

In the first exercise, you have implemented an off-policy RL algorithm in the continuous state space with function approximation. In the next exercise, you saw how to formulate an MPC controller with `acados` and how to wrap it into a differentiable PyTorch module. In the third exercise, you implemented a policy gradient algorithm that works on continuous action spaces. In this exercise you will implement something that is a combination of all those concepts, namely **SAC-FOP**, an algorithm that extends the off-policy actor-critic algorithm **SAC** by using a differentiable MPC controller in the actor.

## Exercise 1: SAC-FOP

- 1.1 In `tanh_gaussian.py` you will find an implementation of a scaled TanhNormal distribution. This means that samples are drawn from a Gaussian distribution, then transformed by a tanh function to bound the samples to the interval  $[-1, 1]$  and then scaled to the bounds of the parameter space  $[p_{\text{low}}, p_{\text{high}}]$ . Implement sampling from a Gaussian in the forward pass of `SquashedGaussian`. Apply the reparameterization trick to allow backpropagation through the samples.
- 1.2 In `sac_fop.py` you will find the class `MpcRlActor`. It uses a usual neural network to predict a parameter distribution. This distribution is used to sample parameters that are input to the MPC controller, which outputs the action. Take a look at the forward pass of the `MpcRlActor`.
- 1.3 In the `train_loop` method of the `SacFopTrainer` class, an implementation of SAC is given, using the `MpcRlActor` class from above. Read through this method to make yourself familiar with SAC. Run `run.py` to train the actor on the Cart Pole example. Statistics and rendered video of the run will be saved in the "output" directory. Open another terminal, then run `pip install tensorboard` and then `tensorboard --logdir=output` to start Tensorboard. It will show you a link in your terminal that you can access for seeing your runs. You will see many statistics being logged, e.g., in the "train" group in the "train/r" chart you can see the cumulative reward being achieved in the rollouts. Since training takes some time, we provide a checkpoint of the networks etc. to start training from and which is used by default. Also statistics and videos from this runs is provided there.