

Ein Generatormodell für die Energienetzsimulation mit VHDL-AMS

Autor: Dr.-Ing. Alexander Graßmann
Bertrandt Ingenieurbüro GmbH
Lilienthalstr. 50-52
85080 Gaimersheim
email:alexander.grassmann@de.bertrandt.com

Abstract.

Dieser Bericht beschreibt die Vorgehensweise bei der Dimensionierung von modernen Fahrzeugbordnetzen. Es wird dabei besonders auf die Einbindung leicht parametrierbarer Generatormodelle eingegangen. Kriterium für das Modell war der Einsatz einer standardisierten Sprache, die auf unterschiedlichen Simulatoren einsetzbar ist. Simulative Verfahren sind für Fahrzeughersteller wie Zulieferer ein unverzichtbares Werkzeug, um Entwicklungszeiten und -kosten bei gleichzeitig steigender Produktkomplexität weiter zu reduzieren („Virtuelle Fahrzeugentwicklung“).

1. Einführung

Durch die steigenden Anforderungen halten Simulationstechniken immer mehr Einzug in die Automobilentwicklung. Insbesondere im Bereich Elektronik liegt ein sehr hohes Innovationstempo vor, welches ohne den Einsatz neuer Technologien nicht mehr beherrschbar wäre. Wichtige Anwendungen sind die Simulation von Steuergeräten und den dazwischen liegenden Bussystemen sowie von Bordnetztopologien. Im Gegensatz zur klassischen Entwicklung elektronischer Schaltungen, in der Simulatoren seit vielen Jahren Einsatz finden, sind in der Fahrzeugentwicklung meist komplette mechatronische Systeme, wie beispielsweise ein Bordnetz zu simulieren.

Für die Simulation eines mechatronischen Systems eignet sich ein *Mixed-Signal-Simulator*, der sowohl analoge als auch digitale Signale verarbeiten kann. Von Anwenderseite her besteht hier in den letzten Jahren – wie auch schon bei rein digitalen Systemen – ein starker Druck zur Standardisierung. Man möchte in der Lage sein, Modelle zwischen unterschiedlichen Simulatoren zu portieren, daneben soll sich der Umlernaufwand der Mitarbeiter beim Toolwechsel in Grenzen halten. Der Programmieraufwand für die Entwicklung der Simulatoren ist relativ hoch, da ein Digitalsimulator, ein leistungsfähiger Analogsimulator sowie eine gute Kopplung zu implementieren sind. Durch die starke Konkurrenzsituation ist dennoch mit sinkenden

Preisen für die Simulatoren zu rechnen. Derzeit kommen erste Hersteller mit Simulatoren auf den Markt. Bereits heute gibt es einen kostenlos erhältlichen VHDL-AMS-Simulator, allerdings ohne eine komfortable Benutzerumgebung. Diese Entwicklung ermöglicht auch Fahrzeugzulieferern den Einsatz von Simulatoren, da sich nun die Investitionskosten deutlich verringern.

Als Beschreibungssprache scheint sich die inzwischen standardisierte Sprache *VHDL-AMS* (*Very High Speed Integrated Circuit Hardware Description Language for Analog and Mixed Signal*) durchzusetzen. Hier wurde der aus der Entwicklung digitaler Schaltungen bekannte Standard IEEE 1076 durch analoge Elemente und Kopplungselemente ergänzt.

Die umfangreichen Modellierungsmöglichkeiten fordern dem Benutzer genaue Kenntnisse über die Bedeutung der Konstrukte ab, es ist daher eine umfangreiche Einarbeitung in die Programmiersprache erforderlich.

2. Simulation von Fahrzeugbordnetzen

Aufgabe einer Bordnetzsimulation ist die Dimensionierung der Energienetzkomponenten. Die steigende Zahl an elektrischen Verbrauchern, vor allem von elektrischen Heizsystemen, belastet das Energienetz immer stärker. Es ist daher genau zu untersuchen, ob die geplante Dimensionierung von Batterie und Generator das Bordnetz zu allen Zeiten mit genügend Energie versorgen kann und

ob die Startfähigkeit des Fahrzeugs gesichert ist (siehe Bild 1).

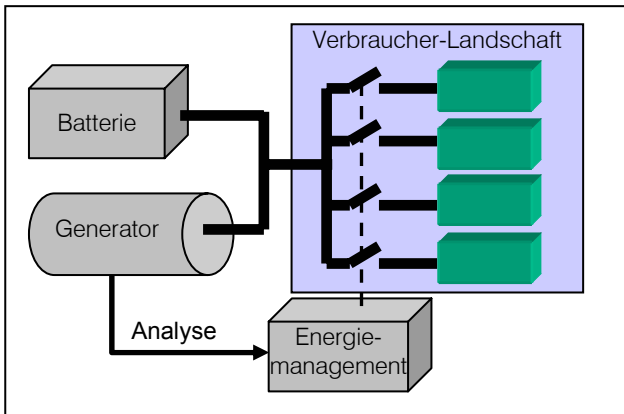


Bild 1: Schematischer Aufbau eines Fahrzeugbordnetzes

Zur Überwachung und Steuerung der Energieverbraucher setzen die meisten Hersteller mittlerweile Energiemanagementsysteme ein, die in kritischen Situationen beispielsweise Heizverbraucher abschalten. Für die Simulation des Energiemanagementsystems kommt meist ein signalflußorientierter Simulator zum Einsatz, der mit dem Mixed-Signal-Simulator zu koppeln ist. Im Rahmen der Simulation muss nun eine Vielzahl kritischer Fahrzyklen nachgestellt werden. Dazu gehört beispielsweise eine Berg- und Stadtfahrt, die jeweils im Sommer und Winter simuliert wird. Wenn man diese Variationen noch mit unterschiedlichen Generatoren und Batterien untersuchen will, entsteht eine große Zahl von Einzelsimulationen, die zur Verringerung der Simulationsdauer und zur Reduzierung der Kosten durch Automatismen unterstützt werden sollten.

Bertrandt setzt für seine Kunden dazu bereits seit mehreren Jahren eine Simulationsautomatisierung ein, welche die Ergebnisse weitgehend automatisch bereitstellt. Über ein Formular lassen sich die Anforderungen an die Simulation eingeben, die Ergebnisse werden vollautomatisch in Form von Powerpoint- und Excel-Dateien erzeugt. Auf diese Weise ist eine Trennung von Spezifikation und eigentlicher Simulation möglich. Die Simulation kann sogar extern erfolgen, beispielsweise durch einen Dienstleister. Hierdurch reduzieren sich Lizenz- und Personalkosten, daneben wird das Ergebnis wesentlich schneller und fehlerfreier bereitgestellt. Die Automatisierung ist jedoch auch für erfahrene Anwender von großem Vorteil (Bild 2).

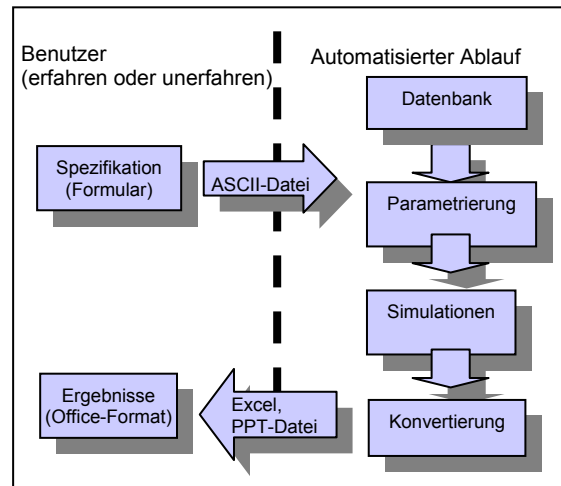


Bild 2: Arbeitsweise der Simulationsautomatisierung bei Bertrandt

Je nach Fragestellung kommen unterschiedliche Modelltypen für Generatoren zum Einsatz. Für die Untersuchung einzelner Stromimpulse verwendet man detaillierte physikalische Modelle (hochdynamische Modelle). Für diese gibt es Optimierungsprogramme zur Anpassung der Modellkennlinien an die Sollkennlinien. Dies erfordert jedoch einen relativ hohen Aufwand, so dass man Generatormodelle häufig durch Tabellenmodelle abstrahiert, was für die Simulation von Energiebilanzen ausreicht und die Rechenzeit erheblich reduziert. Die Messtabellen des Generators sind nun direkt in das Modell einlesbar. Es ist möglich, mehrere Tabellen in einem Modell zu verarbeiten, beispielsweise je eine für Stromlieferfähigkeit, Regler und Temperaturverhalten.

2. Erzeugung eines Tabellenmodells

Bild 3 zeigt schematisch den Aufbau eines Generatormodells.

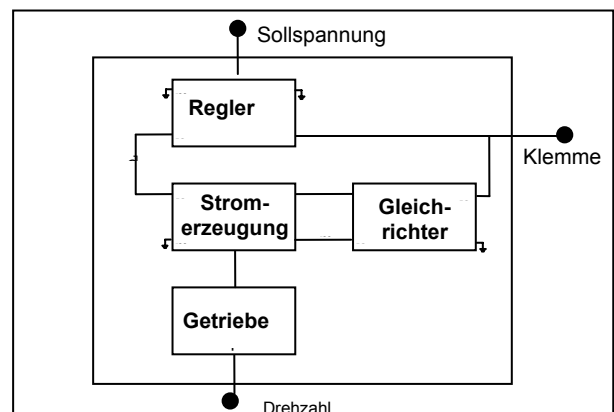


Bild 3: Schematischer Aufbau eines Generators

Die wesentliche Ausgangsgröße ist der erzeugte Strom. Dieser wird durch die Reglerspannung eingestellt und ist durch die Stromlieferfähigkeit begrenzt. Die Stromlieferfähigkeit ist abhängig von Drehzahl, Temperatur und Ausgangsspannung des Generators. Bei der Vermessung entsteht ein Kennlinienfeld, das vergleichbar ist mit der Graphik in Bild 4.

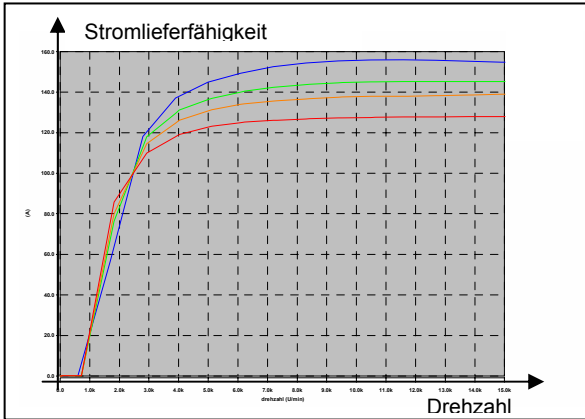


Bild 4: Stromlieferfähigkeit eines Generators

Bild 5 zeigt die Ein- und Ausgangsgrößen der Tabelle. Zwischen Generatorstrom und -spannung ist eine implizite Rückkopplung erforderlich, die im Simulator verarbeitbar sein muss.

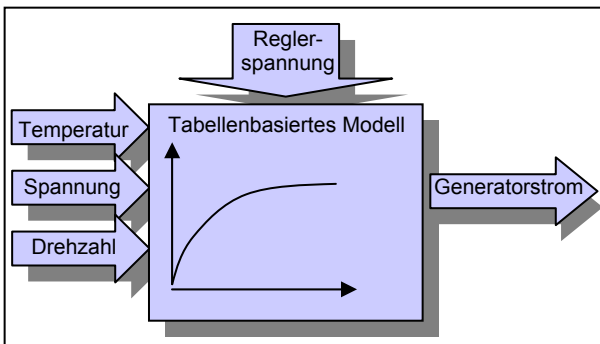


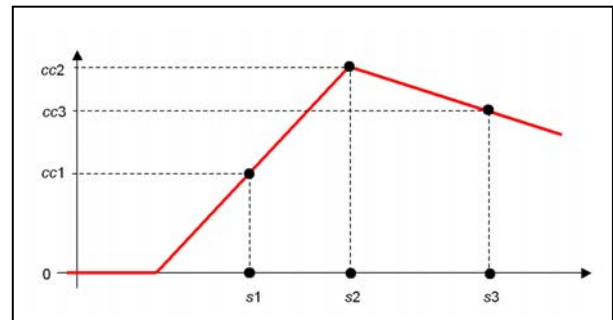
Bild 5: Ein- und Ausgangsgrößen der Tabelle

VHDL-AMS bietet in seinem Sprachumfang kein Konstrukt zur Verarbeitung von Tabellen. Auch eine Dateischnittstelle ist nicht in allen Simulatoren enthalten, so dass für die Realisierung nur die Integration in VHDL-AMS-Konstrukte bleibt.

Vorteil der Kennlinien ist Ihre Monotonie. Es lässt sich leicht eine stückweise stetige Funktion definieren, die das Verhalten der Kennlinie wiedergibt, und für die der Simulator auch jeweils einen eindeutigen Lösungspunkt findet.

```
IF u <= u2 THEN
  IF T <= T2 THEN
    IF n <= n2 THEN
      ...
```

Zwischen den bekannten Stützstellen ist eine Interpolation erforderlich, deren Algorithmus für den Fall einer linearen eindimensionalen Interpolation in Bild 6 zu sehen ist.



Algorithmus:

- Die in der Tabelle erfassten Stützstellen seien $s_1 - s_3$.
- Die zugehörige Messwerte $cc_1 - cc_3$ werden linear interpoliert.
- Links von s_1 und rechts von s_3 wird die Interpolierende linear fortgesetzt.
- Bei Werten kleiner Null wird die Interpolierende Null gesetzt.

Bild 6: Interpolationsalgorithmus des Tabellenmodells

Es ist also eine Kennlinienfeld, das in Form einer Tabelle vorliegt, in VHDL-AMS-Code umzusetzen (s. Bild 7). Dies kann durch ein Excel-Makro erfolgen. Natürlich kann man auch alternative Interpolationsalgorithmen implementieren, die dann durch den Benutzer auswählbar sind.

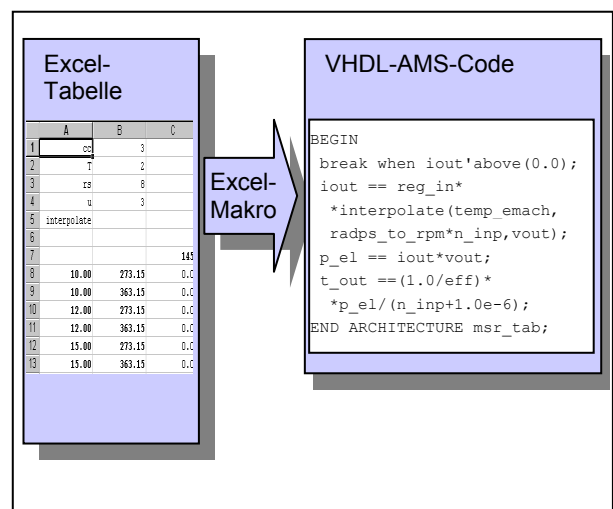


Bild 7: Umsetzung der Excel-Tabelle in VHDL-AMS-Code

Vorteil dieser Methode ist die Simulatorunabhängigkeit, da nicht alle Simulatoren Schnittstellen zum Einlesen von Tabellen bereitstellen.

Bild 8 zeigt ausschnittsweise den VHDL-AMS-Code der Interpolationsfunktion mit 3 Eingangsvariablen. Die Einbindung der Interpolationsfunktion in den restlichen Code ist in Bild 7 zu sehen.

```

FUNCTION interpolate (T: real; n: real; u: real)
  RETURN real IS VARIABLE result : real;
  VARIABLE T1 : REAL := 273.1500;
  VARIABLE T2 : REAL := 363.1500;
  VARIABLE n1 : REAL := 1450.0000;
  VARIABLE n2 : REAL := 1600.0000;
  VARIABLE n3 : REAL := 1800.0000;
  ...
  VARIABLE u1 : REAL := -10.0000;
  VARIABLE u2 : REAL := -12.0000;
  VARIABLE u3 : REAL := -15.0000;
  VARIABLE cc111 : REAL := 0.0000;
  VARIABLE cc121 : REAL := 78.1605;
  VARIABLE cc131 : REAL := 90.0030;
  ...
  VARIABLE grad1 : REAL;
  VARIABLE grad2 : REAL;
  VARIABLE point1 : REAL;
  VARIABLE point2 : REAL;
  VARIABLE grada : REAL;
  VARIABLE gradb : REAL;
  VARIABLE pointa : REAL;
  VARIABLE pointb : REAL;
  ...
BEGIN
  IF u <= u2 THEN
    IF T <= T2 THEN
      IF n <= n2 THEN
        grad1 := (cc211 - cc111)/(T2 - T1);
        grad2 := (cc221 - cc121)/(T2 - T1);
        point1 := cc111 + grad1*(T - T1);
        point2 := cc121 + grad2*(T - T1);
        grada := (point2 - point1)/(n2 - n1);
        pointa := point1 + grada*(n - n1);
        grad1 := (cc212 - cc112)/(T2 - T1);
        grad2 := (cc222 - cc122)/(T2 - T1);
        point1 := cc112 + grad1*(T - T1);
        point2 := cc122 + grad2*(T - T1);
        gradb := (point2 - point1)/(n2 - n1);
        pointb := point1 + gradb*(n - n1);
        grad := (pointb - pointa)/(u2 - u1);
        result := pointa + grad*(u - u1);
        IF result < 0.0 THEN
          RETURN 0.0;
        ELSE RETURN result;
        END IF;
      ELSIF n <= n3 THEN
        ...
      END IF;
    END IF;
  END IF;
END
  
```

Bild 8: VHDL-AMS-Code des Generatormodells (Ausschnitt)

Bei den untersuchten Testfällen gab es keine Konvergenzprobleme. Eine Testbench zur Modellverifikation lieferte die geforderten Werte (s. Bild 9). Von den Größen u (Generatorspannung), T (Temperatur) und n (Drehzahl) wurde jeweils nur eine variiert und der sich ergebende Strom ermittelt.

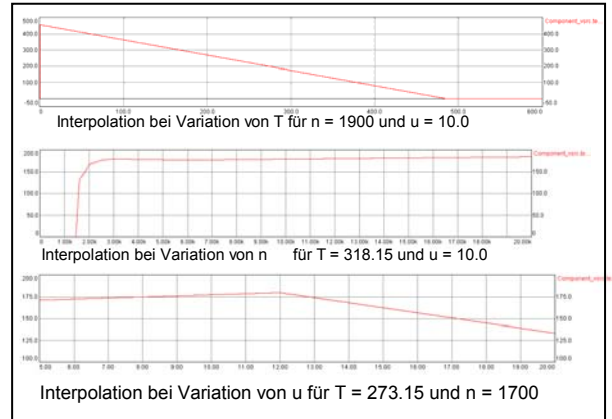


Bild 9: Simulationsergebnisse bei Verifikation des Generatormodells durch eine Testbench

4. Integration in ein virtuelles Fahrzeugbordnetz

Anschließend wurde das Modell in ein vereinfachtes VHDL-AMS-Fahrzeugbordnetz eingebaut, das freundlicherweise vom MSR-Arbeitskreis zur Verfügung gestellt wurde (Bild 10).

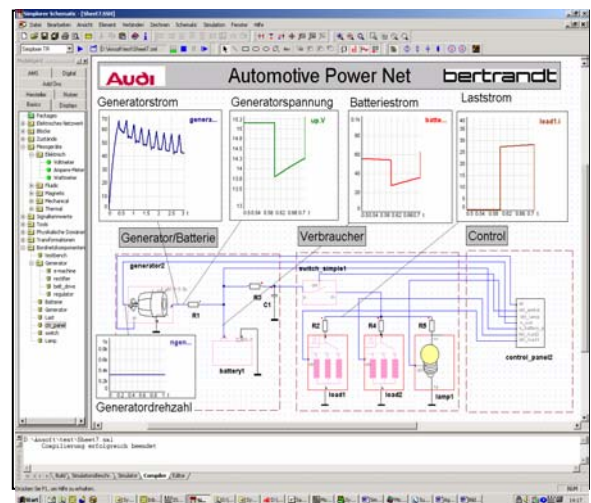


Bild 10: Integration des Generatormodells in ein virtuelles Fahrzeugbordnetz

Auch hier lieferte das Generatormodell die geforderten Werte und es gab keine Konvergenzprobleme.

5. Ausblick

Nächster Schritt wäre die Erweiterung des Bordnetzes aus Bild 10 zu einem Modell aus der aktuellen Fahrzeugentwicklung, ergänzt um eine

Kopplung zu einem Energiemanagementsystem. Die sich ergebende Performanz wäre dann mit vorliegenden Simulationen zu vergleichen.

Für eine weitere Effizienzsteigerung ist die Standardisierung von Kennliniendaten erforderlich. Langfristig ließen sich so Herstellerdaten direkt in die Tabellenmodelle einlesen, die Parametrierung könnte also vollautomatisch erfolgen.

6. Zusammenfassung

Bei der Bertrandt GmbH wurde eine Methode zur einfachen Einbindung von Generatorkennlinien in ein Modell entwickelt. Als Programmiersprache wurde die standardisierte Programmiersprache VHDL-AMS verwendet. Generatormodelle bilden einen wichtigen Bestandteil für Bordnetzsimulationen, die für die Auslegung der Energienetzkomponenten heute unverzichtbar und gut automatisierbar sind.