

Integration TRACE32-Simulink®

Es ist heute üblich, die Simulation und Verifikation möglicher Lösungen vorab durchzuführen. Gerade in der Regelungstechnik haben aus diesem Grund MATLAB® und Simulink® als Entwicklungssoftware Einzug gehalten. Wer seine Regelstrecke im Simulator testet und so mögliche Auswirkungen veränderter Parameter schnell und einfach voraussagen kann, spart viel Zeit und Aufwand.

Aber wie geht es weiter, wenn man seinen optimalen Regelalgorithmus in der Simulation gefunden hat? Wie bringt man diese Lösung auf die echte Steuerungshardware?

Simulink bietet dafür die Möglichkeit der automatischen Codegenerierung. Aber kann man sicher sein, dass das Programm auf der Steuerungshardware genauso reagiert wie die Simulation? Vertrauen ist gut, Kontrolle ist besser! Und oft ist ein Nachweis zwingend vorgeschrieben.

Verifikationsansatz der TU München

Der Lehrstuhl für Flugsystemdynamik der TU München hat in seiner Entwicklung eines Flight-Control-Systems für eine Diamond DA42 (siehe Bild 3) einen effizienten Lösungsweg aufgezeigt.

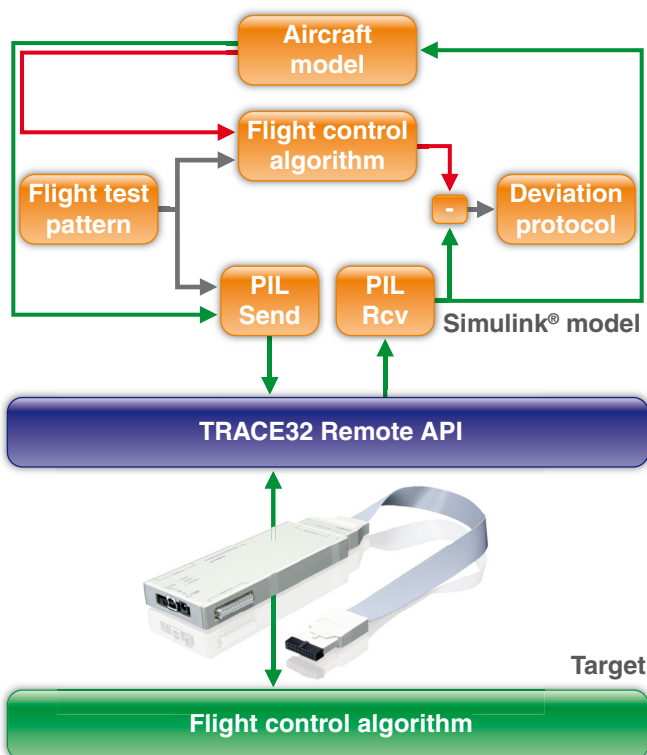


Bild 1: Das reale Reglerverhalten (grüner Pfad) und das simulierte Reglerverhalten (roter Pfad) werden abgeglichen.

Nachdem die Regelungsalgorithmen mit Simulink erstellt und funktional getestet waren, wurde mittels des *Embedded Coders* aus den Regelungsblöcken der entsprechende Programmcode für den Prozessor der Steuerungshardware erzeugt. Unter Verwendung eines TRACE32 Debuggers hat man den generierten Code auf die Steuerungshardware geladen und dort ebenfalls funktional getestet.

Um die Abweichung zwischen simuliertem Reglerverhalten (roter Pfad) und realem Reglerverhalten (grüner Pfad) zu ermitteln, vor allem aber um die numerische Genauigkeit der Steuerungshardware zu bestätigen, wählte man eine Processor-In-the-Loop-Simulation (PIL) (siehe Bild 1). Die PIL-Simulation basiert im Wesentlichen auf den eigens entwickelten Simulink Blöcken „PIL Send“ und „PIL Receive“. Diese wurden entworfen, um die Kommunikation zwischen Simulink und der *TRACE32 Remote API* zu realisieren.

In jedem Durchlauf führt der Flight-Control-Algorithmus auf der Zielhardware einen einzelnen Berechnungsschritt der zeitdiskret arbeitenden Flugregelung durch. Das Simulink Modell stellt die dazu benötigten Eingangsparameter bereit. Die berechneten Stellgrößen werden an das Simulink Modell zurückgeliefert und speisen dort das *Aircraft Model*. In einem parallelen Berechnungszweig ermittelt der simulierte Flight-Control-Algorithmus dieselben Stellgrößen. Durch Differenzbildung werden beide Ergebnisse abschließend verglichen.

Die Erprobung im Teststand ergab eine absolute Abweichung von 10^{-13} . Eine hohe Übereinstimmung, die durch diesen Lösungsansatz elegant und einfach nachgewiesen wurde.

Unter www.lauterbach.com/intsimulink.html finden sich weitergehende Informationen zum Projekt des Lehrstuhls für Flugsystemdynamik der TU München.

TRACE32 Integration für Simulink®

Auf der *embedded world 2012* in Nürnberg wird Lauterbach eine noch engere Kopplung zwischen Simulink und TRACE32 vorstellen.

Für diese Kopplung nutzt Lauterbach die folgende Eigenschaft der Codegenerierung: Der für einen Simulink Block generierte Code beginnt immer mit einer Kommentarzeile, die den Namen und den Modellpfad des Blocks enthält. Die Kommentarzeilen stehen nach dem Laden des generierten Codes im TRACE32 Debugger zur Verfügung. Sie erlauben eine einfache Zuordnung von Simulink Blöcken zu Sourcecodezeilen.

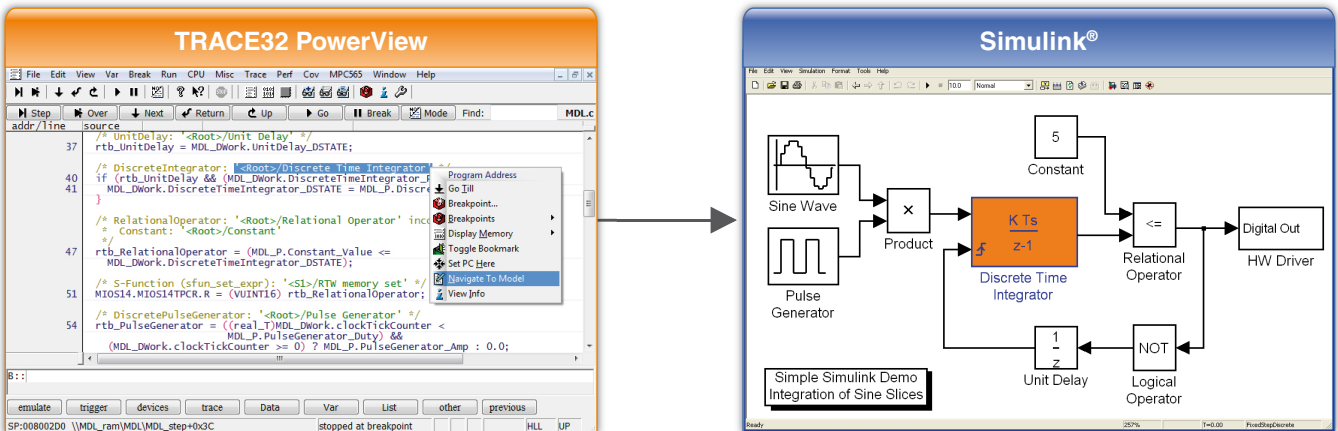


Bild 2: Der zur ausgewählten Sourcecodezeile gehörende Block wird in Simulink markiert.

Navigation von Simulink® zu TRACE32

Als so genannte *Simulink Customization Menus* werden ein globales TRACE32 Menü, sowie TRACE32 Menüs für Blöcke und Signale in Simulink integriert. Über diese Menüs lässt sich der TRACE32 Debugger aus Simulink heraus steuern. Folgende Funktionen sind möglich:

- Blockcode in TRACE32 darstellen
- TRACE32 *Variable Watch Window* für Signale öffnen
- Simulink Build in den TRACE32 Debugger laden
- Block-/Programm-Breakpoints in Simulink setzen und verwalten
- Signal-/Daten-Breakpoints setzen und verwalten
- Programmausführung auf der Steuerungshardware starten und stoppen

Navigation von TRACE32 zu Simulink®

Aus dem TRACE32 Debugger kann für eine ausgewählte Stelle des Quellcodes der dazugehörige Block in Simulink markiert werden (siehe Bild 2).

Ausblick

Sobald die *Simulink Release 2012a* verfügbar ist, wird Lauterbach die verbesserte Funktionalität der Simulink *rtiostream API* dazu nutzen, eine PIL-Simulation, Data-Logging sowie Parameter-Tuning zu integrieren.

MATLAB® und Simulink® sind eingetragene Warenzeichen von The MathWorks, Inc.



Bild 3: Erprobungsflugzeug Diamond DA42 (Quelle: www.diamond-air.at).