PYTHON Exercise I: Simulating a Stochastic Damped Harmonic Oscillator

Michel Verhaegen, Joeri Frederik and Tijmen van Oldenrijk

September 14, 2020

Delft University of Technology
Delft Center for Systems and Control
Mekelweg 2, NL-2628 CD Delft, The Netherlands
J.A.Frederik@tudelft.nl

Chapter 1

Simulating a Stochastic Damped Harmonic Oscillator: An Introduction

1.1 The Harmonic Oscillator

Physicist Sidney Coleman has said: "The career of a young theoretical physicist consists of treating the harmonic oscillator in ever-increasing levels of abstraction." The harmonic oscillator is a pervasive concept in theoretical physics, central in the first order approximation of any vibrating system, from the mass-spring system to quantum fields, but also in other parts of science, in engineering, and in music. In this PYTHON exercise we will simulate the stochastic damped harmonic oscillator in order to introduce in a practical way essential concepts of stochastic processes and the simulation of a stochastic differential equation. The theory is from the introductory text on stochastic processes in physics by Lemons [1].

The physical model we will study is that of an object attached to a spring and submerged in a viscous fluid, where the stochastic effect is caused by collisions with the molecules, so that the random aspect of the damping force is the same as the force that drives Brownian motion . The equation of motion for a deterministic harmonic oscillator ($F = ma$) is $-kx = m\frac{dv}{dt}$, with v the velocity of the object, which leads to an oscillation frequency $\omega = \sqrt{k/m}$. If we add an extra damping force $-\gamma v$, we can write

$$dv(t) = -\omega^2 x(t)dt - \gamma v(t)dt. \tag{1.1}$$

We introduce the force $\beta w(t)$, where $w(t)$ is normally distributed white noise (PYTHON command `randn`) with , and where thermal equilibrium stipulates (fluctuation-dissipation theorem) that $\beta^2/2\gamma = kT/m$, with T the fluid temperature. Thus the (simplest self-consistent, see p. 44 of [1]) 1-dimensional dynamics (in the x-direction) are described by

$$dv(t) = -\omega^2 x(t)dt - \gamma v(t)dt + \beta w(t)dt. \tag{1.2}$$

The variables in this equation are listed in Table 1.1. In order to simulate this process it is important how to generate the white noise sequence $w(t)$ in a sampled data setting. For that purpose we discuss physically realizable white noise sequences first. This is done in section 1.2.

Table 1.1: The quantities in the equation of motion (1.2).

Quantity	Meaning
m	mass of the object
γ	friction coefficient
$x(t)$	displacement of the object in the x-direction at time instant t
k_B	Botzmann contant
T	Temperature of the fluid
$w(t)$	normalized white noise: $\forall t, t' : E[w(t)] = 0 \;\; E[w(t)w(t')] = \delta(t - t')$, with $\delta(t)$ the dirac delta function

1.2 Bandlimited White noise

The quantity $w(t)$ in Table 1.1 is continuous white noise. If we consider the spectrum as the Fourier transform of the Auto-correlation function, the variance of white noise is *infinity*. As such white noise is physically not realizable. In practice the notion of *bandlimited white noise* is used. Such signal has the following spectrum:

$$P_w(\omega) = \begin{cases} \sigma & \text{for } |\omega| < \omega_0 \\ 0 & \text{otherwise} \end{cases} \tag{1.3}$$

The Auto-correlation function of a bandlimited white noise signal is:

$$r_w(\tau) = \int_{-\omega_0}^{\omega_0} \sigma e^{-j\omega\tau} d\omega = 2\sigma \frac{\sin \omega_0 \tau}{\tau} \tag{1.4}$$

We plot this Auto-correlation function for three values of ω_0 in Figure 1.1.

Since the limit $\lim_{t \to 0} \frac{\sin \omega_0 t}{t} = \omega_0$, it can be observed that the value $R_w(0)$ increases with ω_0. This means that the variance increases linearly with ω_0, i.e. the bandwith of the signal. In the limite as the bandwith goes to infinity we have,

$$\lim_{\omega_0 \to \infty} r_w(\tau) = 2\pi\sigma\delta(\tau)$$

For an example of physically generating bandlimited white noise we refer to Example 1.1.

Example 1.1 (Noise generated by an RC circuit.)

[3]] Consider an RC circuit with transfer function given as:

$$G(j\omega) = \frac{a}{\sqrt{\pi}(j\omega + a)}$$

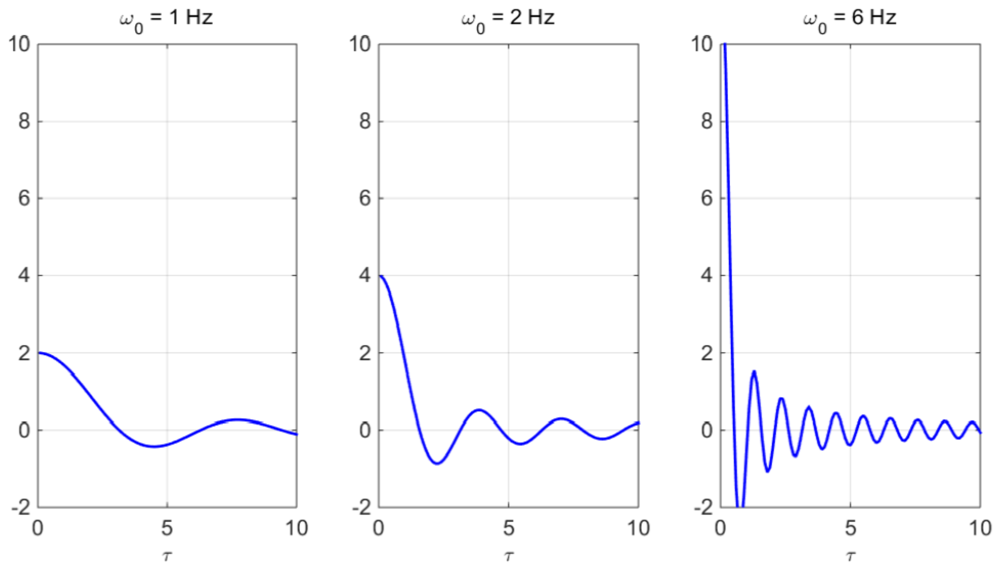

Figure 1.1: The Auto-Correlation function for three bandlimited white noise signals for ω_0 resp. equal to $1, 2, 6$ Hz.

The spectral density of white noise filtered by this transfer function is[1]:

$$P(\omega) = \frac{a^2}{\pi(\omega^2 + a^2)}$$

For $|\omega| < a$ the spectral density is essentially constant. The Auto-correlation function that result as the inverse Fourier transform of this spectral density equals:

$$r(\tau) = ae^{-a|\tau|}$$

Therefore we have the following limits:

$$\lim_{a \to \infty} P(\omega) = \frac{1}{\pi} \qquad \lim_{a \to \infty} r(\tau) = \delta(\tau)$$

That means that for increasing value of a the filtered white noise signal by the RC filter, will resemble more an more more the spectral properties of white noise. It will also have an increasing variance.

1.3 Simulating the equation of motion

As outlined in section 1.2 on band limited white noise, the variance of a physically realizable white noise signal increases with the bandwidth ω_0. In order to

[1]See (3.91) on page 101 of [8] for the definition (in the discrete case).

analyse the equation of motion (1.2) in a discrete time setting, we consider the use of a sampling period Δt sec.

When considering the Discrete Fourier Transform [8] the maximal frequency value that is obtained for a sampled data sequence with sampling periode Δt is proportional to $\frac{1}{\Delta t}$. Based on this fact and the increase of the variance of bandlimited (physically realizable) white noise with the bandwith, a discrete counterpart for the white noise signal in the equation of motion (1.2) is chosen as follows [7]. The discrete sequence of random numbers $\{\tilde{w}(k)\}$ that mimics the properties of $w(t)$ in (1.2) should have mean zero and variance that is equal to $\frac{1}{\Delta t}$. Such a sequence is called discrete white noise [8].

This concept can be used to replace white noise in discretizing different stochastic differential equations [7] and is illustrated in the following RC example.

Example 1.2 (Simulating the noise of the RC circuit of Example 1.1)

When consider the signal $y(t)$ to be generated by filtering the white noise signal $w(t)$ defined in Table 1.1 by the continuous time filter with transfer function:

$$G(s) = \frac{a}{\sqrt{\pi}(s+a)}$$

it should satisfy the stochastic differential equation:

$$\dot{y}(t) + ay(t) = \frac{a}{\sqrt{\pi}} w(t) \quad y(0) = y_0 \tag{1.5}$$

In order to avoid the use of advanced mathematical Ito Integrals, martingales, etc. we make use of the simplified (engineering) approach outlined in [7] to transform the stochastic differential into a difference equation. For that purpose a discrete white noise sequence $\tilde{w}(k)$ is generated with the following properties:

$$\forall k, \ell : E[\tilde{w}(k)] = 0 \quad E[\tilde{w}(k)\tilde{w}(\ell)] = \left\{ \begin{array}{ll} \frac{1}{\Delta t} & k = \ell \\ 0 & k \neq \ell \end{array} \right. \tag{1.6}$$

When the discrete white noise is selected in this way, the stochastic differential equation could be transformed into a (first order) difference equation by applying Eulers backward approximation to the derivative[2]. If the continuous time variable $y(t)$ is approximated by the discrete-time sequence $y(k\Delta t)$ (or in short simply denoted by $y(k)$, then difference equation that result be discretizing the stochastic differential equation (1.5) becomes:

$$\begin{aligned} \frac{y(k) - y(k-1)}{\Delta t} + ay(k) &= \frac{a}{\sqrt{\pi}} \frac{\tilde{w}(k)}{\sqrt{\Delta t}} \\ y(k) - \frac{1}{1 + a\Delta t} y(k-1) &= \frac{a}{\sqrt{\pi}} \frac{\sqrt{\Delta t}}{1 + a\Delta t} \tilde{w}(k) \end{aligned} \tag{1.7}$$

The approximation $y(k)$ is simulated in this way for three values of a, namely $1, 2$ and 5 for a discrete white noise sequence of 1000 samples. From this sequence

[2]The backward Euler method gives an approximation of a derivative via: $\dot{y}(t) \approx \frac{y(k)-y(k-1)}{\Delta t}$.

the Autocorrelation function is approximated by the PYTHON command `xcorr`. The discrete differential equation (1.7) defines a discrete time system with transfer function $G(z)$ given as:

$$G(z) = \frac{a\sqrt{\Delta t}z}{\sqrt{\pi}(1 + a\Delta t)z - \sqrt{\pi}}$$

Here z is the z-transform variable equal to $z = e^{j\omega\Delta t}$ [8]

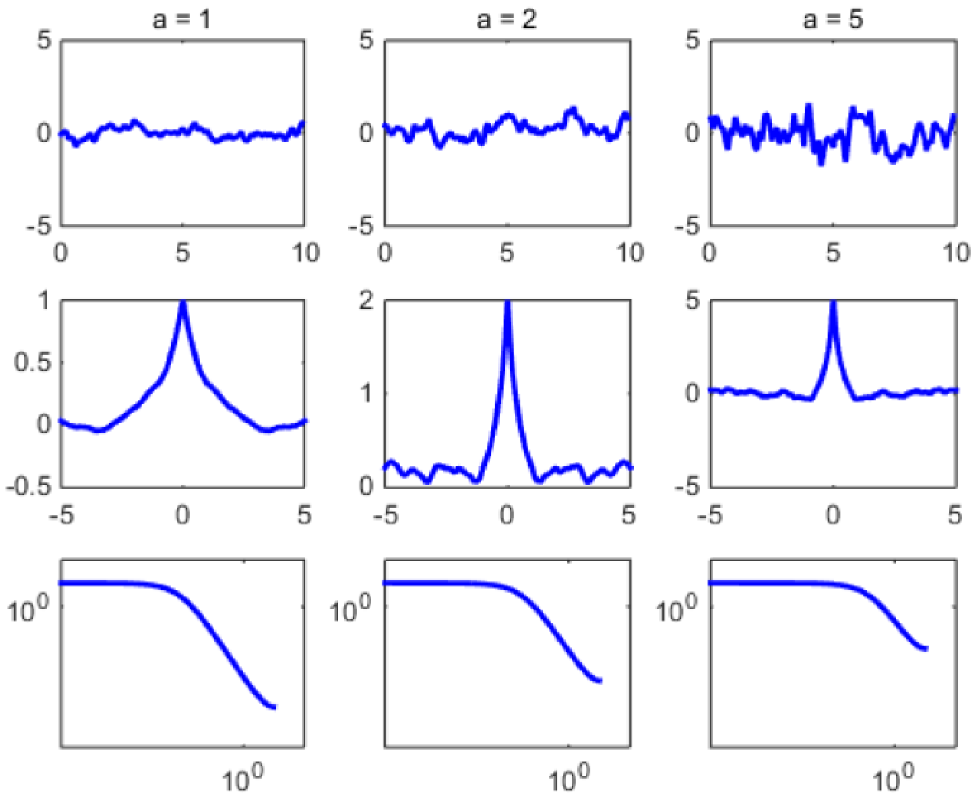

Figure 1.2: Simulating the discrete differential equation (1.7) for three different values of a equal to $1, 2$ and 5 resp. In the first row of figures one particular solution is plotted. The second row contains the approximation of the Autocorrelation function by the PYTHON command `xcorr` and the third row contains the discrete Power spectra.

Chapter 2

Simulating a Stochastic Damped Harmonic Oscillator: The PYTHON exercise

2.1 The equation of motion

In this exercise the task is to simulate the stochastic damped harmonic oscillator by discretizing the Stochastic differential equation (1.2). For that purpose use is made of the following physical values and values relevant to the discretization listed in Table 2.1. In this case we choose β to be 0.8.

Table 2.1: The physical values and sample values for discretizing the equation of motion (1.2).

Quantity	Meaning	Value	unit
N	number of samples	10^4	/
Δt	sampling period	10^{-3}	s
ω	oscillation frequency	3	rad/s
γ	Decay rate	1	rad/s
β	Beta	0.5	rad J/kg

2.2 Exercise

For simulating (1.2), the following Parts need to be solved and reported about. The reporting format is outlined in the next section.

1 point

1. If the random process $x(k)$ in (1.2) is wide-sense stationary (WSS), what can you say about its mean and variance? And what if the process is not WSS?

2. Use the rule $\frac{dx}{dt} \approx \frac{x(k)-x(k-1)}{\Delta t}$ to approximate the derivative operator $\frac{d(.)}{dt}$ and the second order derivative operator $\frac{d^2(.)}{dt^2}$ in (1.2). Subsequently replace the white noise signal $w(t)$ by a discrete white noise sequence as outlined in Section 1.3. Let $x(k)$ be an approximation of $x(t)$ for $t = k\Delta t$, then we ask you to write your result into the following second order difference equation form:

$$x(k) + b_1 x(k-1) + b_2 x(k-2) = b_3 \tilde{w}(k) \tag{2.1}$$

Your answer should consist of two parts:
(1) The derivation of he coefficients 2 points
(2) an analytic expression for the coefficients b_i for $i = 1, 2, 3$ and 1 points
(3) their numerical values making use of the data given in Table 2.1. 1 point

3. Determine the Z-transform of the dynamical system described by the difference equation from the previous question. Also determine the poles, both analytically and numerically. Is this system (BIBO) stable? 2 points

4. Simulate the difference equation (2.1) for $k = 3, \cdots, N$ using the initial conditions $x(1) = x(2) = 0$. Hereby you should generate discrete white noise samples using the PYTHON command `numpy.random.randn`.
Your answer should consist of two parts:
(1) the PYTHON script used to generate your N samples of $x(k)$ and 1 point
(2) a plot of one realization of the sequence $x(k)$. 1 point

5. Let us denote one realization of the solution of the difference equation obtained in part 2 of the exercise by $x(k, \lambda)$ for $\lambda = 1$ and the used white noise sequence by $\tilde{w}(k, 1)$. Then the task is to generate L realizations $x(k, \lambda)$ for $\lambda = 1, \cdots, L$, with each realization generated for a different realization of the discrete time white noise sequence $\tilde{w}(k, \lambda)$. Your answer should consist of two parts:
(1) the PYTHON script used to generate L realisations $x(k, \lambda)$ and 1 point
(2) a plot of all L realizations for $L = 30$. 1 point

6. Using the L realizations of the previous questions, estimate the mean and variance of the stochastic process $x(k)$. Here, you can use the approximation $E[x(k)] \approx \frac{1}{L}\sum_{\lambda=1}^{L} x(k, \lambda)$ and a similar approximation for the variance, or use the available built-in PYTHON commands. Your answer should consist of two parts:
(1) the PYTHON script used to estimate the mean and variance and 1 point
(2) a plot of the estimated mean and variance as a function of time. 1 point

7. Based on the results of the previous question, do you believe that the stochastic process $x(k)$ is WSS? Why or why not? 1 point

8. The mean and variance of the stochastic process $x(k)$ can also be calculated

analytically. This leads to the following expressions:

$$\mu_x(t) = e^{-\gamma t/2}\left(x_0\cos(\omega' t) + \gamma x_0 \frac{\sin(\omega' t)}{2\omega'}\right),$$
$$\sigma_x^2(t) = \frac{\beta^2}{2\gamma\omega^2} + e^{-\gamma t}\left(\frac{\beta^2}{8\gamma\omega'^2\omega^2}\right)\left(-4\omega^2 + \gamma^2\cos(2\omega' t) - 2\gamma\omega'\sin(2\omega' t)\right).$$

Here, ω' is another frequency parameter with the value $\omega' = \sqrt{\omega^2 - \gamma^2/4}$, and x_0 is the value of x at time $t = 0$ (note that in this case this value greatly simplifies the expression).

Compare the theoretical mean and variance with the estimated ones from question 6 for different values of L. Your answer should consist of two parts:
(1) the PYTHON script used to calculate the theoretical mean and variance and
(2) a plot of both the theoretical mean and variance as well as the estimated ones from question 6 as a function of time. Please use one plot with these four functions for $L = 30$, one plot for $L = 100$, and one plot for $L = 1000$, so three plots in total (use lower values for L if your computer runs slowly).

1 point

2 points

9. How do the estimations of the mean and variance vary with respect to the number of realizations L?

1 point

10. Use the results from questions 6-9 and from question 3 to make a final conclusion on whether the stochastic process $x(k)$ is WSS or not, with motivation.

2 points

2.3 Reporting

The report of this PYTHON exercise should consist of the requested answers to the above 10 Parts also nicely ordered in 10 corresponding parts. The PYTHON scripts should be included as text in your reports as well as your PYTHON plots. To make this simple please use the provided template, in which you can find instructions on how to export the Jupyter notebook as a PDF file.

Bibliography

[1] D. Lemons, *An Introduction to Stochastic Processes in Physics*, The Johns Hopkins University Press, , Baltimore, 2002.

[2] A. Einstein, *Investigations on the Theory of Brownian Movement*, Annalen der Physik 17, pp.549-560, 1905.

[3] K.J. Åström, *Noise and Control Design*, Excerpt from book manuscript 'Control of Microsystems', 2010.

[4] Robert Brown, *A brief account of microscopical observations made in the months of June, July and August, 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies*. Philosophical Magazine, Vol. 4, pp. 161–173, 1828.

[5] Tongeang Li and Mark Raizen, *Brownian motion at short time scales*, arXiv:1211.148v1.

[6] Paul Langevin, *Sur la théorie du mouvement brownien*, C. R. Acad. Sci. Paris, Vo. 146, pp. 530–533, 1908.

[7] G. Volpe and G. Volpe, *Simulation of a Brownian particle in an optical trap*, American J. Phys, Vol. 81(3), pp. 224–230, 2013.

[8] M. Hayes, *Statistical Digital Signal Processing and Modeling*, John Wiley & Sons, New York, 1996.