ESE 650 - Learning in Robotics Hidden Markov Models – Gesture Recognition

Vishnu Purushothaman Sreenivasan

Introduction

Problem Statement

The objective of the project is to use data from the IMU of a phone (3 axis gyroscope and 3 axis accelerometer) and predict the gestures performed. We have the 6 following gestures wave, figure 8, circle, pendulum, hammer and fish. We have to use Hidden Markov Models (HMM) to predict the probability of each sequence of observation to belong to gesture.

Description of approach

I removed the magnetometer data and the time stamps. By visually inspecting the variation in the sensor values, I trimmed of the regions where the sensors are moving randomly before starting of the gestures. I vector quantized the 6 dimensional space of accelerometer and gyroscope values into 20 clusters using data from all the gestures. Assuming a total of 15 hidden states, I performed training using EM or Baum-Welch algorithm to my Hidden Markov Model. I used the recursive forward backward algorithm to compute the intermediate values. In order to account for the numerical instabilities I used the scaling technique mentioned in the Lawrence R. Rabiner's paper on "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". For the testing, I computed the log probability of the sequence using the Forward Algorithm. I refer to the equations in the above mentioned paper throughout this report, please refer to the paper for the equations.

Methodology

The methodology can be split into the training and the testing part:

Training

a) I first removed the time stamps and the magnetometer data. I grouped all the data and clustered those using KMeans function in Matlab with 20 clusters.

b) I trained six different models for each gesture using HMM assuming 15 hidden states.

c) Each observation sequence is vector quantized using Euclidean distance to the centroids computed using K means.

d) I randomly initialized my state transition probability matrix (A), my emission matrix (B) and my initial probability matrix (Pi).

$$A = \begin{pmatrix} P(1to1) & \dots & P(1toN) \\ \vdots & \ddots & \vdots \\ P(Nto1) & \dots & P(NtoN) \end{pmatrix}$$

$$B = \begin{pmatrix} P(1emit1) & \dots & P(1emitM) \\ \vdots & \ddots & \vdots \\ P(Nemit1) & \dots & P(NemitM) \end{pmatrix}$$

where, N is the total number of states which is 15 and M is the total number of symbols or the types of emission which is 20.

So, the rows of A and B have to sum to 1.

e) In the training of the HMM, I first performed the forward algorithm according to equations (19), (20) and (21). But I also introduced scaling according to the equations (92a), (92b), (93a) and (93b).

1) Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \le i \le N. \tag{19}$$

2) Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \quad 1 \le t \le T-1$$

$$1 \le j \le N. \tag{20}$$

3) Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{21}$$

That is at each step of time I computed the sum of the α_t for all the states and divided each state by the sum and stored the scaling factor c_t.

$$c_t = \frac{1}{\sum_{i=1}^{N} \alpha_t(i)}. \quad (91)$$

Thus, for a fixed t, we first compute

$$\alpha_t(i) = \sum_{j=1}^{N} \hat{\alpha}_{t-1}(j)\, a_{ij} b_j(O_t). \quad (92a)$$

Then the scaled coefficient set $\hat{\alpha}_t(i)$ is computed as

$$\hat{\alpha}_t(i) = \frac{\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j)\, a_{ij} b_j(O_t)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \hat{\alpha}_{t-1}(j)\, a_{ij} b_j(O_t)}. \quad (92b)$$

By induction we can write $\hat{\alpha}_{t-1}(j)$ as

$$\hat{\alpha}_{t-1}(j) = \left(\prod_{\tau=1}^{t-1} c_\tau \right) \alpha_{t-1}(j). \quad (93a)$$

$$\hat{\alpha}_t(i) = \frac{\sum_{j=1}^{N} \alpha_{t-1}(j) \left(\prod_{\tau=1}^{t-1} c_\tau \right) a_{ij} b_j(O_t)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_{t-1}(j) \left(\prod_{\tau=1}^{t-1} c_\tau \right) a_{ij} b_j(O_t)} = \frac{\alpha_t(i)}{\sum_{i=1}^{N} \alpha_t(i)} \quad (9$$

f) Now I computed the backward recursion variable β_t according to equations (24) and (25).

1) Initialization:

$$\beta_T(i) = 1, \quad 1 \le i \le N. \quad (24)$$

2) Induction:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1})\, \beta_{t+1}(j),$$

$$t = T-1, T-2, \cdots, 1, \quad 1 \le i \le N. \quad (25)$$

But at every time step I rescaled β_t by multiplying with $c_{(t+1)}$.

g) Now I compute $\xi_t(i,j)$ which is the probability of being in state S_i; at time t, and state S_j at time t + 1,

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda). \quad (\text{Testing}$$

$$\xi_t(i,j) = \frac{\alpha_t(i)\, a_{ij} b_j(O_{t+1})\, \beta_{t+1}(j)}{P(O|\lambda)}$$

$$= \frac{\alpha_t(i)\, a_{ij} b_j(O_{t+1})\, \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i)\, a_{ij} b_j(O_{t+1})\, \beta_{t+1}(j)} \quad ($$

But I compute $\xi_t(i,j)$ with rescaled α and rescaled β. I do not normalize the denominator but rather multiply it with c(t+1) as this takes care of the normalization and the scaling

h) I compute the value of γ_t according to the following equations

$$\gamma_t(i) = \hat{\alpha}_t(i)\hat{\beta}_t(i)$$

i) Now I compute all of the above for observation sequence and recompute the A, B and Pi values according to the following equation,

$$\bar{\pi}_i = \frac{\sum_{n=1}^{nex} \gamma_1(i,n)}{nex},$$

$$\bar{a}_{ij} = \frac{\sum_{n=1}^{nex} \sum_{t=1}^{T_n - 1} \xi_t(i,j,n)}{\sum_{n=1}^{nex} \sum_{t=1}^{T_n - 1} \gamma_t(i,n)}$$

$$\bar{b}_j(k) = \frac{\sum_{n=1}^{nex} \sum_{\substack{t=1 \\ O_t^n = k}}^{T_n} \gamma_t(j,n)}{\sum_{n=1}^{nex} \sum_{t=1}^{T_n} \gamma_t(j,n)}$$

where n to nex are the different observation sequences.

j) Now the above procedure is repeated for 100 iterations or if the difference between the parameters is less than 1e-6.

k) The log probability is computed using

$$\log [P(O|\lambda)] = -\sum_{t=1}^{T} \log c_t. \quad (103)$$

Testing

a) The input file is loaded and the gyroscope and accelerometer values are extracted. Using the centroids computed in training the data is vector quantized.

b) Using all the models, forward algorithm is run and the corresponding log probability is obtained using the following equation for each of the model.

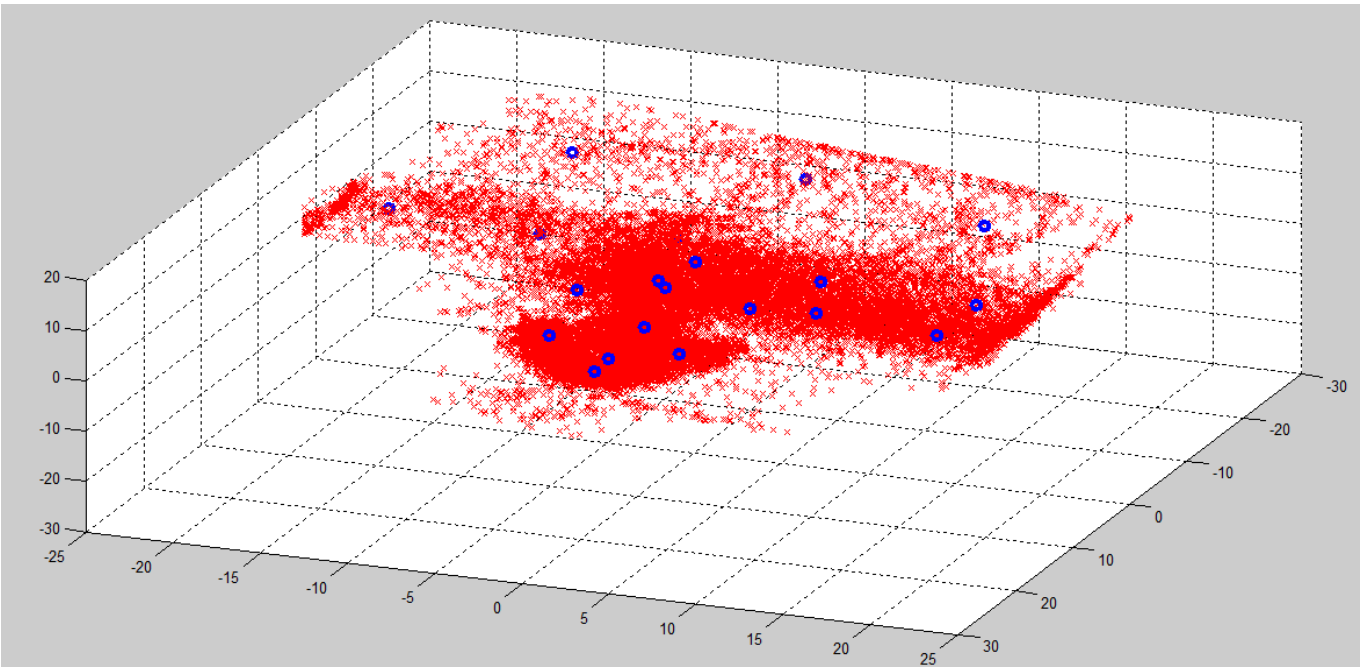
$$\log [P(O|\lambda)] = -\sum_{t=1}^{T} \log c_t. \qquad (103)$$

c) The gesture corresponding to model with the highest log probability is predicted as the gesture.

Results

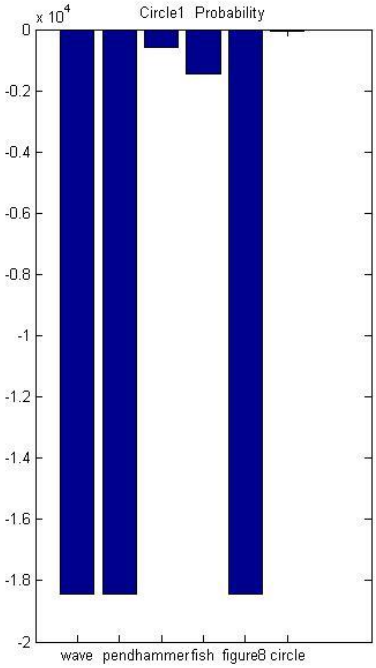
The algorithm worked very well on the training data on cross validation by training using 3 random observation sequences for each gesture and testing on the 2 unseen sequences. One of the problems faced initially was that the probability estimates were not very accurate as the initial observation sequence not corresponding to the gesture was included in the model training. Trimming this region radically improved the probability estimates. A model with 20 observations and 15 states seems to give best result using cross validation.

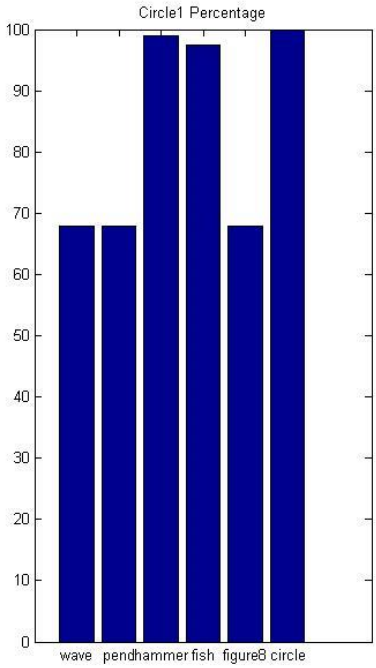
Given below is a PCA'ed clustering in 3 dimensions.

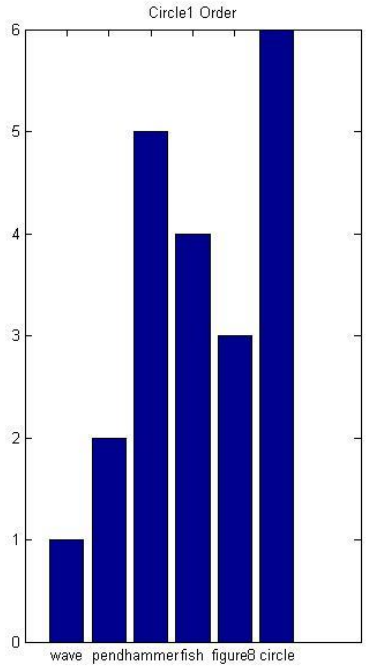


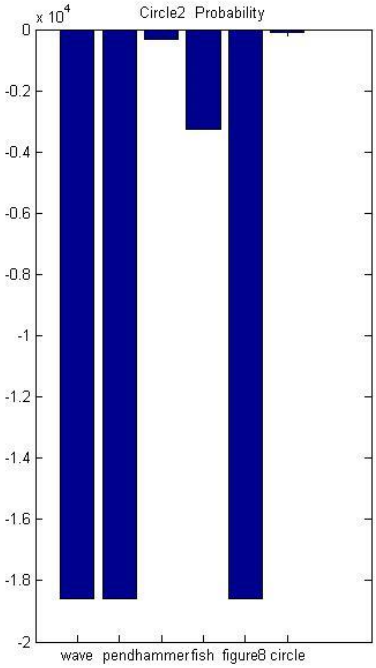
The following are the results of in sample testing on the untrimmed data. Cross validation gave almost exactly similar results.

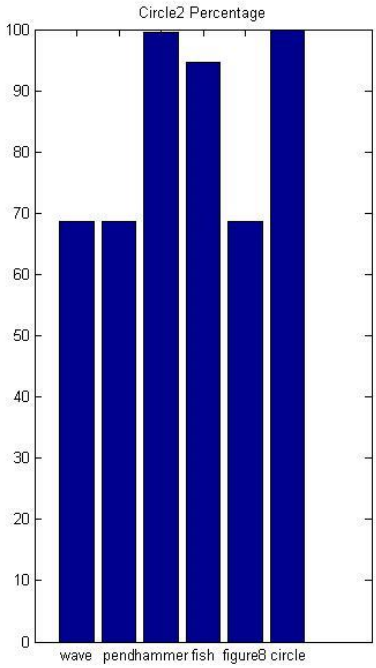
Please zoom in for better visibility.

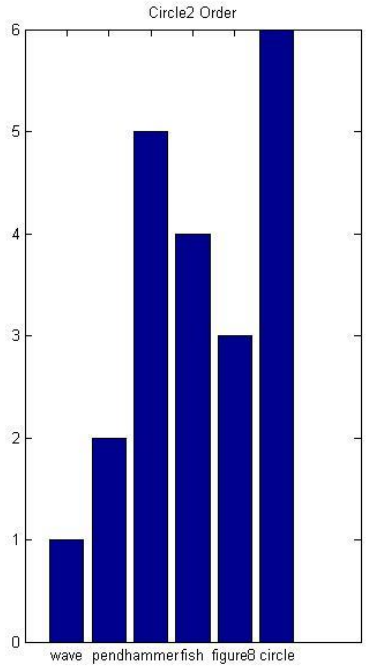

Circle1 Probability
x 10^4
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
-1.4
-1.6
-1.8
-2
wave
pendhammer
fish
figure8
circle

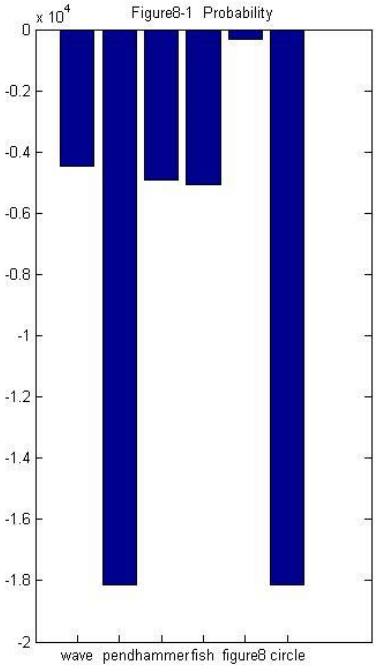

Circle1 Percentage
100
90
80
70
60
50
40
30
20
10
0
wave
pendhammer
fish
figure8
circle


Circle1 Order
6
5
4
3
2
1
0
wave
pendhammer
fish
figure8
circle


Circle2 Probability
x 10^4
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
-1.4
-1.6
-1.8
-2
wave
pendhammer
fish
figure8
circle


Circle2 Percentage
100
90
80
70
60
50
40
30
20
10
0
wave
pendhammer
fish
figure8
circle


Circle2 Order
6
5
4
3
2
1
0
wave
pendhammer
fish
figure8
circle


x 10^4
Figure8-1 Probability
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
-1.4
-1.6
-1.8
-2
wave
pendhammer
fish
figure8
circle

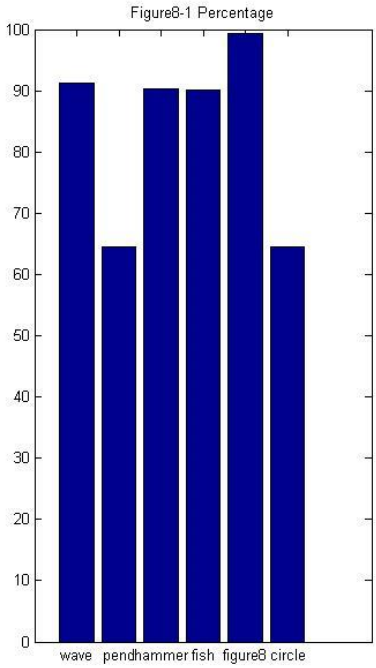


Figure8-1 Percentage
100
90
80
70
60
50
40
30
20
10
0
wave
pendhammer
fish
figure8
circle

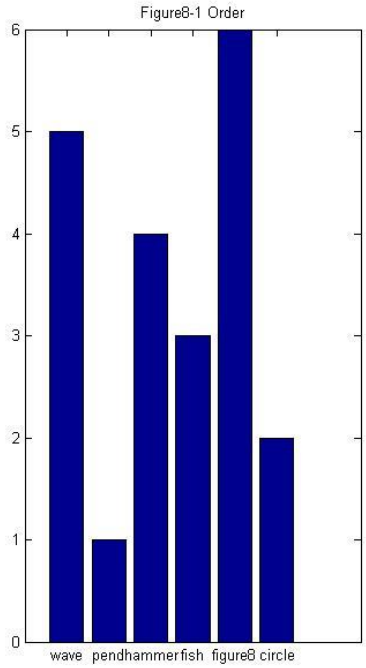
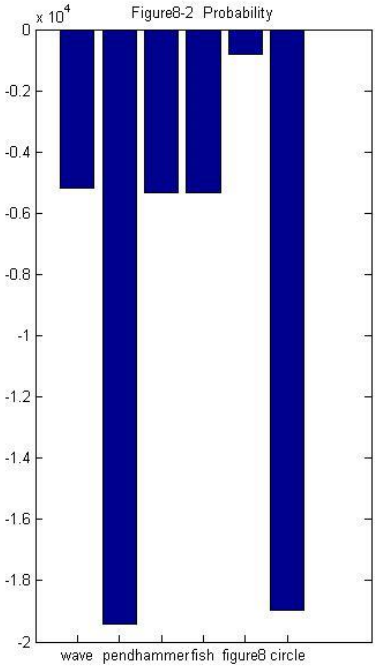


Figure8-1 Order
6
5
4
3
2
1
0
wave
pendhammer
fish
figure8
circle


x 10^4
Figure8-2 Probability
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
-1.4
-1.6
-1.8
-2
wave
pendhammer
fish
figure8
circle

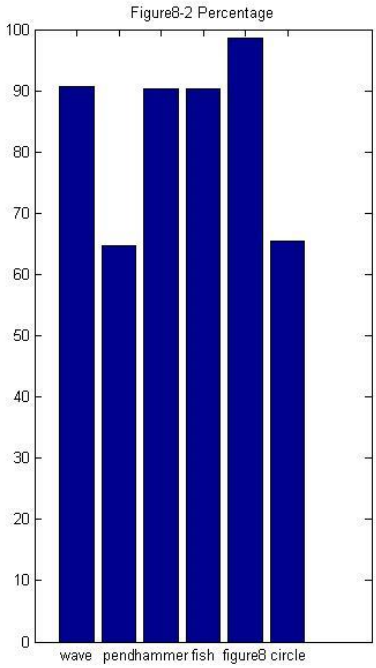


Figure8-2 Percentage
100
90
80
70
60
50
40
30
20
10
0
wave
pendhammer
fish
figure8
circle

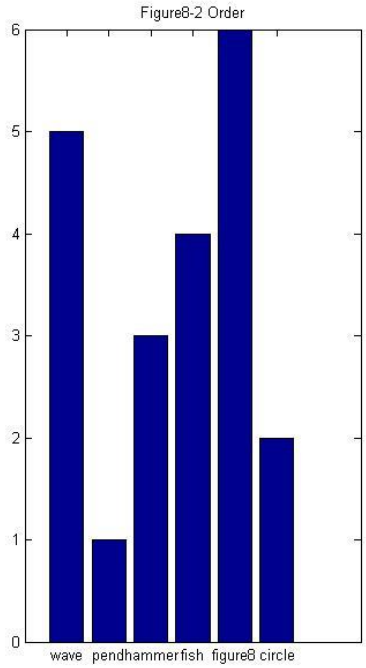
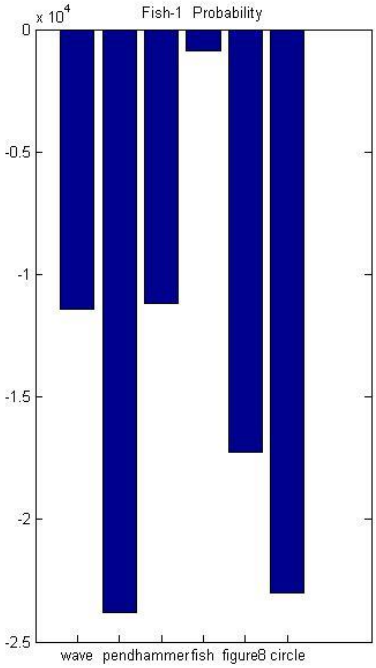
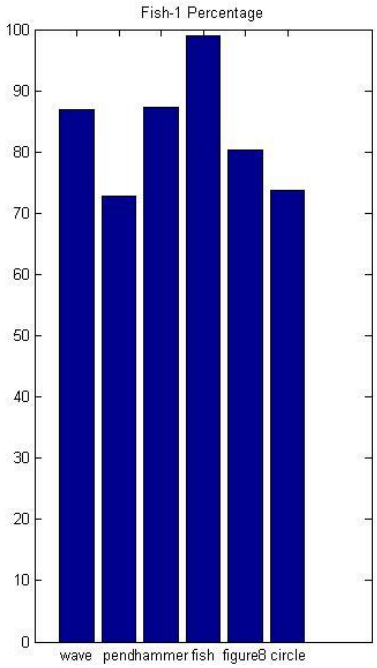
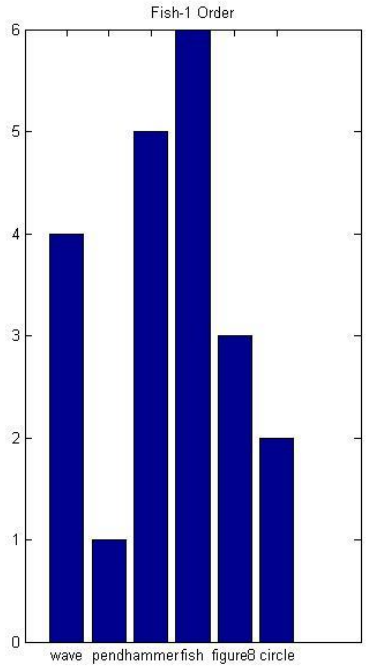
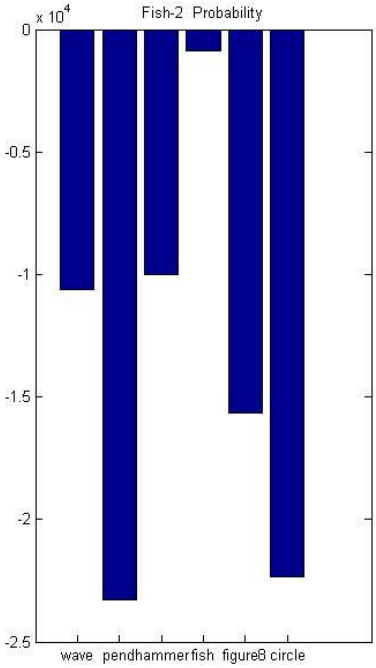


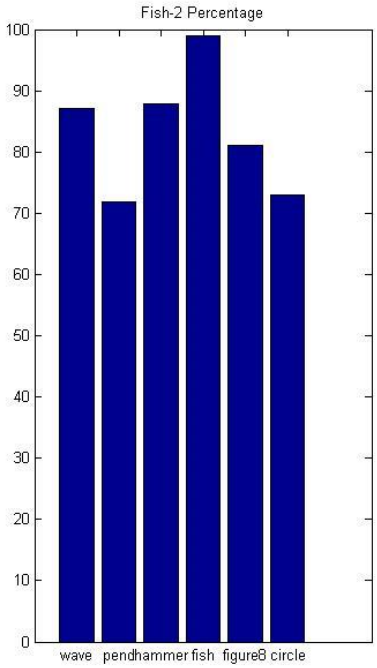
Figure8-2 Order
6
5
4
3
2
1
0
wave
pendhammer
fish
figure8
circle

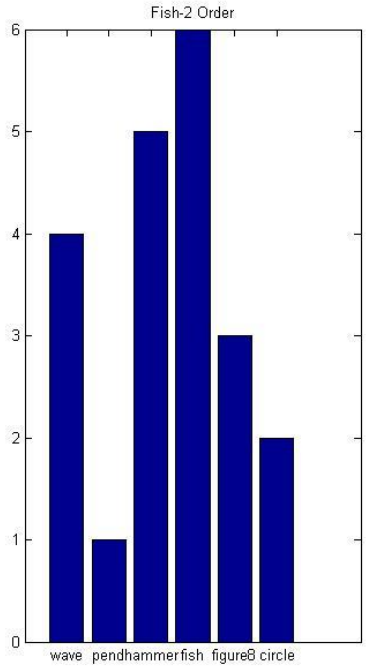

x 10^4
Fish-1 Probability
0
-0.5
-1
-1.5
-2
-2.5
wave
pendhammer
fish
figure8
circle

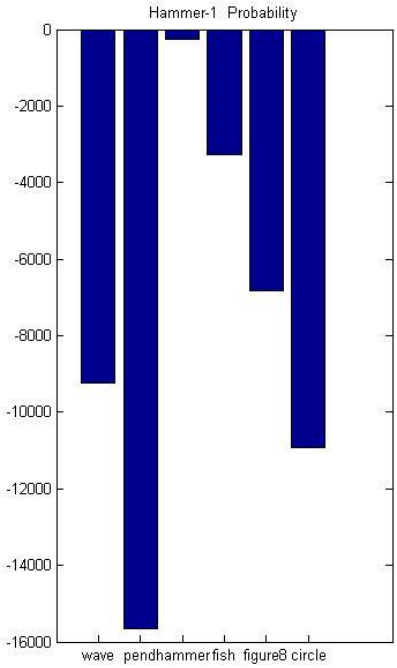

Fish-1 Percentage
100
90
80
70
60
50
40
30
20
10
0
wave
pendhammer
fish
figure8
circle

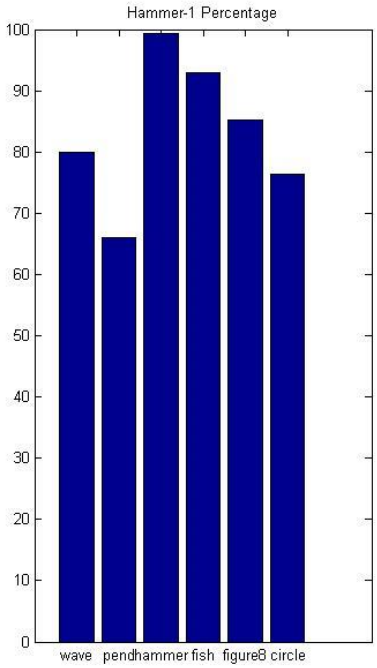

Fish-1 Order
6
5
4
3
2
1
0
wave
pendhammer
fish
figure8
circle

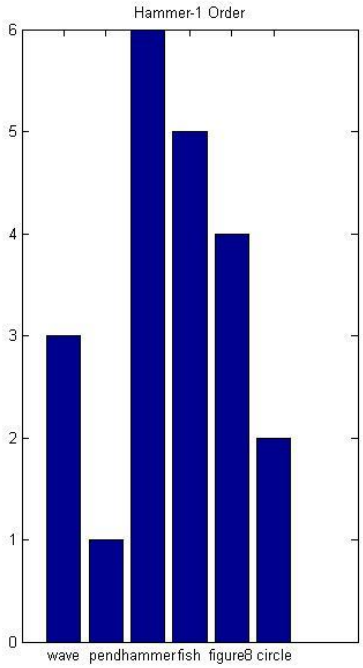

x 10^4
Fish-2 Probability
0
-0.5
-1
-1.5
-2
-2.5
wave
pendhammer
fish
figure8
circle

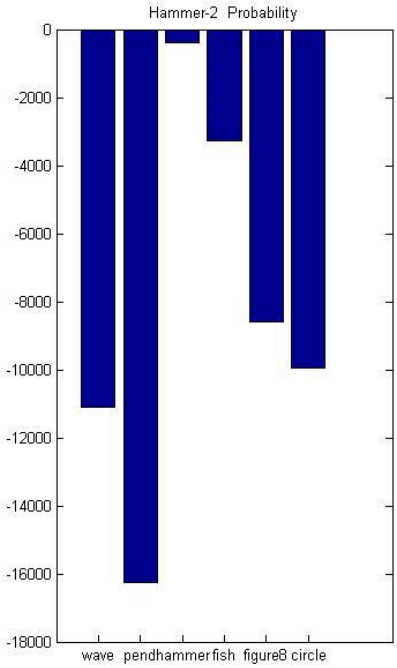

Fish-2 Percentage
100
90
80
70
60
50
40
30
20
10
0
wave
pendhammer
fish
figure8
circle

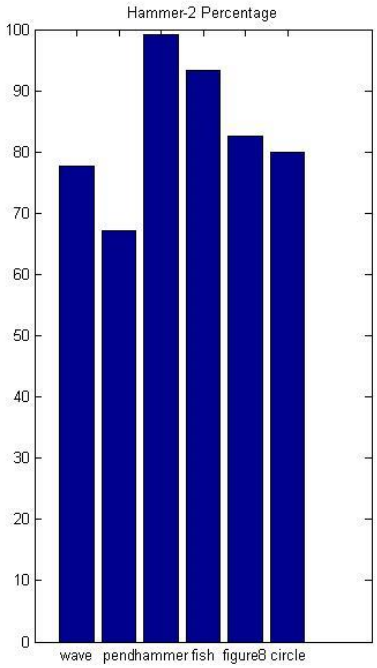

Fish-2 Order
6
5
4
3
2
1
0
wave
pendhammer
fish
figure8
circle

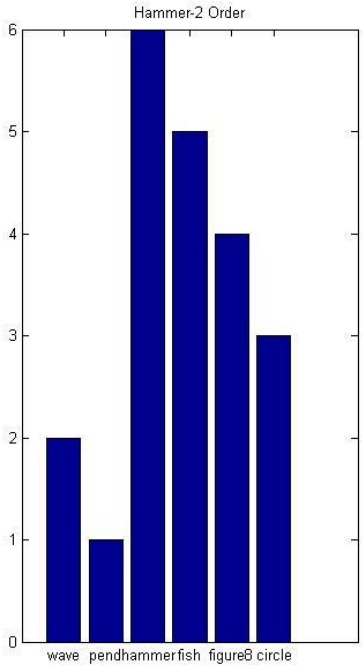

Hammer-1 Probability
0
-2000
-4000
-6000
-8000
-10000
-12000
-14000
-16000
wave
pendhammer
fish
figure8
circle

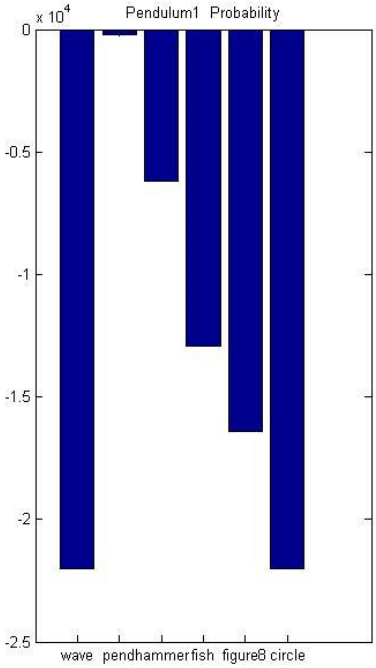

Hammer-1 Percentage
100
90
80
70
60
50
40
30
20
10
0
wave
pendhammer
fish
figure8
circle

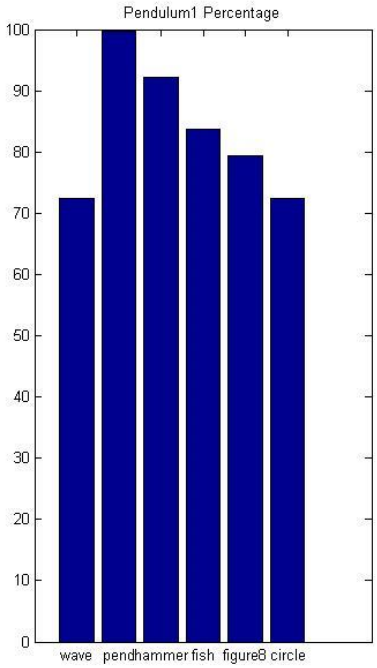

Hammer-1 Order
6
5
4
3
2
1
0
wave
pendhammer
fish
figure8
circle

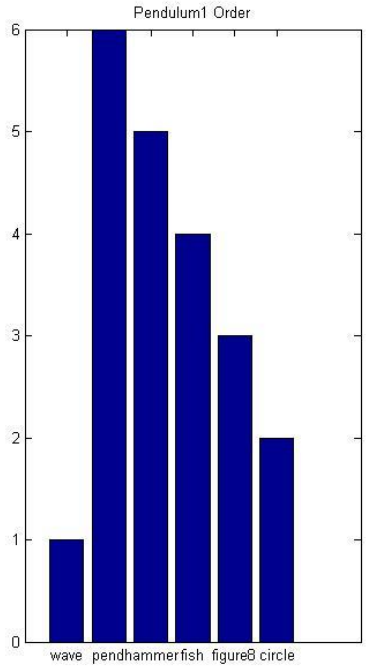

Hammer-2 Probability
0
-2000
-4000
-6000
-8000
-10000
-12000
-14000
-16000
-18000
wave
pendhammer
fish
figure8
circle

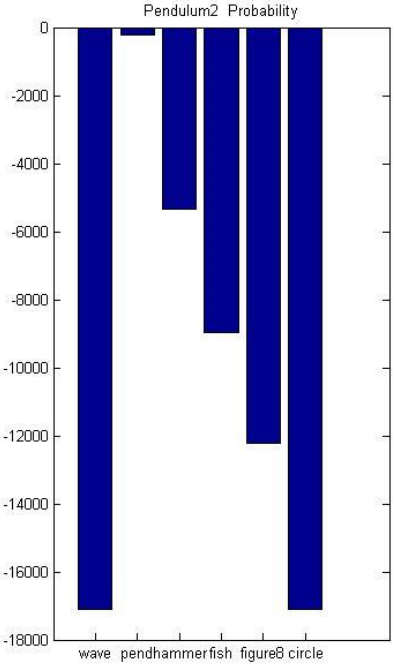

Hammer-2 Percentage
100
90
80
70
60
50
40
30
20
10
0
wave
pendhammer
fish
figure8
circle

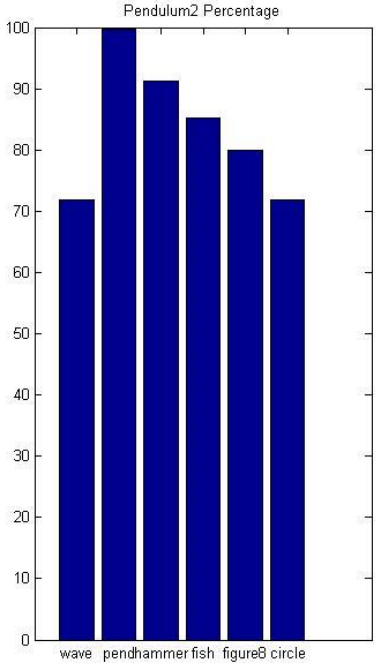

Hammer-2 Order
6
5
4
3
2
1
0
wave
pendhammer
fish
figure8
circle

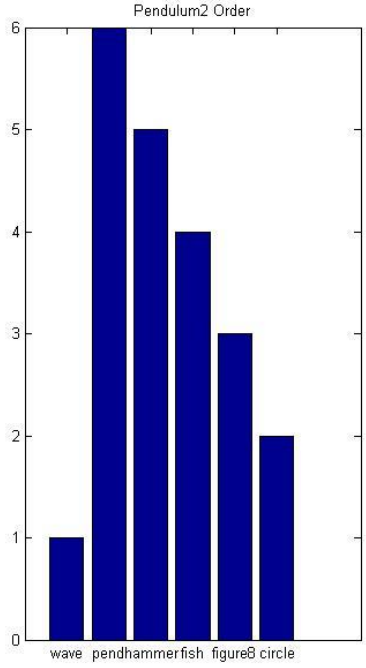

Pendulum1 Probability
x 10^4
0
-0.5
-1
-1.5
-2
-2.5
wave
pendhammer
fish
figure8
circle

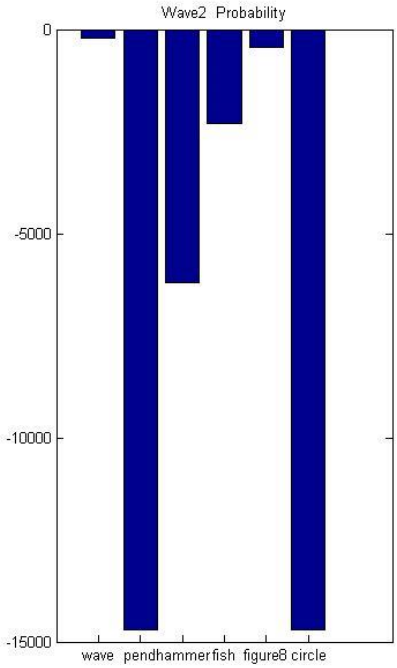

Pendulum1 Percentage
100
90
80
70
60
50
40
30
20
10
0
wave
pendhammer
fish
figure8
circle

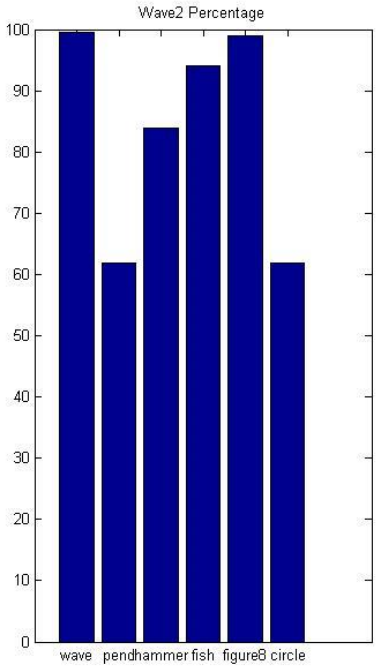

Pendulum1 Order
6
5
4
3
2
1
0
wave
pendhammer
fish
figure8
circle

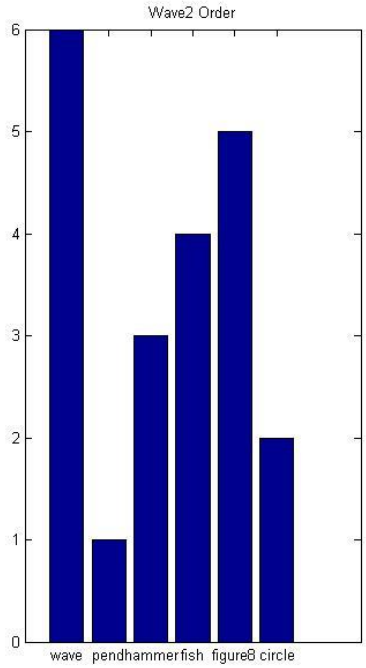

Pendulum2 Probability
0
-2000
-4000
-6000
-8000
-10000
-12000
-14000
-16000
-18000
wave
pendhammer
fish
figure8
circle

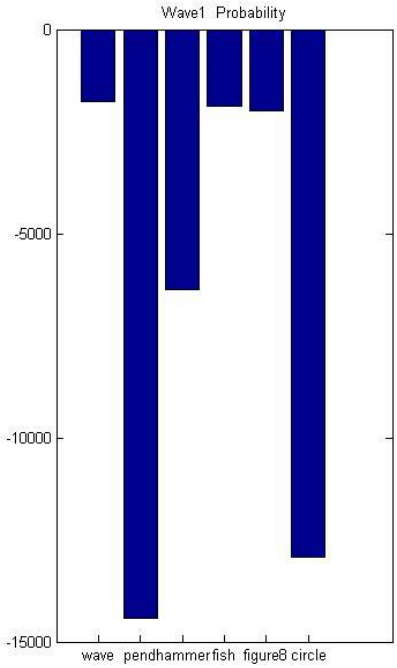

Pendulum2 Percentage
100
90
80
70
60
50
40
30
20
10
0
wave
pendhammer
fish
figure8
circle

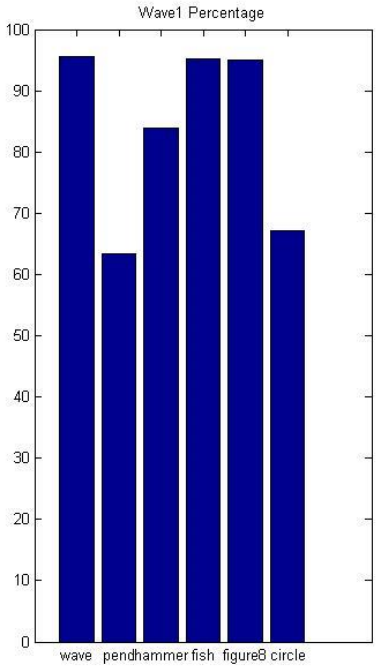

Pendulum2 Order
6
5
4
3
2
1
0
wave
pendhammer
fish
figure8
circle

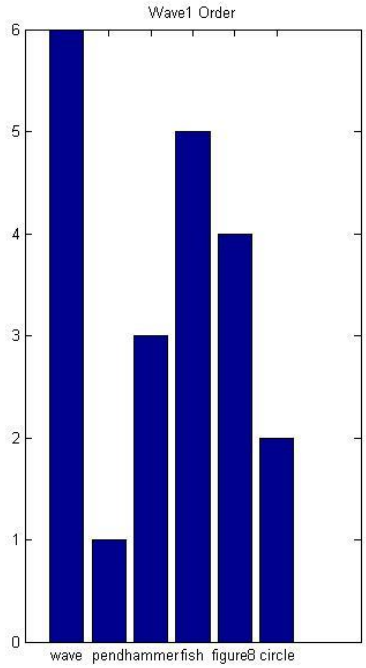

Wave2 Probability
0
-5000
-10000
-15000
wave
pendhammer
fish
figure8
circle


Wave2 Percentage
100
90
80
70
60
50
40
30
20
10
0
wave
pendhammer
fish
figure8
circle


Wave2 Order
6
5
4
3
2
1
0
wave
pendhammer
fish
figure8
circle


Wave1 Probability
0
-5000
-10000
-15000
wave
pendhammer
fish
figure8
circle


Wave1 Percentage
100
90
80
70
60
50
40
30
20
10
0
wave
pendhammer
fish
figure8
circle


Wave1 Order
6
5
4
3
2
1
0
wave
pendhammer
fish
figure8
circle