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

Franciscus 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

2405-8963 © 2015, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.05.039

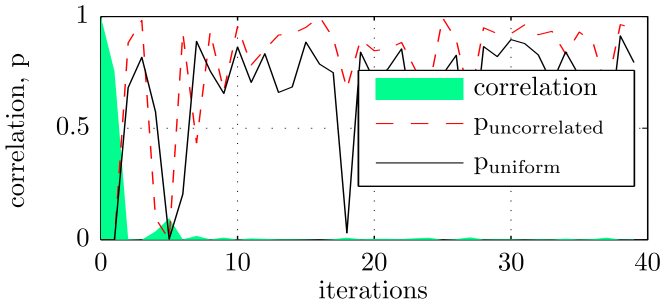


Fig. 1. A high diffusion capacity allows to retrieve random numbers from an algorithm after a few iterations. Here, ten steps allow to recover from a bad seed (see Klöckner et al., 2014, Fig. 3).

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.

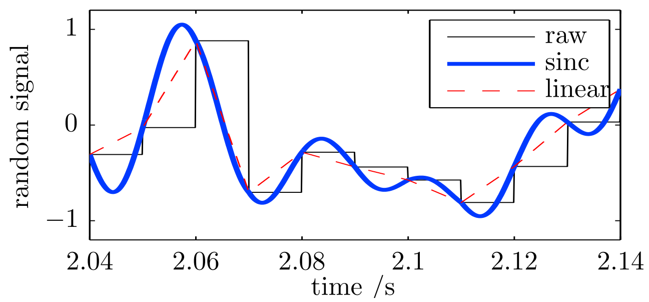


Fig. 2. This study uses the "raw" sample-and-hold noise signal (i.e. no interpolation), the "linear" interpolation, and the continuous "sinc" interpolation (see Klöckner et al., 2014, Fig. 4).

3. EFFECTS ON SOLVER PERFORMANCE

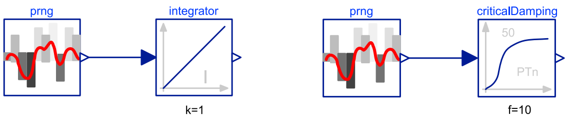
In this section we study the behavior of two commonly used solvers (DASSL & Radau IIA order 5) and the influence of interpolation of the event-free noise signal on the simulation.

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.

time to simulate the system using the DASSL solver is shown. As expected, the computational effort for the sampled method is almost constant, i.e. it is independent of the demanded integrator accuracy. The sample free DIRCS method needs less computational effort than sampled methods for loose tolerances. The effort needed for the sample-free noise signal approaches and eventually exceeds the effort for sampled noise for very tight tolerances. This effect can be explained as follows:

- The sampled system halts the solver at each sample, independent of the tolerance and restarts the calculation. This leads to an almost unchanged simulation effort.
- As long as the error tolerance of the solver is above the amplitude of the noise, the solver can neglect the noisy input of the event-free system. This leads to a reduction of simulation effort.
- At tight tolerances, the solver will try to exactly follow the noise signal. However, when this signal contains discrete steps in its value or in its derivatives, the polynomial-based error estimations of the solver perform badly. For non-interpolated signals, a restart at the time-event is then the better strategy, leading to slightly less computational effort for sampled systems.

A higher order interpolation routine leads to a reduction in function evaluations for the event-free methods. The smoother signal makes it easier for the solver to follow the event-free noisy signal and can therefore reduce the amount of steps. The interpolation function itself though can increase the costs per function evaluation. The results of the simulation using the Radau IIA solver can be seen in the bottom two diagrams of Figure 4. The Radau IIA solver is especially suitable for stiff systems. Since systems with noise are almost always stiff due to the combination of high frequency noise on a low bandwith system. The results from this test show that the Radau IIA solver has a similar behaviour as the DASSL solver. Since the tolerances of both solvers cannot be directly compared, no conclusions about the performance can be made.

3.2 Critical damping system with 50 states

To investigate the effect of noise on nontrivial systems, an example model has been created. A critical damping block with a cut-off frequency of 10 Hz of order 50 is coupled to the noise generator. This yields a system with 50 states. The critical damping block has following transfer function:

$$y = \frac{1}{\left(\frac{s}{\omega} + 1\right)^n} u \tag{4}$$

with $\alpha = \sqrt{2^{\left(\frac{1}{n}\right)} - 1}$ and $\omega = 2\pi\frac{f}{\alpha}$. Here f is the cut off frequency in Hz and n the system order. The set up of the system can be seen in Figure 3b It is expected that in a system with 50 states, a solver restart will have a relatively heavy penalty on the simulation performance. A better interpolated signal is therefore expected to have a positive influence on the performance of the solvers.

The results from the system with 50 states using the DASSL solver with the same setup as used before can be seen in the top two diagrams of Figure 5. The penalty on a higher interpolation routine is not as high as in the simple

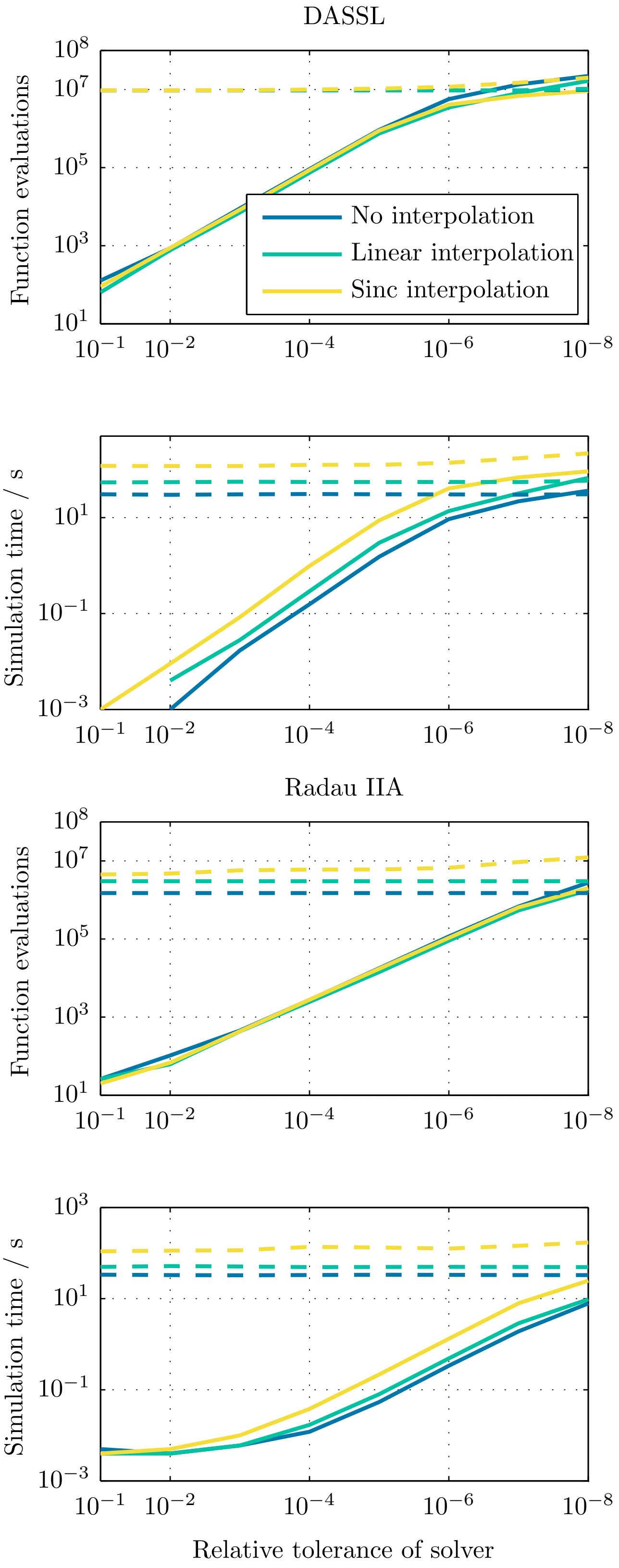


Fig. 4. Function evaluations and simulation time as a function of the integrator tolerance for a simple system and different interpolation methods. Dashed lines represent the sample based method, solid lines the sample free DIRCS method. Sample-free noise reduces computational costs, especially with loose tolerances.

system. When observing the CPU-time for the integration, it becomes clear that at higher integrator tolerances a better interpolation of the event-free noise leads to a better performance of the solver. The simulation time of the sinc interpolated event-free signal is always below the sampled methods. In most real-life systems, an event-free signal can therefore be preferred over a sampled signal. This is even true at very high system accuracies. The results from the same analysis as before using the Radau IIA solver can be seen in the bottom two diagrams of Figure 5. Using the Radau IIA solver, a similar pattern can be observed; the Sinc-interpolated, event-free method is for most tolerances the most suitable interpolation method. Using this combination, the method is always faster as all sampled methods. However, the advantage is minimal for very tight tolerances.

3.3 Error evaluation

Figure 6 shows the mean of the absolute difference between the simulations discussed in the previous subsections and a reference solution simulated with a relative solver tolerance of 1×10^{-9}. The measure is evaluated over 2 500 000 grid points of the simulations lasting 500 s with random numbers generated at 100 Hz. Zero error is limited to 1×10^{-10} for plotting. Note that the values can only be interpreted relative to each other as no meaningful scaling exist .

For the integrator system and unsampled noise, the error measure is linearly dependent on the demanded solver tolerance. This shows the convergence of the simulation using an event-free noise generator. The error using sampled noise is zero, if no interpolation is used; independent of the solver choice. Using the Radau IIA solver, the error is also zero for linear interpolation. As expected, the tolerance has almost no influence on the error of sampled noise, because the step-size is mainly controlled by the events generate by the noise model. Using the sinc interpolation results in medium error levels. This shows the influence of the interpolation complexity on the error level.

In the critical damping simulation, there is no setting with zero error. This system is sufficiently complex to require additional steps in between the events instances. Unless a certain tolerance is demanded, the error value is nearly constant. This is an indicator that the system cannot be sufficiently resolved using the specified tolerance. Using tighter tolerances, results in a kink of the curves and a linear dependency of the error level on the tolerance can again be observed. The system is sufficiently resolved with these tolerances and converges to the reference solution. There is no error advantage of sampled or unsampled noise generators in this region, if the DASSL solver is used. A similar effect is also observed with the integrator system and sampled sinc interpolation, using the DASSL solver.

The Radau IIA solver produces a similar error pattern as the DASSL solver for the critical damping system and unsampled noise. The kink in the curves is at tighter tolerances as compared to the DASSL solver. This corresponds well with the findings concerning function evaluations and simulation time. The sampled noise results in surprisingly high error levels, which cannot be explained in the scope of this paper.

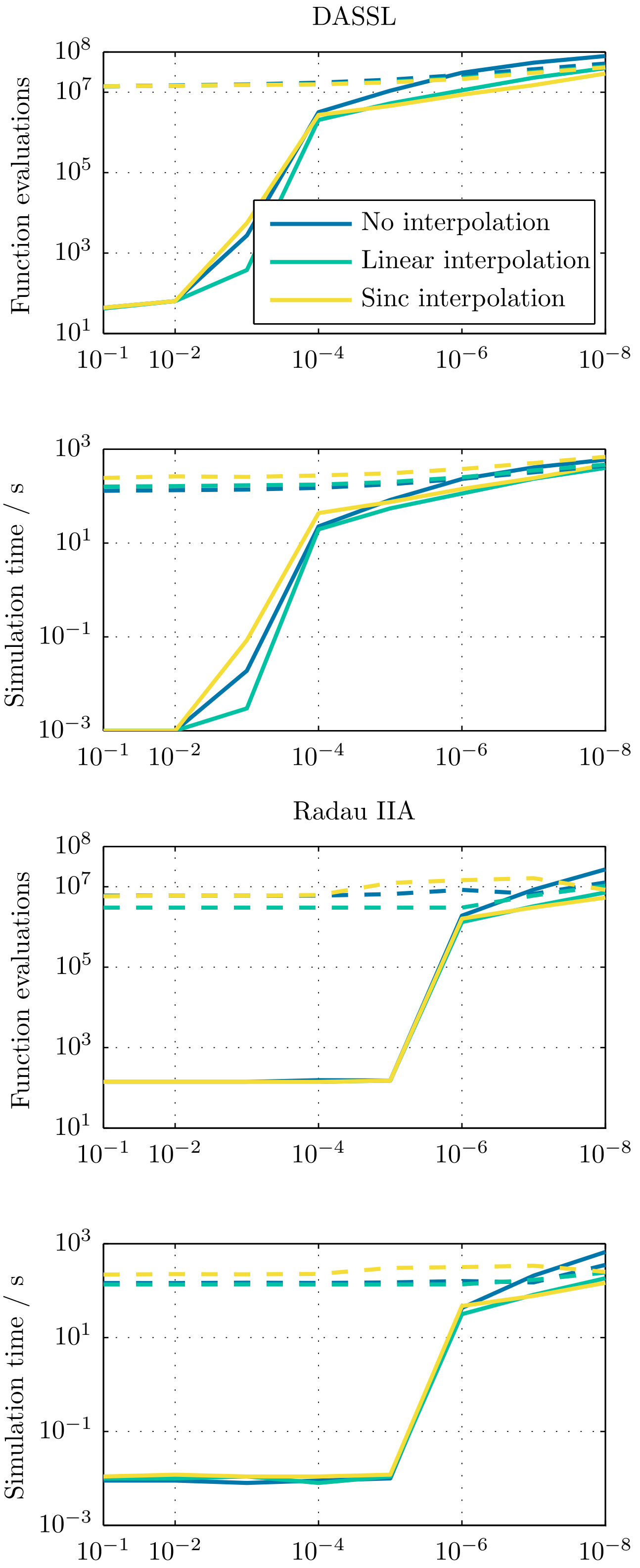


Fig. 5. Function evaluations and simulation time as a function of the integrator tolerance for a complex system and different interpolation methods. The findings from the simple system are confirmed. Dashed lines represent the sample based method, solid lines the sample free DIRCS method.

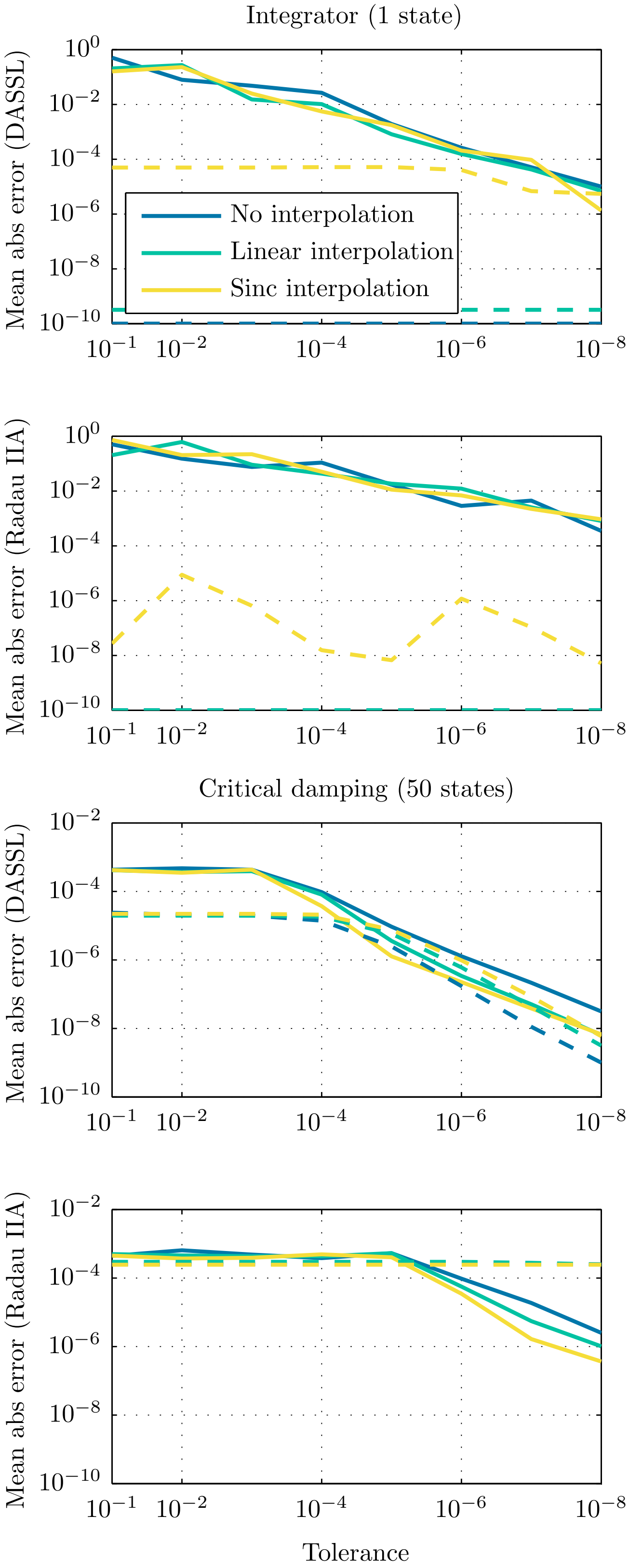


Fig. 6. Comparison of the mean absolute error to reference simulation (tol=1e-9 for 500 samples per second). Dashed lines represent the sample based method, solid lines the sample free DIRCS method. The sample-free methods converge to the reference solution.

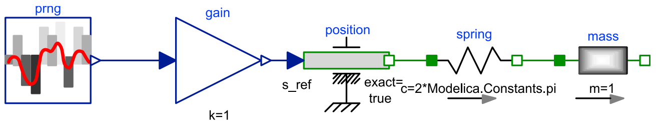


Fig. 7. An undamped spring-mass system is used to show the effects of noise on a physical system simulation.

4. EFFECTS ON A PHYSICAL SYSTEM

To investigate the effects of different noise levels on a physical system, an undamped spring-mass system is modeled using a mass of $m = 1\,\mathrm{kg}$ and a spring stiffness of $c = 2\pi\,\mathrm{N/m}$ (see Fig. 7). The system is perturbed by moving the fixes point of the spring with a normally distributed and smooth interpolated random variable u. The resulting linear system is given by Eq. 5. The system is initialized with a fixed starting position of $x_0 = 1\,\mathrm{m}$ and $\dot{x}_0 = 0\,\mathrm{m/s}$.

$$m \cdot \ddot{x} = -c \cdot (x + u) \tag{5}$$

The system is simulated using the DASSL and Radau IIA solvers with a fixed tolerance of 10^{-4} for a simulation time of 5000 s. The variance of the perturbation is manipulated between $\mathrm{Var}(u) = 10^{-8}\,\mathrm{m}$ and $1\,\mathrm{m}$ and the number of function evaluations and the simulation are recorded. Figure 8 shows the results for both the event-free random number generator and the sampled random number generator. Both solvers show similar dependencies of the performance on the demanded tolerance.

Using a sampled noise confirms the findings from the preceding sections. The number of function evaluations as well as the simulation time are nearly constant for all noise amplitudes. Both only increase at very high noise amplitudes in the order of the natural oscillation of the system, when using the DASSL solver.

An event-free noise generator decreases the number of function evaluations, as well as the simulation time by more than an order of magnitude for small disturbances

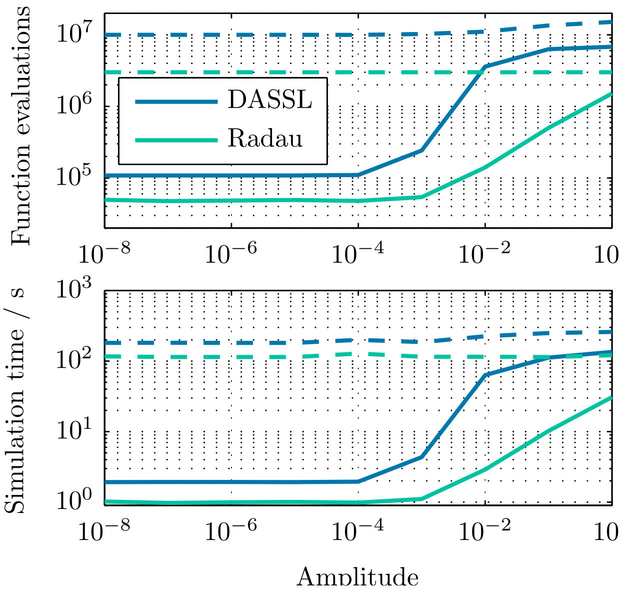


Fig. 8. Function evaluations as a function of the system perturbation. When the perturbation is smaller than the tolerance (10^{-4}), the sample-free noise is ignored.

below a threshold of approximately 10^{-4} m. This threshold correlates to the desired simulation accuracy of 10^{-4} times the natural oscillation amplitude of 1 m. Below the threshold, the system is unaffected by the noise. Above this threshold, the number of function evaluations as well as the simulation time increase with stronger disturbances. If the noise amplitude is in the same order of magnitude as the nominal system oscillation, the event-free simulation outperforms the model with sampled noise by roughly a factor of 2.

We have found that continuous-time noise does not deteriorate simulation performance, as long as the noise amplitude is below the solver accuracy. The solver naturally excludes irrelevant noise from a system simulation. The sampled generator strongly affects the simulation performance, even if the noise is irrelevant to the system's states. Additionally, even if the noise is relevant for the system simulation, an event-free and smooth noise signal improves simulation performance.

5. CONCLUSIONS AND DISCUSSION

In this paper we have investigated the influence of event-free noise on two standard solvers: DASSL and Radau IIA order 5. It is found that both investigated solvers show an almost constant computational effort for the simulation of event-based noise signals, independent of solver tolerance.

The assessed variable step size solvers can handle the event-free noise signals according to the tolerance settings of the solver: Using a tolerances larger than the noise amplitude, the solver mostly ignores the noise. If the tolerance is chosen below the amplitude of the signal, the solver used the event-free noise signal for the solution.

Event-free noise has a lower computational cost compared to the event-based noise at high tolerances for all analyzed models. For low tolerances, event-free noise also outperforms sampled noise, if non-trivial systems are assessed and if continuous interpolations such as the linear or the sinc interpolation are used.

Interpolating the noise signal avoids discontinuous signals and helps the solver to reduce the number of necessary integration steps. Depending on the effort for a single time step, compared to the effort of the interpolation, this can lead to lower or higher CPU times for the simulation. Large systems benefit of the lower amount of function calls, whereas in simple systems the penalty of the interpolation for each function call can be higher than the advantage of reducing the number of function calls. Since a linear interpolation has a low calculation effort, in most cases this leads to a lower simulation time. The sinc interpolation becomes interesting when the model is large and a function call poses high computational cost.

When using sample-free noise, care must be taken on how to choose the solver tolerances. When a high tolerance is selected, the solver will ignore the influence of the sample-free noise. Only if the tolerance is chosen appropriately, the effect of the noise becomes visible in the simulation results. This effect can be used to always include all noise sources in a physical system simulation: only the signals, which influence the behavior of the system, will lead to an increase in simulation times.

Care must also be taken, when heavy tailed distributions like the Cauchy distribution are used. Such distributions have a very large or undefined variance and might not be properly evaluated using a sample-free method, because the solver might not evaluate single large peaks. In this case an event-based method should be used to guarantee proper results.

REPRODUCIBLE RESEARCH

The results of this paper can be reproduced using the code available on `http://dlr-sr.github.io`.

ACKNOWLEDGEMENTS

We thank W. Manbir for his valuable input.

REFERENCES

EASA (2007). Certification specifications for large aeroplanes cs-25. amendment 3. Technical report, European Aviation Safety Agency.

Felgner, F. and Frey, G. (2010). Comparison of 4 numerical solvers for stiff and hybrid systems simulation. In *2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010)*, 1–8. IEEE. doi:10.1109/ETFA.2010.5641330.

Hairer, E. and Wanner, G. (1996). *Solving Ordinary Differential Equations II*, volume 14 of *Springer Series in Computational Mathematics*. Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/978-3-642-05221-7.

Justus, C.G., Hargraves, W.R., and Yalcin, A. (1976). Nationwide assessment of potential output from wind-powered generators. *J. Appl. Meteor.*, 15(7), 673–678. doi:10.1175/1520-0450(1976)015<0673:NAOPOF>2.0.CO;2.

Klöckner, A., van der Linden, F.L.J., and Zimmer, D. (2014). Noise Generation for Continuous System Simulation. In *Proceedings of the 10th International Modelica Conference*, 837–846. Modelica Association and Linköping University Electronic Press, Lund, Sweden. doi:10.3384/ECP14096837.

Lundvall, H. and Fritzson, P. (2005). Event handling in the OpenModelica compiler and run-time system. In *Proceedings of the 46th Conference on Simulation and Modelling of the Scandinavian Simulation Society (SIMS2005)*. Tapir Akademisk Forlag, Trondheim, Norway. URL `http://www.scansims.org/sims2005/SIMS2005_57.pdf`. ISBN 82-519-2093-0.

Márton, L. and van der Linden, F. (2012). Temperature dependent friction estimation: Application to lubricant health monitoring. *Mechatronics*, 22(8), 1078 – 1084. doi:10.1016/j.mechatronics.2012.09.003.

Petzold, L. (1982). Description of dassl: a differential/algebraic system solver. In *10th International Mathematics and Computers Simulation Congress on Systems Simulation and Scientific Computation*. Montreal, Canada. URL `http://www.osti.gov/scitech/servlets/purl/5882821`.