

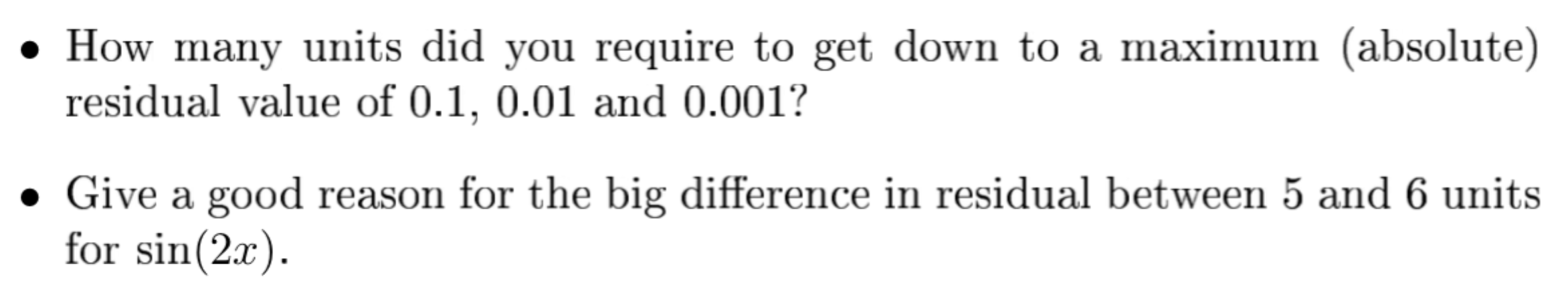
1)

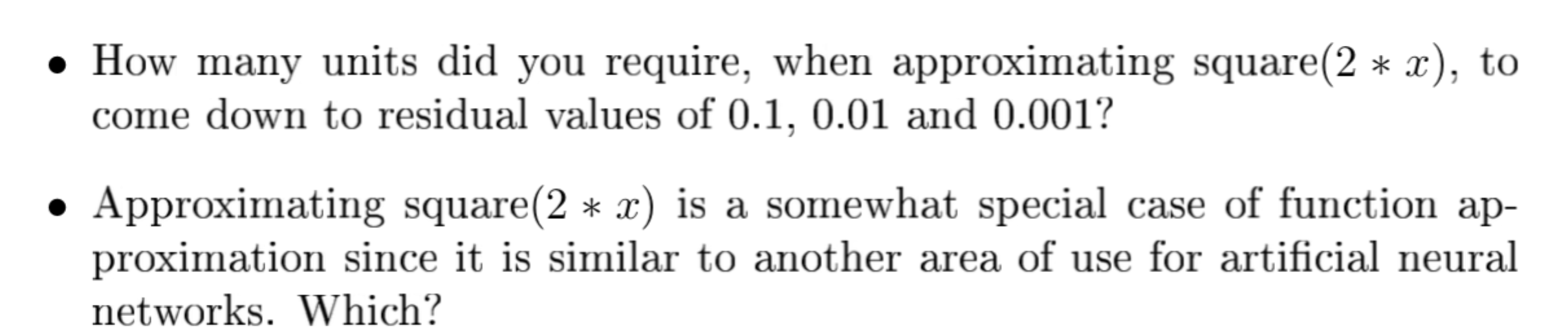
It needs to be larger than 1.

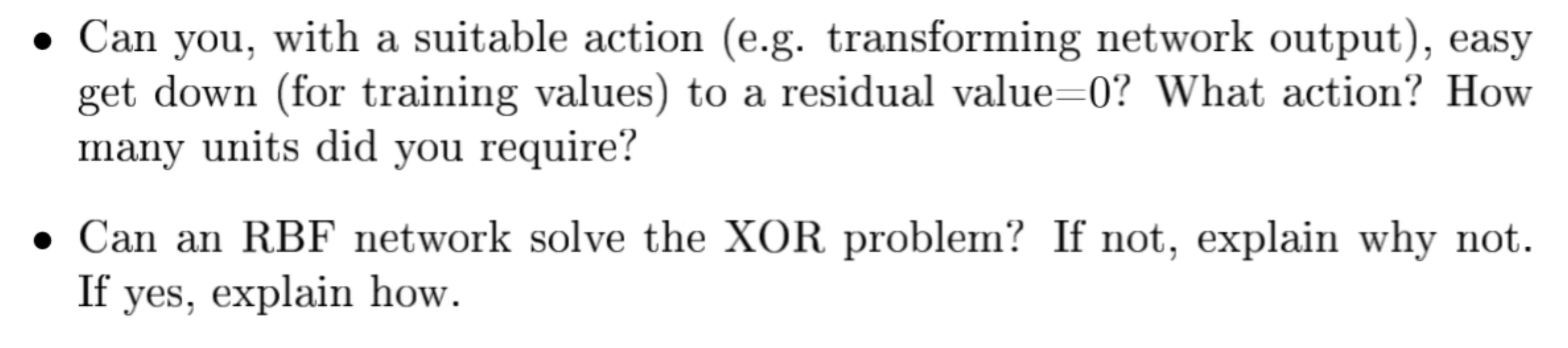
2) There will only one solution for equation (4). The training error will be zero.

3)

4) No. We need to test the model on test set but not on the training set.



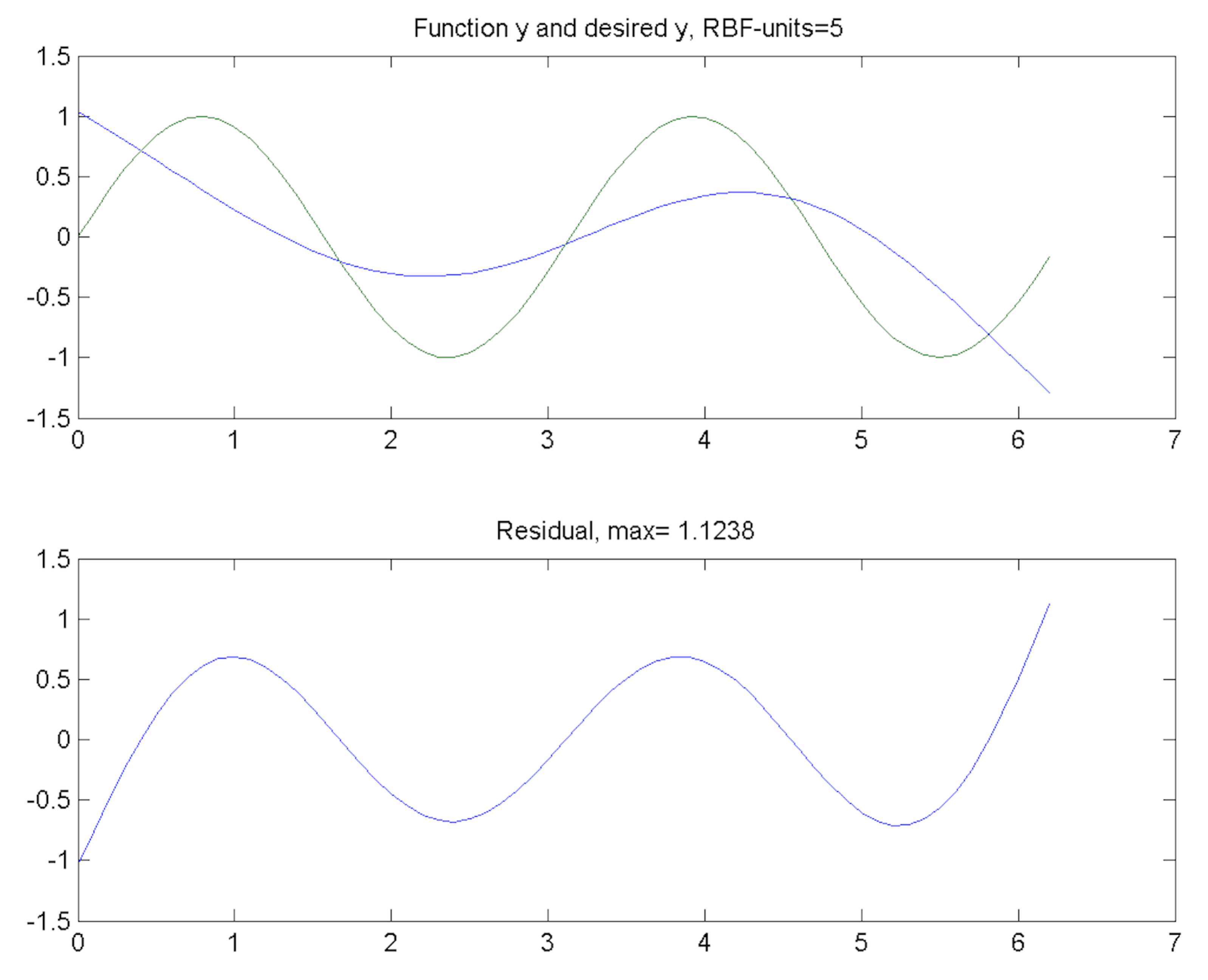


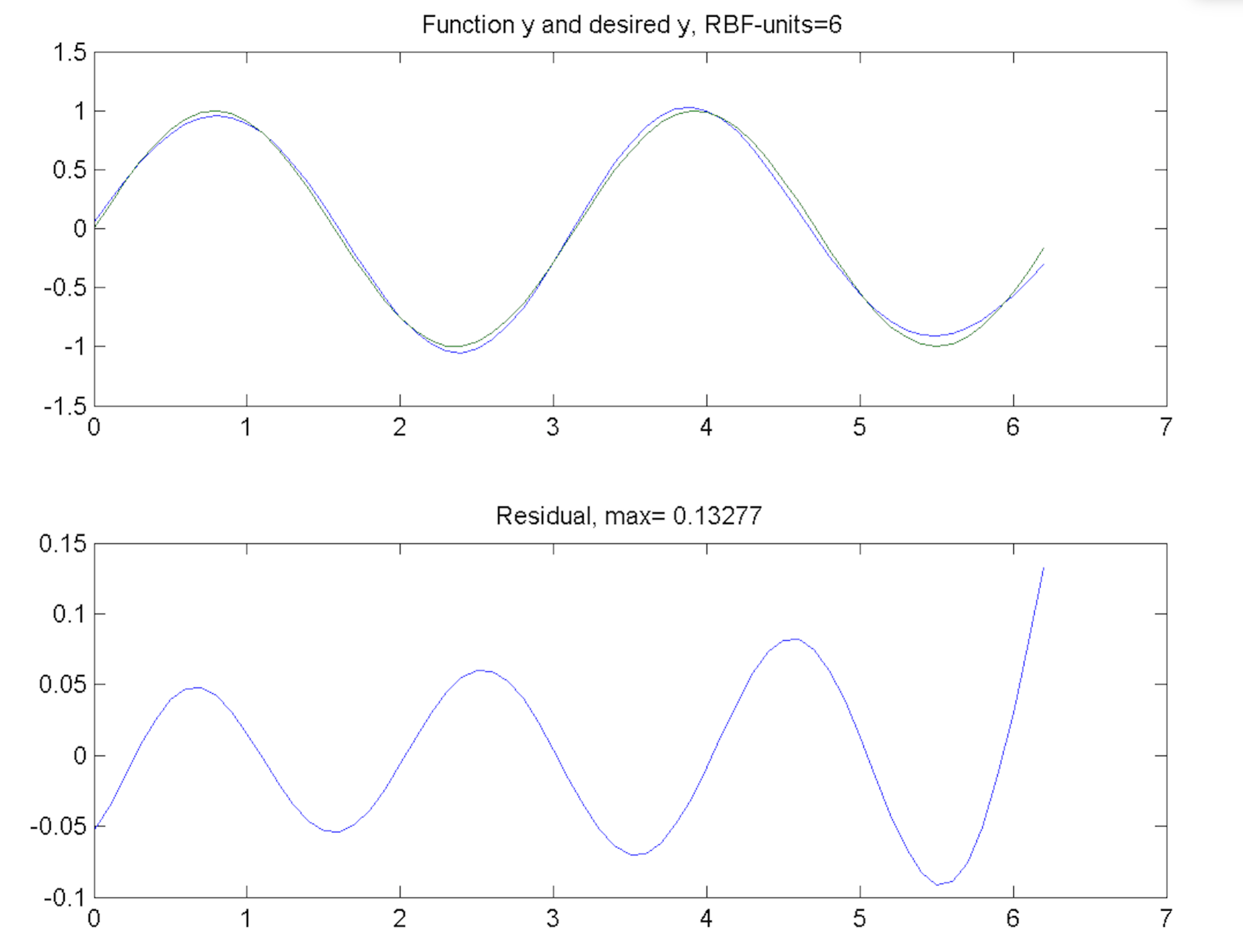


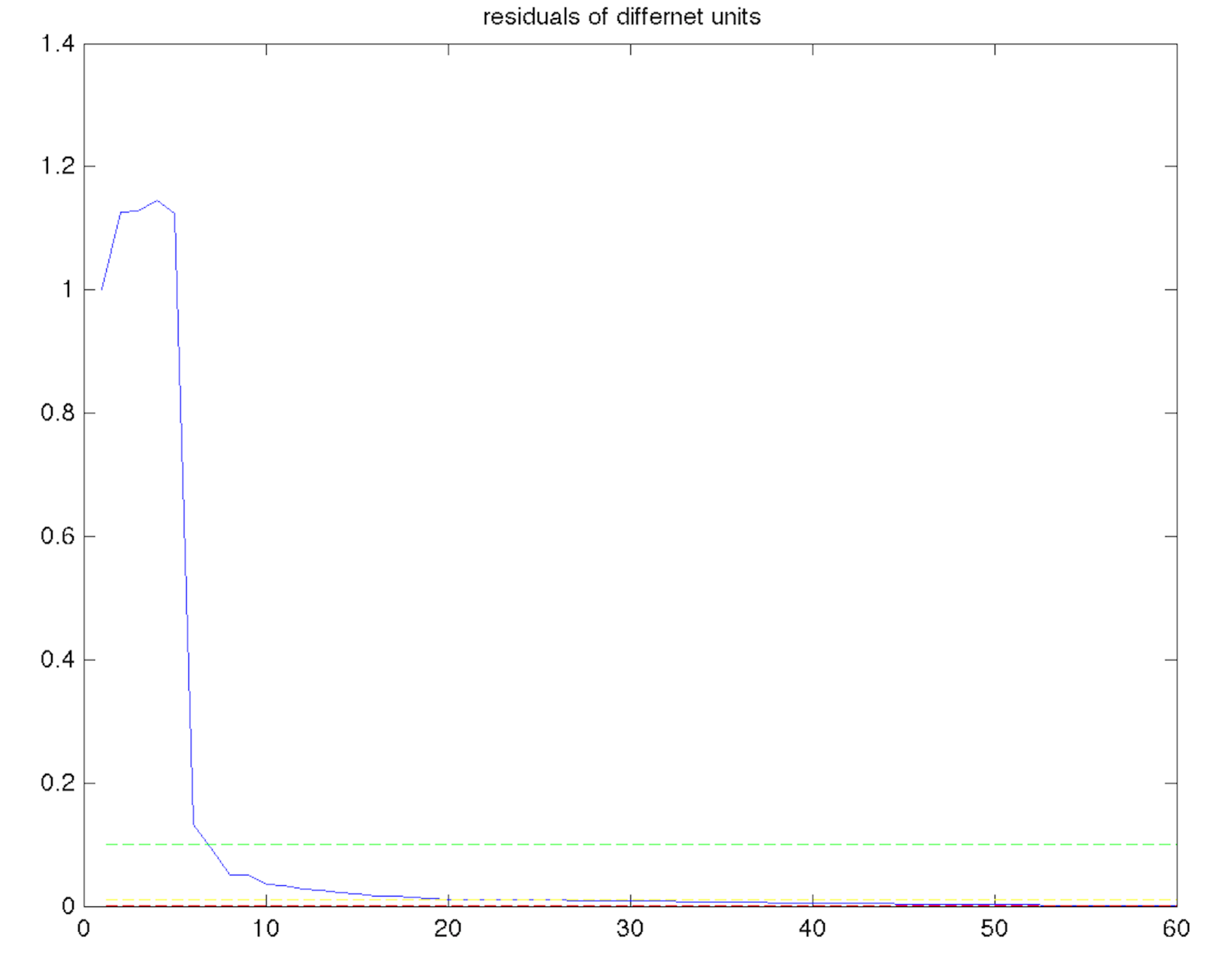
1) 3)

|  |  |
| --- | --- |
| Sin | Square |
| unit = 7 25 56  residual=0.0941 0.0099 0.0009 | unit = 44 59 61  residual=0.0974 0.0065 0.0004 |

2) At some points, more RBF units give better approximation.



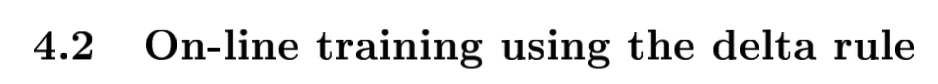


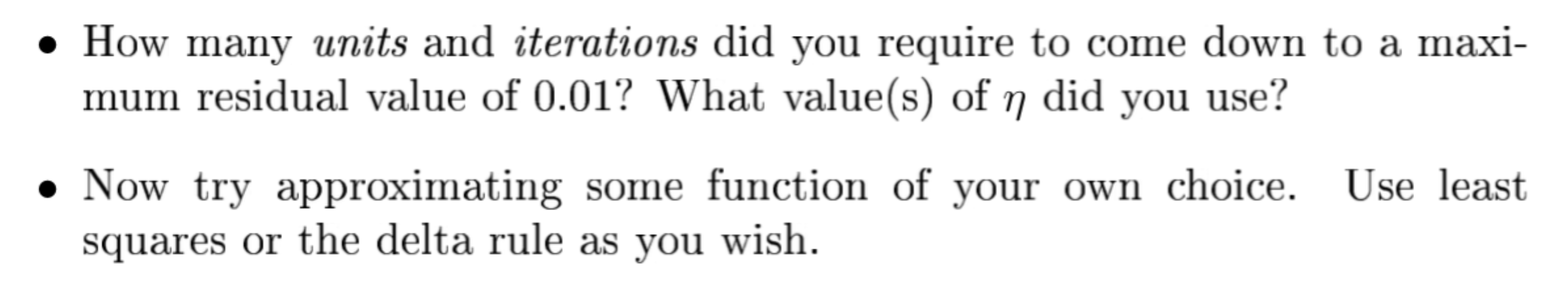


4) Using two layers of feed forward neural network with different hidden units to approximate the Gaussian functions (in lab1).

5) When the number of training samples , the residual value=0.

6) The RBF can solve the XOR problem only when , i.e. . needs to be larger than the patterns or classes of the dataset.



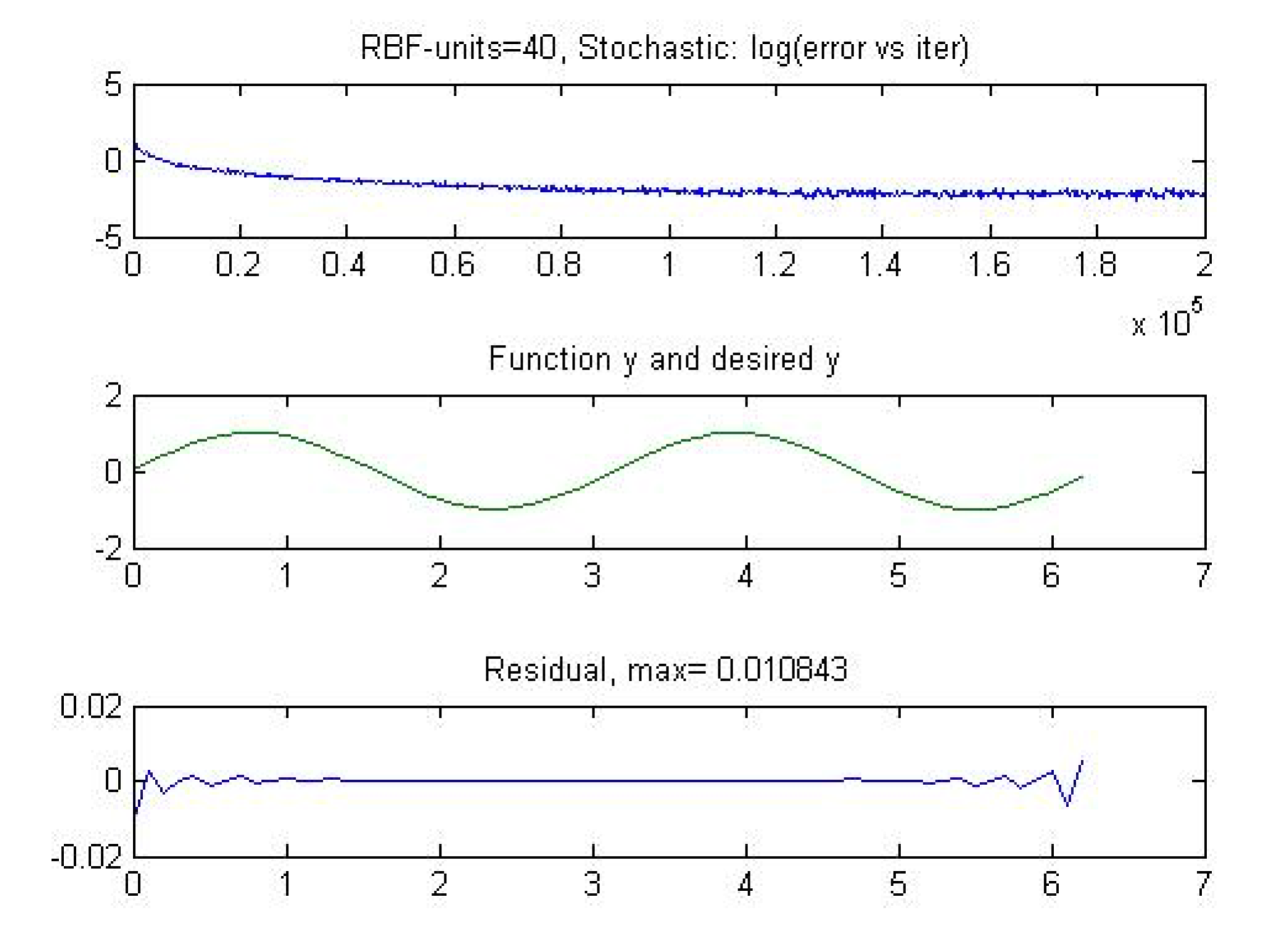


1. units=40; itersub =20; itermax =20000;eta=1;

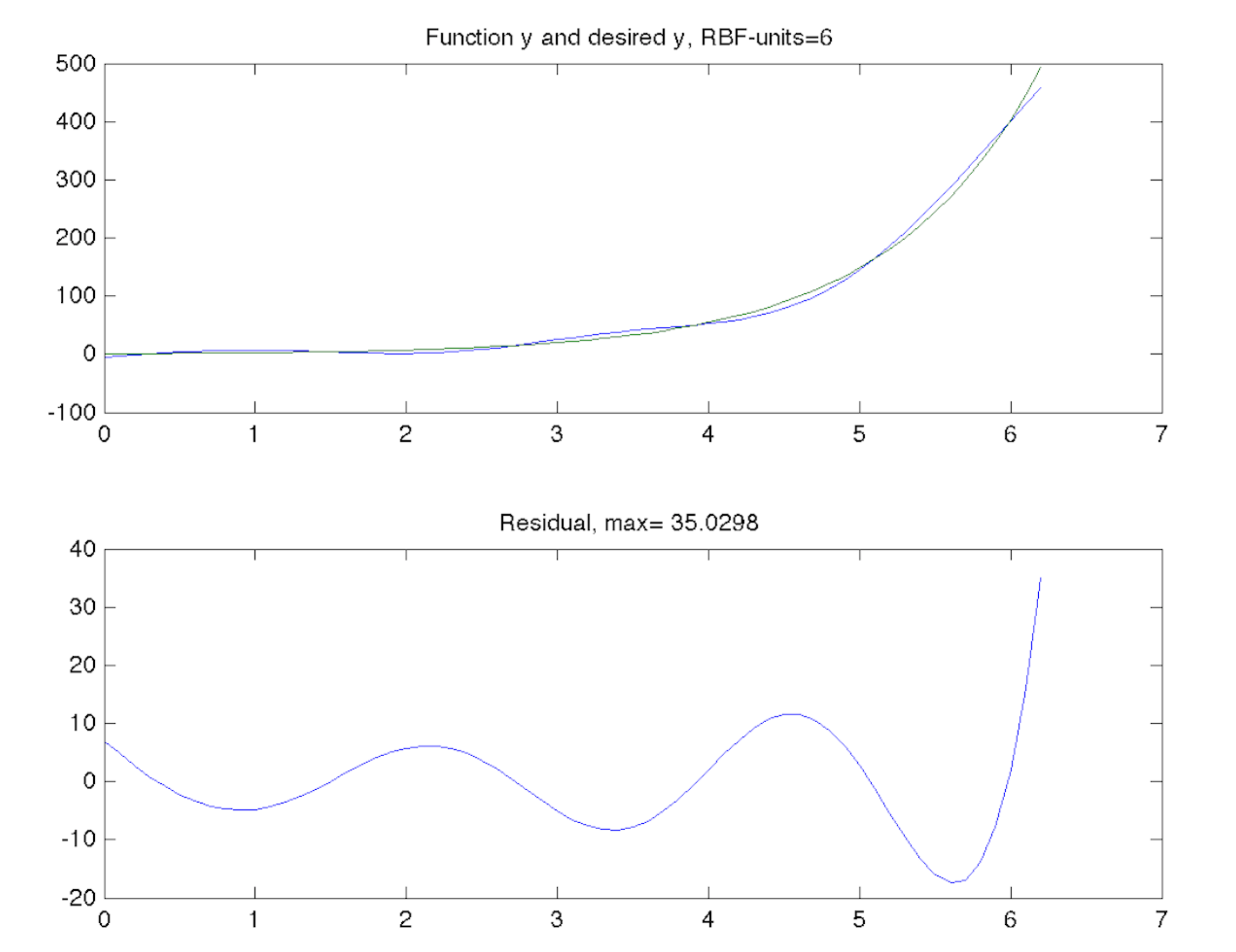
units larger: better but slower (more units to train);

more iterations: better but slower;

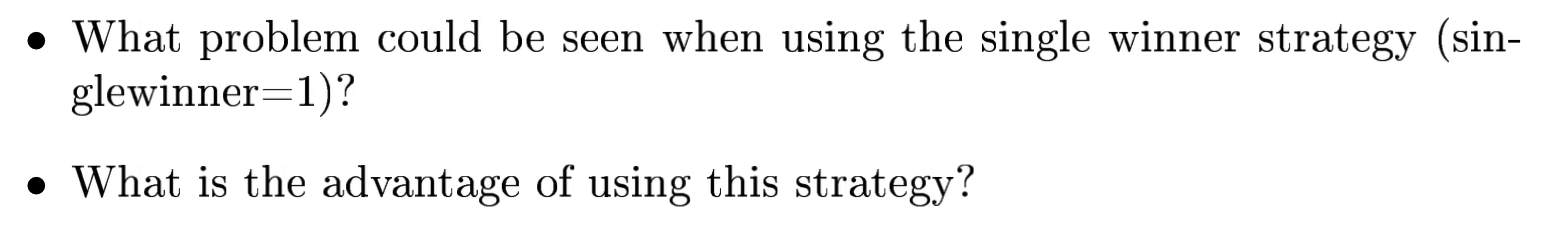
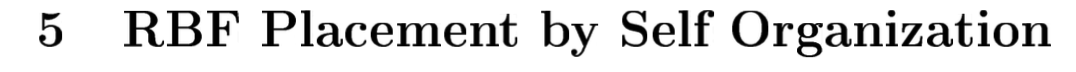
smaller eta: need more time to be optimal (converge slowly).

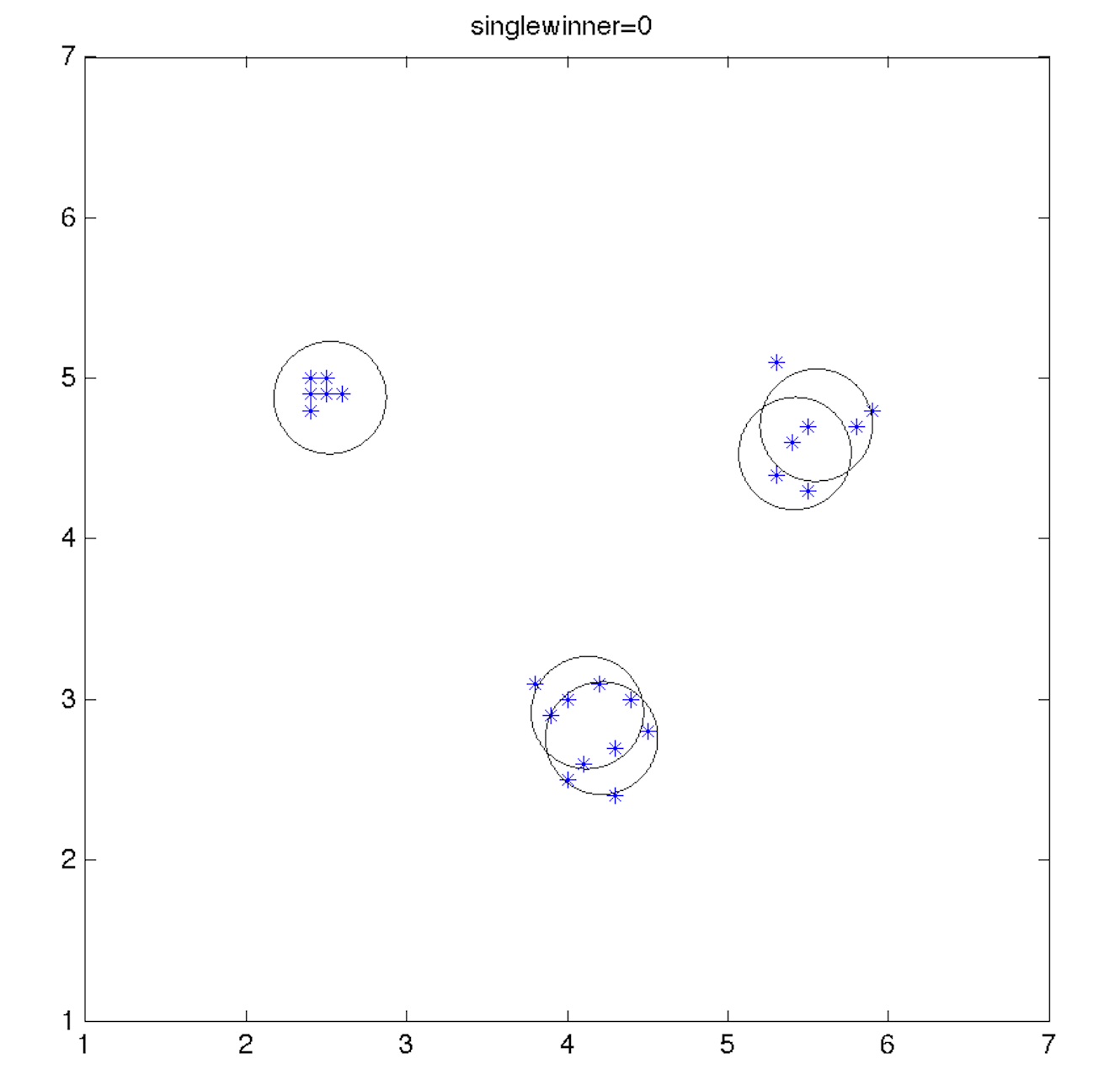
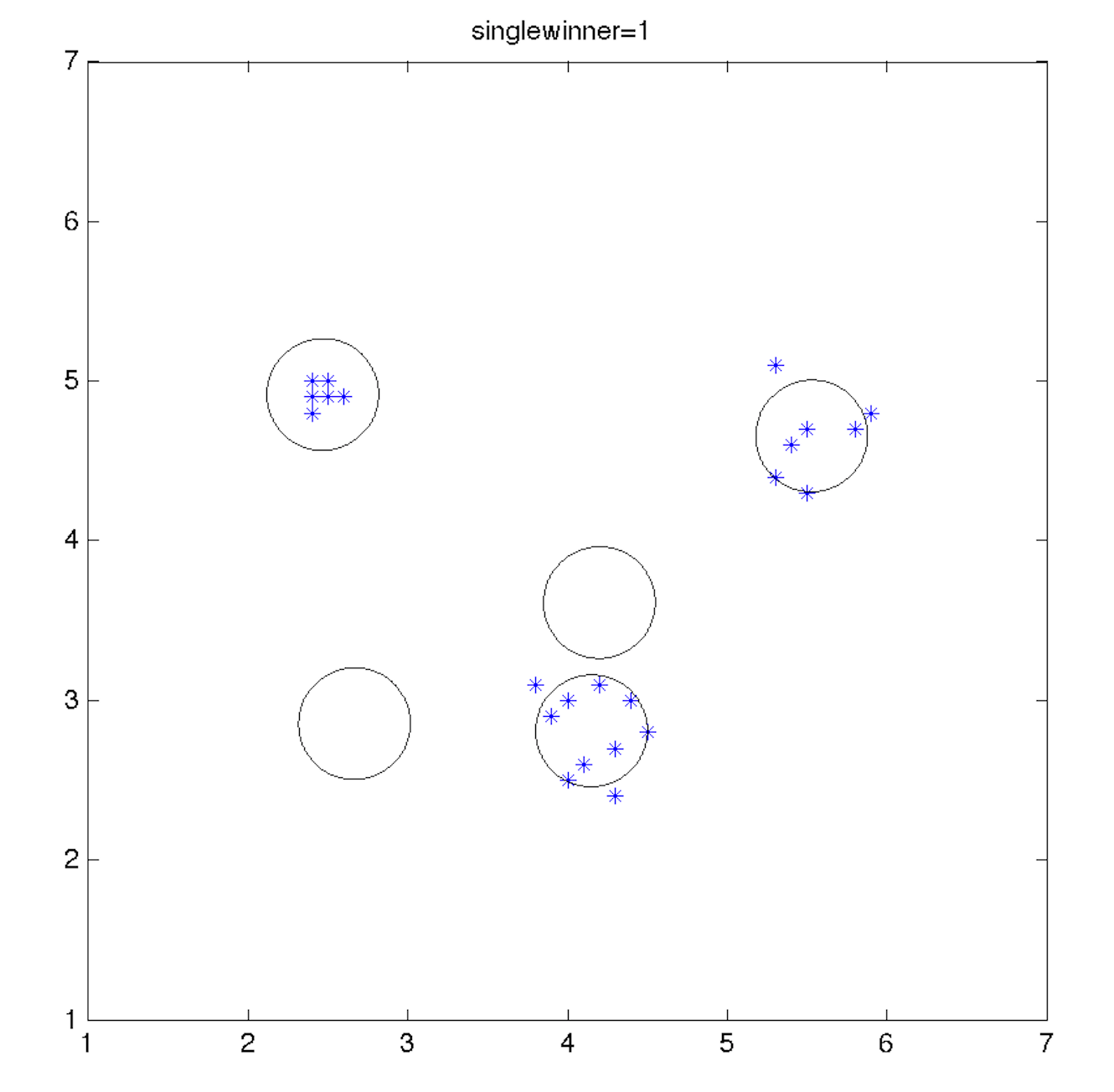


1. Use least squares to approximate exponential function.



|  |  |
| --- | --- |
| Sin | Square |
| unit = 7 25 56  residual=0.0941 0.0099 0.0009 | unit = 44 59 61  residual=0.0974 0.0065 0.0004 |
| exponential |  |
| unit =59 61 62  residual=0.0690 0.0043 0.0000 |  |

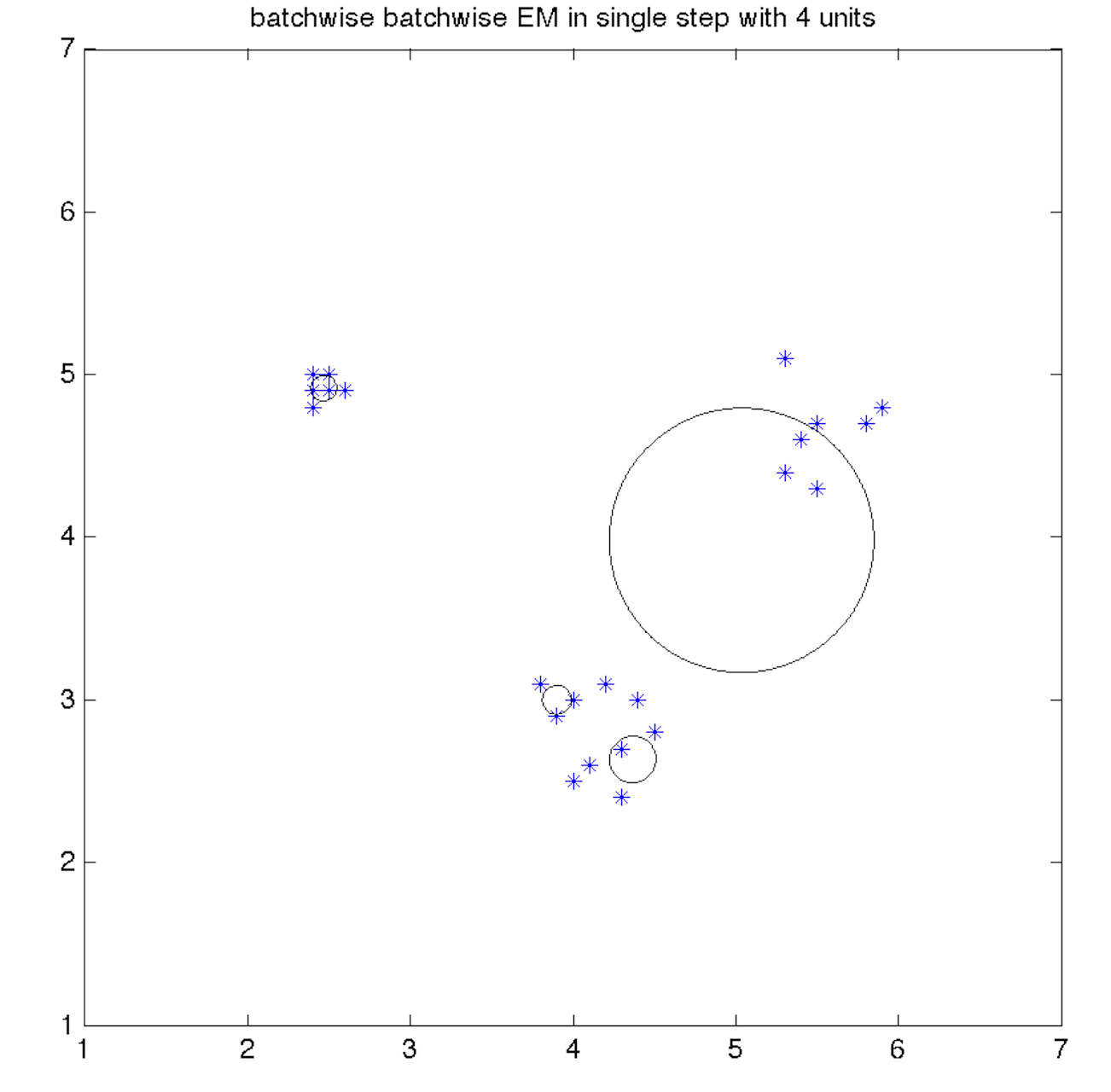
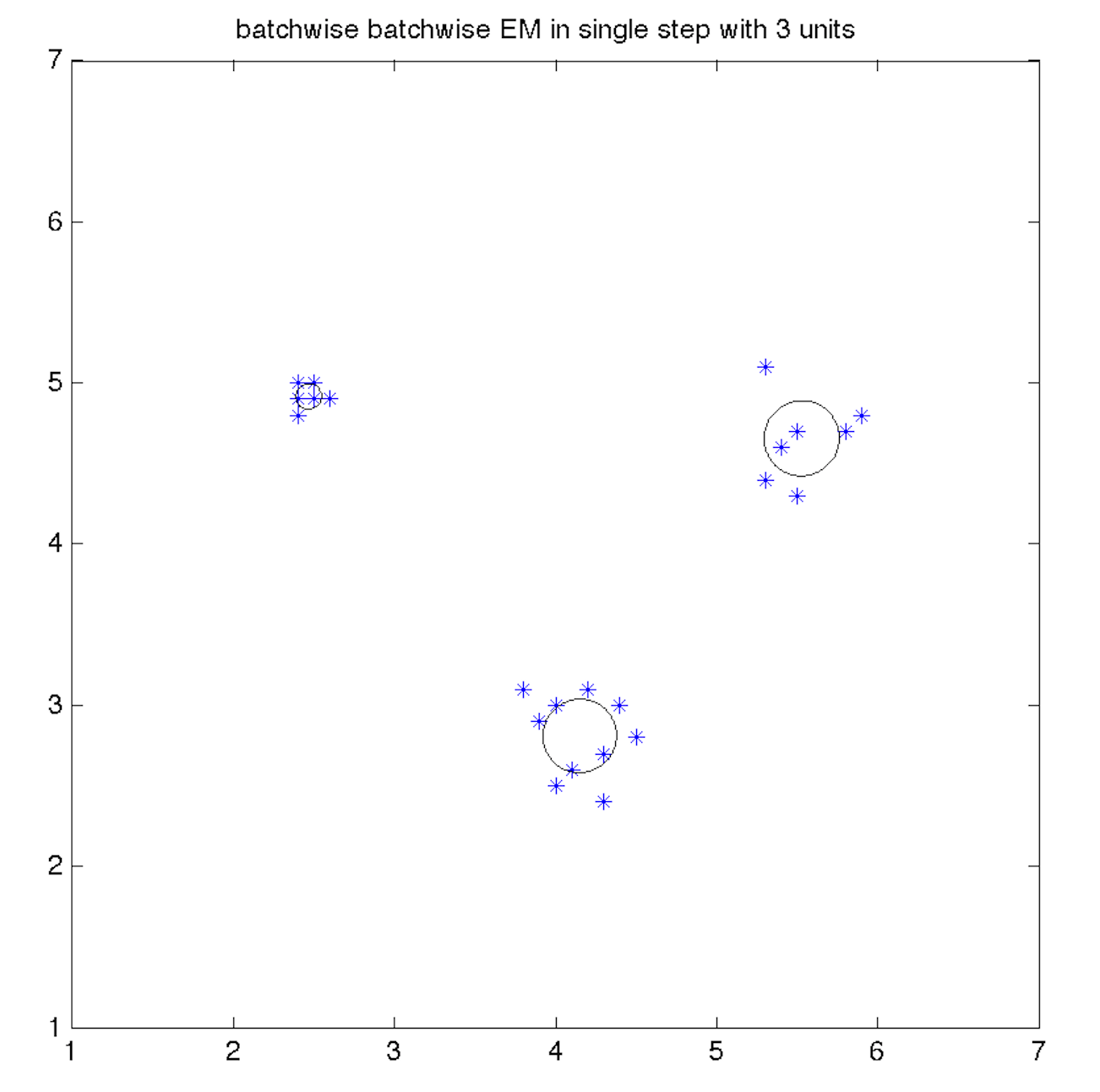


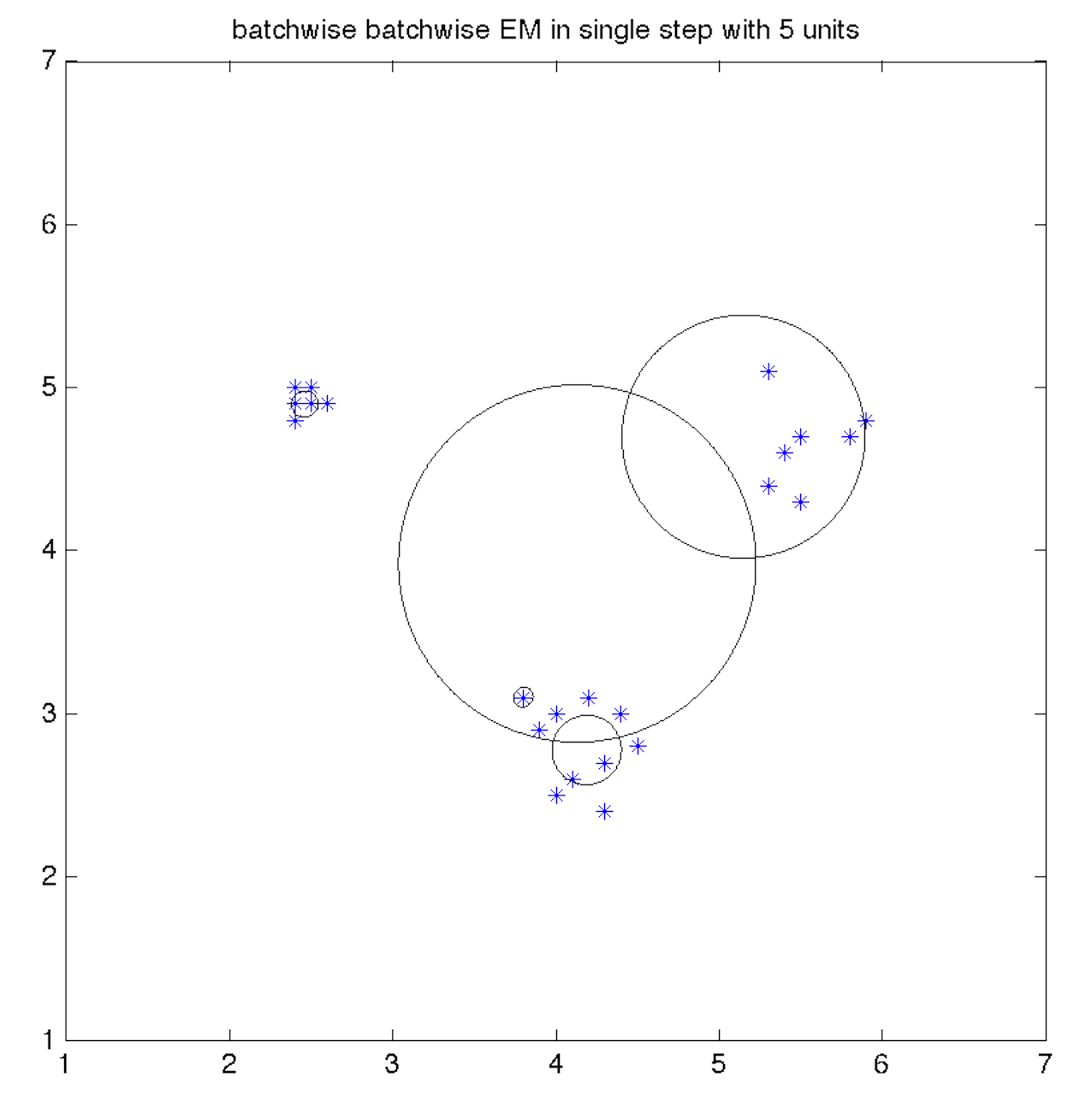
1. When using single winner strategy, only the winner units can get update.  
2. It converges faster and results in less useful RBF units (we can throw away those useless RBF units that do not represent any cluster.)

**EM algorithm**

1. **Batchwise EM (single winner strategy: singlewinner=1)**

Units = 3,4,5

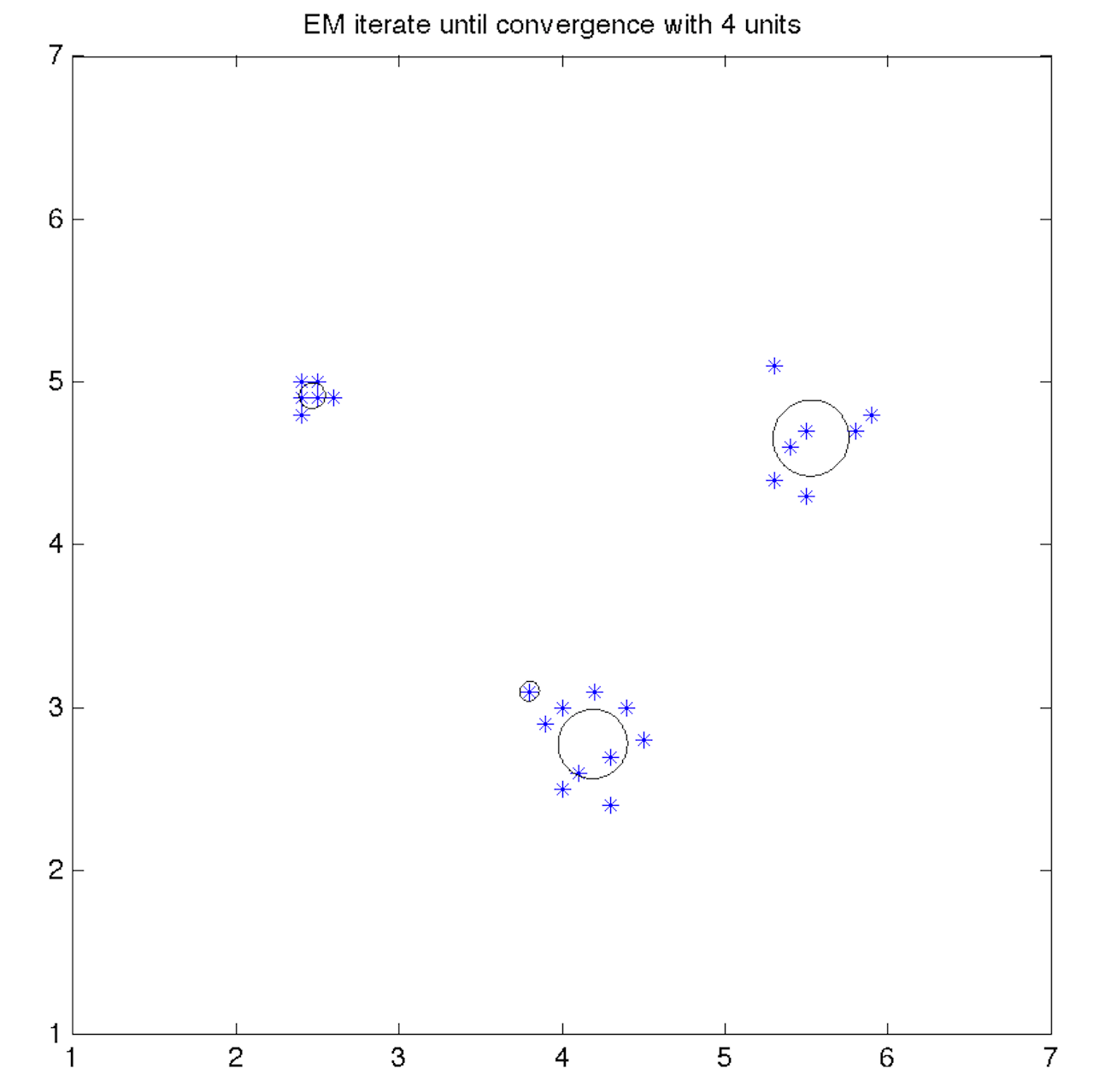
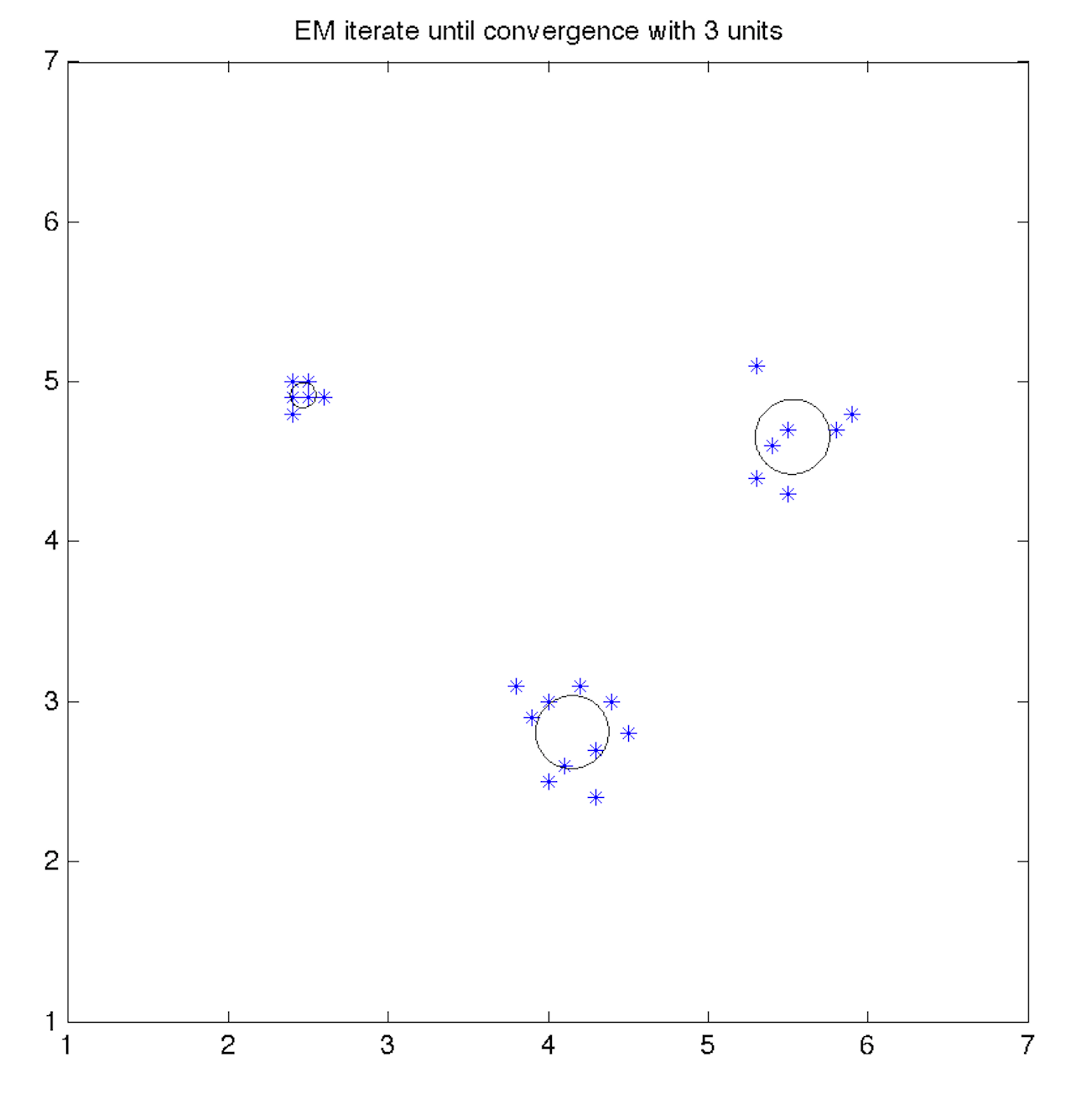


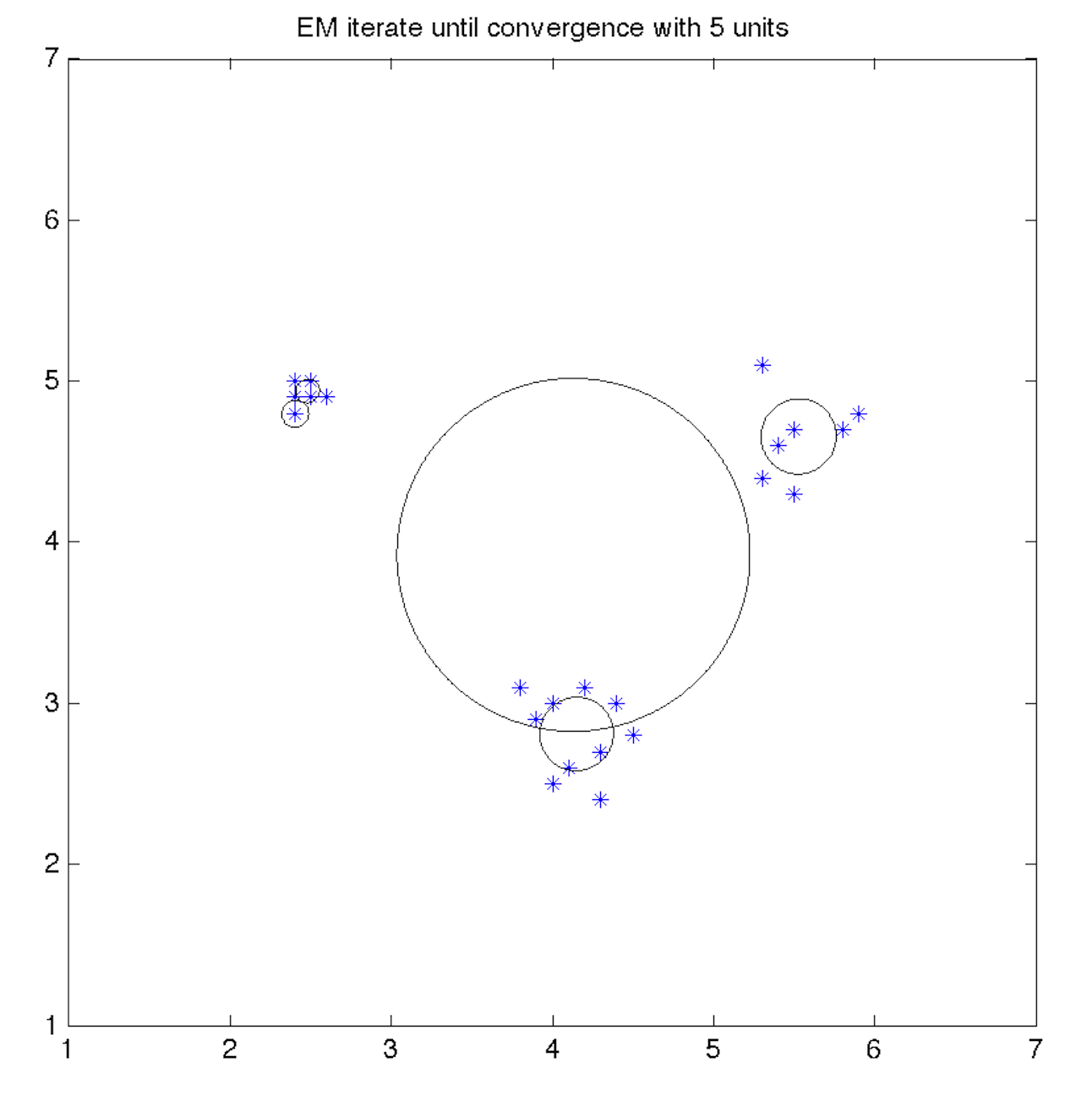


When using single winner strategy, only when units=3, the Batchwise EM can give good clustering result.

1. **Iterative EM (single winner strategy: singlewinner=1)**

Units = 3,4,5



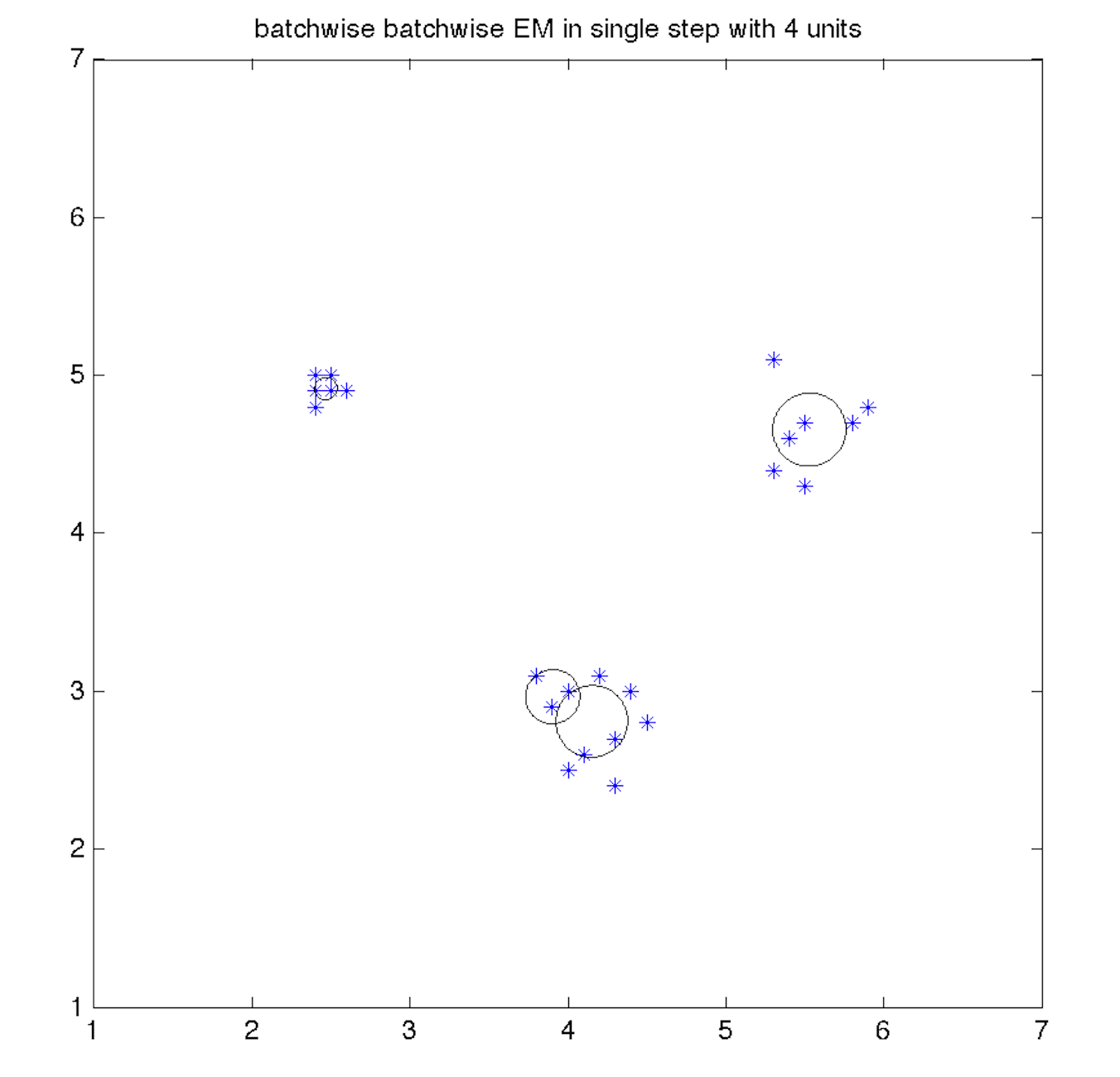
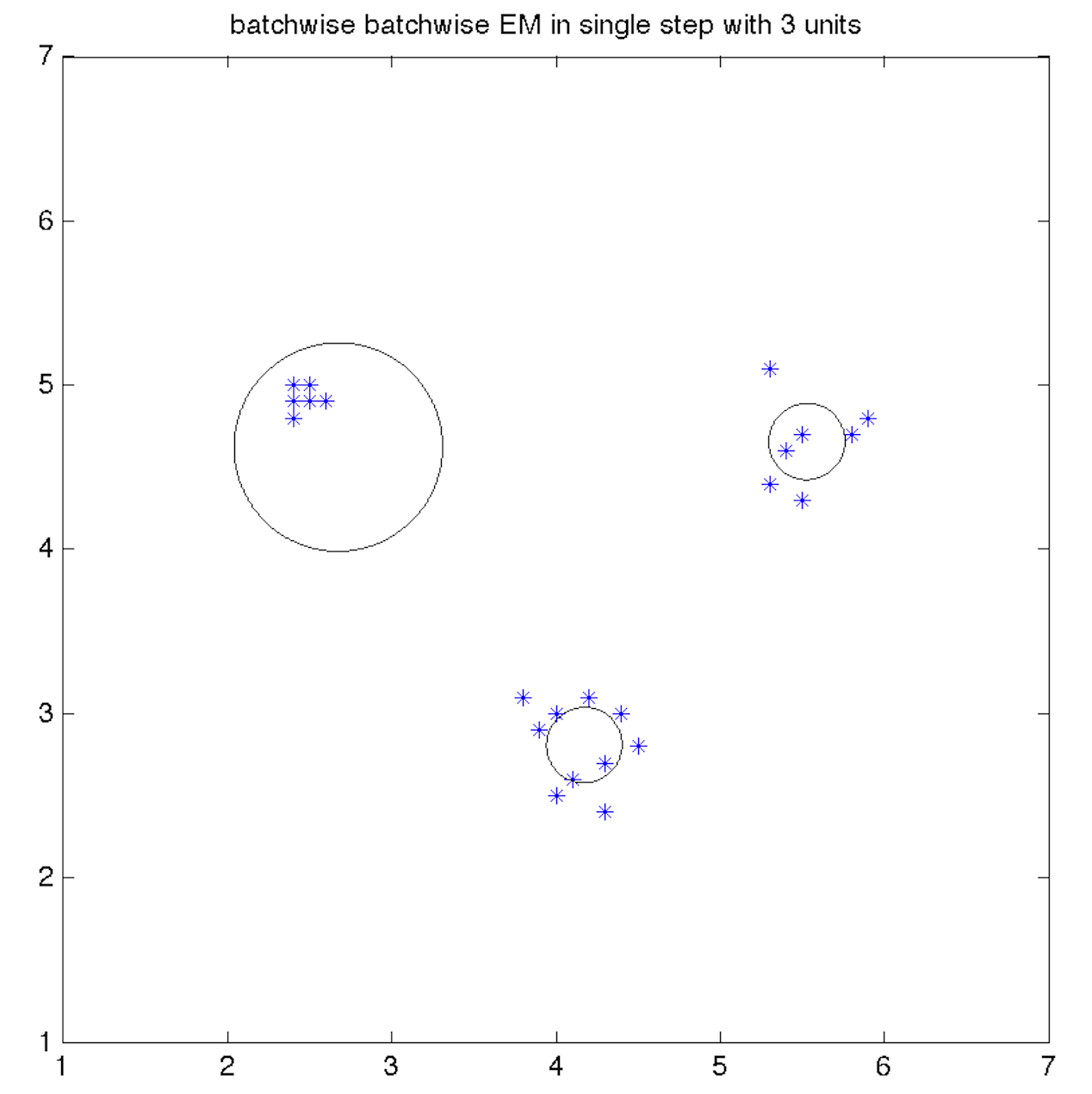


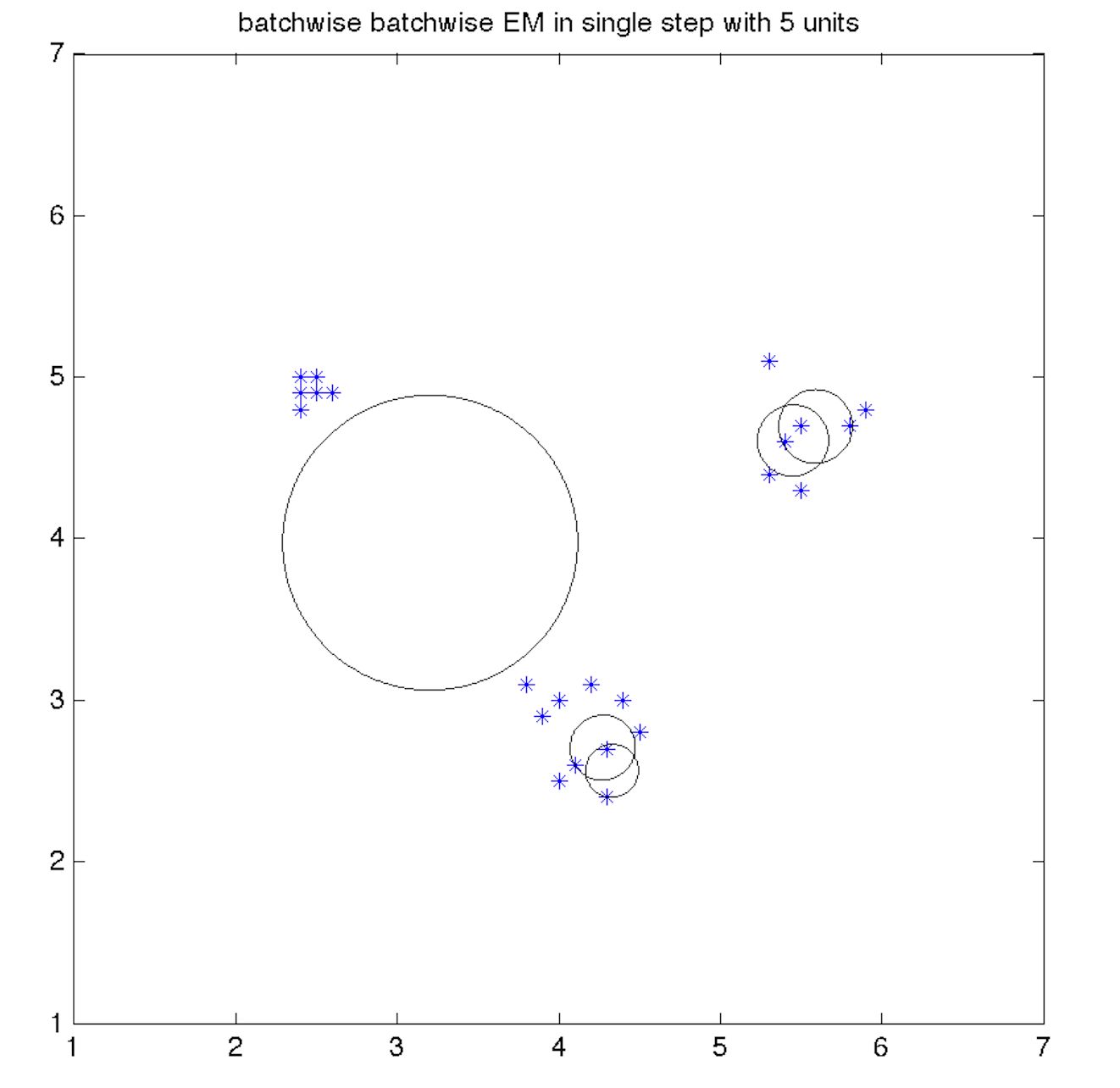
When using single winner strategy, when units=3 or 4, the iterative EM can give good clustering results.

Therefore, iterative EM is slightly better than Batchwise EM. It gives more stable convergence.

1. **Batchwise EM (do not use single winner strategy: singlewinner=0)**

Units = 3,4,5

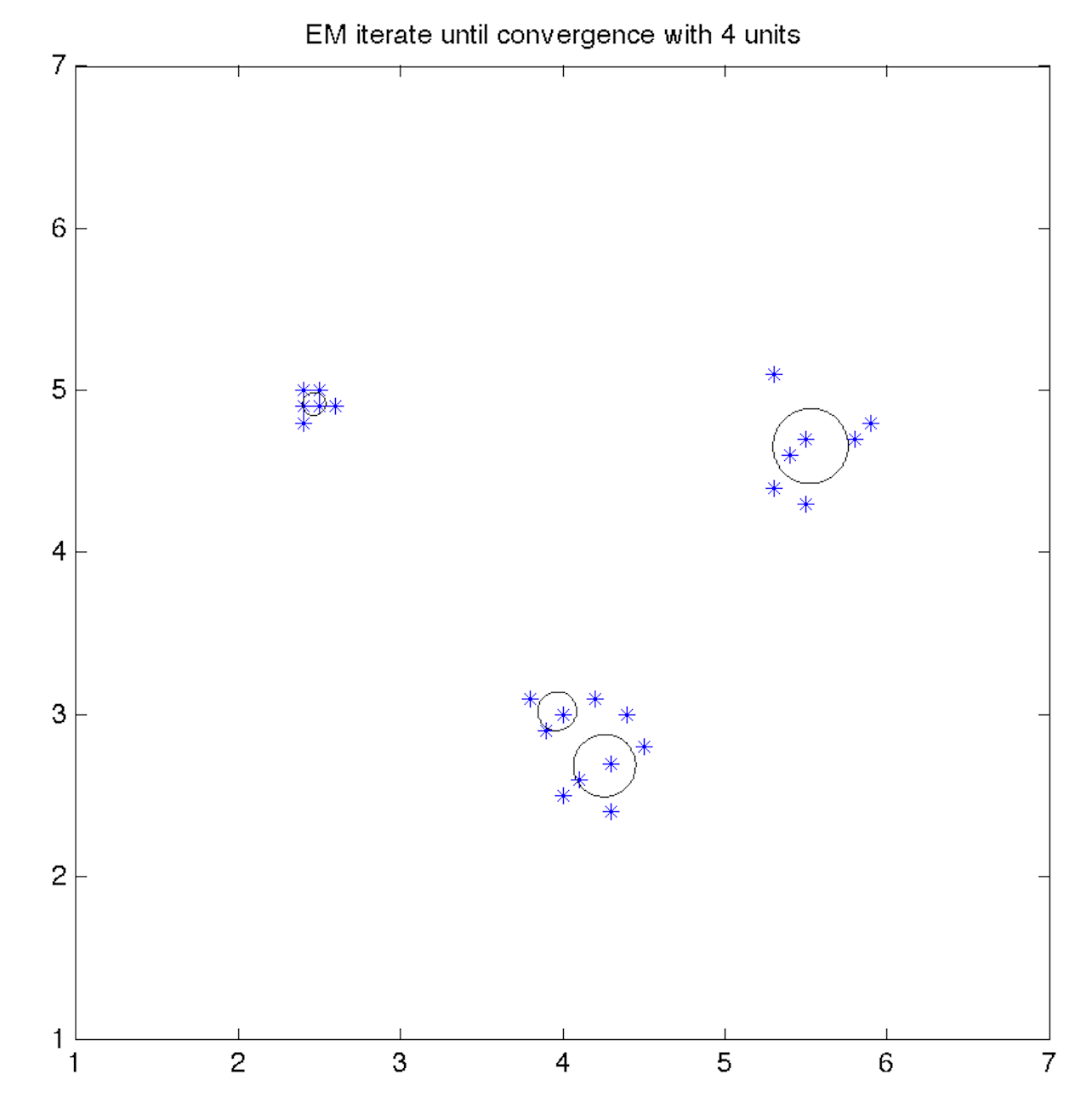
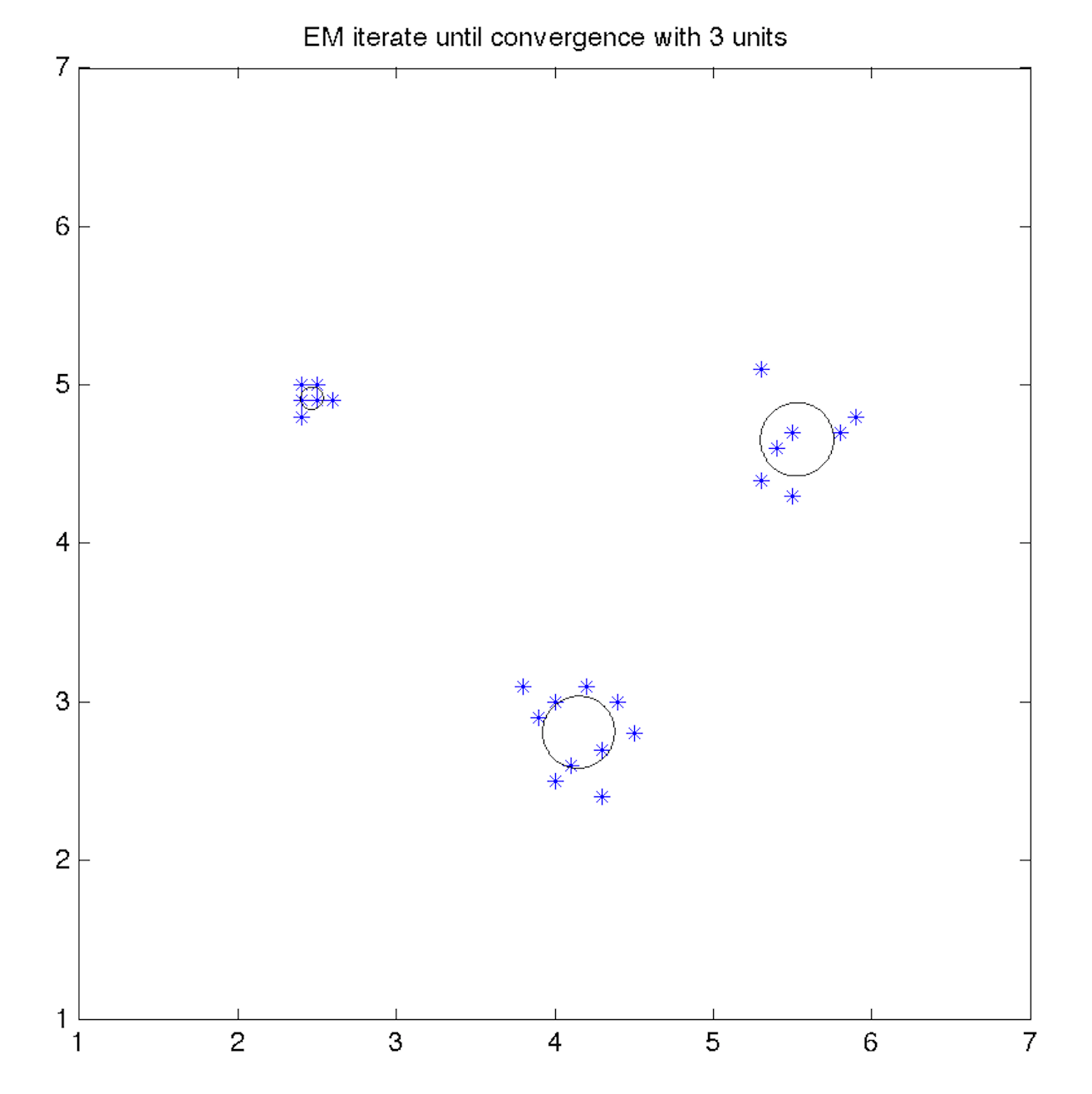


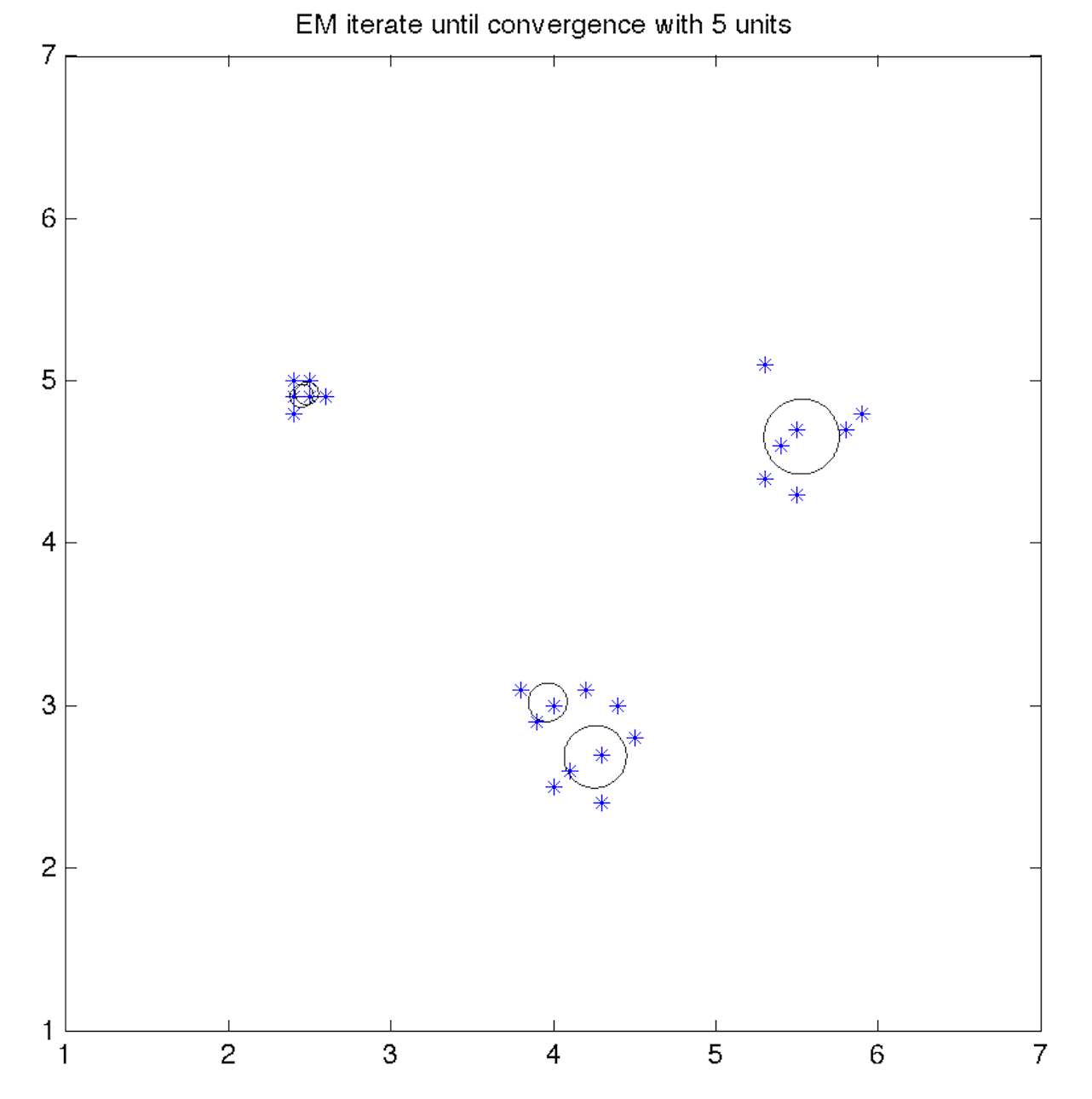


When not using single winner strategy, when units=3 or 4, the Batchwise EM can approximate the clusters. But it is bad on some certain that it is likely to estimate more than two clusters to approximate a cluster!

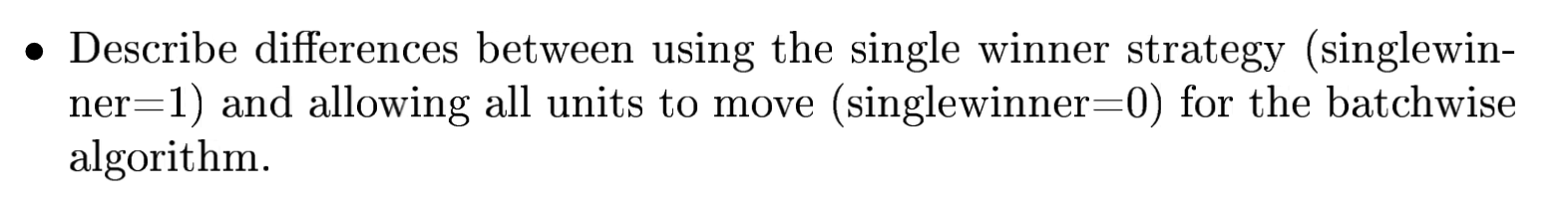
1. **Iterative EM(do not use single winner strategy: singlewinner=0)**

Units = 3,4,5



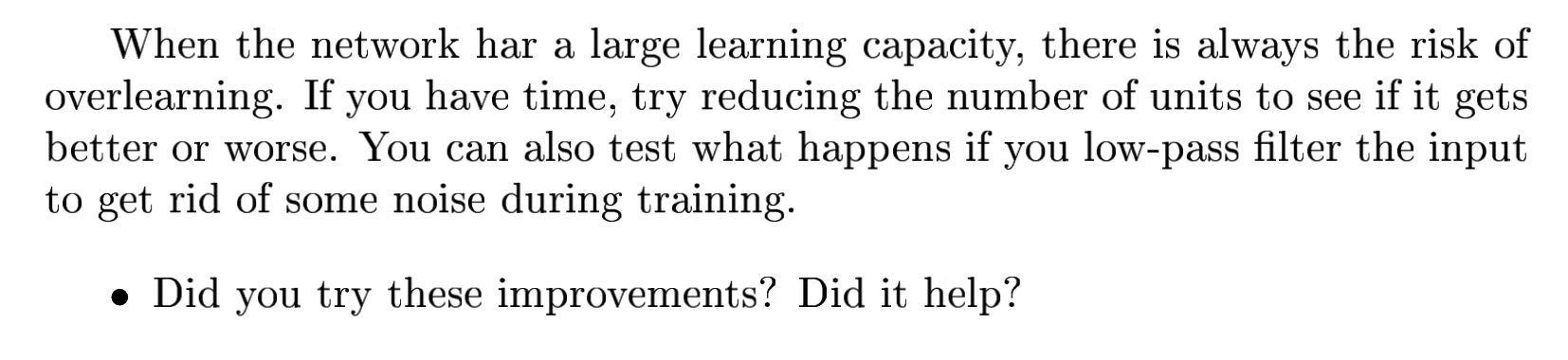


When not using single winner strategy, when units=3 or 4 or 5, the iterative EM can approximate the clusters. But it is bad on some certain that it is likely to estimate more than two clusters to approximate a cluster!



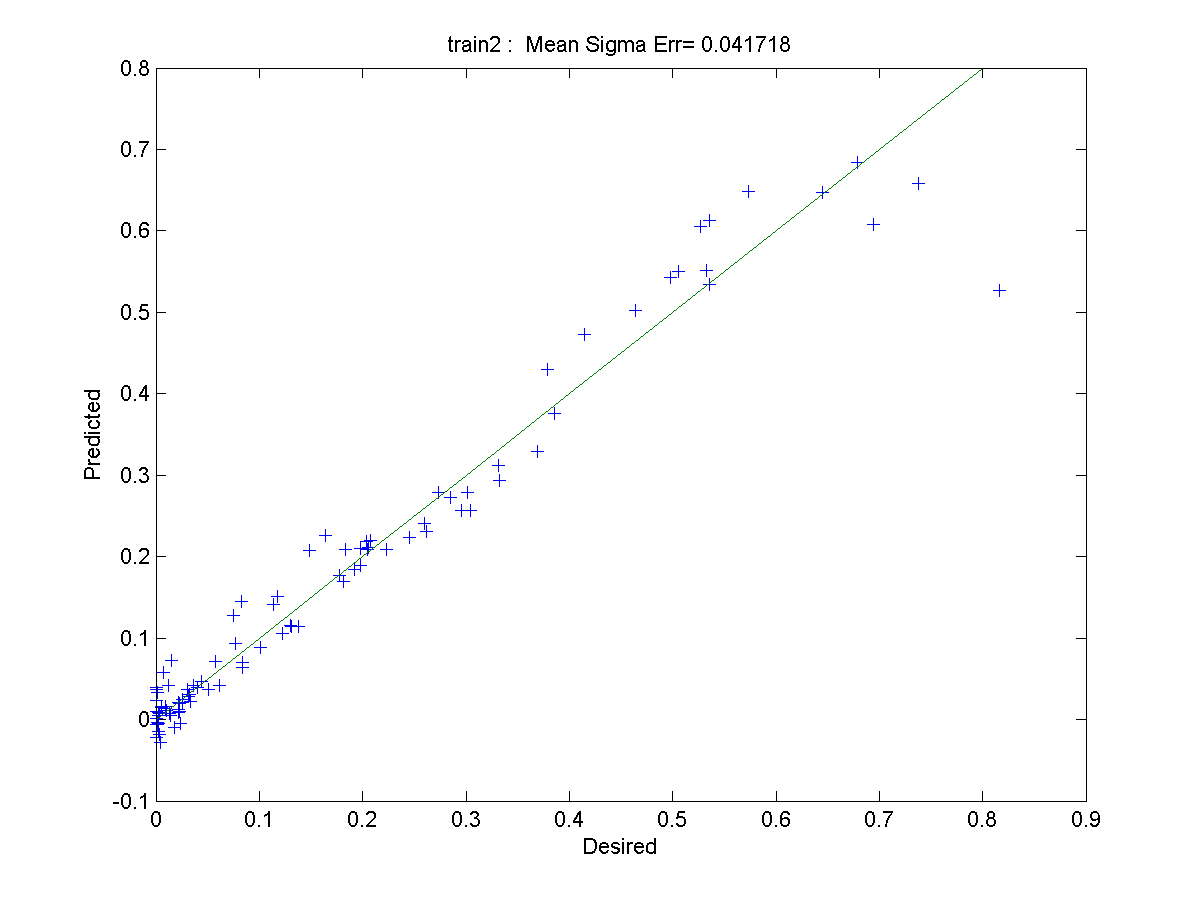
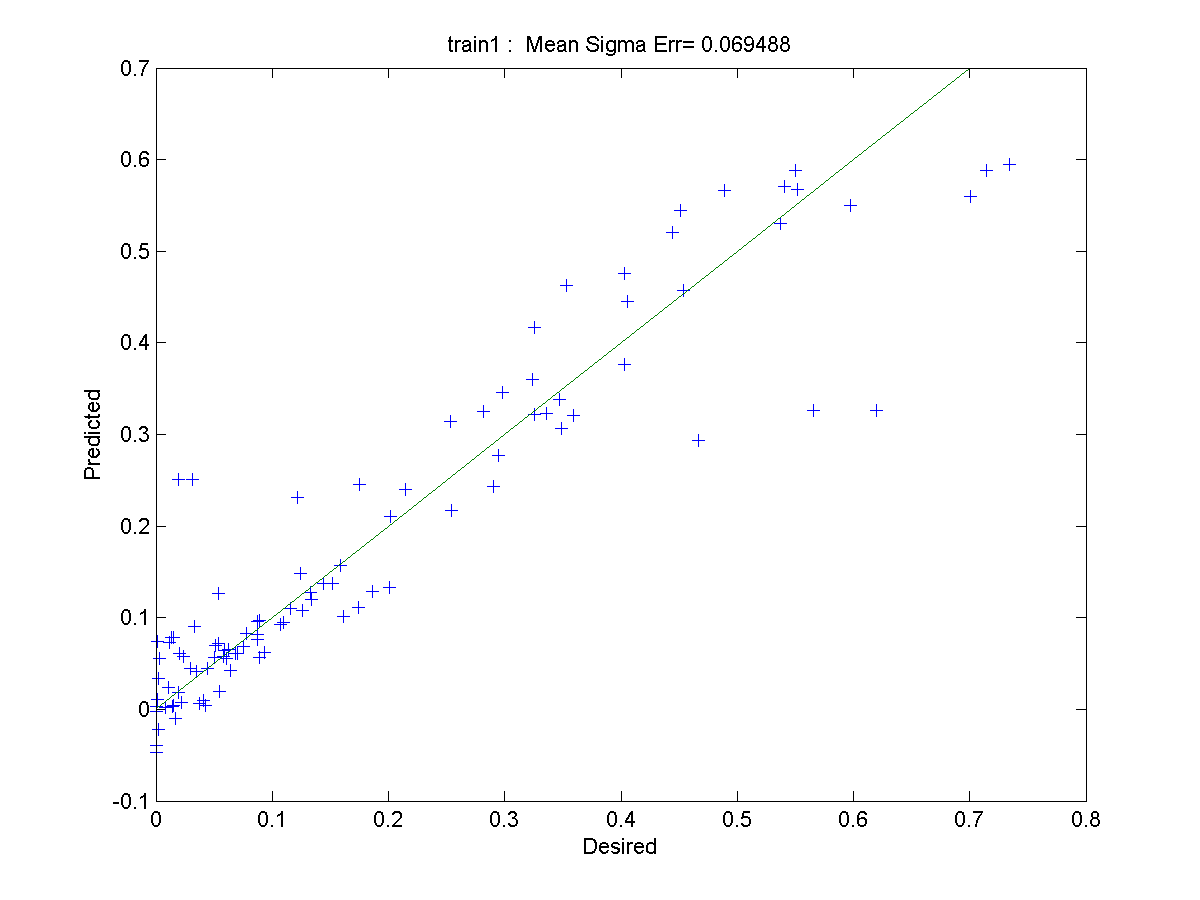
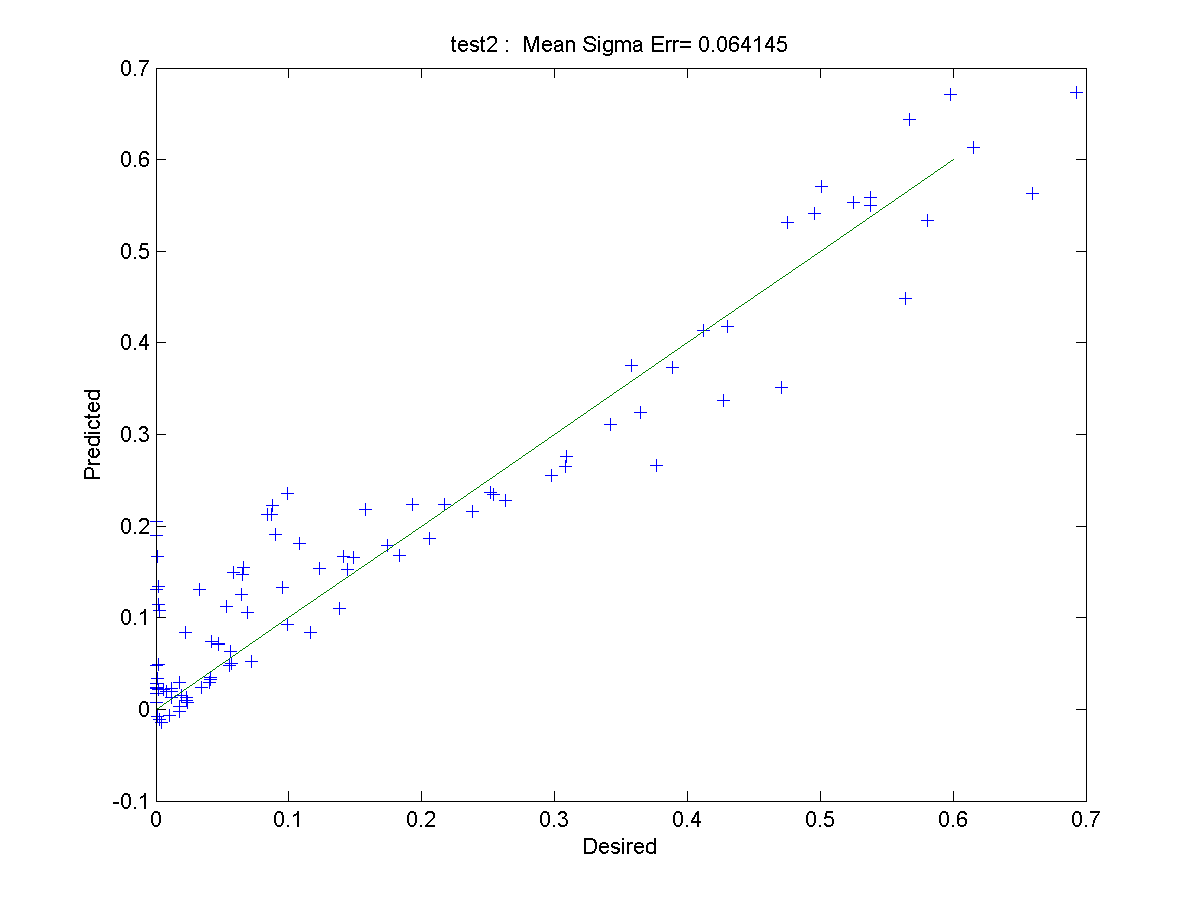
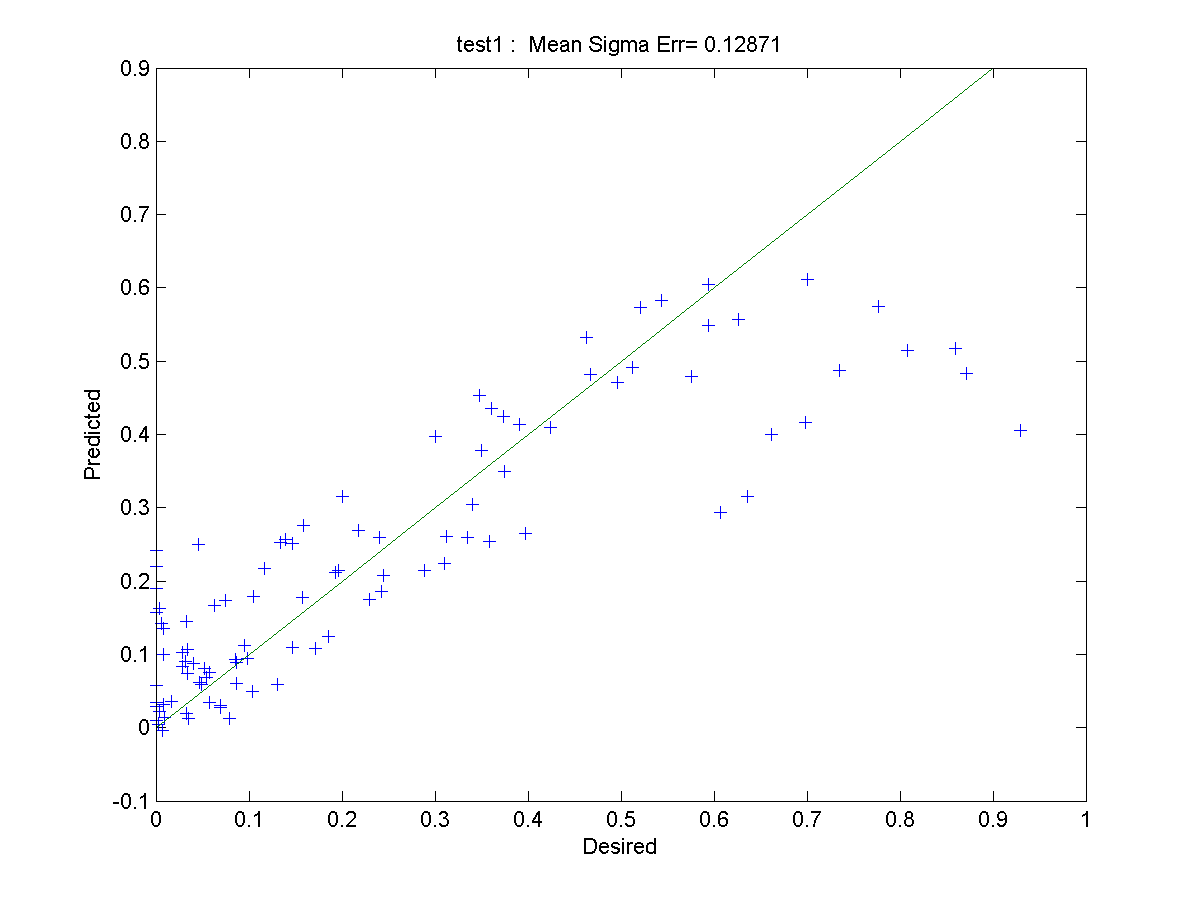
When using the single winner strategy, only winning clusters get update. We need to know how many clusters are there in advance. Otherwise, there will be some useless estimated Gaussians that do not approximate any true cluster.

When allowing all units to move, all the estimated clusters get update in each iteration. We do not need to know how many clusters are there in advance. But there will be more than one estimated Gaussians that approximate a true cluster.

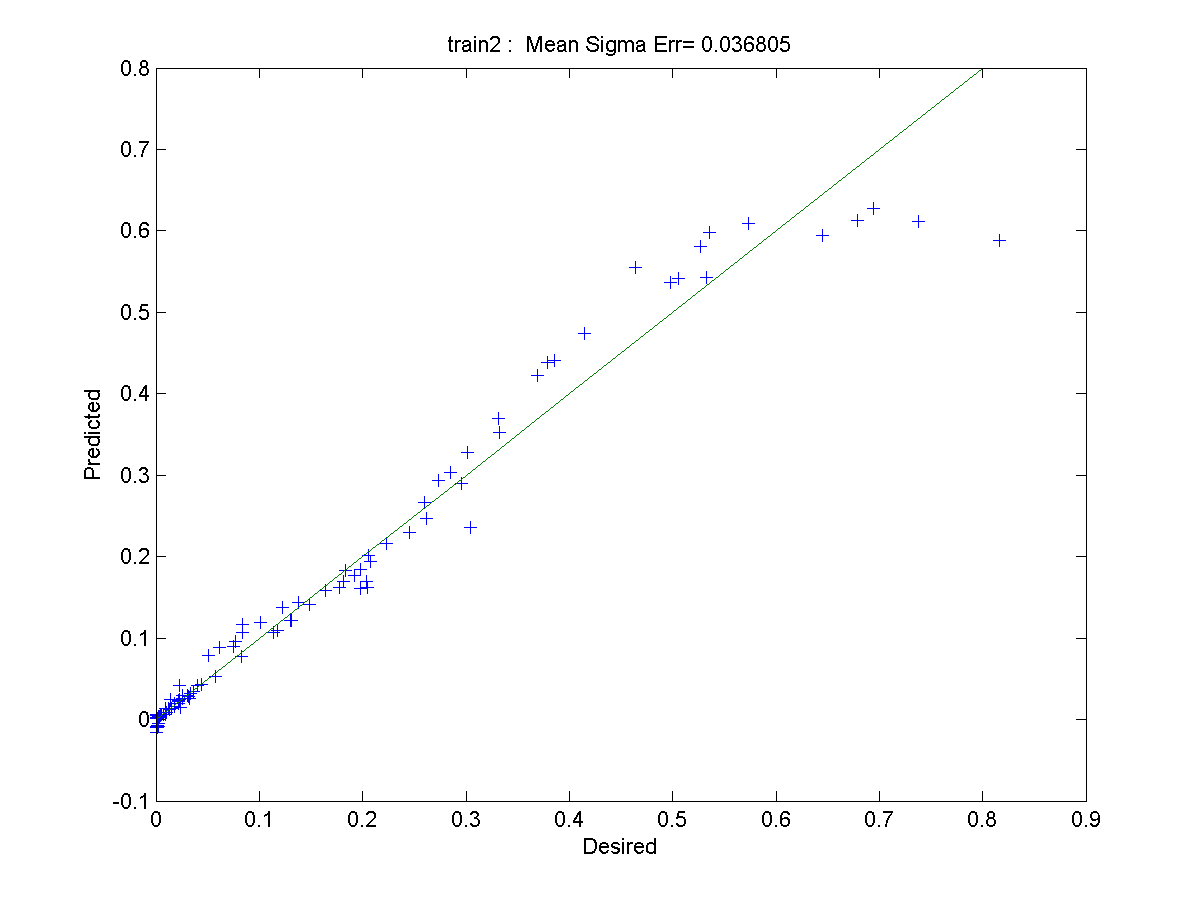
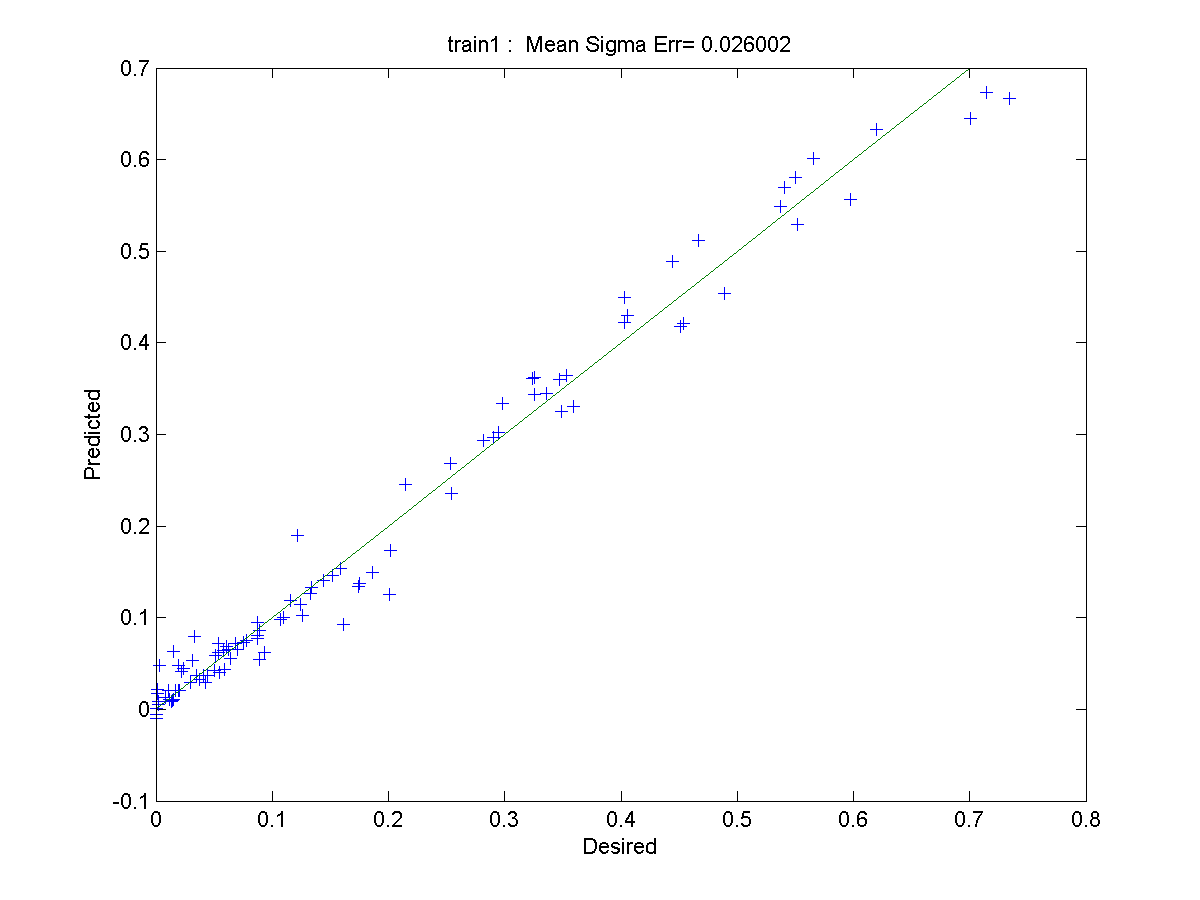
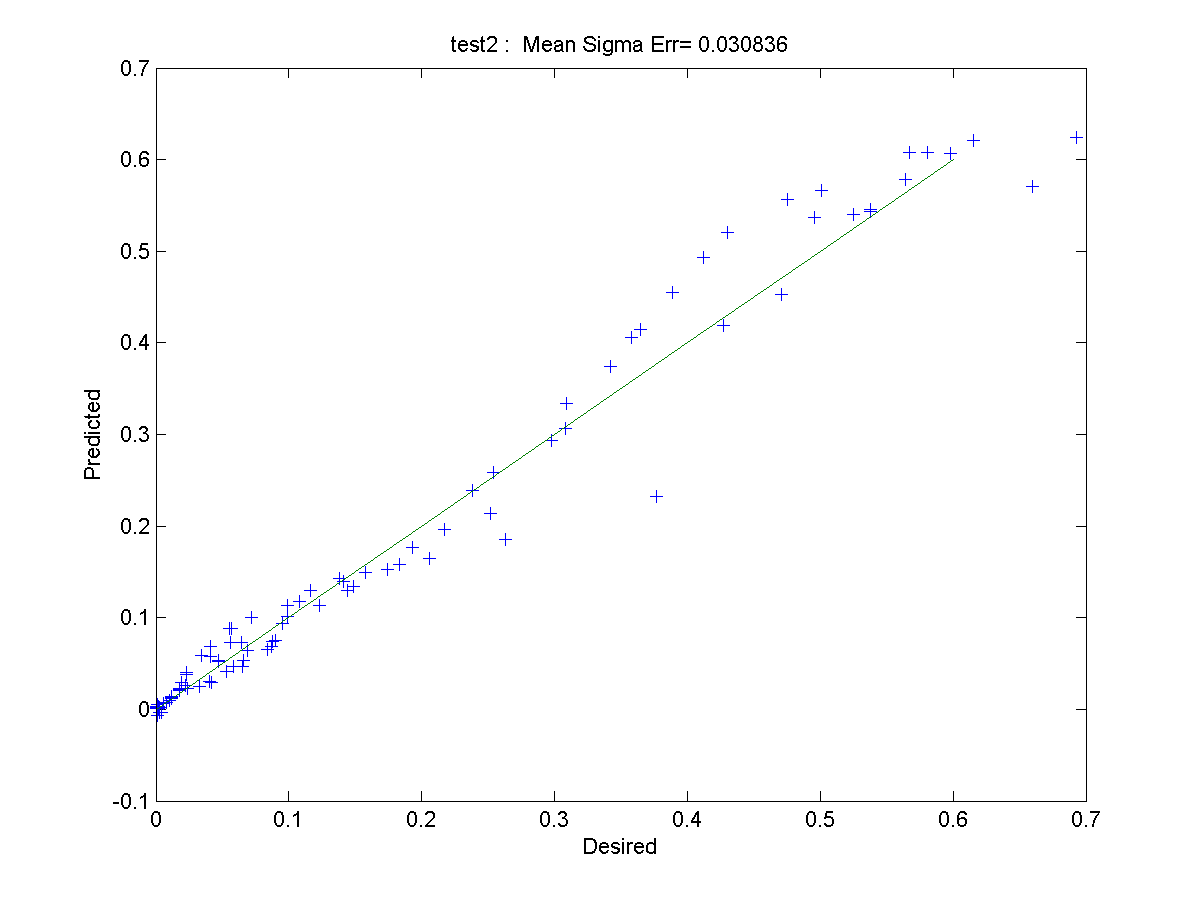
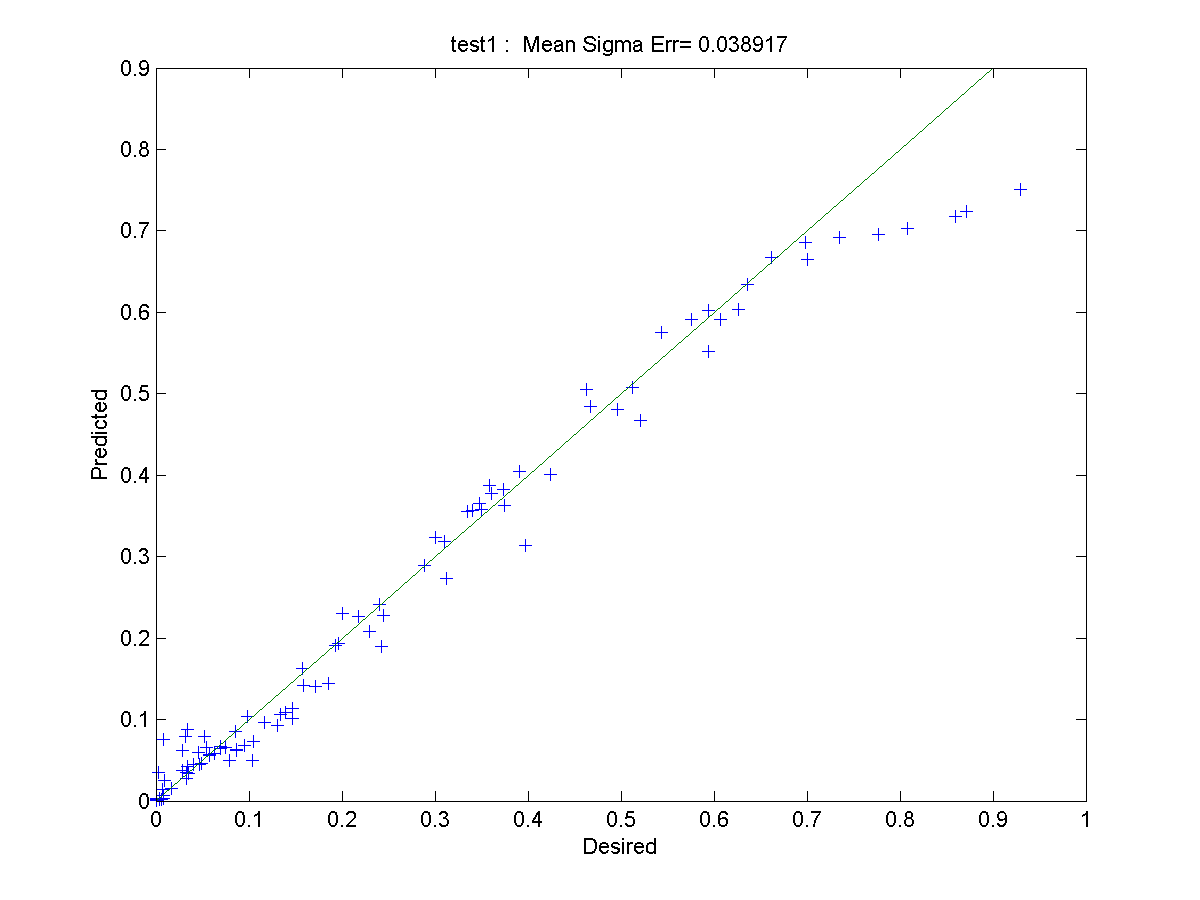


**Improvement1: reducing numbers of units**

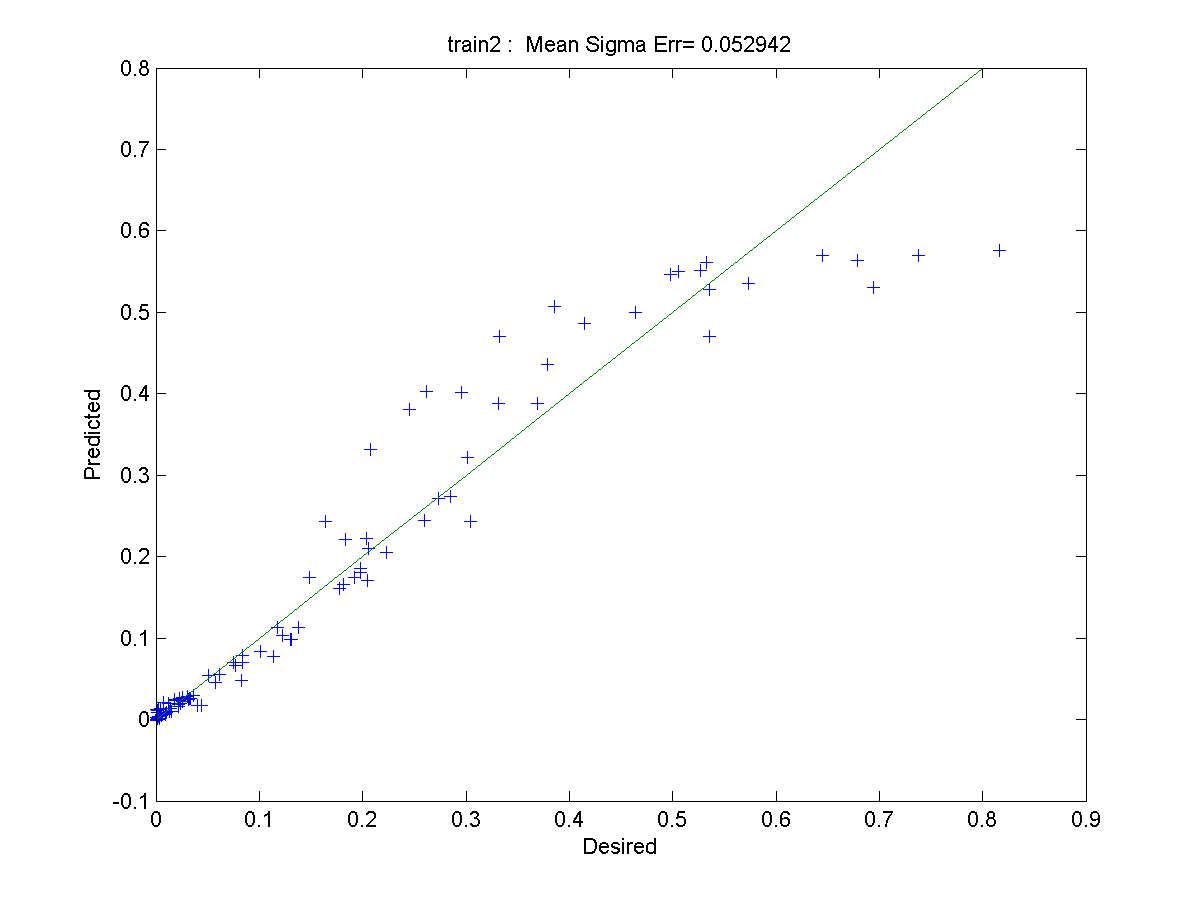
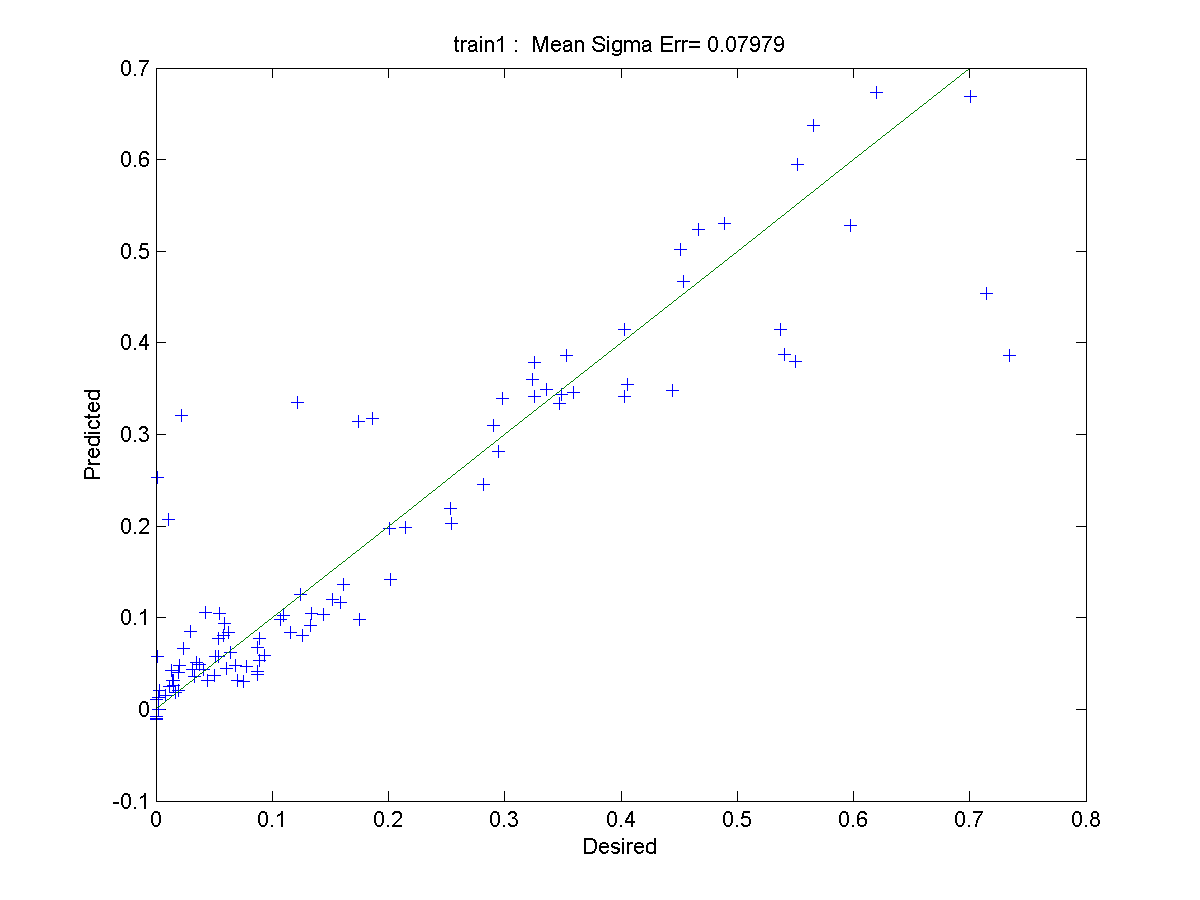
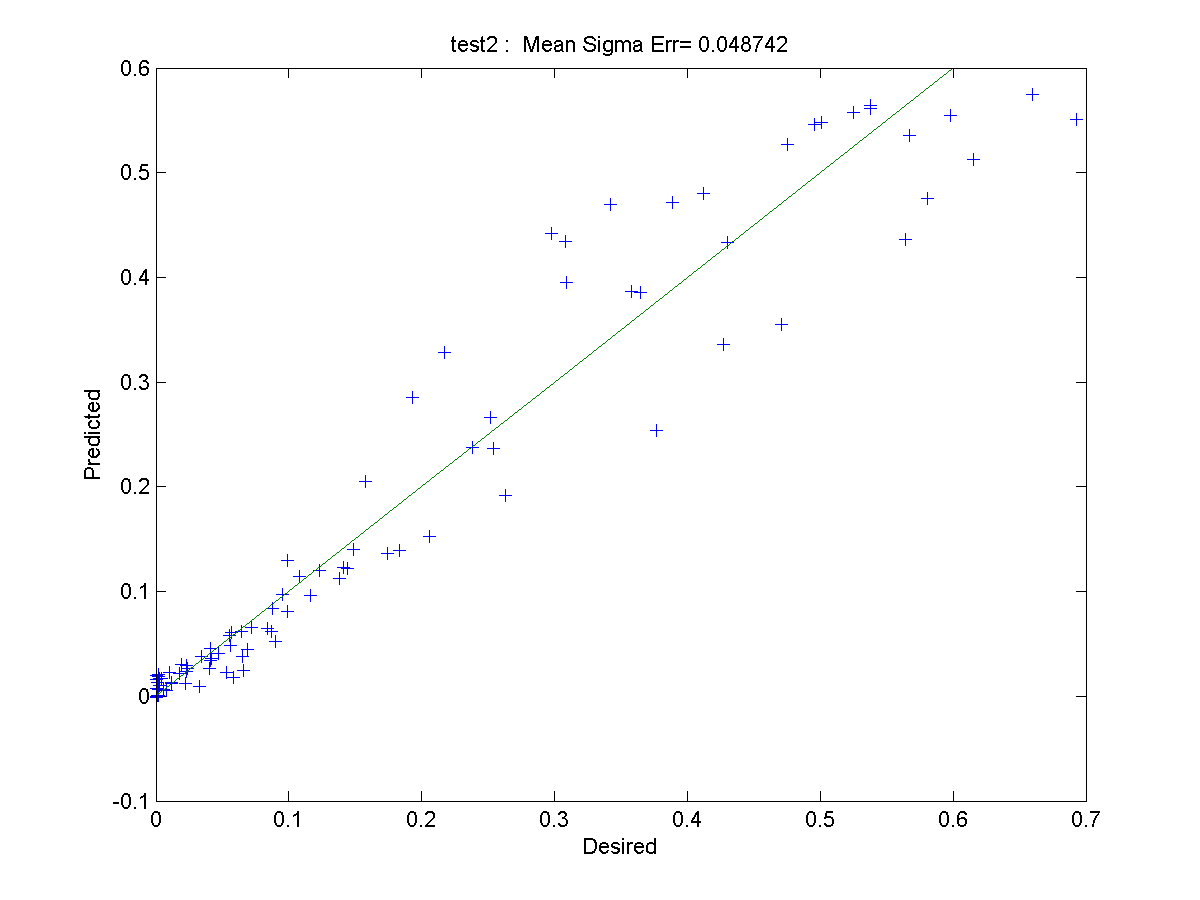
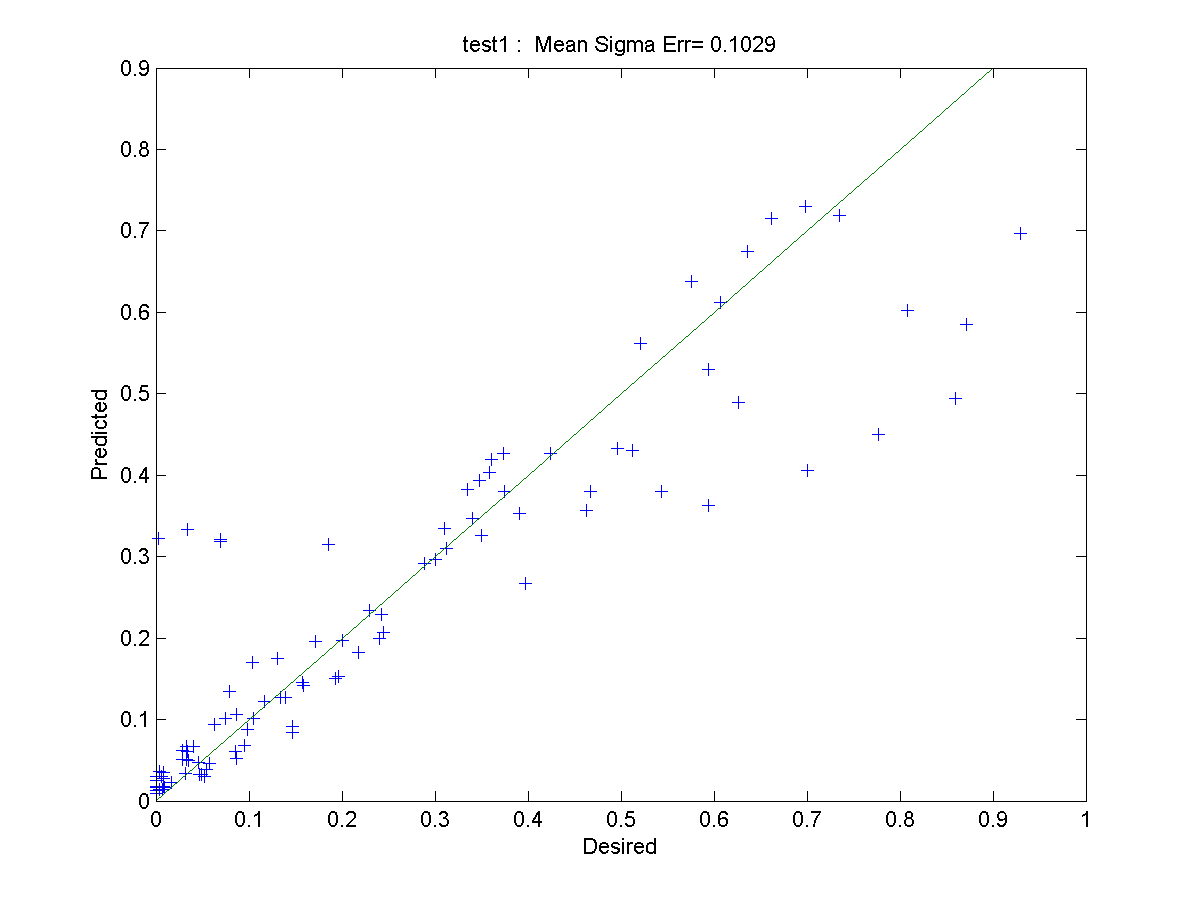
**Units=20**

****

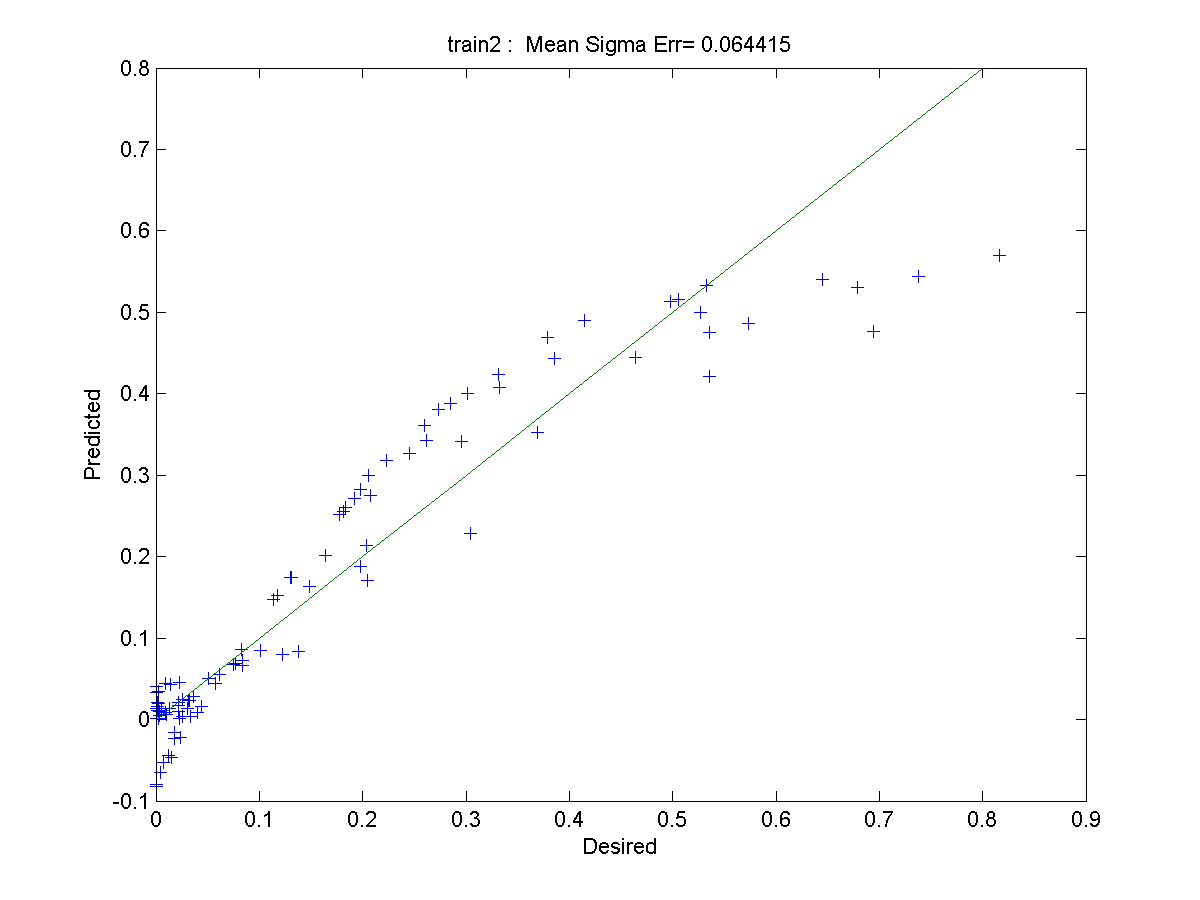
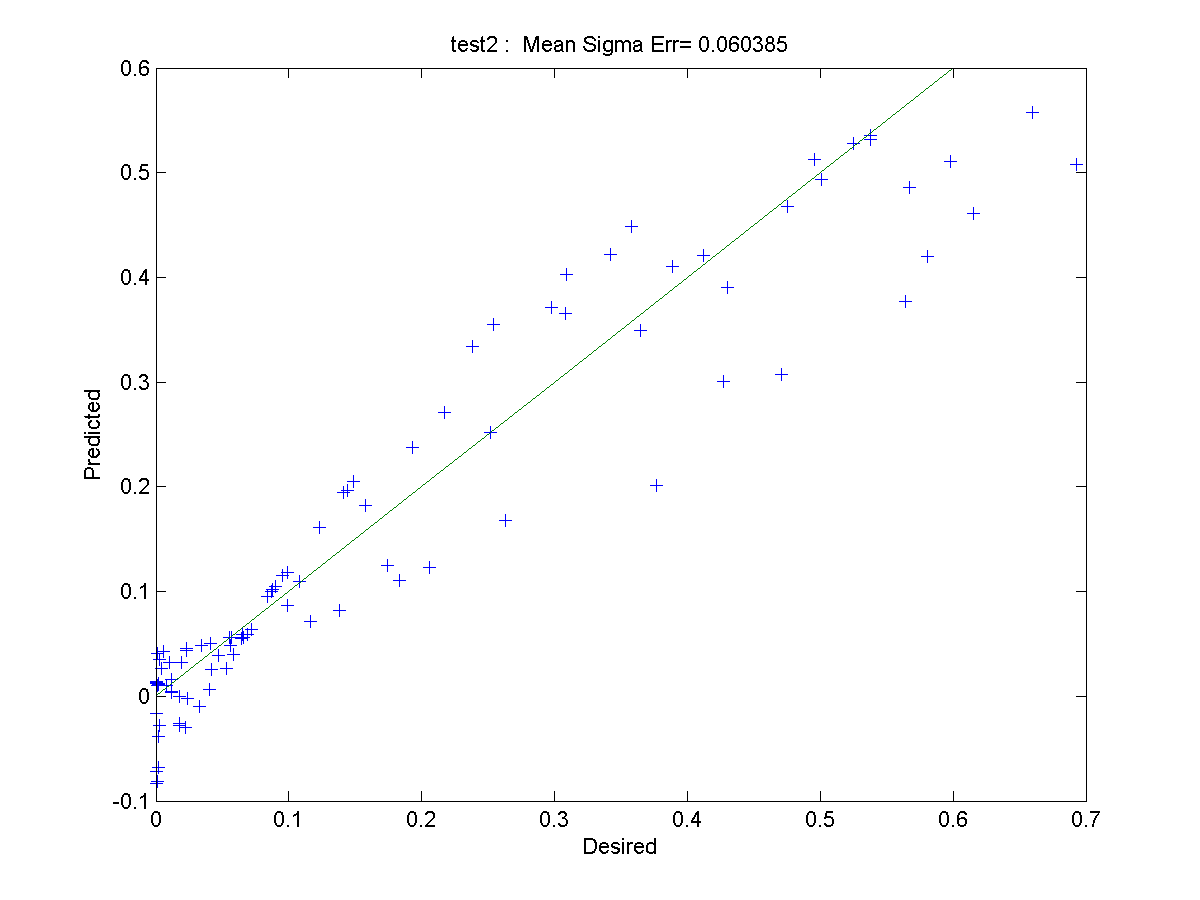
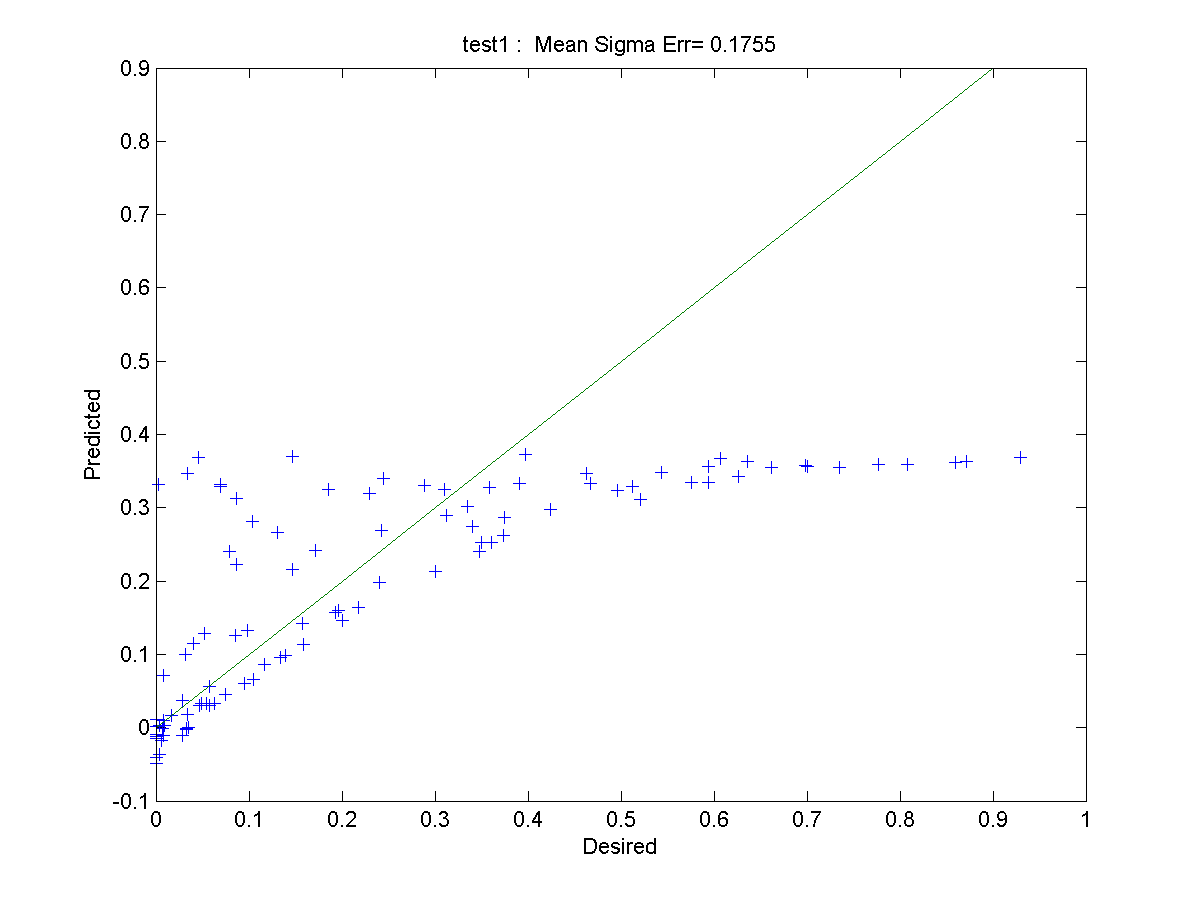
**Units=15:**

****

**Units=10:**

****

**Units=5:**

****

**Both training error and test error will reduce when units change from 20 to 15. When the units further decrease, the error will increase. The following table can show this process.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Units** | **20** | **15** | **10** | **5** |
| **Train1** | **0.069488** | **0.026002** | **0.07979** | **0.14282** |
| **Train2** | **0.04178** | **0.036805** | **0.052942** | **0.064415** |
| **Test1** | **0.12891** | **0.03817** | **0.1029** | **0.1755** |
| **Test2** | **0.064145** | **0.030836** | **0.04874** | **0.060383** |

**The original positions of units and derivation are same.**

**Improvement2: using low pass filter for the input to get rid of noise**

We try with different filters (low pass filter, median filter, moving averaging filter, see link:

<http://se.mathworks.com/help/signal/examples/signal-smoothing.html#zmw57dd0e232> )

But it does not help to improve the results. We find it hard to choose an appropriate filter.

We consider that the main reason is that the dataset is not appropriate to be processed with low pass filter.