

Modifizierte hybride Modellierung zur Vermeidung der Simulation von Sliding Modes

Modified Hybrid Modelling for the Prevention of the Simulation of Sliding Modes

Eckart Giebler

Bei der Simulation von hybriden Systemen kann es prozess- oder modellbedingt zu einem fortlaufenden Springen zwischen diskreten Zuständen, so genannten Sliding Modes kommen. Die Simulation dieses Phänomens bringt Probleme mit sich. Als Alternative zur Anwendung spezieller Simulationsalgorithmen wird in diesem Beitrag ein Vorgehen diskutiert, bei dem die Simulation der Sliding-Mode-Schwingungen durch geeignete hybride Modellierung vermieden wird. Dabei wird die Dynamik während des Sliding Modes beschrieben und einem zusätzlich eingeführten diskreten Zustand zugeordnet. Als Beispiele dienen ein begrenzter PI-Regler mit ARW-Maßnahme und ein chemischer Verfahrensprozess.

In this paper an alternative approach to the application of special numerical algorithms for the simulation of sliding modes is discussed. It will be shown how the simulation of sliding mode oscillations can be avoided by an appropriate hybrid model representation. An anti reset windup controller and a chemical engineering process exemplify that the simulation problem can be solved by the introduction of an additional hybrid state which represents the dynamics during the sliding mode.

Schlagwörter: Sliding Mode, Hybride Modellierung, Simulation hybrider Systeme, hybrider PI-Regler

Keywords: Sliding mode, hybrid modelling, simulation of hybrid systems, hybrid PI controller

1 Einleitung

Gemischt kontinuierlich-diskrete (hybride) Systeme sind in der jüngeren Vergangenheit Gegenstand intensiver Untersuchungen. Dabei stehen Beschreibungsformen sowie Analyse- und Entwurfsmethoden für hybride Systeme im Mittelpunkt [1–5]. Eng damit verbunden ist die Rechnersimulation hybriden Systemverhaltens. Bei der Simulation hybriden Verhaltens muss im Allgemeinen mit verschiedenen Problemen wie z. B. der Laufzeit-Modellgleichungsänderung, der Neuinitialisierung von Zustandsgrößen, dem Auftreten von Dirac-Impulsen in Ableitungsvariablen, der Laufzeitindexänderung und Sliding-Mode-Schwingungen gerechnet werden [6].

In diesem Beitrag wird das letztgenannte Problem der unerwünschten Simulation von Sliding Modes, das heißt

des Springens zwischen diskreten Systemzuständen behandelt. Dazu wird ein Lösungsansatz beschrieben, der die Simulation der Sliding Modes durch eine entsprechende Modellformulierung von vornherein vermeidet. Das Vorgehen wird an zwei Beispielen diskutiert. Zuerst wird das Auftreten von Sliding Modes für einen begrenzten PI-Regler mit ARW-Maßnahme gezeigt. Als Lösung wird eine hybride Modellrepräsentation eingeführt, die das Ein-Ausgangsverhalten des Reglers ohne Sliding-Mode-Simulation korrekt darstellt. Als zweites wird ein Beispiel aus dem verfahrenstechnischen Bereich diskutiert. Dabei wird gezeigt, wie bei einem steifen Simulationsproblem durch hybride Modellformulierung auf implizite Algorithmen verzichtet werden kann und wie entstehende Sliding Modes durch Modifikation des hybriden Modells vermieden werden können.

2 Sliding Modes

Ein gemischt kontinuierlich-diskretes (hybrides) System ist gekennzeichnet durch verschiedene diskrete Zustände, denen jeweils eine kontinuierliche Dynamik zugeordnet ist. Ein Sliding Mode tritt auf, wenn es zu einem dauernden Umschalten zwischen zwei diskreten Zuständen kommt. Im Englischen wird für dieses Phänomen auch der Begriff *Chattering* verwendet. Sliding Modes können in einem realen System beabsichtigt sein, wenn sie z. B. durch eine Zweipunktregelung erzeugt werden. Weiterhin können sie real auftreten, wenn es einen schnellen Wechsel zwischen unterschiedlichen physikalischen Phänomenen gibt. In den beschriebenen Fällen stellen die Sliding Modes in der Regel kein Problem dar oder sind sogar gewollt. Probleme treten jedoch bei der digitalen Simulation solcher Sliding-Mode-Regime auf.

Die beim Sliding Mode auftretende fortdauernde Zustandsumschaltung und die damit verbundene Änderung der kontinuierlichen Systemdynamik bedeutet, dass bei der Simulation nicht modellierte hochfrequente Schwingungen erzeugt werden. Bei der Simulation eines Sliding-Mode-Regimes mit einer festen Schrittweite kommt es zum Schwanken der Lösung der Differenzialgleichung in einem kleinen Bereich, um eine Lösungstrajektorie, welche die Lösungsgebiete der beiden Systemkonfigurationen voneinander trennt. Wird hingegen eine Simulation mit gesteuerter Schrittweite durchgeführt, wird durch die hohe Dynamik der Sliding-Mode-Schwingung die Schrittweite so stark verkleinert, dass die Simulation praktisch zum Stillstand kommt.

Um das Simulationsproblem des Sliding Modes zu lösen, gibt es verschiedene simulationstechnische Ansätze. Bei einem auf Filippov [7] zurückgehenden Lösungsansatz (*equivalence in dynamics*) wird die rechts- und linksseitige Lösungstrajektorie in einem kleinen Bereich um die Schaltfläche durch Vektoren angenähert. Aus diesen wird die durchschnittliche Bewegungsgeschwindigkeit auf der Lösungsfläche ermittelt [8]. Ein anderer Lösungsansatz (*equivalence in control*) stellt einen Zusammenhang zwischen Schrittweite und Fehler beim Verfolgen der Lösungstrajektorie her. Es wird sichergestellt, dass ein adaptives Integrationsschema dem idealen System innerhalb einer vorgegebenen Toleranz folgt [9]. Trotz dieser jüngeren Entwicklungen wird die Behandlung von Sliding-Mode-Problemen durch allgemeine Simulatoren noch nicht unterstützt [6].

3 Lösungsansatz

Die beschriebenen Möglichkeiten zum Umgang mit Sliding-Mode-Problemen setzen auf der algorithmischen Seite bei der Integration der Simulationsmodelle an. Der hier vorgeschlagene Ansatz sieht hingegen vor, die Simulation von Sliding Modes durch eine entsprechende Modellbildung von vornherein zu vermeiden.

In der Simulationspraxis treten Sliding Modes mitunter bei Systemen (unerwartet) auf, die prinzipiell kontinuier-

lich modelliert wurden, wobei schaltende oder stark die Dynamik ändernde Elemente enthalten sind. Dabei kann bei komplexen Systemen bereits das Lokalisieren der Ursache des Sliding Modes schwierig sein. Um das Problem systematisch zu lösen, ist es deshalb vorteilhaft, das reale System, welches gemischt diskret-kontinuierliche Eigenschaften aufweist, auch als solches zu modellieren und entsprechend in ein Simulationsmodell umzusetzen. Dazu soll folgende Definition eingeführt werden [10]:

Definition: Eine *explizit hybride Darstellung* eines gemischt kontinuierlich-diskreten Systems liegt vor, wenn die kontinuierlichen Zustände, die diskreten Zustände und die dazugehörigen Transitionsbedingungen getrennt (explizit) in Gleichungs- oder Blockform formuliert sind und die durch Ereignisse ausgelösten Veränderungen des kontinuierlichen Systemverhaltens über die diskreten Zustände gesteuert werden.

Liegt eine solche explizite Darstellung vor, wird es im Allgemeinen möglich sein, die benachbarten diskreten Zustände zu lokalisieren, zwischen denen es während der Simulation zu anhaltendem Umschalten kommt. Gelingt es nun, die während des Sliding-Mode-Regimes auftretende durchschnittliche Dynamik zu beschreiben, kann für diesen Betriebsfall ein neuer diskreter Zustand eingeführt werden. Im Einzelfall ist zu überprüfen, ob ein durch die neu eingeführte Beschreibung der Dynamik unter Umständen auftretender Modellfehler den jeweiligen Anforderungen der Modellbildung entspricht. Wird der neue diskrete Zustand und die zugehörige kontinuierliche Dynamik in das explizit dargestellte hybride Simulationsmodell zusätzlich eingeführt, ist eine Simulation der Sliding-Mode-Schwingungen nicht mehr nötig. Um ein vollständiges Modell zu erhalten, sind die Transitionsbedingungen für den neuen Zustand zu formulieren und als Ereignisse, die einen Zustandswechsel auslösen, im Simulationsmodell zu implementieren. Bild 1 stellt diesen Sachverhalt dar.

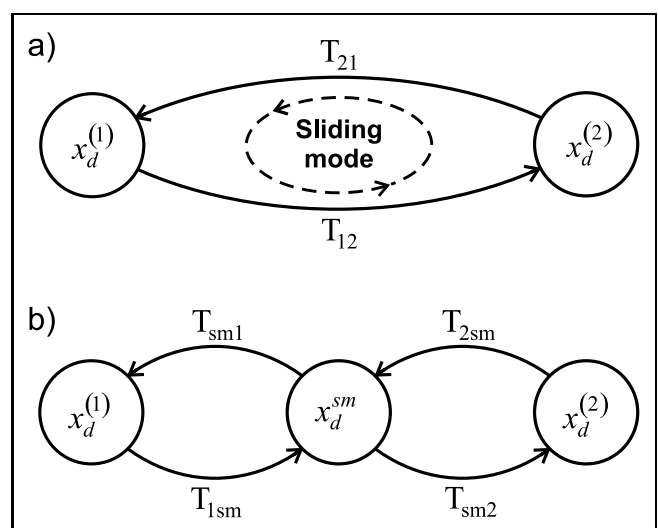


Bild 1: a) System mit 2 diskreten Zuständen b) System mit zusätzlichem Sliding-Mode-Zustand.

4 PI-Regler mit ARW-Begrenzung

Das erste Beispiel eines kontinuierlichen PI-Reglers mit Anti-Reset-Windup-Maßnahmen soll zeigen, wie bereits bei der digitalen Simulation eines relativ einfachen, praktisch bedeutungsvollen Systems Sliding Modes auftreten können. Außerdem lässt sich an dem überschaubaren Beispiel das oben beschriebene Vorgehen gut veranschaulichen.

Im Bild 2a) ist das Blockschaltbild eines PI-Reglers dargestellt, welcher den I-Anteil begrenzt, wenn der Reglerausgang die untere oder die obere Begrenzung u_l bzw. u_u erreicht. Für diesen Regler gilt immer:

$$u_p(t) = k_p e(t), \tag{1}$$

$$u_i(t) = \int_{t_0}^t \dot{u}_i(\tau) d\tau. \tag{2}$$

Dabei erhält der Integrator ein Signal, solange sich die Reglersumme

$$u_{\Sigma}(t) := u_p(t) + u_i(t) \tag{3}$$

innerhalb des Stellbereichs befindet:

$$\dot{u}_i(t) = \begin{cases} \frac{k_p}{T_n} e(t) & u_l < u_{\Sigma}(t) < u_u \\ 0 & \text{sonst} \end{cases} \tag{4}$$

Der Ausgang soll auf den Stellbereich begrenzt werden:

$$u(t) = \begin{cases} u_l & u_{\Sigma}(t) < u_l \\ u_{\Sigma}(t) & u_l \leq u_{\Sigma}(t) \leq u_u \\ u_u & u_{\Sigma}(t) > u_u \end{cases} \tag{5}$$

Neben dem kontinuierlichen Zustand des Reglerintegrators u_i sind also drei diskrete Betriebszustände erkennbar:

1. Normaler Betrieb nach PI-Algorithmus
2. obere Begrenzung
3. untere Begrenzung

Der entsprechend Bild 2a) mit schaltender ARW-Begrenzung realisierte Regler kann auch mit einem rein kontinuierlichen Regler realisiert werden. So zeigt der in [11] vorgeschlagene und in Bild 2b) dargestellte Regler bei einem Anfangszustand 1 (keine Begrenzung wirksam) und nicht zu großen Änderungsgeschwindigkeiten am Reglereingang gleiches Verhalten wie der schaltende Regler¹.

Das Auftreten von Sliding Modes im Regler entsprechend Bild 2a) kann mit einem einfachen Simulationsexperiment nachvollzogen werden. Dazu werden als Reglerparameter festgelegt:

- Verstärkung: $k_p = 1$
- Nachstellzeit: $T_n = 0,1$
- Begrenzungen: $u_u = -u_l = 1$

¹ Zu beachten ist, dass die beiden dargestellten Realisierungen nicht exakt das gleiche Begrenzungsverhalten aufweisen, wie Regler, bei denen als ARW-Maßnahme das inverse Reglerverhalten über einen hochverstärkten Vorwärtszweig rückgekoppelt wird, siehe z. B. [12].

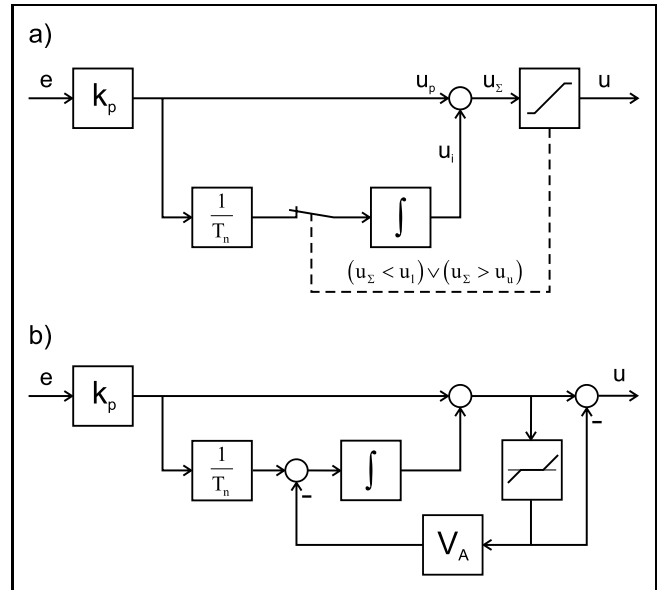


Bild 2: Begrenzter PI-Regler mit ARW-Maßnahme a) mit schaltender Integratorbegrenzung b) mit kontinuierlicher Integratorbegrenzung.

Als Eingangssignal dient ein sinusförmigen Signal der Form:

$$e(t) = \sin(2\pi t). \tag{6}$$

Das Ergebnis einer Simulation mit fester Schrittweite ($h = 10^{-3}$, Runge-Kutta 4. Ordnung) ist in Bild 3 dargestellt. Dabei ist zu erkennen, dass der Regler am Anfang nach dem klassischen PI-Algorithmus arbeitet, dass also die Summe des Proportionalanteils u_p und des Integralanteils u_i (Reglersumme u_{Σ}) das Ausgangssignal u bildet. Übersteigt die Reglersumme den Wert der oberen Begrenzung u_u (hier $t = 0,11$), wird als Ausgangssignal u nur noch dieser obere Begrenzungswert u_u ausgegeben und der Integrator wird angehalten. Dieser Zustand bleibt erhalten bis die Reglersumme u_{Σ} den oberen Begrenzungswert u_u wieder unterschreitet ($t = 0,39$). Nun ist zu beobachten, dass der Integralanteil so weit ansteigt, dass die Reglersumme

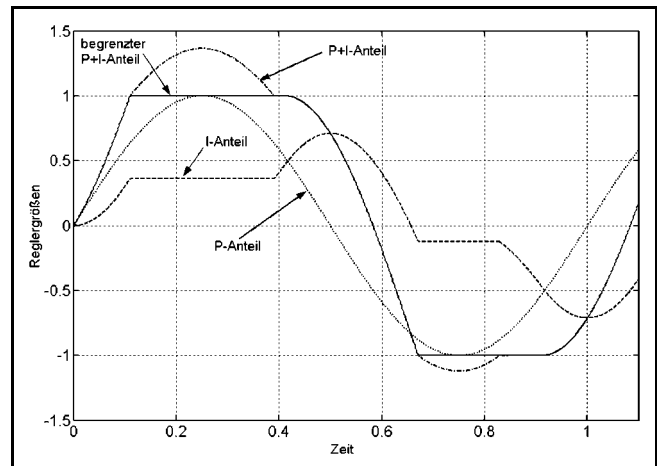


Bild 3: Reglergrößen bei Erreichen der oberen und unteren Begrenzung.

u_Σ die obere Begrenzung wieder erreicht. Das bewirkt wiederum ein Aussetzen der Integration, wodurch die Reglersumme u_Σ erneut unter die obere Begrenzung u_u fällt. Es kommt also zu einem ständigen Umschalten zwischen begrenztem und nicht begrenztem Reglerzustand.

Dieser Umschaltmodus tritt immer genau dann auf, wenn die obere Begrenzung u_u erreicht ist und der Betrag des (positiven) Anstiegs des Integralanteils u_i größer ist als der Betrag des (negativen) Anstiegs des Proportionalanteils u_p :

$$\frac{1}{T_n} e(t) > -\dot{e}(t). \tag{7}$$

Werden die Zeiten zwischen den Umschaltzeitpunkten infinitesimal klein, so verschwindet die Schwankungsbreite der Reglersumme, d.h. die Reglersumme u_Σ bleibt konstant auf dem oberen Grenzwert:

$$\lim_{h \rightarrow 0} u_\Sigma(t) = u_u = const. \tag{8}$$

Eine Ableitung der Gl. (3) nach der Zeit ergibt dann:

$$\dot{u}_p + \dot{u}_i = 0 \Rightarrow \dot{u}_i = -\dot{u}_p. \tag{9}$$

Es kommt also zu einem Steigen des Integralanteils, das dem Fallen des Proportionalanteils proportional ist; der Betrag des P-Anteils „geht“ sozusagen auf den I-Anteil „über“. Ist die Bedingung (7) nicht mehr erfüllt ($t = 0,41$), arbeitet der Regler wieder nach dem klassischen PI-Regelalgorithmus. In Analogie tritt ein Sliding Mode auf, wenn die Reglersumme die untere Begrenzung erreicht und gilt:

$$\frac{e(t)}{T_n} < -\dot{e}(t). \tag{10}$$

Für die beiden Begrenzungszustände des Reglers lassen sich also jeweils zwei verschiedene Regime unterscheiden. Es gibt einen *vollbegrenzten* Zustand, bei dem der Ausgang und der Integrator begrenzt sind und es gibt einen *ausgangsbegrenzten* Zustand, bei dem der Integrator steigt bzw. sinkt. Folglich sind 5 diskrete Zustände erkennbar:

- S1 normaler Betrieb nach PI-Algorithmus
- S2 obere Begrenzung: Integralanteil begrenzt
- S3 obere Begrenzung: Integralanteil steigt
- S4 untere Begrenzung: Integralanteil begrenzt
- S5 untere Begrenzung: Integralanteil sinkt

In Bild 4 ist die Anordnung der Zustände mit den zugehörigen Transitionsbedingungen als Automatengraph dargestellt. Eine Analyse der für die Transitionen nötigen Bedingungen ergibt:

- Der unbegrenzte Zustand wird verlassen, wenn die Reglersumme eine der Begrenzungen erreicht. Sind die Vorzeichen des Anstiegs von Proportional- und Integralanteil gleich wird der vollbegrenzte Zustand erreicht, sonst der ausgangsbegrenzte.
- Die vollbegrenzten Zustände werden verlassen, wenn die Reglersumme die entsprechende Begrenzung wieder erreicht. Die Bedingung (7) bzw. (10) entscheidet, ob

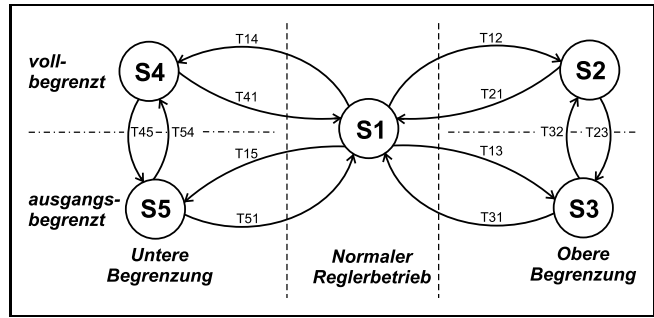


Bild 4: Zustandsautomat des begrenzten PI-Reglers mit ARW-Maßnahme.

der ausgangsbegrenzte Zustand oder der Normalbetrieb erreicht wird.

- Vom ausgangsbegrenzten Zustand wird in den vollbegrenzten Zustand übergegangen, wenn die Bedingung (7) bzw. (10) nicht mehr erfüllt ist.
- Vom ausgangsbegrenzten Zustand wird in den Normalbetrieb übergegangen, wenn der Anstieg des Reglereingangs gegen Null geht.

Die hier formulierten Bedingungen sind in Tabelle 1 für alle möglichen Transitionen dargestellt. Es ist zu erkennen, dass jeweils ein Ereignis in Form einer Näherungsbedingung zum Auslösen des Zustandwechsels führt. Nebenbedingungen entscheiden darüber, welcher Folgezustand erreicht wird. So wird z. B. bei einer linksseitigen Annäherung der Reglersumme u_Σ an die obere Begrenzung u_u^- das Verlassen des Zustandes 1 ausgelöst und die Vorzeichen des Anstiegs von Proportional- und Integralanteil entscheiden darüber, ob der vollbegrenzte Zustand 2 (T12) oder der ausgangsbegrenzte Zustand 3 (T13) erreicht wird.

Nun fehlt noch die Beschreibung der Dynamik für die neu eingeführten Zustände. Die vollbegrenzten Zustände S2 und S4 entsprechen den ursprünglichen Begrenzungszu-

Tabelle 1: Transitionen für den hybriden PI-Regler mit ARW-Begrenzung.

Transition	Ereignis	Nebenbedingung
T 12	$u_\Sigma \rightarrow u_u^-$	$\dot{e}(eT_n^{-1}) > 0$
T 13		$\dot{e}(eT_n^{-1}) < 0$
T 14	$u_\Sigma \rightarrow u_l^+$	$\dot{e}(eT_n^{-1}) > 0$
T 15		$\dot{e}(eT_n^{-1}) < 0$
T 21	$u_\Sigma \rightarrow u_u^+$	$(eT_n^{-1}) < -\dot{e}$
T 23		$(eT_n^{-1}) > -\dot{e}$
T 41	$u_\Sigma \rightarrow u_l^-$	$(eT_n^{-1}) > -\dot{e}$
T 45		$(eT_n^{-1}) < -\dot{e}$
T 31	$-\dot{e} \rightarrow (eT_n^{-1})^-$	-
T 32	$\dot{e} \rightarrow 0^-$	-
T 51	$-\dot{e} \rightarrow (eT_n^{-1})^+$	-
T 54	$\dot{e} \rightarrow 0^+$	-

ständen Gl. (4) und (5). Neu sind die ausgangsbegrenzten Zustände S3 und S5. Für diese nimmt der Ausgang den oberen bzw. unteren Grenzwert an. Der kontinuierliche Zustand u_i hat entsprechend den Gln. (2) und (9) den Verlauf:

$$u_i(t) = u_i(t_0) + \int_{t_0}^t (-\dot{u}_p(\tau)) d\tau, \quad (11)$$

wobei t_0 der Zeitpunkt ist, zu dem der ausgangsbegrenzte Zustand erreicht wurde. Die Integration nach Gl. (11) muss jedoch numerisch nicht ausgeführt werden. Es reicht, das Integral für das Zeitintervall zu lösen, in dem der ausgangsbegrenzte Zustand vorliegt. Eine weitere Vereinfachung ist möglich, da die Reglersumme bei Eintritt in den ausgangsbegrenzten Zustand immer den jeweiligen Grenzwert u_{lim} hat. Die Lösung von Gl. (11) lautet also allgemein:

$$\begin{aligned} u_i(t_{end}) &= u_i(t_0) - (u_p(t_{end}) - u_p(t_0)) \\ &= u_{lim} - u_p(t_{end}). \end{aligned} \quad (12)$$

Der Integrator wird demnach bei Verlassen des ausgangsbegrenzten Zustandes entsprechend neu initialisiert.

Die beiden oben beschriebenen herkömmlichen Reglerrealisierungen (Bild 2) sowie der hier vorgestellte hybride PI-Regler wurden mit dem Simulationssystem MATLAB/Simulink umgesetzt und untersucht. Der Vergleich der beiden herkömmlichen Regler zeigte, dass bei der Realisierung mit der kontinuierlich gestalteten Integratorbegrenzung (Bild 2b) kein Springen zwischen Reglerzuständen, also kein klassischer Sliding Mode mehr auftritt. Trotzdem ist das zugrunde liegende simulationstechnische Problem nicht gelöst. Die große Dynamik der Sliding-Mode-Schwingung wird lediglich durch das Einbringen der hochdynamischen Rückkopplung ersetzt und es wird somit ein steifes System erzeugt. Die Simulation mit gesteuerter Schrittweite ist zwar bei diesem Simulationsmodell möglich; bei entsprechender Genauigkeitsanforderung (große Verstärkung V_A und/oder kleiner zulässiger Simulationsfehler) ist die Simulation ebenfalls zeitaufwändig, da automatisch eine entsprechend kleine Schrittweite eingestellt wird. Gut lösbar ist das steife Problem durch Übergang auf entsprechende implizite Integrationsverfahren. Deren Anwendung erfordert jedoch einen mitunter erheblich größeren Rechenaufwand. Verallgemeinert lässt sich feststellen, dass das Ersetzen eines schaltenden durch einen kontinuierlichen Übergang (*smoothing*) das Sliding-Mode-Problem aus numerischer Sicht nicht löst [2].

Der hybride PI-Regler kann mit dem für MATLAB/Simulink zusätzlich zur Verfügung stehenden Werkzeug Stateflow einfach realisiert werden. Dazu wird der in Bild 4 dargestellte Zustandsautomat graphisch umgesetzt. Die genaue Detektion der laut Tabelle 1 möglichen acht Ereignisse kann im Simulinkmodell durch Hit-Crossing-Blöcke erzwungen werden; die entsprechenden Ereignissignale lösen dann im Stateflowmodell die Zustandswechsel aus. Vergleichende Untersuchungen zeigten, dass sich das Verhalten des ARW-begrenzten PI-Reglers während der kritischen

Betriebszustände S3 und S5 mit der hybriden Reglerrealisierung effizienter und genauer simulieren lässt als mit den klassischen Reglerrealisierungen entsprechend Bild 2.

5 Kontinuierliche Regeneration bei einem nasschemischen Beizprozess

Als zweites Beispiel dient ein Verfahrensprozess aus dem Bereich der Galvano- und nasschemischen Oberflächentechnik, bei dem es durch Veränderungen im chemischen Reaktionsgeschehen zu strukturellen Änderungen im Systemverhalten kommen kann.

Betrachtet wird ein Prozess zum Beizen von Kupferoberflächen, bei dem in einer schwefelsauren Beizlösung durch das Beizmittel Eisen-III-Sulfat Oxid- und Zunderschichten abgelöst werden. Mit dem Ziel, eine umweltschonende Stoffkreislaufschließung zu realisieren, wird beim vorgestellten Prozess eine Regeneration des Eisen-II zu Eisen-III durch Reaktion mit Peroxodisulfat (im weiteren verkürzend als Persulfat bezeichnet) erreicht. Dieser Stoff kann durch elektrolytische Oxidation in einer separaten Regenerationszelle erzeugt werden. Der Prozess wurde im Labor Geschlossene Stoffkreisläufe [13] des Instituts für Automatisierungstechnik der TU Dresden untersucht; für Details zum vorgestellten Verfahren siehe [10;14].

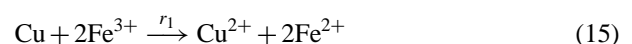
Das Modell eines im Überlauf betriebenen Beizbades kann auf Grundlage von Bilanzgleichungen für Volumen und Masse aufgestellt werden [10]. Die Massebilanz eines gelösten Stoffes lautet vereinfacht:

$$\frac{dm}{dt} = \dot{m}_{in} - \dot{m}_{out} + r, \quad (13)$$

wobei die hereinkommenden Masseströme \dot{m}_{in} durch den Zulauf und durch Einschleppungen sowie die herausgehenden Masseströme \dot{m}_{out} durch den Überlauf und Ausschleppungen bewirkt werden. Weiterhin kommt es zu Masseänderungen gelöster Stoffe durch mit der Geschwindigkeit r ablaufende chemische Reaktionen. Der Zusammenhang zwischen der Masse und dem (hier nicht modellierten) Volumen lässt sich durch die Konzentrationsdefinition herstellen:

$$c := \frac{m}{V}. \quad (14)$$

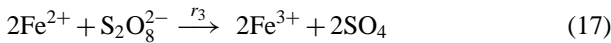
Zur Berechnung der Massebilanzgleichung (13) müssen Geschwindigkeitsgesetze der ablaufenden Reaktionen aufgestellt werden. Für die bei konstanter Temperatur ablaufenden Reaktionen sei lediglich eine Abhängigkeit von der Konzentration berücksichtigt. Drei Reaktionen des Beizsystems mit unterschiedlichen Reaktionsgeschwindigkeiten r_1 , r_2 und r_3 werden modelliert. Das Kupferbeizen durch Eisen-III folgt dem Reaktionsschema:



Als zweites Beizmittel wirkt das Oxidationsmittel Persulfat:



Wird in eine Eisen-II enthaltende Prozesslösung Persulfat eingebracht, findet die Regeneration von Eisen-II zu Eisen-III statt:



Während sich Ware im Beizbad befindet, gilt für die Reaktionsgeschwindigkeiten:

1. Eisen-III-Beizen (Reaktion 1. Ordnung):

$$r_1 = Ak_1c_{\text{Fe}^{3+}} \quad (18)$$

2. Persulfatbeizen (Reaktion 1. Ordnung):

$$r_2 = Ak_2c_{\text{S}_2\text{O}_8^{2-}} \quad (19)$$

3. Regeneration (Reaktion 2. Ordnung):

$$r_3 = Vk_3c_{\text{Fe}^{2+}}c_{\text{S}_2\text{O}_8^{2-}} \quad (20)$$

Für den Vektor \underline{r} der Reaktionsgeschwindigkeiten kann folglich eine allgemeine nichtlineare Funktion eingeführt werden:

$$\underline{r} = \underline{f}_r(\underline{c}, A, V) \quad (21)$$

Das Bild 5 zeigt die prinzipielle Struktur des Konzentrationsmodells für das im Überlauf betriebene Beizbad. Dort ist zu erkennen, dass es beim Verschwinden (bzw. Auftauchen) einer in den Gln. (18) bis (20) enthaltenen Konzentrationsgrößen \underline{c} zu strukturellen Änderungen in der Rückführung über die Reaktionsgeschwindigkeitsfunktion \underline{f}_r kommt. Von außen können solche Änderungen durch Entzug von Fläche A (Ware herausnehmen) oder Volumenzug (Bad ablassen) bewirkt werden.

Simulationstechnisch problematisch wird das beschriebene System, da sich die Reaktionsgeschwindigkeiten r_1 bis r_3 stark unterscheiden. Im betrachteten Fall läuft der Regenerationsprozess mit wesentlich größerer Geschwindigkeit r_3 als die Beizreaktionen (r_1, r_2) ab. Dadurch kommt es beim Zudosieren von Persulfat in die Eisen-II-haltige Beizlösung defacto augenblicklich zum Umsatz der beiden Stoffe. Es ergibt sich folglich ein Modell mit erheblicher Systemsteifigkeit, wenn der Prozess mit den oben eingeführten stetigen Gleichungen der Reaktionsgeschwindigkeit modelliert wird.

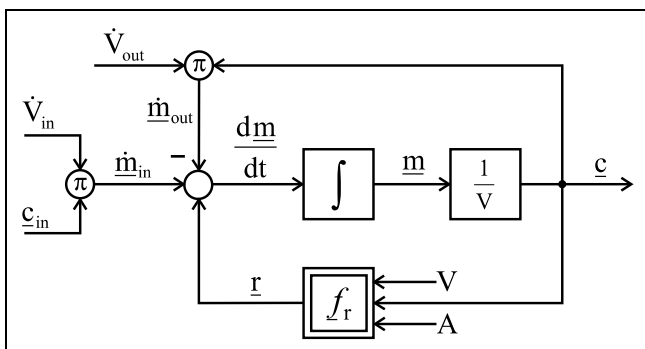


Bild 5: Konzentrationsmodell des im Überlauf betriebenen Beizbades.

Die numerische Lösung des Problems mit einem expliziten Integrationsalgorithmus ist effizient nicht möglich. Implizite Integrationsverfahren sind anwendbar, werden jedoch mit erhöhtem Rechenaufwand bezahlt. Der Zusatzaufwand ist unerheblich, wenn das oben beschriebene Modell isoliert betrachtet wird. In der Simulationspraxis wird das Modell jedoch Teil eines umfangreicheren Simulationsmodells sein. Der numerische Mehraufwand kann in diesem Fall beträchtlich sein.

Eine Lösung des Problems kann wiederum durch eine hybride Systembeschreibung und deren Modifikation erreicht werden. So lassen sich drei diskrete Zustände RS1 bis RS3 festlegen, die als Reaktionszustände bezeichnet werden sollen. In Bild 6 sind die Reaktionszustände als Zustandsautomat mit den zugehörigen Zustandswechseln dargestellt. Bei Simulation mit einem expliziten Algorithmus oder mit fester Schrittweite äußert sich die Systemsteifigkeit bei dieser Modellrealisierung in einer Sliding-Mode-Schwingung zwischen den Reaktionszuständen RS1 und RS3.

Auch hier kann das Problem durch Einführung eines zusätzlichen diskreten Zustandes, welcher speziell die Dynamik während des Sliding Modes beschreibt, elegant gelöst werden. Dazu wird das Wissen, dass zudosiertes Persulfat praktisch sofort umgesetzt wird, gleichungsmäßig ausgedrückt durch:

$$r_{3,\text{S}_2\text{O}_8^{2-}} \stackrel{!}{=} -\dot{m}_{\text{in},\text{S}_2\text{O}_8^{2-}} \quad (22)$$

Für das neue hybride Modell des Beizprozesses wird als weiteres Unterscheidungsmerkmal für die Reaktionszustände das Zudosieren von Persulfat ($\dot{m}_{\text{in},\text{S}_2\text{O}_8^{2-}}$) eingeführt. Somit ergeben sich anstelle des alten Reaktionszustandes RS3 zwei neue Zustände RS3n und RS4n. Der Reaktionszustand RS4n beschreibt mit Gl. (22) das Verhalten beim Zudosieren im Eisenbereich. Mit dem Zustand RS3n wird Zudosieren im Persulfatbereich beschrieben. In Bild 7 ist der neue Zustandsautomat mit den zugehörigen Transitionsbedingungen dargestellt.

Zur Verdeutlichung ist in Bild 8 die Simulation des Sliding-Mode-Regimes mit einem Modell entsprechend Bild 6 beispielhaft dargestellt. Durch Kupferbeizen im Eisenbereich sinkt die Eisen-III-Konzentration. Beim Erreichen eines Konzentrationsgrenzwertes (hier $c_{\text{Fe}^{3+}} = 17 \text{ g l}^{-1}$) wird mit

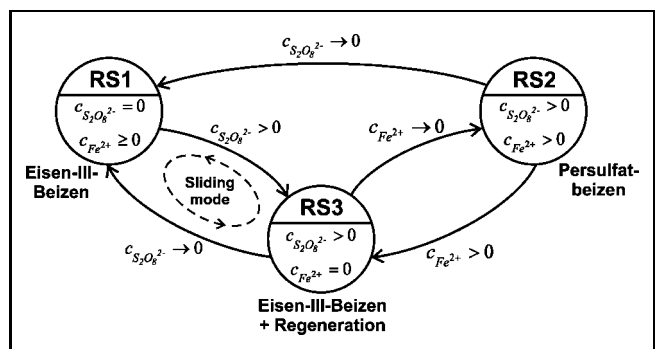


Bild 6: Zustandsautomat des Kupferbeizprozesses.

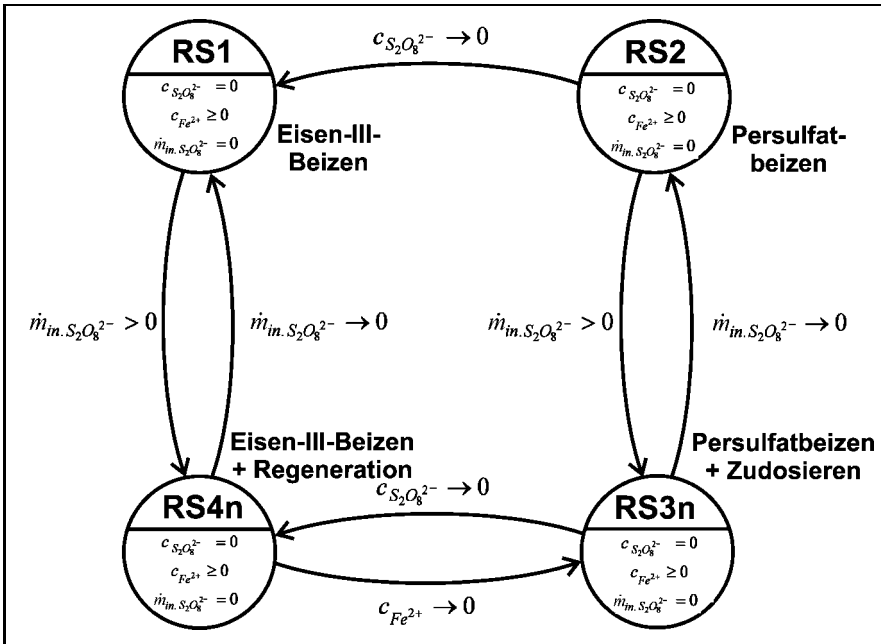


Bild 7: Zustandsautomat des Kupferbeizprozesses ohne Sliding Mode.

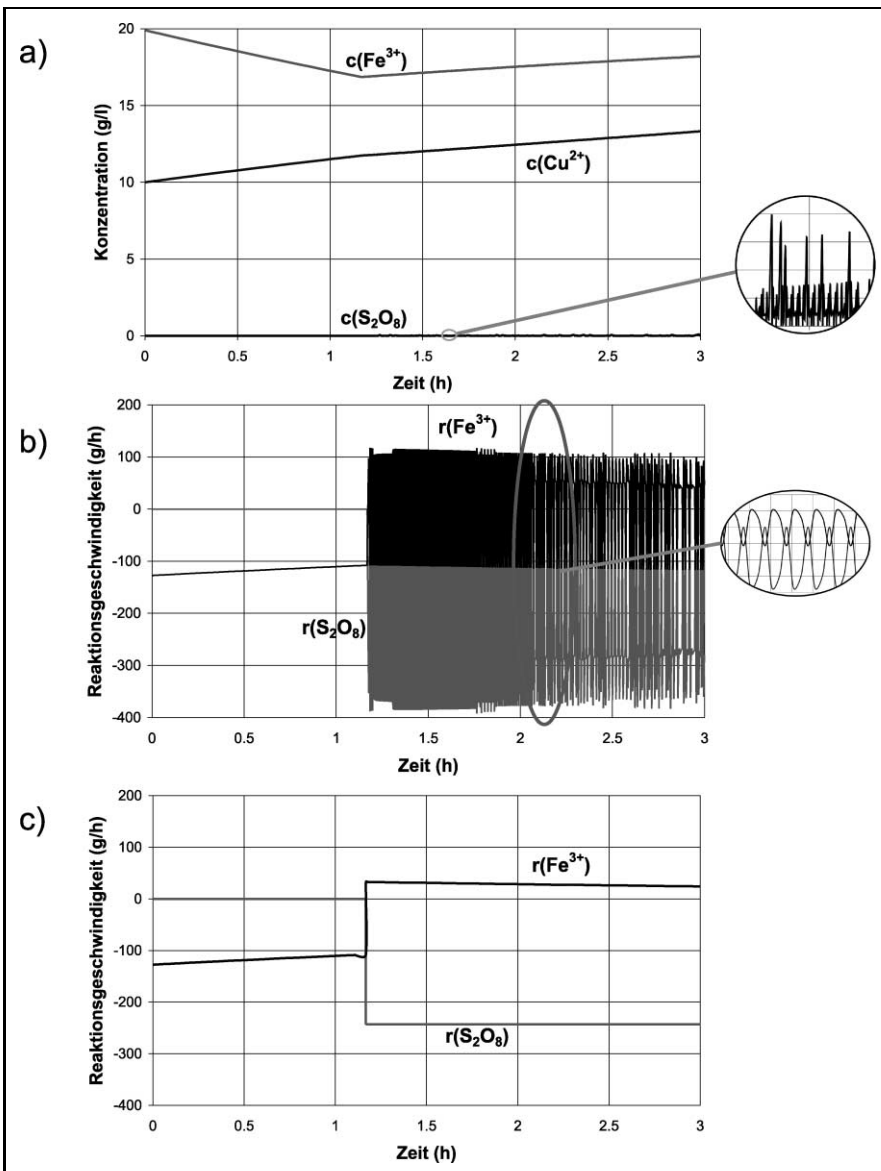


Bild 8: Sliding-Mode bei Simulation einer kontinuierlichen Eisen-III-Regeneration a) Konzentrationsverläufe mit Sliding-Mode b) Reaktionsgeschwindigkeit mit Sliding-Mode c) Reaktionsgeschwindigkeit ohne Sliding-Mode.

dem kontinuierlichen Zudosieren einer hochkonzentrierten Persulfatlösung begonnen. Durch den im Vergleich zu den anderen stattfindenden Reaktionen schnellen Umsatz des Persulfats sinkt dessen Konzentration auf Null. Es kommt zu Sliding-Mode-Schwingungen der simulierten Persulfatkonzentration (Bild 8a) sowie der entsprechenden Reaktionsgeschwindigkeiten (Bild 8b). In Bild 8c ist die Simulation der Reaktionsgeschwindigkeiten mit dem entsprechend Bild 7 umgestalteten hybriden Modell abgebildet. Es treten keine Sliding-Mode-Schwingungen mehr auf. Die Simulation führt zu den gleichen Ergebnissen wie bei der herkömmlichen Formulierung, ist jedoch um mehrere Zehnerpotenzen schneller.

6 Zusammenfassung und Diskussion

Es wurde ein Vorgehen zum Umgang mit Problemen vorgestellt, welche bei der Simulation von Sliding Modes auftreten. Im Gegensatz zu verschiedenen Verfahren, die spezielle Simulationsalgorithmen nutzen, schlägt der hier behandelte Ansatz das Vermeiden von Sliding Modes durch eine vorteilhaft gewählte hybride Modellformulierung vor. Bei diesem Vorgehen sind drei wesentliche Schritte zu erkennen:

1. Explizit hybride Darstellung
2. Beschreibung der Dynamik des Sliding Modes und Zuordnen zu einem neu eingeführten diskreten Zustand
3. Festlegung der Transitionsbedingungen

Das Vorgehen wurde an zwei unterschiedlich gearteten Beispielen erläutert. Zuerst wurde das Auftreten von Sliding Modes bei der Simulation eines begrenzten PI-Reglers mit ARW-Maßnahme aufgezeigt. Als Lösung wurde eine hybride Modellformulierung vorgestellt, bei der je Begrenzung zwei unterschiedliche diskrete Zustände existieren. Einer der beiden Zustände dient dabei der Beschreibung des Reglerverhaltens während des Sliding Modes. Diese Beschreibung konnte soweit vereinfacht werden, dass eine Integration des Reglerzustandes während des Sliding-Mode-Zustandes nicht nötig ist, sondern der Integrator bei Verlassen des Zustandes einmalig auf seinen neuen korrekten Wert gesetzt wird.

Beim zweiten Beispiel aus dem Bereich der Verfahrenstechnik liegt originär ein rein kontinuierlich beschreibbares Systemverhalten vor. Bei Auftauchen oder Verschwinden von Reaktionspartnern treten veränderte Reaktionsmechanismen auf, was als strukturelle Veränderung im Systemverhalten interpretiert werden kann. Das System zeigt erhebliche Steifigkeit, da ein kontinuierlich zudosierter Stoff mit sehr hoher Geschwindigkeit umgesetzt wird. Um den Übergang auf numerisch aufwändigere implizite Integrationsverfahren zu vermeiden, kann auch hier eine adäquate hybride Modellierung hilfreich sein. Dazu wird das Systemverhalten explizit hybrid dargestellt und dann ein separater Zustand eingeführt, welcher das Verhalten während der kontinuierlichen Regeneration beschreibt. Dazu wird die reale Abhängigkeit der Reaktionsgeschwindigkeit von

der Konzentration vernachlässigt, indem ein sofortiger Umsatz angenommen wird. Die Simulation der hohen Dynamik der schnellen Reaktion wird also umgangen, indem eine unendlich hohe Reaktionsgeschwindigkeit angenommen wird und somit eine Integration des entsprechenden Vorgangs nicht mehr nötig ist.

Wie die behandelten, relativ überschaubaren Beispiele zeigen, erfordert das Vermeiden von Sliding-Mode-Simulationen durch Modellmodifikation einen zusätzlichen Aufwand. Möglich ist das Vorgehen nur, wenn die Quelle der Sliding-Mode-Schwingung eindeutig lokalisiert werden kann und eine Beschreibung der Dynamik des Sliding-Mode-Regimes möglich ist. Ferner müssen alle Transitionsbedingungen formuliert und im Simulationsmodell überwacht werden. Es bleibt zu untersuchen, ob das hier lokal angewendete Vorgehen für komplexe Systeme geeignet ist. Das gilt insbesondere, wenn Sliding Modes in einer Reihe von Teilmodellen oder über mehrere Teilmodelle verteilt entstehen. Offen bleibt auch, ob der erhöhte Modellierungsaufwand noch lohnt, wenn zukünftig die in Abschnitt 2 angesprochenen numerischen Methoden zur Lösung von Sliding-Mode-Problemen in Standardsimulationswerkzeugen verfügbar sind.

Das im zweiten Beispiel beschriebene Vorgehen zur Vermeidung von Systemsteifigkeit ist besonders für die Simulation komplexer Modelle geeignet, wenn einige lokal abgrenzbare steife Teilsysteme vorhanden sind. Dann besteht die Möglichkeit, die Modelle mit überschaubarem Aufwand umzugestalten, sodass eine Integration mit expliziten Algorithmen und somit eine Rechenzeitminimierung möglich ist. Ist bei komplexen Modellen eine durchgehende Problemdarstellung durch explizite hybride Modellierung nicht mehr realisierbar bzw. wird der Aufwand dafür zu hoch, sind implizite Integrationsalgorithmen unvermeidbar.

Vorteilhaft bei dem in dieser Arbeit vorgestellten Vorgehen ist das tiefere Prozessverständnis, welches durch die konsequent explizite Darstellung des hybriden Systemverhaltens zu erwarten ist. Ferner erübrigt ein durch die Modellbildung vermiedenes Sliding-Mode-Problem simulationstechnische Zusatzmaßnahmen. Das bedeutet, dass die Dynamik des Sliding-Mode-Regimes bereits bei der Modellbildung und nicht erst während der Simulation ermittelt wird.

Danksagung

Dieser Beitrag ist im Rahmen des vom BMBF geförderten Verbundvorhabens „Umstellung bestehender galvanotechnischer Anlagen auf eine stoffverlustminimierte Prozesstechnik bei gleichzeitiger Kostensenkung“ entstanden. Projektträger ist die Umweltforschung und -technik des DLR. Der Autor dankt für die Unterstützung.

Literatur

- [1] Engell, S.: Modellierung und Analyse hybrider dynamischer Systeme. at – Automatisierungstechnik 45 (1997) 4, 152–161.
- [2] Cellier, F.E.; Elmqvist, H.; Otter, M. u. a.: Guidelines for Modeling and Simulation of Hybrid Systems. IFAC World Congress. Sydney 1993, 1219–1225.

- [3] Branicky, M.S.: Studies in hybrid systems: Modeling, analysis and control. Diss. Massachusetts Institute of Technology, Cambridge 1995.
- [4] Flaus, J.-M.: Modeling and analysis of hybrid dynamical systems: an introduction. JESA – European journal of automation 32 (1998) 7/8, 797–830.
- [5] Schaft, A.v.d.; Schumacher, H.: An introduction to hybrid dynamic systems. (Lecture notes in control and information Science 251) London, Berlin, Heidelberg (Springer) 2000.
- [6] Mosterman, P.J.: An Overview of Hybrid Simulation Phenomena and Their Support by Simulation Packages. in Hybrid Systems: Computation and Control. Berlin, Heidelberg, New York (Springer) 1999 Lecture Notes in Computer Science 1569, 165–177.
- [7] Filippov, A.F.: Differential equation with discontinuous right-hand sides. Matematicheskii Sbornik 51 (1960).
- [8] Mosterman, P.J.; Zhao, F.; Biswas, G.: Sliding Mode Model Semantics and Simulation for Hybrid Systems. Hybrid Systems V. Berlin, Heidelberg, New York (Springer) 1999 Lecture Notes in Computer Science 1567, 218–237.
- [9] Zhao, F.; Utkin, V.: Adaptive Simulation and Control of Variable-structure Control Systems in Sliding Regimes. Automatica 32 (1996) 7, 1037–1042.
- [10] Giebler, E.: Modellbildung und Simulation für die Prozessführung von Verfahrensprozessen der Galvano- und nasschemischen Oberflächentechnik. Diss. TU Dresden. Düsseldorf (VDI-Verlag) 2002 (VDI-Fortschritt-Berichte 952).
- [11] Noisser, R.: Anti-Reset-Windup-Maßnahmen bei Eingrößenregelungen. at – Automatisierungstechnik 35 (1987) 1, 32–39.
- [12] Bühler, H.: Anti-Reset-Windup-Maßnahmen bei stetigen Reglern. at – Automatisierungstechnik 36 (1988) 5, 190–191.
- [13] Hauser, S.; Neumann, K.-H.; Reich, A.: Stoffkreislaufschließung. Versuchsanlage für die Automatisierung geschlossener Stoffkreisläufe. Metalloberfläche 51 (1997) 9, 670–677.
- [14] Thiele, W.; Wildner, K.; Matschiner, H. u. a.: Offenlegungsschrift OS DE 198 50 530 A. Kreislaufverfahren zum Beizen von Kupfer und Kupferlegierungen (2000).

Manuskripteingang: 04. Februar 2002.



Dr.-Ing. Eckart Giebler hat in Dresden und Valladolid (E) studiert und ist seit 1998 als wissenschaftlicher Mitarbeiter im Labor Geschlossene Stoffkreisläufe des Instituts für Automatisierungstechnik (Prof. K. Janschek) der TU Dresden beschäftigt. Hauptarbeitsfeld: Modellbildung und Simulation von Verfahrensprozessen der Galvano- und nasschemischen Oberflächentechnik.

Adresse: TU Dresden, Institut für Automatisierungstechnik, D-01062 Dresden, Tel.: (0351) 463-32243, E-Mail: giebler@ifa.et.tu-dresden.de

Oldenbourg! für Ingenieure

Helmut E. Scherf

Modellbildung und Simulation dynamischer Systeme

Mit Matlab- und Simulink-Beispielen
2003. 192 Seiten

Oldenbourg Lehrbücher für Ingenieure

€ 24,80

ISBN 3-486-27285-3

Ausgehend von der Problembeschreibung werden die einzelnen Beispiele umfassend und detailliert bis zur Lösung dargestellt.

Anne Angermann / Michael Beuschel / Martin Rau / Ulrich Wohlfarth

Matlab - Simulink - Stateflow

Grundlagen, Toolboxen, Beispiele
Oldenbourg Lehrbücher für Ingenieure

2002. 418 Seiten

€ 29,80

ISBN 3-486-25979-2

"... die beste mir bekannte Einführung in Matlab."
(Prof. Dr. H. Forster, Fachhochschule Nürnberg)



Neuerscheinungen

Oldenbourg Wissenschaftsverlag
Rosenheimer Straße 145, D-81671 München
Telefon 089/4 50 51-0, Fax 089/4 50 51-204
Bestellungen: www.oldenbourg-verlag.de

Frieder Grupp / Florian Grupp

MATLAB 6 für Ingenieure

Grundlagen und Programmierbeispiele
Oldenbourg Lehrbücher für Ingenieure

2002. 105 Seiten

€ 19,80

ISBN 3-486-25957-1

Die elementare Einführung in MATLAB, mit der auch der Neueinsteiger rasch mit den Grundlagen von MATLAB vertraut wird.

Oldenbourg

