

ENTWICKLUNG VON ALGORITHMEN ZUR SIMULATORKOPPLUNG UNTER VERWENDUNG DER FMI-SCHNITTSTELLE FÜR ANWENDUNGEN IN DER GEBÄUDESIMULATION

C. Clauß, K. Majetta und R. Meyer

Fraunhofer IIS/EAS Dresden, Zeunerstraße 38, 01069 Dresden, Germany

KURZFASSUNG

Mit der Schaffung des Functional-Mockup-Interfaces (FMI) und der Unterstützung des Exports und Imports von Functional-Mockup-Units (FMUs) durch immer mehr Tools nimmt der Modellaustausch zu. FMUs für Co-Simulation stellen Modelle einschließlich zugehöriger Simulationsalgorithmen zur Verfügung. Für die Co-Simulation mehrerer FMUs sind Masteralgorithmen erforderlich, die die einzelnen FMUs aufrufen und den Austausch von Werten (Koppelgrößen) zwischen den FMUs durchführen. Der Beitrag befasst sich mit der Entwicklung effektiver Masteralgorithmen, die auf Anwendungen in der Gebäudesimulation angepasst sind. Es werden grundlegende Masteralgorithmen erprobt und daraus Gesichtspunkte, Anforderungen und Vorschläge für die künftige Entwicklung von Masteralgorithmen abgeleitet.

ABSTRACT

By reason of the development of the FMI standard and the support of FMU export and import by an increasing number of tools, the exchange of models becomes a frequent task. FMUs for co-simulation provide models together with their simulation algorithms. For the co-simulation of several FMUs master algorithms are needed which call the FMUs as well as interchange coupling variables. In this paper some basic master algorithms are presented which are implemented, and ready to use. These algorithms are tested at two applications. Based on the test results requirements and suggestions are derived for adapting the master algorithms to applications in building simulation.

EINLEITUNG

Die Anwendung von Modellierung und Simulation zur Erforschung des Verhaltens von Systemen, zur Erprobung, zur Optimierung und zur robusten Auslegung ist unbestritten. Dafür wurde Software geschaffen, Simulatoren, umfangreiche Modellbibliotheken und Simulationsumgebungen, und dieser Prozeß hält an. Es gibt relativ universelle Ansätze, die viele physikalische Gebiete sowie Anwendungsbereiche abdecken, aber auch spezialisierte, sehr leistungsfähige Simulatoren für dedizierte Anwendungen. Die Modellbildung ist oft

ein sehr aufwendiger Vorgang, der bereits Simulation mit einbezieht.

In jüngster Vergangenheit hat die Einführung des Functional-Mockup-Interfaces (FMI) die Modellierungs- und Simulationspraxis verändert. Jetzt können Modelle zwischen allen Simulatoren, die dieses Interface unterstützen, beliebig ausgetauscht werden. Damit erhöht sich der Anwendungsbereich einmal entwickelter Modelle beträchtlich. Umfassende, beliebig detaillierte Systemmodelle können aufwandsparend und einfach in ihren „angestammten“ Umgebungen entwickelt und breit angewendet werden. Auch die Simulation im Gebäudebereich profitiert von dieser Entwicklung.

Ein zum Austausch aufbereitetes und mit dem FMI versehenes Modell, das aus differential-algebraischen Gleichungen besteht, heißt Functional-Mockup-Unit (FMU). Es enthält ein standardisiertes Beschreibungsfile (im XML-Format) und Quellcode oder üblicherweise ausführbaren Code (*.DLL/*SO) mit C-Funktionen, deren Namen und Argumente vorgeschrieben sind und die von der die FMU „importierenden“ Software aufgerufen werden. Man unterscheidet FMUs für den Modellaustausch (model exchange), die selbst keinen Lösungsalgorithmus enthalten und nach entsprechender Verknüpfung einem Lösungsverfahren für differential-algebraische Gleichungen unterworfen werden, und FMUs für Co-Simulation, die ihren eigenen Löser (Simulator) besitzen. Formal zeigt sich dieser Unterschied darin, dass FMUs für Co-Simulation ihre eigene Simulationszeit mitführen und „von außen“ aufgefordert werden müssen, Simulationsschritte auszuführen (durch Aufruf der „doStep“-Funktion). Bei FMUs für Modellaustausch gibt es keine „doStep“-Funktion. In diesem Beitrag steht die Co-Simulation, d.h. die Kopplung von mehreren FMUs für Co-Simulation, im Mittelpunkt. Dafür ist ein sogenannter Masteralgorithmus notwendig, der den Datenaustausch organisiert und die einzelnen FMUs auffordert, Zeitschritte zu rechnen, um eine Co-Simulation zu vollziehen.

Es ist zu erwarten und teilweise auch schon Praxis, dass FMUs für die verschiedenen bei der Gebäudesimulation relevanten Disziplinen (Bauphysik, Bauklimatik, technische Gebäudeausrüstung, Wetter, Nutzungsszenarien,

Gesamtregelungen, ...) erstellt, zu parametrisierbaren FMU-Bibliotheken zusammengefaßt und dann gekoppelt werden. In diesem Beitrag wird untersucht, welche Anforderungen und Vorschläge zu beachten sind, um Masteralgorithmen an die Besonderheiten von Co-Simulationen im Gebäudebereich anzupassen. Zunächst werden im folgenden Abschnitt allgemeine Masteralgorithmen beschrieben. Im dritten Abschnitt werden zwei Kopplungsbeispiele getestet, um danach aus den Ergebnissen Erweiterungen an die allgemeinen Masteralgorithmen für die genannten oder ähnlich geartete Beispiele abzuleiten.

GRUNDLEGENDE ALGORITHMEN UND TESTIMPLEMENTIERUNG

Mathematische Aufgabe der Co-Simulation

In (Petridis et al. 2015) wurde die Aufgabe der Co-Simulation mathematisch formuliert als

$$x = \sum_{i=1}^m Q_i S_i(P_i x) \quad (1)$$

Dabei bedeuten x die Gesamtheit der Variablen, die zwischen den Simulatoren ausgetauscht werden, und S_i die mit den m FMUs bereitgestellten m Simulatoren. P_i sowie Q_i sind Matrizen, die den Verbindungsgraphen, das heißt die Verknüpfung der Ein- und Ausgänge der Simulatoren, beschreiben. Bei Simulatoren ohne Ausgängen entfällt in dieser Formel der entsprechende Summand, bei Simulatoren ohne Eingängen fehlt das Argument $P_i x$.

Grundlegende Algorithmen

Ebenfalls in (Petridis et al. 2015) wurden elementare Lösungsverfahren für (1) angegeben, die für die Lösung nichtlinearer Gleichungen bekannt sind: Das Gauß-Jacobi-Verfahren

$$x^{k+1} = \sum_{i=1}^m Q_i S_i(P_i x^k) \quad (2)$$

benötigt eine Startlösung x_0 und nutzt (1) als Iterationsvorschrift, der Iterationsindex ist k . Bei diesem Verfahren können die einzelnen Simulatoren parallel abgearbeitet werden. Beim Gauß-Seidel-Verfahren (3) ist die Festlegung einer Reihenfolge $r = (r_1, \dots, r_m)$ erforderlich, in der die Simulatoren aufgerufen werden. Dabei greifen die einzelnen Simulatoren auf die Ergebnisse der aktuellen Iteration zu, falls die „Vorgänger“ bereits simuliert wurden, andernfalls auf die Ergebnisse des letzten Iterationsschrittes.

$$\xi^1 := x^k$$

$$\xi^{i+1} := Q_{r_i} S_{r_i}(P_{r_i} \xi^i) + \xi^i - Q_{r_i} Q_{r_i}^T \xi^i \quad (3)$$

$$i = 1(1)m$$

$$x^{k+1} := \xi^{m+1}$$

Dabei ist parallele Abarbeitung nur bedingt möglich. Weiterhin kann auf (1) das Newton-Verfahren (4) angewendet werden, das Ableitungsberechnungen für die Jacobimatrix benötigt und bei einer Startlösung, die „in der Nähe“ der gesuchten Lösung liegt, quadratisch konvergiert.

$$x^{k+1} = \left(\frac{\partial S}{\partial x} - I\right)^{-1} \left(\frac{\partial S}{\partial x} x^k - S x^k\right) \quad (4)$$

Zur Vereinfachung empfiehlt es sich, zunächst den Verbindungsgraphen der gekoppelten Simulatoren, der durch die Matrizen P_i und Q_i gegeben ist, zu analysieren und alle vorhandenen Zyklen (Teilmengen von FMUs, bei denen es Ausgänge gibt, die, verbunden über weitere FMUs, von sich selbst abhängen) festzustellen. Die Simulatoren sind dann entsprechend der Reihenfolge des gerichteten Verbindungsgraphen aufzurufen und die oben genannten Verfahren (2), (3) oder (4) brauchen nur noch auf die gegebenenfalls vorhandenen Zyklen angewendet werden. Bei diesem Vorgehen wird die Aufrufreihenfolge, die sich de facto bei den einzelnen Verfahren bei Vorhandensein eines gerichteten Graphen einstellen würde, a priori ausgenutzt und die Komplexität des Iterationsverfahrens reduziert.

Weiterhin wird nicht mit Zeitfunktionen $x(t)$ über dem kompletten zu simulierenden Zeitintervall $[0, T]$ gearbeitet, sondern entsprechend des FMI-Standards mit Teilintervallen $[0, T] = \cup [t_c, t_{c+1}]$, die durch Kommunikationszeitpunkte t_c begrenzt sind, an denen die Simulation unterbrochen wird und der Datenaustausch erfolgt. Die Kommunikationsschrittweite $h_c = t_{c+1} - t_c$ kann konstant vorgegeben werden oder variabel sein, gesteuert zum Beispiel durch das Verfahren der Richardson-Extrapolation (Schierz 2013).

Die Anwendungsmöglichkeit der oben genannten Verfahren kann durch die Eigenschaften der beteiligten Simulatoren (FMUs) eingeschränkt werden. So erfordern die Verfahren beim Iterieren wiederholte Simulationen des gleichen Zeitintervalls, d.h. die Simulatoren müssen zurücksetzbar sein auf einen Zustand, um von dort eine neue Simulation mit geänderten Eingängen fortzusetzen. Ob eine FMU diese hohen Anforderungen erfüllt, wird in deren Attributen festgehalten, die zu Beginn der Simulation ausgewertet werden. Masteralgorithmen müssen deshalb sowohl den Verbindungsgraphen als auch die Attribute der FMUs auswerten, um danach die Strategie der Co-Simulation festzulegen.

Testimplementierung

(Bastian et al. 2011) stellt eine Testimplementierung eines Masters für Co-Simulation vor, die inzwischen am Fraunhofer IIS EAS weiterentwickelt wurde. Diese Implementierung enthält die Grundalgorithmen (3) und (4), wobei (3) noch auf genau eine Iteration pro Zeitschritt eingeschränkt werden kann. Die oben erwähnte Reihenfolge des Aufrufs der Simulatoren muss vom Nutzer vorgegeben werden, es erfolgt noch keine automatische Auswertung des Verbindungsgraphen. In einem Konfigurationsfile werden die Anzahl von FMUs und Koppelgrößen, das zu simulierende Zeitintervall, die vorzugebende feste Kommunikationsschrittweite, die Auswahl des Verfahrens und dessen Verfahrensparameter, der Verbindungsgraph und die vom Anwender vorzugebende Abarbeitungsreihenfolge beschrieben. Das Konfigurationsfile steuert den Masteralgorithmus der Testimplementierung. Eine graphische Oberfläche unterstützt den Anwender bei der Erzeugung des Konfigurationsfiles. Insbesondere können die FMUs platziert und auch graphisch verknüpft werden. Es ist aber auch möglich, das Konfigurationsfile zu generieren und ohne die Oberfläche zu arbeiten. Aktuelle Entwicklungen an der Testimplementierung betreffen die Implementierung des Verfahrens (2) und Algorithmen zur Berechnung konsistenter Anfangswerte.

Der Master wurde an einer Vielzahl mathematischer Testbeispiele erprobt. Wie zu erwarten zeigte sich, dass bei zyklischen Abhängigkeiten Iterationsverfahren erforderlich sind und das Newtonverfahren zwar am aufwendigsten ist, aber auch die besten Konvergenzeigenschaften besitzt. Wenn das Gauß-Seidel-Verfahren ohne Iterationen konvergiert, kann die Lösungsgenauigkeit durch kleinere Kommunikationsschrittweite verbessert werden. Es galt nun, die implementierten Verfahren an Beispielen der Gebäudesimulation zu erproben und anzupassen.

ANWENDUNG FÜR GEBÄUDESIMULATION

Für die Erprobung der Kopplungsalgorithmen stehen zwei Beispiele aus der Gebäudesimulation zur Verfügung, das Modell eines real existierenden Schul- und Sportzentrums und das Modell eines hypothetischen Reihenhauses.

Beispiel Schul- und Sportzentrum Lohr

Das Schul- und Sportzentrum Lohr besteht aus dem Gebäudekomplex eines Gymnasiums und einer Mittelschule mit einer Sport- und einer Schwimmhalle. Es stehen moderne regenerative Energieträger, Speicheranlagen und ebenso moderne Regelungssysteme zur Verfügung. Um die enorm hohen Betriebskosten durch den Verbrauch an Öl und Gas zu reduzieren, wurde durch EA Systems Dresden

(EA Systems Dresden, 2015) im Rahmen eines Forschungsprojektes ein Modell des Energiemanagers als Basis für den Entwurf und die Optimierung der Regelungsanlage erstellt, das vor allem die Energieflüsse und die Verbräuche erfasst.

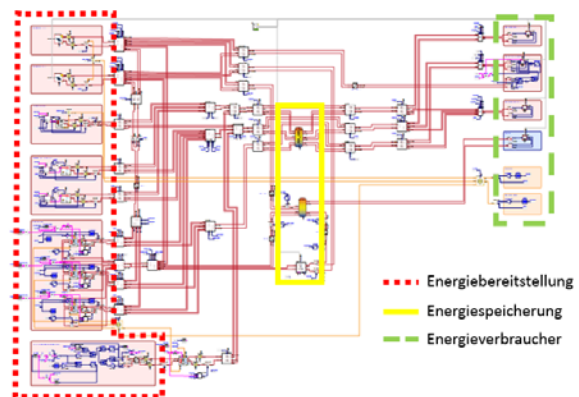


Abbildung 1: Graphische Repräsentation des Teilmodells „Anlage“ des Schul- und Sportzentrums

Aus diesem Modell wurden zwei Teilmodelle abgeleitet und zu Testzwecken als FMUs Version 1.0 mit SimulationX 3.6.5 für die Co-Simulation bereitgestellt.

Die FMU „Anlage“ enthält das Modell der Anlage der Energiezentrale des Schul- und Sportzentrums. Modelliert sind Wärmeenergiebereitstellung, -speicherung sowie Wärmeverbraucher. Dazu wurden Modelle aus der von EA Systems entwickelten GreenBuilding-Modelica-Bibliothek verwendet. Die Energiebereitstellung erfolgt durch zwei Solarthermieanlagen, ein Blockheizkraftwerk, einen Gaskessel für Zeiten der Spitzenlast, mehrere Wärmepumpen sowie eine Wärmerückgewinnungsanlage. Die gewonnene Wärmeenergie wird in drei thermischen Puffern gespeichert und in verschiedenen Last-Heizkreisen verbraucht. Die Modellierung der Lasten wird durch in Textfiles beschriebenen Lastkurven realisiert. Abbildung 1 zeigt die graphische Repräsentation des Modells. Die Modellbestandteile für Energiebereitstellung, Energiespeicherung und Energieverbrauch sind im Bild unterschiedlich eingerahmt. Die Anlage enthält auch die Angaben zum Wetter, die teilweise als Koppelgrößen an den Regler übergeben werden, wie zum Beispiel die Außentemperatur.

Die FMU „Regler“ enthält das Reglermodell. Der Regler steuert, basierend auf den aktuellen Temperaturzuständen in Wärme-, Kälte- und Eisspeicher, das An- und Abschalten der Erzeuger sowie die Volumenströme zwischen Erzeugern und Abnehmern.

Der Regler hat 46 Ausgangsgrößen. Davon sind 30 kontinuierliche Größen, wie die Vorlauftemperaturen für Heizkreise, Temperaturvorgaben an die Wärmepumpen, Vorgaben für Durchflüsse an Absorber und Abnehmer und weitere. Einer der kontinuierlichen Ausgänge ist nicht an der Anlage

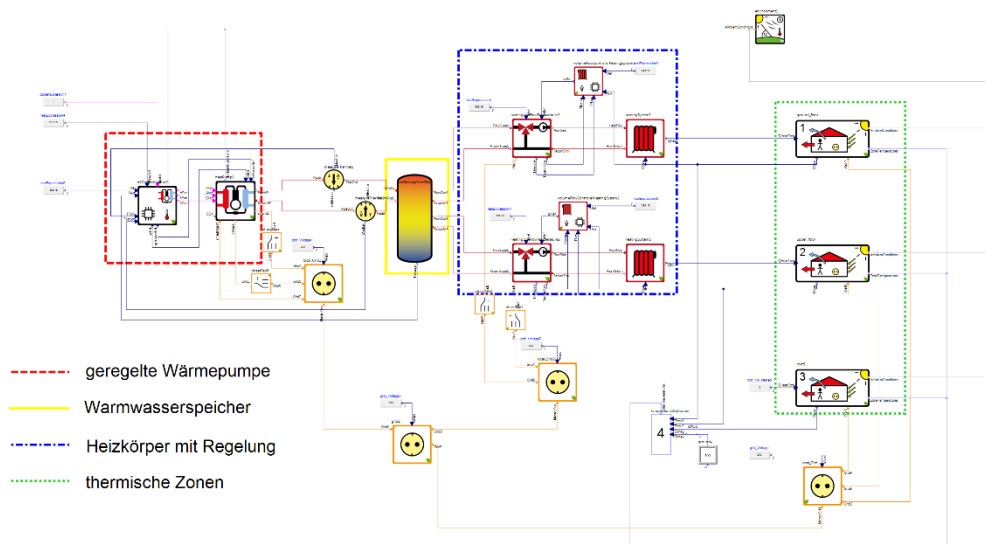


Abbildung 2: Graphische Repräsentation des Modells "Reihenhaus"

angeschlossen. Die restlichen 16 Ausgangsgrößen sind Boolesche Größen für An- und Abschaltbefehle, zum Beispiel für die Wärmepumpen. Für den Regler gibt es 82 kontinuierliche Eingangsgrößen, die den Zustand der Anlage beschreiben, wie verschiedene Temperaturen und aktuelle Verbräuche, zum Beispiel von Öl und Gas. Insgesamt gibt es 127 Koppelgrößen.

Beispiel Reihenhaus

Das Modell eines Reihenhauses wird durch drei thermische Zonen (Erdgeschoss, erstes Obergeschoss und Dachboden) beschrieben. Die Wärmeversorgung der Zonen erfolgt über Heizkörper, wobei das Dachgeschoss nicht beheizt wird. Je nach gewünschter Zonentemperatur wird der Volumenstrom des Vorlaufs der Heizungen geregelt. Die Warmwasserversorgung erfolgt über einen Speicherkessel, der durch eine Wärmepumpe versorgt wird. Abbildung 2 zeigt die graphische Repräsentation des Reihenhaus-Modells. Die einzelnen Modellbestandteile sind eingerahmt. Zum Zweck der gekoppelten Simulation wurde das Modell zerlegt und durch zwei FMUs für Co-Simulation bereitgestellt.

Die FMU „Anlage“ enthält das Anlagenteil des Modells, das aus der Wärmepumpe, dem Speicherkessel und den Heizkörpern einschließlich Rohrleitungen besteht. Diese FMU wurde aus einem Modelica-Modell, das mit SimulationX 3.7.1 modelliert vorliegt, als FMU 2.0 für Co-Simulation exportiert.

Die thermischen Zonen wurden mit dem an der TU Dresden entwickelten Tool NANDRAD (Nicolai et al. 2012) modelliert, das um die Möglichkeit des Exports von FMUs für Co-Simulation erweitert worden ist. NANDRAD beschreibt das Drei-Zonen-Gebäude detailliert als PDE und exportierte die FMU „Zonen“, die in Abbildung 2 dem Block „thermische

Zonen“ (rechts im Bild), erweitert um den Block des Wetters, entspricht. Die FMU „Zonen“ hat vier kontinuierliche Eingänge, die die Wärmeströme in die beiden beheizten Räume beschreiben. Jeder Wärmestrom teilt sich in eine Konvektions- und eine Strahlungskomponente. Die 22 Ausgänge der FMU „Zonen“, die allesamt kontinuierliche Größen sind, umfassen zehn Größen des Wetters, zwei Soll-Temperaturvorgaben für die Räume, vier Temperaturen der beheizten Zonen, wovon nur zwei an die Anlage angeschlossen sind, und weitere sechs nicht angeschlossene Größen für den Energieverbrauch, sowie die Set-Points für Kühlung und den Wärmeeintrag durch Personen. Die FMU „Zonen“ unterstützt FMI 2.0 und gestattet keine wiederholten Simulationen des gleichen Kommunikationsintervalls.

Das gekoppelte Problem hat somit insgesamt 18 Koppelvariablen, vier werden von der FMU „Anlage“ berechnet, 14 von der FMU „Zonen“.

GEKOPPELTE SIMULATION

Im Folgenden wird die Co-Simulation der beiden Beispiele mit der oben genannten Testimplementierung von Master-Algorithmen beschrieben.

Beispiel Schul- und Sportzentrum Lohr

Beide FMUs entsprechen dem FMI 1.0-Standard, der kein Zurücksetzen von Kommunikationsintervallen erlaubt. Damit kann unter den implementierten Algorithmen nur der Gauß-Seidel-Algorithmus mit genau einem Iterationsschritt pro Zeitpunkt angewendet werden. Mit konstanter Kommunikationsschrittweite von fünf Minuten ist eine Jahressimulation möglich und liefert Ergebnisse, die denen der ungekoppelten Simulation entsprechen. In Abbildung 3 und 4 sind die Jahresverläufe einiger kontinuierlicher Koppelgrößen angegeben.

Diese Jahressimulation dauert mit PC (64 Bit Windows7; Prozessor: Intel(R) Core(TM)i7-4600U

CPU@2.1 GHz, 8 GByte Arbeitsspeicher) zirka 14,6 Stunden. Davon benötigt die FMU „Regler“ 11,6 Stunden und die FMU „Anlage“ lediglich 2,9 Stunden. Die Zeit, die der Masteralgorithmus verbraucht, ist vernachlässigbar.

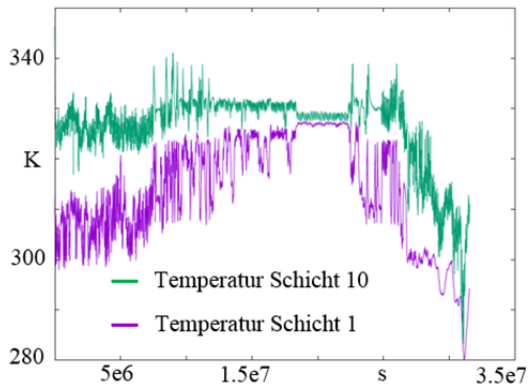


Abbildung 3: Temperaturen in zwei Schichten des Warmwasserspeichers – Ausgabe der Anlage

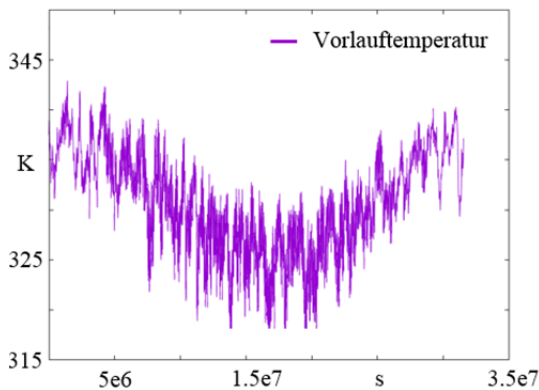


Abbildung 4: Vorlauftemperatur eines Heizkreises – Ausgabe des Reglers

Beispiel Reihenhaus

Beim Beispiel „Reihenhaus“ liegen zwar FMUs nach FMI 2.0 vor, jedoch kann die FMU „Zonen“ nicht zurückgesetzt werden, so dass auch hier nur der Gauß-Seidel-Algorithmus mit einem Iterationsschritt pro Zeitpunkt angewendet werden kann.

zeigt die Ansicht der gekoppelten FMUs in der Test-Implementierung des Masters. Auch diese Simulation liefert mit einer Kommunikationsschrittweite von einer Minute akzeptable Ergebnisse. Abbildung 6 zeigt den Temperaturverlauf einer Raumzone über einem Jahr, Abbildung 7 den konvektiven Anteil des Wärmestromes in diesen Raum, der in der Anlage zur Regelung der Raumtemperatur erzeugt wird.

Diese Jahressimulation benötigt mit oben genanntem PC 42 Minuten. Die von NANDRAD erzeugte FMU „Zonen“ beansprucht dabei mit 40,4 Minuten den Großteil der Simulationszeit. Die Anlagen-FMU rechnet nur 1,1 Minuten.

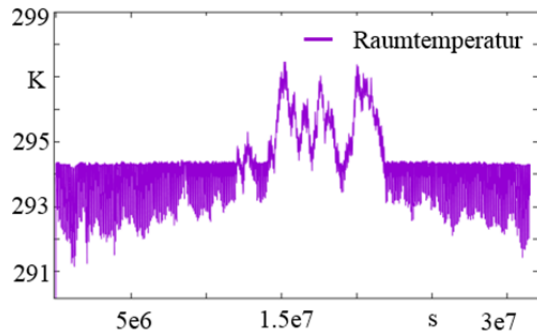


Abbildung 6: Raumtemperatur einer thermischen Zone

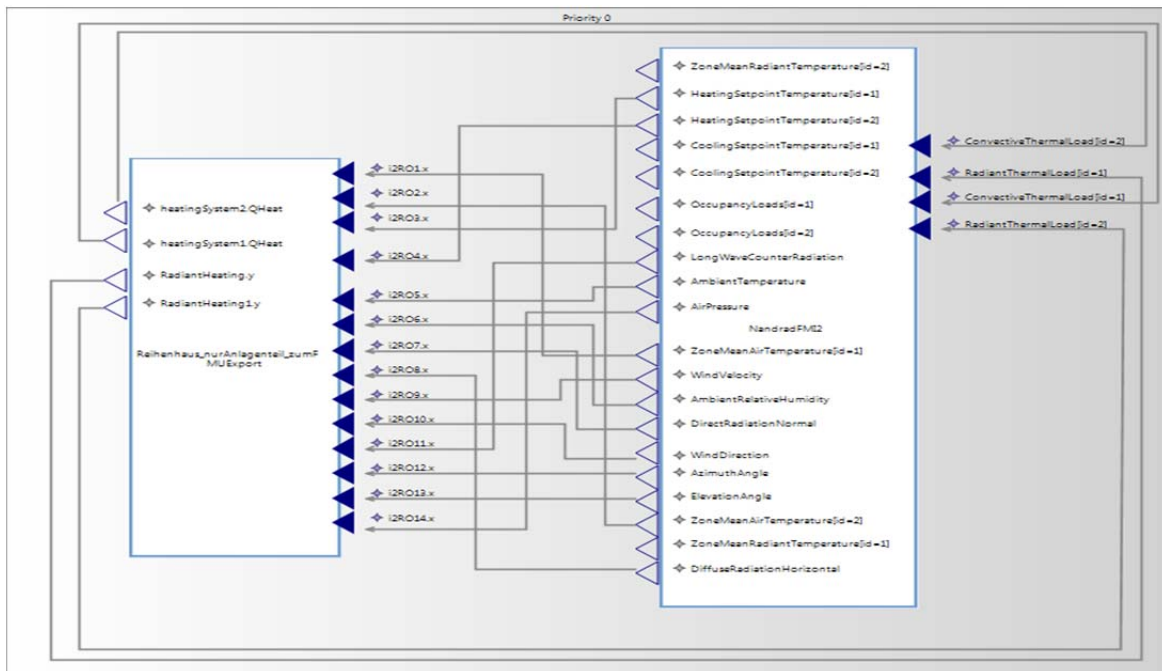


Abbildung 5: Gekoppelte FMUs "Reihenhaus"

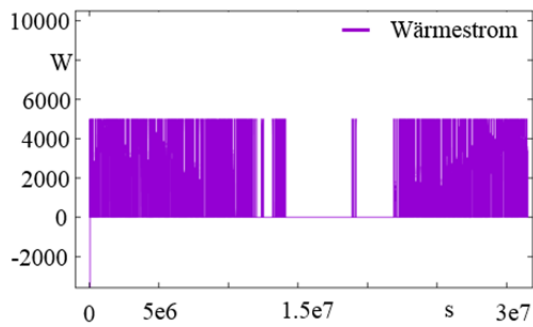


Abbildung 7: Wärmestrom - Ausgabe der Anlage

ANGEPASSTE ALGORITHMEN

Diese beiden Beispiele zeigen, dass nach Verbesserungen und Anpassungen der vorhandenen Algorithmen an derartige Co-Simulationen gesucht werden sollte. Nachfolgend werden Gesichtspunkte skizziert, die für die Entwicklung von spezialisierten Algorithmen für die oben angegebenen und ähnliche Anwendungsfälle wichtig sind.

- Zwar ist FMI 2.0 eingeführt, jedoch wird auch FMI 1.0 noch immer verwendet. Ein Masteralgorithmus sollte deshalb beide Versionen verarbeiten können.
- Mehrfache Simulationen eines Kommunikationsintervalles (FMI-Attribut `canGetAndSetFMUstate`) werden nicht immer oder sogar oft nicht unterstützt. Damit sind die iterierenden Grundalgorithmen nicht anwendbar. Es ist deshalb erforderlich, dass die Masteralgorithmen auch ohne Iterationen sinnvolle Ergebnisse liefern. Dafür eignet sich dann das Gauß-Seidel-Verfahren mit genau einer Iteration, sinnvoll gewählter Reihenfolge des Aufrufs der FMUs und Verwendung einer „kleinen“ Kommunikationsschrittweite.

- Wenn keine Iterationen über vorhandenen Zyklen möglich sind, sollte der Master wenigstens Informationen über die erreichte Genauigkeit ermitteln. Dies ist an den Stellen möglich, wo angenommene Werte mit berechneten verglichen werden können. Damit kann die angewendete Kommunikationsschrittweite beurteilt werden und Vorschläge für eine bessere Schrittweite sind möglich.
- Bei den Beispielen sind an beiden Kopplungen FMUs mit deutlichen Unterschieden in der Performance beteiligt, die in der Modellkomplexität und den numerischen Verfahren der FMUs begründet sind. Die FMUs für Raumzonen und Anlagentechnik können um Größenordnungen langsamer sein als die schnellen FMUs der Regler. Falls Iterationen möglich sind, sollten die langsamen FMUs möglichst wenig zurückgesetzt werden.
- Es wurden Untersuchungen mit unterschiedlich langen Kommunikationsschrittweiten durchgeführt. Wie zu erwarten, zeigt sich, dass mit steigender Kommunikationsschrittweite die Performance steigt und die Genauigkeit sinkt. Tabelle 1 zeigt dies am Beispiel „Reihenhaus“. In Abbildung 8 ist die Verschlechterung des Simulationsergebnisses an einer Zonentemperatur zu sehen. Bei diesem Beispiel wäre mit der Schrittweite von einer Stunde vermutlich ein Kompromiß zwischen Performance und Genauigkeit gefunden, weil eine weitere Vergrößerung der Schrittweite die erforderliche Rechenzeit kaum verbessert, aber das Ergebnis wesentlich verschlechtert. Derartige Studien können zum Finden einer geeigneten festen Kommunikationsschrittweite dienen (gegebenenfalls auch automatisiert).

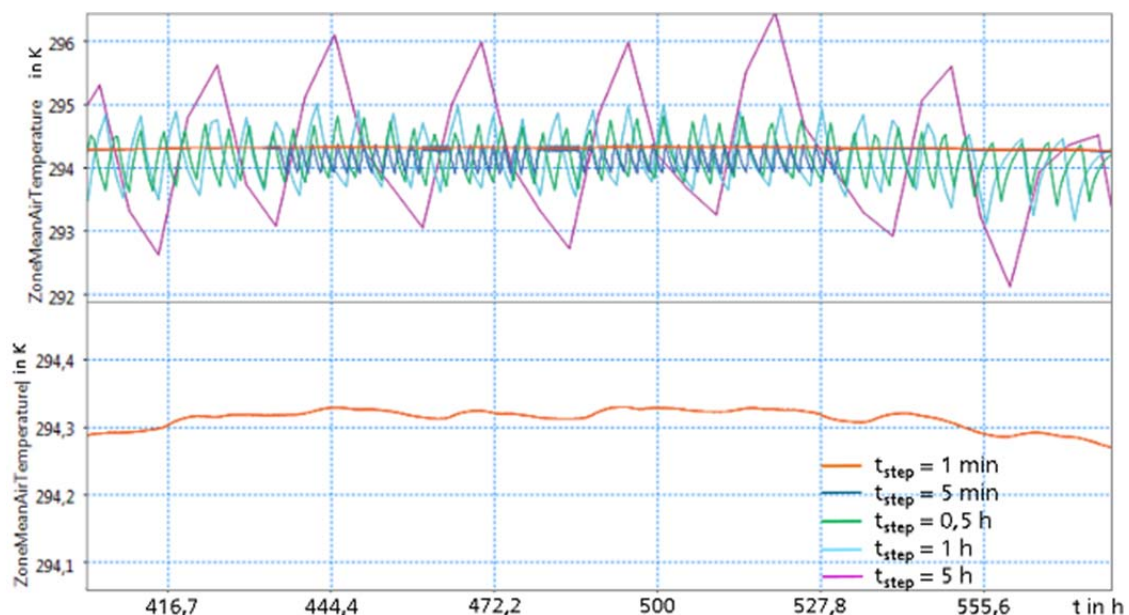


Abbildung 8: Schrittweitenstudien „Reihenhaus“

Tabelle 1: CPU-Zeiten bei unterschiedlicher Kommunikationsschrittweite für das Reihenhaus und Simulation über 31 Tage

Schrittweite	1 min	5 min	0.5 h	1 h	5h
CPU Zonen	3.2 min	3.5 min	1.9 min	1.4 min	1.3 min
CPU Anlage	5.7 s	13.8 s	8.5 s	5.6 s	2 s

- Zur Verbesserung der Genauigkeit bei der Verwendung einer großen Kommunikationsschrittweite ist es sinnvoll, die Eingänge genauer zu erfassen als mit stückweise konstanten Zeitfunktionen. Die FMUs sollten deshalb höhere Zeitableitungen an den Eingängen verarbeiten können, und der Master sollte diese übergeben oder bereitstellen, damit die Eingangsfunktionen interpoliert bzw. extrapoliert werden können. Dies betrifft auch Kopplungen ohne Zyklen.
- Bei den Beispielen gibt es digitale, z.B. Boolesche Signale für das Schalten von Anlagen. Die Schaltzeitpunkte müssen genau erfasst werden. Das legt nahe, asynchron zu arbeiten: Die FMU mit digitalen Ausgängen sollte mit kleinerer Schrittweite aufgerufen werden, um die Schaltzeitpunkte genau zu detektieren. Die FMU mit den digitalen Eingängen kann dann jeweils bis zum Schaltzeitpunkt rechnen. Dies ist insbesondere dann sinnvoll, wenn die FMU mit den digitalen Eingängen rechenintensiv ist und daher eine schlechte Performance besitzt.
- Der Masteralgorithmus besitzt noch kein Verfahren zur Berechnung konsistenter Anfangswerte. Dies ist notwendig, wenn bei FMUs Ausgänge direkt von Eingängen abhängig sind. Auch hierzu empfiehlt es sich, die oben angeführten Iterationsverfahren für den Startzeitpunkt (vor dem ersten „doStep“) anzuwenden. Dies ist auch möglich, wenn beteiligte FMUs keine Iteration erlauben. In diesem Fall können mehrfach hintereinander ausgeführte Initialisierungen angewendet werden. Das Setzen der Anfangswerte wurde implementiert. Damit kann wenigstens vermieden werden, dass zum Beispiel Temperaturvariablen mit dem default-Startwert Null (Kelvin) beginnen. Für Iterationsverfahren zur Bestimmung der Anfangswerte können die Startwerte zum Beginn der Iterationen genutzt werden.
- Bei den betrachteten Beispielen wird deutlich und es ist allgemein bei der Gebäudesimulation zu erwarten, dass sehr viele Koppelvariablen auftreten. Es ist zu vermeiden, dass diese

manuell verbunden werden müssen. Die Verbindungen könnten automatisch über Namenskonventionen erfolgen oder von weiteren in der Gebäudesimulation verwendeten Tools generiert werden.

- Es fällt auf, dass viele Ausgangsvariablen ausschliesslich von der Zeit und nicht von Eingangsgrößen abhängen. Dies sind zum Beispiel Wetterdaten und Daten zur Belegung. Es ist sinnvoll, diese als Zeitfunktionen zu kennzeichnen. Dann ist der Master in der Lage, diese Variablen aus den Iterationen a priori auszuschließen und einfach „weiterzureichen“. Damit kann die Anzahl der Iterationsvariablen deutlich reduziert werden. Es muss noch geprüft werden, ob eine derartige Kennzeichnung bereits innerhalb des vorhandenen FMI-Standards möglich ist.
- Aus gleichen Gründen ist man gut beraten, die Beschreibung ausschließlich von der Zeit abhängiger Größen in einer separaten FMU zu führen, die dann vor allen anderen FMUs gerechnet wird.

Es ist erforderlich, weitere und insbesondere umfangreichere Beispiele zu untersuchen und die genannten Ideen in der Implementierung umzusetzen. Insofern der Koppelalgorithmus auf Besonderheiten aus dem Gebäudebereich zugeschnitten wird, ist es notwendig, diese Besonderheiten zu beschreiben, (automatisch) zu erkennen und gegebenenfalls im FMI mitzuteilen, um im Masteralgorithmus darauf reagieren zu können. Dies ist eine künftige Forschungsaufgabe.

ZUSAMMENFASSUNG

Das FMI gestattet die Kopplung von Simulationstools nach einer einheitlichen Vorgehensweise, sofern die Tools FMI unterstützen. Für die Anwendungen in der Gebäudesimulation ergeben sich dadurch neue Möglichkeiten, weil bisher getrennt agierende Tools samt ihren Modellierungsmöglichkeiten und Modellbibliotheken zusammengeführt werden können.

Im Beitrag werden grundlegende Masteralgorithmen vorgestellt, die teilweise in einer Implementierung anwendungsbereit vorliegen. Bei deren Anwendung auf zwei typische Koppelsituationen aus der Gebäudesimulation zeigt sich, dass zwar brauchbare Ergebnisse erzielt werden können, jedoch Verbesserungen an den Algorithmen hinsichtlich Performance als auch Genauigkeit sinnvoll sind. Einmal sollten die Zeitsteuerung und die Anfangswertberechnung verbessert werden, was für beliebige Kopplungen nützlich ist. Andererseits ist es für die spezielle Koppelsituation aus dem Gebäudebereich wichtig, unterschiedlich performante FMUs angepasst zu behandeln, mit möglichst

„großer“ Schrittweise zu arbeiten und dennoch Maßnahmen zu ergreifen, Ereignisse wie Schaltvorgänge genau zu erfassen. Dazu werden Ideen skizziert, die weiter ausgearbeitet, implementiert und erprobt werden müssen.

DANKSAGUNG

Die Arbeiten für diesen Beitrag erfolgen im Verbundprojekt „Entwicklung der Kopplungstechnologie von Komplexmodellen für Bauteil-, Raum- und Gebäudesimulation mit Modelica-basierten Anlagen-, Regelungs- und Nutzermodellen“, Förderkennzeichen 03ET1215C.

Gefördert durch:



aufgrund eines Beschlusses
des Deutschen Bundestages

Die Autoren danken den Projektpartnern J. Bastian und T. Blochwitz (ITI), A. Nicolai und A. Paepke (TUD), T. Schwan und M. Wicke (EA Systems Dresden), und K. Petridis, A. Klein und T. Schierz, sowie dem Kollegen S. Seidel für Diskussionen und Unterstützung.

LITERATUR

- Bastian, J., Clauß, C., Wolf, S., Schneider, P. 2011. Master for CoSimulation Using FMI. 8th International Modelica Conference, Dresden, March 20-22, 2011.
- EA Systems Dresden, 2015. Leistungsübersicht. Die neue Generation intelligenter Energiekonzepte. Firmenschrift, über info@ea-energie.de
- FMI: Projektwebseite, <https://www.fmi-standard.org/>
- Nicolai, A., Paepke, A. 2012. Die Gebäudesimulationsplattform NANDRAD – Physikalisches Modell, Umsetzungskonzept und Technologien im Überblick. BauSIM 2012, Berlin, 26.-28. September 2012
- Petridis, K., Clauß, C. 2015. Test of basic co-simulation algorithms using FMI. 11th International Modelica Conference, 2015, Versailles.
- Schierz, T. 2013. Modulare Zeitintegration gekoppelter Differentialgleichungssysteme in der technischen Simulation. Fortschr.-Ber. VDI Reihe 20 Nr. 447. Düsseldorf, VDI Verlag 2013.
- Schierz, T., Arnold, M., Clauß, C. 2012. Cosimulation with communication step size control in an FMI compatible master algorithm. 9th International Modelica Conference, 2012, Munich