

# Funktionswurzeln und Lernalgorithmen für abgetastete Systeme

Roland Kempf, Harald Staab

Technische Universität Darmstadt, Institut für Automatisierungstechnik,  
Landgraf-Georg Straße 4, D-63482 Darmstadt  
Telefon: (06151) 16-3542, Fax: (06151) 16-2507,  
E-Mail: rdkempf@rtr.tu-darmstadt.de

## Zusammenfassung:

Durch die zeitliche Abtastung eines zeit- und wertekontinuierlichen Systems erhält man ein zeitdiskretes System. Kann man während des Zeitintervalls den Eingangsgrößenvektor  $\mathbf{u}$  als konstant voraussetzen, so läßt sich ein funktionaler Zusammenhang zwischen der alten und neuen Zustandgröße angeben:  $\mathbf{x}(k+1) = \mathbf{f}_{\mathbf{u}}(\mathbf{x}(k))$ , wobei der Eingangsgrößenvektor  $\mathbf{u}$  als Scharparameter für die Funktionen  $\mathbf{f}_{\mathbf{u}}$  betrachtet werden kann.

Die Menge der abgetasteten Systeme bildet allerdings nur eine kleine Teilmenge in der Menge der zeitdiskreten Systeme. Sie zeichnen sich unter anderem auch dadurch aus, dass die Transitionsfunktionen  $\mathbf{f}_{\mathbf{u}}$  beliebige sogenannte Funktionswurzeln besitzen muß, d.h. es gibt Funktionen  $\mathbf{f}_{\mathbf{u}}^{1/q}(\mathbf{x})$ , so dass die  $q$ -fache Hintereinanderschaltung  $\mathbf{f}_{\mathbf{u}}^{1/q} \circ \mathbf{f}_{\mathbf{u}}^{1/q} \circ \dots \circ \mathbf{f}_{\mathbf{u}}^{1/q}$  genau die Funktion  $\mathbf{f}_{\mathbf{u}}$  ergibt.

Will man nun ein solches abgetastetes System identifizieren, so muß das Ergebnis diese Eigenschaft bzw. daraus abgeleitete Eigenschaften besitzen. Dies schränkt den Suchraum während einer Identifikation erheblich ein. Zum Teil können dadurch auch unerwünschte dynamische Verhaltensweisen, wie etwa chaotisches Verhalten, bei den Systemen vermieden werden, die sich andernfalls während des Identifikationsprozesses ergeben können. Besonders bei Lernalgorithmen, die ohne eine vollständige Beschreibung der internen Zustandsgrößen auskommen, wie z.B. "backpropagation through time", erhöht sich dadurch die Lerngeschwindigkeit und Erfolgsquote.

In diesem Beitrag werden einige Grundlagen im Bereich der Funktionswurzeln angesprochen und dargestellt wie man die Eigenschaften von Funktionen mit beliebigen Funktionswurzeln ausnutzen kann, um Lernverfahren zu modifizieren. Getestet wird ein modifizierter "backpropagation through time"-Algorithmus zur Identifikation eines einfachen Neuro-Fuzzy Systems.

# 1 Problemstellung

Bei der Identifikation abgetasteter wertekontinuierlicher Systeme ist das Ziel eine Funktion  $f$  zu finden, die den Übergang des Zustandes  $x(k)$  eines Systems zum Zeitpunkt  $k \cdot \Delta t$  auf den Zustand  $x(k + 1)$  zum Zeitpunkt  $(k + 1) \cdot \Delta t$  je nach Belegung der möglichen Eingabewerte  $u(k)$  beschreibt, also:

$$x(k + 1) = f(x(k), u(k)), \text{ wobei } f \text{ stetig und } k \in \mathbb{N}. \quad (1)$$

Die Gleichung (1) beschreibt allerdings eine weit größere Funktionsklasse als die der abgetasteten wertekontinuierlichen Systeme. Betrachtet man verschiedene stetige Funktionen  $f$ , so können nach Anwendung der Gleichung (1) sehr unterschiedliche Dynamiken entstehen. Insbesondere

- konvergentes Verhalten, z.B. für  $f(x) = 0.5x + 0.25$ ,
- divergentes Verhalten, z.B. für  $f(x) = 2x - 0.5$ ,
- oszillatorisches Verhalten, z.B. für  $f(x) = 1 - x$ , und sogar
- chaotisches Verhalten, z.B. für  $f(x) = \begin{cases} 2x & \text{für } x \leq 1, \\ 2 - 2x & \text{für } x > 1. \end{cases}$

Abbildung 1 zeigt die Funktionen und Zeitreihen, die durch sie definierten dynamischen Systeme.

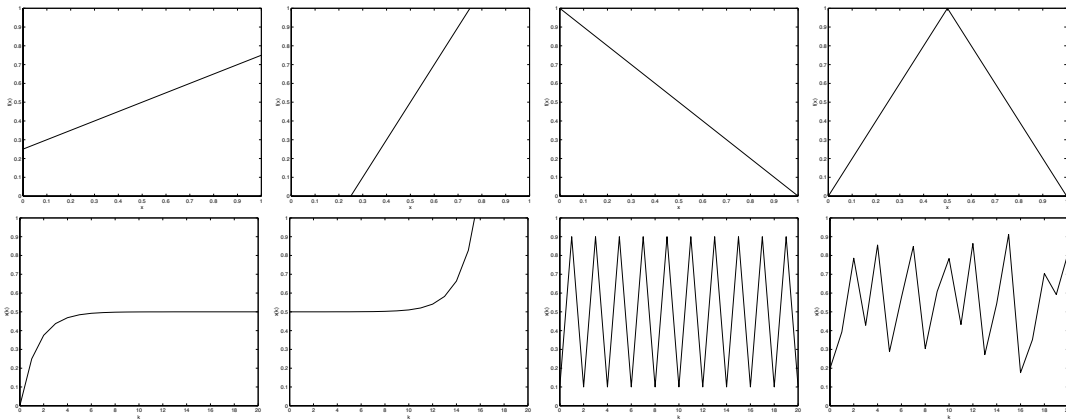


Abbildung 1: Verschiedene Funktionen (oben) erzeugen Zeitreihen (unten) mit qualitativ unterschiedliche Dynamiken.

Allerdings tauchen nicht alle dieser Verhaltensweisen auch in abgetasteten wertekontinuierlichen Systemen auf. Bei der Identifikation ist daher darauf zu achten, Funktionstypen mit solchen Verhalten auszuschließen. Dazu muß man die erlaubten Verhaltensweisen, bzw. die zugelassenen Funktionen  $f$  näher beschreiben. Wie im nächsten Abschnitt erläutert wird, besitzen abgetastete wertekontinuierliche Systeme, im Gegensatz zu vielen zeitdiskreten Systemen, beliebige sogenannter Funktionswurzeln.

## 2 Existenz beliebiger Funktionswurzeln

Betrachtet wird ein abgetastetes wertekontinuierliches System, dessen Eingabewerte  $u$  während eines Abtastschrittes als konstant vorausgesetzt wird. Dieses System wird eindeutig durch eine Gleichung der Form wie Gleichung (1) bzw. durch

$$x(T + \Delta t) = f_u(x(T)) \quad (2)$$

mit  $T = k \cdot \Delta t$  beschrieben.

Tastet man das System mit der doppelten Abtastrate ab, so ergibt sich eine andere Funktion  $f_u^{1/2}$  die, anstelle von  $f_u$ , das System beschreibt. Mit dieser Funktion gilt:

$$x(T + \frac{1}{2} \cdot \Delta t) = f_u^{1/2}(x(T)) \quad (3)$$

$$x(T + \Delta t) = f_u^{1/2}(x(T + \frac{1}{2} \cdot \Delta t)) \quad (4)$$

und somit

$$x(T + \Delta t) = f_u^{1/2}(f_u^{1/2}(x(T + \Delta t))) \quad (5)$$

Vergleicht man Gleichung (2) und Gleichung (5), so erhält man die folgende Beziehung zwischen den beiden Funktionen  $f_u$  und  $f_u^{1/2}$ :

$$f_u = f_u^{1/2} \circ f_u^{1/2}. \quad (6)$$

Bei  $q$ -facher Abtastrate erhält man analog Funktionen  $f_u^{1/q}$  mit

$$f_u = \underbrace{f_u^{1/q} \circ \dots \circ f_u^{1/q}}_{q\text{-fach}}. \quad (7)$$

Die Funktionen  $f_u^{1/q}$  nennt man auch  $q$ -te Funktionswurzeln von  $f_u$ , weil in Analogie zur Multiplikation ihre  $q$ -fache Anwendung der ursprünglichen Funktion  $f_u$  gleicht. Durch die  $p$ -fache Hintereinanderschaltung der Funktion  $f_u^{1/q}$  sind dann auch die Funktionen  $f_u^{p/q}$  festgelegt.

Für eine Übergangsfunktion  $f_u$  eines abgetasteten wertekontinuierlichen Systems muß es also für jeden positiven Bruch  $p/q$  stetige Funktionen  $f_u^{p/q}$  geben.

## 3 Auswirkung der Existenz von Funktionswurzeln auf abgetastete wertekontinuierlicher Systeme

Exemplarisch werden die Auswirkungen der Existenz von Funktionswurzeln auf abgetastete wertekontinuierlicher Systeme nun anhand eines linearen, eindimensionalen System demonstriert. Anschließend werden nichtlineare, eindimensionale Systeme behandelt.

### 3.1 Lineares System

Ein lineares, eindimensionales System läßt sich bei in einem Abtastschritt festgesetzten Eingabewerten  $u$ , nach Transformation einer Ruhelage in den Ursprung allgemein wie folgt beschreiben:

$$x(T + \Delta t) = f_u(x(T)), \text{ mit } f_u(x) = a_1 \cdot x. \quad (8)$$

Die Übergangsfunktion der doppelt abgetasteten Systems wird nun durch eine Funktionswurzel von  $f_u(x)$  beschrieben. Da das doppelt abgetastete System selbst eindimensional und linear sein muß, ist die Funktionswurzel  $f_u^{1/2}$  durch folgenden Ansatz beschreibbar:

$$f_u^{1/2}(x) = a_2 \cdot x. \quad (9)$$

Nach Gleichung (5) ergibt sich dann

$$a_1 \cdot x = f_u(x) = f_u^{1/2}(f_u^{1/2}(x)) = a_2 \cdot a_2 \cdot x \quad (10)$$

und somit  $a_1 = a_2^2$ . Damit die lineare Funktionswurzel  $f_u^{1/2}(x)$  existiert, muß daher  $a_1 \geq 0$  sein. Analog muß auch  $a_2 \geq 0$  sein, damit eine vierte lineare Funktionswurzel  $f^{1/4}(x)$  von  $f(x)$  existiert, die gleichzeitig eine zweite lineare Funktionswurzel von  $f^{1/2}(x)$  darstellt. Somit ist der Parameter  $a_2$  durch  $a_2 = +\sqrt{a_1} = a_1^{1/2}$  eindeutig festgelegt.

Analog überlegt man sich, dass ungerade  $q$ -fache lineare Funktionswurzeln von  $f_u$  durch  $f^{1/q}(x) = a_1^{1/q} x$  eindeutig festgelegt sind. Kombiniert man die Überlegungen zu geraden und ungeraden Funktionswurzeln von  $f_u$  so ergibt sich  $f^{1/q}(x) = a_1^{1/q} x$  und  $f^{p/q}(x) = a_1^{p/q} x$  für beliebige Brüche  $p/q$ , wobei  $p, q \in \mathbb{N}$ .

Da das Systemverhalten stetig in der Zeit ist, können in diesem Fall auch "reellen Funktionswurzeln" definiert werden durch:

$$f_u^{t/\Delta t}(x) = a_1^{t/\Delta t} x = e^{t/\Delta t \cdot \ln(a_1)} x = e^{t/\tau} x, \text{ wobei } \tau = \frac{\Delta t}{\ln(a_1)}. \quad (11)$$

Schreibt man Gleichung (11) um, so erhält man die bekannte Darstellung des Zeitverlaufs der Zustandsgröße  $x(T + t)$  zwischen zwei Abtastzeiten  $T$  und  $T + \Delta t$  durch:

$$x(T + t) = f_u^{t/\Delta t}(x(T)) = e^{t/\tau} x(T), \text{ mit der Zeitkonstanten } \tau = \frac{\Delta t}{\ln(a_1)}. \quad (12)$$

Durch die Forderung, dass alle Funktionswurzeln auch linear sein sollen, ist es somit möglich, den kontinuierlichem Zeitverlauf der Zustandsgröße eindeutig zu rekonstruieren. Bei den Untersuchungen wurde außer Acht gelassen, dass es womöglich nichtlineare Funktionswurzeln von linearen Funktionen gibt. Im nichtlinearen Fall ist diese Frage allerdings wichtig.

### 3.2 Nichtlineares System

Bei nichtlinearen Systemen ist die Bestimmung der Funktionswurzeln aufgrund eines fehlenden Ansatzes nicht so einfach möglich, wie die Bestimmung von linearen Funktionswurzeln von linearen Systemen. Funktionswurzeln sind in der Regel auch nicht eindeutig

festgelegt [4, 2]. Trotzdem lassen sich noch einige Aussagen über eindimensionale, nichtlineare abgetastete Systeme treffen. Es gilt folgender

Satz:

Eine eindimensionale, stetige Funktion  $f$ , die ein reelles, beschränktes und abgeschlossenes Intervall  $X$  in sich selbst abbildet besitzt nur dann beliebige stetige Funktionswurzeln  $f^r$  mit  $r \in \mathbb{R}$  in  $X$ , wenn die Funktion die Funktion  $f$  monoton steigend auf  $X$  ist.

Beweis:

Angenommen die Funktion  $f$  ist nicht monoton steigend  $X$ . Dann gibt es Punkte  $x_1 < x_2$  aus  $X$  mit  $f(x_1) > f(x_2)$ . Da  $f^t(x)$  stetig in  $t$  ist, gibt es auch ein  $t \in (0, 1)$ , so dass  $f^t(x_1) = f^t(x_2)$  und damit folgt  $f(x_1) = f^{1-t}(f^t(x_1)) = f^{1-t}(f^t(x_2)) = f(x_2)$  im Widerspruch zu  $f(x_1) > f(x_2)$ .  $\square$

Dieser Satz hat eine große Auswirkung auf mögliche Dynamikarten von eindimensionalen, nichtlinearen abgetasteten Systemen, welche sich durch eine solche Funktion  $f$  beschreiben lassen. Denn nach [1] sind solche Systeme entweder konvergent oder bestimmt divergent. Oszillationen oder gar chaotische Dynamiken sind ausgeschlossen.

## 4 Anwendungsbeispiele

Die Überlegungen aus den vorangegangenen Abschnitten wurden benutzt, um den Lernalgorithmus für ein rekurrentes NeuroFuzzy-System zu modifizieren und damit die Geschwindigkeit und Erfolgsquote für Lernvorgänge zu verbessern.

### 4.1 Aufbau der NeuroFuzzy-Systeme

Die benutzten NeuroFuzzy-Systeme haben je eine Eingangsgröße  $u$  und eine Zustandsgröße  $x$ . In der ersten Neuronenschicht findet die Fuzzifizierung der Größen  $x$  und  $u$  statt, indem die Zugehörigkeitsfunktionen  $\mu_j^x(x)$  und  $\mu_p^u(u)$  ausgewertet werden. Die nächste Neuronenschicht bestimmt durch Multiplikation der entsprechenden Zugehörigkeitswerte die Aggregation der Wahrheitswerte für die einzelnen Regeln. Die Implikation wird durch die Gewichte  $r_{q,j}$  auf der Verbindungen zum Summationsneuron realisiert. Das Summationsneuron schließlich übernimmt Akkumulation und Defuzzifizierung gleichzeitig. Die Zugehörigkeitsfunktionen werden dabei so gewählt, dass der Nenner, der in der Defuzzifizierungsmethode "Center of Singletons" (CoS) [5] auftritt, den Wert 1 für alle Belegungen von  $x$  und  $u$  hat. Als Zugehörigkeitsfunktionen werden hier durchgehend normierte Dreiecksfunktionen verwendet. Der schematische Aufbau des NeuroFuzzy-Systems ist in Abbildung 2 zu sehen.

Der Ausgang  $v$  wird nun zeitverzögert als Eingangswert  $x$  zurückgeführt. Damit entsteht ein dynamisches System, welches in Anlehnung an Gleichung (1) folgende Gleichung realisiert:

$$x(k+1) = v(k) = f(x(k), u(k)), \text{ wobei } f \text{ stetig und } k \in \mathbb{N}. \quad (13)$$

Die Parameter  $r_{q,j}$  dieses Systems sollen nun so bestimmt werden, dass es vorgegebene Zeitreihen approximiert. Die Bestimmung erfolgt mit Hilfe des Backpropagation-Algorithmus.

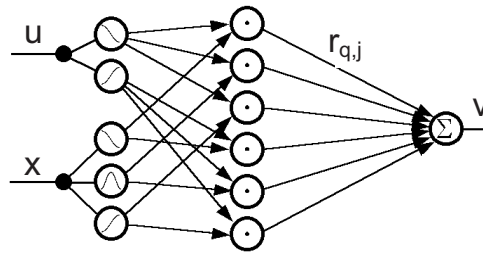


Abbildung 2: Schematischer Aufbau des NeuroFuzzy-Systems.

## 4.2 Backpropagation-Algorithmus

Die Funktion  $f$  läßt sich durch die Singletonpositionen  $r_{q,j}$  parametrisieren und wie folgt schreiben:

$$f(x, u) = \sum_{q,j} r_{q,j} \cdot \mu_j^x(x) \cdot \mu_q^u(u). \quad (14)$$

In jedem Zeitschritt wird das Ergebnis  $v(k) = f(x(k), u(k))$  mit dem gewünschten Ergebnis  $\tilde{v}(k)$  verglichen und der quadratische Fehler  $e(k) = (v(k) - \tilde{v}(k))^2$  über die Zeit zu einem Gesamtfehler  $e_{ges}$  aufsummiert. Anschließend werden die Singletonpositionen  $r_{q,j}$  im  $l$ -ten Lernschritt nach der Gleichung

$$r_{q,j}(l+1) = r_{q,j}(l) - h \cdot \frac{\partial e_{ges}(r_{q,j})}{\partial r_{q,j}} \quad (15)$$

verändert. Mit Hilfe der Kettenregel ergibt sich einfach

$$r_{q,j}(l+1) = r_{q,j}(l) - h \cdot \sum_k (v(k) - \tilde{v}(k)) \cdot \mu_j^x(x(k)) \cdot \mu_q^u(u(k)). \quad (16)$$

Es wurde bewußt kein seriell-paralleles Lernen [3] verwendet, bei dem nach jedem Schritt der Zustand  $x(k)$  im Modell auf den Wert  $\tilde{v}(k)$  des zu identifizierenden Systems gesetzt wird anstatt den Modellausgang  $v(k)$  zu benutzen, um das Verfahren dann auf Systeme anwenden zu können, deren Zustand sich zum Teil nicht beobachten läßt.

Die Schrittweite  $h$  wurde per SuperSAB-Verfahren adaptiert, weil dies zu besseren Ergebnissen als bei konstanter Schrittweite, bzw. mit einem "momentum term" oder "resilient backpropagation" [6] führte.

## 4.3 Diesel Turbolader

Für das erste Beispiel lagen Daten des Turboladers eines Dieselmotors vor. Die Einspritzmenge  $v(k) = x(k+1)$  im Turbolader hängt stark von der Einspritzmenge  $x(k)$  im vorangegangenen Zyklus und der Pedalstellung  $u(k)$  ab. Die weiteren Einflußgrößen wurden vernachlässigt und die Daten wurden durch einen linearen Filter leicht geglättet und zwischen 0 und 1 skaliert. Aufgrund des transparenten Ansatzes des NeuroFuzzy-Systems

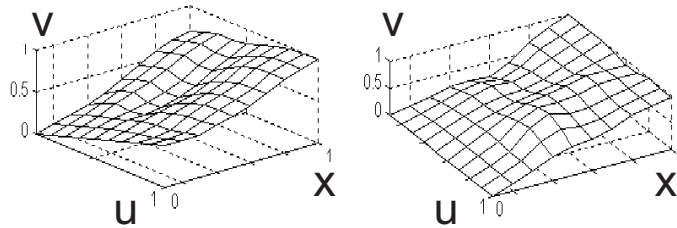


Abbildung 3: Übergangsfunktion  $f(x, u)$  vor (links) und nach (rechts) dem Lernprozeß.

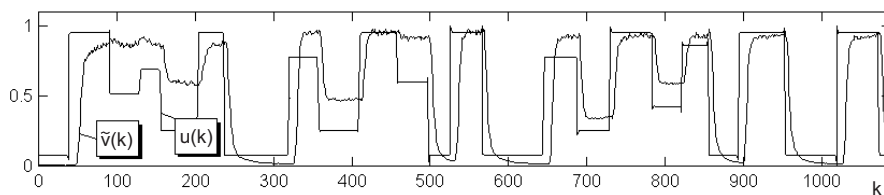


Abbildung 4: Normierte Einspritzmenge  $\tilde{v}$  und normierte Pedalstellung  $u$  über den Abtastschritten  $k$ .

war eine Anfangsinitialisierung der Parameter sehr leicht möglich. Abbildung 3 zeigt die Funktion  $f(x, u) = v$  vor und nach dem Lernvorgang. Abbildung 4 zeigt die behandelten Daten und Abbildung 5 die Zeitreihen vor und nach der Anwendung des Lernalgorithmus.

Das gute Ergebnis dieses Lernvorganges liegt darin begründet, dass eine sehr gute Anfangsinitialisierung möglich war. Um den Effekt der Anfangsinitialisierung testen zu können, wurde eine einfacheres Systemverhalten, ein  $PT_1$ -Verhalten zu erlernen und zwar von einer willkürlich initialisierten Regelbasis.

#### 4.4 $PT_1$ -Verhalten und der modifizierter Backpropagation-Algorithmus

Bei einer willkürlichen Anfangsbelegung der Regelbasis als Ausgangspunkt für das Erlernen einer  $PT_1$ -Verhaltens können prinzipiell konvergentes, oszillatorisches und sogar chaotisches Verhalten auftreten. Man kann die Beispielfunktionen 1, 3 und 4 aus dem ersten Abschnitt durch folgende Wahl der Singletonpositionen erreichen:

	$r_{1,1}$	$r_{1,2}$	$r_{1,3}$
Beispiel 1	0.25	0.5	0.75
Beispiel 3	1	0.5	0
Beispiel 4	0	1	0

Insbesondere können kleine Veränderungen in den Parametern  $r_{q,j}$ , wie in Abbildung 6 zu sehen ist, zu unterschiedlichem Verhalten führen. Die linke Funktion in dieser Abbildung hat drei Fixpunkte die rechte Funktion dagegen nur einen. Da die Fixpunkte zugleich Ruhelagen sind, hat das linke System zwei stabile und eine instabile Ruhelage, das rechte

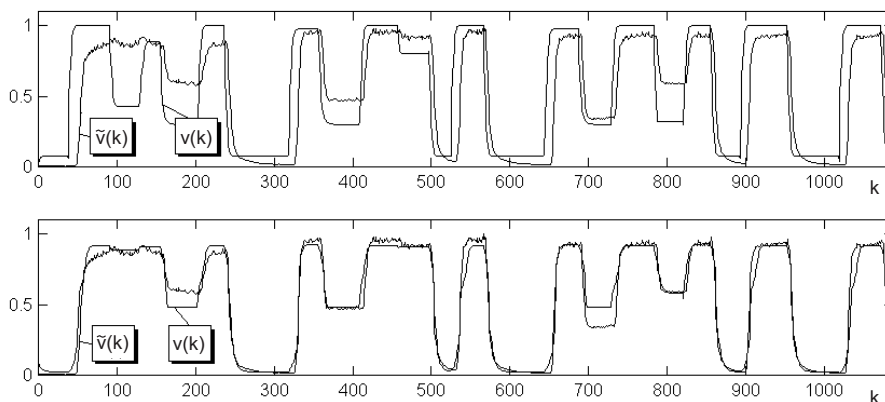


Abbildung 5: Vergleich der Daten  $\tilde{v}$  mit den Modellergebnissen  $v$  vor (oben) und nach (unten) dem Lernprozeß.

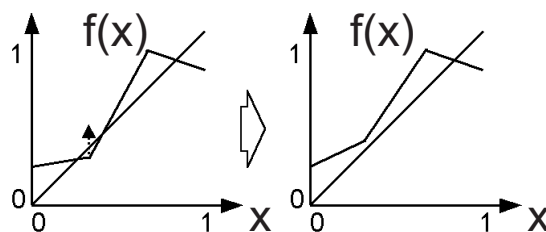


Abbildung 6: Kleine Veränderungen in den Parametern können zu einem sehr unterschiedlichen dynamischen Verhalten führen. Die linke Funktion erzeugt eine instabile und zwei stabile Ruhelagen, die rechte Funktion erzeugt nur eine einzige stabile Ruhelage.

System nur eine einzige, stabile Ruhelage. Für ein  $PT_1$ -Verhalten wird eine eindeutige, stabile Ruhelage erwartet. Weiterhin sollen negative Steigungen ausgeschlossen werden, wie in Abschnitt 3 gezeigt wurde. Um diese Punkte zu erfüllen wird der Raum der zulässigen Singletonpositionen eingeschränkt.

Dazu wird der Eingabewert  $u$  so gewählt, dass nur eine der Zugehörigkeitsfunktionen  $\mu_q^u(u)$  von Null verschieden ist. Diese Werte werden mit  $p_q^u$  bezeichnet. Damit ist die Funktion  $f$  in diesem Fall nur von den Singletonpositionen  $r_{q,j}$  abhängig, wobei  $q$  fest und  $j$  beliebig ist. Wählt man nun auch die  $x$ -Werte so, dass nur eine der Zugehörigkeitsfunktion  $\mu_j^x(x)$  von Null verschieden ist, so ist der Funktionswert gleich der speziellen Singletonposition  $r_{q,j}$ . Diese  $x$ -Werte werden mit  $p_j^x$  bezeichnet. Für je zwei benachbarte  $x$ -Werte  $p_j^x$  und  $p_{j+1}^x$ , werden die Singletonpositionen als Punkt in der Ebene dargestellt, wie in Abbildung 7 zu sehen ist. Wenn die Funktion die Winkelhalbierende schneidet, so darf sie nicht fallen. Diese entspricht dem verbotenen Bereich  $A$  in Abbildung 7. Ist die Steigung der Geraden beim Schnitt mit der Winkelhalbierenden größer als 1, gibt es einen



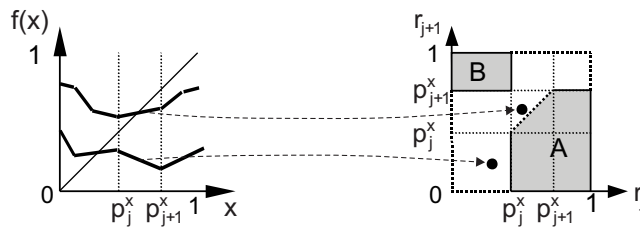


Abbildung 7: Nebeneinanderliegende Parameter  $p_q^x$  und  $p_{q+1}^x$  dürfen bei abgetasteten Systemen bestimmte Wertepaare nicht annehmen. Diese liegen in den grauen Bereichen A und B.

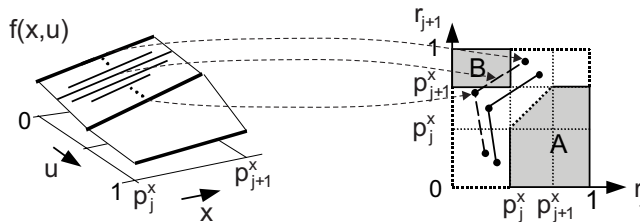


Abbildung 8: Die Verbindungslinie zwischen Parameterpaaren  $(r_{q,j}, r_{q+1,j})$  und  $(r_{q,j+1}, r_{q+1,j+1})$  dürfen nicht in die Bereiche A und B eindringen.

instabilen Fixpunkt und somit noch mindestens zwei stabile Fixpunkte im Intervall  $[0, 1]$ . Dieser verbotene Fall ist durch den Bereich B gekennzeichnet. Liegt die Eingabe  $u$  nun zwischen zwei Werten  $p_q^u$  und  $p_{q+1}^u$ , hat das den gleichen Effekt auf die Funktion  $f$ , wie die Nutzung von Singletonpositionen, die sich aus der Gewichtung der jeweiligen Singletonpositionen ergeben. Daher liegen die Funktionswerte für  $p_q^x$  und  $p_{q+1}^x$  in diesem Fall auf der Verbindungsgeraden zwischen  $(r_{q,j}, r_{q+1,j})$  und  $(r_{q,j+1}, r_{q+1,j+1})$  wie in Abbildung 8 zu sehen ist.

Vor jeder Berechnung der Zeitreihe und dem Vergleich mit den realen Daten werden nun die Singletonpositionen so verschoben, dass keine der Verbindungslinien in die verbotenen Bereich A oder B durchkreuzen. Da die Verschiebung minimal sein soll, muß somit ein Optimierungsproblem mit Randbedingungen gelöst werden.

Das Ergebnis dieser Modifikation ist in Abbildung 9 zu sehen. Man erkennt, dass der Fehler mit dem modifizierten Algorithmus schneller abfällt und das Ergebnis nach einer festen Zahl von Schritten besser ist. Simulationen bei Systemen, denen Zustand aufgrund einer hinzugefügten nichtumkehrbaren Ausgabefunktion nicht eindeutig festzustellen war, zeigten, dass die Erfolgsquote bei Verwendung des modifizierten Backpropagation-Algorithmus höher ist als bei dem einfachen.

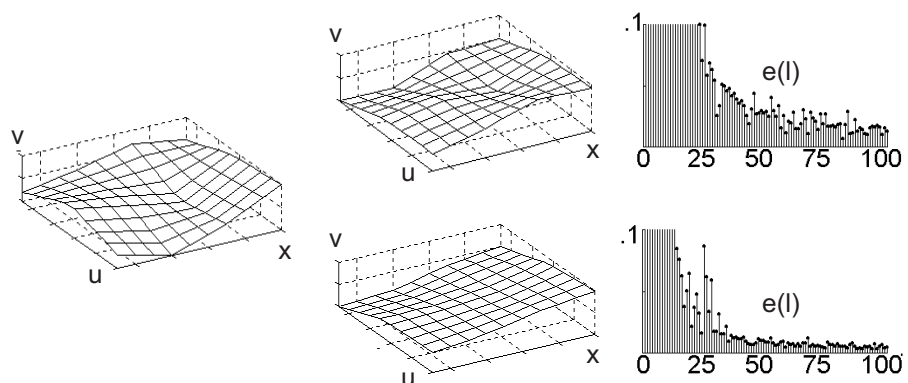


Abbildung 9: Die Übergangsfunktion einer  $PT_1$ -Gliedes vor (links) und nach (Mitte) dem Lernprozeß. Die Übergangsfunktion oben wurde mit dem unmodifizierten Bachpropagation-Algorithmus erreicht, die Übergangsfunktion unten mit dem modifizierten.

## 5 Zusammenfassung

Abgetastete wertekontinuierliche Systeme sind nur eine Teilmenge der zeitdiskreten Systeme. Sie zeichnen sich durch die Existenz beliebiger Funktionswurzeln aus. Diese Eigenschaft und daraus abgeleitete Eigenschaften können dazu genutzt werden, den Suchraum bei Identifikationsaufgaben einzuschränken, um somit mit einer höheren Erfolgsquote zu erzielen und schneller zu einer Identifikation zu kommen.

## Literatur

- [1] R. Kempf, On  $\Omega$ -limit sets of discrete-time dynamical systems, angenommen zu Veröffentlichung im Journal of Difference Equations and Applications.
- [2] M. Kuczma, B. Choczewski, R. Ger, Iterative Functional Equations, Cambridge Univ. Press, Cambridge, 1990
- [3] O. Nelles, S. Ernst, R. Isermann, Neuronale Netze zur Identifikation nichtlinearer, dynamischer Systeme: Ein Überblick, at 45 (1997) 6, S. 251-262
- [4] P. Protzel et al., Anwendung und Entwicklung Neuronaler Verfahren zur Autonomen Prozess-Steuerung, Abschlussbericht des BMBF Verbundprojekt 01 IN 505 B, 1999
- [5] Fuzzy logic and fuzzy control - Terms and definitions, VDI/VDE-Richtlinien, VDI/VDE 3550 Blatt 2, Beuth Verlag GmbH, Berlin, 2000
- [6] A. Zell, Simulation neuronaler Netze, Oldenburg Verlag, München, 1997