Introduction to Stochastic Differential Equations
MATH 6490–1 – Spring 2008
Homework 1

Due Date: Friday, May 2 at 5:00 PM

This homework has 140 points plus 30 bonus points available but, as always, homeworks are graded out of 100 points. Full credit will generally be awarded for a solution only if it is both correctly and efficiently presented using the techniques covered in the lecture and readings, and if the reasoning is properly explained. If you used software or simulations in solving a problem, be sure to include your code, simulation results, and/or worksheets documenting your work. If you score more than 100 points, the extra points do count toward your homework total.

1 Theoretical Calculations

1.1 Conditional Distributions of Partial Sums of Gaussian Random Sequence (15 points)

Suppose that $\{X_j\}_{j=1}^{\infty}$ is a sequence of independent, identically distributed Gaussian random variables with mean μ and standard deviation σ, and define the partial sums $S_n = \sum_{j=1}^{n} X_j$. Calculate and simplify the conditional probability density function for S_m given the value of S_n, where $n > m$.

1.2 Angular Distribution of Gaussian Random Variable Pair (20 points)

Suppose that X_1 and X_2 are jointly Gaussian random variables with mean 0 and the same variance σ^2 and correlation coefficient ρ.

a. (**10 points**) Derive the joint probability density function of the polar coordinate representation (R, Θ), where $X_1 = R\cos\Theta$ and $X_2 = R\sin\Theta$.

b. (**5 points**) Derive the marginal probability density function for the angle variable Θ.

c. (**5 points**) Calculate the probability that X_1 and X_2 will have the same sign.

1.3 Servomechanism Model (55 points plus 30 bonus points)

A servomechanism is designed to stabilize a system against external, unpredictable disturbances through a feedback control. For this problem, we will satisfy ourselves with a model in which time is discretized, so that, at the nth time step, the state of the system is denoted X_n, the disturbance is denoted Z_n, and the state of the controller is denoted by C_n. We model the control feedback loop through a stochastic update rule:

$$X_{n+1} = X_n + C_n + Z_{n+1}, \qquad n = 0, 1, 2, \ldots$$

where the disturbances $\{Z_n\}_{n=0}^{\infty}$ are independent, identically distributed Gaussian random variables with mean 0 and standard deviation σ_Z. Moreover, the control must obey the causality rule: for each $n \geq 0$, C_n can only depend on X_0, X_1, $X_2, \ldots, X_n$ and Z_1, $Z_2, \ldots Z_n$. (The controller cannot respond to the future!) A good control will minimize the fluctuations in X_n as well as in C_n (so that the controller does not have to work so hard). Finally, we assume the initial state of the system X_0 is a Gaussian random variable with mean μ_0 and standard deviation σ_0.

a. (**5 points**) Explain whether or not the state of the controller and/or the state of the system is a Markov process with respect to the filtration generated by the disturbances $\{Z_n\}_{n=0}^{\infty}$ and the initial state X_0.

b. (**5 points**) Explain whether or not X_n must be a Gaussian random variable.

c. (**10 points**) Write down a general formula for the variance of X_n in terms of the statistics of the noise and the controller states. At this point, make no assumption about the control beyond the general condition described above.

d. (**5 points**) Show that, for $n \geq 1$, the standard deviation of X_n must be at least equal to σ_Z, regardless of how the controller is designed.

e. (**10 points**) Suppose that the servomechanism runs for a long time and settles down into a statistically stationary state, meaning that all *statistics* of (but not necessarily the individual realizations of) X_n and C_n approach constant values at large n. In particular, we define

$$\bar{\sigma}_C^2 \equiv \lim_{n\to\infty} \mathrm{Var}\,(C_n), \qquad \bar{\sigma}_X^2 \equiv \lim_{n\to\infty} \mathrm{Var}\,(X_n).$$

Show that, no matter how the control is designed, the following inequality must hold:

$$\bar{\sigma}_X \geq \frac{1}{2}(\bar{\sigma}_C + \sigma_Z^2/\bar{\sigma}_C) \tag{1}$$

f. (**10 points**) Suppose the control is linear and local in time: $C_n = a(X_n - b)$, where a and b are real constants. Calculate the mean state of the system $\langle X_n \rangle$ as well as the covariance of the state of the system $\langle (X_n - \langle X_n \rangle)(X_m - \langle X_m \rangle) \rangle$ for all $m \geq 0$ and $n \geq 0$. You should express your answer explicitly in terms of given parameters of the problem.

g. (**5 points**) For what values of a and b do the mean and variance of the system state approach constant values (thereby suggesting a statistically stationary state)? What are these asymptotic values?

h. (**5 points**) Show explicitly using your results from the previous part that for these values of a and b, the inequality (1) is satisfied by the linear controller.

i. (**30 bonus points**) Extend the above results to the case in which the controller has a time delay of D time steps, so that the control C_n applied at time n can depend only on X_0, X_1, $X_2, \ldots X_{n-D}$ and Z_1, $Z_2, \ldots Z_{n-D}$.

2 Numerical Computations

2.1 Correlated Gaussian Random Variable Simulation (50 points)

a. (**15 points**) Write a code to generate a pair of Gaussian random variables X_1 and X_2 with arbitrary means $\mu_j = \langle X_j \rangle$, arbitrary positive variances $\sigma_j^2 = \langle (X_j - \mu_j)^2 \rangle$, and arbitrary correlation coefficient $-1 \leq \rho \leq 1$. You should explain your code clearly (through comments in the code or on a separate page), describing how it relates to the theory of simulating random variables developed in the lectures. If you have an algorithm that improves upon the ideas described in class, explain the basis and justification for your method. You should write your own Gaussian random number generator with two versions: the Box-Muller Method and the Polar Marsaglia Method. You may use existing software for simulating a uniform random number on the unit interval.

b. (**5 points**) Modify your code to simulate N independent pairs of Gaussian random variables, with each pair having the statistical properties described above. (In particular, the random variables within each pair are not independent if $\rho \neq 0$.) Compare how fast your code can simulate a given number N of pairs of Gaussian random variables when you use the Box-Muller method as compared to the Polar Marsaglia method. Try a few large values of N, and report the times required for the simulations and the times required per pair of random numbers to be simulated. Comment on the behavior of the algorithms for not-so-large and for large values of N.

c. (**5 points**) Simulate $N = 1000$ pairs of random variables $\{(X_1^{(n)}, X_2^{(n)}\}_{n=1}^{N}$ using both versions of your code, and calculate the following statistical estimators:

- Sample means:

$$\hat{\mu}_j = \frac{1}{N} \sum_{n=1}^{N} X_j^{(n)}$$

- Sample variances:

$$\hat{\sigma}_j^2 = \frac{1}{N-1} \sum_{n=1}^{N} (X_j^{(n)} - \hat{\mu}_j)^2$$

- Sample covariance:

$$\hat{C}_{12} = \frac{1}{N-1} \sum_{n=1}^{N} (X_1^{(n)} - \hat{\mu}_1)(X_2^{(n)} - \hat{\mu}_2)$$

Compare these sample estimators to the correct theoretical values.

You may be puzzled by why the sample variance has $N-1$ rather than N in the denominator. The reason is statisical: dividing by N would produce an underestimation bias of the variance. That's because you're computing the fluctuations from your *sample* mean, but your *sample* mean is itself a random variable with variance σ^2/N. To correct for this (since you're trying to estimate the mean-square fluctuations with respect to the *true* mean μ), the denominator for $\hat{\sigma}^2$ is made to be $N-1$. Of course, this makes little practical difference when N is large.

d. (**10 points**) Assuming your code is working properly, suppose you compile it and send the executable to someone who does not have access to the source code. All she knows is that the code generates a user-specified N copies of 2 random variables, and each pair is generated in an identical fashion, independently of the other pairs. How should she decide based on data from the simulations whether, according to the underlying probability distribution, the random variables within each pair are independent or not? Apply your method for determining dependence/independence to some data generated by your code for cases in which the random variables within the pair are uncorrelated, weakly correlated, and strongly correlated.

e. (**10 points**) If you worked part b of Problem 1.2, verify your theoretical calculation through random variable simulations with your code. To do so, divide the range of the random variable into bins of reasonably small size, and for each bin, calculate the fraction of simulations in which the random variable landed in that bin. This value should be compared against the theoretical probability for the random variable to fall in that bin. One good way to display the comparison is to plot these two sets of values as histograms versus the value of the random variable, with bins identified by their central value. Perform these comparisons for an uncorrelated, weakly correlated, and strongly correlated example.

f. (**5 points**) If you worked part c of Problem 1.2, verify your theoretical answer through random variable simulation for an uncorrelated, a weakly correlated case, and a strongly correlated case.