Stochastic DEQs and hybrid computing

1 Introduction

Stochastic differential equations are not only interesting object by themselves but also of great interest in financial mathematics and other fields.[1] This application note was inspired by [Huang et al. 2018] which focuses on the Black-Scholes stochastic differential equation[2] (*SDE*) of the general form

$$\dot{y} = \lambda y + \sigma y W \tag{1}$$

with some initial condition $y(0)$. W denotes a Wiener process[3], λ is a *drift parameter*, and σ is the *volatility*.

The goal of this application note is not to solve this particular SDE but to use it to demonstrate how a hybrid computer, i. e., an analog computer coupled with a digital computer, can be employed to treat problems like this in general. Accordingly, the Wiener process is implemented by an electronic noise generator without caring too much about its actual output signal's stochastic properties.[4] In the same manner the parameters λ and σ were chosen arbitrarily so that no overloads occured at the end of the individual computing runs.

Equation (1) can be directly tranformed into an analog computer setup as shown in figure 1. With a time scaling factor of $k_0 = 10^3$ set at the integrator, the noise source should have a bandwidth of 50 kHz to 100 kHz. Lower bandwidth noise requires smaller k_0 to avoid overloads, since the positive feedback λy is the dominating term and would result in an exponentially increasing output signal on its own.

[1] A great introduction can be found in [Evans 2013].
[2] See [Gronbach et al. 2012, pp. 110 ff.].
[3] See [Evans 2013, pp. 41 f.].
[4] Here, a HP3722A set to infinite sequence length with a Gaussian noise bandwidth ot 50 kHz at an amplitude of 3.16 V_{RMS} was used.

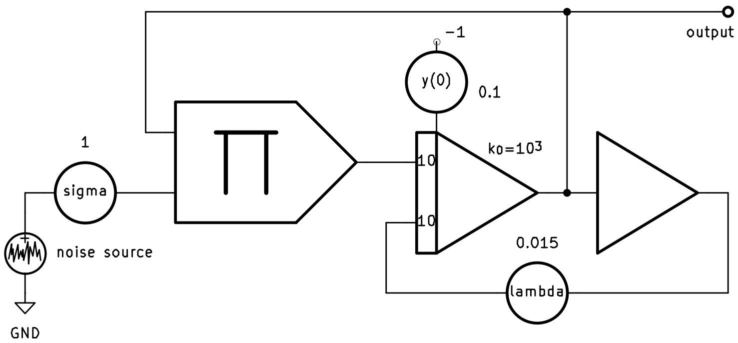

Figure 1: Analog computer setup for equation (1)

2 Hybrid computer setup

SDEs typically require many computer runs in order to gather enough data for later analysis. The idea is to perform a number (several thousand) IC/OP cycles on the analog computer reading the output value after the OP interval and storing it for later processing on the digital computer. In the following THE ANALOG THING[5] is controlled by an Arduino Mega 2560 using the `THAThc` library.[6] This library implements a number of methods that can be used to control the analog computer from the microcontroller, gathering data and sending it to an attached digital computer via USB. The corresponding control program is shown below. It can be directly compiled from within the Arduino IDE.[7] It requires the libraries `TimerThree`[8] and `TimerFive`[9] to be installed. In addition to this the `THAThc` library (see above) is required, too.

The overall program is shown in figure 2. First, `THAThc.h` must be included. `setup()` is pretty minimal and only intializes the serial line interface and instantiates a `THAThc` singleton. The main computations are done within the central `loop()`.

[5]See https://the-analog-thing.org.
[6]See https://github.com/anabrid/THAThc.
[7]See https://www.arduino.cc/en/software.
[8]See https://github.com/PaulStoffregen/TimerThree.
[9]See https://github.com/VincentLim/TimerFive.

```
#include "THAThc.h"
#define BAUD_RATE 250000

void setup() {
  Serial.begin(BAUD_RATE);
  THAThc.begin();
}

void loop() {
  THAThc.enable();                  // Configure the hybrid controller
  THAThc.set_ic_time(1);            // IC time is one millisecond
  THAThc.set_op_time(2);            // OP time is two milliseconds
  THAThc.set_channels(1);           // Only one ADC channel is used

  // Perform repetitive runs and sample result at the end of each run
  for (unsigned long i = 0; i < 10000; i++) {
    THAThc.single_run();            // Perform a single IC/OP cycle
    THAThc.block();                 // Block until this cycle finished

    float result;
    THAThc.sample_adc(&result);     // Read the value at the integrator's output.
    Serial.print(String(i) + "\t"); // Print number of run
    Serial.print(result, 3);        // and the corresponding result
    Serial.print("\n");
  }

  for( ; ; delay(1000));            // Stop
}
```

Figure 2: Control program

The result of each analog computer run is sent to the serial line and can be captured in the serial monitor of the Arduino IDE and copied to a file such as `test.dat`. `gnuplot`[10] a histogram can be plotted using the following commands within `gnuplot`:

```
binwidth=.01
bin(x,width)=width*floor(x/width)
plot '2.dat' using (bin($2,binwidth)):(1.0) smooth freq
```

A typical histogram is shown in figure 3.

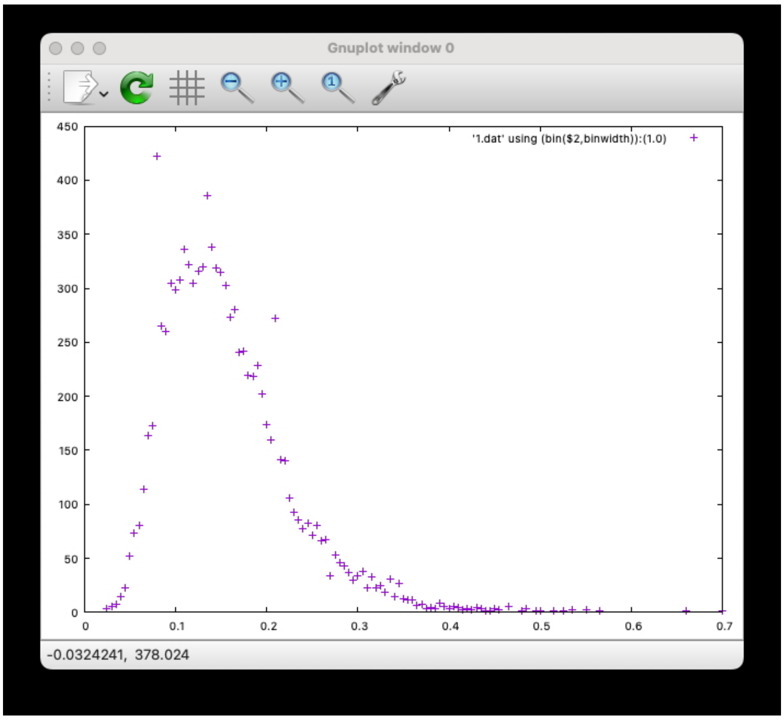

Figure 3: Histogram of the data generated by the hybrid computer setup

[10]See `https://gnuplot.sourceforge.net`.

Happy analog computing!

References

[Evans 2013] Lawrence C. Evans, *An Introduction to Stochastic Differential Equations*, American Mathematical Society, 2013

[Gronbach et al. 2012] Thomas Müller-Gronbach, Erich Nowak, Klaus Ritter, *Monte Carlo-Algorithmen*, Springer Verlag, 2012

[Huang et al. 2018] Yipeng Huang, Ning Guo, Simha Sethumadhavan, Mingoo Seok, Yannis Tsividis, "A Case Study in Analog Co-Processing for Solving Stochastic Differential Equations", 23rd DSP 2018, Shanghai, China