An Investigation on Loose Coupling Co-Simulation with the BCVTB

Irene Hafner[1*], Bernhard Heinzl[1,2], Matthias Rössler[1]

[1]Simulation Services, dwh GmbH, Neustiftgasse 57-59, 1070 Vienna, Austria; *irene.hafner@dwh.at

[2]Institute for Analysis and Scientific Computing, Vienna University of Technology, Wiedner Haupstraße 8-10,
1040 Vienna, Austria

Simulation Notes Europe SNE 23(1), 2013, 45 - 50
DOI: 10.11128/sne.23.tn.10173
Received: February 15, 2013; Revised: March 20, 2013;
Accepted: March 30, 2013;

Abstract. This paper introduces several methods of cooperative simulation. Apart from the general classification and method descriptions, the numerical stability and consistency of one loose coupling approach is discussed. It is shown that consistency is maintained, although possibly of lower order, and zero stability persists as long as no algebraic dependencies between partial systems occur. The methodology of Jacobi-Type loose coupling is applied for a case study using the co-simulation tool BCVTB (see [1]). The study shows that this tool is well suited for the fast co-simulation of many instances of certain simulators, but allows no synchronisation step size control and only equidistant synchronization step sizes.

Introduction

In times of increasing environmental awareness, the prediction of energy and resource consumption is becoming more and more important. A very important auxiliary means for this issue is computer-aided mathematical simulation. Since the simulation of whole production halls including building geometry, machinery, control systems and building services needs detailed modelling of all parts, where every system requires an individual modelling approach, the method of cooperative simulation needs to be considered. Co-simulation allows the overall simulation of complex systems consisting of partial systems requiring different modelling approaches, solver step sizes or even solver algorithms.

1 Co-Simulation Types

In general, co-simulation methods are divided into two types, depending on whether data exchange takes place iteratively in every time step or only at specified synchronisation time steps.

1.1 Loose coupling

Simulations coupled via loose coupling exchange data only at certain points in time. These synchronisation references do not have to be predefined or equidistant, but hereafter only co-simulation methods at fixed, equidistant times are considered for reasons given in section 4.2. In the following a system of two partial systems depending on each other is given.

$$\dot{x_1} = f_1(x_1, y_2) \tag{1}$$

$$y_1 = g_1(x_1, y_2) \tag{2}$$

$$\dot{x_2} = f_2(x_2, y_1) \tag{3}$$

$$y_2 = g_2(x_2, y_1) \tag{4}$$

Equations (1) and (2) describe System 1 and equations (3) and (4) describe System 2. $y_i, i \in \{1,2\}$ are required in the respective other system to calculate the internal state variables. These values are synchronised at given points in time. Depending on the synchronisation order, two methods of loose coupling are distinguished.

Gauß-Seidl type. At the start time of the simulation, the initial values for all variables are exchanged. For each following synchronisation reference, the data exchange follows the procedure shown in Figure 1. Between two synchronization references, without loss of generality in System 1 the values of y_2 are extrapolated and the states of the internal variables are calculated at the individual time steps defined by the solver for System 1.

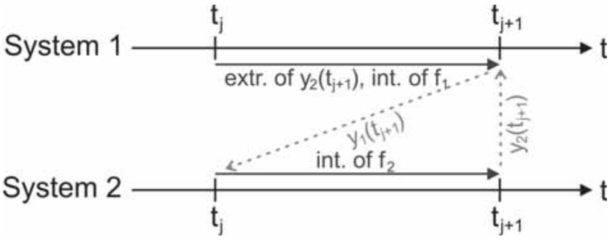

Figure 1: Overview of loose coupling co-simulation with the Gauß-Seidl type between two synchronisation references

As soon as t_{j+1} is reached, the values of y_1 until t_{j+1} are transferred to System 2, so the values needed at the time steps demanded by the solver for System 2 between t_j and t_{j+1} can be interpolated instead of extrapolated. At t_{j+1}, the values for y_2 are again transferred to System 1.

Jacobi type. This method allows all systems to calculate in parallel between two synchronisation references, which also means that each partial system has to extrapolate the values needed from the other systems. For the given example, values of of y_1 and y_2 are exchanged simultaneously at each synchronisation reference and extrapolated until the next data exchange takes place. A sketch of the method is illustrated in Figure 2.

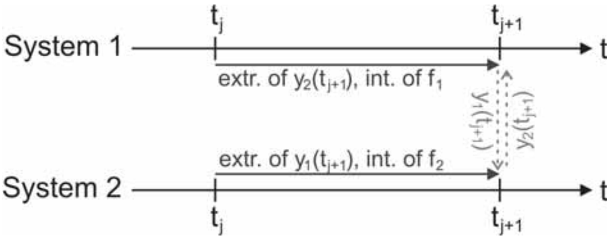
t_j
t_{j+1}
System 1
t
extr. of y_2(t_{j+1}), int. of f_1
y_1(t_{j+1})
y_2(t_{j+1})
extr. of y_1(t_{j+1}), int. of f_2
System 2
t
t_j
t_{j+1}

Figure 2: Overview of loose coupling co-simulation with the Jacobi type between two synchronisation references

1.2 Strong coupling

Co-simulation methods using strong coupling iterate the values needed from other partial systems in every time step until a specified accuracy is achieved. This approach obviously leads to far more accurate results but also boosts computing times.

2 Numerical Background

Coupling the simulation of two or more equation systems of course influences the behaviour of the underlying solver algorithms. Hence it is very important to investigate these effects to be able to determine consequences regarding numerical stability.

2.1 Consistency

For the analysis of consistency with co-simulation, let a multi-step method for solving ordinary differential equations be given as in (5):

$$\sum_{j=0}^{k} \alpha_{k-j} y_{i+1-j} = \Delta t \cdot \Phi_f(t_{i+1-j}, y_{i+1-j}, \Delta t), \qquad (5)$$

where Φ stands for the increment function of the method and Δt for the (equidistant) step size between t_i and t_{i+1}, $i = 1, \dots n$.

Consistency defines the method's error per step. The consistency error is defined by

$$\tau^{i+k}(\Delta t) := \sum_{j=0}^{k} \alpha_{k-j} y(t_{i+1-j}) - \Delta t \cdot \Phi_f(t_{i+1-j}, y(t_{i+1-j}), \Delta t). \qquad (6)$$

A method is called consistent if

$$\lim_{\Delta t \to 0} \left(\frac{\tau^{i+k}(\Delta t)}{\Delta t} \right) = 0 \qquad (7)$$

is fulfilled for arbitrary initial values. A method is called consistent of order p if there exists a constant $C > 0$ so that

$$\left\| \tau^{i+k}(\Delta t) \right\| \leq C \cdot (\Delta t)^{p+1} \qquad (8)$$

To discuss consistency of a loose coupling co-simulation method, a linear one-step method is considered, so (5) becomes

$$\alpha_0 y_i + \alpha_1 y_{i+1} = \Delta t \left(\beta_0 f(t_i, y_i) + \beta_1 f(t_{i+1}, y_{i+1}) \right) \quad (9)$$

The consistency error of this method is calculated by

$$\begin{aligned} \tau^{i+1}(\Delta t) &= \alpha_0 y(t_i) + \alpha_1 y(t_{i+1}) \\ &\quad - \Delta t \left(\beta_0 f(t_i, y(t_i)) + \beta_1 f(t_{i+1}, y(t_{i+1})) \right). \end{aligned} \qquad (10)$$

In the case of a co-simulation, the values for $y(t_{i+1})$ are needed from another partial system and hence are not known but extrapolated during one time step.

Let $y_c(t_{i+1})$ be the extrapolated value. Therefore follows the consistency error

$$\begin{aligned}\tau_c^{i+1}(\Delta t) =&\alpha_0 y(t_i)+\alpha_1 y(t_{i+1})\\ &-\Delta t\,(\beta_0 f(t_i,y(t_i))+\beta_1 f(t_{i+1},y_c(t_{i+1})))\\ =&\alpha_0 y(t_i)+\alpha_1 y(t_{i+1})\\ &-\Delta t(\beta_0 f(t_i,y(t_i))+\beta_1 f(t_{i+1},y(t_{i+1}))\\ &-\beta_1 f(t_{i+1},y(t_{i+1}))+\beta_1 f(t_{i+1},y_c(t_{i+1})))\\ =&\tau^{i+1}(\Delta t)\\ &+\Delta t\cdot\beta_1\cdot(f(t_{i+1},y(t_{i+1}))-f(t_{i+1},y_c(t_{i+1})))\end{aligned}$$

in the co-simulation. To determine consistency, we consider

$$\begin{aligned}&\left\|\frac{\tau_c^{i+k}(\Delta t)}{\Delta t}\right\| =\\ &\left\|\frac{\tau^{i+k}(\Delta t)}{\Delta t}+\beta_1\cdot(f(t_{i+1},y(t_{i+1}))-f(t_{i+1},y_c(t_{i+1})))\right\|\\ \leq&\left\|\frac{\tau^{i+k}(\Delta t)}{\Delta t}\right\|\\ &+|\beta_1|\cdot\|f(t_{i+1},y(t_{i+1}))-f(t_{i+1},y_c(t_{i+1}))\|\\ \leq&\left\|\frac{\tau^{i+k}(\Delta t)}{\Delta t}\right\|+L\cdot|\beta_1|\cdot\|y(t_{i+1})-y_c(t_{i+1})\|\end{aligned}$$

where the Lipschitz continuity of f with Lipschitz constant L conditions the last inequality. If the most simple extrapolation, i.e. taking the last known value, $y(t_i)$, for $y_c(t_{i+1})$, is applied and the Taylor approximations for $y(t_{i+1})$ and $y(t_i)$ around $t_i+\alpha\Delta t$ for an arbitrary $\alpha\in(0,1)$ are considered, the constant terms cancel each other out in the subtraction. Thus follows

$$\left\|\frac{\tau_c^{i+k}(\Delta t)}{\Delta t}\right\|\leq\left\|\frac{\tau^{i+k}(\Delta t)}{\Delta t}\right\|+L|\beta_1|\cdot O(\Delta t). \qquad (11)$$

Hence for a method of consistency order 1 the consistency order is maintained in a co-simulation, for methods of higher order consistency is maintained but of lower order.

2.2 Zero stability

A zero stabile numerical method yields a bounded solution of $y(t)=0$ for arbitrary initial conditions. To determine zero stability for a method, the first characteristic polynomial is needed. For a given multi-step method (see (5)), the first characteristic polynomial is given in (12):

$$\rho(\zeta):=\sum_{j=0}^{k}\alpha_j\zeta^j \qquad (12)$$

A method is called zero stabile if every zero λ of the first characteristic polynomial fulfils $|\alpha|<1$ and every zero with $|\alpha|=1$ is a single zero.

In the following zero stability of a linear one-step method in a loosely coupled co-simulation is considered. Regarding (9), we see that the characteristic polynomial depends solely on the equation's left side. Co-simulation inflicts changes only on the calculation of f when using $y(t_{i+1})$, so only the right side is being affected. This means that co-simulation does not influence zero stability as long as only ordinary differential equation systems are considered. For differential-algebraic equation systems, [2] states that zero stability is maintained as long as no algebraic interdependencies between the partial systems occur.

3 Co-Simulation with the BCVTB

The Building Controls Virtual Test Bed (BCVTB) is a co-simulation tool which has been developed at the University of California, Berkeley and allows co-simulation of the building simulation software EnergyPlus [6], simulators of the object-oriented standard Modelica [4], MATLAB [3] and its toolboxes Simulink and Simscape, Radiance [7] and Functional Mockup Interfaces [5].

BCVTB is based on Ptolemy and provides on the one hand so-called *simulator actors* which are part of the BCVTB environment and define the simulators and corresponding source files to be co-simulated. On the other hand, functions or function blocks for the communication with BCVTB are provided for each participating simulator. The communication itself takes place via so-called BSD sockets, which have also been developed at the University of California for inter-process communication. BCVTB allows only loose coupling co-simulation of Jacobi type with predefined, equidistant synchronisation references.

4 Case Studies

4.1 Production hall in Energy Plus

In the following case study, the model of a production hall is co-simulated with the BCVTB. The machines located in the different halls emit heat which has to be transferred to the respective rooms. Figure 3 shows the sketch of the building model to be simulated.

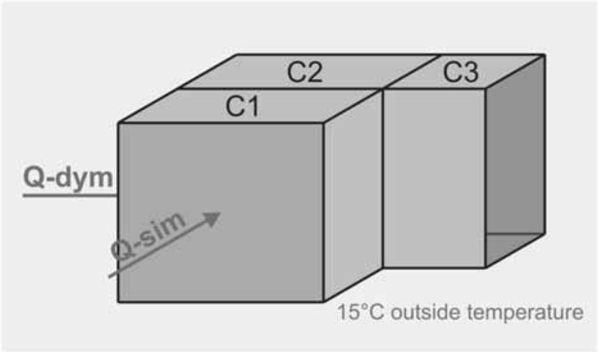

Figure 3: Overview of the building to be simulated.

The building itself is modelled in EnergyPlus, the machines are implemented in Simscape and Dymola respectively. A sketch of the communication via BCVTB is shown in Figure 4. The solvers used for the simulation of the individual models can be found in the corresponding brackets.

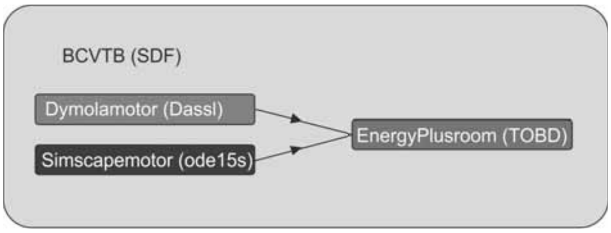

Figure 4: Overview of the intended communication between the individual simulators via BCVTB.

Both machines are switched on at 8 a.m. in the morning and switched off at 4 p.m. in the afternoon. The temperature of the environment of the hall is defined in a weather data sheet which is needed as an input to the EnergyPlus model. Figure 5 shows the heat emission of both machines and Figure 6 shows the room temperature in all rooms over one week responding to the heat insertion.

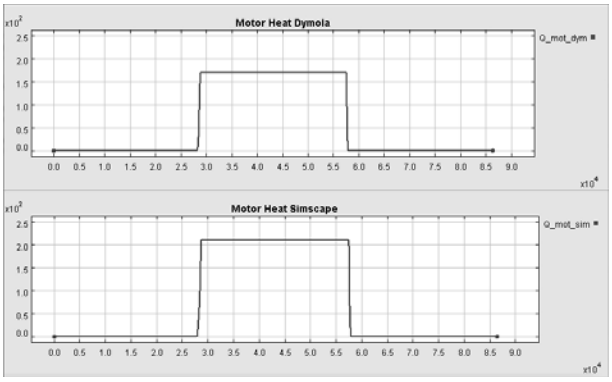

Figure 5: Heat loss of the machine models.

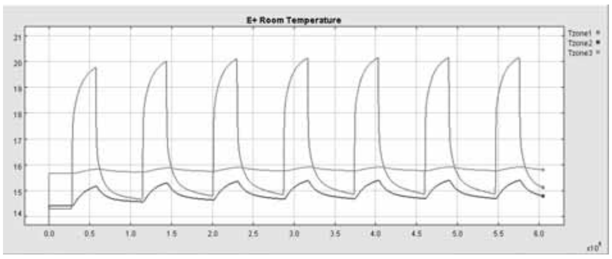

Figure 6: Room temperature over one week.

Regarding Figure 6 it becomes clear that the room temperature reacts clearly to the heat loss of the machines but still manages to cool down without further ado (like HVAC systems controlling the temperature). It is also evident that the room containing the machines (C1) reacts to the emission with much faster temperature increase than the rooms without a machine. Furthermore, since room C3 contains a window, its temperature is also able to cool down rather rapidly. Between two synchronisation references, every simulation uses the time steps given by its own solver and at each reference the data needed from other partial systems is exchanged. This procedure is shown in Figure 7.

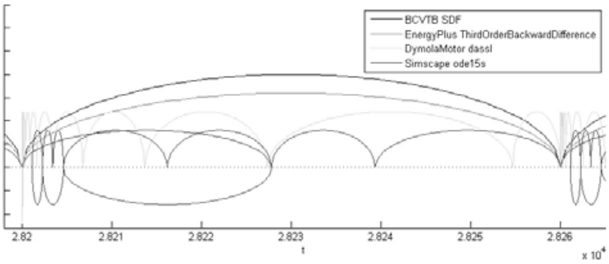

Figure 7: Steps taken by the individual solvers between two synchronisation references.

It can be observed that the solvers for the machine models need to discard steps after the data exchange since they are intended too large in the first place. In EnergyPlus the overall step size is also defined for the internal calculation as no major changes take place internally and furthermore EnergyPlus fires data exchange at every internal step, so the co-simulation step size has to be the very same as the step size for EnergyPlus.

In Simscape and Dymola it is possible to calculate internal states at additional points in time after minor modifications of the predefined communication devices. To keep the room temperature in an interval convenient for human beings even in high summer or in a well insulated hall, a control (e.g. in Simulink) can be added to the model but since this requires a feedback (see Figure 8), a loop has to be broken by the insertion of a time lag of one step into the control model.

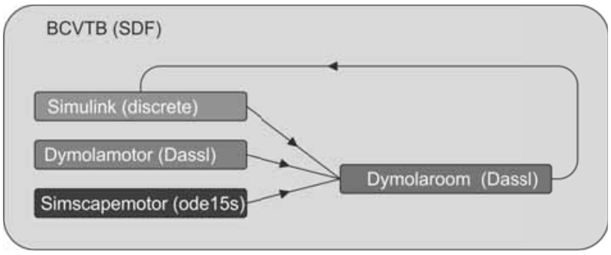

Figure 8: Overview of the communication via BCVTB with the inclusion of temperature control.

4.2 Experiment with step size control

Since BCVTB in general offers also a *continuous director* meaning an ODE solver with step size control for the overall simulation, this possibility has also been tried out. The basic model for this experiment was the model of a motor implemented in Simscape which transfers its waste heat to BCVTB where it is simply plotted. The BCVTB model for this scenario is given in Figure 9.

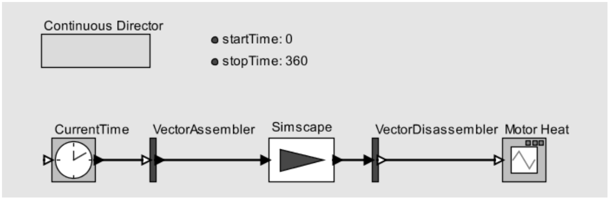

Figure 9: BCVTB model with only one partial model, a machine implemented in Simscape.

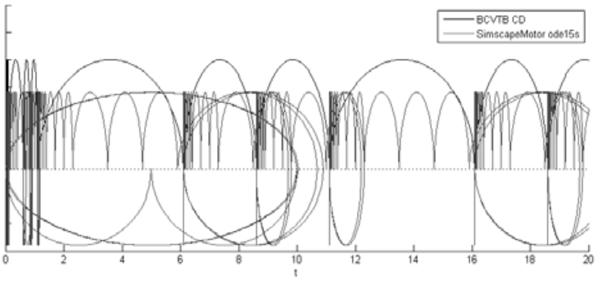

Figure 10: Steps taken by the *continuous director* of BCVTB and ode 15s in Simscape at the beginning of the simulation.

So far, everything works fine. However, the use of a continuous solver does not make much sense for this system, since nothing actually takes place in the BCVTB model. This can also be seen by inspecting the steps taken by the *continuous director*, which show periodic behaviour after a few smaller steps at the beginning, which are shown in Figure 10. Even after the beginning of the heat emission of the machine at $t = 200s$ the *continuous director* maintains its periodic behaviour of taking one bigger and two smaller steps consequently (see Figure 11).

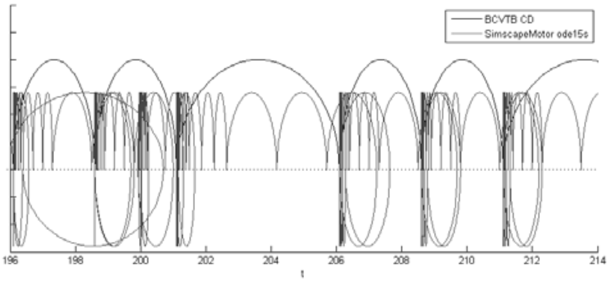

Figure 11: Steps taken by the *continuous director* of BCVTB and ode15s in Simscape at the beginning of the motor heat emission.

Furthermore the Simscape solver has to make many redundant steps since the time for synchronization has to be iterated.

If an integrator is inserted to calculate the energy consumption of the machine (see Figure 12), which justifies the usage of step size control, a somehow predictable performance occurs: up to switching the motor on, the same behaviour as in the model without integrator can be observed. As soon as the motor starts to emit heat, the BCVTB solver realizes that the output of the Simscape *simulator actor* has changed gravely and so it

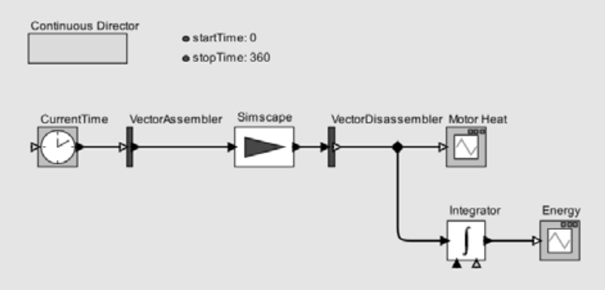

Figure 12: BCVTB model with an integrator determining the energy discharged by a machine.

wants to step back to iterate the time of change more precisely. Simscape in the meantime does not know BCVTB has stepped back and hence does not step back itself but waits for BCVTB to fire the next synchronization event – which can never occur in the future since it already happens in the past. This, however, means that BCVTB waits for Simscape to respond to its call and Simscape never does since it does not look back, so both simulators would wait forever for each other. This quite simple experiment proves that the use of a *continuous director* for the overall simulation is not suited for co-simulation with BCVTB as either the model has to be simple enough that the solver does never need to step back, which means a fixed step solver would be just as well or even better suited, or a deadlock would occur if the overall solver is obliged to discard steps.

5 Conclusion

This article shows that in a loose coupling co-simulation consistency is maintained but possibly of lower order and zero stability can be held with restrictions to algebraic dependencies. Apart from the numerical analysis, the possibilities and limits of co-simulation with the BCVTB have been investigated with respect to the number of participating simulators, diversity of models and methods of numerical solution approaches. Related work has shown that BCVTB allows the co-simulation of many instances of several simulators in a quick and rather easy way, but as described in this paper BCVTB covers only loose coupling co-simulation at equidistant points in time since step size control leads to a deadlock, see section 4.2. Further research will aim the investigation of additional co-simulation and multi-rate simulation methods, consideration of numerical issues and testing of other given co-simulation tools with regard to generality and provided coupling methods.

Acknowledgement

This work has partially been supported by the Austrian Research Promotion Agency (FFG).

References

[1] Wetter M. *Co-Simulation of Building Energy and Control Systems with the Building Controls Virtual Test Bed*. Journal of Building Performance Simulation. 2011; 4(3):185-203. doi: 10.1080/19401493.2010.518631.

[2] Kübler R, Schielen W. Two methods of simulator coupling. *Mathematical and Computer Modelling of Dynamical Systems*. 2000; 6(2): 93-113. doi: 10.1076/1387-3954(200006)6:2;1-M;FT093.

[3] mathworks: Products and Services [Internet]. The MathWorks, Inc.: c1994-2014 [cited 2014 Nov 13]. Available from: `http://uk.mathworks.com/products/index.html?s_tid=gn_loc_drop`

[4] modelica: Modelica and the Modelica Association [Internet]. Modelica Association c2000-2014 [cited 2014 Nov 13]. Available from:`https://www.modelica.org/`

[5] fmi-standard: Functional Mock-up Interface [Internet]. Modelica Association c2000-2014 [cited 2014 Nov 13]. Available from: `https://www.fmi-standard.org/`

[6] eere.energy: EnergyPlus Energy Simulation Software [Internet]. U.S. Department of Energy [cited 2014 Nov 13]. Available from: `http://apps1.eere.energy.gov/buildings/energyplus/energyplus_documentation.cfm`

[7] radsite: Radiance [Internet]. Building Technology and Urban Systems Department c2014 [cited 2014 Nov 13]. Available from: `http://radsite.lbl.gov/radiance/`

[8] Trčka M. *Co-simulation for Performance Prediction of Innovative Integrated Mechanical Energy Systems in Buildings* [dissertation]. [Faculty of Architecture, Building and Planning, (NL)]. Technische Universiteit Eindhoven; 2008.

[9] Hafner I. *Möglichkeiten der Co-Simulation mit dem Building Controls Virtual Test Bed für den Bereich der objektorientierten Modellbildung physikalischer Systeme* [diploma thesis]. [Institute of Analysis and Scientific Computing, (AT)]. Vienna University of Technology; 2013.