

EtherCAT ermöglicht Intellectual Property (IP) im Bereich Motion Control ohne regelungstechnisch bedingten Dynamikverlust

EtherCAT enables Intellectual Property (IP) for Motion Control without limiting the Performance

Prof. Dr.-Ing. Jens Onno Krahl, MSc. Christoph Klarenbach, FH-Köln, Köln, Deutschland
Jens_Onno.Krahl@FH-Koeln.de – Christoph.Klarenbach@FH-Koeln.de

Kurzfassung

Eine wesentliche Eigenschaft der auf EtherCAT basierenden XFC (eXtreme Fast Control) Technology besteht darin, dass die Funktionalitäten dedizierter Spezialsteuerungen zentral zusammengefasst werden können. Neben Kostenersparnissen sind deutlich leistungsfähigere Algorithmen möglich, die sich flexibel an andere, im zentralen IPC bekannte, Randbedingungen anpassen lassen. Auch Servoantriebe sind unterlagerte Spezialsteuerungen im Sinne des gesamten Automatisierungssystems. Der Beitrag zeigt die Machbarkeit und den Nutzen, einen Servoantrieb in seine Einzelkomponenten zu zerlegen und den Regelkreis mit Hilfe von XFC direkt in TwinCAT zu schließen. Durch diese Vorgehensweise wird der Einsatz von Motion Control IP möglich, was Innovationen begünstigt.

Abstract

An important feature of the EtherCAT-based XFC (eXtreme Fast Control) technology is that the functionalities of dedicated special controllers can be integrated in the central control unit. In addition to cost savings, significantly higher performance algorithms are possible, which can be easily adapted to other known constraints in the central control unit. From an automation system perspective, Servo Drives are subordinate special controllers. The article shows the feasibility and benefits of taking a Servo Drive apart into its individual components and including the control loop directly in TwinCAT using XFC. This approach enables the use of Motion Control IP, which effectively promotes innovation.

1 Einleitung

In der Automatisierungstechnik ist heute der intelligente Antriebsregler Standard. Insbesondere in der Servoantriebstechnik sind viele Geräte modular aufgebaut. Die Grundgeräte sind leistungsmäßig skaliert und durch die Stromstärke gekennzeichnet. Über Optionskarten werden die Geräte mit den gewünschten Schnittstellen ausgestattet:

- Durch die Feedbackoption kann zwischen Resolver und diversen Encoderschnittstellen gewählt werden.
- Mit einer Feldbusoptionskarte können alternativ diverse Feldbusse bzw. Industrial Ethernet Varianten eingesetzt werden.
- Oft kann man auch eine SPS/PLC Option integrieren.

Leider ist die Modularisierung bei vielen Herstellern ähnlich konzipiert, aber nicht identisch bzw. kompatibel. Von einem übergreifenden Standard, wie z.B. dem Norm Asynchronmotor (ASM) bzw. dem Industrie PC (IPC) sind Servoantriebe weit entfernt. Dadurch, dass weder Hardware noch Software standardisiert sind, entsteht kaum Wettbewerb. Den Preis dafür zahlen die Maschinenhersteller über relativ hohe Produktpreise.

2 Modulares Motion Control

Bis vor wenigen Jahren gab es keine Alternative zum intelligenten Antriebsregler, weil die Bandbreiten der Feldbusse

und die Verarbeitungsleistung der Prozessoren in den Steuerungen zu gering waren. Erste Forschungsergebnisse zu einem offenen Antriebsregler wurden in [1] vorgestellt.

In diesem Beitrag soll gezeigt werden, welche sinnvolle Modularisierung mit standardisierten Schnittstellen heute technisch möglich ist. Eingesetzt werden dafür folgende Technologien:

2.1 EtherCAT

Als Schnittstelle wird Sercos via EtherCAT entsprechend der IEC 61800-7 genutzt. Sercos III und EtherCAT sind die mit Abstand am besten für Motion Control einsetzbaren Feldbusse. Der Vorteil von EtherCAT ist die deutlich höhere Verbreitung, die flexibleren Busstrukturen und die Distributed Clocks (DC) Funktionalität zur hochgenauen Synchronisation von Antrieben und Encodern.

Ein Grund für die hohe effektive Bandbreite von EtherCAT (und SERCOS III) ist der Summenrahmen. Statt jeden Netzwerkteilnehmer über eine IP Adresse anzusprechen, werden möglichst viele Daten in einen gemeinsamen Summenrahmen zusammengepackt. Dieser benötigt nur einen Ethernet-Header für viele Feldbusknoten, die er dann nacheinander durchläuft. Der Preis für die hohe Effizienz ist, dass der EtherCAT Slave Controller (ESC) nicht einen Standard Ethernet Media Access Controller (MAC) nutzen kann. Stattdessen wird ein spezieller MAC mit on-the-fly frame processing benötigt.

Bei EtherCAT – und den anderen Industrial Ethernet Feldbussen – hat sich der aus der PC-Welt bekannte RJ45 Stecker sowie der M12 Stecker durchgesetzt. Der M12 Stecker wird meist außerhalb des Schaltschranks z.B. für Encoder verwendet. Da Ethernet immer galvanisch trennende Übertrager mit 1500 V Spannungsfestigkeit verwendet, sind keine zusätzlichen Optokoppler notwendig [2, 3].

In der Automatisierungstechnik wird zwischen Prozessdaten und Servicedaten unterschieden. Die Prozessdaten – in der Antriebstechnik z.B. die Führungsgröße Stromsollwert und die Regelgröße Stromistwert – müssen in Echtzeit mit einer hohen Updaterate und geringem Jitter übertragen werden. Die Stromregler-Updatefrequenz eines Servoantriebs beträgt heute meist 62,5 μ s. Die Prozessdaten werden deshalb mit synchronen EtherCAT Frames übertragen.

Servicedaten werden benötigt, wenn ein Parameter wie beispielsweise die Proportionalverstärkung K_P eines Reglers verändert werden soll. Das ist normalerweise relativ selten notwendig. Auch eine taktsynchrone Übertragung ist meist nicht erforderlich. Die Servicedaten können deshalb meist mit asynchronen EtherCAT Frames übertragen werden, die bei Bedarf vom EtherCAT Master zwischen den synchronen Frames gesendet werden. Diese Übertragung von Servicedaten erfolgt bei EtherCAT über sogenannte Mailboxen.

2.2 IGBT Endstufen

Bei der Leistungselektronik für Antriebe dominiert der IGBT. Egal ob eine ASM mit einem einfachen Frequenzumrichter (FU) drehzahl geregelt werden soll, eine Handhabungsaufgabe mit einem Roboter angetrieben werden soll oder eine Werkzeugmaschine hochpräzise gesteuert wird, die Leistungselektronik unterscheidet sich meist nur im Aufwand für die Strommessung. Bei der Steuerelektronik ist das nicht der Fall. Während der FU mit einem preisgünstigen 8-Bit μ Controller betrieben werden kann, werden bei Antrieben für Werkzeugmaschinen eher 32-Bit Controller mit DSP Erweiterungen oder gar einer Floating Point Unit (FPU) verwendet.

Die Roboterhersteller gehen einen anderen Weg. Wegen der Verkopplung der sechs Achsen wird nur der Stromregler über den Antriebscontroller geschlossen. Die Drehzahlregel- und Lageregelkreise werden in einem achsübergreifenden Motion-Controller geschlossen. Hierfür eignet sich besonders ein Intel-kompatibler Industrie PC mit FPU. Das Preis / Rechenleistungsverhältnis ist unübertroffen. Der Bildschirm des IPC kann zur Visualisierung genutzt werden.

2.3 Stromregelung im FPGA

Bei der Stromregelung erhält man dann die besten Ergebnisse, wenn die Totzeit des Systems minimiert wird. Die Totzeit setzt sich aus verschiedenen Teilen zusammen:

- Schaltfrequenz (umgekehrt proportional)
- Strommessung
- Zeit zur Berechnung der Regel-Algorithmen

Wenn zur Strommessung Sigma-Delta ($\Sigma\Delta$) Modulatoren mit digitalen Dezimierungsfiltren genutzt werden und der Regel-Algorithmus innerhalb eines FPGAs gerechnet wird, hängt die erzielbare Bandbreite fast ausschließlich von der Schaltfrequenz ab. Hier ist immer ein Kompromiss notwendig: entweder Bandbreite (= hohe Schaltfrequenz) oder geringe Schaltverluste. Die in der Praxis gewählten Schaltfrequenzen liegen zwischen 2 und 16 kHz. Wenn die Schaltfrequenz 8 kHz beträgt, und jede Flanke (steigend und fallend) berechnet wird, beträgt die Stromregler-Updatefrequenz 16 kHz. Mit 8 kHz Schaltfrequenz lassen sich bis zu 4 kHz Stromregler-Bandbreite erreichen [4, 5].

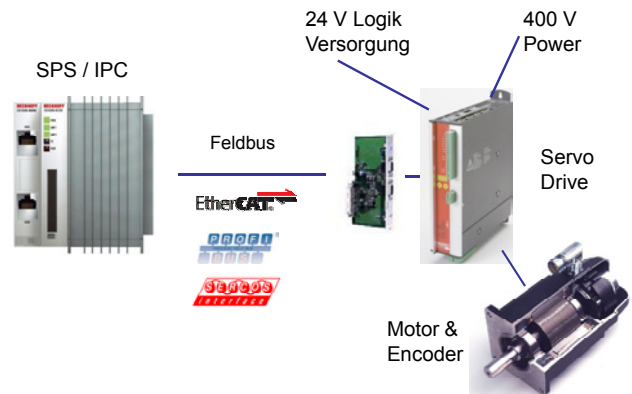


Bild 1: Physikalische Anordnung einer Motion Control Konfiguration für die Automatisierungstechnik. Der Antrieb ist über eine Feldbusschnittstelle (Optionskarte) mit der Steuerung verbunden.

2.4 Lage- und Drehzahlregelkreis im IPC

Durch die Distributed Clocks (DC) Funktionalität von EtherCAT ist es nicht mehr notwendig, den Winkelsensor (Resolver bzw. Encoder) direkt an den Antrieb anzuschließen. Es reicht aus, die Antriebe und die Lagerückführungen über EtherCAT mit dem Motion Controller (IPC) zu vernetzen. Eine solche physikalische Anordnung ist in **Bild 1** dargestellt. Unabhängig von dieser physikalischen Anordnung kann gewählt werden, ob Drehzahl- und Lageregel einzeln jeweils im Antrieb (dezentral) oder gemeinsam in der Steuerung (zentral) geschlossen werden. Da die Übermittlung der Stellgrößen bzw. Regelgrößen und die Abarbeitung des Regelalgorithmus in Summe innerhalb eines Zyklus erfolgt, ist der zentrale Ansatz genauso schnell wie der meist genutzte dezentrale Ansatz, **Bild 2**.

Aus Kostengründen ist folgende Vorgehensweise sinnvoll:

- Bei Frequenzumrichtern und beim Not-Halt durch einen Feldbus-Fehler wird eine optionale sensorlose Drehzahlregelung genutzt.
- Bei weniger anspruchsvollen Servo-Applikationen ist eine pauschale Zuordnung nicht so einfach möglich.
- Bei dynamischen Mehrachs Anwendungen ist das Schließen von Drehzahl- und Lageregel im Motion Controller mit schneller FPU sinnvoll.

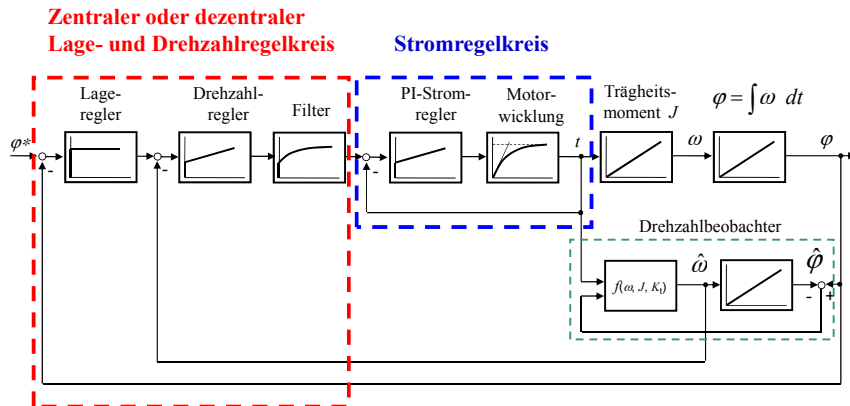


Bild 2: Signalflussplan einer zentralen bzw. dezentralen Kaskadenregelung

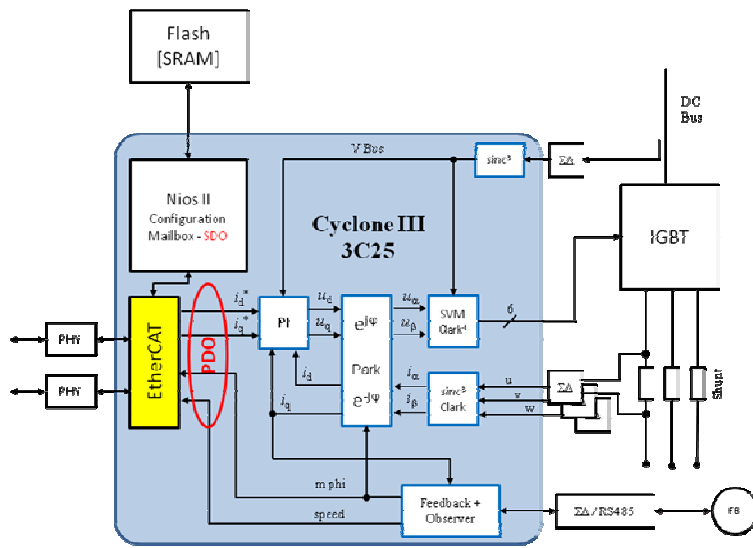


Bild 3: Signalflussplan einer FPGA basierten feldorientierten Stromregelung [5]

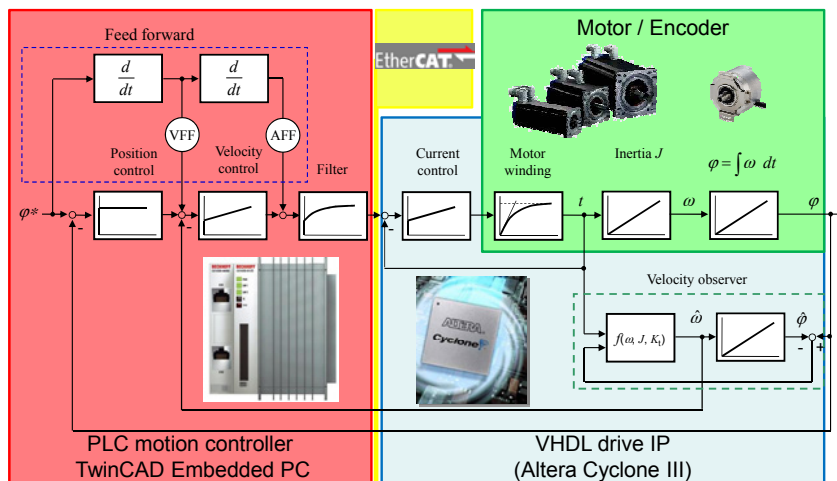


Bild 4: Die zentralen Motion Control Funktionen sind mit TwinCAT realisiert [6]

Bild 3 zeigt das Blockschnittbild der im FPGA realisierten logischen Anordnung der Stromregelung einer Achse für einen feldorientiert geregelten Antrieb. Damit das Blockschnittbild nicht zu komplex wird, sind die Signalpfade zur Konfiguration größtenteils nicht gezeichnet. Der Soft Core Prozessor NIOS II (Altera) kann über das Dual Port RAM des EtherCAT-Moduls die Mailbox abarbeiten und Chip-intern die einzelnen VHDL-Module konfigurieren und parametrieren. Im Falle eines Feldbusfehlers kann die CPU auch den Antrieb, entsprechend des konfigurierten Modus, kontrolliert stillsetzen – entweder über einen zusätzlichen, mit dem NIOS II implementierten Drehzahlregler oder über ein kontrolliertes „Kurzschließen“ des Motors.

Der Stromregler wird im FPGA gerechnet, Drehzahl- und Lage-regler in der Automatisierungssoftware TwinCAT von Beckhoff Automation, **Bild 4**. Das TwinCAT-Softwaresystem verwandelt jeden kompatiblen PC in eine Echtzeitsteuerung mit Multi-SPS-System und NC/CNC-Achsregelung.

Wenn die Steuerung für eXtreme Fast Control (XFC) konfiguriert ist, werden in einem 62,5 µs Zyklus erst die Istwerte via EtherCAT gelesen, dann werden die Regel-Algorithmen im IPC gerechnet und danach werden die neuen Sollwerte via EtherCAT an die Antriebe übermittelt. Die hochgenaue Abtastung und Synchronisation erfolgt im Antriebsregler mit den EtherCAT Distributed Clocks Trigger-Signalen, **Bild 5**.

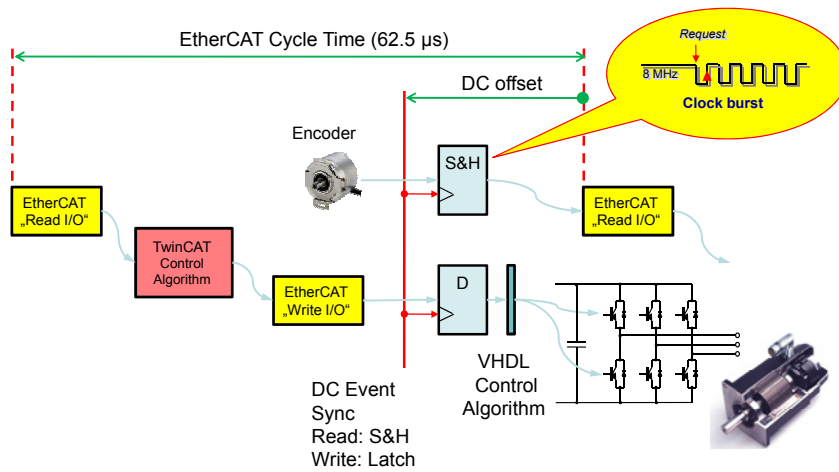


Bild 5: Zeitlicher Ablauf der Drehzahl- und Lageregelung:

1. EtherCAT-„Read I/O“ liest die Regelgrößen
2. TwinCAT rechnet die Regelalgorithmen
3. EtherCAT-„Write I/O“ schreibt Stellgrößen

Die Synchronisation erfolgt mit dem EtherCAT DC-Sync.-Signal:

3 Motion Control IP

Bisher ist der Einsatz von Intellectual Property (IP) bei Antriebsreglern kaum möglich. Nur wenige Hersteller öffnen die Geräte für Maschinenbauunternehmen oder Endkunden. Diese Vorgehensweise liegt vor allem an der geringen Leistung früherer μ Controller bzw. DSPs. Diese wurden oft mit proprietären Betriebssystemen in Assembler programmiert. Die einzelnen Tasks waren stark verzahnt und zeitkritisch. Ein Öffnen dieser Struktur für externe Programmierer war immer problematisch. Durch die heute verfügbare Hardware ist ein Antrieb viel weniger problematisch. Durch die Möglichkeit, mehrere Algorithmen innerhalb eines FPGAs parallel in VHDL zu implementieren, ist die Anzahl der Tasks innerhalb von Antrieben stark gesunken. Die aufwendigen Algorithmen können mit der extrem leistungsfähigen CPU / FPU des IPCs gerechnet werden. Auch die Nutzung der installierten Leistung ist viel effizienter, da im Motion Controller immer Compiler Technologie zu Einsatz kommt. In den universellen Standard-Antrieben wird sehr viel interpretiert, weil Austausch und Parametrierung der Antriebe ohne die Notwendigkeit eines Entwicklungssystems mit integriertem Compiler erfolgen soll. Bei einem Gerätetausch müssen deshalb nur die im Motion Controller (IPC) gespeicherten Parameter via Feldbus in den Antrieb geladen werden. Des Weiteren ist es relativ selten, dass bei einer dezentralen Struktur alle μ Controller mit der gleichen Auslastung betrieben werden. Aus diesen Gründen ist die notwendige Rechenleistung bei einer zentralen Bearbeitung und Einsatz eines Compilers deutlich geringer als die Summe der Rechenleistungen der einzelnen CPUs bei dezentraler Bearbeitung.

Bei einem IPC basierten Motion Controller ist der Einsatz von IP naheliegend:

- Es gibt eine Vielzahl von standardisierten Schnittstellen.
- Die Programmierung von kundenspezifischen Algorithmen in Steuerungen (IEC 61131) ist selbstverständlich.

- Meist kann der IPC selbst als Entwicklungssystem genutzt werden.

Im Folgenden werden zwei unterschiedliche Ansätze betrachtet:

- Einsatz von Standard IP in Einzelachsen und
- Einsatz von maschinenspezifischen IP bei Mehrachsensystemen

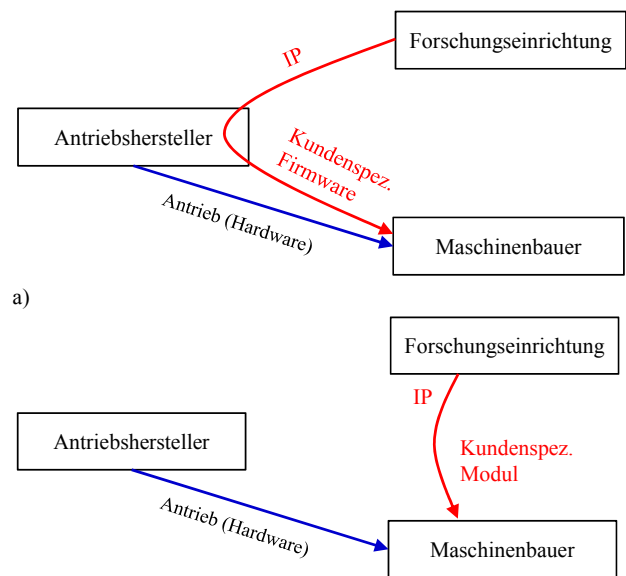


Bild 6: Informationsfluss bei der Implementierung von kundenspezifischen Algorithmen

- a) Heutige Vorgehensweise
- b) Vorgehensweise mit Motor Control IP

3.1 Universelles Motor Control IP

In den letzten Jahren forschen und entwickeln viele Forschungseinrichtungen an der sensorlosen Regelung von Synchron- und Asynchronmaschinen oder anderen Projekten. Durch die fehlenden offenen und standardisierten Schnittstellen muss von den Hochschulen immer wieder die erforderliche spezielle Hardware erstellt werden [7]. Dadurch und zusätzlich durch die Realisierung in handoptimierter Festkommaarithmetik geht viel Zeit verloren. Wenn das Projekt erfolgreich beendet wird, ist es nicht selbstverständlich, dass das erarbeitete Know-How genutzt wird. Nur wenn ein Antriebshersteller Interesse zeigt, werden die Algorithmen von den Entwicklern des Herstellers im Antrieb zur Verfügung gestellt. Wenn ein Endkunde Interesse zeigt, muss der Hersteller trotzdem mit ins Boot geholt werden. Das gelingt erfahrungsgemäß nur großen Endkunden, **Bild 6**.

Mit dem neuen Ansatz kann eine Universität z.B. ein sensorless Motor Control IP erstellen, dessen Algorithmen von der CPU / FPU des Motion Controllers ausgeführt werden. Eine Beteiligung des Antriebsherstellers ist durch die standardisierten, offenen Schnittstellen nicht mehr notwendig. Das senkt die Kosten und ermöglicht innovativere Ansätze. In diesem Beispiel – sensorlose, permanent erregte Synchronmaschine – bekommt das Motor Control IP folgende

1. Eingangssignale:
 - Spannungsraumzeiger u_a, u_b
 - Stromraumzeiger i_a, i_b
2. Ausgangssignale:
 - Trägersignal u_{inject}
 - Rotorlagewinkel φ
3. Parameter:
 - Längsinduktivität L_d
 - Querinduktivität L_q
 - Wicklungswiderstand R
 - Drehmomentkonstante K_T

Da die Signale Floating-Point nutzen können, ist keine zusätzliche Skalierung erforderlich. Stattdessen können Si Einheiten genutzt werden, **Bild 7**.

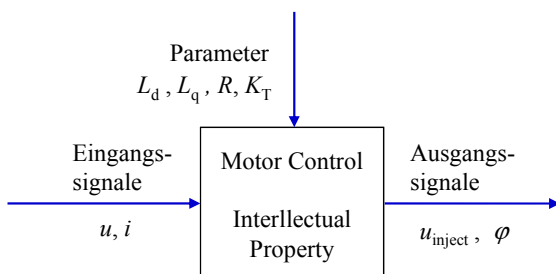


Bild 7: Motor Control IP (sensorlose, permanent erregte Synchronmaschine) mit offenen Schnittstellen und den zugehörigen Signalen (Fließkomma)

3.2 Maschinenspezifisches Motion Control IP

Ein weiterer Ansatz ist es, dem Maschinenhersteller IP zu ermöglichen. Dabei kann es sich um spezielle Regelungstechnik für Mehrachssysteme oder eine maschinenspezifische Verkopplung von mehreren Achsen handeln. Bei der Herstellung von Spritzgussmaschinen gehört es beispielsweise zum Kern-Know-How wie die Achse zum Schließen der Form, die Temperatur des Kunststoffes und der Einspritzvorgang zusammenarbeiten. Der Maschinenhersteller möchte dieses Know-How weder dem Antriebslieferanten noch dem Endkunden offenlegen.

Ebenso kann es sich um eine spezielle Linearisierung eines Kniegelenks handeln, oder um die Verkopplung der Achsen eines speziellen Roboters.

Matlab-Simulink von the MathWorks ist „das“ Standard Tool zur Simulation von Regelkreisen. In fast jeder Entwicklungsabteilung findet sich mindestens eine Lizenz. Trotz der hohen Verbreitung in den Entwicklungsgruppen hat sich Matlab-Simulink innerhalb der Maschinensteuerungen nicht etabliert. Dafür gibt es mehrere Gründe:

- Die Lizenz ist sehr teuer.
- Viele Maschinenbauer erwarten langfristige Verfügbarkeit und Kompatibilität der verwendeten Komponenten.
- Erst seit einigen Jahren basieren die Steuerungen auf IPC Technologie.
- Endkundenspezifische Änderungen sind schwierig.

3.3 Implementierung von IP

Der einfachste und bekannteste Weg mit Matlab-Simulink entwickelte Technologien in einer Maschine zu nutzen, ist der C-Codegenerator. Dieser generiert aus einem Matlab MDL-File C-Quelltext, der von einem Standard C-Compiler verarbeitet werden kann. Natürlich müssen die Eingangssignale (Regelgrößen und Führungsgrößen) und die Ausgangssignale (Stellgrößen und ev. Zustandsgrößen) noch hardware-spezifisch verschaltet werden. Hierbei ist auch darauf zu achten, dass Kenngrößen von Reglern weiterhin parametrierbar sind und nicht als Konstanten in den Quelltext integriert werden. Zum Beispiel ist eine Anpassung der Proportionalverstärkung eines Drehzahlreglers notwendig, wenn sich das Trägheitsmoment von Motor und/oder Last durch mechanische Umbauarbeiten ändert.

Wichtig ist, dass man mit Matlab-Simulink den IP-Block zwar erstellt, innerhalb der Maschinensteuerung aber weder eine Matlab Lizenz noch spezielle Kenntnisse erforderlich sind. Der Maschinennutzer sieht eine Funktionalität, die sich in die Maschinesteuerung, z.B. TwinCAT einfügt. Das Einbinden hat mit den Tools der Steuerung zu erfolgen. Das Einbinden eines IP-Moduls mit automatisch generiertem C-Quelltext eignet sich für abgekapselte, eher kleinere Aufgaben, da der C-Quelltext meist sehr schlecht les- und wartbar ist. Bei der Inbetriebnahme können von den Steuerungswerkzeugen nur die Zustandsgrößen dargestellt werden.

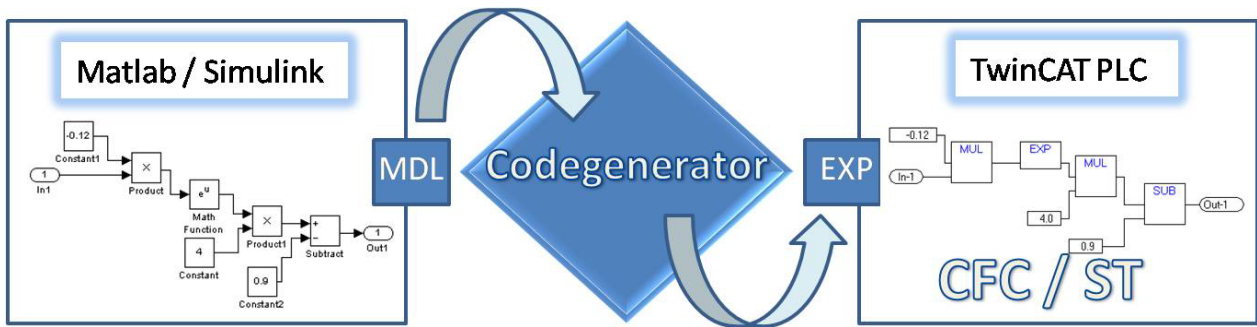


Bild 8: Matlab Simulink nach Continuous Function Chart (CFC) Compiler [8]

Ein anderer, zweistufiger Ansatz wird in [8] gezeigt. Der Schwerpunkt liegt in der Nutzung von Matlab Simulink zur Entwicklung von komplexer Steuerungs- und Regelungstechnik mit der Möglichkeit der Simulation der Maschine. In einem ersten Schritt wird die in Matlab Simulink beschriebene Regelungsarchitektur unter Wahrung der Struktur und der Reglerparameter automatisch in einen steuerungsspezifischen Quelltext Cross-Kompiliert. Gewählt wurde die Programmiersprache Continuous Function Chart (CFC), die zwar nicht in IEC 61131-3 enthalten ist, aber von den marktüblichen Herstellern unterstützt wird. Dieser Quelltext kann bei Bedarf von den Inbetriebnahmingenieuren bearbeitet und geändert werden, **Bild 8**.

In einem zweiten Schritt wird der Steuerungsquelltext für den Prozessor der Steuerung kompiliert. Die Inbetriebnahmewerkzeuge der Steuerung können uneingeschränkt genutzt werden. Matlab Simulink Lizenzen und Kenntnisse sind nicht erforderlich. Für den Endkunden muss die maschinenspezifische Struktur nicht offengelegt werden.

Auch wenn im Laufe der Zeit kleinere Änderungen gefordert werden, ist es nicht unbedingt notwendig, diese in der Entwicklungsabteilung mit Matlab Simulink zu realisieren, da die strukturkonforme Darstellung eine Wiedereinarbeitung ermöglicht bzw. erleichtert, **Bild 9**.

4 Zusammenfassung

Im vorliegenden Beitrag wird ein Konzept vorgestellt, bei dem die Regelkreise eines Servoantriebes über EtherCAT mit XFC-Technologie und Distributed-Clocks zentral geschlossen werden. Die Vorteile dieses EtherCAT-/FPGA-Servo-Konzeptes sind:

- Durch die digitale Signalverarbeitung in einem FPGA werden die Vorteile von analoger und digitaler Regelungstechnik kombiniert.
- Die Rückführung der Motorlage und/oder Last-Lagegeber mit Resolver oder Encoder kann modular über eine bzw. mehrere separate „EtherCAT-Klemmen“ realisiert werden.
- Als IGBT-Leistungsendstufe genügt ein „einfacher Frequenzumrichter“ mit schneller EtherCAT-Anschaltung auf FPGA-Basis.
- Es besteht keine Einschränkung der Regelkreisbandbreite bei Schaltfrequenzen bis 8 kHz $\rightarrow T_a = 62,5 \mu s$.
- Kundenspezifische Regelstrukturen können problemlos z. B. in einer IEC 61131-Umgebung mit Floating-Point-Algorithmen erstellt werden.

- Durch die offene Architektur wird der Einsatz vom Motion-Control-IP des Maschinenherstellers bzw. dritter Anbieter möglich, was Innovationen begünstigt.

Level		
3	Entwicklungsabteilung Matlab Simulink	Simulation der Maschine
	↓ MDL to CFC Compiler	
2	Inbetriebnahme- und Serviceabteilung TwinCAT Entwicklung	Test der Maschine
	↓ CFC to Binary Compiler	
1	Maschinennutzer TwinCAT Runtime	Betrieb der Maschine

Bild 9: Durch zweistufiges Kompilieren wird Matlab Simulink nur in der Entwicklungsabteilung benötigt.

5 Literatur

- [1] G. Pritschow, C. Kramer, G. Willuweit: Architektur eines Offenen Antriebreglers, wt Werkstattstechnik 2003 Heft 5, Springer Verlag.
- [2] www.ethercat.org
- [3] Jens Onno Krahn: Motion Control mit FPGA und Echtzeit-Ethernet, drives & motion special: Ethernet für Motion Control, GIT Verlag, Darmstadt, April 2008.
- [4] www.altera.com, www.ebv.com
- [5] Jens Onno Krahn, Christoph Klarenbach: FPGA based Field Oriented Current Controller for High Performance Servo Drives, PCIM Power Conversion Intelligent Motion, Nürnberg, Mai 2008.
- [6] www.beckhoff.com
- [7] C. Brecher, M. Merz, C. Wenzel: Optimisation of Servo Control for Highly Dynamic Axes for Ultra Precision Freeform Machining, Proceedings of the PCIM Europe Conference. Nürnberg, 22.-24. Mai 2007.
- [8] G. Bayrak, A. Wannagat, B. Vogel-Heuser: Automatische Modelltransformation zwischen Matlab/Simulink und der Steuerung. SPS/IPC DRIVES 2007, Seite 509.