Zeitschrift: Bulletin Electrosuisse
Herausgeber: Electrosuisse, Verband für Elektro-, Energie- und Informationstechnik
Band: 94 (2003)
Heft: 11

Artikel: Matlab/Simulink - TINA PRO - Rational ROSE : ein Blick in den Werkzeugkasten eines Ingenieurs
Autor: Balczarczyk, Thomas / Friedlos, Ralph / Kessler, Andreas
DOI: https://doi.org/10.5169/seals-857566

Nutzungsbedingungen
Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Mehr erfahren

Conditions d'utilisation
L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. En savoir plus

Terms of use
The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. Find out more

Download PDF: 01.07.2025

ETH-Bibliothek Zürich, E-Periodica, https://www.e-periodica.ch

Matlab/Simulink – TINA PRO – Rational ROSE: ein Blick in den Werkzeugkasten eines Ingenieurs

Software-Tools unterstützen die Arbeit des Elektroingenieurs

Die Anforderungen an Systeme und deren Entwickler werden infolge der rasant wachsenden Komplexität der Technik ständig höher. Aus diesem Grunde ist der Ingenieur heutzutage stark auf die Unterstützung durch technische Software für Berechnungen, Analysen und Simulationen angewiesen. Computer Aided Engineering (CAE) geniesst daher einen entsprechend hohen Stellenwert. An der Hochschule für Technik in Rapperswil (HSR) findet CAE bereits in der Ausbildung zum Elektroingenieur Verwendung. Der vorliegende Beitrag beschreibt die wichtigsten im Unterricht eingesetzten Programme. Neben dem bekannten Programmpaket Matlab/Simulink werden auch zwei weniger verbreitete Programme – TINA PRO und Rational ROSE – erläutert.

Thomas Balczarczyk, Ralph Friedlos, Andreas Kessler, Matthias Moser, Stefan Mullis, Guido Niedermann, Mario Stocker, Leonhard Züger

Seit einigen Jahren wird an der HSR der Umgang mit Entwicklungssoftware gelehrt. So legt die HSR grossen Wert darauf, dass ihre Studenten Kompetenz im Umgang mit praxisrelevanten Entwicklungstools aus den Bereichen Mathematik, digitale Signalverarbeitung, Regelungstechnik, Elektronik sowie Informatik erwerben. Nachfolgend werden die Programme Matlab/Simulink (für mathematische Berechnungen und Visualisierungen), TINA PRO (für Zeichnen, Simulieren und Analysieren digitaler, analoger oder gemischter Elektronikschaltungen) und Rational ROSE (eine Entwicklungsumgebung für objektorientierte[1] Analyse und Design) beschrieben.

Der vorliegende Beitrag wurde von Studierenden der HSR in Rahmen des Moduls «Berufsbezogenes Schreiben» erarbeitet (Kasten I). Das beschriebene Programmpaket Rational ROSE wurde von ihnen in einem Softwareprojekt[2] verwendet.

Das Programmpaket Matlab/Simulink

Der Ingenieur ist heute stark auf technische Software für Berechnungen, Analyse und Simulation angewiesen. Daher lernt der angehende Ingenieur schon in der Ausbildung an der HSR, mit mathematischen, computerunterstützten Hilfsmitteln umzugehen und sie richtig anzuwenden.

Matlab und Simulink sowie ihre Toolboxen sind in Wissenschaft und Technik weit verbreitete und bekannte Programme. Die Funktionalität von Matlab umfasst dabei mathematische Berechnungen und Visualisierung der Ergebnisse. Seine leistungsfähige Sprache ermöglicht es dem Anwender, seine Aufgaben interaktiv zu analysieren sowie Lösungen zu entwickeln und Prototypen zu erstellen. Matlab wird in Industriezweigen wie Automobilbau, Luft- und Raumfahrt, Umwelt, Telekommunikation, Computerindustrie, Finanzen und Medizin angewendet. Der Name Matlab verrät auch schon den mathematischen Hintergrund: Er steht für MATrix LABoratory; die numerischen Berechnungen erfolgen also mit Hilfe von Matrizen.

Simulink ist eine Entwicklungsplattform für System-Modellierung und Simulation, wobei es als Berechnungsgrundlage Matlab benützt. Es handelt sich dabei um eine grafische Entwicklungsumgebung auf der Basis von Blockdiagrammen zur Modellierung, Analyse und Simulation einer grossen Anzahl dy-

Das Modul «Berufsbezogenes Schreiben»

Zukunftsweisende Ausbildung an der Hochschule für Technik in Rapperswil (HSR)

Ingenieuren haftet das Image an, die von ihnen durchgeführten Arbeiten mehr schlecht als recht dokumentieren zu können. Die Fähigkeit, technische Texte zu verfassen, wird kaum gefördert. Es stellt sich als besondere Herausforderung, technische Texte so zu formulieren, dass sie für Dritte inhaltlich und sprachlich verständlich sowie logisch in Aufbau und Gliederung sind. Im Rahmen des Moduls «Berufsbezogenes Schreiben» der HSR erarbeiten die Studenten zu einem von ihnen gewählten Thema einen solchen Text.

Das Modul «Berufsbezogenes Schreiben» wird seit dem Studienjahr 2002/03 im 3. Studienjahr der Fachrichtung Elektrotechnik angeboten. Es umfasst 17 zweistündige Lektionen. Während des Semesters erwerben die Studenten das für das Planen, Strukturieren und Formulieren von verständlichen technischen Texten erforderliche Rüstzeug. Als Abschluss formulieren sie einen Text, in welchem sie das Gelernte anwenden. Der vorliegenden Fachbeitrag ist das Resultat dieses Moduls.

Weitere Informationen: Prof. Dr. Annette Verhein Jarren, averhein@hsr.ch

Lesen Sie dazu auch den Bericht in der Rubrik «News», Seite 53.

```
% Kommentare in M-Files beginnen mit einem Prozentzeichen.
% Dieses M-File entspricht einem Script File.

>>load handel
% Ausschnitt aus "Der Messias" von Georg Friedrich Händel
% Sampling-Rate Fs ist 8192 Samples/sec
% Klangvektors y enthält 73113 Samples, das entspricht
% 8.925 Sekunden Spieldauer.

>>Y=fft(y);
% Klangvektor y mit der schnellen Fouriertransformation
% in den Frequenzbereich überführen.

>>f=(0:Fs/length(y):Fs-Fs/length(y))';
% Aus der Sampling-Rate und der Länge des Klangvektors
% sollte zur Veranschaulichung noch die Frequenzachse
% richtig skaliert werden.

>>plot(f,abs(Y))
% Zeichnet die im Spektrum enthaltenen Frequenzen auf die
% Frequenzachse f. Es ist der Betrag des Spektrums zu zeichnen,
% da die Fouriertransformation ein komplexes Spektrum ergibt.
```

Bild 1 Matlab-Befehle zur Frequenzuntersuchung eines Musikstücks
Eingetragen sind Grundfrequenz sowie die ersten drei Oberwellen der Violine.

namischer, nicht linearer und ereignisgesteuerter Systeme.

Vielfache Anwendung

Im Bereich der *digitalen Signalverarbeitung* können mit Matlab Fourier-Transformationen sowie Auto- und Kreuzkorrelationsfunktionen schnell gelöst werden, und die Genauigkeit von Filtern lässt sich verbessern. Ein Algorithmus für die digitale Signalverarbeitung kann vor seiner Portierung auf ein echtzeitfähiges, digitales System in einer Offline-Simulation getestet werden.

In der *Mobilkommunikation* können mit Hilfe von Matlab beispielsweise verschiedene Modulationsverfahren oder Kanalkodes untereinander verglichen, die Auswirkungen von unterschiedlichen fehlerkorrigierenden Kodes auf ein zu übertragendes Datenpaket getestet oder aber auch ganze Telefongespräche durch Rauschunterdrückung verständlicher gemacht werden.

Im Bereich der *Regelungstechnik* ermöglicht das Programmpaket Matlab/Simulink, ohne grossen Aufwand die Funktionsweise vieler Regler und Regelstrecken zu testen. Die «Control System Toolbox» von Simulink ist auf die Regelungstechnik zugeschnitten und vereinfacht die Simulation wesentlich.

Im Folgenden wird aufgezeigt, wie Matlab und Simulink angewendet werden können[3].

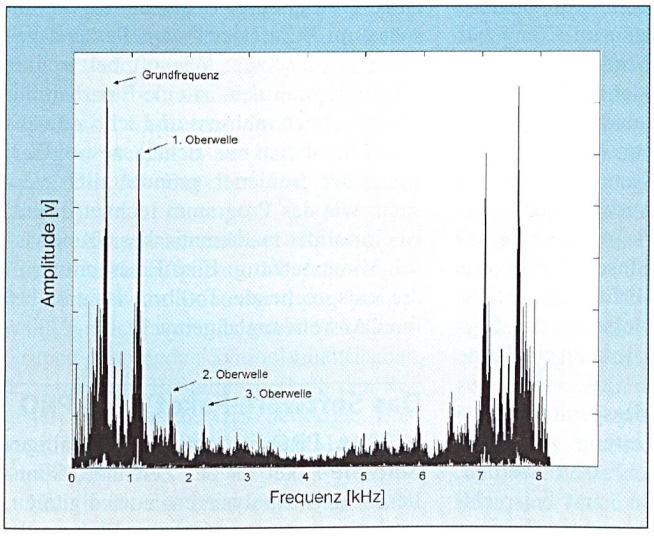

Bild 2 Frequenzspektrum des Musikstücks

Matlab

Wie bereits erwähnt, basiert Matlab auf der Matrix als Datenelement und umfasst numerische Analyse, Matrix-Rechenart, Signalverarbeitung und Grafik. Die Handlichkeit der Ein- und Ausgabe und die bewährten Algorithmen ermöglichen das Finden von Lösungen in kürzester Zeit.

Daneben kann Matlab aber noch vieles mehr, wie etwa das Exportieren von Daten in eine Excel-Datei, das Erzeugen ausführbarer Dateien oder das Generieren von Reporten.

M-Files und Toolboxen

Nach dem Aufstarten präsentiert sich Matlab übersichtlich in drei Fenstern, dem *Command Window*, der *Command History* und dem *Workspace*. Im Command Window – der Befehlszeile – gibt man die Befehle ein, die in der Command History abgespeichert werden. Die gespeicherten Variablen sind im Workspace ersichtlich.

Mit dem Befehl *edit* wird der M-File-Editor[4] von Matlab geöffnet, der für grössere Befehlsfolgen – verglichen mit dem Command Window – ein angenehmeres Arbeiten erlaubt. Damit kann das M-File unter einem geeigneten Namen gespeichert und später wieder benutzt werden.

M-Files (.m) sind ein wichtiger Bestandteil von Matlab, denn alle Toolboxen sind auf der Basis der M-Files aufgebaut. Es wird dabei zwischen *Script Files* und *Function Files* unterschieden. Ein Script File ersetzt dabei die Eingabe der Befehle im Command Window und ermöglicht dadurch ein schnelleres und bequemeres Arbeiten. Function Files hingegen sind Funktionen, denen Argumente oder Parameter übergeben werden können und die auch wieder Funktionswerte zurückgeben.

Matlab verfügt über mehr als 20 Toolboxen[5]. Da es den Rahmen dieses Fachbeitrags sprengen würde, über alle zu berichten, soll stellvertretend die *Signal Processing Toolbox* anhand eines kleinen Beispiels genauer erklärt werden.

Die Signal Processing Toolbox

Diese Toolbox ist eine Sammlung von etwa 200 Tools für die Signalverarbeitung, die von der Signalgenerierung über Filter-Design und -Implementierung bis hin zur spektralen Analyse von Signalen reichen. Die Toolbox enthält zudem eine Vielzahl von Funktionen, die in der Befehlszeile eingegeben werden können und bietet auch einige grafische Bedienoberflächen, etwa für das Filter Design oder die spektrale Signalanalyse.

Anhand eines Beispiels zur spektralen Signalanalyse soll nachfolgend ein kurzes Musikstück von etwa 8,9 Sekunden

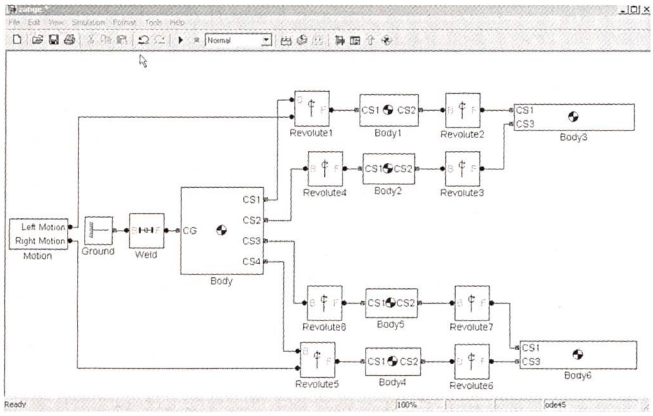

Bild 3 Modell einer «Greifzange», dargestellt mit SimMechanics

Länge auf vorhandene Frequenzen untersucht werden. In Bild 1 ist der für die Durchführung der Signalanalyse verwendete Matlab-Programmkode dargestellt.

Wie aus dem Frequenzspektrum aus Bild 2 ersichtlich ist, sind bei etwa 567 Hz und 1,134 kHz markante Spitzen vorhanden. Die erste Spitze zeigt die Grundfrequenz der Violine an, die zweite Spitze markiert deren erste Oberwelle. Knapp zu erkennen sind auch die zweite und dritte Oberwelle der Violine. Die Aufgabe des Benutzers ist nun die richtige Interpretation des Spektrums hinsichtlich Symmetrie, Einhaltung des Abtasttheorems und weiterer Aspekte der Signalanalyse.

Simulink

Simulink stellt eine Erweiterung von Matlab dar, welche die grafische Modellierung, die Simulation und die Analyse von Systemen unterstützt. Die meisten Anwendungen dieses Programms finden sich in der Regelungstechnik, im Digital-Signal-Processor-Design oder in der Nachrichtentechnik. Dem Entwickler ermöglicht die Visualisierung von komplexen Systemen mittels einfacher Symbole in einem Blockschema einen verständlicheren Einblick in das Verhalten eines Prozesses. Prozessgrössen, wie beispielsweise Geschwindigkeit, Leistung oder Effizienz lassen sich so mit einfachen Mitteln darstellen und vergleichen. Entscheidungen über die Parameterwahl – z.B. einer Regelstrecke – können so einfacher verifiziert werden.

Aus der Vielzahl von Toolboxen, die zur Verfügung stehen, soll die SimMechanics-Toolbox näher vorgestellt werden, da sie relativ neu und ihr Bekanntheitsgrad daher noch nicht sehr hoch ist.

Prozess-Darstellung mittels Simulink

Zum Erstellen eines Modells mit Simulink wird mit der *Simulink-Library* gearbeitet. Dort wählt man die benötigten Operatoren – beispielsweise ein Verzögerungsglied oder einen Integrator – für ein Blockdiagramm aus und platziert sie in einem *Modell Editor* (.mdl).

Die einzelnen Blöcke werden dann durch Leitungen miteinander verbunden, wobei Pfeile die logische Richtung des Signals angeben. Mit einem Doppelklick auf einen Modellblock erscheint das Eingabefeld für die Block-Parameter, in das – je nach Art des Blocks – einzelne oder mehrere Parameter eingegeben werden können. Die Parameter können aber auch durch Variable ersetzt werden, die dann im Command Window definiert oder mittels M-File geladen werden. Zur Anregung des Systems steht eine Auswahl von Energie- oder Informationsquellen[6] zur Verfügung. Zur Messung der Signale kann ein Scope – als virtuelles Messgerät – an eine Leitung angeschlossen werden. So kann beispielsweise der Zeitverlauf eines Signals betrachtet werden. Messwerte können auch als Variable – etwa als Array – abgespeichert und später mit Matlab dargestellt oder weiter bearbeitet werden.

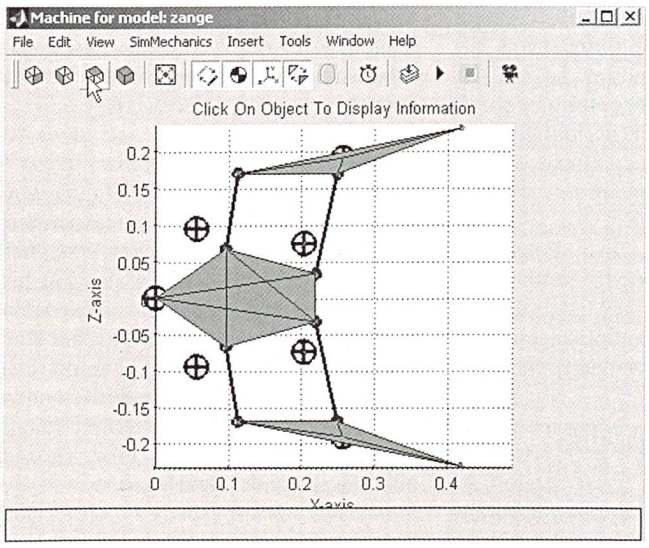

Bild 4 Simulation des «Greifzangen»-Modells

Die SimMechanics-Toolbox

SimMechanics ist eine neuere Toolbox aus dem Hause *The MathWorks*. Wie der Name schon sagt, liegt ihr Einsatzgebiet in der Simulation und Darstellung von mechanischen Bewegungen. Die wichtigsten Elemente dieser Toolbox sind *Bodys*, *Joints*, *Actuators* und *Sensors*.

Mit den Bodys lassen sich «reale» Körper darstellen. Dazu müssen Gewicht, Massenträgheitsmoment, Schwerpunkt und die Beziehung zu anderen Teilen des Systems bekannt sein. Die Verbindungen zwischen den Bodys – also die Gelenke – werden durch Joints realisiert, bei denen die Art des Gelenkes und die Ausrichtung der Drehbewegung gewählt werden kann. Mit den Actuators schliesslich können Kräfte auf das System ausgeübt werden: bei den Bodys als eine gerichtete Kraft oder bei den Joints als Winkelgeschwindigkeit, beziehungsweise Winkelbeschleunigung. Messungen im Modell wie etwa die Bestimmung des Drehwinkels eines Gelenks können über Sensoren durchgeführt werden.

In Bild 3 ist zur Illustration das Modell einer Greifzange mit dem dazugehörigen Simulations-Fenster (Bild 4) dargestellt[7].

Fazit Matlab/Simulink

Das Programmpaket Matlab/Simulink ist in Forschung und Industrie ein wichtiges Werkzeug für die Entwicklungstätigkeit und das wissenschaftliche Schaffen. Der zukünftige Ingenieur sollte diese Programme daher ansatzweise schon vor seinem Berufseinstieg kennen. Die schrittweise Einführung von Matlab/Simulink während des Studiums an der HSR ermöglicht es, diese Programme später im Berufsleben schneller und gezielter einzusetzen: Wie oft bei technischen Programmen, ist eine Einarbeitung per Handbuch mühsam und sehr aufwändig. Matlab ist nur richtig anwendbar, wenn der Bediener grundsätzlich versteht, wie das Programm rechnet. Daher ist ein solides mathematisches Grundwissen Voraussetzung. Eine Einarbeitung in die entsprechende Toolbox ist erst bei ihrer Anwendung angebracht.

Das Software-Paket TINA PRO

TINA PRO ist ein leistungsfähiges Software-Paket für das Zeichnen, Simulieren und Analysieren von digitalen, analogen oder gemischten Elektronikschaltungen.

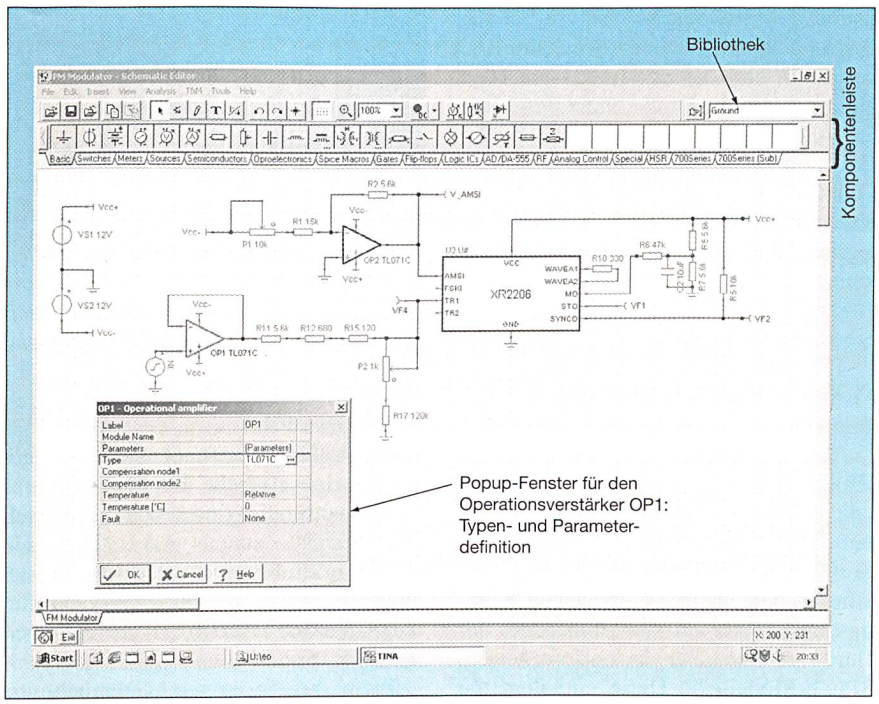

Bild 5 Benutzeroberfläche von TINA PRO mit geöffnetem Popup-Fenster in der Zeichenebene zur Wahl eines spezifischen Operationsverstärkers

Experten sagen dem Tool einen mächtigen Boom im Bereich der Schaltungssimulation voraus. Elektroingenieure finden TINA PRO ein leicht zu bedienendes Werkzeug mit hoher Leistungsfähigkeit, während Studenten seine vielfältigen Anwendungsmöglichkeiten im Übungsbereich begrüssen.

TINA PRO wurde von der ungarischen Firma Designsoft[8] entwickelt und wird in der Schweiz von Simsoft[9] vertrieben. Anfänglich war TINA PRO nur für Ausbildungszwecke erhältlich, findet nun aber allmählich den Weg in die Industrie.

Die Benutzeroberfläche von TINA PRO

Die Benutzeroberfläche des Schaltungseditors ist übersichtlich gestaltet und gut strukturiert. Am oberen Bildschirmrand befindet sich zusätzlich zur Menüleiste und einer Symbolleiste die Komponentenleiste. Sie bildet das Herzstück für das Erstellen von Schemata (Bild 5).

Die Komponentenleiste ist wie ein Karteiregister aufgebaut. Über die Register können die Bauteilgruppen ausgewählt werden, worauf dann die zu dieser Gruppe gehörenden Symbolschaltflächen der einzelnen Bauteile oberhalb des Registers erscheinen. Das benötigte Element wird dann durch einen Mausklick angewählt.

Hat man ein Bauelement gewählt, kann es mit der Maus auf die Zeichnungsebene gezogen werden. Das Element erscheint gestrichelt und kann nun gedreht oder gespiegelt werden. Mit Hilfe des Rasters wird das Bauteil ausgerichtet und durch einen Mausklick platziert. Zugleich wird es automatisch beschriftet und mit Standardwerten versehen. Auf diese Weise wird mit jeder weiteren Komponente verfahren. Zur Beschleunigung des Vorgangs kann mit der rechten Maustaste ein Menü geöffnet werden; über dieses kann das zuletzt verwendete Bauteil als Kopie nochmals platziert oder zum Standard definiert werden, wodurch bei jedem Mausklick eine Kopie erstellt wird.

Auf ein Bauteil, das nicht als Schaltfläche in der Komponentenleiste enthalten ist, kann mittels der Bibliothek direkt zugegriffen werden. Im Textfeld der Bibliothek muss dazu nur der Name des gewünschten Bauteils eingegeben werden. Zusätzlich bietet eine Komponenten-Suchfunktion weitere Hilfe.

Nach dem Doppelklick auf ein Symbol in der Zeichenebene erscheint ein Popup-Fenster, in welches Beschriftung, Werte und Parameter des Bauteils angepasst werden können (Bild 5). Im selben Fenster kann auch eine vordefinierte Komponente ausgewählt werden. Bei Speisungen wählt man in einem weiteren Fenster Signalform, Frequenz und Amplitude.

Für das Verbinden der Bauteile sind das Fadenkreuz und das Raster eine gute Hilfe. Im Menü der rechten Maustaste kann zudem festgelegt werden, dass Leitungen mitgeführt werden, sobald eine Komponente verschoben wird. Das Ergebnis ist jedoch nicht immer zufriedenstellend.

Schemata lassen sich auf diese Weise schnell und einfach erstellen. Eine Vielzahl von individuellen Einstellungsmöglichkeiten ermöglichen ein noch komfortableres Arbeiten und erlauben es zudem, die Darstellung (Layout) firmenspezifischer zu gestalten. So können etwa Hintergrundfarbe und Komponentenfarben frei gewählt und die verschiedenen Farbeinstellungen in Farbschemata abgespeichert werden. Sie sind dann in jedem Schema jederzeit aufrufbar. Es kann auch festgelegt werden, was alles eingeblendet werden soll – beispielsweise die Beschriftung von Elementen, Werte von Widerständen sowie ihre Einheiten oder das Raster.

Des Weiteren können die bestehenden Symbole der Schaltelemente abgeändert oder nach eigenen Wünschen frei erstellt werden. Zusätzlich können aber auch die Standardbeschriftung der Symbole festgelegt, die einzelnen Register der Komponentenleiste mit Symbolen ergänzt oder Elemente, die man nie braucht, gelöscht werden. Der Komponentenleiste selbst können auch neue Register hinzugefügt werden, beispielsweise für firmeneigene Bauteile.

Analysearten

In TINA PRO gibt es eine Vielzahl von Analysearten, wie etwa

- Analyse von Gleichstromschaltungen
- Symbolische und numerische Analyse in Bodeform
- Fourieranalyse
- Toleranzanalyse
- Analysen mit variablen Parameter von Elementen
- Signal-Rauschabstand
- Netzwerkanalyse
- Digitalanalyse
- Analyse mit virtuellen Instrumenten (z.B. Einsatz eines virtuellen Oszillografen).

Bei der Auswertung werden die Werte der analysierten Daten in Diagrammen verschiedenfarbig dargestellt, wobei die einzelnen Kurven beschriftet und die Achsen verschieden skaliert werden können. Mit wenigen Mausklicks lassen sich zudem Minimal- oder Maximalstellen anzeigen. Diagramme, Berechnungen, grafische Ausgaben und Messresultate können in TINA PRO mit Beschriftungen und vom System oder vom Anwender generierten Gleichungen kommentiert werden.

Bild 6 stellt Amplituden- und Phasengang der in Bild 7 gezeigten Schaltung für ein Aktivfilter in Form eines Bodedi-

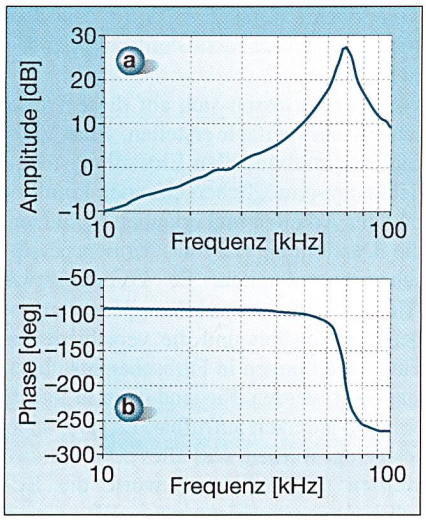

Bild 6 Frequenzgang des Aktivfilters

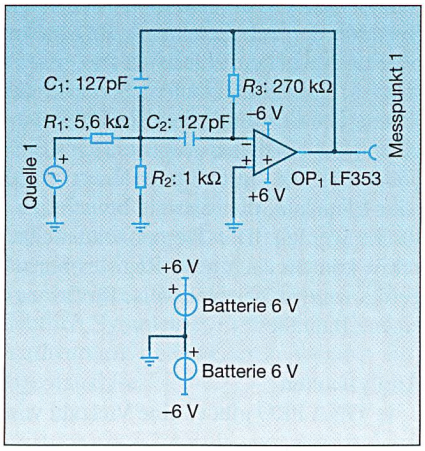

Bild 7 Beispielschaltung für ein Aktivfilter

agramms dar. Ausser dem Bodediagramm kann eine Vielzahl anderer Darstellungsarten wie etwa Pol-Nullstellendiagramm, Nyquistdiagramm oder Zeitdiagramm gewählt werden.

Ist die Stossantwort im Zeitbereich bestimmt, kann unter Einsatz der Fourieranalyse das kontinuierliche Spektrum nicht-periodischer Signale ermittelt werden. Das Spektrum (Bild 8) wird dabei entweder als Amplituden/Phasenspektrum oder als Amplitudenverteilung von Sinus- und Kosinus-Komponenten dargestellt. Das Programm berechnet zudem die Koeffizienten der Fourierreihe und die harmonische Verzerrung (Klirrfaktor) periodischer Signale.

Mit dem *Equation Editor* zeigt TINA PRO die Übertragungsfunktion in Abhängigkeit des Laplace-Operators *s*, dies sowohl in numerischer als auch in symbolischer Bodeform. Die symbolische Analyse ermittelt dabei die exakten Formeln der Übertragungsfunktionen, äquivalente Wirk- und Scheinwiderstände sowie Antworten analoger linearer Schaltungen. Durch die symbolische Analyse können Pole und Nullstellen linearer Schaltungen berechnet und dargestellt werden. Eine andere leistungsfähige Funktion von TINA PRO ist der *Interpreter*. Mit dem Equation-Editor kann man beliebige Funktionen eintragen und lineare Gleichungssysteme lösen, differenzieren, integrieren und vieles mehr. Der Interpreter kann vorhergehende Resultate und Bezugsstromkreisparameter auch weiterverarbeiten und dann zeichnen.

Vorhandene und selbst erstellte Bauteile

Bauteile können in TINA PRO unterschiedlich erzeugt werden. Eine Vielzahl von Elementen wird dem Benützer bereits in der bestehenden Bibliothek zur Verfügung gestellt. Werden jedoch spezifische oder neue Komponenten benötigt, findet man auf den Homepages der Hardware-Hersteller die entsprechenden SPICE-Modelle[10]. Diese enthalten die Spezifikationen, die TINA PRO benötigt, um ein Bauteil zu integrieren. Die Modelle können meistens kostenlos heruntergeladen und anschliessend einem Bauteil in TINA PRO hinterlegt werden. Falls der Benützer eigene Hardware herstellt und somit weder in der Bibliothek noch auf einer Homepage ein Modell finden kann, bietet sich die Möglichkeit an, diese selber zu definieren.

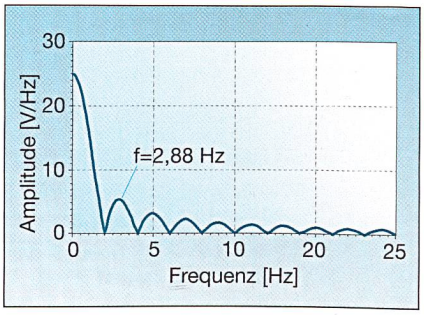

Bild 8 Fourierspektrum dargestellt mit TINA PRO

Fazit TINA PRO

TINA PRO zeichnet sich speziell durch seine einfache Handhabung aus. Ein Neuanwender findet sich schnell auf der Oberfläche zurecht und erstellt – mit ein wenig Probieren – sein erstes Projekt ohne Hilfe. Das Handbuch muss erst für weitergehende Operationen zur Hilfe genommen werden.

Beim regelmässigen Gebrauch im Labor stellt man allerdings fest, dass die simulierten Schaltungen zum Teil beachtliche Abweichungen zur Realität aufweisen. Dies liegt jedoch nicht nur an der Realitätsferne des Algorithmus – also der schlechten analytischen Nachbildung –, sondern auch an den beschönigten Angaben der Bauteilhersteller. Mit dem Einsatz von TINA PRO lässt sich aber – sofern der Elektronikdesigners über ausrei-

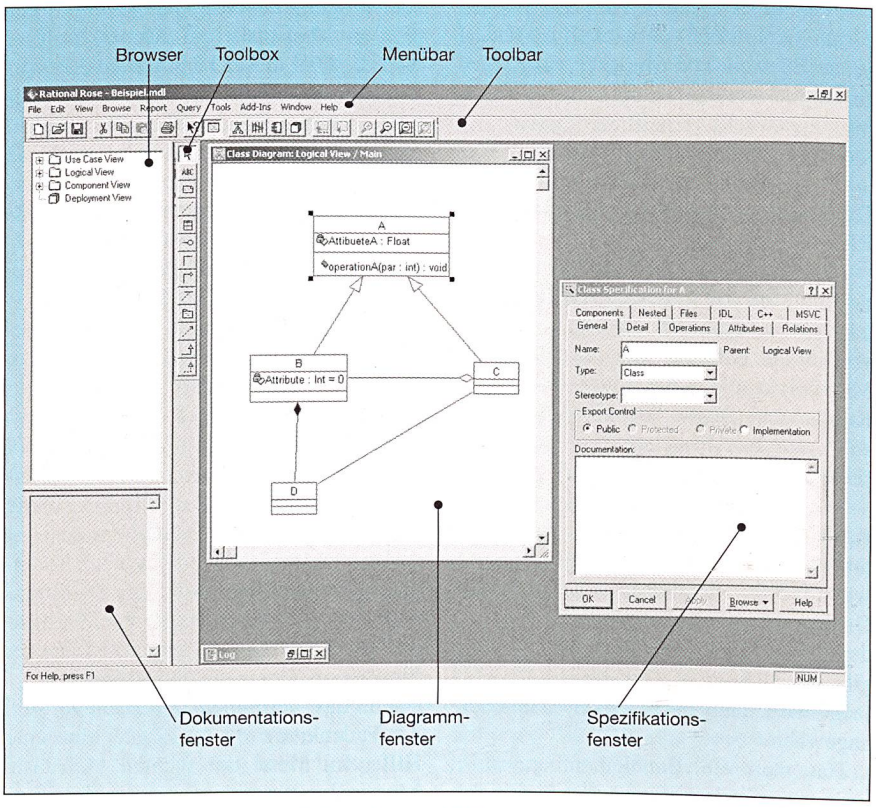

Bild 9 Das Graphical User Interface (GUI) von Rational ROSE

chende berufliche Erfahrung verfügt – eine beachtliche Effizienz bei der Entwicklung von Schaltungen erreichen.

Das Software-Paket Rational ROSE

Heute gibt es kaum ein Gerät, das keine Software enthält. Je komplexer das Gerät ist, desto grösser der Softwareumfang und desto grösser ist die Gefahr, dass man bei der Software-Entwicklung schnell den Überblick verliert. Um das Arbeiten in Entwicklungsteams mit mehreren Programmierern zu koordinieren und Parallelitäten, Namenskonflikte und Programmierfehler möglichst zu vermeiden, empfiehlt sich ein professionelles Vorgehen sowohl bei der Planung als auch bei der Gestaltung der Software.

Bei komplexen Projekten geht man meistens «objektorientiert» vor. Objektorientiertes Programmieren teilt sich in objektorientierte Analyse (OOA), objektorientiertes Design (OOD) und objektorientierte Implementation (OOI) auf. Die Firma Rational bietet unter dem Produktenamen *Rational ROSE* eine grafische Entwicklungsumgebung an, welche OOA und OOD ermöglicht. Rational ROSE ist somit die Schnittstelle zwischen dem abstrakten Modell eines Softwaresystems und dessen Sourcecode.

Diagrammtyp	Beschreibung
Use-Case-Diagramm	Beschreibt das Zusammenwirken von Personen mit einem Softwaresystem. Es kann sich an Stelle der Person auch um ein anderes Softwaresystem handeln.
Klassendiagramm	Ist ein zentraler Bestandteil. Es beschreibt die statische Struktur der Objekte in einem System sowie deren Beziehungen untereinander.
Interaktionsdiagramm	Zeigt den zeitlichen Ablauf der Zusammenarbeit von Objekten.
Sequenzdiagramm	Zeigt die zeitliche Aufrufstruktur mit wenigen Klassen.
Kollaborationsdiagramm	Zeigt die zeitliche Aufrufstruktur mit wenigen Meldungen.
Paketdiagramm	Zeigt, in welchem Modul welche Klasse zu finden ist. Aufteilung in Unterprojekte, Bibliotheken, Übersetzungseinheiten.
Zustandsdiagramm	Stellt das dynamische Verhalten des Programms dar.
Aktivitätsdiagramm	Zeigt die Geschäftsprozesse bei parallelen Prozessen.
Implementierungsdiagramm	Eignet sich besonders für die Darstellung von verteilten Anwendungen und Komponenten.
Komponentendiagramm	Zeigt die Zusammenhänge der Software (Quellcode, Binärcode und ausführbaren Programmen).

Tabelle Diagramme für die Darstellung der verschiedenen Ansichten in der Unified Modeling Language (UML)

Die Benutzeroberfläche von Rational ROSE

Die grafische Benutzeroberfläche (Graphical User Interface, GUI) von Rational ROSE stellt die eigentliche Stärke der Software dar. Mit ihr können beispielsweise Klassen-, Kollaborations- oder Sequenzdiagramme (Tabelle) per Mausklick schnell und unkompliziert dargestellt, dokumentiert und geändert werden. Dies ermöglicht eine effiziente Entwicklung komplexer Softwaresystemmodelle mit der von Rational Software Corporation entwickelten Unified Modeling Language (UML, Kasten II).

Wie Windows verfügt auch Rational ROSE über eine Befehlszeile und ein Toolbar. Weiter werden die nachfolgend beschriebenen Fenster *Browser*, *Toolbox*, *Dokumentations-*, *Diagramm-* und *Log-Fenster* benutzt.

Der Browser

Der Browser ist ein hierarchisch aufgebautes Navigationstool, welches es erlaubt, die Namen und Icons von Interaktionsdiagrammen (Tabelle), Klassen, Geschäftsprozessen (Use-Cases), Zustands- oder Entwicklungsdiagrammen übersichtlich anzuzeigen. Er stellt das eigentliche Herzstück von Rational ROSE dar; durch ihn wird das Softwareprojekt zusammengehalten und verwaltet.

Das Dokumentations-Fenster

Das Dokumentationsfenster wird gebraucht, um die verschiedenen Elemente des Softwareprojekts (Klassen, Beziehungen, Attribute, Methoden) zu beschreiben. Durch Anklicken eines Elements können Informationen für die Dokumentation im Fenster eingegeben und durch nochmaliges Anklicken editiert werden. Diese effiziente und geordnete Möglichkeit, parallel zur Entwicklung die Dokumentation zu führen, hilft, den Überblick über die zahlreichen Diagramme, Attribute und Methoden zu behalten.

Das Diagrammfenster

Im Diagrammfenster werden die gewünschten Elemente des Modells grafisch dargestellt. Jedes Icon, welches im Browser sichtbar ist, stellt ein Element dar. Klassen oder Pakete, welche nicht angezeigt werden sollen, können aus dem Diagrammfenster gelöscht werden, ohne dass sie verloren gehen. Will man nun z.B. eine bereits modellierte Klasse im Diagrammfenster betrachten, so zieht man sie aus dem Browser einfach ins Fenster. Allfällige Verbindungen zwischen momentan dargestellten Klassen werden sofort sichtbar. Weiter kann das ganze Modell in logische Pakete zerlegt werden, um es übersichtlicher darzustellen.

Das Spezifikationsfenster

Durch Doppelklicken auf das zu editierende Modellelement öffnet sich das zugehörige Spezifikationsfenster. Es ermöglicht einen schnellen und unkomplizierten Zugriff auf die Eigenschaften eines Modellelements. Einige dieser Modelleigenschaften werden ebenfalls im Icon, welches das Element repräsentiert, grafisch dargestellt und können per Maus direkt bedient werden.

Die Toolbox

Die Toolbox zeigt die spezifischen Werkzeuge, welche durch Anklicken ausgewählt werden können. Wird der Diagrammtyp (Tabelle) gewechselt, so wird die Toolbox automatisch angepasst. Falls das aktuelle Modell schreibgeschützt geöffnet wurde, erscheint die Toolbox nicht.

Das Log-Fenster

Rational ROSE verwendet das Log-Fenster, um über Fehler, Kompilierfortschritte und Resultate von ausgeführten Befehlen zu informieren. Eine solche Meldung wird zusätzlich mit Zeit und Datum versehen [1].

Arbeitsweise

Der Grundgedanke der Entwickler von UML besteht darin, eine einheitliche Notation für viele Einsatzgebiete zu haben. Entsprechend der Absicht der Hersteller sollten alle Softwaresysteme mit der UML darstellbar sein.

In einem ersten Arbeitsschritt wird das das zu entwickelnde Softwaresystem mit

Object Management Group

Systemanbieter und Anwender objektorientierter Techniken haben sich 1989 zur Object Management Group (OMG) zusammengeschlossen. Die OMG verfolgt das Ziel, Standards und Spezifikationen für verteilte objektorientierte Anwendungen zu schaffen.
Die von Booch, Rumbaugh und Jacobson bei der Rational Software Corporation entwickelte Unified Modeling Language (UML) wurde 1997 von der OMG als Standard akzeptiert.

der UML beschrieben, mit welcher versucht wird, einen Rahmen für Prozesse zur Verfügung zu stellen. Das bedeutet, dass alle Vorgehensweisen und Notationen von konkreten Prozessen mit dieser Notation darstellbar sind. Damit können die Programmierer ihre Prozesse mit der UML beschreiben.

Die UML besteht aus verschiedenen Diagrammen, die wiederum verschiedene grafische Elemente besitzen. Die Bedeutung der Elemente ist genau festgelegt.

Die Tabelle zeigt eine Zusammenstellung der für die Darstellung der verschiedenen Ansichten vorhandenen Diagramme.

Ist das gesamte Softwaresystem mit der UML beschrieben, kann in Rational ROSE Code für verschiedene Sprachen generiert werden. Im Falle von C++ beispielsweise werden von Rational ROSE nach dem Einstellen der verschiedenen Parameter für Klassen, Variablen und für die Verbindungen zwischen den Objekten entsprechende Header-Files (.h) und Source-Files (.cpp) generiert, die dem Klassennamen entsprechen[11] [2].

Fazit Rational ROSE

Handhabung

Durch Drag & Drop ist Rational ROSE grundsätzlich einfach in der Handhabung. Der Aufruf der verschiedenen Diagramme ist ebenfalls sehr einfach. Änderungen können durch Anklicken in das entsprechende Objekt vorgenommen werden. Das Wechseln zwischen OOD und OOI funktioniert reibungslos. Attribute und Funktionen können problemlos hinzugefügt oder Einstellungen geändert werden, ohne dass bereits implementierter Code verändert wird.

Einarbeitung

Ohne fachliche Unterstützung ist vor allem die Parametrisierung der einzelnen Klassen sowie ihrer Variablen und Verknüpfungen sehr unübersichtlich. Es empfiehlt sich daher, einen Kurs zu besuchen oder eine kompetente Ansprechperson zur Verfügung zu haben.

Matlab/Simulink – TINA PRO – Rational ROSE: coup d'œil dans la boîte à outils d'un ingénieur

Les outils de logiciel assistent l'ingénieur électricien dans son travail

Les exigences posées aux systèmes et à leurs développeurs deviennent de plus en plus sévères par suite de la complexité toujours croissante de la technique. Aussi l'ingénieur est-il de nos jours largement tributaire de l'assistance de logiciel technique pour ses calculs, analyses et simulations. Le Computer Aided Engineering (CAE) joue de ce fait un rôle important. A la Haute école technique de Rapperswil (HSR), le CAE sert déjà à la formation des ingénieurs électriciens. L'article décrit les principaux programmes utilisés à l'enseignement. Outre les programmes bien connus Matlab/Simulink, il expose également deux programmes moins répandus – TINA PRO et Rational ROSE.

Referenzen

[1] Model-Driven Development with UML. http://www.rational.com/media/products/rose/D185H_Rose.pdf
[2] *Heide Balzert:* Lehrbuch der Objektmodellierung – Analyse und Entwurf. Spektrum, Akademischer Verlag, Heidelberg, Berlin, 1999.

Angaben zu den Autoren

Thomas Balczarczyk, Ralph Friedlos, Andreas Kessler, Matthias Moser, Stefan Mullis, Guido Niedermann, Mario Stocker, Leonhard Züger: Studenten im 3. Studienjahr an der Hochschule für Technik in Rapperswil (HSR), Klasse E2000, CH-8640 Rapperswil, Abteilung Elektrotechnik. – *Kontakt:*
Thomas Balczarczyk: tbalczar@hsr.ch
Ralph Friedlos: r1friedl@hsr.ch
Andreas Kessler: akessler@hsr.ch
Matthias Moser: mmoser@hsr.ch
Stefan Mullis: smullis@hsr.ch
Guido Niedermann: gniederm@hsr.ch
Mario Stocker: mstocker@hsr.ch
Leonhard Züger: l1zueger@hsr.ch

[1] Objektorientiert: Zusammengehörende Daten und Methoden werden in Objekte zusammengefasst (Kapselung).
[2] Projekt PSA (Project Support Application): Software-Projekt-Agenda. Dauer: ein Semester im Umfang von zehn C++-Klassen.
[3] www.mathworks.ch
[4] Der Buchstabe M steht hier für Matlab.
[5] Financial-Toolbox, Wavelet-Toolbox usw.
[6] Informationsquellen: dienen beispielsweise zum Einspeisen eines Bits.
[7] Mit freundlicher Genehmigung von Marco Pfister, The Mathworks Schweiz
[8] www.designsoftware.com
[9] www.simsoft.ch
[10] SPICE: Simulation Program with Integrated Circuits Emphasis. Der SPICE-Algorithmus wurde an der Universität von Berkeley entwickelt und ist heute der Defacto-Standard für Schaltungssimulationen.
[11] www.sigs-datacom.de/sd/publications/os/1998/02/OBJEKTspektrum_UM_kompakt.htm