prozesssteuerungselement nach einem der Ansprüche 29 bis 34, wobei der Funktionsblock einen Speicher zum Speichern eines Satzes von Steuerparameter-Sollwerten und eines Satzes von Hilfs- und Stellparameter-Grenzen beinhaltet und wobei die Optimierer-Routine konfiguriert ist, um den Satz optimaler Zielwerte für die Stellparameter zu bestimmen, die dazu führen, dass die Steuerparameter sich an den Steuerparameter-Sollwerten befinden, die Hilfs- und Stellparameter sich innerhalb der Hilfs- und Stellparametergrenzen befinden und die Zielfunktion minimiert oder maximiert wird.

36. Prozesssteuerungselement nach Anspruch 35, bei dem der Speicher ebenfalls einen Satz von Steuerparameter-Sollwertgrenzen speichert und die Optimierer-Routine ausgebildet ist, um den Satz optimaler Zielwerte für die Stellparameter zu erzeugen, die die Zielfunktion maximieren oder minimieren, während jeder der Steuerparameter innerhalb der Steuerparameter-Sollwertgrenzen und jeder der Hilfsparameter und Stellparameter innerhalb der Hilfs- und Stellparametergrenzen gehalten wird, wenn eine Lösung nicht existiert, die die Steuerparameter bei den Steuerparameter-Sollwerten und die Hilfs- und Stellparameter innerhalb der Hilfs- und Stellparametergrenzen hält.

37. Prozesssteuerungselement nach Anspruch 36, bei dem der Speicher weiterhin einen Satz von Prioritätsangaben für die Hilfsparameter speichert und die Optimierer-Routine ausgebildet ist, um einen Satz von Ziel-Stellparametern zu erzeugen, die die Zielfunktion maximieren oder minimieren, während jeder der Steuerparameter innerhalb der Steuerparameter-Sollwertgrenzen gehalten wird, und während es einem oder mehreren der Hilfsparameter auf der Grundlage der Prioritätsangaben für die Hilfsparameter gestattet wird, die Hilfsparametergrenzen zu verletzen, wenn eine Lösung nicht existiert, bei der jeder der Steuerparameter innerhalb der Steuerparameter-Sollwertgrenzen und jeder der Hilfsparameter und Stellparameter innerhalb der Hilfs- und Stellparametergrenzen gehalten wird.

38. Prozesssteuerungselement nach einem der Ansprüche 29 bis 37, bei dem die Steuerroutine zum Erzeugen eines prädiktiven Wertes für jeden der Steuer-, Hilfs- und Stellparameter und zum Bereitstellen der Prädiktionswerte von jedem der Steuer-, Hilfs- und Stellparameter für die Optimierer-Routine ausgebildet ist und bei dem die Optimierer-Routine für die Verwendung der prädiktiven Werte für die Steuer-, Hilfs- und Stellparameter ausgebildet ist, um die Zielwerte für die Teilsatz der Steuer- und Hilfsparameter zu bestimmen.

39. Verfahren zum Ausführen einer Steuerung eines Prozesses (50), der eine erste Anzahl von Steuer- und Hilfsgrößen aufweist, die durch eine zweite Anzahl von Stellgrößen gesteuert werden, wobei das Verfahren folgende Schritte aufweist:

Bestimmen einer Sprungantwortmatrix, die eine Antwort auf jede der Steuer- und Hilfsgrößen auf Änderungen in jede der Stellgrößen definiert;

Auswählen eines Teilsatzes der Steuer- und Hilfsgrößen, wobei der Teilsatz die gleiche oder eine kleinere Anzahl von Steuer- und Hilfsgrößen als Stellgrößen hat;

Erstellen einer quadratischen Steuermatrix aus den Antworten innerhalb der Antwortmatrix für den ausgewählten Teilsatz von Steuer- und Hilfsgrößen und die Stellgrößen; und

während jedes Prozess-Scans:

Durchführen einer Messung von jeder von dem ausgewählten Teilsatz der Steuer- und Hilfsgrößen;

Berechnen eines optimalen Betriebszielwertes für jede von dem ausgewählten Teilsatz der Steuer- und Hilfsgrößen;

Durchführen einer Steuerungsroutine mit mehrfacher Eingabe/mehrfacher Ausgabe unter Verwendung der Zielwerte für jede von dem ausgewählten Teilsatz der Steuer- und Hilfsgrößen, der Messungen des ausgewählten Teilsatzes der Steuer- und Hilfsgrößen und der Steuermatrix, um einen Satz von Stellparametersignalen zu erzeugen; und

Verwenden der Stellparametersignale, um den Prozess zu steuern.

40. Verfahren nach Anspruch 39, wobei die Schritte der Auswahl des Teilsatzes und des Erzeugens einer quadratischen Steuermatrix vor dem Onlinebetrieb des Prozesses (50) durchgeführt werden.

41. Verfahren nach Anspruch 39 oder 40, wobei der Schritt der Auswahl des Teilsatzes von Steuer- und Hilfsgrößen das Auswählen einer der Steuer- und Hilfsgrößen als am besten auf eine der Stellgrößen ansprechend beinhaltet.

42. Verfahren nach einem der Ansprüche 39 bis 41, wobei der Schritt der Auswahl eines Teilsatzes der Steuer- und Hilfsgrößen das Auswählen einer der Steuer- und Hilfsgrößen als am besten auf eine der Stellgrößen ansprechend beinhaltet, und zwar auf der Basis einer Verstärkungsantwort von der einen der Steuer- oder Hilfsgrößen auf die eine der Stellgrößen.

43. Verfahren nach einem der Ansprüche 39 bis 42, wobei der Schritt der Auswahl eines Teilsatzes der Steuer- und Hilfsgrößen das Auswählen von einer der Steuer- oder Hilfsgrößen als am besten auf eine der Stellgrößen ansprechend beinhaltet, und zwar auf der Basis einer Antwortzeit der einen der Steuer- oder Hilfsgrößen auf die eine der Stellgrößen.

44. Verfahren nach einem der Ansprüche 39 bis 43, wobei der Schritt der Auswahl eines Teilsatzes der Steuer- und Hilfsgrößen das Auswählen von einer der Steuer- und Hilfsgrößen als am besten auf eine der Stellgrößen ansprechend beinhaltet, und zwar auf der Basis einer Kombination einer Verstärkungsantwort und einer Antwortzeit der einen der Steuer- und Hilfsgrößen auf die eine der Stellgrößen.

45. Verfahren nach einem der Ansprüche 39 bis 44, wobei der Schritt der Auswahl des Teilsatzes von Steuer- und Hilfsgrößen das automatische Auswählen des Teilsatzes von Steuer- und Hilfsgrößen auf der Grundlage der Antwortmatrix beinhaltet.

46. Verfahren nach einem der Ansprüche 39 bis 45, wobei der Schritt der Auswahl des Teilsatzes der Steuer- und Hilfsgrößen ein Bestimmen einer Bedingungsanzahl einer Matrix beinhaltet, die den ausgewählten Teilsatz der Steuer- und Hilfsgrößen beinhaltet.

47. Verfahren nach einem der Ansprüche 39 bis 46, wobei der Schritt des Erzeugens der quadratischen Steuermatrix folgendes beinhaltet: ein Verwenden von Steuer- und Hilfsgrößen des Prozesses (50) als Eingaben und von Stellgrößen des Prozesses (50) als Ausgaben und ein Auswählen einer Anzahl von Steuer- und Hilfsgrößen, die kleiner als die Anzahl von Stellgrößen ist, und ein Auswählen von mindestens einer der Stellgrößen als eine SteuergröÙeneingabe.

48. Verfahren nach einem der Ansprüche 39 bis 47, wobei der Schritt der Durchführung der Steuerungsroutine mit mehrfacher Eingabe/mehrfacher Ausgabe den Schritt des Durchführens einer modellbasierten prädiktiven Steuerungsroutine enthält.

49. Verfahren nach Anspruch 48, wobei der Schritt der Berechnung eines optimalen Betriebszielwertes das Ausführen einer linearen oder quadratischen Programmierung beinhaltet.

50. Verfahren nach Anspruch 49, wobei der Schritt der Durchführung einer linearen Programmierung das Verwenden einer Zielfunktion enthält, die einen optimalen Betrieb des Prozesses (50) auf der Grundlage der ersten Anzahl der Steuer- und Hilfsgrößen und der zweiten Anzahl der Stellgrößen definiert.

51. Verfahren nach Anspruch 50, das weiterhin das Auswählen von einer aus einem vorgegebenen Satz von Zielfunktionen als die Zielfunktion beinhaltet, die innerhalb der linearen Programmierung zu verwenden ist.

Es folgen 16 Seiten Zeichnungen

Anhängende Zeichnungen

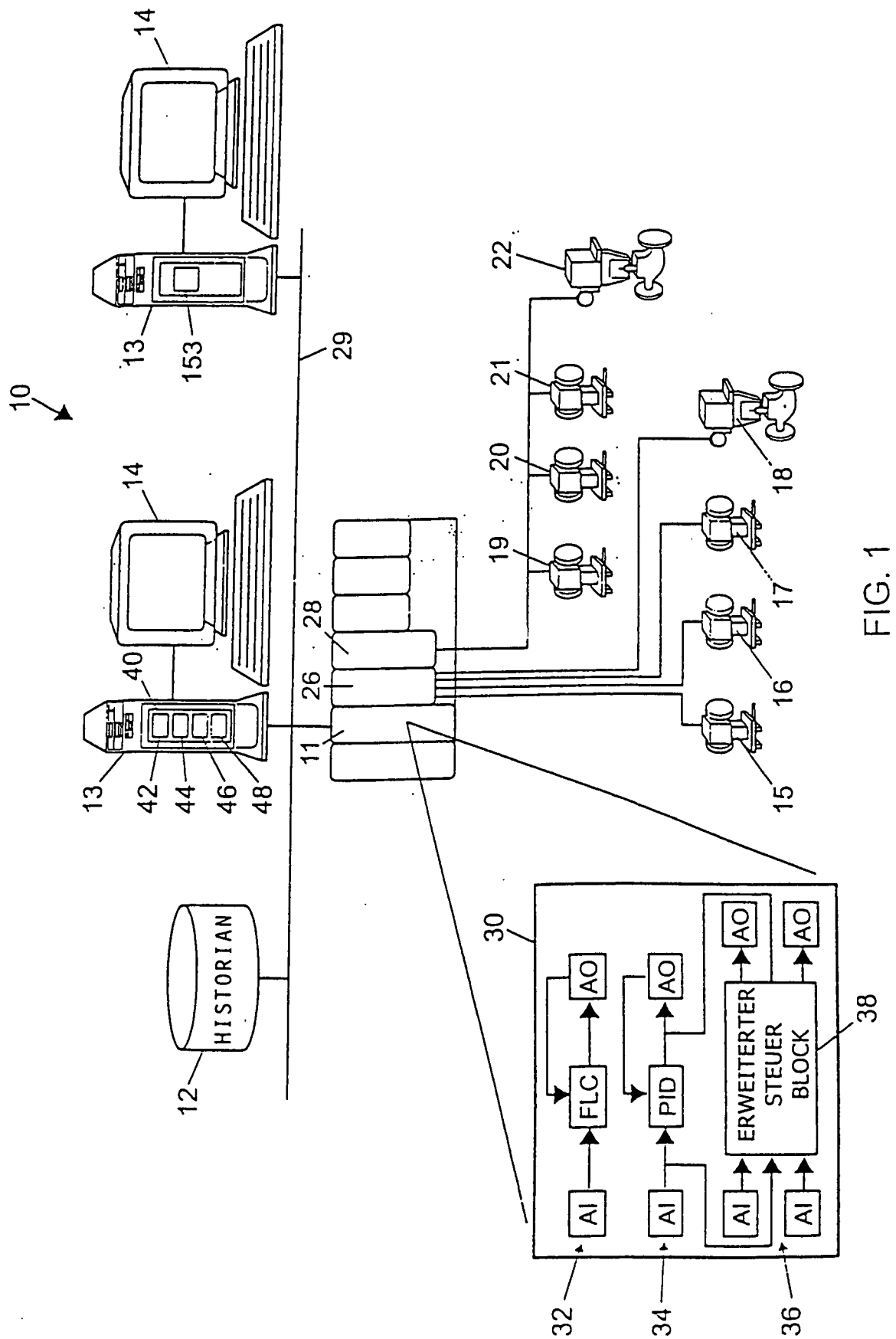
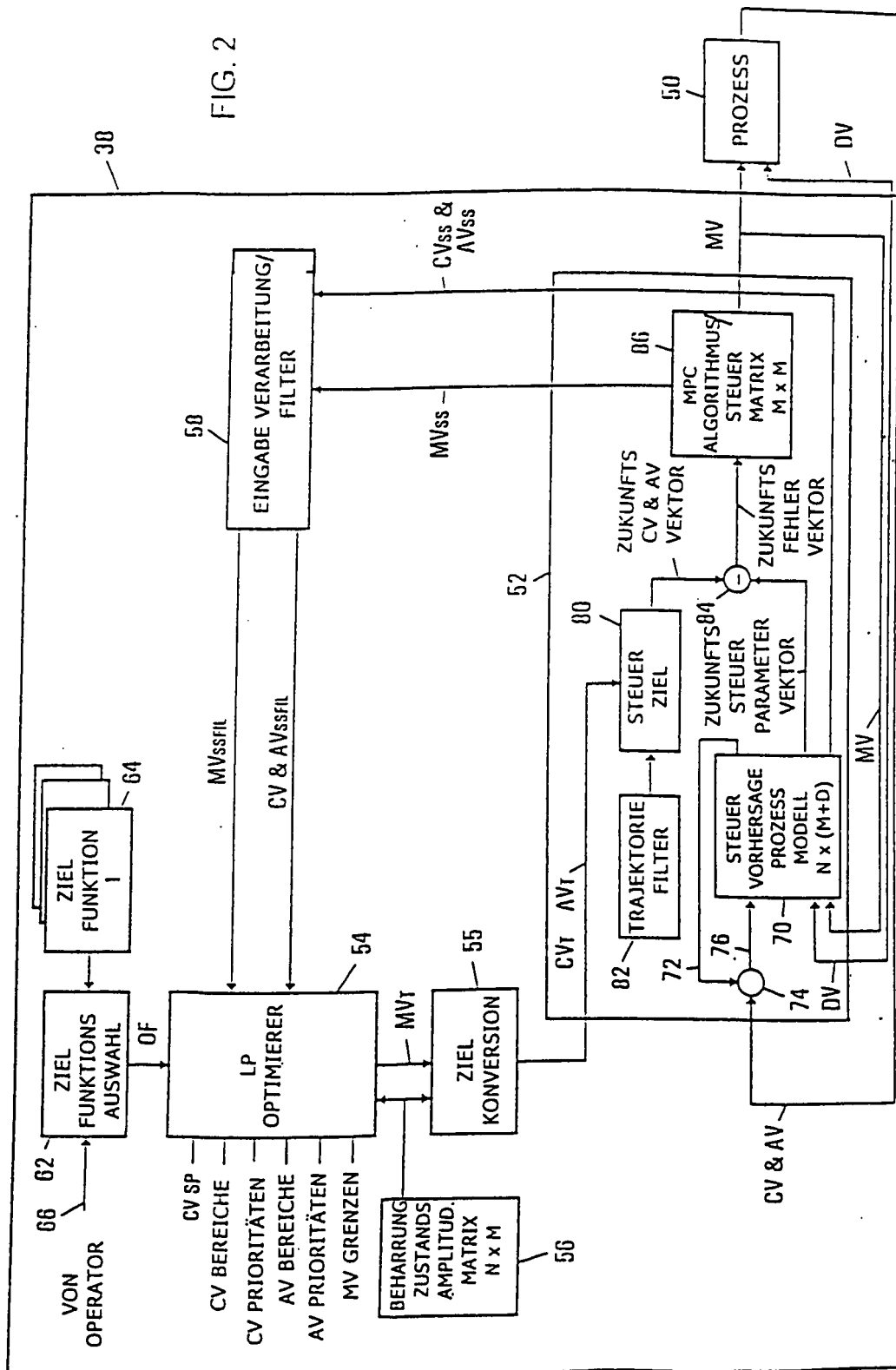


FIG. 2



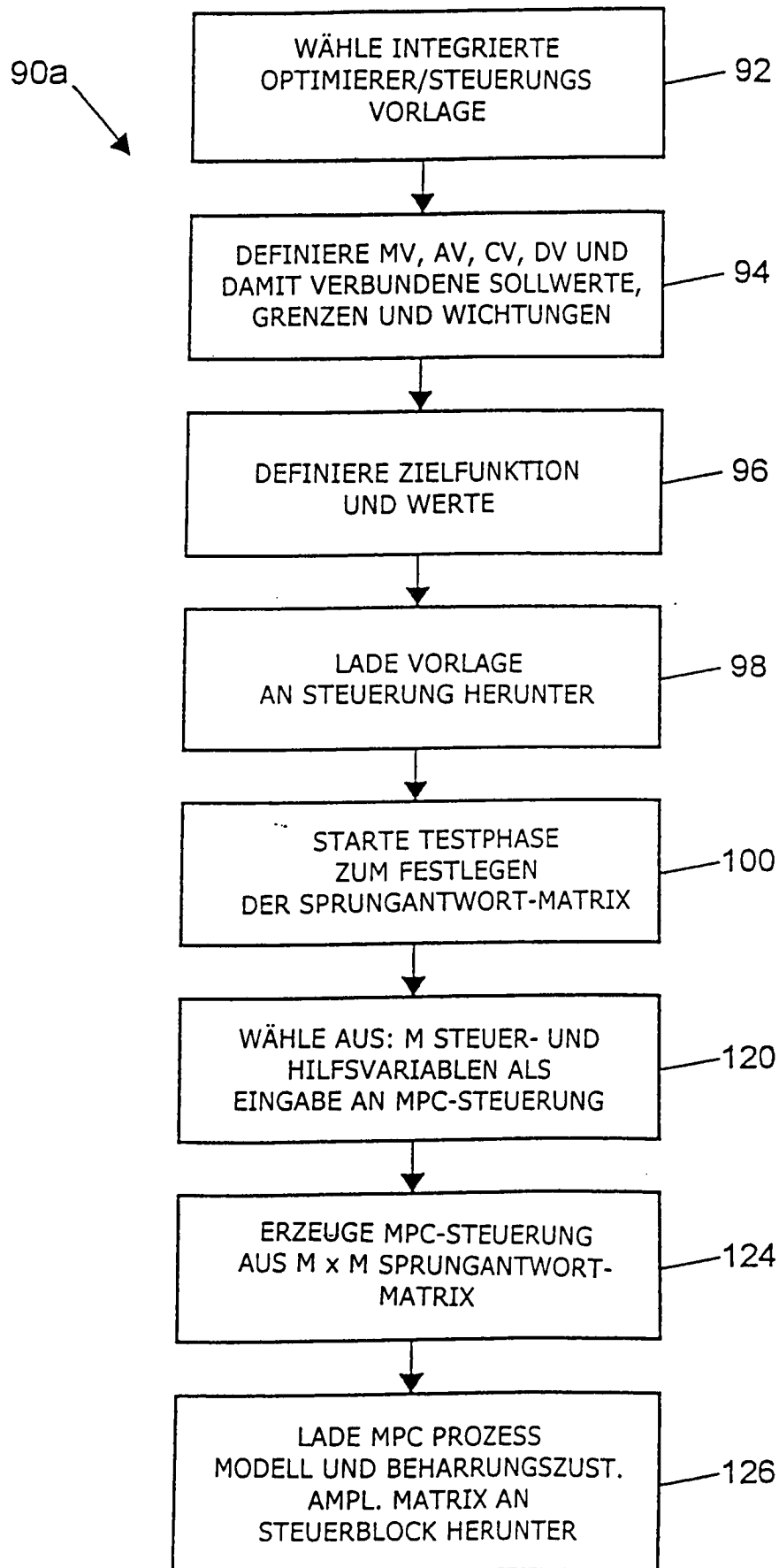


FIG. 3

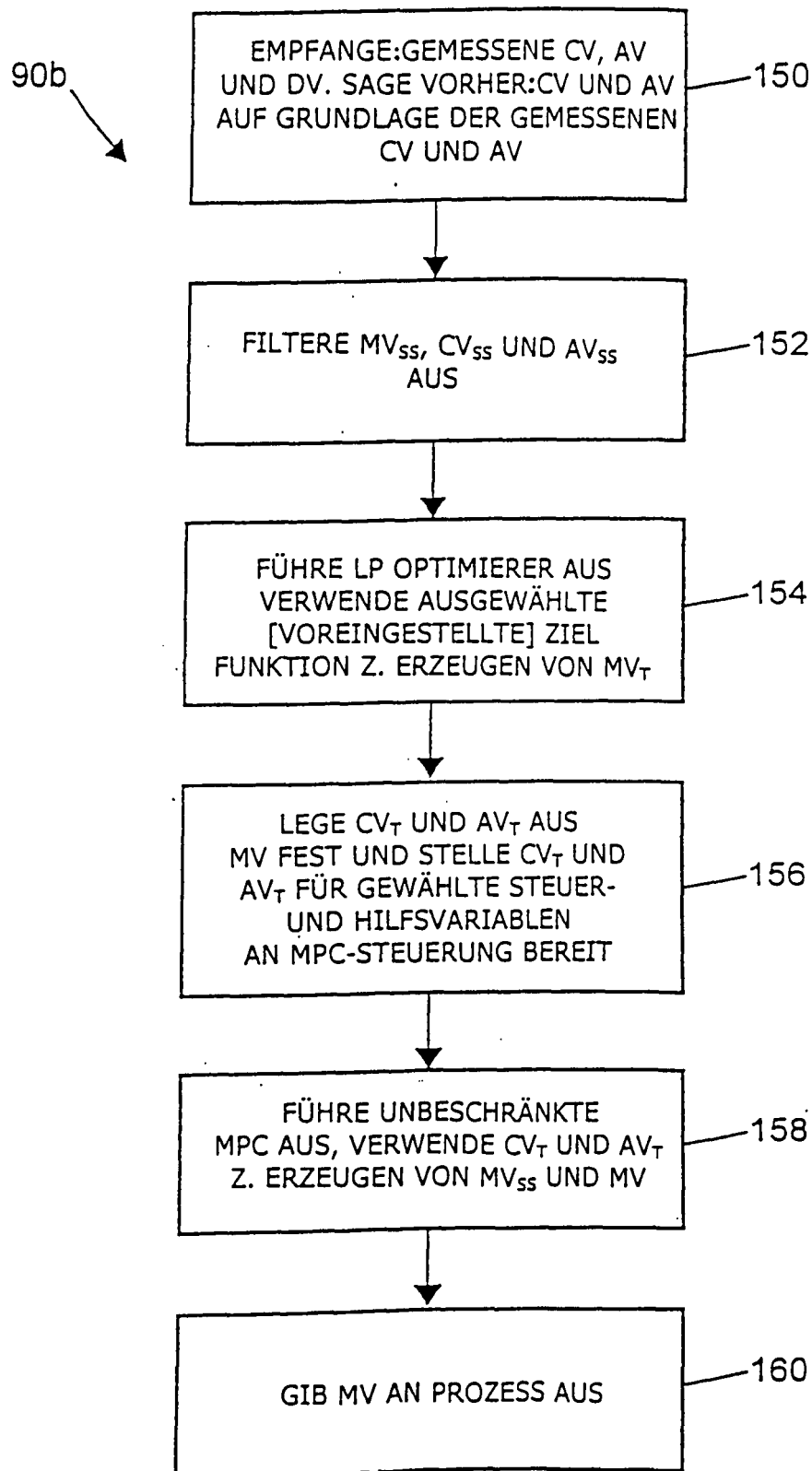


FIG. 4

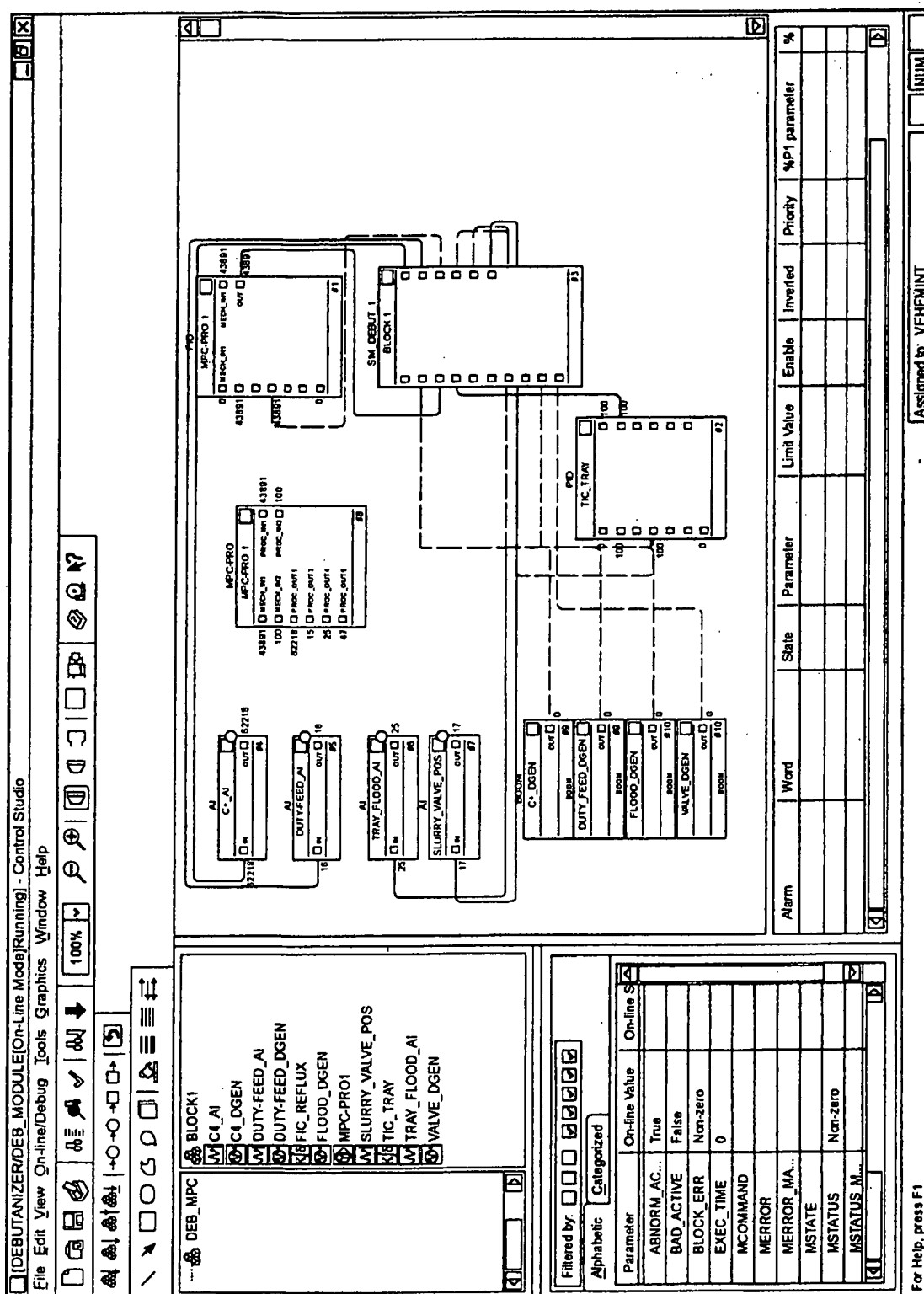


Fig. 5

MPC-PRO1 Properties

Controlled

Manipulated

Disturbance

Constraint

Deleted Items

Description	Low Limit	High Limit	Path	Fail Condition	Optimization	Priority	Profit
REBOILER DUTY/FEED	0	100	//DEB MPC MODULE/DUTY_FEED_AJOUT	Manual	Maximize	0	0
TRAY FLOODING	5	85	//DEB MPC MODULE/TRAY_FLOOD_AJOUT	Simulate	Minimize	0	0
SLURRY REBOILER VAL...	5	85	//DEB MPC MODULE/SLURRY_VALVE_POS/OUT	No Action	< none >	0	0

Total number of process inputs: 2

Total number of process outputs: 4

History sample rate: 1 sec

Close

Add...

Move

Modify...

Delete

<<Advanced

FIG. 6

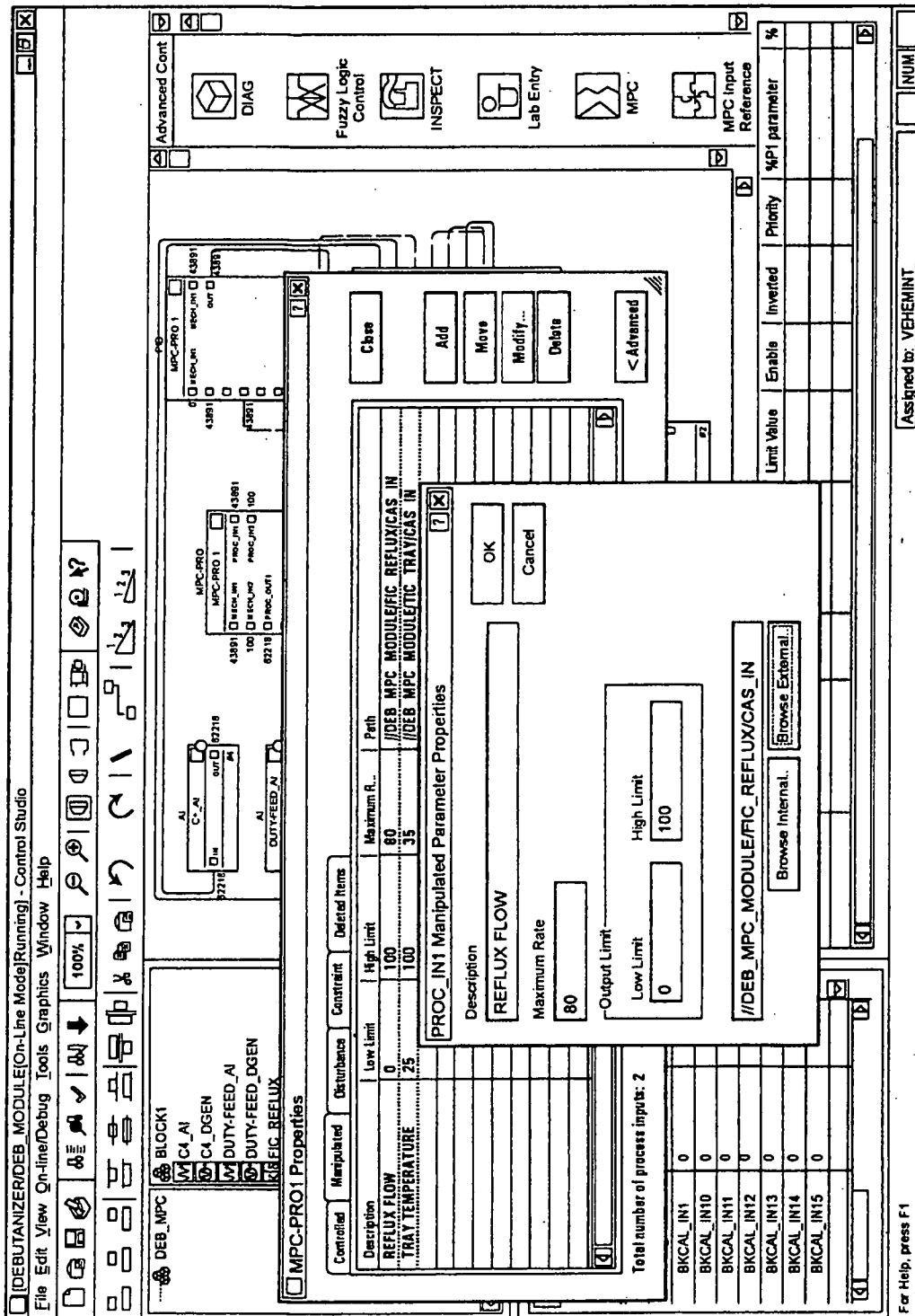


FIG. 7

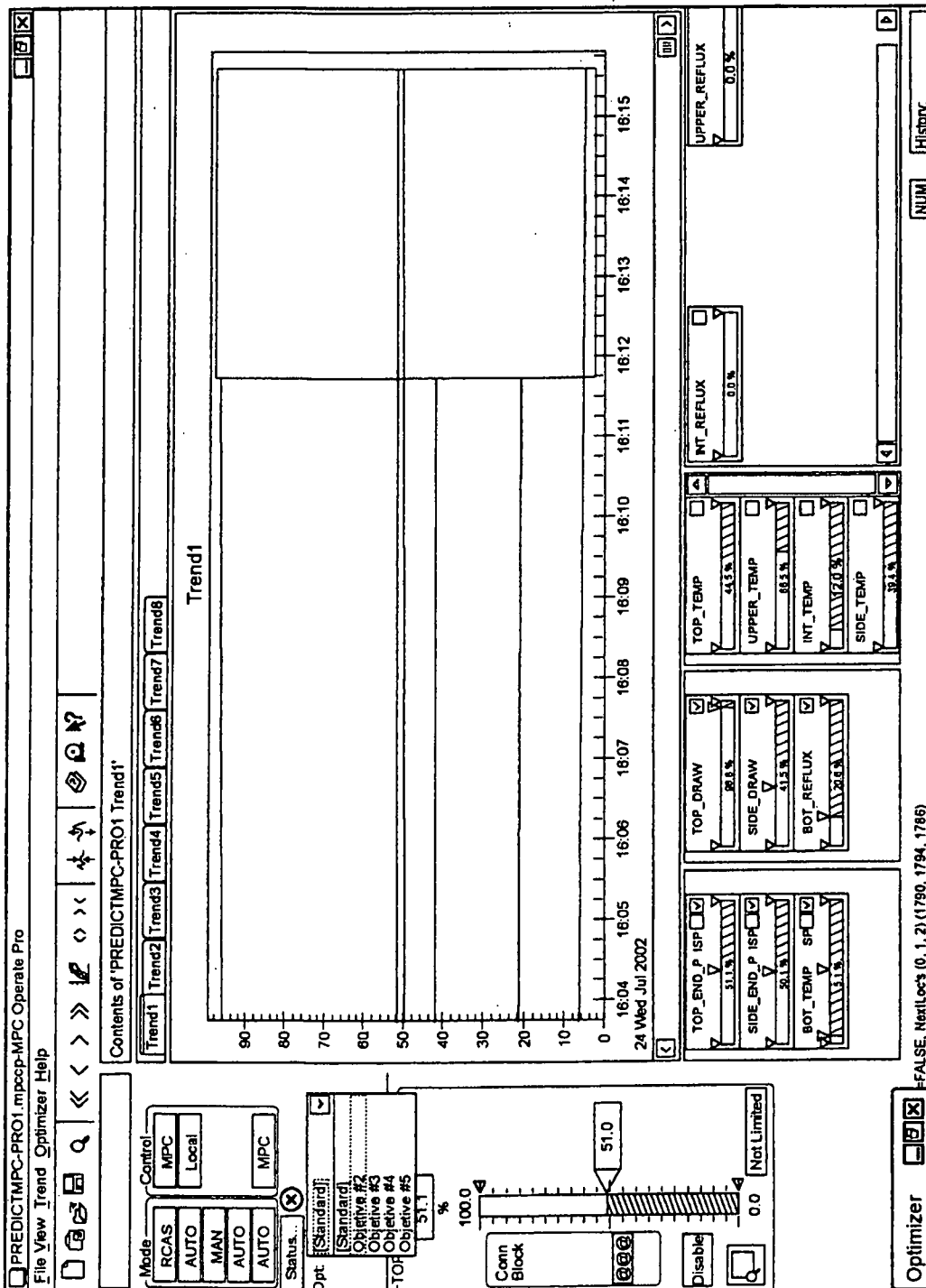
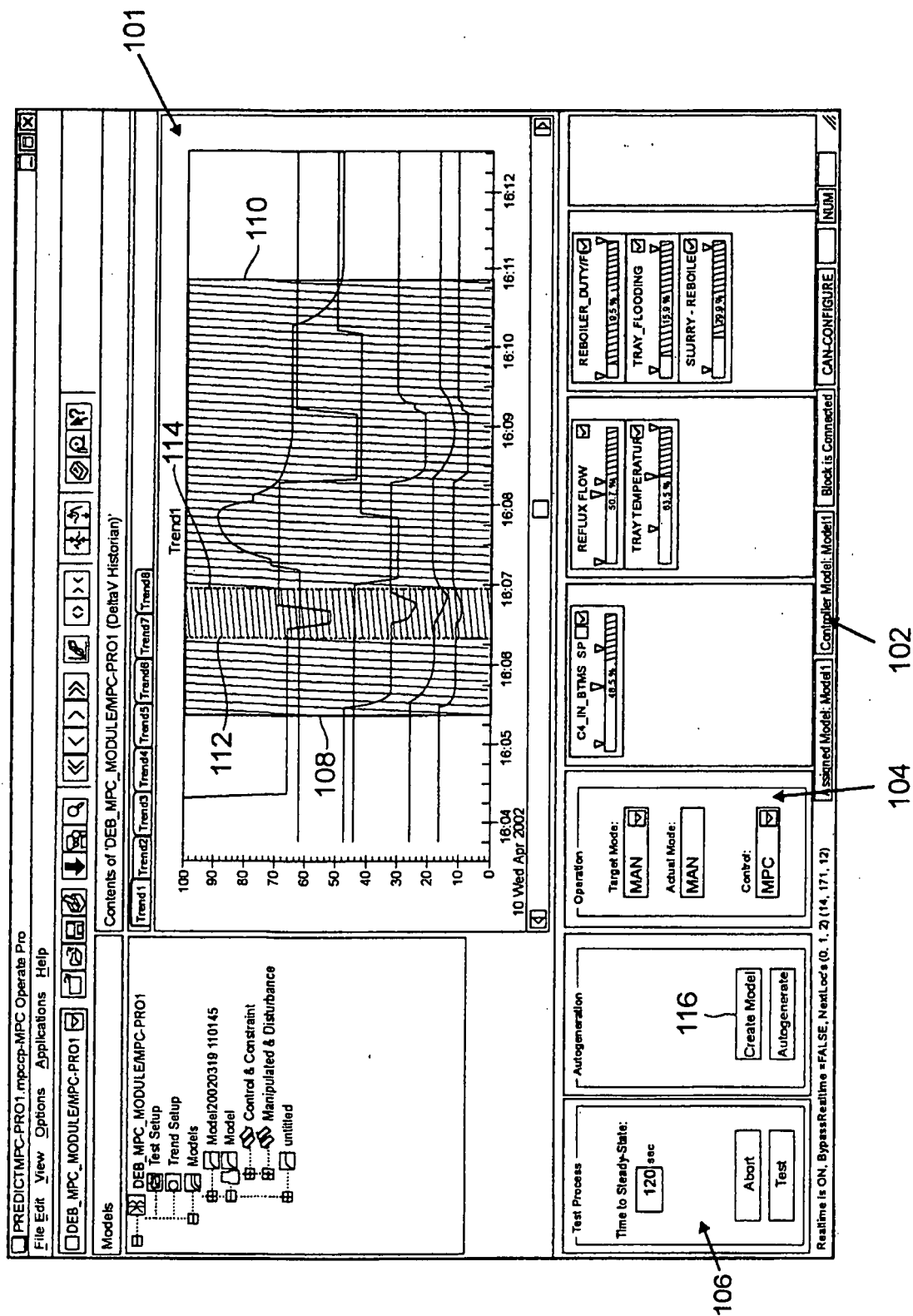


FIG. 8



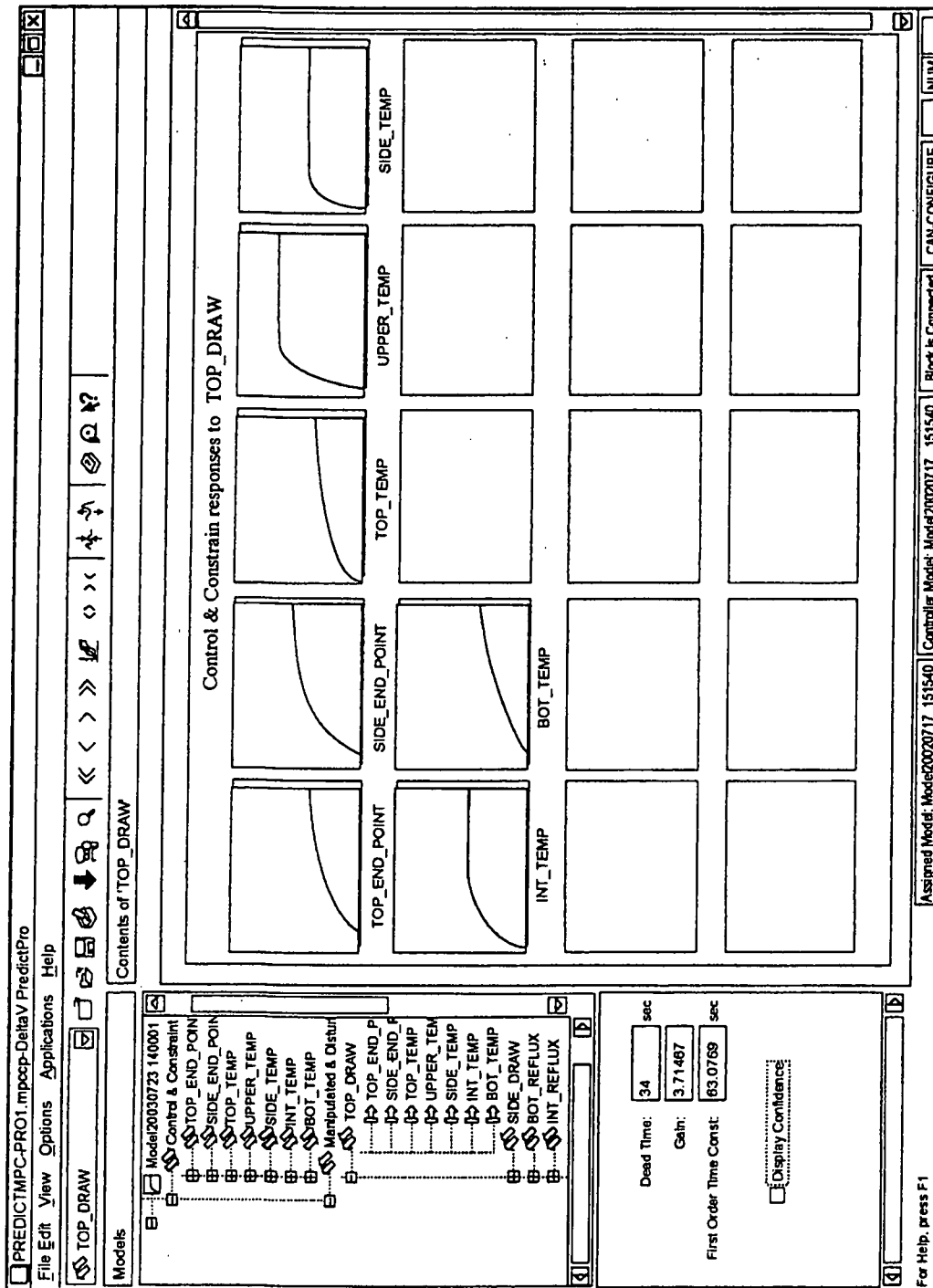


FIG. 10

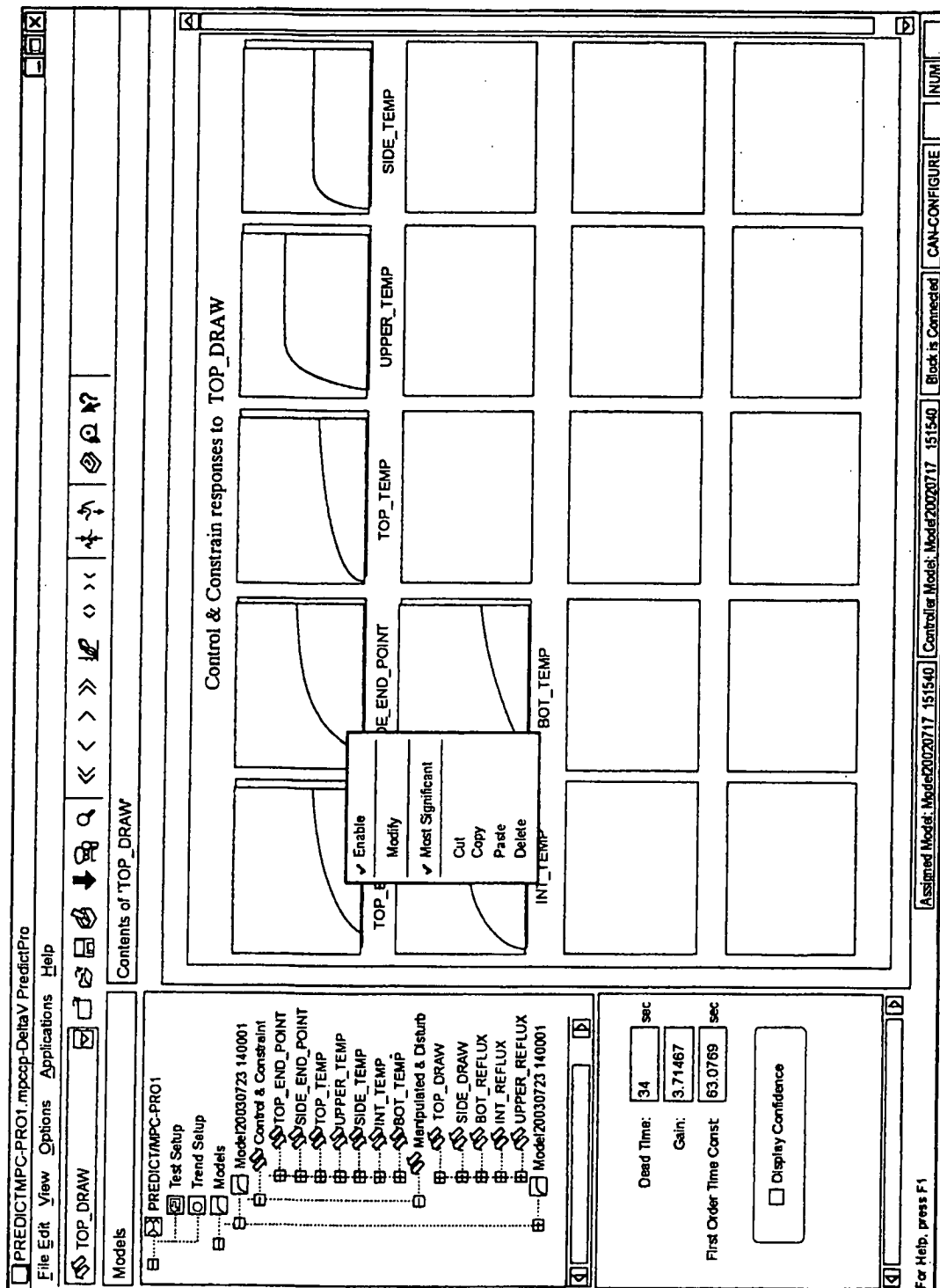


FIG. 11

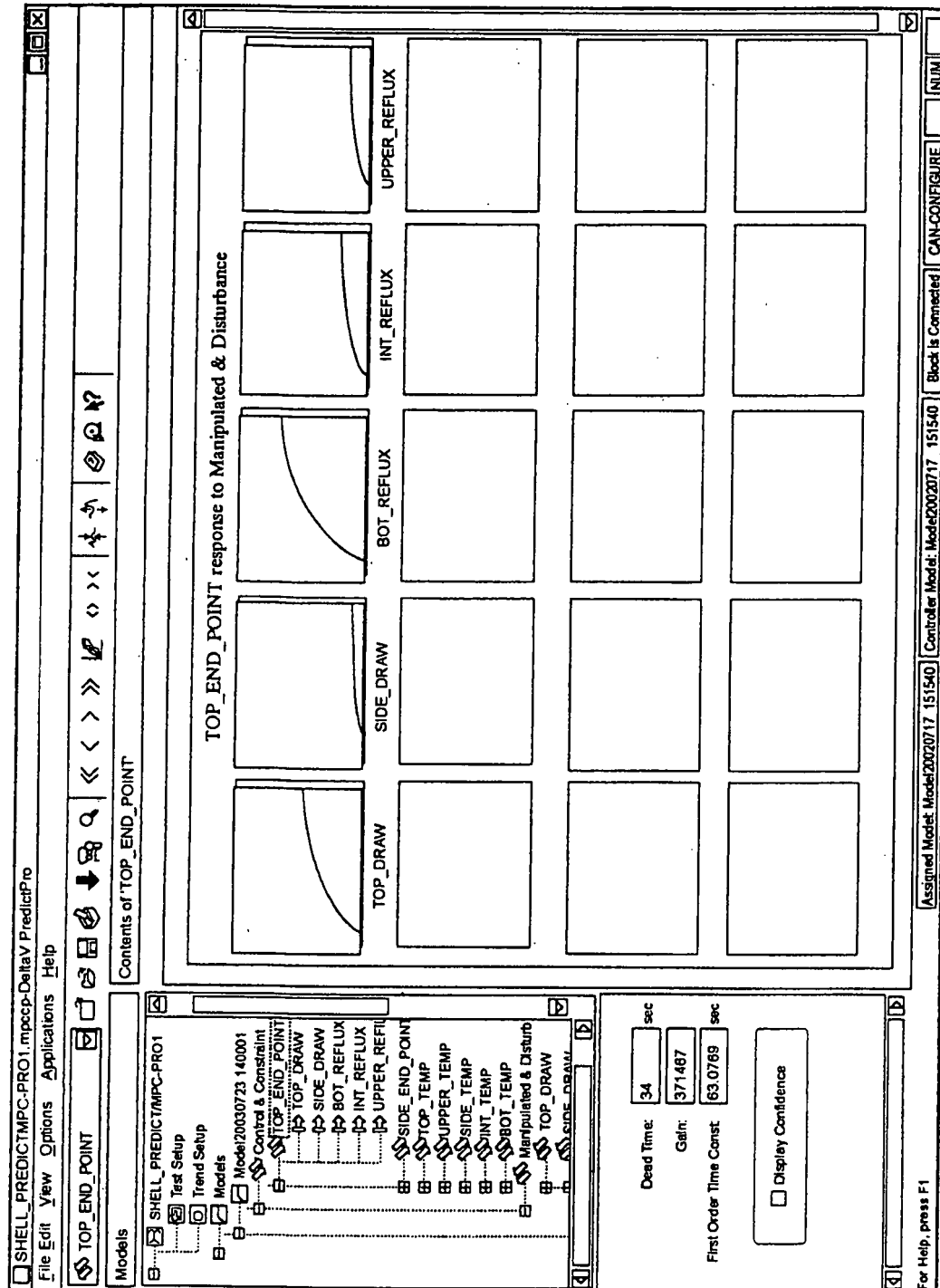


FIG. 12

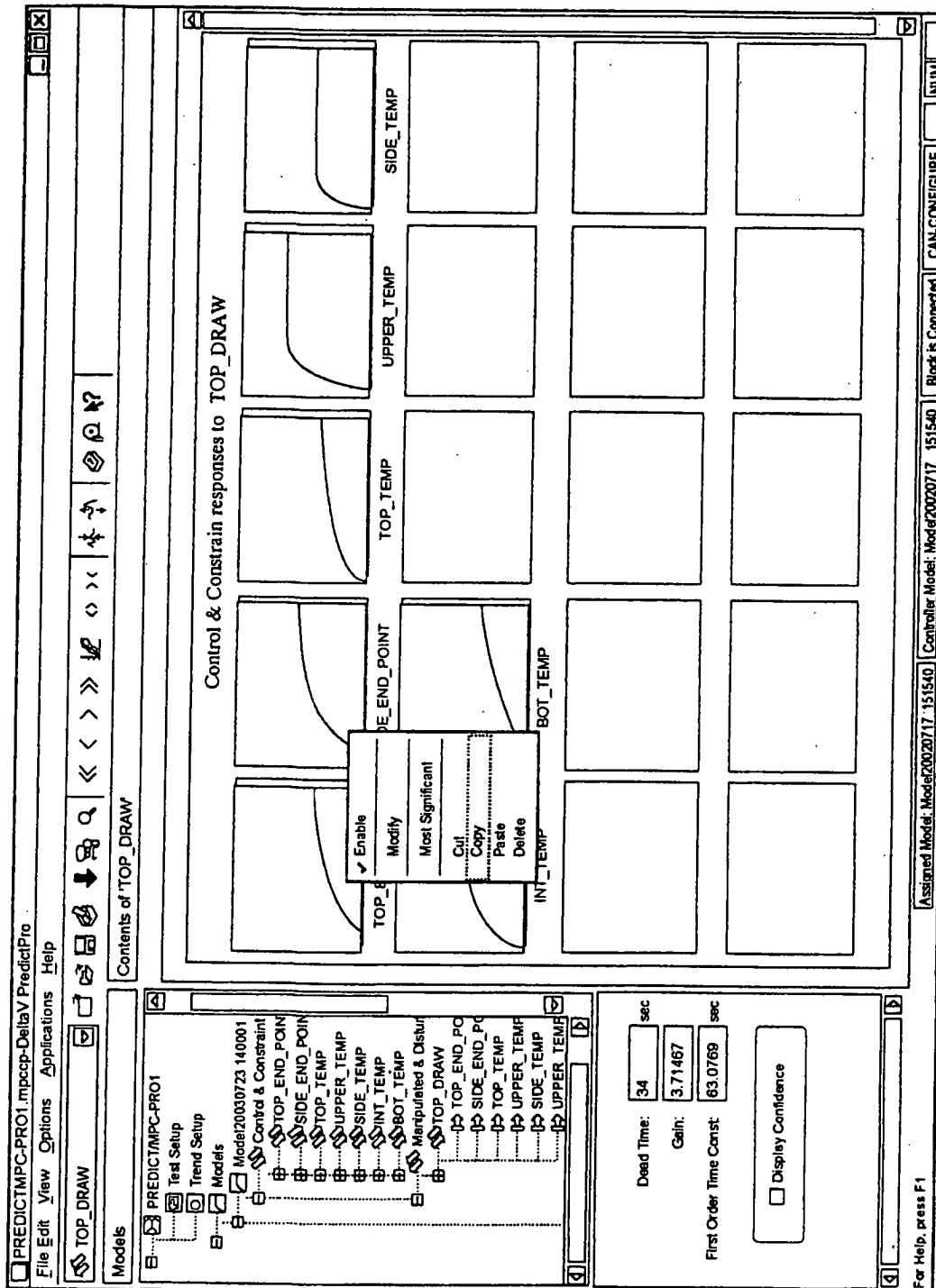


FIG. 13

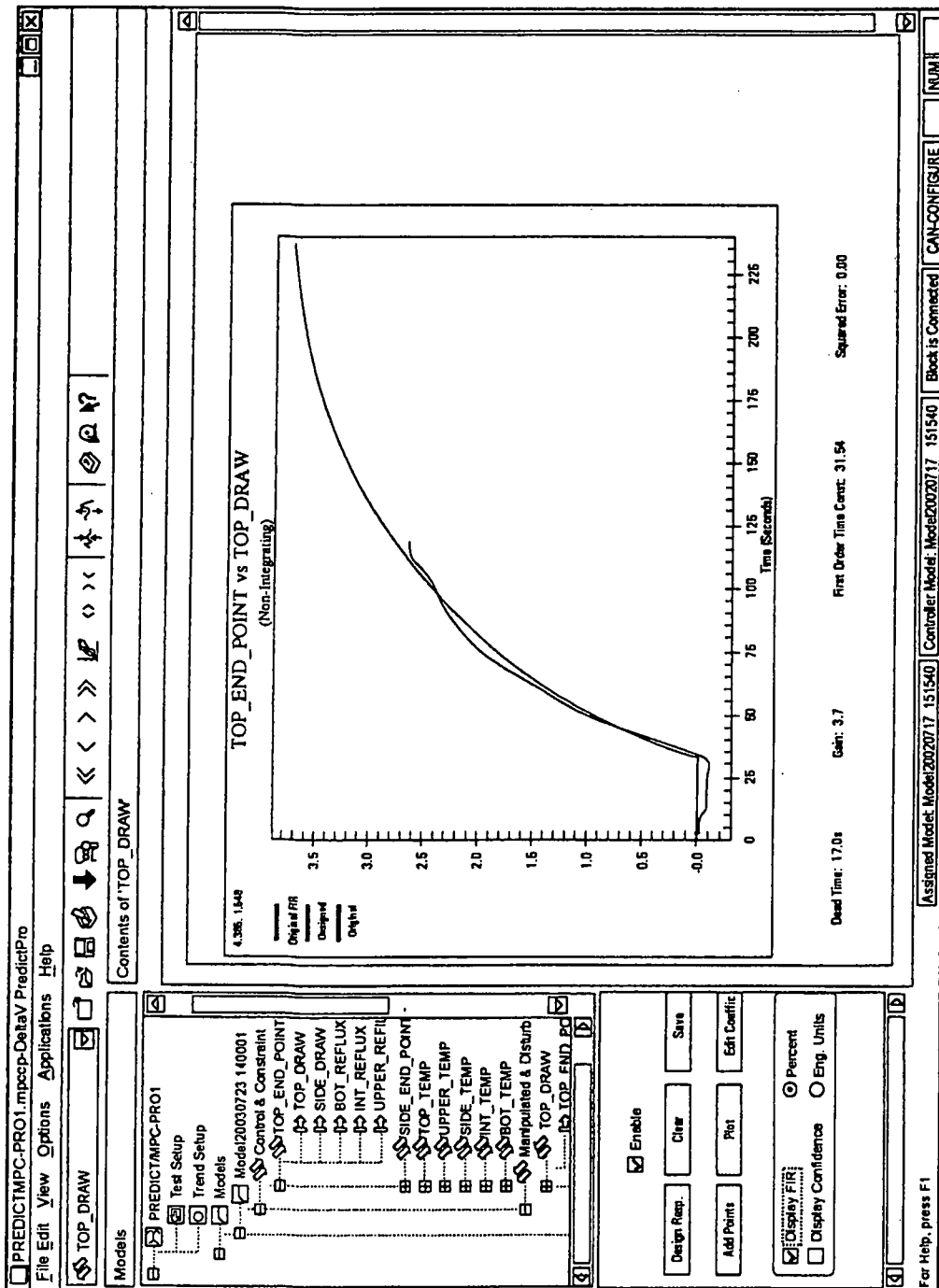


FIG. 14

Optimizer

Inputs (MV)

Descriptor	Current Value	Target Value	Units	Unit Value	Min/Max
TOP_DRAW	96.85	100.00	%	0.10	None
SIDE_DRAW	41.50	41.08	%	0.10	None
BOT_REFLUX	20.48	18.38	%	0.10	Min

Outputs (CV, AV)

Descriptor	Current Value	SP	Target SP	TSP Active	Prediction	Units	Unit Value
TOP_END_POINT	51.10	51.00	51.00	Yes	51.10	%	1.00
SIDE_END_POINT	50.07	50.00	50.00	Yes	50.10	%	1.00
TOP_TEMP	44.40		45.30	Not in ctrl	44.31	%	1.00
UPPER_TEMP	66.58		70.84	Not in ctrl	66.79	%	1.00
BOT_TEMP	4.85	50.00	5.03	Yes	4.89	%	1.00
INT_TEMP	11.84		10.90	Not in ctrl	11.94	%	1.00
SIDE_TEMP	39.32		40.94	Not in ctrl	39.70	%	1.00

-rightclick on Yes/No
 -rightclick on Value
 -rightclick on Min/Max/None
 -rightclick on value

FIG. 15

MPCDiagnostics - PREDICT/MPC-PRO1

File Help

PREDICT/MPC-PRO1
Control and Constraints

Type	Description	Failure	Value	Status	Standard De...
✓	TOP_END_POINT		51.0092	GoodNonCa..	
✓	SIDE_END_POINT		50.0062	GoodNonCa..	
✓	BOT_TEMP		2.1097	GoodNonCa..	
✓	TOP_TEMP		43.6979	GoodNonCa..	
✓	UPPER_TEMP		67.4156	GoodNonCa..	
✓	INT_TEMP		9.38708	GoodNonCa..	
✓	SIDE_TEMP		38.5107	GoodNonCa..	

Disturbance

Type	Description	Value	Status
✗ Error	INT_REFLUX	0	
✗ Error	UPPER_REFLUX	0	

Manipulated

Type	Description	F...	Value	Status	Back Cal Value	Back Cal Status	Standard Deviation
✓	TOP_DRAW		99.7101	Good Cascade	99.7101	Good Cascade	
✓	SIDE_DRAW		40.981	Good Cascade Const	40.981	Good Cascade	
✓	BOT_REFLUX		18.6473	Good Cascade Const	18.6473	Good Cascade	

Block Abnormal Conditions - Description

Type	Failure
------	---------

Ready

FIG. 16