

# Eine Strategie zur Gewinnung Neuronaler Modelle für nichtlineare dynamische Systeme

Jörg Schultz, Karlsruhe

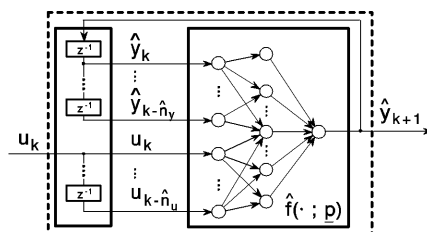
## 1 Kurzfassung

Zur Systemanalyse oder Reglersynthese ist in den meisten Fällen ein Modell der zu untersuchenden Strecke erforderlich. Dazu ist zunächst eine möglicherweise aufwendige theoretische Modellbildung notwendig. Es hat sich gezeigt, daß Künstliche Neuronale Netze dazu geeignet sind, nichtlineare dynamische Systeme ohne vorherige theoretische Modellbildung zu identifizieren. Die Schwierigkeit liegt jedoch zunächst darin, die für das Modell notwendigen Informationen über die anzusetzende Modellstruktur zu ermitteln. Im ersten Teil des Beitrags wird eine neue Methode vorgestellt, mit der die notwendigen Verzögerungen der Ein- und Ausgangsgröße, welche die Dynamik charakterisieren, über die Schätzung des Gradientenvektors der Systemnichtlinearität ermittelt werden können. Ist die Modellstruktur bestimmt, kann das nichtlineare dynamische Modell durch numerische Optimierung ermittelt werden. Aufgrund des Approximationsfehlers des Neuronalen Netzes und der verzögerten Rückkopplung der Ausgangsgröße stellt sich eine dynamische Fehlerfortpflanzung ein, die zu größeren Abweichungen zwischen System- und Modellverhalten führen kann. Daher wird im zweiten Teil ein iteratives Verfahren vorgeschlagen, bei dem weitere Trainingsdaten gezielt an den Stellen des Eingangsraumes hinzugefügt werden, an denen eine Modellverbesserung notwendig ist. Abschließend wird die Methode zur Modelloptimierung am Beispiel eines Rührkesselreaktors veranschaulicht.

## 2 Einführung

Klassische systemtheoretische Methoden stoßen bei komplexen Prozessen an ihre Grenzen. So ist beispielsweise die theoretische Modellbildung für derartige Prozesse mit großem Aufwand verbunden. Neuronale Netze [1, 2] bieten hier den Vorteil, daß man ohne theoretische Modellbildung, allein mittels Ein-/Ausgangsdaten des Prozesses, ein diskretes nichtlineares Black-Box-Modell erhält. Eine mögliche Netztopologie besteht darin, das Künstliche Neuronale Netz (KNN) mit einer vorgeschalteten Komponente zu versehen, die aus Verzögerungselementen besteht (Abb. 1).

Abbildung 1: Dynamisches Modell bestehend aus Künstlichem Neuronalem Netz zur Approximation der Nichtlinearität und einem vorgeschaltetem Block mit Zeitverzögerungen.



Im ersten Teil des Beitrags wird eine Methode zur Strukturermittlung vorgestellt, mit der die notwendigen Vergangenheitswerte der Ein- und Ausgangsgröße bestimmt werden können. Sie basiert auf der Linearisierung der Systemnichtlinearität und der Schätzung der Tangentialebene aufgrund der verfügbaren Meßdaten. Dabei wird über die Konditionierung von schrittweise aus den Prozeßdaten aufgebauten Matrizen zunächst der am weitesten zurückliegende notwendige Vergangenheitswert der Ausgangsgröße bestimmt. Das resultierende Modell der Tangentialebene ist linear in den Parametern. Die letzte gut konditionierte Matrix wird zur Berechnung des Gradienten benutzt, mit dem dann Rückschlüsse auf die weiteren, für das Modell relevanten Verzögerungen der Ausgangsgröße sowie der Eingangsgröße möglich sind.

Mit der damit bekannten Modellstruktur kann durch eine numerische Optimierung ein Neuronales Netz zur Approximation der Prozeßnichtlinearität trainiert werden. Wegen des verbleibenden Approximationsfehlers führt die verzögerte Rückkopplung der Modellausgangsgröße zu einer dynamischen Fehlerfortpflanzung. Zur Reduktion des dynamischen Modellfehlers ist eine alleinige Verkleinerung des Netzfehlers nicht ausreichend, auch die Interpolationseigenschaften des KNN müssen verbessert werden.

Die herkömmliche Vorgehensweise besteht darin, daß die Menge der für das Training benutzten Systemdaten ohne weitere Analyse vergrößert wird. Das im zweiten Teil vorgestellte iterative Verfahren führt dagegen eine Trainingsdatenerweiterung gezielt an den Stellen durch, die für die signifikanten Abweichungen zwischen der Modell- und Systemausgangsgröße verantwortlich sind. Damit gelingt es, iterativ eine Modellanpassung so lange vorzunehmen, bis das dynamische Modellverhalten den Anforderungen entspricht, ohne den Trainingsdatenumfang unnötig stark ansteigen zu lassen. Unter Verwendung eines Mehrschicht-Perzeptrons wird die iterative Methode zur Verbesserung des Modells eines Rührkesselreaktors veranschaulicht.

### 3 Identifikation des nichtlinearen dynamischen Prozesses

Ein nichtlinearer dynamischer SISO-Prozeß, beschrieben durch

$$y_{k+1} = f(y_k, \dots, y_{k-n_y}, u_k, \dots, u_{k-n_u}), \quad (1)$$

soll ohne vorherige theoretische Modellbildung identifiziert werden. Es wird vorausgesetzt, daß der Prozeß nicht sprungfähig und im betrachteten Arbeitsgebiet BIBO-stabil ist. Weiterhin wird angenommen, daß die Nichtlinearität  $f(\cdot)$  zeitinvariant, stetig differenzierbar und beschränkt ist. Die maximalen (unbekannten) Zeitverzögerungen, die den Prozeßausgang beeinflussen, werden für die zurückgekoppelte Ausgangsgröße mit  $n_y$  und für die Eingangsgröße mit  $n_u$  bezeichnet.

Das Modell (Abb. 1) wird gemäß der nachfolgenden Differenzgleichung

$$y_{k+1} = \hat{f}(y_k, \dots, y_{k-\hat{n}_y}, u_k, \dots, u_{k-\hat{n}_u}; \underline{p}) \quad (2)$$

mit den maximalen Modellverzögerungen  $\hat{n}_y, \hat{n}_u$  eingeführt. Der Vektor  $\underline{p}$  besteht aus den Modellparametern, die z.B. für das Mehrschicht-Perzeptron die Gewichte und Schwellwerte darstellen. Die Parameter  $\underline{p}$  sind so zu bestimmen, daß (2) das Ein-/Ausgangsverhalten von (1) möglichst gut nachbildet. Der herkömmliche Ansatz besteht in der Minimierung [4] des Gütekriteriums

$$J = \frac{1}{2} \sum_k (y_{k+1} - \hat{y}_{k+1})^2 = \frac{1}{2} \sum_k \left( y_{k+1} - \hat{f}(y_k, \dots, y_{k-\hat{n}_y}, u_k, \dots, u_{k-\hat{n}_u}; \underline{p}) \right)^2 \quad (3)$$

hinsichtlich des Vektors  $\underline{p}$  unter Verwendung einer endlichen Anzahl von Prozeßdaten.

Ohne theoretische Modellbildung sind die anzusetzenden maximalen Verzögerungen  $\hat{n}_y, \hat{n}_u$  unbekannt und müssen deshalb hinreichend groß gewählt werden. Der Rechenaufwand in den nachfolgenden Schritten des Identifikationsvorgangs, z.B. bei der Bestimmung der Modellparameter, sollte möglichst gering sein. Daher ist es vorteilhaft, nur die relevanten Verzögerungen, die zur Beschreibung des dynamischen Prozesses erforderlich sind, ins Modell aufzunehmen. Eine Methode zur Ermittlung dieser Verzögerungen wird im nächsten Abschnitt vorgestellt.

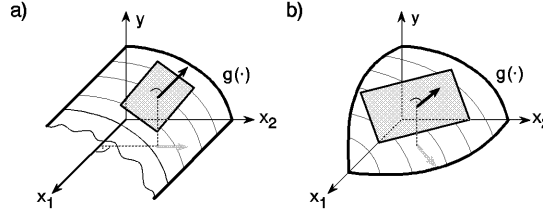
#### 4 Ermittlung der Modellstruktur durch Linearisierung

Eine nichtlineare Funktion  $g(x_1, x_2, \dots, x_n) = g(\underline{x})$  ist von einem Eingangsterm  $x_i$  im betrachteten Bereich unabhängig, wenn die partielle Ableitung von  $g(\cdot)$  nach der Variablen  $x_i$  in diesem gesamten Bereich verschwindet:

$$\frac{\partial g(\cdot)}{\partial x_i} = 0. \quad (4)$$

Dies ist gleichbedeutend mit der Aussage, daß der Gradientenvektor senkrecht auf der Achse derjenigen Variablen steht, von der die Funktion unabhängig ist (Abb. 2).

Abbildung 2: Die Funktion  $y = g(x_1, x_2)$  ist im Fall a) von  $x_1$  unabhängig; der Gradientenvektor steht im gesamten Definitionsbereich senkrecht auf der  $x_1$ -Achse. Im Fall b) gilt dies nicht; jeder Eingangsterm wird benötigt.



Es ist  $\hat{f}(\cdot; \underline{p})$  hinsichtlich unnötiger Eingangsterme zu untersuchen. Der Eingangsraum wird durch die Verzögerungen der Ein- und Ausgangsgröße aufgespannt. Diese werden im folgenden als Eingangsterme bezeichnet und im Vektor

$$\underline{x} = \left[ y_k, \dots, y_{k-\hat{n}_y}, u_k, \dots, u_{k-\hat{n}_u} \right]^T \quad (5)$$

zusammengefaßt. Ein Eingangsterm in (5) ist nach (4) unnötig, wenn das zu dem Eingangsterm gehörende Element des Gradientenvektors  $\frac{\partial}{\partial x_i} \hat{f}(\cdot; \underline{p})$  im gesamten Arbeitsbereich Null ist. Es wäre nun denkbar, ein KNN mit den Eingangstermen (5) zu trainieren und dieses Modell zu benutzen, um den Gradienten  $\frac{\partial}{\partial x_i} \hat{f}(\cdot; \underline{p})$  zu berechnen. Um aber das aufwendige Ermitteln von  $\hat{f}(\cdot; \underline{p})$  mit der zu großen Anzahl an Eingangstermen zur Berechnung des Gradientenvektors zu vermeiden, kann dieser mit Hilfe der gemessenen Ein-/Ausgangsdaten des Prozesses geschätzt werden.

Die Taylorreihenentwicklung von  $\hat{f}(\cdot; \underline{p})$  am Punkt  $\underline{x}_p$  lautet

$$y_{k+1} = \hat{f}(\underline{x}_p) + \Delta \underline{x}^T \left. \frac{\partial \hat{f}}{\partial \underline{x}} \right|_{\underline{x}_p} + O(\|\Delta \underline{x}\|^2) \quad (6)$$

mit  $\Delta \underline{x} = \underline{x} - \underline{x}_p$ . Eine Näherung zur Bestimmung der Abweichung des Ausgangs  $y_{k+1}$  von  $\hat{f}(\underline{x}_p)$  stellt die Tangentialebenenbeschreibung

$$\Delta y_{k+1} = y_{k+1} - \hat{f}(\underline{x}_p) = \Delta \underline{x}^T \left. \frac{\partial \hat{f}}{\partial \underline{x}} \right|_{\underline{x}_p} = \Delta \underline{x}^T \underline{m} \quad (7)$$

dar, die man unter Vernachlässigung des Restterms  $O(\|\Delta \underline{x}\|^2)$  erhält. Dabei bezeichnet  $\underline{m}$  den gesuchten Gradientenvektor, dessen Schätzung  $\hat{\underline{m}}$  mit der Least-Squares-Methode durchgeführt werden kann. Mit  $N$  Meßdaten aus der Umgebung von  $\underline{x}_p$  erhält man unter Berücksichtigung von (7) die Meßgleichung

$$\begin{bmatrix} \Delta y_{k+1}^1 \\ \vdots \\ \Delta y_{k+1}^N \end{bmatrix} = \begin{bmatrix} \Delta \underline{x}_1^T \\ \vdots \\ \Delta \underline{x}_N^T \end{bmatrix} \hat{\underline{m}} + \hat{\underline{\epsilon}} = \begin{bmatrix} \Delta y_k^1 & \cdots & \Delta y_{k-\hat{n}_y}^1 & \Delta u_k^1 & \cdots & \Delta u_{k-\hat{n}_u}^1 \\ \vdots & & \vdots & \vdots & & \vdots \\ \Delta y_k^N & \cdots & \Delta y_{k-\hat{n}_y}^N & \Delta u_k^N & \cdots & \Delta u_{k-\hat{n}_u}^N \end{bmatrix} \hat{\underline{m}} + \hat{\underline{\epsilon}} = \underline{A} \hat{\underline{m}} + \hat{\underline{\epsilon}}. \quad (8)$$

In (8) bezeichnet  $\hat{\underline{\epsilon}}$  den Linearisierungsfehler. Für eine erfolgreiche Schätzung ist es notwendig, daß  $\underline{A}$  gut konditioniert ist. Im folgenden wird gezeigt, daß die gute Konditionierung nicht nur von den benutzten Daten abhängt, sondern auch von den im Modell einbezogenen Verzögerungstermen. In diesem Zusammenhang wird herausgearbeitet, wie dabei die maximale notwendige Verzögerung  $n_y$  ermittelt werden kann.

Es wird vorausgesetzt, daß die angenommenen maximalen Verzögerungen  $\hat{n}_y, \hat{n}_u$  größer als die tatsächlichen im System vorhandenen Verzögerungen  $n_y, n_u$  sind und somit  $\hat{n}_y > n_y$  und  $\hat{n}_u > n_u$  gilt. Ohne Einschränkung der Allgemeinheit wird zur Erleichterung der folgenden Ausführungen darüberhinaus angenommen, daß  $\Delta_m = \hat{n}_y - n_y < \hat{n}_u - n_u$  ist.

Wenn  $f(\cdot)$  bekannt wäre, dann ergäbe sich die Linearisierung von (1) zu

$$\begin{aligned} \Delta y_{k+1} &= m_1 \Delta y_k + \cdots + m_{n_y+1} \Delta y_{k-n_y} + \\ &+ m_{n_y+2} \Delta u_k + \cdots + m_{n_y+n_u+2} \Delta u_{k-n_u} + \varepsilon \end{aligned} \quad (9)$$

mit einem Linearisierungsfehler  $\varepsilon$ . In den folgenden Betrachtungen werden die Linearisierungen von (1) bei vorhergehender Verschiebung des Index  $k$  um  $r$  benötigt. Sie würden sich mit zu

$$\begin{aligned} \Delta y_{k+1-r} &= m_1^r \Delta y_{k-r} + \cdots + m_{n_y+1}^r \Delta y_{k-n_y-r} + \\ &+ m_{n_y+2}^r \Delta u_{k-r} + \cdots + m_{n_y+n_u+2}^r \Delta u_{k-n_u-r} + \varepsilon^r \end{aligned} \quad (10)$$

berechnen, wobei der Index  $k$  vorher um  $0 \leq r \leq \Delta_m$  verschoben wurde. In (10) bezeichnet  $\varepsilon^r$  den Linearisierungsfehler. Es lassen sich Vektoren  $\Delta \underline{y}_{k-\nu} = [\Delta y_{k-\nu}^1, \dots, \Delta y_{k-\nu}^N]^T$ ,  $r-1 \leq \nu \leq n_y+r$  und  $\Delta \underline{u}_{k-\mu} = [\Delta u_{k-\mu}^1, \dots, \Delta u_{k-\mu}^N]^T$ ,  $r \leq \mu \leq n_u+r$  mit Prozeßmeßdaten aufbauen. Diese Spaltenvektoren genügen (10).

Für  $\varepsilon^r = 0$  folgt mit (10), daß die Vektoren  $\Delta \underline{y}_{k-j} = [\Delta y_{k-j}^1, \dots, \Delta y_{k-j}^N]^T$ ,  $0 \leq j \leq \Delta_m - 1$  in (8) linear abhängen von  $\Delta \underline{y}_{k-i}$ ,  $j+1 \leq i \leq j+n_y+1$  und von  $\Delta \underline{u}_{k-i} = [\Delta u_{k-i}^1, \dots, \Delta u_{k-i}^N]^T$ ,  $j+1 \leq i \leq j+n_u+1$ . Für den linearen Fall, d.h.  $\varepsilon^r = 0$  wurde in [3] gezeigt, daß sich ein Rangabfall in der Matrix  $\underline{A}$  in (8) für den Fall  $\hat{n}_y > n_y$  ergibt.

Da aber im nichtlinearen der Linearisierungsfehler  $\varepsilon^r \neq 0$  vorliegt, sind die Spaltenvektoren von  $\underline{A}$  in (8) nicht exakt linear abhängig. Für den Fall, daß der Linearisierungsfehler sehr klein ist, d.h. daß (10) nur in der nächsten Umgebung des Linearisierungspunktes betrachtet wird, kann jedoch aus einer schlechten Konditionierung von  $\underline{A}$  auf einen "Rangabfall" geschlossen werden. In diesem Fall ist  $\underline{A}$  nämlich schlecht konditioniert. Dies wird ausgenutzt, um die maximale Ausgangsverzögerung für das Modell (2) zu ermitteln. Dazu werden schrittweise die Matrizen

$$\underline{A}_\kappa = \left[ \Delta \underline{y}_{k-\kappa}, \dots, \Delta \underline{y}_{k-\hat{n}_y}, \Delta \underline{u}_{k-\kappa}, \dots, \Delta \underline{u}_{k-\hat{n}_u} \right] \quad (11)$$

für  $\kappa \leq \hat{n}_y$  aufgebaut. Für  $\hat{n}_y - n_y \leq \kappa \leq \hat{n}_y$  ist die Matrix  $\underline{A}_\kappa$  gut konditioniert, da die Spaltenvektoren von  $\underline{A}_\kappa$  linear unabhängig sind. Wird der Spaltenvektor  $\Delta \underline{y}_{k+1-\hat{n}_y+n_y}$  für  $\kappa = \hat{n}_y - n_y - 1$  hinzugefügt, so folgt mit (10), daß  $\underline{A}_{\hat{n}_y-n_y-1}$  schlecht konditioniert ist. Dieser Wechsel von einer gut zu einer schlecht konditionierten Matrix  $\underline{A}_\kappa$  zeigt an, daß eine Verzögerung mehr als notwendig betrachtet worden ist. Wird der Index der letzten noch gut konditionierten Matrix  $\underline{A}_\kappa$  mit  $\kappa^*$  bezeichnet, so ergibt sich die unbekannte maximale Ausgangsverzögerung  $\hat{n}_y$  des Modells wegen  $\kappa^* = \hat{n}_y - n_y$  zu

$$n_y = \hat{n}_y - \kappa^*. \quad (12)$$

Mit der letzten gut konditionierten Matrix  $\underline{A}_{\hat{n}_y-n_y} = \underline{A}_{\kappa^*}$  kann schließlich der Gradientenvektor  $\hat{\underline{m}}$  geschätzt werden. Ist ein Element des Gradientenvektors  $\hat{\underline{m}}$  nahezu null, so ist im untersuchten Punkt  $\underline{x}_p$  die Ausgangsgröße  $y_{k+1}$  von der dem Element zugehörigen Verzögerung unabhängig.

Um von den absoluten Werten der Konditionierung  $cond\{\underline{A}_\kappa\}$  unabhängig zu werden, ist es angebracht, jeweils das Verhältnis  $cond\{\underline{A}_{\kappa-1}\}/cond\{\underline{A}_\kappa\}$  zweier aufeinanderfolgend aufgebauten Matrizen heranzuziehen. Untersuchungen zur Ermittlung der Modellordnung von linearen Systemen, sind in dieser Form in [6] bereits 1971 durchgeführt worden.

Das gesamte Verfahren zur Bestimmung der Modellordnung läuft wie folgt ab:

- 1 ) Ansetzen einer hinreichend hohen Anzahl an Verzögerungen  $\hat{n}_y, \hat{n}_u$ .
- 2 ) Schrittweises Berechnen der Konditionen  $c_\kappa = cond\{\underline{A}_\kappa\}$  für  $\kappa \leq \hat{n}_y$ , bis der Übergang von einer gut zu einer schlecht konditionierten Matrix detektiert wird. Die maximale Verzögerung der Ausgangsgröße wird mit (12) berechnet.
- 3 ) Schätzen des Gradientenvektors  $\hat{\underline{m}}$  für das Modell mit der letzten gut konditionierten Matrix  $\underline{A}_{\kappa^*}$ .
- 4 ) Feststellen der Elemente des Gradientenvektors, die dem Betrage nach nahe Null sind.

Die Ergebnisse einer solchen Untersuchung gelten lediglich in der Umgebung des betrachteten Punktes  $\underline{x}_p$ . Um zu einer Aussage über die für das Systemverhalten relevanten Verzögerungen zu kommen, sind Ergebnisse für eine ausreichend große Anzahl verschiedener Punkte  $\underline{x}_{p,i}$  des

Arbeitsbereichs heranzuziehen. Vernachlässigt werden können schließlich nur die Verzögerungen, die sich für alle untersuchten Punkte als nicht relevant erwiesen haben.

Die vorgestellte Methode benötigt so viele Prozeßdaten, daß in der Umgebung mehrerer Punkte  $\underline{x}_p$  sowohl eine gute Aussage hinsichtlich der Konditionierung als auch der Berechnung des Gradienten möglich ist. Die Daten müssen so verteilt sein, daß  $\underline{A}_{\kappa^*}$  gut konditioniert ist. Da das System BIBO-stabil ist, können beliebige Eingangssequenzen auf den Prozeß geschaltet werden. Somit stellen diese Forderungen keine grundsätzliche Einschränkung dar.

**Beispiel 1: Ermittlung der Modellstruktur eines Rührkesselreaktors** Ein Rührkesselreaktor soll ohne theoretische Modellbildung identifiziert werden. Das nachfolgend angegebene kontinuierliche Modell [5] des Prozesses wird nur zu Verifikationszwecken herangezogen. Es lautet:

$$\dot{\underline{x}} = \begin{bmatrix} -0.957 & a_{12}(\underline{x}) \\ -0.323 & a_{22}(\underline{x}) \end{bmatrix} \underline{x} + \begin{bmatrix} 0 \\ 1.548 \end{bmatrix} u$$

mit

$$a_{12}(\underline{x}) = \begin{cases} b_1(\underline{x}) \left( e^{b(x_2)} - 6.568 \cdot 10^{-15} \right) & : x_2 \neq 0 \\ 21.449 (0.279 - x_1) & : x_2 = 0 \end{cases} \quad b_1(\underline{x}) = \frac{1.05 \cdot 10^{14} (0.279 - x_1)}{x_2}$$

$$a_{22}(\underline{x}) = 0.468 a_{12}(\underline{x}) - 1.815 \quad b(x_2) = \frac{-34.289}{1.05 + x_2}.$$

Die Kühlmitteltemperatur ist die Eingangsgröße, die Reaktortemperatur und die Konzentration des zuffließenden Mediums sind die Zustandsgrößen. Es soll ein diskretes Ein-/Ausgangsmodell mit der Kühlmitteltemperatur als Eingangsgröße und der Reaktortemperatur als Ausgangsgröße ermittelt werden. Durch Euler-Approximation und der Betrachtung um die Ruhelage  $[x_1, x_2]_R = [0.72, 0.05]$  ergibt sich als Differenzgleichung  $y_{k+1} = f(y_k, y_{k-1}, u_k, u_{k-1})$ , wobei  $n_y = 1$  und  $n_u = 1$  gilt.

Für den Modellansatz wird  $y_{k+1} = \hat{f}(y_k, y_{k-1}, y_{k-2}, u_k, u_{k-1}, u_{k-2}; \underline{p})$  ( $\hat{n}_y = 2, \hat{n}_u = 2$ ) gewählt. Betrachtet man die Punkte  $\underline{x}_{p1}^T = [5.3, 3.3, 2.1, 3.5, 3.4, -1.5] \cdot 10^{-3}$  und  $\underline{x}_{p2}^T = [0, 0, 0, 0, 0, 0]$  aus dem Arbeitsraum, so berechnet sich die Folge der Matrixkonditionen  $c_\kappa = \text{cond}\{\underline{A}_\kappa\}$  zu  $\{c_2, \dots, c_{-1}\} = \{1.4, 3.8, 137.5, 1310\}$  für  $\underline{x}_{p1}$  und zu  $\{c_2, \dots, c_{-1}\} = \{1.6, 3.5, 124.5, 953\}$  für  $\underline{x}_{p2}$ . Der Übergang von einer gut zu einer schlecht konditionierten Matrix  $\underline{A}_\kappa$  tritt für beide Punkte von  $c_1$  zu  $c_0$  auf. Folglich repräsentiert  $\underline{A}_1$  die letzte gut konditionierte Matrix. Mit (12) folgt  $n_y = \hat{n}_y - \kappa^* = 1$ . Somit ist für den Modellansatz  $y_{k+1} = \hat{f}(y_k, y_{k-1}, u_k, u_{k-1}; \underline{p})$  zu wählen. Die gut konditionierte Matrix  $\underline{A}_1$  wird benutzt, um  $\hat{m}$  für  $\underline{x}_{p1}, \underline{x}_{p2}$  zu schätzen. Diese sind  $\underline{m}_1^T = [2.2, -1.6, 0.16, -0.23]$  und  $\underline{m}_2^T = [2.6, -1.7, 0.2, -0.21]$ . Somit kann keine der Verzögerungen weiter entfernt werden. Damit weist das Modell die gleiche Struktur wie die Systemdifferenzgleichung auf.

## 5 Modellverbesserung durch Trainingsdatenerweiterung

Nach Bestimmung der Modellstruktur mit der oben beschriebenen Vorgehensweise, besteht der weitere Ablauf zur Identifikation des Prozesses (Abb. 3) aus der Minimierung des Gütekriteriums (3) mit geeignet gewählten Trainingsdaten.

Sie sind so zu wählen, daß der Teil des Eingangsraums von (2), der den betrachteten Arbeitsbereich darstellt, gesamt und gut verteilt abgedeckt ist. Zum anderen sollte aber der Datenumfang wegen des erforderlichen Rechenaufwands beim Training nicht zu groß werden.

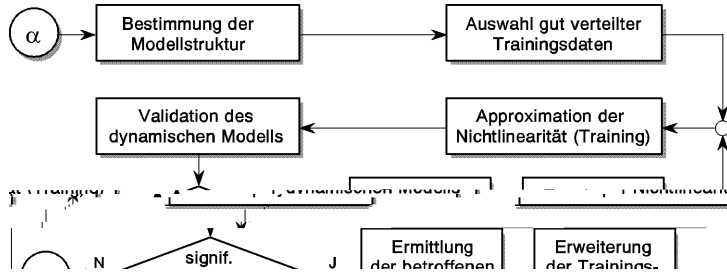


Abbildung 3: Ablaufdiagramm zur Identifikation nichtlinearer dynamischer Systeme und iterativen Modellverbesserung.

Eine Möglichkeit [7] besteht in der äquidistanten Einteilung der Achsen des Eingangsraums. Durch Füllen jedes so entstandenen Hyperkubus mit einer begrenzten Anzahl von Prozeßdaten werden die obigen Forderungen erfüllt.

Nach dem Training des Neuronalen Netzes  $\hat{f}(\cdot; \underline{p})$  liegt ein Parametervektor  $\underline{p}$  vor. Anschließend wird eine dynamische Validation des neuronalen Modells durchgeführt. Aufgrund des Approximationsfehlers des Netzes und der dynamischen Rückkopplung der Modellausgangsgröße stellt sich eine dynamische Fehlerfortpflanzung

$$e_{k+1} = y_{k+1} - \hat{y}_{k+1} = f(y_k, \dots, y_{k-n_y}, u_k, \dots, u_{k-n_u}) - \hat{f}(\hat{y}_k, \dots, \hat{y}_{k-n_y}, u_k, \dots, u_{k-n_u}; \underline{p}) \quad (13)$$

ein. Steigt der Fehler  $e_k$  über eine vorgegebene Schranke, so ist das Modell zu verbessern.

Der Fehler  $e_k$  resultiert aus dem Approximations- und dem Interpolationsfehler des Netzes  $\hat{f}(\cdot; \underline{p})$ . Der Approximationsfehler kann zwar durch Training weiter reduziert, aber nie zu Null gemacht werden können. Die Verbesserung der Netzinterpolation besteht herkömmlicherweise [8] in der globalen Erweiterung der Trainingsdatenmenge und des erneuten Trainings. Bei einer solchen Vorgehensweise kann der Fall auftreten, daß Bereiche unberücksichtigt bleiben, in denen eine Modellverbesserung zwingend wäre. Dann wird es bei einem mit dem erweiterten Datensatz trainierten Netz zu keiner wesentlichen dynamischen Modellverbesserung kommen. Zum anderen können Daten in Bereichen hinzugefügt werden, in denen keine Modellverbesserung notwendig wäre. Dies führt zu einer unnötigen Vergrößerung der Trainingsdatenmenge. Um die geschilderten Nachteile zu überwinden, ist eine gezielte Analyse der Ursache des fehlerhaften Modellverhaltens erforderlich.

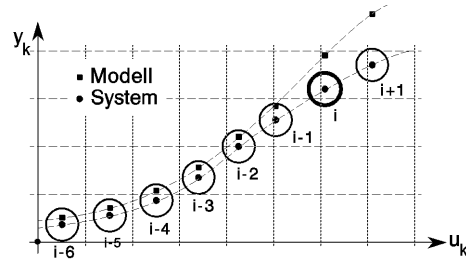
Dies kann über die Betrachtung des Eingangsraumes der Nichtlinearität geschehen, der durch die Verzögerungsterme aufgespannt wird. Dazu wird der Zustandsvektor  $\underline{v}_k$  des Prozesses  $f(\cdot)$  zum k-ten Abtastzeitpunkt

$$\underline{v}_k = [y_k, \dots, y_{k-n_y}, u_k, \dots, u_{k-n_u}]^T \quad (14)$$

eingeführt. Analog geschieht das für den Modellzustand:  $\hat{\underline{v}}_k = [\hat{y}_k, \dots, \hat{y}_{k-n_y}, u_k, \dots, u_{k-n_u}]^T$ . Sie beschreiben den zum k-ten Abtastzeitpunkt vorliegenden Zustand des Systems bzw. des

Modells in dem durch die Eingangsvariablen aufgespannten Eingangsraum (Abb. 4).

Abbildung 4: Der Eingangsraum für einen Prozeß  $y_{k+1} = f(y_k, u_k)$  ist wiedergegeben. Es sind die System- und die Modellzustände  $\underline{v}_k$  und  $\hat{\underline{v}}_k$  für  $i - 6 \leq k \leq i + 1$  dargestellt. Das Kreisgebiet kennzeichnet die definierte Umgebung eines Systemzustandes, in der der zugehörige Modellzustand liegen darf.



Aufgrund der dynamischen Fehlerfortpflanzung wird sich  $\Delta v_k = \|\underline{v}_k - \hat{\underline{v}}_k\| \neq 0$  als Differenz zwischen System- und Modellzustand einstellen. Dabei bezeichnet  $\|\cdot\|$  die Euklidische Vektornorm. Ein Maß für das dynamische Modellverhalten stellt die Differenz  $\Delta v_k$  dar. Überschreitet sie eine vorgegebene Schranke  $v_{max}$ , so liegt ein signifikanter Fehler vor. Die Ursache für das fehlerhafte Modellverhalten zum Zeitpunkt  $k = i$  ist das schlechte Interpolationsverhalten des KNN in den Umgebungen der Systemzustände für  $k < i$ . Genau in diesen Bereichen muß die Interpolation des KNN verbessert werden, was durch Erweiterung der Trainingsdatenmenge mit Daten aus diesen Bereichen geschieht. Da nicht genau festgelegt werden kann, zu welchem Zeitpunkt die Ursache der signifikanten dynamischen Modellabweichung vorliegt, wird man sich auf eine endliche Anzahl  $k_{max}$  von vorangehenden Bereichen beschränken. Sie sind die Umgebung der Systemzustände  $\underline{v}_k$  für  $i - k_{max} \leq k < i$ . Die Trainingsdaten werden mit Daten erweitert, die in diesen – wegen der Euklidischen Norm – hyperkugelförmigen Bereichen liegen.

Mit der oben beschriebenen Vorgehensweise wird die Trainingsdatenmenge an den Stellen des Eingangsraumes erweitert, die als Ursache fehlerhaften Modellverhaltens erkannt werden und damit gezielt eine bessere Interpolation erreicht.

Die Verbesserung des Modells durch Hinzunahme weiterer geeigneter Prozeßdaten und erneutem Training wird iterativ solange durchgeführt, bis keine signifikanten Abweichungen zwischen Modell- und Systemzuständen mehr auftreten (Abb. 3). Der Ablauf der Methode wird im folgenden Beispiel verdeutlicht.

**Beispiel 2: Verbesserung des Rührkesselreaktormodells** Für die Nichtlinearität  $\hat{f}(\cdot; \underline{p})$  des Rührkesselreaktormodells  $\hat{y}_{k+1} = \hat{f}(\hat{y}_k, \hat{y}_{k-1}, u_k, u_{k-1}; \underline{p})$  wurde ein Mehrschicht-Perzeptron mit zwei verdeckten Schichten angesetzt. In jeder Schicht befinden sich 6 Neuronen. Die Aktivierungsfunktion wurde für alle Neuronen zur Sigmoidfunktion gewählt.

Der initial aufgebaute Datensatz bestand aus 792 gut im Eingangsraum verteilten Prozeßdaten. Weiterhin lag eine Menge von 4000 Daten vor, die zur Erweiterung dienen konnte. Diese Daten wurden mittels der Differenzgleichung des Rührkesselreaktors aus Beispiel 1 in der Simulation erzeugt. Das Training des KNN mit dem Levenberg-Marquard-Algorithmus führte nach 600 Epochen zu einem bezogenen quadratischen Fehler

$$SSE = \frac{1}{N} \sum_{k=1}^N (y_k - \hat{y}_k)^2 = 6.78 \cdot 10^{-7}. \quad (15)$$



Die Verläufe der Ausgangsgröße des Systems und des Modells und der Differenz  $\|\Delta \underline{v}_k\|$  sind in den nachfolgenden Abbildungen wiedergegeben. Die Grenze für den signifikanten Fehler ist zu  $v_{max} = 0.021$  gewählt. Wie aus Abb. 5a zu entnehmen ist, liegen zu den Zeitpunkten  $\{i_0\} = \{26, 116, 228, 272, 290\}$  signifikante Abweichungen zwischen dem Modell- und Prozeßverhalten vor. Es werden Bereiche zur Erweiterung mit Daten betrachtet, die bis zu  $k_{max} = 5$  Abtastschritten vor der fehlerhaften Abtastzeit liegen. Die maximale Anzahl an erweiterungsmöglichen Daten für jeden Bereich wurde auf 10 gesetzt. Es konnten 231 Datentupel  $\{\underline{x}_k, y_{k+1}\}$  in dem Datenpool der 4000 Prozeßdaten zur Erweiterung der Trainingsdatenmenge gefunden werden.

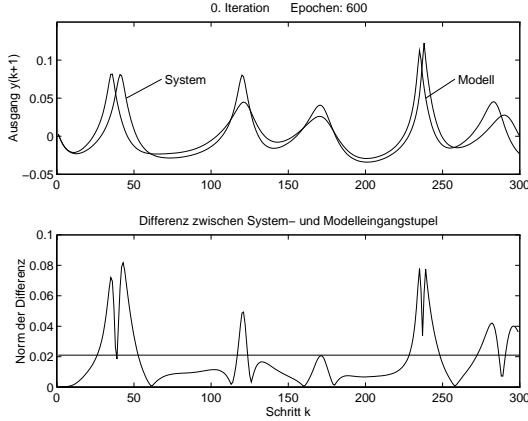


Abbildung 5a: Verläufe der Ausgangsgröße von System und Modell nach dem Training mit der initialen Trainingsmenge mit 792 Daten (oben) und der zugehörigen Norm  $\|\underline{v}_k - \hat{\underline{v}}_k\|$  (unten).

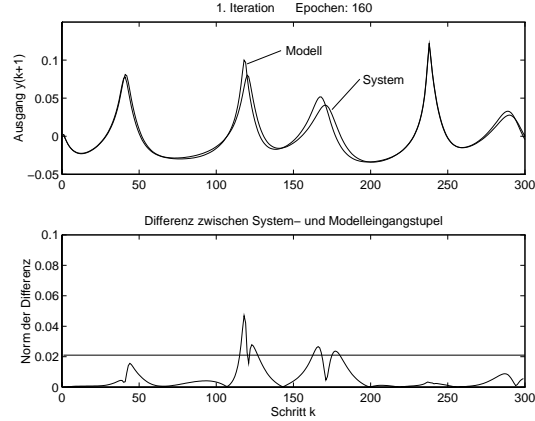


Abbildung 5b: Verläufe der Ausgangsgröße von System und Modell nach dem Training mit dem einmal erweiterten Trainingssatz mit 1023 Daten (oben) und der zugehörigen Norm  $\|\underline{v}_k - \hat{\underline{v}}_k\|$  (unten).

Das Netz wurde 160 Epochen mit 1023 Daten weitertrainiert, bis zu einem Fehler  $SSE = 6.18 \cdot 10^{-7}$ . Der dynamische Verlauf des Modellausgangsgröße (Abb. 5b) zeigt, daß bei  $\{i_0\} = \{26, 228, 272, 290\}$  keine signifikanten Differenzen mehr auftreten. Dafür liegt ein Fehler bei  $i = 115$  vor. Hier hat sich das Interpolationsverhalten des Netzes nicht genügend verbessert. Daneben sind kleinere signifikante Fehler bei  $\{i\} = \{162, 175\}$  auftreten. In der nächsten Iteration wurde  $k_{max} = 10$  gewählt und maximal 10 Datentupel zur Erweiterung eines Bereiches erlaubt. Es lagen somit 3 signifikante Fehler vor, für die 10 vorherliegende Bereiche zu erweitern waren. Es fanden sich dazu 232 neue Datentupel, die zur Trainingsmenge hinzugefügt wurden. Das Netz wurde weitere 400 Epochen bis zu einem Fehler von  $SSE = 6.49 \cdot 10^{-7}$  trainiert. Der sich ergebende Verlauf des Modells ist in Abb. 6 dargestellt. Es treten keine signifikanten Fehler mehr auf.

## 6 Zusammenfassung

Zur Vermeidung einer unter Umständen aufwendigen theoretischen Modellbildung kann man Künstliche Neuronale Netze mit Rückkopplung der Ausgangsgröße und externen Verzögerungen als Black-Box-Identifikationsmodelle verwenden. Die in diesem Artikel benutzte Aufspaltung in eine dynamische Komponente und in die zu approximierende Nichtlinearität erfordert zwei Schritte bei der Identifikation. Zunächst ist die Modellstruktur zu bestimmen. Dazu wurde ein Verfahren vorgestellt, mit dem die notwendigen Verzögerungsterme des Modells über

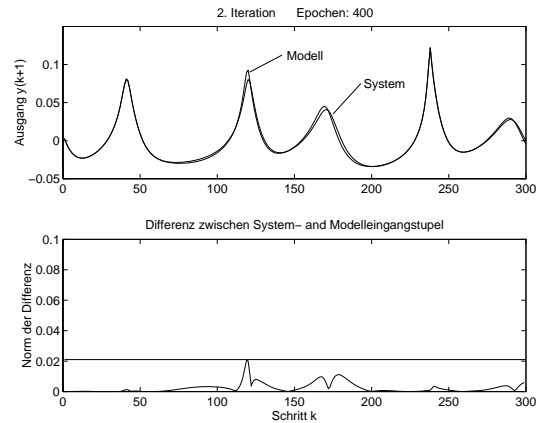


Abbildung 6: Verläufe der Ausgangsgröße von System und Modell nach dem zweiten Training mit 1255 Daten (oben) und der zugehörigen Norm  $\|\underline{v}_k - \hat{\underline{v}}_k\|$  (unten). Es treten keine signifikanten Fehler mehr auf.

die Konditionierung aus Prozeßdaten aufgebauter Matrizen und den damit geschätzten Gradientenvektoren ermittelt werden können. Es wird bei dieser Vorgehensweise die Erzeugung eines überdimensionierten neuronalen Modells vermieden.

Nach der Bestimmung der Modellverzögerungen und dem erfolgten Netztraining wird die Modellvalidation vorgenommen. Dazu wurde ein iteratives Verfahren zur Verbesserung des dynamischen Modellverhaltens vorgestellt. Es basiert auf der Analyse des Eingangsraumes des KNN und der Erweiterung der Trainingsdatenmenge. Durch die gezielte Verbesserung der Interpolation des Netzes an einzelnen Stellen, wird der Zuwachs des Datenumfangs begrenzt gehalten.

Durch die vorgestellten Verfahren und der betrachteten Modellstruktur ist eine größere Transparenz bei der Identifikation mit KNN gewonnen. Wenn die Modellstruktur basierend auf den Prozeßdaten ermittelt ist, kann die Ursache für fehlerhaftes Verhalten nur in der Approximation und der Interpolation der Nichtlinearität liegen. Die Anwendung der vorgestellten iterativen Methode erlaubt eine zielgerichtete Verbesserung des Modells.

## Literatur

- [1] Hunt, K. J., Sbarboro, D., et.al.: Neural Networks for Control Systems - A Survey, Automatica, 28(6), pp. 1083-1112, 1992.
- [2] Narendra, K. S., Parthasarathy, K.: Identification and Control of Dynamical Systems Using Neural Networks, IEEE Trans. on Neural Networks, 1(1), pp. 4-27, March 1990.
- [3] Lee, R. C. K.: Optimal Estimation, Identification and Control, MIT Press, 1964.
- [4] Press, W. H.: Numerical Recipes in C, Cambridge University Press, Second Edition, 1992.
- [5] Föllinger, O.: Nichtlineare Regelungssysteme I,II, R. Oldenbourg, München, 7. Ausgabe, 1993.
- [6] Woodside, C. M.: Estimation of the Order of Linear Systems, Automatica, Vol. 7 (1971), pp. 727-733,
- [7] Schultz, J.: Improvement of nonlinear dynamic models containing neural networks based on domain partitioning, ECC '95, Roma, Italy, Sept. 5-8, 1995, pp. 1191-1196.
- [8] Brown, M.; Harris, C.: Neurofuzzy Adaptive Modelling and Control, Prentice Hall, 1994.