

*Working Papers*



Technische Hochschule  
Ingolstadt

*Zukunft in  
Bewegung*



*Prof. Dr. Wolfgang Krämer*

***LabVIEW-Simulationsprogramme  
und Fallstudien zur modell-  
prädiktiven Regelung***

***Abstract***

Es werden LabVIEW-Simulationsprogramme für das regelungstechnische Praktikum beschrieben, die intensiven Gebrauch von MATLAB-Skript-Nodes machen.

Ein verfahrenstechnischer Prozess und ein Antriebssystem werden modellprädiktiv geregelt. Die Auswirkungen von Begrenzungen, die Teil der Regelungsaufgabe sind, werden untersucht.

## Vorwort

*When all is said  
and all is done,  
usually more is said than done.<sup>1</sup>*

In diesem Bericht stelle ich die von mir während meines Forschungsfreisemesters an der University of Wisconsin – Madison zum Zweck meiner Weiterbildung durchgeführten Arbeiten und erzielten Ergebnisse vor.

Meinen Fachkollegen an der THI bin ich zu tiefem Dank dafür verpflichtet, dass sie während meines Forschungsfreisemesters meine Lehrveranstaltungen übernommen und mir damit die Forschungsarbeiten in Madison ermöglicht haben. Ferner danke ich der Fakultät für Maschinenbau und der Hochschulleitung für die Gewährung des Freisemesters.

Mein besonderer Dank gilt meinem Kollegen Prof. James B. Rawlings am Department of Chemical and Biological Engineering der University of Wisconsin – Madison für die freundliche Aufnahme in seine Forschungsgruppe, seine großzügige fachliche Unterstützung und Offenheit. Die Diskussionen mit ihm waren stets sehr interessant und hilfreich. Außerdem danke ich den Doktoranden (graduate students) Michael Risbeck und Nishith Patel für ihre Vorarbeiten, auf denen ich gut aufbauen konnte, und ihre wertvolle Unterstützung während meines Aufenthalts.

Wie im Folgenden ausgeführt, erstellte ich Simulationsmodelle mit Hilfe der weit verbreiteten Systemdesignsoftware LabVIEW und konnte mich dabei in dieses komplexe Programmiersystem einarbeiten. Damit kann ich dieses zukünftig im Messtechnikpraktikum an der THI einsetzen.

Außerdem führte ich Fallstudien zur modellprädiktiven Regelung durch, wobei ich die Programmiersprache MATLAB verwendete. Neben dem Kennenlernen dieser modernen Regelungsmethode konnte ich dabei meine Erfahrungen mit MATLAB vertiefen und neue Funktionen dieser weltweit eingesetzten Software kennenlernen. Somit war das Forschungssemester für meine zukünftige Arbeit an der THI äußerst lohnend.

Ingolstadt, im November 2014, Wolfgang Krämer

---

<sup>1</sup> Verfasser unbekannt.

## Inhalt

Vorwort .....	1
1 Einleitung.....	3
2 LabVIEW-Simulationsprogramme mit MATLAB-Skript-Nodes.....	4
2.1 Rechnerkonfiguration und Softwarewerkzeuge.....	4
2.2 Beschreibung eines Simulationsprogramms .....	7
3 Fallstudien zur modellprädiktiven Regelung (MPC) .....	13
3.1 Einführung in lineare MPC .....	13
3.2 Regelung eines Rührkesselreaktors .....	17
3.2 Regelung eines elastischen Antriebs .....	22
3.3.1 Lageregelung mit Servointegrierer .....	23
3.3.2 Lageregelung mit Störgrößenschätzung.....	27
3.3.3 Vergleich .....	31
3.4 Fazit .....	31
Literaturhinweise .....	32
Anhang A: MATLAB-Funktionen für die Wassersäulen-Simulation .....	33
Anhang B: Parameters for Standpipe Simulation .....	35

# 1 Einleitung

Dieser Bericht besteht aus zwei voneinander ziemlich unabhängigen Teilen:

- Im nächsten Kapitel wird über die Entwicklung von Echtzeit-Simulationsprogrammen für den Einsatz im regelungstechnischen Praktikum am Department of Chemical and Biological Engineering der University of Wisconsin - Madison berichtet. Dabei kommen die bekannte Systemdesignsoftware LabVIEW und die Programmiersprache MATLAB zum Einsatz. Insbesondere werden eine Programmstruktur und Regeln für die intensive Benutzung von MATLAB-Skript-Nodes für die Echtzeitberechnungen vorgestellt.
- In Kapitel 3 werden Fallstudien zur modellprädiktiven Regelung (Model Predictive Control, MPC), einem modernen Verfahren der Regelungstechnik, durchgeführt. Als Regelstrecken werden je ein typisches Beispiel aus dem Maschinenbau bzw. der Antriebstechnik und aus der Verfahrenstechnik verwendet. Es wird insbesondere untersucht, wie sich die Möglichkeit, Begrenzungen bei der Formulierung der Regelungsaufgabe direkt zu berücksichtigen, auf das Regelverhalten auswirken. Diese Möglichkeit ist ein Spezifikum von MPC.

## 2 LabVIEW-Simulationsprogramme mit MATLAB-Skript-Nodes

Um die zunehmende Zahl von Bachelorstudierenden (undergraduate students) am Department of Chemical and Biological Engineering der University of Wisconsin - Madison zu bewältigen, muss das Labor für das regelungstechnische Praktikum ausgebaut werden. Aus Raum- und Kostengründen sollen jedoch keine weiteren Versuche aufgebaut werden, sondern es sollen Simulationen erstellt werden, die das Verhalten der realen Versuchsaufbauten möglichst realistisch nachbilden.

Für folgende Versuchsaufbauten, die in [1] beschrieben sind, werden Simulationen erstellt:

- Zwei-Tank-System bestehend aus zwei übereinander angeordneten, hintereinander geschalteten Behältern. Dieses Experiment wird für die Untersuchung von dynamischen Systemantworten verwendet.
- Zwei hintereinander geschaltete Rührkesselreaktoren, in denen Malachitgrün mit Natronlauge reagiert. Der Versuchsaufbau dient der Anwendung von Identifikationsverfahren.
- Hoher Wasserbehälter (Wassersäule) für die Entwicklung einer Füllstandregelung.
- Kontinuierlicher beheizbarer Rührkessel für kombinierte Temperatur- und Füllstandregelung (Mehrgrößenregelung).

### 2.1 Rechnerkonfiguration und Softwarewerkzeuge

Bei den vorhandenen Praktikumsversuchen erfolgt die Datenerfassung, Steuerung und Regelung mit Hilfe von Rechnern, die LabVIEW-Programme ausführen. LabVIEW ist ein weit verbreitetes Programmiersystem der Firma National Instruments für Personal Computer zur Datenerfassung, Steuerung und Regelung, s. z.B. [2], [3]. LabVIEW-Programme werden als virtuelle Instrumente (Virtual Instruments, VIs) bezeichnet. Sie bestehen aus einer Benutzeroberfläche, Frontpanel genannt, und einem Blockschaltbild, in dem die Programmfunktionen auf grafische Weise in Form von Signalflussplänen programmiert werden.

Bei den Praktikumsversuchen erfolgt die Erfassung von analogen Messsignalen und die Ausgabe von analogen Stell- und Steuersignalen an die Versuchsaufbauten über ein Interface, das A/D- und D/A-Wandler enthält und mit dem

Rechner über eine serielle Schnittstelle kommuniziert. Bild 2.1 zeigt die mess- und regelungstechnische Konfiguration der vorhandenen Versuche.

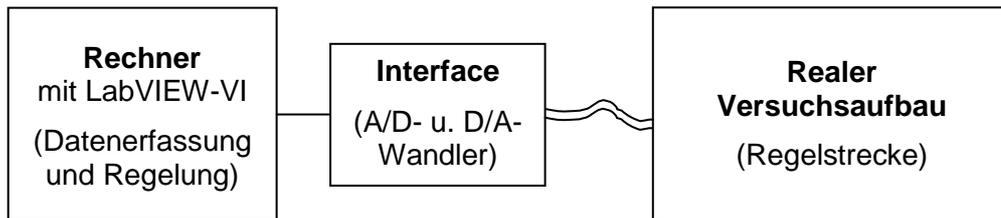


Bild 2.1: Mess- und regelungstechnische Konfiguration der vorhandenen Laborversuche

Um eine hohe Realitätsnähe der neuen Versuche zu erreichen, bei denen das Verhalten des Versuchsaufbaus simuliert wird, sollen für die Datenerfassung, Steuerung und Regelung die gleichen VIs wie bei den realen Versuchsaufbauten verwendet werden. Außerdem sollen analoge Mess- und Stellsignale über das Interface eingelesen bzw. ausgegeben werden, um etwaige Effekte der A/D- und D/A-Wandlungen und der analogen Signalübertragung zu berücksichtigen. Damit bietet es sich an, einen zweiten Rechner für die Simulation des Versuchsaufbaus sowie ein zweites Interface zu verwenden. Die Konfiguration der neuen Versuche zeigt Bild 2.2.

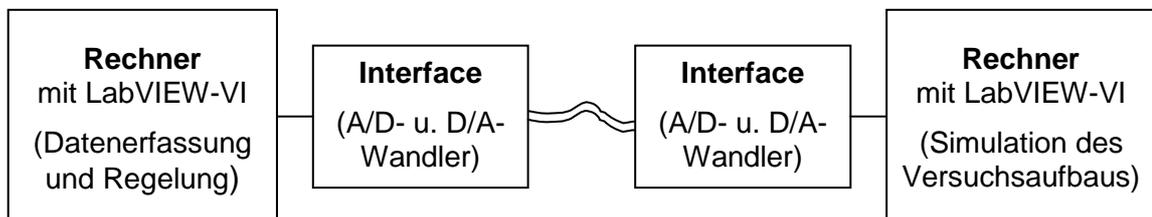


Bild 2.2: Rechnerkonfiguration der neuen Versuche

Wie in Bild 2.2 angegeben, werden auch die Simulationsprogramme als LabVIEW-VIs erstellt, um die Anzahl der verwendeten Softwarewerkzeuge in Grenzen zu halten. Wie oben erwähnt, werden die VIs in Form von Blockschart- bzw. Signalfussbildern grafisch programmiert. Blockschartbilder werden bei der Programmierung von komplexeren Systemen, wie sie die Modelle der Versuchsaufbauten darstellen, schnell unübersichtlich. Deshalb wird von der Möglichkeit Gebrauch gemacht, in LabVIEW sog. MATLAB-Skript-Nodes zu verwenden, in denen MATLAB-Skripte programmiert werden können. Diese Kombination von LabVIEW und MATLAB hat folgende Vorteile:

- Ansprechende Benutzeroberflächen können in Form von LabVIEW-Frontpanels einfach erstellt werden.
- Es ergeben sich übersichtliche LabVIEW-Blockschartbilder.

- Bewährte und wohlbekannte MATLAB-Funktionen, z.B. für numerische Integration, können verwendet werden
- Mess- und Stellsignale werden auf beiden Rechnern mit Hilfe entsprechender LabVIEW-Funktionen und der Interfaces auf einheitliche Weise eingelesen bzw. ausgegeben.

Außerdem gibt es in der Forschungsgruppe von Prof. Rawlings viel Knowhow zur MATLAB-Programmierung.

Bei der Erstellung der Simulationsprogramme hat es sich als sinnvoll erwiesen, folgende Regeln zu beachten:

- Für die Initialisierung und für die Durchführung der Simulation wird jeweils ein MATLAB-Skript-Node verwendet.
- Berechnungen werden so weit wie möglich in den MATLAB-Skript-Nodes durchgeführt. Dies ergibt übersichtliche LabVIEW-Blockschaltbilder.
- Die Simulationsberechnungen werden in einer MATLAB-Funktion (m-File) zusammengefasst. Im entsprechenden Skript-Node wird nur diese Funktion aufgerufen. Die MATLAB-Funktion kann damit zunächst ohne die Verwendung von LabVIEW entwickelt und getestet werden.
- Eingangsgrößen der MATLAB-Funktion werden in Arrays zusammengefasst. Die MATLAB-Funktion liefert ihre Ausgangsgrößen zusammengefasst in Clustern, die bei Bedarf im LabVIEW-Blockschaltbild in ihre Komponenten zerlegt werden. Dies führt zu einem kompakten Aufruf der MATLAB-Funktion.

Die letzten beiden Regeln erhöhen auch die Zuverlässigkeit der Simulationsprogramme. Wenn umfangreiche Berechnungen direkt in Skript-Nodes durchgeführt werden, treten immer wieder unverständliche Laufzeitfehler auf; ob dies ein generelles Problem oder auf die Programminstallation zurückzuführen ist, ist unklar.

Abschließend sei noch bemerkt:

- Die Simulationsberechnungen müssen selbstverständlich in Echtzeit durchgeführt werden.
- Bei den Simulationen wird Mess- und Prozessrauschen modelliert, um eine möglichst hohe Realitätsnähe zu erreichen.

## 2.2 Beschreibung eines Simulationsprogramms

In diesem Abschnitt wird beispielhaft das Simulationsprogramm des Versuchsaufbaus „Wassersäule“ beschrieben, an dem der grundsätzliche Aufbau aller erstellten Simulationsprogramme gut erkennbar ist<sup>2</sup>. Dieser Versuchsaufbau ist in Bild 2.3 schematisch dargestellt; er wird im Praktikum für die Entwicklung einer Füllstandregelung verwendet.

In Abhängigkeit des „Valve input signals“ wird der Zufluss in das „Standpipe“ über das „Control valve“ stetig eingestellt. Der Abfluss erfolgt über das „Normal outflow valve“ und über das „Step disturbance valve“; hierbei handelt es sich um manuelle Schaltventile, die entweder völlig geöffnet oder völlig geschlossen werden können. Der Füllstand wird über den „Level sensor“, einen Drucksensor erfasst und als „Sensor output signal“ ausgegeben. Bei beiden Signalen handelt es sich um elektrische Spannungen.

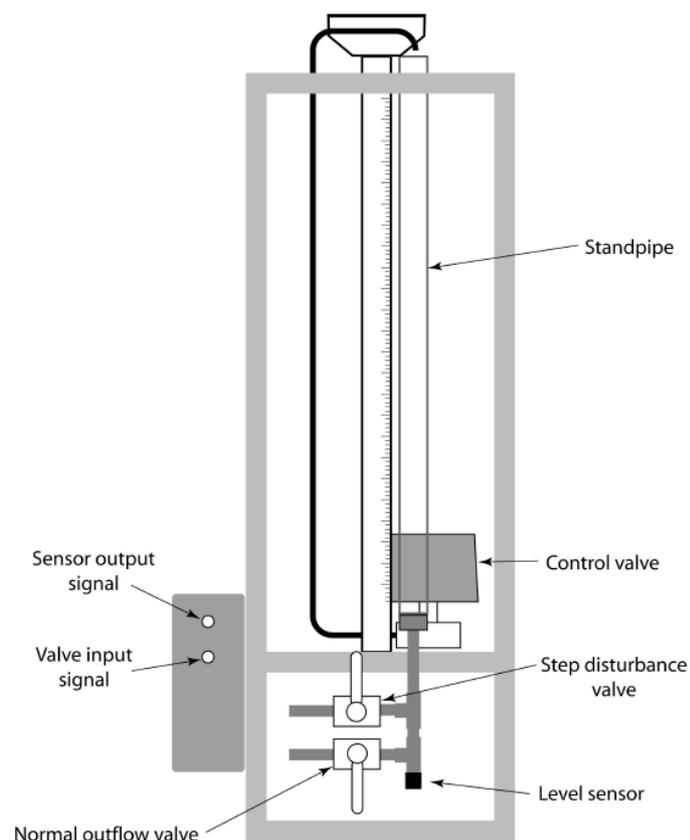


Bild 2.3: Schematische Darstellung der Wassersäule aus [1]

<sup>2</sup> Für alle Simulationen wurden LabVIEW-Frontpanels und –Blockschaltbilder ohne MATLAB-Skript-Nodes von M. Risbeck erstellt. Meine Aufgabe bestand in der Entwicklung der Programmstruktur mit Skript-Nodes, der Umsetzung der Berechnungen in Skript-Nodes und der Parametrierung der Modelle.

Das Verhalten des Füllstands  $h$  wird durch die Dgl.

$$A \frac{dh}{dt} = q_i - C_d a \sqrt{2gh} \quad (2.1)$$

beschrieben. Dabei ist  $A$  die Querschnittsfläche des "Standpipes" und  $q_i$  der Zufluss;  $C_d$  und  $a$  sind der Durchflusskoeffizient bzw. die Querschnittsfläche der Abflussventile;  $g$  ist die Erdbeschleunigung. Der zweite Summand der rechten Seite stellt offensichtlich den Abfluss nach dem Torricellischen Ausflussgesetz dar.

Die Verzögerung des Zuflussventils wird berücksichtigt, indem der Zufluss als Lösung der Dgl.

$$T_1 \dot{q}_i + q_i = q_{i,ss} \quad (2.2)$$

ermittelt wird, wobei der stationäre Zufluss  $q_{i,ss}$  rein proportional zur vorgegebenen Ventilspannung ist; d.h. der Zufluss folgt der Ventilspannung mit einer Verzögerung erster Ordnung. Bei der Simulation werden die beiden Dgln. numerisch gelöst.

Bild 2.4 zeigt das LabVIEW-Frontpanel der Wassersäulensimulation. Rechts befindet sich – wie in allen erstellten Frontpanels – ein Foto des realen Versuchsaufbaus, in dem die wesentlichen Komponenten bezeichnet sind. Die große Balkenanzeige stellt den Füllstand der Wassersäule dynamisch dar. Die kleine Balkenanzeige zeigt den Zufluss, der an der realen Anlage am „Rotameter“ abgelesen werden kann. Die entsprechenden Werte werden auch numerisch angegeben. Mit Hilfe der Schaltflächen „Disturbance Outflow Valve“ und „Normal Outflow Valve“ können die Abflussventile geöffnet und geschlossen werden. Über die Schaltfläche „Extra Noise“ kann das Messrauschen vergrößert werden; dies erfolgt beim realen Experiment durch Verlegen der Zuflussleitung, so dass das Wasser in den Behälter „stärker hineinplätschert“.

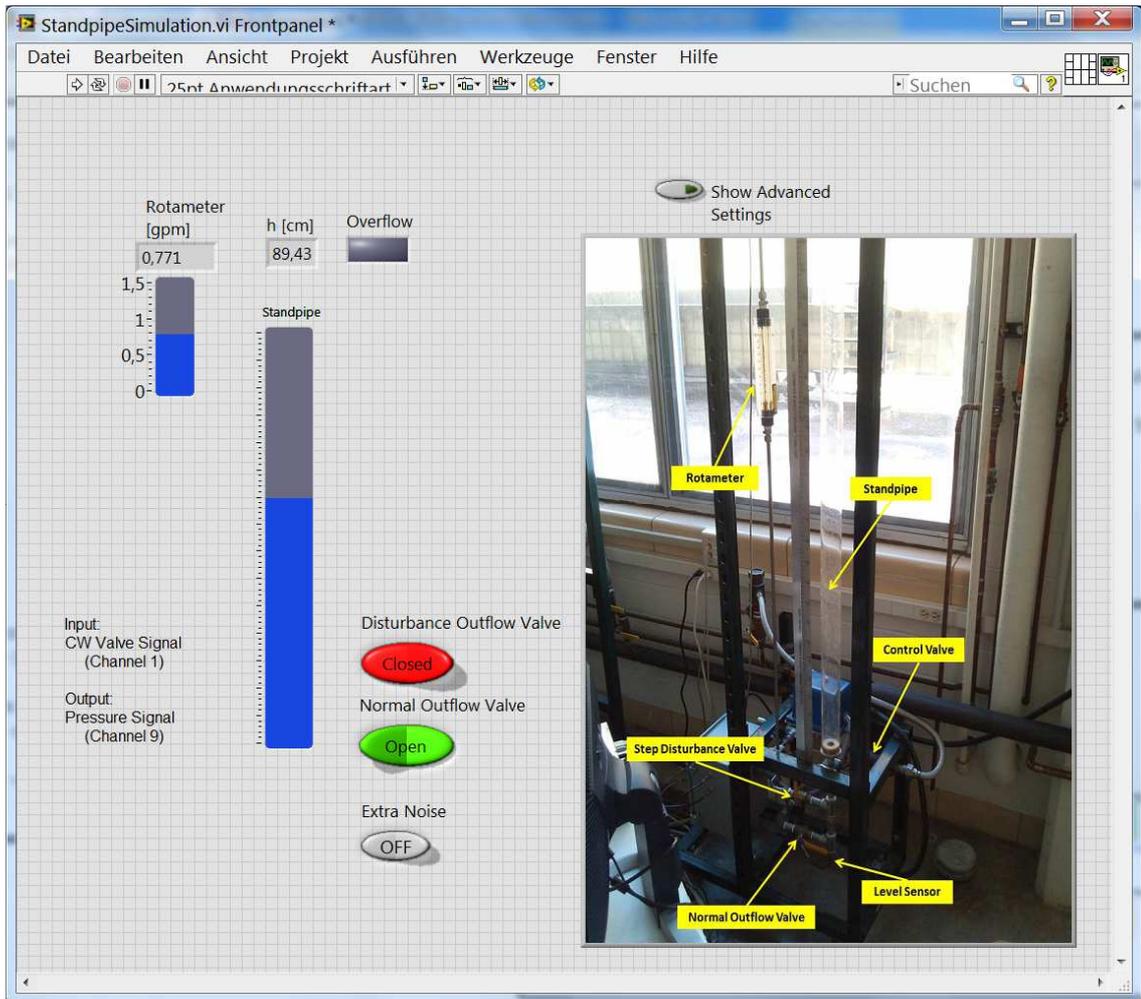


Bild 2.4: Frontpanel der Wassersäulensimulation.

Mit der Schaltfläche „Show Advanced Settings“ können bei allen Simulationen hinter dem Foto verborgene Einstelloptionen und Anzeigen eingeblendet werden, s. Bild 2.5. Bei der Wassersäulensimulation kann mit der Schaltfläche „Manual Control“ und dem Drehknopf „Inlet Valve Voltage“ die Eingangsspannung des Stellventils direkt, ohne die Verwendung eines Steuerprogramms, vorgegeben werden. Der „Time Warp“ erlaubt die Beschleunigung oder Verlangsamung der Simulation für Test- und Entwicklungszwecke. Die Anzeigefläche „Running Slow“ zeigt an, ob die Berechnungen in Echtzeit ausgeführt werden. Dauern die Berechnungen eines Zeitschritts zu lange, wird die Anzeigefläche rot. Über die Schaltfläche „A/D Boxes“ kann das Ansprechen des Interfaces ein- bzw. abgeschaltet werden.

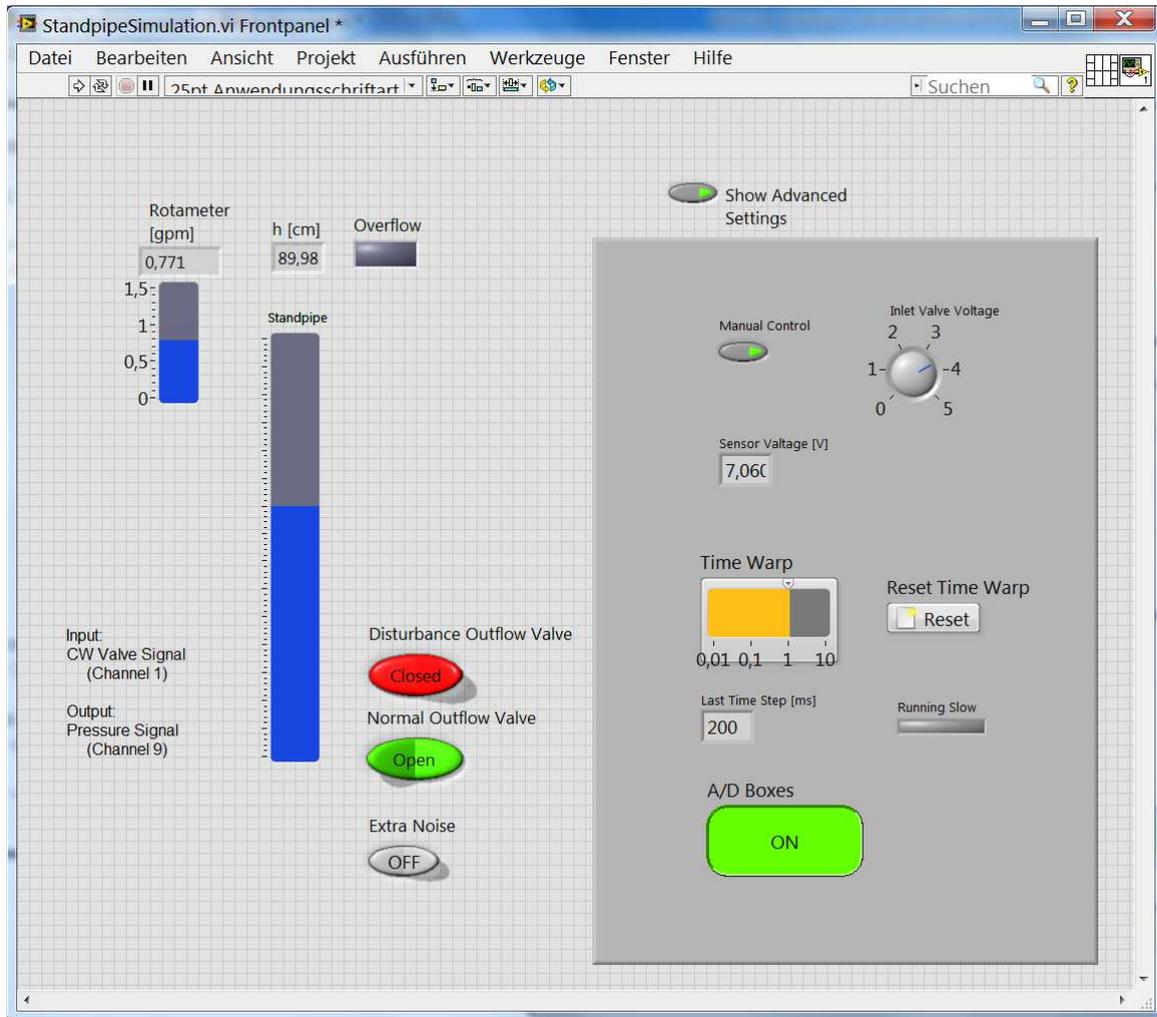


Bild 2.5: Frontpanel der Wassersäulensimulation mit eingeblendeten „Advanced Settings“

Bild 2.6 zeigt das Blockschaltbild des VIs. Im linken Matlab-Skript-Node werden die Simulationsberechnungen initialisiert. Insbesondere werden alle konstanten Parameter definiert und in der Struktur `param` zusammengefasst. Alle in diesem Node definierten Variablen werden im MATLAB-Workspace gespeichert und über diesen an den zweiten Skript-Node übergeben.



Sämtliche Berechnungen während der Simulation, d.h. alle Echtzeitberechnungen, erfolgen in der großen while-Schleife, deren Ausführung über den Wait-Block („Armbanduhr“) gesteuert wird, so dass die Berechnungen regelmäßig im Abstand der im Block „Time Step“ definierten Zeitschrittweite wiederholt werden.

Die Echtzeitberechnungen erfolgen in der MATLAB-Funktion `Standpipe_calc`, die im zweiten Skript-Node aufgerufen wird. Die Eingangsgrößen dieser Funktion werden im Array `inputs` zusammengefasst; sie werden zum größten Teil über die Einstellinstrumente des Frontpanels vorgegeben. Die Ausgangsgrößen von `Standpipe_calc` sind im Cluster `outputs` zusammengefasst, das nach Durchführung der Berechnungen in seine Komponenten zerlegt wird. Diese werden überwiegend am Frontpanel angezeigt.

Im linken unteren Teil der while-Schleife wird überprüft, ob die Simulationsrechnungen innerhalb der festgelegten Zeitschrittweite „Time Step“ erfolgen. Die Ergebnisse werden in den „Advanced Settings“ angezeigt.

Anhang A enthält ein ausführlich kommentiertes Listing der Funktion `Standpipe_calc`. In dieser Funktion werden zunächst die Eingangsgrößen aufbereitet und ggf. mit Rauschen beaufschlagt. Die Integration der Dgln. erfolgt mit Hilfe der Funktion `ode45`, d.h. es wird ein Runge-Kutta-Verfahren vierter Ordnung mit Schrittweitensteuerung verwendet, das für die Lösung nicht steifer Differentialgleichungssysteme geeignet ist. Die Ableitungen, die `ode45` numerisch integriert, werden in der Funktion `Standpipe_DE` berechnet. Zum Schluss werden die Ausgangsgrößen bestimmt.

Die Werte der konstanten Parameter, die bei der Initialisierung gesetzt werden, sind – wie für alle Simulationsprogramme – in einem Parameterdokument zusammengefasst. In diesem ist auch beschrieben, wie die Parameter bestimmt wurden, so dass der Ursprung der Werte nachvollziehbar ist bzw. sie verbessert werden können, falls die Simulationsergebnisse deutlich von den Ergebnissen der realen Versuchsaufbauten abweichen. Das Parameterdokument befindet sich in Anhang B.

### 3 Fallstudien zur modellprädiktiven Regelung (MPC)

#### 3.1 Einführung in lineare MPC

Die modellprädiktive Regelung (Model Predictive Control, MPC) ist ein modernes Verfahren der Regelungstechnik, das insbesondere in der chemischen Industrie erfolgreich eingesetzt wird. Es wird im Folgenden kurz vorgestellt, soweit es für die Fallstudien erforderlich ist; die Darstellung folgt [4].

Bild 3.1 zeigt schematisch eine vollständige modellprädiktive Regelung mit dem Regler (regulator), der Zustandsschätzung (estimator), der Sollwertvorgabe (target selector) sowie der Regelstrecke (plant). Es handelt sich um eine zeitdiskrete Regelung.

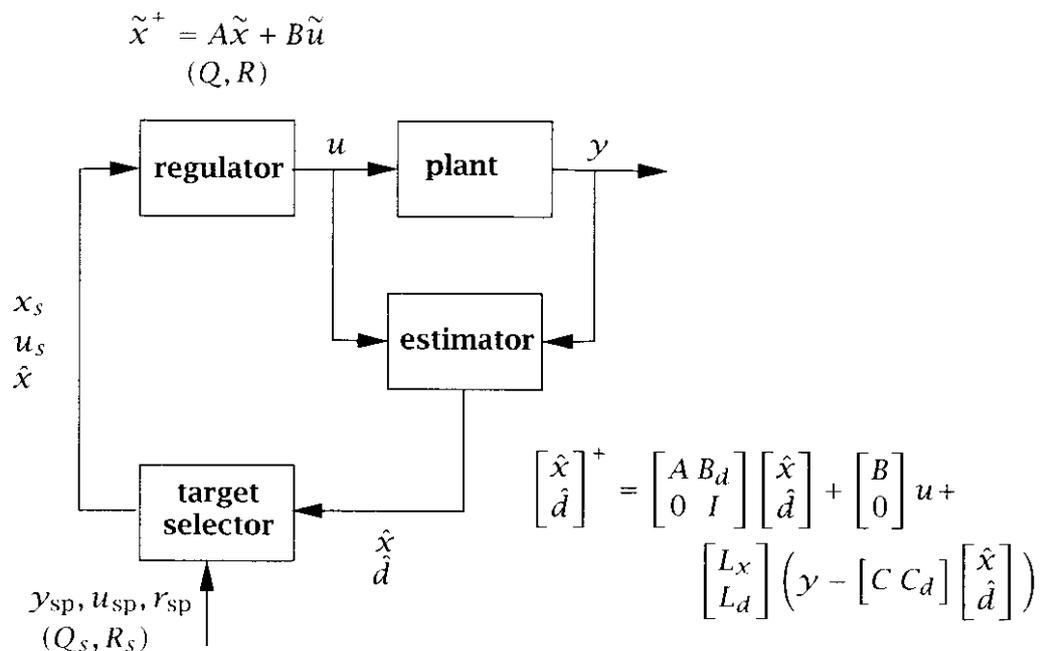


Bild 3.1: Modellprädiktive Regelung aus [4]

#### Der Regler

Ausgangspunkt für den Entwurf eines linearen MPC-Reglers ist die lineare zeitdiskrete Beschreibung der Regelstrecke in Zustandsraumbeschreibung

$$x(k+1) = A \cdot x(k) + B \cdot u(k), \quad k = 0, 1, 2, \dots \quad (3.1)$$

und

$$y(k) = C \cdot x(k) \quad (3.2)$$

mit der Anfangsbedingung  $x(0) = x_0$ . Dabei sind wie üblich  $x \in R^n$  der Zustandsvektor,  $u \in R^m$  der Stellgrößenvektor,  $y \in R^p$  der Ausgangsgrößenvektor oder Vektor der gemessenen Größen,  $A \in R^{n \times n}$  die Systemmatrix,  $B \in R^{n \times m}$  die Eingangsmatrix und  $C \in R^{p \times n}$  die Ausgangsmatrix.

Aufgabe des Reglers ist, den Anfangszustand  $x(0)$  in den Endzustand null zu überführen, so dass das quadratische Kriterium

$$V = \frac{1}{2} \sum_{k=0}^{N-1} [x(k)' Q x(k) + u(k)' R u(k)] + \frac{1}{2} x(N)' P_f x(N) \quad (3.3)$$

mit den symmetrischen Gewichtsmatrizen  $Q \geq 0$ ,  $R > 0$  und  $P_f \geq 0$  minimal wird; das Matrizenpaar  $(A, Q)$  muss stabilisierbar sein.  $N$  ist der zeitdiskrete Vorhersage- bzw. Prädiktionshorizont. Zusätzlich sollen die Zustands- und Stellgrößenbegrenzungen

$$Du(k) + Gx(k) \leq d, \quad k = 0, \dots, N-1 \quad (3.4)$$

erfüllt werden. Durch diese Begrenzungen wird die Regelungsaufgabe nichtlinear. Die Minimierung des Kriteriums (3.3), so dass die Bedingungen (3.1) und (3.4) erfüllt werden, stellt ein sog. quadratisches Programm dar, das in jedem Abtastzeitschritt gelöst wird.

Ohne Begrenzungen (3.4) ergibt die Lösung des quadratischen Programms eine lineare Zustandsrückführung (optimale Rückführung)

$$u(k) = -K \cdot x(k) \quad (3.5)$$

mit der Rückführmatrix

$$K = (B' \Pi(1) B + R)^{-1} B' \Pi(1) A. \quad (3.6)$$

Dabei wird die Matrix  $\Pi(1)$  als Lösung der rückwärts-Riccati-Iteration

$$\Pi(k-1) = Q + A' \Pi(k) A - A' \Pi(k) B (B' \Pi(k) B + R)^{-1} B' \Pi(k) A, \quad (3.7)$$

$$k = N, N-1, \dots, 2$$

mit der Endbedingung  $\Pi(N) = P_f$  bestimmt. Für einen unendlichen Horizont,  $N \rightarrow \infty$ , ergibt dies den wohlbekannten Riccati-Regler bzw. „Linear Quadratic Regulator (LQR)“. Man erhält den LQR auch, indem man die Matrix  $\Pi$  als Lösung der diskreten algebraischen Riccati-Gl.

$$A' \Pi A - \Pi - A' \Pi B (B' \Pi B + R)^{-1} B' \Pi A + Q = 0 \quad (3.8)$$

bestimmt und anstatt  $\Pi(1)$  in Gl. (3.6) einsetzt. Dieselbe Rückführung ergibt sich bei der oben beschriebenen Iteration unabhängig vom Horizont  $N$ , wenn man als Gewichtsmatrix für die Endwerte der Zustandsgrößen  $P_f = \Pi$  wählt. Dies wird

in [5] empfohlen, um die bekannten guten Eigenschaften des LQR zu erhalten, wenn keine Begrenzungen wirksam sind.

Das quadratische Programm mit Begrenzungen (3.4) wird numerisch gelöst.

### Die Zustandsschätzung

In der Regel sind die  $n$  Zustandsgrößen  $x$  unbekannt; nur die  $p < n$  Ausgangsgrößen  $y$  werden gemessen. Für die Rückführung (3.5) bzw. die Lösung des quadratischen Programms werden deshalb die Zustandsgrößen mit Hilfe eines linearen stationären Kalmanfilters bzw. eines „Linear Quadratic Estimators (LQE)“ geschätzt. Dabei wird davon ausgegangen, dass auf die Regelstrecke Prozess- und Messstörungen  $w$  bzw.  $v$  wirken, so dass sie durch die Gln.

$$x(k+1) = A \cdot x(k) + B \cdot u(k) + w(k), \quad (3.9)$$

$$y(k) = C \cdot x(k) + v(k) \quad (3.10)$$

beschrieben wird. Die Störungen werden als mittelwertfreies weißes Rauschen mit den Kovarianzmatrizen

$$E(w(k) \cdot w(k)') = Q_w, \quad E(v(k) \cdot v(k)') = R_v \quad (3.11)$$

modelliert;  $w$  und  $v$  seien unkorreliert. Die Schätzung erfolgt in zwei Schritten. Im Korrekturschritt werden die Schätzwerte  $\hat{x}(k)$  der Zustandsgrößen  $x$  zum Zeitpunkt  $k$  berechnet:

$$\hat{x}(k) = \hat{x}^-(k) + L(y(k) - C\hat{x}^-(k)). \quad (3.12)$$

Dann erfolgt die Prädiktion der Schätzwerte für den nächsten Zeitpunkt  $k+1$ :

$$\hat{x}^-(k+1) = A\hat{x}(k) + Bu(k) \quad (3.13)$$

Die Korrekturmatrix  $L$  ist

$$L = PC'(CPC'+R)^{-1}, \quad (3.14)$$

mit der Lösung  $P$  der diskreten Riccati-Gl.

$$APA'-P - APC'(CPC'+R)^{-1}CPA'+Q = 0. \quad (3.15)$$

### Die Sollwertvorgabe

Wie oben erwähnt hat der Regler die Aufgabe, die Zustandsgrößen  $x$  und nach Gl. (3.2) auch die Ausgangsgrößen  $y$  zu null zu machen. Falls die Sollwerte  $y_{sp} \neq 0$  sind, werden Werte  $x_s$  und  $u_s$  für die Zustands- bzw. Stellgrößen berechnet, die im stationären Zustand  $y = y_{sp}$  ergeben. Da in den Fallstudien nur die

Störgrößenausregelung betrachtet wird, d.h.  $y_{sp} = 0$  ist, wird auf diese Berechnung hier nicht eingegangen. Der Regler arbeitet dann mit den Abweichungen

$$\tilde{x} = x - x_s \quad \text{und} \quad \tilde{u} = u - u_s \quad (3.16)$$

der Zustands- bzw. Stellgrößen anstatt mit  $x$  und  $u$ . Die stationären Zustands- und Stellgrößen  $x_s$  bzw.  $u_s$  hängen auch von eventuell auftretenden konstanten Störungen ab, wie im Folgenden erläutert wird.

### Berücksichtigung von Störgrößen

Konstante nicht gemessene Störgrößen sollen zu keinen bleibenden Regelabweichungen führen. Solche Störgrößen werden mit zeitdiskreten Integratoren der Form

$$d(k+1) = d(k) + w_d(k), \quad (3.17)$$

modelliert, die durch weißes Rauschen  $w_d$  angeregt werden. Es wird ein Störmodell der Ordnung  $n_d$  angesetzt, d.h.  $d \in R^{n_d}$  und  $w \in R^{n_d}$ . Die Gln. (3.17) werden mit den Zustands- und Ausgangsgln. für die Zustandsschätzung (3.9) bzw. (3.10) kombiniert, so dass sich die erweiterte Zustandsraumbeschreibung

$$\begin{bmatrix} x \\ d \end{bmatrix}^+ = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + w, \quad (3.18)$$

$$y = \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} + v \quad (3.19)$$

ergibt. Entsprechend diesem Vorgehen werden also – im Unterschied zu einer Störgrößenschätzung [6] – immer alle Zustands- und Störgrößen geschätzt. Damit das System detektierbar ist, müssen die Matrizen  $B_d$  und  $C_d$  so gewählt werden, dass

$$\text{Rang} \begin{bmatrix} I - A & -B_d \\ C & C_d \end{bmatrix} = n + n_d \quad (3.20)$$

ist. Daraus ergibt sich, dass

$$n_d \leq p \quad (3.21)$$

sein muss, d.h. die Anzahl der modellierten Störgrößen darf die Zahl der Messgrößen nicht überschreiten.

Es ist offensichtlich, dass mit  $m$  Stellgrößen nicht mehr als  $m$  Sollwerte korrekt eingestellt werden können. Sollte  $m < p$  sein, werden aus den  $p$  Messgrößen  $n_c \leq m$  Regelgrößen

$$r(k) = HCy(k), \quad (3.22)$$

$r \in R^{n_c}$  ausgewählt, die stationär genau auf ihre Sollwerte  $r_{sp}$  eingestellt werden sollen. Hierfür ist es nach [4] notwendig, dass  $p$  Störgrößen geschätzt werden, d.h.  $n_d = p$  ist. Dabei wird keine physikalisch richtige Modellierung der Störungen vorausgesetzt; nur die Bedingung (3.20) muss erfüllt sein. Dies ist günstig, wenn die Wirkungsweise der Störungen unbekannt ist. Können jedoch die Störungen physikalisch richtig modelliert werden, ist auch mit Störmodellen niedrigerer Ordnung  $n_d < p$  eine stationär genaue Störgrößenausregelung möglich. Dafür müssen in der erweiterten Zustandsraumbeschreibung (3.18) und (3.19) die richtigen Störeingangs- und -ausgangsmatrizen  $B_d$  bzw.  $C_d$  verwendet werden.

Mit einem Kalman-Filter für das erweiterte System (3.18), (3.19) können Schätzwerte  $\hat{x}$  und  $\hat{d}$  für die Zustands- und Störgrößen ermittelt werden. Die Werte  $\hat{d}$  werden als Schätzung der stationären Störgrößen betrachtet, d.h.  $\hat{d}_s = \hat{d}$ . Dann treten für  $n_c = m$  mit den stationären Zustands- und Stellgrößen

$$\begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} I - A & -B \\ HC & 0 \end{bmatrix}^{-1} \begin{bmatrix} B_d \hat{d}_s \\ r_{sp} - HC_d \hat{d}_s \end{bmatrix} \quad (3.23)$$

keine bleibenden Abweichungen der Regelgrößen  $r$  auf. Mit den so bestimmten stationären Werten  $x_s$  und  $u_s$  werden die Abweichungen (3.16) berechnet.

### 3.2 Regelung eines Rührkesselreaktors

Als erste Fallstudie wird die modellprädiktive lineare Regelung eines kontinuierlichen Rührkesselreaktors nach [4] betrachtet.

#### Beschreibung der Regelstrecke

Eine schematische Darstellung des Reaktors, in dem eine irreversible Reaktion  $A \rightarrow B$  erster Ordnung in einer Flüssigphase abläuft, zeigt Bild 3.2. Sein Verhalten wird durch folgendes nichtlineare Differentialgleichungssystem beschrieben:

$$\begin{aligned} \frac{dc}{dt} &= \frac{F_0(c_0 - c)}{\pi r^2 h} - k_0 c \exp\left(\frac{-E}{RT}\right), \\ \frac{dT}{dt} &= \frac{F_0(T_0 - T)}{\pi r^2 h} + \frac{-\Delta H}{\rho C_p} k_0 c \exp\left(\frac{-E}{RT}\right) + \frac{2U}{r \rho C_p} (T_c - T), \\ \frac{dh}{dt} &= \frac{F_0 - F}{\pi r^2}. \end{aligned}$$

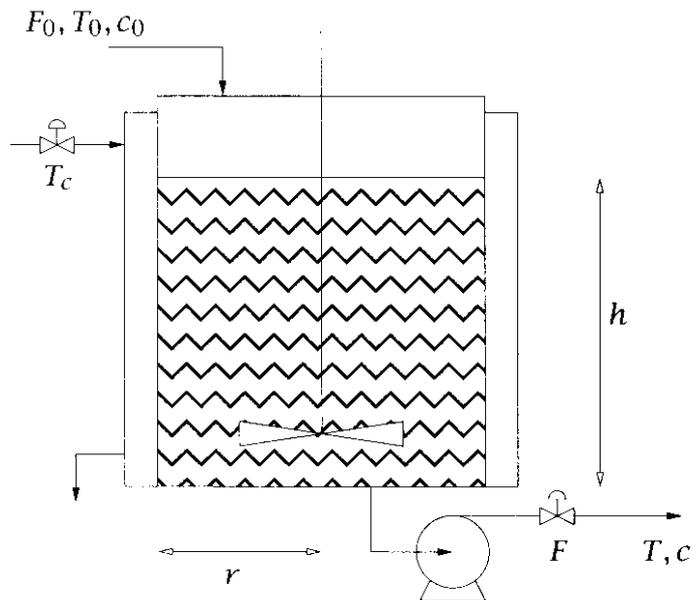


Bild 3.2: Schematische Darstellung des Rührkesselreaktors aus [4]

Die Regelgrößen sind der Füllstand  $h$  und die molare Konzentration  $c$  des Reaktanden A. Die Reaktortemperatur  $T$  ist eine zusätzliche Zustandsgröße. Die Stellgrößen sind die Kühlmitteltemperatur  $T_c$  und der Abflussvolumenstrom  $F$ . Der Zuflussstrom  $F_0$  wirkt als Störung. Die Parameter sind in Tabelle 3.1 angegeben.

Tabelle 3.1: Parameter des Rührkesselreaktors aus [4]

Parameter	Nominal value	Units
$F_0$	0.1	$\text{m}^3/\text{min}$
$T_0$	350	K
$c_0$	1	$\text{kmol}/\text{m}^3$
$r$	0.219	m
$k_0$	$7.2 \times 10^{10}$	$\text{min}^{-1}$
$E/R$	8750	K
$U$	54.94	$\text{kJ}/\text{min} \cdot \text{m}^2 \cdot \text{K}$
$\rho$	1000	$\text{kg}/\text{m}^3$
$C_p$	0.239	$\text{kJ}/\text{kg} \cdot \text{K}$
$\Delta H$	$-5 \times 10^4$	$\text{kJ}/\text{kmol}$

Die Zustandsgrößen haben im stationären Zustand die Werte

$$c_s = 0,87778 \text{ kmol/l}, \quad T_s = 324,5 \text{ K}, \quad h_s = 0,659 \text{ m};$$

die Stellgrößen sind

$$T_{c,s} = 300\text{K}, \quad F_s = 0,1 \text{ m}^3/\text{min}.$$

Alle Zustandsgrößen werden gemessen.

Für die Reglerberechnungen wird ein linearisiertes Modell

$$x(k+1) = Ax(k) + Bu(k) + b_d d(k), \quad y(k) = Cx(k)$$

für die Abweichungen

$$x = y = \begin{bmatrix} c - c_s \\ T - T_s \\ h - h_s \end{bmatrix}, \quad u = \begin{bmatrix} T_c - T_{c,s} \\ F - F_s \end{bmatrix}, \quad d = F_0 - F_{0,s}$$

bestimmt. Bei einer Abtastzeit von  $T_0 = 1$  min sind die Modellmatrizen

$$A = \begin{bmatrix} 0.268, & -0.00338 & -0.00728 \\ 9.703 & 0.3279 & -25.44 \\ 0 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$B = \begin{bmatrix} -0.00537 & 0.1655 \\ 1.297 & 97.91 \\ 0 & -6.637 \end{bmatrix}, \quad b_d = \begin{bmatrix} -0.1175 \\ 69.64 \\ 6.637 \end{bmatrix}.$$

Als Gewichtsmatrizen werden

$$Q = \text{Diag}(1/c_s^2, 1/T_s^2, 1/h_s^2), \quad R = \text{Diag}(1/T_{c,s}^2, 1/F_s^2)$$

gewählt.

### Regelung ohne Begrenzungen

Für die Regelung wird ein linearer MPC nach den Gln. (3.5) – (3.7) mit den angegebenen Gewichtsmatrizen verwendet; es werden keine Begrenzungen berücksichtigt.

Zunächst wird der Einfluss des Vorhersagehorizonts untersucht, indem die Eigenwerte des geschlossenen Regelkreises für verschiedene Horizonte  $N$  berechnet werden. Die Gewichtsmatrix der Endwerte wird der Einfachheit halber  $P_f = Q$  gesetzt; diese Festlegung wird in der Praxis häufig verwendet.

Sowohl die Regelstrecke als auch der geschlossene Regelkreis haben einen reellen Eigenwert und ein konjugiert komplexes Eigenwertpaar. Bild 3.2.a zeigt alle Eigenwerte in der komplexen Zahlenebene, Bild 3.2b zeigt den reellen Eigenwert in Abhängigkeit von  $N$ . Man erkennt, dass die Eigenwerte sehr schnell gegen ihre Grenzwerte für  $N \rightarrow \infty$  konvergieren, so dass schon mit kleinen Horizonten gutes Regelkreisverhalten zu erwarten ist.

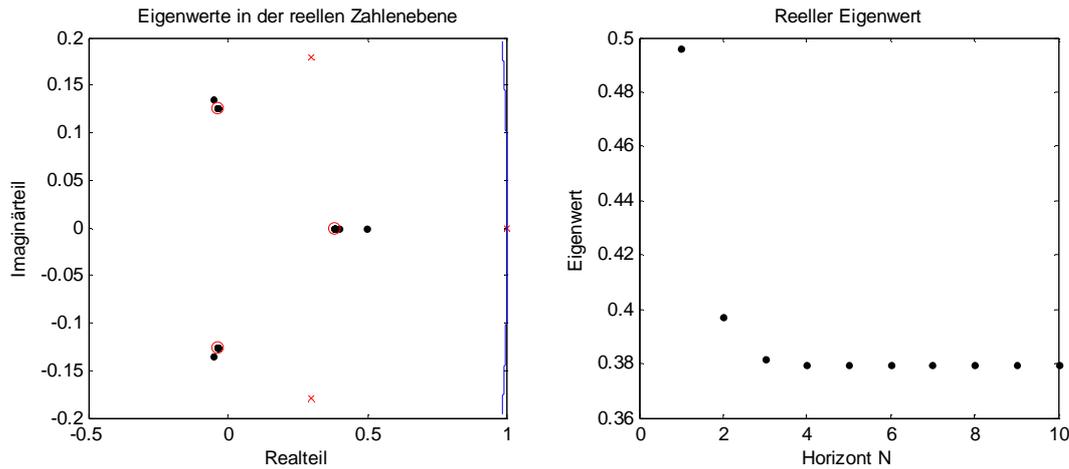


Bild 3.2: Eigenwerte des geschlossenen Regelkreises in Abhängigkeit des Horizonts  $N$ . (×: Eigenwerte der Regelstrecke; ○: Eigenwerte für  $N \rightarrow \infty$ , blau: Einheitskreis)

Zur Untersuchung des Regelverhaltens wird eine Simulation aus [4] nachvollzogen. Dabei wird für die Reglerberechnungen eine in der Forschungsgruppe neu geschriebene MATLAB-Funktion `Impc_matlab` verwendet und getestet, die für die Lösung des oben beschriebenen quadratischen Programms die Funktion `quadprog` aus der Optimization Toolbox verwendet. Als äußere Erregung des Regelkreises wird von einer nicht gemessenen Störung in Form einer Erhöhung des Zuflusses  $F_0$  um zehn Prozent zum Zeitpunkt  $t = 10$  min ausgegangen. Der Horizont des MPC ist  $N = 5$ , als Gewichtsmatrix für den Endzustand wird die Lösung der stationären Riccati-Gl. (3.8), d.h.  $P_f = \Pi$ , verwendet. Ein Störmodell (3.17) der Ordnung  $n_d = 3$  dient der Vermeidung bleibender Regelabweichungen. Damit ist ein Zustandsschätzer erforderlich, obwohl alle Zustandsgrößen der Regelstrecke gemessen werden. Als Störein- und -ausgangsmatrix werden für das erweiterte Modell (3.18)

$$B_d = \begin{bmatrix} 0 & 0 & 0,1655 \\ 0 & 0 & 97,91 \\ 0 & 0 & -6,637 \end{bmatrix} \quad \text{bzw.} \quad C_d = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

gewählt. Die Kovarianzmatrizen sind

$$Q_w = \text{Diag}(\varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, 1) \quad \text{und} \quad R_v = 2,5 \cdot 10^{-7} \text{Diag}(c_s^2, T_s^2, h_s^2)$$

mit  $\varepsilon = 10^{-8}$ ; weitere Details s. [4]. Damit ergibt sich das gleiche Simulationsergebnis wie in [4], womit bestätigt ist, dass die neue Funktion `Impc_matlab` für Regelaufgaben ohne Begrenzungen korrekt arbeitet.

Nun wird die Störung auf 25 % von  $F_0$  erhöht. Alle Parameter der Regelung bleiben unverändert. Es wird verglichen, wie sich Regelkreise mit dem nichtli-

nearen und mit dem linearisierten Modell der Regelstrecke verhalten. Bild 3.3 zeigt das Ergebnis der Simulationen, links den Verlauf der Zustandsgrößen und rechts die Stellgrößenverläufe. Man erkennt, dass die Nichtlinearität der Regelstrecke einen deutlichen Einfluss auf die Konzentration  $c$  sowie auf die Stellgröße Kühlmitteltemperatur  $T_c$  hat. Bei den anderen Größen gibt es geringere Unterschiede zwischen der nichtlinearen und der linearen Simulation. In beiden Fällen haben die Regelgrößen Konzentration  $c$  und Füllstand  $h$  keine bleibenden Abweichungen, d.h. die Störung wird vollständig ausgegletzt, da die Bedingung (3.20) erfüllt und  $n_d = p = 3$  ist.

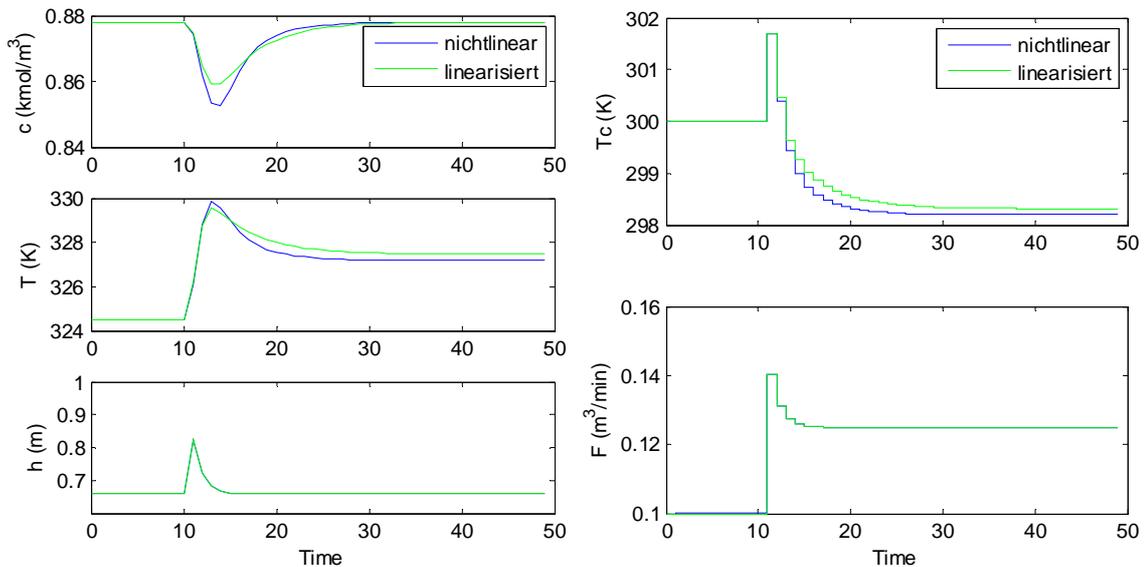


Bild 3.3: Verhalten der Regelung ohne Begrenzung bei einer Störung des Zuflusses um 25 % mit nichtlinearem und linearisiertem Streckenmodell, Zeit in min

### Regelung mit Begrenzungen

Die Regelgrößen  $c$  und  $h$  weichen während des in Bild 3.3 gezeigten Einschwingvorgangs zeitweise erheblich von ihren stationären bzw. Sollwerten ab. Um diese Abweichungen zu verringern, wird versucht, Zustandsgrößenbegrenzungen bei den Reglerberechnungen zu berücksichtigen. Hierfür werden die Ungleichungen (3.4) verwendet; die Funktion `Impc_matlab` wird etwas erweitert. Die Simulationen erfolgen mit dem nichtlinearen Regelstreckenmodell.

Zunächst wird eine Begrenzung  $c \geq c_{\min} = 0,86 \text{ kmol/m}^3$  für die Konzentration vorgegeben. Dies hat keinerlei Auswirkungen auf das Regelkreisverhalten. Der Grund dafür ist, dass der Regler mit den geschätzten Zustandsgrößen arbeitet

und die Schätzwert  $\hat{c}$  der Konzentration deutlich über dem wahren Wert liegt. Auch größere Werte  $c_{\min} < c_s$  führen zu keinem anderen Verhalten.

Sodann wird versucht, die starke Überhöhung des Füllstands durch Vorgabe einer Begrenzung  $h \leq h_{\max} = 0,7 \text{ m}$  zu verringern. Dadurch verändert sich der Maximalwert des Füllstands jedoch nicht. Der maximale Füllstand wird zur Zeit  $t = 11 \text{ min}$  erreicht, d.h. unmittelbar nach dem Eintreten der Störung; er ist die erste Regelabweichung, die auftritt. Dieser Wert kann vom Regler nicht beeinflusst werden, da der Regler erst auf diese Abweichung reagiert. Für  $t > 11 \text{ min}$  ergeben sich mit der Begrenzung allerdings niedrigere Füllstände. Um den maximalen Füllstand zu reduzieren, muss die Abtastzeit verringert werden.

Die Abweichungen der Stellgrößen von ihren stationären Werten sind gering, so dass eine Stellgrößenbegrenzung nicht sinnvoll erscheint.

### 3.2 Regelung eines elastischen Antriebs

Die Lageregelung elastischer Antriebe stellt eine typische Aufgabenstellung der Antriebstechnik dar [6], [7]. Bild 3.4 zeigt einen Zweimassenschwinger als einfachstes Modell eines ungedämpften elastischen Antriebs.

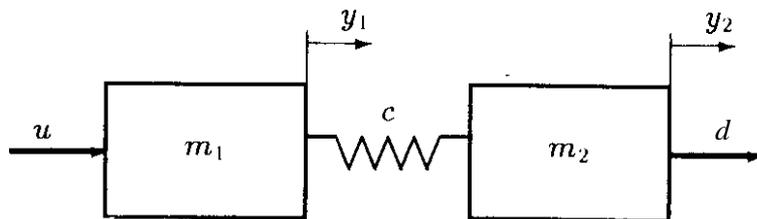


Bild 3.4: Modell eines elastischen Antriebs aus [7]

Bei den Positionen von Motor und Last  $y_1$  bzw.  $y_2$  kann es sich um translatorische oder rotatorische Freiheitsgrade handeln. Dementsprechend sind  $m_1$  und  $m_2$  die Massen oder Trägheitsmomente von Motor bzw. Last. Die Feder mit der Steifigkeit  $c$  beschreibt eine elastische Verbindung von Motor und Last, z.B. durch ein Getriebe. Die Stellgröße  $u$  stellt ein Antriebsmoment oder eine Antriebskraft dar, die Störgröße  $d$  ist ein Lastmoment oder eine Lastkraft.

Das Verhalten des Antriebs wird durch die Bewegungsgln.

$$M\ddot{y} + Ky = b_b u + b_{b,d} d$$

mit der regulären Massen- und der singulären Steifigkeitsmatrix

$$M = \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \quad \text{bzw.} \quad K = \begin{bmatrix} c & -c \\ -c & c \end{bmatrix}$$

beschrieben. Die Eingangsvektoren für die Stell- und die Störgröße sind

$$b_b = [1 \quad 0]' \quad \text{bzw.} \quad b_{b,d} = [0 \quad 1]'$$

Mit dem Zustandsvektor  $x = [y' \quad \dot{y}']'$  ergibt sich die Zustandsraumbeschreibung

$$\dot{x} = Ax + bu + b_d d$$

mit der Systemmatrix und den Eingangsvektoren für die Stell- und Störgröße

$$A = \begin{bmatrix} 0 & I \\ -M^{-1}K & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ b_b \end{bmatrix} \quad \text{bzw.} \quad b_d = \begin{bmatrix} 0 \\ b_{b,d} \end{bmatrix}.$$

Das System hat doppelt integrierendes Verhalten, da  $\det K = 0$  ist.

Alle Zustandsgrößen werden gemessen, d.h.  $C = I$ . Die Regelgröße ist die Lastposition

$$r = y_2 = Hy = Hx$$

mit dem Ausgangsvektor  $H = [0 \quad 1 \quad 0 \quad 0]$ .

Für die folgenden Untersuchungen werden die dimensionslosen Parameter  $m_1 = m_2 = 1$  und  $c = 1/2$  gewählt. Damit ergibt sich die Eigenkreisfrequenz der ungedämpften Schwingung des Antriebs  $\omega = \sqrt{\left(\frac{1}{m_1} + \frac{1}{m_2}\right)} \cdot c = 1$  bzw. die Perio-

dendauer  $T = \frac{2\pi}{\omega} = 2\pi$ .

Für die zeitdiskrete Regelung wird eine Abtastzeit  $T_0 = T/20$ , ein eher kleiner Wert, gewählt und damit eine zeitdiskrete Zustandsraumbeschreibung (3.1), (3.2) berechnet. Als Regelungsaufgabe wird die Ausregelung eines konstanten Lastmoments, d.h. einer konstanten nicht gemessenen Störung, mit einem MPC betrachtet. Um stationäre Genauigkeit der Regelgröße zu gewährleisten, wird entweder ein Servointegrator [6] oder eine Störgrößenschätzung wie in Abschn. 3.1 beschrieben verwendet.

### 3.3.1 Lageregelung mit Servointegrator

Die übliche lineare Rückführung (3.5) der Zustandsgrößen  $x$  wird um eine integrierende Rückführung der Regelgröße  $r$  erweitert, um stationäre Genauigkeit bei konstanten Störgrößen zu erreichen. Dies entspricht dem Vorgehen in der klassischen Regelungstechnik, z.B. bei der Verwendung von PID-Reglern. Die integrierende Rückführung kann nach [6] auch als Störgrößenbeobachter für

konstante Störungen betrachtet werden. Das zeitkontinuierliche Regelgesetz lautet damit

$$u(t) = -kx(t) - k_I \int_0^t r(\tau) d\tau.$$

Der Integrierer des Regelgesetzes wird mit der Zustandsraumbeschreibung der Regelstrecke kombiniert; mit der zusätzlichen Zustandsgröße  $x_I = \int r dt$  ergibt sich das erweiterte Modell

$$\begin{bmatrix} \dot{x}_I \\ \dot{x} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & c_R \\ 0 & A \end{bmatrix}}_{A_e} \underbrace{\begin{bmatrix} x_I \\ x \end{bmatrix}}_{x_e} + \underbrace{\begin{bmatrix} 0 \\ b \end{bmatrix}}_{b_e} u + \underbrace{\begin{bmatrix} 0 \\ b_d \end{bmatrix}}_{b_{d,e}} d$$

Dieses Modell, das nun dreifach integrierendes Verhalten hat, wird diskretisiert und dem Reglerentwurf nach Abschnitt 3.1 zugrunde gelegt, wodurch gleichzeitig die Rückführverstärkungen  $k$  und  $k_I$  bzw. der erweiterte Rückführvektor  $k_e = [k_I \quad k']^T$  für den erweiterten Zustandsvektor  $x_e$  berechnet werden.

Als Gewichtsmatrizen bzw. -faktor für den Reglerentwurf werden

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 10 & -10 \\ 0 & 0 & 0 & -10 & 10 \end{bmatrix} \quad \text{und} \quad R = 0,1$$

gewählt. Von den Zustandsgrößen werden also der Zustand  $x_I$  des Integrierers, die Regelgröße  $r = y_2$  und die Geschwindigkeitsdifferenz  $\dot{y}_1 - \dot{y}_2$  gewichtet.  $x_I$  muss gewichtet werden, damit das Matrizenpaar  $(A_e, Q)$  stabilisierbar ist; die Gewichtung der Geschwindigkeitsdifferenz verbessert die Dämpfung des geschlossenen Regelkreises.

### Regelung ohne Begrenzungen

Zunächst wird der Einfluss des Vorhersagehorizonts  $N$  eines linearen MPC nach den Gln. (3.5) – (3.7) untersucht, indem für  $P_f = Q$  die Eigenwerte des geschlossenen Regelkreises für  $N = 1, 2, \dots, 30$  berechnet werden. Bild 3.5 zeigt das Ergebnis in der komplexen Zahlenebene.

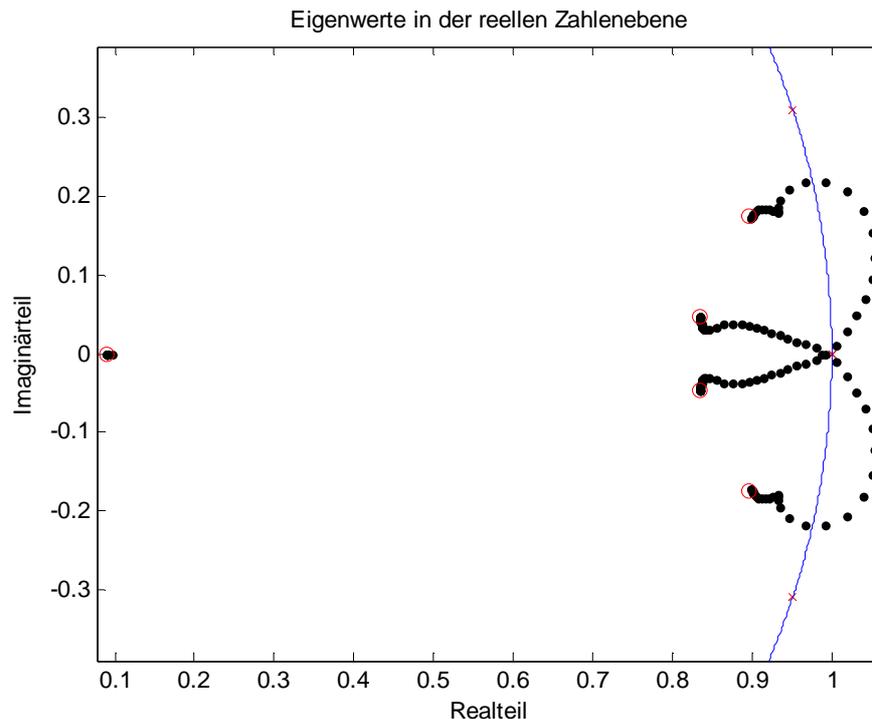


Bild 3.5: Eigenwerte des geschlossenen Regelkreises in Abhängigkeit des Horizonts  $N$ . (x: Eigenwerte der Regelstrecke; o: Eigenwerte für  $N \rightarrow \infty$ , blau: Einheitskreis)

Man erkennt, dass der Regelkreis für kleine Horizonte instabil ist; erst für  $N \geq 11$  ergibt sich ein stabiler geschlossener Regelkreis. Der reelle Pol strebt sehr schnell gegen seinen Grenzwert für  $N \rightarrow \infty$ .

Zur Illustration des Regelverhaltens wird die Störsprungantwort berechnet, d.h. es wird davon ausgegangen, dass die Störgröße eine Einheitssprungfunktion darstellt. Ein Vorhersagehorizont von  $N = 20$  wird gewählt, der der Periodendauer der Schwingung der Regelstrecke entspricht. Die Regelung erfolgt ohne Zustandsschätzung und Sollwertvorgabe; der Sollwert ist null. Als Gewichtsmatrix für den Endzustand wird die Lösung der stationären Riccati-Gl. (3.8), d.h.  $P_f = \Pi$ , verwendet. In Bild 3.6 ist die Störsprungantwort dargestellt, links die Verläufe der Motor- und Lastposition und rechts der Stellgrößenverlauf.

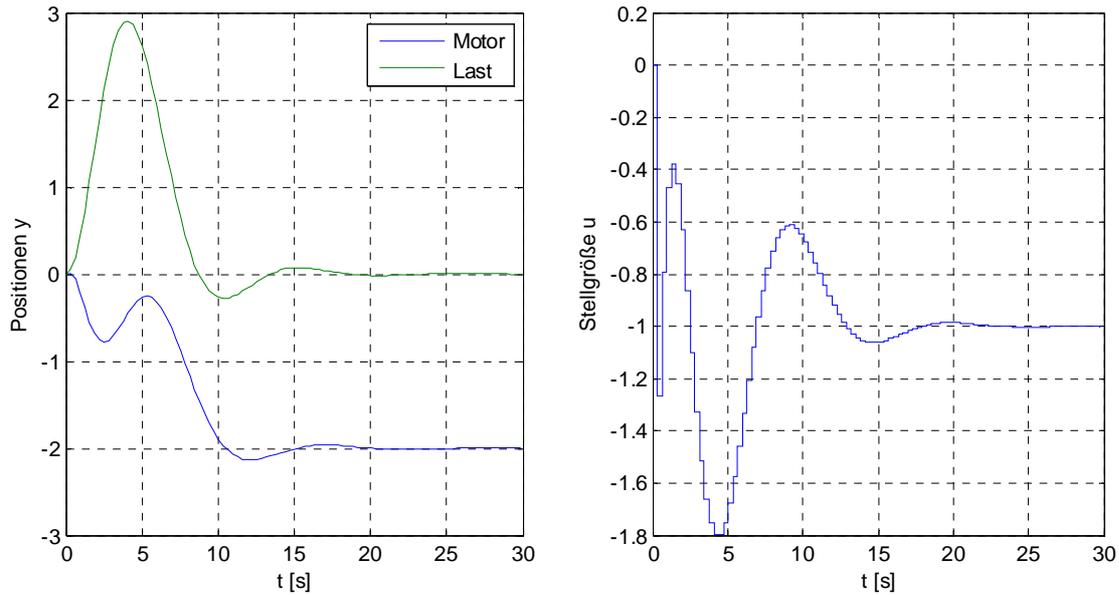


Bild 3.6: Störspungantwort der Regelung ohne Begrenzungen

Die Störspungantwort ist gut gedämpft; die Einbruchtiefe beträgt  $y_{2\max} \approx 2,9$ ; die Ausregelzeit<sup>3</sup> ist  $T_{\text{aus}} = 12,45$  s.

### Regelung mit Begrenzungen

Es wird versucht, die Stellgröße, d.h. das Motormoment mit Hilfe von Ungleichungen (3.4) auf  $|u| \leq u_{\max}$  zu begrenzen. Bild 3.7. zeigt die Störspungantwort für  $u_{\max} = 1,2$ ; dies stellt eine starke Begrenzung dar.

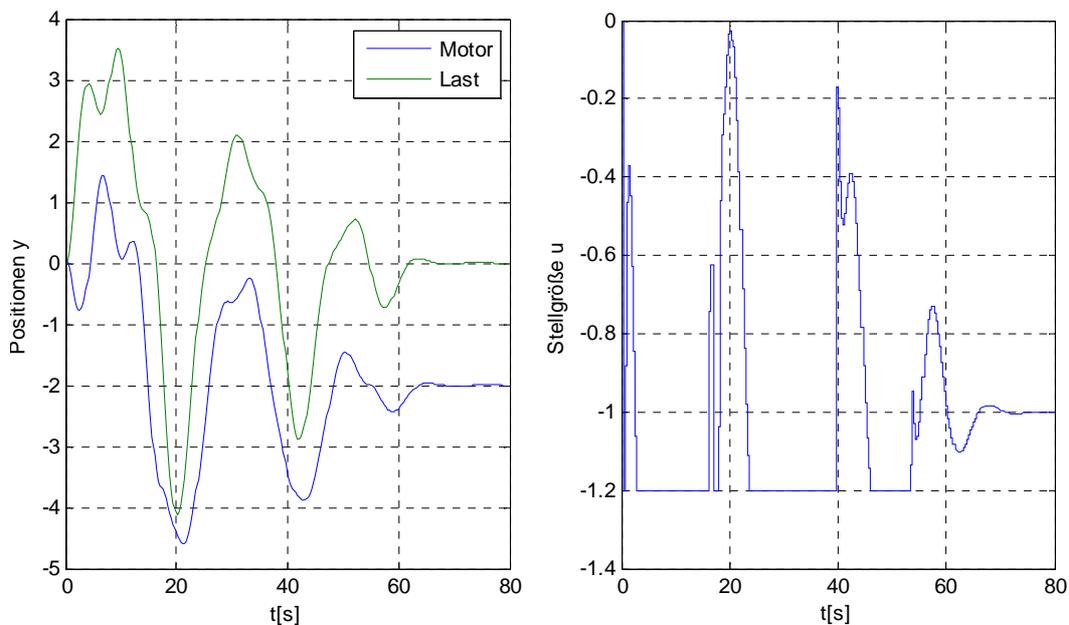


Bild 3.7: Störspungantwort der Regelung mit  $|u| \leq 1,2$

<sup>3</sup> Die Ausregelzeit wird folgendermaßen definiert:  $|y_R(t)| \leq 0,1$  für  $t \geq T_{\text{aus}}$ .

Die Stellgrößenbegrenzung wird exakt eingehalten. Selbstverständlich ergeben sich durch die Begrenzung im Vergleich zum unbegrenzten Fall nach Bild 3.6 eine deutlich größere Einbruchtiefe und Ausregelzeit. Die Stellgröße kann nicht stärker begrenzt werden, da der geschlossene Regelkreis für  $u_{\max} \leq 1,15$  instabil wird. Diese Instabilität ist unabhängig vom Horizont  $N$ . Starke Stellgrößenbegrenzungen können also zu Instabilität führen, so dass bei der Auslegung von Antrieben ausreichend starke Motoren gewählt werden sollten, zumal da maximale Störungen oft nicht genau bekannt sind.

Nun wird versucht, die Regelgröße, d.h. die Lastposition  $r = y_2$ , auf  $|y_R| \leq y_{R\max}$  zu begrenzen. Das wäre eine einfache Möglichkeit, um die Einbruchtiefe zu verringern. Dies gelingt nicht; mit  $y_{R\max} = 2,5$ , d.h. schon mit einer geringen Begrenzung, stellt sich bei der Störsprungantwort ein stabiler Grenzyklus mit großer Amplitude ein. Mit größeren Horizonten, z.B.  $N = 50$ , kann der Grenzyklus vermieden werden, aber auch damit ergibt sich kein akzeptables Regelverhalten.

### 3.3.2 Lageregelung mit Störgrößenschätzung

Um bleibende Regelabweichungen zu vermeiden, wird nun ein Störmodell nach Abschnitt 3.1 verwendet werden. Die Reglerberechnungen basieren auf dem nicht erweiterten Antriebsmodell der Ordnung  $n = 4$ , s. Abschn 3.3. Die Gewichtsmatrizen werden zu

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & -10 \\ 0 & 0 & -10 & 10 \end{bmatrix}, \quad R = 25$$

gewählt. Von den Zustandsgrößen werden also wiederum die Regelgröße  $r = y_2$  und die Geschwindigkeitsdifferenz  $\dot{y}_1 - \dot{y}_2$  gewichtet. Mit diesen Gewichtungen ergibt sich, wie sich noch zeigen wird, eine vergleichbare Störsprungantwort wie mit dem Servointegrator. Bild 3.8 zeigt den Verlauf der Eigenwerte des geschlossenen Regelkreises mit einem MPC in Abhängigkeit des Vorhersagehorizonts  $N$  für  $P_f = Q$ . In diesem Fall ergeben sich für alle Horizonte  $N$  stabile Regelkreise.

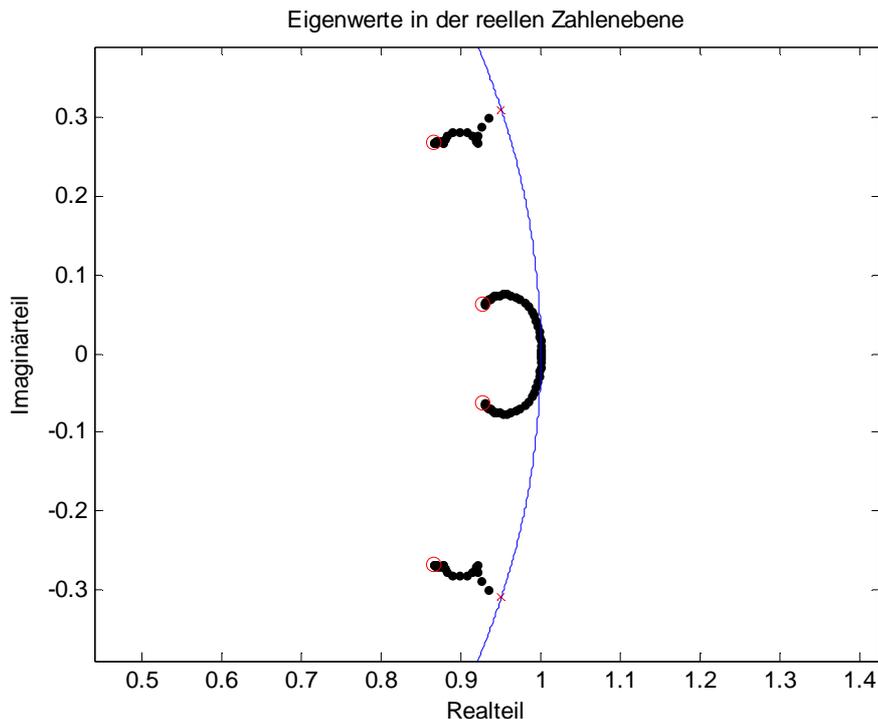


Bild 3.8: Eigenwerte des geschlossenen Regelkreises in Abhängigkeit des Horizonts  $N$ . (x: Eigenwerte der Regelstrecke; o: Eigenwerte für  $N \rightarrow \infty$ , blau: Einheitskreis)

Die Simulation der Störsprungantwort erfolgt wie zuvor mit  $N = 20$ ; als Gewichtsmatrix für den Endzustand wird wiederum die Lösung der stationären Riccati-Gl. (3.8), d.h.  $P_f = \Pi$ , verwendet.

Es wird ein Störmodell (3.18) der Ordnung  $n_d = 1$  mit  $B_d = \bar{b}_d$  und  $C_d = 0$  gewählt; dabei ist  $\bar{b}_d$  der Störeingangsvektor des nicht angegebenen zeitdiskreten Antriebsmodells. Damit wird die Störung physikalisch richtig modelliert und stationäre Genauigkeit der Regelgröße mit  $n_d < p$  erzielt, vgl. die Erläuterungen zu Gl. (3.22). Als Gewichtsmatrizen für das Kalman-Filter für die Zustands- und Störgrößenschätzung werden

$$Q_w = \text{Diag}(\varepsilon, \varepsilon, \varepsilon, \varepsilon, 1) \quad \text{und} \quad R_v = \text{Diag}(\varepsilon, \varepsilon, \varepsilon, \varepsilon)$$

mit  $\varepsilon = 10^{-5}$  gewählt, wobei der Einfluss von  $\varepsilon$  auf das Regelverhalten gering ist.

### Regelung ohne Begrenzungen

Bild 3.9 zeigt die Störsprungantwort des Regelkreises mit MPC ohne Begrenzungen. Sowohl die Verläufe der Positionen und der Stellgröße als auch die Einbruchtiefe und Ausregelzeit sind mit denen der Regelung mit Servointegrierer, Bild 3.6, vergleichbar.

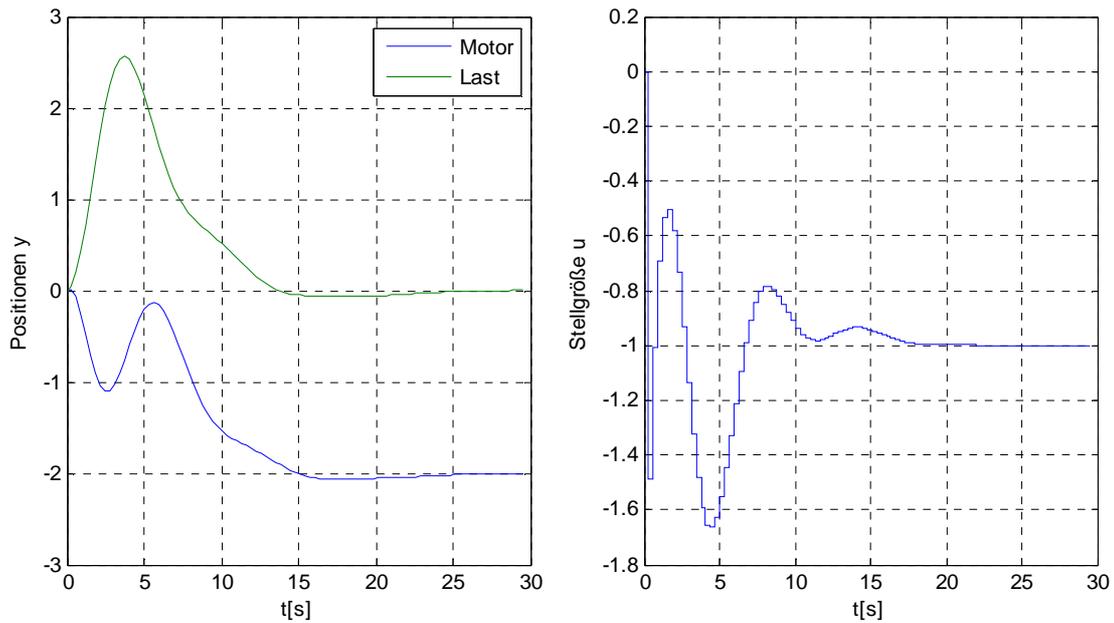


Bild 3.9: Störsprungantwort der Regelung ohne Begrenzungen

### Regelung mit Begrenzungen

Zunächst wird wieder die Stellgröße betragsmäßig auf  $u_{\max} = 1,2$  begrenzt. Bild 3.10 zeigt das sich damit ergebende Regelverhalten. Die Begrenzung führt zu keiner Vergrößerung der Einbruchtiefe und nur zu einer geringfügig verlängerten Ausregelzeit. Das Regelverhalten ist besser als mit dem Servointegrator mit Stellgrößenbegrenzung, vgl. Bild 3.7.

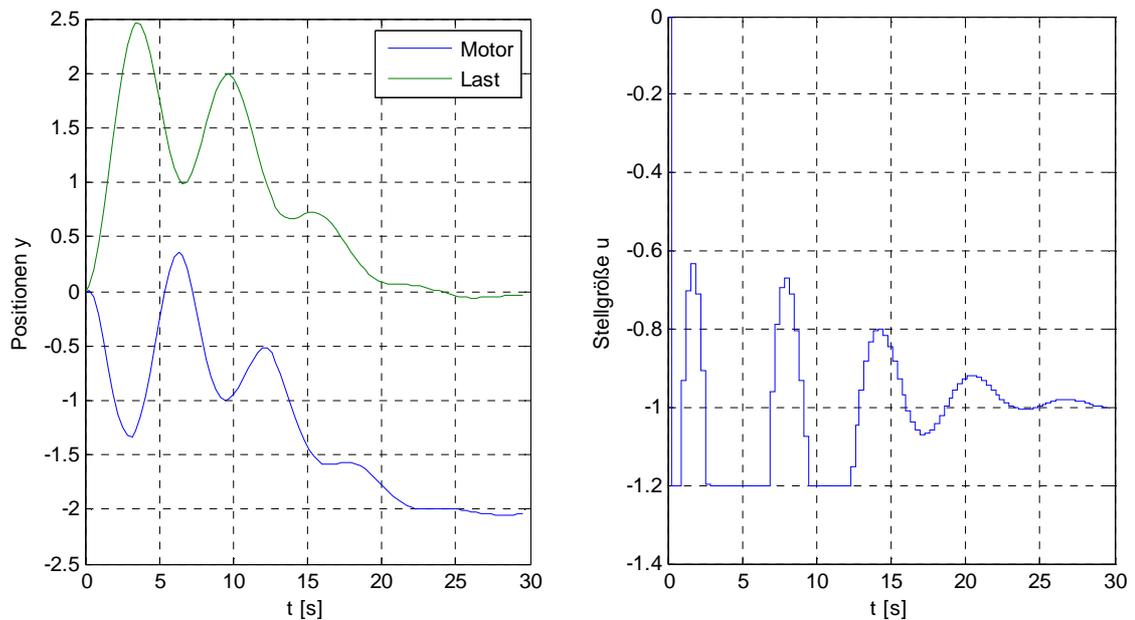


Bild 3.10: Störsprungantwort der Regelung mit  $|u| \leq 1,2$

Es ist möglich, die Stellgröße noch stärker zu begrenzen, ohne dass der Regelkreis instabil wird. Dies ist aber aus praktischer Sicht irrelevant, da sich dann ein viel zu schwach gedämpftes Regelkreisverhalten ergibt.

Nun wird die Regelgröße auf  $|r| \leq r_{\max}$  mit  $r_{\max} = 2$  begrenzt. Wie Bild 3.11 zeigt, ist dies möglich, führt aber natürlich zu größeren Ausschlägen der Stellgröße und zu heftigeren Motorbewegungen. Die Regelgröße kann noch stärker begrenzt werden, womit selbstverständlich auch die negativen Auswirkungen zunehmen. So ergibt sich z.B. für  $r_{\max} = 0,5$  betragsmäßig eine maximale Stellgröße  $|u|_{\max} \approx 20$  und eine maximale Motorposition  $|y_1|_{\max} \approx 4$ . Realistischer Weise kann die Regelgröße also nur in Maßen begrenzt werden.

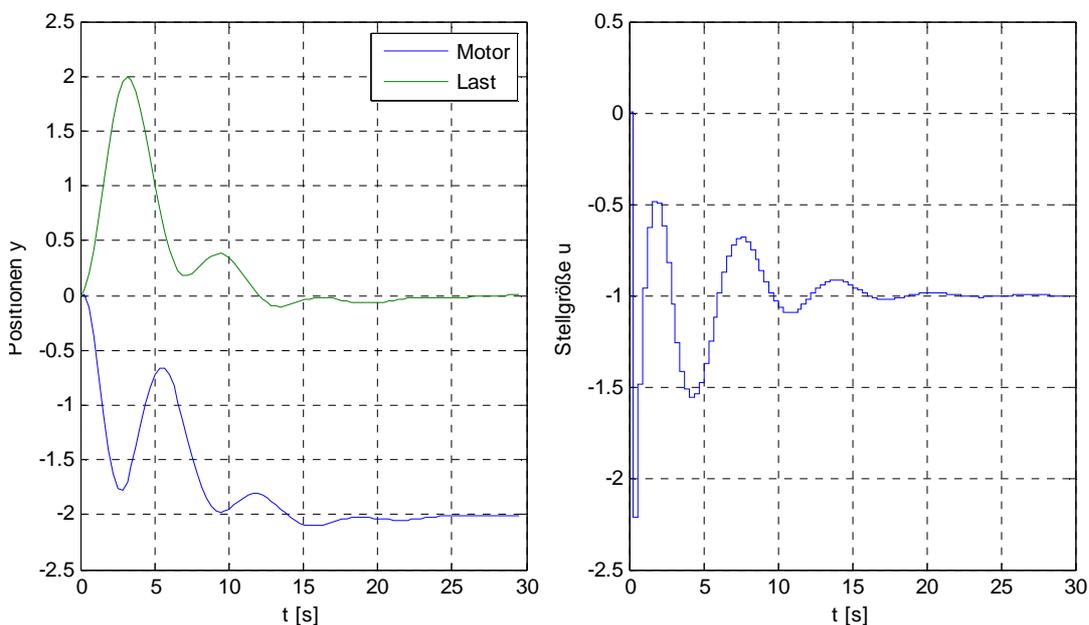


Bild 3.11: Störsprungantwort der Regelung mit  $|r| \leq 2$

Es ist auch möglich, sowohl die Regel- als auch die Stellgröße zu begrenzen. So ergibt sich z.B. für  $r_{\max} = 2$ ,  $u_{\max} = 1,5$  oder für  $r_{\max} = 2,25$ ,  $u_{\max} = 1,2$  jeweils ordentliches Regelverhalten; die Begrenzungen werden eingehalten. Wird allerdings ein Wert der genannten Kombinationen von  $r_{\max}$  und  $u_{\max}$  weiter verringert, wird das quadratische Programm (3.1)-(3.4) unlösbar. Da in der Praxis die maximalen Störgrößen häufig nicht genau bekannt sind, sollte man bei der gleichzeitigen Begrenzung der Stell- und der Regelgröße sehr vorsichtig sein, um dieses Problem zu vermeiden.

### 3.3.3 Vergleich

Vergleicht man die Regelungen mit Servointegrierer und mit Störgrößenschätzung, so scheint für die Regelung des elastischen Antriebs die Störgrößenschätzung vorteilhaft zu sein. Stellgrößenbeschränkungen führen zu einer geringeren Beeinträchtigung des Regelverhaltens, und die Stellgröße kann stärker beschränkt werden, ohne dass der Regelkreis instabil wird. Außerdem kann mit der Störgrößenschätzung auch die Regelgröße beschränkt werden. Selbst eine gleichzeitige Beschränkung der Regel- und der Stellgröße ist möglich, wobei jedoch wie oben erwähnt vorsichtig vorgegangen werden sollte.

### **3.4 Fazit**

Lineare MPC ist eine attraktive Methode der Regelungstechnik. Ihre Besonderheit besteht darin, dass Begrenzungen von Stell- und Zustandsgrößen explizit vorgegeben und berücksichtigt werden können. Wie die Fallstudien zeigen, ist jedoch bei der Formulierung von Begrenzungen, insbesondere von Zustandsgrößenbegrenzungen Vorsicht geboten, da zu restriktive Vorgaben dazu führen können, dass der Regelkreis instabil wird oder dass das quadratische Programm, mit welchem die Stellgrößen berechnet werden, unlösbar wird. Unter Umständen können Zustandsgrößenbeschränkungen unwirksam sein.

Die Fallstudien zeigen ebenfalls, dass das Konvergenzverhalten der Riccati-Iteration (3.7) stark unterschiedlich sein kann. Dieses Verhalten kann Hinweise auf die Wahl des Vorhersagehorizonts  $N$  geben.

Auf neuere Weiterentwicklungen von MPC, insbesondere nichtlineare MPC [8], verteilte MPC [9], hybride MPC und ökonomische MPC [10], bei der sich das Optimierungskriterium nicht auf technische sondern auf wirtschaftliche Größen, wie z.B. den Ertrag einer Chemieranlage, bezieht, wird in diesem Bericht nicht eingegangen.

## Literaturhinweise

- [1] Rawlings, J. B.; M. D. Graham; W. H. Ray: Process Dynamics and Control Laboratory Manual CBE 470, Second Edition. Department of Chemical and Biological Engineering; University of Wisconsin – Madison, 1999.
- [2] Systemdesignsoftware NI LabVIEW. [www.ni.com/labview/d](http://www.ni.com/labview/d), aufgerufen am 10.10.2014.
- [3] Georgi, W.; E. Metin,,: Einführung in LabVIEW, 5. Auflage. München, 2012.
- [4] Rawlings, J. B.; D. Q. Maine: Model Predictive Control: Theory and Design. Nob Hill Publishing, Madison, WI, 2009.
- [5] Rawlings, J. B.; D. Q. Maine: Postface to “Model Predictive Control: Theory and Design”. Nob Hill Publishing, Madison, WI, 2009.
- [6] Juen, G.; Lageregelung elastischer Antriebe dargestellt an einem Radioteleskop. VDI-Verlag, Düsseldorf, 1987.
- [7] Krämer, W.: Lageregelung elastischer Antriebe durch Ausgangsrückführungen. VDI-Verlag, Düsseldorf, 1991.
- [8] Angeli, D.; J. B. Rawlings: Receding Horizon Cost Estimation and Control for Nonlinear Plants. 8th IFAC Symposium on Nonlinear Control Systems (NOLCOS), Bologna, Italy, September 2010.
- [9] Brett, T. S.; S. J. Wright; J. B. Rawlings: Cooperative Distributed Model Predictive Control for Nonlinear Systems. J. Proc. Cont. 21, 2011, S. 698-704.
- [10] Rawlings, J. B.; D. Angeli; C. Bates: Fundamentals of Economic Model Predictive Control. IEEE Conference on Decision and Control (CDC), S. 3851-3861, Maui, HI, December 2012.

## Anhang A: MATLAB-Funktionen für die Wassersäulen-Simulation

### Funktion Standpipe\_calc

```
function outputs = Standpipe_calc(inputs, outputs, dt, param, ...
                                options)

% Inputs
v_Valve = inputs(1);      % Valve voltage
normOut = inputs(2);     % Normal outflow valve open / closed
distOut = inputs(3);     % Disturbance outflow valve open / closed
much_noise = inputs(4);  % Much measurement noise due to splashing

% Steady state inflow rate
qIn_ss = max([0, (param.vIntercept + param.vSlope * v_Valve) * ...
             .qIn_Max]);

% Noise of inflow rate
if qIn_ss == 0
    noise_q = 0;
else
    noise_q = param.stdev_q * randn(1);
end

% Cd*A for outflow
param.Cda = (normOut + distOut) * param.CdA;

% Initial conditions
qIn = outputs(1);
h = outputs(2);

% Solve DEs
[t,x] = ode45(@(t, x) Standpipe_DE(t, x, qIn_ss, noise_q, param),
             ...
             [0 dt/2 dt], [qIn, h], options);

% Determine Outputs
outputs(1) = x(3,1);      % outputs(1) = qIn
if x(3,2) >= param.hMax
    outputs(2) = param.hMax; % outputs(2) = h
    outputs(3) = 1;        % outputs(3) = overflow
else
    outputs(2) = x(3,2);
    outputs(3) = 0;
end
outputs(4) = param.sIntercept + param.sSlope*outputs(2) ...
            + (param.stdev_v_lo + much_noise*(param.stdev_v_hi ...
            - param.stdev_v_lo))*randn(1);      % sensor voltage
end % of function Standpipe_calc
```

## Funktion Standpipe\_DE

```
function dxdt = Standpipe_DE( t, x, qIn_ss, noise_q, param)
%
%Calculation of derivatives for standpipe simulation
%
% x(1): Inflow rate (qIn)
% x(2): Level in standpipe (h)

dxdt = zeros(2,1);          % column vector

% Outflow rate
qOut = param.Cda * sqrt(2 * param.g * max([x(2), 0]));

% Derivatives
dxdt(1) = (qIn_ss - x(1)) / param.T1_Valve;
dxdt(2) = (max([x(1)+noise_q, 0]) - qOut)/param.A;

end
```

## Anhang B: Parameters for Standpipe Simulation

Remark: For all calculations in the model, the units cm and s are used. For display, the quantities are converted to US-units, if necessary.

The diameter of the standpipe  $d = 4.5$  cm was measured at the lab setup. Therefore, the cross sectional area of the standpipe is  $A = 15.5$  cm<sup>2</sup>.

According to the manual, the height of the standpipe is  $h_{\text{Max}} = 150$  cm.

Also according to the manual, there are the following correlations for the valve and sensor voltages:

- $h = 0$  cm  $\rightarrow$   $V_{\text{Sensor}} = 3$  V
- $h = 150$  cm  $\rightarrow$   $V_{\text{Sensor}} = 10$  V
- $V_{\text{Valve}} = 1$  V  $\rightarrow$  Valve completely closed
- $V_{\text{Valve}} = 5$  V  $\rightarrow$  Valve completely open

Therefore, the following parameters hold:

- for the sensor:  $s_{\text{Intercept}} = 3$  V,  $s_{\text{Slope}} = 0.0467$  V/cm
- for the relative opening of the valve  $v_{\text{Intercept}} = -0.25$ ,  $v_{\text{Slope}} = 0.25 / V$

The product  $C_d A_V$  of the flow rate coefficient and the cross sectional area of the outflow valve is determined from the lab report of X. Teng, I. Tonner, and W.

Han from April 17, 2013. From this,  $\frac{C_d A_V \sqrt{2g}}{A} = 0.1285 \frac{\sqrt{\text{cm}}}{\text{s}}$ , c.f. p. 7. Therefore,  $C_d A_V = 0.045$  cm<sup>2</sup>. This value is used for both, the nominal outflow and the step disturbance valve, as the valves are equal. The little height difference of the valves is neglected. (It could easily be accounted for in the simulation, if desired).

In afore mentioned report, the following regression results are given:

- Level at steady state:  $h = 93$  cm/V  $\cdot$   $V_{\text{Valve}} - 203$  cm (figure 1)
- Flow rate at steady state:  $q = 0.072$  gpm/V  $\cdot$   $V_{\text{Sensor}} + 0.0929$  gpm (figure 3)
- For the sensor voltage, as above:  $V_{\text{Sensor}} = 0.0467$  V/cm  $\cdot$   $h + 3$  V

From these data, the maximum inflow rate, if the valve is fully open, is calculated to be  $q_{\text{in,Max}} = 75.1$  cm<sup>3</sup>/s.

The time lag of the valve is assumed to be  $T_{1\text{Valve}} = 0.5$  s.

For the noises, the following standard deviations were chosen:

- $s_q = 1$  cm<sup>3</sup>/s for the inflow rate
- $s_{V,lo} = 0.1$  V for the sensor voltage in normal operation
- $s_{V,hi} = 0.5$  V for the sensor voltage with inflow splashing, c.f. section 10 of part D in the manual.



*Prof. Dr. Wolfgang Krämer*

***LabVIEW-Simulationsprogramme  
und Fallstudien zur modell-  
prädiktiven Regelung***

***Impressum***

**Herausgeber**

Der Präsident der Technischen Hochschule Ingolstadt  
Esplanade 10, 85049 Ingolstadt  
Telefon: +49 841 9348-0  
Fax: +49 841 9348-2000  
E-Mail: [info@thi.de](mailto:info@thi.de)

**Druck**

Hausdruck

Die Beiträge aus der Reihe „Arbeitsberichte – Working Papers“  
erscheinen in unregelmäßigen Abständen. Alle Rechte,  
insbesondere das Recht der Vervielfältigung und Verbreitung  
sowie der Übersetzung vorbehalten. Nachdruck, auch  
auszugsweise, ist gegen Quellenangabe gestattet,  
Belegexemplar erbeten.

**Internet**

Alle Themen aus der Reihe „Arbeitsberichte – Working Papers“,  
können Sie unter der Adresse [www.thi.de](http://www.thi.de) nachlesen.