Assignment 2

Beike Lu
03651597

Exercise 1

GMM parameters:

Mean:

-0.0193569975974723	-0.0166476340171238
-0.0431927060954337	0.0445861695119822
-0.0147174949022227	-0.0796388103975139
0.0261856969632315	0.0617286574477437

Covariance Matrices:

```
val(:,:,1) =

   1.0e-03 *

    0.7437   -0.5916
   -0.5916    0.6102

val(:,:,2) =

   1.0e-03 *

    0.1748    0.2615
    0.2615    0.3975
```

```
val(:,:,3) =

   1.0e-03 *

    0.3944    0.2167
    0.2167    0.1276

val(:,:,4) =

    0.0011   -0.0004
   -0.0004    0.0002
```

Prior:

0.297167121370254
0.239974104467071
0.201139958128008
0.261718816034666

Exercise 2

The log likelihood value of each sequence can be shown blow:

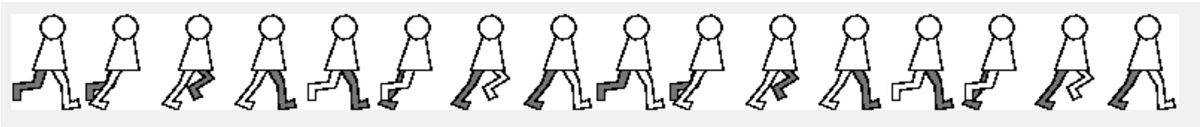
```
loglikelh =

 -511.4069 -570.6697 -387.9167 -427.3069 -437.5989 -426.1784 -473.3031 -400.2880 -377.1776 -401.0614
```

according to the question, if likelihood value <-120, the classification would be gesture 2, so all of the sequence belong to **gesture 2.**

Exercise 3

Policy iteration:

Start from state 8:



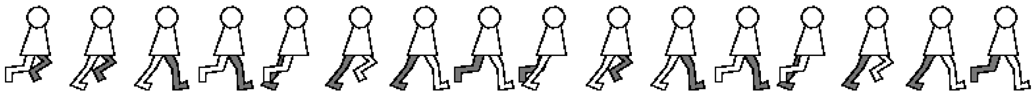
1. Reward Matrix:

0	0	0	0
0	1	-1	-1
1	-1	-1	0
-1	0	1	0
-1	-1	0	1
0	0	0	0
0	0	0	0
-1	1	-1	0
-1	0	1	-1
0	0	0	0
0	0	0	0
-1	0	0	0
1	0	-1	0
-1	0	-1	1
0	0	-1	0
0	0	0	0

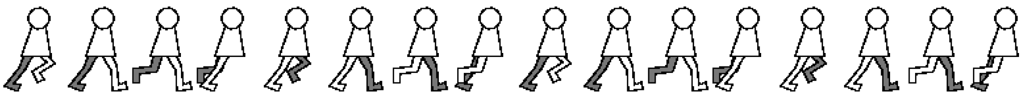
2. The value of gamma is **0.99.**
   If decrease gamma, the iterations needed to achieve convergence are fewer.
   If increases gamma, the number of iterations required to obtain acceptable accuracy rises rapidly.

3. The policy iteration is considered as convergence if the policy hasn´t change for 3 iterations in a row. Therefore, it takes usually 5-7 iterations to reach convergence. If the iteration is considered as soon as policy doesn´t change, the number of iterations would be 3-5 accordingly.

4. Start from 10:

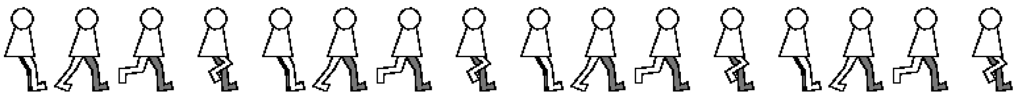


Start from 3:



Q-learning:

Start from state 16:



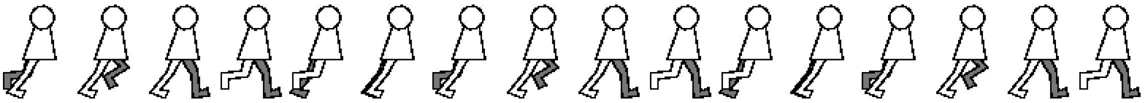
1. epsilon: 0.1
   alpha: 0.1
2. A pure greedy policy leads to no improvements in robot performance, because this doesn´t allow for the exploration step. Since the Q-function was initialized with zeros, a pure greedy policy resulted in always actions 1 and new Q values of zeros. Therefore, the robot can not explore all the possibilities of movements, so it can not reach the optimal policy.

   If the epsilon-greedy is used, a random action is taken for exploration. Therefore, the Q-function would not be stuck in the endless circle of zeros.

   Epsilon determines the probability of taking a random step. Therefore, the higher the epsilon value, the more adventurous the robot. Low value of epsilon lead to slow convergence, high value of epsilon result no convergence at all.

3. 10000 steps

4. Start from state 5



Start from state 12: