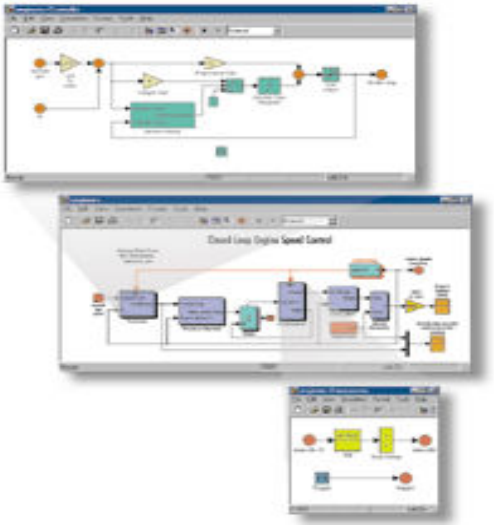
Modellbasierte Softwareentwicklung

Kunde	Wärtsilä
Ziel-System	Embedded System (Motorola MPC561)
Programmier-Sprache	ANSI-C, Matlab-Simulink
Technologien	Modell basierte Softwareentwicklung
Speziell	Innerhalb kürzester Zeit einen Regler zu entwerfen, simulieren, verifizieren und anschliesend automatisch davon C-Code generieren zu lassen.



Quelle: TheMathworks

Einleitung

Bei der Entwicklung von Software für eingebettete Systeme sieht sich der Softwareentwickler einer Reihe verschiedener Herausforderungen gegenüber. Neben den üblichen engen Zeitvorgaben, müssen schon während der Designphase Vorhersagen getroffen werden, wie sich ein System in den unterschiedlichsten Fällen verhalten wird. Mit konventionellen Entwurfs-, Test- und Implementierungsverfahren muss allerdings oft bis in späte Entwicklungsphasen hinein abgewartet werden, ob die Software wirklich alle in sie gesetzten Erwartungen erfüllt - meist ist dies erst dann der Fall, wenn die Software auf der letztendlich vorgesehenen Zielhardware eingesetzt wird.
Eine Alternative hierzu stellt das Model-Based-Design mit Werkzeugen von *TheMathworks* dar, das sich als Methode für den Entwurf der Software für Steuerungen und Regelungen fest etabliert hat.

Aufgabe

Sotronik erhielt den Auftrag, einen neuen Drehzahlregler für einen Grossdieselmotor zu realisieren. Der bestehende Drehzahlregler wurde als separates, externes System betrieben, das ständig über einen CAN-Bus die errechneten Einspritzmengen an die Motorensteuerung übermittelte.

Neu sollten diese Einspritzmengen, unter Berücksichtigung der Geschwindigkeitsvorgaben, direkt innerhalb der Motorensteuerung berechnet werden. Die grosse Herausforderung hierbei bestand darin, dass die Motorensteuerung aus mehreren verteilten Modulen besteht, von dem jedes Einzelne die Kontrolle über jeweils einen Zylinder übernimmt (maximal 14Zylinder). Der neu zu entwickelnde Drehzahlregler musste somit als verteilter Regler realisiert werden, bei dem jede Einheit die Einspritzmenge der Vorgängereinheit und deren Auswirkung auf die Motordrehzahl mit der eigenen Einspritzmenge korrigiert.

Zur Entwicklung dieser Software wurde *Matlab-Simulink* eingesetzt, um den Drehzahlregler modellbasierend vorab simulieren und nach der Verifikationsphase am PC automatisch den zugehörigen C-Code generieren lassen zu können.

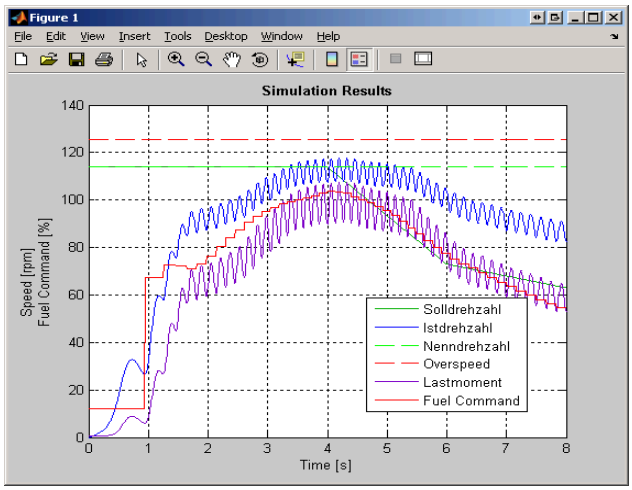
Entwurf der Reglersoftware

Basierend auf einem bestehenden Verhaltensmodell der Motordrehzahl, konnte mit dem Entwurf der Reglersoftware begonnen werden. Da auf dem Zielsystem (Motorola MPC561-Mikrocontroller) keine Fliesskommaberechnungen zugelassen wurden, mussten sämtliche Berechnungen in Festkomma-Arithmetik ausgeführt werden.
Mit Hilfe von Simulationen konnte ermittelt werden, wie sich verschiedene Festkomma-Berechnungen auf die Reglersoftware auswirken und so sichergestellt werden, dass die optimalen Datengrößen und Skalierungs-Faktoren verwendet werden. *Matlab-Simulink* mit der Erweiterung *Fixed-Point Toolbox* stellt hierzu einige nützliche Werkzeuge zur Verfügung.

Die Reglersoftware wurde in umfassenden Simulationen getestet. Hierbei konnten die Auswirkungen zahlreicher externer Einflüsse simuliert und die Ergebnisse automatisch visualisiert werden, wodurch Designfehler frühzeitig aufgedeckt wurden.

Nach der Behebung der Fehler und erneuter Simulationen war der Erfolg der getroffenen Maßnahmen sofort sichtbar, was insbesondere bei der Reglerdimensionierung und dem Parametertuning sehr hilfreich war.

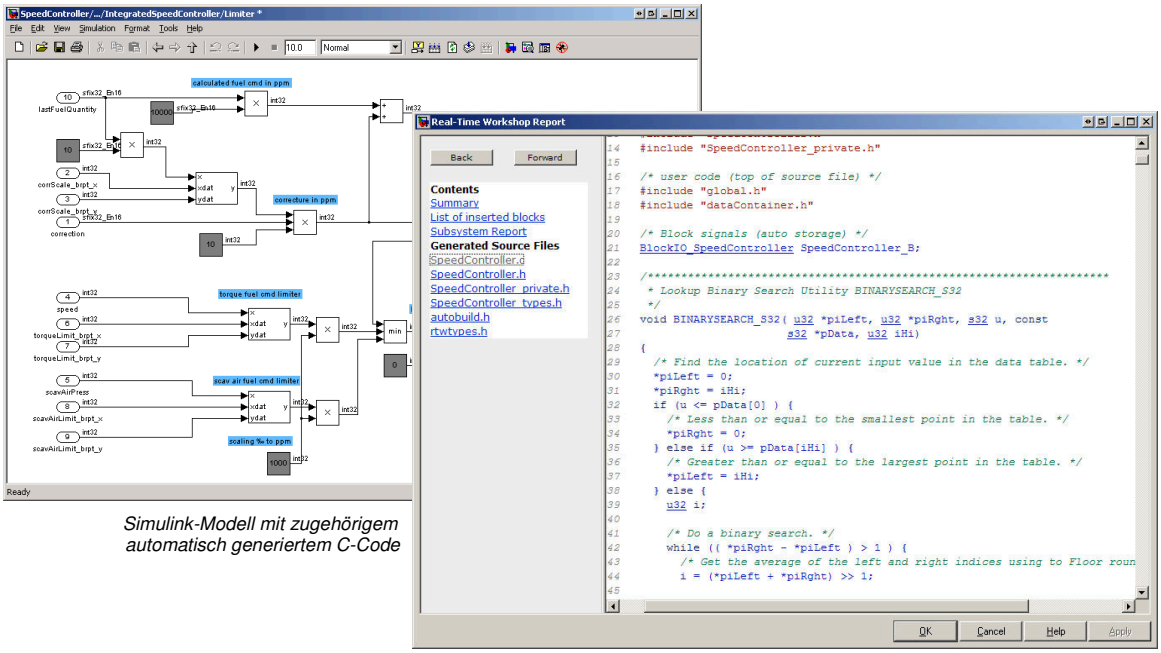
Als die Simulationen schließlich zeigten, dass sämtliche Anforderungen erfüllt waren, konnte aus dem Regler-Modell automatisch C-Code generiert und dieser in die bereits bestehende Motorensteuerung integriert und kompiliert werden.



Ausschnitt aus Simulationsresultaten

C-Code-Generierung

Mit *Matlab-RealTimeWorkshop-EmbeddedCoder* kann aus einem Modell automatisch ANSI-C-Code für die abschließende hardwarespezifische Implementierung von Embedded Software erzeugt werden. Während seiner Erzeugung wird dieser Programmcode automatisch auf höchste Geschwindigkeit und Speichereffizienz optimiert (teilweise abhängig von der Konfiguration des Entwicklers). Die automatische Codegenerierung aus einem getesteten Modell vermeidet Fehler, die bei einer manuellen Übersetzung einer Software-Spezifikation in Programmcode eingeschleppt werden können und vermeidet den Zeitaufwand für die Codierung.



Simulink-Modell mit zugehörigem automatisch generiertem C-Code

Nutzen für die Software Entwicklung

Durch den Einsatz von *Matlab-Simulink* konnten die nachfolgenden, wesentlichen Vorteile erzielt werden:

- Verständliches System und weniger Dokumentationsaufwand durch die Verwendung von graphischen Modellen.
- Kürzere Entwicklungszeit und geringere Fehleranfälligkeit gegenüber manueller Codierung durch die automatische Codegenerierung nach vorangegangener Simulation am PC.
- Geringe Einarbeitungszeit durch die Nutzung vorgefertigter Bibliotheken und der leicht verständlichen, graphischen Oberfläche von *Matlab-Simulink*.