Research Plan:
Using Iterative Simulation for Timing Analysis of
Complex Real-Time Systems
MdH Research Planning Course 2007
Workshop 3 – Research Plan Final Report

Yue Lu
ylu01@mdh.se
March 22\textsuperscript{nd}, 2007
Table of Content

1 Central Terminology ........................................................................................................... 3
2 Research Overview ............................................................................................................. 4
3. Current Research Issues .................................................................................................. 6
  3.1 Improvements about previous version simulator ............................................................ 6
  3.2 Iterative simulation algorithm (ISA) ....................................................................... 6
  3.3 Iterative simulation algorithm implementation ....................................................... 7
  3.4 Tool prototype / Framework development ............................................................... 8
  3.5 Results comparison ................................................................................................. 8
4. Research Methods ......................................................................................................... 9
  4.1 Hypothesis ................................................................................................................ 9
  4.2 Research steps ......................................................................................................... 9
  4.3 Expected results ................................................................................................... 9
    4.3.1 Current Tool Development ............................................................................... 9
    4.3.2 Result Comparison ......................................................................................... 9
  4.4 Evaluation ............................................................................................................... 10
  4.5 Expected contribution ......................................................................................... 10
5. Overview of Filed of Research and Central literature ............................................. 11
  5.1 Previous work ....................................................................................................... 11
  5.2 Seminal papers ..................................................................................................... 11
  5.3 Central literature ................................................................................................ 11
  5.4 Related research projects .................................................................................. 12
    5.4.1 Simulation tools used for real time system .................................................. 12
  5.5 Key conferences and leading research groups .................................................. 13
    5.5.1 Key/targeting conferences ............................................................................ 13
    5.5.2 Leading researchers and research groups .................................................. 13
6. Project Plan .................................................................................................................. 15
  6.1 Activities ............................................................................................................... 15
  6.2 Planned papers .................................................................................................... 15
  6.3 Milestones and Time plan ................................................................................ 15
    Time plan ................................................................................................................. 15
7. References ................................................................................................................... 16
1 Central Terminology

- **RTOS**
  Real-Time Operating System, an operating system designed to be used in real time systems.

- **Simulation**
  Simulation is a method which can be used for analysis of response time. When using simulation, a detailed model of the system is executed in a simulator environment. Simulation can be used to evaluate the performance of the system when it has to be altered, which helps to reduce the risks of failure.

- **Genetic Algorithm**
  Genetic Algorithms (GAs) are the search algorithms based on idea of selection and genetics. It follows the rule given by Charles Darwin of survival of the fittest. It does a random search within a defined space for solving a problem.

- **Iterative Simulation Algorithm**
  Iterative Simulation Algorithm is not only to do the simulation randomly but also search for the worst case scenario of a system. This is an iterative process where each iteration finds even more extreme scenarios, until the termination point is reached, e.g. no more worst case is found in each iteration. It can be regarded as the special instance of Genetic Algorithm.

- **Iterative Simulation Controller**
  It is the kind of software, which will call simulator many times, collect data from each execution of simulator, process the data, guide simulator to find the most interesting test case, e.g. the task with the highest response time, buffer status etc. It is the core part of iterative simulation framework.

- **Tasks**
  A process in a real time system usually with some deadline and a period.

- **Response time**
  The time in which system gives output after taking input.

- **Worst case response time**
  The maximum possible response time of a task. This occurs when all tasks in the system wants to execute at the same time, and with the individual worst-case execution time.

- **Worst case execution time (WCET)**
  The longest possible execution time of the task.
2 Research Overview

The size and complexity of software in embedded systems is increasing with every new product launched in the market, for instance ABB Robotics Control System. When changing such complex systems i.e. adding a new feature, it is hard to predict the consequences on the behavior. This is especially the case for temporal properties of the system, e.g. response time, which depends upon factors that one cannot figure out during the implementation, such as execution time. This is a big problem when developing industrial and embedded systems, which often are real-time systems.

The root cause is that such real time systems can not be analyzed by the existing analytical methods for response time analysis, i.e. intrusive methods and non-intrusive. The first way, in which the system is redesigned to fit an existing analysis method, e.g. rate monotonic analysis, is very costly since it requires a lot of work and there is a high risk of causing new bugs. For some systems analytical scheduability analysis is infeasible [14] due to the high pessimism, since an analysis would always give a negative result - even though the real system works perfectly. [20] A non-intrusive method is when the system is not changed, but instead to construct a model of the existing system that can be analyzed, using e.g. model checking and simulation. The former cannot be scale well with system size and complexity, e.g. state space explosion issue. Whereas the latter shows the way of telling how the system will work after the changes have been made in terms of prediction result, by analyzing the model of a system. Nowadays, there have been a number of methods and tools developed for timing analysis, Rate-Monotonic Analysis for periodic tasks, Real-Time Calculus for tasks described using arrival curves, and Times using Timed Automata. [2] Some of these techniques, e.g., implemented in Times can deal with systems with complex release patterns, but do not scale well with system size and complexity due to issues, e.g. state space explosion; the others are scalable but can not handle systems with complex structures.

Therefore, it is interested in developing simulation-based analysis of real-time systems. However, since not all behaviors of a model are explored (it is only a random subset) by simulation, there is no way of knowing if the "worst case" of a model has been found, or if even more extreme behaviors are possible than the ones have observed in the simulation.

It has to be considered as one kind of global optimization problem, e.g. how to guide the simulator find the more “interesting” test cases of the target system. There are many methods existing in the global optimum in a large search space arena. The most famous ones are Simulated Annealing (SA), and Genetic Algorithm (GA).

The key point of SA is to choose better Annealing Scheduling, which emulates the annealing scheduling in the nature, and depends on the time budget used so far[4]. For the complex system, i.e. there are many tasks involved; latter is impossible to reach in the short time, i.e. a week, month, even one year.[3] Furthermore, SA may pick the neighbor which may not lead to the worst case, although it seems to be, in our application. SA is a
good at local search, and may arise the local minimum, but not strong in the global search. [3] It should use various methods to jump out the current pit. [4]

Although GA is widely and successfully adopted in different application and the positive results returned. But the difficulty is to find the set of parameters of genetic operations, e.g. cross over, mutation, and selection. In our application, some of generated test cases will switch seeds with its present generation test case, some has to change the seed randomly, and some should choose the next round seeds with probability from scheduled, “excellent” seeds which lead to the interesting test cases, e.g. highest response time of some task. But when the system is complex, it is difficult to choose such sets of parameters and decide which generic operations will be the most dominant or should have the same probability.

The main focus of this filed research is to find an improved simulation technology based on analysis method for real time systems while applying the new, better global optimization technology. The idea is randomly, do many simulations, find the most interesting ones (e.g. the 1% with highest response time for some task in the beginning), and then use these as starting points of new simulations in order to find a better approximation of the worst-case scenario. This is an iterative process where each iteration finds even more extreme scenarios, until the termination point is reached. Therefore, it is named as “Iterative Simulation”, which can be regarded as the specific instance of GA.

The research will not only contains a literature study, which includes explaining the main existing timing analysis methods in detail, important parameters of the methods, developing iterative simulation algorithm, providing a comparison among results from above methods, but also implementing the framework which integrates a set of tools, and framework evaluation and application in the practical (toy example + complex system model). For the future perspective, this framework will try to help their engineers to find the worst case of specific properties of the system, test and analyze the system, improve system performance in the industrial.
3. Current Research Issues

In this section, the improvement about previous version simulator, Iterative simulation algorithm and its implementation, and its framework design will be introduced.

3.1 Improvements about previous version simulator

Previous simulator has to be improved with a set of parameters, e.g. how to let the existing simulator starts running from the previous time instance to the point that user specified or required; How to get a better random number generator in order to make the better stochastic model; How to make the simulator run-time errors which are not easy or possible to be corrected in short time, more “distinctive” and can be caught correctly by Iterative Simulator Controller; How to synchronize TASKs threads that will fetch the generated “better” random numbers, and thread which will generate the random numbers without any conflict and waste; How to implement scheduled seeds file recorder with the specific format for both simulator and ISC etc.

Currently, simulator is improved and can fit for above requirements.

3.2 Iterative simulation algorithm (ISA)

As it is described, Iterative simulation algorithm (ISA) can be regarded as the specific application of GA in real-time system. The main difference between GA and iterative simulation is ISA only uses mutation. I.e. each individual in the new generation is based on exactly one individual from the previous state of iteration, not.replying on the other “upper”/ “father” iteration state.

It can be described as follows, firstly Iterative Simulation Controller (ISC), will call simulator to perform a large set of (random) simulations of the model, e.g. 1000 test cases, then ISC will select a fraction of these simulations test cases with the most extreme values for some properties, e.g. the 1% of test cases with highest response time for task X or resource usage. Thirdly, in each selected simulation, the ISC will perform new round simulations of the model from appropriate system state by calling simulator with a set of parameters, e.g. seeds, simulation time that are scheduled and new used in next round simulation. And then get back to second step but with picking all the worst cases found, iteratively. This process is not repeated until no worse case is found in the next iteration of the process, which is also defined as “stopping criteria”.

- Framework testing, e.g. integrated testing of ISC + improved simulator + tracealyzer which has already been deployed in ABB successfully, as toy example and target complex system model
- Results comparisons among Iterative simulation, analytical analysis and timed
model checking.

### 3.3 Iterative simulation algorithm implementation

ISA is implemented as iterative simulation controller (ISC), which is the core part of the Iterative Simulation Framework (ISF). ISF integrates a set of tools, e.g. simulator, analysis tool, tracealyzer, results board currently, and some other tools to be developed in the future which will enhance the framework functionalities and performance.

ISC is divided into two parts; the first one is Iterative Simulation Controller 1, which is also referred as “Test-case generator”, and generates certain number of test cases, e.g. 1,000. The following diagram depicts how it works.

![Figure 1: Iterative Simulation Controller 1 working process](image)

The other part is Iterative Simulation Controller N, which is responsible for works except for test cases generator. The following diagram depicts how it works.
### 3.4 Tool prototype / Framework development

Currently, first version of the prototype is developed by C# and can be distributed in the .Net 2.0 environment. The further version tool development is in the pipeline.

### 3.5 Results comparison

The evaluation results from the ISA will be compared with the ones that are from analytical response time analysis, model checking (MC) tool e.g. timestool, etc.
4. Research Methods

4.1 Hypothesis

1. Backward time of each iterative simulation is 0.8 times of happening time instance of aiming task with interesting properties. It is used when the scenario is retrieved by the simulator, and will be adjusted upon the request.

2. Because of time and resource limitation, e.g. PC performance, the current time of Iterative simulation is supported to 3. Also, the most important task right now is to implement the algorithm and make the tool prototype as soon as possible, and post the observation to the results and compare it with other methods. Therefore, the software architecture is not good enough. One further version which enhance to improve the program structure and performance is in the action now, and first version will be released based on it.

4.2 Research steps

My research method consists of the following steps:

- Literature survey, e.g. simulation technologies which include continuous and discrete event simulation. Relevant search algorithms e.g. Simulated Annealing (SA), Genetic Algorithm (GA) etc.
- Study and development of iterative simulation algorithm, which can be considered as one specific application / instance of GA in real-time system application.
- Framework implementation, e.g. develop the Iterative Simulator Controller, framework configurator and other make-up tools, and improve the existing simulator.
- Evaluate the algorithm, framework, and compare the generated results with the ones from other analytical analysis technologies, e.g. response time analytical analysis and timed model checking.

4.3 Expected results

4.3.1 Current Tool Development

Here is only briefly introduction about the tool implemented. Currently, this version software can be distributed in the DotNet 2.0 framework, which means the further / enhanced version can be installed and used without any version problems / conflicts, e.g. DLL hell.

4.3.2 Result Comparison

Hopefully, the result, e.g. the most interesting case with highest response time of certain
task found, will locate between the domain between analytical analysis and model checking. The former is the safe but pessimistic. And since there are some abstractions, time assumption, the model pattern proposed, when applying Model Checking tool that we plan to use. Therefore, the result is safe and accurate. Anyhow, we will get conclusion after the result for each comes out.

### 4.4 Evaluation

The complex example which consists of a few tasks, buffer operations, virtual tasks that is from industry application will be evaluated by Iterative Simulation Framework, in a few days. And further evaluation will be planned.

### 4.5 Expected contribution

By introducing Iterative Simulation, we proposed a better / advanced simulation technology with applying new global optimization technology, to find the most interesting case of target system based on timing analysis, e.g. worst case response time of certain task. And framework will be presented, and results comparison will be done.
5. Overview of Filed of Research and Central literature

5.1 Previous work

The previous work has been performed by Johan Kraft, who is the senior PhD student in the research cluster LEGACY, Mälardalen University. It has shown the motivation for the research, completed basic literature study and algorithm design. E.g. in the literature part, the following results among response time analysis, model checking and simulation is present.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Response Time analysis</th>
<th>Model checking</th>
<th>Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements.</td>
<td>It require some assumption to be fulfilled</td>
<td>Requires a detailed analyzable model of the system, which can take a lot of time to develop and maintain</td>
<td>Requires a detailed analyzable model of the system, which can take a lot of time to develop and maintain</td>
</tr>
<tr>
<td></td>
<td>• All tasks must be periodic</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Tasks are independent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>State space explosion problem</td>
<td>No</td>
<td>yes</td>
<td>No</td>
</tr>
<tr>
<td>Accuracy</td>
<td>Safe but Pessimistic</td>
<td>Safe and Accurate</td>
<td>Accurate but not safe (Optimistic)</td>
</tr>
<tr>
<td>Learning threshold</td>
<td>Low</td>
<td>High</td>
<td>Medium</td>
</tr>
</tbody>
</table>

Figure 3: Comparison among Response Time analysis, Model checking and Simulation from the literature study

5.2 Seminal papers

There is very little study about iterative simulation nowadays. From the literature study, there is one tool declared to support the iterative simulation, but with the other meaning, e.g. to observe the simulation step by step and follow the evaluation of the model states with the variables updates. There is one Master’s thesis about Iterative simulation literature study. [1] And some interesting papers in the global optimization filed, like [4] etc.

5.3 Central literature

The Genetic Algorithms (GA) is related to the iterative simulation algorithm. However, in the iterative simulation, each individual in the new generation is based on exactly one individual from the previous state of iteration, not replying on the other “upper”/ “father” iteration state. By which means ISA only uses mutation. It is also possible to set the specific value of seed in the simulator, in the iterative simulation framework. Therefore,
the selection is involved also in ISA. Anyhow, ISA can be regarded as one specific instance of GA.

5.4 Related research projects

5.4.1 Simulation tools used for real time system

**STRESS**
It is used for the analyses of hard real time systems. It is a language based simulator and can be used to study the general behaviour of an application. Algorithms for resource sharing and task scheduling can be defined in STRESS. It is mostly used for testing resource sharing and scheduling algorithms. STRESS tool include a simulator, a presentation tool and examples of modelling language. It does not support modelling distributions of execution times or memory allocation. [14, 1]

**DRTSS**
It is a simulation framework that “allows its users to easily construct discrete-event simulators of complex, multi-paradigm, distributed real-time systems” [1]. We can build a simulator by picking it from different set of algorithms and protocols in DRTSS; we can also add new algorithms and protocols to the original set. DRTSS is a part of PERTS family which was developed by University if Illinois at Urbana-Champaign. DRTSS helps in getting more detailed analysis of a system. [14, 1]

**ART-ML**
ART-ML (architecture and real-time behaviour modelling language) has been developed at Mälardalen University at Department of computer engineering, and it has been used in an industrial case study [14]. It was designed as a mean to describe complex real time systems and is an imperative language similar to ANSI C, and it is based on the concepts of tasks. ART-ML has support for probabilistic modelling through the “chance” statement. The simulation time is advanced explicitly through an “execute” statement, which “consumes” CPU time according to a given probability distribution. [14]

**ARTISST**
It’s an event driven simulation tool which gives us the accurate simulation of real time systems. It is used for simulating the complex systems, Real time operating system (RTOS), interacting with an external environment by way of events. [15]

The ARTISST framework comes with a set of default modules, and a set of default schedulers and task models. It uses C or C++ as the general programming language which allows the simulation to be implemented very closely to the applications which are real, by allowing the code to be reused. The simulator of the ARTISST is fully customizable as it doesn’t depend upon any operating system due to its extensible object oriented architecture. The result of the simulation can be seen in the form of a Gantt chart or series of evaluation metrics. The main advantage of using the ARTISST is that it takes
all the system overhead in to account. [15]

Figure 4: ARTISST Framework

5.5 Key conferences and leading research groups

5.5.1 Key/targeting conferences

• Winter Simulation Conference ’07
• To be updated in the future

5.5.2 Leading researchers and research groups

LAND: Laboratory for modeling, analysis and development of networks and computer systems

LAND is part of the Computer Science Department and the Systems Engineering and Computer Science Program (PESC/COPPE) of the Federal University of Rio de Janeiro / Brazil (UFRJ) and is located at COPPE. The research group is composed by professors, undergraduate, master and PhD students. The main research topics are:

• Development of new algorithms to solve models of computer and communication systems
• Development of tools for performance/availability analysis of computer and communication systems
• Development of multimedia applications
• Development of models for multimedia network mechanisms and multimedia traffic.

You can visit their homepages at www.land.ufrj.br.

System Simulation: the Shortest Route to Applications

This site features information about discrete event system modeling and simulation. It includes discussions on descriptive simulation modeling, programming commands, techniques for sensitivity estimation, optimization and goal-seeking by simulation, and what-if analysis.
Advancements in computing power, availability of PC-based modeling and simulation, and efficient computational methodology are allowing leading-edge of prescriptive simulation modeling such as optimization to pursue investigations in systems analysis, design, and control processes that were previously beyond reach of the modelers and decision makers. You can visit their homepages http://home.ubalt.edu/ntsbarsh/simulation/sim.htm.

**IBM simulation producer**

IBM Simulation Producer is an application simulation capture program, ideal for quickly producing application simulations. Similar to a "ScreenCam," it automatically captures images and interactions in an application, such as mouse clicks and keyboard commands. Designed for SME who needs to create robust e-learning application simulations quickly. Creates simulations rapidly for interactive learning, even when/where the real application has not been deployed. Details include: SME authoring, No downloads or plug-ins to view content, Compact simulation file sizes, Creates training documentation - MS Word based, AICC/SCORM standards compliant etc.

**UPPAAL**

Uppaal is an integrated tool environment for modeling, validation and verification of real-time systems modeled as networks of timed automata, extended with data types (bounded integers, arrays, etc.). The tool is developed in collaboration between the Department of Information Technology at Uppsala University, Sweden and the Department of Computer Science at Aalborg University in Denmark. More information about the tool and groups is available at www.uppaal.com website.

**TIMES groups**

TIMES (a Tool for Implementation and Modeling of Embedded Systems) is a tool set for modeling, schedulability analysis, and synthesis of (optimal) schedules and executable code. It is appropriate for systems that can be described as a set of tasks which are triggered periodically or sporadically by time or external events.

This group is primarily intended as a forum for discussions between users of the tool. It may also be used to announce new tool releases and related papers. To prevent spam, messages from new members are always moderated. More information about the TIMES tool is available at the www.timestool.com.
6. Project Plan

6.1 Activities

Previous work from the beginning of my PhD study in Winter Nov, 2006 until now has been Iterative Simulation Algorithm Implementation, prototype built up and presentation of the current work.

6.2 Planned papers

1. Iterative simulation of complex real-time system based on timing analysis.
   This paper will not only contains a literature study, which includes explaining the main existing timing analysis methods in detail, important parameters of the methods, developing iterative simulation algorithm, providing a comparison among results from above methods, but also implementing the framework which integrates a set of tools, and framework evaluation and application in the practical (toy example + complex system model). Hopefully, it will be regarded as the central literature in the iterative simulation field.

2. Improved Iterative Simulation algorithm.
   After the complex system model is evaluated, the result of Iterative Simulation will be compared with other methods, and the improvements will be explored along with accumulating more knowledge in the optimization filed.

6.3 Milestones and Time plan

<table>
<thead>
<tr>
<th>Spring 2007</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>April 2007</td>
<td>Paper submit to 13th IEEE PTCSA Conference</td>
</tr>
<tr>
<td>May 2007</td>
<td>Tool improvement and maintenance</td>
</tr>
<tr>
<td>June to August 2007</td>
<td>Tool maintenance and development</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Summer 2007</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>August 2007</td>
<td>13th IEEE PTCSA Conference</td>
</tr>
<tr>
<td>December 2007</td>
<td>Winter Simulation Conference ’07</td>
</tr>
</tbody>
</table>

Time plan

- In the end of March, first paper, which is aiming at the 13th IEEE RTCSA conference, will be published. In this paper, literature study, basic iterative control algorithm, framework implementation, a toy example or complex system model will be evaluated. Moreover, several results generated from the existing technology will be compared, e.g. analytical response-time analysis, timed model checking tool etc.
- Afterwards, there are some activities will be involved. E.g. tool improvements will be in the pipeline, the application of complex example will be done.
7. References


[6] Real time scheduling-part1 by Damir isovic Mälardalen University.


[14] Uppaal Website www.uppaal.com


[16] Discrete Event simulation- what is simulation by ib expert in simulations 20th august 2006 web link: http://www.ibrightsolutions.co.uk/support/simulation/simulation.htm#des


[18] Introduction temporal analyzability late in the lifecycle of complex Real-time
systems by Anders Wall, Johan Anderson, Jonas Neander, Christer Norström and Martin Lembke at Department of Computer Engineering, Mälardalen University.


[23] Simulation system for real-time planning, scheduling, and control. Year of Publication: 1996 
ISBN: 0-7803-3383-7 by Glenn R. Drake and Jeffrey S. Smith