at – Automatisierungstechnik 41 (1993) 9 © R. Oldenbourg Verlag

Übersichtsaufsatz

Lernende Regelung von Totzeitprozessen

Learning control of processes with dead-time

Matthias Seitz, Andreas Kurz und Henning Tolle, Darmstadt

Dipl.-Ing. Matthias Seitz ist wissenschaftlicher Mitarbeiter am Fachgebiet Regelsystemtheorie und Robotik der Technischen Hochschule Darmstadt. Hauptarbeitsfelder: Lernende Systeme, Bildverarbeitung für Anwendungen in der Robotik.

Adresse: Technische Hochschule Darmstadt, Fachgebiet Regelsystemtheorie und Robotik, Landgraf-Georg-Str. 4, D-64283 Darmstadt, Tel. (06151) 167404.

Dipl.-Ing. Andreas Kurz ist wissenschaftlicher Mitarbeiter am Fachgebiet Regelsystemtheorie und Robotik der Technischen Hochschule Darmstadt. Hauptarbeitsfelder: Lernende Systeme, Neuronale Netze, autonome Fahrzeuge.

Adresse: siehe oben, Tel. (06151) 167410.

Prof. Dr. rer. nat. Dipl.-Ing. Henning Tolle ist Ordinarius an der Technischen Hochschule Darmstadt und Direktor am Institut für Regelungstechnik, wo er das Fachgebiet Regelsystemtheorie und Robotik leitet. Hauptarbeitsfelder: Regelkreissynthese, Lernende Regelkreise, Künstliche Intelligenz, Lokal verallgemeinernde (interpolierende) assoziative Speicher, Robotik.

Adresse: siehe oben, Tel. (06151) 163442.

Zum Entwurf von Regelsystemen mit unbekannter oder mathematisch nur schwer modellierbarer Strecke eignen sich neben adaptiven Verfahren vor allem lernende Konzepte. Zur Regelung solcher Prozesse mit a priori unbekannter Totzeit wird ein prädiktiver Regler in den lernenden Regelkreis integriert. Hierfür muß die unbekannte Prozeßtotzeit ermittelt werden. Eine Gegenüberstellung einiger Verfahren, die die Totzeit identifizieren, sowie ihre Anwendung in einem simulierten Beispiel erörtert der folgende Aufsatz.

Beside adaptive methodes especially learning concepts are suitable for the design of control systems with unknown or hardly modelizable processes. A predictive controller is integrated within the learning control loop to make the control of processes with a priori unknown dead-time possible. For this the dead-time has to be identified. The paper gives a comparison of some methods identifying the dead-time as well as a discussion of their application in a simulated example.

1 Einführung

Der Mensch ist in der Lage, komplexe nichtlineare Prozesse verschiedenster Art zufriedenstellend zu regeln. Dabei hat er keine mathematischen Vorstellungen über die Prozeßstruktur, sondern erlernt durch Anweisungen oder Ausprobieren, situationsabhängig geeignete Aktionen vorzunehmen. Bei Wiederauftreten gleicher oder ähnlicher, früher schon erfolgreich behandelter Situationen erinnert er sich an seine Handlungen und reagiert auf der Basis dieser Erinnerungen.

Diese Fähigkeiten des Menschen können, insoweit sie die direkte sensomotorische Koordination und nicht bewußte Schlußfolgerungen auf der linguistischen Ebene betreffen, durch lernende Regelkreise nachgebildet werden. Die von *Saridis* 1977 in [1] beschriebenen statistisch/stochastischen Ansätze wurden von *Ersü* 1983 durch einen einfachen deterministischen Ansatz ergänzt [2]. Bild 1 zeigt den in [2] konzipierten Regelkreis LERNAS, wobei die aktuelle Prozeßsituation $\boldsymbol{\varphi}$ zum Abtastzeitpunkt k (entspricht der Zeit $t = kT_0$ mit der Abtastzeit T_0) durch die Eingänge $\boldsymbol{u}$ und die Ausgänge $\boldsymbol{y}$ des Prozesses und – zur Charakterisierung der Prozeßdynamik – ergänzend durch weitere vergangene Werte dieser Größen beschrieben wird:

$$\boldsymbol{\varphi}(k) = \begin{pmatrix} \boldsymbol{\xi}(k) = [\boldsymbol{u}^T(k-1), \ldots, \boldsymbol{u}^T(k-n_u)]^T \\ \boldsymbol{\psi}(k) = [\boldsymbol{y}^T(k), \ldots, \boldsymbol{y}^T(k-n_y)]^T \end{pmatrix}. \quad (1)$$

Wesentliche Elemente von LERNAS sind zwei Assoziativspeicher (AMS) [3], in denen zur jeweils betrachteten Situation einmal die Prozeßantwort $\boldsymbol{y}(k+1)$ abgelegt wird, was ein prädiktives Prozeßmodell ergibt, und zum anderen eine über ein Gütekriterium als günstig errechnete Stellaktion $\boldsymbol{u}(k)$, wodurch ein Regler realisiert wird. Da beide Speicher über zufällig auftretende Situationen trainiert und im allgemeinen in davon abweichenden Situationen betrachtet werden, sind sie so angelegt, daß für Zwischensituationen die Ausgangswerte automatisch interpoliert werden: Insofern liefert das prädiktive Prozeßmodell nur eine Approximation der Prozeßsituation, weshalb dafür $\hat{\boldsymbol{y}}(k+1)$ statt $\boldsymbol{y}(k+1)$ geschrieben wird. Auch der vom Regler ausgegebene Stellwert ist nur eine Approximation des günstigen Stellwertes, d.h. $\boldsymbol{u}(k) = \hat{\boldsymbol{u}}_{\text{opt}}(k)$. Als rein tabellarische Ablagen können die Speicher beliebige nichtlineare Zusammenhänge im Rahmen der Interpolationsunschärfe charakterisieren. Der lernende Regelkreis ermittelt also für beliebige nichtlineare Prozesse daran angepaßt optimierte nichtlineare Rege-

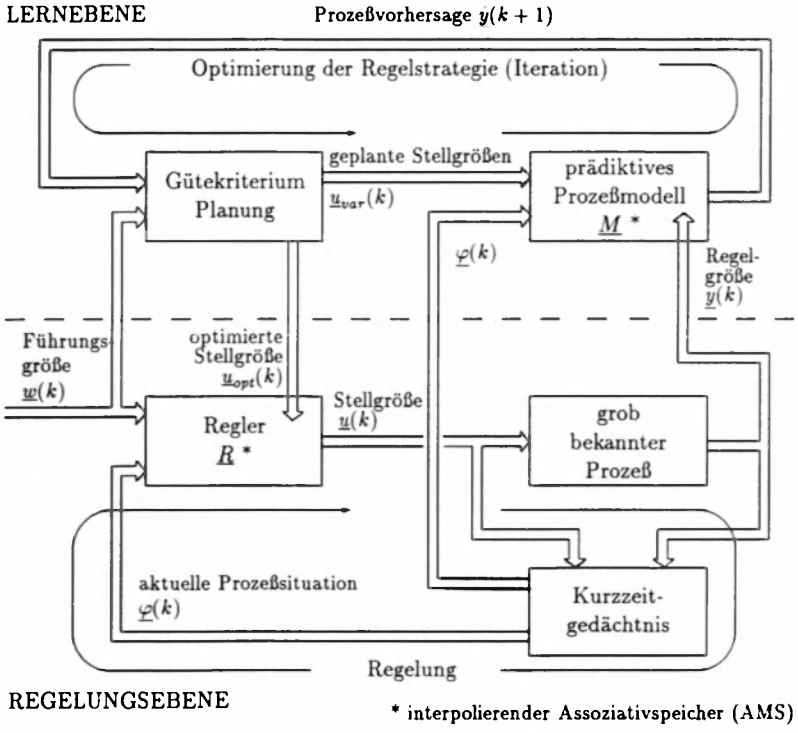


Bild 1. Struktur des lernenden Regelkreises LERNAS nach Ersü [2] ohne Vorkehrungen zur Regelung von totzeitbehafteten Prozessen.

lungen. Vertiefende Darstellungen zu den Speichern und Regelkreisansätzen findet man in [3] und [4], eine ausführliche Erläuterung über die Gesamtheit der in diesem Zusammenhang durchgeführten Untersuchungen in [5].

Für die Konfigurierung des Regelkreises ist neben einer auf Kenntnissen über die Dynamik der Regelstrecke fußenden Festlegung der Abtastzeit insbesondere der Umfang der mit zu erfassenden Vergangenheit, also die Festlegung der Parameter n_u und n_y in (1), von Bedeutung.

Von der adaptiven Regelung her weiß man, daß z. B. beim Prozeßmodell die so beschriebene Dynamik linearisiert im allgemeinen höchstens dritter Ordnung zu sein braucht, wobei aber vielfach explizit eine Totzeit mit berücksichtigt werden muß. Dies bedeutet, daß man zumeist bei $\boldsymbol{y}$ mit $\boldsymbol{y}(k)$, $\boldsymbol{y}(k-1)$ und $\boldsymbol{y}(k-2)$ auskommen wird, während für $\boldsymbol{u}$ die Werte $\boldsymbol{u}(k-d-1)$ und $\boldsymbol{u}(k-d-2)$ mit der Totzeit $T_t \approx dT_0$ im Eingangsraum des Speichers zu berücksichtigen sind. Somit sind für den Umfang der Historien n_u und n_y zwar heuristische Werte angegeben, doch stellt sich bei unbekannter Totzeit T_t zusätzlich das Problem, die richtigen Stellgrößen $\boldsymbol{u}(k-d-1)$ und $\boldsymbol{u}(k-d-2)$ aus der Menge aller vergangenen Stellgrößen $\boldsymbol{u}(k-1), \ldots, \boldsymbol{u}(k-d_{\max}-2)$ mit $0 < d < d_{\max}$ zu finden. Berücksichtigt man alle diese möglichen Stellgrößen, erhöht sich der Eingangsraum des Modellspeichers beträchtlich, so daß es an sich wünschenswert erscheint, die Menge der zu berücksichtigenden Stellgrößen auf das notwendige Minimum zu beschränken. Man kann dies durch die Nutzung von Prozeßkenntnissen und Vorversuche mit verschieden umfangreichen Modellspeichern erreichen.

Man kann sich aber auch fragen, inwieweit nicht eine automatische Selektion des Modellspeicherumfangs aufgrund der Meßsignale am Prozeß denkbar ist. Mit dieser Frage befaßt sich der vorliegende Aufsatz.

Zur Vereinfachung wird zunächst angenommen, daß der Vergangenheitsumfang, der die eigentliche Einschwingdynamik charakterisiert, bekannt sei und nur die Vergangenheit der vorgenommenen Stellaktionen für eine separat erfaßbare Totzeit offen ist. Für die Charakterisierung der Leistungsfähigkeit der Totzeitermittlung und lernenden Regelung bei Vorliegen einer Totzeit wurde der im Bild 2 gezeigte einfache nichtlineare Prozeß benutzt, der aus einem Totzeitglied, einer nichtlinearen Kennlinie und einem linearen Glied mit PT_1-Verhalten zusammengesetzt ist. Ergänzend wurde im linearen Teil auch ein PDT_1-Ansatz betrachtet. Da sich aber dadurch keine wesentlichen Veränderungen bei den Ergebnissen zeigten, wird hier darauf nicht näher eingegangen.

Grundsätzlich verursachen mit größeren Totzeiten behaftete Prozesse regelungstechnische Probleme, da die Stellgröße erst nach einer Verzögerung von d Abtastschritten ($T_t = dT_0$) auf die Regelgröße einwirkt. Dies führt in den meisten Fällen – so auch bei LERNAS bei einem Prozeß mit PT_1-Verhalten und $d = 5$ – zur Instabilität der Regelung (vgl. Bild 7a). Um dies grundsätzlich zu verbessern, kann ein Smith-Prädiktor (vgl. z. B. [6]) benutzt werden. Abschnitt 2 diskutiert entsprechend die Erweiterung des lernenden Regelkreises durch Vorhersage der Regelgröße um die Totzeit in die Zukunft. Zu der dafür notwendigen Identifikation der Totzeit werden drei Ansätze behandelt: Im Abschnitt 3.1 die Ausdehnung des Situationsraums im prädiktiven Prozeßmodell bis zu $n_u > d$, eine – wie sich zeigt – wenig günstige Lösung, im Abschnit 3.2 der alternative Ansatz von mehr als d parallelen Speichern, was eine deutliche Verbesserung bringt aber auch noch aufwendig ist, und im Abschnitt 3.3 die Nutzung von Kreuzkorrelationen, eine sehr einfache, aber nicht ausreichend exakte Methode.

Aufgrund eines Vergleichs der Vor- und Nachteile der drei Ansätze, ergänzt um Aussagen zu der hier nicht näher erörterten aber betrachteten Entfaltung [7], wird im

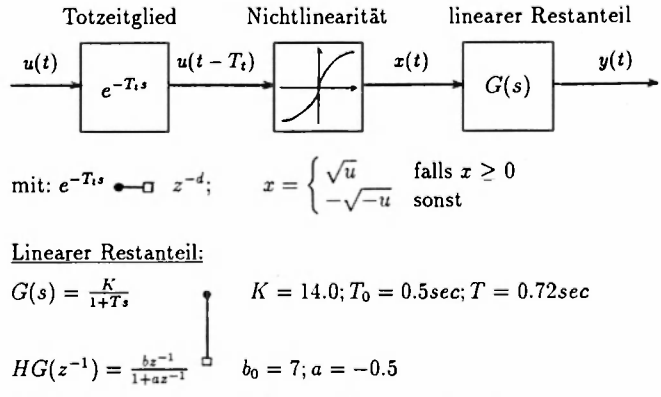
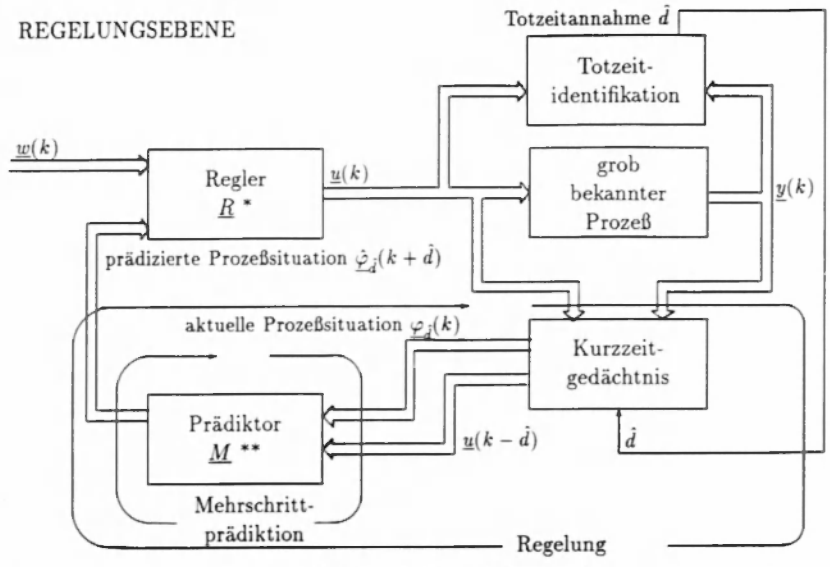


Bild 2. Struktur des simulierten Prozesses mit einer Totzeit von $d = T_t/T_0$ Abtastschritten.

Abschnitt 4 eine Kombination von Kreuzkorrelation und parallelen Speichern als zweckmäßiges Vorgehen zur Totzeitidentifikation vorgeschlagen. In den Schlußbemerkungen werden dann noch einige ergänzende Hinweise, u.a. zur Regelung von Strecken mit Allpaßverhalten, gemacht.

2 Lernende prädiktive Regelung totzeitbehafteter Systeme

Der lernende Regelkreis, wie er im Bild 1 abgebildet ist, ermöglicht im allgemeinen nicht die Regelung von Totzeitprozessen, da die Regelung aufgrund der Totzeit verspätet auf die aktuelle Prozeßsituation $\varphi(k)$ gemäß (1) reagiert. Dies zeigt sich am instabilen Führungsverhalten von LERNAS im Bild 7a, wo der im Bild 2 eingeführte Prozeß mit einer Totzeit von $d = 5$ Abtastschritten simuliert wurde. Um die Regelungsstrategie zu verbessern, muß die Regelgröße um die Totzeit in die Zukunft vorhergesagt werden, was im folgenden durch eine Prädiktion in Anlehnung an den Smith-Prädiktor (vgl. z. B. [6]) realisiert wird. Voraussetzung hierfür ist, daß die Totzeit bekannt ist. Wenn die unbekannte Prozeßtotzeit aus den Eingangs- und Ausgangsdaten des Prozesses ermittelt werden soll, sind spezielle Identifikationsmethoden notwendig. Im Bild 3 ist die um einen Prädiktor und eine Totzeitidentifikationsmethode erweiterte Regelungsebene von LERNAS (vgl. Bild 1) dargestellt, womit eine Regelung von Totzeitprozessen ermöglicht wird.



* Assoziativspeicher (AMS) ** Mehrschrittprädiktion durch prädiktives Prozeßmodell gemäß Gln. 4-6

Bild 3. Zur Regelung totzeitbehafteter Prozesse ist in die Regelungsebene von LERNAS (vgl. Bild 1), ein Prädiktor und ein Identifikationsverfahren zur Ermittlung der Totzeit zu integrieren.

Zur Prädiktion wird zweckmäßigerweise das in der Lernebene vorhandene Prozeßmodell (vgl. Bild 1) verwendet, welches damit zweifach benutzt wird. Während klassische Verfahren (vgl. z. B. [6]) die Modellierung des Prozesses durch zumeist lineare Parametermodelle realisieren, die eine bekannte lineare Prozeßstruktur voraussetzen, wird der Prozeß in LERNAS durch seine in einem Assoziativspeicher (AMS) abgelegten Eingangs- und Ausgangswerte modelliert, wobei keine Strukturvorgaben getroffen werden. Der AMS kann dabei in Anlehnung an die Informationsverarbeitung in der menschlichen Kleinhirnrinde [3] einen nichtlinearen, mehrdimensionalen Zusammenhang abspeichern, so daß dieser Zusammenhang durch Interpolation der abgespeicherten Daten jederzeit ausgelesen werden kann.

Zum Erlernen des Prozeßverhaltens wird in dem zur Prädiktion benutzten prädiktiven Prozeßmodell die Zuordnung $\boldsymbol{M}$ gemäß (3) abgespeichert, bei der die um die Totzeit verzögerte Stellgröße $\boldsymbol{u}(k-\hat{d}-1)$ und die für Totzeitprozesse im Abtastzeitpunkt k interessante Prozeßsituation (vgl. Gl. (1))

$$\varphi_{\hat{d}}(k) = \begin{pmatrix} \xi(k-\hat{d}) = [\boldsymbol{u}^T(k-\hat{d}-1), \ldots, \\ \qquad \boldsymbol{u}^T(k-\hat{d}-n_u)]^T \\ \psi(k) = [\boldsymbol{y}^T(k), \ldots, \boldsymbol{y}^T(k-n_y)]^T \end{pmatrix} \tag{2}$$

als Modelleingänge und die aktuelle Regelgröße $\boldsymbol{y}(k)$ als Modellausgang abgespeichert werden:

$$\boldsymbol{M}: \quad \boldsymbol{u}(k-\hat{d}-1), \varphi_{\hat{d}}(k-1) \rightarrow \boldsymbol{y}(k). \tag{3}$$

Dabei stellt $\hat{d}$ die getroffene Totzeitannahme dar. Durch Assoziation des so abgespeicherten Zusammenhanges kann der Modellspeicher die Regelgröße um einen Abtastschritt in die Zukunft vorhersagen, wenn die Größen $\boldsymbol{u}(k-\hat{d})$ und $\varphi_{\hat{d}}(k)$ als Eingangsgrößen am Modell anliegen:

$$\hat{\boldsymbol{y}}(k+1) = \boldsymbol{M}[\boldsymbol{u}^T(k-\hat{d}), \varphi_{\hat{d}}^T(k)]^T. \tag{4}$$

Um die Regelgröße einen weiteren Schritt in die Zukunft zu prädizieren, wird die Prozeßsituation im nächsten Abtastschritt mit dem soeben geschätzten Wert berechnet:

$$\hat{\varphi}_{\hat{d}}(k+1) = \begin{pmatrix} \xi(k-\hat{d}+1) = [\boldsymbol{u}^T(k-\hat{d}), \ldots, \\ \qquad \boldsymbol{u}^T(k-\hat{d}-n_u+1)]^T \\ \hat{\psi}(k+1) = [\hat{\boldsymbol{y}}^T(k+1), \boldsymbol{y}(k), \ldots, \\ \qquad \boldsymbol{y}^T(k-n_y+1)]^T \end{pmatrix}. \tag{5}$$

Hieraus ergibt sich durch erneutes Auslesen des Modellspeichers die um einen weiteren Abtastschritt in die Zukunft prädizierte Regelgröße:

$$\hat{\boldsymbol{y}}(k+2) = \boldsymbol{M}[\boldsymbol{u}^T(k-\hat{d}+1), \hat{\varphi}_{\hat{d}}^T(k+1)]^T. \tag{6}$$

Dieser Vorgang kann solange wiederholt werden, bis die zur Prädiktorregelung notwendige, um $\hat{d}$ Schritte in die Zukunft prädizierte Prozeßsituation $\hat{\varphi}_{\hat{d}}(k+\hat{d}) = [\xi^T(k), \hat{\psi}^T(k+\hat{d})]^T$ bestimmt ist. Schaltet man sie zusammen mit dem Sollwert $\boldsymbol{w}$ auf den Regler, so kann dieser anhand der abgespeicherten Regelstrategie $\boldsymbol{R}$ eine Stellgröße erzeugen, die den Einfluß der Prozeßtotzeit kompensiert:

$$\boldsymbol{u}(k) = \boldsymbol{R}[\boldsymbol{w}^T(k), \hat{\varphi}_{\hat{d}}^T(k+\hat{d})]^T. \tag{7}$$

An dieser Stelle ist darauf hinzuweisen, daß die Prädiktion keine exakten Werte, sondern lediglich Schätzgrößen hervorruft. Dieser Prädiktionsfehler pflanzt sich bei jedem Iterationsschritt fort, so daß insbesondere bei großen Totzeiten eigentlich nicht vernachlässigbare Fehler entstehen, die zumeist aber durch die Regelung ausgeregelt werden.

Wie die Bilder 7b bis d gegenüber Bild 7a zeigen, funktioniert die Regelung des simulierten Totzeitprozesses in LERNAS mit dem hier eingeführten Prädiktor recht gut, vorausgesetzt die unbekannte Prozeßtotzeit wurde identifiziert [7].

3 Verfahren zur Identifikation der Totzeit

In der Literatur findet man Identifikationsverfahren, die die Totzeit zumeist im Rahmen der Identifikation des Gesamtprozesses ermitteln. Die gebräuchlichsten Methoden sind Parameterschätz- und Korrelationsverfahren (s. z. B. [8], [9]). Beide Verfahren sind im allgemeinen nur bei linearen Prozessen anwendbar, wobei Parameterschätzverfahren von einer Strukturvorgabe in Form eines parametrischen Modells ausgehen. Da mit LERNAS Prozesse unbekannter Struktur (insbesondere nichtlineare Prozesse) geregelt werden, sind die angesprochenen Verfahren in diesem Zusammenhang nicht allgemein genug.

Es besteht jedoch die Möglichkeit, eine Schätzung der Gewichtsfunktion des unbekannten Prozesses durch Entfaltung (vgl. z. B. [9]) vorzunehmen, anhand derer auf die Prozeßtotzeit geschlossen werden kann. Diese Methode bewies in [7] zwar ihre Funktionstüchtigkeit auch bei nichtlinearen Prozessen, doch erfordert die dabei auftretende Inversion der Autokorrelationsmatrix erhebliche Rechenzeit, was die Echtzeitfähigkeit dieses Identifikationsverfahrens beeinträchtigt.

Im folgenden werden allgemein gültige Verfahren vorgestellt, die mit Hilfe der Assoziativspeicher in LERNAS die Totzeit ohne wesentlichen Rechenzeitaufwand identifizieren. Zum einen umfaßt dies die Methode, den Eingangsraum des prädiktiven Prozeßmodells zu erweitern (Abschnitt 3.1), und zum anderen den Signalfehlertest mit parallelen Speichern (Abschnitt 3.2). Da diese Methoden jedoch einen großen Speicherplatzbedarf haben, wird im Abschnitt 3.3 ein einfaches Korrelationsverfahren vorgestellt, das die Totzeit zwar nicht exakt, aber zumeist im Rahmen ihrer Größenordnung ermittelt. Schließlich werden die Verfahren im Abschnitt 3.4 hinsichtlich ihrer Auswirkungen bei der mit LERNAS durchgeführten Regelung des im Bild 2 eingeführten Prozesses vergleichend gegenübergestellt.

3.1 Erweiterung des Situationsraumes

Zur Ermittlung der Totzeit anhand der aufgetretenen Prozeßeingangs- und -ausgangsdaten ist es notwendig, jene Stellgrößen $\boldsymbol{u}(k-d-1), \ldots, \boldsymbol{u}(k-d-n_u)$ aus der Menge aller vergangenen Stellgrößen herauszufinden, die die aktuelle Regelgröße $\boldsymbol{y}(k)$ erzeugt haben. Für unbekanntes d liegt es deshalb nahe, den Situationsvektor $\boldsymbol{\varphi}_d(k)$ für Totzeitprozesse (vgl. Gl. (3)) mit allen vergangenen und relevanten Stellgrößen $\boldsymbol{u}(k-1), \ldots, \boldsymbol{u}(k-d_{\max}-n_u)$ mit $0 < d < d_{\max}$ zu erweitern, um mit den im Modellspeicher abgespeicherten Daten, die Stellgrößen, die die aktuelle Regelgröße beeinflussen, herauszufinden. Dabei stellt $d_{\max}$ die maximal mögliche Totzeit dar:

$$\boldsymbol{\varphi}_d(k) = \begin{pmatrix} \tilde{\xi}(k) = [\boldsymbol{u}^T(k-1), \ldots, \\ \qquad \boldsymbol{u}^T(k-d_{\max}-n_u)]^T \\ \psi(k) = [\boldsymbol{y}^T(k), \ldots, \boldsymbol{y}^T(k-n_y)]^T \end{pmatrix}. \quad (8)$$

Die Idee zur Identifikation der Totzeit beruht auf der Tatsache, daß alle Stellgrößen $\boldsymbol{u}(k-i)\ \forall i \notin [d+1, d+n_u]$ den Prozeßausgang $\boldsymbol{y}$ zum aktuellen Abtastzeitpunkt k nicht beeinflussen. Für alle übrigen Stellgrößen $\boldsymbol{u}(k-d-1), \ldots, \boldsymbol{u}(k-d-n_u)$ ergibt sich dagegen ein im allgemeinen nicht konstanter Zusammenhang mit der abgespeicherten Ausgangsgröße $\boldsymbol{y}(k)$, der sich in der abgespeicherten Datenstruktur erkennen läßt (Näheres s. [7]).

Da der AMS keine exakten, sondern auf der Basis von Interpolationen geschätzte Größen ausgibt, kann es insbesondere bei unzureichend trainiertem Speicher, wie er zu Beginn der Regelung vorliegt, zu Fehlidentifikationen kommen, die nach ausreichendem Training jedoch gegenüber den korrekten Identifikationen in der Minderheit sind. Um diesem Sachverhalt Rechnung zu tragen, werden in einem Vektor $\boldsymbol{dc}$, der im folgenden Totzeitzähler genannt wird, die Identifikationsergebnisse registriert. Anfangs gilt $\boldsymbol{dc} = \boldsymbol{0}$. In jedem Zeitschritt wird die Totzeit $\hat{d} = i\ \forall i \in [0, d_{\max}]$ durch Analyse des Modellspeicherinhaltes geschätzt und die i-te Komponente des Totzeitzählers dc_i inkrementiert, so daß nach einer gewissen Lernphase dc_d die größte aller Totzeitzählerkomponenten ist. Durch Ermittlung der maximalen Komponente des Totzeitzählers läßt sich somit die Totzeit ermitteln.

Im Bild 4 ist der zeitliche Verlauf der Totzeitzählerkomponenten $dc_0(k), \ldots, dc_6(k)$ dargestellt, wobei der im Bild 2 beschriebene Prozeß mit einer Totzeit von $d = 5$ Abtastschritten simuliert wurde. Man sieht, daß die Komponente $dc_5(k)$ erst nach etwa 470 Schritten maximal wird und so LERNAS erst dann von der richtigen

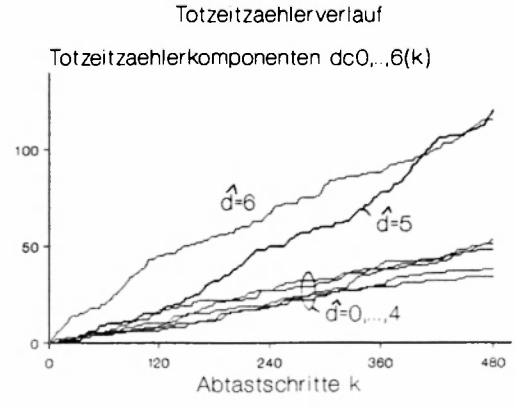


Bild 4. Totzeitzählerverlauf zur Identifikation der Totzeit $d = 5$ bei Erweiterung des Situationsraumes mit 6 zusätzlichen Stellvergangenheiten. Die Totzeit wird erst nach 470 Abtastschritten erfolgreich ermittelt.

Totzeitannahme $\hat{d} = d = 5$ ausgeht. Der Grund für diese lange Phase der Fehlidentifikation liegt darin, daß das prädiktive Prozeßmodell mit erweitertem Eingangsraum nur langsam trainiert werden kann, denn eine Erweiterung des Eingangsraumes führt beim AMS zu einer explosionsartigen Vermehrung von Speicherzellen. Die Anzahl an zu trainierenden Speicherzellen des Modells, die für eine Erkennung der anliegenden Eingangssituation notwendig sind, steigt nämlich mit der Potenz der Anzahl der Eingangsgrößen. Die große Anzahl an Speicherzellen erhöht den Trainingsaufwand des Speichers, was zu einem langsamen Aufbau des Prozeßmodells führt. Dies soll in den folgenden Ansätzen vermieden werden.

3.2 Signalfehlertest mit parallelen Prozeßmodellen

Um den Eingangsraum des prädiktiven Prozeßmodells klein zu halten, wird die gesamte Stellvergangenheit $\xi(k)$ in (8) auf $(d_{max} + 1)$ parallele Speicher verteilt abgelegt, wobei jedes einzelne Prozeßmodell $\boldsymbol{M}_i$ entsprechend (2) und (3) trainiert wird:

$$\boldsymbol{M}_i: \quad \boldsymbol{u}(k-i-1), \varphi_i(k-1) \rightarrow \boldsymbol{y}(k) \; \forall \, i \in [0, d_{max}] \tag{9}$$

mit:

$$\varphi_i(k) = \begin{pmatrix} \xi(k-i) = [\boldsymbol{u}^T(k-i-1), \ldots, \\ \boldsymbol{u}^T(k-i-n_u)]^T \\ \psi(k) = [\boldsymbol{y}^T(k), \ldots, \boldsymbol{y}^T(k-n_y)]^T \end{pmatrix}. \tag{10}$$

Jedes Modell $\boldsymbol{M}_i$ repräsentiert eine Totzeitannahme $\hat{d} = i$, und es gilt nun herauszufinden, welcher Modellspeicher $\boldsymbol{M}_d$ den Prozeß am besten nachbildet. Dies kann durch Bewertung des Fehlers zwischen dem Prozeßausgang und dem Ausgangssignal von jeweils einem Prozeßmodell beurteilt werden [9]:

$$\boldsymbol{e}_i(k) = \boldsymbol{y}(k) - \boldsymbol{M}_i[\boldsymbol{u}^T(k-i-1), \varphi_i^T(k-1)]^T \; \forall \, i \in [0, d_{max}]. \tag{11}$$

Der kleinste Signalfehler $|\boldsymbol{e}_d(k)|$ ergibt sich für den Modellspeicher $\boldsymbol{M}_d$ mit der richtigen Totzeitannahme.

Betrachtet man auch hier den zeitlichen Verlauf der Totzeitzählerkomponenten $dc_i(k)$ im Bild 5, so ergibt sich

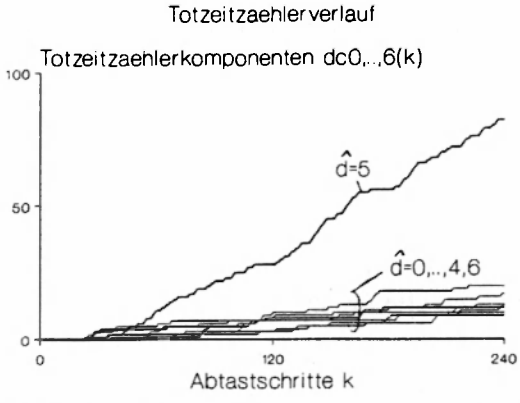


Bild 5. Totzeitzählerverlauf durch Signalfehlertest mit 7 parallelen Modellspeichern bei einem Prozeß mit $d = 5$. Die Totzeit wird bereits nach etwa 60 Abtastschritten korrekt geschätzt.

im Vergleich zu Bild 4 eine deutlich schnellere Identifikation der Prozeßtotzeit $d = 5$. Trotzdem ist vor allem bei großen Totzeiten auch bei dieser Methode der Speicherplatzaufwand erheblich, da dann viele Speicher mit den unterschiedlichen Totzeitannahmen $0 \leq \hat{d} \leq d_{max}$ trainiert werden müssen. Es wäre deshalb wünschenswert, die ungefähre Größenordnung der Totzeit $\hat{d}$ zu kennen, so daß weniger Totzeitannahmen (die sich nur geringfügig von der ungefähren Größenordnung der Totzeit unterscheiden) in Betracht gezogen und dementsprechend weniger Modellspeicher eingerichtet werden müssen.

3.3 Maximalwertermittlung der Kreuzkorrelationsfunktion

Ein Maß für die Verwandtschaft zweier Signale stellt ihre Kreuzkorrelierte dar. Um herauszufinden, welcher Stelleingriff $\boldsymbol{u}(k-d-1)$ den aktuellen Prozeßausgang $\boldsymbol{y}(k)$ zuletzt beeinflußt hat, berechnet man die Kreuzkorrelationsfunktion zwischen Prozeßeingang und -ausgang[1])

$$R_{uy}(\tau) \approx \frac{1}{k-\tau-1} \sum_{\kappa=\tau}^{k} u(\kappa-\tau)\, y(\kappa). \tag{12}$$

Diese Funktion wird in der Korrelationsanalyse [9] zur Identifikation der Gewichtsfunktion $g(k)$ eines linearen Systems mit dem Eingang x und dem Ausgang y (vgl. Bild 2) benutzt:

$$R_{uy}(\tau) = \sum_{k=0}^{\infty} g(k)\, R_{ux}(\tau-k). \tag{13}$$

Nichtlineare Prozesse, wie sie mit LERNAS geregelt werden können, sind mit dieser Gleichung nicht zu identifizieren, da sie nicht durch eine Gewichtsfunktion $g(k)$ beschrieben werden können. Betrachtet man die Struktur des im Bild 2 eingeführten, nichtlinearen Prozesses, so gilt für den nichtlinearen Anteil mit Totzeit jedoch das Bussange'sche Theorem [10], nach dem die Kreuzkorrelierte $R_{ux}(\tau)$ am Ausgang der Nichtlinearität proportional zur Autokorrelierten $R_{uu}(\tau-d)$ des normalverteilten Eingangssignales ist:

$$R_{ux}(\tau) = \frac{1}{N} \sum_{k=0}^{N-1} \frac{\mathrm{d}x(k)}{\mathrm{d}u(k-d)} R_{uu}(\tau-d) = c R_{uu}(\tau-d). \tag{14}$$

Der gesamte Prozeß nach Bild 2 gehorcht dann entsprechend (13) folgendem Zusammenhang (mit $\bar{g}(k) = cg(k)$):

$$R_{uy}(\tau) = \sum_{k=0}^{\infty} \bar{g}(k)\, R_{uu}(\tau-d-k). \tag{15}$$

Die Autokorrelierte R_{uu} ist maximal, wenn ihr Argument Null ist. Somit wird die in (14) betrachtete Kreuzkorrelierte $R_{ux}(\tau)$ maximal, wenn $\tau = d$ ist. Durch Berechnung

[1]) Im folgenden werden die Gleichungen der besseren Übersicht wegen für Eingrößensysteme beschrieben. Man kann jedoch leicht z. B. in (12) einen Vektor $\boldsymbol{R}_{uy}(\tau)$ für Mehrgrößenprozesse einführen.

der Kreuzkorrelierten zwischen Eingangs- und Ausgangssignal läßt sich also direkt auf die Totzeit schließen, wenn das System gedächtnislos und der nichtlineare Zusammenhang hystereselos, stetig und ohne Begrenzung ist. Es stellt sich nun die Frage, welche Anforderungen an einen linearen Prozeßteil mit Gedächtnis zu stellen sind, damit der gefundene Zusammenhang nach (14) auch für dynamische Prozesse bestehen bleibt.

Da die Gewichtsfunktion eines stabilen Verzögerungsprozesses erster Ordnung, wie er im Bild 2 dargestellt ist, monoton fällt, ergibt sich hierfür im allgemeinen auch ein Maximum der Kreuzkorrelierten $R_{uy}(\tau)$ bei $\tau = d$. Der zeitliche Verlauf der Komponenten der Kreuzkorrelationsfunktion $R_{uy}(0), \ldots, R_{uy}(d_{max})$ ist im Bild 6 dargestellt und veranschaulicht die Funktionsfähigkeit des Verfahrens bei dem im Bild 2 eingeführten, simulierten Prozeß mit einer Totzeit von $d = 2$.

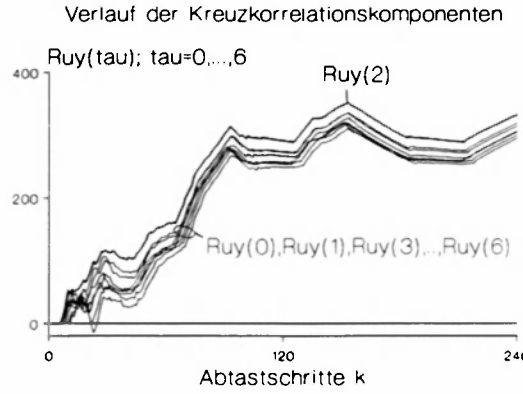


Bild 6. Zeitlicher Verlauf der Komponenten $R_{uy}(\tau)$ der Kreuzkorrelierten. Die Totzeit $d = 2$ des simulierten Prozesses mit PT_1-Verhalten kann ermittelt werden, weil die Komponente $R_{uy}(d)$ den größten Signalverlauf aufweist. Die Identifikation ist zuverlässig, wenn sich der Abstand der Signalverläufe nicht mehr wesentlich verändert.

Es muß jedoch erwähnt werden, daß eine monoton fallende Gewichtsfunktion nicht bei allen Prozessen gegeben ist. Systeme mit D-Anteil, großer Dynamik, insbesondere schwingende Systeme, haben dagegen das Maximum ihrer Gewichtsfunktion im allgemeinen nicht im Ursprung, so daß hierbei auch die Kreuzkorrelierte nicht für $\tau = d$ maximal ist. Versuche mit einem PDT_1-Prozeß ergeben aber, daß die Unterschiede im Führungsverhalten bei dieser Methode, trotz einer für diesen Prozeß um einen Abtastschritt zu groß geschätzten Totzeit, gegenüber dem Verfahren mit Signalfehlertest nicht wesentlich waren. Ob eine Prädiktion mit der groben Größenordnung $\tilde{d}$ der Totzeit für das Führungsverhalten von LERNAS ausreicht, hängt aber von den Genauigkeitsanforderungen des Einzelfalles ab.

Durch die Bestimmung der maximalen Kreuzkorrelationskomponente ist somit ein einfaches Verfahren mit vergleichsweise geringem Aufwand zur Totzeit-

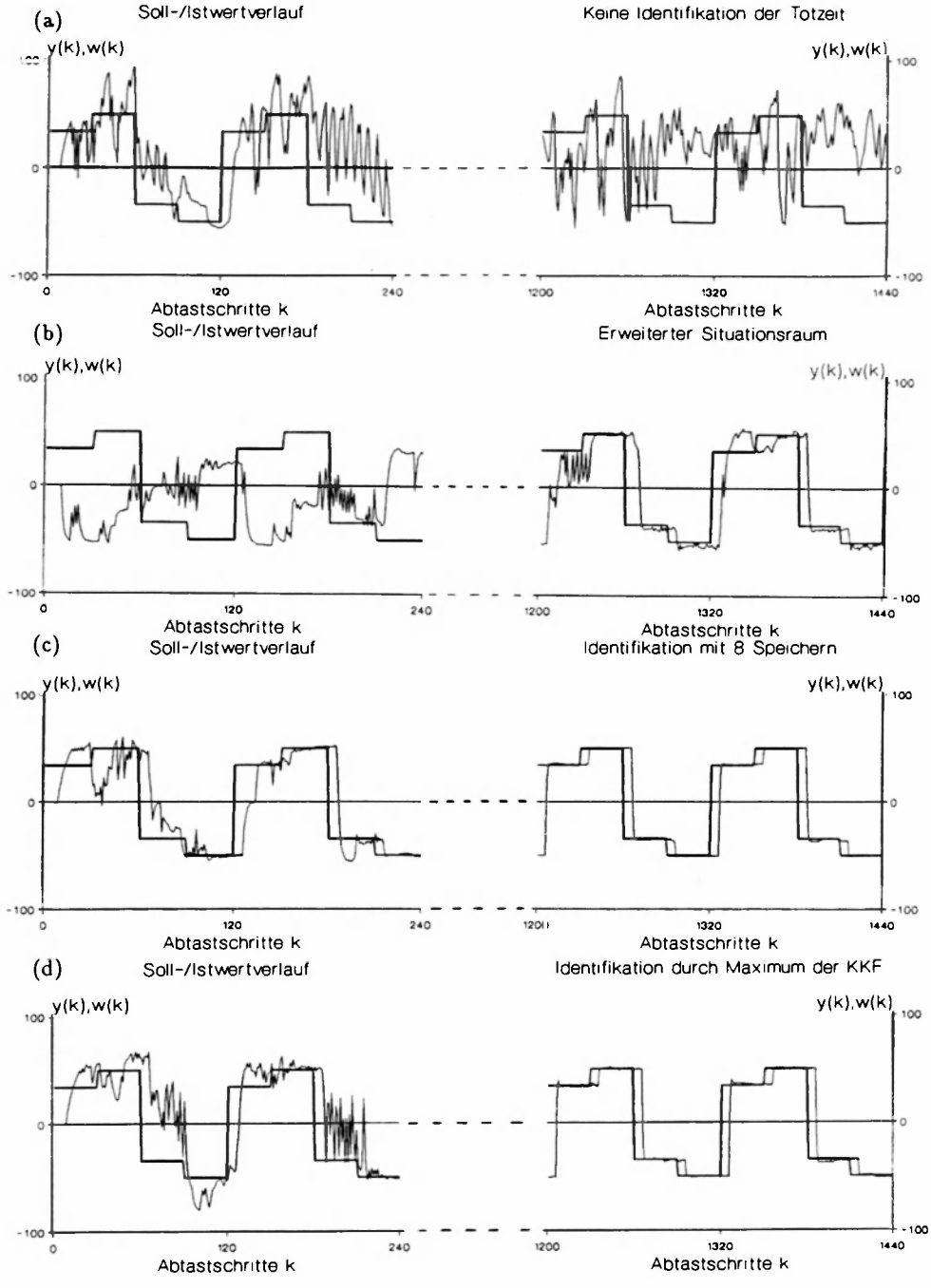


Bild 7. Führungsverhalten des LERNAS bei dem Prozeß nach Bild 2 mit der Totzeit $d = 5$. Ohne Prädiktor (a) ergibt sich ein instabiles Verhalten. Mit Prädiktor (b) bis (d) zeigt sich bei einem Prozeßmodell mit erweitertem Eingangsraum (b) nur eine grobe Soll-/Istwertanpassung, ansonsten ergibt sich ein zufriedenstellendes Führungsverhalten bei Identifikation durch Signalfehlertest (c) ebenso wie bei Maximalwertermittlung der Kreuzkorrelierten (d).

identifikation gefunden worden, das für Prozesse erster Ordnung mit Tiefpaßverhalten die Totzeit richtig schätzt und ansonsten zumindest die Größenordnung der Totzeit liefert, die zumeist schon ausreicht, um den Prozeß zu regeln.

3.4 Vergleich der Verfahren

Nachdem gezeigt wurde, daß die in den Abschnitten 3.1 bis 3.3 beschriebenen Verfahren mit den angesprochenen Randbedingungen die Totzeit identifizieren können, müßte dies im wesentlichen auch zu ähnlichen Ergebnissen im Soll-/Istwertverhalten bei einer LERNAS-Regelung des nichtlinearen Prozesses mit PT_1-Verhalten (Bild 2) und relativ großer Totzeit von 5 Abtastschritten führen. Ein Vergleich der Soll-/Istwertverläufe nach 1200 Abtastschritten und anfangs leeren AMS-Speichern im Bild 7 zeigt, daß die Einführung von Prädiktor und Totzeitidentifikation nach Bild 3 zwar prinzipiell die Regelung von Totzeitprozessen ermöglicht (vgl. Bild 7a mit den Bildern 7b bis d), sich wegen der verschiedenen Identifikationsmethoden aber Unterschiede im Führungsverhalten ergeben (vgl. Bilder 7b bis d).

	Erweiterung des Situationsraumes (Absch. 3.1)	Parallele Prozeßmodelle (Absch. 3.2)	Maximum der Kreuzkorrelierten (Absch. 3.3)	Entfaltung (vgl. [9])
Speicherplatz	sehr hoch	hoch	sehr niedrig	sehr niedrig
Rechenzeit (Regelungsebene)	niedrig	niedrig	niedrig	sehr hoch
Rechenzeit (Lernebene)	sehr hoch	niedrig	niedrig	niedrig
Lernaufwand (Identifikationszeit)	hoch bei grossen Totzeiten	niedrig	niedrig	niedrig
Anforderungen an den Prozeß	wenig Eingänge	keine	g(k) monoton fallend, kein Hochpaß, kleine Dynamik, kl. Verzögerungsordnung, Nichtlinearität ohne Gedächtnis und Begrenzung	kl. Dynamik, Nichtlinearität ohne Gedächtnis und Begrenzung
Strukturerkennung	möglich	möglich	nicht möglich	eingeschränkt möglich
Erkennung wechselnder Totzeiten	erneuter Lernaufwand	erneuter Lernaufwand	sofort möglich	sofort möglich
numerische Instabilität	keine	keine	keine	evtl. singuläre Autokorrelationsmatrix

Bild 8. Qualitativer Vergleich von Verfahren zur Totzeitidentifikation.

Das gegenüber den anderen Methoden relativ schlechte Führungsverhalten von LERNAS bei einem Prozeßmodell mit erweitertem Eingangsraum (Bild 7b) läßt sich vor allem bei großen Totzeiten mit der hohen Anzahl an zu trainierenden Speicherzellen begründen[2]), die das Erlernen des Prozeßverhaltens verlangsamen und erschweren. Dadurch wird meistens eine nicht optimale Regelungsstrategie geplant, was zu einer relativ schlechten Regelung im Bild 7b führt.

Dagegen zeigt der Signalfehlertest ebenso wie die Methode, die die Totzeit durch Maximalwertermittlung der Kreuzkorrelierten bestimmt (Bilder 7c, d), bei dem untersuchten Prozeß nach 1200 Abtastschritten ein zufriedenstellendes Führungsverhalten, weil das Prozeßverhalten durch das prädiktive Modell relativ schnell erlernt wird.

Als Zusammenfassung des Vergleichs der Verfahren ist im Bild 8 eine Tabelle dargestellt, die qualitative Aussagen hinsichtlich der wichtigsten Kriterien zur Beurteilung der Verfahren gegenüberstellt. Dabei werden auch die Anwendbarkeit der Verfahren bei zeitvarianten Totzeiten und ihre Nutzung zur Erkennung der Prozeßstruktur verglichen.

Bei den Verfahren mit assoziativen Speichern ergeben sich Schwierigkeiten, einen eventuellen Wechsel der Totzeit festzustellen. Für diesen Fall eignen sich die Korrelationsverfahren besser, denn das Maximum der Korrelationsfunktion hängt direkt von einer Zeitverschiebung τ ab, die der Totzeitschätzung $\hat{d}$ entspricht, und ermöglicht die sofortige Erkennung eines Totzeitwechsels ohne zusätzlichen Lernaufwand.

Eine andere Anwendungsmöglichkeit der dargestellten Methoden zur Totzeitidentifikation ist die automatische Dynamikidentifikation zur Ermittlung der Prozeßstruktur. Statt vergangener Stellgrößen können in (9) natürlich auch verschiedene Prozeßausgangswerte $y(k-i)$ auf die parallelen Speicher aufgeschaltet werden. Durch Signalfehlertest kann dann überprüft werden, welche vergangenen Ausgangsgrößen $y(k-1)$, $y(k-2)$, ... zur Prozeßdynamik beitragen. Ähnliches ist auch für einen Modellspeicher mit erweitertem Eingangsraum denkbar. Somit bestehen Verfahren, mit denen die Zusammensetzung des Situationsvektors nach (2) durch Bestimmung der Totzeit und der notwendigen Historien n_u und n_y ermittelt werden kann. Eine Strukturerkennung zur Bestimmung der Historien n_u und n_y wie oben angesprochen ist durch Korrelationsverfahren nicht möglich, da zwischen der Regelgröße $y(k)$ und den Zuständen $y(k-1), y(k-2), \ldots$ über die Autokorrelationsfunktion von Natur aus ein Zusammenhang besteht, unabhängig davon, ob die Zustände die Regelgröße aufgrund der Prozeßdynamik beeinflussen oder nicht. Als Ergebnis kann man festhalten, daß zumindest zur groben Bestimmung der Größenordnung der Totzeit die Maximalwertermittlung der Kreuzkorrelationsfunktion eine Methode mit sehr geringem Aufwand darstellt. Zur genaueren Ermittlung der Totzeit ist der Signalfehlertest mit parallelen Modellspeichern die wirkungsvollste und universellste Methode, wobei durch Verknüpfung der zuletzt genannten Verfahren die Anzahl der notwendigen Speicher reduziert werden kann, was im nächsten Abschnitt behandelt wird.

[2]) Während bei den anderen Verfahren der Eingangsraum des AMS 2 Eingangsgrößen umfaßt, sind es hier mindestens 7 Eingänge.

4 Totzeitidentifikation durch effektive Verknüpfung zweier Verfahren

Der Vergleich der Methoden ergab, daß durch den Signalfehlertest die Totzeit zwar prinzipiell für alle Prozesse ermittelt werden kann, doch nur unter Einsatz von großer Speicherplatzkapazität. Vor diesem Hintergrund ergibt sich die Möglichkeit, zur effektiveren Ermittlung der Totzeit zwei Identifikationsverfahren zu verwenden, nämlich zum einen den Signalfehlertest, der ohne Vorwissen über die Totzeit aufwendig ist, und zum anderen die Maximalwertermittlung der Kreuzkorrelierten, die die Totzeit ohne wesentlichen Aufwand, aber nur ungenau ermittelt.

Dabei wird zunächst die Größenordnung der Totzeit $\tilde{d}$ durch das Korrelationsverfahren bestimmt und danach durch Vergleich der Vorhersagen paralleler Prozeßmodelle die Totzeit genauer ermittelt. Der Vorteil der Kombination beider Methoden liegt darin, daß für den Signalfehlertest nicht mehr so viele Speicher erforderlich sind wie bei unbekannter Größenordnung der Totzeit. Durch die Maximalwertermittlung der Kreuzkorrelierten wird die Totzeit in den meisten Fällen zu groß geschätzt [7], so daß somit für eine Totzeit d mit $\tilde{d} - \Delta d/2 < d < \tilde{d} + \Delta d/2$ nur noch $\Delta d/2 \ll d_{max}$ zusätzliche Modellspeicher erforderlich sind, was bei einer großen Totzeit weit weniger ist als die dann anzusetzende, maximal mögliche Totzeit d_{max}.

Wie Simulationen in [7] ergaben, führt die genauere Schätzung der Totzeit gegenüber der Maximalwertermittlung der Kreuzkorrelierten alleine zu einem qualitativ nur wenig verbesserten Führungsverhalten. Die erforderliche Regelungsgüte ist jedoch anwendungs- und prozeßgebunden, so daß man sich für solche Prozesse, die auch mit falscher Totzeitannahme geregelt werden können, mit der ermittelten ungefähren Größenordnung der Totzeit zufrieden geben kann und für Prozesse, die eine genauere Prädiktion erfordern, zusätzlich parallele Speicher zur Totzeitidentifikation einrichtet.

5 Schlußfolgerungen

Zusammenfassend lassen sich die Vor- und Nachteile der Verfahren so darstellen, daß die Methoden mit Assoziativspeichern viel Speicherplatz, dagegen aber wenig Rechenzeit beanspruchen. Für diese Verfahren ist es generell von Bedeutung, daß zumindest die grobe Größenordnung der Totzeit bekannt ist, um Rechenzeit- bzw. Speicherplatzaufwand in Grenzen zu halten. Große Totzeiten verursachen vor allem bei Erweiterung des Situationsraumes eine Verschlechterung der Regelung, was die Anwendbarkeit für Prozesse mit großer Totzeit einschränkt. Durch die Maximalwertbestimmung der Kreuzkorrelierten kann die Größenordnung der Totzeit ermittelt werden. Diese Methode ist zur exakten Identifikation der Totzeit jedoch nur bei Prozessen mit einem linearen Prozeßanteil, der durch eine monoton fallende Gewichtsfolge gekennzeichnet ist, sowie mit gedächtnisloser Nichtlinearität ohne Begrenzung anwendbar. Einzig und allein der Signalfehlertest mit parallelen Prozeßmodellen ist uneingeschränkt für alle Prozesse anwendbar. Um Speicherplatz zu sparen, kann durch eine Kombination der beiden zuletzt angesprochenen Verfahren die Totzeit ohne wesentliche Erhöhung des Rechenzeitaufwandes exakt bestimmt werden. Die Anwendbarkeit der vorgestellten Verfahren im lernenden Regelkreis ist hinsichtlich zeitvarianter Totzeiten und automatischer Dynamikerkennung beurteilt worden.

Weiterhin fragt man sich, ob die vorgestellten Verfahren zur Regelung von Totzeitprozessen auch auf andere nichtphasenminimale Prozesse angewendet werden können. Da ein Allpaß eine Näherung für Totzeitprozesse ist, stellt sich die Frage, ob der Prädiktorregler ausreicht, um ein Allpaßverhalten des Prozesses zumindest teilweise zu kompensieren und trotzdem eine stabile Regelung zu gewährleisten. Experimentelle Untersuchungen einer Prädiktorregelung von Allpaßprozessen ergaben aber, daß eine Regelung von Allpaßprozessen mit den hier untersuchten Ansätzen zur Regelung von Totzeitprozessen in LERNAS nicht realisierbar ist [7]. Die Schwierigkeiten liegen dabei in der Identifikation des Allpaßgliedes und der Modellierung des allpaßfreien Prozeßanteiles, vor allem dann, wenn der Prozeß weitgehend unbekannt ist. Der Prädiktor ist nämlich nur in der Lage, den verspäteten Stelleinfluß, nicht aber die anfängliche Prozeßausrichtung entgegen der Sollvorgabe zu kompensieren. Eine laufende Arbeit am Fachgebiet Regelsystemtheorie und Robotik der TH Darmstadt beschäftigt sich mit der Möglichkeit, Totzeit- und Allpaßsysteme in LERNAS durch variable Abtastzeit, die durch das Ausgangssignal des Prozesses automatisch bestimmt wird, regeln zu können.

Schließlich ist bei den aufgezeigten Ergebnissen und Schlußfolgerungen zu berücksichtigen, daß der notwendige Identifikationsaufwand bei konstanter Totzeit nur solange aufrecht erhalten werden muß, wie die Totzeit noch nicht ausreichend zuverlässig ermittelt wurde. Somit sind zwei Phasen bei der lernenden Regelung von Totzeitprozessen zu unterscheiden:

- In der Identifikationsphase wird die Totzeit unter Inkaufnahme eines relativ großen Speicherplatzaufwandes geschätzt. Das Führungsverhalten wird anfangs insbesondere wegen unzureichender Modellierung infolge falscher Totzeitannahmen und dementsprechend ungenauer Prädiktion beeinträchtigt.
- In der Regelungsphase ist die Totzeit als bekannt anzusehen, so daß lediglich Prädiktionsfehler die Qualität der Regelung vermindern können. Diese Fehler werden im allgemeinen aber ausgeregelt.

Ein Parallellaufen beider Phasen ist nur bei zeitvarianten Totzeiten erforderlich, wobei dann die Maximalwertermittlung der Kreuzkorrelierten als Identifikationsverfahren zur Anzeige einer Totzeitänderung ausreicht.

Danksagung

Für den Hinweis auf das Bussang'sche Theorem sind wir Dr.-Ing. *J. Cezanne* vom Fachgebiet Theorie der Signale der TH Darmstadt zu Dank verpflichtet.

Literatur

[1] *Saridis, G. N.:* Self-Organizing Control of Stochastic Systems. Control and Systems Theory, Vol. 4. Verlag: Marcel Dekker Inc., New York 1977.

[2] *Ersü, E.:* On the Application of Associative Neural Network Models to Technical Control Problems. In: Location and Orientation in Biology and Engineering. Hrsg.: *D. Varju* und *H. U. Schnitzler*. Springer-Verlag, Berlin, Heidelberg 1984.

[3] *Tolle, H., Militzer, J.,* und *Ersü, E.:* Zur Leistungsfähigkeit lokal verallgemeinender assoziativer Speicher und ihren Einsatzmöglichkeiten in lernenden Regelungen. messen-steuern-regeln 32 (1989) 3, S. 98–105.

[4] *Tolle, H.:* Lernende Regelung für die Automatisierungstechnik. Proceedings zum Aussprachetag des VDI/VDE-GMA-FA 1.6. Langen 1991.

[5] *Tolle, H.,* und *Ersü, E.:* Neurocontrol – Learning Control Systems Inspired by Neuronal Architectures and Human Problem Solving Strategies. Springer-Verlag, Berlin, Heidelberg 1992.

[6] *Cao, C. T.:* Entwurf eines robusten prädiktiven Reglers für totzeitbehaftete Regelsysteme. Automatisierungstechnik 37 (1989) 3, S. 104–110.

[7] *Seitz, M.:* Regelung von Totzeitprozessen mit dem lernenden Regelkreiskonzept LERNAS. Interner Bericht am FG Regelsystemtheorie und Robotik im Institut für Regelungstechnik der TH Darmstadt 1992.

[8] *Kurz, H.,* und *Goedecke, W.:* Digital Parameter-Adaptive Control of Processes with Unknown Dead Time. Automatica 17 (1981) 1, S. 245–252.

[9] *Unbehauen, H.:* Regelungstechnik III. Verlag Fr. Vieweg & Sohn, Braunschweig, Wiesbaden 1987.

[10] *Papoulis, A.:* Probability, Random Variables and Stochastic Processes. McGraw-Hill International Editions 1984.

Verfügbar unter
lediglich die vom Gesetz vorgesehenen Nutzungsrechte gemäß UrhG

Manuskripteingang: 12. Mai 1992.