

# NICHTLINEARE, LERNFÄHIGE FLUGREGELUNG AM BEISPIEL EINES UNBEMANNTEN FLUGGERÄTES MIT SYSTEMSCHÄDEN

Thomas Krüger, Philipp Schnetter, Peter Vörsmann  
Institut für Luft- und Raumfahrtsysteme - Technische Universität Braunschweig  
Hermann-Blenk-Str. 23, 38108 Braunschweig, Deutschland

## Zusammenfassung

Im Rahmen des Forschungsprojektes *Bürgernahe Flugzeug* am *Campus Forschungsflughafen* in Braunschweig forscht das Institut für Luft- und Raumfahrtsysteme an adaptiven Flugregelungsverfahren. Dabei ist ein Regler auf Basis dynamischer Inversion, auch Eingangs-/Ausgangslinearisierung genannt, implementiert und mit lernfähigen neuronalen Netzen erweitert worden. Die Grundidee der dynamischen Inversion besteht darin, für ein nichtlineares System ein arbeitspunktunabhängiges lineares Eingangs-/Ausgangsverhalten zu realisieren. Dabei wird eine Transformation des Systems durchgeführt, so dass durch Zustandsrückführung die Inversion der Dynamik des Ursprungssystems ermöglicht wird. Auf Grund von Modellungenauigkeiten ist es jedoch nicht möglich, die Systemgleichungen exakt zu invertieren, so dass sich in der Praxis ein Inversionsfehler einstellt. Um diesen auszugleichen, wird das Regelungssystem mit neuronalen Netzen erweitert. Dieses Regelungskonzept bietet nicht nur die Möglichkeit, resultierende Inversionsfehler im Standardbetrieb auszugleichen, sondern erlaubt es auch, mit erweiterten Lernverfahren auf signifikante Änderungen der Systemdynamik auf Grund von Schäden am Flugzeug zu reagieren. Die Ergebnisse zeigen, dass dieses Regelungskonzept ein leistungsfähiges Verfahren zur adaptiven Flugregelung darstellt. Effektiv trainierte neuronale Netze ermöglichen das stabile Ausregeln von Modellunsicherheiten, bei denen das gleiche Fluggerät ohne adaptiven Regelkreis instabil wird. Das vorgestellte Regelungskonzept ist damit Grundlage für ähnliche Untersuchungen zu bemannten Flugzeugen wie sie im Projekt *Bürgernahe Flugzeug* geplant sind.

## 1 Einleitung

Die Nutzung unbemannter Flugsysteme (Unmanned Aircraft Systems - UAS) hat in den letzten Jahren, nicht zuletzt für die wissenschaftliche Anwendung, stark zugenommen. Dabei lässt sich feststellen, dass die Vorteile geringer Größe und niedriger Kosten häufig Probleme in der Flugregelung nach sich ziehen, wie zum Beispiel Modellunsicherheiten oder die Empfindlichkeit gegenüber Wind und Turbulenz. Viele dieser Fluggeräte nutzen konventionelle kaskadisch aufgebaute Regler, wie sie in [1] beschrieben sind. Diese zeigen unter Nominalbedingungen angemessenes Verhalten, können aber

bei starken atmosphärischen Störungen an ihre Leistungsgrenze gelangen, was bei der Durchführung verschiedener wissenschaftlicher Kampagnen deutlich wurde [2, 3]. Die Lerneigenschaften künstlicher neuronaler Netze (KNN) bieten in diesem Zusammenhang die Möglichkeit das Regelungssystem unter Echtzeitbedingungen auch an Systemschäden anzupassen. Dabei können KNN verschieden eingesetzt werden: als Regelungselemente [4], zur Identifizierung aerodynamischer Parameter [5] oder speziell für die rein neuronale Regelung kleiner UAS [6, 7].

Das Regelungsprinzip der dynamischen Inversion, auch Eingangs-/Ausgangslinearisierung genannt, ist in Kombination mit neuronalen Netzen in der Lage, nichtlineare Effekte zu kompensieren und ist in den letzten Jahren umfangreich diskutiert worden [8–11]. Auftretende Inversionsfehler können durch ein KNN, das mit dem Backpropagation Algorithmus trainiert wird, approximiert und ausgeglichen werden [12]; die Erweiterung mit einem robustifizierendem Term und der sogenannten e-Modifikation [13] garantiert stabiles Lernverhalten. Eine weitere Möglichkeit, die Leistungsfähigkeit der neuronalen Netze zu verbessern, ist die Nutzung alternativer Lernverfahren. Dafür ist das Konzept der strukturvariablen Regelung auf das Training neuronaler Netze adaptiert worden, was eine Steigerung der Konvergenzgeschwindigkeit des Lernalgorithmus' unter Beachtung der Gesamtstabilität ermöglicht. Dies wird erreicht, indem das KNN in einen sogenannten Gleitzustand (Sliding Mode) überführt wird, wodurch eine dynamische Bestimmung der Lernrate möglich ist [14–16].

Die hier vorgestellten Ergebnisse sind mit dem unbemannten Kleinflugzeug *CAROLO P200* und der dazugehörigen Simulationsumgebung entstanden. Das Fluggerät ist in Abb. 1 dargestellt. Dieses UAS hat ein Gewicht von 5 kg, eine Spannweite von 200 cm und kann mit elektischem Antrieb ca. 60 Minuten bei einer Reisegeschwindigkeit von 20 m/s operieren. Als Steuerorgane stehen neben dem Schub Quer- und Höhenruder zur Verfügung. Die Simulation nutzt die nichtlinearen Bewegungsgleichungen, identifizierte Aktuator- und Sensormodelle, ein komplett modelliertes integriertes Navigationssystem [17] und ein Atmosphärenmodell mit einem Dryden-Turbulenzspektrum [1], so dass möglichst realistische Simulationsergebnisse erzeugt werden.



Abbildung 1: Das unbemannte Flugsystem CAROLO P200 im Landeanflug.

## 2 Theorie des Regelungsverfahrens

### 2.1 Dynamische Inversion

Gegeben sei ein nichtlineares Eingrößensystem (SISO - single-input-single-output):

$$\begin{aligned} \dot{x} &= f(x) + g(x) \cdot u, \\ y &= h(x), \end{aligned} \quad (1)$$

wobei  $f$ ,  $g$  und  $h$  Vektorfelder im Wertebereich  $D \subset \mathbb{R}^n$  darstellen und als hinreichend glatte Abbildungen angesehen werden können. Das Prinzip der dynamischen Inversion ist es, einem nichtlinearen System ein lineares Eingangs-/Ausgangsverhalten aufzuprägen. Dies setzt voraus, dass der Ausgang  $y$  direkt vom Eingang  $u$  steuerbar ist. Dazu kann die Ableitung  $\dot{y}$  geschrieben werden als:

$$\dot{y} = \frac{\partial h}{\partial x} \cdot \dot{x} = \frac{\partial h}{\partial x} \cdot [f(x) + g(x) \cdot u] = L_f h(x) + L_g h(x) \cdot u, \quad (2)$$

wobei

$$\frac{\partial h}{\partial x} \cdot f(x) = L_f h(x) \quad (3)$$

als Lie-Ableitung bezeichnet wird. Ist  $L_g h(x)$  bei  $x = x^0$  gleich null und damit  $\dot{y}$  nicht direkt steuerbar, muss die Gleichung  $r$  mal abgeleitet werden bis der Zusammenhang

$$y^{(r)} = L_f^r h(x) + L_g L_f^{r-1} h(x) \cdot u, \quad (4)$$

mit  $L_g L_f^{r-1} h(x) \neq 0$  gefunden ist. Für ein Mehrgrößensystem (MIMO - multiple-input-multiple-output) mit  $m$  Ausgaben kann (4) geschrieben werden als:

$$\begin{bmatrix} y_i^{(r_i)} \end{bmatrix}_{m \times 1} = \vec{b}(x) + A(x) \cdot \vec{u}, \quad (5)$$

mit

$$A(x) = \begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1(x) & \dots & L_{g_m} L_f^{r_1-1} h_1(x) \\ L_{g_1} L_f^{r_2-1} h_2(x) & \dots & L_{g_m} L_f^{r_2-1} h_2(x) \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r_m-1} h_m(x) & \dots & L_{g_m} L_f^{r_m-1} h_m(x) \end{bmatrix}, \vec{b}(x) = \begin{bmatrix} L_f^{r_1} h_1(x) \\ L_f^{r_2} h_2(x) \\ \vdots \\ L_f^{r_m} h_m(x) \end{bmatrix}. \quad (6)$$

Das lineare Eingangs-/Ausgangsverhalten kann über eine Zustandsrückführung realisiert werden:

$$\vec{u} = A^{-1}(x) \cdot [\vec{v} - \vec{b}(x)], \quad (7)$$

was einer Invertierung der Systemdynamik entspricht. Die Pseudosteuergröße  $\vec{v}$  wird generiert, um das Stellsignal  $u$  zu bestimmen, das den gewünschten Ausgang des Systems  $y^{(r)}$  zur Folge hat. Die Kombination von (7) und (5) reduziert die Systemdynamik zu einer Integratorkette, die es ermöglicht lineare Regelungsansätze zu nutzen:

$$\begin{bmatrix} y_i^{(r_i)} \end{bmatrix}_{m \times 1} = \vec{v}_i. \quad (8)$$

### 2.2 Fehlerdynamik

Auf Grund von Parameterunsicherheiten entspricht die modellierte Dynamik, die für die Inversion genutzt wird (mit Index  $\hat{\cdot}$  gekennzeichnet), nicht der des wirklichen Systems. Daher kann man annehmen, dass:

$$A(x) \neq \hat{A}(x), \quad \vec{b}(x) \neq \hat{b}(x). \quad (9)$$

Daraus folgt, dass

$$\vec{y}^{(r)} = \vec{b}(x) + A(x) \cdot \hat{A}^{-1}(x) \cdot [\vec{v} - \hat{b}(x)] \neq \vec{y}_R^{(r)}, \quad (10)$$

wobei  $\vec{y}_R^{(r)}$  die  $r$ -te Ableitung des gewünschten Referenzsignals darstellt. Für ein MIMO-System kann der Inversionsfehler geschrieben werden als:

$$\Delta = \begin{bmatrix} y_i^{(r_i)} \end{bmatrix}_{m \times 1} - \begin{bmatrix} y_{R,i}^{(r_i)} \end{bmatrix}_{m \times 1}. \quad (11)$$

Die Abweichung zwischen tatsächlichem  $\vec{y}$  und gewünschtem Ausgang des Systems  $\vec{y}_R$  ist beschrieben durch:

$$\vec{e} = \begin{bmatrix} e_1 \\ \vdots \\ e_m \end{bmatrix} = \begin{bmatrix} y_1 - y_{R,1} \\ \vdots \\ y_m - y_{R,m} \end{bmatrix}, \quad (12)$$

woraus der Fehlerzustandsvektor folgt

$$\vec{\chi}(t) = \begin{bmatrix} e(t) \\ \dot{e}(t) \\ \vdots \\ e^{(r-1)} \end{bmatrix}. \quad (13)$$

Damit lässt sich die Fehlerdynamik mit einer linearen Differentialgleichung beschreiben:

$$\dot{\vec{e}}^{(r)} + c_{r-1} \vec{e}^{(r-1)} + \dots + c_1 \dot{\vec{e}} + c_0 \vec{e} = \vec{e}^{(r)} + \vec{c}^T \vec{\chi} = \Delta, \quad (14)$$

wobei  $\vec{c}^T$  den Vektor der Reglerverstärkungen darstellt. Dies bedeutet, dass der Inversionsfehler als Eingangssignal der Fehlerdynamik interpretiert werden kann. Mit dem Zustandsvektor  $\vec{\chi}$  kann die Zustandsraumdarstellung der Fehlerdynamik bestimmt werden durch:

$$\dot{\vec{\chi}} = A_E \vec{\chi} + \vec{b}_E \Delta, \quad (15)$$

mit

$$A_E = \begin{bmatrix} 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ -c_0 & -c_1 & \dots & -c_{r-1} \end{bmatrix}, \quad \vec{b}_E = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \quad (16)$$

Die Stabilität der Fehlerdynamik kann durch positive Parameter  $c_i > 0$  sichergestellt werden, woraus folgt, dass  $A_E$  eine Hurwitz-Matrix ist. Aus der Ljapunov-Funktion

$$A_E^T P_E + P_E A_E = -Q_E \quad (17)$$

folgt, dass  $P_E$  und  $Q_E$  positive Matrizen sein müssen.

### 2.3 Reglerentwurf

Der vorgestellte Regelkreis besteht aus drei kaskadierten Schleifen mit relativem Grad eins für Dämpfung, Lageregelung und Bahnführung. Jede der Regelschleifen besteht aus einem Kommandofilter zur Generierung der Referenzsignale, einem Linearregler und einem Inversionsblock. Die vereinfachte Reglerarchitektur ist in Abb. 2 dargestellt.

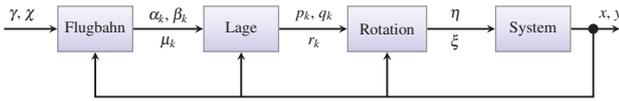


Abbildung 2: Aufbau der Reglerkaskade.

Im innersten Regelkreis der Rotationsdynamik erfolgt die Steuerflächenzuweisung auf Höhen- und Querruder. Da kein Seitenruder verwendet wird, wird die Gierbewegung nicht direkt geregelt, so dass die gewünschten Roll- und Nickmomente den Steuerflächen zugeordnet werden müssen. Dabei wird von der Entkopplung der jeweiligen Bewegungen ausgegangen. Zur Berechnung der Ruderausschläge wird ein inkrementeller Ansatz gewählt [11], dass heißt es werden nur die nötigen Zusatzausschläge zu den aktuell anliegenden berechnet. Damit ergeben sich die Stellsignale für Höhenruder  $\eta_C$  und Querruder  $\xi_C$  zu

$$\eta_C = \hat{\eta}_0 + \delta\eta \quad \xi_C = \hat{\xi}_0 + \delta\xi, \quad (18)$$

mit

$$\delta\eta = \frac{\delta C_m}{C_{m\eta}} \quad \delta\xi = \frac{\delta C_l}{C_{l\xi}}. \quad (19)$$

### 2.4 Erweiterung mit neuronalen Netzwerken

Die zur Ausregelung des Inversionfehlers  $\Delta$  verwendeten neuronalen Netzwerke sind forwärtsgerichtete Netze, die in der verdeckten Schicht Tangens Hyperbolicus- und in den anderen Schichten lineare Transferfunktionen verwenden. Der detaillierte Aufbau des Regelungssystems ist, an der Rotationsdynamik verdeutlicht, in Abb. 3 erkennbar. Allgemein lässt sich die Vorwärtspropagation eines Neurons berechnen mit:

$$o_j^L = f \left( \sum_{i=1}^n w_{ij} \cdot o_i^{L-1} + b_j \right), \quad (20)$$

wobei  $j$  das Neuron der Schicht  $L$  und  $i$  die  $n$  Neuronen der Schicht  $L-1$  bezeichnet;  $b$  ist der Biaswert von Neuron  $j$  und  $f$  die Transferfunktion. Dies kann in allgemeiner Vektorschreibweise zusammengefasst werden:

$$\vec{y} = \mathbf{w}^{(2)} \cdot \vec{f}^{(2)}(\mathbf{w}^{(1)} \vec{x}). \quad (21)$$

Dabei ist  $\vec{x}$  der Eingabevektor,  $\mathbf{w}^{(1)}$  und  $\mathbf{w}^{(2)}$  sind die Gewichtsmatrizen des Netzwerkes und  $\vec{f}^{(2)}$  bezeichnet die Transferfunktion der verdeckten Schicht. Der resultierende quadratische Fehler, als Maß der Ausgabegüte, wird schließlich bestimmt mit:

$$E_j = \frac{1}{2} (y_{t,j} - y_j)^2. \quad (22)$$

Unter Verwendung eines Gradientenabstiegsverfahrens ohne Moment, welches für diese Reglerarchitektur durchaus gängig ist [8–10], kann die Gewichtsänderung geschrieben werden als:

$$\Delta w_{ij} = -\mu \frac{\partial E_j}{\partial w_{ij}}. \quad (23)$$

Die hierbei eingeführte Lernrate  $\mu > 0$  hat einen starken Einfluss auf die Schrittweite des Lernverfahrens und muss im Sinne der Stabilität daher mit Bedacht gewählt werden. Nach mehrfacher Anwendung der Kettenregel kann die Gewichtsänderung  $\Delta w_{ij}$  aus Termen der Vorwärtspropagation ermittelt werden [18]. Um den Inversionsfehler, wie in Abb. 3 dargestellt, auszuregulieren, muss dieser nichtlineare Parameter vom neuronalen Netz approximiert werden. Wie in [12] gezeigt, kann ein KNN mit einer verdeckten Schicht und sigmoid geformter Transferfunktion beliebige, nichtlineare Funktionen bis auf einen Approximationsfehler  $\vec{\epsilon}$  annähern:

$$\vec{\Delta} = \mathbf{w}_*^{(2)} \cdot \vec{f}^{(2)}(\mathbf{w}_*^{(1)} \vec{x}) + \vec{\epsilon}(\vec{x}), \quad (24)$$

wobei  $\|\epsilon\| < \bar{\epsilon}$  und  $\bar{\epsilon}$  nach oben begrenzt ist. Die Matrizen  $\mathbf{w}_*^{(1)}$  und  $\mathbf{w}_*^{(2)}$  repräsentieren die optimalen Gewichte, bei denen  $\bar{\epsilon}$  minimal ist. Damit lässt sich die Fehlerdynamik aus (15) erweitern:

$$\dot{\vec{\chi}} = \mathbf{A}_E \cdot \vec{\chi} + \mathbf{b}_E (\Delta - v_{ad} - v_r); \quad (25)$$

hierbei ist die Ausgabe des KNN mit  $v_{ad}$  gegenzeichnet.

$$v_{ad} = \mathbf{w}^{(2)} \cdot \vec{f}^{(2)}(\mathbf{w}^{(1)} \vec{x}) \quad (26)$$

Der Term  $v_r$  kennzeichnet einen robustifizierenden Anteil [13], der für die Ljapunov-Stabilität des Gesamtsystems von Relevanz ist:

$$v_r = [k_{r0} + k_{r1} \cdot (\|\mathbf{w}\|_F + \bar{\mathbf{w}}_*)] \cdot \zeta. \quad (27)$$

Hierbei sind  $k_{r0}$  und  $k_{r1}$  positive Parameter,  $\zeta = \vec{\chi}^T \mathbf{P}_E \mathbf{b}_E$  bezeichnet die gefilterte Fehlergröße und  $\|\mathbf{w}\|_F$  ist die Frobeniusnorm der Matrix  $\mathbf{w}$ , die nach oben begrenzt ist:

$$\|\mathbf{w}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n w_{ij}^2} \leq \bar{\mathbf{w}}_*. \quad (28)$$



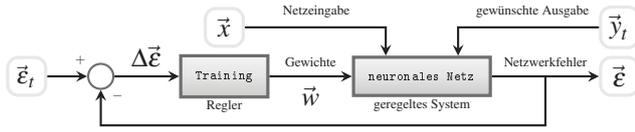


Abbildung 4: Das Training eines neuronalen Netzes als Regelungsprozess

Ausgangssignal  $\bar{y}$ , woraus sich der aktuelle Fehler  $\bar{\epsilon}$  ergibt. Um dieses Verfahren umzusetzen, kann man die Schaltlinie  $\vec{S}$  des Lernalgorithmus' unterschiedlich definieren [15, 16]. Die Änderung der Verbindungsgewichte unter Verwendung des SMC-Ansatzes stellt im weitesten Sinne lediglich eine Erweiterung des normalen Backpropagation-Algorithmus' um die Schaltlinie  $\vec{S}$  dar :

$$\Delta \vec{w} = \left( \frac{\partial \bar{y}(\vec{w}, \vec{x}, \vec{y}_d)}{\partial \vec{w}(t)} \right)^T \cdot \mu \cdot \text{diag}(\text{sign}(\vec{S})) \cdot |\bar{\epsilon}|. \quad (38)$$

Dieses Lernverfahren kann demzufolge in (34) und (35) angewendet werden, was die dynamische Berechnung der Lernraten ermöglicht. Zur Herleitung des Verfahrens wird zunächst die Schaltlinie  $\vec{S}$  aus (38) definiert:

$$\dot{\vec{S}} = \dot{\bar{\epsilon}} + \lambda \cdot \bar{\epsilon}. \quad (39)$$

Für  $S = 0$  befindet sich das System direkt auf der Schaltlinie, auf der der Netzwerkfehler gegen 0 läuft, sofern der Faktor  $\lambda$  positiv gewählt wird:

$$\dot{\vec{S}} = \dot{\bar{\epsilon}} + \lambda \cdot \bar{\epsilon} = 0 \quad \Rightarrow \quad \bar{\epsilon} = \bar{\epsilon}(t_0) \cdot e^{-\lambda(t-t_0)}. \quad (40)$$

Im Weiteren wird die Ableitung des Netzwerkfehlers  $\bar{\epsilon}$  über die Taktrate des Regelkreises  $T_s$  bestimmt.

$$\dot{\bar{\epsilon}} \approx \frac{\bar{\epsilon}(t) - \bar{\epsilon}(t - T_s)}{T_s} \quad (41)$$

Der Index  $k$  beschreibt den aktuellen Trainingsschritt:

$$\vec{S}(t) = \vec{S}_k \quad ; \quad \vec{S}(t + T_s) = \vec{S}_{k+1} \quad ; \quad \dots \quad (42)$$

Um die Stabilität des Lernverfahrens nachzuweisen, wird gefordert, dass sich der Betrag der Schaltlinienfunktion  $\vec{S}$  mit jedem Zeitschritt verringert.

$$|\vec{S}_{k+1}| < |\vec{S}_k| \quad (43)$$

Unter Verwendung von (39) und (41) können dann  $\vec{S}_k$  sowie  $\vec{S}_{k+1}$  berechnet werden:

$$\vec{S}_k = \dot{\bar{\epsilon}}_k + \lambda \cdot \bar{\epsilon}_k = \left( \lambda + \frac{1}{T_s} \right) \bar{\epsilon}_k - \frac{1}{T_s} \bar{\epsilon}_{k-1} \quad (44)$$

$$\vec{S}_{k+1} = \dot{\bar{\epsilon}}_{k+1} + \lambda \cdot \bar{\epsilon}_{k+1} = \left( \lambda + \frac{1}{T_s} \right) \bar{\epsilon}_{k+1} - \frac{1}{T_s} \bar{\epsilon}_k. \quad (45)$$

Der unbekannte Faktor  $\bar{\epsilon}_{k+1}$  aus (45) muss nun approximiert werden, wofür der Fehler  $\bar{\epsilon}_{k+1}$  umgeschrieben wird:

$$\bar{\epsilon}_{k+1} = \bar{\epsilon}_k + \Delta \bar{\epsilon}_k. \quad (46)$$

Anschließend wird die Änderung des Fehlers  $\Delta \bar{\epsilon}_k$  nach einem Zeitschritt bestimmt:

$$\Delta \bar{\epsilon}_k = \bar{\epsilon}_{k+1} - \bar{\epsilon}_k = (\bar{y}_{t,k+1} - \bar{y}_{k+1}) - (\bar{y}_{t,k} - \bar{y}_k) = \Delta \bar{y}_{t,k} - \Delta \bar{y}_k. \quad (47)$$

Der Faktor  $\Delta \bar{y}_{t,k}$  ist aus dem aktuellen Propagationsschritt des KNN bekannt; für  $\Delta \bar{y}_k$  wird eine Taylorreihenentwicklung angenommen:

$$\Delta \bar{y}_k = \frac{\partial \bar{y}_k(\vec{w}_k, \vec{x}_k)}{\partial \vec{w}_k} \Delta \vec{w}_k + \frac{\partial \bar{y}_k(\vec{w}_k, \vec{x}_k)}{\partial \vec{x}_k} \Delta \vec{x}_k. \quad (48)$$

Die partiellen Ableitungen  $\frac{\partial \bar{y}_k(\vec{w}_k, \vec{x}_k)}{\partial \vec{w}_k}$  und  $\frac{\partial \bar{y}_k(\vec{w}_k, \vec{x}_k)}{\partial \vec{x}_k}$  können durch Rückpropagation des Netzausganges  $y_k$  durch das KNN bestimmt werden. Der Parameter  $\Delta \vec{w}_k$  ist in (38) gegeben und für die Bestimmung von  $\vec{S}_{k+1}$  bereits bekannt. Außerdem wird für kleine Zeitschritte  $T_s$  eine minimale Änderung der Netzwerkeingaben angenommen. Um die Bedingung aus (43) zu erfüllen, sind zwei Parameter zu bestimmen: die Lernrate  $\mu$  und  $\lambda$ , welches im Sinne der Stabilität positiv gewählt werden muss. Die Lernrate  $\mu$  kann desweiteren für jeden Regelungsschritt individuell bestimmt werden. Im Sinne der Deutlichkeit werden die Substitutionen  $a_i$  und  $b_i$  eingeführt, wobei  $i$  als Index der jeweiligen Netzausgabe fungiert. Für die beschriebene Anwendung ist ( $i = 1$ ), so dass:

$$a_i = \left( \lambda + \frac{1}{T_s} \right) \left( \epsilon_{k,i} + \Delta y_{d,k,i} - \left( \frac{\partial \bar{y}_k}{\partial \vec{x}_k} \Delta \vec{x}_k \right)_i \right) - \frac{1}{T_s} \epsilon_{k,i} \quad (49)$$

$$b_i = \left( \lambda + \frac{1}{T_s} \right) \left( \frac{\partial \bar{y}_k}{\partial \vec{w}_k} \left( \frac{\partial \bar{y}_k}{\partial \vec{w}_k} \right)^T \text{diag}(\text{sign}(\vec{S}_k)) |\bar{\epsilon}_k| \right)_i. \quad (50)$$

Zur stabilen, dynamischen Kalkulation von  $\mu$  erhält man schließlich:

$$\left\{ \begin{array}{l} \left\{ -\frac{S_{k,i}}{b_i} + \frac{a_i}{b_i} < \mu < \frac{S_{k,i}}{b_i} + \frac{a_i}{b_i} \right\} \text{ for } (S_{k,i} > 0 \wedge b_i > 0) \vee (S_{k,i} < 0 \wedge b_i < 0) \\ \left\{ \frac{S_{k,i}}{b_i} + \frac{a_i}{b_i} < \mu < -\frac{S_{k,i}}{b_i} + \frac{a_i}{b_i} \right\} \text{ for } (S_{k,i} > 0 \wedge b_i < 0) \vee (S_{k,i} < 0 \wedge b_i > 0). \end{array} \right. \quad (51)$$

## 4 Simulationsergebnisse

Die Tests der Reglerarchitektur sind mit der in Abschnitt 1 beschriebenen Simulationsumgebung durchgeführt worden. Die Eigenschaften des Regelkreises im Nominalfall ohne atmosphärische Störungen sind in Abb. 5 für die Längsbewegung dargestellt. Der Soll-Ist-Vergleich für die Seitenbewegung verläuft qualitativ ähnlich und ist hier daher nicht dargestellt. Die Regelungsaufgabe dieses Referenzszenarios besteht in einer schnellen Abfolge zweier Höhenkommandos. Es ist erkennbar, dass der Regelkreis mit hoher Genauigkeit arbeitet, wobei auch deutlich wird, dass die Zeitkonstanten der Referenzmodelle von innen nach außen größer werden, was geringe Zeitverzögerungen im Bahn- und Anstellwinkel zur Folge hat.

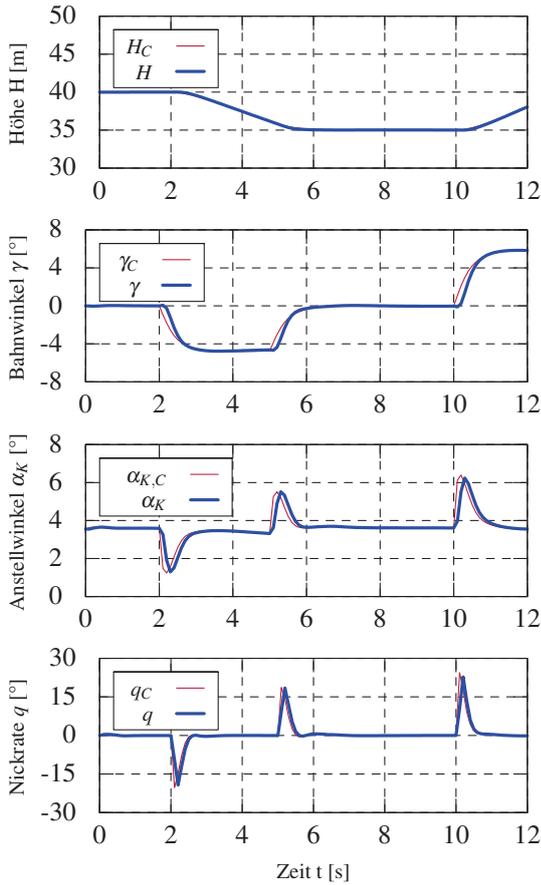


Abbildung 5: Vergleich von Soll- und Istwerten der Längsbewegung für alle Regelschleifen.

Nach der Validierung der grundlegenden Reglereigenschaften wird die Architektur mit neuronalen Netzen erweitert, und über eine Zeit von 180 Sekunden getestet. Um die Reaktion des Regelkreises, besonders des erweiterten SMC-Lernverfahrens, in Gegenwart signifikanter Nichtlinearitäten zu testen, werden die Simulationen mit atmosphärischen Störungen und zum Teil mit Systemschäden durchgeführt. Um die höhere Konvergenzgeschwindigkeit des SMC-Lernverfahrens gegenüber dem Standardverfahren zu prüfen, werden beide Regelkreise unter gleichen Bedingungen getestet. Der Verlauf der Testbahn ist in Abb. 6 für Längs- und Seitenbewegung dargestellt und stellt mit einer Abfolge verschiedener Kurven-, sowie Sink- und Steigmanöver ein angemessenes Szenario für ein kleines unbemanntes Flächenflugzeug dar. Die Trajektorie ist durch kubische Bezièr-Splines definiert, was den Vorteil einer mathematisch definierten Flugbahn und damit der genauen Berechnung einer seitlichen Bahnabweichung hat [7].

Zur Analyse der Robustheit des erweiterten Reglers in Gegenwart von Systemschäden sind diese vereinfacht simuliert worden. Dazu ist die Spannweite von 2 m auf 1.2 m reduziert worden, was eine starke Reduzierung der Flügelfläche, der Aerodynamik sowie der Querrudereffektivität und somit eine deutliche Änderung der Systemdynamik mit sich bringt. Zusätzlich ist ein Windfeld mit Windvektorkomponenten von  $u = 1 \text{ m/s}$ ,  $v = 2 \text{ m/s}$  und  $w = 1 \text{ m/s}$ , sowie ein Turbulenz-

spektrum simuliert worden, so dass die Windgeschwindigkeit in Böen bis  $5 \text{ m/s}$  steigt. Die für ein Flugzeug dieser Größe anspruchsvollen Randbedingungen lassen somit nicht unerhebliche Inversionsfehler erwarten. Der Regler soll nicht nur die Bahnführungsgenauigkeit sicherstellen, sondern auch die Geschwindigkeit bei  $20 \text{ m/s}$  konstant halten. In Tab. 1 sind die Ergebnisse der Längs- und Seitenbewegung (Höhenfehler  $\Delta H$  und seitliche Bahnablage  $d$ ) für den Basisregler, sowie die Erweiterungen mit verschiedenen trainierten Netzen dargestellt. Das Gradientenabstiegsverfahren ist mit GA, das Sliding-Mode-Training mit SMC gekennzeichnet. Als Bewertungsgröße der Reglereigenschaften ist der mittlere quadratische Fehler (MSE) sowie die Standardabweichung des Höhenfehlers  $\Delta H$  und der seitlichen Bahnabweichung  $d$  ermittelt. Es ist erkennbar, dass sich die Ergebnisse der verschiedenen Lernverfahren nicht maßgeblich unterscheiden; hinsichtlich des auftretenden Inversionsfehlers ist in Abb. 7 der Vergleich zwischen Basisregler und GA-trainierten Netzwerken sichtbar. Wie zu erwarten, ist die Regelgenauigkeit in der Längsbewegung durch die veränderte Dynamik deutlich reduziert, wobei keiner der Regler instabil wird. In der Seitenbewegung werden die Kompensationseigenschaften des KNN deutlich; Durch die reduzierte Querruderwirksamkeit entsteht ein Inversionsfehler, der beim Basisregler zu Schwingungen der Bahnablage  $d$  führt. Durch Aufschaltung eines GA- oder SMC-trainierten Netzes hingegen kann diese durch den Inversionsfehler induzierte Schwingung komplett gedämpft werden, so dass der Regler zwar eine geringere Bahnführungsgenauigkeit aufweist, aber stabil bleibt.

|            | Basisregler   | KNN mit GA  | KNN mit SMC   |
|------------|---|---|---|
| $\Delta H$ | $MSE = 0.078 \text{ m}^2$<br>$\sigma = 0.279 \text{ m}$ | $MSE_{GD} = 0.115 \text{ m}^2$<br>$\sigma_{GD} = 0.339 \text{ m}$ | $MSE_{SMC} = 0.108 \text{ m}^2$<br>$\sigma_{SMC} = 0.329 \text{ m}$ |
| $d$        | $MSE = 0.452 \text{ m}^2$<br>$\sigma = 0.666 \text{ m}$ | $MSE_{GD} = 0.417 \text{ m}^2$<br>$\sigma_{GD} = 0.640 \text{ m}$ | $MSE_{SMC} = 0.411 \text{ m}^2$<br>$\sigma_{SMC} = 0.635 \text{ m}$ |

Tabelle 1: Eigenschaften der unterschiedlichen Regler.

Um die unterschiedliche Konvergenzgeschwindigkeit der beiden Lernverfahren in Extremsituationen zu analysieren, wird die Systemdynamik noch einmal deutlich verändert. Dazu werden die Trägheitsmomente des Flugzeuges um die x- und y-Achse verdoppelt. Auch wenn diese Änderung der Dynamik nicht alle physikalischen Effekte einbezieht, sorgt sie für ein erheblich trägeres Flugzeug und damit für eine anspruchsvolle Regelungsaufgabe. Die Ergebnisse hierzu, unter Verwendung der gleichen Windbedingungen wie zuvor, sind die Abb. 8 dargestellt. Es ist ersichtlich, dass der Regler mit GA-Training bei ca. 30 Sekunden instabil wird, während das Flugzeug ein Kurvenmanöver im Steigflug durchführt. Augenscheinlich sind Trägheit und daraus folgend der Inversionsfehler so groß, dass das Flugzeug in den Strömungsabriss gelangt. Der Regler mit SMC-trainierten Netzen hingegen kann die gesamte Bahn bei verringerter Regelgüte stabil durchfliegen. Diese Betrachtung zeigt, dass gerade bei hohen Inversionsfehlern die verbesserte Konvergenzgeschwindigkeit des SMC-Trainings durch die dynamische Bestimmung der Lernrate positive Auswirkungen auf das Regelverhalten hat; dies gilt besonders für Extremsituation.

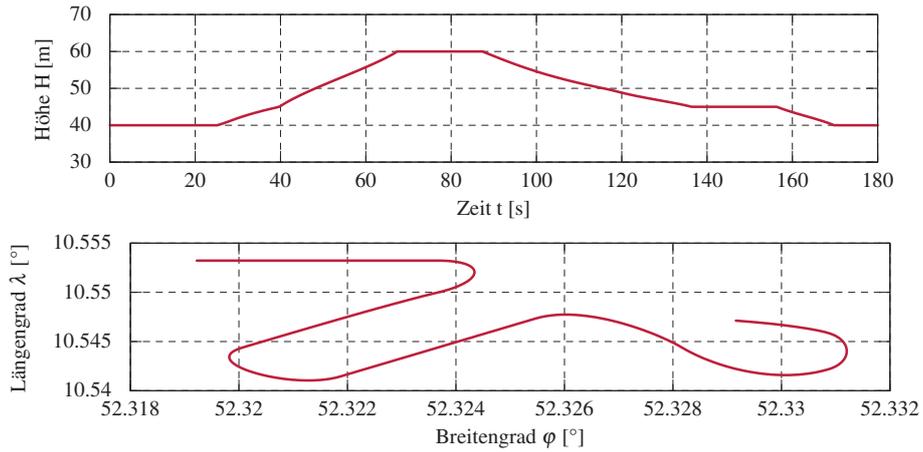


Abbildung 6: Profil der Testbahn in Längs- und Seitenbewegung.

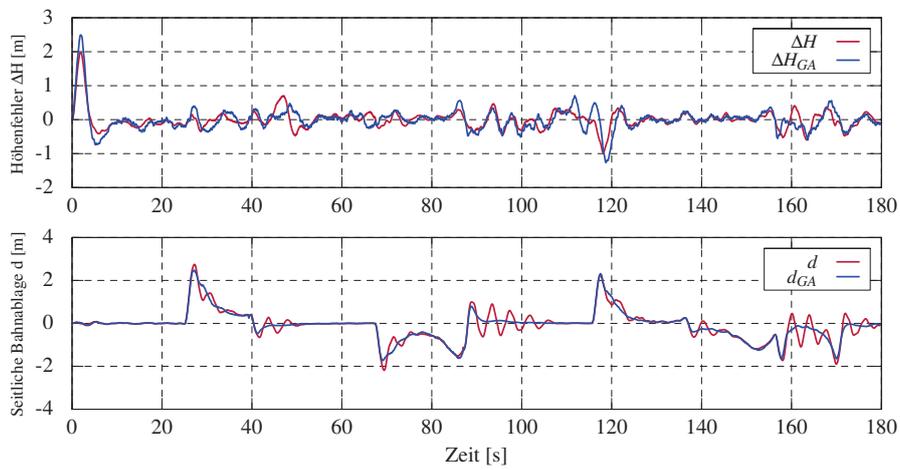


Abbildung 7: Vergleich der Bahnführungsgenauigkeit von Basisregler und mit GA-trainierter KNN-Erweiterung.

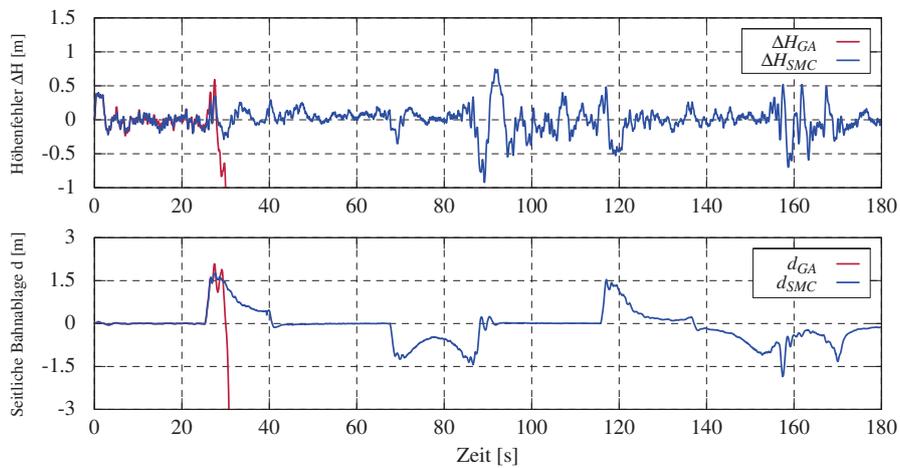


Abbildung 8: Vergleich der Lernverfahren in Gegenwart signifikanter Inversionsfehler.

## 5 Zusammenfassung

In dieser Arbeit ist das Regelungsverfahren der dynamischen Inversion für ein unbemanntes Keinflugzeug angewendet und durch neuronale Netze erweitert worden, die nach dem Prinzip der Sliding-Mode-Regelung trainiert werden. Dieser Ansatz erlaubt die dynamische Bestimmung der Lernrate für jeden Regelungsschritt, was die Eigenschaften des beschriebenen Flugregelungssystems deutlich verbessert. Während das bekannte Trainingsverfahren des Gradientenabstiegs bereits beachtliche Nichtlinearitäten auszuregulieren vermag, ermöglicht die höhere Konvergenzgeschwindigkeit des SMC-Algorithmus' die Reaktion auf signifikante Systemschäden ohne instabil zu werden. In diesem Zusammenhang stellt das vorgestellte Lernverfahren eine sinnvolle Erweiterung der adaptiven Regelung auf Basis dynamischer Inversion dar. Weiterführende Untersuchungen werden sich mit der Implementierung dieses Regelkreises für bemannte Flugzeuge befassen.

## Literatur

- [1] R. Brockhaus. *Flugregelung*. Springer Verlag, ISBN 3-540-55416-5, Berlin, 1994.
- [2] A. Kroonenberg. *Airborne Measurement of Small-Scale Turbulence with special regard to the Polar Boundary Layer*. PhD thesis, Zentrum für Luft- und Raumfahrt, Technische Universität Braunschweig, 2009.
- [3] T. Krüger, C.-S. Wilkens, M. Reinhold, P. Selam, B. Böhm, and P. Vörsmann. Ergebnisse des ANDROMEDA-Projektes - Automatische Luftbildgewinnung mit Unbemannten Kleinflugzeugen. In *German Aeronautics and Astronautics Congress*, Hamburg, Germany, September 2010. DGLR. ISBN:978-3-932182-68-5, PaperID 161314.
- [4] S. Ferrari and R.F. Stengel. Online Adaptive Critic Flight Control. *Journal of Guidance, Control and Dynamics*, 27(5):777–786, 2004.
- [5] J. Seifert. *Identifizierung nichtlinearer aerodynamischer Derivative mit einem Modularen Neuronalen Netzwerk*. PhD thesis, Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, Institut für Systemdynamik und Flugmechanik, 2003.
- [6] T. Dierks and S. Jagannathan. Output Feedback Control of a Quadrotor UAV Using Neural Networks. *IEEE Transactions on Neural Networks*, 21(1):50–66, 2010.
- [7] T. Krüger, M. Mößner, A. Kuhn, J. Axmann, and P. Vörsmann. Sliding Mode Online Learning for Flight Control Applications in Unmanned Aerial Systems. In *WCCI - World Congress on Computational Intelligence*, Barcelona, Spain, July 2010. IEEE. pp. 3738-3745.
- [8] J. Burken, N. T. Nguyen, and B. J. Griffin. Adaptive Flight Control Design with Optimal Control Modification on an F-18 Aircraft Model. In *AIAA Infotech@Aerospace Conference*, Atlanta, Georgia, April 2010. AIAA 2010-3364.
- [9] A. Calise, S. Lee, and M. Sharma. Development of a reconfigurable flight control law for the x-36 tailless fighter aircraft. In *AIAA Guidance, Navigation, and Control Conference*, Denver, CO, September 2000. AIAA 2000-3940.
- [10] R. Rysdyk and A. J. Calise. Robust Nonlinear Adaptive Flight Control for Consistent Handling Qualities. *IEEE Transactions on Control Systems Technology*, 13(6):896–910, 2005.
- [11] F. Holzapfel. *Nichtlineare adaptive Regelung eines unbemannten Fluggerätes*. PhD thesis, Lehrstuhl für Flugmechanik und Flugregelung, Technische Universität München, 2004.
- [12] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.
- [13] F. L. Lewis, A. Yegildirek, and Kai Liu. Multilayer neural-net robot controller with guaranteed tracking performance. *IEEE Transactions on Neural Networks*, 7(2):388–399, 1996.
- [14] O. Kaynak, K. Erbatır, and M. Ertugrul. The Fusion of Computationally Intelligent Methodologies and Sliding-Mode Control - A Survey. *IEEE Transactions on Industrial Electronics*, 48(1):4–17, 2001.
- [15] A. V. Topalov and O. Kaynak. Online Learning in Adaptive Neurocontrol Schemes with a Sliding Mode Algorithm. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 31(3):445–450, 2001.
- [16] A. Nied, S. I. Seleme, G. G. Parma, and B. R. Menezes. On-line neural training algorithm with sliding mode control and adaptive learning rate. *Neurocomputing*, 70:26872691, 2007.
- [17] T. Martin, S. Winkler, and P. Vörsmann. GPS/INS Integration for Autonomous Mini and Micro Aerial Vehicle Navigation Considering Time Correlated Sensor Noise. In *ION 63<sup>rd</sup> Annual Meeting*, Cambridge, Massachusetts, USA, 23 - 25 April 2007. p. 710-715.
- [18] R. Rojas. *Neural Networks - A Systematic Introduction*. Springer-Verlag, Berlin, 1996.