# How the plots were generated

All .fig files can be found in Latex/figures

The following instructions are given to regenerate the figures from the raw data or form the code. It hopefully gives enough context to the figures so that one can understand how to modify them if needed. For more information on the functioning of the code, see document “How to use the code”.

## Traj only scenarios (PI2 GMM 31 tune exploration noise/)

Fig.1 a

Original figures: PI2 GMM 31 tune exploration noise/ results 31/ Nshape conf/ example run / intTraj.fig

To generate the plot:

In PI2 GMM 31 tune exploration noise/ Run main.m with p = readProtocol('protocol\_NShape.txt'); (6th line)

It is the first plot that appears.

Then add to it the demo trajectories that are in “Nshape.mat”

Fig.1 b

Original figures: PI2 GMM 31 tune exploration noise/ results 31/ Pshape conf/noise 01/ example run/ finalTraj.fig

To generate the plot:

In PI2 GMM 31 tune exploration noise/ Run main.m with p = readProtocol('protocol\_PShape.txt'); (6th line)

It is the first plot that appears.

Then add to it the demo trajectories that are in “Pshape.mat”

Fig.2 up left

Original figures: PI2 GMM 31 tune exploration noise/ results 31/ Nshape conf/ example run / finalTraj.fig

This plot was the last plot generated by running main.m with protocol\_NShape, to which the initial trajectory and the via-points were added.

To re generate the plot:

Load results\_protocol1.mat, plot results\_this\_protocol.D\_init.traj\_y + results\_this\_protocol.Dfin.traj\_y

Then use plotStreamLines.m with the variables found in results\_this\_protocol.GMR

Then, from plotViapoints.m, execute cell N shape 5.

Fig.2 down left

Original figures: PI2 GMM 31 tune exploration noise/ results 31/ Pshape conf/ example run / finalTraj.fig

This plot was the last plot generated by running main.m with protocol\_PShape. It was the last plot, to which the initial trajectory and the via-points were added.

To re generate the plot:

Load results\_protocol1.mat, plot results\_this\_protocol.D\_init.traj\_y + results\_this\_protocol.Dfin.traj\_y

Then use plotStreamLines.m with the variables found in results\_this\_protocol.GMR

Then, from plotViapoints.m, execute cell N shape 5.

Fig.2 top right

Original figure: PI2 GMM 31 tune exploration noise/ results 31/ Nshape conf/ meanLearningCurve.fig

To generate figure: load “PI2 GMM 31 tune exploration noise/ results 31/ Nshape conf/ results\_protocol1.mat”, then execute “plots.m”. It is the first figure

Fig.2 down right

Original figure: PI2 GMM 31 tune exploration noise/ results 31/ Pshape conf/ meanLearningCurve.fig

To generate figure: load “PI2 GMM 31 tune exploration noise/ results 31/ Pshape conf/ results\_protocol1.mat”, then execute “plots.m”. It is the first figure

Fig.3 left

Original figure: PI2 GMM 31 tune exploration noise/ results 31/ compa norm veloNoise/ Nshape/ Basic/ colormap mean float.fig

This figure is based on the data contained in PI2 GMM 31 tune exploration noise/ results 31/ compa norm veloNoise/ Nshape/ Basic/ vNoiseNormMeanBasic.mat

The data was generated using the script testNoiseTuningOriginal.m loading Nshape\_GMR beforehand.

Fig.3 right

Original figure: PI2 GMM 31 tune exploration noise/ results 31/ compa norm veloNoise/ Nshape/ Basic/ colormap mean float.fig

This figure is based on the data contained in PI2 GMM 31 tune exploration noise/ results 31/ compa norm veloNoise/ Nshape/ Rect/ vNoiseNormMeanBasic.mat

The data was generated using the script testNoiseTuningOriginal.m loading Nshape\_GMR beforehand.

## Box SEDS scenario (PI2 GMM 31 tune exploration noise/)

The box scenario can be run with “main.m” using “protocol\_boxSEDS.txt”. It is necessary to open “runPI2GMMLearning.m” and to uncomment lines 800-804 and 823-826 to activate the effect of the box.

Fig.4

Original figure: PI2 GMM 31 tune exploration noise/ results 31/SEDS box task conf/ example run/ initTraj.fig

The figure is the first one generated in the box scenario. The box must be added to the figure using plotViapoints.m and executing cell “Plot box”. The demonstration trajectories come from “shapeBoxSEDS3.mat”

Fig.5 left

Original figure: PI2 GMM 31 tune exploration noise/ results 31/SEDS box task conf/ example run/ finalTraj.fig

This plot was the last plot generated by running main.m with protocol\_boxSEDS.txt. It was the last plot, to which the initial trajectory and the via-points were added.

To re generate the plot:

Load PI2 GMM 31 tune exploration noise/ results 31/SEDS box task conf/ example run/result.mat

plot results\_this\_protocol.Dfin.traj\_y

Then use plotStreamLines.m with the variables found in results\_this\_protocol.GMR

Then, from plotViapoints.m, execute cell “Plot box”.

Fig.5 Right

Original figure: PI2 GMM 31 tune exploration noise/ results 31/SEDS box task conf/ noise0.04/ medLearningCurve.fig

To generate figure: load “PI2 GMM 31 tune exploration noise/ results 31/ SEDS box task conf/ noise0.04/results.mat”, then execute “plots.m”. It is the second figure

### Divergent force field scenario (PI2 GMM 35 further kp learning (diverent field)/)

The “divergent force field” scenario” can be run from PI2 GMM 35 further kp learning (diverent field)/main.m using “protocol\_test.txt”

Fig. 6

This is the first picture generated by this scenario. Then the via-points are added by executing the cell “sine corrected” from plotViapoint.m

Fig. 7 top left

To generate figure:

Load PI2 GMM 35 further kp learning (diverent field)/new results report/n\_dim\_kp 2/kp0 20/results\_protocol2.mat

Run plots.m -> it is the second figure

Fig. 7 top right

To generate figure:

Load PI2 GMM 35 further kp learning (diverent field)/new results report/n\_dim\_kp 2/kp0 20/results\_protocol2.mat

Run plots.m -> it is the fourth figure

Fig. 7 down left & right

To generate figure

Load PI2 GMM 35 further kp learning (diverent field)/new results report/n\_dim\_kp 2/kp0 20/results\_protocol2.mat

Run plot\_stiffness.m

Fig. 8

Load PI2 GMM 35 further kp learning (diverent field)/new results report/n\_dim\_kp 2/kp0 20/results\_protocol2.mat

Run plots.m -> it is the first figure

## Stochastic force field scenario (PI2 GMM 36 further kp learning (selective kp))

The “stochastic force field” scenario” can be run from PI2 GMM 36 further kp learning (selective kp)/main using “protocol\_test.txt”

Isotropic or diagonal stiffness can be selected by changing n\_dim\_kp = 1 / 2 at line 17 of runPI2GMMLearning.m

Fig. 9

The figure is the first one generated when running the scenario with kp0 = 10

The via-points are added by executing the cell “Sine corrected” in plotViapoints.m

The force field was added by executing “plot simple force field.m”

Fig. 10 Top left

This figure was the last one generated when in one run of the scenario with n\_dim\_kp = 2, kp0 = 10

The exact data for this plot has not been saved in a .mat format (it was a later execution just for illustration purpose). However similar plots can be obtained by loading “PI2 GMM 36 further kp learning (selective kp)\new Results 36 report\cost func 2\n\_dim\_kp 2\kp0 10\results\_protocol1.mat” and then plotting results\_this\_protocol(x).Dfin(1).q, results\_this\_protocol(x).Dfin(2).q and results\_this\_protocol(x).Dfin(1).traj\_y.

Viapoints and force field are added as for fig.9

Fig. 10 Top right

Figure generated by loading “PI2 GMM 36 further kp learning (selective kp)\new Results 36 report\cost func 2\n\_dim\_kp 2\kp0 10\results\_protocol1.mat” and executing plots.m. It is the third figure.

Fig. 10 Bottom left & right

Figure generated by loading “PI2 GMM 36 further kp learning (selective kp)\new Results 36 report\cost func 2\n\_dim\_kp 2\kp0 10\results\_protocol1.mat” and executing plot\_stiffness.m

Fig. 11

Figure generated by loading “PI2 GMM 36 further kp learning (selective kp)\new Results 36 report\cost func 2\n\_dim\_kp 2\kp0 10\results\_protocol1.mat” and executing plots.m. It is the first figure.

Fig. 12

Figure based on the two following files:

Isotropic: “PI2 GMM 36 further kp learning (selective kp)\new Results 36 report\cost func 2\n\_dim\_kp 1\kp0 10\results\_protocol1.mat

Diagonal: “PI2 GMM 36 further kp learning (selective kp)\new Results 36 report\cost func 2\n\_dim\_kp 2\kp0 10\results\_protocol1.mat

Fig. 13

Inkscape file for this figure can be found in Digging task photos/HWsetting.svg

## HW experiment (PI2GMM41 HW imp/)

Fig. 14 Left

Load “PI2GMM41 HW imp\results\gravel\01.02.2015\kp0 150 offset 0 gravel Cost func\results\_protocol1.mat”

Execute utils\plots.m

Fig. 14 Right

Load “PI2GMM41 HW imp\results\gravel\01.02.2015\kp0 400 offset 0 gravel Cost func\results\_protocol1.mat”

Execute utils\plots.m

Fig. 15

The data for this plot can be found in “PI2GMM41 HW imp\results\gravel\01.02.2015\”. The files “InitPolyKpxxx.mat” are recordings of 5 executions of the initial policy with initial uniform stiffness xxx . They are stored in result.Ds(1,1:5)

The costs for those execusions have been extracted in totCostKpxxx.

The plot shows the mean value +- 1 std for the different values of initial Kp

The two red crosses are the final cost values after learning starting with Kp = 150 and Kp =400. This values can be found in “PI2GMM41 HW imp\results\gravel\01.02.2015\kp0 xxx offset 0 gravel Cost func\results\_protocol1.mat” -> results\_this\_protocol.cost(end,3)

# How to run the code

## main.m

The software is run from “main.m”. This script reads the file “protocol\_x.txt”, where the parameters of the learning algorithm are given for the different experiments to be executed.

Then the GMM or SEDS model is learned using the demonstration trajectories in the “DemoName.mat” file specified in “protocol.txt” (demo\_set parameter). The GMM and SEDS libraries are used for this purpose (<http://lasa.epfl.ch/sourcecode/>)

Then the function “runPI2GMMLearning.m” is called to carry out the learning iterations.

## TrajDrawer.m

## Result

For each line in the protocol file, the results are stored in a variable called “results\_this\_protocol” saved to the folder containing the main script under the name “results\_protocol$.mat”, where $ is the line number.

The parameters specified in the protocol file are stored under results\_this\_protocol.p for future reference. The parameters that are not specified in the protocol file are usually indicated by the name of the folder containing the result.

## Parameters

The parameters are described in the “protocol\_x.txt” files.

There are some extra protocols that are not defined in the protocol files but in “main.m” or “runPI2GMMLearning.m”:

* Dimensions
* Stiffness dimensions
* Number of sub-trials