A Random Oscillator

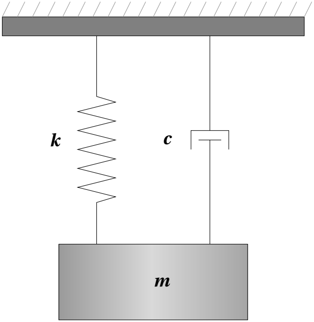
Prof. Americo Cunha Jr

americocunha.org

@AmericoCunhaJr

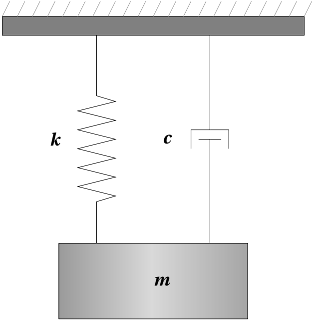
@AmericoCunhaJr @AmericoCunhaJr

Mass-Spring-Damper Oscillator



$$m\,\ddot{x} + c\,\dot{x} + k\,x = 0 \quad \dot{x}(0) = v_0 \quad x(0) = x_0$$

Mass-Spring-Damper Oscillator



$$m\,\ddot{x} + c\,\dot{x} + k\,x = 0 \quad \dot{x}(0) = v_0 \quad x(0) = x_0$$

What happens if the stiffness k is random ?

Parametric probabilistic approach

Probability space: $(\Omega, \Sigma, \mathcal{P})$

Stiffness k is modeled as a random variable

$$K : \omega \in \Omega \mapsto K(\omega) \in \mathbb{R}.$$

Displacement x has to be modeled as a random process

$$X_t : (\omega, t) \in \Omega \times \mathcal{T} \mapsto X(\omega, t) \in \mathbb{R},$$

which respect the stochastic equation of motion

$$m\,\ddot{X}(\omega, t) + c\,\dot{X}(\omega, t) + K(\omega)\,X(\omega, t) = 0,$$

$$\dot{X}(\omega, 0) = v_0, \quad X(\omega, 0) = x_0.$$

Contruction of the probabilistic model

<u>Known theoretical information:</u>

- positive support – $\mathrm{Supp}\, p_K \subset (0, +\infty) \implies K > 0 \;\; a.s.$
- finite variance – $\mathbb{E}\left\{K^2\right\} < +\infty$
- known mean – $\mathbb{E}\left\{K\right\} = \mu_K$
- inverse finite variance – $\mathbb{E}\left\{K^{-2}\right\} < +\infty$

Contruction of the probabilistic model

Known theoretical information:

- positive support – $\mathrm{Supp}\, p_K \subset (0, +\infty) \Longrightarrow K > 0 \quad a.s.$
- finite variance – $\mathbb{E}\left\{K^2\right\} < +\infty$
- known mean – $\mathbb{E}\left\{K\right\} = \mu_K$
- inverse finite variance – $\mathbb{E}\left\{K^{-2}\right\} < +\infty$

$$p_K(k) = \mathbb{1}_{(0,+\infty)}(k) \frac{1}{\mu_K} \frac{\delta_K^{-2\delta_K^{-2}}}{\Gamma(\delta_K^{-2})} \left(\frac{k}{\mu_K}\right)^{\delta_K^{-2}-1} \exp\left\{-\frac{k/\mu_K}{\delta_K^2}\right\}$$

Gamma distribution is obtained via MaxEnt Principle.

rand_oscilator.m (1/7)

```
clc
clear all
close all

t0  = 0.0;    % initial time of analysis (s)
t1  = 30.0;   % final time of analysis (s)
Ndt = 300;    % number of time steps

m  = 1.0;     % system mass (kg)
c  = 0.1;     % damper constant (N.s/m)
k  = 5.0;     % stiffness constant (N/m)
x0 = 1.0;     % initial position (m)
v0 = 1.0;     % initial velocity (m/s)
```

rand_oscilator.m (2/7)

```
% define stochastic parameters
rng_stream = RandStream('mt19937ar','Seed',30081984);
RandStream.setGlobalStream(rng_stream);

% number of samples
Ns = 256;
% preallocate memory for displacement and velocity
Qd = zeros(Ndt,Ns);
Qv = zeros(Ndt,Ns);
% stiffness mean (N/m)
mean_k = k;
% stiffness coef. var
coefvar_k = 0.15;
% generate stiffness with Gamma distribution (N/m)
k = gamrnd(1/coefvar_k^2,mean_k*coefvar_k^2,[Ns,1]);
```

rand_oscilator.m (3/7)

```
% init. cond. and interval of analysis
IC = [x0 v0]; tspam = linspace(t0,t1,Ndt);

% Monte Carlo method
for n=1:Ns

    % system of equations
    dydt=@(t,y)[0 1; -k(n) -c]*y;

    % ODE solver Runge-Kutta45
    [t,y] = ode45(dydt,tspam,IC);

    % time series of system displacement (m)
    Qd(:,n) = y(:,1);

    % time series of system velocity (m/s)
    Qv(:,n) = y(:,2);
end
```

rand_oscilator.m (4/7)

```
% sample mean
Qd_smean = mean(Qd');
Qv_smean = mean(Qv');

% temporal mean
Qd_tmean = mean(Qd);
Qv_tmean = mean(Qv);

% std. dev.
Qd_std  = std(Qd');
Qv_std  = std(Qv');

% confidence band
Pc = 95;
r_plus = 0.5*(100 + Pc); r_minus = 0.5*(100 - Pc);
Qd_upp = prctile(Qd',r_plus);
Qv_upp = prctile(Qv',r_plus);
Qd_low = prctile(Qd',r_minus);
Qv_low = prctile(Qv',r_minus);
```

rand_oscilator.m (5/7)

```
% histogram of temporal mean
Nbins = round(sqrt(Ns));
[Qd_bins,Qd_freq] = randvar_pdf(Qd_tmean,Nbins);
[Qv_bins,Qv_freq] = randvar_pdf(Qv_tmean,Nbins);

% kernel density estimator for temporal mean
[Qd_ksd,Qd_supp] = ksdensity(Qd_tmean);
[Qv_ksd,Qv_supp] = ksdensity(Qv_tmean);
```

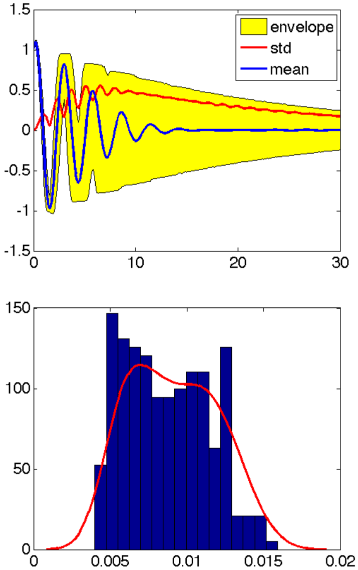
rand_oscilator.m (6/7)

```
figure(1)
fh1 = plot(t,Qd_smean,'b','linewidth',3); hold on
fh2 = plot(t,Qd_std,'r','linewidth',3);
fh3 = fill([t' fliplr(t')],[Qd_upp fliplr(Qd_low)],'y');
uistack(fh3,'top');
uistack(fh2,'top');
uistack(fh1,'top');
legend('envelope','std','mean')
hold off

figure(2)
fh1 = plot(t,Qv_smean,'b','linewidth',3); hold on
fh2 = plot(t,Qv_std,'r','linewidth',3);
fh3 = fill([t' fliplr(t')],[Qv_upp fliplr(Qv_low)],'y');
uistack(fh3,'top');
uistack(fh2,'top');
uistack(fh1,'top');
legend('envelope','std','mean')
hold off
```

rand_oscilator.m (7/7)

```
figure(3)
bar(Qd_bins,Qd_freq,1.0);
hold on
plot(Qd_supp,Qd_ksd,'r','linewidth',3)
hold off

figure(4)
bar(Qv_bins,Qv_freq,1.0);
hold on
plot(Qv_supp,Qv_ksd,'r','linewidth',3)
hold off
```

Random oscillator response



References

A. Cunha Jr, *Modeling and quantification of physical systems uncertainties in a probabilistic framework*, In: S. Ekwaro-Osire; A. C. Gonçalves; F. M. Alemayehu (Org.), **Probabilistic Prognostics and Health Management of Energy Systems**, Springer, 2017. `http://dx.doi.org/10.1007/978-3-319-55852-3_8`

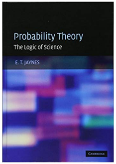
E. T. Jaynes, **Probability Theory: The Logic of Science**, Cambridge University Press, 2003.

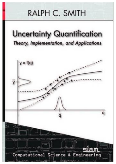
R. C. Smith, **Uncertainty Quantification: Theory, Implementation, and Applications**, SIAM, 2013.

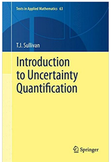
T. J. Sulivan, **Introduction to Uncertainty Quantification**, Springer, 2015.

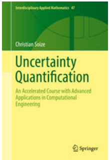
C. Soize **Uncertainty Quantification: An Accelerated Course with Advanced Applications in Computational Engineering**, Springer, 2017.

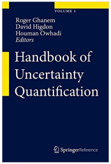
R. Ghanem, D. Higdon and H. Owhadi (Editors) **Handbook of Uncertainty Quantification**, Springer, 2017.











How to cite this material?

A. Cunha Jr, *A Random Oscillator*, 2021.

@AmericoCunhaJr

@AmericoCunhaJr @AmericoCunhaJr