

Identifikation dynamischer Systeme (experimentelle Modellbildung)

Auf dem Wege der theoretischen Modellbildung erhält man in der Regel die grundlegende Struktur des mathematischen Modells des Prozesses und auch einige Parameter. Die Parameter werden dabei aus physikalischen Prozesskoeffizienten oder Grunddaten des Prozesses berechnet, sofern dies möglich ist. Manche Prozessabläufe sind jedoch nicht genau bekannt, und bei der Bestimmung der Parameter sind die Unsicherheiten oft groß. Als Ergänzung zur theoretischen Modellbildung bietet sich deshalb die *experimentelle Modellbildung* an, die *Prozess-Identifikation* genannt wird. Hierbei verwendet man gemessene Signale und ermittelt das zeitliche Verhalten des Systems innerhalb bestimmter Klassen von mathematischen Modellen.

Für mechatronische Systeme ist dieses Gebiet von Interesse, um theoretische Modelle zu validieren und um zur Adaption der modellgestützten Funktionen eine Online-Identifikation bzw. Parameterschätzung zu integrieren. Dies wird z.B. für modellgestützte Steuerungen und Regelungen, zur Kompensation des nichtlinearen Verhalten wie z.B. bei Reibung, Hysterese und vom Betriebspunkt abhängigem Verhalten, zur Fehlererkennung und Prozessoptimierung benötigt.

Im Folgenden wird eine kurzgefasste Darstellung mit Hinblick auf mechatronische Systeme gegeben. Eine ausführliche Darstellung ist z.B. in Eykhoff (1974), Young (1984), Isermann (1992), Ljung (1999) zu finden.

7.1 Identifikationsmethoden

7.1.1 Allgemeines Vorgehen

Es wird ein linearer Prozess nach Bild 7.1 betrachtet. Der Prozess sei stabil, so dass eine eindeutige Beziehung zwischen Eingangssignal $u(t)$ und Ausgangssignal $y_u(t)$ existiert. Auf den Prozess wirken Störsignale ein, die als Störsignalkomponente $y_z(t)$ dem Nutzsignal $y_u(t)$ am Ausgang überlagert gedacht werden können. Die *Aufgabe der Identifikation* besteht dann darin, aus dem bekannten Eingangssignal $u(t)$ und dem gestörten Ausgangssignal $y(t)$ ein mathematisches Modell des Prozesses P zu ermitteln. Dies kann man sich so vorstellen, dass ein Fehler $e(t)$ zwischen Prozess

und Modell gebildet wird (siehe Bild 7.2), der im Laufe der Messzeit bei Einsatz eines Identifikationsverfahrens kleiner wird. Wenn bei der Identifikation lediglich Ein- und Ausgangssignale zur Verfügung stehen, kann auch nur das Ein-/Ausgangsverhalten bestimmt werden. Dann geht nur der steuerbare und beobachtbare Teil des Prozesses in das ermittelte Modell ein.

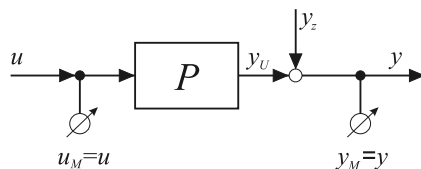


Bild 7.1. Linearer Prozess mit gestörtem Ausgangssignal

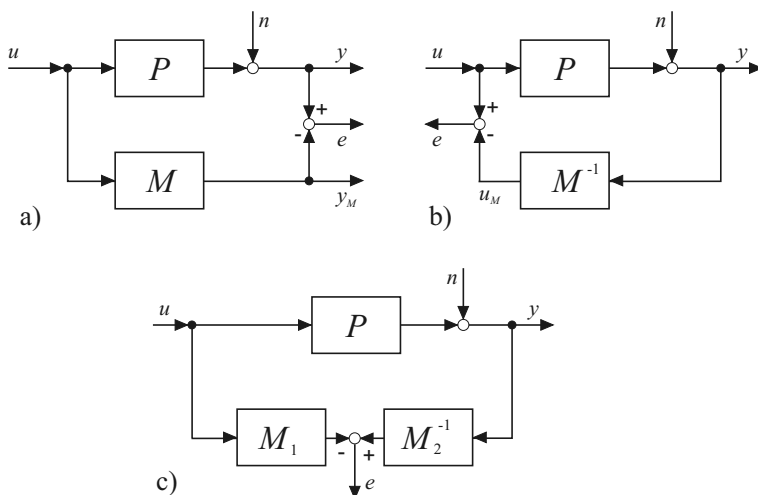


Bild 7.2. Bildung verschiedener Fehlersignale zwischen Prozess P und Modell M : a) Ausgangsfehler $e = y - Mu$; b) Eingangsfehler $e = u - M^{-1}y$; c) Verallgemeinerter Fehler $e = M_2^{-1}y - M_1u$

Bei einem Identifikationsexperiment sind grundsätzlich mehrere *Beschränkungen* zu beachten: Die Messzeit ist beschränkt, $T_M \leq T_{M,max}$; die Testsignalhöhe ist beschränkt, $u_0 \leq u_{0,max}$; die Ausgangssignaländerung ist beschränkt, $y_0 \leq y_{0,max}$. Ferner besteht das Störsignal $y_z(t)$ oft aus mehreren Komponenten: Stochastischer, stationärer Anteil $n(t)$, instationärer Anteil (Drift) $d(t)$ und nicht definierbare Anteile $h(t)$, wie z.B. Sprünge, Impulse.

7.1.2 Klassifikation von Identifikationsmodellen

Im Laufe der letzten 30 Jahre sind viele verschiedene Identifikationsmethoden entwickelt worden., z.B. Eykhoff (1974), Strobel (1975), Leonhard (1973), Young (1984), Ljung und Söderström (1985), so dass zum Verständnis der Unterschiede *Klassifikationsmerkmale* notwendig sind. Beschränkt man sich dabei auf lineare Prozesse mit konzentrierten Parametern und mit einem Ein- und Ausgang, dann kann man folgende Merkmale angeben:

- 1) *Typ der mathematischen Modelle*
Parametrische Modelle: Gleichungen, die die Parameter explizit enthalten (Differentialgleichungen, Übertragungsfunktionen).
Nichtparametrische Modelle: Funktionen in Form von Wertetafeln oder Kurvenverläufen (Gewichtsfunktionen, Frequenzgangwerte).
- 2) *Typ der im Modell verwendeten Signale*
Kontinuierliche Signale: Amplitude und Zeit haben einen kontinuierlichen Wertebereich.
Zeitdiskrete Signale: Diskrete Zeit und kontinuierliche Amplitude (Abtastsignale).
- 3) *Eingangssignale*
Determiniert oder stochastisch.
- 4) *Fehler zwischen Modell und Prozess*
Ausgangs-, Eingangs-, verallgemeinerter Fehler (siehe Bild 7.2)
- 5) *Kopplung von Prozess und Rechner*
Offline oder online.
- 6) *Art der Signalverarbeitung*
Blockverarbeitung oder Echtzeitverarbeitung.
- 7) *Art der algorithmischen Verarbeitung*
Nichtrekursiv oder rekursiv.
- 8) *Art der nichtrekursiven Verarbeitung*
Direkt (ein Rechenlauf) oder *iterativ* (mehrere Läufe).

7.1.3 Identifikationsmethoden

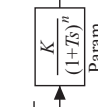
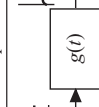
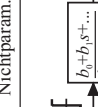
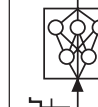
Die wichtigsten Identifikationsmethoden lassen sich wie folgt charakterisieren (vgl. Tabelle 7.1).

7.1.4 Testsignale

Nichtparametrische Modelle

Die *Frequenzgangmessung* mit periodischen Testsignalen dient zur direkten Ermittlung der Frequenzgangwerte. Gut bewährt hat sich die Auswertung durch orthogonale Korrelation, die in sogenannten Frequenzgangmessplätzen enthalten ist. Die erforderliche Messzeit ist bei mehreren erforderlichen Frequenzwerten relativ groß,

Tabelle 7.1. Übersicht der wichtigsten Identifikationsmethoden. ZVS: Zeitvariante Systeme, MGS: Mehrgrößensysteme, NLS: Nichtlineare Systeme

Ein-gangs-signal	Modell	Aus-gangs-signal	Identifika-tions-Methode	Mess-/Aus-Zulässiges Störsignalwertgerät	Digitalrechner Kopplung		Datenverarbeitung		Erreich-bare Ge-nauigkeit	Erweiterbarkeit		Anwendungs-beispiele
					offline	online	Block	Echtzeit		ZVS	MGS	NLS
			Kennwert-ermittlung	• Schreiber • Daten-speicher • PC	-	-	-	-	klein	-	-	Grobes Modell, Reglereinstellung
			Fourier-Analyse	• Schreiber • Daten-speicher • PC	X	-	X	-	mittel	-	X	Überprüfung theoretischer Modelle
			Frequenz-gang-messung	• Schreiber • Daten-speicher • F.G. Mess-gerät	X	-	-	-	sehr groß	-	X	Überprüfung theoretischer Modelle Entwurf klas-scher Regler
			Korrelation	• Korrelator • Prozess-rechner • PC	X	X	X	X	groß	X	X	Erkennung Signalzusam-menhänge Laufzeit-Identifikation
			Modell-anpassung	• Analog-rechner • PC • Prozess-rechner • PC	-	X	-	X	mittel	X	-	Adaptive Regelung
			Parameter-schätzung	• Prozess-rechner • PC	X	X	X	X	groß	X	X	Entwurf digitaler Regler Adaptive Regelung Fehlererkennung
			Adaption Neuronales Netz	• Prozess-rechner • PC	X	X	X	-	sehr groß	-	X	Entwurf nicht-linearer Regler Adaptive Regelung Fehlererkennung

die erreichbare Genauigkeit sehr groß. Die Frequenzgangmessung ist besonders bei Prozessen mit mehreren Resonanzfrequenzen zu empfehlen.

Die *Fourieranalyse* wird hauptsächlich für lineare Prozesse mit kontinuierlichen Signalen zur Ermittlung des Frequenzganges aus Sprung- und Impulsantwortfunktionen angewandt. Sie ist ein einfaches Verfahren mit relativ kleinem Rechenaufwand, relativ kurzer Messzeit und empfiehlt sich nur für Prozesse mit kleinem Störsignal/Testsignal-Verhältnis.

Die *Korrelationsanalyse* arbeitet im Zeitbereich und ist für lineare Prozesse sowohl mit zeitkontinuierlichen als auch zeitdiskreten Signalen im offenem Regelkreis geeignet. Zulässige Eingangssignale sind stochastische oder periodische Signale. Als Ergebnis erhält man Korrelationsfunktionen bzw. in Sonderfällen Gewichtsfunktionen. Korrelationsverfahren werden bei Prozessen mit großem Störsignal/Testsignal-Verhältnis bevorzugt angewandt. Der Rechenaufwand ist gering.

Die *Spektralanalyse* wird unter denselben Bedingungen wie die Korrelationsanalyse verwendet. Die Auswertung geschieht jedoch über den Frequenzbereich. Es werden Spektren berechnet. Das Ergebnis sind Frequenzgangwerte.

Als Apriori-Information dieser Methoden für nichtparametrische Modelle muss nur bekannt sein, dass der Prozess linearisierbar ist. Eine bestimmte Modellstruktur muss nicht angenommen werden. Deshalb eignen sich diese nichtparametrischen Methoden sowohl für Prozesse mit konzentrierten als auch verteilten Parametern mit beliebig komplizierter Struktur. Sie werden bevorzugt zur Überprüfung theoretisch abgeleiteter Modelle verwendet, denn dann ist man besonders daran interessiert, keine bestimmte Modellstruktur annehmen zu müssen.

Parametrische Modelle

Bei den Identifikationsmethoden für parametrische Modelle muss eine bestimmte Modellstruktur angenommen werden. Falls die angenommene Struktur zutrifft, sind wegen der höheren Apriori-Information im Zusammenhang mit statistischen Ausgleichsmethoden genauere Ergebnisse zu erwarten.

Am Einfachsten sind die Methoden der *Kennwertermittlung*. Hierbei werden aus gemessenen Antwortfunktionen auf nichtperiodische Testsignale bestimmte Kennwerte wie z.B. Verzugszeit und Ausgleichszeit entnommen, und es werden aufgrund von Tabellen und Diagrammen die Parameter einfacher Modelle bestimmt. Diese Kennwertermittlung ist aber bei Auswertung von Hand nur für einfache Prozesse anwendbar, wenn nur kleine Störsignale einwirken.

Die *Referenzmodellmethoden* oder *Modellabgleichmethoden* wurden für analog realisierte Modelle mit kontinuierlichen Signalen entwickelt, sie liefern nach Annahme einer bestimmten Modellstruktur die Parameter von Differentialgleichungen oder Differenzengleichungen. Für die Eingangssignale gilt meist nur die Voraussetzung, dass alle interessierenden Eigenfrequenzen des Prozesses genügend angeregt werden. Die Referenzmodell-Verfahren haben jedoch zugunsten der Parameterschätzverfahren an Bedeutung verloren.

Parameterschätzmethoden gehen von Differenzengleichungen oder Differentialgleichungen beliebiger, aber nicht zu hoher Ordnung und Totzeit aus. Durch speziell für dynamische Prozesse entwickelte statistische Ausgleichsverfahren werden Funk-

tionen von Fehlersignalen minimiert. Es sind beliebige Eingangssignale und auch größere Störsignale zulässig. Parameterschätzmethoden lassen sich deshalb vielseitig einsetzen und erlauben auch bei ungünstigem Störsignal/Nutzsignal-Verhältnis noch die Ermittlung von Modellen, auch im geschlossenen Regelkreis.

Zur Identifikation nichtlinearer Prozesse ohne wesentliche Apriori-Kenntnisse der Modellstruktur eignen sich *künstliche neuronale Netze*, die aus mathematisch formulierten Neuronen als nichtlineare Modellmodule zusammengesetzt werden.

Im Allgemeinen ist zu empfehlen, künstlich erzeugte Testsignale zu verwenden, da die im Betrieb auftretenden Signale die Prozesse selten genügend anregen. An diese Testsignale werden im Allgemeinen folgende Forderungen gestellt:

- Einfach und reproduzierbar zu erzeugen
- einfach mathematisch beschreibbar
- realisierbar mit den gegebenen Stellanrichtungen
- anwendbar auf den Prozess
- gute Anregung der interessierenden Prozessdynamik und des Signalamplitudenbereiches.

Bild 7.3 zeigt einige häufig verwendete Testsignale. Bei Korrelationsmethoden und Parameterschätzmethoden für lineare Prozesse hat sich das **Pseudo-Rausch-Binär-Signal** (PRBS) sehr gut bewährt. Speziell bei Flugzeugen wird gerne ein 3211-Signal verwendet, das als Kombination von Rechteckimpulsen oder Teile eines PRBS aufgefasst werden kann. Für nichtlineare Prozesse ist ein amplitudenmoduliertes APRBS geeignet.

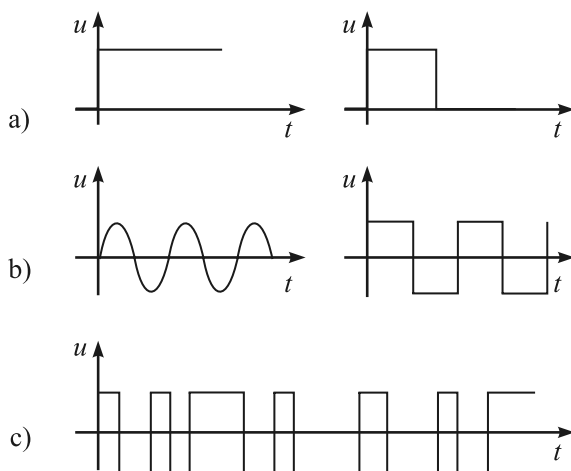


Bild 7.3. Häufig verwendete Testsignale: a) nichtperiodisch: Sprungfunktion und Rechteckimpuls; b) Periodisch: Sinus- und Rechteckschwingung; c) Stochastisch: Diskretes binäres Rauschen

7.1.5 Geschlossener Regelkreis

Proportional wirkende Prozesse können im Allgemeinen bei geöffneten Regelkreisen identifiziert werden. Bei integral wirkenden Prozessen ist dies jedoch häufig nicht möglich, da entweder störende Driftsignale einwirken oder aber der Betrieb des Prozesses dies nicht für längere Zeit zulässt (Wegdriften des Betriebspunktes). In solchen Fällen und auch bei instabilen Prozessen muss man deshalb im geschlossenen Regelkreis identifizieren (siehe Bild 7.4). Wenn ein externes Signal, z.B. die Führungsgröße w , messbar ist, dann kann der Prozess mit Korrelations- oder Parameterschätzmethoden identifiziert werden, siehe Isermann (1992). Wirkt jedoch kein messbares externes Signal ein, z.B. lediglich das Störsignal y_z , dann können Parameterschätzmethoden unter Beachtung von Identifizierbarkeitsbedingungen eingesetzt werden.

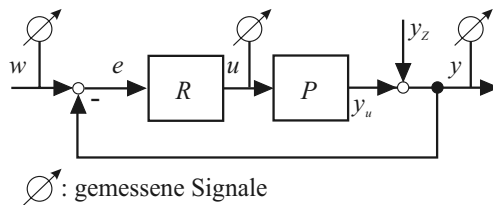


Bild 7.4. Zur Identifikation im geschlossenen Regelkreis

7.1.6 Art der Anwendung

Einen ausschlaggebenden Einfluss auf die Wahl des mathematischen Modells, die Identifikationsmethode, die erforderliche Genauigkeit des Modells und die einzusetzenden Geräte hat die Anwendung des identifizierten Modells.

Für Anwendungen in mechatronischen Systemen oder zu ihrem Entwurf sind besonders die Parameterschätzmethoden von Bedeutung, die online und in Echtzeit arbeiten können und relativ einfach auch auf bestimmte Klassen von nichtlinearen Prozessen übertragen werden können. Deshalb sollen im Folgenden diese Methoden und bestimmte neuronale Netze für nichtlineare Mehrgrößensysteme betrachtet werden.

7.2 Parameterschätzung für zeitdiskrete Signale

Es wird davon ausgegangen, dass sich der Prozess durch eine lineare Differenzengleichung

$$\begin{aligned} y_u(k) + a_1 y_u(k-1) + \dots + a_m y_u(k-m) = \\ = b_1 u(k-d-1) + \dots + b_m u(k-d-m) \end{aligned} \quad (7.2.1)$$

beschreiben lässt. Hierbei sind

$$\begin{aligned} u(k) &= U(k) - U_{00}, \\ y_u(k) &= Y_u(k) - y_{00} \end{aligned} \quad (7.2.2)$$

die Änderungen der absoluten Signalwerte $U(k)$ und $Y_u(k)$ von den Gleichwerten U_{00} und Y_{00} . k ist die diskrete Zeit $k = t/T_0 = 0, 1, 2, \dots$, T_0 ist die Abtastzeit, $d = T_t/T_0 = 0, 1, 2, \dots$ die diskrete Totzeit mit der Totzeit T_t . Die zugehörige z -Übertragungsfunktion lautet

$$\begin{aligned} G_P(z) &= \frac{y_u(z)}{u(z)} = \frac{B(z^{-1})}{A(z^{-1})} z^{-d} = \\ &= \frac{b_1 z^{-1} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + \dots + a_m z^{-m}} z^{-d}. \end{aligned} \quad (7.2.3)$$

Das Messsignal enthalte ein stationäres, stochastisches Störsignal

$$y(k) = y_u(k) + n(k) \quad \text{mit} \quad E\{n(k)\} = 0. \quad (7.2.4)$$

Die Aufgabe besteht darin, die unbekannten Parameter a_i und b_i aus N gemessenen Ein- und Ausgangssignalen zu bestimmen.

7.2.1 Methode der kleinsten Quadrate (LS)

Bezeichnet man die bis zu einem Zeitpunkt $(k-1)$ bestimmten Parameter mit a'_i und b'_i , dann gilt mit dem gestörten Ausgangssignal

$$\begin{aligned} &y(k) + a'_1 y(k-1) + \dots + a'_m y(k-m) \\ &- b'_1 u(k-d-1) - \dots - b'_m u(k-d-m) = e(k), \end{aligned} \quad (7.2.5)$$

wobei anstelle von 0 wie bei der entsprechend geschriebenen (7.2.1) der *Gleichungsfehler* (Residuum) eingeführt wird. Dieser entspricht einem verallgemeinerten Fehler (Bild 7.2), was aus der Form

$$A'(z^{-1})y(z) - B'(z^{-1})z^{-d}u(z) = e(z) \quad (7.2.6)$$

ersichtlich ist (siehe Bild 7.5). Die Eigenschaft, dass e linear von den gesuchten Parametern abhängig ist (Linearität den Parametern), ermöglicht eine direkte Parameterschätzung.

In (7.2.5) kann man $y'(k|k-1)$ als Einschnitt-Vorhersage aufgrund von Messungen bis $(k-1)$ auffassen und schreiben

$$y'(k|k-1) = \psi^T(k) \Theta' \quad (7.2.7)$$

mit dem Datenvektor

$$\begin{aligned} \psi^T(k) &= [-y(k-1) \dots - y(k-m) | u(k-d-1) \dots \\ &\quad u(k-d-m)] \end{aligned} \quad (7.2.8)$$

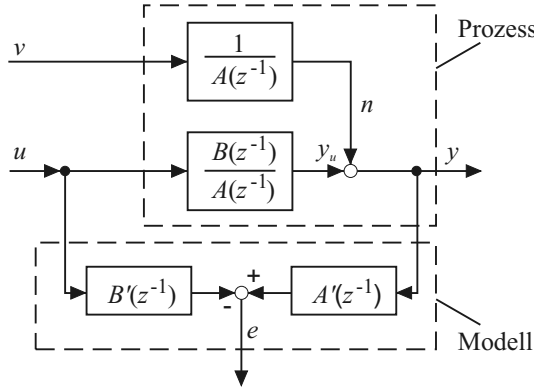


Bild 7.5. Modellanordnung bei der Methode der kleinsten Quadrate mit Gleichungsfehler (verallgemeinerter Fehler)

und dem Parametervektor

$$\Theta' = [a'_1 \dots a'_m | b'_1 \dots b'_m]^T. \quad (7.2.9)$$

(7.2.5) lautet dann

$$y(k) = \psi^T(k) \Theta' + e(k). \quad (7.2.10)$$

Die gemessenen Signale für $k = m + d, \dots, m + d + N$ werden in Vektoren angeordnet, z.B.

$$\mathbf{y}^T(m + d + N) = [y(m + d) \dots y(m + d + N)]. \quad (7.2.11)$$

Dann gilt

$$\mathbf{y}(m + d + n) = \Psi(m + d + N) \Theta' + \mathbf{e}(m + d + N), \quad (7.2.12)$$

wobei Ψ eine $((N + 1), 2m)$ -Datenmatrix ist. Nun wird die Summe der Gleichungsfehlerquadrate

$$V = \sum_{k=m+d}^{m+d+N} e^2(k) = \mathbf{e}^T(m + d + N) \mathbf{e}(m + d + N) \quad (7.2.13)$$

bezüglich der unbekannten Parameter minimiert

$$\left. \frac{dV}{d\Theta'} \right|_{\Theta'=\hat{\Theta}} = -2\Psi^T[\mathbf{y} - \Psi\Theta'] = \mathbf{0}. \quad (7.2.14)$$

Hieraus folgt die (nichtrekursive) Schätzgleichung der Methode der kleinsten Quadrate (LS: least squares, ausführliche Ableitung siehe z.B. Isermann (1992))

$$\hat{\Theta} = [\Psi^T \Psi]^{-1} \Psi^T \mathbf{y}. \quad (7.2.15)$$

Die Matrix

$$\mathbf{P} = [\Psi^T \Psi]^{-1} \quad (7.2.16)$$

hat die Dimension $(2m, 2m)$. Damit ihre Inverse existiert, muss gelten

$$\det[\Psi^T \Psi] = \det \mathbf{P}^{-1} \neq 0 \quad (7.2.17)$$

und damit die Verlustfunktion V ein Minimum annimmt, muss

$$\frac{\partial^2 V}{\partial \Theta \partial \Theta^T} = \Psi^T \Psi \quad (7.2.18)$$

positiv definit sein. Beides wird erfüllt durch

$$\det[\Psi^T \Psi] = \det \mathbf{P}^{-1} > 0. \quad (7.2.19)$$

Die Bedingung schließt ein, dass das Eingangssignal *fortdauernd angeregt* ist und der Prozess *stabil* ist. Von einer Parameterschätzung wird im Allgemeinen erwartet, dass sie erwartungstreu ist für endliche N (Θ_0 wahre Parameter)

$$E \{ \hat{\Theta}(N) \} = \Theta_0 \quad (7.2.20)$$

und konsistent im quadratischen Mittel, d.h.

$$\lim_{N \rightarrow \infty} E \{ \hat{\Theta}(N) \} = \Theta_0, \quad (7.2.21)$$

$$\lim_{N \rightarrow \infty} E \left\{ \left[\hat{\Theta}(N) - \Theta_0 \right] \left[\hat{\Theta}(N) - \Theta_0 \right]^T \right\} = \mathbf{0}. \quad (7.2.22)$$

Für die Methode der kleinsten Quadrate wird nach Einsetzen von (7.2.12) in (7.2.15)

$$\begin{aligned} E \{ \hat{\Theta}(N) \} &= \Theta_0 + E \left\{ \left[\Psi^T \Psi \right]^{-1} \Psi^T \mathbf{e} \right\} = \\ &= \Theta_0 + \mathbf{b}. \end{aligned} \quad (7.2.23)$$

Damit der Bias (systematischer Schätzfehler) $\mathbf{b}$ verschwindet, dürfen Ψ^T und $\mathbf{e}$ nicht korreliert sein. Hieraus folgt, dass $e(k)$ nicht korreliert sein darf und $E \{ e(k) \} = 0$. Eine erwartungstreu Schätzung wird dann erreicht, wenn das Störsignal $n(k)$ über ein Störsignalfilter

$$G_v(z) = \frac{n(z)}{v(z)} = \frac{1}{A(z^{-1})} \quad (7.2.24)$$

erzeugt wird, wobei $v(k)$ ein diskretes weißes Rauschen ist (Bild 7.5). Da dieses Filter jedoch praktisch nicht existiert, liefert die Methode der kleinsten Quadrate im Allgemeinen Schätzungen mit Bias. Diese systematischen Schätzfehler sind dabei um so größer, je größer die Varianz σ_n^2 der Störsignale im Vergleich zur Varianz des Nutzsignals σ_{yu}^2 ist.

Für die Kovarianzmatrix gilt mit $E \{ \hat{\Theta} \} = \Theta_0$ (also $\mathbf{b} = \mathbf{0}$)

$$\begin{aligned} cov[\Delta \Theta] &= E \left\{ \left[\hat{\Theta} - \Theta_0 \right] \left[\hat{\Theta} - \Theta_0 \right]^T \right\} \sigma_e^2 E \{ \mathbf{P} \} = \\ &= \sigma_e^2 E \left\{ \left[\frac{1}{N+1} \Psi^T \Psi \right]^{-1} \right\} \frac{1}{N+1} \\ &= \sigma_e^2 \left\{ \Phi^{-1} (N+1) \right\} \frac{1}{N+1}. \end{aligned} \quad (7.2.25)$$

σ_e^2 ist die Varianz von $e(k)$. Φ ist eine Matrix, deren Elemente aus Korrelationsfunktionen bestehen. Für $N \rightarrow \infty$ wird (7.2.22) erfüllt. $E \{ \mathbf{P} \}$ ist proportional zur Kovarianzmatrix der Parameterschätzfehler.

Aufgrund der nicht erwartungstreuen Schätzung kann die Methode der kleinsten Quadrate nur bei nicht oder nur wenig gestörten Prozessen eingesetzt werden. Ein großer Vorteil ist aber die Schätzung der Parameter $\hat{\Theta}$ in einem Rechenlauf, also nicht durch iterative Verfahren, was durch die Verwendung eines Fehlersignals ermöglicht wurde, das linear in den Parametern ist.

7.2.2 Rekursive Methode der kleinsten Quadrate

Die bisher betrachtete Methode der kleinsten Quadrate ist nichtrekursiv, d.h. man muss erst alle Daten speichern und dann $\hat{\Theta}(N)$ (in einem Lauf) berechnen. Wenn man jedoch nach jeder neuen Messung von $u(k)$ und $y(k)$ die neuesten Parameterschätzwerte $\hat{\Theta}(k)$ haben möchte (Echtzeitanwendungen), dann kann man (7.2.15) in eine rekursive Schreibweise umformen und erhält die RLS-Schätzgleichung (recursive least-squares)

$$\hat{\Theta}(k+1) = \hat{\Theta}(k) + \gamma(k) [y(k+1) - \psi^T(k+1)\hat{\Theta}(k)], \quad (7.2.26)$$

$$\gamma(k) = \frac{1}{\psi^T(k+1)\mathbf{P}(k)\psi(k+1) + 1} \mathbf{P}(k)\psi(k+1), \quad (7.2.27)$$

$$\mathbf{P}(k+1) = [I - \gamma(k)\psi^T(k+1)\mathbf{P}(k)]. \quad (7.2.28)$$

Zum Start wird $\hat{\Theta}(0) = \mathbf{0}$ gesetzt und es werden große Varianzen $\mathbf{P}(0) = \alpha \mathbf{I}$, $\alpha = 100 \dots 1000$, angenommen. Wegen numerischer Vorteile werden bei der Implementierung sogenannte Wurzelfilterverfahren z.B. DSFC, DSFI oder DUDC bevorzugt, siehe Isermann (1992).

7.2.3 Modifikation der Methode der kleinsten Quadrate

Um bei den praktisch vorkommenden farbigen Störsignalen erwartungstreue Parameterschätzungen zu erhalten, sind mehrere Modifikationen der ursprünglichen Methode der kleinsten Quadrate vorgeschlagen worden. Hierzu zählen z.B. die Methoden der *verallgemeinerten kleinsten Quadrate* (GLS) und der *erweiterten kleinsten*

Quadrate (ELS) und die *Maximum-Likelihood-Methode*, siehe Isermann (1992). In der praktischen Anwendung hat sich jedoch gezeigt, dass diese Methoden nur dann bessere Ergebnisse als LS liefern, wenn die Messzeiten sehr groß sind. Deshalb reicht in vielen Fällen die LS- oder RLS-Methode, in Form von z.B. DSFI aus, wenn die Störsignale nicht sehr groß sind im Vergleich zu den anregenden Eingangssignalen.

7.3 Parameterschätzung für zeitkontinuierliche Signale

Die Parameterschätzmethoden für dynamische Prozesse wurden zunächst hauptsächlich für Prozessmodelle mit zeitdiskreten Signalen im Zusammenhang mit digitalen Regelungen entwickelt. Für manche Anwendungen, wie z.B. zur Überprüfung theoretischer Modelle und zur Fehlerdiagnose, ist jedoch die Parameterschätzung für Modelle mit zeitkontinuierlichen Signalen erforderlich.

7.3.1 Methode der kleinsten Quadrate

Es wird ein stabiler Prozess mit konzentrierten Parametern betrachtet, der durch eine lineare, zeitinvariante Differentialgleichung

$$\begin{aligned} a_n y_u^{(n)}(t) + a_{n-1} y_u^{(n-1)}(t) + \dots + a_1 y_u^{(1)}(t) + y_u(t) = \\ = b_m u^{(m)}(t) + b_{m-1} u^{(m-1)}(t) + \dots + b_1 u^{(1)}(t) + b_0 u(t) \quad m < n \end{aligned} \quad (7.3.1)$$

beschrieben werden kann. Dabei wird angenommen, dass die Ableitungen des Ausgangssignals

$$y^{(j)}(t) = d^j y(t)/dt^j, \quad j = 1, 2, \dots, n \quad (7.3.2)$$

und des Eingangssignales für $j = 1, 2, \dots, m$ existieren. $u(t)$ und $y(t)$ sind die Änderungen

$$\begin{aligned} u(t) &= U(t) - U_{00}, \\ y(t) &= Y(t) - Y_{00} \end{aligned} \quad (7.3.3)$$

der absoluten Signalwerte $U(t)$ und $Y(t)$ von den Gleichwerten U_{00} und Y_{00} . Zu (7.3.1) gehört die Übertragungsfunktion

$$\begin{aligned} G_P(s) &= \frac{y_u(s)}{u(s)} = \frac{B(s)}{A(s)} = \\ &= \frac{b_0 + b_1 s + \dots + b_{m-1} s^{m-1} + b_m s^m}{1 + a_1 s + \dots + a_{n-1} s^{n-1} + a_n s^n}. \end{aligned} \quad (7.3.4)$$

Das messbare Signal $y(t)$ enthalte ein überlagertes Störsignal $n(t)$, also

$$y(t) = y_u(t) + n(t). \quad (7.3.5)$$

Setzt man (7.3.5) in (7.3.1) ein und führt einen Gleichungsfehler $e(t)$ ein, dann wird (entsprechend zu (7.2.10))

$$y(t) = \boldsymbol{\psi}^T(t) \boldsymbol{\Theta}' + e(t) \quad (7.3.6)$$

mit

$$\boldsymbol{\psi}^T(t) = [-y^{(1)}(t) \dots -y^{(n)}(t) | u(t) \dots u^{(m)}(t)], \quad (7.3.7)$$

$$\boldsymbol{\Theta}' = [a'_1 \dots a'_n | b'_0 \dots b'_m]^T. \quad (7.3.8)$$

Es werden nun Ein- und Ausgangssignale zu den diskreten Zeitpunkten $t = kT_0$, $k = 0, 1, 2, \dots, N$, mit der Abtastzeit T_0 gemessen und ihre Ableitungen gebildet. Dann entstehen $N + 1$ Gleichungen

$$y(k) = \boldsymbol{\psi}^T(k) \boldsymbol{\Theta}' + e(k). \quad (7.3.9)$$

Dieses Gleichungssystem wird nun in eine Matrixdarstellung gebracht

$$\mathbf{y} = \boldsymbol{\Psi} \boldsymbol{\Theta}' + \mathbf{e}. \quad (7.3.10)$$

Durch Minimieren der Verlustfunktion

$$V = \mathbf{e}^T(N) \mathbf{e}(N) = \sum_{k=0}^N e^2(k) \quad (7.3.11)$$

erhält man über $dV/d\boldsymbol{\Theta}' = \mathbf{0}$ wie in Abschnitt 7.2 den Parameterschätzwertvektor nach der Methode der kleinsten Quadrate

$$\hat{\boldsymbol{\Theta}}(N) = [\boldsymbol{\Psi}^T \boldsymbol{\Psi}]^{-1} \boldsymbol{\Psi}^T \mathbf{y}. \quad (7.3.12)$$

Die Existenz einer eindeutigen Lösung setzt voraus, dass die Matrix $\boldsymbol{\Psi}^T \boldsymbol{\Psi}$ positiv definit ist. Nach Division durch die Messzeit erhält man als Elemente dieser Matrix Korrelationsfunktionsschätzwerte $\hat{R}(\tau)$ der Signalableitungen für $\tau = 0$, also ohne Zeitverschiebung.

In der äußeren Gestalt ist viel Ähnlichkeit zur Methode der kleinsten Quadrate für Modelle mit zeitdiskreten Signalen zu erkennen. Deshalb können eine Reihe von Darstellungen direkt übernommen werden, wie z.B. die rekursive Form und numerisch verbesserte Versionen. Besondere Probleme treten allerdings bei der Konvergenz und der Ermittlung der benötigten Ableitungen der Signalwerte auf.

Eine *Konvergenzanalyse* zeigt, dass sich bei der LS-Methode für kontinuierliche Signale auch dann keine erwartungstreuen Schätzwerte ergeben, wenn das Fehlersignal $e(t)$ statistisch unabhängig ist. Deshalb treten bei gestörten Prozessen grundsätzlich Bias auf.

Wenn die erforderlichen *Ableitungen der Signale* direkt messbar sind (wie z.B. bei Fahrzeugen), können diese Messwerte in die Datenmatrix $\boldsymbol{\Psi}$ eingesetzt bzw. die in $[\boldsymbol{\Psi}^T \boldsymbol{\Psi}]/(N + 1)$ stehenden Korrelationsfunktionen direkt berechnet werden. Sind die Ableitungen jedoch nicht messbar, dann müssen sie aus den abgetasteten Signalen $u(t)$ und $y(t)$ bestimmt werden. Hierzu gibt es im Wesentlichen zwei Möglichkeiten. Die *numerische Differentiation* in Verbindung mit Interpolationsverfahren (Spline, Newton) kann jedoch Störsignaleinflüsse nicht genügend abschwächen. Besser bewährt hat sich ein *Zustandsvariablenfilter* (ZVF)

$$F(s) = \frac{y_f(s)}{y(s)} = \frac{1}{f_0 + f_1 s + \dots + f_{n-1} s^{n-1} + s^n}, \quad (7.3.13)$$

(siehe Bild 7.6). Dies ist ein Tiefpassfilter in Zustandsdarstellung, das sowohl die Ableitungen liefert als auch Störsignale ausfiltert. Damit werden sowohl $u(t)$ als auch $y(t)$ gefiltert. Die Wahl der Filterparameter f_i ist weitgehend frei. Es wird die Auslegung als Butterworth-Filter empfohlen, Peter (1993).

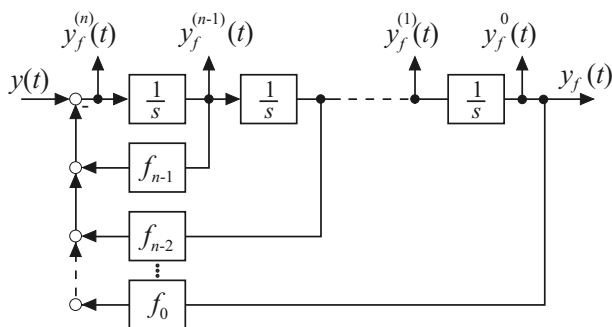


Bild 7.6. Zustandsvariablenfilter zur Bestimmung von Signalableitungen

Bei kleinen Störsignal/Nutzsignal-Verhältnissen hat sich diese Methode der kleinsten Quadrate gut bewährt.

Bei größeren Störsignalen sollten konsistente Parameterschätzmethoden eingesetzt werden. Hierzu eignet sich die *Methode der Hilfsvariablen*.

Da die Parameterschätzmethoden schließlich wieder auf zeitdiskrete Schätzalgorithmen führen, können viele Ergebnisse und Schätzmethoden für zeitdiskrete Modelle sinngemäß übernommen werden.

7.4 Zeitvariante Systeme

Bei den bisher betrachteten Identifikationsmethoden wurde angenommen, dass die Parameter konstant sind. Dies trifft jedoch in der Praxis selten zu, da die Parameter sich durch interne oder externe Einflüsse mit der Zeit ändern. Besonders häufig ist der Fall, dass für kleine Signaländerungen lineares Verhalten um den Betriebspunkt angenommen werden darf. Bei Änderungen des Betriebspunktes macht sich dann das in Wirklichkeit nichtlineare Verhalten bemerkbar. Man kann dann versuchen, mit linearen Modellen, aber zeitinvarianten Parametern, weiterzukommen. Hierzu eignen sich *rekursive Parameterschätzmethoden*, die im Folgenden am Beispiel der Methode der kleinsten Quadrate kurz beschrieben werden.

Um die jüngeren Messwerte stärker zu gewichten als die älteren Messwerte, kann man die Fehlerquadrate mit Gewichten $w(k)$ versehen, die gemäß

$$w(k) = \lambda^{N^*-k}, \quad 0 < \lambda < 1 \quad (7.4.1)$$

exponentiell abnehmen, wie man aus

$$w(N^*) = 1, \quad w(N^* - 1) = \lambda, \quad w(N^* - 2) = \lambda^2, \dots \quad (7.4.2)$$

sieht.

Hierbei wird λ *Vergessensfaktor* genannt.

Die Verlustfunktion lautet dann

$$V = \sum_{k=N^*-N}^{N^*} w(k) e^2(k) = \sum_{k=N^*-N}^{N^*} \lambda^{N^*-k} e^2(k). \quad (7.4.3)$$

Aus der gewichteten Methode der kleinsten Quadrate folgen dann die rekursiven Schätzalgorithmen

$$\hat{\Theta}(k+1) = \hat{\Theta}(k) + \gamma(k)[y(k+1) - \psi^T(k+1)\hat{\Theta}(k)], \quad (7.4.4)$$

$$\gamma(k) = \frac{1}{\psi^T(k+1)\mathbf{P}(k)\psi(k+1) + \lambda} \mathbf{P}(k)\psi(k+1), \quad (7.4.5)$$

$$\mathbf{P}(k+1) = [\mathbf{I} - \gamma(k)\psi^T(k+1)]\mathbf{P}(k) \frac{1}{\lambda}. \quad (7.4.6)$$

Der Einfluss des Vergessensfaktors lässt sich aus der Inversen der Kovarianzmatrix erkennen, die bei der Ableitung der rekursiven Algorithmen benötigt wird

$$\mathbf{P}^{-1}(k+1) = \lambda \mathbf{P}^{-1}(k) + \psi(k+1)\psi^T(k+1). \quad (7.4.7)$$

$\mathbf{P}^{-1}$ ist dabei proportional zur Informationsmatrix

$$J = \frac{1}{\sigma_e^2} E \{ \Psi^T \Psi \} = \frac{1}{\sigma_e^2} E \{ \mathbf{P}^{-1} \}. \quad (7.4.8)$$

Durch $\lambda < 1$ wird die Information des letzten Schritts verkleinert bzw. es werden die Kovarianzwerte vergrößert. Dadurch werden schlechtere alte Schätzwerte angenommen, so dass die neuen Messwerte ein größeres Gewicht erhalten.

Für $\lambda = 1$ (kein nachlassendes Gedächtnis) gilt

$$\begin{aligned} \lim_{k \rightarrow \infty} E \{ \mathbf{P}(k) \} &= \mathbf{0}, & \lim_{k \rightarrow \infty} \mathbf{P}^{-1}(k) &= \infty \\ \lim_{k \rightarrow \infty} E \{ \gamma(k) \} &= \mathbf{0}. \end{aligned} \quad (7.4.9)$$

Mit $\lambda < 1$ entstehen endliche Grenzwerte

$$\begin{aligned} \lim_{k \rightarrow \infty} E \{ \mathbf{P}(k) \} &= \mathbf{P}(\infty), & \lim_{k \rightarrow \infty} E \{ \mathbf{P}^{-1}(k) \} &= \mathbf{P}^{-1}(\infty), \\ \lim_{k \rightarrow \infty} E \{ \gamma(k) \} &= \gamma(\infty). \end{aligned} \quad (7.4.10)$$

Deshalb gehen bei großen k die neuen Messwerte mit konstantem Gewicht ein und nicht mit immer kleiner werdendem Gewicht wie bei $\lambda = 1$. Der Schätzalgorithmus

bleibt dadurch empfindlich für eventuelle Parameteränderungen. Wegen der kleiner werdenden effektiven Mittelungszeit wird allerdings der Störsignaleinfluss größer. Bei der Wahl von λ muss man deshalb einen geeigneten Kompromiss schließen zwischen der Fähigkeit, Parameteränderungen gut zu folgen (λ klein) und Störsignale gut zu eliminieren (λ groß). Die exponentielle Gewichtung erlaubt jedoch nur relativ *langsame Parameteränderungen*, da $\boldsymbol{y}(k)$ sich wie $\boldsymbol{P}(k+1)$ nur langsam (exponentiell) ändert.

Von besonderer Bedeutung ist bei $\lambda < 1$, dass der Prozess fortlaufend angeregt werden muss. Sonst nimmt $\boldsymbol{P}^{-1}(k+1)$ wegen $\boldsymbol{\psi}(k+1) \approx 0$ laufend ab bzw. $\boldsymbol{P}(k+1)$ und $\boldsymbol{y}(k)$ laufend zu und der Schätzalgorithmus wird immer empfindlicher, so dass er schließlich divergiert. Man kann deshalb $\lambda(k)$ variabel machen, z.B. in Abhängigkeit vom Informationsinhalt oder Eigenwert des Parameterschätzalgorithmus, Kofahl (1988), Isermann (1992).

7.5 Nichtlineare Prozesse

Zur Identifikation nichtlinearer Prozesse eignen sich Parameterschätzverfahren und künstliche neuronale Netze. Im Folgenden werden die nichtlinearen Prozesse nach stetig differenzierbar und nicht stetig differenzierbar unterschieden.

7.5.1 Parameterschätzung mit klassischen nichtlinearen Modellen

Wenn der Prozess stetig differenzierbare Beziehungen zwischen den Ein- und Ausgangssignalen aufweist, dann können polynomiale Approximation eingesetzt werden. Man unterscheidet allgemein anwendbare Strukturen wie z.B. Volterra-Reihen oder Kolmogorov-Gabor-Polynome, oder spezielle Strukturen wie z.B. Hammerstein-, Wienermodelle oder nichtlineare Differenzengleichungen (NDE-Modelle), siehe z.B. Eykhoff (1974), Haber und Unbehauen (1990), Isermann et al. (1992). Man kann aber auch einen Ausgangsfehler bilden und die Maximum-Likelihood-Methode anwenden. Dies ergibt aber meist umfangreiche, iterativ ablaufende Rechenprozeduren.

Einfachere Verhältnisse ergeben sich somit, wenn man mit nichtlinearen Modellen arbeitet, die *linear in den Parametern* sind. Hierzu schreibt man die lineare Differenzengleichung mit dem Zeitverschiebeoperator q^{-1} ($q^{-i}y(k) = y(k-i)$) wie folgt

$$A(q^{-1})y(k) = B(q^{-1})q^{-d}u(k) + D(q^{-1})v(k) \quad (7.5.1)$$

mit Polynomen entsprechend (7.2.3).

Folgende nichtlineare, dynamische Modelle sind dann linear in den Parametern

1) *Allgemeines Hammerstein-Modell:*

$$\begin{aligned} A(q^{-1})y(k) = & B_1(q^{-1})u(k) + \\ & + B_2(q^{-1})u^2(k) + \dots + D(q^{-1})v(k). \end{aligned} \quad (7.5.2)$$

2) *Parametrisches Volterra-Modell:*

$$\begin{aligned}
 A(q^{-1})y(k) = & B_1(q^{-1})u(k) + \\
 & + \sum_{\alpha} B_2(q^{-1})u(k)[q^{-\alpha}u(k)] + \\
 & + D(q^{-1})v(k).
 \end{aligned} \tag{7.5.3}$$

3) *Nichtlineares Modell nach Lachmann (1983) (NDE):*

$$\begin{aligned}
 A_1(q^{-1})y(k) + \sum_{\alpha} A_2(q^{-1})y(k)[q^{-\alpha}y(k)] = \\
 = B(q^{-1})u(k) + D(q^{-1})v(k).
 \end{aligned} \tag{7.5.4}$$

Für diese Modelle können die einfachen Parameterschätzmethoden LS, RLS, RELS direkt angewendet werden, Lachmann (1983), Isermann et al. (1992).

7.5.2 Künstliche neuronale Netze

Die Anwendung der bewährten Parameterschätzmethoden auf nichtlineare Prozesse setzt eine relativ genaue Kenntnis der Struktur des nichtlinearen Prozessmodells voraus. Es sind deshalb für einen allgemeinen Ansatz Identifikationsmethoden für nichtlineare Prozesse von Interesse, die ohne wesentliche Strukturkenntnis auskommen und flexibel einsetzbar sind. Diese Eigenschaften haben *künstliche neuronale Netze*, die aus mathematisch formulierten Neuronen zusammengesetzt werden.

Diese mathematisch formulierten Neuronen wurden zunächst dazu verwendet das Verhalten von biologischen Neuronen zu beschreiben, McCulloch und Pitts (1943). Die Zusammenschaltung in Netzen erlaubte dann, statische Zusammenhänge zwischen Ein- und Ausgangssignalen zu beschreiben, Rosenblatt (1958), Widrow und Hoff (1960). Im Folgenden werden künstliche neuronale Netze (KNN) betrachtet, die Eingangssignale $\mathbf{u}$ in Ausgangssignale $\mathbf{y}$ abbilden, Bild 7.7. In der Regel sind die freien Parameter eines KNN nicht bekannt. Dann müssen sie durch die Verarbeitung von gemessenen Signalen $\mathbf{u}$ und $\mathbf{y}$ adaptiert oder „trainiert“ oder „gelernt“ werden, Hecht-Nielsen (1990), Haykin (1994). Es handelt sich also um ein Problem der System-Identifikation. Wenn die Ein- und Ausgänge zu Gruppen oder Mustern (Clustern) zusammengefasst werden, liegt eine Aufgabe der Klassifikation im Zusammenhang mit z.B. einer Mustererkennung vor, Bishop (1995). Im Folgenden wird das Problem der Identifikation nichtlinearer Systeme betrachtet (supervised learning). Dabei wird die Eigenschaft von KNN genutzt, eindeutig nichtlineare Zusammenhänge beliebig genau approximieren zu können. Im Folgenden werden zunächst KNN für das *statische Übertragungsverhalten* betrachtet, Hafner et al. (1992), Preuß und Tresp (1994), die dann für das *dynamische Verhalten* erweitert werden, Ayoubi (1996), Nelles (1997), Isermann et al. (1997), Nelles (2001).

Künstliche neuronale Netze für das statische Verhalten

Neuronale Netze sind universelle Approximatoren und somit eine Alternative zu klassischen Polynomansätzen. Von Vorteil sind die geringe erforderliche Apriori-

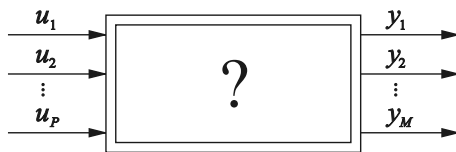


Bild 7.7. System mit P Eingängen und M Ausgängen, das durch ein künstliches neuronales Netz approximiert werden soll

Kenntnis über die Prozessstruktur und eine einheitliche Behandlung von Ein- und Mehrgrößenprozessen. Im Folgenden wird davon ausgegangen, dass ein nichtlineares System mit P Eingangsgrößen und M Ausgangsgrößen approximiert werden soll, Bild 7.7.

Neuronenmodell

Bild 7.8 zeigt das Signalflussbild eines Neurons. Im *Eingangsoperator* (Synaptische Funktion) wird zunächst ein Ähnlichkeitsmaß zwischen dem Eingangsvektor $\mathbf{u}$ und dem (gespeicherten) Gewichtsvektor $\mathbf{w}$ und daraus die skalare Zwischengröße x gebildet, z.B. durch das Skalarprodukt

$$x = \mathbf{w}^T \mathbf{u} = \sum_{i=1}^P w_i u_i = |\mathbf{w}^T| |\mathbf{u}| \cos \phi \quad (7.5.5)$$

oder durch den quadratischen Abstand

$$x = \|\mathbf{u} - \mathbf{w}\|^2 = \sum_{i=1}^P (u_i - w_i)^2. \quad (7.5.6)$$

Wenn $\mathbf{w}$ und $\mathbf{u}$ ähnlich sind, wird x im ersten Fall groß und im zweiten Fall klein.

Die Zwischengröße x wirkt nun auf die *Aktivierungsfunktion* und bildet die Ausgangsfunktion y , auch Aktivierung genannt

$$y = \gamma(x - c). \quad (7.5.7)$$

Bild 7.9 zeigt einige Beispiele dieser im allgemeinen nichtlinearen Funktionen. c ist eine Konstante, die eine Parallelverschiebung in x -Richtung bewirkt, auch Schwellwert genannt.

Netzstruktur

Die einzelnen Neuronen werden zu einem Netz verschaltet, Bild 7.10. Dabei unterscheidet man verschiedene Schichten mit parallel angeordneten Neuronen (layer): die Eingangsschicht (input layer), die erste, zweite, ..., Zwischenschicht oder versteckte Schicht (hidden layer) und die Ausgangsschicht (output layer). Die Eingangsschicht dient im Allgemeinen zur Normierung der Eingangssignale und wird oft nicht mitgezählt, so dass der Netzaufbau mit der ersten Zwischenschicht beginnt. Bild 7.10

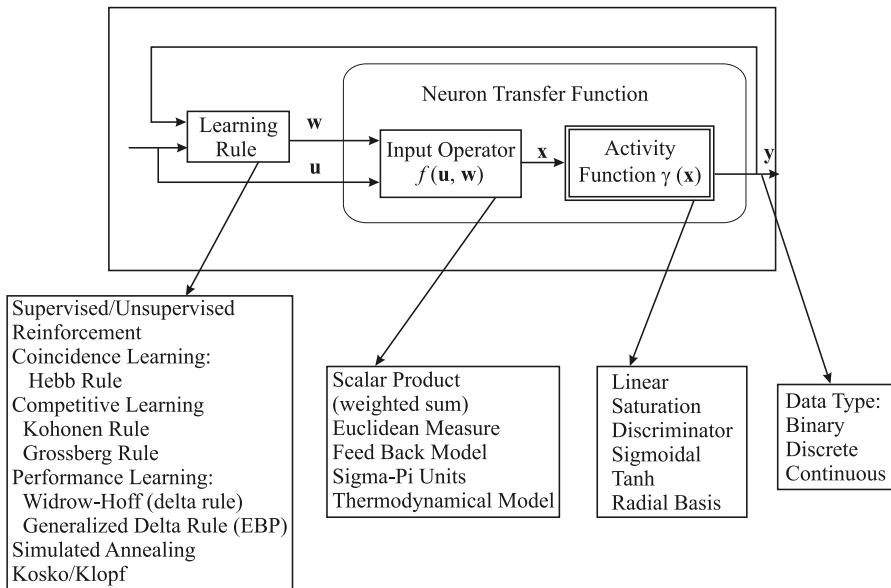


Bild 7.8. Allgemeines Neuronenmodell

zeigt die wichtigsten internen Verbindungen (links): vorwärts (feedforward), rückwärts (feedback), seitwärts (lateral), rückgekoppelt (recurrent). Die Eingangssignale können bezüglich ihres Wertebereiches binär, diskret oder kontinuierlich sein. Binäre und diskrete Signale werden z.B. zur Klassifikation, kontinuierliche Signale zur Identifikation verwendet.

Das Multi-Layer Perzeptron-Netz (MLP)

Aus dem allgemeinen Neuronenmodell nach Bild 7.8 folgt das Perzeptron genannte Neuron in Bild 7.11. Als Aktivierungsfunktion werden meist die Sigmoidalfunktion oder der Hyperbeltangens verwendet, also mehrfach differenzierbare nichtlineare Funktionen, die auch außerhalb ihres linearen Bereiches $y \neq 0$ ergeben und deshalb eine globale Wirkung mit Extrapolationsfähigkeit besitzen. Die Gewichte w_i sind im Eingangsoperator vorgesehen, also im Signalfluss vor der Aktivierungsfunktion.

Versieht man jede Schicht mit parallel angeordneten Perzeptronen und schaltet die Schichten hintereinander, dann erhält man ein vorwärtsgerichtetes mehrschichtiges Perzeptron-Netz, Bild 7.12. Dabei wirkt jedes der P Eingangssignale auf jedes Perzeptron, so dass also in der ersten Zwischenschicht mit K Perzeptronen ($K \times P$) Gewichte w_i vorhanden sind. Das Ausgangsneuron ist im Allgemeinen ein Perzeptron mit einer linearen Aktivierungsfunktion, Bild 7.13.

Die Adaption der Gewichte w_i aufgrund gemessener Ein- und Ausgangssignale erfolgt in der Regel über die Minimierung einer quadratischen Verlustfunktion

a) Hyperbeltangens (Tangens Hyperbolicus)

$$y = \frac{e^{(x-c)} - e^{-(x-c)}}{e^{(x-c)} + e^{-(x-c)}} = 1 - \frac{2}{1 + e^{2(c-1)}}$$

b) Sigmoidal Funktion

$$y = \frac{1}{1 + e^{-(x-c)}}$$

c) Begrenzer

$$y = \begin{cases} 1 & ; x - c \geq 1 \\ x - c & ; |x - c| < 1 \\ -1 & ; x - c \leq -1 \end{cases}$$

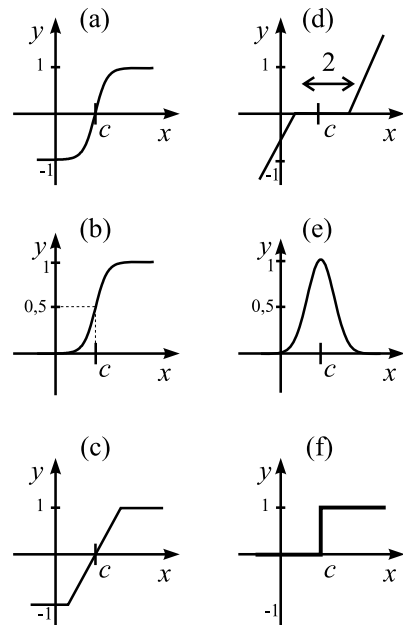
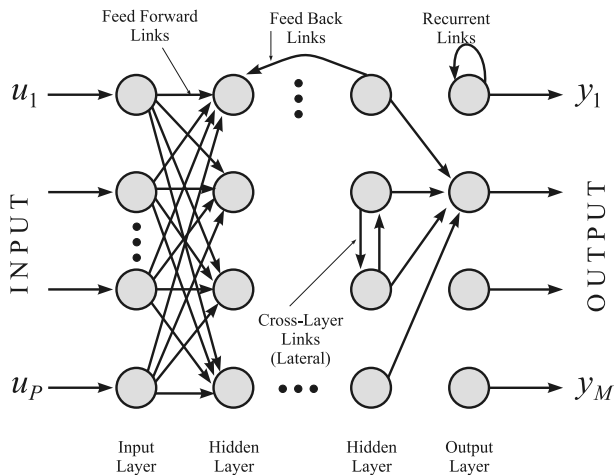
d) Tote-Zone

$$y = \begin{cases} 0 & ; |x - c| \leq 1 \\ x - c - 1 & ; x - c > 1 \\ x - c + 1 & ; x - c \leq -1 \end{cases}$$

e) Gaussfunktionen $y = e^{-(x-c)^2}$

f) Binäre Funktion

$$y = \begin{cases} 0 & ; x - c < 0 \\ 1 & ; x - c > 0 \end{cases}$$

**Bild 7.9.** Beispiele für Aktivierungsfunktionen**Bild 7.10.** Netzstruktur: Schichten und Verbindungen in einem neuronalen Netz

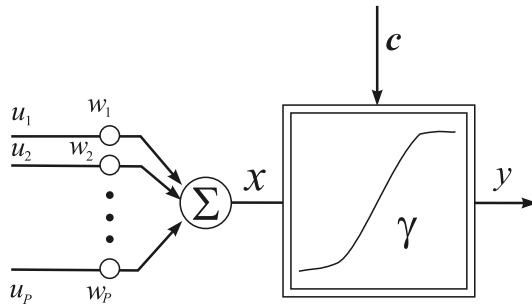


Bild 7.11. Perzeptron als Neuron mit Gewichtung w_i und Addition der Eingangssignale (Skalarprodukt) und nichtlinearer Aktivierungsfunktion

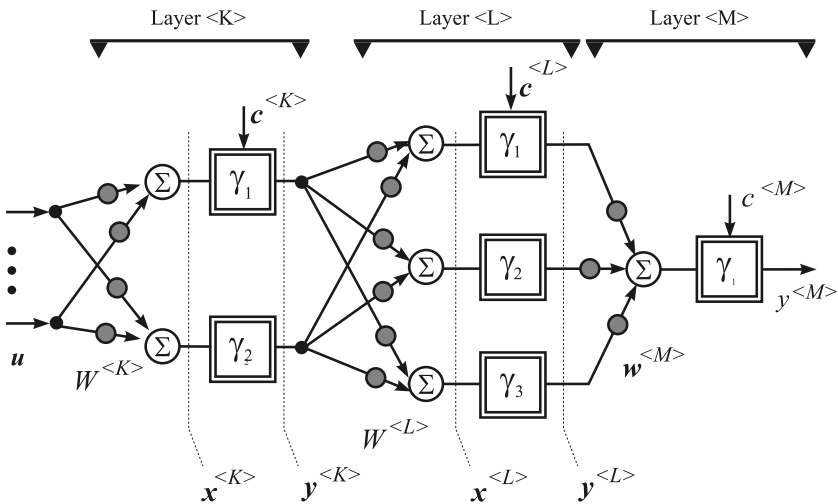


Bild 7.12. Vorwärtsgerichtetes mehrschichtiges Perzeptron-Netz (MLP-Netz). 3 Schichten mit $(2 \times 3 \times 1)$ Perzeptronen. $< K >$ ist die erste verdeckte Schicht

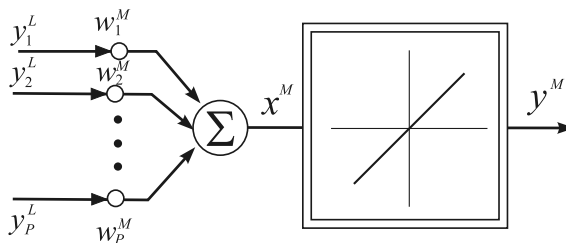


Bild 7.13. Ausgangsneuron als Perzeptron mit linearer Aktivierungsfunktion (Ausgangspzeptron)

$$J(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N-1} e^2(n) \quad (7.5.8)$$

$$e(n) = y(n) - \hat{y}(n),$$

wobei e der Ausgangsfehler, y das gemessene Ausgangssignal und $\hat{y}$ das Netzausgangssignal sind. Dabei wird wie bei der Methode der kleinsten Fehlerquadrate zur Parameterschätzung

$$\frac{dJ(\mathbf{w})}{d\mathbf{w}} = \mathbf{0} \quad (7.5.9)$$

gebildet. Wegen der nichtlinearen Abhängigkeit ist jedoch keine direkte Lösung möglich. Deshalb wird z.B. ein Gradientenverfahren zur numerischen Optimierung eingesetzt. Wegen des dabei erforderlichen Zurückrechnens des Ausgangsfehlers über mehrere Schichten wird dies „error back propagation“ genannt oder auch „Delta-Regel“. Die sogenannte Lernrate η muss dabei geeignet gewählt (probiert) werden. Gradientenverfahren erlauben aber bei vielen unbekannten Parametern grundsätzlich nur eine langsame Konvergenz und nicht immer reproduzierbare Ergebnisse.

Das Radial-Basisfunktions-Netz (RBF)

Beim Neuron des Radial-Netz-Basisfunktions-Netzes, Bild 7.14, wird im Eingangsoperator ein quadratisches Abstandsmaß

$$x = \|\mathbf{u} - \mathbf{c}\|^2 \quad (7.5.10)$$

gebildet und der Aktivierungsfunktion

$$G_m = \gamma_m(\|\mathbf{u} - \mathbf{c}\|^2) \quad (7.5.11)$$

zugeführt. Diese Aktivierungsfunktion setzt sich aus radialen Basisfunktionen in Form von meist Gauß-Funktionen zusammen, so dass

$$\gamma_m = \exp \left[-\frac{1}{2} \left(\frac{(u_1 - c_{m1})^2}{\sigma_{m1}^2} + \frac{(u_2 - c_{m2})^2}{\sigma_{m2}^2} + \dots + \frac{(u_P - c_{mP})^2}{\sigma_{mP}^2} \right) \right]. \quad (7.5.12)$$

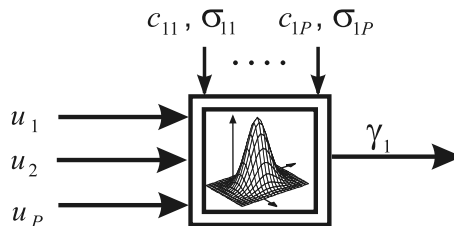


Bild 7.14. Neuron mit Radial-Basisfunktion

Die Zentren c_j und Standardabweichungen σ_j werden a priori festgelegt, so dass sich die Gauß-Glocken z.B. gleichförmig im Anregungsgebiet $\mathbf{u}$ verteilen. Die Aktivierungsfunktion bestimmt dabei die Abstände jedes Eingangssignals zum Zentrum der zugehörigen Basisfunktion. Diese Radial-Basisfunktionen wirken jedoch nur lokal um ihre Zentren und besitzen daher wenig Extrapolationsfähigkeit, da ihre Ausgänge bei größeren Abständen Null werden.

Radial-Basisfunktions-Netze bestehen meist aus nur einer Schicht, Bild 7.15. Die Ausgänge γ_i werden in einem Ausgangsneuron vom Perzeptrontyp, Bild 7.13, gewichtet und summiert, so dass

$$y = \sum_{m=1}^M w_m \gamma_m (||\mathbf{u} - \mathbf{c}||^2). \quad (7.5.13)$$

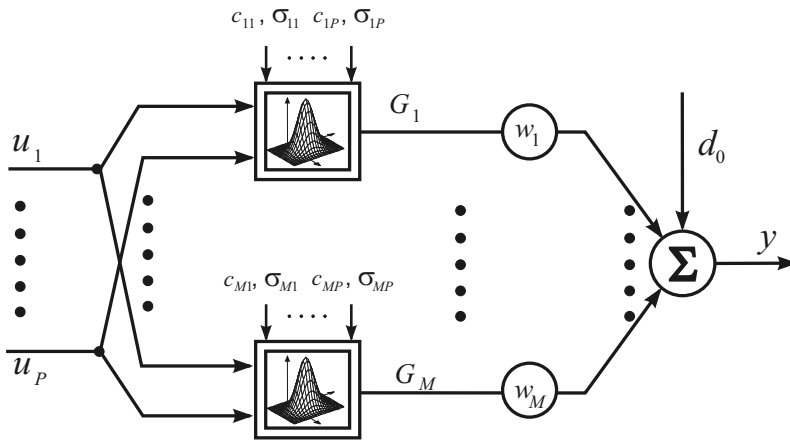


Bild 7.15. Vorwärtsgerichtetes einschichtiges Radial-Basis-Funktionsnetz

Da die Gewichte am Ausgang der Schicht also im Signalfuss nach der nicht-linearen Aktivierungsfunktion liegen, ist das Fehlersignal linear in den Parametern und es lässt sich die Parameterschätzmethode der kleinsten Quadrate in der expliziten Form anwenden. Dadurch wird eine wesentlich schnellere Konvergenz im Vergleich zu den MLP-Netzen mit Gradientenverfahren möglich. Wenn allerdings die Zentren und Standardabweichungen auch optimiert werden, muss wiederum ein nichtlineares numerisches Optimierungsverfahren angewandt werden.

Lokal lineare Netzwerkmodelle

Das lokal lineare Netzwerkmodell ist ein erweitertes Radial-Basisfunktions-Netz Nelles (1997), Nelles (2001). Die Gewichte der Ausgangsschicht werden durch eine lineare Funktion der Netzwerkeingänge ersetzt (7.5.14). Des Weiteren wird das RBF Netzwerk normiert, so dass die Summe aller Basisfunktionen 1 ist. Jedes Neu-

ron enthält also ein lokal lineares Modell mit seinen entsprechenden Validierungsfunktionen, siehe Bild 7.16. Die Validierungsfunktionen bestimmen die Gebiete des Eingangsraumes, in dem jedes Neuron aktiv ist. Die allgemeine Struktur der lokalen Netzwerkmodelle wird ausführlich in Murray-Smith und Johansen (1997) diskutiert.

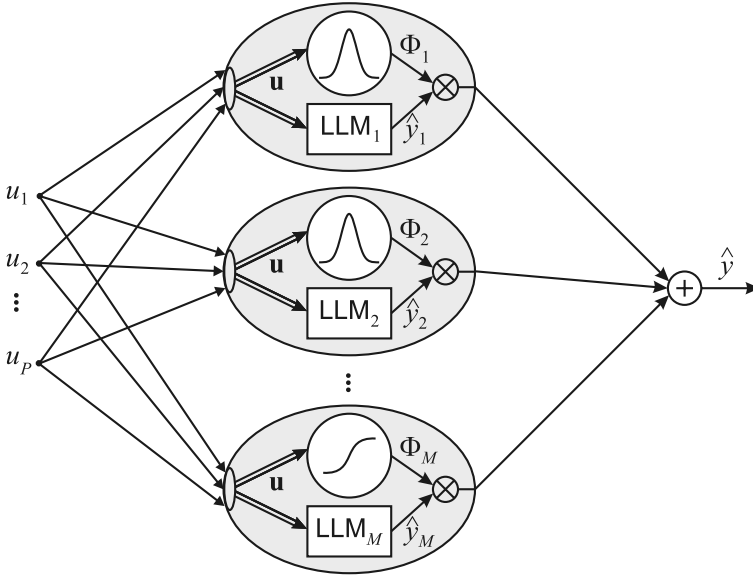


Bild 7.16. Lokal lineares Netzwerkmodell (LOLIMOT)

Das betrachtete lokale Netzwerkmodell verwendet normierte Gauß-Funktionen (7.5.15) und eine achsenorthonogale Partitionierung des Eingangsraumes. Deswegen können die Validierungsfunktionen aus eindimensionalen Zugehörigkeitsfunktionen bestehen und das Netzwerk kann als Takagi-Sugeno Fuzzy Modell interpretiert werden, siehe Abschnitt 7.5.3.

Der Ausgang des lokal linearen Modells ist

$$\hat{y} = \sum_{i=1}^M \Phi_i(\mathbf{u}) (w_{i,p} + w_{i,1}u_1 + \cdots + w_{i,p}u_p) \quad (7.5.14)$$

und mit der normierten Validierungs-Gauß-Funktion

$$\Phi_i(\mathbf{u}) = \frac{\mu_i(\mathbf{u})}{\sum_{j=1}^M \mu_j(\mathbf{u})} \quad (7.5.15)$$

gilt hier

$$\mu_i(\mathbf{u}) = \sum_{j=1}^p \exp \left(-\frac{1}{2} \left(\frac{(u_j - c_{i,j})^2}{\sigma_{i,j}^2} \right) \right). \quad (7.5.16)$$

Die Zentren $\mathbf{c}$ und Standardabweichungen σ sind Parameter, die nichtlinear wirken, während die lokalen Modellparameter w_i linear auf den Ausgang wirken. Der lokal lineare Modellbaum-Algorithmus (LOLIMOT) wird für das Trainieren verwendet. Er besteht aus einem äußeren Kreis, in welchem der Eingangsraum unterteilt wird, indem die Parameter der Validierungsfunktionen bestimmt werden, und einem inneren Kreis, in dem die Parameter der lokal linearen Modelle durch die lokal gewichtete Schätzung mit der Methode der kleinsten Quadrate optimiert werden.

Der Eingangsraum wird somit in einer achsenorthogonalen Weise unterteilt und man erhält Hyper-Rechtecke, in deren Zentren die Gauß-Funktionen $\mu_i(u)$ positioniert werden. Die Standardabweichungen dieser Gauß-Funktionen werden proportional zur Größe der Hyper-Rechtecke gewählt, um die veränderliche Granularität zu repräsentieren. Die nichtlinear eingehenden Parameter $c_{i,j}$ und $\sigma_{i,j}$ werden durch eine nichtlineare Optimierung bestimmt. LOLIMOT startet mit einem einzelnen linearen Modell, welches für den gesamten Eingangsraum gültig ist. Bei jeder Iteration spaltet sich ein lokal lineares Modell in zwei neue Untermodelle auf. Nur das lokale Modell, welches (lokal) die geringste Güte erzielt, wird für die weitere Verfeinerung verwendet. Die Spaltung entlang aller Eingangsachsen wird verglichen und die Alternative mit der bestens Güte wird realisiert, siehe Bild 7.17.

Die großen Vorteile dieses Ansatzes mit lokalen Modellen sind die integrierte Strukturidentifikation und der sehr schnelle und robuste Trainingsalgorithmus. Die Modellstruktur wird so der Komplexität des Prozesses angepasst. Jedoch kann die explizite Anwendung der zeitaufwändigen Optimierungsalgorithmen auch vermieden werden, wenn die Struktur einmal gefunden wurde.

Eine andere lokal lineare Modellstruktur ergeben die so genannten *Hinging Hyperplane Modelle*, beschrieben in Töpfer (Ernst) (1998), Töpfer (2002). Diese Modelle können als Erweiterung des LOLIMOT Netzwerkes mit Bezug auf die Partitionierung interpretiert werden. Während der LOLIMOT-Algorithmus auf die achsenorthogonale Spaltung beschränkt ist, erlauben die Hinging Hyperplane Modelle einen achsenschrägen Unterteilung des Eingangsraumes. Diese komplexere Partitionierungsstrategie führt jedoch zu einem zunehmenden Aufwand für den Modellaufbau. Trotzdem ist dies bei Auftreten von stark nichtlinearem Modellverhalten und höher-dimensionalen Eingangsräumen notwendig.

In diesem Abschnitt wurden die grundlegenden Strukturen von drei künstlichen neuronalen Netzen beschrieben. Diese Modelle sind gut für die Approximation von gemessenen Ein-/Ausgangsdaten eines statischen Prozesses geeignet, vgl. auch Hafner et al. (1992), Preuß und Tresp (1994). Hierfür müssen die Trainingsdaten so gewählt werden, dass der betrachtete Eingangsraum möglichst gleichmäßig mit Daten bedeckt ist. Nach dem Training steht ein parametrisches mathematisches Modell des statischen Prozessverhaltens zur Verfügung. Entsprechend ist eine direkte Berechnung der Ausgangswerte $\hat{y}$ für beliebige Eingangskombinationen $\mathbf{u}$ möglich.

Ein Vorteil der automatischen Trainingsverfahren ist die Möglichkeit, beliebig verteilte Daten im Trainings-Datensatz zu verwenden. Es ist nicht notwendig die Daten an genau definierten Positionen zu messen, wie es z.B. bei Rasterkennfeldern nötig ist, siehe Abschnitt 7.5.4. Dies reduziert den Aufwand bei der praktischen Anwendung.

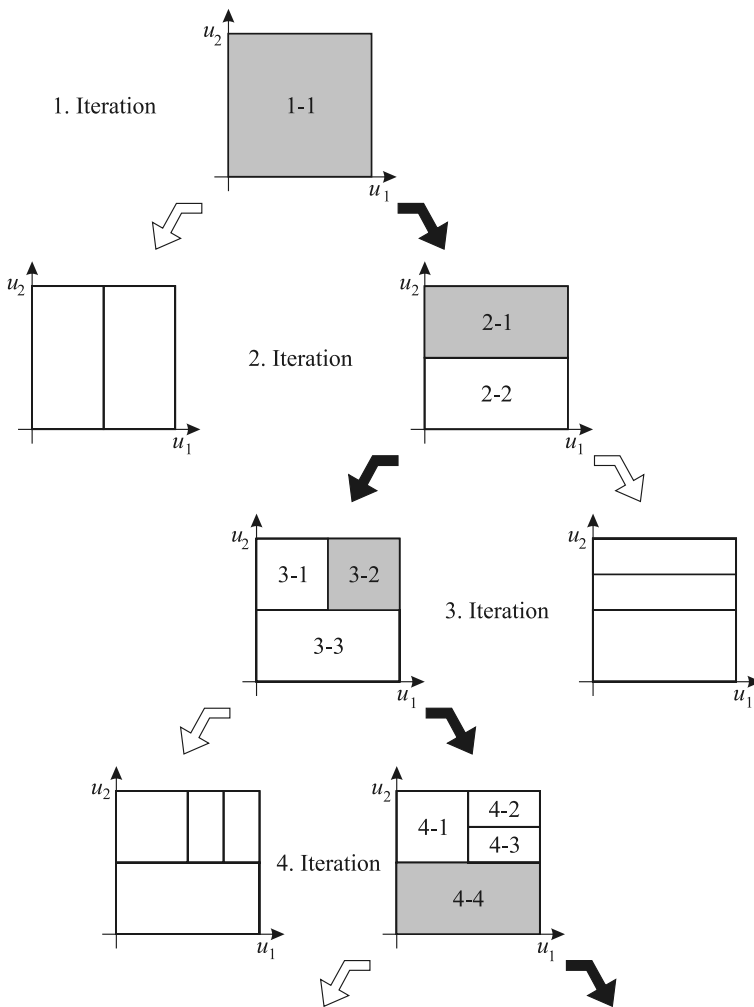


Bild 7.17. Baumstruktur des LOLIMOT-Algorithmus

Beispiel 7.1: Künstliche Neuronale Netze für das statische Verhalten eines Verbrennungsmotors

Als Beispiel wird das stationäre Motorverhalten eines 6-Zylinder Ottomotors betrachtet. Hierbei muss das Motordrehmoment abhängig vom Drosselklappenwinkel und der Motordrehzahl gesteuert werden. Bild 7.18 zeigt die 433 vorhandenen Datenpunkte, die am Motorprüfstand gemessen wurden.

Für die Approximation wurde ein MLP Netzwerk eingesetzt. Nach dem Training folgt die Approximation der gemessenen Daten, wie in Bild 7.19 gezeigt. Hierfür werden 31 Parameter benötigt. Offensichtlich zeigt das neuronale Netz gute Interpolations- und Extrapolations-Fähigkeiten. Dies bedeutet, dass auch Teilbereiche mit

nur wenigen Trainingsdaten recht gut approximiert werden können, Holzmann et al. (1997).

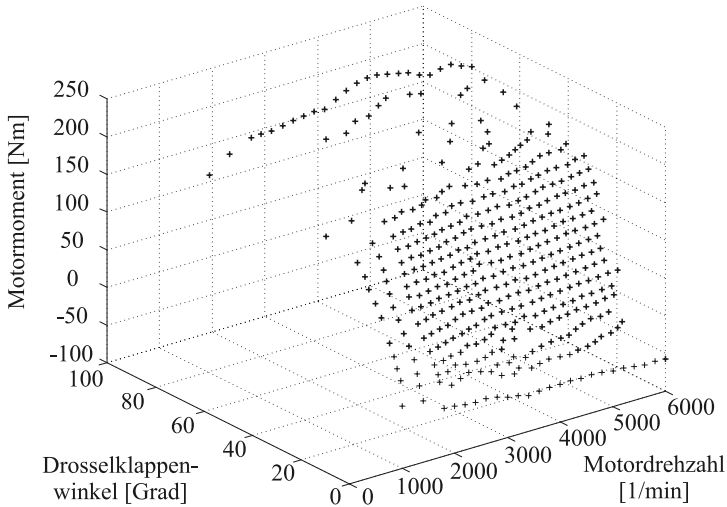


Bild 7.18. Gemessene Motorkennfelddaten eines Benzinmotors (2,5 l, V6): ungleichförmig verteilt, 433 Stützstellen

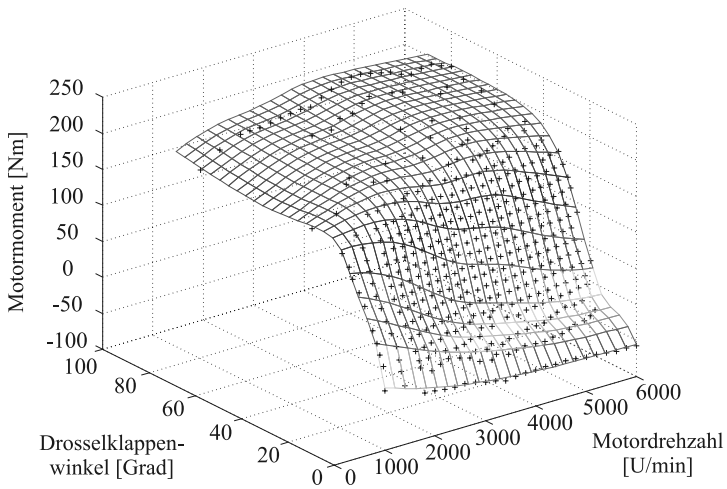


Bild 7.19. Approximation der Motorkennfelddaten von Bild 7.18 mit einem MLP-Netz ($2 \times 6 \times 1$): 31 Parameter

b) Künstliche neuronale Netze für das dynamische Verhalten

Dynamische neuronale Netze gehen aus der Erweiterung der statischen Netze hervor. Dabei kann man neuronale Netze mit externer oder interner Dynamik unterscheiden, Nelles et al. (1997), Isermann et al. (1997). KNN mit *externer Dynamik* haben statische MLP- oder RBF-Netze als Basis und schalten das gemessene zeitdiskrete Eingangssignal $u(k)$ über Filter $F_i(q^{-1})$ und das gemessene Ausgangssignal $y(k)$ oder das NN-Ausgangssignal $\hat{y}(k)$ über Filter $G_i(q^{-1})$ dem NN auf. q^{-1} ist hierbei ein Zeitverschiebeoperator

$$y(k)q^{-1} = y(k-1). \quad (7.5.17)$$

Im einfachsten Fall sind die Filter reine Verzögerungsketten, Bild 7.20a),

$$\hat{y}(k) = f_{NN}[u(k), u(k-1), \dots, \hat{y}(k-1), \hat{y}(k-2), \dots], \quad (7.5.18)$$

wobei die zeitverschobenen Abtastwerte die Eingangssignale des NN bilden. Die Struktur in Bild 7.20a) bildet ein zum Prozess *paralleles Modell* (entspricht dem Ausgangsfehlermodell bei der Parameterschätzung). In Bild 7.20b) wird das gemessene Ausgangssignal auf den Netzeingang geschaltet. So entsteht ein *seriell-paralleles Modell* (entspricht dem Gleichungsfehlermodell bei der Parameterschätzung). Von Vorteil ist bei diesen Strukturen, dass im Prinzip dieselben Methoden zur Adaption verwendet werden können wie bei den statischen KNN. Nachteilig sind allerdings die starke Vergrößerung des Eingangsraumes, eventuelle Stabilitätsprobleme und das nur iterativ berechenbare statische Verhalten.

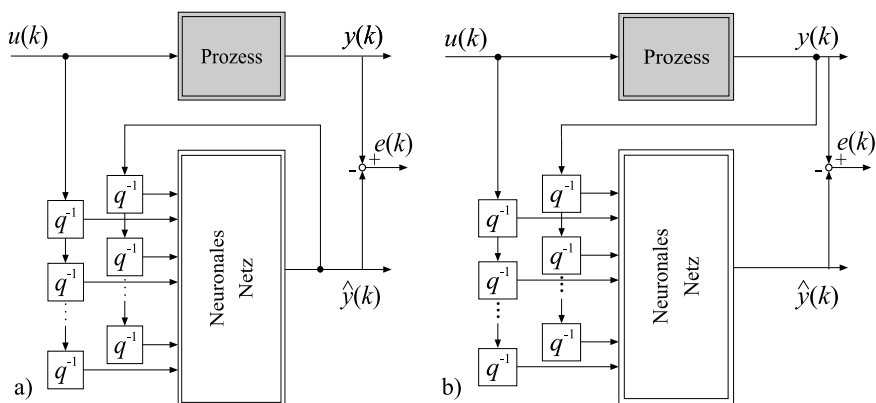


Bild 7.20. Künstliches Neuronales Netz mit externer Dynamik: a) Paralleles Modell; b) Seriell-paralleles Modell

Bei den KNN mit *interner Dynamik* werden die dynamischen Elemente im Inneren der Netzstruktur realisiert. Man unterscheidet z.B. voll vernetzte Strukturen, partiell rekurrente Netze und lokal rekurrente global feedforward Netze (LRGF), Nelles et al. (1997), Nelles (2001). Bei den letzteren wird die Struktur der statischen NN

beibehalten und es wird das einzelne Neuron dynamisiert, siehe Bild 7.21. Dabei kann man unterscheiden: Synapsendynamik, Aktivierungsdynamik oder Rückführungsdynamik. Am Einfachsten ist die Aktivierungsdynamik nach Ayoubi (1996). Hier wird jedes Perzeptron mit einem linearen Übertragungsglied zweiter Ordnung versehen, Bild 7.22. Die dynamischen Parameter a_i und b_i werden adaptiert. Statisches und dynamisches Verhalten lassen sich einfach trennen und die Stabilität stellt kein Problem dar.

Üblicherweise werden für die LRGF-Strukturen mehrschichtige Perzeptren eingesetzt. Es können aber auch RBF Strukturen mit Dynamik am Ausgang eingesetzt werden, wenn man Hammerstein-Strukturen erwartet, Ayoubi (1996).

Die Adaption dieser dynamischen NN erfolgt meist über entsprechend erweiterte Gradientenverfahren, Nelles et al. (1997), Nelles (2001).

Aufgrund der Grundstrukturen der KNN lassen sich spezielle Strukturen mit besonderen Eigenschaften zusammensetzen. Erweitert man z.B. bei einem RBF-Netzwerk die Ausgangsgewichte durch lineare Funktionen der Netzeingangsgrößen, dann stellt jedes Neuron ein lokales lineares Modell dar. Damit lässt sich eine Baumstruktur (LOLIMOT: local linear model tree) aufbauen, Nelles (1997). Mit Hilfe einer iterativen Unterteilung des Eingangsraumes und einer Parameterschätzung mit Linearität in den Parametern entsteht so ein KNN mit schneller Konvergenz.

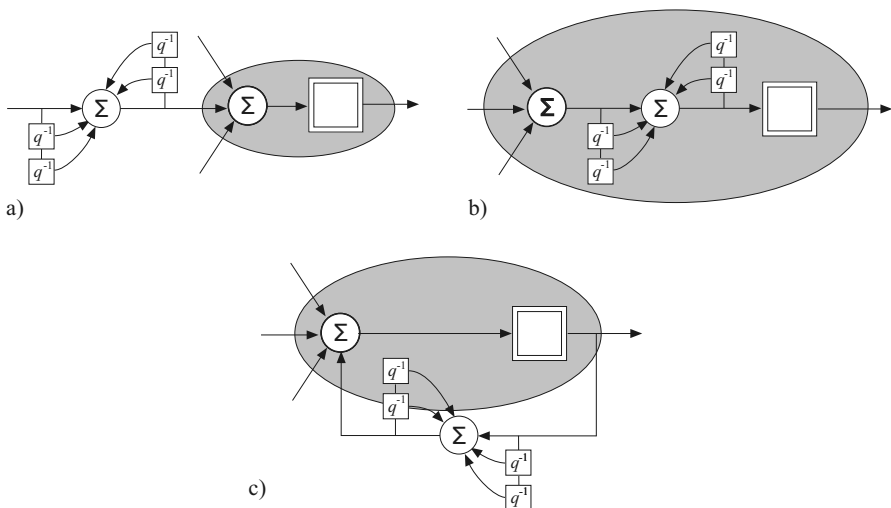


Bild 7.21. Dynamische Neuronen für neuronale Netze mit interner Dynamik: a) Lokale Synapsendynamik; b) Lokale Aktivierungsdynamik; c) Lokale Rückführungsdynamik

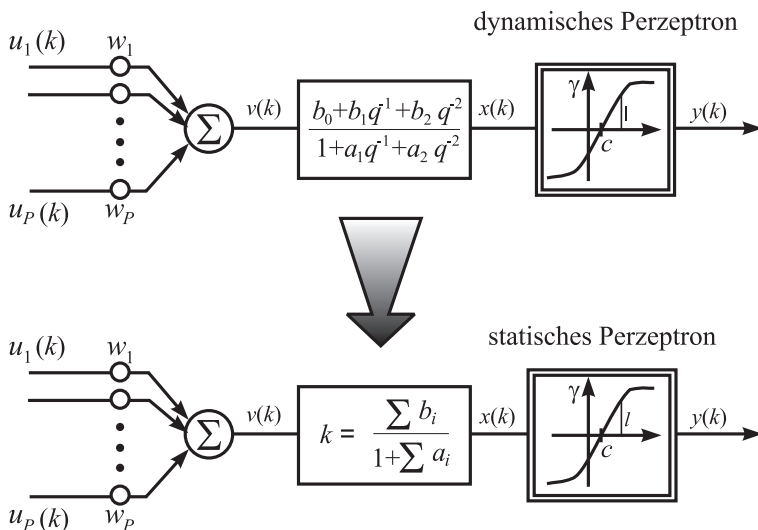


Bild 7.22. Dynamisches Perzeptron nach Ayoubi (1996)

Beispiel 7.2: Künstliche neuronale Netze für die Identifikation des dynamischen Verhaltens der Emissionen von Verbrennungsmotoren

Das folgende Beispiel zeigt die Anwendung und Ausführung eines dynamisch neuronalen Netzwerkes zur Identifikation der Abgase eines Verbrennungsmotors. Genaue Abgasmodelle können z.B. beim Regelungsentwurf, der modellbasierten Optimierung des Motors, modellbasierten Fehlererkennung und -diagnose etc. verwendet werden. Leider ist es kaum möglich die Emissions-Modelle durch theoretische Modellbildung alleine zu erhalten, da die Verbrennung und die Abgaszusammensetzung komplexen thermodynamischen und chemischen Reaktionen unterliegen, die durch lokale Bedingungen und Störungen beeinflusst werden. Neuronale Netze eignen sich jedoch gut für die experimentelle Modellierung bzw. Identifikation des Abgasverhaltens basierend auf gemessenen Ein-/Ausgangsdaten, Hafner et al. (2000).

Es wird nun ein dynamisches NOx Modell eines 1,9 L TDI Dieselmotors identifiziert. Die Daten wurden bei einem bestimmten Betriebspunkt mit der Motordrehzahl von 1000 U/min und einem Motordrehmoment von 130 Nm gespeichert. Die Stellgrößen für die Regelung des Motors wurden als Eingänge für das Netze gewählt, d.h. die Position des Abgasrückführventils (EGR), Winkel des Einspritzbeginns (Θ_{inj}) und Position der Leitschaufel des Turboladers mit variabler Geometrie (VGT). Entsprechend Bild 7.23 wurde das Modell mit einem internen dynamischen Multi-Layer Perzeptron-Netz (DMLP) mit fünf versteckten Neuronen und einem Ausgangsneuron realisiert. Die Dynamik des Prozesses wurde durch die gewählten Filter-Übertragungsfunktionen realisiert. In diesem Fall führten die Polynome erster Ordnung A_i und B_i zu recht befriedigenden Ergebnissen.

Die Messungen selbst müssen gut ausgelegt sein, um den Prozess mit den relevanten Amplituden und Frequenzen anzuregen. Der resultierende Signalverlauf des

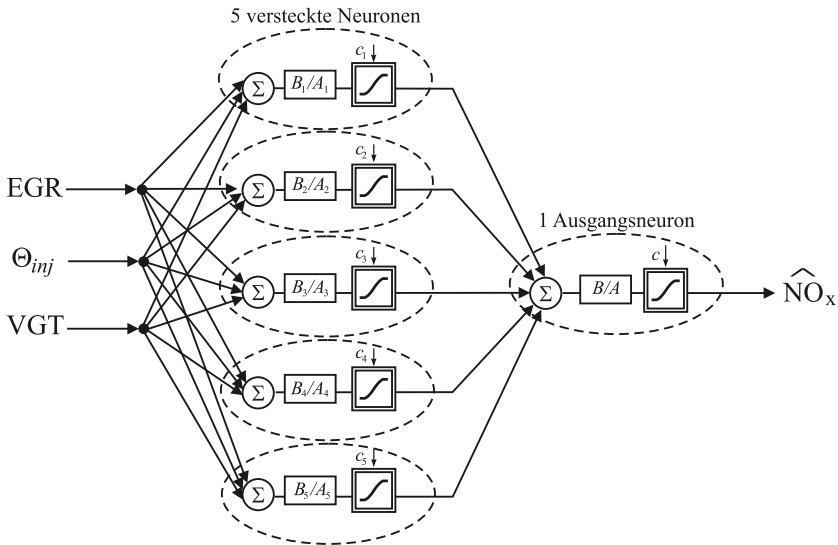


Bild 7.23. Struktur eines dynamischen künstlichen neuronalen Netzes zur Modellbildung der NO_x-Emissionen mit den Eingängen EGR (Abgasrückführventil), θ_{inj} (Einspritzbeginn) und VGT (Position des stellbaren Turboladers)

Experimentes und die gemessenen Trainingsdaten sind in Bild 7.24 dargestellt. Entsprechend der dominierenden Zeitkonstanten des Prozesses, welche 1-3 Sekunden bezüglich EGR und VTG betragen, wurde die Abtastzeit mit $T_0 = 100$ ms gewählt.

Die Trainingsdaten (ca. 9000 gemessene Punkte für Ein- und Ausgänge der drei Modelle) werden nun für die Schätzung der Modellparameter eingesetzt. Mit dem DMLP-Trainings-Algorithmus benötigt man 45 Minuten um ein dynamisches Modell mit einem quadratischen Mittelwert-Fehler von 0,08 zu erzeugen. Bild 7.25a) illustriert die Güte des Modells, wenn es zur Generalisierung auf nicht zum Training verwendete Daten angewandt wird. Der Vergleich zwischen dem gemessenen und simulierten Verhalten beweist die Qualität des identifizierten Netzwerkes.

Ein zusätzlicher Vorteil der dynamischen Modellierung ist die Möglichkeit, das statische nichtlineare Verhalten des Prozesses aus dem dynamischen Modell zu berechnen, Bild 7.25b). Wie erwartet, steigen die NO_x Emissionen mit früher Einspritzung an (kleinere Zahlen in Θ_{inj} und EGR).

□

7.5.3 Fuzzy-logische Modelle

Eine weitere Möglichkeit der Kennfelddarstellung ergibt sich über Fuzzy-Logik Methoden, siehe z.B. Bothe (1993), Kiendl (1996), Preuß (1992), Zadeh (1965). Zur Veranschaulichung zeigt Bild 7.26 die Grundstruktur eines Fuzzy-Logik-Systems.

Bei der *Fuzzifizierung* werden die deterministischen Eingangswerte e_1 und e_2 unter Verwendung von Zugehörigkeitsfunktionen in „unscharfe“ linguistische Werte

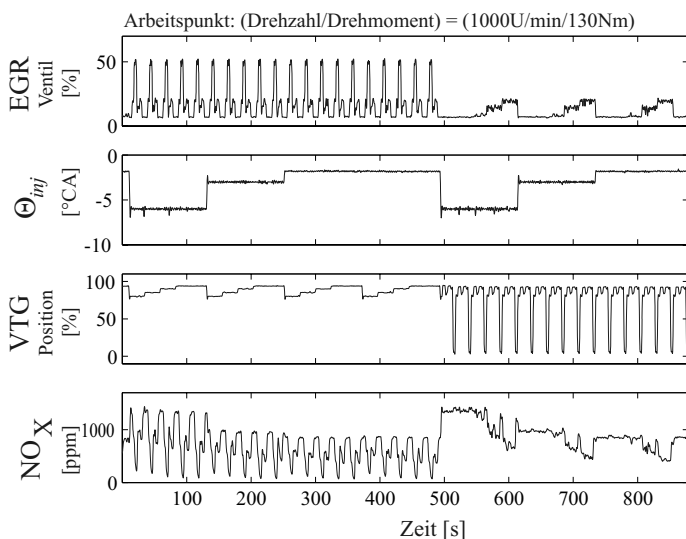


Bild 7.24. Dynamisch gemessene Signale zum Training des KNN. Die drei Stellsignale regen den Prozess mit verschiedenen Amplituden und Frequenzen an (1,9 L TDI Dieselmotor mit einer Drehzahl von 1000 U/min und einem Drehmoment von 130 Nm)

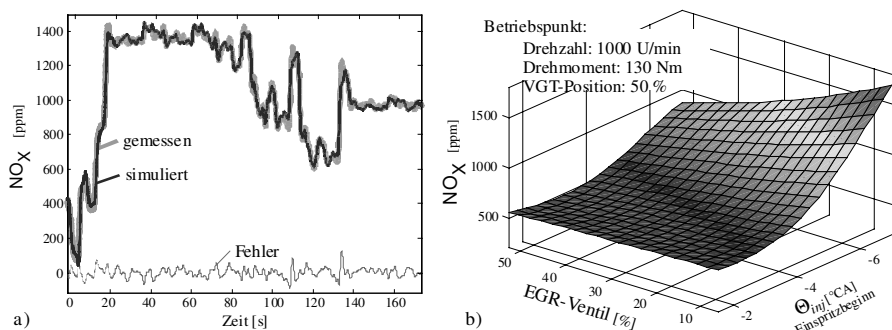


Bild 7.25. Identifikationsergebnisse: a) Generalisierung des dynamischen NOx-Modells; b) Berechnung des statischen Prozessverhaltens aus dem dynamischen neuronalen Netze

umgewandelt, die dann in *Fuzzy-Wenn-Dann-Regeln* verarbeitet werden. Am Ausgang der Regeln stehen wiederum Werte in unscharfer Form zur Verfügung, die zusammengefasst werden (Akkumulation) und durch die *Defuzzifizierung* in deterministische Ausgangswerte (hier u) umgewandelt werden. Durch die linguistische Beschreibung der unscharfen Werte wird eine Systembeschreibung durch Fuzzy-Logik mit einer überschaubaren Anzahl von Regeln transparent und gezielt modifizierbar.

Um einen Satz von Fuzzy-Regeln zu finden, der ein spezielles Ein-/Ausgangsverhalten eines Systems nachbildet, werden sogenannte Fuzzy-Identifikationsverfahren eingesetzt. Verbreitete Verfahren dieser Art basieren z.B. auf konventionellen Para-

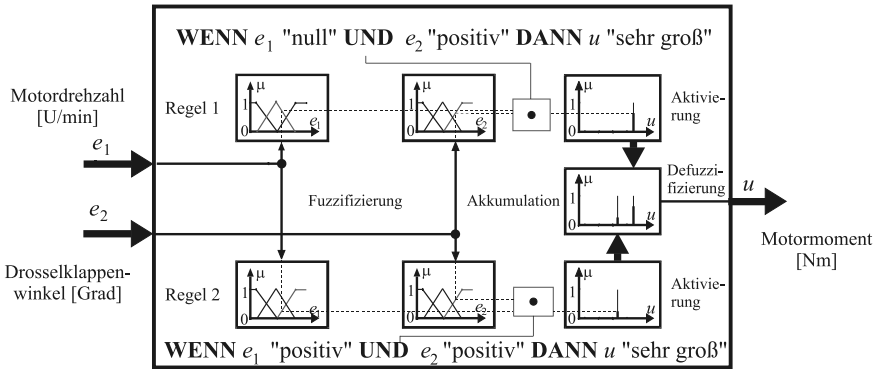


Bild 7.26. Grundstruktur eines Fuzzy-Logik-Systems

meterschätzverfahren, Pfeiffer (1995) oder der Kombination von Fuzzy-Logik und Neuronalen Netzen, Kosko (1992), Preuß und Tresp (1994). Durch derartige Neuro-Fuzzy Verfahren ist es möglich, die Transparenz einer Fuzzy-Darstellung mit dem Adaptionsvermögen Neuronaler Netze zu kombinieren. Um einen Eindruck von der Qualität einer Fuzzy-Systembeschreibung zu vermitteln, zeigt Bild 7.27 die Darstellung des als Anwendungsbeispiel betrachteten Motorkennfeldes. Analog zu Bild 7.19 stellen die schwarzen Kreuze die auf einem Motorenprüfstand aufgenommenen Messdaten dar.

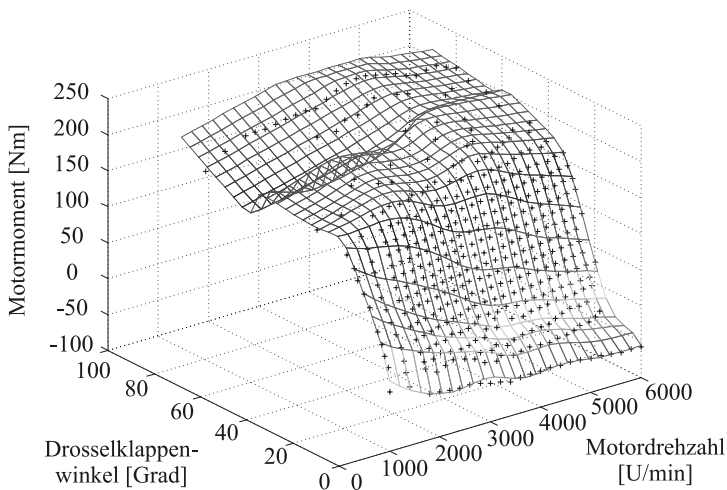


Bild 7.27. Fuzzy-Logik-basierte Kennfelddarstellung (12 Fuzzy Wenn-Dann-Regeln)

Die Güte der Darstellung hängt entscheidend von der Zahl der im Fuzzy-System verwendeten Zugehörigkeitsfunktionen und Regeln ab. Wird ein relativ ein-

faches Fuzzy-System gewählt, tauchen Probleme insbesondere bei der Darstellung komplizierter, stark nichtlinearer Kennfelder auf. Wählt man eine zu komplexe Beschreibung, wird das dargestellte Kennfeld zu wellig, d.h. der oft gewünschte leichte Glättungscharakter der Messdaten geht verloren. Bei der in Bild 7.27 dargestellten Kennfeldbeschreibung wurden jeweils 12 Zugehörigkeitsfunktionen pro Eingang sowie 12 Fuzzy-Regeln verwendet. Zur Identifikation des Fuzzy-Systems wurde ein Neuro-Fuzzy Verfahren eingesetzt, Jang (1993).

Beim direkten Vergleich mit der Darstellung durch ein Neuronales Netz (Bild 7.19) zeigen sich etwas schlechtere Ergebnisse des Fuzzy-Identifikationsverfahrens, hauptsächlich in Kennfeldbereichen, in denen nur wenig Messdaten zur Verfügung stehen. Die Rechenzeit hinsichtlich der Generierung des Kennfeldes aus Messdaten liegt sowohl bei den Neuro- als auch bei den Neuro-Fuzzy Verfahren in einer ähnlichen Größenordnung.

7.5.4 Kennfeld-Darstellungen für nichtlineares statisches Verhalten

Zur besonders anschaulichen und flexiblen Darstellung nichtlinearer statischer Zusammenhänge werden Kennfelder besonders für nichtlineare Steuerungen und Regelungen angewandt. In einem Motorsteuergerät sind beispielsweise bis zu 100 Kennfelder für das Motor- und Abgasmanagement implementiert, Bosch (1995).

Wegen der aus Kostengründen beschränkten Rechenzeit und des beschränkten Speicherplatzes ist es in Kfz-Steuergeräten üblich, derartige Kennfelder als Stützstellen eines Rasters (Gitternetz) abzuspeichern und unter Verwendung von Flächeninterpolationsverfahren auszuwerten.

a) Rasterkennfelder

Kennfeldmodelle bestehen aus einem Satz von Datenpunkten, Knoten oder Stützstellen, die in einem mehrdimensionalen Gitter positioniert sind, z.B. als Funktion von X und Y . Jede Stützstelle besteht aus zwei Komponenten. Die skalaren Datenpunkt-Höhen Z approximieren eine nichtlineare Funktion an ihrer Position. Die Stützstellen sind auf Gitterlinien an den Kreuzungspunkten z.B. in einem ROM des Steuergerätes gespeichert und im Voraus festgelegt, siehe Bild 7.28. Üblicherweise werden die Datenhöhen direkt an den Gitterpunkten experimentell ermittelt. Dann kann eine Optimierung des Modells vermieden werden.

Im Folgenden wird der häufigste realisierte zweidimensionale Fall betrachtet. Die Berechnung des Ausgangswertes Z für gegebene Eingangswerte X und Y zwischen den Gitterpunkten besteht aus zwei Schritten. Zunächst werden die Indizes der vier umliegenden Gitterstützstellen ermittelt. Daraufhin wird eine lineare Flächeninterpolation durchgeführt, Schmitt (1995).

Zur Ermittlung des gesuchten Wertes Z werden die vier ermittelten Stützstellen $Z(k, l)$ jeweils mit der gegenüberliegenden Teilfläche gewichtet und aufsummiert, 7.29. Das Ergebnis der Summation muss dann noch durch die Gesamtfläche des Rasterrechtecks geteilt werden, Schmitt (1995), Töpfer (2002). Somit folgt die Interpolationsgleichung

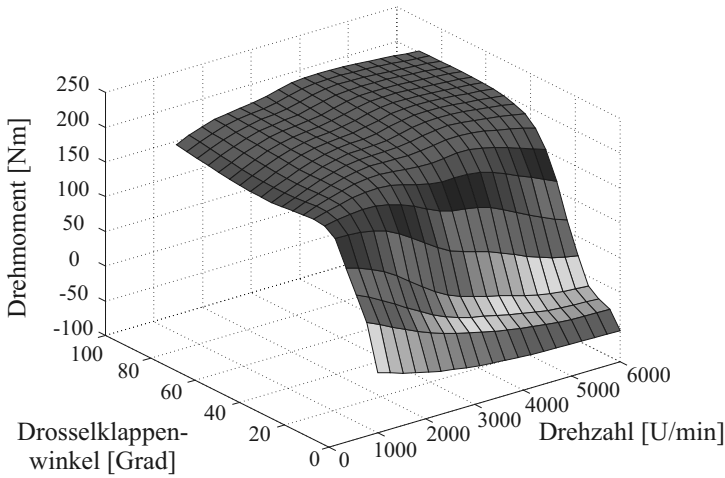


Bild 7.28. Rasterkennfeld eines 2.5l - V6 Ottomotors, gewonnen durch Messungen an allen 400 gleichmäßig verteilten Stützstellen

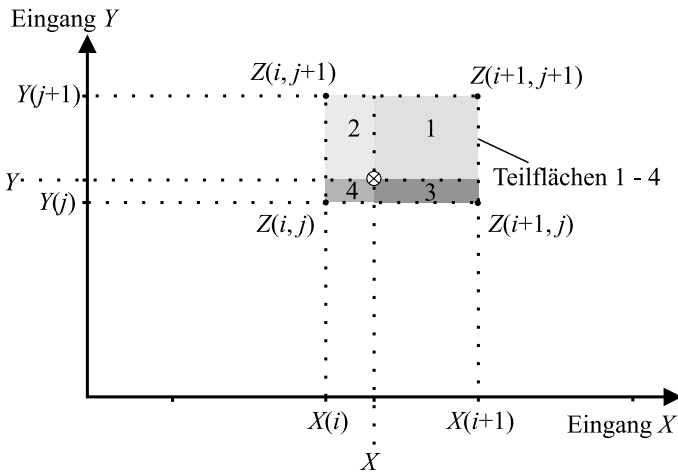


Bild 7.29. Darstellung der zur Interpolation in 3D-Kennfeldern benötigten Teilflächen

$$\begin{aligned}
 Z(X, Y) = & \underbrace{[Z(i, j) (X(i+1) - X)(Y(j+1) - Y)]}_{\text{Fläche 1}} \\
 & + \underbrace{Z(i+1, j) (X - X(i))(Y(j+1) - Y)}_{\text{Fläche 2}} \\
 & + \underbrace{Z(i, j+1) (X(i+1) - X)(Y - Y(j))}_{\text{Fläche 3}} \\
 & + \underbrace{Z(i+1, j+1) \cdot (X - X(i))(Y - Y(j))}_{\text{Fläche 4}} \\
 & \div \underbrace{[(X(i+1) - X(i))(Y(j+1) - Y(j))]}_{\text{Gesamtfläche}}.
 \end{aligned} \tag{7.5.19}$$

Aufgrund des relativ einfachen Berechnungsalgorithmus wird dieses Flächeninterpolationsverfahren insbesondere bei Echtzeitanwendungen eingesetzt. Die Genauigkeit des Verfahrens hängt stark von der Anzahl der verwendeten Gitterstützstellen ab. Zur Abbildung relativ „glatter“ Kennfelder reicht eine geringe Anzahl von Stützstellen aus, während bei ausgeprägtem nichtlinearen Verhalten das Gitter entsprechend engmaschiger gewählt werden muss. Eine Grundvoraussetzung für den Einsatz des Flächeninterpolationsverfahrens ist, dass die Stützstellenwerte im gesamten Arbeitsbereich bekannt sein müssen. Diese Bedingung ist allerdings in vielen Fällen nicht erfüllt. Außerdem ist eine automatische Adaption der auf Rasterbasis dargestellten Kennfelder nur schwer möglich.

Gitterbasierte Kennfelder (Rasterkennfelder) gehören zu den *nichtparametrischen Modellen*. Sie haben den Vorteil, dass eine Änderung einzelner Datenpunkthöhen leicht möglich ist. Von Nachteil ist jedoch die exponentielle Zunahme der Daten mit zunehmender Zahl P von Eingangsgrößen, entsprechend N^P für N Stützstellen pro Eingang. Deshalb sind sie im Allgemeinen auf ein- und zweidimensionale Fälle beschränkt. Die Bestimmung der Datenpunkthöhen aus Messungen an beliebigen Stützstellen und adaptive Kennfelder sind durch rekursive Parameterschätzung in Müller (2003) beschrieben.

b) Kennfelder in parametrischer Form

Mehrdimensionale Kennfelder können auch als *parametrische Modelle*, also in Gleichungsform mit relativ wenigen Parametern, in Form von Polynommodellen, neuronalen Netzwerken oder Fuzzymodellen dargestellt werden. Ein Polynommodell lautet z.B.

$$\begin{aligned} Z(X, Y) = & Z_0 + a_1 X + a_2 X^2 + \dots + b_1 Y + b_2 Y^2 + \dots \\ & + c_1 XY + c_2 XY^2 + c_3 Y^2 X + \dots \end{aligned} \quad (7.5.20)$$

Diese Modelle benötigen im Allgemeinen weniger Modellparameter (siehe Beispiel 7.1 und Bild 7.19) und deshalb weniger Speicherplatz. Jedoch ist der Rechenaufwand im Allgemeinen höher als bei Rasterkennfeldern. Durch ihre parametrische Form sind diese Modelle besser geeignet zur Modellidentifikation, da sie infolge der Parameterschätzung einen Störsignalausgleich (Repression) durchführen und eine Festlegung der Eingangsdaten nicht an bestimmte Stützstellen gebunden ist. Polynommodelle sind z.B. Grundlage einer Vermessung mit den Methoden des „Design-of-Experiments“ (DoE), siehe z.B. Retzlaff et al. (1978), Kleppmann (1998), Röpke (2005). Sie sind aber auf statische Modelle beschränkt und nicht so flexibel wie z.B. neuronale Netzmodelle, siehe Zimmerschied et al. (2005).

7.5.5 Parameterschätzung für nicht stetig differenzierbare nichtlineare Prozesse (Reibung und Lose)

Nicht stetig differenzierbare Nichtlinearitäten treten bei mechanischen Systemen besonders in Zusammenhang mit Reibung und Lose auf und bei elektromagnetischen Komponenten in Form der Magnetisierungs-Hysterese.

a) Prozesse mit Reibung

In Abschnitt 4.7 wurde angegeben, wie man trockene und viskose Gleitreibung in Differentialgleichungen berücksichtigen kann. Die trockene Reibung tritt als Konstante in Abhängigkeit der Geschwindigkeit auf, deren Vorzeichen von der Richtung der Geschwindigkeit abhängig ist. Für den Ruhezustand entsteht aus den dynamischen Beziehungen eine Hysteresekurve in Abhängigkeit der Position.

Zur Identifikation von Prozessen mit Reibung kann zunächst die Hysteresekurve direkt durch langsame stetige oder sprungförmige Änderungen der Eingangsgröße $u(t)$ jeweils in einer Richtung und Messung von $y(t)$ punktweise ermittelt werden.

Beschreibt man die Hysteregeraden durch

$$\begin{aligned} y_+(u) &= K_{0+} + K_{1+}u \\ y_-(u) &= K_{0-} + K_{1-}u, \end{aligned} \quad (7.5.21)$$

dann lassen sich die Parameter durch Anwendung der Methode der kleinsten Quadrate aus jeweils $v = 1, 2, \dots, N - 1$ gemessenen Kennlinienpunkten wie folgt ermitteln

$$\hat{K}_{1\pm} = \frac{N \sum u(v)y_{\pm}(v) - \sum u(v) \sum y_{\pm}(v)}{N \sum u^2(v) - \sum u(v) \sum u(v)}, \quad (7.5.22)$$

$$\hat{K}_{0\pm} = \frac{1}{N} \left[\sum y_{\pm}(v) - \hat{K}_{1\pm} \sum u(v) \right]. \quad (7.5.23)$$

Da die Differentialgleichungen linear in den Parametern sind, kann man bei Prozessen mit trockener und viskoser Reibung im bewegten Zustand die direkten Methoden der Parameterschätzung anwenden. Hierzu eignen sich als Prozessmodelle sowohl Differentialgleichungen als auch Differenzgleichungen. In manchen Fällen ist es zweckmäßig, nicht nur eine richtungsabhängige trockene Reibung sondern auch *richtungsabhängige dynamische Parameter* anzusetzen, z.B. in Form der Differenzgleichungen

$$y(k) = - \sum_{i=1}^m a_{1+} y(k-i) + \sum_{i=1}^m b_{1+} u(k-i) + K_{0+}, \quad (7.5.24)$$

$$y(k) = - \sum_{i=1}^m a_{1-} y(k-i) + \sum_{i=1}^m b_{1-} u(k-i) + K_{0-}. \quad (7.5.25)$$

K_{0+} und K_{0-} können hierbei als richtungsabhängige Gleichwertparameter aufgefasst werden. Dann lassen sich folgende Methoden zur Gleichwertschätzung anwenden, Isermann (1992):

- Implizite Schätzung der Gleichwertparameter K_{0+} , K_{0-} ;
- explizite Schätzung der Gleichwertparameter. Hierzu Differenzenbildung $\Delta y(k)$ und $\Delta u(k)$ und Parameterschätzung für

$$\Delta y(k) = - \sum_{i=1}^m \hat{a}_i \Delta y(k-i) + \sum_{i=1}^m \hat{b}_i \Delta u(k-i) \quad (7.5.26)$$

unter Annahme richtungsunabhängiger Dynamikparameter $\hat{a}_i$ und $\hat{b}_i$. Für jede Bewegungsrichtung werden dann $\hat{K}_{0+}$ und $\hat{K}_{0-}$ getrennt bestimmt.

Bei dieser Parameterschätzung mit *richtungsabhängigem Modell* ist als zusätzliche Identifikationsbedingung zu beachten, dass die Bewegung ohne Umkehr in einer Richtung erfolgt, d.h. es muss gelten

$$\dot{y}(t) > 0 \text{ oder } \dot{y}(t) < 0, \quad (7.5.27)$$

welches man z.B. dadurch überprüft, dass

$$\Delta y(k) > \epsilon \text{ oder } \Delta y(k) < \epsilon$$

für alle k mit $\epsilon \geq 0$. Ein Testsignal für proportional wirkende Prozesse, das diese Bedingungen erfüllt, hat Maron (1996) vorgeschlagen, Bild 7.30. Durch einen linearen Anstieg wird die Bewegung in einer Richtung mit bestimmter Geschwindigkeit erzeugt. Dann erfolgt ein Sprung zur Anregung der höheren Frequenzen mit einem Übergang zu einem Beharrungszustand. Die Parameterschätzung muss jeweils bei Bewegungsumkehr (in Bild 7.30 die Punkte 1, 2, 3, ...) abgebrochen und neu gestartet oder mit den Werten derselben Bewegungsrichtung fortgesetzt werden.

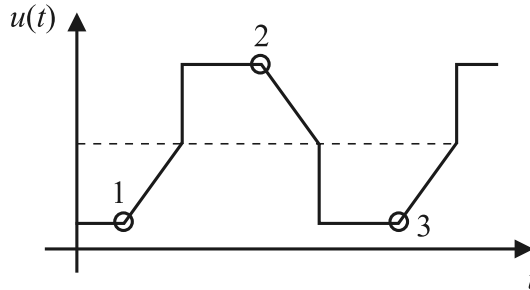


Bild 7.30. Testsignal zur Parameterschätzung von Prozessen mit trockener Reibung

Die Hysterese Kennlinie berechnet sich aus dem statischen Verhalten nach (7.5.24), (7.5.25)

$$y_+(u) = \frac{\hat{K}_{0+}}{1 + \sum \hat{a}_{i+}} + \frac{\sum \hat{b}_{i+}}{1 + \sum \hat{a}_{i+}} u, \quad (7.5.28)$$

$$y_-(u) = \frac{\hat{K}_{0-}}{1 + \sum \hat{a}_{i-}} + \frac{\sum \hat{b}_{i-}}{1 + \sum \hat{a}_{i-}} u. \quad (7.5.29)$$

Zur Validierung der Parameterschätzung aus dem dynamischen Verhalten kann diese so berechnete Kennlinie mit der direkt aus dem statischen Verhalten gemessenen Kennlinie, siehe oben, verglichen werden.

Für rotierende Antriebe wurde von Held und Maron (1988) ein besonderes Parameterschätzverfahren angegeben, bei dem das gemessene Drehmoment mit der

Drehzahlbeschleunigung korreliert und hieraus das Trägheitsmoment bestimmt wird. Dann kann die Reibungsmomentkennlinie in nichtparametrischer Form ermittelt werden.

Die hier beschriebenen Methoden zur Identifikation reibungsbehafteter Prozesse wurden mit sehr gutem Erfolg praktisch erprobt und zur digitalen Regelung mit Reibungskompensation eingesetzt, siehe Maron (1996), Raab (1993). Weitere Methoden sind z.B. in Armstrong-Hélouvy (1991), Canudas de Wit (1988) beschrieben.

b) Systeme mit Lose (Tote Zone)

Als Beispiel wird wiederum ein mechanischer Schwinger mit einer *Lose* oder *toten Zone* der Breite $2y_t$ betrachtet, Bild 7.31. Für den Schwinger ohne Lose gilt

$$m\ddot{y}_2(t) + d\dot{y}_2(t) + cy_2(t) = cy_3(t). \quad (7.5.30)$$

Die Lose lässt sich wie folgt beschreiben

$$y_3(t) = \begin{cases} y_1(t) - y_t & \text{für } y_1(t) > y_t \\ 0 & \text{für } -y_t \leq y_1(t) \leq y_t \\ y_1(t) + y_t & \text{für } y_1(t) < -y_t \end{cases}. \quad (7.5.31)$$

Diese Gleichung führt zu der in Bild 7.31b) eingezeichneten nichtlinearen Kennlinie. Für den Fall, dass sich die Lose an einem Anschlag befindet, so dass $y_1(t) > y_t$, gilt somit

$$m\ddot{y}_2(t) + d\dot{y}_2(t) + cy_2(t) + cy_t = cy_1(t) \quad (7.5.32)$$

und am anderen Anschlag, $y_1(t) < -y_t$,

$$m\ddot{y}_2(t) + d\dot{y}_2(t) + cy_2 - cy_t = cy_1(t). \quad (7.5.33)$$

Die Lose tritt also als Konstante auf, deren Vorzeichen vom Vorzeichen von $y_1(t)$ abhängt. Für den Bereich innerhalb der Lose ist $y_3(t) = 0$ und es gilt somit das Eigenverhalten des Schwingers

$$m\ddot{y}_2(t) + d\dot{y}_2(t) + cy_2(t) = 0, \quad (7.5.34)$$

wenn der Punkt 3 (z.B. durch eine hier nicht betrachtete Reibung) festgehalten wird. Wenn der Punkt 3 nicht festgehalten wird und sich innerhalb der Lose frei bewegen kann, fallen die Federkräfte weg. Es ist dann $y_2 = y_3$ und man muss in (7.5.30) und (7.5.34) $c = 0$ setzen.

Als vereinfachtes Ersatzschaltbild erhält man für die Bereiche außerhalb der Lose Bild 7.32. Man kann die Auswirkung der Lose in diesen Bereichen als Gleichwertverschiebung des Eingangssignales mit wechselndem Vorzeichen auffassen.

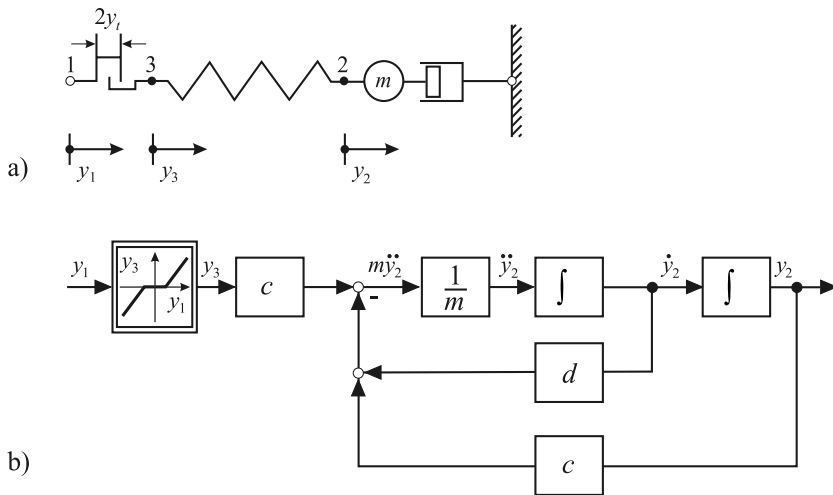


Bild 7.31. Mechanischer Schwinger mit Lose (tote Zone): a) Schematische Anordnung; b) Blockschaltbild für die Fälle $y_1(t) > y_t$ und $y_1(t) < -y_t$

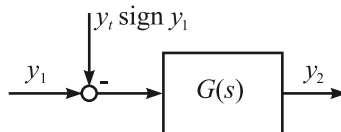


Bild 7.32. Ersatzschaltbild für ein lineares System mit Lose für $|y_1(t)| > |y_t|$

7.6 Aufgaben

- 1) Die folgende lineare Gleichung erster Ordnung lautet:

$$y(k) + a_1 y(k-1) = b_1 u(k-1).$$

Die gemessenen Ein- und Ausgangssignale $u(k)$ und $y(k)$ mit $k = 1 \dots N$ sind vorgegeben. Wie lautet die Schätzgleichung zur Berechnung der Parameter a_1 und b_1 mit der Methode der kleinen Quadrate?

Was müssen Sie in der Schätzgleichung ändern, wenn die Totzeit $d = 5$ hinzugefügt wird?

- 2) Leiten Sie die Schätzgleichung für die Schätzung der Gewichtsfunktion $g(k)$, $k = 0 \dots n$ (Impulsantwort) eines linearen Prozesses her, in dem Sie die Methode der kleinen Quadrate mit den vorgegebenen Ein- und Ausgangssignalen $u(k)$ und $y(k)$ ($k = 1 \dots N$) verwenden.
- 3) Ein lineares System erster Ordnung ist gegeben

$$y(k) + a_1 y(k-1) = b_1 u(k-1)$$

mit folgenden Messungen

k	0	1	2	3	4	5	6	7	8	9	10
$u(k)$	0	1	-1	1	1	1	-1	-1	0	0	0
$y(k)$	0	0	0	0	0	0	1	0	0	0	0

Berechnen Sie die Schätzwerte der Parameter a_1 und b_1 .

- 4) Bestimmen Sie die rekursiven Schätzgleichungen der Methoden der kleinsten Quadrate für den linearen Prozess erster Ordnung

$$y(k) + a_1 y(k-1) = b_1 u(k-1).$$

Wie muss die Schätzgleichung geändert werden, wenn die Totzeit $d = 2$ hinzugefügt wird?

- 5) Die Differentialgleichung erster Ordnung

$$y(t) + a_1 \dot{y}(t) = b_0 u(t)$$

beschreibt ein lineares System im kontinuierlichen Zeitbereich. Das Ein- und das Ausgangssignal $u(t)$ und $y(t)$ und ihre Ableitung $\dot{y}(t)$ werden am zeitdiskreten Abtastsignal $k = 1 \dots N$ gemessen. Bestimmen Sie die Gleichungen, um die Parameter a_1 und b_1 der Differentialgleichung mit der Methode der kleinsten Quadrate zu schätzen.

- 6) Welches sind die Voraussetzungen für die Anwendung der direkten Parameterschätzung (z.B. LS Methoden) für die Parameter der polynomialen nichtlinearen Modelle nach Abschnitt 7.5.1?
- 7) Skizzieren Sie ein Neuron eines MLP und eines RBF Netzwerkes.
- 8) Das Trainieren von MLP Netzwerken und das Trainieren der Gewichte der Ausgangsschicht von RBF Netzwerken soll verglichen werden. Welche Optimierungsmethoden sind anwendbar? Beschreiben Sie die verschiedenen Optimierungsmethoden.
- 9) Betrachten Sie ein Zwei-Schichten Perzeptron-Netzwerk mit P Eingängen, K versteckten Neuronen und einem Ausgang. Leiten Sie eine Gleichung für die gesamte Anzahl der Netzwerk-Gewichte her.
- 10) Bestimmen Sie die Gleichungen für die Interpolation der Kennfelddarstellung mit einem Eingang und einem Ausgang. Welches Interpolationsverfahren erscheint in diesem eindimensionalen Fall?



<http://www.springer.com/978-3-540-32336-5>

Mechatronische Systeme

Grundlagen

Isermann, R.

2008, XIX, 623 S., Softcover

ISBN: 978-3-540-32336-5