

Echtzeitsimulation physikalischer Systeme mit Simscape

Von [Steve Miller](#) und [Jeff Wendlandt](#)

Wenn man ein physikalisches System wie ein Fahrzeug, ein Flugzeug oder einen Roboter durch ein virtuelles System, das in Echtzeit simuliert wird, ersetzt, lassen sich die Kosten für das Testen von Hardware und Software erheblich verringern. Bei einer Echtzeitsimulation werden alle Ein- und Ausgaben der virtuellen Umgebung synchron mit der realen Umgebung eingelesen bzw. aktualisiert. Erreicht die Simulationszeit 5, 50 oder 500 Sekunden, dann ist in der realen Welt genauso viel Zeit vergangen. Man kann also rund um die Uhr über beliebige Zeiträume hinweg testen. Und dies außerdem unter Bedingungen, bei denen mit Material- oder sogar Personenschäden zu rechnen wäre. Zudem können solche Tests bereits vor der Verfügbarkeit von Hardware-Prototypen beginnen.

Damit man physikalische Systeme in Echtzeit simulieren kann, muss die Modellkomplexität zu den eingesetzten Solver-Typen sowie deren Einstellungen passen. Weiterhin muss geeignete Simulations-Hardware zur Verfügung stehen, die eine Echtzeitausführung unterstützt und Ergebnisse liefert, die möglichst genau denen einer PC-Simulation entsprechen. Variiert man diese Faktoren, dann ist die Simulation oft schneller, dafür aber ungenauer – oder umgekehrt. Simscape™ bietet eine Reihe von Funktionen, die die Konfiguration von Modellen für Echtzeitsimulationen erleichtern. Dieser Artikel zeigt beispielhaft, wie ein pneumatischer Stellantrieb für die Echtzeitsimulation konfiguriert wird (Abbildung 1). Die beschriebenen Schritte sind prinzipiell auf andere Hardware übertragbar.

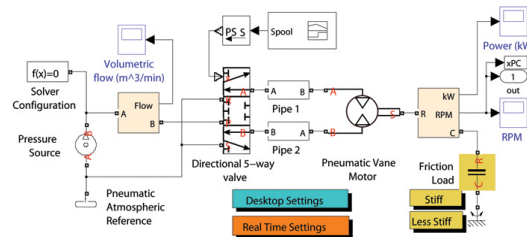


Abb. 1: Simscape-Modell eines pneumatischen Stellantriebs, das für die Echtzeitsimulation konfiguriert werden soll.

Von der Simulation im PC zur Echtzeitsimulation

Das Simscape-Modell wird in fünf Schritten für die Echtzeitsimulation vorbereitet:

1. Ermittlung von Referenzergebnissen mit einem Solver mit variabler Schrittweite.
2. Begutachtung der in der Simulation aufgetretenen Schrittweiten.
3. Wahl eines Solvers mit fester Schrittweite und Konfiguration einer Fixed-Cost Simulation.
4. Ermittlung der richtigen Balance zwischen Genauigkeit und Simulationsgeschwindigkeit.
5. Simulation des Modells auf der Echtzeit-Plattform.

1. Ermittlung von Referenzergebnissen mit einem Solver mit variabler Schrittweite

Ein Solver mit variabler Schrittweite kann seine Schrittweite verkleinern und dadurch Ereignisse sowie die Systemdynamik genau abbilden. Für Echtzeitsimulationen sind diese Solver allerdings ungeeignet, weil Echtzeit-Targets zur Berechnung eines Zeitschrittes eine bestimmte Mindestzeit benötigen. Man verwendet darum stattdessen implizite oder explizite Solver mit fester Schrittweite.

Um sicherzustellen, dass die mit Solvern mit fester Schrittweite erhaltenen Ergebnisse korrekt sind, benötigt man Referenzergebnisse. Hierzu führt man zuvor Simulationen mit einem Solver mit variabler Schrittweite durch und verringert die Fehlertoleranzen so lange, bis die Simulationsergebnisse identisch bleiben (Abbildung 2). Die empfohlenen Solver mit variabler Schrittweite für Simscape-Modelle sind ode15s und ode23t.

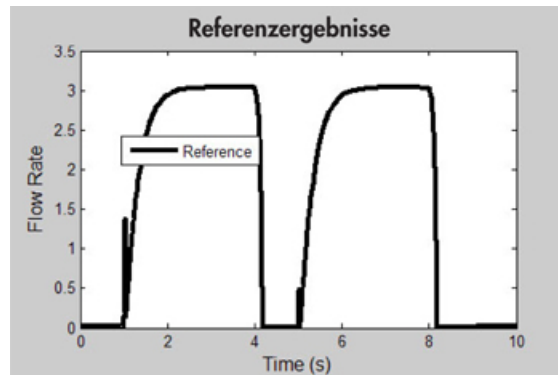


Abb. 2: Referenzergebnisse aus einer Simulation mit variabler Schrittweite.

2. Begutachtung der in der Simulation aufgetretenen Schrittweiten

Ein Solver mit variabler Schrittweite variiert die Schrittweite so, dass die vorgegebenen Fehlertoleranzen eingehalten werden und er auf Nulldurchgangsereignisse reagieren kann. Verringert der Solver etwa die Schrittweite plötzlich auf einen extrem kleinen Wert wie 10^{-15} s, bedeutet dies, dass er gerade versucht, ein Nulldurchgangsereignis wie eine Flussumkehr oder das Schließen eines Schalters korrekt zu identifizieren. Für einen Solver mit fester Schrittweite kann die Erfassung solcher Ereignisse ein Problem darstellen, wenn die Fähigkeit zur Echtzeitsimulation eine größere Schrittweite erfordert.

Mit folgenden MATLAB®-Befehlen wird ein Diagramm erzeugt, das die Veränderung der Schrittweite im Simulationsverlauf abbildet:

```
semilogy
(tout(1:end-1),diff(tout),'-*');
title
('Step Size vs. Simulation Time','F
ontSize',14,'FontWeight','bold');
xlabel
('Simulation Time(s)','FontSize',12);
ylabel
('Step Size(s)','FontSize',12);
```

Aus diesem Diagramm (Abbildung 3) wird deutlich, dass das Reibungsmodell angepasst werden muss (s. Abbildung 1, Friction Load), damit das Systemmodell echtzeitfähig wird.

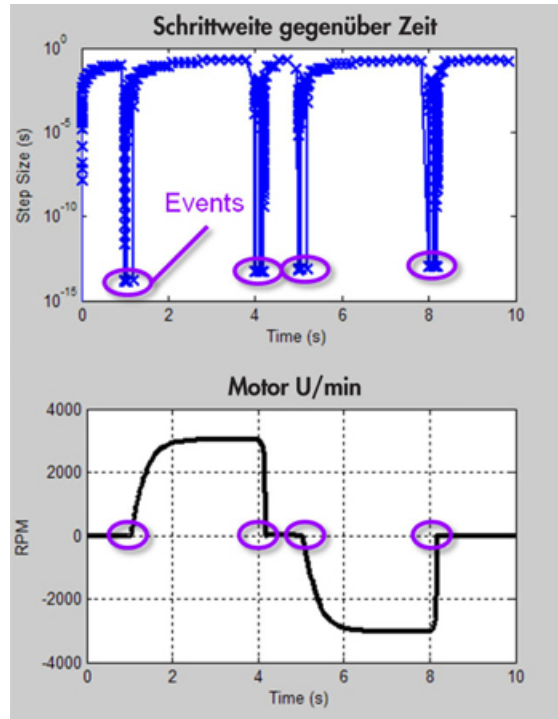


Abb. 3: Oben: Während der Simulation beobachtete Schrittweiten. Unten: Diagramm der Motorgeschwindigkeit. Nulldurchgangsereignisse bedeuten, dass der Motor auf 0 U/min abgebremst wird oder wieder neu anfährt. Um das Systemmodell echtzeitfähig zu machen, muss eventuell das Reibungsmodell verändert werden.

3. Wahl eines Solvers mit fester Schrittweite und Konfiguration einer Fixed-Cost Simulation

Ein entsprechender Solver mit fester Schrittweite muss leistungsfähig sein, aber auch bei einer für die Echtzeitsimulation geeigneten Schrittweite ausreichend genaue Ergebnisse liefern. Darum wird das Modell bei verschiedenen Schrittweiten sowohl mit einem impliziten als auch mit einem expliziten Solver mit fester Schrittweite simuliert und die Ergebnisse verglichen (Abbildung 4). Der implizite Solver liefert in diesem Beispiel die genaueren Ergebnisse.

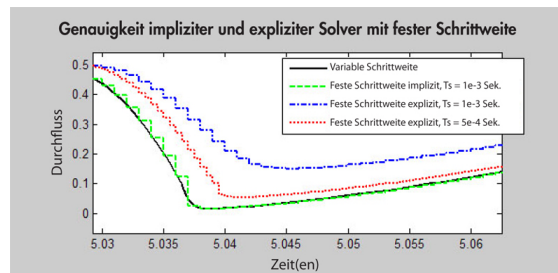


Abb. 4: Vergleich der Simulationsergebnisse eines Solvers mit variabler Schrittweite mit denen eines expliziten und eines impliziten Solvers mit fester Schrittweite. Für den expliziten Solver sind kleinere Zeitschritte erforderlich, um die gleiche Genauigkeit zu erzielen wie der implizite Solver.

Wenn die Ausführungszeit länger wird als die Abtastzeit, entstehen Überläufe (Abbildung 5). Um dies zu vermeiden, wird hier eine Fixed-Cost Simulation eingerichtet, die die Zahl der pro Zeitschritt ausgeführten Iterationen beschränkt.

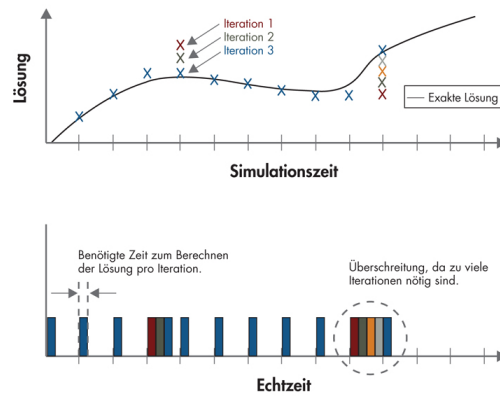


Abb. 5: Mechanismus eines Überlaufs durch Zeitüberschreitung. Oben: Genaue Lösungen erfordern manchmal mehrere Iterationen. Unten: Eine Zeitüberschreitung tritt dann auf, wenn nicht alle Iterationen innerhalb der für die Simulation festgelegten Schrittweite berechnet werden können.

Wie Abbildung 5 zeigt, müssen sowohl bei impliziten als auch bei expliziten Solvern häufig mehrere Iterationen für jedes physikalische Simscape-Netzwerk durchgeführt werden. Die Zahl dieser Iterationen kann man begrenzen, indem man die Checkbox „Use fixed-cost runtime consistency iterations“ anklickt und die gewünschte Zahl nichtlinearer Iterationen im Solver Configuration-Block festlegt. Das beste Kosten/Genauigkeits-Verhältnis erzielt man in der Regel mit zwei oder drei Iterationen.

Simscape erlaubt lokale Solver. Man kann also die steifen Anteile eines Modells durch implizite Solver mit fester Schrittweite lösen, während der Rest des Modells von expliziten Solvern mit fester Schrittweite gelöst wird (Abbildung 6). Auf diese Weise minimiert man die Zahl der je Zeitschritt durchzuführenden Berechnungen und erhöht die Wahrscheinlichkeit, dass das Modell echtzeitfähig ist.

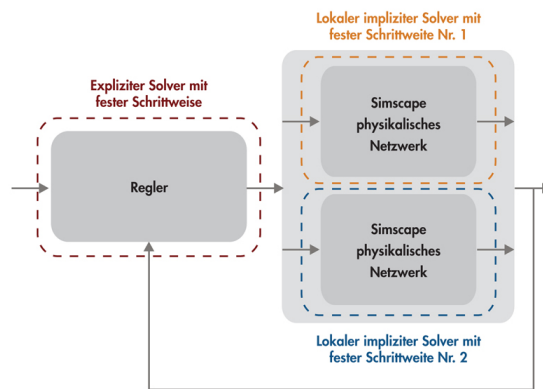


Abb. 6: Ein Modell kann verschiedene Solver nebeneinander enthalten. Bei solchen lokalen Solvern berechnen implizite Solver die steifen Anteile eines Modells und explizite Solver den übrigen Teil. Dies minimiert die Ausführungszeit bei dennoch hoher Genauigkeit.

4. Die richtige Balance zwischen Genauigkeit und Geschwindigkeit

Während jedes Zeitschritts liest das Echtzeitsystem die Eingaben, berechnet die Simulationsergebnisse für den folgenden Zeitschritt und schreibt die Ausgaben. Wenn es dazu nicht den gesamten Zeitschritt braucht, bleibt der Prozessor während der restlichen Zeit inaktiv.

Die Herausforderung besteht also in der Bestimmung von Einstellungen, die genaue Ergebnisse liefern und gleichzeitig eine Echtzeitsimulation erlauben (Abbildung 7). Dabei muss man grundsätzlich Kompromisse zwischen Genauigkeit und Geschwindigkeit machen. Durch Auswahl eines rechenintensiveren Solvers, Erhöhung der Anzahl nichtlinearer Iterationen oder Verkleinerung der Schrittweite erhöht man die Genauigkeit und verringert den Leerlauf. Gleichzeitig wächst aber das Risiko, dass die Simulation nicht

mehr in Echtzeit ausgeführt werden kann. Durch jeweils gegenteilige Einstellungen erhöht man den Leerlaufanteil, verringert aber die Genauigkeit.

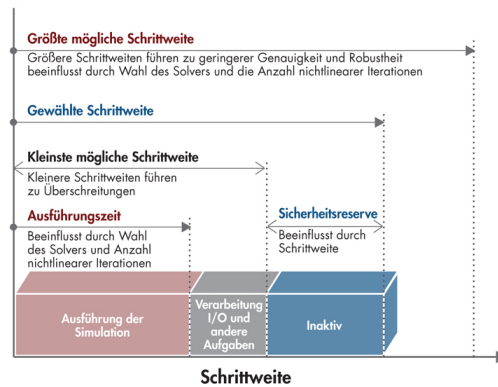


Abb. 7: Die Auswahl des Solvertyps, der Zahl nichtlinearer Iterationen sowie der Schrittweite ist immer ein Kompromiss. Für jedes Modell müssen diese Faktoren so gewählt werden, dass sie höchstmögliche Genauigkeit und Robustheit garantieren, aber auch ausreichend Leerlauf als Sicherheitsreserve lassen.

Für das vorliegende Simscape-Modell erhält man mit dem lokalen Solver Backward Euler und einer Beschränkung auf zwei Iterationen genaue Simulationsergebnisse bei akzeptabler Geschwindigkeit.

5. Simulation des Modells auf der Echtzeit-Plattform

Das vorgestellte Modell lief auf einem 700 MHz-Prozessor in Echtzeit und benötigte für die erforderlichen Berechnungen nur 62 % der zur Verfügung stehenden Schrittweite. Die Ergebnisse der Echtzeitsimulation waren identisch mit denen, die mit denselben Solver-Einstellungen auf einem PC erhalten wurden. Sie lagen außerdem sehr nahe an den Referenzergebnissen des Solvers mit variabler Schrittweite (Abbildung 8).

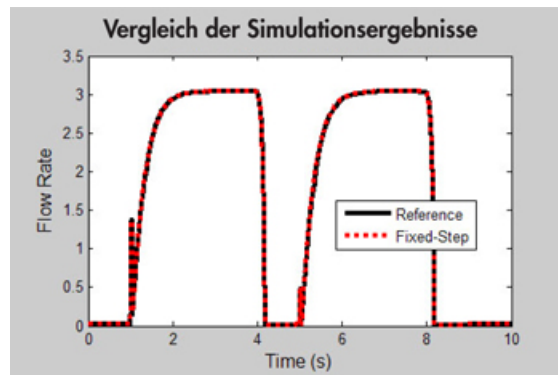


Abb. 8: Vergleich der Simulationsergebnisse für den Solver mit variabler Schrittweite mit denen des für die Echtzeitsimulation konfigurierten Modells.

Erweiterung dieses Ansatzes

Der in diesem Artikel vorgestellte Ansatz ist nicht auf einen bestimmten Modelltyp beschränkt. Er wurde auf über 30 Modelle aus verschiedenen Anwendungsbereichen und physikalischen Domänen angewandt. Diese Modelle enthielten hydraulische, elektrische, mechanische, pneumatische oder thermische Elemente und repräsentierten unter anderem hydromechanische Servoventile, bürstenlose Gleichstrommotoren, hydraulische Pipelines mit Druckstoßeffekten oder pneumatische Antriebssysteme mit Stick-Slip-Effekt. Alle Modelle konnten mit xPC Target™ auf einem Intel® Core 2 Duo E6700 (2,66 GHz) ausgeführt werden. Der höchste Anteil für die reine

Ausführung der Simulation an einem Zeitschritt betrug weniger als 18 %, was eine komfortable Sicherheitsreserve für die I/O-Verarbeitung und alle weiteren Aufgaben lässt. Im Mittel beanspruchte die Ausführung der Simulation 3,9 % eines Zeitschritts, der niedrigste Wert lag bei 0,0006 %.

Eingesetzte Produkte

- [MATLAB®](#)
- [Simulink®](#)
- [Simscape™](#)
- [Simulink Coder™](#)
- [xPC Target™](#)

Weitere Informationen

- [Video: Konfiguration von Simscape-Modellen für die Echtzeitsimulation](#)
- [Interaktives Modell: Echtzeitsimulation eines hybrid-elektrischen Fahrzeugs](#)
- [Echtzeitsimulation physikalischer Systeme mit Simscape](#)

Weitere Artikel und Newsletter-Abonnement unter www.mathworks.de/newsletters.