

# Elektronikdesign für Mechatroniksysteme

## Einheitliche Modellierungs- und Simulationsumgebung zur Systemanalyse

**Besonders in der Automobil-Industrie stehen interdisziplinäre Entwicklungsteams vor großen Herausforderungen bei der Integration und vollständigen Verifikation zukünftiger hybrider Architekturen. Das Zusammenwirken von einzelnen mechatronischen Komponenten stellt einen wesentlichen Aspekt der Verifikation des Zielsystems dar. Im Gegensatz zu der heute häufig angewandten Methode des so genannten ‚Cut and Try‘ mit realer Hardware ermöglicht die Verwendung von Hochsprachen eine frühzeitige Verifikation mit Hilfe von virtuellen Prototypen im Sinne eines computergestützten Prototypings auf Basis von mathematischen Beschreibungen.**

INGO NICKELEIT

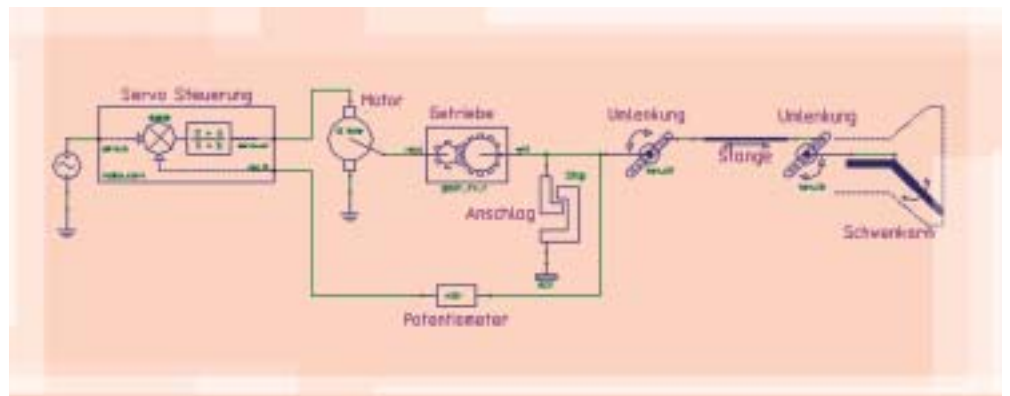


Abb. 1: Mechatronisches System mit elektromechanischen Komponenten zur Dimensionierung der elektronischen Steuerung mit ‚SystemVision‘ von Mentor



Ingo Nickleit verfügt über langjährige Erfahrungen als Schaltungsingenieur für Telekommunikationssysteme und ist seit 5 Jahren als Applications Ingenieur bei Mentor Graphics im Bereich ‚Analog Mixed Signal Systeme‘ tätig.

Zu den Industriebereichen mit stark steigendem mechatronischem Anteil am Gesamtsystem gehören Luftfahrt, Medizintechnik, der große Bereich der Industrieanlagensteuerungen und in besonderem Maße die Automobilindustrie. Schon aktuelle Mittelklassewagenmodelle sind mit einer Vielzahl von Mechatronikkomponenten bestückt. Diese sind Bestandteil der Systeme zur Erhöhung der Fahrsicherheit und des Komforts, wie zum Beispiel Airbag und elektronische Sitzeinstellung oder Fahrerinformation- (GPS) und Entertainment-Einrichtungen. Die Anwendung in der Automobilindustrie setzt darüber hinaus voraus, dass große Stückzahlen mit hoher Zuverlässigkeit gefertigt werden können. Der zu erwartende Einstieg der Automobilhersteller in 42-Volt-Systeme, Multiplex-Datenbus-Architekturen und alternative Antriebskonzepte erfordert neue Strategien zur Systemmodellierung und -verifikation. Mit der

Verfügbarkeit von einheitlichen Modellierungsstandards und Werkzeugen zur Analyse interdisziplinärer Systeme bietet sich eine hervorragende Chance für einen reibungslosen Ein- und Umstieg in künftige Technologien.

### Design- und Verifikation mechatronischer Systeme

Mechatronische Systeme sind gekennzeichnet durch eine starke Wechselwirkung zwischen nicht-elektrischen und elektrischen Komponenten. Dabei sind die nicht-elektrischen physikalischen Effekte keineswegs nur auf die Mechanik beschränkt, sondern kommen speziell im Automobil auch aus den Bereichen Hydraulik, Thermodynamik usw. Doch auch in der elektronischen Welt gibt es mehrere, meist unabhängige Disziplinen. Auf Systemebene müssen Algorithmen, Schnittstellen und Kommunikationsprotokolle festgelegt werden. ▶

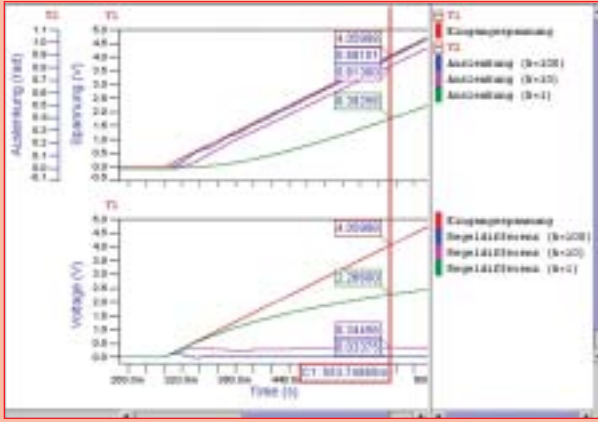


Abb. 2: Dynamik der Servosteuerung für eine Verstärkung von 1, 10 und 100

Mit Hilfe von entsprechenden Pflichtenheften entwickeln dann Schaltungsdesigner auf Komponentenebene analoge (Leistungs-) Treiber und digitale Steuerungen. Darüber hinaus besteht häufig eine enge Verbindung zur Software, die sich mit zukünftigen konfigurierbaren Systemen noch verstärken wird. Für eine effiziente Analyse aller Interaktionen ist eine einheitliche Modellierungs- und Simulationsumgebung unverzichtbar.

In der Systemmodellierung favorisieren branchenweit immer mehr Unternehmen den IEEE Standard 1076.1 [1], meist VHDL-AMS genannt. Dieser ist eine seit 1999 existierende Erweiterung des bestehenden VHDL-Standards und ergänzt die digitalen Eigenschaften um analoge Verhaltensbeschreibungen in Form von (Differential-) Gleichungen. Im Gegensatz zu früheren analogen Beschreibungssprachen kann mit dieser Kombination echtes Mixed-Signal-Verhalten in einem geschlossenen Modell vollständig charakterisiert werden. Der Standard enthält darüber hinaus auch Anweisungen, wie diskontinuierliche Interaktionen während der Simulation zu synchronisieren sind. Aus Sicht des Entwicklers führt die kompatible und konsistente Verarbeitung seiner Modelle durch verschiedene Simulatoren zu einer erhöhten Sicherheit. Es gibt bereits mehrere Simulatoren mit VHDL-AMS-Unterstützung. Studien des ‚Manufacturer Supplier Relationship‘-Konsortiums (MSR) belegen, dass ein Modellaustausch weitestgehend ohne Modifikationen möglich ist [2]. Der offene Standard VHDL-AMS bietet daher mit seinen analogen Erweiterungen zur Beschreibung von Effekten aus verschiedenen physikalischen Disziplinen und der Unterstützung durch führende EDA-Werkzeuganbieter eine geeignete Grundlage zur Vereinheitlichung der Systemmodellierung.

Im Bereich der Systemzulieferer und Integratoren ergeben sich für den gesamten Entwicklungsprozess wesentliche Vorteile und Möglichkeiten zur Kosteneinsparung. Im Gegensatz zu der heute häufig angewandten Methode des sogenannten ‚Cut and Try‘ mit realer Hardware

werden, um kritische Abhängigkeiten zu identifizieren, wobei der Modellierungsaufwand auf wenige Standards beschränkt bleibt. Jeder Entwickler gewinnt somit rasch Einblick in das Zusammenwirken der Komponenten im System. Die Optimierung wird direkt im Hinblick auf das Verhalten des Gesamtsystems durchgeführt – und nicht mehr ausschließlich für einzelne Komponenten. In jedem Entwicklungsstadium kann die Zuverlässigkeit an beliebigen Spezifikationspunkten untersucht werden. Während des gesamten Entwicklungsprozesses wird die Kommunikation in der Zulieferkette und zwischen den Entwicklern unterschiedlicher physikalischer Disziplinen deutlich verbessert.

### Mechatronik-Praxisbeispiel: Schwenkarmsteuerung

Ein typisches Beispiel für die Entwicklung eines mechatronischen Systems ist die Servoansteuerung eines Schwenkarms, welches im Automobil auf eine Lichtweitenregelung angewandt werden könnte. Abbildung 1 zeigt ein Blockschaltbild des elektromechanischen Systems.

Die Dimensionierung erfolgt zunächst sehr abstrakt in der sogenannten Laplace oder S-Domäne. D.h. alle Komponenten werden über ihre allgemeine mathematische Übertragungsfunktion in VHDL-AMS beschrieben. Somit lassen sich Topologie und dynamisches Regelverhalten eindeutig festlegen, ohne spezielle Überlegungen wie z.B. die Verbindung verschiedener Domänen zu berücksichtigen. Im VHDL-AMS-Modell sind in diesem Fall die externen Anschlüsse sogenannte Quantity Ports, die eben nicht der Energieerhaltung unterliegen und deshalb in der Regel zur deutlichen Beschleunigung der Simulation führen. Auch das physikalische Verhalten der Komponenten lässt sich mit VHDL-AMS beliebig abstrahieren. Ausgehend von den Kenndaten des Motorherstellers (Tab. 1) können direkt die Parameter (Pole p; Grund-

ermöglicht die Verwendung von Modellierungshochsprachen eine frühzeitige Verifikation mit Hilfe von virtuellen Prototypen im Sinne eines ‚Math-based Computer Aided Prototyping‘. Durch eine Gesamtsystembetrachtung in einer einheitlichen Umgebung können alle relevanten physikalischen Bereiche miteinander verknüpft

verstärkung G) des Laplace-Modells aus Verstärkung und Zeitkonstanten mit folgenden Formeln bestimmt werden.

$$p_e = -\frac{1}{\tau_e} = -\frac{L}{R} \tag{1}$$

$$p_m = -\frac{1}{\tau_m} = -\frac{J}{B} \tag{2}$$

$$G(s)_e = \frac{1}{(sL + R)} \tag{3}$$

$$G(s)_m = \frac{1}{(sJ + B)} \tag{4}$$

Unter Verwendung der Werte aus Tabelle 1 ergeben sich für  $p_e = 1,08 \text{ krad/s}$  (172,5 Hz),  $p_m 33,5 \text{ rad/s}$  (5,3 Hz) und unter der Annahme dass s gegen Null geht für  $G(0)_e = 0,4545$  bzw.  $G(0)_m = 177,6 \cdot 10^3$ .

Die Beschreibung des Gleichstrommotors erhält damit die im Kasten gezeigte Struktur:

```
-- DC Motor Laplace Modell
library IEEE;
use ieee.math_real.all;
entity DC_MotorS is
    generic (
        ge: real := 0.4545;
        gm: real := 177.6e3;
        pe: real := 172.5;
        pm: real := 5.3;
        kT: real := 3.43;
        kE: real := 3.43);
    port (
        quantity motor_cmd, load_t: in real;
        quantity motor_out : out real);
end entity DC_MotorS;

architecture laplace of DC_MotorS is
    constant we : real := pe*math_2_pi;
    constant wm : real := pm*math_2_pi;
    constant nume : real_vector := (0 => we * ge);
    constant dene : real_vector := (we, 1.0);
    constant numm : real_vector := (0 => wm * gm);
    constant denm : real_vector := (wm, 1.0);

    quantity motor_diff: real := 0.0;
    quantity motor_e : real := 0.0;
    quantity motor_m : real := 0.0;

begin
    motor_diff == motor_cmd - kE * motor_out;
    motor_e == kT * motor_diff'ltf(nume, dene);
    motor_m == motor_e - load_t;
    motor_out == motor_m'ltf(numm, denm);
end architecture laplace;
```

Das Attribut 'ltf (Laplace-Transfer-Funktion) ist eine mächtige AMS-Erweiterung von VHDL. Sie ermöglicht kompakten Code, exakte Lösungen bei kürzeren Simulationszeiten und ist deshalb eine sehr effiziente Methode zur Systemverifikation. Im Sinne einer Top-Down-Entwurfsmethodik können nun anhand der Laplace-Darstellung die elektroni-

sche Systemspezifikation bereits in einer konzeptionellen Entwicklungsphase überprüft werden:

- ▶ Schwenkdynamik ( $0^\circ$  bis  $60^\circ$ )  $\leq 300$  ms (mit 0,2 Nm Belastung bei maximalem Ausschlag)
- ▶ Servoregelkreisfehler (im eingeschwungenen Zustand)  $\leq 1$  %
- ▶ Servophasenreserve  $\geq 35^\circ$

Abbildung 2 zeigt den Verlauf der ersten beiden Kriterien abhängig von der Verstärkung der Servosteuerung. Für einen Gewinn von

**Tabelle 1: Kenndaten des Motors aus dem Beispiel Schwenkarmsteuerung**

Kennwert	Formelzeichen	Wert
Eingangswiderstand	$R$	2,2 Ohm
Eingangsimpedanz	$L$	2,03 mH
Achsenträgheitsmoment	$J$	168 kg/m <sup>2</sup>
Drehmomentkonstante	$k_T$	3,43 mNm/A
EMK-Konstante	$k_E$	3,43 mV/rad/s
Dämpfungsfaktor	$B$	5,63 $\mu$ Nm/rad/s

ca. 100 bewegt sich der Schwenkarm innerhalb von 300 ms in die Endlage und der verbleibende Fehler ist kleiner 1 % ( $0,034/4,06 = 0,83$  %). Allerdings beträgt die Phasenreserve nur  $10^\circ$ .

Die Bestimmung der Kompensation erfolgt mit dem Wurzelortskurvenverfahren. Von besonderem Interesse ist das Zusammenwirken der Motorpolstelle bei 5,3 Hz mit dem inhä-

## Anzeige

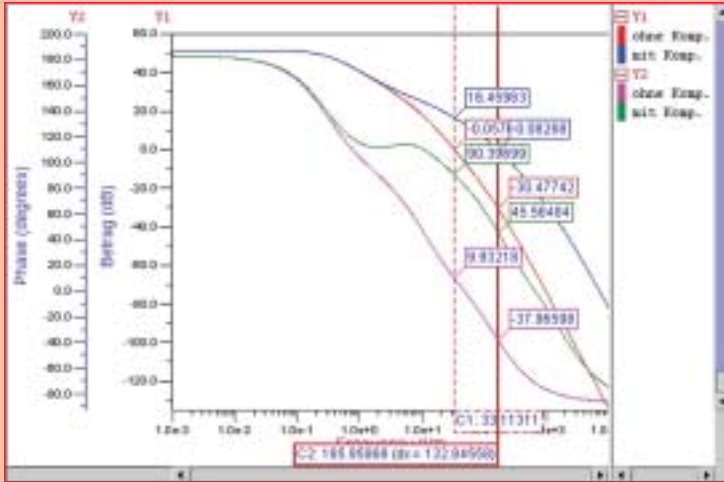


Abb. 3: Bode-Diagramm für eine Verstärkung von 100 mit und ohne Kompensation

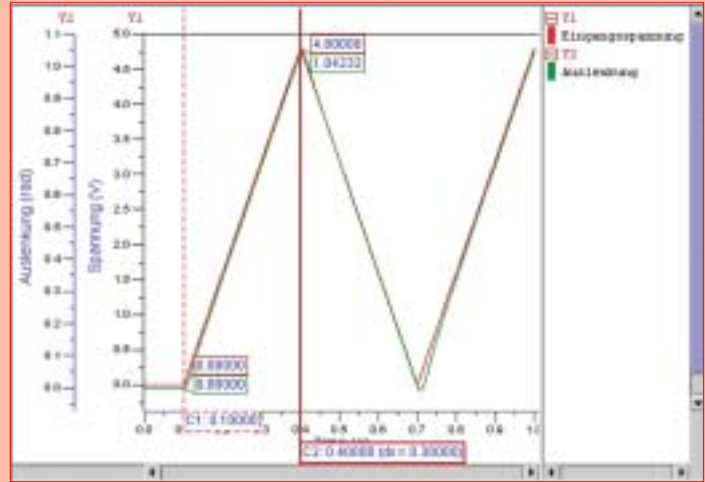


Abb. 4: Gesamtsystem Verifikation unter Berücksichtigung der elektromechanischen Effekte

renten Integrator (0 Hz). Bei zunehmendem Gewinn bilden diese ein konjugiert komplexes Polpaar, das sich zur positiven Halbebene bewegt. Daraus resultiert die geringe Phasenreserve bei einem Gewinn von 100. Durch Einfügen eines PDT<sub>1</sub>-Gliedes mit einer Nullstelle bei 5 Hz und einem unkritischen Pol bei 2 kHz verlagert sich die konjugiert komplexe Polstelle bei hohem Gewinn von der imaginären Achse weg in die negative Halbebene hinein. Das System ist stabil. Aus dem Bode-Diagramm in Abbildung 3 ist der Effekt der Kompensation ersichtlich. Die Phasenreserve beträgt 45°, womit auch die dritte Spezifikation erfüllt ist.

Schließlich erfolgt die Implementierung und Verifikation mit elektromechanischen Komponenten. Das dynamische Verhalten wird nun in der jeweiligen physikalischen Disziplin über die charakteristischen Gleichungen beschrieben. Ein solches VHDL-AMS-Modell des oben beschriebenen Gleichstrommotors hat die im folgenden Kasten gezeigte Struktur:

Mit dem Bibliotheksaufufr IEEE\_proposed werden am Anfang die benötigten physikalischen Natures geladen. Die externen Anschlüsse sind konservative Terminals vom Typ electrical und rotational\_v, die fest mit den jeweiligen, in der Architektur definierten, physikalischen Größen eines analogen Zweiges verknüpft sind. Als Parameter werden nun direkt die Kenngrößen des Motors übergeben und die in die bekannten Differentialgleichungen eingesetzt. Hier steckt die eigentliche Funktion des Modells, wobei das Attribut 'dot

die Ableitung nach der Zeit darstellt. In dieser Phase des Systementwurfs erfolgt eine Gesamtverifikation auf einer funktionalen Abstraktionsebene, auf der mittels Verhaltensmodellierung die physikalischen Effekte deren Wechselwirkungen in einem Mechatroniksystem im Sinne eines Math-based Computer Aided Prototyping detailliert berücksichtigt werden. Abbildung 4 zeigt das Verhalten des ‚realen‘ Systems – die

```
library ieee_proposed;
use ieee_proposed.mechanical_systems.all;
use ieee_proposed.electrical_systems.all;

entity DC_MotorP is
  generic ( r_wind: resistance; -- Motor Eingangs
            -- widerstand [Ohm]
            kt      : real;      -- Drehmoment Koeff.
            -- [N*m/Amp]
            l      : inductance; -- Eingangsindukti
            -- vtaet [Henrys]
            d      : real;      -- Dämpfungskoeff.
            -- [N*m/(rad/sec)]
            j      : mmoment i ); -- Trägheitsmoment
            -- [kg*meter**2]
  port ( terminal p1, p2 : electrical;
        terminal shaft_rotv : rotational_v);
end entity DC_MotorP;

architecture physical_beh of DC_MotorP is
  quantity v across i through p1 to p2;
  quantity w across torq through shaft_rotv to
    rotational_v_ref;

begin
  torq == -1.0 * kt * i + d * w + j * w'dot;
  v == kt * w + i * r_wind + l * i'dot;

end architecture physical_beh;
```

Spezifikationen werden mit der gewählten Dimensionierung eingehalten.

### Literatur

- [1] IEEE Standard VHDL Analog and Mixed-Signal Extensions – IEEE Std 1076.1-1999, IEEE, 1999.
- [2] Moser E., „VHDL-AMS – Werkzeuge im Vergleich – ein Referenz-Beispiel und seine Umsetzung“, ASIM Fachtagung, Bielefeld, 2002, <http://www-rt.upb.de/-ASIM/FG455/Start.html>
- [3] Schmidt G., „Grundlagen der Regelungstechnik“, Springer Verlag, Berlin, 1989
- [4] Ashenden, P.J.; Peterson, G.D.; Teegarden, D.A., „The System Designer’s Guide To VHDL-AMS“, Morgan Kaufmann Publishers, San Francisco, 2003
- [5] SystemVision User’s Guide, Mentor Graphics

Beitrag als PDF im Internet:

[www.publish-industry.net](http://www.publish-industry.net)  **more @ click DV63251**

### Anzeige