Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 48-1 (2015) 280–285

Effects of Event-Free Noise Signals on Continuous-Time Simulation Performance

Francisscus L. J. van der Linden * **Andreas Klöckner** *
Dirk Zimmer *

* *DLR German Aerospace Center, Institute of System Dynamics and Control, 82234 Weßling, Germany*
(e-mail: franciscus.vanderlinden@dlr.de).

Abstract: Generating stochastic input signals such as noise in physical systems is traditionally implemented using discrete random number generators based on discrete time-events. Within the Modelica community, time-event free random number generators have recently been proposed in order to increase the performance of system simulations. However, the impact of such signals on commonly used solvers, such as DASSL or Radau IIA, is still under discussion. In order to provide better understanding for modeling practitioners, we examine the influence of event-free noise models on simulation performance. To this end, we conduct practical simulation experiments with systems of three sizes, two solvers, and different parameters. Results indicate that step-size control can handle event-free noise generators well and that they outperform sampled generators. The findings can be related to other time-dependent system inputs.

Keywords: Numerical simulation, Random number generators, Continuous time systems, Integrators, Software performance

1. INTRODUCTION

Noise or other stochastic input signals are omnipresent in realistic system simulation. Adding noise to a nominal system simulation is especially important for assessing a system's performance or to evaluate a controller's properties. However, simulation of natural fluctuations is not limited to control design, but also applies to various other fields such as aircraft airworthiness requirements (e.g. EASA, 2007), estimating power outcomes of wind energy farms (e.g. Justus et al., 1976), or interpretation of experimental sensor readings (e.g. Márton and van der Linden, 2012).

Typical noise generators are discrete-time processes, relying on recursively perturbing an internal state. Each perturbation of this state is represented by a time-event in the simulation. The high frequency of typical noise signals thus causes a high number of time-events. This results in small step-sizes for the ODE solver and consequently high computational cost. See e.g. Felgner and Frey (2010), where the influence of different solvers is investigated on continuous, stiff and hybrid systems.

Most modelers today use robust ODE or DAE solvers that are suitable for highly stiff systems such as DASSL (Petzold, 1982) or Radau IIA (see e.g. Hairer and Wanner, 1996). Especially multi-step methods like DASSL suffer from the large number of time-events since the restart at each time-event is computationally expensive (see e.g. Lundvall and Fritzson, 2005). But also for implicit Runge-Kutta method as Radau IIA, the enforced step-sizes are often much lower than what would be required for the demanded precision.

Recent work therefore proposes to generate event-free continuous-time noise signals (Klöckner et al., 2014). The signals are generated directly as a function of time. This eliminates the need to generate events. Instead, it puts the step-size control of the ODE solver in charge. However, the performance impact of such signals on the ODE solvers is not yet fully understood. The general proposition claimed is that step-size control will handle the influences of such signals reasonably well, if suitably smooth interpolation functions are used and the frequency content is bounded. In this case, the polynomial approximations used for error estimations should work adequately.

Here, we investigate the effects of such event-free noise on the integrator accuracy and cost (i.e. number of function calls and run-time). Our expectations are that (a) sampled noise introduces a relatively constant cost for all accuracies due to the step-size being limited by event instances, that (b) variable step-size integrators can indeed handle event-free noise signals by selecting suitable step-sizes, that (c) event-free noise outperforms sampled noise for low accuracies by allowing larger step-sizes, that (d) smooth interpolations further decrease the cost of noise simulation.

1. We thus first introduce sampled and event-free noise signals as used in this study in Sec. 2.
2. The influence of the noise signals on a simple integrator model's performance is then compared as a function of the desired accuracy in Sec. 3.
3. The example model is extended to a critical damping with 50 states in Sec. 3.2.
4. We finally show the influence of the noise amplitude relative to the system states in Sec. 4.

Although we use noise signals in this work, the results are also relevant for other types of signals. These include e.g. interpolation tables or sine waves, as long as the signals

10.1016/j.ifacol.2015.05.039

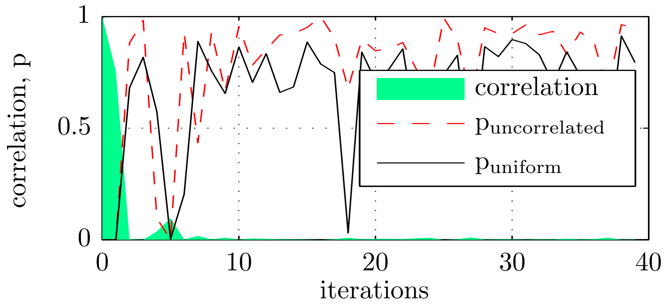


Fig. 1. A high diffusion capacity allows to retrieve random numbers from an algorithm after a few iterations. Here, ten steps allow to recover from a bad seed (see Klöckner et al., 2014, Fig. 3).

are generated directly as a function of the time rather than of the system states.

2. RANDOM SIGNALS

For this work, we use the Modelica Noise library (Klöckner et al., 2014). It allows to modularly compose a random number generator, a probability density function, and an interpolation function for the noise signal. The signal is then readily available for complex multi-physics simulations built on the modeling language Modelica.

Several standard sampled random number generators are provided, which all make use of a discrete-time state vector $\boldsymbol{s}$. The model generates events every Δt seconds and iterates the state vector from $\boldsymbol{s}^{\mathrm{pre}}$ to $\boldsymbol{s}^{\mathrm{new}}$ in order to yield a new random value r:

$$\begin{aligned} \boldsymbol{s}^{\mathrm{new}} &= f(\boldsymbol{s}^{\mathrm{pre}}), \\ r &= g(\boldsymbol{s}^{\mathrm{new}}). \end{aligned} \tag{1}$$

The library additionally introduces a new, continuous-time type of random number generator: DIRCS Immediate Random with Continuous Seed (DIRCS). It relies heavily on the diffusion capacity of certain random number generators: They deliver high-quality random numbers after a few iterations of the algorithm on a poor, non-random seed. Simple generators recover reasonably well after a few steps (see Fig. 1). This ability is exploited by seeding the random number generator with a simple function of time, such as shown in Eq. 2. The approach completely eliminates the need for discrete states in the noise model.

$$\textbf{int}\ \mathrm{s}\,[2]\ = (\textbf{int}*)\ (\&\mathrm{time}); \tag{2}$$

In this work, we use uniformly distributed random numbers generated by a simple, multiple recursive generator with the two states s_1 and s_2. The same generator is used for the discrete-time algorithm as well as within the DIRCS algorithm in order to yield comparable results in terms of run-time. The quality of the random number is not of interest in this study. The algorithm used is given in Eq. 3. The parameters are heuristically chosen to be $a_i = 134775813$ and $c = 2147483629$.

$$\begin{aligned} s_1^{\mathrm{new}} &= \sum a_i \cdot s_i^{\mathrm{pre}} + 1 \mod c \\ s_2^{\mathrm{new}} &= s_1^{\mathrm{pre}} \\ r &= s_1^{\mathrm{new}} / c \end{aligned} \tag{3}$$

The library also provides three different types of interpolation: The first implements a sample-and-hold behavior, the second is a linear interpolation, and the third applies a smooth interpolation using the sinc function as kernel. The three interpolations are shown conceptually in Fig. 2. The sinc interpolation has very good low-pass characteristics. In this study, we use all three interpolation functions in order to compare the effect of the interpolation's smoothness on the solver performance. Note that all interpolations can be used with the sampled as well as the sample-free method.

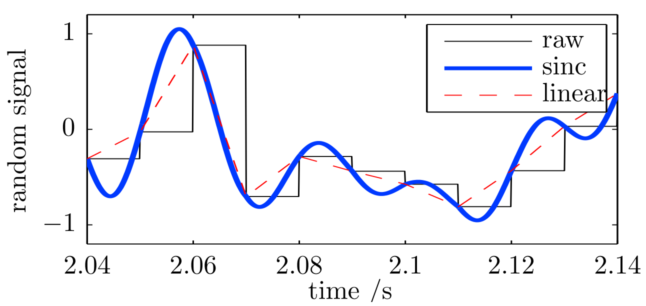


Fig. 2. This study uses the "raw" sample-and-hold noise signal (i.e. no interpolation), the "linear" interpolation, and the continuous "sinc" interpolation (see Klöckner et al., 2014, Fig. 4).

3. EFFECTS ON SOLVER PERFORMANCE

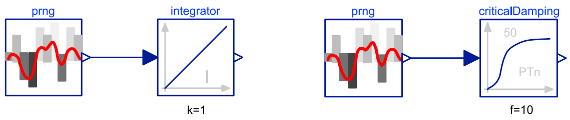
In this section we study the behavior of two commonly used solvers (DASSL & Radau IIA order 5) and the influence of interpolation of the event-free noise signal on the simulation.

To study these effects, two systems have been analyzed: A trivial system with one state and a larger system with 50 states. These systems will be studied in the following sections. The proposed systems are simulated using the DASSL and Radau IIA order 5 solvers implemented in Dymola 2015 on a Windows based computer (Intel Xeon E5-1620, 16GB ram). The influence of the solver accuracy on the number of function evaluations and the simulation time is assessed. The systems are simulated 5000 seconds to minimize the influence of initialization effects.

3.1 Single state integrator system

To study the effects of event-free noise on a simple example, a system is set up using Dymola combining a noise generator and an integrator (see Figure 3a). This system represents a simple model with only a single state. The noise is configured to produce a uniform noise on the interval [-1e-3,1e-3].

In the top two diagrams of Figure 4, the amount of evaluations of the function $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, t)$ and the computational



(a) simple intgrator (b) critical damping

Fig. 3. Noise generator coupled to two systems: A simple integrator as a trivial system with one state and a critical damping as a system with 50 states.