

# Po2-4

## Online-Selbst-Identifikation eines dynamischen Systems

Benedikt Böttcher<sup>1</sup>, Peter Zentgraf<sup>2</sup>, Adrian Zeitler<sup>3</sup>

### 1 Abstract

Im Rahmen der Digitalisierung der industriellen Produktion mit Industrie 4.0 und des autonomen Fahrens erlangen möglichst detaillierte digitale Abbilder von Maschinen und Fahrzeugen für den vollständigen Zeitraum ihres Einsatzes eine zentrale Bedeutung. Sowohl durch geplante Veränderungen des Arbeitspunkts aber auch durch Abnutzung und Verschleiß kommt es zu Abweichungen des Systemverhaltens. In diesem Posterbeitrag werden die Entwicklung und die Anwendung eines Identifikations-Algorithmus für die automatisierte Modellierung eines dynamischen Systems basierend auf einem gemessenen Ein- und Ausgangssignal beschrieben.

Der in MATLAB und Simulink programmierte Algorithmus läuft parallel zum realen System auf einer embedded Hardware ab, so dass eine Identifikation online und selbstständig ausgeführt wird. Die prozess- oder verschleißbedingten Veränderungen des Systemverhaltens werden damit unmittelbar bei der Modellierung während des Betriebs berücksichtigt. Im Anschluss an die Identifikation sind die Selbst-Optimierung, die Vorsteuerung und die Selbst-Konfiguration des Systems, z.B. die Anpassung von Reglern, basierend auf dem aktuellen Modell durchführbar.

### 2 Aufbau und Ablauf der Online-Selbst-Identifikation

Der Algorithmus der Online-Selbst-Identifikation ist in einem Simulink-Modell realisiert und wird nach der C-Code-Generierung auf einem Mikrocontroller oder einem IPC ausgeführt, der über das TCP/IP-Internetprotokoll die Messwerte für das Ein- und Ausgangssignal von dem zu modellierenden realen System empfängt. Die Teilaufgaben des Algorithmus stellen die Aufbereitung der empfangenen Messsignale und die experimentelle Modellbildung mit der linearen Regression dar (siehe Abbildung 1). Im Anschluss an eine Identifikation wird das Ausgangssignal mit einer Differenzgleichung (DcGL) online und in Echtzeit simuliert. Die Entwicklung des Algorithmus umfasst die Formulierung eines automatisierbaren Verfahrens für die Überführung der geschätzten DGL in eine DcGL unter Anwendung der Tustin-Methode. Schließlich resultieren ein fortlaufend aktualisiertes Modell des Systems und der simulierte Systemausgang.

---

<sup>1</sup> M.Sc. Benedikt Böttcher, KRONES AG, Neutraubling

<sup>2</sup> Prof. Dr.-Ing. Peter Zentgraf M.Sc, Technische Hochschule Rosenheim, Rosenheim

<sup>3</sup> B.Eng. Adrian Zeitler, Technische Hochschule Rosenheim, Rosenheim

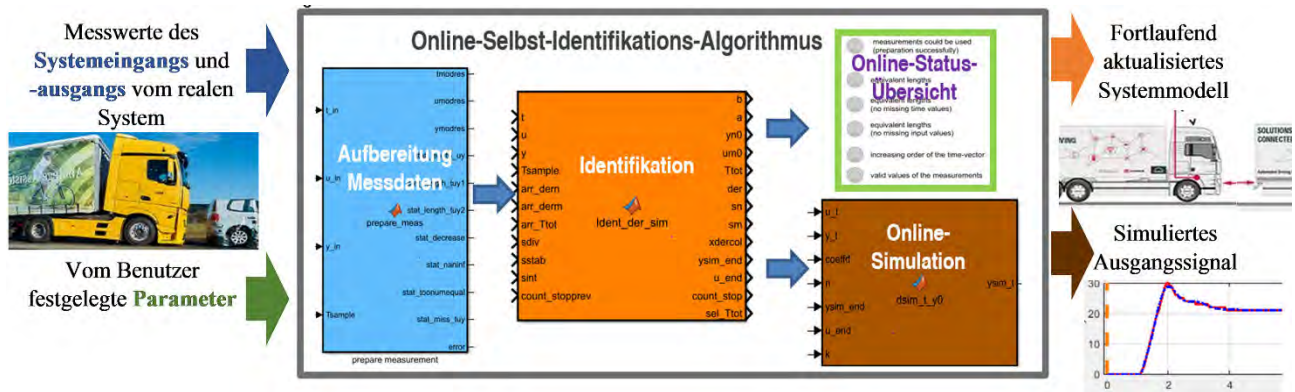


Abbildung 1: Aufbau und Ablauf der Online-Selbst-Identifikation eines dynamischen Systems

Als Abgrenzung zu den Algorithmen der System Identification Toolbox [1] wird mit dem hier vorgestellten Identifikations-Algorithmus eine vollständig automatisierte Modellbildung eines im Betrieb befindlichen Systems mit einem veränderlichen Systemverhalten auf einer embedded Hardware ermöglicht. Des Weiteren sind zusätzliche Verfahren für die Optimierung der Güte des resultierenden Modells, z.B. die Variation des Koeffizienten für die Division der DGL (siehe Abschnitt 3), und die Echtsimulation des Ausgangs durch ein neues Verfahren für die automatisierte Überführung der identifizierten DGL in eine DcGL implementiert.

### 3 Lineare Regression für die Schätzung der Parameter einer inhomogenen Differentialgleichung mit beliebiger Ordnung

Die Systembeschreibung erfolgt durch eine inhomogene Differentialgleichung (DGL), deren Koeffizienten  $\underline{a}$  und  $\underline{b}$  und Anfangsbedingungen  $\underline{y}_{n0}$  mit der linearen Regression geschätzt werden (siehe Gl.1).

$$\begin{aligned}
 b_n * y^{(n)} + b_{n-1} * y^{(n-1)} + \dots + b_1 * \dot{y} + b_0 * y &= \\
 = a_0 * u + a_1 * \dot{u} + \dots + a_{m-1} * u^{(m-1)} + a_m * u^{(m)} & \quad \text{(Gl. 1)}
 \end{aligned}$$

mit:  $\underline{a} = [a_m, a_{m-1}, \dots, a_0] \in \mathbb{R}^{m+1}$      $\underline{b} = [b_n, b_{n-1}, \dots, b_0] \in \mathbb{R}^{n+1}$      $\underline{y}_{n0} = [\tilde{y}_0, \dot{y}_0, \dots, y_0^{(n-1)}]$

Für die Schätzung der unbekannt Parameter wird das Gleichungssystems

$$\underline{\tilde{y}} = H * \underline{x} + \underline{e} \quad \text{(Gl. 2)}$$

basierend auf den eingehenden Messsignalen des Algorithmus aufgestellt. Der Vektor  $\underline{x}$  beinhaltet die unbekannt Koeffizienten der DGL und die Anfangsbedingungen des Ausgangssignals. Als bekannte Größen werden das Ein- und Ausgangssignal  $\underline{\tilde{u}}$  und  $\underline{\tilde{y}}$  innerhalb des Mess-Zeitraums von  $\tilde{t}_1 = 0s$  bis  $\tilde{t}_p = (p - 1) * T_{sample}$  mit der Anzahl an Messwerten  $p$  vorausgesetzt.

In jeder Zeile des oberen Gleichungssystems (Gl. 2) wird der Schätzwert für den Systemausgang  $H(i, :) * \underline{x}$  an einem Zeitpunkt  $\tilde{t}_i$  mit dem gemessenen Systemausgang  $\tilde{y}_i$  verglichen. Die Differenz zwischen Mess- und Schätzwert wird als Schätzfehler  $e_i$  bezeichnet. Das Gleichungssystem wird unter Verwendung der Pseudo-Inversen gelöst, sodass die Summe der Quadrate der Schätzfehler minimal ist [2].

Die Formulierung des Gleichungssystems der linearen Regression basierend auf der oberen inhomogenen DGL erfolgt mit dem in [3] erläuterten Verfahren. Neben den Koeffizienten der DGL können die Anfangsbedingungen mithilfe dieses Verfahrens geschätzt werden. Nur durch dieses Merkmal wird eine Online-System-Identifikation während des Betriebs des Systems, also im Allgemeinen außerhalb dessen Ruhelage, ermöglicht.

Um die Modellgüte zu verbessern, finden die Variation des Nenner- und des Zählergrads  $n$  und  $m$  der zur DGL korrespondierenden kontinuierlichen Übertragungsfunktion und die Variation der Koeffizienten der DGL für die Division innerhalb der Online-Selbst-Identifikation statt. Dafür ist ein allgemein formuliertes Gleichungssystem in Abhängigkeit dieser variierenden Parameter implementiert. Zum Beispiel resultiert für die Division der DGL mit dem unbekanntem Koeffizienten  $b_n$  :

$$y(t) - y_0 = -\sum_{i=1}^{n-1} \widetilde{b}_{n-i} * \left( y_{(i)}(t) - \frac{1}{i!} * t^i * y_0 \right) - \widetilde{b}_0 * y_{(n)}(t) + \sum_{i=1}^{n-1} y_{i0} * \frac{1}{i!} * t^i + \sum_{i=0}^{m-1} \widetilde{a}_{m-i} * \\ * \left( u_{(n+i-m)}(t) - \frac{1}{(n+i-m)!} * t^{n+i-m} * u_0 \right) + \widetilde{a}_0 * u_{(n)}(t) - \sum_{i=1}^{m-1} u_{i0} * \frac{1}{(n+i-m)!} * t^{n+i-m} \quad (\text{Gl. 3})$$

In der oberen Gleichung wird die  $r$ -malige Integration eines Signals mit dem Index ( $r$ ) abgekürzt. Die Parameter  $u_{i0}$  und  $y_{i0}$  sind Zwischengrößen, die von den gesuchten Koeffizienten und den Anfangsbedingungen abhängen. Eine zusätzliche Verbesserung der Modellgüte wird durch die Schätzung einer Totzeit im Identifikations-Algorithmus erreicht. Hierfür wird das gemessene Ausgangssignal auf der Zeitachse nach links vor der Identifikation verschoben.

Um die resultierenden Modelle mit den variierenden Parametern hinsichtlich der Beschreibung des realen Systemverhaltens vergleichen zu können, werden verschiedene Maße für die Güten der Modelle, die von dem Schätzfehler  $\underline{e}$  abhängen, im Algorithmus berechnet. Für die Ermittlung des Schätzfehlers ist das Ausgangssignal für den gemessenen Eingang zu simulieren.

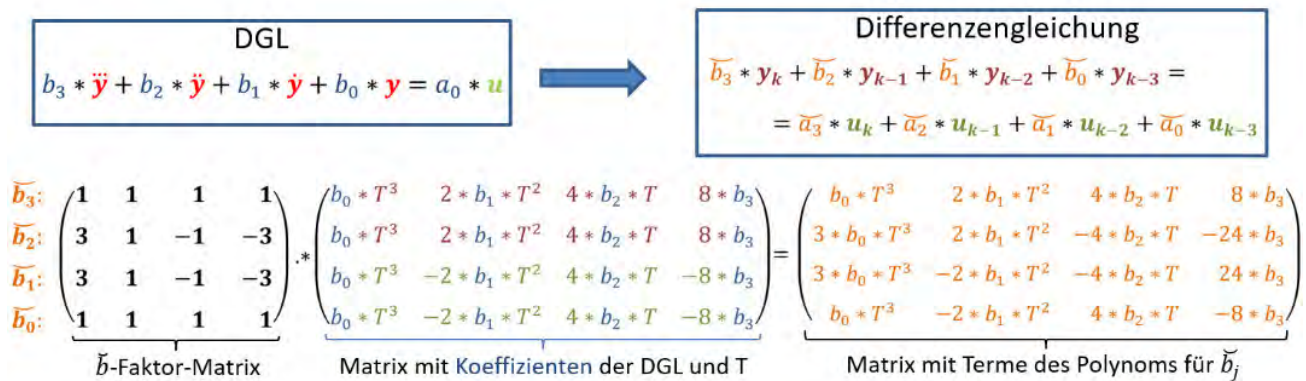
#### 4 Randbedingung für die Online-Realisierung

Das oben vorgestellte Verfahren zur experimentellen Modellbildung mit den Freiheitsgraden für die Optimierung der Modellgüte muss so angepasst werden, dass die Identifikation online, also während des Betriebs des Systems, auf einer embedded Hardware automatisiert ausgeführt wird. Die grundsätzliche Randbedingung der Online-Realisierung ist daher die Generierung von C-Code aus dem Simulink-Modell des Identifikations-Algorithmus.

Für die Simulation des Ausgangssignals einer identifizierten kontinuierlichen Übertragungsfunktion stehen weder die MATLAB-Funktion *lsim()*, in der *tf*- und *ss*-Objektvariablen angelegt werden und die keine C-Code-Generierung unterstützt, zur Verfügung. Noch kann der *Transfer-function* Block, dessen Übertragungsfunktion während der Code-Ausführung nicht anpassbar ist, eingesetzt werden. Als zentrale Randbedingung für die Online-Selbst-Identifikation ist das simulierte Ausgangssignal mit einer korrespondierenden Differenzengleichung (DcGL) zu berechnen.

#### 5 Simulation des Ausgangssignals mit einer Differenzengleichung

Für die Überführung der identifizierten DGL in eine DcGL ist ein Algorithmus in der Online-Selbst-Identifikation implementiert. Dieser verwendet ein neues Rechenzeit-effizientes Verfahren für die vollständig automatisierte Überführung einer DGL mit beliebiger Ordnung unter der Anwendung der Tustin-Methode. Nachfolgend ist das Ergebnis des Verfahrens für die Koeffizienten des Ausgangssignals der DcGL für eine DGL 3.Ordnung veranschaulicht.



Das Verfahren umfasst die allgemeine Formulierung von zwei  $(n+1) \times (n+1)$  Matrizen. Die elementweise Multiplikation der beiden Matrizen ergibt eine Matrix, in deren Zeilen die Terme der Polynome für die Berechnung der unbekannt Koeffizienten  $\bar{b}_j$  der Differenzgleichung enthalten sind. Dieses Verfahren wird auch für die Simulation des Ausgangssignals in Echtzeit als weitere Funktion des Identifikations-Algorithmus eingesetzt (siehe Abbildung 1).

### 6 Test des Algorithmus an einem autonom fahrenden Roboter

Nach der C-Code-Generierung des Simulink-Modells ist die Validierung des Algorithmus der Online-Selbst-Identifikation an verschiedenen realen Systemen, z.B. ein autonom fahrender Roboter, erfolgt. Während des Betriebs des Roboters fanden die Variation der Masse der Zuladung und damit eine Veränderung des Systemverhaltens statt. Der auf einem Mikrocontroller ausgeführte Identifikations-Algorithmus ermöglichte es, fortlaufend detaillierte Modelle des fahrenden Roboters für die variierenden Zuladungen zu generieren. In einer Identifikation wurde das folgende Modell  $G_{ident}(s)$  mit der Fit-Güte  $J_{fit}$  ermittelt:

$$G_{ident}(s) = \frac{0,0021*s^3 + 0,0010*s^2 + 0,1366*s + 0,1352}{0,0195*s^4 + 0,1280*s^3 + 1*s^2 + 2,7686*s + 2,8977} \quad \text{mit } J_{fit} = 98,62 \%$$

Benutzter Koeffizient für Division der DGL:  $b_2$       Ermittelte Totzeit:  $T_{tot} = 0,15 \text{ s}$

In den unteren Subplots sind die zeitlichen Verläufe für den gemessenen Systemeingang, für den gemessenen und simulierten Systemausgang und für den Schätzfehler im Identifikationszeitraum gezeigt.

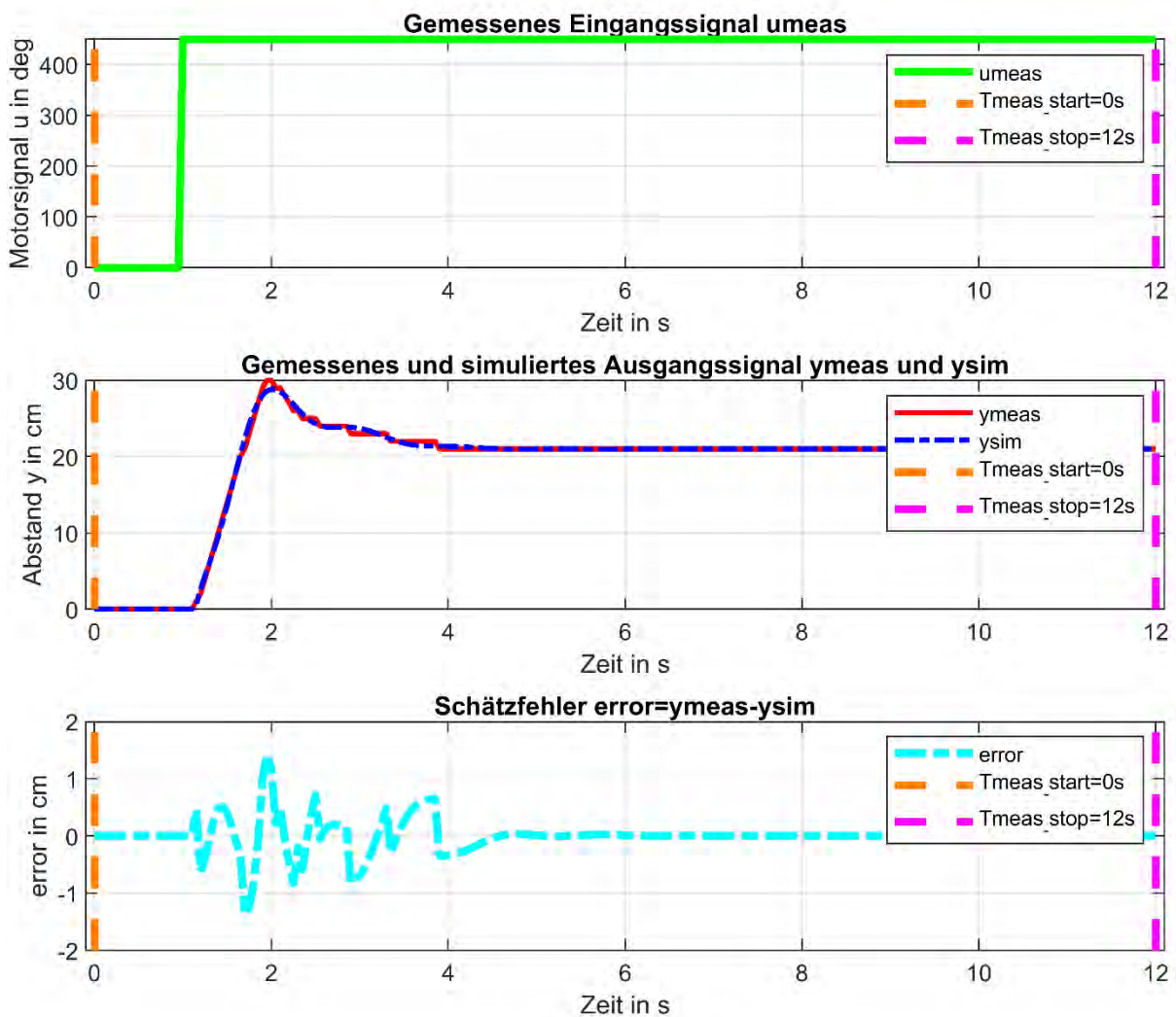


Abbildung 2: Zeitliche Verläufe des Systemein-, des Systemausgangs und des Schätzfehlers

## 7 Literatur

- [1] Bohn, C., & Unbehauen, H. (2016). Identifikation dynamischer Systeme, Clausthal-Zellerfeld, Bochum: Springer Vieweg-Verlag
- [2] Zentgraf, P. (2019). Ein neues Verfahren zur Modellierung linearer Systeme. *atp magazin 11/12 2019*, Rosenheim: Vulkan-Verlag GmbH
- [3] Ljung, L. (2019). System Identification Toolbox User's Guide, Natick: The MathWorks, Inc.