Noise Generation for Continuous System Simulation

Andreas Klöckner   Franciscus L. J. van der Linden   Dirk Zimmer
{andreas.kloeckner, franciscus.vanderlinden, dirk.zimmer}@dlr.de

German Aerospace Center (DLR), Institute of System Dynamics and Control,
82234 Weßling, Germany

The problem of introducing *disturbances to a nominal system* eventually becomes an issue, when simulating real-world systems. Especially, when dealing with controlled systems, important tasks are to check whether the controller is able to reject realistic disturbances, and to assess the performance of the system including noise. The problem is not limited to the field of control design, but is also of interest in e.g. specification of aircraft airworthiness requirements with respect to turbulence, estimation of the power outcome of wind energy farms or when interpreting contaminated sensor readings of experiments.

This noise contribution must be *specified carefully*, because it can have a strong impact on the system's performance. However, there are no convenient means of specifying noise properties in Modelica, such that typical approaches implement ad-hoc modifications of a simple random signal.

The *simulation performance* of the model should not be decreased by adding noise to the system. However, injecting noise into a continuous system typically decreases the simulation speed drastically, because standard noise generators are sampled systems, which generate time events by definition. These time events lead in most cases to integrator restarts, imposing a big penalty on simulation performance.

In this work, we present ways to solve the two issues outlined above in an integrated library:

1. We describe a general procedure to *specify a suitable noise signal* by means of selecting a high-quality random number generator, a probability distribution and a power spectrum.

2. By providing a *continuous noise* signal formulation using a sample-free generator and a smooth interpolation, we provide means for continuous noise generation. Avoiding events and using a smoothly filtered signal speeds up the simulation as compared to standard methods.

3. The methods and processes are integrated in a *library with convenient user interfaces*. This enables a user to easily specify a desired noise signal and to use it in complex simulation models.

Figure 1 shows the timing of an actuator simulation without noise, with conventional `LinearSystems` noise and with the new `Noise` library. The noisy simulation is relieved of unnecessary time events, using the `Noise` library. Its speed is increased by a factor of 2.5.

		Events	
Model	Time	Time	State
No Noise	0.037 s	1	12
`LinearSystems`	27.5 s	15000	1642
`Noise`	10.5 s	1	1791

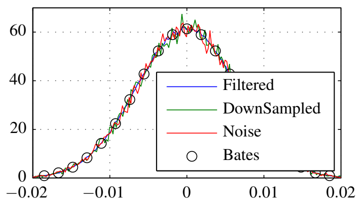

Figure 1: Timing of an actuator simulation and the distribution of a random variable