

Gerhard Schlicht, Marco Schmid

Motion embedded

Robotik- und Motion-Anwendungen sind Mechatronik pur, bei denen Mechanik, Elektronik, Regelungstechnik und Applikationssoftware parallel zu entwickeln sind. Grafische Entwicklungsmethoden machen diesen komplexen Mix aus Disziplinen beherrschbar.

Viele Ingenieure skizzieren ihre Ideen und Gedankenmodelle auf Papier, beispielsweise als Blockschaltbild, das als Grundlage für die anschließende Programmierung in einer textbasierten Sprache dient. Im Gegensatz dazu beginnt die grafische Programmierung bereits einen Schritt früher, beim Skizzieren: Das Gedankenmodell wird direkt als Blockschaltbild auf dem Bildschirm entworfen. Daraus generiert die grafische Programmierumgebung automatisch den Maschinencode.

Mit dieser in der embedded Systementwicklung sowie Mess- und Automatisierungstechnik etablierten Methode lassen sich ebenso Robotik-Systeme auf einem hohem Abstraktionsniveau beschreiben und analysieren. Das vermittelt ein Gefühl für die Zusammenhänge, für wichtige Systemparameter und für das Verhalten des Gesamtsystems. Beispielsweise können Entwickler die Drehmomentmessung in ihrer physikalischen Einheit (Nm) oder die Berechnung einer „S-Kurve“ zur sanften Ansteuerung einer Achse über Funk-

tionsblöcke ohne Detailkenntnisse der an sich komplexen Teilaufgaben einfügen.

Auch weniger erfahrene Entwickler können dank der interaktiven Tools anspruchsvolle Aufgaben wie die Identifikation von Regelstrecken und die Auslegung von Regelkreisen lösen sowie komplexe Modelle aus klassischen Mathematikprogrammen einbinden. Selbst fortgeschrittene Konzepte wie Trajektorien-Berechnung, kaskadierte Regelkreise, neuronale Netze, Fuzzy-Control und Neuro-Fuzzy sind machbar.

Bewegungen unter Kontrolle

Linearachsen, Schlitten oder Knickarme sind wichtige Elemente in der Robotik. Ihre Ansteuerung verlangt normalerweise fundiertes Know-how über Bewegungsplanung und Regelungstechnik – vor allem, wenn mehrere Achsen synchronisiert werden sollen. Die Bewegungen solcher Multi-DOF-Systeme mit vielen Freiheitsgraden (DOF: Degrees of Freedom) beruhen auf komplexen mathematischen Modellen, die in Echtzeit umgesetzt wer-

den müssen. Mit Hilfe der Mathematik- und Analysefunktionen von LabVIEW sowie Toolboxen für Motion, Mess- und Regelungstechnik wird diese Komplexität überschaubar. Die Bewegungsbahn ist dabei auf vier verschiedene Arten programmierbar:

- Ein- und mehrachsige Bewegungssteuerung mit parametrierbarer Sollgeschwindigkeit und Beschleunigung über Funktionsblöcke.
- Manuelle Vorgabe (Teach-In) über Joystick und Speicherung des „gelernten“ Bewegungsablaufs.
- Über 3D-CAD-Software, die ein visuelles Überprüfen der simulierten Bewegungsbahn eines am Bildschirm entworfenen Robotermodells ermöglicht. Die Modelle werden als „Virtual-Reality“-Dateien aus dem CAD exportiert, in LabVIEW importiert, dort in 3D visualisiert und auf der Zielplattform live ausgeführt. Die Bewegungen erhalten ihre Feinabstimmung über den Vergleich der virtuellen Modelle mit den realen Abläufen. Diese Technik wurde beispielsweise zur Fernsteuerung eines Baggers in 1000 m Tiefe unter dem Meeresspiegel eingesetzt.
- Kontinuierliche Berechnung der Bewegungsbahn über die inverse Kinematik in Echtzeit direkt auf der Zielplattform.

24 Achsen grafisch programmiert

Für Einsätze in Katastrophengebieten hat das Forschungsinstitut Nanyang Polytech-

(Bilder: Schmid Engineering)

nic in Singapur einen sechsbeinigen Roboter auf Basis von LabVIEW und der Hardwareplattform ZBrain entwickelt. Die Aufgabe des Roboters ist, bei Katastropheneinsätzen in schwer zugänglichen Hohlräumen nach eingeschlossenen Personen zu suchen. Deshalb wählten die Entwickler ein Laufprinzip mit sechs autonomen Beinen. Deren Mechanik und die Steuerung der insgesamt 24 Freiheitsgrade (6x4) waren die Schlüsselfunktionen des Projekts.

Zusätzlich enthält der Roboter Subsysteme für Bildverarbeitung, Entfernungsmessung und drahtlose Kommunikation. Eine intelligente Kamera lokalisiert und vermisst Objekte mit leistungsfähigen Bildverarbeitungsalgorithmen, um einen bestimmten Suchbereich zu ermitteln oder um Farben zu identifizieren. Mit der stromsparenden CPU-Plattform ZMobile und der grafischen Entwicklungsumgebung LabVIEW wurde der flexible, sechsbeinige Roboter innerhalb weniger Monate realisiert.

Die zentralen Bewegungsmechanismen finden sich abgewandelt in vielen Industrie-Anwendungen wieder, zum Beispiel in Knickarmrobotern und Handling-Systemen. Die Algorithmen können daher direkt in industriellen Applikationen übernommen werden.

Die Akzeptanz grafischer Tools ist bei Entwicklern von Embedded-Software mit den Möglichkeiten gekoppelt, bei Bedarf die Abläufe und das Timing der Software sowie die Hardware-I/Os direkt beeinflussen zu können – und zwar bis hinunter auf Assembler- und Registerebene. Dazu müssen sich C-Code und Assembler-Routinen in den grafischen Kontext einbetten lassen. Je nach Aufgabenstellung können Entwickler das passende Programmiermodell (textuelle Mathematik, Simulationsknoten, Datenfluss-Programmierung, Zustandsdiagramme) wählen und die Elemente im grafischen Kontext der Funktionsskizze zusammenfügen. Die Grundlage hierfür bilden die unterschiedlichsten Funkti-

onsblöcke (Virtuelle Instrumente/VIs), die mit Verbindungslinien zu einem Blockschaltbild verdrahtet werden. Aus dem grafischen Blockschaltbild wird automatisch der funktionsfähige C-Code generiert. Dieser C-Code wird über Standardtools mit einem Echtzeit-Kernel verlinkt und als deterministische Firmware in das Bootflash des Zielsystems geladen.

Lösungsneutrale Applikations-Frameworks für Funktionen wie Prozess I/O, Fehlerbehandlung, Datenverarbeitung und -speicherung, grafische Bedienerschnittstelle und Web-Server stehen bereits als echtzeitfähige Multitasking-Umgebung zur Verfügung und verkürzen die Einarbeitung. Darüber lassen sich die noch theoretischen Algorithmen mit realen Sensoren und Aktoren verbinden und testen. Das deckt Entwicklungsfehler in einer frühen Projektphase auf, in der ein Wechsel des Regelungskonzepts oder der Hardwareplattform, noch keine hohen Kosten verursacht. Der Übergang

Greifen, tragen, drehen, schneiden, biegen, formen...

Höchstleistung auch unter
schwierigsten Anforderungen: ODU ROB –
der robuste Rundsteckverbinder für höchste
Effizienz in der Automatisierungstechnik.

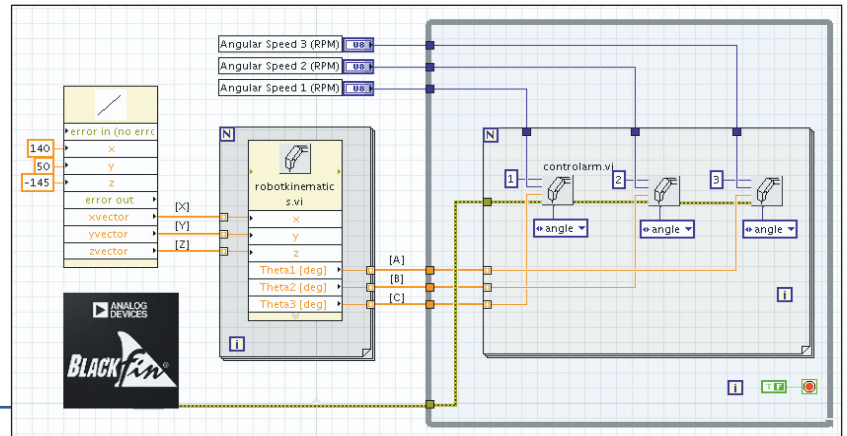
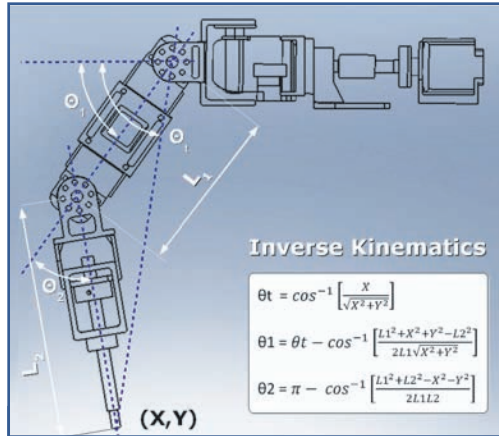


ODU ROB – der robuste Rundsteckverbinder für Industrieroboter und Robotersysteme

- einfache Montage durch Einrasten der Kontakte
- IP 67 im gesteckten Zustand
- Brennbarkeitsklasse UL 94-VO
- Kabelgehäuse in jede Richtung drehbar



ODU Steckverbindingssysteme
www.odu.de



Robotik wird beherrschbar: Komplexe Bewegungsabläufe wie eine inverse Kinematik (*links*), sind als Funktionsbausteine in die grafische Programmierumgebung LabVIEW integriert (*rechts*).

zwischen den Phasen der Produktentwicklung wird damit fließend. Der grafisch programmierte Code, zum Beispiel ein spezieller PID-Regel-Algorithmus aus der Machbarkeits-Studie, ist weitgehend plattformunabhängig und lässt sich auf Prototypen wie auch auf die spätere Serien-Hardware portieren. Für die dafür notwendige Flexibilität sorgt National Instruments Konzept rekonfigurierbarer Ein-/Ausgänge auf den Plattformen CompactRIO und SBRIO, bei denen die Hardwarefunktionen skaliert und über High-Level-Funktionsblöcke angespro-

chen werden können. Auch die Festlegung auf die CPU-Plattform kann aufgrund der grafischen und damit plattform-neutralen Systemsprache ohne Einfluss auf die Software-Entwicklung erst in einem späteren Projektstatus erfolgen. Die Referenzplattform ZMC (Measurement, Motion & Control) von Schmid Engineering basiert auf dem stromsparenden RISC-Prozessor Blackfin (BF548) der Firma Analog Devices und verschiedenen Schnittstellen (Analog-, Digital- und Kommunikations-I/O). Die flexible Hardware schafft zusammen mit

dem grafischen Multitasking und den deterministischen Echtzeit-Diensten die Voraussetzung, in kürzester Zeit MSR-, Robotik- und Motionaufgaben zu realisieren. sk

Gerhard Schlicht

ist Verkaufsleiter bei der Firma CC&I Computer Communication & Interface in Gauting bei München.

Marco Schmid

ist Geschäftsführer der Firma Schmid Engineering in Münchwilen, Schweiz.

■ Kompaktsteuerungen:

0,21 µs Verarbeitungsgeschwindigkeit

Die „Melsec FX3“-Steuerungsfamilie von Mitsubishi Electric hat ein neues Mitglied: Die speziell für kleinere Automatisierungen im Maschinen- und Anlagenbau entwickelte Kompaktsteuerung FX3G ist durch Erweiterungs-, Funktions- und Netzwerkmodule flexibel einsetzbar. Sie basiert auf dem gleichen technischen Konzept wie

die Kompakt-SPS-Serien Melsec FX3U und Melsec FX3UC. Alle notwendigen Komponenten sind in den betriebsbereiten Grundgeräten bereits enthalten. Die Verarbeitungsgeschwindigkeit beträgt 0,21 µs je logischer Anweisung, der Speicher reicht für bis zu 32 000 Programmschritte. Zudem verfügt die Steuerung neben dem konventionellen Systembus für bis zu acht Erweiterungs-, Funktions- und Netzwerkmodule über ein zweites Bussystem. Bis zu sechs Module – zwei Schnittstellenmodule (RS232 oder RS485) und vier analoge Funktionsadapter – lassen sich an diesen Adapterbus zusätzlich

anschießen. Die eigens für die Melsec-FX3-Familie entwickelten Adaptermodule können ohne zusätzliche Programmierung eingebunden und durch spezielle Datenregister direkt angesprochen werden. Eine dritte Erweiterungsmöglichkeit für serielle Kommunikationsverbindungen (RS232, RS422 oder RS485) und analoge Funktionen bieten bis zu zwei Erweiterungskarten, die an der Frontseite der Gehäuse eingesteckt werden. Die Programmierung erfolgt mit der Programmiersoftware „GX Developer“.

Die Kompaktsteuerung unterstützt die Anbindung an Netzwerke wie Profibus-DP, CC-Link, Ethernet, Devicenet, CANopen und AS-Interface ebenso wie den Datenaustausch über bis zu drei serielle Schnittstellen. In die Gehäusefront integriert sind eine USB- und eine RS422-Programmierschnittstelle.

Mitsubishi Electric Europe B.V.
Tel. (0 21 02) 4 86 – 0
www.mitsubishi-automation.de



Kleinsteuerungen
von 4E/A bis 32E/A
Bollrath elektronik
Burloer Str. 31 46414 Rhede
Tel.: 02872-2503 Fax: 02872-6907
www.bollrathelektronik.de