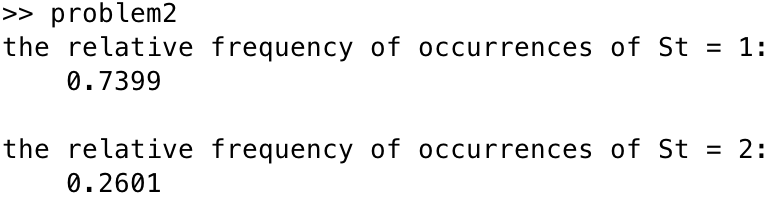
**Problem1**

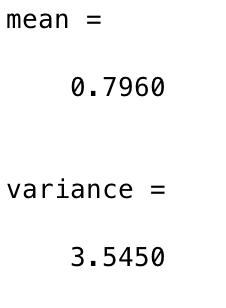
,

**problem2**

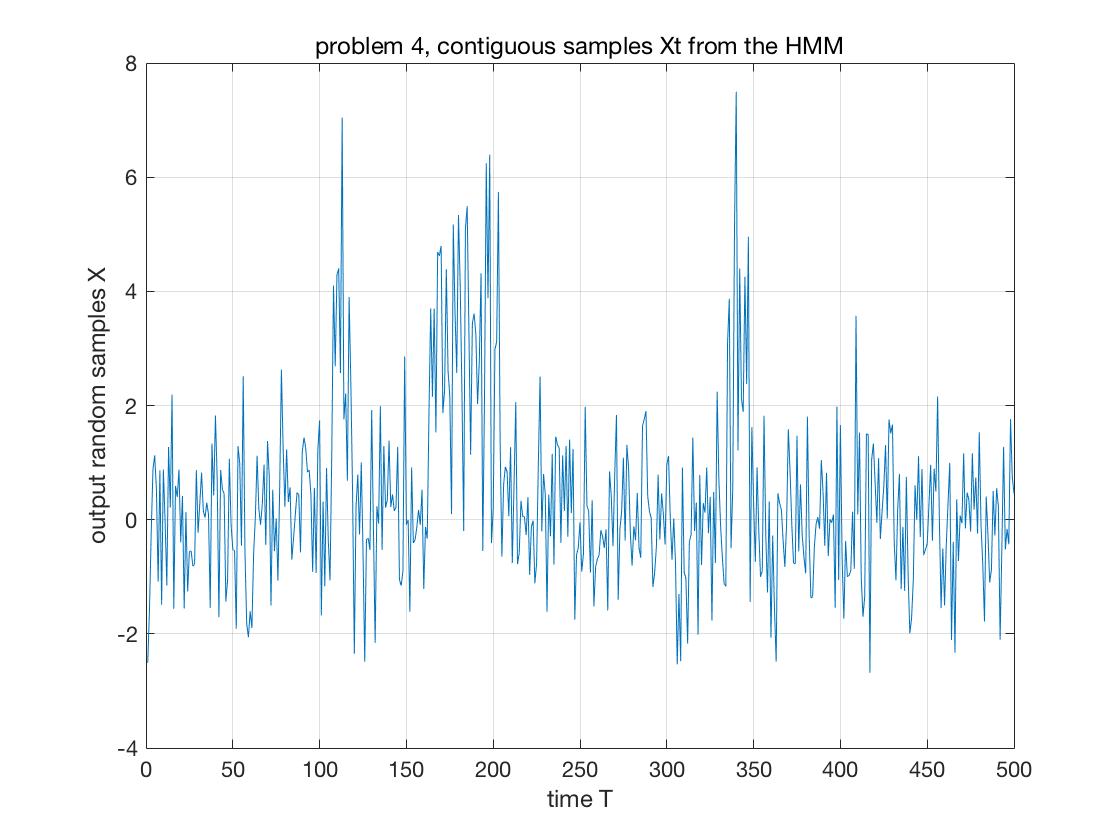
below picture is the result from MATLAB, the relative frequency of occurrences of St = 1 and St = 2 are 0.7399 and 0.2601 respectively, which approximately equals to P(St);



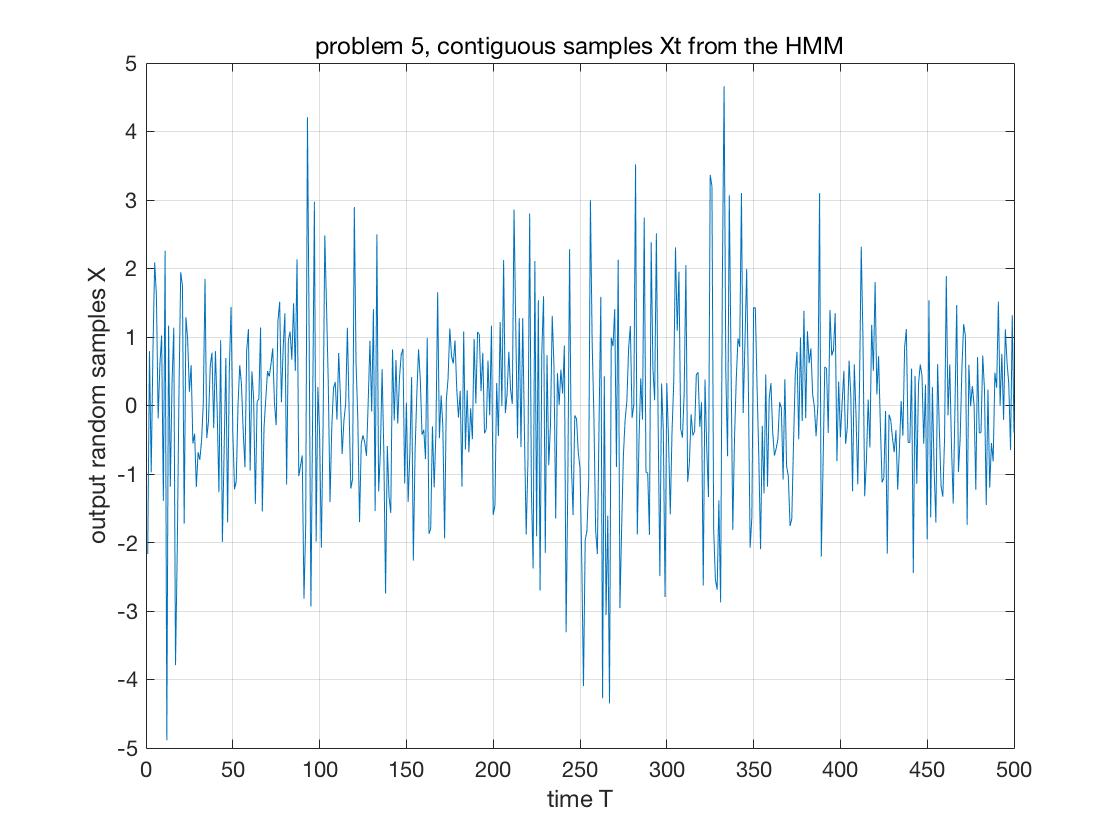
**problem 3**



**problem 4**



**problem 5**



similarity:

the variance keeps the same, so the shape that output points spreads out is quite same with the problem 4.

difference:

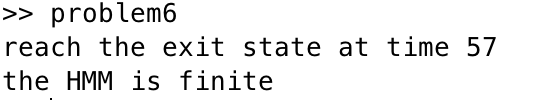
since the mean of two Gaussian distribution is the same, so for one output point is difficult to distinguish which state it belongs to.

**Problem 6:**

Change the original, square transition probability matrix into the following:

mc = MarkovChain([0.75;0.25], [0.97 0.02 0.01;0.03 0.95 0.02]);

run the code we get the output



**problem 7:**

figure 1:

distribution 1, mean[0;0], covariance[2 1;1 4]

distribution 2, mean[0;0], covariance[2 0;0 4]

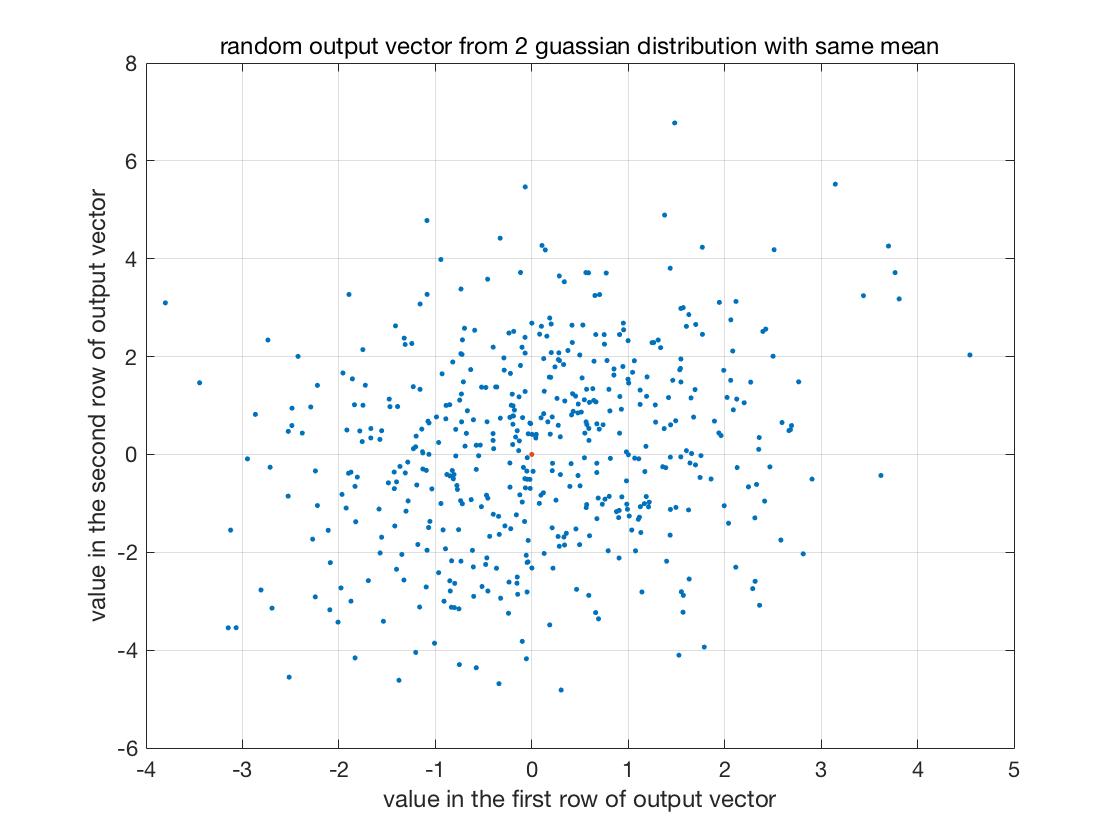
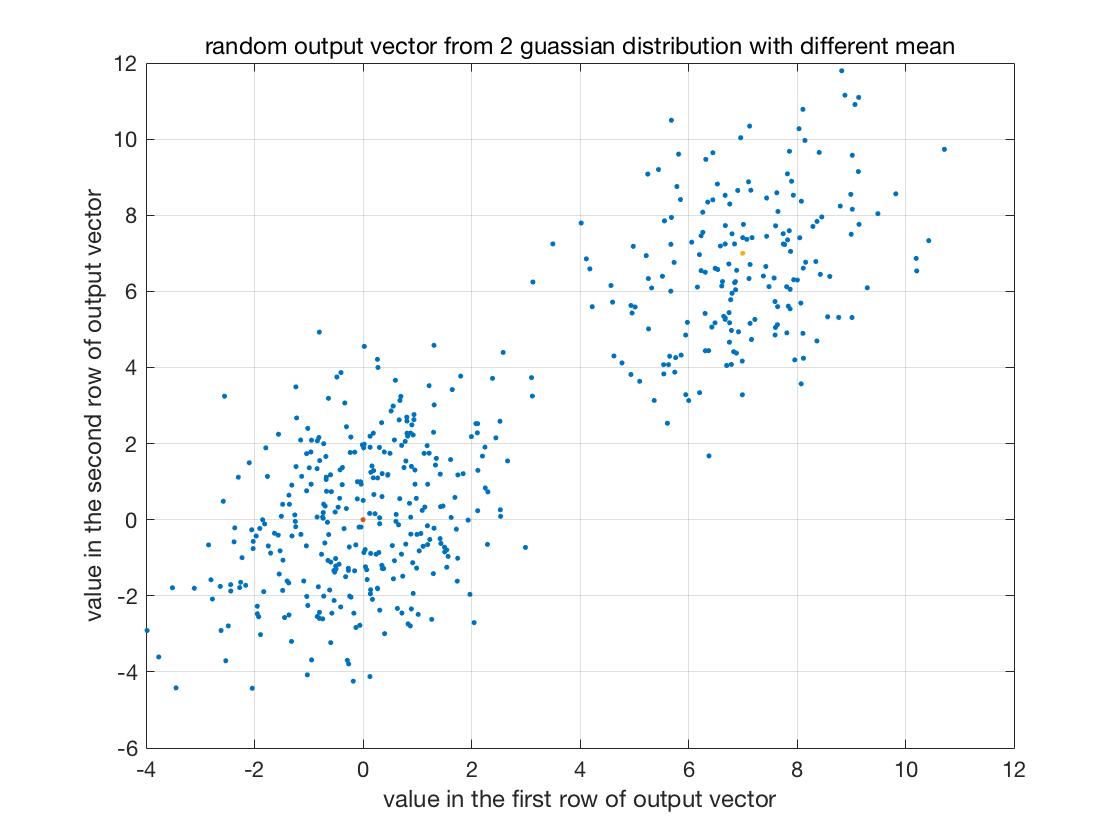


figure 2:

distribution 1, mean[0;0], covariance[2 1;1 4]

distribution 2, mean[5;5], covariance[2 1;1 4]



the result above indicates all codes are right. and from the second figure, it shows that all data points are clearly divide into 2 clusters because of different means.

Red points show where the mean vector locates.

QUESTION 6

In the code below, simply use the function if to decide whether the loop reaches the exit state. If yes, jump out of the loop. Otherwise, the loop continues.

if index == nS + 1

        disp(['reach the exit state at time ', num2str(t)])

        break

    end

In order to test this code also works for finite-durations HMM, the probability transition matrix needs to be modified while the initial state keeps the same.

The following is the outcome from MATLAB, easily find the loop terminated at time 43 because it reaches the exit state, the corresponding length of output sequences should be T-1=42.

>> problem6

reach the exit state at time 43

the HMM is finite, the length of output sequences is 42

Question 7

In order to test the code is also suitable for vector-value, the output distribution needs to be modified. Here we use 2-dimension GAUSSIAN distribution instead of 1-dimension. The following is how 2-dimension GAUSSIAN distributions are formulated in the code. Here the given GaussD function is used to build a object/distribution.

u1 = [0;0];

cov1 = [2 1;1 4];

u2 = [0;0];

cov2 = [2 0;0 4];

u3 = [7;7];

cov3 = [2 1;1 4];

g1 = GaussD('Mean',u1,'Covariance',cov1);

g2 = GaussD('Mean',u2,'Covariance',cov2);

g3 = GaussD('Mean',u3,'Covariance',cov3);