7 Optimierung des Spaltweitenreglers mit Neuronalen Netzen

Innerhalb des vorgestellten Prozeßführungssystems sind Spaltweitenregelung und Lichtbogenerkennung die wichtigsten Komponenten. Durch diese Systemteile ist ein sicherer Prozeßverlauf gewährleistet. Zerstörungen auf den Elektrodenoberflächen können ausgeschlossen werden. Neben dem Aspekt der Sicherheit soll das Prozeßführungssystem gleichzeitig einen abtragsintensiven Betrieb anstreben. Diese Forderung kann nur erfüllt werden wenn:

- Möglichst wenig Fehlentladungen auftreten, da das gehäufte Auftreten von Lichtbögen eine Bewegungsspülung auslöst und der Prozeß dadurch für Sekunden unterbrochen wird.
- Der Leerlaufanteil nicht zu groß wird, da Leerläufe für den Abtragsprozeß einen Zeitverlust bedeuten. Dieser Zeitverlust wird zwar durch die Art der Generatorsteuerung vermindert, ist aber nicht gänzlich zu beseitigen.

Da mit Zunahme der Leerlaufimpulse die Tendenz zur Lichtbogenbildung abnimmt, sind diese Forderungen konträr. Weder Lichtbogenerkennung noch Spaltweitenregelung sind für sich genommen in der Lage, diese Ziele zu verfolgen, da beide autark voneinander arbeiten. Die Spaltweitenregelung kann einen Prozeß mit geringerem Leerlaufanteil anstreben. Es ist für den Regler jedoch nicht möglich, eine Zunahme der Anzahl der Fehlentladungen zu erkennen, da der Spaltweitenregler zur Prozeßbewertung lediglich die relativen Leerlauf- und Kurzschlußhäufigkeiten heranzieht. Die Spaltweitenregelung kann demnach nicht ihre Regelstrategie an den sich ändernden Prozeßverlauf anpassen und beispielsweise bei stark ansteigendem Lichtbogenanteil die Anzahl der tolerierbaren Leerläufe erhöhen, um damit den Prozeß zu stabilisieren. Für die Lichtbogenerkennung ist die Regelstrategie nicht relevant. Eine längere Folge von Fehlentladungen wird zu einer Bewegungsspülung führen, die den Regelvorgang unterbricht.

Um das Ziel eines abtragsintensiven Erosionsvorgangs zu erreichen, bedarf es demnach einer übergeordneten Instanz in Form einer Prozeßoptimierung. Dieses Optimiersystem hat die Aufgabe, auf der Basis einer Prozeßbeurteilung den Spaltweitenregler an den sich stetig ändernden Prozeß anzupassen, damit ein möglichst abtragsintensiver Betrieb erreicht wird. Es handelt sich bei dieser Optimierung demnach um eine *Adaptive Control Optimization* (ACO) im Sinne von [Kur72].

7.1 Konzeption der Prozeßoptimierung

Für die Gestaltung der Regleradaption bieten sich verschiedene Möglichkeiten an. In früheren Arbeiten wurden mathematische Optimierungsverfahren eingesetzt, bei denen mittels systematischer Variation die Spaltweitenregelung adaptiert wurde [Deh92]. Neuere Ansätze verwenden Methoden und Techniken aus der künstlichen Intelligenz innerhalb der Optimierungsalgorithmen.

Dabei haben sich zwei Ansätze für regelungstechnische Problemstellungen bewährt. Zum einen die Fuzzy-Logik, die bereits innerhalb der Spaltweitenregelung eingesetzt wird, zum anderen die Neuronalen Netze.

Neuronale Netze sind Modelle der Gehirnfunktion. Sie versuchen, in Struktur und Funktionsweise Gehirnzellkomplexe nachzubilden (Bild 7.1) und dadurch eine Simulation menschlicher Denkvorgänge zu erzielen. Ein Neuronales Netz besteht aus einer Vielzahl gleichartiger Elemente, die parallel und unabhängig voneinander Informationen speichern, verarbeiten und weiterleiten. Das menschliche Gehirn besteht aus etwa 10 Milliarden Nervenzellen oder Neuronen.

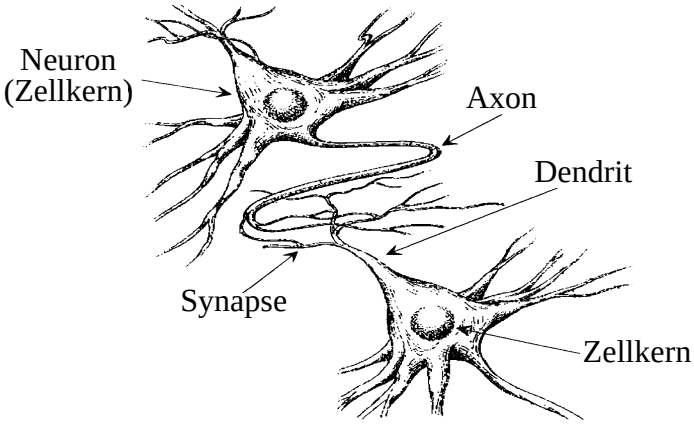


Bild 7.1: Schematische Darstellung zweier Neuronen

Eine wesentliche Eigenschaft Neuronaler Netze ist die Fähigkeit, Sachverhalte zu erlernen. Damit wird es möglich, Systeme zu realisieren, die sich leicht an veränderte Situationen anpassen lassen. In Tabelle 7.1 werden Fuzzy Logik und Neuronale Netze im Hinblick auf ihre Anwendbarkeit zur Adaption des Spaltweitenreglers verglichen.

Fuzzy Logik	Neuronale Netze
Regelwissen muß verfügbar sein	Regelwissen nicht notwendig
aufwendiges „Tuning“ der Regelbasis und Zugehörigkeitsfunktionen	„Tuning“ entfällt, dafür Training des Netzes, wobei Lernvorgang nicht garantiert konvergiert.
nicht lernfähig	lernfähig

Tabelle 7.1: Vergleich von Fuzzy-Logik und Neuronalen Netzen

Der Vergleich der beiden Ansätze ergibt, daß ein Neuronales Netz zur Adaption des Spaltweitenreglers Vorteile gegenüber einer Fuzzy-Adaption aufweist. So ist zu der Adaptionsaufgabe kein Apriori Wissen vorhanden, wie es für die Entwicklung eines Fuzzy-Systems notwendig ist. Hier bietet ein Neuronales Netz Vorteile, da es lediglich Daten benötigt, die es ihm ermöglichen, sich an die Adaptionsaufgabe anzupassen. Dies geschieht in einer der Anwendung vorangestellten Lernphase. Durch diese Eigenschaft wird es möglich, die Optimierungsstrategie auf einfache Weise zu verbessern oder zu erweitern, indem die dem Neuronalen Netz in der Lernphase präsentierten Daten verändert werden.

Für das System zur Regelung der Spaltweite bietet sich demnach die Kombination eines Fuzzy-Spaltweitenreglers mit einer Adaption auf der Basis eines Neuronalen Netzes an. Solche kombinierten Systeme werden unter dem Oberbegriff *Neuro-Fuzzy* zusammengefaßt. Die Art und Weise, wie diese Kombination erfolgt, erlaubt eine genauere Differenzierung. So unterscheidet [NKR94] *hybride Neuro-Fuzzy-Regler* und *kooperativer Neuro-Fuzzy-Regler*.

Bei *hybriden Neuro-Fuzzy-Reglern* liegt eine einheitliche Architektur vor, die sich meist an die Struktur eines Neuronalen Netzes anlehnt. Dazu kann entweder ein Fuzzy-Regler als spezielles Neuronales Netz interpretiert oder mit Hilfe eines Neuronalen Netzes implementiert werden.

Bei einem *kooperativen Neuro-Fuzzy-Regler* arbeiten Neuronales Netz und Fuzzy-Regler grundsätzlich unabhängig voneinander. Die Kopplung besteht darin, daß einige Parameter des Fuzzy-Reglers vom Neuronalen Netz optimiert werden. Bei dem in dieser Arbeit entwickelten Neuro-Fuzzy-System wird eine solche Art der Kombination angewendet.

Ein wichtiges Unterscheidungsmerkmal zwischen kooperativen Neuro-Fuzzy-Systemen ist der Zeitpunkt, zu dem die Optimierung des Fuzzy-Reglers erfolgt. So ist es möglich, den Fuzzy-Regler vor dem eigentlichen Betrieb, also *offline*, durch ein Neuronales Netz zu erzeugen oder anzupassen. Dies wird von einigen Entwurfs-Werkzeugen für Fuzzy-Systeme bereits automatisiert unterstützt. Der Benutzer braucht dem Entwurfs-Werkzeug lediglich Daten zur Verfügung zu stellen, die das gewünschte Regler-Verhalten wiedergeben. Das Werkzeug erzeugt daraufhin automatisiert den Fuzzy-Regler mit sämtlichen charakteristischen Parametern [Kos92]. Innerhalb der Funkenerosion wird in [Raa99] ein solcher Ansatz verfolgt. Um während der Bearbeitung auf sich ändernde Prozeßbedingungen zu reagieren, muß die Anpassung des Spaltweitenreglers jedoch während des Erosionsvorganges, also *online*, durchgeführt werden.

Adaptiert das Neuronale Netz während des Bearbeitungsvorgangs den Fuzzy-Regler, so wird es damit zum permanent aktiven Bestandteil des Prozeßführungssystems. Eine Optimierung kann jederzeit durch sich verschlechternde Prozeßbedingungen ausgelöst werden und zu einer Anpassung des Fuzzy-Spaltweitenreglers durch das Neuronale Netz führen. Ein solcher Ansatz wurde für das Optimierungssystem innerhalb dieser Arbeit gewählt.

7.2 Funktionsweise und Anwendung künstlicher Neuronaler Netze

Künstliche Neuronale Netze (nachfolgend Neuronale Netze) basieren auf einem vereinfachten datentechnischen Verarbeitungsmodell der biologischen Neuronen in den Gehirnzellen. Jedes einzelne dieser biologischen Neuronen wird dabei durch einen Summenverstärker mit Schwellwertschaltung nachgebildet.

Grundelement des in dieser Arbeit verwendeten Neuronalen Netzes ist das sog. *Perceptron* [WiHo60]. Dabei handelt es sich um einen Netzknoten, der über zahlreiche Eingänge (a_1, ... a_n) verfügt, die jeweils mit Gewichten gekoppelt sind (w_1, ... w_n).

Die Berechnung der Netzaktivität *net* erfolgt durch multiplikative Verknüpfung des Eingangsvektors mit dem aus den Gewichten gebildeten Vektor. Dabei wird ein Schwellwert θ mit einbezogen (Bild 7.2).

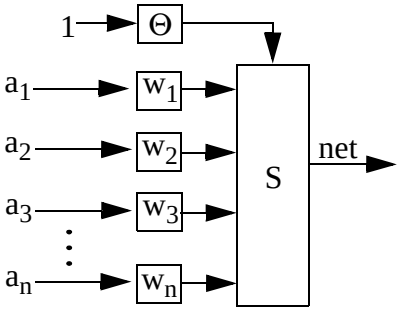
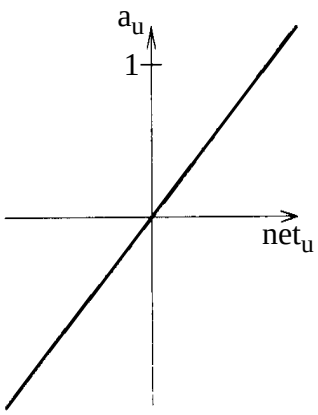


Bild 7.2: Modell der Netzaktivität eines Perceptrons

Die Gesamtbeziehung zur Berechnung der Netzaktivität eines Neurons ist somit:

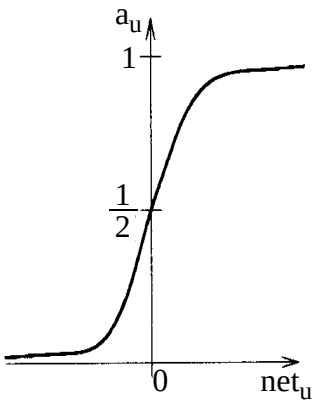
$$net = \sum_{i=1}^{n} a_i w_i + \Theta \qquad \text{(Gl 7.1)}$$

Die Ausgabe (a_u) des Perceptrons kann in Abhängigkeit von der Netzaktivität *net*, je nach Netzmodell, durch verschiedene Ausgabefunktionen erfolgen. Eingesetzt wurden die *Lineare*- und die *Sigmoide*-Ausgabefunktion (Bild 7.3).



Lineare Funktion

$a_u = c_u * net_u$



Sigmoide Funktion

$a_u = \frac{1}{1+e^{-net_u}}$

Bild 7.3: Ausgabefunktionen für Neuronale Netze

Ein Neuronales Netz besteht aus einer Vielzahl derartiger Perceptronen (=Neuronen), deren Ein- und Ausgänge miteinander verknüpft sind. Eine häufig verwendete Netz-Topologie ordnet die Neuronen in Schichten an. Die Ausgänge der Neuronen einer Schicht bilden dabei die Eingänge jedes Neurons in der Folgeschicht (Bild 7.4). Derartige Netz-Topologien werden auch als *Feed-Forward-Netze* bezeichnet.

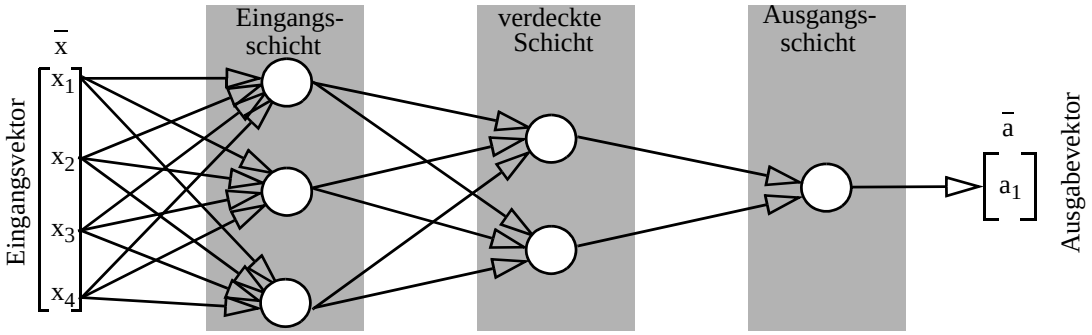


Bild 7.4: Schematischer Aufbau eines mehrschichtigen Feed-Forward Netzwerkes

Je nach Verbindung der Neuronen zur Netzwerkumgebung werden die Schichten unterschiedlich bezeichnet. So gibt es eine Eingangsschicht zur Aufnahme des Eingabevektors $\bar{x}$. Die weitere Informationsverarbeitung findet in den Neuronen der verdeckten Schicht(en) und der Ausgabeschicht statt. Die Anzahl der Neuronen in der Eingabeschicht und in den verdeckten Schichten ist, abhängig von der Komplexität der zu bearbeitenden Aufgabe, beliebig wählbar. Die Ergebnisse der Neuronen in der Ausgabeschicht repräsentieren den Ausgabevektor $\bar{a}$. Damit ist die Anzahl der Neuronen in der Ausgangsschicht auf die Dimension des Ausgangsvektors festgelegt.

Ein Neuronales Netz ist in der Lage, beliebige nichtlineare Zusammenhänge zwischen Eingangs- und Ausgangsdaten zu erlernen. Dazu müssen für jedes Perceptron im Netz die Gewichte $w_1, \ldots, w_n$ sowie der Schwellwert θ angepaßt werden. Der Vorgang der Gewichtsanpassung wird als Training oder Lernen des Netzes bezeichnet. Dem Neuronalen Netz werden dabei Eingaben präsentiert, um dann die Gewichte der einzelnen Neuronen entsprechend dem Fehler, also der Differenz zwischen der gewollten und der tatsächlichen Ausgabe, anzupassen. Für mehrschichtige Feed-Forward-Netze wird am häufigsten der *Backpropagation*-Lernalgorithmus angewendet [RHW86]. Bei diesem Algorithmus wird das Fehlersignal rückwärts von der Ausgabeschicht hin zur Eingabeschicht propagiert. Der grundlegende Ansatz besteht darin, die Verbindungsgewichte des Netzwerkes nach den Regeln eines Gradientenabstiegs so zu verändern, daß für einen Eingabevektor $\bar{x}_j$ die Abweichungen zwischen dem vom Netzwerk berechneten Ausgabevektor $\bar{a}_j$ und dem gewünschten Ausgabevektor $\bar{y}_j$ minimal werden.

Als Fehlermaß E wird die Summe der quadratischen Abweichungen zwischen berechneter und gewünschter Ausgabe definiert. Das Lernziel besteht darin, den über alle Trainingsbeispiele summierten Fehler E zu minimieren (Gl 7.2).

$$E = \sum_{i=1}^{az} (y_j - a_j)^2 \rightarrow Min \qquad \text{(Gl 7.2)}$$

y_j : gewünschter Ausgabevektor zum Trainingsbeispiel j
a_j : berechneter Ausgabevektor zum Trainingsbeispiel j
az: Anzahl der Trainingsbeispiele

Die Lernaufgabe wird beim Backpropagation-Algorithmus durch zyklische Wiederholung der folgenden Schritte gelöst:

1. Berechnen von E durch Anlegen aller Trainingsbeispiele
2. Bestimmen des Gradienten durch Analyse des Fehlers
3. Verändern der Netzwerkgewichte in Gradientenrichtung

Das Verfahren wird abgebrochen, wenn E einen vorgegebenen Wert unterschreitet oder eine bestimmte Anzahl von Lernschritten erreicht ist.

Gegenüber anderen Verfahren der künstlichen Intelligenz weisen Neuronale Netze folgende Vorteile auf, die in der beabsichtigten Anwendung innerhalb der Regleradaption zum Tragen kommen:

- Lernfähigkeit
  Durch die während der Lernphase vorgenommene Anpassung der Gewichtsfaktoren wird eine explizite algorithmische Lösung des Problems umgangen.
- Fähigkeit zur Generalisierung
  Einem trainierten Neuronalen Netz können auch Eingabemuster präsentiert werden, die außerhalb der bekannten Lernmenge liegen. Das Netz wird dann, von seinen bekannten Datenmustern ausgehend, Ausgaben für die unbekannten Daten erzeugen, die den erlernten Ausgaben ähneln.
- Wissensspeicherung
  Informationen sind nicht an spezifischen Speicherplätzen (Adressen) lokalisiert, sondern als Muster über das gesamte Netzwerk verteilt (distributive Wissensspeicherung).

Ein wichtiges Einsatzgebiet Neuronaler Netze besteht in der Approximation unbekannter Funktionen. Dies wird durch die Fähigkeit erreicht, Zusammenhänge zwischen Ein- und Ausgängen (Übertragungsfunktionen) durch Trainingsdaten (Stützstellen) zu erlernen.

Die Mustererkennung ist ein weiteres Anwendungsgebiet Neuronaler Netze. Die Aufgabe besteht darin, die Muster der Meßdaten, die den Merkmalsraum R^n aufspannen, einem Element des Entscheidungsraumes R^m zuzuordnen. Die Entscheidung wird zumeist umgesetzt, indem die zu einer Messung gehörenden Daten einer bestimmten Klasse von Zuständen zugeordnet werden (Klassifikation). Dieses Anwendungsgebiet ist auf die zu lösende Problemstellung der Adaption des Spaltweitenreglers übertragbar.

7.3 Anwendung von Neuronalen Netzen zur Adaption des Spaltweitenreglers

Das Prinzip der Adaption des Spaltweitenreglers durch ein Neuronales Netz beruht darauf, charakteristische Prozeßzustände zu identifizieren und geeignete Adaptionsmaßnahmen durchzuführen. Das Neuronale Netz wird dabei im Sinne einer Mustererkennung eingesetzt.

Dem Netz werden während der Trainingsphase zahlreiche Adaptionsmaßnahmen präsentiert, die jeweils für bestimmte Prozeßzustände eine Verbesserung im Sinne der Zielkriterien erreichen konnten. In der Anwendungsphase ist das Netz dann in der Lage, auftretende Prozeßzustände wiederzuerkennen. Da der Merkmalsraum groß ist, werden während des Bearbeitungsvorgangs in der Regel andere Prozeßzustände auftreten als die in der Trainingsphase erlernten. Durch die Fähigkeit zur Generalisierung ist das Neuronale Netz jedoch in der Lage, Ähnlichkeiten zu den erlernten Prozeßzuständen zu erkennen. Dadurch werden Ausgaben generiert, die ebenfalls den erlernten Ausgaben ähneln.

Während des Erosionsvorgangs ist die Regleradaption nicht den engen zeitlichen Randbedingungen unterworfen, die der Erosionsprozeß aufgrund seiner Impulsperiode vorgibt. Für die schnelle Reaktion auf Prozeßereignisse sind Lichtbogenerkennung und Spaltweitenregelung verantwortlich. Aufgabe der Regleradaption ist es, Trends in der Entwicklung der Prozeßstabilität zu erkennen und den Spaltweitenregler entsprechend anzupassen. Um den Einfluß kurzfristiger Schwankungen in den Bewertungsgrößen zu reduzieren, werden die Eingangsgrößen über ein längeres Zeitintervall betrachtet und dann Mittelwerte herangezogen.

Zur Adaption des Fuzzy-Spaltweitenreglers bieten sich unterschiedliche Möglichkeiten an. Je tiefer in die Struktur des Fuzzy-Reglers eingegriffen wird, desto mehr Einstellparameter müssen von einer Optimierung angepaßt werden. Tiefe Eingriffe in die Reglerstruktur erlauben zwar größte Freiheitsgrade bei der Anpassung, erhöhen dafür jedoch die Komplexität der Optimierung. Tabelle 7.2 zeigt die verschiedenen Eingriffsmöglichkeiten auf und bewertet diese.

Möglichkeit zur Regleradaption	Bewertung
Änderung der Zugehörigkeitsfunktionen	• Sehr viele Einstellparameter, die von der Optimierung geändert werden müssen • Wirkungsweise der Optimierung schwer verständlich
Änderung der Regelbasis	• Sehr viele Einstellparameter, die von der Optimierung geändert werden müssen • Regler kann in weiten Grenzen verändert werden
Änderung der Skalierungsfaktoren am Reglerausgang	• Änderungsmöglichkeiten am Regler sind begrenzt • Wenige Einstellparameter • Wirkungsweise der Optimierung gut verständlich

Tabelle 7.2: Bewertung der Möglichkeiten zur Adaption des Fuzzy-Reglers

Aus der Bewertung geht hervor, daß eine Adaption der Skalierungsfaktoren am Regler-Ausgang die aussichtsreichste Variante ist. Alle anderen Möglichkeiten erfordern von der Adaption die Änderung sehr vieler Einstellparameter, wodurch die Komplexität des Adaptionsalgorithmus beträchtlich ansteigt. Durch die Anpassung der Zugehörigkeitsfunktionen oder der Regelbasis würde die Struktur des Reglers gänzlich verändert. Diese Struktur wurde bereits bei der Auslegung des Spaltweitenreglers (Kapitel 6) im Rahmen von Abtragsversuchen ermittelt. Dadurch läßt sich abschätzen, daß eine Änderung der Reglerstruktur kein größeres Verbesserungspotential bietet als eine Anpassung der Skalierungsfaktoren am Reglerausgang.

Mit der Erläuterung des Spaltweitenreglers wurden die Skalierungsfaktoren am Regler-Ausgang bereits eingeführt (siehe Bild 6.20). Diese Faktoren können getrennt für die Zustell- und Rückzugsbewegung eingestellt werden (S_V und S_R) und dienen zur Verstärkung der Ausgangsgröße des Fuzzy-Reglers.

Nachdem die Zielparameter der Prozeßoptimierung feststehen, gilt es, Meßgrößen festzulegen, die den Prozeßverlauf möglichst zuverlässig im Hinblick auf Stabilität und Abtragsleistung beschreiben. Folgende Bewertungsgrößen stehen zur Verfügung:

- Relative Kurzschlußhäufigkeit
- Relative Leerlaufhäufigkeit
- Relative Lichtbogenhäufigkeit
- Bahnfortschritt (Arbeitsfortschritt)
- Von der Zündverzögerungszeit abgeleitete Größen

Die Zündverzögerungszeit oder von ihr abgeleitete Größen wurden aufgrund der reduzierten Aussagekraft im Zusammenhang mit der eingesetzten Generatortechnologie nicht als Bewertungsgröße eingesetzt (siehe Kapitel 6).

Eine Verwendung des Bahnfortschritts ist ebenfalls problematisch, da dieser auch von der Werkzeugelektrodengeometrie abhängig ist. So steigt beispielsweise bei keilförmiger Werkzeugelektrode die im Eingriff befindliche Fläche mit zunehmender Einsenktiefe an. Dadurch wird sich der Bahnfortschritt verlangsamen, obwohl die Prozeßbedingungen sich nicht verschlechtert haben.

Die Prozeßoptimierung erhält deshalb als Bewertungsgrößen die relativen Kurzschluß-, Leerlauf- und Lichtbogenhäufigkeiten. Außerdem dienen die aktuellen Einstellwerte der Skalierungsfaktoren S_V und S_R als Eingangsdaten. Durch diese Rückführung wird erreicht, daß zukünftige Adaptionen auf Basis der aktuellen Einstellwerte vorgenommen werden. Es ergibt sich ein Neuronales Netz, wie es in Bild 7.5 skizziert ist.

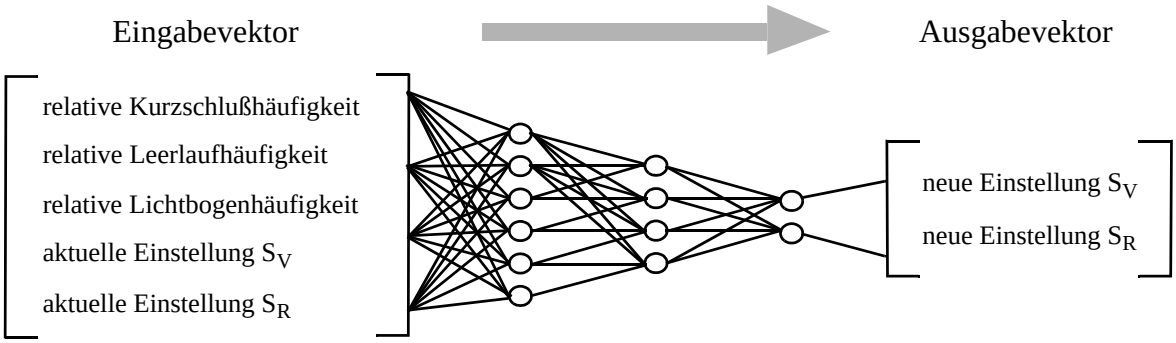
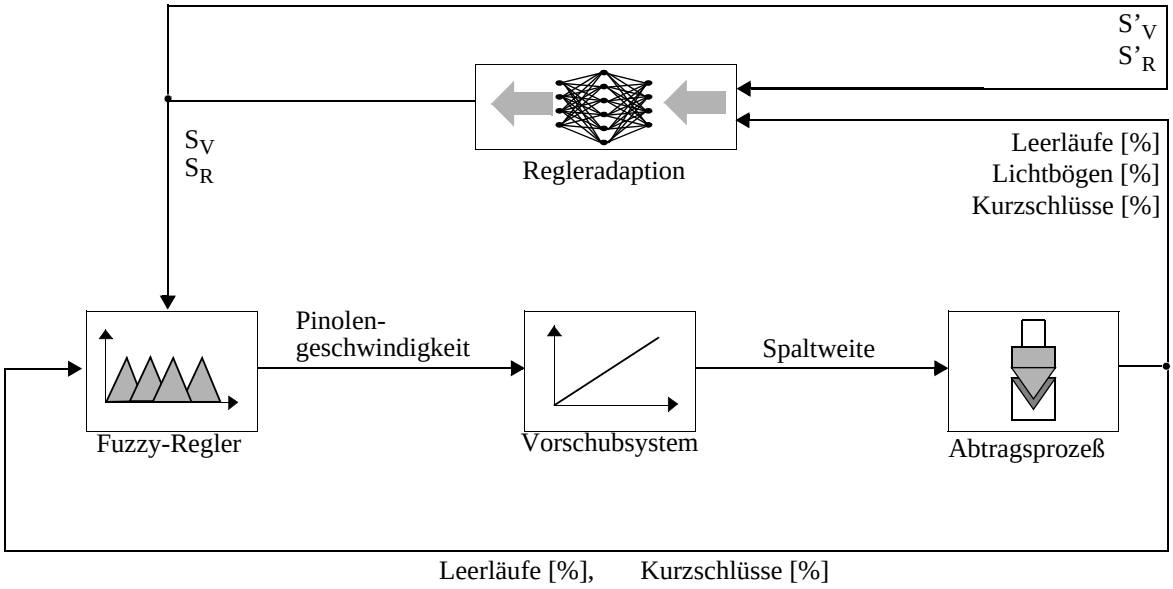


Bild 7.5: Schematische Darstellung des Neuronalen Netzes zur Spaltweitenregler-Adaption

Bild 7.6 zeigt die Spaltweitenregelung und die Regleradaption im Blockdiagramm.



Legende: S_R : Rückwärts-Skalierung S'_R : Letzte Einstellung Rückwärts-Skalierung
S_V : Vorwärts-Skalierung S'_V : Letzte Einstellung Vorwärts-Skalierung

Bild 7.6: Blockdiagramm der Regleradaption

7.4 Die Wirkung der Skalierungsfaktoren auf den Erosionsprozeß

Um die Einflußmöglichkeiten einer Adaption abzuschätzen, die den Spaltweitenregler durch Änderung von Skalierungsfaktoren am Reglerausgang anpaßt, muß die Wirkung dieser Skalierungsfaktoren auf den Erosionsprozeß untersucht werden. Dieser wird erkennbar durch eine systematische Variation der Skalierungsfaktoren während des Erosionsvorgangs und bei gleichzeitiger Aufzeichnung der Lichtbogen-, Kurzschluß- und Leerlaufhäufigkeiten. Bild 7.7 zeigt das Prozeßverhalten bei konstantem Rückwärts-Skalierungsfaktor S_R und veränderter Vorwärts-Skalierung S_V.

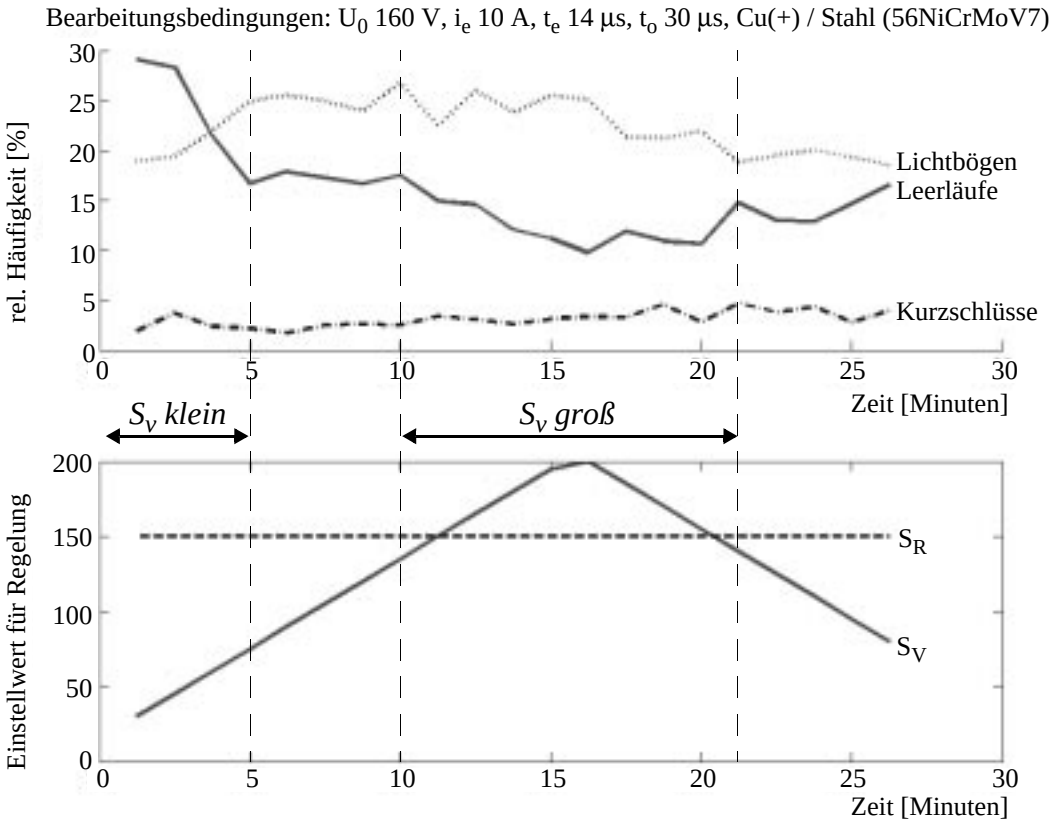


Bild 7.7: Einfluß der Vorwärts-Skalierung S_V auf den Prozeßverlauf

Aus Bild 7.7 geht hervor, daß die Leerlaufhäufigkeit am stärksten von der Vorwärts-Skalierung S_V beeinflußt wird. So führt eine geringe Vorwärts-Skalierung zu einem Anstieg des Leerlaufanteils. Auch die Anzahl der beobachteten Fehlentladungen wird durch die Vorwärts-Skalierung beeinträchtigt. Ein erhöhter Leerlaufanteil stellt sich bei vergrößerter Erosionsspaltweite ein. In diesem Arbeitsbereich sinkt die Gefahr der Lichtbogenbildung und damit die Anzahl der Fehlentladungen. Umgekehrt führt eine erhöhte Vorwärts-Skalierung S_V zu einem abnehmenden Leerlauf- und steigenden Lichtbogenanteil. Die Anzahl der Kurzschlüsse ist dagegen weitgehend unbeeinflußt von der Vorwärts-Skalierung S_V.

Den Einfluß der Rückwärts-Skalierung S_R zeigt ein Versuch mit konstanter Vorwärts-Skalierung und variiertem S_R. Der Verlauf der Leerlauf-, Kurzschluß und Lichtbogenhäufigkeiten ist für diesen Fall in Bild 7.8 dargestellt.

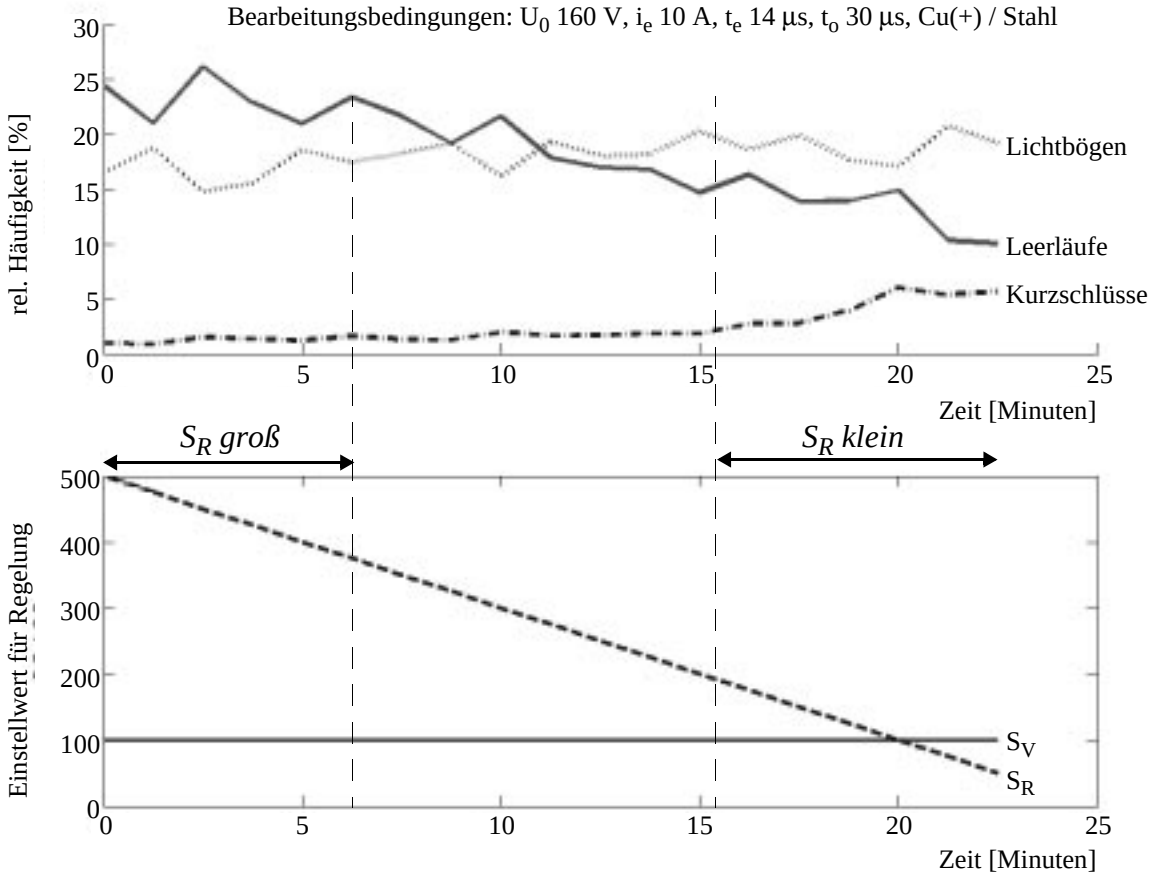


Bild 7.8: Einfluß der Rückwärts-Skalierung S_R auf den Prozeßverlauf

Anhand von Bild 7.8 ist zu erkennen, daß die Rückwärts-Skalierung S_R hauptsächlich auf die Kurzschluß- und Leerlaufhäufigkeiten einwirkt. Da bei dem verwendeten Spaltweitenregler die Rückwärts-Skalierung ausschließlich bei zu hohem Kurzschlußanteil die Bewegung der Pinole beeinflußt, sorgen verringerte S_R-Werte für einen langsameren Rückzug im Kurzschlußfall. Eine Brückenbildung, die zu einem Elektrodenkurzschluß führte, bleibt auf diese Weise länger bestehen.

Eine hohe Rückwärts-Skalierung S_R läßt die Pinole bei Kurzschluß schnell zurückziehen. Der Kurzschlußanteil nimmt auf diese Weise ab. Allerdings führt ein hohes S_R zu einem vermehrten Leerlaufanteil. Die Anzahl der Fehlentladungen wird durch S_R nur in geringem Maße beeinflußt.

Insgesamt zeigen die Versuche, daß es durch Änderung der Skalierungsfaktoren S_V und S_R möglich ist, den Prozeß in der gewünschten Weise zu beeinflussen. Leerlauf- Kurzschluß- und Lichtbogenhäufigkeiten können reduziert werden, um die Abtragsleistung zu steigern.

7.5 Gewinnung und Bereitstellung von Trainingsdaten

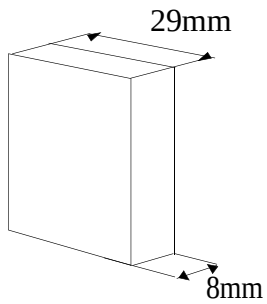
Die Genauigkeit, mit der das Neuronale Netz die Eingangsgrößen auf die Ausgangsgrößen abbildet, wird maßgeblich von der Qualität der eingehenden Informationen bestimmt. Aufgrund der Vielzahl möglicher Eingaben ist es unmöglich, das Neuronale Netz mit der Gesamtheit aller denkbaren Informationen zu trainieren. Für die Bereitstellung der Lernbeispiele sind Abtragsversuche notwendig. Um den Aufwand zu reduzieren, werden nur begrenzt Trainingsdaten bereitgestellt. Außerdem erhöhen umfangreiche Trainingsdaten die in der Lernphase notwendige Rechenzeit. Andererseits erhöht ein eventuell vorhandener Anteil redundanter Informationen in den Trainingsdaten die Robustheit des Gesamtsystems.

Das Ziel muß es daher sein, dem Neuronalen Netz diejenigen Daten zur Verfügung zu stellen, die zum Erlernen der Adaptionsaufgabe am besten geeignet sind. Diese Daten entsprechen Adaptionsmaßnahmen, mit denen, ausgehend von einem bestimmten Prozeßzustand und einer definierten Einstellung des Spaltweitenreglers, eine Verbesserung im Prozeßverhalten erzielt werden konnte. Die Aufgabe des Neuronalen Netzes ist es, einen abtragsintensiven Erodierbetrieb mit geringem Anteil an Fehlentladungen, Leerläufen und Kurzschlüssen einzustellen. Bezogen auf die Eingangsdaten des Neuronalen Netzes bedeutet dies eine Abnahme der Kurzschlüsse, Leerläufe und Lichtbögen.

Um erfolgreiche Adaptionsmaßnahmen zu gewinnen, wurden anhand einer Standard-Erosionsaufgabe bei unterschiedlichen Maschineneinstellungen Abtragsversuche durchgeführt. Bild 7.9 zeigt die Versuchsbedingungen.

Maschineneinstellungen:

U_0: 160 V	i_e: 12 A	t_e:15 µs	t_0:30 µs
U_0: 160 V	i_e: 10 A	t_e:14 µs	t_0:30 µs
U_0: 160 V	i_e: 8 A	t_e:12 µs	t_0:30 µs
U_0: 200 V	i_e: 6 A	t_e:10 µs	t_0:30 µs



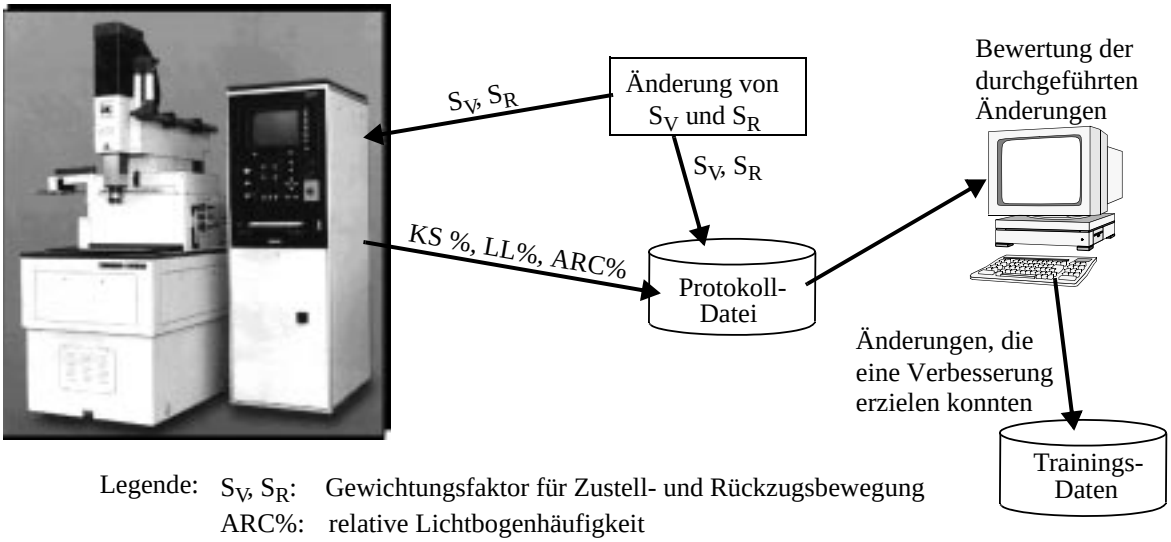
Elektrode: E-Cu (+)

Werkstück: 56NiCrMoV7

Bild 7.9: Erosionsversuche zur Gewinnung von Trainingsdaten

Während dieser Versuche wurden durch einen Bediener Adaptionsmaßnahmen am Spaltweitenregler durchgeführt. Die fortlaufende Protokollierung der relativen Kurzschluß-, Leerlauf- und Lichtbogen-Häufigkeiten während der gesamten Versuchsdauer ermöglichte die spätere Auswertung der Adaptionen.

Mit Hilfe einer anschließenden Analyse der Protokolldaten konnten die Parameteränderungen identifiziert werden, die den Prozeßverlauf im Sinne der Zielkriterien positiv beeinflußt haben. Bild 7.10 zeigt den beschriebenen Ablauf.



Legende: S_V, S_R: Gewichtungsfaktor für Zustell- und Rückzugsbewegung
ARC%: relative Lichtbogenhäufigkeit
LL%: relative Leerlaufhäufigkeit
KS%: relative Kurzschlußhäufigkeit

Bild 7.10: Verfahrensablauf bei der Gewinnung von Trainingsdaten

Während der Versuche zur Gewinnung von Trainingsdaten konnten einige Einstellregeln identifiziert werden, die in bestimmten Prozeßsituationen (Leerlauf-, Kurzschluß- und Lichtbogenhäufigkeiten) einen positiven Einfluß auf den Bearbeitungsverlauf ausüben konnten. In Tabelle 7.3 sind diese Einstellregeln aufgeführt.

Prozeßzustand	Adaptionsmaßnahme
erhöhter Kurzschlußanteil	S_R vergrößern
erhöhter Leerlaufanteil	S_V vergrößern, S_R verkleinern
erhöhter Lichtbogenanteil	S_V verkleinern

Tabelle 7.3: Wirkung der Trainingsdaten auf die Zielgrößen

Diese Adaptions-Strategien fanden in Form erfolgreicher Adaptionsmaßnahmen ihren Niederschlag in den Trainingsdaten für das Neuronale Netz. Da der Mindestumfang der notwendigen Lernstichproben nur schwer abschätzbar ist, erfolgte die Ermittlung der erforderlichen Beispielanzahl experimentell.

Um mit einem neuronalen Netz eine Identifikationsaufgabe befriedigend zu lösen, wird in [Mül96] eine Lernmenge empfohlen, die mindestens dreimal so groß ist, wie die Summe der Ein- und Ausgänge des Neuronalen Netzes (21 im vorliegenden Fall). Durch eigene Experimente mit unterschiedlichen Trainingsdaten konnte für das zur Regleradaption einzusetzende Neuronale Netz eine Lernstichprobe mit einer Größe von 73 Datensätzen als ausreichend ermittelt werden.

7.6 Konzeption und Training des Netzwerkes

Im Rahmen der Netzwerkkonzeption erfolgt die Festlegung des Neuronenmodells und der Netzwerkarchitektur. Für das verwendete Feed-Forward-Netzwerk bedeutet die Bestimmung der Topologie im wesentlichen die Definition der Netzschichten. Als Ausgabefunktion wurde für die Neuronen der verdeckten Schicht die *Sigmoide*-Funktion (siehe Bild 7.3) verwendet. Diese Funktion ist vorteilhaft, wenn das Neuronale Netz nichtlineare Zusammenhänge erlernen soll. Das Ergebnis der *Sigmoiden*-Funktion ist auf den Wertebereich 0 bis 1 beschränkt. Deshalb kann diese Funktion nicht für die Neuronen der Ausgabeschicht eingesetzt werden, da die Ausgaben des Netzwerkes in einem größeren Wertebereich liegen müssen (S_R liegt zwischen 50 und 500, S_V liegt zwischen 50 und 200). Damit das Neuronale Netz auch Ausgaben in einem größeren Wertebereich erzeugen kann, wurde innerhalb der Neuronen der Ausgangsschicht die lineare Ausgabefunktion eingesetzt (siehe Bild 7.3).

Um die Anzahl der Neuronen in der verdeckten Schicht möglichst optimal festzulegen, sind Kenntnisse des Lernverhaltens von Backpropagation-Netzwerken notwendig. Bild 7.11 zeigt den typischen Verlauf eines Lernvorganges. Dabei ist der Fehler zu erkennen, den das Neuronale Netz bei Eingabe der Trainingsdaten erzeugt (Trainingsfehler). Außerdem ist in Bild 7.11 der Fehler dargestellt, den das Neuronale Netz bei der Präsentation von unbekannten Daten erzeugt (Testfehler). Die Anwendung einer dem Neuronalen Netz unbekannten Datenmenge (Teststichprobe) hat sich durchgesetzt, um den Lernerfolg zu überprüfen. Durch eine derartige Kreuzvalidierung kann die Fähigkeit des Netzes zur Generalisierung beurteilt werden. Da das Neuronale Netz in der praktischen Anwendung ständig mit Eingangsdaten konfrontiert wird, die nicht der ursprünglich trainierten Datenmenge entstammen, ist die Fähigkeit zur Abstraktion wesentlich für die Einsatzbereitschaft des Netzes. Aus diesem Grund wurde neben den Datensätzen zum Training des Neuronalen Netzes mit dem gleichen Verfahren auch eine zweite Datensammlung ermittelt, die als Teststichprobe zur Prüfung dient.

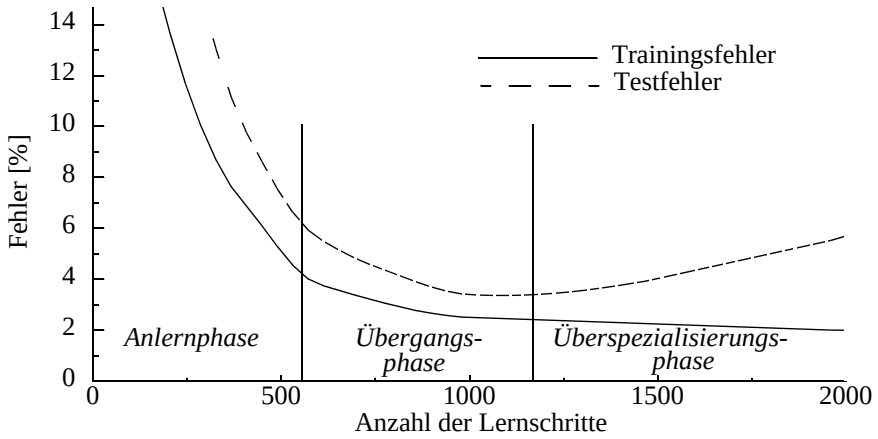


Bild 7.11: Verlauf des Trainings- und Testfehlers bei Variation der Anzahl der Lernschritte

Anhand von Bild 7.11 ist erkennbar, daß der Fehler in der *Anlernphase* stark abnimmt. In der *Übergangsphase* stagniert die Verbesserung des Fehlers. Bei weiterer Anwendung des Lernalgorithmus verschlechtert sich in der *Überspezialisierungsphase* der Fehler bei der Teststichprobe. Das Neuronale Netz verliert in dieser Phase die Fähigkeit zur Generalisierung, da es sich zufällige Gemeinsamkeiten der Lernstichprobe einprägt.

Das Minimum der Fehlerrate über die Anzahl der Lernschritte ist somit ein aussagekräftiges Gütekriterium für den Erfolg des Lernvorgangs. Eine 100prozentige Wiedererkennung der Trainingsdaten wird in der Regel nie erreicht und ist in der Anwendung auch nicht erwünscht.

Die Auswahl der für die Anwendung optimalen Netz-Topologie ist aufgrund der großen Variationsmöglichkeiten schwierig. In der praktischen Anwendung werden deshalb automatische Suchalgorithmen für diese Aufgabe eingesetzt. Eine Auswahl kann beispielsweise durch Selektion auf der Basis der natürlichen Evolution [Rech73] erfolgen. Das verwendete Werkzeug zur Konstruktion und Simulation der Neuronalen Netzwerke (Matlab, Neural Network Toolbox, Version 6, Release 12) bietet keine automatisierte Netzauswahl.

Durch Versuche mit einem neuronalen Netz, das nur über eine verdeckte Schicht verfügt, konnte aus den Trainingsdaten mit geringem Fehler die gewünschte Funktion approximiert werden. Bild 7.12 zeigt den Trainingsfehler sowie den Fehler bei der Teststichprobe bei einem derartig gestalteten Neuronalen Netz mit unterschiedlichen Neuronenzahlen in der verdeckten Schicht.

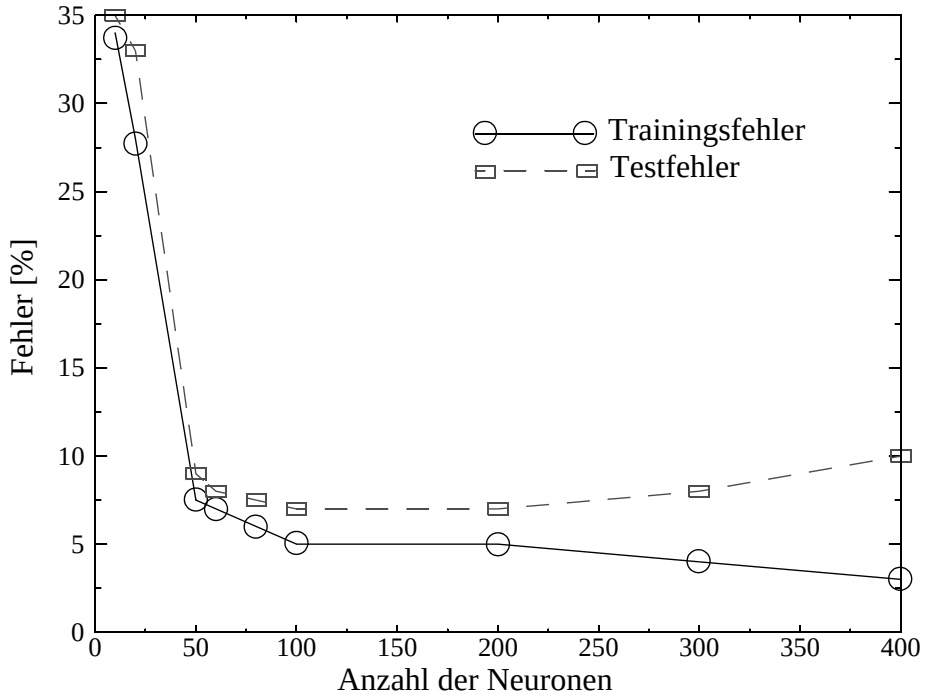


Bild 7.12: Lern- und Testfehler eines Neuronalen Netzes mit einer verdeckten Schicht und unterschiedlichen Neuronenzahlen

Anhand von Bild 7.12 wird deutlich, daß bei diesem Netzwerk mindestens 50 Neuronen notwendig sind, um einen Lernerfolg zu erzielen. Wird die Anzahl der Neuronen auf über 200 ausgedehnt, dann verliert das Netz zunehmend die Fähigkeit zur Generalisierung. Damit die Rechenzeit zur Generierung der Netzausgabe reduziert werden kann und gleichzeitig ein geringer Fehler sichergestellt wird, bietet sich ein Netz mit 80 Neuronen in der verdeckten Schicht an.

7.7 Praktischer Einsatz der Prozeßoptimierung

Um das mittels Matlab trainierte Neuronale Netz als Teil des Prozeßführungssystems während der funkenerosiven Bearbeitung einsetzen zu können, sind weitere Schritte notwendig. Zunächst wird das trainierte Neuronale Netz in einer Datei abgelegt. Für jedes Perceptron innerhalb des Neuronalen Netzes werden die durch das Training angepaßten Eingangsgewichte sowie der Schwellwert (siehe Bild 7.2) in Form einer Matrix gespeichert.

Ein Programm-Modul des Prozeßführungssystems lädt diese Datei in den Speicher des Prozeßrechners. Der Algorithmus zur Berechnung der Ausgänge des Neuronalen Netzes wurde innerhalb des Prozeßführungssystems implementiert. Diese Berechnung wird zyklisch während der Bearbeitung durchgeführt. Als Eingangsdaten werden die gemittelten Prozeßdaten verwendet (Kurzschluß-, Leerlauf-, und Lichtbogenhäufigkeiten), die seit der letzten Berechnung beobachtet wurden.

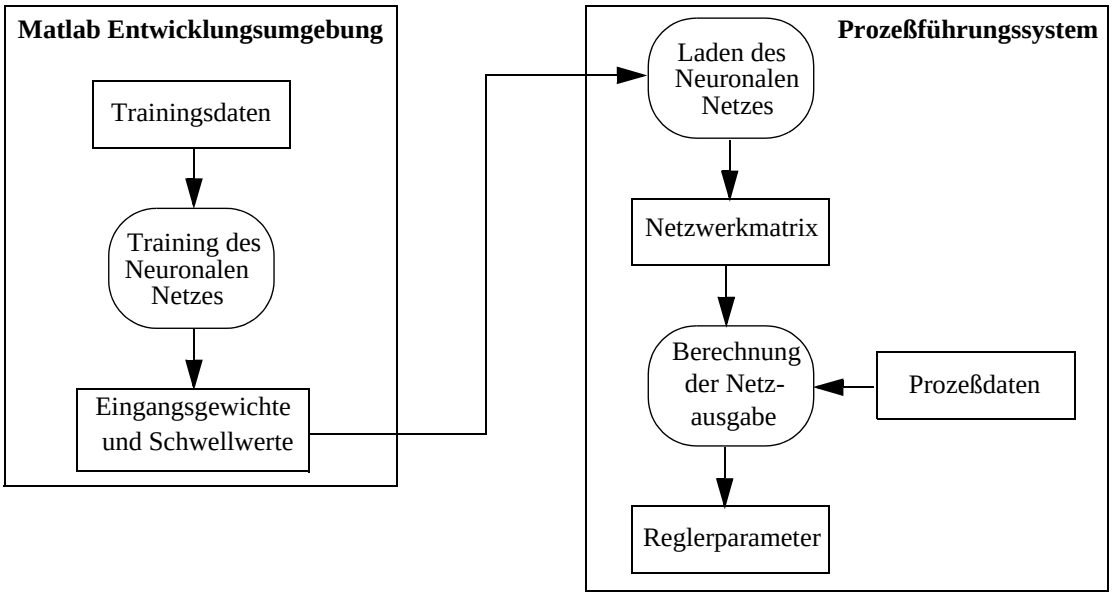


Bild 7.13: Einsatz des Neuronalen Netzes innerhalb des Prozeßführungssystems

Zur Demonstration der Arbeitsweise des entwickelten Prozeßoptimierungssystems wurden während eines Erosionsvorgangs die Eingangsdaten des Neuronalen Netzes (relative Leerlauf-, Kurzschluß- und Lichtbogenhäufigkeiten) sowie die Adaptionsmaßnahmen protokolliert. Bild 7.14 stellt die Daten im zeitlichen Verlauf dar.

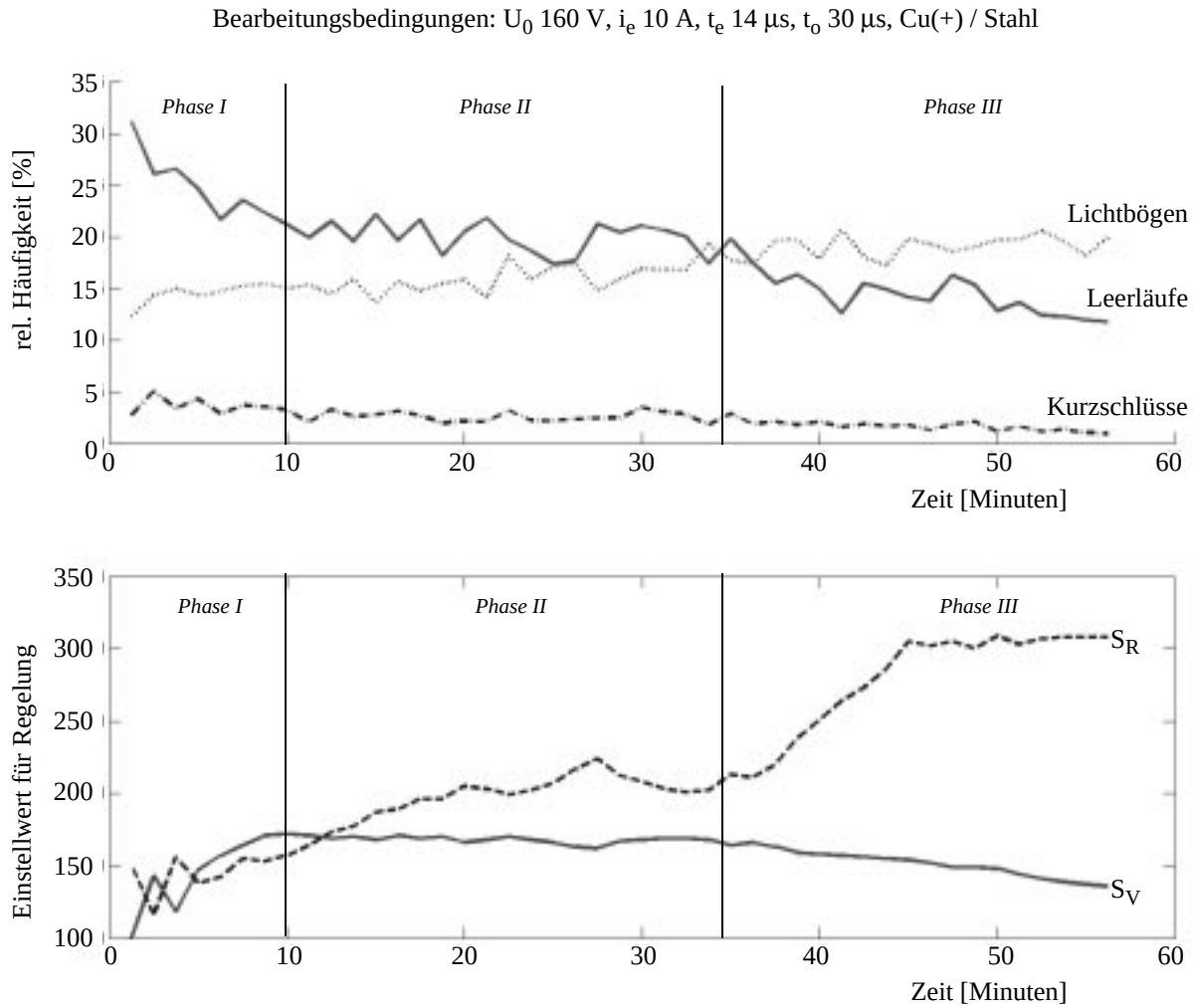


Bild 7.14: Arbeitsweise des Neuronalen Netzes zur Regleradaption

Da sich der Prozeßverlauf mit zunehmender Einsenktiefe verändert, wurde die Darstellung in Bild 7.14 in Phasen 3 aufgeteilt. Diese Aufteilung trägt zusätzlich zum besseren Verständnis bei.

Phase I ist gekennzeichnet durch hohe Kurzschluß- und Leerlaufanteile. Diese Problematik ist typisch für den Beginn des Erosionsvorgangs. Da die beiden Elektrodenstirnflächen nicht exakt parallel ausgerichtet sind, kommt es punktuell zu Kurzschlüssen. Die Ausregelung dieser Kurzschlüsse sorgt für ansteigende Leerlaufhäufigkeiten. Das Neuronale Netz erhöht die Vorwärtsskalierung S_V, um den Leerlaufanteil zu reduzieren.

Phase II ist geprägt durch einen immer noch erhöhten Kurzschlußanteil (2 - 3% im Vergleich zum Endzustand in Phase III 0,5 - 1%), wobei die Leerlaufhäufigkeit bereits erfolgreich reduziert ist. Das Neuronale Netz erhöht die Rückwärtsskalierung S_R, um durch schnelleren Elektrodenrückzug im Kurzschlußfall Brückenbildungen zügig aufheben zu können.

Phase III weist einen abnehmenden Leerlaufanteil auf. Im selben Maße, wie die Leerlaufhäufigkeit abnimmt, ist eine Zunahme des Lichtbogenanteils zu beobachten. Dies folgt aus den Spülverhältnissen, die sich mit zunehmender Eindringtiefe verschlechtern. Um den Anstieg der Fehlentladungen zu mindern, verringert das Neuronale Netz die Vorwärtsskalierung S_V. Gleichzeitig wird die Rückwärtsskalierung S_R stark erhöht, um den Kurzschlußanteil noch weiter zu reduzieren.

Zum Nachweis der Leistungsfähigkeit des entwickelten Neuro-Fuzzy-Regelsystems wurden zwei typische Bearbeitungsbeispiele ausgewählt. Ziel der Bearbeitung ist es, eine möglichst schnelle Einsenkung herzustellen, wobei die Werkstückoberflächen nicht durch lichtbogenartige Fehlentladungen beschädigt werden dürfen. Der Verschleiß an den Werkzeugelektroden sollte möglichst klein sein, um eine hohe Abbildungsgenauigkeit sicherzustellen.

Bei der ersten Bearbeitungsaufgabe handelt es sich um die Herstellung einer kugelförmigen Einsenkung von 6 mm Tiefe. Durch die zu Beginn der Bearbeitung geringen Elektrodenstirnflächen stellt dieser Abtragsversuch hohe Anforderungen an die Lichtbogenerkennung und Spaltweitenregelung. Bild 7.15 zeigt die Versuchsbedingungen und das Arbeitsergebnis.

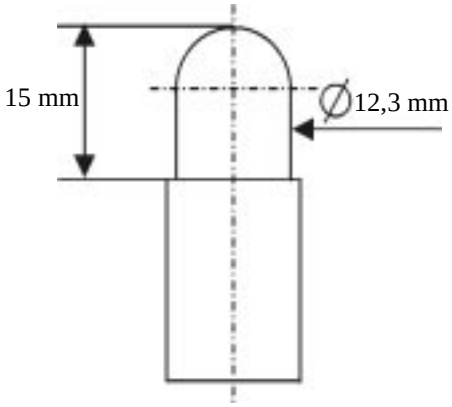
Skizze der Elektrode	Versuchsbedingungen	Bearbeitungsergebnis:
	Werkstück: 56NiCrMoV7 Werkzeug: E-Cu (+) Einsenktiefe: 6 mm Maschineneinstellung: U_0: 160 V i_e: 12 A t_e: 15 µs t_0: 30 µs	Bearbeitung ohne Neuro-Adaption: Bearbeitungszeit: 101,6 Minuten relativer Verschleißυ: 8,13 % Bearbeitung mit Neuro-Adaption: Bearbeitungszeit: 82,3 Minuten relativer Verschleißυ: 8,48 %

Bild 7.15: Abtragsversuch zur Herstellung einer kugelförmigen Einsenkung

Das Neuronale Netz in der vorgestellten Form verkürzte die Bearbeitungszeit um 19 %, wobei der Werkzeugelektrodenverschleiß nur geringfügig angestiegen ist.

Im nächsten Abtragsversuch wurde mittels einer spitz zulaufenden, keilförmigen Werkzeugelektrode eine Einsenkung erzeugt. Die Geometrie dieser Elektrode erschwert die Spülbedingungen und stellt höchste Anforderungen an die Spaltweitenregelung. Da die Stirnfläche der Elektroden zu Bearbeitungsbeginn sehr klein ist, wurde bis zu einer Einsenktiefe von 4 mm zunächst nur ein kleiner Strom von 6 A verwendet, der dann auf 8 A erhöht wurde. Durch diese Maßnahme läßt sich der Elektrodenverschleiß an der Keilspitze reduzieren und die Formgenauigkeit erhöhen.

Diese Stromänderung während der Bearbeitung demonstriert gleichzeitig, daß das Neuronale Netz in der Lage ist, sich flexibel an Bearbeitungsbedingungen anzupassen, die sich aufgrund äußerer Einflüsse verändern. Bild 7.16 zeigt das Bearbeitungsergebnis.

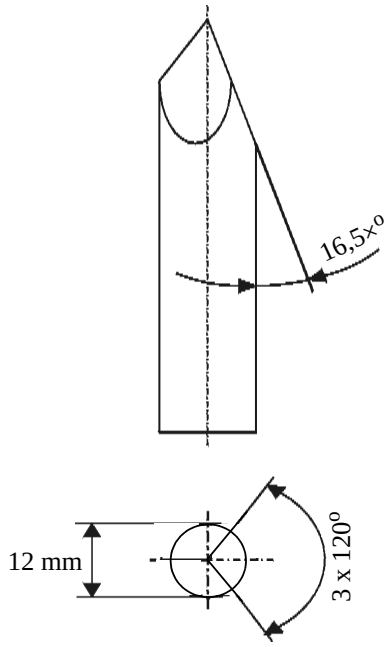
Skizze der Elektrode	Versuchsbedingungen	Bearbeitungsergebnis:
	Werkstück: 56NiCrMoV7 Werkzeug: E-Cu (+) Einsenktiefe: 10 mm Maschineneinstellung bis 4mm Einsenktiefe: U_0: 200 V i_e: 6 A t_e: 10 µs t_0: 30 µs Maschineneinstellung ab 4mm Einsenktiefe: U_0: 200 V i_e: 8 A t_e: 12 µs t_0: 30 µs	Bearbeitung ohne Neuro-Adaption: Bearbeitungszeit: 126,7 Minuten relativer Verschleißυ: 7,68 % Bearbeitung mit Neuro-Adaption: Bearbeitungszeit: 106,6 Minuten relativer Verschleißυ: 7,75 %

Bild 7.16: Abtragsversuch zur Herstellung einer keilförmigen Einsenkung

Durch das Neuronale Netz ergibt sich eine Zeitersparnis von 16 %, bei nahezu gleichbleibendem Werkzeugelektrodenverschleiß.

Die Erosionsversuche lassen erkennen, daß es durch die vorgestellte Regleradaption in Form eines Neuronalen Netzes möglich ist, gegenüber der ausschließlich auf Fuzzy-Technik basierten Spaltweitenregelung eine weitere Verbesserung des Prozeßverlaufes zu erzielen. Da das Neuronale Netz durch Anpassung der Skalierungsfaktoren am Reglerausgang nur indirekt den Erosionsprozeß beeinflussen kann, sind die erzielten Leistungssteigerungen geringer als die Verbesserung, die sich bei der Einführung des leerlauf- / kurzschlußbasierten Fuzzy Spaltweitenreglers einstellten.