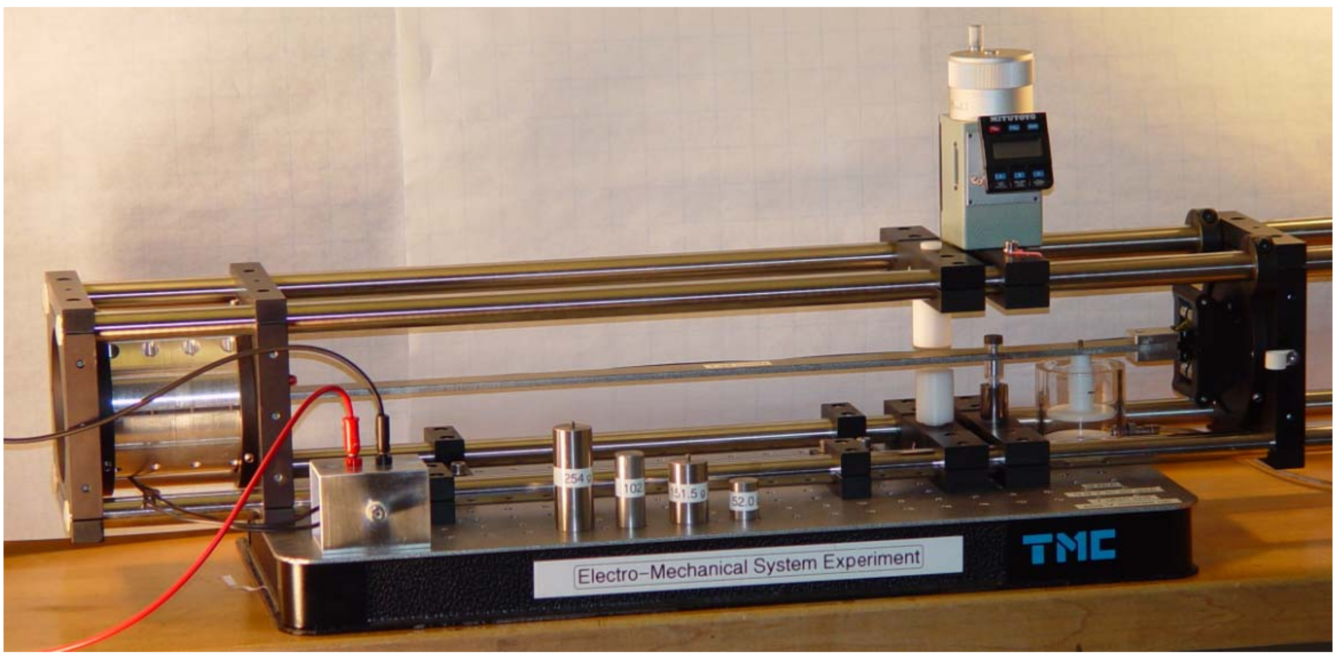
Electro-Mechanical System: Stochastic Gaussian Input

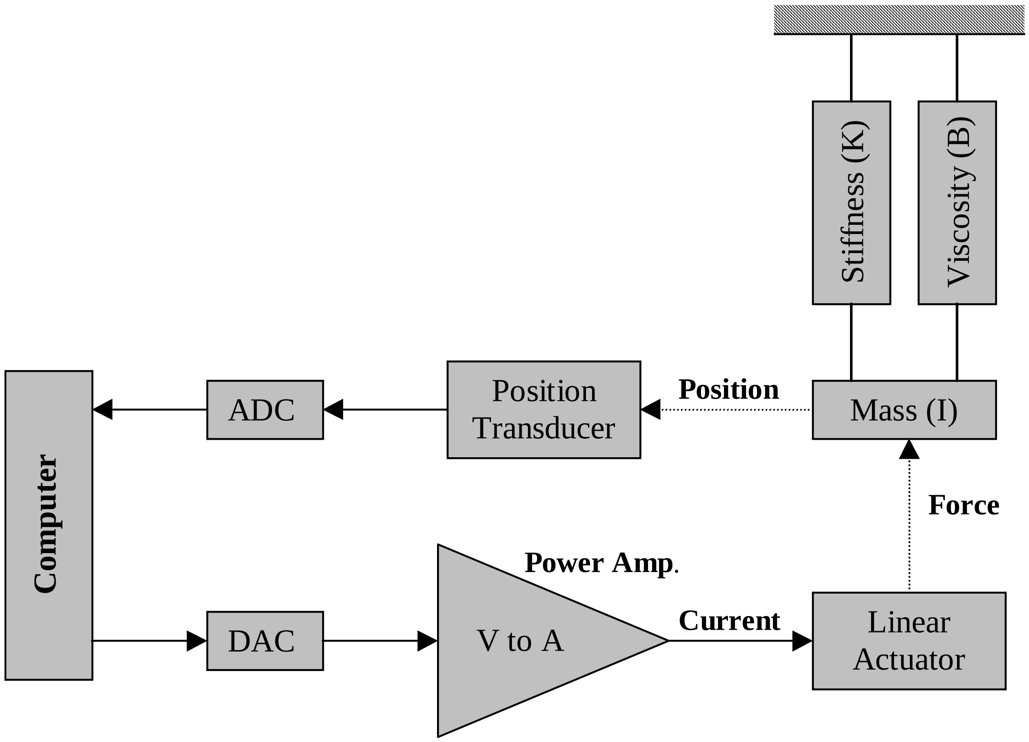
Ian Hunter, MIT, 15 February 2018

Introduction

A copper-wire coil (acting as a linear actuator) is attached to the free end of a 25 mm wide by 4 mm thick stainless steel beam which protrudes 615 mm from a large clamp (see Figure below). The coil moves vertically by about 12 mm (peak to peak) within an air gap formed by two pairs of permanent magnets. The top pair of permanent magnets create a 0.9 T magnetic field in the gap. The bottom pair of permanent magnets produce a -0.9 T magnetic field. These magnetic fields are at right angles to current flowing in the coil. The Lorentz force generated by the coil current interacting with the magnetic field is at right angles to both the current and magnetic field. The beam is consequently deflected by this force (a positive current produces an upward force and deflects the beam upward). The beam deflection is measured by an inductive position sensor (Fastar) mounted 500 mm from the clamp. The position sensor may be calibrated using a Mitutoyo Digital Micrometer (1 μm resolution) mounted above the position sensor.



A dashpot type damper, mounted 560 mm from the clamp, may be filled with a liquid (e.g. water) to provide viscous damping. A set of calibrated masses (approximately 50 g increments) may be mounted above the dashpot to provide known forces to deflect the beam for measurement of the static beam stiffness and for determination of the coil current to force relation.



Static Characteristics

Some important static characteristics of the system (obtained using the Mathcad module titled "ElectroMechanicalSystemStaticMeasurements.mcd") are listed below.

$\text{CoilCurrentMax} := 0.8 \cdot \text{A}$ $\qquad$ $\text{CoilResistance} := 7 \cdot \text{ohm}$

$\text{CoilVoltageMax} := \text{CoilResistance} \cdot \text{CoilCurrentMax}$ $\qquad$ $\text{CoilVoltageMax} = 5.6 \cdot \text{V}$

To be safe always limit the voltage to about 4 V and the current to 0.6 A (i.e. set the power supply current limit to 0.6 A).

$$\text{PositionSensorVoltageToPosition} := 2.5 \cdot 10^{-3} \cdot \frac{\text{m}}{\text{V}}$$

$$\text{BeamStiffness} := 750 \cdot \frac{\text{N}}{\text{m}}$$

$$\text{CoilCurrentToForce} := 9 \cdot \frac{\text{N}}{\text{A}}$$

$$\text{CoilVoltageToForce} := \frac{\text{CoilCurrentToForce}}{\text{CoilResistance}} \qquad \text{CoilVoltageToForce} = 1.286 \cdot \frac{\text{N}}{\text{V}}$$

$$\text{CoilForceToVoltage} := \frac{1}{\text{CoilVoltageToForce}} \qquad \text{CoilForceToVoltage} = 0.778 \cdot \frac{\text{V}}{\text{N}}$$

Generate Gaussian White Noise

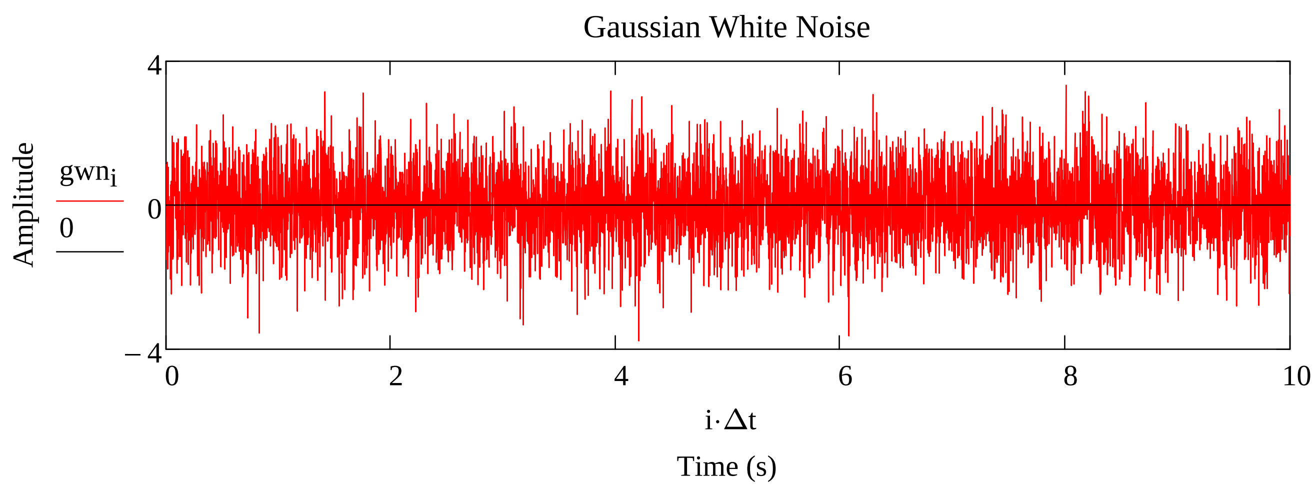
$I := 5000$ **Maximum sample (number of samples = I+1)**

$i := 0 .. I$

$$gwn_i := \sum_{j=1}^{12} rnd(1) - 6$$ **white Gaussian signal**

$\Delta t := 0.002 \ s$ **Time between samples**

Sampling rate $= \frac{1}{\Delta t} = 500 \quad Hz$

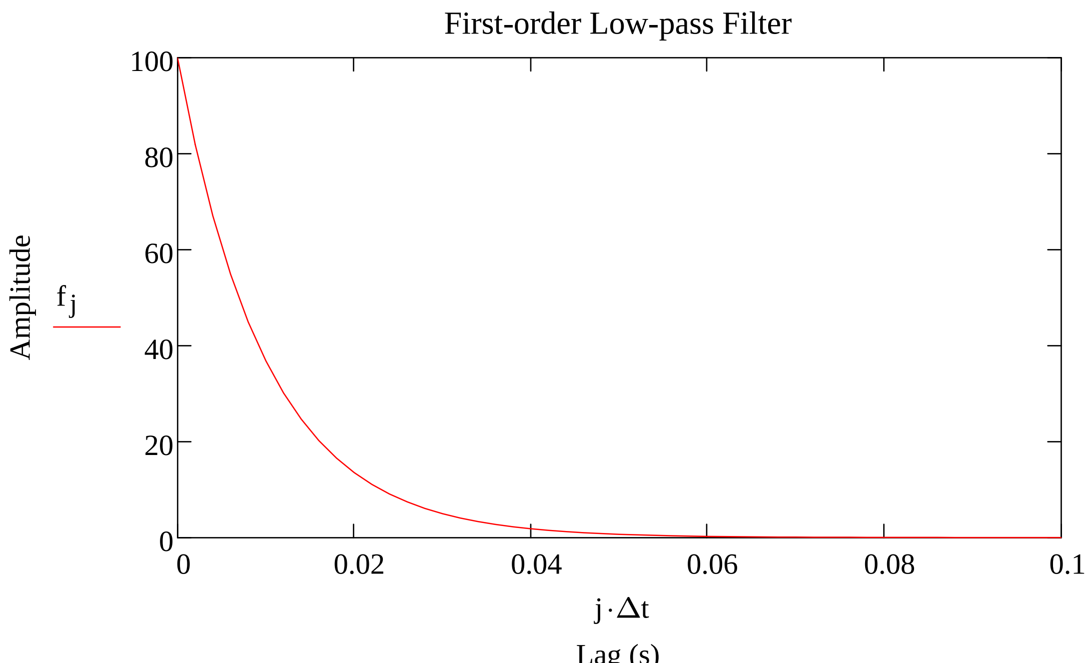


We could use this Gaussian white noise signal directly as the input. Gaussian white noise has a flat power spectrum (all frequencies up to the Nyquist are equally represented on average). It is sometimes better to "shape" the input spectrum by filtering the Gaussian white noise to produce a Gaussian stochastic input having a desired power spectrum (more details on input design are provided in other notes). Although it is not really required here we will do this to illustrate the process. To keep things simple we will use a first-order low-pass filter (as opposed to a more complicated filter) to filter the Gaussian white noise so that it will have more power at lower frequencies than at higher frequencies.

Define First-Order Low-Pass Filter

$\tau := 0.01$ $\quad\quad J := 50$ $\quad\quad j := 0..J$

$$f_j := \frac{1}{\tau}\cdot\exp\left(-\frac{j\cdot\Delta t}{\tau}\right)$$



Convolve the Gaussian White Noise with the Filter

$$x_i := \Delta t\cdot\sum_{j=0}^{\min((i\ \ J))}\left(f_j\cdot gwn_{i-j}\right)$$

This implements numeric convolution.

Compute Means and Standard Deviations

$$\mu(x) := \frac{1}{I+1}\cdot\sum_{i=0}^{I} x_i \qquad \mu_x := \mu(x)$$

mean

$$\sigma(x) := \left|\begin{aligned} &\mu \leftarrow \frac{1}{I+1}\cdot\sum_{i=0}^{I} x_i \\ &\sqrt{\frac{1}{I+1}\cdot\sum_{i=0}^{I}\left(x_i-\mu\right)^2} \end{aligned}\right. \qquad \sigma_x := \sigma(x)$$

standard deviation

Give the Input the Desired Mean and Standard Deviation

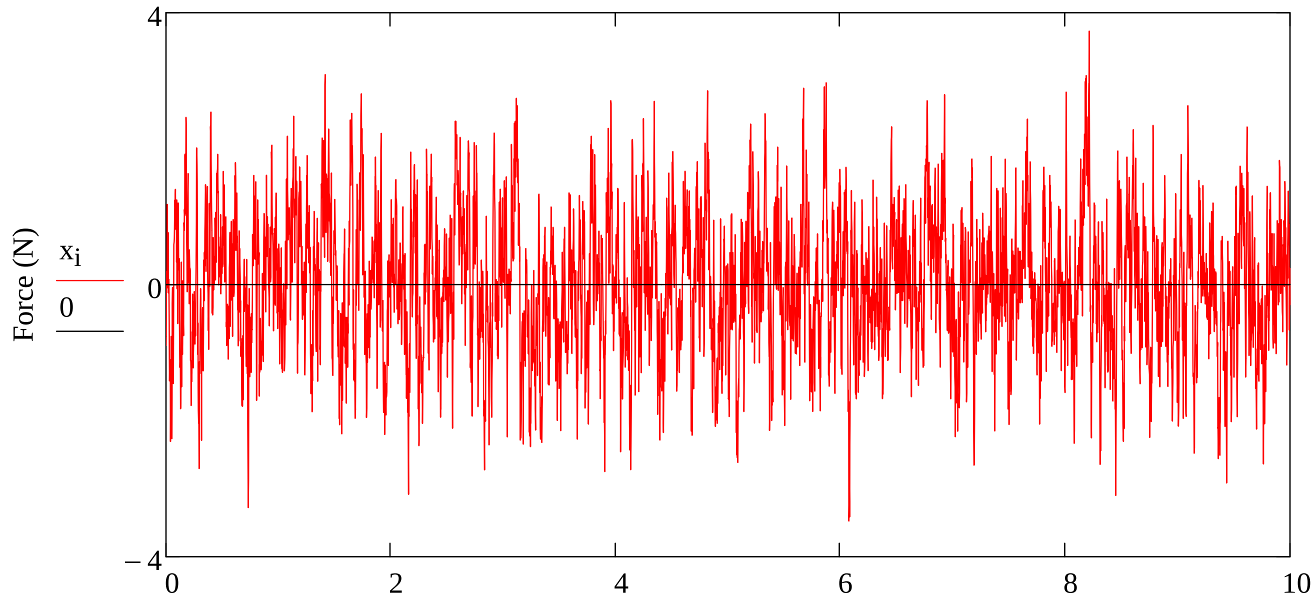
$$x_i := \frac{1.0 \cdot (x_i - \mu_x)}{\sigma_x}$$ **Set the input to have mean of 0.0 and standard deviation of 1 N**

Check $\mu_x := \mu(x)$ $\mu_x = 0$

$\sigma_x := \sigma(x)$ $\sigma_x = 1$

Remember: the standard deviation squared is the signal variance

Input Signal

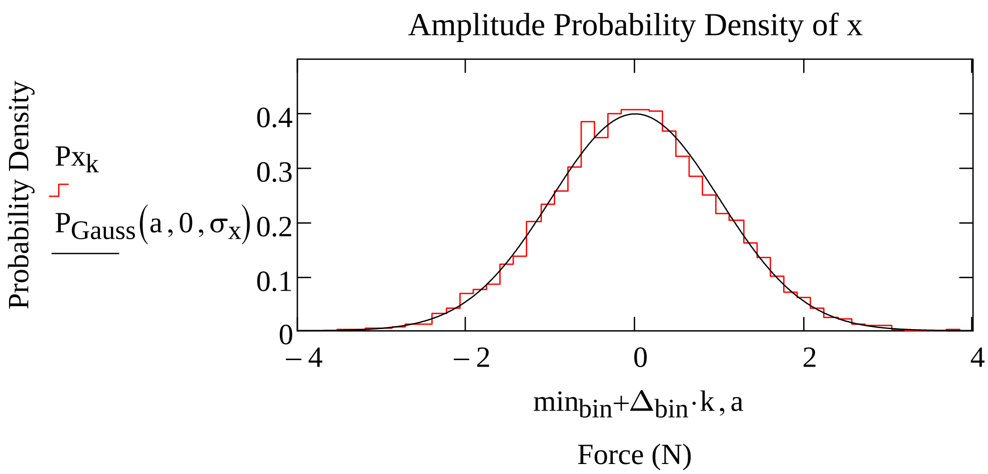


We can check that the input has a Gaussian probability density, Px , by calculating its probability density function and comparing it to a Gaussian probability density with the same mean and SD.

$K := 50 \qquad min_{bin} := -4\cdot\sigma_x \qquad k := 0 .. K \qquad \Delta_{bin} := 8\cdot\sigma_x\cdot K^{-1}$

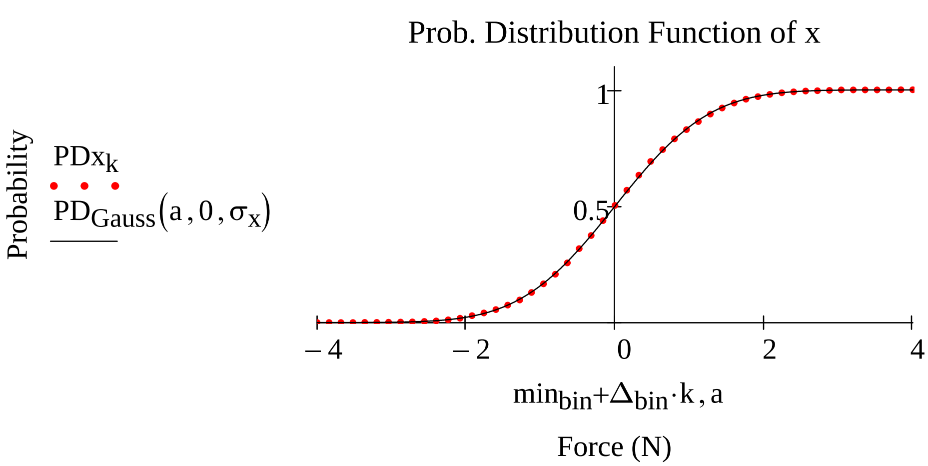
$$Px_k := \frac{1}{\Delta_{bin}\cdot(I+1)}\cdot\sum_{i=0}^{I} \mathrm{if}\left[\left(min_{bin}+\Delta_{bin}\cdot k\right) < x_i \le \left[min_{bin}+\Delta_{bin}\cdot(k+1)\right], 1, 0\right]$$

$$P_{Gauss}(x,\mu,\sigma) := \frac{1}{\sigma\cdot\sqrt{2\cdot\pi}}\cdot e^{-\frac{1}{2}\cdot\left(\frac{x-\mu}{\sigma}\right)^2}$$ Gaussian probability density function



$$PDx_k := \frac{1}{I+1}\cdot\sum_{i=0}^{I} \mathrm{if}\left[x_i \le \left(min_{bin}+\Delta_{bin}\cdot k\right), 1, 0\right]$$ Probability distribution function of x

$$PD_{Gauss}(x,\mu,\sigma) := \int_{-10}^{x} \frac{1}{\sigma\cdot\sqrt{2\cdot\pi}}\cdot e^{-\frac{1}{2}\cdot\left(\frac{x-\mu}{\sigma}\right)^2}\,dx$$ Gaussian probability distribution function

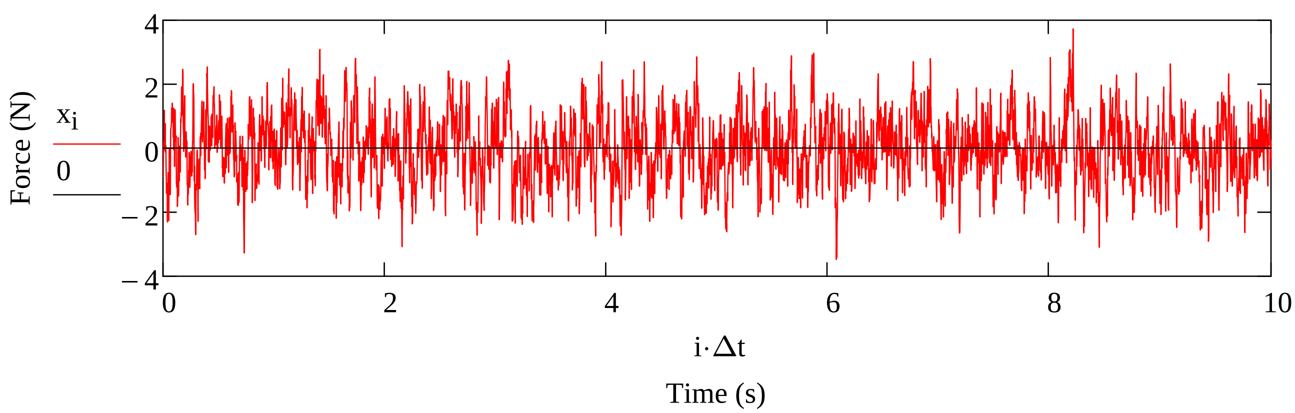


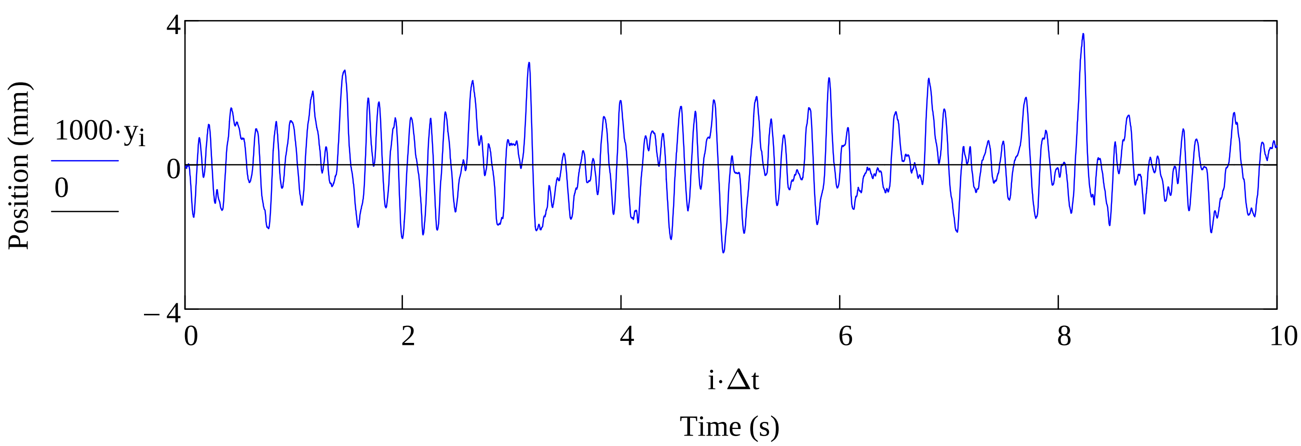
Lets now convert this force signal to a voltage (multiply by CoilForceToVoltage) and send it out the digital to analog converter (DAC) to the power amplifier (voltage to current) which drives the voice-coil actuator (current to force). At the same time we record the displacement response (system output) of the beam via the analog to digital converter (ADC) and convert it to a displacement (PositionSensorVoltageToPosition).

$$go = 0$$

$$
y := \left|
\begin{array}{l}
\text{if } \; go = 1 \\
\quad \left|
\begin{array}{l}
time \leftarrow timer(\Delta t) \\
\text{for } \; i \in 0 .. I \\
\quad \left|
\begin{array}{l}
timer(0) \\
err \leftarrow da\left(0, CoilForceToVoltage \cdot x_i\right) \\
y_i \leftarrow PositionSensorVoltageToPosition \cdot ad(0) \cdot V
\end{array}
\right. \\
err \leftarrow da(0, 0.0) \\
y
\end{array}
\right. \\
0 \quad \text{otherwise}
\end{array}
\right.
$$

$\mu_y := \mu(y)$ $\sigma_y := \sigma(y)$ $y_i := y_i - \mu_y$ Set y to zero mean



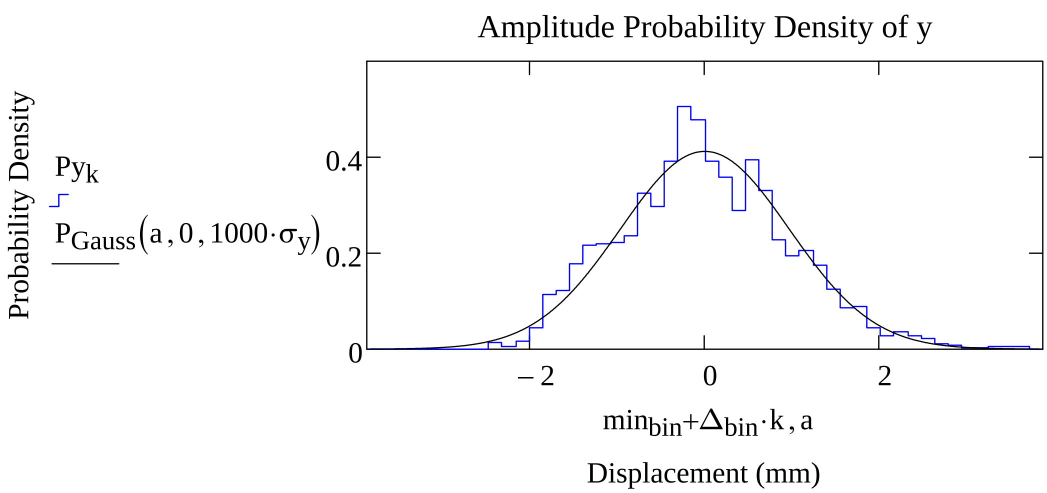


If the system is linear and the input is Gaussian (but not necessarily white) then the output should be Gaussian. We can check that the output has a Gaussian probability density, Py , by calculating its probability density function and comparing it to a Gaussian probability density with the same mean and standard deviation.

$$K := 50 \qquad \Delta_{bin} := \frac{8000 \cdot \sigma_y}{K} \qquad min_{bin} := -4000 \cdot \sigma_y \qquad k := 0 .. K$$

$$Py_k := \frac{1}{\Delta_{bin} \cdot (I+1)} \cdot \sum_{i=0}^{I} \text{if}\left[\left(min_{bin} + \Delta_{bin} \cdot k\right) < 1000 \cdot y_i \le \left[min_{bin} + \Delta_{bin} \cdot (k+1)\right], 1, 0\right]$$

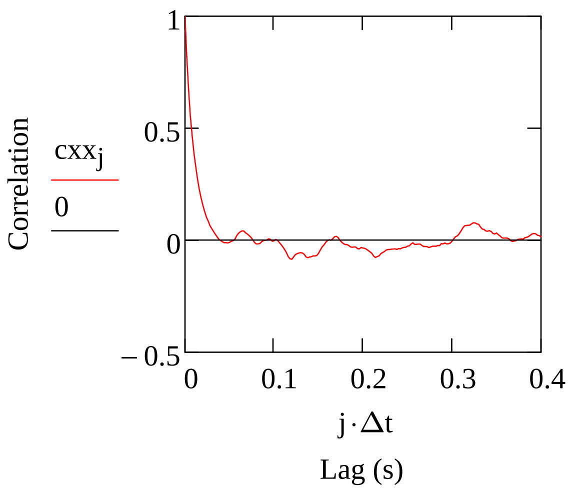
$$P_{Gauss}(y, \mu, \sigma) := \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot e^{-\frac{1}{2} \cdot \left(\frac{y-\mu}{\sigma}\right)^2}$$

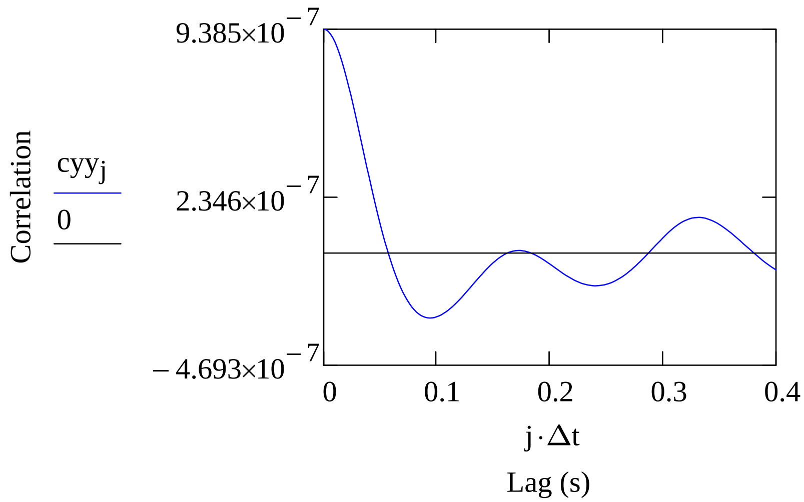


Determine input and output biased auto-correlation functions (Note: means are 0.0)

$J := 200 \qquad j := 0 .. J$

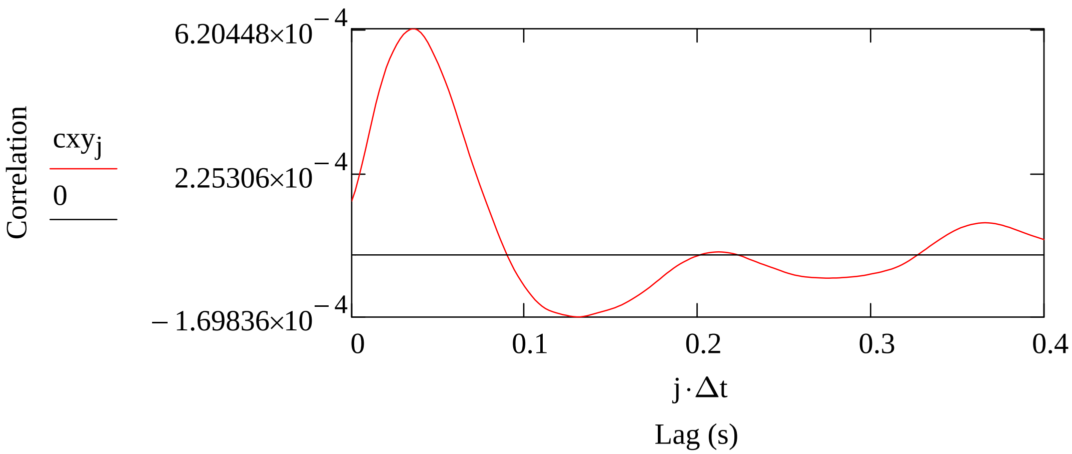
$$cxx_j := \frac{1}{I+1} \cdot \sum_{i=j}^{I} \left(x_{i-j} \cdot x_i\right) \qquad cyy_j := \frac{1}{I+1} \cdot \sum_{i=j}^{I} \left(y_{i-j} \cdot y_i\right)$$





Determine the input-output cross-correlation function.

$$cxy_j := \frac{1}{I+1} \cdot \sum_{i=j}^{I} \left(x_{i-j} \cdot y_i\right)$$



Toeplitz Matrix

Consider a one dimensional array

$$A := \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}$$

A special matrix called a Toeplitz matrix may be formed from A as follows

$j := 0 .. 5$ $\qquad\qquad$ $k := 0 .. 5$

$$\text{ToeplitzMatrix}_{j,k} := A_{|j-k|}$$

$$\text{ToeplitzMatrix} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & 1 & 2 & 3 & 4 \\ 2 & 1 & 0 & 1 & 2 & 3 \\ 3 & 2 & 1 & 0 & 1 & 2 \\ 4 & 3 & 2 & 1 & 0 & 1 \\ 5 & 4 & 3 & 2 & 1 & 0 \end{pmatrix}$$

Notice that the Toeplitz matrix has a special type of symmetry

Estimation of the system impulse response function (system identification)

In order to estimate the system impulse response function from the auto and cross-correlation functions we first form a Toepltiz matrix, Cxx, from the input auto-correlation function, cxx.

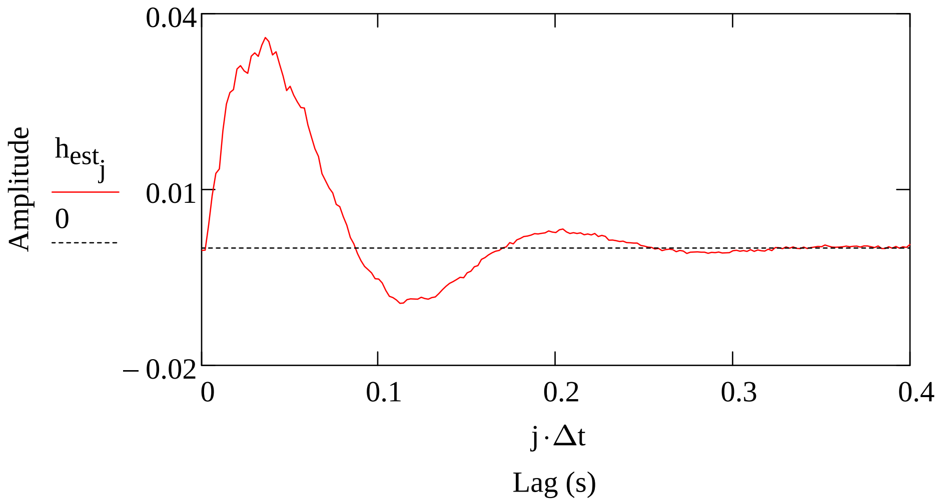
$j := 0..J$ $\qquad$ $k := 0..J$

$Cxx_{j,k} := cxx_{|j-k|}$ $\qquad$ Form the Toeplitz matrix from the input auto-correlation function

We can now estimate the system impulse response function by solving the following matrix equation involving the Toeplitz matrix, Cxx, and the input-output cross-correlation function, cxy.

$h_{est} := \frac{1}{\Delta t}\cdot\left(Cxx^{-1}\cdot cxy\right)$ $\qquad$ Solve for h via Toeplitz matrix inversion (i.e. the input auto-correlation function is deconvolved from the cross-correlation function)

Note that we have solved this matrix equation by inverting the Toeplitz matrix using the Mathcad standard matrix inversion algorithm. This is the "brute force" approach and can be rather slow and subject to numerical problems when the determinant of the Toeplitz matrix is near zero. Specialized Toeplitz matrix inversion algorithms exist which are very fast. The singular value decomposition (SVD) algorithm (rather slow but numerically robust) may also be used when the determinant of the Toeplitz matrix is near zero to avoid numerical problems.



$h_m(t, Gain, \omega_n, \zeta) := Gain\cdot\omega_n\cdot\exp(-\zeta\cdot\omega_n\cdot t)\cdot\frac{\sin\left(\sqrt{1-\zeta^2}\cdot\omega_n\cdot t\right)}{\sqrt{1-\zeta^2}}$ $\qquad$ Continuous second-order low-pass under-damped impulse response function

$h_{model_j} := h_m(j\cdot\Delta t, Gain, \omega_n, \zeta)$ $\qquad$ sampled version of the impulse response function

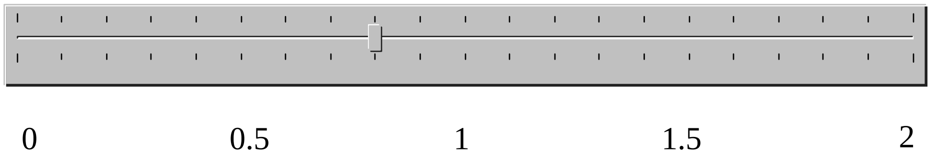
Define objective function to minimize

We define the sum of squared error between the measured impulse response and the model

$$SS_{error}(Gain, \omega_n, \zeta) := \sum_{j=0}^{J}\left(h_{est_j} - h_m(j\cdot\Delta t, Gain, \omega_n, \zeta)\right)^2$$

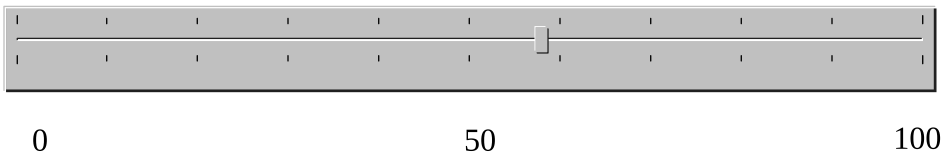
Estimate Parameter Values by Hand

Gain :=



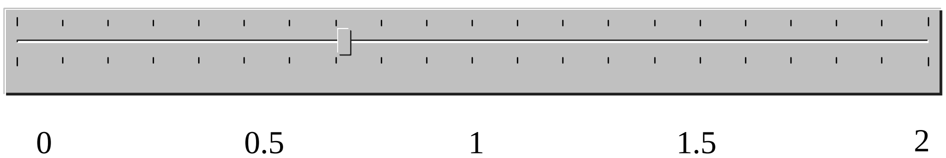
$\text{Gain} = 8 \times 10^{-4} \quad \frac{m}{N}$

ω_n :=



$\omega_n = 58 \quad \text{rad /s}$

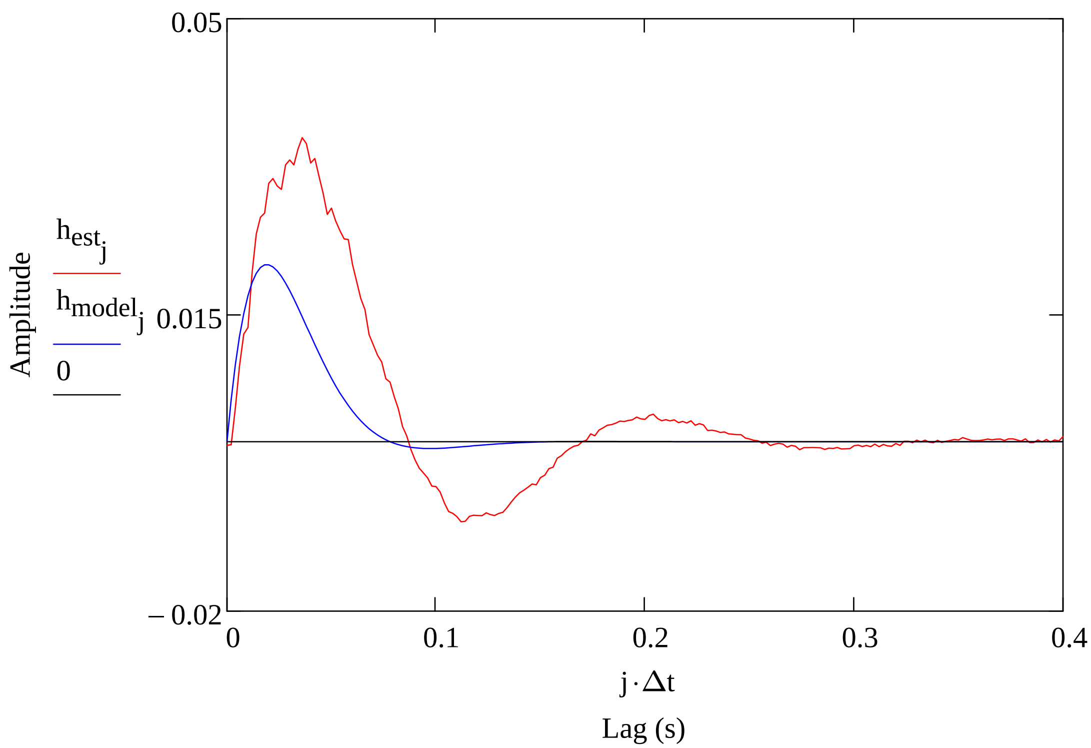
ζ :=



$\zeta = 0.72$

$$SS_{error}(\text{Gain}, \omega_n, \zeta) = 9.631 \times 10^{-3}$$

$$h_{model_j} := h_m(j \cdot \Delta t, \text{Gain}, \omega_n, \zeta)$$



Estimate parameters by minimizing objective function using nonlinear minimization

We will use a nonlinear minimization technique implemented by the function called Minimize() which attempts to find the parameters which minimize the function $SS_{error}(\ )=0$. Minimize() uses the Levenberg-Marquardt nonlinear minimization method. The parameter values set manually above are used as initial estimates for the minimization algorithm.

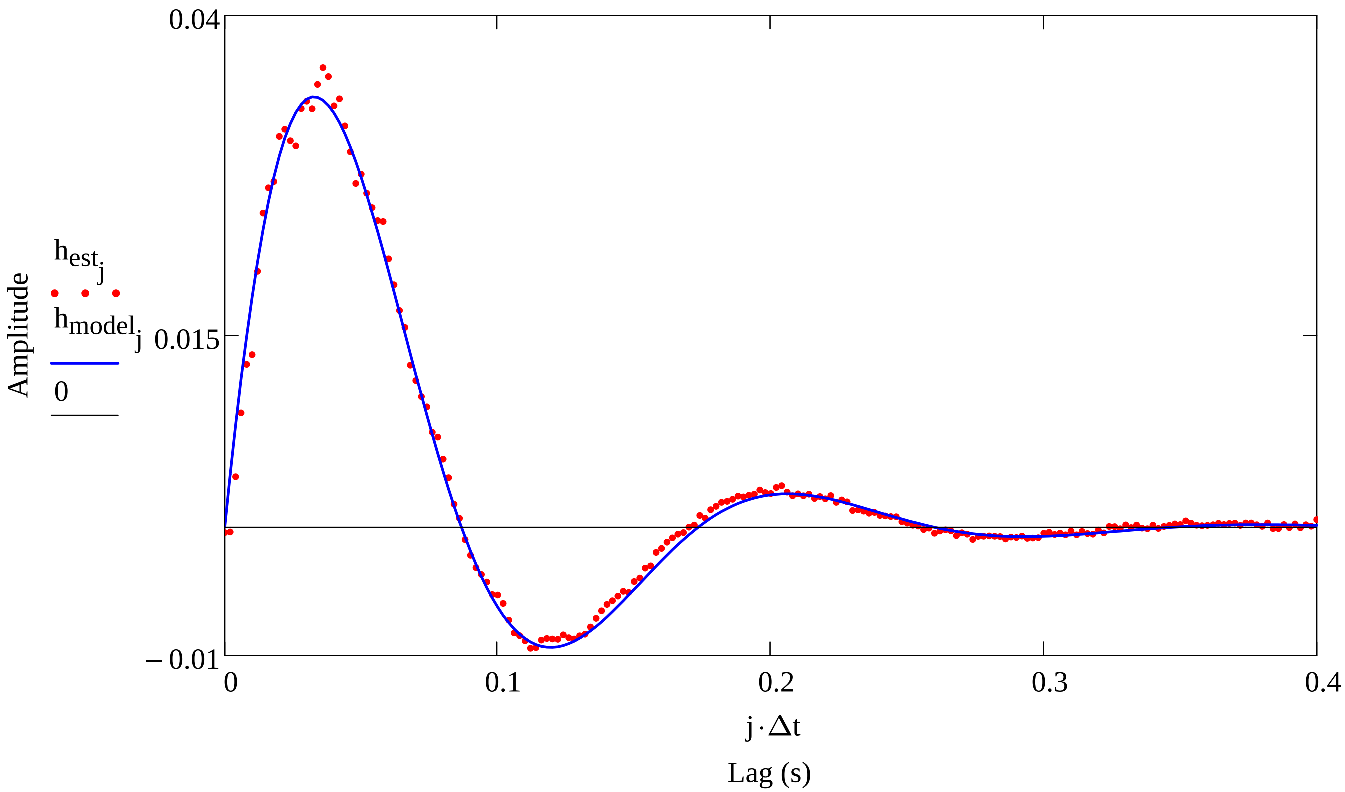
$TOL := 0.001$ Set the minimization termination condition

$$\begin{pmatrix} Gain \\ \omega_n \\ \zeta \end{pmatrix} := \text{Minimize}\left(SS_{error}, Gain, \omega_n, \zeta\right)$$ Note that this can take a few minutes

Results

$Gain = 1.392 \times 10^{-3}\ \frac{m}{N}$ $\omega_n = 39.113\ \frac{rad}{s}$ $\zeta = 0.377$

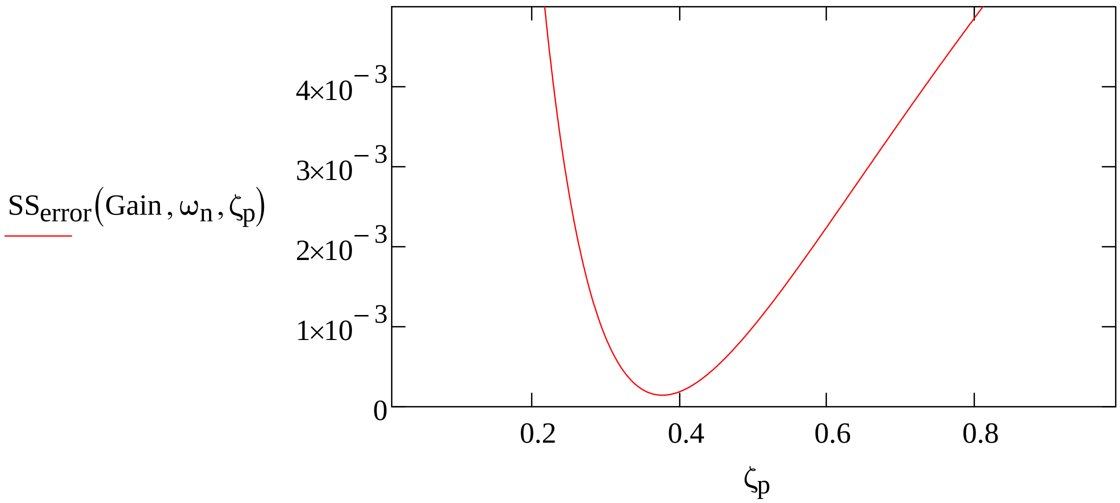
$h_{model_j} := h_m\left(j \cdot \Delta t, Gain, \omega_n, \zeta\right)$ $SS_{error}\left(Gain, \omega_n, \zeta\right) = 1.419 \times 10^{-4}$



Note that the fit of the second-order model is excellant.

Parameter Sensitivity Analysis

The figure below shows how the SS increases as ζ is increased or decreased from its optimal value.



Find the corresponding I, B, and K

The two transfer functions are related by

$$\frac{Gain\cdot\omega_n^2}{s^2 + 2\cdot\zeta\cdot\omega_n\cdot s + \omega_n^2} = \frac{1}{I\cdot s^2 + B\cdot s + K} = \frac{\frac{1}{I}}{s^2 + \frac{B}{I}\cdot s + \frac{K}{I}}$$

We can see that

$$\omega_n^2 = \frac{K}{I} \quad \text{and} \quad Gain\cdot\omega_n^2 = \frac{1}{I} \quad \text{and} \quad 2\cdot\zeta\cdot\omega_n = \frac{B}{I}$$

$$\omega_n = \sqrt{\frac{K}{I}} \qquad Gain = \frac{1}{K} \qquad \zeta = \frac{B}{2\cdot\sqrt{I\cdot K}}$$

And that

$$K := \frac{1}{Gain} \qquad B := \frac{2\cdot\zeta}{Gain\cdot\omega_n} \qquad I := \frac{1}{Gain\cdot\omega_n^2}$$

The compliance of the beam is $Gain = 1.392\times10^{-3}$ m/N

The stiffness is $K = 718.413$ N/m which is similar to the value determined using static testing

The inertia is $I = 0.47$ $N.s^2/m$ or kg

The viscosity is $B = 13.833$ N.s/m

Determine output, y_{est}, from convolution of input, x, with impulse response function, h_{est}

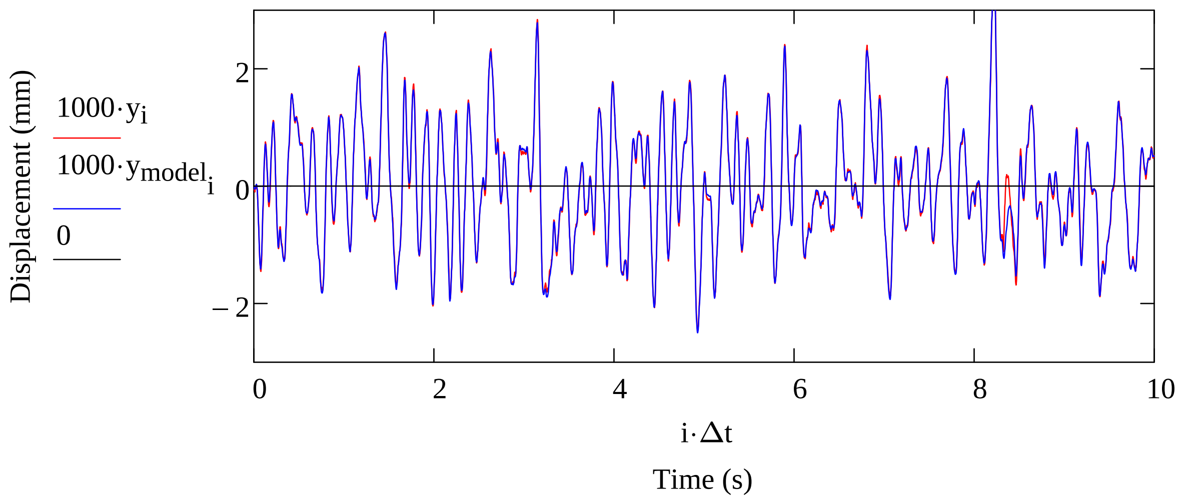
$$y_{est_i} := \Delta t \cdot \sum_{j=0}^{\min((i\ \ J))} \left(h_{est_j} \cdot x_{i-j}\right)$$

Note: this is the numeric counterpart to the convolution integral.

Determine output, y_{model}, from convolution of x with impulse response function, h_{model}

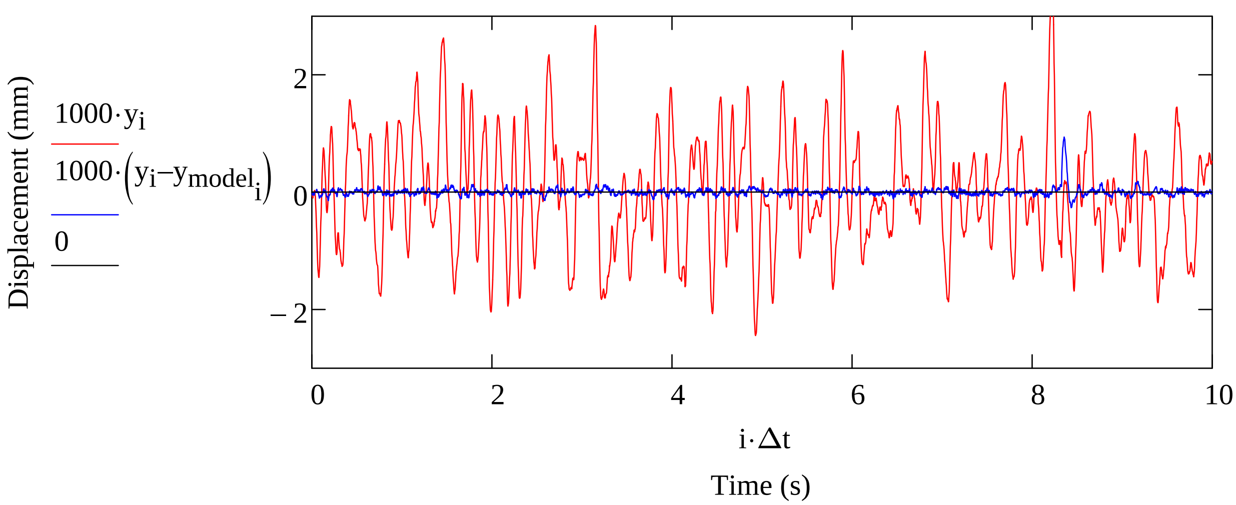
$$y_{model_i} := \Delta t \cdot \sum_{j=0}^{\min((i\ \ J))} \left(h_{model_j} \cdot x_{i-j}\right)$$

We now plot the actual output and the output predicted by the second-order transfer function model.



Notice how well we have predicted the output with the 3 parameter linear dynamic model.

We can also plot the difference between the actual and predicted outputs. These errors or residuals are actually due to a combination of modeling error and noise (perhaps added to the output).



Determine the output variance accounted for, VAF, by h_{est} and h_{model}

A quantitative measure of the success of the prediction is the variance accounted for (VAF) by the model.

NonParametric Prediction Error $\quad error_{est_i} := y_{est_i} - y_i$

$$VAF_{est} := 100 \cdot \left(1 - \frac{\sigma(error_{est})^2}{\sigma(y)^2} \right) \qquad VAF_{est} = 99.619$$

Parametric Prediction Error $\quad error_{model_i} := y_{model_i} - y_i$

$$VAF_{model} := 100 \cdot \left(1 - \frac{\sigma(error_{model})^2}{\sigma(y)^2} \right) \qquad VAF_{model} = 99.413$$

It is important to note that the variance accounted for by the non-parametric model, h_{est}, will always be greater than the variance accounted for by the parametric model, h_{model} . This is because the parametric second-order model used here only has 3 free parameters ($Gain, \omega_n, \zeta$) whereas the non-parametric model essentially has as J+1 parameters (the impulse response function values).