VisibleSim Manual

Julien Bourgeois, Benoît Piranda, Thadeu Knychala Tucci, André Naz

April 25, 2015

Contents

1 Introduction

VisibleSim is a general discrete event simulator (DES) for modular robot systems.

2 Installation

3 User applications in VisibleSim

3.1 Examples of applications

3.2 Implementing a new application

3.3 Running an application

3.3.1 C++ application

3.3.2 Meld application

3.3.3 Command line arguments

4 Embedded debugger

5 Local clock Simulation

VisibleSim supports local clock simulation. This section details the clock model used by the simulator. The programming API is available in VisibleSim Doxygen documentation.

5.1 Systematic model for clocks

[1] proposes a general model for oscillators:

$$x(t) = x_0 + y_0 t + \frac{1}{2} D t^2 + \epsilon(t) \tag{1}$$

where t is the simulation time (real-time), $x(t)$ is the local time, x_0 is the time offset, y_0 is the frequency offset, D is the frequency drift and $\epsilon(t)$ is the random noise. $\epsilon(t)$ is a not deterministic. [3] suggests to measure the Power Spectral Density (PSD) of the noise $S_y(f)$ and to design a filter with the frequency response $H(f)$ to simulate the oscillator noise.

5.2 Model parameter estimation through experiments

We used hardware Blinky Blocks to compute realistic clock models. Blinky Blocks are equipped with a micro-controller ATxmega256A3 that holds a 16-bit Real Time Counter (RTC). The RTC can be plugged to different oscillators. We choose to study clock behavior using the most precise internal oscillator available: a 32.768 kHz (actually divided down to 1.024 kHz) calibrated RC oscillator with a precision of 1% (at 3V and 25°C) and a resolution of 1 ms[1].

Typical RC oscillators are cheap, but generally suffer from poor accuracy over temperature and supply voltage. Moreover, they generally show important variations of nominal output frequency[2].

We conducted experiments on hardware Blinky Blocks in order to compute model parameters. We used a system composed of 6 blocks to collect time reference points $< t, x^i(t) >$ with i the block unique identifier every 5 seconds during 24 hours. The real-time t was provided by a computer. We assume the computer clock to be perfect. The set of blocks were powered with 5V and 0.36A. We observed that the behavior of the clock of a block varies slightly from an execution to another. The variations seems independent from the location of the block in the system, from the distance to the block plugged to the power supply, and from the distance to the block that receives the real-time t. Figure 1 shows the clock drift and the noise $\epsilon(t)$ distribution. We assume that Blinky Blocks clocks follow (1)'s model. We decided to model noise using a normal law $\mathcal{N}(0, (residual\ standard\ error)^2)$. Figure 5.2 shows experimental parameter values using polynomial regression.

Figure 3 shows the distribution of the parameter values. The parameters D, y_0 and the residual standard error seems normally distributed. As a consequence, we randomly generate clock parameters according to normal laws with the corresponding mean and standard deviation.

[1] see AVR1003: Using the XMEGA^{TM} Clock System (`http://www.atmel.com/images/doc8072.pdf`) for more details.

[2] see `http://www.maximintegrated.com/en/app-notes/index.mvp/id/2154` for more details.

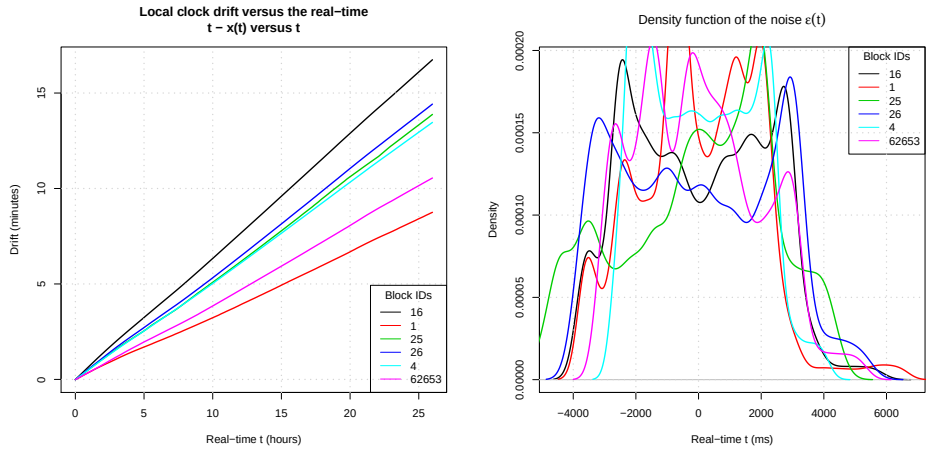

Figure 1: Local clock drift (t - x(t)) and noise ($\epsilon(t)$) distribution.

Parameter	Min	Mean	Max	Standard-deviation
D (ms^{-2})	-1.613992e-11	-1.179717e-11	-7.991859e-12	3.060884e-12
y_0 (ms^{-1})	0.9896537	0.9922277	0.9949096	0.001851285
x_0 (ms)	-5984.141	-3532.051	-785.9812	1921.629
Residual standard error (ms)	1688.103	2080.197	2423.646	294.832

Figure 2: Parameters

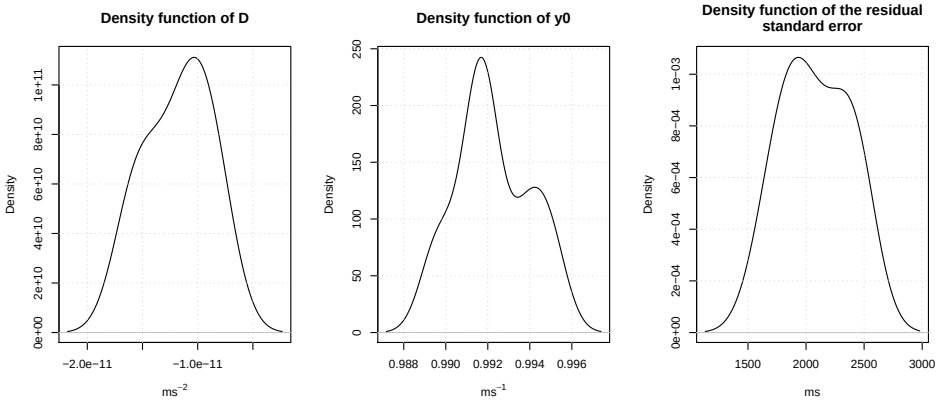

Figure 3: Parameter distributions.

5.3 Clock simulation in DES

[3] explains how to enhance DES with efficient local clock simulation. For a given simulation and a given block i, Points in time need to be consistent:

$$t_2 < t_1 \iff x^i(t_2) \leq x^i(t_1) \tag{2}$$

$$t_2 = t_1 \iff x^i(t_2) = x^i(t_1) \tag{3}$$

$$t_2 > t_1 \iff x^i(t_2) \geq x^i(t_1) \tag{4}$$

Consistency equations (2) (3) (4) are obvious but not trivial to ensure when considering noise simulation. For example, if we consider that $t_2 = t_1 + 1$, $D > 0$, $y_0 > 0$, $\epsilon(t_1) > 0$ and $\epsilon(t_2) < 0$, then $x(t_2) < x(t_1)$ which is inconsistent.

[3] proposes to use a caching strategy by storing the last calculated simulation times and local times. These reference points are then used to ensure time consistency. We use that approach in VisibleSim.

References

[1] David W Allan. Time and frequency(time-domain) characterization, estimation, and prediction of precision clocks and oscillators. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 34(6):647–654, 1987.

[2] Liangping Ma, Hua Zhu, Gayathri Nallamothu, Bo Ryu, and Heidi Howard. Understanding linear regression for wireless sensor network time synchronization. In *Proceedings of the 2007 International Conference on Wireless Networks, June 25-28, 2007, Las Vegas, Nevada, USA*, pages 325–328, 2007.

[3] Felix Ring, Anetta Nagy, Georg Gaderer, and Patrick Loschmidt. Clock synchronization simulation for wireless sensor networks. In *Sensors, 2010 IEEE*, pages 2022–2026. IEEE, 2010.