

A Statistical Approach for Validation of Task Simulation Models with Intricate Temporal Execution Dependencies

Yue Lu, Johan Kraft, Thomas Nolte and Christer Norström
 Mälardalen Real-Time Research Centre
 Mälardalen University, Västerås, Sweden
 {yue.lu, johan.kraft, thomas.nolte, christer.norstrom}@mdh.se

Abstract

This paper presents a novel approach to validation of temporal simulation models extracted from real industrial control systems containing intricate task execution dependencies, by introducing existing mature statistical methods to the context. The proposed approach firstly collects sampling distributions of response time data of tasks in both the modeled system and the model, in terms of simple random samples (SRS). The second step of the approach is to compare the sampling distributions using a non-parametric Kolmogorov-Smirnov test. After evaluating a fictive system model inspired by a real robotic control system, the proposed algorithm shows the possibility of identifying the temporal differences between a target system and its extracted model, i.e., whether the model is a sufficiently accurate approximation of the target system. The approach makes few assumptions on the system design and scales to very large and complex systems.

1 Introduction

To date, most existing embedded real-time software systems have been developed in a traditional code-oriented manner. Many such systems are maintained over extended periods of time, sometimes spanning decades, during which systems become larger and increasingly complex. As a result, these systems are difficult and expensive to maintain and verify. There are many industrial embedded systems consisting of millions of lines of C code, and containing 50 - 100 tasks or more, out of which many tasks have real-time constraints. One example of such systems is the robotic control systems developed by ABB. Looking closer at these systems, the adhering tasks exhibit strong temporal dependencies, e.g., asynchronous message-passing, globally shared state variables and runtime changeability of periods and priorities of tasks, which vary the execution time

of tasks radically.

Model-based analysis of complex real-time systems has the potential of not only allowing for response-time analysis of such systems, but also facilitating migration toward a component based real-time system by e.g., analyzing the timing properties of the existing code and wrapping it into components.

A major issue when using model-based timing analysis is how to obtain the necessary analysis model, which should be a subset of the original program focusing on behavior of significance for task scheduling, communication and allocation of logical resources. For many systems, manual modeling would be far too time-consuming and error-prone. Two methods for automated model extraction are proposed in [1]. A tool for automated model extraction is in development, named MXTC - Model eXtraction Tool for C. The MXTC tool targets large implementations in C, consisting of millions of lines of code, and is based on program slicing.

However, there is one important issue to be raised, i.e., *model validity*, which is defined as *the process of determining whether a simulation model is an accurate representation of the system, for the particular objectives of the study* [2]. As a model is an abstraction of the system, some system details may be omitted in the model, for instance by using probabilistic modeling. Thus, the results from a simulation of such models may not be identical to the recordings of the system, e.g., with regard to the exact task response time. In order to convince system experts to use simulation-based methods, the models should reflect the system with a satisfactory level of significance, i.e., as a sufficiently accurate approximation of the actual system. Therefore, an appropriate validation process should be performed before using the models.

In this paper, we present a novel approach for validation of temporal simulation models extracted from real industrial control systems containing intricate task execution dependencies. That is, to consider this particular problem as a statistical problem, then, which could be solved by using existing, mature methods from statistic field. The pro-

posed method uses sampling distributions collected by simple random samples (SRS) [3] at first, and then produces results concerning whether the model is a sufficiently accurate approximation of the target system according to a non-parametric Kolmogorov-Smirnov test.

2 RTSSim Simulation Models

The models, as the outcome of the tool MXTC, are described using the modeling language in RTSSim [5], which describes both architecture and behavior of task-oriented systems developed in C. An RTSSim simulation model consists of a set of tasks, sharing a single processor. Each task in RTSSim is a C program, which executes in a “sandbox” environment with similar services and runtime mechanisms as a normal real-time operating system, e.g., task scheduling, inter-process communication (message queues) and synchronization (semaphores). The default scheduling policy of RTSSim is Fixed-Priority Preemptive Scheduling (FPPS) and each task has scheduling attributes such as priority, periodicity, offset and jitter. RTSSim allows for three types of selections which are directly controlled by simulator input data:

1. selection of execution times in `execute` statements,
2. selection of task jitter, and
3. selection of task behaviors, depending on the system environment, e.g., random number of external events generated by sensors.

In RTSSim, Monte Carlo simulation is realized by providing randomly generated input data. A more thorough description of RTSSim can be found in [5].

3 Model Validation

Related Work

For the sake of space, we only briefly introduce the related work concerning the model validation process. There are various methods to do the comparison; these methods are either objective or subjective. Subjective methods are often used for validation of simulation models; examples of subjective methods are Face Validation, Graphical Comparisons and Sensitive Analysis [6], which are highly dependent on domain expertise and prone to human error. Objective methods use mathematical methods to compare the outputs from the real system with the output from the simulation model. In [7], authors presented a notation of model equivalence based on observable property equivalence which is used to compare results of a model and an actual system. A method in [8] is presented for automated validation of models extracted from real-time systems by checking if the model can generate the same event sequences as the recorded event sequences from the system using a model checker.

Problem Formulation

We are given a model S' which is extracted from a real system (or modeled system) S containing a task set Γ including m tasks, where $m \in \mathbb{N}$. Let $RT_{samples}(S', \tau_i)$ and $RT_{samples}(S, \tau_i)$ denote the sampling distributions of the response time measured for a task τ_i in S' and S respectively. The goal of the problem is then to find: whether for every task τ_i in the task set Γ , its underline population (from which the sampling distribution of a task τ_i is) of response time in S and S' is statistically significant or not.

Descriptive Statistics of Response-Time Distributions

Table 1 shows the numerical summary of the center and the spread (or variability) of sampling distributions of the response time (RT) data of tasks in the Model 1 containing intricate execution dependencies, used for the evaluation in Section 5. In Table 1, *Std. Dev.*, *Coef.*, *Q1* and *Q3* represents *standard deviation*, *coefficient of variation*, *first quartile* and *third quartile* respectively. As we can see, the skewness of sampling distributions for all the tasks are right (positive) skewed (i.e., the numerical representation of tasks' skewness are positive; in the view of graph, the sampling distribution has relatively few high values, and the mass of the distributions is concentrated on the left of the figure). Therefore, we add the *five-number summary* introduced in [3] consisting of *Min*, *Q1*, *Median*, *Q3* and *Max* to Table 1. Further, in order to have an overview statistic description of these sampling distributions, we include some common numerical descriptions as well, such as *Mean*, *Standard deviation*.

Table 1. Descriptive statistics of the tasks in the system model M1 used in the evaluation.

	<i>DRIVE</i>	<i>IO</i>	<i>CTRL</i>	<i>PLAN</i>
<i>Samples</i>	199994	400000	199990	199988
<i>Mean</i>	222.08	125.0	1967.3	2002.9
<i>Std. Dev</i>	14.291	45.576	390.09	412.46
<i>Coef.</i>	0.06435	0.36461	0.19829	0.20593
<i>Skewness</i>	6.7334	0.00128	0.38184	7.0644
<i>Min</i>	220	0	1074	332
<i>Q1</i>	220	100	1594	1631
<i>Median</i>	220	125	1919	1931
<i>Q3</i>	220	150	2339	2376
<i>Max</i>	420	250	6954	45957

Using Non-parametric Test

By using a conventional statistical procedure (*parametric test*), e.g., t-test, analysis of variance (ANOVA), the underline assumptions are: E.g., in ANOVA, firstly, each individual at the sampling distribution is supposed to be independent. While in our case, due to intricate task execution de-

dependencies in the system, an upcoming RT data may not be independent with the RT data previously recorded because the value of a Globally Shared State Variable (GSSV) can impact the selection of control branches for the upcoming RT data; Secondly, in both t-test and ANOVA, the underline population is assumed to follow a normal distribution. However, the sampling distribution of RT data of the CTRL task (i.e. the task on focus) is a multimodal distribution having several peaks, when the number of samples is large enough i.e. 199 990 (refer to row `Samples` for the CTRL task in Table 1). Further, by using a commercial statistic analysis software *EasyFit*, according to the results given by a Goodness of Fit (GOF) test, i.e., Chi-square test at α -value of 0.05¹, the obtained results clarify that the sampling distributions of RT data of all the tasks cannot fit to any of the 65 known distributions, e.g. Normal, Uniform, Student's t, Lognormal. Further, in t-test, the mean value μ_0 of the population has to be known beforehand, which is not the fact in this work. Therefore, parametric test cannot be applied in this work. Instead, we use the two sample Kolmogorov-Smirnov [4] (hereafter KS test) which is non-parametric and makes no assumptions on the underline population of a sampling distribution.

4 Algorithm

Simple Random Sample

In order to eliminate bias on the sampling, which is a key issue of selecting samples from the population of all individuals concerning the desired information, the technique of simple random samples (SRS) [3] is adopted. SRS gives every possible sample of a given size the same chance to be chosen. For instance, Monte Carlo simulation is used as a way of implementing SRS to collect sampling distributions of RT data of tasks in the extracted RTSSim model, in terms of giving each possible input to the RTSSim simulator an equal chance to be chosen, e.g., task jitter, number of external events generated by sensors etc. This is done by an embedded random number generator `rnd_inst()`² in the RTSSim simulator, which is an improved version of the Pseudo-random number generator used in C. Moreover, empirical results showed that the distribution of random numbers given by `rnd_inst()` is conforming to the uniform distribution, which assures that for each selection in RTSSim input data, all possible values in any range are equally likely to be chosen. Analogously, the sampling distributions of RT data of tasks in the real system can be collected based on measurements given a randomized system input. Some

¹ $\alpha = 0.05$ means that we are requiring that the RT data of tasks give evidence against H_0 so strong that it would happen no more than 5% of the time when H_0 is true.

²For the sake of space, the detailed implementation of `rnd_inst()` will not be given in this paper.

of the outliers (extreme values) which are caused by errors, e.g., hardware failure, have to be removed from the sampling distributions.

StatiVal

The proposed method, *StatiVal*, is shown in Algorithm 1. The algorithm returns the result concerning if the modeled system S and the model S' are statistically significantly. Further, τ_* is the task on focus. The outline of *StatiVal* is as follows:

1. Construct the sampling distribution of n RT data of all the tasks in both the system S and the model S' by using the randomized measurement `RandMeasure()` and Monte Carlo simulation `MonteCarlo()` respectively (refer to lines 1 to 6 in Algorithm 1).
2. Use KS test to compare if the sampling distribution of RT data of task τ_* in both S and S' are statistically significant. If the result is positive, then Algorithm 1 draws the conclusion C_1 , i.e., *the model S' is not a sufficiently accurate approximation of the system S due to an improper model extraction process*, and finally, stops the entire validation process (refer to lines 7 to 8 in Algorithm 1). Otherwise, *StatiVal* goes to Step 3.
3. Remove τ_* which has been already evaluated in Step 2 from the task set Γ in both S and S' , then iteratively compare sampling distributions of the remaining tasks in both S and S' by using KS test, unless a positive answer is given; Otherwise, *StatiVal* will draw the conclusion C_0 , i.e., *the model S' is a sufficiently accurate approximation of the system S after a proper model extraction process* (refer to lines 9 to 16 in Algorithm 1).

KS test is conducted by using a commercial software *XLSTAT*, which is a plug-in to EXCEL. Moreover, the number of data to handle by XLSTAT is practically dependent on the number of data dealt by EXCEL. In our experimental computer, due to that EXCEL 2003 is in use, the upper bound of the number of data in KS test is 65 535 (i.e. $2^{16} - 1$). Correspondingly, the upper bound of the number of samples in sampling distributions of all the tasks in both the system S and the model S' in Algorithm 1 is 60 000; Otherwise, the extra samples will be discarded in KS test unless EXCEL 2007 is used. Note that more investigation on optimizing the number of samples in the sampling distributions of RT data of tasks in S and S' will be given as the part of the future work.

5 Evaluation

Currently, we are not able to perform the model validation process concerning the extracted model and a real system (or a modeled system). Therefore, in this work, we examine the idea by using a fictive model, i.e., Model 1 (M1). M1 has similar architecture and analysis problems

Algorithm 1 *StatiVal*(Γ, Γ')

```

1: for all  $\tau_i$  such that  $1 \leq i \leq m$  in  $\Gamma$  in  $S$  do
2:    $X_{\tau_i} \leftarrow x_{i,1}, \dots, x_{i,j}, \dots, x_{i,m} \leftarrow \text{RandMeasure}(n, rt, \text{rnd\_inst}())$ 
3: end for
4: for all  $\tau'_i$  such that  $1 \leq i \leq m$  in  $\Gamma'$  in  $S'$  do
5:    $X'_{\tau'_i} \leftarrow x'_{i,1}, \dots, x'_{i,j}, \dots, x'_{i,m} \leftarrow \text{MonteCarlo}(n, rt, \text{rnd\_inst}())$ 
6: end for
7:  $ret \leftarrow 0$ 
8:  $ret \leftarrow \text{kstest}(X_{\tau_*}, X'_{\tau'_*}, \alpha)$ 
9: if  $ret = 0$  then
10:   $\Gamma \leftarrow \text{remove}(\Gamma, \tau_*)$ 
11:   $\Gamma' \leftarrow \text{remove}(\Gamma', \tau'_*)$ 
12:  while  $k < m - 1$  and  $ret = 0$  do
13:     $ret \leftarrow \text{kstest}(X_{\tau_i}, X'_{\tau'_i}, \alpha)$ 
14:  end while
15: end if
16: return  $ret$ 

```

as an industrial real-time application in use at ABB. Although FPPS is used as base, in M1, one task changes its priority during runtime. The task model is presented in Table 2. The outcome of the simulation model M1 in which there are no changes applied, is used as the outcome of the real system, and then we change M1 regarding the following three change scenarios which are initially introduced in [9] and outlined in Table 3. Finally, we compare the outputs against the original model to investigate the performance of the method. If the result returned by StatiVal is negative (NO), then the conclusion C_0 is given; Otherwise, C_1 is obtained.

Table 2. Tasks and task parameters for M1. The lower numbered priority is more significant, i.e., 0 stands for the highest priority.

Task	Period (μs)	Offset (μs)	Priority
DRIVE	2000	12000	1
IO	5000	500	3
CTRL	10000 or 20000	0	4 or 2
PLAN	40000	0	5

Table 3. Results obtained by using StatiVal concerning different models.

Change Scenarios	Changes Description	Results of StatiVal
Case 1	PLAN: Prio = 6	NO (C_0)
Case 2	PLAN: Prio = 2	YES (C_1)
Case 3	DUMMY: Prio = 6, T = 5000, C = 25000	NO (C_0)

As shown in Table 3, StatiVal considers the results of validation between the system and the models in Case 1 and Case 3 as negative (i.e. NO). Whilst for case 2, the corresponding result given by StatiVal is positive (i.e. YES), which shows that the system and the model are statistically different. Moreover, the validation results are in line with the results in [9], which used the same evaluation model and change scenarios but using a different validation method.

6 Conclusions and Future Work

This paper has presented ongoing work on validation of temporal simulation models extracted from real industrial control systems containing intricate task execution dependencies. In particular, we have presented and validated the method by using a fictive system model inspired by a real system, which shows that the proposed method has the potential to identify differences between the system and the extracted models. As part of future work, an effort will be spent on evaluating more scenario changes on the fictive model. Moreover, the possibility of evaluating the proposed method on real systems will be explored.

References

- [1] J. Kraft, J. Huselius, A. Wall, and C. Norström, "Extracting simulation models from complex embedded real-time systems," in *Real-Time in Sweden 2007*, August 2007.
- [2] A. M. Law, "How to build valid and credible simulation models," in *WSC '08. Winter Simulation Conference*, 2008, pp. 39–47.
- [3] D. S. Moore, G. P. McCabe, and B. A. Craig, *Introduction to the practice of statistics*, 6th ed. New York, NY 10010: W. H. Freeman and Company, 2009.
- [4] A. M. Law and D. M. Kelton, *Simulation Modeling and Analysis*. McGraw-Hill Higher Education, 1999.
- [5] J. Kraft, "RTSSim - A Simulation Framework for Complex Embedded Systems," Mälardalen University, Technical Report, March 2009.
- [6] O. Balci, "How to assess the acceptability and credibility of simulation results," in *WSC '89*. New York, NY, USA: ACM, 1989, pp. 62–71.
- [7] J. Andersson, A. Wall, and C. Norström, "Validating temporal behavior models of complex real-time systems," in *SERPS'04*, September 2004.
- [8] J. Huselius, J. Andersson, H. Hansson, and S. Punnekkat, "Automatic generation and validation of models of legacy software," in *RTCSA '06*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 342–349.
- [9] F. Nemati, J. Kraft, and C. Norström, "Validation of temporal simulation models of complex real-time systems," in *CORCS'08*, July 2008.