



1,2 Gute Verbindung: The MathWorks Produkte und B&R Automation Studio.

Modellierung und Simulation ersparen teure Experimente an Prototypen und sind daher in der mechanischen Konstruktion längst etablierte Entwicklungsmethoden. In der Softwareentwicklung scheiterte deren Implementierung häufig am hohen Modellierungsaufwand und am Systembruch zwischen Simulation und Zielsystem-Programmierung. Durch automatische Codegenerierung und spezifisch für B&R Automation Studio geschaffene Systemerweiterungen steht mit MATLAB / Simulink von The Mathworks eine durchgängige Entwicklungsumgebung für Automatisierungslösungen zur Verfügung, die den Weg zur robusten Steuerungssoftware verkürzt und die Entwicklungseffizienz hebt.

Autoren: Franz Enhuber / B&R, Ing. Peter Kemptner / x-technik

Modellbasierter Steuerungsentwurf

„Elektronik ist Glückssache, und Software ist ein Abenteuerspiel“, sagte vor nicht allzu langer Zeit ein älterer Maschinenbau-Ingenieur zu mir. Dass er mit dieser polemischen Aussage nicht Recht haben kann, beweisen Millionen Maschinen, die ohne software-basierende Steuerungselektronik nicht annähernd das könnten, was heute Stand der Technik im Maschinenbau ist. Woher die Meinung kommt, ist aber auch klar: Nicht selten muss bei neu konstruierten Maschinen und Anlagen noch an der Software nachgebessert werden, ehe der Hersteller sie ruhigen Gewissens an die ersten Kunden übergeben kann.

Die Schuld dafür liegt nicht notwendigerweise bei den Entwicklern der Steuerungsprogramme und der Regelungsalgorithmen. Diese müssen nicht nur unter engen Termin- und Kostenvorgaben Software mit vorhersagbarer Leistung und konkurrenzfähigem Funktionsumfang entwickeln, sondern können meist erst in späten Phasen der Gesamtentwicklung zu testen beginnen, dann nämlich, wenn auch die Zielhardware und die mechanischen Einheiten zur Verfügung stehen. Dann erst stellt sich heraus, ob die Software wirklich alle in sie gesetzten Erwartungen erfüllt oder ob sich in der Anfangsphase bislang unentdeckte Fehler eingeschlichen haben.

Gängige Praxis Simulation

In vielen Bereichen ist es daher mittlerweile üblich, bereits in den frühen Phasen der Entwicklung das Verhalten von Teilen oder Baugruppen anhand im Computer generierter Modelle der Umgebung zu testen. Kein Automobilhersteller wartet, bis er durch die Zerstörung teurer Prototypen in Crash-Tests Konstruktionsmängel entdecken kann, und auch im Maschinenbau ist die Simulation vor dem Musterbau gängige Praxis als natürlicher Teil des Konstruktionsprozesses. Überall dort, wo Gefahr für Mensch oder Maschine besteht, ist die Simulation als Methode längst anerkannt, und die Zeit- und Kostenersparnis führt zu erheblichen Verbesserungen der Wettbewerbsfähigkeit.

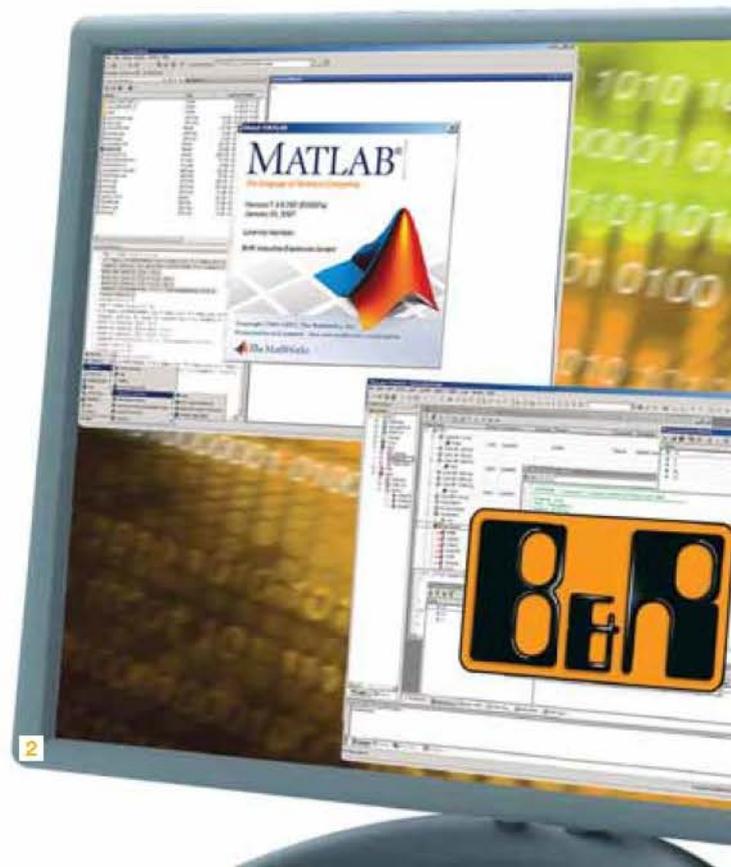
Natürlich kann keine Simulation jemals den Anspruch auf Vollständigkeit erheben, sodass letzte Erkenntnisse auch weiterhin nur aus Tests am realen Objekt gewonnen werden können. Das ist auch eines der durchaus validen Argumente der Simulationskritiker, denn auch bei guter Übereinstimmung des Simulationsmodells mit der Realität sind Redesigns zwischen Prototyp und Serie nicht auszuschließen. Das zweite Argument ist der befürch-

tete Zeitaufwand für die Entwicklung gültiger Simulationsmodelle. Hier ist zum Einen abzuwägen, welcher Aufwand durch die später zu erreichende Einsparung von Entwicklungsarbeit und Prototypenbau sowie durch die raschere Fertigstellung zu rechtfertigen ist. Zum Anderen ist für Nutzen einer Simulation von entscheidender Wichtigkeit, die für den Entwicklungserfolg wesentlichen Kenngrößen heraus zu filtern und mit größtmöglicher Präzision nachzustellen. Oberflächlichkeit zugunsten eines rascheren Projektfortschrittes ist ebenso wenig hilfreich wie der Versuch, Lückenlosigkeit zu erreichen.

Rapid Prototyping in der Automatisierung

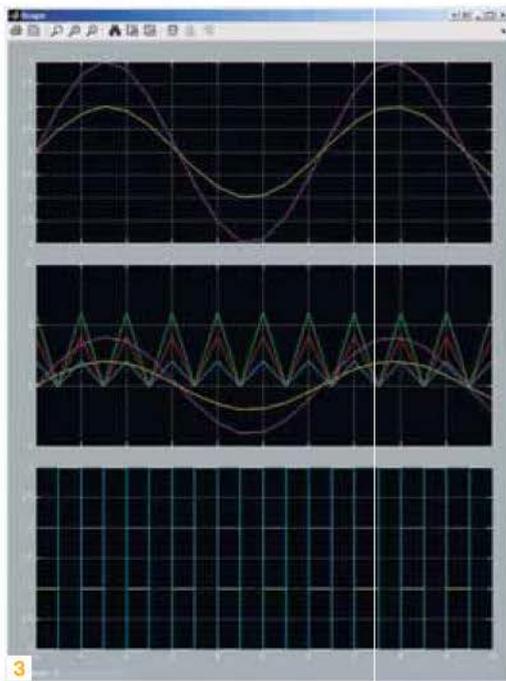
Auf den ersten Blick überraschend, ist die Simulation im Bereich der Softwareentwicklung für Steuerungs- und Regelungssysteme weniger verbreitet als etwa in der Elektronikentwicklung oder in der mechanischen Konstruktion. Das liegt zum Teil daran, dass es als sehr aufwändig gilt, das Verhalten einer Regelstrecke mit

↳ Fortsetzung Seite 36



3 The MathWorks bietet einen durchgängigen Entwicklungspfad vom Softwaremodell der zu steuernden Anwendung bis zum automatisch generierten Programmcode für B&R Automation Studio.

4 In Kooperation mit der FH Salzburg entwickelte Bernecker + Rainer systemkonforme Erweiterungen für ein nahtloses Zusammenspiel von The MathWorks und Automation Studio.



den zu berücksichtigten mechanischen Eigenschaften des Zielsystems nicht nur mathematisch zu formulieren, sondern danach in Programmcode umzusetzen.

„Schon länger gibt es daher Softwaretools wie Simulink® vom amerikanischen Hersteller The Math Works mit einer visuell orientierten, interaktiven Umgebung für Aufbau, Verwaltung und Simulation von Systemmodellen. In grafischer Form angelegt und hierarchisch strukturiert, helfen die mit MATLAB® recht einfach zu erstellenden Softwaremodelle den Entwicklern, sämtliche Funktionen ihrer Systeme effektiv zu modellieren, exakt zu dokumentieren und effizient zu kommunizieren.“, sagt Franz Enhuber, Verantwortlicher bei Bernecker + Rainer für spartenübergreifende Projekte. Die steigende Komplexität heutiger Maschinen und Anlagen mit einer Vielzahl von Aktuatoren und Sensoren bringt eine Zunahme der Fehlermöglichkeiten, aber auch der Angriffspunkte für Optimierungen. Mit immer mehr gegenseitigen Abhängigkeiten der einzelnen Teile von einander legen solche Systeme die Verwendung modellbasierter Entwicklungssysteme für die Software nahe.

„Allerdings ist die Simulation eines Prozesses nur die halbe Miete“, fährt Enhuber fort. „Nur wenn von der Formulierung der Aufgabenstellung über die Konstruktion bis zur Portierung in das Zielsystem Durchgängigkeit herrscht, lassen sich die Effizienzpotenziale in der Entwicklung von Automatisierungssoftware heben.“

Durchgängigkeit als Erfolgsfaktor

Anders gesagt: Hat ein Techniker den Realprozess, beispielsweise eine Regelstrecke, als Aufgabenstellung möglichst getreu nachgebildet und, wie das bei Simulink® Standard ist, in der selben Umgebung die zur Aufgabenstellung passende Lösung, in unserem Beispiel einen Regler, konstruiert, so ist er in der Lage, wesentliche Kenngrößen des Reglers, wie Störgrößenverhalten oder Führungsverhalten anhand des Softwaremodells zu optimieren, bis es seinen Anforderungen entspricht. Bereits das ist ein Vorteil, denn die Modellierung erlaubt präzise Tests im Vorfeld und erspart Kosten durch die Belegung des Zielsystems und die Produktion von Ausschuss während der Testphase, ganz zu schweigen von der Verkürzung der Inbetriebnahmezeiten.

Allerdings fand an dieser Stelle bisher meist ein Systembruch statt: Das Ergebnis musste in einer für das Zielsystem verständlichen Sprache nachprogrammiert werden. Das birgt einerseits die Gefahr einer nicht ganz exakten Übersetzung und macht andererseits jede Anpassung mühsam, da der gesamte Prozess noch einmal durchlaufen werden muss. Um diesen Aufwand zu vermeiden, werden erforderliche Anpassungen vielfach direkt im Zielsystem durchgeführt, sodass zwischen den modellierten und den verwendeten Versionen Unterschiede entstehen, was die Nachvollziehbarkeit und Wartung enorm erschwert.

Automatische Codegenerierung

MATLAB / Simulink versteht sich daher als vollständige Entwicklungsumgebung, die während des gesamten Entwicklungsprozesses nicht verlassen werden muss. Der Systembruch wird durch eine automatische Codegenerierung vermieden, die aus dem erfolgreich simulierten Systemmodell kompilierbaren Echtzeit-Quellcode für das Zielsystem erzeugt.

Nicht nur wird dadurch die aufwändige und fehleranfällige manuelle Programmierung der Steuersoftware vollkommen vermieden, sondern können die Erkenntnisse aus den Tests in der Echtumgebung direkt in das ursprüngliche Modell einfließen, was den Anpassungsaufwand deutlich reduziert und vollständige Tests vor der Portierung erlaubt. Durch diese Vorgehensweise profitiert neben der Erstentwicklung auch die Entwicklung von Produktvarianten und, nicht zu unterschätzen, die laufende Produktpflege.

Die Codegenerierung erfolgt in The MathWorks im sogenannten Real-Time Workshop Embedded Coder, und zwar in Form strukturierter, effizienter C-Sources, die von B&R Automation Studio kompiliert werden können und ressourcenschonend auf dem Zielsystem laufen. Da eine allgemeine Lösung die spezifischen Ansprüche eines Automatisierungsproduktes natürlich nicht berücksichtigen kann, entwickelte B&R in Kooperation mit der FH Salzburg systemkonforme Erweiterungen,



*Franz Enhuber,
Verantwortlicher
bei Bernecker +
Rainer für sparten-
übergreifende
Projekte: „Mit der
engen Kopplung
zu The MathWorks
Produkten bietet
Bernecker +
Rainer seinen
Kunden einen
Wettbewerbsvorteil
durch die leichtere
Umsetzung neuer
technischer
Möglichkeiten.“*

die ein nahtloses Zusammenspiel von The MathWorks Produkten und der B&R Entwicklungsumgebung Automation Studio gewährleisten.

„Dabei handelt es sich im Wesentlichen um „in“ und „out“ Blöcke zur Definition von Prozessvariablen zur Übergabe vom Modell zur Steuerung und umgekehrt, und um interne Parameterdefinitionen für komplexere Blöcke, die den Zugriff der Steuerung auf diese gewährleisten, auch wenn der betreffende Baustein keine Schnittstellen (Anschlüsse) nach außen aufweist“, erläutert Franz Enhuber. Bei der vierten Erweiterung handelt es sich um eine organisatorische Schnittstelle zwischen der Steuerung und der Simulationsumgebung.

Für Automation Studio entstehen aus diesen Systemerweiterungen bei der Codegenerierung als Teil des C-Codes Variablen, die während des Kompilierens zur richtigen Zuweisung der Prozessvariablen an

die Peripherie dienen. Das „Debugging“, also die detaillierte Fehlersuche, kann ohne Einschränkungen sowohl in Automation Studio erfolgen als auch im Modell.

Solcherart generierter Code kann durch die definierten Übergabevariablen beliebig zu bestehenden Softwareteilen hinzugefügt werden, was eine modulare Entwicklung begünstigt. So ist es etwa möglich, neue, zusätzliche Funktionalität isoliert zu entwickeln und an eine bereits existierende Applikation anzufügen.

Rasche Amortisation bei Neuentwicklung

Ein konkretes Beispiel illustriert den Effizienzgewinn in der Entwicklungsarbeit: „Aus Gründen der Komplexität musste das Streckenverhalten einer Regelstrecke für eine Vorsteuerung in Software modelliert und ein entsprechender Regler programmiert werden.“, schildert Franz Enhuber.

„Ab der fertigen Definition wurde drei Monate lang die Regelstrecke in C codiert, das Aufsetzen des Reglers dauerte dann noch ein weiteres Monat.“ Nur um heraus zu finden, wie groß der Effizienzgewinn wirklich ist, wurde der selbe Softwareentwickler beauftragt, die selbe Aufgabe noch einmal zu lösen, diesmal vollständig mit The MathWorks. „Einen halben Nachmittag später lieferte der Mitarbeiter das fertige Ergebnis ab, und es war in allen Funktionen vom händisch programmierten nicht zu unterscheiden.“

Der Erfolg hängt natürlich von der individuellen Aufgabenstellung ab. Je komplexer die Steuerungs- und Regelungsaufgaben sind, desto größer wird der Nutzen sein. Gleiches gilt für Hersteller von Maschinen mit hohem Risiko für Mensch oder Maschine. Generell bietet Bernecker + Rainer durch gute Abstimmung mit The MathWorks Produkten seinen Kunden einen Wettbewerbsvorteil durch die leichtere Umsetzung neuer technischer Möglichkeiten. „Speziell Unternehmen, die bereits heute The MathWorks für die mechanische Konstruktion verwenden, finden einen leichten Einstieg, da sie von dieser Seite bereits eine gute Ausformulierung vieler für eine vollständige Simulation erforderlicher Teile haben.“

① KONTAKT

**Bernecker + Rainer
Industrie-Elektronik Ges.m.b.H.**
B&R Straße 1
5142 Eggelsberg
Tel. +43-7748-6586-0
www.br-automation.com

Der **ANTRIEB**

zum
ERFOLG

Besuchen Sie uns:
Halle 1, OG Stand Nr. 1944



Linearantriebe
Spindelhubgetriebe

Planetengetriebe
Kupplungen

Automatisierungstechnik
Sonderlösungen