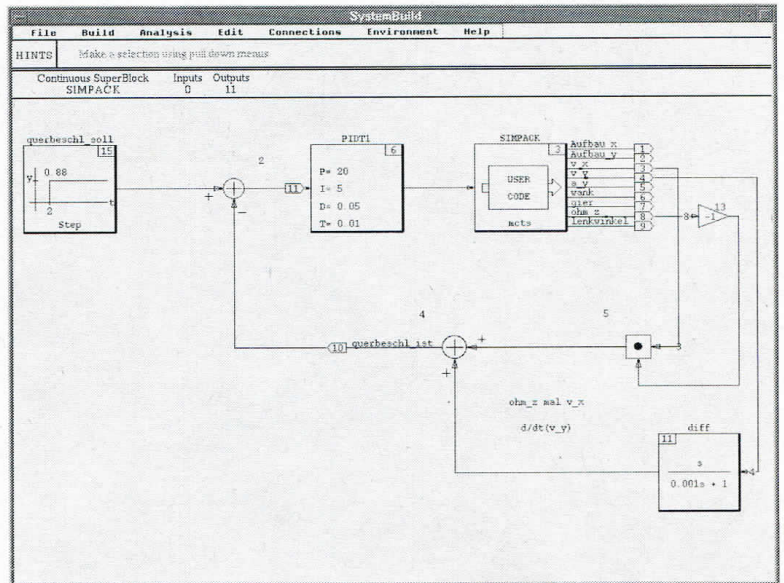


SIMAX: SIMPACK-MATRIXx-Interface

Die fortschreitende Verknüpfung der Bereiche Mechanik, Elektronik und Regelungstechnik erfordert die Integration von Software-Entwurfspaketen der einzelnen Fachgebiete für die interdisziplinäre Entwicklung neuer Produkte. Die Schnittstelle SIMAX ermöglicht eine Kopplung von SIMPACK mit dem regelungstechnischen Entwurfswerkzeug MATRIXx.

Wolf Krüger, Deutsche Forschungsanstalt für Luft- und Raumfahrt, Oberpfaffenhofen

Die fortschreitende Miniaturisierung elektronischer Komponenten sowie der Zwang zu Leichtbau und neuen Werkstoffen führt zu einer immer stärker werdenden Integration der Elektronik, und hier insbesondere der Regelungstechnik, in mechanische Systeme. Zahlreiche mathematische Programme zur Entwicklung, Auslegung und Simulation regelungstechnischer Algorithmen sind in den letzten Jahren auf den Markt gekommen, unter ihnen das Programmpaket MATRIXx (*Xmath/ System-Build*). Mit seinen verschiedenen Toolboxes lassen sich sowohl Methoden der klassischen wie auch der neueren Regelungstechnik im Frequenz- und im Zeitbereich anwenden. Modelle können direkt in Blockschaltbildern aufgebaut werden, in denen sich Verknüpfungen und Schaltelemente auf einfache Weise darstellen lassen. Hauptproblem dieser Vorgehensweise ist der Aufbau der Reglstrecke, also des zu regelnden mechanischen oder mechanischen Modells. Dieses muß weiterhin „per Hand“ aufgestellt werden und ist deswegen häufig entweder recht einfach, oder, im Falle komplexer Modelle, unhandlich. An dieser Stelle bietet es sich an, das MKS-Programm



SIMPACK als flexiblen und mächtigen Modellgenerator für Blocksimulationsprogramme zu nutzen. Die Kopplung zwischen SIMPACK und dem mathematischen/regelungstechnischen Entwurfspaket MATRIXx/System-Build ist auf drei verschiedene Weisen realisiert, wie im folgenden gezeigt wird.

DAS LINEAR SYSTEM INTERFACE

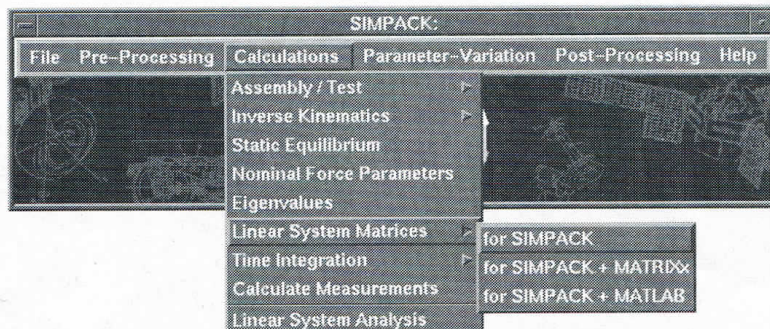
Bewegungsdifferentialgleichungen lassen sich in ihrer linearisierten Darstellung in der sog.

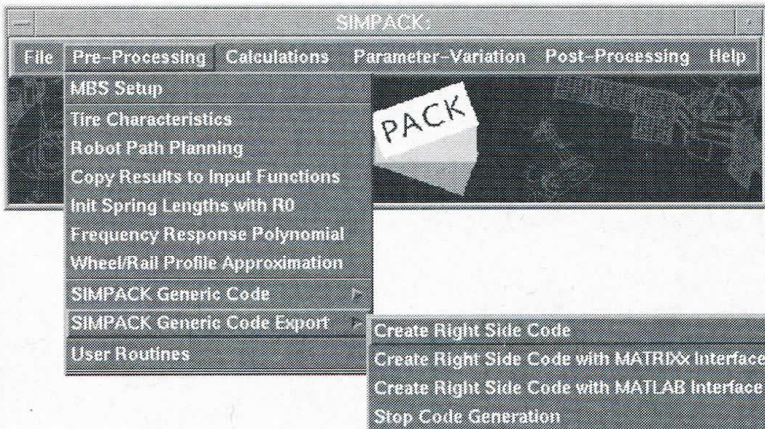
Zustandsform schreiben:

$$\dot{x} = A \cdot x + B \cdot u$$

$$y = C \cdot x + D \cdot u$$

Ein solches System mit dem Eingangsvektor u und dem Ausgangsvektor y läßt sich in *SystemBuild* direkt verarbeiten. SIMPACK linearisiert „auf Knopfdruck“ ein gewünschtes Modell um einen gewählten Arbeitspunkt und schreibt die Matrizen A, B, C, D schnittstellengerecht heraus. Diese Vorgehensweise hat den Vorteil, daß die zu übertragenden Daten plattformunabhängig (ASCII-Format) sind, d.h. SIMPACK und MATRIXx brauchen nicht auf dem gleichen Rechner installiert zu sein.





Auf diese Weise sind definitionsgemäß nur lineare Modelle übertragbar, was für viele regelungstechnische Aufgaben jedoch ausreichend ist. Die gesamte weitere Arbeit, Reglerauslegung und Auswertung, erfolgt nach dem Modelltransfer in MATRIXx.

Will der Benutzer ein nichtlineares SIMPACK-Modell in SystemBuild einbinden, so kann er auf das Symbolic Code oder das Function Call Interface zurückgreifen.

DAS SYMBOLIC CODE INTERFACE

Das Symbolic Code Interface nutzt die Fähigkeit SIMPACKs, die kompletten nichtlinearen Bewegungsgleichungen eines Modells in plattformunabhängigem FORTRAN- oder C-Code zu erzeugen. Diese können anschließend von einem beliebigen Integrator gelöst werden.

Für die Nutzung unter SystemBuild heißt das, daß man einen modellabhängigen Code erhält, der mit Hilfe einer Schnittstellenroutine an einen SystemBuild UserCodeBlock angeschlossen wird. Damit liegt das Modell in MATRIXx vor.

Der Vorteil dieser Vorgehensweise liegt in der Verfügbarkeit des kompletten nichtlinearen Modells in MATRIXx sowie der Portierbarkeit des erzeugten SIMPACK-Codes auf andere Systemumgebungen, wie z.B. den PC. Wie beim Linear System Interface erfolgt anschließend die gesamte weitere Auswertung in MATRIXx.

DAS FUNCTION CALL INTERFACE

Die dritte und umfassendste Schnittstelle bietet das Function Call Interface. SIMPACK und MATRIXx müssen dabei auf der

gleichen Plattform installiert sein. Mit dieser Schnittstelle läßt sich die Modellbildungsfunktionalität SIMPACKs modellunabhängig in MATRIXx integrieren. Diese Verbindung baut ebenfalls auf dem SystemBuild UserCodeBlock auf. Zwei Besonderheiten zeichnen das Function Call Interface aus: zum einen ermöglichen es die Integrioren DASSL und ODASSL, sowie der sogenannte „Implicit UserCodeBlock“, daß auch mechanische Modelle mit kinematischen Schleifen simuliert werden können. Modelle dieser Art führen zu sogenannten differential- algebraischen Gleichungen (DAE), die für viele Integrioren nicht auswertbar sind. Des weiteren ist es beim Function Call Interface möglich, die Integrationsergebnisse des geregelten Modells an SIMPACK zurückzugeben, so daß sie in der SIMPACK 2D-Auswertung und insbesondere der 3D-Animation analysiert werden können.

Die folgende Übersicht verdeutlicht noch einmal die Unterschiede zwischen den drei Schnittstellen:

