

Kontinuierlich-ereignisdiskrete Prozessmodelle für die Simulationsumgebung DiVA

R. Köhler, S. Küpper, M. Zeitz

Institut für Systemdynamik und Regelungstechnik, Universität Stuttgart

Pfaffenwaldring 9, D-70550 Stuttgart, Germany

email: {koeehler,Stephan.Kuepper,zeitz}@isr.uni-stuttgart.de, fax: ++49-711-685-6371

Abstract

Köhler R., Küpper S., Zeitz M.: Continuous-Time/Discrete-Event Process Models for the Simulation Environment DiVA. Most simulation models of chemical processes describe time continuous systems. But in practice, one often encounters discontinuities which arise for instance at the transition of laminar to turbulent flow or at the phase transitions of a material mixture. Also interventions, due to process control may lead to discontinuities. For the resulting continuous-time/discrete-event process models, a special treatment is required that detects the discrete event times at which discontinuities are initiated. The event handling is necessary for steering of the numerical routines. This contribution describes the mathematical representation of continuous-time/discrete-event differential algebraic process models in the model library of the simulation environment DiVA. Moreover, the event handling in DiVA as well as the automatic code generation of continuous-time/discrete-event simulation models for Dn>A are presented.

1 Einführung

Für die Analyse, Auslegung und Optimierung verfahrenstechnischer Anlagen und Prozesse stellt die Simulation ein unverzichtbares Hilfsmittel dar. Zur Simulation zeitkontinuierlicher Systeme werden Modelle verwendet, die mit differential-algebraischen Gleichungen (DAE) oder partiellen Differentialgleichungen (PDE) beschrieben werden. In der Praxis hat man es jedoch oft mit Systemmodellen zu tun, die zu bestimmten Zeitpunkten Unstetigkeiten aufweisen. Unstetigkeiten entstehen einerseits aufgrund physikalisch-chemischer Eigenschaften des Systems, wie zum Beispiel beim Übergang von einer laminaren zur turbulenten Strömung oder beim Phasenübergang eines Stoffgemischs. Andererseits können unstetige Vorgänge auch gewollte Änderungen durch äußere Einflüsse sein, wie zum Beispiel Prozess-Steuierungen oder diskrete Reglereingriffe. Zu diskreten Zeitpunkten auftretende Vorgänge, die Unstetigkeiten auslösen, werden als *Ereignisse* bezeichnet. Systeme, die sowohl kontinuierliche als auch ereignisdiskrete Eigenschaften besitzen, werden *gemischt kontinuierlich-ereignisdiskret* oder *hybrid* genannt.

Bei der Simulation von kontinuierlich-ereignisdiskreten Modellen mit numerischen Verfahren für kontinuierliche Modelle führen Unstetigkeitsstellen, d.h. Ereignisse, zu erheblichen Effizienzverlusten bei der Integration oder sogar zum Abbruch der Integration. Aus diesem Grund ist für die Simulation gemischt kontinuierlich-ereignisdiskreter Modelle eine spezielle Ereignisbehandlung nötig, welche die Ereigniszeitpunkte erkennt, die Integration an solchen Stellen unterbricht und nach dem Schaltvorgang wieder fortsetzt [1].

In der an dem Stuttgarter Institut entwickelten und eingesetzten Simulationsumgebung DiVA zur dynamischen Simulation, Analyse und Optimierung verfahrenstechnischer Anlagen und Prozesse stehen numerische Routinen für die Behandlung von differential-algebraischen Simu-

lationsmodellen zur Verfügung [2, 3,4,5]. Für gemischt kontinuierlich-ereignisdiskrete Modelle ist eine spezielle Ereignisbehandlung implementiert, die Unstetigkeitsstellen erkennt und die Ansteuerung der numerischen Verfahren übernimmt.

Dieser Beitrag behandelt die Darstellung und Implementierung gemischt kontinuierlich-ereignisdiskreter Simulationsmodelle in DIVA. Dazu wird zunächst die differential-algebraische Formulierung kontinuierlich-ereignisdiskreter Modelle beschrieben. Anschließend wird gezeigt, wie in DIVA diese Modelle implementiert und wie Ereignisse numerisch behandelt werden. Schließlich wird auf die automatische Codierung von Modellen mit Unstetigkeiten für DIVA eingegangen [6].

2 Differential-algebraische Prozessmodelle mit Unstetigkeiten

In der Simulationsumgebung DIVA werden zeitkontinuierliche Modelle in der linear-impliziten differential-algebraischen Form mit einem differentiellen Index kleiner/gleich eins verwendet:

$$\mathbf{B}(\mathbf{x}, \mathbf{u}, \mathbf{h}, \mathbf{p}, t) \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{h}, \mathbf{p}, t) \quad t > t_0, \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad (1)$$

Dabei hängen die B-Matrix und der f-Vektor von den Zuständen $x(t)$, den Eingangsgrößen $u(t)$, den Modellparametern p , den Hilfsgrößen $h(x,u,p,t)$ und der Zeit t ab. Hilfsgrößen werden zur Berechnung von mehrfach vorkommenden funktionellen Teilausdrücken verwendet. Für eine tiefer gehende Erklärung der Funktion und Verwendung von Hilfsgrößen wird auf [7, 8] verwiesen. Der Anfangswert x_0 erfüllt die algebraischen Gleichungen von (1).

Für Prozessmodelle mit Unstetigkeiten wird (1) mit unstetigen Modellparametern $p_i = P_i(m)$, $i = 1(1)P$, unstetigen Hilfsgrößen $h_j = h_j(m)$, $i = 1(1)H$ und binären Schaltparametern $r_{ii} \in \{0,1\}$, $i = 1(1)M$ erweitert:

$$\mathbf{B}(\mathbf{x}, \mathbf{u}, \mathbf{h}, \hat{\mathbf{h}}, \mathbf{p}, \hat{\mathbf{p}}, \mathbf{m}, t) \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{h}, \hat{\mathbf{h}}, \mathbf{p}, \hat{\mathbf{p}}, \mathbf{m}, t) \quad t > t_0, \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad (2)$$

Dabei können mit Hilfe der binären Schaltparameter m_j auch unstetige Anteile in den Elementen der B-Matrix und des f-Vektors formuliert werden. Dazu werden die Elemente b_{ij} der B-Matrix und die Komponenten f_i des f-Vektors in stetige Anteile \bar{b}_{ij} bzw. \bar{f}_i und unstetige Anteile \hat{b}_{ij} bzw. \hat{f}_i aufgeteilt:

$$b_{ij} = \bar{b}_{ij}(\mathbf{x}, \mathbf{u}, \mathbf{h}, \mathbf{p}, t) + \hat{b}_{ij}(\mathbf{x}, \mathbf{u}, \mathbf{h}, \hat{\mathbf{h}}, \mathbf{p}, \hat{\mathbf{p}}, \mathbf{m}, t), \quad i, j = 1(1)n \quad (3)$$

$$f_i = \bar{f}_i(\mathbf{x}, \mathbf{u}, \mathbf{h}, \mathbf{p}, t) + \hat{f}_i(\mathbf{x}, \mathbf{u}, \mathbf{h}, \hat{\mathbf{h}}, \mathbf{p}, \hat{\mathbf{p}}, \mathbf{m}, t), \quad i = 1(1)n \quad (4)$$

Sowohl die unstetigen Anteile \hat{b}_{ij} bzw. \hat{f}_i von \mathbf{B} bzw. \mathbf{f} als auch die unstetigen Hilfsgrößen h_i und Modellparameter p_j werden durch konditionale Ausdrücke in Abhängigkeit von Schaltparametern m_j dargestellt.

Die erweiterte Modellform (2) besitzt ebenso wie (1) eine feste Modellordnung n . Änderungen der Ordnung während der Simulation sind in DIVA nicht vorgesehen.

Für eine Teilmenge der Schaltparameter m_j mit $k \in K_p \subseteq \{1, 2, \dots, M\}$ nimmt eine unstetige Hilfsgröße h_i bzw. ein unstetiger Parameter p_i verschiedene Werte $f_{i,t}$ bzw. $p_{j,t}$ an, je nachdem welcher der Schaltparameter m_k gerade den Wert 1 hat:

$$\hat{h}_i = \hat{h}_{i,k} \quad \text{falls } m_k = 1 \wedge m_l = 0 \quad \forall l \in K_h \text{ mit } l \neq k \quad (5)$$

$$\text{für } k \in K_h \subseteq \{1, 2, \dots, M\}$$

$$\hat{p}_i = \hat{p}_{i,k} \quad \text{falls } m_k = 1 \wedge m_l = 0 \quad \forall l \in K_p \text{ mit } l \neq k$$

$$\text{für } k \in K_p \subseteq \{1, 2, \dots, M\}$$

Dabei wird vorausgesetzt, dass von den Schaltparametern $1^{\frac{3}{4}}$ für $k \in K^{\wedge, p}$ jeweils immer genau einer den Wert 1 und alle anderen den Wert 0 haben, so dass die Zuordnungen (5) und (6) immer eindeutig sind.

Die Spezifikation der unstetigen Anteile ($\frac{3}{4}$ und $!$; von B bzw. f in Abhängigkeit von Schaltparametern m ; lässt sich ähnlich formulieren:

$$\hat{b}_{ij}(\mathbf{x}, \mathbf{u}, \mathbf{h}, \hat{\mathbf{h}}, \mathbf{p}, \hat{\mathbf{p}}, \mathbf{m}, t) = \begin{cases} \hat{b}_{ij,1}(\mathbf{x}, \mathbf{u}, \mathbf{h}, \hat{\mathbf{h}}, \mathbf{p}, \hat{\mathbf{p}}, t) & \text{falls Bedingung}_{b,1} \\ \dots \\ \hat{b}_{ij,L}(\mathbf{x}, \mathbf{u}, \mathbf{h}, \hat{\mathbf{h}}, \mathbf{p}, \hat{\mathbf{p}}, t) & \text{falls Bedingung}_{b,L} \end{cases} \quad (7)$$

$$\hat{f}_i(\mathbf{x}, \mathbf{u}, \mathbf{h}, \hat{\mathbf{h}}, \mathbf{p}, \hat{\mathbf{p}}, \mathbf{m}, t) = \begin{cases} \hat{f}_{i,1}(\mathbf{x}, \mathbf{u}, \mathbf{h}, \hat{\mathbf{h}}, \mathbf{p}, \hat{\mathbf{p}}, t) & \text{falls Bedingung}_{f,1} \\ \dots \\ \hat{f}_{i,L}(\mathbf{x}, \mathbf{u}, \mathbf{h}, \hat{\mathbf{h}}, \mathbf{p}, \hat{\mathbf{p}}, t) & \text{falls Bedingung}_{f,L} \end{cases} \quad (8)$$

Dabei können als Bedingungen beliebige Wertekombinationen (Bitmuster) verschiedener Schaltparameter η_j , verwendet werden, beispielsweise

$$\begin{aligned} \text{Bedingung}_{b,1} &= (m_2 = 1) \wedge (m_3 = 1) \wedge (m_6 = 0) \\ \text{Bedingung}_{b,2} &= (m_2 = 0) \wedge (m_3 = 1) \wedge (m_6 = 1) \\ &\dots \\ \text{Bedingung}_{b,L} &= (m_2 = 0) \wedge (m_3 = 0) \wedge (m_6 = 1) \end{aligned}$$

Die Bedingungen $\text{Bedingung}^{\wedge}$ bzw. Bedingung^f für $I = 1(1)L$ müssen sich gegenseitig ausschließen, d.h. es muss immer genau eine dieser Bedingungen zutreffen, so dass eine eindeutige Auswahl erfolgen kann.

Die Unstetigkeiten in Hilfsgrößen f_i , Modellparametern p ; sowie in Modellfunktionen G^{\wedge} und f_i lassen sich demnach auf diskrete Änderungen der Schaltparameter m ; zurückführen, welche in den Ereigniszeitpunkten t_{ev} stattfinden. Die Ereigniszeitpunkte werden mit Hilfe von den Schalteingangs- oder \wedge -Funktionen

$$\varphi_e = \varphi_e(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}, \mathbf{h}, \hat{\mathbf{h}}, \mathbf{p}, \hat{\mathbf{p}}, t), \quad e = 1(1)E \quad (9)$$

definiert. Ein Ereignis findet zum Zeitpunkt $t_{ev} > i_0$ statt, wenn die zugehörige y -Funktion (9) das Vorzeichen wechselt:

$$\begin{aligned} t_{ev} : \quad & \varphi_e[\mathbf{x}(t_{ev}), \dot{\mathbf{x}}(t_{ev}), \mathbf{u}(t_{ev}), \mathbf{h}, \hat{\mathbf{h}}, \mathbf{p}, \hat{\mathbf{p}}, t_{ev}] = 0 \\ & \wedge \text{sgn}[\varphi_e(t_{ev}^-)] \neq \text{sgn}[\varphi_e(t_{ev}^+)], \quad e \in \{1, 2, \dots, E\} \end{aligned} \quad (10)$$

Zur Definition von *expliziten* Ereignissen genügt die Angabe der bekannten Zeitpunkte t_{ev} . In den übrigen Fällen spricht man von *impliziten* Ereignissen.

Die Änderungen der Schaltparameter m , in den Ereigniszeitpunkten i_e , werden durch Schaltfunktionen S definiert:

$$\mathbf{m}(t_{ev}^+) = S[\mathbf{m}(t_{ev}^-), \varphi_1^+, \varphi_1^-, \varphi_2^+, \varphi_2^-, \dots, \varphi_E^+, \varphi_E^-] \quad (11)$$

Diese Funktionen bestimmen in Abhängigkeit von Werten $m(t_{ev}^-)$ der Schaltparameter vor einem Ereignis und den eingetretenen Vorzeichenwechseln der \wedge -Funktionen (9) neue Werte $m(<+)$ für die Schaltparameter. In (11) bezeichnet φ^+ (φ^-) für $e \in \{1, 2, \dots, E\}$ einen Nulldurchgang der y -Funktion $<p_e$ in positiver (negativer) Richtung. Zu den Schaltfunktionen S gehören die Anfangsbedingungen für die Schaltparameter:

$$\mathbf{m}(t_0) = \mathbf{m}_0$$

Die Anfangsbedingungen bilden zusammen mit den Schaltfunktionen (11) das *Schaltwerk* zur Definition der ereignisdiskreten Vorgänge in einem gemischt kontinuierlich-ereignisdiskreten Modell.

Die binären Schaltparameter m_i können als Stellen eines Petri-Netzes (Bedingungs®reignis-Systems) [9] interpretiert werden. Damit ist eine Darstellung des kompletten Schaltwerks (11) als Petri-Netz möglich. Darauf wird im Zusammenhang mit der Codierung gemischt kontinuierlich-ereignisdiskreter Modelle für DivA mit Hilfe des Code-Generators im Abschnitt 4 eingegangen.

3 Implementierung von unstetigen Prozessmodellen in DivA

Im folgenden wird die Architektur der Simulationsumgebung DivA soweit erklärt, wie dies für Beschreibung der Implementation und Behandlung von unstetigen Prozessmodellen erforderlich ist. Danach wird gezeigt, wie die mathematische Darstellung (2)-(12) solcher Modelle in DivA programmtechnisch realisiert ist. Darauf aufbauend wird die numerische Behandlung von Ereignissen in DivA erläutert.

3.1 DivA Grundelemente für unstetige Prozessmodelle

In der Simulationsumgebung DivA für verfahrenstechnische Anlagen und Prozesse werden in Anlehnung an ein modulares Anlagenfließbild die mathematischen Modelle einzelner Anlagenteile (Apparate, Blöcke) miteinander verkoppelt [2, 3, 5]. Das dynamische Verhalten einzelner Blöcke sowie der gesamten Anlage wird durch die Differential-Algebra-Gleichungen (DAE) (1) bzw. (2-12) beschrieben. Diese sind als sogenanntes DAE-Blockmodell, welches auch als DivA-Grundelement bezeichnet wird, in mehreren FORTRAN-Unterprogrammen codiert (siehe Abb. 1). Für die vollständige Beschreibung eines kontinuierlichen generischen Blockmodells sind die FORTRAN-Unterprogramme zur Modellinitialisierung (*I-Teil*), Funktionsauswertung (*F-Teil*) und Datenausgabe (*A-Teil*) obligatorisch. Im *I-Teil* erfolgen die Parametrierung, Adressberechnung und die Belegung der Struktur der Jacobi-Matrizen des Blockmodells. Der *F-Teil* berechnet die funktionalen Ausdrücke des DAE-Systems. Der *A-Teil* wird für die Ausgabe der

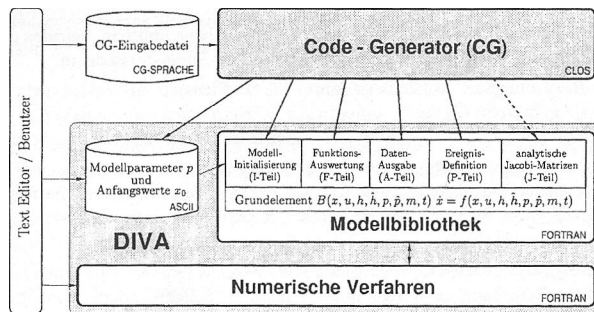


Abbildung 1: Architektur der Simulationsumgebung DivA mit den drei Ebenen: Numerische Verfahren, Modellbibliothek für Grandelemente mit der Ereignisdefinition im P-Teil und Code-Generator [7,8, 10].

Simulationsdaten benötigt. Für die Auswertung analytischer Jacobi-Matrizen, die bei der numerischen Integration benötigt werden, kann optional ein *J-Teil* ergänzt werden. Für die Definition von Blockmodellen mit gemischt kontinuierlich-ereignisdiskreten Funktionen ist zusätzlich ein FORTRAN-Unterprogramm zur Ereignis-Definition (*P-Teil*) erforderlich. Außerdem wird zur Parametrierung eines generischen Blockmodells ein Blockdatenfile mit den Modellparametern und Anfangswerten der Modellzustände benötigt.

In der Modellbibliothek liegen einige Grundelemente bereits fertig programmiert vor. Um neue Blockmodelle zu programmieren, wurde für die Simulationsumgebung DIVA der Code-Generator (CG) [10,7,8] entwickelt, der aus einer Eingabedatei die benötigten FORTRAN-Unterprogramme sowie das Blockdatenfile automatisch erzeugt (Abb. 1).

Nachfolgend wird genauer betrachtet, wie die Modellgleichungen (2-12) in den einzelnen FORTRAN-Unterprogrammen enthalten sind. Dazu zeigt Abbildung 2 die Definition der verschiedenen Teile gemischt kontinuierlich-ereignisdiskreter Modelle in den einzelnen FORTRAN-Unterprogrammen. Außerdem ist dort das Zusammenspiel der Unterprogramme mit der DIVA-Numerik dargestellt, worauf im folgenden Kapitel eingegangen wird.

Die von den Schaltparametern τ_j abhängigen Elemente der linken und rechten Seite des DAE-Systems (2) werden im F-Teil definiert. Optional sind im J-Teil die von den Schaltparametern m abhängigen analytischen Ausdrücke für die Jacobi-Matrix-Einträge codiert. Die Zuweisung der Werte der unstetigen Hilfsgrößen h und der unstetigen Modellparameter p in Abhängigkeit von Schaltparametern m ; entsprechend (5) bzw. (6) erfolgt im P-Teil, in dem außerdem die Ereigniszeiten t_{ev} entsprechend (10) mit Hilfe von \wedge -Funktionen (9) oder durch Angabe expliziter Ereigniszeitpunkte t_{ex} definiert werden. Die Initialisierung (12) des Schaltwerks zum Zeitpunkt t_0 erfolgt im I-Teil, die Schaltfunktionen (11) werden wiederum im P-Teil definiert.

3.2 Behandlung von Ereignissen in DIVA

Zur Erklärung der Ereignisbehandlung bei der Simulation in DIVA wird zunächst der für unstetige Gmdelemente erweiterte Programmablauf bei der numerischen Integration betrachtet. Dazu zeigt Abbildung 2 das nachfolgend beschriebene Zusammenspiel der DIVA-DAE-Numerik mit den FORTRAN-Unterprogrammen eines Gmdelements mit Unstetigkeiten:

- Die Initialisierung des Schaltwerks gemäß (12) erfolgt beim üblichen Aufruf des I-Teils zur Initialisierung und Parametrierung des Modells vor Simulationsbeginn.
- Bei der Auswertung der funktionalen Ausdrücke der Elemente der rechten und linken Seiten des DAE-Systems (2) durch Aufrufen des F-Teils werden konditionale Ausdrücke der Form (3) bzw. (4) mit (7) bzw. (8) in Abhängigkeit der Schaltparameter r_n bestimmt.
- Nach jedem Integrationsschritt wird der P-Teil aufgerufen, um die Werte der v -Funktionen (9) sowie die nächste explizite Ereigniszeit t_{ex} zu bestimmen. Außerdem erfolgt die Zuweisung der Werte der unstetigen Hilfsgrößen h gemäß (5) sowie der unstetigen Modellparameter p gemäß (6).
- In den Ereigniszeitpunkten t_{ev} wird ebenfalls der P-Teil aufgerufen. Zusätzlich zu den oben angeführten Berechnungen werden hierbei sämtliche Schaltfunktionen (11) ausgewertet und damit die neuen Werte der Schaltparameter m ; bestimmt.

Demnach hat der P-Teil je nach Zeitpunkt des Aufrufs durch die DIVA-Numerik-Routinen verschiedene Aufgaben zu erfüllen: Bei jedem Aufruf werden die y -Funktionen (9) ausgewertet, die nächste explizite Ereigniszeit t_{ex} bestimmt und die Werte der unstetigen Hilfsgrößen h und Parameter p zugewiesen. Dagegen werden die Schaltfunktionen (11) nur beim Eintreten eines Ereignisses ausgewertet. Welche Aufgaben der P-Teil bei einem Aufruf zu erfüllen hat, wird

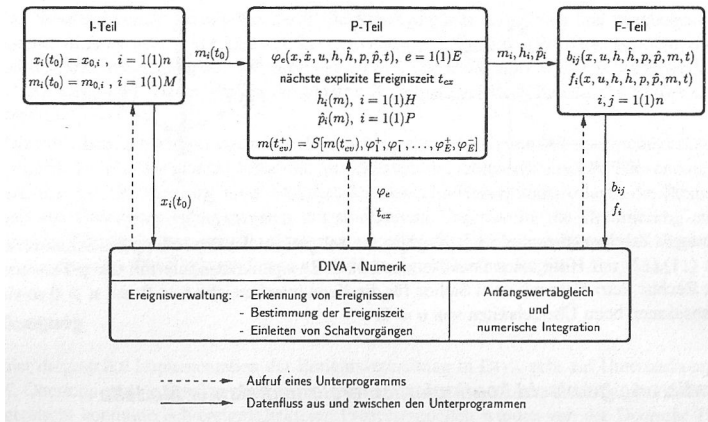


Abbildung 2: Zusammenspiel der DIVA-DAE-Numerik mit den FORTRAN-Unterprogrammen zur Modell-Initialisierung (I-Teil), Ereignis-Definition (P-Teil) und Funktions-Auswertung (F-Teil).

durch das Setzen bestimmter *Schalttags* durch die DIVA-Ereignisverwaltung bestimmt. Die Ereignisbehandlung geschieht dann in den folgenden drei Schritten:

- **Erkennen von Ereignissen:** Bei jedem Integrationsschritt werden die Vorzeichen aller ψ -Funktionen (9) auf eine Änderung überprüft. Ein Vorzeichenwechsel zeigt gemäß (10) ein Ereignis an.
- **Bestimmung des Schaltzeitpunkts t_{ev} :** Durch einen iterativen Suchalgorithmus wird das Intervall, in dem der erkannte Vorzeichenwechsel liegt, verkleinert. Der Zeitpunkt t_{ev} des Ereignisses gilt als hinreichend genau bestimmt, wenn der Betrag der ψ -Funktion eines vom Benutzer definierten Toleranzwert ϵ_ψ nicht überschreitet und die ψ -Funktion ihr Vorzeichen gegenüber dem letzten Integrationsschritt geändert hat:

$$|\varphi(t_{ev})| \leq \epsilon_\psi \text{ und } \text{sgn}[\varphi(t_{k-1})] \neq \text{sgn}[\varphi(t_{ev})] \quad (13)$$

- **Ausführung des Schaltvorgangs:** Ist die Integration an einem Schaltzeitpunkt t_{ev} angelangt, so werden durch Auswerten der Schaltfunktionen (11) die neuen Werte für die Schaltparameter z_{ij} zugewiesen. Falls $\dot{x}(t)$ und $x(t)$ das dadurch veränderte DAE-System nicht mehr erfüllen, wird ein Abgleich zur Bestimmung konsistenter Anfangswerte für die weitere Integration vorgenommen. Die Änderungen an den Schaltparametern m_{ij} oder der Abgleich können weitere Vorzeichenwechsel in den ψ -Funktionen und damit weitere implizite Ereignisse verursachen. In diesem Fall spricht man von *statischem Schalten*. Die entsprechenden Schaltvorgänge werden ebenfalls ausgeführt.

Für explizite Ereignisse erübrigen sich die beiden ersten Schritte. Stattdessen wird nur der jeweils nächste explizite Ereigniszeitpunkt t_{ex} betrachtet und parallel zur Erkennung impliziter Ereignisse höchstens bis zu diesem Zeitpunkt integriert.

Neben der Ereignisbehandlung bei der numerischen Integration sind in DIVA auch spezielle Algorithmen zur Berechnung von Parameter-Sensitivitäten von unstetigen Modellen implementiert. Diese Sensitivitäten werden für die Lösung von Optimierungsaufgaben benötigt.

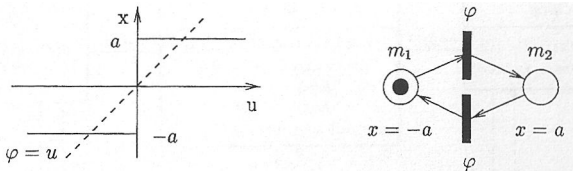


Abbildung 3: Zweipunktkenlinie $\chi = a \cdot \text{sgn}[u]$ als Beispiel für die Darstellung des Schaltwerks S nach (11),(12) mit Hilfe eines Petri-Netzes. Links: Zweipunktkenlinie mit der χ -Funktion $\psi = u$; Rechts: Petri-Netz mit zwei Stellen für die Kennlinienbereiche $u < 0$ und $u > 0$ sowie den Transitionen beim Überschreiten von $u = 0$.

4 Codierung gemischt kontinuierlich-ereignisdiskreter Modelle

Für die automatische Codierung gemischt kontinuierlich-ereignisdiskreter Modell mit Hilfe des für DiVA entwickelten Code-Generators wird das Schaltwerk (11), (12) als Petri-Netz dargestellt [61], wobei hier ausschließlich sogenannte Bedingungsfl-Ereignis-Systeme (Netze) [9] gemeint sind. Ein solches Petri-Netz besteht aus Stellen (Kreise) und Transitionen (Balken), einer Flussrelation (Pfeile) sowie einer Anfangsmarkierung (schwarze Punkte). Abbildung 3 zeigt ein Petri-Netz für das Beispiel einer Zweipunktkenlinie.

Bei der Darstellung eines Schaltwerks entspricht eine Stelle einem Schaltparameter m_i . Diese kann entweder eine Marke enthalten ($m_i^* = 1$) oder frei sein ($m_i = 0$). Die Anfangsmarkierung $m_i(<0)$ des Petri-Netzes entspricht der Anfangsbedingung (12). Transitionen entfernen entsprechend der Flussrelation Marken in den sogenannten Eingangsstellen und setzen in den Ausgangsstellen neue Marken. Dieser Vorgang ist an Ereigniszeitpunkte gekoppelt. Transitionen ändern also die Markierung im Petri-Netz und damit die Werte der Schaltparameter m beim Auftreten von Ereignissen. Somit repräsentieren die Parameterwerte zusammen mit der Flussrelation die Schaltfunktionen (11), d.h. das Verhalten des Schaltwerks beim Eintreten von Ereignissen.

Ein Schaltwerk S wird in Form eines oder mehrerer Petri-Netze in der Code-Generator-Eingabedatei definiert. Hierzu werden zunächst die Stellen, die Anfangsmarkierung (12) und schließlich alle Ereigniszeitpunkte in Form von $y>$ -Funktionen (9) oder expliziten Zeitpunkten t_{ex} angegeben. Die Flussrelationen der Transitionen werden zusammen mit den zugehörigen Ereigniszeitpunkten definiert. Bei der Definition von unstetigen Gleichungsteilen gemäß (7) und (8) sowie der Werte unstetiger Hilfsgrößen (5) und Modellparameter (6) werden die Stellen m_i als Boolesche Variablen in konditionalen Ausdrücken verwendet.

5 Zusammenfassung

Gemischt kontinuierlich-ereignisdiskrete Prozessmodelle werden in der Simulationsumgebung DiVA in einem linear-impliziten differential-algebraischen Modell dargestellt, das mit unstetigen Elementen erweitert ist. Die unstetigen Elemente umfassen Modellparameter, Hilfsvariablen sowie Anteile in den Elementen der Matrix B und den Komponenten des Vektors f . Die Ereignisse werden durch den Vorzeichenwechsel von Schalteingangsfunktionen ψ ausgelöst. Ein Ereignis führt zu einer Neubestimmung von binären Schaltparametern mit Hilfe eines Schaltwerks. Mittels der binären Schaltparameter werden die Werte und Funktionen der o.g. unstetigen Elemente festgelegt. Zur Erkennung von Ereignissen ist in DiVA eine spezielle Ereignisbehandlung implementiert. Um numerische Probleme aufgrund von Unstetigkeiten zu verhindern, wer-

den die Schaltzeitpunkte iterativ bestimmt, die Schaltvorgänge ausgeführt und gegebenenfalls konsistente Anfangswerte neu berechnet. Zur automatischen Codierung gemischt kontinuierlich-ereignisdiskreter Prozessmodelle mit Hilfe des DIVA-Code-Generators wird das Schaltwerk als Petri-Netz dargestellt. Dabei werden die binären Schaltparameter als Stellen des Petri-Netzes interpretiert.

Die beschriebenen Möglichkeiten zur Behandlung gemischt kontinuierlich-ereignisdiskreter Prozessmodelle in DIVA wurden bisher für verschiedene verfahrenstechnischen Anwendungen genutzt, u.a. für die Steuerung einer Mehrbett-Druckwechselladsorptions-Anlage, die Prozessführung von Chromatographieprozessen mit simuliertem Gegenstrom, die Optimierung einer präparativen Flüssigchromatographie, Rezeptsteuerungen für Batchprozesse und das automatische Anfahren einer Destillationskolonne mit Hilfe ereignisdiskreter Regler.

Danksagung

Die hier dargestellte Implementation der Ereignisverwaltung in DIVA geht auf Untersuchungen von T. Obertopp und A. Spieker zurück, was hiermit dankend anerkannt wird. Die Behandlung von gemischt kontinuierlich-ereignisdiskreten Prozessmodellen wurden von der Deutsche Forschungsgemeinschaft im SFB412gefordert.

Literatur

- [1] P. I. Barton and C. C. Pantelides. The modelling of combined discrete/continuous processes. *AIChEJournal* 40:966-979, 1994.
- [2] K. D. Mohl, A. Spieker, R. Köhler, E. D. Gilles, and M. Zeitz. DWA - A simulation environment for chemical engineering applications. In *Informatics, Cybernetics and Computer Science (ICCS-97). Collected Volume of Scientific Papers*, pages 8-15. Donetsk State Technical University, Donetsk/Ukraine, 1997.
- [3] K. D. Mohl, A. Spieker, E. Stein, and E. D. Gilles. DIVA - Eine Umgebung zur Simulation, Analyse und Optimierung verfahrenstechnischer Prozesse. In A. Kuhn and S. Wenzel, editors, *Simulationstechnik, 11. ASIM-Symposium in Dortmund*, pages 278-283. Vieweg Verlag, Braunschweig/Wiesbaden, 1997.
- [4] T. Obertopp, A. Spieker, and E. D. Gilles. Optimierung hybrider Prozesse in der Verfahrenstechnik. In *UMSICHT-Schriftenreihe Band 7, Oberhausener UMSICHT-Tage „Rechneranwendungen in der Verfahrenstechnik“, 15.-16. September 1998*, pages 5.1-5.18, Stuttgart, 1998. Fraunhofer IRB Verlag.
- [5] R. Köhler, K.D. Mohl, H. Schramm, M. Zeitz, A. Kienle, M. Mangold, E. Stein, and E.D. Gilles. Method of lines within the simulation environment diva for chemical processes. In W. Schiesser, A. Vande Wouwer, and P. Saucez, editors, *Adaptive Method of Lines*. CRC Press, Boca Raton/USA, 2001.
- [6] Stephan Küpper. Automatische Codierung gemischt-kontinuierlich-ereignisdiskreter Prozessmodelle für die Simulationsumgebung DIVA. Studienarbeit, ISR, Universität Stuttgart, 2000.
- [7] S. Räumsschüssel. *Rechnerunterstützte Vorverarbeitung und Codierung verfahrenstechnischer Modelle für die Simulationsumgebung DTV.A*. Fortschritt-Berichte Nr. 20/270, VDI-Verlag, Düsseldorf, 1998.
- [8] R. Köhler. Beschreibung Code-Generator Version 2.0 zur Erzeugung von DIVA-Grundelementen. Interner ISR-Bericht 00-2, Universität Stuttgart, September 2000.
- [9] B. Baumgarten. *Petri-Netze: Grundlagen und Anwendungen*. Spektrum Akademischer Verlag, 2nd edition, 1996.
- [10] R. Köhler, S. Räumsschüssel, and M. Zeitz. Code generator for implementing differential algebraic models used in the process simulation tool DFVA. In A. Sydow, editor, *15th IMACS World Congress*, vol. 3, 621-626, Berlin, Germany, 1997. Wissenschaft und Technik Verlag.