

# TS-Toolbox

Matlab-Toolbox zur nichtlinearen Systemidentifikation mittels lokal affiner Tagaki-Sugeno-Modelle

Version: 1.3 vom 14.9.2020

Prof. Dr.-Ing. Andreas Kroll, FG Mess- und Regelungstechnik, FB 15 Maschinenbau, Universität Kassel

URL: <http://www.uni-kassel.de/go/mrt>

Author: Axel Dürrbaum (<mailto:axel.duerrbaum@mrt.uni-kassel.de>)

## Contents

---

- [Aufgabe: Nichtlineare Systemidentifikation und Regression](#)
- [Modellansatz: lokal affine Tagaki-Sugeno-Modelle \(TS\)](#)
- [Funktionsprinzip](#)
- [Clusterung](#)
- [Verfügbare Zugehörigkeitsfunktionen](#)
- [Lokale TS-Modelle](#)
- [Modellgütemaße](#)
- [Visualisierung](#)
- [Dokumentation](#)
- [Verfügbare Objekte](#)
- [Verfügbare Funktionen](#)
- [Installation](#)
- [Musterprojekte](#)
- [Implementierung](#)
- [Benötigte Software](#)
- [Geplante Erweiterungen](#)

## Aufgabe: Nichtlineare Systemidentifikation und Regression

---

- für statische MISO-Modelle

$$y(t) = f(u_1(t), \dots, u_m(t))$$

- oder dynamische MISO-Modelle

$$y(t) = f(u_1(t), \dots, u_1(t - m_1), \dots, u_m(t - 1), \dots, u_m(t - m_m), \dots, y(t - 1), \dots, y(t - n))$$

## Modellansatz: lokal affine Tagaki-Sugeno-Modelle (TS)

---

Überlagerung der  $c$  lokal affinen Teilmodelle  $y_i(x)$  zu einem Gesamtmodell

$$\hat{y}(t) = \sum_{i=1}^c \mu_i(z) \cdot \hat{y}_i(x)$$

- mit den Eingangssignalen  $u(t)$  und dem Ausgangssignal  $y(t)$ ,
- der Scheduling-Variablen  $z(u, y)$ ,
- der Zugehörigkeitsfunktionen  $\mu_i(z)$ ,
- der Regressor-Variablen  $x(u, y)$
- und den lokalen TS-Modellen  $\hat{y}_i(x)$

## Funktionsprinzip

---

- Datensatz  $\{u(t), (y(t))\}$ , ggf. Normierung und Split in Identifikations- und Validierungsdaten
- Ggf. Anpassung der Standard-Einstellungen der Hyperparameter \* Clustering  $\nu, c, \epsilon_{FCM}$  \* Multistart \* NL-Optimierung
- Vorgabe der Anzahl der lokalen Modelle  $c$  und des Unschärfeparameters  $\nu$
- Clustering zur Ermittlung der Partitionierung bzw. Lage der Teilmodelle im Scheduling-Raum, Multistartstrategie mit Auswahl des besten Ergebnisses auf Basis des Modellfehlers auf Identifikationsdaten
- Initiale Schätzung der lokalen Modelle mittels Least-Squares-Verfahren (lokal oder global)
- Optionale Optimierung der Zugehörigkeitsfunktionen  $\mu_i$  und/oder der lokalen Teilmodelle  $\hat{y}_i$  mittels nichtlinearer Optimierung der Simulation (Matlab-Funktion `lsqnonlin`)
- Unterschiedliche Wahl der Scheduling- und Regressor-Variablen möglich
- Validierung auf neuen Daten

## Clustering

---

Eingangs- ( $u$ ) oder Produktraum ( $u|y$ )

Implementierte Algorithmen:

- Abstandsnormen: Euklid, Mahalanobis
- Fuzzy C-Means (FCM)
- Gustafson-Kessel (GK)

## Verfügbare Zugehörigkeitsfunktionen

---

- FCM-Type-Funktionen
- Gauss-Funktionen

## Lokale TS-Modelle

---

- Linear:  $y(t) = \sum_{i=0}^n a_i \cdot u_i(t) + a_0$
- ARX:  $y(t) = A \cdot y(t - dt) + B \cdot u(t) + C$
- OE:  $y(t) = A \cdot y(t - dt) + B \cdot u(t) + C + e(t)$

## Modellgütemaße

---

auf Identifikations- und Validierungsdaten

- Maximum Absolute Error (MAE)
- Sum of Squared Errors (SSE)
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- Normalized Mean Squared Error (NMSE)
- Best Fit Rate (BFR)
- ??? Akaike Information Criterion (AIC)
- ??? Bayesian Information Criterion (BIC)

## Visualisierung

---

- Clustering (2D, n-dimensional als mehrfache 2D)
- ??? Zugehörigkeitsmaße / Regelaktivierung
- Residuen
- ??? Residualhistogramm
- Simulation oder 1-Schritt-Prädiktion auf Identifikations- oder Validierungsdaten

## Dokumentation

---

### Verfügbare Objekte

---

- Daten Parameter, Methoden
- Modell Parameter, Methoden

### Verfügbare Funktionen

---

Funktionen, die nicht auf Objekten arbeiten

## Installation

---

ToDo

Verzeichnisse

- TS\_Toolbox
- TS\_Toolbox/Functions
- TS\_Toolbox/Examples

## Musterprojekte

---

im Verzeichnis `Examples`

- statisch: Akademisches Beispiel `Test_LS_Akad`
- statisch: Friedmann-Funktion 2D/3D `Test_LS_Friedman`
- statisch: Kompressor-Kennlinie 3D `Test_LS_Kompressor`
- dynamisch: Narendra (SISO) `Test_ARX_Narendra.m`

- dynamisch: Regelkappe (SISO) `Test_ARX_Throttle.m`
- dynamisch: Drosselkappe IAV (MISO) `Test_ARX_Ladedruck.m`

## Implementierung

Objektorientierte Realisierung:

- Objekt **Daten**:  $u(t), y(t)$
- Objekt **Modell**: Daten, Premisse, Konklusion
- Objekt **Premisse**: Scheduling / Zugehörigkeitsfunktion
- Objekt **Konklusion**: Regresser / lokale Modelle (ARX/OE)

## Benötigte Software

- Matlab R2019a oder höher (Windows/Linux/macOS)
- Matlab Fuzzy Toolbox (FCM)
- Matlab Optimization Toolbox (lsqnonlin)

## Geplante Erweiterungen

Aufgabe	Zeitraum	Prio	Status
Toolbox als OO-Klasse in Matlab Identifikation+Optimierung TS-Modelle statisch + ARX/OE (mg)	1	5/20	Erledigt 100%
ARX MISO-Modelle Optimierung MISO Tests Matlab-Optimierungsverfahren/-parameter	2	5/20 – 6/20	Erledigt: 100%
Subklassen für Modelle (LS/ARX/OE) und Datensätze		6/20	60%
Testsignalentwurf (mg)		6/20 - 7/20	0%
Regelung (as)		7/20	0%
Strukturselektion (mk)			10%
Maximum-Likelihood (jf)			0%
BETS (fw)			0%
Datascreening (da)			0%
Implementierung weitere Cluster-Verfahren			0%

\$Id: tsm\_Manual.m | Thu Dec 3 09:43:02 2020 +0100 | Axel Dürrbaum \$