MACHINE LEARNING IN ROBOTICS
Assignment 1

Stefan Röhrl
Technische Universität München, Arcisstraße 21, Munich, Germany
Email: stefan.roehrl@tum.de

I. Exercise

a) For k = 2 the best order with the lowest average error overall test-sets are:

$$p_1 = 5 \tag{1}$$
$$p_2 = 2 \tag{2}$$

With these polynomial orders the total errors are:

$$err_{xy} = 0.2254 \tag{3}$$
$$err_\theta = 0.0797 \tag{4}$$

The parameters determined with k=2 are listed below:

$$\begin{pmatrix}
a_{1,1}= & -0.0034 & a_{2,1}= & -0.0013 & a_{3,1}= & -0.0005 \\
a_{1,2}= & 0.9219 & a_{2,2}= & -0.0020 & a_{3,2}= & -0.0002 \\
a_{1,3}= & -0.0112 & a_{2,3}= & 0.0124 & a_{3,3}= & 1.0008 \\
a_{1,4}= & 0.0087 & a_{2,4}= & 0.4703 & a_{3,4}= & 0.0001 \\
a_{1,5}= & 0.0014 & a_{2,5}= & 0.0001 & a_{3,5}= & 0.0000 \\
a_{1,6}= & 0.0009 & a_{2,6}= & -0.0035 & a_{3,6}= & -0.0004 \\
a_{1,7}= & -0.0018 & a_{2,7}= & 0.0004 & a_{3,7}= & -0.0001 \\
a_{1,8}= & 0.0001 & a_{2,8}= & 0.0002 & & \\
a_{1,9}= & 0.0069 & a_{2,9}= & -0.0154 & & \\
a_{1,10}= & -0.0005 & a_{2,10}= & -0.0009 & & \\
a_{1,11}= & -0.0000 & a_{2,11}= & 0.0000 & & \\
a_{1,12}= & -0.0000 & a_{2,12}= & 0.0008 & & \\
a_{1,13}= & 0.0000 & a_{2,13}= & -0.0000 & & \\
a_{1,14}= & -0.0000 & a_{2,14}= & -0.0000 & & \\
a_{1,15}= & -0.0001 & a_{2,15}= & 0.0034 & & \\
a_{1,16}= & 0.0000 & a_{2,16}= & 0.0000 & &
\end{pmatrix}$$

b) For k = 5 the best order with the lowest average error overall test-sets are:

$$p_1 = 6 \tag{5}$$
$$p_2 = 3 \tag{6}$$

With these polynomial orders the total errors are:

$$err_{xy} = 0.2253 \tag{7}$$
$$err_\theta = 0.0796 \tag{8}$$

The parameters determined with k = 5 are listed below:

$$\begin{pmatrix}
a_{1,1}= & -0.0078 & a_{2,1}= & -0.0015 & a_{3,1}= & -0.0005 \\
a_{1,2}= & 0.9210 & a_{2,2}= & -0.0021 & a_{3,2}= & -0.0004 \\
a_{1,3}= & -0.0074 & a_{2,3}= & 0.0131 & a_{3,3}= & 0.9985 \\
a_{1,4}= & 0.0086 & a_{2,4}= & 0.4704 & a_{3,4}= & -0.0000 \\
a_{1,5}= & -0.0012 & a_{2,5}= & -0.0005 & a_{3,5}= & 0.0000 \\
a_{1,6}= & 0.0777 & a_{2,6}= & 0.0071 & a_{3,6}= & -0.0004 \\
a_{1,7}= & -0.0035 & a_{2,7}= & 0.0008 & a_{3,7}= & -0.0001 \\
a_{1,8}= & 0.0002 & a_{2,8}= & 0.0002 & a_{3,8}= & 0.0000 \\
a_{1,9}= & 0.0007 & a_{2,9}= & -0.0166 & a_{3,9}= & 0.0015 \\
a_{1,10}= & -0.0005 & a_{2,10}= & -0.0009 & a_{3,10}= & 0.0000 \\
a_{1,11}= & 0.0001 & a_{2,11}= & 0.0000 & & \\
a_{1,12}= & -0.0503 & a_{2,12}= & -0.0080 & & \\
a_{1,13}= & 0.0000 & a_{2,13}= & -0.0000 & & \\
a_{1,14}= & -0.0000 & a_{2,14}= & -0.0000 & & \\
a_{1,15}= & 0.0015 & a_{2,15}= & 0.0037 & & \\
a_{1,16}= & 0.0000 & a_{2,16}= & 0.0000 & & \\
a_{1,17}= & -0.0000 & a_{2,17}= & -0.0000 & & \\
a_{1,18}= & 0.0071 & a_{2,18}= & 0.0013 & & \\
a_{1,19}= & -0.0000 & a_{2,19}= & 0.0000 & &
\end{pmatrix}$$

c) In the following you will find the requested plots. The trajectory is estimated with the saved parameters.

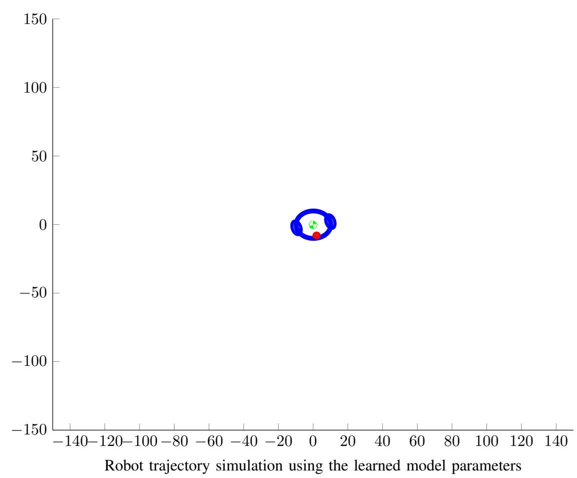

Figure 1. Estimated robot trajectory with input $(0, 0.05)$

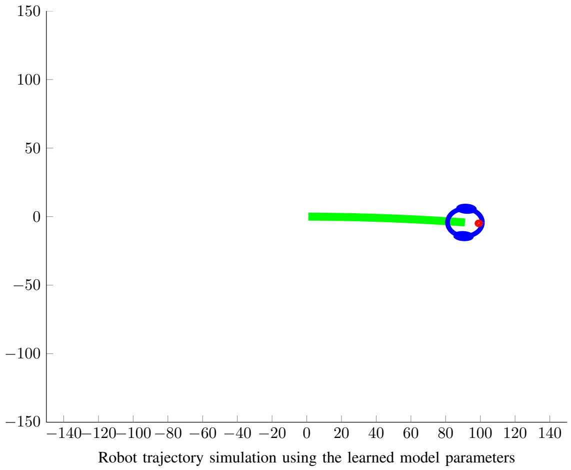

Figure 2. Estimated robot trajectory with input $(1, 0)$

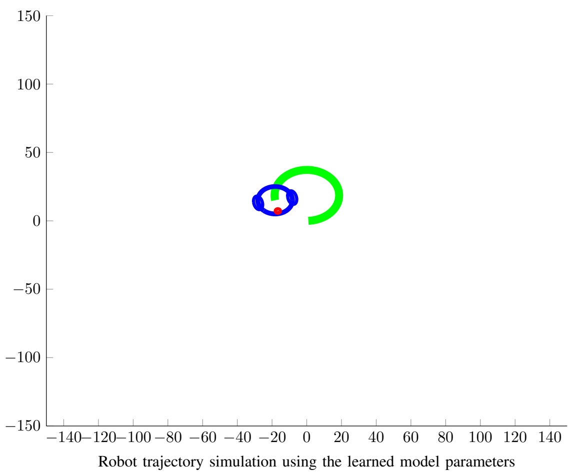

Figure 3. Estimated robot trajectory with input $(1, 0.05)$

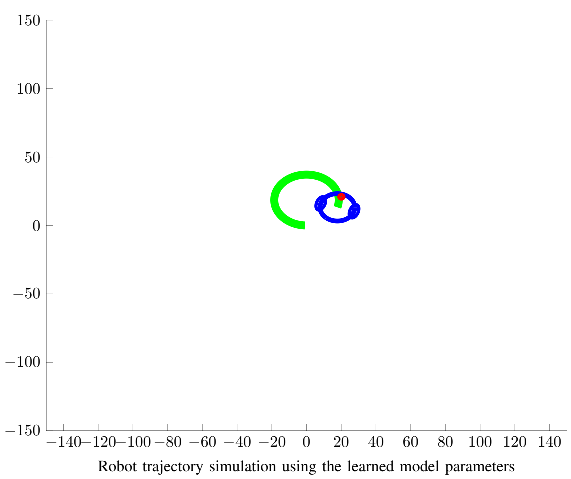

Figure 4. Estimated robot trajectory with input $(-1, -0.05)$

II. Exercise

a) The learned values for the skin model from 20 pictures of George W. Bush with 20% of the centered image are given below:

$$\mu_s = \begin{pmatrix} 176.9146 \\ 129.1766 \\ 103.8827 \end{pmatrix}$$

$$\Sigma_s = \begin{pmatrix} 1726.4 & 1431.6 & 1447.9 \\ 1431.6 & 1426.8 & 1449.8 \\ 1447.9 & 1449.8 & 1593.5 \end{pmatrix}$$

b) The learned values for the background model learned from all background images are these:

$$\mu_b = \begin{pmatrix} 103.3421 \\ 98.9777 \\ 87.1115 \end{pmatrix}$$

$$\Sigma_b = \begin{pmatrix} 5537.8 & 4744.9 & 4294.5 \\ 4744.9 & 4793.4 & 4517.4 \\ 4294.5 & 4517.4 & 5033.1 \end{pmatrix}$$

c) The following pictures (cf. Fig. 5 and Fig. 6) show the normalized LikValues ($p(x|skinmodel)$) for two unknown pictures

Figure 5. Normalized Skin Model Likelihood of SampleImage.png

Figure 6. Normalized Skin Model Likelihood of SampleImage2.png

d) The next pictures (cf. Fig. 7 and Fig. 8) show the normalized LikValues ($p(x|backgroundmodel)$) for two unknown pictures

Figure 7. Normalized Background Model Likelihood of SampleImage.png

Figure 8. Normalized Background Model Likelihood of SampleImage2.png

e) Binary classification images (white belongs most likely to skin, black belongs to background)

Figure 9. Binary classification of SampleImage.png

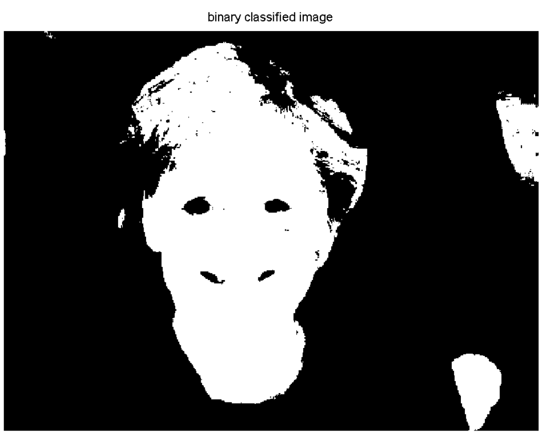

Figure 10. Binary classification of SampleImage2.png

f) The face area is surrounded by a red rectangle .

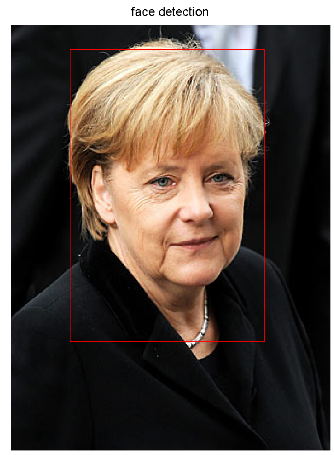

Figure 11. Face detection of SampleImage.png

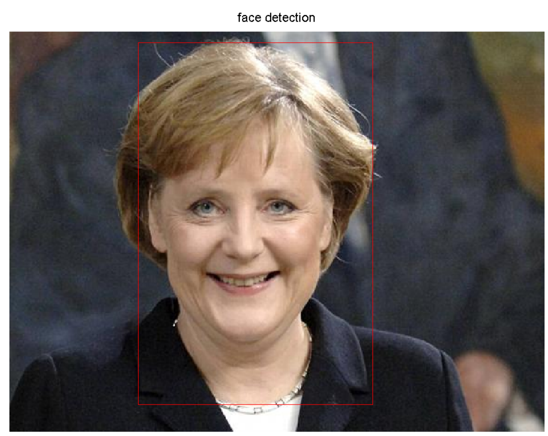

Figure 12. Face detection of SampleImage2.png

III. Exercise

At first sight one cloud think that the K-means and NUBS algorithm would produce some similar results. But this is only partly true for the o gesture (cf. Fig. 14 and Fig. 17). Even here there a differences in how the points are divided. K-means produces for every gesture a good separation in clear sections of the gesture. NUBS has sometimes problems the divide the sections properly. A good example is the o gesture (cf. Fig. 17) where you can find a yellow cluster, a blue cluster an then again a yellow cluster. Especially in the l and x gesture (cf. Fig. 16, 18) NUBS leads in the x,y space to clusters which are laying above each other. Where as K-means is able to separate also this gestures properly (cf. Fig. 13, 15). If there would not be the initialization with the same vectors K-means and NUBS would produce a different output every time. But as the clustering results show K-means might be a better tool for gesture recognition. If you don't know which gesture you will have to recognize you would have to choose a random initialization which could lead to bad results. Therefore you could determine some input vectors for the K-means algorithm by first performing the NUBS as a kind of prepossessing. A comparison of the results for the two algorithms can be found below:

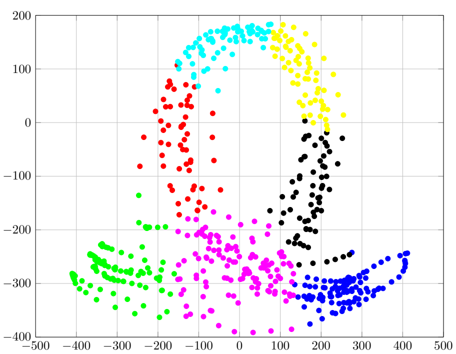

Figure 13. K-means on gesture l

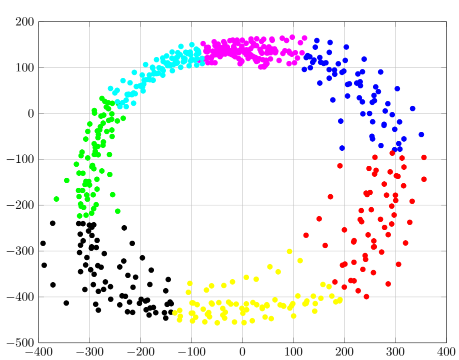

Figure 14. K-means on gesture o

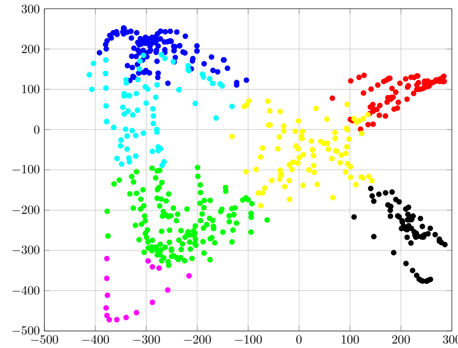

Figure 15. K-means on gesture x

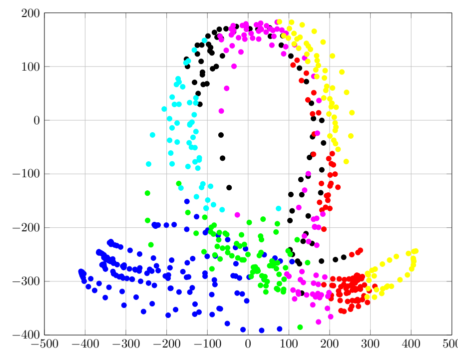

Figure 16. NUBS on gesture l

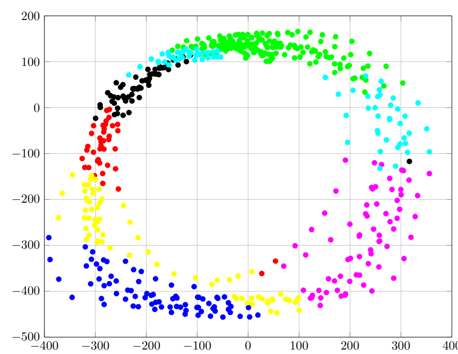

Figure 17. NUBS on gesture o

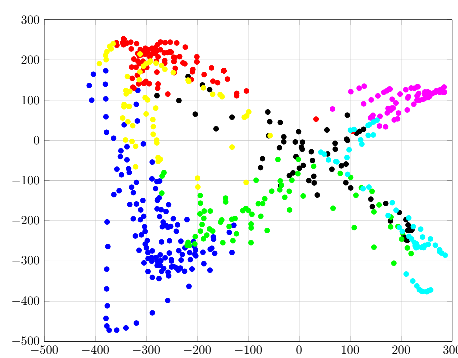

Figure 18. NUBS on gesture x