

Identifikation eines nichtlinearen dynamischen Mehrgrößensystems mit rekurrenten neuronalen Netzen im Vergleich zu lokal-affinen Zustandsraummodellen

Stephan Godt, Martin Kohlhasse

Center for Applied Data Science Gütersloh, FH Bielefeld

Schulstraße 10, 33330 Gütersloh

E-Mail: {stephan.godt, martin.kohlhasse}@fh-bielefeld.de

1 Einführung

Die Systemidentifikation versucht anhand von Beobachtungsdaten Prozesse mit Hilfe von Modellen bestmöglich zu beschreiben. Diese Modelle ermöglichen es nicht nur das Verhalten eines komplexen Systems abzubilden, sondern dieses auch in gewissem Maße vorherzusagen. Diese Fähigkeit findet besonders in Bereichen der Automatisierungs- und Regelungstechnik ihren Nutzen, vor allem in Teilgebieten wie beispielsweise der modellbasierten Prozesssteuerung und Prozessoptimierung. Der Schlüssel bei jeder Modellierungsaufgabe ist es, eine geeignete Modellstruktur zur Abbildung des untersuchten Prozesses zu wählen. Hier finden bei komplexen Prozessen besonders nichtlineare dynamische Ansätze ihre Anwendung. Die Verfahren zur nichtlinearen dynamischen Modellbildung können allgemein in zwei Kategorien unterteilt werden. Der erste Ansatz verwendet eine externe Dynamik zur Schätzung der Modelle, wobei der aktuelle Modellausgang \hat{y} auf Basis verzögerter Ein- und Ausgangsgrößen geschätzt wird. In der Systemidentifikation finden in diesem Bereich häufig Modelle mit einer NARX-Struktur (Nichtlinear autoregressiv mit exogenem Eingang (Nonlinear ARX)) ihre Anwendung. Verfahren mit einer internen Dynamik, die der zweiten Kategorie zugeordnet werden können, haben keine externe Kopplung verzögerter Ein- und Ausgänge. Hier werden die

DOI: 10.58895/ksp/1000124139-10 erschienen in:

Proceedings – 30. Workshop Computational Intelligence: Berlin, 26. - 27. November 2020

DOI: 10.58895/ksp/1000124139 | <https://www.ksp.kit.edu/site/books/m/10.58895/ksp/1000124139/>

internen Zustände innerhalb des Modells lokal zurückgegeben. Häufige Vertreter dieser Kategorie sind NLSS (*general NonLinear State-Space*)-Modelle. Auch der Einsatz von rekurrenten neuronalen Netzen in diesem Bereich ist ein viel diskutiertes Thema in der heutigen Forschung. Es wurden bereits erste Verbindungen und Gemeinsamkeiten dieser Verfahren aufgezeigt und eine gewisse Vergleichbarkeit in der Modellgenauigkeit nachgewiesen [1], [2].

In dieser Arbeit werden die Untersuchungen hinsichtlich der Vergleichbarkeit der beiden Ansätze aufgegriffen und zwei Methoden zur Systemidentifikation bezüglich ausgewählter Modellcharakteristika miteinander verglichen. Zum einen kommt ein lokal-affines Zustandsraummodell zum Einsatz, welches das Systemverhalten mit Hilfe von lokal-linearen Modellen abbildet, die auf Basis aufgenommener Prozessdaten geschätzt werden. Dies erfolgt mit dem Programmpaket DYLAMOT (DYnamic Local Affine MOdeling Toolbox). Für die lokal linearen Modelle wird dabei eine NARX-Struktur gewählt, wobei die Parameter anhand einer Ausgangsfehlerschätzung bestimmt werden. Zum anderen werden für die Systemidentifikation rekurrente neuronale Netze, genauer „Long short-term memory“-Netze (LSTM), verwendet. Diese besitzen Rückkopplungen zwischen Neuronen verschiedener Schichten, sodass Prozessinformationen über einen Zeitraum erhalten bleiben. Bei einem möglichen Einsatz der Verfahren an realen Prozessen ergeben sich verschiedene Herausforderungen. Einerseits besteht die Möglichkeit, dass Prozesse außerhalb ihres üblichen Einsatzbereichs betrieben werden, was ein gewisses Maß an Extrapolationsfähigkeit in den verwendeten Modellen voraussetzt. Andererseits wirken auf reale Prozesse häufig Störeinflüsse, sodass es sinnvoll ist, die Empfindlichkeit der eingesetzten Modelle beispielsweise gegenüber dem Einfluss von Rauschen zu untersuchen.

Nach einer kurzen Vorstellung der zwei Methoden muss in einem ersten Schritt eine ausreichende Modellgüte mit beiden Verfahren erzielt werden. Daraufhin werden in einem nächsten Schritt die verwendeten Modelle hinsichtlich ihrer Extrapolationsfähigkeit und ihrer Rauschempfindlichkeit näher betrachtet. Die Untersuchungen werden zunächst an einem zuvor definierten Testprozess vorgenommen. Folgend werden ausgewählte Aspekte zusätzlich an einem realen nichtlinearen Mehrgrößensystem untersucht.

2 Modellbildung

In diesem Beitrag sollen dynamische Prozesse, die mit Hilfe nichtlineare Differentialgleichungen in der Form $\dot{x}(t) = f(t, x(t), u(t))$ und $y(t) = h(x(t), u(t))$ abgebildet werden können, durch datenbasierte Ansätze dargestellt werden. Dabei sollen die funktionalen Zusammenhänge zwischen f und h untersucht werden. Zur Modellierung der nichtlinearen dynamischen Systeme werden lokal - affine Zustandsraummodelle sowie tiefe rekurrente neuronale Netze eingesetzt. Beide Verfahren werden im Folgenden kurz vorgestellt.

2.1 Tiefe rekurrente neuronale Netze

In einem *Feedforward*-Netzwerk werden die Informationen auf einem direkten Weg durch das Netz geleitet, d.h. von der Eingabeschicht durch die einzelnen verdeckten Netzwerkschichten (*hidden layers*) zur Ausgabeschicht. Dabei berücksichtigen die Neuronen (*hidden units*) in den jeweiligen Schichten ausschließlich den aktuellen Netzeingang. Ein klassisches einschichtiges *Feedforward*-Netzwerk stellt eine Funktionsabbildung von einem Eingang x zu einem Ausgang \hat{y} (skalare Größen) dar und wird für die Netzwerkschicht j wie folgt beschrieben:

$$\hat{y} = \sum_{j=1}^{d_1} \gamma_j \sigma(\alpha_j \cdot (x + \beta_j)). \quad (1)$$

Hierbei sind die Parameter α und β an die Dimension von x angepasste Größen. Die Variablen α , β und γ stellen die Modellparameter des Netzes dar mit $\theta = \{\alpha_j, \beta_j, \gamma_j\}_{j=1}^{d_1}$. Als Aktivierungsfunktion wird eine Sigmoidfunktion σ herangezogen [3]:

$$y = \sigma(x) = \frac{1}{1 + \exp(-x)}. \quad (2)$$

Der Ausdruck $\sigma(\alpha_j \cdot (x + \beta_j))$ beschreibt in diesem Fall die *hidden units*, die zusammen die verdeckte Netzwerkschicht (*hidden layer*) repräsentieren [1].

Feedforward-Netzwerke können Informationen nur von einem Zeitschritt in den nächsten übertragen. Um Informationen über einen längeren Zeitraum beizubehalten, bieten sich rekurrente neuronale Netze (RNNs) an. Eine effizien-

ente Art von RNNs sind Long-Short Term Memory (LSTM)-Netze, die aufgrund ihrer inneren Struktur Informationen auch über einen längeren Zeitraum speichern können [4]. Ein vollständiges LSTM-Netzwerk besteht aus einer oder mehreren verketteten LSTM-Schichten und einer abschließenden vollständig verketteten linearen Netzwerkschicht. Die Eigenschaft, Informationen über einen längeren Zeitraum zu speichern, wird durch zwei Arten von internen Zuständen bestärkt sowie der Möglichkeit, den Informationsfluss zu und aus diesen Zuständen zu steuern. Als innere Zustände einer LSTM-Schicht zum Zeitpunkt t dienen der verdeckte Zustand \mathbf{h}_t , auch *hidden state* genannt, und der Zellstatus \mathbf{c}_t , der für das Langzeitgedächtnis des Netzwerks verantwortlich ist. Der *hidden state* sowie der Zellstatus repräsentieren dabei sowohl den Ausgang einer LSTM-Schicht zum Zeitpunkt $t - 1$ als auch den Eingang der folgenden LSTM-Schicht zum Zeitpunkt t . Auf Basis der Eingangsgrößen \mathbf{x}_t und des *hidden states* \mathbf{h}_{t-1} aus der vorherigen Netzwerkschicht wird der Informationsfluss in und aus dem Zellstatus über sogenannte *gates* gesteuert. Hierbei handelt es sich um ein *forget gate* \mathbf{f}_t , um ein *input gate* \mathbf{i}_t , einem *cell gate* \mathbf{g}_t und einem *output gate* \mathbf{o}_t . Das *forget gate* \mathbf{f}_t entscheidet, welche Informationen aus dem Zellstatus entfernt werden sollen. Hierfür wird eine Sigmoidfunktion σ (Gleichung (2)) herangezogen, deren Ausgabe zwischen 0 (vollständiges Vergessen) und 1 (Information vollständig beibehalten) liegt. Das *input gate* \mathbf{i}_t entscheidet, ebenfalls auf Basis einer Sigmoidfunktion, welche Werte dem Zellstatus hinzugefügt werden sollen. Dafür wird durch das *cell gate* \mathbf{g}_t mithilfe einer tanh-Funktion eine Kandidatenliste erstellt. Die Kombination aus *input gate* und *cell gate* führt zu einer Aktualisierung des Zellstatus \mathbf{c}_t . Zum Schluss gibt das *output gate* \mathbf{o}_t auf Basis einer Sigmoidfunktion an, welche Informationen aus dem aktuellen Zellstatus \mathbf{c}_t in den Ausgang der Netzwerkschicht fließen.

Die *gates* lassen sich wie folgt definieren:

$$\mathbf{f}_t = \sigma(\mathbf{W}_{\text{xf}}\mathbf{x}_t + \mathbf{b}_{\text{xf}} + \mathbf{W}_{\text{hf}}\mathbf{h}_{t-1} + \mathbf{b}_{\text{hf}}) \quad (3)$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_{\text{xi}}\mathbf{x}_t + \mathbf{b}_{\text{xi}} + \mathbf{W}_{\text{hi}}\mathbf{h}_{t-1} + \mathbf{b}_{\text{hi}}) \quad (4)$$

$$\mathbf{g}_t = \tanh(\mathbf{W}_{\text{xg}}\mathbf{x}_t + \mathbf{b}_{\text{xg}} + \mathbf{W}_{\text{hg}}\mathbf{h}_{t-1} + \mathbf{b}_{\text{hg}}) \quad (5)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{\text{xo}}\mathbf{x}_t + \mathbf{b}_{\text{xo}} + \mathbf{W}_{\text{ho}}\mathbf{h}_{t-1} + \mathbf{b}_{\text{ho}}). \quad (6)$$

Hierbei repräsentieren \mathbf{W}_{xf} , \mathbf{W}_{xi} , \mathbf{W}_{xg} und \mathbf{W}_{xo} die Gewichtsmatrizen des Eingangs \mathbf{x}_t und die Matrizen \mathbf{W}_{hf} , \mathbf{W}_{hi} , \mathbf{W}_{hg} , \mathbf{W}_{ho} die Gewichtsmatrizen des vorherigen *hidden states* für das jeweilige *gate*. Darüber hinaus spiegeln die Vektoren \mathbf{b}_{xf} , \mathbf{b}_{xi} , \mathbf{b}_{xg} und \mathbf{b}_{xo} den Bias am Eingang und \mathbf{b}_{hf} , \mathbf{b}_{hi} , \mathbf{b}_{hg} , \mathbf{b}_{ho} den Bias im *hidden state* des jeweiligen *gates* wider. Die Gewichtsmatrizen und Bias-Vektoren enthalten dabei die Parameter der LSTM-Schichten. Zur Berechnung des Zellstatus \mathbf{c}_t und des aktuellen *hidden states* \mathbf{h}_t werden die *gates* wie folgt verwendet:

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \mathbf{g}_t \quad (7)$$

$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t). \quad (8)$$

Zur Aktualisierung des Zellstatus \mathbf{c}_t erfolgt eine Addition der Hadamard - Produkte (\circ) aus dem *forget gate* \mathbf{f}_t und dem vorherigen Zellstatus \mathbf{c}_{t-1} sowie aus dem *input gate* \mathbf{i}_t und dem *cell gate* \mathbf{g}_t . Der *hidden state* \mathbf{h}_t kann folgend auf Basis von \mathbf{o}_t und \mathbf{c}_t berechnet werden. Abschließend bildet der Ausgang der letzten LSTM-Schicht den Eingang für eine vollständig verkettete lineare Netzwerkschicht, die den finalen Netzausgang y_t ausgibt. Eine Übersicht über den Aufbau von LSTM-Schichten sowie deren Verkettung kann [5] entnommen werden.

Ein weiterer herauszustellender Fakt im Kontext von LSTM-Netzwerken ist deren strukturelle Ähnlichkeit zu den in der Systemidentifikation etablierten NLSS-Modellen. Beide Strukturen können unter gewissen Annahmen in die jeweilig andere überführt werden. Detaillierte Untersuchungen zur Vergleichbarkeit von NLSS-Modellen und LSTM-Netzen können [1] und [2] entnommen werden.

2.2 LOLIMOT

Eine weitere Möglichkeit der datenbasierten Modellbildung stellt der LOLIMOT (LOcal LInear MOdel Tree) - Algorithmus dar. Dieser gehört zu den inkrementellen heuristischen Konstruktionsalgorithmen und bildet nichtlineare Prozesse durch Überlagerung von mehreren lokal-affinen Teilmodellen ab, was den Vorteil einer leichten Interpretierbarkeit bietet [9].

Der Modellausgang eines lokal-affinen Modells wird wie folgt definiert:

$$\hat{y} = \sum_{i=1}^M (w_{i0} + w_{i1}x_1 + \dots + w_{in}x_n) \Phi_i(\mathbf{z}). \quad (9)$$

Der Modellausgang mit M affinen Teilmodellen ergibt sich aus der Summe der einzelnen Teilmodelle, gewichtet mit einer Aktivierungsfunktion $\Phi_i(\mathbf{z})$. Dabei wird jedes Teilmodell i durch n Eingangsgrößen $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T$ und $n+1$ Parametern $\mathbf{w}_i = [w_{i0} \ w_{i1} \ \dots \ w_{in}]^T$ bestimmt. Die Aktivierung hängt von dem Eingangsvektor der Aktivierungsfunktionen $\mathbf{z} = [z_1 \ z_2 \ \dots \ z_n]^T$ ab, welcher dem Arbeitspunkt des lokal-affinen Modelles entspricht. Wie \mathbf{x} wird auch \mathbf{z} aus den insgesamt zur Verfügung stehenden Eingangsgrößen bestimmt [7]. Der Gültigkeitsbereich der Teilmodelle wird durch eine normierte Aktivierungsfunktion beschrieben, welche den Ausgang eines jeden Teilmodells in Abhängigkeit des z -Regressors gewichtet und wie folgt definiert ist:

$$\Phi_i(\mathbf{z}) = \frac{\mu_i(\mathbf{z})}{\sum_{m=1}^M \mu_m(\mathbf{z})}. \quad (10)$$

Hier spiegelt $\mu_i(\mathbf{z})$ den Grad der Zugehörigkeit zu den Gültigkeitsbereichen wider, wobei die Summe der Aktivierungsfunktionen für eine sinnvolle Interpretation gleich Eins sein muss: $\sum_{i=1}^M \Phi_i(\mathbf{z}) = 1$. Um eine einfach zu interpretierende Modellstruktur sowie einen glatten Übergang zwischen den Teilmodellen zu erzeugen, werden als Zugehörigkeitsfunktion Gaußglocken verwendet

$$\mu_i(\mathbf{z}) = \exp\left(-\frac{1}{2} \frac{(z_1 - c_{i1})^2}{\sigma_{i1}^2}\right) \cdot \dots \cdot \exp\left(-\frac{1}{2} \frac{(z_n - c_{in})^2}{\sigma_{in}^2}\right), \quad (11)$$

wobei c_{ij} das Zentrum und σ_{ij} die Standardabweichung der Gaußglocke des i -ten Teilmodells in der j -ten Dimension des z -Regressors sind. Die Größen c_{ij} und σ_{ij} ergeben sich dabei aus der Position und Größe der jeweiligen Gültigkeitsbereiche der Teilmodelle im Eingangsraum [7]. LOLIMOT zeichnet sich durch eine achsen-orthogonale Partitionierung des Eingangsraums aus, was eine einfache, schnelle und vor allem transparente Strukturoptimierung ermöglicht. Dabei wird schrittweise das lokal schlechteste Teilmodell durch eine achsen-orthogonale, mittige Teilung in zwei neue Teilmodelle aufgeteilt.

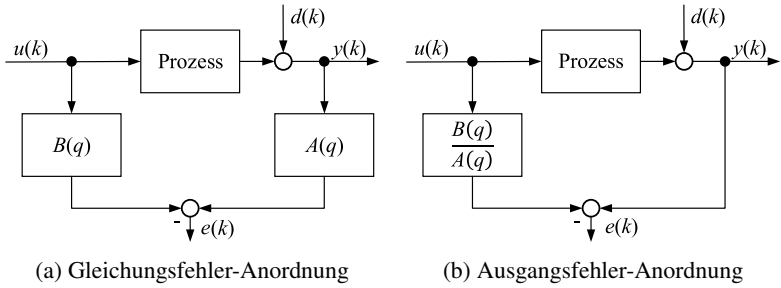


Bild 1: Darstellung der Gleichungsfehler-Anordnung (NARX) für die Parameterschätzung in a) und der Ausgangsfehler-Anordnung für die Strukturoptimierung in b).

Das lokal schlechteste Teilmodell kann beispielsweise anhand der lokalen Verlustfunktion V :

$$V_{i,\text{lokal}} = \sum_{k=1}^N (y(k) - \hat{y}(k))^2 \Phi_i(\mathbf{z}(k)) \quad (12)$$

beurteilt werden, welche den quadrierten Modellfehler, gewichtet mit der jeweiligen Aktivierungsfunktion, aufsummiert. Dabei wird aus allen möglichen Teilungen diejenige ausgewählt, die zu dem geringsten globalen Fehler führt. Ein großer Vorteil der in Gleichung (9) beschriebenen Struktur kommt bei der Parameterschätzung zum Tragen. Da der Modellausgang linear in seinen Parametern ist, können lineare Verfahren, wie die Methode der gewichteten kleinsten Fehlerquadrate, eingesetzt werden.

LOLIMOT wird im Folgenden zur Modellierung eines nichtlinearen dynamischen Systemmodells verwendet, sodass die bisher betrachtete Modellstruktur um eine Dynamik erweitert werden muss. Dazu wird eine NARX-Struktur zur Schätzung der Parameter verwendet, wobei ein Ein-Schritt-Prädiktionsfehler basierend auf einer Gleichungsfehler-Anordnung (Bild 1a) minimiert wird. Hierbei stellt $B(q)$ das Zählerpolynom und $A(q)$ das Nennerpolynom der Übertragungsfunktion des Modells dar, wobei q mit $q^{-1}x(k) = x(k-1)$ den Zeitverzögerungsoperator darstellt. Des Weiteren repräsentiert $d(k)$ eine mögliche Störung [7]. Der Modellausgang des dynamischen lokal-affinen Gesamtmo-

dells kann wie folgt beschrieben werden:

$$\hat{y}(k) = \sum_{i=1}^M (b_{i1}u(k-1) + \dots + b_{in}u(k-n) - a_{i1}y(k-1) - \dots - a_{in}y(k-n) + \xi_i)\Phi_i(\mathbf{z}(k)). \quad (13)$$

Dabei wird exemplarisch ein SISO-System (*Single-Input-Single-Output*) n -ter Ordnung zugrunde gelegt. Es werden verzögerte Eingangsgrößen mit den Parametern a_i und verzögerte Ausgangsgrößen mit den Parametern b_i sowie ein Offset ξ als Regressoren verwendet. Die Minimierung des Gleichungsfehlers hinsichtlich der Parameterschätzung hat den Vorteil, dass auch bei dynamischen Systemen lineare Schätzverfahren, wie die Methode der kleinsten Fehlerquadrate, anwendbar sind, da Linearität in den Parametern gegeben ist. Die Strukturoptimierung basiert dagegen auf dem Ausgangsfehler (NOE (Nichtlinearer Ausgangsfehler)-Struktur), Bild 1b. Dies hat den Vorteil, dass die durch die Rückkopplung bedingte Fehlerfortpflanzung, aufgrund der Parameteroptimierung in der Gleichungsfehler-Anordnung, in der Strukturoptimierung vermieden werden kann [8]. Die Gleichung (13) kann auch in einer parameterveränderlichen Form dargestellt werden:

$$\hat{y}(k) = \sum_{i=1}^n b_i(\mathbf{z}(k))u(k-i) - \sum_{i=1}^n a_i(\mathbf{z}(k))\hat{y}(k-i) + \xi(\mathbf{z}(k)). \quad (14)$$

Das Software-Paket DYLAMOT (DYnamic Local Affine MOdeling Toolbox) beinhaltet Erweiterungen des LOLIMOT-Verfahrens zur Identifikation nichtlinearer dynamischer Prozesse mit Hilfe von lokal-affinen Modellen, wodurch eine bessere Modellgüte erzielt werden kann. Dabei verwendet DYLAMOT zur Partitionierung des Eingangsraums den heuristischen LOLIMOT - Konstruktionsalgorithmus. Eine der implementierten Erweiterungen ist die Verzögerung des arbeitspunktbeschreibenden z -Regressors entsprechend der jeweiligen Modelleingangsgrößen. Hierbei erfolgt die Änderung der Modellparameter in Abhängigkeit von $\mathbf{z}(k)$ zum Zeitpunkt k nicht gleichzeitig, sondern jeweils verzögert mit der Verzögerung i der dazugehörigen Eingangsgrößen $u(k-i)$ oder $\hat{y}(k-i)$ [7]. Ein großer Vorteil der Verzögerung der z -Regressoren ist ein stark verbessertes Eingangs- /Ausgangsverhalten im Vergleich zu einem

Modell mit gleichzeitiger Parameteränderung. Während die verzögerte Parameteränderung das Eingangs- /Ausgangsverhalten verbessert, wird der Offset mit einer unabhängigen Zähler-Dynamik ausgestattet, wodurch besonders die Interpretierbarkeit der Modelle erhöht wird. Durch die Erweiterung werden für jedes Teilmodell Offset-Parameter mit verzögerter Parameteränderung geschätzt. Der Modellausgang mit verzögerter Parameteränderung und Offset-Zähler-Dynamik ergibt sich dann zu:

$$\begin{aligned}\hat{y}(k) = & \sum_{i=1}^n b_i(\mathbf{z}(k-i))u(k-i) - \sum_{i=1}^n a_i(\mathbf{z}(k-i))\hat{y}(k-i) \\ & + \sum_{i=1}^n \xi_i(\mathbf{z}(k-i)).\end{aligned}\tag{15}$$

Durch die eingeführte Erweiterung ist der Offset im Modell abhängig von unterschiedlichen Zeitpunkten, wodurch die Flexibilität des Modells gesteigert wird. Die verzögerte Parameteränderung ist dabei für die Verwendung mehrerer Offset-Parameter zwingend, da sonst alle Offset-Parameter zu einem zusammengefasst werden können [7]. Durch die vorgenommenen Erweiterungen der Modellstruktur ergibt sich, dass jedes Teilmodell in der Ausgangsfehler-Anordnung (Bild 1b) geschätzt wird. Hierdurch fließen die Parameter nichtlinear in den Ausgangsfehler ein, sodass nichtlineare Optimierungsverfahren zur Schätzung der Parameter herangezogen werden müssen [9].

Durch die getroffenen Erweiterungen der Modellstruktur lässt sich eine minimale Zustandsraumrealisierung erzeugen, die in Beobachternormalform dargestellt werden kann [7]. Dies ermöglicht einen Vergleich der vorgestellten Modellstrukturen gemäß [2].

3 Anwendungsbeispiel - Testprozess

Bei dem Vergleich der vorgestellten Methoden liegt der Fokus auf der zu erreichenden Modellgüte. Des Weiteren werden Aspekte wie die Extrapolationsfähigkeit sowie die Rauschempfindlichkeit der verwendeten Modelle untersucht. Der Vergleich findet zunächst an einem ausgewählten Testprozess statt. Dies

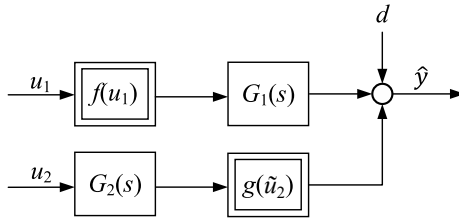


Bild 2: Strukturbild des verwendeten Testprozesses.

erfolgt zum einen mit dem Programmpaket DYLAMOT, zum anderen unter Verwendung der *Deep Learning Toolbox* in MATLAB [10].

3.1 Systembeschreibung - Testprozess

Die Struktur des Testprozesses kann Bild 2 entnommen werden. Bei dem Testprozess handelt es sich um ein nichtlineares dynamisches MISO (*Multiple Input, Single Output*)-System, basierend auf einem Hammerstein- und einem Wiener Modell. Zudem wird der Ausgang \hat{y} mit einem weißen Rauschen d beaufschlagt. Zur Anregung des Prozesses werden APRB (Amplituden- moduliertes Pseudo Rausch Binär)-Signale verwendet und auf beide Eingänge u_1 und u_2 aufgeschaltet. Um eine hohe Modellgüte zu erzielen, müssen dabei die entscheidenden Frequenz- und Amplitudenbereiche des Systems hinreichend angeregt werden. Hierfür werden APRB-Signale mit einer Länge von 2046 Sekunden und einer Taktzeit von 2 Sekunden gewählt. Die Amplituden basieren dabei auf einer Gleichverteilung. Die statischen Nichtlinearitäten

$$f(u_1) = \frac{1}{1 + \exp(-u_1)} = \frac{1}{2} \left(1 + \tanh\left(\frac{u_1}{2}\right) \right) \quad (16)$$

$$g(\tilde{u}_2) = \arctan(\tilde{u}_2) + \tilde{u}_2 \quad (17)$$

werden für $\{(u_1, u_2) \in \mathbb{R}^2 \mid 0, 1 \leq u_j \leq 0, 9 \forall j \in \{1, 2\}\}$ betrachtet. Die Übertragungsfunktionen $G_1(s)$ und $G_2(s)$ werden wie folgt definiert:

$$G_1(s) = \frac{1}{5s^3 + 66s^2 + 73s + 12} \quad (18)$$

$$G_2(s) = \frac{1}{s^3 + 2s^2 + 6s + 5}. \quad (19)$$

Bei den Untersuchungen am Testprozess werden $N = 204600$ Datenpunkte aufgenommen, von denen 70% zum Trainieren der Modelle und 30 % als Testdatensatz verwendet werden. Zur Beurteilung der jeweiligen Modellgüte wird die Abweichung zwischen dem gemessenen Wert y und dem vorhergesagten Wert \hat{y} bestimmt: $e = y - \hat{y}$. Zudem wird der RMSE auf den Testdaten berechnet:

$$\text{RMSE} = \sqrt{\frac{1}{N_T} \sum_{t=1}^{N_T} (y_t - \hat{y}_t)^2}, \quad (20)$$

wobei N_T die Anzahl an Testdatenpunkten beschreibt. Für die Untersuchungen muss eine geeignete Auswahl der verschiedenen Hyperparameter gefunden werden. Für eine ausreichende Modellgüte wird in DYLAMOT zur Modellbildung die Anzahl der lokal linearen Modelle zwischen 4 und 8 variiert. Des Weiteren wird die Ordnungszahl der lokalen Modelle zwischen 2 und 4 angepasst. Für das LSTM-Netzwerk wird die Anzahl der Neuronen, die Anzahl der verdeckten Schichten (*hidden layers*) und die Anzahl der Epochen für das Training so variiert, dass sich eine vergleichbare Modellgüte einstellt.

3.2 Modellvergleich - Testprozess

Mit beiden Verfahren kann bei den Untersuchungen eine hohe Modellgüte erreicht werden. Das beste Ergebnis wird in DYLAMOT mit einem lokal-linearen Modell, bestehend aus 7 Teilmodellen vierter Ordnung, erzielt. Bei dem LSTM-Netzwerk wird das beste Ergebnis unter der Verwendung einer LSTM-Schicht mit 128 Neuronen erzielt. Dabei wird beim Training eine maximale Anzahl von 500 Epochen gewählt und zum Anlernen der ADAM (*adaptive moment estimation*)-Algorithmus [11] mit einer konstanten Lernrate von 0.01 verwendet. In Bild 3a ist ein Ausschnitt der zur Anregung verwendeten

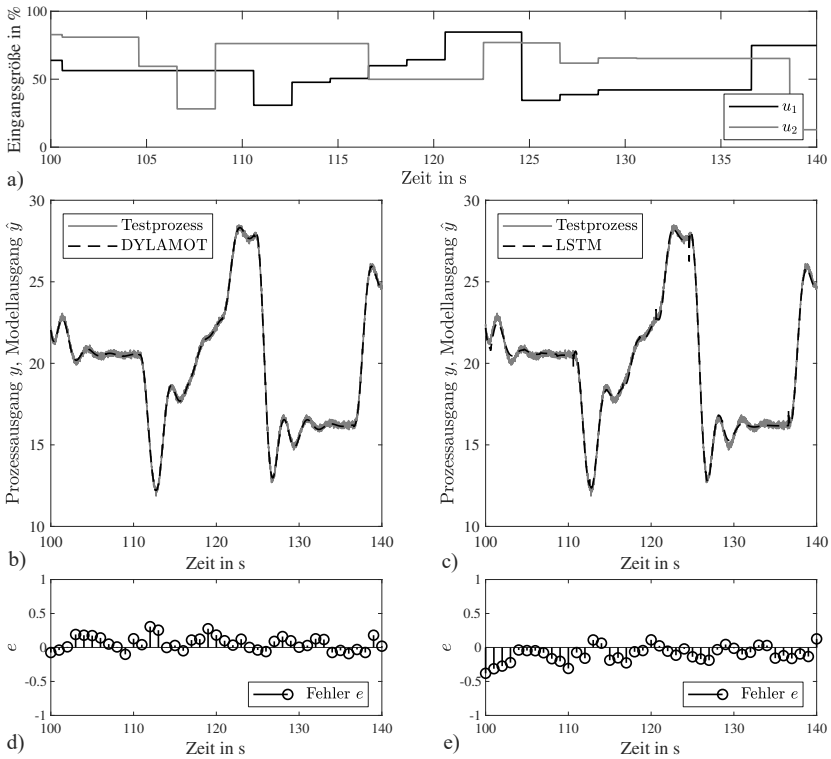


Bild 3: Vergleich der Modelle am Testprozess: a) verwendete Eingangsgrößen; b) Testprozessausgang und Modellausgang DYLAMOT; c) Testprozessausgang und Modellausgang LSTM; d) Fehler e Modellausgang DYLAMOT und e) Fehler e Modellausgang LSTM.

APRB-Signale für die beiden Eingangsgrößen u_1 und u_2 dargestellt. Aufgrund der Vergleichbarkeit zur Heizstrecke werden hierbei prozentuale Größen angegeben. In den Bildern 3b und c sind für beide Modelle die gemessenen und geschätzten Werte (y bzw. \hat{y}) der Ausgangsgröße zu sehen, basierend auf einem separaten Testdatensatz. Ebenfalls dargestellt ist der Fehler e bzw. die Abweichung zwischen gemessenem und geschätztem Wert Bilder 3d und e. Die hohe Modellgüte wird anhand des Vergleichs zwischen den gemessenen und geschätzten Werten in den Bildern 3b und c deutlich. Hierbei werden bei beiden Verfahren keine großen Abweichungen in den Verläufen festgestellt. DYLAMOT erreicht dabei eine geringfügig bessere Modellgüte als das

LSTM-Netz. Dies spiegelt sich auch beim RMSE wider. Dieser nimmt für den gesamten Testdatensatz bei DYLAMOT einen Wert von 0,132 an und unter Verwendung des LSTM-Netzes einen Wert von 0,298.

Für die Untersuchung der Extrapolationsfähigkeit werden die Amplituden der verwendeten APRB-Signale bei der Modellbildung verringert. Die Modellbildung findet mit APRB-Signalen statt, deren Amplituden aus dem Intervall $[0.3, 0.7]$ stammen. Daraufhin erfolgt die Generalisierung dieser Modelle auf Testdaten, deren Ursprung eine Anregung mit Amplituden aus dem Intervall $[0.1, 0.9]$ ist. Die trainierten Modelle werden folglich auf Daten getestet, deren minimalen und maximalen Werte die min. bzw. max. antrainierten Modellausgänge unter- bzw. überschreiten. In den Darstellungen 4a und b sind für beide Methoden die Eingangsgrößen aufgetragen sowie die Grenzen des antrainierten Bereichs. In den Bildern 4c und d sind die Modelle auf den amplitudenmodifizierten Testdaten generalisiert, was durch das Überschreiten der min. und max. Modellausgänge aus dem Training deutlich wird. Bei einem Vergleich der Eingangsdaten mit den jeweiligen Prozess- und Modellausgängen werden die Extrapolationsbereiche sichtbar. Bei dem Vergleich der Verläufe in den Bildern 4c und d wird deutlich, dass in diesen Bereichen unter der Verwendung von DYLAMOT die Abweichungen zwischen gemessenem und geschätztem Wert deutlich kleiner sind als beim LSTM-Netzwerk. Dieses Ergebnis wird auch vom RMSE widerspiegelt, der in diesem Fall für DYLAMOT 0,7554 und für das LSTM-Netzwerk 1,5957 beträgt. Ein Grund für dieses Ergebnis ist im Aufbau von LSTM-Netzen zu finden. Die einzelnen LSTM-Schichten basieren auf Sigmoid- und Tanh-Funktionen, die für die Aktualisierung der Zellzustände verantwortlich sind. Diese weisen entsprechend ihrer Verläufe einen unteren und einen oberen Sättigungsbereich auf, in denen das Wachstum gegen Null strebt. Diese Bereiche werden bei der Verwendung von Eingangsdaten, die außerhalb des antrainierten Modellbereichs liegen, erreicht. Diese Einschränkungen bestehen unter der Verwendung von DYLAMOT nicht. DYLAMOT weist ein lineares Extrapolationsverhalten auf, sodass keine unteren oder oberen Sättigungsbereiche erreicht werden können.

Ein weiterer Untersuchungspunkt in diesem Beitrag zielt auf die Empfindlichkeit der erstellten Modelle gegenüber möglichen Rauscheinflüssen ab. Für die Untersuchung werden die erstellten Modelle herangezogen und mit Rauschen

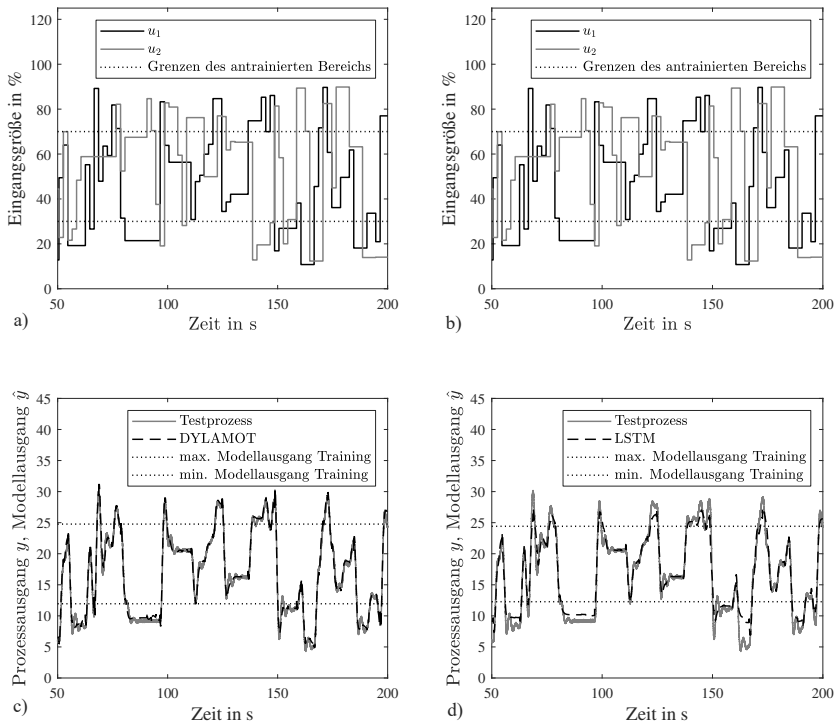


Bild 4: Extrapolationsverhalten am Testprozess: a),b) verwendete Eingangsgrößen; c) Modellausgang DYLAMOT bei Generalisierung auf amplitudenmodifizierten Testdatensatz; d) Modellausgang DYLAMOT bei Generalisierung auf amplitudenmodifizierten Testdatensatz.

beaufschlagt. Dafür wird der Testprozess mit weißem Rauschen angeregt. Der Prozessausgang y wird dann zur Generalisierung der Modelle herangezogen. Zur Anregung wird ein weißes Rauschen mit einer Abtastzeit von 0,01 Sekunden auf beide Eingänge aufgeschaltet. Für u_1 wird zusätzlich eine spektrale Leistungsdichte von 8W/Hz gewählt und für u_2 ein Wert von 4W/Hz. In Darstellung 5 sind für beide Methoden die gemessenen und geschätzten Werte (y bzw. \hat{y}) der Ausgangsgröße zu sehen. Beim Vergleich der Abbildungen ist deutlich zu erkennen, dass bei den gewählten Modell- und Rauschparametern das DYLAMOT-Modell (Bild 5a) stärker auf den verrauschten Eingang reagiert als das LSTM-Netz (Bild 5b). Dies spiegeln ebenfalls die mittels FFT (*fast Fourier transform*) ermittelten Grenzfrequenzen des Prozessausgangs und

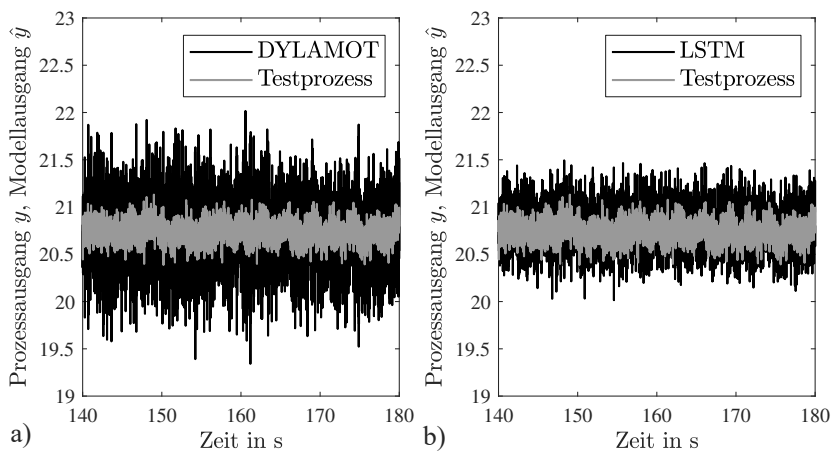


Bild 5: Empfindlichkeit der erstellten Modelle gegen weißem Rauschen: a) DYLAMOT; b) LSTM.

Tabelle 1: Grenzfrequenzen des Testprozesses und der erstellten Modelle.

Ausgang	Grenzfrequenz rad/s
Testprozess (Bode)	0,5000
Testprozess (FFT)	0,4947
DYLAMOT (FFT)	0,7727
LSTM (FFT)	0,5283

der jeweiligen Modellausgänge wider, die der Tabelle 1 entnommen werden können. Zur Kontrolle wird ebenfalls die Grenzfrequenz des Testprozesses aus dem Bode-Diagramm ermittelt. Die ermittelten Grenzfrequenzen aus dem Bode-Diagramm und der FFT sind wie erwartet beinahe identisch. Die größte Grenzfrequenz bei der Untersuchung weist DYLAMOT auf, was den Ergebnissen in Darstellung 5 entspricht. In dem unter der Verwendung von DYLAMOT erstellten Modell werden höhere Frequenzen deutlich weniger stark gedämpft als durch den eigentlichen Prozess und das erstellte LSTM-Modell. Eine Erklärung dafür bietet die Modellordnung, die für DYLAMOT gewählt wurde, um eine akzeptable und vergleichbare Modellgüte für den gewählten Testprozess zu erhalten. Man kann sagen, dass eine höhere Modellordnung zwar zu einer

besseren Prozessabbildung auf den Testdaten führt, das Modell aber anfälliger für Störeinflüsse macht.

4 Anwendungsbeispiel - Heizstrecke

Während im vorherigen Abschnitt die Untersuchungen an einem ausgewählten Testprozess stattfanden, wird nun ein realer nichtlinearer dynamischer Prozess herangezogen. Hierbei wird wie zuvor die jeweilige Modellgüte verglichen und die Extrapolationsfähigkeit untersucht. Auf eine Untersuchung der Rauschempfindlichkeit wird am realen Prozess verzichtet.

4.1 Systembeschreibung - Heizstrecke

Für die Untersuchungen wird eine Heizstrecke der Firma „ELWE Technik“ verwendet. Es handelt sich dabei um ein Labormodell, das über ein xPC Target-System unter Matlab-Simulink angesteuert werden kann. In der Heizstrecke verbaut sind neben einer Heizung und einem Lüfter auch Sensoren, mit denen die Lufttemperatur und der Luftmassenstrom erfasst werden können. Des Weiteren befindet sich eine manuell verstellbare Drosselklappe am Lufteinlass. Das System mit zwei Eingangs-, zwei Ausgangs- und einer Störgröße kann somit als Mehrgrößensystem angesehen werden. Bei den Eingangsgrößen des Systems handelt es sich um die Lüfterleistung P_L und die Heizleistung P_H . Bei der für die Untersuchung betrachteten Ausgangsgröße handelt es sich um die gemessene Temperatur T . Zur Anregung des Prozesses werden ebenfalls APRB-Signale verwendet und auf beide Eingänge u_1 und u_2 aufgeschaltet. Dabei stammen die Amplituden aus dem Intervall $[10,90]\%$ der jeweils maximal möglichen Leistungen. Sonstige Einstellungen zur Systemanregung gleichen den Angaben, die für die Untersuchungen am Testprozess getroffen worden sind, und können dem vorherigen Kapitel entnommen werden.

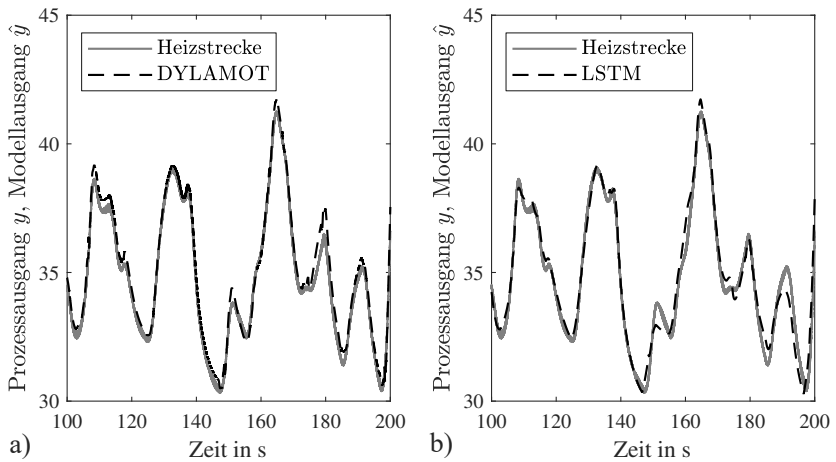


Bild 6: Vergleich der Modelle an der Heizstrecke: a) DYLAMOT; b) LSTM.

4.2 Modellvergleich - Heizstrecke

Mit beiden Verfahren kann eine vergleichbar hohe Modellgüte erreicht werden. Dabei findet die jeweilige Modellbildung mit denselben Hyperparametern statt, die auch für die Untersuchungen am Testprozess gewählt worden sind. Die Modellgüte wird anhand des Vergleichs zwischen den gemessenen und geschätzten Werten (Bilder 6a und b) deutlich. Hierbei kommt es zwar an einigen Stellen in den Verläufen zu geringen Abweichungen, für den gesamten Verlauf kann aber von einer ausreichend hohen Modellgüte gesprochen werden. DYLAMOT erreicht dabei ein geringfügig besseres Ergebnis als das LSTM-Netz. Dies spiegelt sich auch beim RMSE wider. Dieser nimmt für den gesamten Testdatensatz bei DYLAMOT einen Wert von 0,3884 an und unter Verwendung des LSTM-Netzes einen Wert von 0,5121.

Für die Untersuchung der Extrapolationsfähigkeit werden wie am Testprozess die Amplituden der zur Anregung verwendeten APRB-Signale bei der Modellbildung verringert. Die Modellbildung findet mit APRB-Signalen statt, deren Amplituden aus dem Intervall $[30,70]\%$ der jeweils maximal möglichen Leistungen stammen. Daraufhin erfolgt die Generalisierung dieser Modelle auf Testdaten, deren Ursprung eine Anregung mit Amplituden aus dem Intervall

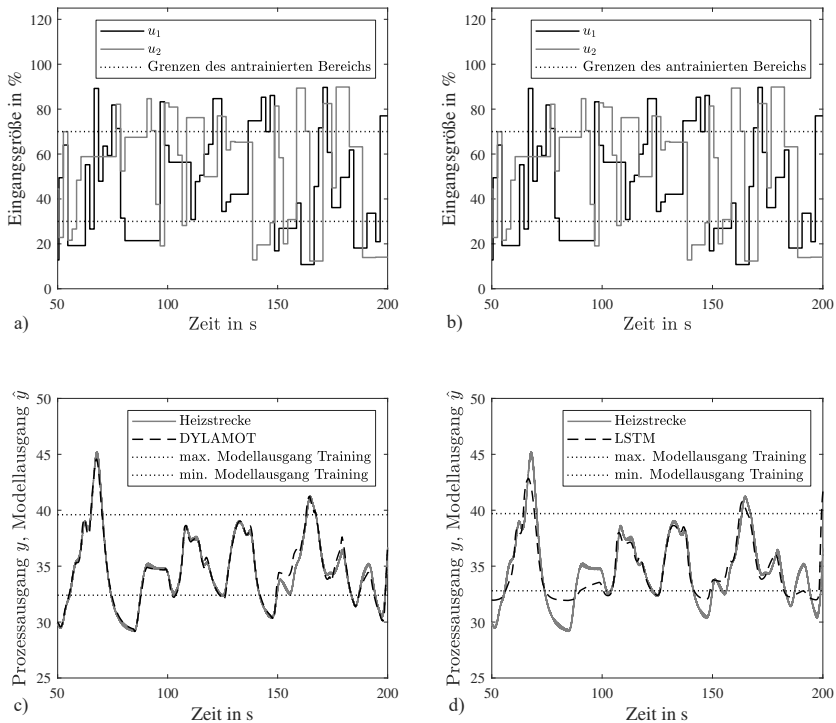


Bild 7: Extrapolation an der Heizstrecke: a), b) verwendete Eingangsgrößen; c) Modellausgang DYLAMOT bei Generalisierung auf amplitudenmodifizierten Testdatensatz; d) Modellausgang DYLAMOT bei Generalisierung auf amplitudenmodifizierten Testdatensatz.

[10,90]% ist. In den Darstellungen 7a und b sind für beide Methoden die Eingangsgrößen aufgetragen sowie die Grenzen des antrainierten Bereichs. In den Bildern 7c und d werden wie am Testprozess die Modelle auf den amplitudenmodifizierten Testdaten generalisiert, was durch das Überschreiten der min. und max. Modellausgänge aus dem Training deutlich wird. Durch den Vergleich der Eingangsdaten mit den jeweiligen Prozess- und Modellausgängen werden die Extrapolationsbereiche sichtbar. Hierbei werden die Ergebnisse aus den Untersuchungen am Testprozess bestätigt. Auch an der Heizstrecke kann mit DYLAMOT eine bessere Modellgüte erzielt werden als mit dem LSTM-Netzwerk. Beim Vergleich der Verläufe in den Bildern 7c und d wird deutlich, dass unter der Verwendung von DYLAMOT die Abweichungen zwi-

schen gemessenem und geschätztem Wert unter- und oberhalb des antrainierten Bereichs deutlich kleiner sind als beim LSTM-Netzwerk. Dieses Ergebnis wird auch vom RMSE widergespiegelt, der in diesem Fall für DYLAMOT 0,9848 und für das LSTM-Netzwerk 1,8220 beträgt.

5 Zusammenfassung und Ausblick

In diesem Beitrag werden tiefe rekurrente neuronale Netze (LSTM-Netze) bzw. lokal-affine Zustandsraummodelle (DYLAMOT) zur Abbildung nichtlinearer dynamischer Systeme verwendet. Dabei wird die jeweils mit den Verfahren erzielte Modellgüte verglichen. Zusätzlich wird ein Vergleich hinsichtlich der Extrapolationsfähigkeit und der Rauschempfindlichkeit gezogen. Es wird gezeigt, dass unter den gewählten Rahmenbedingungen in beiden Anwendungen eine hohe Modellgüte erzielt werden kann, DYLAMOT jedoch am Testprozess sowie an der Heizstrecke ein geringfügig besseres Ergebnis erzielt. Des Weiteren wird dargestellt, dass bei der Verwendung von DYLAMOT die Abweichungen zwischen gemessenem und geschätztem Wert außerhalb der antrainierten Bereiche deutlich kleiner sind als beim LSTM-Netzwerk. Zudem wird gezeigt, dass bei diesen Rahmenbedingungen das DYLAMOT-Modell stärker auf den verrauschten Eingang reagiert als das LSTM-Netz. Als weiterer Schritt in dem Methodenvergleich kann der Speicherbedarf der Modelle untersucht werden. Dies ist besonders bei einer Verwendung der Modelle im industriellen Umfeld von Bedeutung, wo Rechen- und Speichermöglichkeiten häufig begrenzt sind. Ein weiterer zu untersuchender Aspekt ist die zu erzielende Modellgüte bei der Verwendung verschiedener Anregungssignale. APRB-Signale können häufig in der Praxis, aufgrund der hohen Dynamik, nicht eingesetzt werden. Es werden stetige Anregungssignale gesucht, die in der Praxis sowohl einsetzbar sind als auch eine ausreichende Eingangsraumabdeckung vorweisen, um eine ausreichende Modellgüte zu erzielen.

Förderung

Dieses Vorhaben wurde aus Mitteln des Europäischen Fonds für regionale Entwicklung (EFRE) gefördert (Förderkennzeichen 34.EFRE-0300119).

Literatur

- [1] L. Ljung, C. Andersson, K. Tiels und T. B. Schön. „Deep Learning and System Identification“. In: *Proc., IFAC Congress* Berlin. 2020.
- [2] M. Schüssler, T. Munker und O. Nelles. „Deep Recurrent Neural Networks for Nonlinear System Identification“. In: *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, S. 448-454. IEEE, 2019.
- [3] J. Sjöberg, H. Hjalmarsson und L. Ljung. „Neural Networks in System Identification“. Linköping University. 1994.
- [4] S. Hochreiter und J. Schmidhuber. „Long Short - Term Memory“. In: *Neural computation* 9.8. S. 1735-1780. Princeton University Press. 2019.
- [5] J. Gonzalez und W. Yu. „Non-linear system modeling using LSTM neural networks“. In: *IFAC-PapersOnLine* 51.13. S. 485-489. 2018.
- [6] O. Nelles. „Nonlinear System Identification“. Berlin Heidelberg: Springer. 2001.
- [7] R. Zimmerschied. „Identifikation nichtlinearer Prozesse mit dynamischen lokal-affinen Modellen: Maßnahmen zur Reduktion von Bias und Varianz“. Dissertation, In: *Fortschritt-Bericht VDI Nr. 1150, Reihe 8*. Düsseldorf: VDI-Verlag. 2008.
- [8] D. Schröder. „Intelligente Verfahren: Identifikation und Regelung nichtlinearer Systeme“. Berlin Heidelberg: Springer. 2010.
- [9] L. Ljung. „System Identification: Theory for the User“. Prentice Hall. 1999.

- [10] M. H. Beale, M. T. Hagan und H. B. Demuth. „Deep Learning Toolbox: User’s Guide“. www.mathworks.com. 2020.
- [11] D. P. Kingma und J. L. Ba. „Adam: A method for stochastic optimization“. In: *International Conference on Learning Representations*, 12. 2014.