Final Project

This project has two parts, both aimed at a numerical investigation of the behavior of stochastic processes, by using Monte Carlo methods. The first assignment involves generating random processes that possess known (from analytical computations) properties, and developing numerical tools to obtain these properties from the generated data set. Whereas the second assignment involves analyzing data from a real experiment, conducted in the wind tunnel and containing, therefore, a sequence of noisy measurements. The idea is to apply the methods developed in the first part of the project, and gain an insight into the processes underlying the recorded data.

Some of the sub-assignments offer a choice between several problems, and some have a bonus part for those who find the compulsory assignments too easy. In both compulsory and bonus parts you are encouraged to solve the problem in more than one way and further analyze any interesting issues involved in the tasks. Graduate students are expected to include several bonus assignments in their project, at least those in sections: 1.1.D, 1.1.E, and 2.2.

The Appendix summarizes Matlab functions that can be potentially useful, however, other software packages (such as python, Octave) have similar built-in capabilities, and the tasks can be solved using any programming language of your choice.

This project is to be done and submitted by each student alone. The submission will include two stages: 1. Submitting your solution of the project tasks (typed/ neatly written), including the relevant plots and your computer code; 2. Giving an oral 5 minute presentation of your main conclusions regarding the processes under investigation (please prepare one page/ slide with a few bullets) during an individual meeting with the teacher. The meetings will be scheduled towards the end of the semester.

1 Langevin Equation

Consider the following stochastic differential equation:

$$X'(t) + \alpha X(t) = Z(t), \quad t \geq 0, X(0) = 0 \tag{1}$$

If the input $Z(t)$ is given by a zero-mean white Gaussian noise, it is called the *Langevin equation*, after the scientist who formulated it in 1908 to describe the Brownian motion of a free particle. Thus, $X'(t)$ represents the acceleration of the particle, which is equal to the force on the particle due to friction (proportional to the particle's velocity: $\alpha X(t)$) and the force due to random collisions $Z(t)$. Its solution was developed by Uhlenbeck and Ornstein in 1930.

1.1 Ornstein-Uhlenbeck Process

1.1.A Simulating realizations of the process

Simulate a number N of realizations of the process given by Eq. 1, where $Z(t)$ is a zero-mean white Gaussian noise with the autocorrelation function given by:

$$R_Z(t_1, t_2) = \sigma^2 \delta(t_1 - t_2) \tag{2}$$

where δ is the Dirac delta function. How the variance of the random variable Zd_i, inputted at each simulation step i, should be scaled according to the chosen simulation time step Δt? *Hint:*
Notice, that in a numerical simulation we are approximating the following expression:

$$E\left[\int_0^t Z(t_1)dt_1 \int_0^t Z(t_2)dt_2\right] \approx E\left[\sum_{i=1}^{\frac{t}{\Delta t}} Zd_i \Delta t \sum_{j=1}^{\frac{t}{\Delta t}} Zd_j \Delta t\right] \tag{3}$$

where $Zd_n, \quad n = 1, 2, ..$ is a random iid sequence with $R_{Zd}(i,j) = \sigma_d^2 \delta_{i,j}$. Compute the left side of Eq. 3 in terms of σ and the right side - in terms of σ_d, and obtain a relation between them.
Choose Δt sufficiently small, the final simulation time t_{final} and the number of experiments N sufficiently large for solution of the following tasks in this subsection. Try different choices of Δt, t_{final}, N, and of σ, α, and present your conclusions.
Plot a few realizations of the process $X(t)$ as a function of time. Plot a histogram of $X(t_{final})$ with a superimposed Gaussian pdf. How is the random variable $X(t_{final})$ distributed?

1.1.B Expected value of $X(t)$

Compute (analytically) the expected value of $X(t)$ as a function of time.
Hint: Obtain a differential equation for it by taking the expected value of Eq. 1, and solve it. Recall that a solution of a first-order differential equation $x'(t) + ax(t) = g(t)$, $t \geq 0, x(0) = 0$ is given by:

$$x(t) = \int_0^t e^{-a(t-\eta)} g(\eta) d\eta \tag{4}$$

Compute and plot the ensemble average from N realizations of $X(t)$ as a function of time and compare it to the theoretical expected value. Compute the RMS (root mean square) error between them.
Compute the time average of $X(t)$ from each sample function and investigate whether it approaches the expected value:

$$< X(t) >_T \xrightarrow[T \to t_{final}]{} E[X(0)] \tag{5}$$

Plot $< X(t) >_T$ as a function of T for a few realizations. Compute the RMS error between $E[X(0)]$ and $< X(t) >_{t_{final}}$ based on N sample functions and compare to the error previously computed for the ensemble average.

1.1.C Autocorrelation function of $X(t)$

In order to derive an analytical expression for $R_x(t)$ take the following steps:

1. Using the definition of the mean square derivative, show that:

$$R_{Z,X'}(t_1, t_2) = \frac{\partial}{\partial t_2} R_{Z,X}(t_1, t_2) \tag{6}$$

2. Obtain a differential equation for $R_{Z,X}(t_1, t_2)$ by multiplying Eq. 1 by $Z(t_1)$ and taking the expected value. Solve it (as in Eq. 4) for $t_2 \geq 0$ and initial condition $R_{Z,X}(t_1, 0) = 0$.

3. Obtain a differential equation for $R_X(t_1, t_2)$ by multiplying Eq. 1 by $X(t_2)$ and taking the expected value. Solve it (as in Eq. 4) for $t_1 \geq 0$ and initial condition $R_X(0, t_2) = 0$. Notice that the right hand side of the equation contains $R_{Z,X}(t_1, t_2)$, for which an expression obtained in the step above should be substituted.

4. Bonus: Compute $R_X(t_1, t_2)$ directly: by substituting the expressions for $X(t_1)$, $X(t_2)$ (recall Eq. 4) into the definition of autocorrelation.

5. Substitute $t_1 = t$ and $t_2 = t + \tau$ in the expression obtained for $R_X(t_1, t_2)$. Is the process $X(t)$ stationary? WSS? becomes stationary/WSS when t approaches infinity? Is it Gaussian? Is it Markovian? Provide a detailed explanation for your answers.

Compute $R_X(t, t)$ as an ensemble average from N realizations of $X(t)$ and plot it as a function of time along with the theoretical value. What value does it approach as the time approaches infinity? At what time t_S does $R_X(t_S, t_S)$ reach 95% of its value at infinity?
Write a computer code for calculating $< X(t)X(t + \tau) >_{t_{final}}$ for each sampling function, while using the samples starting from time t_S (explain why), and for some $\tau = [0, ...\tau_{final}]$ (explain your choice of τ_{final}). Compute an ensemble average of $< X(t)X(t + \tau) >_{t_{final}}$ as a function of τ and plot it along with the theoretical autocorrelation function $R_X(\infty, \infty + \tau) = R_X(\tau)$. How many realizations are needed to provide a good estimate for $R_X(\tau)$? Plot on the same graph $< X(t)X(t + \tau) >_{t_{final}}$ for a few realizations. Is the process $X(t)$ in the steady state (for large t) mean ergodic? correlation ergodic?

1.1.D Power Spectral Density of $X(t)$

Derive an analytical expression for PSD of $X(t)$ in the steady state by computing the Fourier Transform of $R_X(\tau)$.
Compute the periodogram estimate of the PSD for each realization as a function of frequency for the frequency range $[10^{-2}, ..10^{1}]$ with a logarithmic spacing.
Compute an ensemble average of the periodogram estimates from N realizations as a function of frequency and plot it along with the theoretical PSD. Additionally, plot a few of the periodogram estimates. State your observations and conclusions.
Bonus: In order to smooth the periodogram estimate based on ensemble average the following procedure is suggested: divide the time interval $[t_S, ...t_{final}]$ into n sub-intervals of equal length, compute the periodogram estimate based on each sub-interval and store for each realization their average. This method is called *Bartlett's smoothing procedure*. Compute again the ensemble average and plot it along with the previous estimate. Try to estimate the PSD from a single realization of $X(t)$ (experiment with the number of sub-intervals, t_{final}, and Δt). Try to obtain a PSD estimate from a single realization using built-in methods of your preferred software, for example, in Matlab this could be fft(...), pwelch(...). These functions get as parameters: the frequency range, the sampling rate of the signal ($1/\Delta t$), the number of smoothing windows, and etc. Experiment with these parameters and try to obtain meaningful results.

1.1.E Cross Power Spectral Density

Repeat all tasks in section 1.1.D for cross PSD $S_{Z,X}(f)$. Hint: Recall that the periodogram estimate of $S_X(f)$ is given by $\frac{1}{T}\widetilde{x}(f)\widetilde{x}^*(f)$, where $\widetilde{x}(f)$ is the DFT of the measurement sequence $[X(0), X(\Delta t), X(2\Delta t)...X(T)]$. Analogously, $S_{Z,X}(f)$ can be estimated as: $\frac{1}{T}\widetilde{z}(f)\widetilde{x}^*(f)$.
Compute the theoretical value of the coherence function and its estimation based on the periodogram estimates of $S_{Z,X}(f)$ and $S_X(f)$. Plot both as a function of frequency. Can the process $X(t)$ be viewed as a response of a linear system to an input of white Gaussian noise? What is the transfer function $H(f)$ of this system?
Bonus: Compute the transfer function $H(f)$ from the periodogram estimate of $S_{Z,X}(f)$ and the known PSD of the input signal. Plot a bode diagram of this estimate along with theoretical values of gain and phase of $H(f)$. Can the transfer function be estimated from one realization of $X(t)$? Can the transfer function be estimated if the signal $X(t)$ is measured with noise? Assume this noise is zero mean and uncorrelated with other signals, and try to estimate $H(f)$ for various values of coherence.

1.2 Models Based on Langevin Equation

Processes based on Langevin equation are widely used in many fields of science for modeling various phenomena. For example, Ornstein-Uhlenbeck process with some modifications is used in financial mathematics to model currency exchange rates; in evolutionary biology to model phenotype development; and in engineering it is extensively used to model the turbulent flow of wind in the atmosphere, turbulent boundary layers, motion through viscous fluids, and etc.
In this section you will explore different models, all based on the Langevin equation. You are to choose one process from sub-section 1.2.A and one process from sub-section 1.2.B. For each of them repeat all the tasks of section 1.1, and elaborate on the properties of the process that have changed. In cases that the derivation of analytical expressions of correlation/PSD functions is hard, they can be evaluated numerically (e.g. approximating $\int_0^T g(t)dt$ by $\sum_{k=1}^{T/\Delta t} g(k\Delta t)\Delta t$ and computing it numerically). If additional constants are involved, set them to any values of your choice, or try different options.

1.2.A Gaussian noise

In this sub-section $Z(t)$ is zero mean and Gaussian.

1. The stochastic input to Langevin equation is not white, but equals $X(t)$ - the Ornstein-Uhlenbeck process. The new process $Y(t)$ is, thus, obtained as follows:

$$\begin{aligned} Y'(t) + \beta Y(t) = X(t), \quad t \geq 0, Y(0) = 0 \\ X'(t) + \alpha X(t) = Z(t), \quad t \geq 0, X(0) = 0, \qquad R_z(t_1, t_2) = \sigma^2\delta(t_1 - t_2) \end{aligned} \tag{7}$$

   In addition to analysis of $Y(t)$, analyze (using all appropriate tools) the correlation between $X(t)$ and $Y(t)$.

2. Motion of a particle happens in a three dimensional space, its velocity has, therefore, three components

$\vec{X}(t) = \begin{bmatrix} X_1(t) \\ X_2(t) \\ X_3(t) \end{bmatrix}$, and the process becomes:

$$\begin{aligned} \vec{X}'(t) + \alpha \vec{X}(t) &= \vec{Z}(t), \quad t \geq 0, \vec{X(0)} = [0, 0, 0]^T \\ R_{\vec{z}}(t_1, t_2) &= \begin{bmatrix} \sigma_1^2 & \nu\sigma_1^2 & \mu\sigma_1^2 \\ \nu\sigma_1^2 & \sigma_2^2 & \eta\sigma_2^2 \\ \mu\sigma_1^2 & \eta\sigma_2^2 & \sigma_3^2 \end{bmatrix} \delta(t_1 - t_2) \end{aligned} \tag{8}$$

Choose the values of ν, μ, η such that the three noise components are correlated. In addition to analyzing the process components $X_1(t), X_2(t), X_3(t)$, analyze (using all appropriate tools) the correlation between these components, and between different process and noise components, i.e. $X_i(t)$ and $Z_j(t)$ for at least one choice of $i \neq j$.

3. Consider harmonic oscillations in fluid, generated by adding a term, related to potential energy, to Langevin equation. This term is proportional to particle's position $Y(t)$, the derivative of which is the particle's velocity $X(t)$:

$$\begin{aligned} Y'(t) &= X(t), \quad t \geq 0, Y(0) = 0 \\ X'(t) + \alpha X(t) + kY(t) &= Z(t), \quad t \geq 0, X(0) = 0 \\ R_z(t_1, t_2) &= \sigma^2 \delta(t_1 - t_2) \end{aligned} \tag{9}$$

Analyzing the processes $Y(t)$, $X(t)$ (using all appropriate tools), consider using phase portraits (plot of $Y'(t)$ vs. $Y(t)$) for a better insight into the process dynamics.

1.2.B Heavy tailed noise

In this section consider the Langevin equation (Eq. 1), where the random force $Z(t)$ has a non-Gaussian distribution. In turbulence models the distributions of interest are those with heavy (not exponentially bounded) tails, that is, they have heavier tails than the exponential distribution. Can you think of a physical reason why forces distributed this way are more appropriate for describing a particle exposed to turbulence? Consider the following distributions (all can be simulated by Matlab built-in functions).

1. Log-normal
2. Pareto
3. Weibull
4. Any other heavy tailed distribution of your choice

Bonus Options: Cauchy, Lévy

1.2.C Bonus Section

One of the Langevin-type models of turbulence is the *Pitchfork bifurcation normal form*, given below: the noise induces random switching between the two metastable equilibria. This will be clear from looking at the plot of a simulated realization of Eq. 10.

$$X'(t) - \lambda X(t) + \mu X^3(t) = Z(t), \quad t \geq 0, X(0) = 0 \tag{10}$$

The stochastic force $Z(t)$ is usually assumed to be a generalized Brownian motion, called *fractional Brownian motion (fBm)*. Unlike the classical Brownian motion driving the Ornstein-Uhlenbeck process, the increments of fBm are not necessarily independent. fBm process is Gaussian with zero mean for all t, and autocorrelation:

$$R_z(t_1, t_2) = \frac{1}{2}(|t_1|^{2H} + |t_2|^{2H} - |t_1 - t_2|^{2H}) \tag{11}$$

where $H \in (0, 1)$ (called a Hurst parameter) defines the smoothness of the motion. If $H = 0.5$ we obtain a classical Brownian motion with independent increments. The increment process of fBm is called fractional Gaussian noise, and can be generated in Matlab by wfbm(H,...).

2 Wind Tunnel Experiment: Aerial Rotations

An experienced skydiver performed right and left 360 degrees turns in the vertical wind tunnel in a belly-to-earth position approximately during 2 minutes (see Fig. 1). His body posture was recorded at 240 Hz by the X-Sens body movement tracking suit, which he was wearing underneath his jumpsuit. The X-Sens suit is equipped with 16 miniature inertial sensors that are fixed at strategic locations on the body. Each unit includes a 3D accelerometer, 3D rate gyroscope, 3D magnetometer, and a barometer. Each measurement set at a given sampling instant includes the orientation of 23 body segments (pelvis, four spine segments, neck, head, shoulders, upper arms, forearms, hands, upper legs, lower legs, feet, toes) relative to the inertial frame, expressed by quaternions. These measurements were interpreted in terms of Euler angles of the following 15 joints (all together 45 degrees-of-freedom (DOFs)): abdomen, thorax, neck, shoulders, elbows, wrists, hips, knees, ankles. The order of Euler rotations and the accustomed neutral skydiving posture are shown in Fig. 2.

Fig. 1: Aerial rotations in the wind tunnel

Thus, the available data matrix has 45 rows and $t_{final}\cdot$ 240 columns: each row contains an angle of a specific DOF during the experiment time t_{final}. Additionally, the measurements of the body inertial linear and angular velocities are available (1 by $t_{final}\cdot$ 240 vectors).
For viewing the experiment see the supplemental video and for visualizing the posture measurements use the supplemental Matlab script (see instructions in Appendix 3.3; this tool is optional: animation will help to better understand the experimental procedure).
In the neutral posture the skydiver doesn't move vertically or horizontally, and doesn't rotate (i.e. falls with constant speed, called the 'terminal velocity'). The motion is caused by changing the posture and thus deflecting the surrounding airflow. Even small posture changes can result in significant aerodynamic forces and moments and enable a skydiver to perform a great variety of maneuvers, such as rotations, rolls, flips, saltas, etc.
It has been long known that multiple DOFs in the human body are not independent, but are organized by the Central Nervous System into *movement patterns*: combinations of DOFs that are activated synchronously and proportionally as a single unit. In many sports (e.g. running, cycling, skating, skiing) it was discovered that athletes need only 6-8 movement patterns for their activity. In this project you will investigate the following hypotheses with relation to body flight:

1. Aerial rotations can be performed by the means of only one movement pattern.
2. Less than 8 movement patterns are involved in tunnel rotations: some of them are needed to control the vertical and horizontal motion (in order to stay in the center of the tunnel during the maneuver).
3. The plant comprising the airflow and the skydiver's body, actuated by the 'turning' pattern, behaves as a low pass filter.

2.1 Karhunen-Loeve Expansion

The body posture $\vec{P}(t)$ (vector 45 by 1) at each instant of time can be represented as a sum of a neutral posture $\vec{P}_{neutral}$ and a weighted combination of movement patterns:

$$\vec{P}(t) = \vec{P}_{neutral} + \sum_{i=1}^{45} u_i(t)\vec{e}_i \tag{12}$$

where $\vec{e}_i$ is the eigenvector (45 by 1) representing the movement pattern i, and $u_i(t)$ is its corresponding weight, which can be viewed as the control signal: activation amplitude of the movement pattern i at time t. The aim of this section is computing the movement patterns and the matching control signals from the experimental data, and exploring the first two hypotheses, stated above. To accomplish this, follow the tasks below:

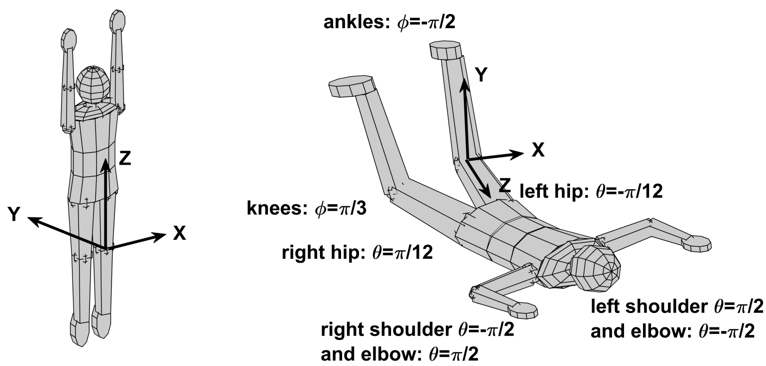

Fig. 2: Pose defined by all zero Euler angles (left), and a standard neutral pose (right). For each limb the Euler angles ψ, θ, ϕ are rotations around Z, Y, X axis of the coordinate system attached to the parent limb.

1. Plot a few of the measured signals (velocities, posture DOFs) as a function of time. Are those signals random or deterministic? State the possible reasons for your answer.

2. State under which conditions the mean and the covariance of the skydiver's posture $\vec{P}$ can be approximated as time averages of the measurements. Assume that these conditions hold and compute the neutral posture as

$$\vec{P}_{neutral} =< \vec{P}(t) >_{t_{final}} \tag{13}$$

   Let $d\vec{P}(t)$ denote the sequence of posture changes, i.e.: $d\vec{P}(t) \equiv \vec{P}(t) - \vec{P}_{neutral}$. Compute the covariance matrix of $d\vec{P}$ as

$$K_{dP} =< d\vec{P}(t)d\vec{P}^T(t) >_{t_{final}} \tag{14}$$

3. Compute the eigenvalues and eigenvectors of K_{dP} and the Karhunen-Loeve expansion of $d\vec{P}$. Find the representation of $\vec{P}(t)$ given in Eq. 12 (optional: check the result as explained in Appendix 3.3). Plot the normalized eigenvalues λ_i of the obtained movement patterns in a descending order. Plot the control signals $u_i(t)$ of the 15 most dominant movement patterns (with the largest λ) as functions of time. What conclusions can be made regarding the amount of meaningful patterns? Can the patterns with $\lambda < 0.2$ be neglected? Plot the measured yaw rate signal and compare it to the control signals of dominant patterns. State your conclusions.

4. Consider the control signals $u_i(t)$ of the first 4-6 patterns (with the largest λ), and the measurements $z_j(t), \quad j = 1, 2, ..$ conveying inertial motion (angular and linear velocities). Under which conditions the cross correlation $R_{u_i,z_j}(t_1, t_2)$ is a function of only $\tau = t_2 - t_1$? Assume that these conditions hold and compute the correlation coefficient $\rho_{u_i,z_j}(\tau)$ as a time average. Organize the results into relevant plots. At which τ is $|\rho|$ getting its maximal values? Explain why. Is there evidence in favor of the assumption that each movement pattern is responsible for generating a certain type of angular/ linear motion? Suggest ways (experimental and/or theoretical) to validate this assumption.

5. (Bonus): In human movement sciences it is accustomed to find movement patterns by the means of Principal Component Analysis (PCA), which is the Singular Value Decomposition (SVD) of the matrix $[d\vec{P}(t_1), d\vec{P}(t_2), ...d\vec{P}(t_{final})]^T$ ($t_{final}\cdot$ 240 by 45). Compute the SVD and verify the previously obtained results. Explain why PCA is mathematically equivalent to the Karhunen-Loeve expansion. In PCA, the eigenvectors $\vec{e}_i$ are called Principal Components. Can you explain why, and explain the geometrical interpretation of control signals $u_i(t)$?

2.2 Bonus: Estimation of the transfer function

The goal in this section is to explore the third hypothesis. Recall that the maneuver performed in the experiment involved rotations in the horizontal plane, and assume that the first Principal Component, i.e. the movement pattern with the largest λ, is the one producing these rotations. The transfer function from the angle of this pattern ($u_1(t)$) to yaw rate can be estimated via the PSD and cross PSD of these signals.
State conditions under which the estimation of PSD from a single realization of a random process may produce meaningful results. (Recall the task in 1.1.D and Bartlett's smoothing procedure.)

Assume that these conditions hold and estimate the plant under investigation. Present the relevant plots. Why is it reasonable to expect that this system will resemble behavior of a low pass filter? Subplot a theoretical gain and phase of a suitable low pass filter. Explain the inevitable discrepancies between the theoretical and estimated transfer functions at low and high frequencies. Can you think of reasons to expect a different (from a low pass filter) behavior (utilize your knowledge in aerodynamics and control theory, and imagination)?

3 Appendix: Useful Matlab Functions

3.1 Generating a vector of iid random variables

The following functions can be used to generate a vector (1 by N) of uncorrelated Gaussian random variables $\sim N(\mu, \sigma^2)$:

1. X=normrnd(μ, σ, 1, N)
2. pd=makedist('Normal', 'mu', μ, 'sigma', σ)
   X=random(pd,1,N)

To view the histogram of the generated data with a superimposed normal pdf:

1. histfit(X, 100) This will generate a histogram with 100 bars
2. x=-100:100
   plot(x, pdf(pd, x)) This will plot a theoretical pdf curve

Other distributions can be generated and viewed in a similar way:

1. Lognormal: pd=makedist('lognormal', 'mu', μ, 'sigma', σ)
2. Pareto: pd=makedist('gp', 'k', k, 'sigma', σ, 'theta', 0)
3. Weibull: pd=makedist('weibull', 'B', k, 'A', σ)

Notice, that these distributions produce a random variable X=random(pd,1,N) with a non-zero expected value:

$$\text{Lognormal: } E[X] = e^{\mu+\sigma^2/2}$$
$$\text{Pareto: } E[X] = \frac{\sigma}{1-k}$$
$$\text{Weibull: } E[X] = \sigma\Gamma(1+\frac{1}{k})$$

where $\Gamma(.)$ function can be computed by Matlab's function gamma(.).
Cauchy and Lévy distributions are private cases of the *stable* distribution with parameters:

1. Cauchy: pd= makedist('Stable', 'alpha', 1, 'beta', 0, 'gam', σ, 'delta', 0)
2. Lévy: pd=makedist('stable', 'alpha', 0.5, 'beta', 1, 'gam', σ, 'delta', 0)

3.2 Useful functions

- Generating a vector $f = [10^a, ..10^b]$ with a logarithmic spacing: f=logspace(a,b)
- Plotting y vs x such that the axis X has a logarithmic spacing: semilogx(x,y)
- Computing RMS (root mean square) from elements of a vector: rms(X)
- Computing a cumulative sum of elements of a vector: cumsum(X)
- Computing exponential of a matrix e^A: expm(A) (as opposed to an exponential of a scalar x: exp(x))
- Operations with complex numbers: finding the real part real(X), imaginary part imag(X), and complex conjugate conj(X)
- Computing logarithms: $ln(X)$: log(X), $log_{10}(X)$: log10(X)
- Computing eigenvalues (in a diagonal matrix D) and the corresponding eigenvectors (columns of matrix V) of a matrix A: [V,D] = eig(A)
- Computing singular value decomposition of a matrix $A = U \cdot S \cdot V^T$: [U, S, V]=svd(A) (S - diagonal matrix with singular values; U, V - unitary matrices)
- Importing data from *file.txt*: T=importdata('file.txt')

3.3 Visualizing the posture sequences

In order to obtain a better understanding of the posture measurements and computed movement patterns, the following Matlab scrips are available. The scripts assign the posture sequences to a human avatar and present its animation.

1. Viewing the measured postures:

   T=importdata('DOF_measurements.txt')

   animate_mov_data(T.data)

2. Viewing the same postures, but represented via Eq. 12:

   animate_mov_component_matrix(neutral_pose, PC, signals)

   where: neutral_pose is $\vec{P}_{neutral}$; PC is the matrix (45 by 45) containing the eigenvectors $[\vec{e}_1, .. \quad \vec{e}_{45}]$; and signals is the matrix (45 by $t_{final}\cdot$ 240) containing control signals $\begin{bmatrix} u_1(t_0), .. & u_1(t_{final}) \\ ... & \\ u_{45}(t_0), .. & u_{45}(t_{final}) \end{bmatrix}$

3. Animating a movement pattern, represented by its eigenvector $\vec{e}$:

   animate_mov_component(neutral_pose, PC, recording)

   where neutral_pose is $\vec{P}_{neutral}$; PC is the eigenvector $\vec{e}$; and recording - is a flag 1 or 0 to enable/disable the animation recording, respectively. If enabled - the animation will be saved to the working directory. The animation will be 10 sec long, and the control signal for the inputted movement pattern will be a sine wave: $\vec{u}(t) = sin(2 \cdot \pi \cdot 0.25 \cdot t)$