

**Entwurf neuronaler Netze zur Prädiktion von Prozessgrößen
in einer Schmelzwanne für Floatglas
- Teil 2 -**

Dipl.-Ing. Martin Handreg
Brandenburgische Technische Universität Cottbus-Senftenberg
Lehrstuhl Regelungssysteme und Leittechnik

1. Einleitung

In den letzten zwei Jahrzehnten erlangten verbesserte Regelungs- und Optimierungsmethoden zunehmend Einfluss in der Glasindustrie. Dazu gehören analytische Prozessmodelle zum besseren Verständnis oder moderne prädiktive Regelkonzepte zur übergeordneten Prozessoptimierung (z.B. [1], [2]). Die Grundlage einer modernen Regelstrategie ist das hinterlegte mathematische Prozessmodell. Die Auswahl eines geeigneten Modells hängt von den Eigenschaften des Prozesses ab, die es abbilden soll. Eine Beschreibung des Prozessverhaltens und der regelungstechnischen Problemstellungen erfolgte bereits in [3]. Ein Abgleich von möglichen Prozessmodellen (z.B. Übertragungsfunktionen, Zustandsmodelle, CFD) und den Anforderungen, denen das Modell gerecht werden muss (z.B. Abbildung der Prozesseigenschaften, Modellierungsaufwand, benötigtes Prozesswissen) offenbarte das Potential, mit künstlich neuronalen Netzen das komplexe und nichtlineare Prozessverhalten in einer Schmelzwanne für Flachglas nachbilden zu können.

Auf Basis eines realen Datensatzes einer Flachglasanlage wurden in [3] erste neuronale Prozessmodelle aufgestellt und der Einfluss verschiedener Parameter untersucht, u. a. der Ein- und Ausgangsgrößen, der verborgenen Neuronen und vergangenen Messzeitpunkte. Die Simulationsergebnisse waren vielversprechend. Als weitere Entwurfparameter wurden die Trainings- und Aktivierungsfunktionen genannt. Der Einfluss dieser Parameter auf die Genauigkeit der neuronalen Netze wird nachfolgend erläutert. Desweiteren wird eine mögliche modellbasierte prädiktive Regelstrategie beschrieben.

2. Entwicklungsumgebung

Die Programmierung erfolgte mit der Software MATLAB. Der Aufbau und die Wirkungsweise eines NARX-Netzes (**N**onlinear **A**uto**R**egressive with **E**Xternal Input) sowie Ausführungen zum gegebenen Datensatz sind bereits in [3] beschrieben. Zum besseren Verständnis zeigt Abbildung 2.1 erneut die vereinfachte Übersicht der gegebenen Messanordnung.

Im Rahmen weiterer Untersuchungen wurden insgesamt fünf geeignete neuronale Netze evaluiert, an denen der Einfluss der Netzparameter (Trainings- und Aktivierungsfunktion) untersucht wird (Abbildung 2.2). Die NARX-Netze nutzen zur Berechnung des aktuellen Ausgangswertes die vergangenen (TDL, **T**apped **D**elay **L**ines) Messwerte der Ein- und Ausgangsgrößen. Das NIO-Netz (**N**onlinear **I**nput **O**uput) zieht nur die vergangenen Eingangsgrößen, das NAR-Netz (**N**onlinear **A**uto**R**egressive) nur die vergangenen Ausgangsgrößen heran.

Die Methodik der Simulationen ist wie folgt aufgebaut: die verschiedenen neuronalen Netze wurden stets mit den Daten der ersten 12 Stunden trainiert und anschließend mit für das Netz unbekanntem Daten getestet. Nachfolgend sind stets die Testergebnisse dargestellt.

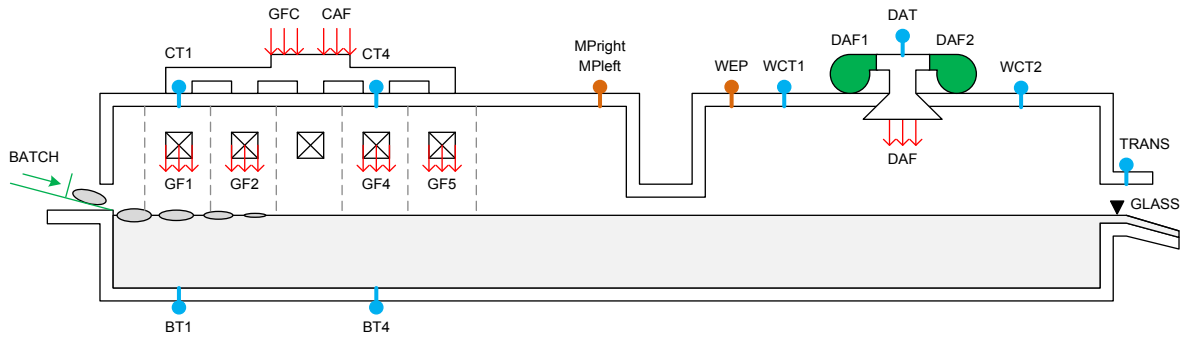


Abbildung 2.1: Vereinfachte Übersicht der Messanordnung

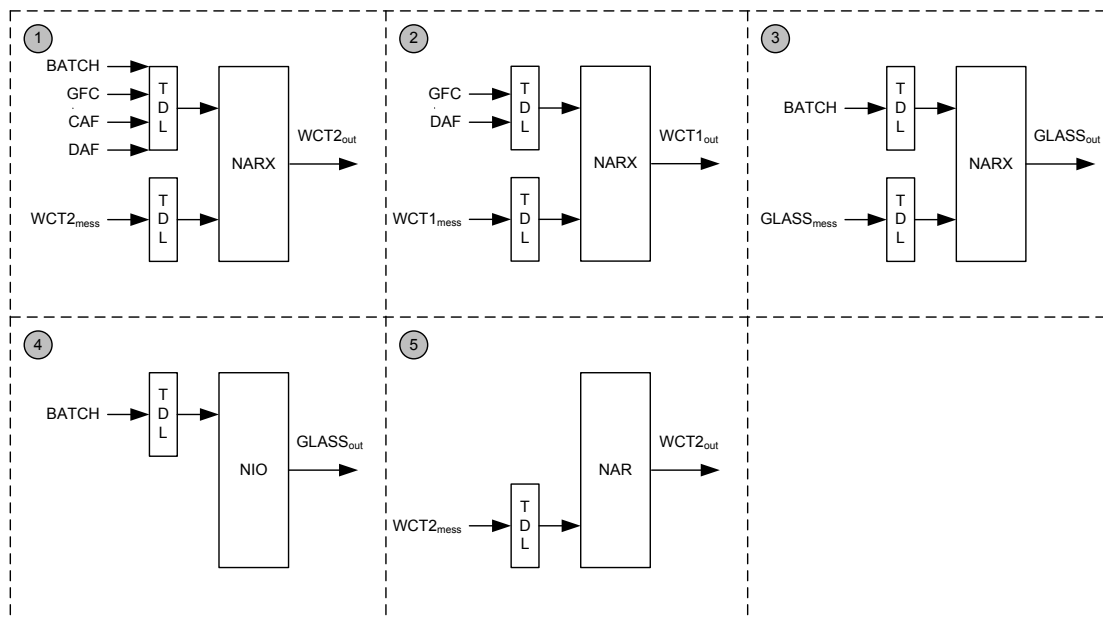


Abbildung 2.2: Übersicht der als geeignet eingestuft neuronalen Netze

3. Simulationsergebnisse

3.1. Einfluss der Trainingsmethoden

Der Trainingsprozess beinhaltet das wiederholte Anpassen der Gewichte und Schwellwerte zur Minimierung einer vorgegebenen Gütefunktion. Als Gütefunktion wurde der quadratische Fehler zwischen dem berechneten Netzausgang und dem tatsächlich gemessenen Ausgang gewählt. MATLAB bietet zwölf verschiedene Trainingsfunktionen an, auf die an dieser Stelle nicht näher eingegangen wird. Die meisten basieren auf der Gradientenberechnung der Gewichte. Bei allen Algorithmen wird das bekannte Verfahren der Backpropagation angewendet. Eine Auflistung und mathematische Beschreibung aller verfügbaren Trainingsmethoden findet sich in [4].

Die Abbildung 3.1 zeigt die Testergebnisse für alle zwölf verfügbaren Trainingsmethoden am Beispiel der Gewölbetemperatur WCT2 (Netz 1) im hinteren Bereich der Abstehwanne. In jedem Diagramm werden die vom Netz berechneten Ausgangsgrößen (blau) den tatsächlich gemessenen Werten (schwarz) gegenübergestellt. Die zwei Trainingsmethoden, mit denen die höchste Genauigkeit (gemessen an der Standardabweichung) erzielt wurde, sind rot hervorgehoben. Aus Gründen der Übersichtlichkeit wurde auf die Achsenbeschriftung verzichtet.

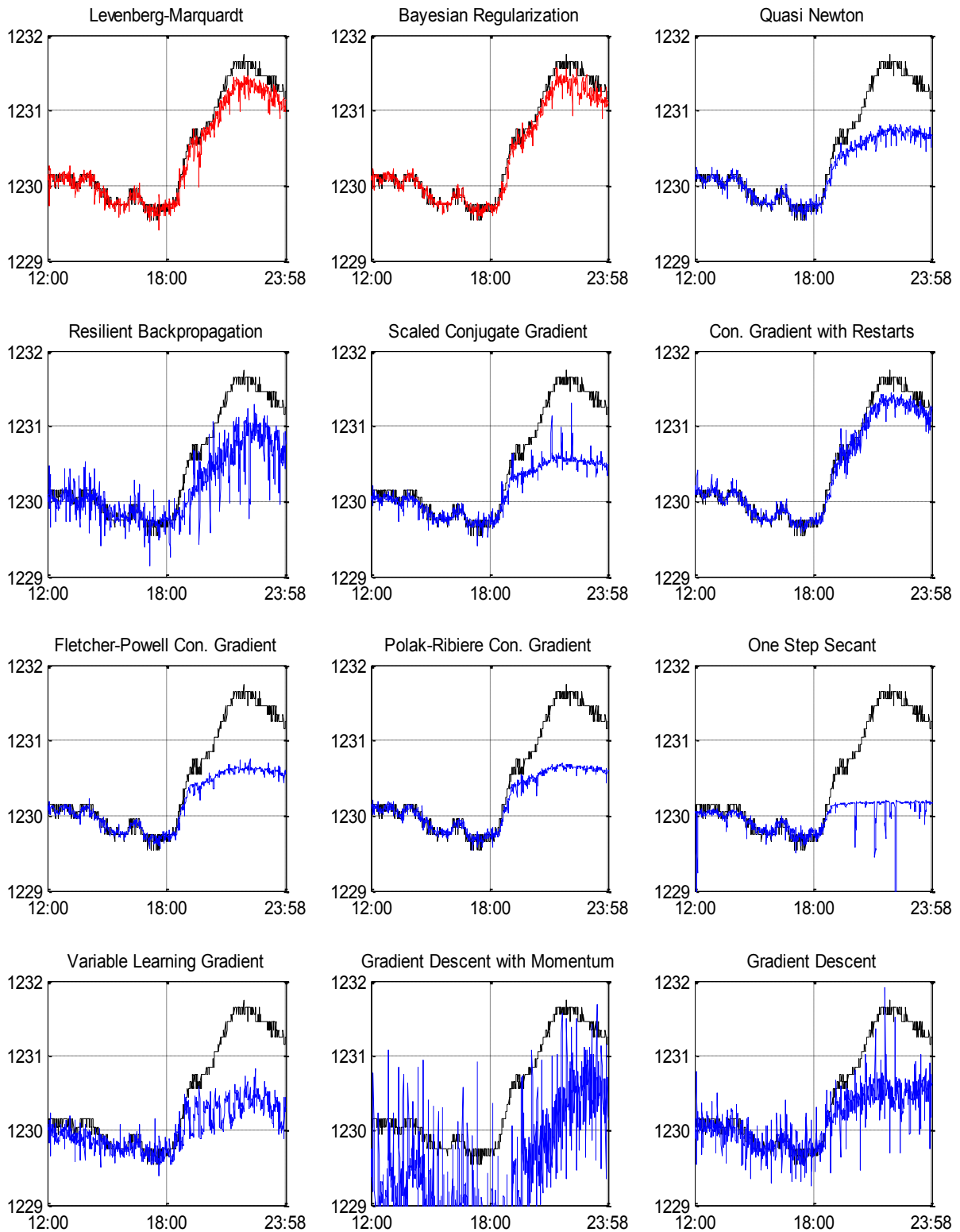


Abbildung 3.1: Zeitliche Verläufe der tatsächlichen (schwarz) und berechneten Ausgangsgröße WCT2 (Netz1) im Testzeitraum für verschiedene Trainingsmethoden (Bemerkung: Auf der Abszisse ist die Zeit [hh:mm] und auf der Ordinate die Temperatur [K] dargestellt. Die verwendete Trainingsmethode ist Titel des Diagramms.)

Bei vier der fünf Netze wurde mit dem Levenberg-Marquardt-Algorithmus (LM), der sich als eine Kombination aus dem Newton- und dem klassischen Gradientenverfahren versteht, und dem Bayesian Regularization Algorithmus (BR) die höchste Genauigkeit erzielt. Neben den Standardabweichungen dienen die Anzahl der Iterationen und die benötigte Trainingszeit als Bewertungskriterium. Die LM-Methode benötigte im Schnitt 9 Iterationen und 0,23 Sekunden Trainingszeit. Diese Werte stellen mit Abstand Bestwerte dar. Der BR-Algorithmus erzielt eine ebenso hohe Genauigkeit, jedoch sind die Trainingszeiten (im Schnitt 17,5 Sekunden) und deutlich höheren Iterationen von Nachteil. Der LM-Algorithmus wurde bereits in anderen Anwendungsgebieten zum Training von neuronalen Netzen erfolgreich eingesetzt, u.a. in [5], [6] und [7]. Für den gegebenen Anwendungsfall ist dieser Algorithmus zu empfehlen.

3.2. Einfluss der Aktivierungsfunktionen

Mit der Aktivierungsfunktion wird festgelegt, welche Wirkung den gewichteten Eingangssignalen zukommt. Somit haben diese Funktionen entscheidenden Einfluss auf das Netzverhalten. Es wird in lineare, sigmoide und Basisfunktionen sowie Funktionen mit Zweipunktverhalten unterschieden. Eine detaillierte Beschreibung aller 15 in MATLAB verfügbaren Aktivierungsfunktionen findet sich in [4].

Abbildung 3.2 (folgende Seite) zeigt die Trainings- und Testergebnisse von Netz 1 für alle Aktivierungsfunktionen. Aus Gründen der Übersichtlichkeit wurde erneut auf eine Achsenbeschriftung verzichtet. Die berechneten Ausgangssignale sind blau dargestellt. Das beste Ergebnis ist rot hervorgehoben. Die jeweilige Aktivierungsfunktion wurde in beiden Schichten verwendet. Es ist zu erkennen, dass lediglich die lineare Funktion *purelin* geeignet ist.

Da in beiden Schichten eine Aktivierungsfunktion festgelegt wird, ergeben sich 225 mögliche Kombinationen, die allesamt getestet und ausgewertet wurden. Die Kombination *purelin/purelin* führt bei drei von fünf Netzen zu der höchsten Genauigkeit. Die Tauglichkeit der begrenzenden Aktivierungsfunktionen (*poslin*, *satlin*, *satlins*, *tansig*, *logsig*, *elliotsig*) hängt maßgeblich vom Wertebereich der Ausgangsgröße ab. Es ist darauf zu achten, dass das Netz in einem sinnvollen und vollständigen Wertebereich trainiert wird. Für eine MPC-Strategie (**M**odel **P**redictive **C**ontrol) sind diese Funktionen daher ungeeignet. Verlässt die Regelgröße den trainierten Wertebereich, liefert das neuronale Prozessmodell zwangsläufig fehlerhafte Ausgangswerte, die zu kontraproduktiven Stelleingriffen führen können.

Für NARX-Netze verwendet MATLAB standardmäßig die sigmoide *tansig*-Funktion in der ersten Schicht. Die Begrenzungseigenschaft dieser Funktion hat bei sensiblen und turbulenten Prozessen mit minimalen Zeitverzögerungen den Vorteil, dass eventuelle Ausreißer oder Messrauschen unterdrückt werden. Die für die vorliegende Aufgabenstellung interessierenden Prozessgrößen unterliegen nur geringfügigen Änderungen. Ausreißer oder unverhältnismäßige Änderungen sind für die Temperaturen in der Abstehwanne und den Glasstand aus physikalischer Sicht ausgeschlossen. Aus diesem Grund kann auf die Begrenzungseigenschaft verzichtet und die lineare Aktivierungsfunktion verwendet werden.

Ein weiterer Vorteil der linearen Aktivierungsfunktion ist der im Vergleich zur *tansig*-Funktion geringere Rechenaufwand. Netze, die viel Rechenaufwand erfordern (Mehrgrößennetze, hohe Anzahl an Neuronen oder Tapped Delay Lines), liefern mitunter stark verrauschte Ausgangssignale (vergleiche [3]). Dieser Nachteil kann mit der linearen Aktivierungsfunktion reduziert werden. Abbildung 3.3 (Seite 6) verdeutlicht diesen Sachverhalt am Beispiel von Netz 2. Auf der linken Seite ist der Einfluss der TDL-Werte mit der Aktivierungsfunktion *purelin* geringer als mit der Aktivierungsfunktion *tansig* (rechts).

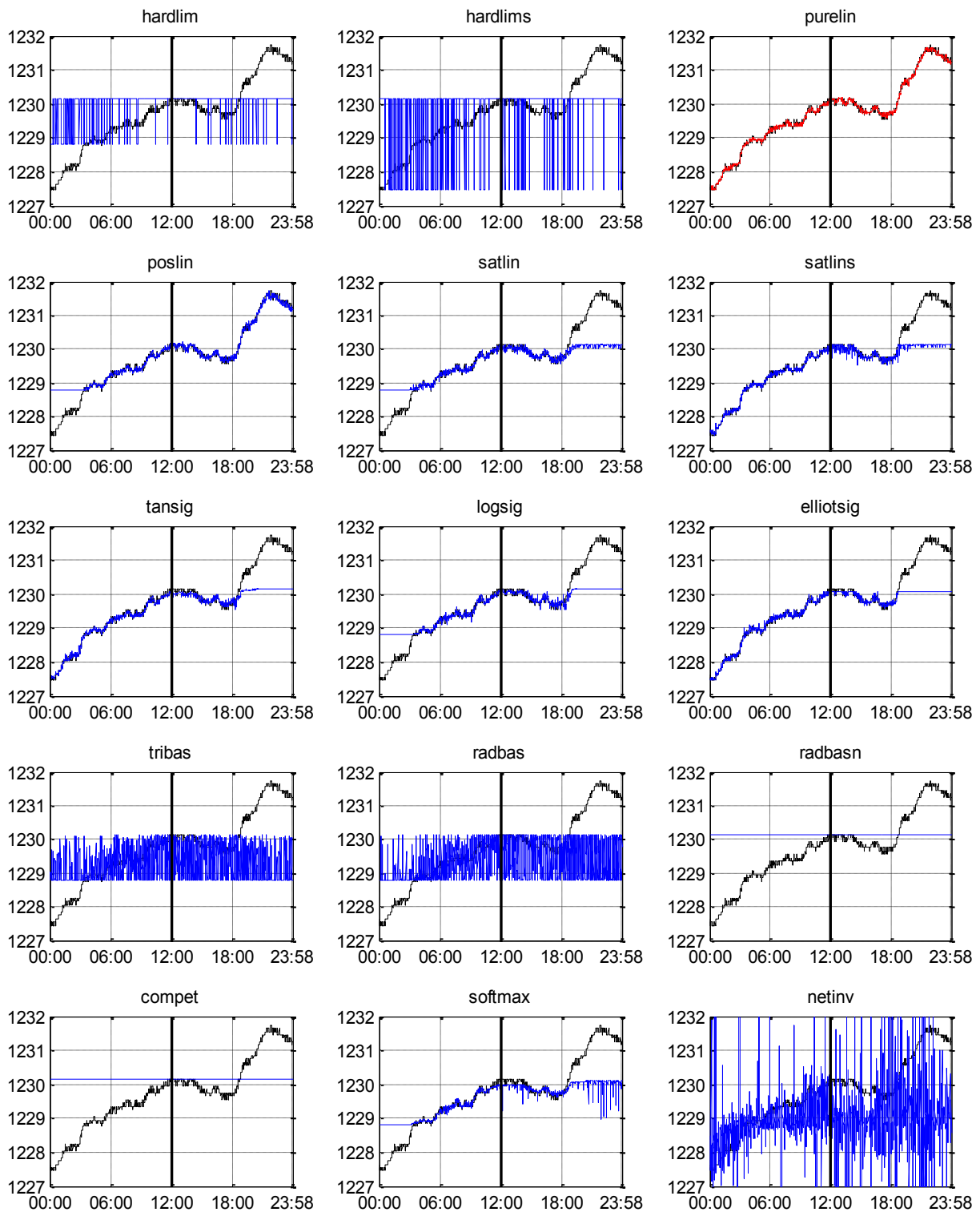


Abbildung 3.2: Simulationsergebnisse (blau) von Netz 1 (WCT2) im Trainings (0-12Uhr) - und Testzeitraum (12-24) für verschiedene Aktivierungsfunktionen (Bemerkung: Auf der Abszisse ist die Zeit [hh:mm], auf der Ordinate die Temperatur [K] dargestellt. Die in beiden Schichten verwendete Aktivierungsfunktion ist Titel des Diagramms.)

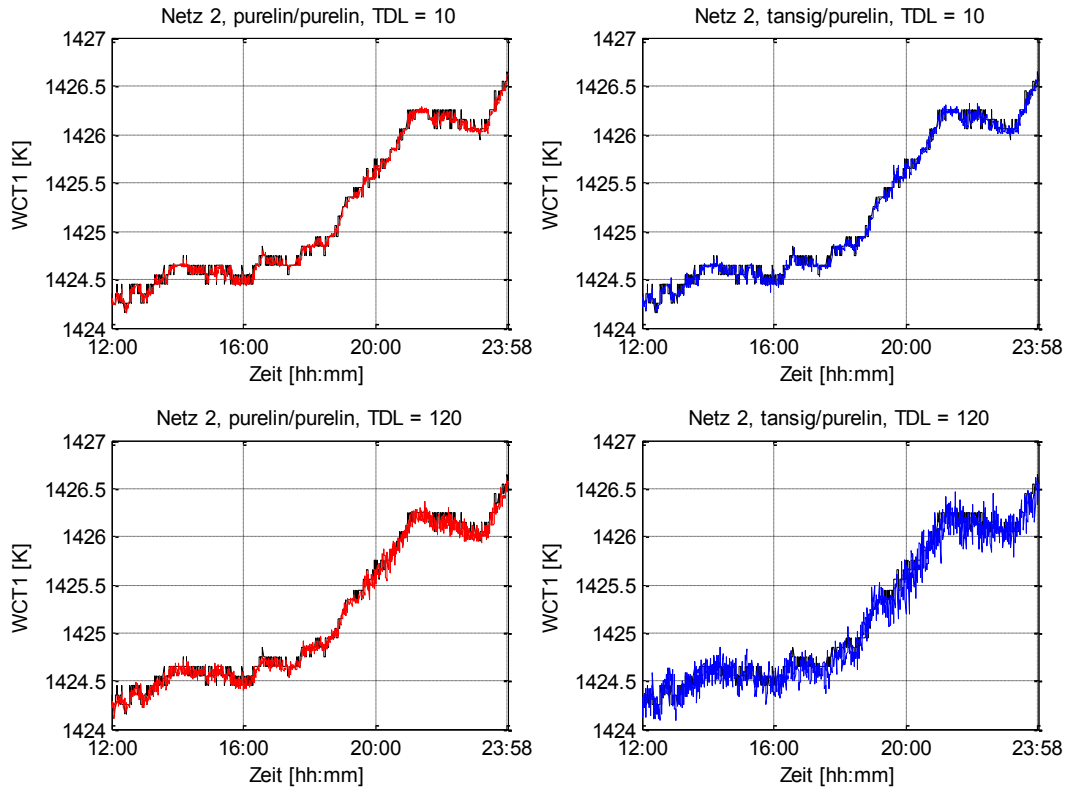


Abbildung 3.3: Gegenüberstellung der berechneten Ausgangswerte im Testzeitraum für Netz 2 bei TDL=10 (oben) / TDL=120 (unten) und der in der ersten Schicht verwendeten Aktivierungsfunktionen *purelin* (links) / *tansig* (rechts).

3.3. Reproduzierbarkeit

In mehreren Untersuchungen ist aufgefallen, dass Netze trotz gleicher Konfiguration nach einem erneuten Training mit identischen Daten nicht immer die gleichen Ausgangswerte liefern. Diese Streuung steigt mit zunehmendem Rechenaufwand. Abbildung 3.4 verdeutlicht diesen Sachverhalt am Netz 1. Das Netz wurde zehnmal hintereinander mit den gleichen Daten trainiert und getestet. Die Standardabweichungen schwanken im Mittel um 15%. Bei noch rechenintensiveren Netzen/Konfigurationen wurden Streuungen von 30% ermittelt.

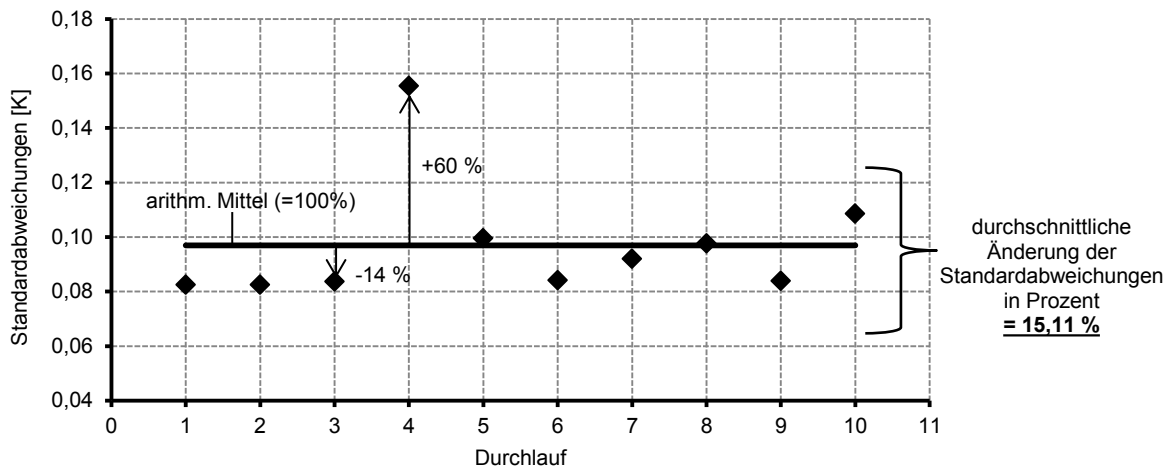


Abbildung 3.4: Standardabweichungen von Netz 1 für TDL=120 bei 10 Simulationsdurchläufen

Für eine belastbare Analyse zum Einfluss der einzelnen Netzparameter ist eine hundertprozentige Reproduzierbarkeit der Ergebnisse unabdingbar. Die Ursache dieser Streuung wurde in der zufälligen Initialisierung der Gewichte und Schwellwerte vor dem ersten Iterationsschritt identifiziert. Diese Werte definieren den Ausgangspunkt im Gradientenabstiegsverfahren. Aus diesem Grund wurde der offene Quellcode der MATLAB-Trainingsfunktion modifiziert. Anstatt mit Zufallszahlen werden die Wichtungsmatrizen mit konstanten Werten (Einsen) initialisiert. Durch diese Modifikation liefert jedes Netz nach erneutem Training mit identischen Daten auch identische Ausgangswerte.

Die Vermutung, dass durch diesen Eingriff die bisherigen Ergebnisse zum Einfluss der einzelnen Netzparameter nicht mehr gültig sind, bestätigte sich nicht. Es traten kleine Abweichungen auf, der jeweilige Trend ist jedoch vollständig reproduzierbar. Abbildung 3.5 stellt zur Bestätigung die Standardabweichungen (Netz 1) zum Einfluss der TDL-Werte direkt gegenüber. Die Reproduzierbarkeit gilt auch für die anderen erstellten Netze. Die Ergebnisse zum Einfluss aller anderen Netzparameter (Neuronen, Trainingsfunktion, Aktivierungsfunktion etc.) wurden ebenfalls bestätigt.

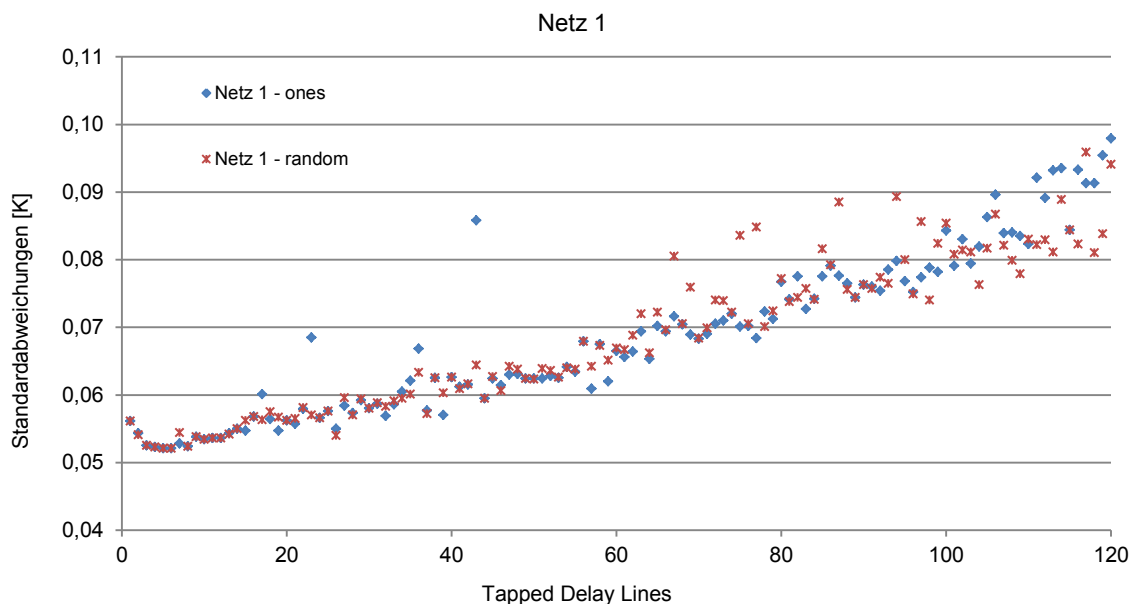


Abbildung 3.5: Vergleich der Standardabweichungen von Netz 1 in Abhängigkeit der Tapped Delay Lines bei konstanter (ones) und zufälliger (random) Initialisierung der Wichtungsmatrizen

4. Modellbasierte prädiktive Regelstrategie

Die künstlich neuronalen Netze lassen sich nun in verschiedene Regelstrategien einbetten. Dabei kann grundsätzlich unterschieden werden, ob die Netze als Prozessmodell oder direkt als Regler dienen sollen. Detaillierte Ausführungen zu bereits bekannten neuronalen Regelstrategien finden sich unter anderem in [4], [8] und [9]. Um dem gegenwärtigen Trend in der Glasindustrie gerecht zu werden, wird nachfolgend gezeigt, wie neuronale Netze in einer modellbasierten prädiktiven Regelstrategie integriert werden können.

Jede MPC-Strategie setzt sich aus einer modellgestützten Prädiktion und dynamischer Optimierung zusammen. Es wird nach einer optimalen Stellgrößenfolge gesucht, mit der sich die Regelabweichungen im Prädiktionshorizont minimieren lassen. Die berechnete optimale Stellgröße $u_{\text{opt}}(t)$ wird anschließend an den Prozess ausgegeben (Abbildung 4.1).

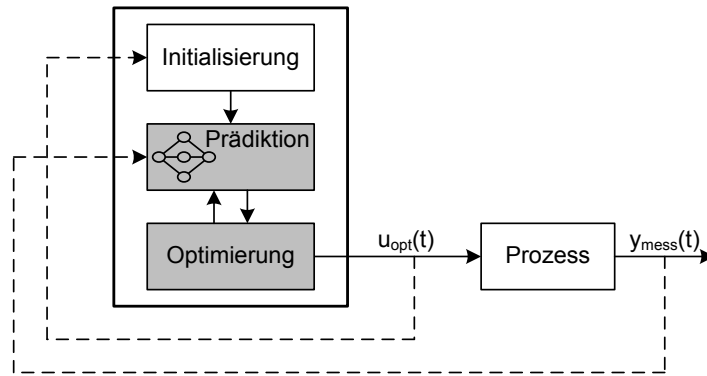


Abbildung 4.1: Prinzipieller Wirkungsplan einer neuronalen prädiktiven Regelung

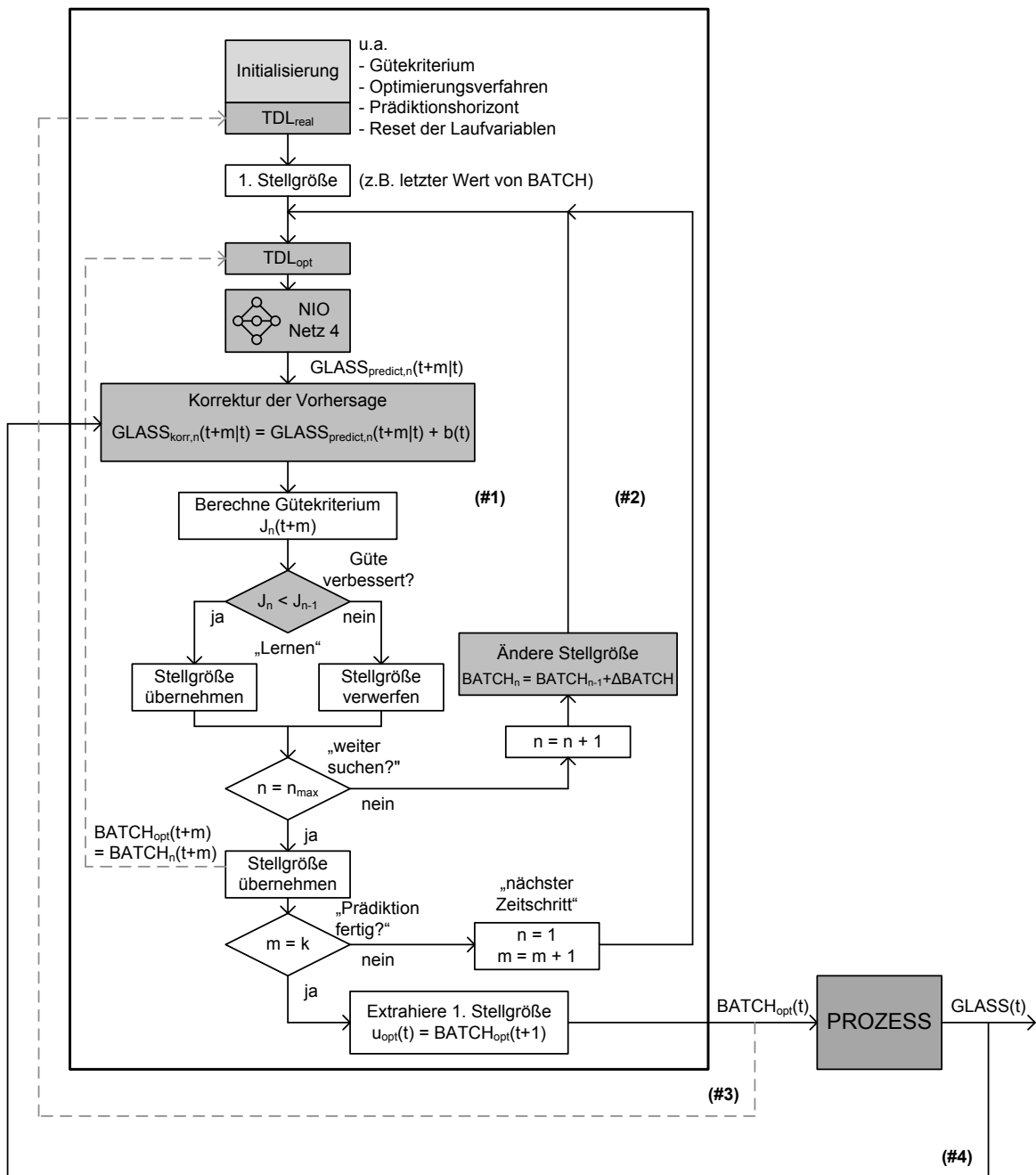


Abbildung 4.2: Neuronale prädiktive Regelstrategie am Beispiel von Netz 4

In der in Abbildung 4.2 beispielhaft dargestellten Regelstrategie wird das NIO-Netz 4 zur Prädiktion herangezogen. Dieses Netz berechnet anhand der vergangenen Messwerte der Gemeindefuhr BATCH den aktuellen Glasstand GLASS. Die Prädiktion des zukünftigen Glasstandes erfolgt über den Horizont k . Der aktuelle Zeitschritt im Prädiktionszeitraum wird mit m gekennzeichnet. In der inneren Schleife (#1) wird zu jedem Prädiktionszeitpunkt m eine optimale Stellgröße ermittelt, mit der die Regelabweichung zu diesem Zeitpunkt minimal wird. Diese Schleife wird n -mal wiederholt. Anschließend wird diese Optimierung für den nächsten Zeitpunkt $m+1$ im Prädiktionszeitraum wiederholt, solange bis der Horizont k erreicht ist. Diese wiederholte Ein-Schritt-Optimierung ist notwendig, da das neuronale Netz stets die vergangenen Eingangsgrößen zur Berechnung des neuen Ausgangswertes benötigt. Die Optimierung kann in diesem Fall als eine Kombination aus einer lernfähigen Evolutionsstrategie und einer einfachen systematischen Suche verstanden werden. Als Gütekriterium würde die Regelabweichung zwischen dem bekannten Sollwert und dem vorhergesagten Glasstand genügen, da die Optimierung zu jedem Prädiktionszeitpunkt durchgeführt wird.

Das Ergebnis der beiden Schleifen (#1 und #2) ist eine zu jedem Zeitpunkt optimale Stellfolge der Gemeindefuhr BATCH, um die berechneten zukünftigen Regelabweichungen zu minimieren. Nach der Optimierung wird lediglich der erste Wert dieser Stellfolge an den Prozess ausgegeben und die Optimierung beginnt im nächsten Abtastintervall von neuem. Um den Regelkreis zu schließen, ist es notwendig, den tatsächlichen Messwert des Glasstandes in der Prädiktion zu berücksichtigen (#4). Hier findet sich in [10] eine geeignete Möglichkeit zur Korrektur der Vorhersage. Es wird der Fehler $b(t)$ zwischen dem ein Abtastintervall zuvor vorhergesagten und dem tatsächlich gemessenen Glasstand berechnet. Dieser Fehler wird dem in der aktuellen Prädiktion ermittelten Glasstand addiert. Somit werden eventuelle Störungen zu jedem Abtastzeitpunkt berücksichtigt.

5. Zusammenfassung und Ausblick

Auf Basis eines realen Datensatzes einer Flachglasanlage wurden neuronale Netze identifiziert, die das vorhandene Prozessverhalten sehr genau nachbilden. Im Vordergrund stand die Evaluation einer geeigneten Konfiguration der Netze. Der Einfluss der individuell einstellbaren Netzparameter wurde eingehend untersucht. Von den zwölf verfügbaren Trainingsmethoden ist aufgrund der kurzen Trainingszeit und geringen Anzahl an Iterationen der Levenberg-Marquardt-Algorithmus geeignet. Als Aktivierungsfunktionen wird in beiden Schichten die lineare Funktion *purelin* empfohlen. Für die Netze 1,2,3 und 5 (siehe Abbildung 2.2) genügt ein TDL-Wert von 10. Das NIO-Netz 4 benötigt zur genauen Berechnung des zukünftigen Glasstandes einen TDL-Wert von 120. Die Anzahl der Neuronen kann bei allen Netzen auf 10 gesetzt werden. Die Standardabweichungen, die die fünf evaluierten Netze bei optimaler Konfiguration im Testzeitraum erzielen, sind sehr zufriedenstellend.

- Netz 1: $S = 0,0534K$
- Netz 2: $S = 0,0512K$
- Netz 3: $S = 0,0292mm$
- Netz 4: $S = 0,0607mm$
- Netz 5: $S = 0,0506K$

Weiterhin wurde beispielhaft eine Möglichkeit zur Nutzung der neuronalen Prozessmodelle zur prädiktiven Regelung vorgestellt. Es muss an dieser Stelle erwähnt werden, dass weiterführende Untersuchungen noch ausstehen, da die Möglichkeiten realer Tests vorerst nicht gegeben sind. Gegenwärtiger Schwerpunkt aktueller Untersuchungen sind vielmehr konzeptionelle Überlegungen zum Optimierungsverfahren. Bei einer Nutzung von NARX-Netzen, die neben den vergangenen Eingangsgrößen auch die Ausgangsgrößen heran-

ziehen, steigt zusätzlich die Komplexität. Auch hier ist noch konzeptionelle Arbeit notwendig. Um auf sich ändernde Prozessdynamiken reagieren zu können, wäre es sinnvoll, das Netz in bestimmten Intervallen neu zu trainieren. Weiterhin könnten die TDL-Werte unterschiedlich gewichtet werden, so dass näher zurückliegenden Messwerten eine höhere Bedeutung zukommt.

Literatur

- [1] H. Loch, D. Krause; "Mathematical Simulation in Glass Technology"; Springer-Verlag Berlin Heidelberg, 2002.
- [2] E. Muijsenberg; "Modeling of glass melting processes - How reliable & validated simulation tools can help to improve glass melting"; Vortrag, 11. ESG-Konferenz Maastricht, 03.-06.06.2012.
- [3] M. Handreg; "Entwurf neuronaler Netze zur Prädiktion von Prozessgrößen in einer Schmelzwanne für Floatglas"; HVG-Mitteilung Nr. 2168, 2015.
- [4] Benutzerhandbuch "Neural Network Toolbox - User's Guide", MATLAB R2014a, 2014.
- [5] N.S. Popov, D.J. Dozic, M. Stankovic, G.M. Krajoski, D. Stanisic; "Development of a Closed Loop FES System Based on NARX Radial Based Network"; IFMBE Proceedings 50, S.70ff, 2015.
- [6] E. Diaconescu; "The use of NARX Neural Networks to predict Chaotic Time Series"; WSEAS Transactions on Computer Research, Volume 3, S.182ff., 2008.
- [7] H. Yonaba, F. Anctil, V. Fortin; "Comparing Sigmoid Transfer Functions for Neural Network Multistep Ahead Streamflow Forecasting"; Journal of Hydrologic Engineering, Vol. 15, Nr. 4, S.275ff., 2010
- [8] S. Zakharian, P. Ladewig-Riebler, S. Thoer; "Neuronale Netze für Ingenieure - Arbeits- und Übungsbuch für regelungstechnische Anwendungen"; Vieweg-Verlag Braunschweig Wiesbaden, 1998.
- [9] T. Knohl; "Anwendung künstlicher neuronaler Netze zur nichtlinearen adaptiven Regelung", Fortschritt-Berichte VDI, Reihe 8, Nr. 885, 2001.
- [10] R. Dittmar, B.-M. Pfeiffer; "Modellbasierte prädiktive Regelung - Eine Einführung für Ingenieure"; Oldenbourg Wissenschaftsverlag GmbH, München, 2004.