

# Standardisierung dynamischer Modelle zur Analyse, Synthese, Simulation und Echtzeitrealisierung

Karlheinz Wolfmüller

Hochschule Heilbronn

*karlheinz.wolfmueller@hs-heilbronn.de*

**Abstract** - The nonlinear state space model with the potential- and flow variables  $\varepsilon(t)$  and  $\phi(t)$  as state space variables is introduced here as a model standard of a dynamic system with symbolic variables. In a general operating point a linearized matrix model is derived depending on the symbolic state space variables, the inputs and the system parameters. With this model all analysis-, synthesis-, simulation- and realtime realizing steps are implemented analytically with symbolic tools. The Software modules of the solution approaches are strictly realized depending on parameters. A standardized simulink model for simulation and for a realtime application is presented. The solution approaches are completely parameter adaptive. The methods presented here are verified for example by a nonlinear unstable process model and by some MIMO-models with nonlinear decoupling control methods.

## 1 Einleitung

Dynamische Modelle beschreiben das Zeitverhalten realer Vorgänge abstrahiert mit der Sprache der Mathematik und bilden die Basis sämtlicher Aktivitäten, die sich mit der Dynamik eines Systems befassen. In der realen Welt werden dabei immer Zustandsänderungen beschrieben, die durch Materie-, Energie- und Informations-, Transport-, Wandlungs- und Speichervorgängen verursacht werden. Bei der Modellbildung werden dabei verschiedene Formen von Differentialgleichungen mit unterschiedlichen Eigenschaften eingesetzt. Die physikalische Interpretierbarkeit der Modelle hat in der Praxis eine besondere Bedeutung. Durch die Einführung von Potential- und Flußgrößen lassen sich die Energiespeicher in einem System methodisch verallgemeinert darstellen. Mit entsprechenden Analogie Beziehungen läßt sich der Bezug in den Systemen der Elektrotechnik, Mechanik, Thermodynamik, Hydrodynamik und Gasdynamik herstellen. Die nichtlineare Zustandsraum-Modelldarstellung hat in diesem Zusammenhang eine herausragende Bedeutung. Leistungsfähige Modellbildungs-, Analyse-, Synthese-, Simulations- und Echtzeitprogrammierungs-Tools bieten mittlerweile die Voraussetzungen Forschungs-, Entwicklungs- und Entwurfsprojekte auf dem Gebiet der Systemdynamik weitgehend methodisch mit möglichst vielen analytischen und möglichst wenig numerischen Lösungsschritten durchzuführen. Wird diese Zielvorgabe konsequent eingehalten, darf man erwarten, dass am Ende immer eine ganze Problemklasse standardisiert gelöst wird. Die Softwaremodule der Problemlösung sollten sich allgemein ohne programmiertechnische Änderungen einsetzen

lassen. Spezielle problempezifische Lösungen sollten nur noch durch Strukturauswahl, Parametervariation und Anpassung erzeugt werden. Häufig wird bei einem RT-Projekt aus einer grundsätzlichen Aufgabe durch numerische Lösungsschritte ein spezielles Problem mit speziellen Lösungen gemacht. Diese Vorgehensweise ist ineffizient und schränkt die Chancen auf ein Optimum der Lösung stark ein. Es wird hier der Versuch unternommen auf analytischem Wege standardisierte Lösungen bei der Modellbildung, der Systemanalyse und bei der Synthese von Simulations-, Steuerungs-, Regelungs- und Signalaufbereitungs-Algorithmen mit Hilfe von symbolischen Werkzeugen aufzuzeigen. An einem exemplarischen Beispiel wird mit MATLAB/Simulink an einem instabilen nichtlinearen Streckenmodell z.B. ein Parameter adaptiver PI-Zustandsregler mit frei wählbarer Dynamik und einem nichtlinearen Beobachter mit variabler Schätzfehlerdynamik realisiert. Das Konzept kann als struktureller Lösungsstandard für viele Projekte verwendet werden. Alle physikalischen Parameter können im laufenden Simulations- und Echtzeitbetrieb verändert werden. Die Simulink-Lösungen der Simulations- und der Echtzeitprogramme enthalten nur Subsystem-Module, die konsequent auf analytischem Wege mit den Werkzeugen der symbolischen MATLAB-Function Library „RT2lib“ [4] und einem MATLAB Function-Code Generator erzeugt wurden. Diese Programmbibliothek benutzt die MATLAB Symbolic Toolbox und wurde im RT2-Labor der Hochschule Heilbronn entwickelt.

## 2 Dynamische Modelle

Die dynamische Modellbildung erfolgt in der Regel gemäß **Abbildung 1** in mehreren Schritten. Von zentraler Bedeutung dabei ist die Definition des Systems mit der Festlegung von eindeutigen Systemgrenzen. Damit liegt fest, wieviele innere Energiespeicher bzw. dynamische Freiheitsgrade  $n$  im System an den Zustandsänderungen beteiligt sind. Im Modell werden diese Energiespeicher durch die inneren Zustandsgrößen  $x_i$ ,  $i = 1, \dots, n$  abgebildet. Die kausalen Ursachen des Systems sind die bekannten  $p$  Eingangsgrößen  $u_1, \dots, u_p$  und die  $r$  unbekanntes Störgrößen  $z_1, \dots, z_r$  über die das System aus der Umgebung ausschließlich beeinflusst werden kann. Die Wirkungen sind im Inneren des Systems durch die Zustandsgrößen repräsentiert. In der Umgebung allerdings sind die Wirkungen nur an den  $q$  Ausgängen  $y_1, \dots, y_q$  sichtbar. Nach dieser Systemdefinition gibt es verschiedene Möglichkeiten die funktionellen Zusammenhänge zwischen Ursachen und Wirkungen darzustellen.

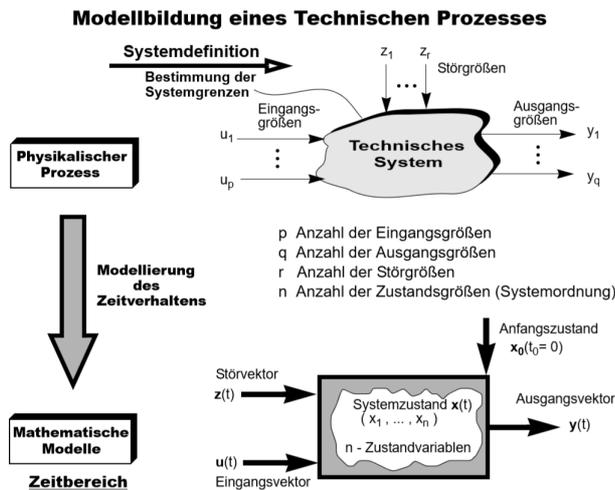


Abbildung 1. Modellbildung eines Prozesses [2],[3]

Mit der Aufstellung der Systemgleichungen lässt sich das Systemverhalten als

- Übertragungsmodell und/oder als
- Zustandsraum-Modell

abbilden. Jedes dieser Modelle verkörpert eine eigene Modellklasse mit unterschiedlichen Eigenschaften.

### 2.1 Übertragungsmodelle

Zur Darstellung eines MIMO-Systems  $n$ -ter Ordnung werden im allgemeinen  $q$  nichtlineare Übertragungsdifferentialgleichungen

$$t f_k(y_k^{(n)}, \dots, \dot{y}_k, y_k; \mathbf{u}^{(m)}, \dots, \dot{\mathbf{u}}, \mathbf{u}) = 0$$

$$m \leq n \quad k = 1, \dots, q \quad (1)$$

für jeden Ausgang benötigt, die das kausale Systemverhalten der Wirkungen (Ausgängen  $y$ ) in Abhängigkeit der Ursachen (Eingängen  $u$ ) und der Systemeigenschaften explizit mit den Größen aus der Umgebung in einem absoluten Koordinatensystem beschreiben. Existieren nur lineare Zusammenhänge bzw. linearisiert man das System Gl. (1) um einen Betriebspunkt  $(y_B, u_B)$  und entsprechende Ableitungen davon, ergeben sich  $(q \times p)$  lineare Übertragungsdifferentialgleichungen

$$y_k^{(n)} + a_{n-1,k,j} y_k^{(n-1)} + \dots + a_{1,k,j} \dot{y}_k + a_{0,k,j} y_k =$$

$$b_{m,k,j} u_j^{(m)} + b_{m-1,k,j} u_j^{(m-1)} + \dots + b_{1,k,j} \dot{u}_j + b_{0,k,j} u_j$$

$$m \leq n, \quad k = 1, \dots, q; \quad j = 1, \dots, p \quad (2)$$

jeweils des  $j$ -ten Eingangs zum  $k$ -ten Ausgang in einem absoluten oder relativen Koordinatensystem. Im Laplace-Frequenzbereich kann das Modell Gl. (2) auch mit Hilfe einer komplexen  $(q \times p)$  Übertragungsmatrix dargestellt werden [3].

### 2.2 Zustandsraum-Modelle

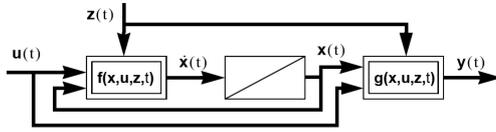
Zustandsraum-Modelle bilden das Systemverhalten in zwei Schritten ab. Im 1.Schritt werden die inneren dynamischen Freiheitsgrade bzw. die inneren Energiezustände in ihren zeitliche Veränderungen jeweils mit einer Differentialgleichung 1.Ordnung in Abhängigkeit vom inneren Gesamtzustand  $x(t)$  und den Eingängen  $u(t)$  zu einem bestimmten Zeitpunkt  $t$  abgebildet. In diesem Modellierungsschritt ergeben sich  $n$  Differentialgleichungen 1.Ordnung die zu einem Funktionsvektor Gl. (3)

$$\dot{x} = f(x, u, t) \quad (3)$$

$$y = g(x, u, t) \quad (4)$$

zusammengefaßt werden können. Dieser Teil des Modells wird als nichtlineare Zustandsdifferentialgleichung bezeichnet. Die Wirkungen in der äußeren Umgebung werden im 2.Schritt durch einen nichtlinearen, algebraischen Funktionsvektor des Ausgangs  $y(t)$  der Länge  $q$  in Abhängigkeit von den inneren Zuständen  $x(t)$  und den Eingängen  $u(t)$  zu einem Zeitpunkt  $t$  dargestellt. Gl.(4) wird als nichtlineare Ausgangsgleichung bezeichnet. Die Zustandsdifferentialgleichung Gl.(3) und die Ausgangsgleichung Gl.(4) sind die Zustandsgleichungen des nichtlinearen (NL-)Zustandsraum-Modells.

Die **Abbildung 2** zeigt das Strukturbild des NL-Zustandsraum-Modells.



**Abbildung 2.** Strukturbild NL-Zustandsraum-Modell

Linearisiert man das NL-Modell Gl. (3), Gl. (4) um einen Betriebspunkt  $(\mathbf{x}_B, \mathbf{y}_B)$  z.B. der Ruhelage mit

$$0 = \mathbf{f}(\mathbf{x}_B, \mathbf{u}_B), \quad \mathbf{y}_B = \mathbf{g}(\mathbf{x}_B, \mathbf{u}_B) \quad (5)$$

und der Lösung  $[\mathbf{x}_B(\mathbf{y}_B); \mathbf{u}_B(\mathbf{y}_B)]$

ergibt sich für die Abweichungen

$$\Delta \dot{\mathbf{x}} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}_B, \mathbf{u}_B} \cdot \Delta \mathbf{x} + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{\mathbf{x}_B, \mathbf{u}_B} \cdot \Delta \mathbf{u} \quad (6)$$

$$\Delta \mathbf{y} = \left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right|_{\mathbf{x}_B, \mathbf{u}_B} \cdot \Delta \mathbf{x} + \left. \frac{\partial \mathbf{g}}{\partial \mathbf{u}} \right|_{\mathbf{x}_B, \mathbf{u}_B} \cdot \Delta \mathbf{u} \quad (7)$$

ein Modell in einem relativen Koordinatensystem um den Betriebspunkt. Verändert man nun den Betriebspunkt im physikalischen Wertebereich einer Ausgangstrajektorie  $\mathbf{y}(t)$  so entsteht ein Modell für die Abweichungen um einen zeitveränderlichen Betriebspunkt

$$\Delta \dot{\mathbf{x}} = \mathbf{A}(\mathbf{x}(t)) \cdot \Delta \mathbf{x} + \mathbf{B}(\mathbf{u}(t)) \cdot \Delta \mathbf{u} \quad (8)$$

$$\Delta \mathbf{y} = \mathbf{C}(\mathbf{x}(t)) \cdot \Delta \mathbf{x} + \mathbf{D}(\mathbf{u}(t)) \cdot \Delta \mathbf{u} \quad (9)$$

in einer formalen Vektor- und Matrizenschreibweise. Die Modellmatrizen  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ ,  $\mathbf{D}$  sind zeitabhängig von  $\mathbf{x}(t)$  und  $\mathbf{u}(t)$ . Zu erwähnen ist, dass am Betriebspunkt das Änderungsverhalten dieses Modells exakt dem Verhalten des nichtlinearen Originalmodells am Betriebspunkt entspricht. Damit wird das Systemverhalten in einem absoluten bzw. relativen Koordinatensystem im  $n$ -dimensionalen Zustandsraum  $X^n$  beschrieben.

### 2.2.1 Physikalische Definition der Zustandsgrößen

In den technischen Disziplinen Elektrotechnik, Mechanik, Thermodynamik, Hydrodynamik und Gasdynamik lassen sich die inneren Zustandsgrößen als abstrakte Potentialgrößen  $\varepsilon(t)$  und Flussgrößen  $\phi(t)$  von Speicherbauteilen interpretieren [3], wie sie z.B. in Netzwerken benutzt werden. Die Potentialgröße repräsentiert direkt den Energiezustand eines

kapazitiven Speichers. Die Zustandsdifferentialgleichung

$$\dot{\varepsilon}(t) = \frac{1}{C(\varepsilon)} \cdot \phi(t) \quad (10)$$

beschreibt mit der Bauteilkapazität  $C$  den Zusammenhang mit der Flussgröße. Die Flussgröße repräsentiert direkt den Energiezustand eines induktiven Speichers. Die Zustandsdifferentialgleichung

$$\dot{\phi}(t) = \frac{1}{L(\phi)} \cdot \varepsilon(t) \quad (11)$$

beschreibt mit der Bauteilinduktivität  $L$  den Zusammenhang mit der Potentialgröße.

Es wird nun noch ein Energiewandler als Verbraucherbauteil mit der reellen Bauteilgröße

$$R = f(\varepsilon, \phi) = \varepsilon / \phi \quad (12)$$

eingeführt, in dem mit der Wirkleistung

$$p = \varepsilon \cdot \phi = R \cdot \phi^2 \quad (13)$$

eine gesteuerte thermische Flussquelle thermische Energie erzeugt, die die Temperatur  $T(t)$  des Systems als thermische Potentialgröße beeinflusst in Wechselwirkung mit der Temperatur  $T_U(t)$  der Umgebung.

Physikal. System	Potentialgröße $\varepsilon(t)$	Flussgröße $\phi(t)$
Elektrotechnik	<u>Spannung</u> $u$ /[V]	<u>Strom</u> $i$ /[A]
Mechanik - Translation	<u>Geschwindigkeit</u> $v$ /[m/s]	<u>Kraft</u> $F$ /[N]
- Rotation	<u>Kreisfrequenz</u> $\omega$ /[rad/s]	<u>Moment</u> $M$ /[N/m]
Thermodynamik	<u>Temperatur</u> $T$ /[°K]	<u>Wärmestrom</u> $\dot{Q}$ /[W]
Hydrodynamik	<u>Druck</u> $p$ /[Pa]	<u>Volumenstrom</u> $\dot{V}$ /[m³/s]
Gasdynamik	<u>Druck</u> $p$ /[Pa]	<u>Volumenstrom</u> $\dot{V} = \frac{\dot{m}}{\rho}$ /[m³/s]

**Tabelle 1.** Analogien der Potential- und Flussgrößen

Da grundsätzlich derartige Bauteile als thermische Energiewandler in jedem System vorhanden sind, werden zur Beschreibung der Temperatur der Bauteile im System immer dynamische thermische Modelle

von thermischen Massen benötigt. Thermische Energiespeicher sind grundsätzlich kapazitive Speicher. In der **Tabelle 1** sind der Potential- und der Flussgröße die Analogiegrößen in den einzelnen technischen Disziplinen zugeordnet.

Zur Ansteuerung der Eingänge  $\mathbf{u}$  und zur Realisierung der Anfangszustände  $\mathbf{x}_0(t_0)$  der Energiespeicher sind noch eine ideale Potentialquelle und eine ideale Flussquelle mit den entsprechenden Impedanzbedingungen notwendig. In der **Abbildung 3** sind die 5 Netzwerkbauerelemente zusammengefasst, die in Parallel- und/oder Reihenschaltung zu komplexen dynamischen Netzwerk-Modellen [3] zusammen geschaltet werden können. Durch Anwendung der Knoten- und Maschensätze bieten diese Modelle die

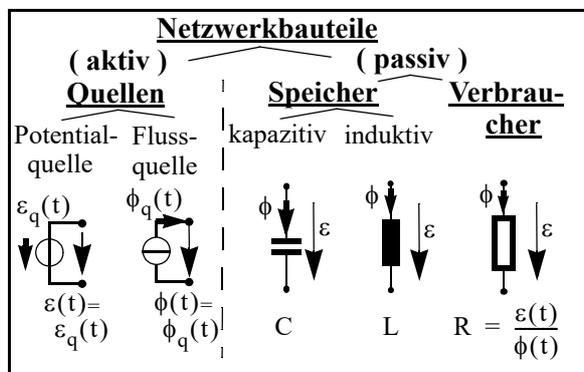


Abbildung 3. Übersicht der Netzwerkbauerelemente für die dynamische Modellbildung

idealen Voraussetzungen zur systematischen Aufstellung der Modellgleichungen [3].

### 3 Analyse

Lineare Übertragungsmodelle können direkt in die speziellen Zustandsräume der Regelungsnormform (RNF) und der Beobachtungsnormform (BNF) überführt werden [1]. Dabei sind die Zustandsvariablen direkt mit den Ausgängen und deren Ableitungen verknüpft. Sie sind also nur noch bedingt oder nicht mehr physikalisch interpretierbar.

#### 3.1 Eigenschaften von Übertragungsmodellen

Lineare Übertragungsmodelle werden vornehmlich in der klassischen Regelungstechnik im Laplace-Bildbereich zur Frequenzanalyse und dem Entwurf klassischer PID-Standardregelkreise insbesondere bei SISO-Systemen eingesetzt. Komplexe nichtlineare Mehrgrößensysteme, deren innere physikalische Eigenschaften interessieren, lassen sich mit Übertragungsmodellen nur schwer oder überhaupt nicht physikalisch analysieren.

#### 3.2 Eigenschaften von Zustandsraum-Modellen

Interessiert man sich für die physikalischen inneren Zusammenhänge und Größen eines Systems bieten die nichtlinearen Zustandsraum-Modelle Gl. (3) und Gl. (4) die besten Voraussetzungen. Wählt man zusätzlich die Zustandsvariablen wie es in Kap. 2.2.1 beschrieben wird, ist die bestmögliche physikalische Referenz gewährleistet. Diese Modellform kann als physikalischer Modellstandard zur Beschreibung des Zeitverhaltens eines Systems angesehen werden. Zusätzlich ist die mathematische Form und Struktur **Abbildung 2** der Modellgleichungen relativ einfach und prädestiniert zur Verarbeitung mit einem Digitalrechner. Als exemplarisches Beispiel soll hier das allgemein bekannte akademische Lehrmodell „Magnetfeld Schwebekörper“ dienen. In dem MATLAB-Programm **Code 1** wird konsequent mit symbolischen Variablen das NL-Zustandsraummodell gemäß Gl. (3), Gl. (4) im MATLAB-Workspace gebildet.

```

%%
% Parameter Magnetfeldregelung
%-----
syms k kd g m R L Fm f_NL g_NL x u y w
%%
% Nichtlineares Zustandsraum-Modell
%=====
% NL-Zustandsdgl      x'=f_NL(x,u,t)
%-----
f_NL(1,1) = x(2);
f_NL(2,1) = -kd/m*x(2) - k/m*x(3) ...
             ^2/x(1)^2 + g;
f_NL(3,1) = -(R/L)*x(3) + 1/L*u

% NL-Ausgangsglg     y=g_NL(x,u,t)
%-----
g_NL = x(1)

% Matlab-function Generator
create_matfkt('NL_zgl_matfkt', f_NL,
g_NL)
    
```

Code 1. Symbolische NL-Zustandsgleichungen

Die verwendeten MATLAB-functions sind Bestandteile der im RT2-Labor der Hochschule Heilbronn entwickelten MATLAB-function Library „RT2lib“ [4]. Der MATLAB-function Code Generator „create\_matfkt(...)“ in der letzten Zeile generiert automatisch den Programm-Code der NL-Zustandsgleichungen  $\mathbf{f}(\mathbf{x}, \mathbf{u})$ ,  $\mathbf{g}(\mathbf{x}, \mathbf{u})$  in einem MATLAB-function m-File mit dem Namen 'NL\_zgl\_matfkt'. Dieser Code Generator bildet aus einer symbolischen Ausdruck-Liste den MATLAB-Code einer MATLAB-

function in einem m-File mit dem Namen des 1.Übergabe-Strings.

Benutzt man darüber hinaus noch die Betriebspunkt-darstellung Gl. (6), Gl. (7), Gl. (8) und Gl. (9) so können formal die Analysewerkzeuge der linearen Systemtheorie [1] angewendet werden. Es besteht damit die Möglichkeit ein System entlang bestimmter Betriebstrajektorien bezüglich seiner Eigenschaften zu betrachten. Zum Beispiel können Steuerbarkeit, Beobachtbarkeit und Stabilität mit der Eigenwertanalyse mit analytischen Werkzeugen an veränderlichen Betriebspunkten untersucht werden. Der MATLAB-Programm **Code 2** führt diesen Schritt durch.

```
%%
% Stationärer Betriebspunkt
%-----
[xB, uB]=nl_statb(f_NL,g_NL,x,u,y)
%%
% Linearisiertes Modell
%-----
[Asym, Bsym, Csym, Dsym] = ...
    nl2lin(f_NL, g_NL, x, u, y)
%%
% Analyse am Betriebspunkt
%-----
% Symbolische Eigenwerte
Eigsym=eig(Asym)
```

**Code 2.** Stationärer Betriebspunkt mit linearisiertem Modell und Eigenwertanalyse

Führt man die Analyseschritte konsequent mit symbolischen Mathematik-Tools durch, sind die analytischen Ergebnisse abhängig von den veränderlichen Systemparametern und sind so allgemeingültig interpretierbar. Dies kann als erheblicher Fortschritt gegenüber rein numerischen Lösungen angesehen werden.

## 4 Synthese

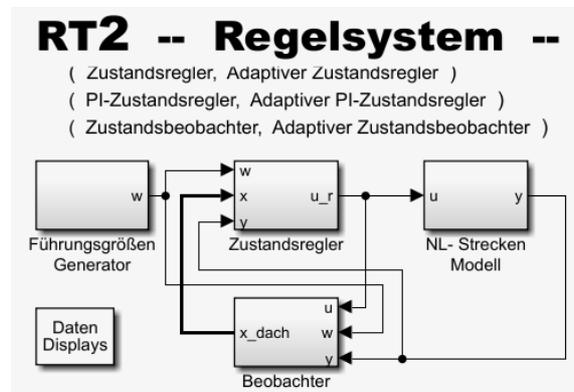
Es gibt kaum allgemein anwendbare, mathematische Werkzeuge zur Synthese von Steuerungs-, Regelungs- und Beobachtungs-Algorithmen an NL-Zustandsraum-Modellen. Alle bekannten linearen Synthesewerkzeuge [1] können natürlich auf die linearisierten Betriebspunkt-Modelle Gl. (8), Gl. (9) angewendet werden. In der „RT2lib“ werden MATLAB-functions bereitgestellt, die die gängigen Syntheseschritte konsequent analytisch an den linearisierten Betriebspunkt-Modellen umsetzen. Dabei wird die MATLAB Symbolic-Toolbox eingesetzt. Nach aktuellem Stand

werden in der „RT2lib“ Werkzeuge für die folgenden analytischen Syntheseschritte zur Verfügung gestellt:

- Zustandsregler (variable Pole und Parameter),
- PI-Zustandsregler (variable Pole und Parameter),
- Adaptiver Zustandsregler (variable Pole und Parameter),
- Adaptiver PI-Zustandsregler (variable Pole und Parameter),
- Linearer Mehrgrößen-Entkopplungsregler (variable Pole und Parameter),
- Nichtlinearer Mehrgrößen-Entkopplungsregler (variable Pole und Parameter),
- Beobachter (variable Dynamik und Parameter),
- NL-Beobachter (variable Dynamik und Parameter).

## 5 Simulation

In **Abbildung 4** ist ein mit Simulink realisiertes Simulationsmodell dargestellt, das als Regelsystem-Standard betrachtet werden kann.



**Abbildung 4.** Standardisiertes Simulink Regelsystem

**Abbildung 5** zeigt die Simulink-Lösung des Subsystems des standardisierten NL-Streckenmodells. Es entspricht strukturell exakt dem mathematischen Modell von **Abbildung 2**. Es können hier z.B. auch die Systemparameter dynamisch während der Simulation bzw. im laufenden Echtzeitbetrieb verändert werden. Sämtliche im Regelsystem beteiligten Komponenten passen sich dynamisch an. Strukturell identisch ist die Simulink Lösung des NL-Beobachters, der als adaptiver Beobachter mit einem von der Messgröße dynamisch gesteuerten Betriebspunkt ent-

worfen wurde. Für die Realisierung der Regelung können sämtliche aufgeführten Zustandsregler-Typen

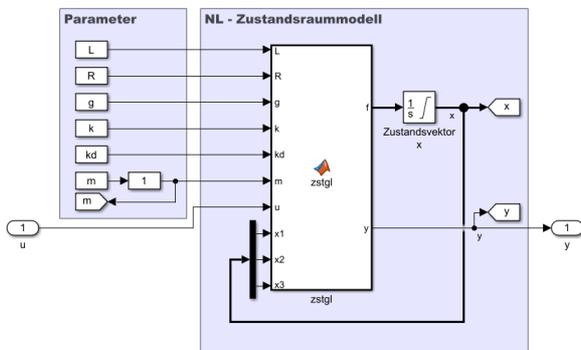


Abbildung 5. NL-Streckenmodell, Simulink-Standard

ausgewählt werden. Auch hier ist die interne Struktur gleich. Beobachter und Regler sind vollständig Systemparameter abhängig und variabel in der Dynamik durch veränderliche Pole.

## 6 Echtzeitrealisierung

Mit den aktuell verfügbaren Prozessorleistungen ist die Realisierung der Parameter adaptiven Regler- und Beobachter-Algorithmen unter harten Echtzeitbedingungen in der Regel kein grundsätzliches Problem mehr. Der maximale algorithmische Aufwand existiert in der Praxis natürlich nur in der Entwicklungsphase. Nach der Systemoptimierung bzgl. der Zielkriterien der konkreten Anwendung sind viele Parameter im praktischen Betrieb als konstant anzusehen. Der Rechenaufwand kann sich dann stark reduzieren. Das Regelsystem gemäß **Abbildung 4** läßt sich direkt ohne Änderung der Subsysteme mit Real-Time Simulink oder anderen Hardware Support - Packages mit entsprechendem C-Code Generator als Echtzeitprozess bei diskreter Abtastzeit implementieren. Es ist nur die Strecke durch die Sensor/Aktor-Interfaces der Realtime-Hardware an der Stellgröße  $u$  und der Messgröße  $y$  zu ersetzen. Die Algorithmen der Echtzeit- und der Simulationslösung sind nach der Diskretisierung identisch. Das hier vorgestellte Beispiel wurde mit der Simulink Real-Time Toolbox und PC-Hardware mit PCI-NI6229 I/O-Interfaces unter harten Echtzeitbedingungen (Rt-ontime Kernel) realisiert. Selbst mit einem Debian LINUX-Kernel auf einem Raspberry PI-3 konnte eine Abtastzeit von  $T_A = 1$  msec für den Regelungsprozess im Rt-Status eingehalten werden.

## 7 Anwendungsbeispiele

Die hier vorgestellte Vorgehensweise der analytischen Modellierung, Analyse, Synthese, Simulation und Echtzeitrealisierung wurde in mehreren Projekten schon angewendet und verifiziert. Exemplarisch werden hier z.B. genannt:

1. NL-3Tank MIMO-Modell  
(Entkopplungsregelung mit 2 Eingängen, 2 Ausgängen, 3 Zuständen),
2. NL-gasdynamisches MIMO-Modell  
(NL-Entkopplungsregelung mit 3 Eingängen, 3 Ausgängen, 4 Zuständen).

In allen Projekten sind die Lösungen der einzelnen Schritten mit den Programm-Modulen strukturell identisch, obwohl völlig unterschiedliche physikalische Problemstellungen behandelt werden.

## 8 Zusammenfassung

Das NL-Zustandsraum-Modell mit den physikalisch definierten Zustandsgrößen wird hier als Standardmodell eingeführt. Sämtliche Modellierungs-, Analyse-Synthese-, Simulations- und Echtzeitrealisierungsschritte werden als analytische Lösungen mit symbolischen Ausdrücken umgesetzt. Damit wird eine dynamische Parametrierung der Lösungen in allen Modulen möglich. In den Entwicklungsphasen eines Projekts ist damit die größte Flexibilität mit bestmöglichem Optimum der Lösung gegeben. Es liegt immer eine Lösung der physikalischen Problemklasse vor. Numerische Lösungsschritte, die die Allgemeingültigkeit einschränken, werden konsequent vermieden.

## 9 Referenzen

- [1] Otto, Föllinger. Regelungstechnik. Heidelberg: Hüthig Verlag 1992, 7. Auflage
- [2] Karlheinz, Wolfmüller. Systemdynamik und Simulation 2. Hochschule Heilbronn, WS2017/18
- [3] Karlheinz Wolfmüller. Modellbildung und Simulation. Hochschule Heilbronn, WS2013/14
- [4] Karlheinz, Wolfmüller. RT2lib, MATLAB-function Library. RT2-Labor, Hochschule Heilbronn, 2018