AE 4803 Robotics and Autonomy
Professor Evangelos Theodorou
Homework 3

Luis Pimentel — lpimentel3@gatech.edu
Jackson Crandell — jackcrandell@gatech.edu

November 25, 2020

## Problem 1.

We formulate our experiment with the following scalar system:

$$A = [0.4]$$
$$B = [0.9]$$
$$Q = [0.01]$$
$$R = [0.001]$$

MATLAB's ***dlqr*** function computes the optimal gain K = 0.3964.

We formulate the following Reinforcement Learning optimization problem and solving using gradient ascent with Finite Differencing gradient estimation:

$$\underset{\theta}{\text{maximize}} \quad \mathbb{E}\Big[R(\boldsymbol{\tau})\Big]$$
$$R(\boldsymbol{\tau}) = \sum_{t=0}^{N} r(\boldsymbol{x}_t, \boldsymbol{u}_t, t)$$
$$r(\boldsymbol{x}_t, \boldsymbol{u}_t, t) = -\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u}$$
$$\boldsymbol{u} = -\boldsymbol{\theta}\boldsymbol{x}$$

This results in an optimal $\boldsymbol{\theta}^\star = 0.3990$. Our convergence criterion fulfills that the gradient is sufficiently small for some $\epsilon$ or that our Reward begins to decrease after increasing for a certain number of times.

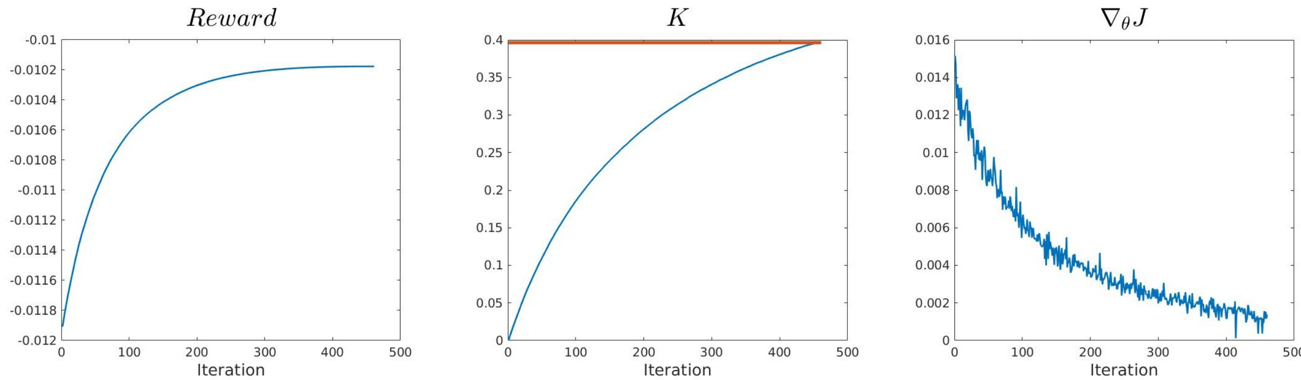


Figure 1: Results using Finite Differencing policy gradient estimation.

## Problem 2.

**2.1**) For the derivation of the REINFORCE Gradient we begin with the following cost function:

$$J(\boldsymbol{\theta}) = \int p(\boldsymbol{\tau})R(\boldsymbol{\tau})\, d\boldsymbol{\tau}$$

A trajectory can be expressed as $\boldsymbol{\tau} = (\boldsymbol{x_0}, \boldsymbol{u_0}, \ldots, \boldsymbol{x_{N-1}}, \boldsymbol{u_{N-1}}, \boldsymbol{x_N})$ with states $\boldsymbol{x} \in \mathbb{R}^\ell$ and controls $\boldsymbol{u} \in \mathbb{R}^p$ over the time horizon T = Ndt. R($\boldsymbol{\tau}$) is the accumalated cost over a trajectory and $p(\boldsymbol{\tau})$ represents the path probability of the trajectory, which using Bayesian and Markov properties can be expressed as:

$$p(\boldsymbol{\tau}) = p(\boldsymbol{x}_0) \prod_{i=0}^{N-1} p(\boldsymbol{x}_{i+1}|\boldsymbol{x}_i, \boldsymbol{u}_i)p(\boldsymbol{u}_i|\boldsymbol{x}_i; \boldsymbol{\theta})$$

$$R(\boldsymbol{\tau}) = \sum_{t=0}^{N-1} r(\boldsymbol{x}_t, \boldsymbol{u}_t, t)$$

The $p(\boldsymbol{u}_i|\boldsymbol{x}_i; \boldsymbol{\theta})$ term in path probability represents the parametrized policy where $\boldsymbol{\theta} \in \mathbb{R}^n$. We begin our derivation by the gradient of the cost function with respect to $\boldsymbol{\theta}$, $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$.

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \left( \int p(\boldsymbol{\tau})R(\boldsymbol{\tau})\, d\boldsymbol{\tau} \right)$$

$$\int \nabla_{\boldsymbol{\theta}} p(\boldsymbol{\tau})R(\boldsymbol{\tau})\, d\boldsymbol{\tau}$$

We use the following log property:

$$\nabla_{\boldsymbol{\theta}} log(p(\boldsymbol{\tau})) = \frac{1}{p(\boldsymbol{\tau})}\nabla_{\boldsymbol{\theta}} p(\boldsymbol{\tau})$$

$$\nabla_{\boldsymbol{\theta}} p(\boldsymbol{\tau}) = p(\boldsymbol{\tau})\nabla_{\boldsymbol{\theta}} log(p(\boldsymbol{\tau}))$$

To make the substitution and rewrite the following as an expectation:

$$\int p(\boldsymbol{\tau})\nabla_{\boldsymbol{\theta}} log(p(\boldsymbol{\tau}))R(\boldsymbol{\tau})\, d\boldsymbol{\tau}$$

$$E_{p(\boldsymbol{\tau})}\Bigg[\nabla_{\boldsymbol{\theta}} log(p(\boldsymbol{\tau}))R(\boldsymbol{\tau})\Bigg]$$

We start by calculating $\nabla_{\boldsymbol{\theta}} log(p(\boldsymbol{\tau}))$

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} log(p(\boldsymbol{\tau})) &= \nabla_{\boldsymbol{\theta}} log\Bigg(p(\boldsymbol{x}_0)\prod_{i=0}^{N-1} p(\boldsymbol{x}_{i+1}|\boldsymbol{x}_i,\boldsymbol{u}_i)p(\boldsymbol{u}_i|\boldsymbol{x}_i;\boldsymbol{\theta})\Bigg) \\
&= \nabla_{\boldsymbol{\theta}} log(p(\boldsymbol{x}_0)) + \nabla_{\boldsymbol{\theta}} log\Bigg(\prod_{i=0}^{N-1} p(\boldsymbol{x}_{i+1}|\boldsymbol{x}_i,\boldsymbol{u}_i)\Bigg) + \nabla_{\boldsymbol{\theta}} log\Bigg(\prod_{i=0}^{N-1} p(\boldsymbol{u}_i|\boldsymbol{x}_i;\boldsymbol{\theta})\Bigg) \\
&= \nabla_{\boldsymbol{\theta}} log(p(\boldsymbol{x}_0)) + \nabla_{\boldsymbol{\theta}}\sum_{i=0}^{N-1} log(p(\boldsymbol{x}_{i+1}|\boldsymbol{x}_i,\boldsymbol{u}_i)) + \nabla_{\boldsymbol{\theta}}\sum_{i=0}^{N-1} log(p(\boldsymbol{u}_i|\boldsymbol{x}_i;\boldsymbol{\theta})) \\
&= \sum_{i=0}^{N-1}\nabla_{\boldsymbol{\theta}} log(p(\boldsymbol{u}_i|\boldsymbol{x}_i;\boldsymbol{\theta}))
\end{aligned}$$

From this we rewrite our policy gradient as

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = E_{p(\boldsymbol{\tau})}\Bigg[\sum_{i=0}^{N-1}\nabla_{\boldsymbol{\theta}} log(p(\boldsymbol{u}_i|\boldsymbol{x}_i;\boldsymbol{\theta}))R(\boldsymbol{\tau})\Bigg]$$

We can further simplify this by calculating $p(\boldsymbol{u}_i|\boldsymbol{x}_i;\boldsymbol{\theta})$ given the parametrized policy with Gaussian noise $\epsilon \sim \mathcal{N}(0,\boldsymbol{I})$:

$$u(\boldsymbol{x},\boldsymbol{\theta},t_k) = \boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{\theta} + \boldsymbol{B}(\boldsymbol{x})\epsilon(t_k)$$

We start by expressing $p(\boldsymbol{u}_i|\boldsymbol{x}_i;\boldsymbol{\theta})$ as the multi-variate Gaussian Distribution:

$$p(\boldsymbol{u}_i|\boldsymbol{x}_i;\boldsymbol{\theta}) = \frac{1}{(2\pi)^{m/2}|\boldsymbol{B}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})^T|^{1/2}} exp\Bigg(-\frac{1}{2}(\boldsymbol{u}-\boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{\theta})^T\Big(\boldsymbol{B}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})^T\Big)^{-1}(\boldsymbol{u}-\boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{\theta})\Bigg)$$

We then express $log(p(\boldsymbol{u}_i|\boldsymbol{x}_i;\boldsymbol{\theta}))$ as:

$$log(p(\boldsymbol{u}_i|\boldsymbol{x}_i;\boldsymbol{\theta})) = log\Bigg(\frac{1}{(2\pi)^{m/2}|\boldsymbol{B}\boldsymbol{B}^T|^{1/2}}exp\Big(-\frac{1}{2}(\boldsymbol{u}-\boldsymbol{\Phi}\boldsymbol{\theta})^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}(\boldsymbol{u}-\boldsymbol{\Phi}\boldsymbol{\theta})\Big)\Bigg)$$

$$= log\Bigg(\frac{1}{(2\pi)^{m/2}|\boldsymbol{B}\boldsymbol{B}^T|^{1/2}}\Bigg) + log\Bigg(exp\Big(-\frac{1}{2}(\boldsymbol{u}-\boldsymbol{\Phi}\boldsymbol{\theta})^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}(\boldsymbol{u}-\boldsymbol{\Phi}\boldsymbol{\theta})\Big)\Bigg)$$

$$= -log\Big((2\pi)^{m/2}|\boldsymbol{B}\boldsymbol{B}^T|^{1/2}\Big) - \frac{1}{2}(\boldsymbol{u}-\boldsymbol{\Phi}\boldsymbol{\theta})^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}(\boldsymbol{u}-\boldsymbol{\Phi}\boldsymbol{\theta})$$

$$= -log\Big((2\pi)^{m/2}|\boldsymbol{B}\boldsymbol{B}^T|^{1/2}\Big) - \frac{1}{2}(\boldsymbol{u}^T-\boldsymbol{\theta}^T\boldsymbol{\Phi}^T)\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}(\boldsymbol{u}-\boldsymbol{\Phi}\boldsymbol{\theta})$$

$$= -log\Big((2\pi)^{m/2}|\boldsymbol{B}\boldsymbol{B}^T|^{1/2}\Big) + \Bigg(-\frac{1}{2}\boldsymbol{u}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1} + \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\Phi}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\Bigg)(\boldsymbol{u}-\boldsymbol{\Phi}\boldsymbol{\theta})$$

$$= -log\Big((2\pi)^{m/2}|\boldsymbol{B}\boldsymbol{B}^T|^{1/2}\Big) - \frac{1}{2}\boldsymbol{u}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\boldsymbol{u} + \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\Phi}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\boldsymbol{u} + \frac{1}{2}\boldsymbol{u}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\boldsymbol{\Phi}\boldsymbol{\theta}$$

$$-\frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\Phi}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\boldsymbol{\Phi}\boldsymbol{\theta}$$

$$= -log\Big((2\pi)^{m/2}|\boldsymbol{B}\boldsymbol{B}^T|^{1/2}\Big) - \frac{1}{2}\boldsymbol{u}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\boldsymbol{u} + \boldsymbol{\theta}^T\boldsymbol{\Phi}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\boldsymbol{u} - \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\Phi}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\boldsymbol{\Phi}\boldsymbol{\theta}$$

We then compute the gradient of this expression with respect to $\boldsymbol{\theta}$ and substitute our parametrized policy $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{\theta} + \boldsymbol{B}\epsilon_k$:

$$\nabla_{\boldsymbol{\theta}} log(p(\boldsymbol{u}_i|\boldsymbol{x}_i;\boldsymbol{\theta})) = \boldsymbol{\Phi}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\boldsymbol{u} - \boldsymbol{\Phi}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\boldsymbol{\Phi}\boldsymbol{\theta}$$

$$= \boldsymbol{\Phi}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\Bigg(\boldsymbol{\Phi}\boldsymbol{\theta} + \boldsymbol{B}\epsilon_k\Bigg) - \boldsymbol{\Phi}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\boldsymbol{\Phi}\boldsymbol{\theta}$$

$$= \boldsymbol{\Phi}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\boldsymbol{\Phi}\boldsymbol{\theta} + \boldsymbol{\Phi}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\boldsymbol{B}\epsilon_k - \boldsymbol{\Phi}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\boldsymbol{\Phi}\boldsymbol{\theta}$$

$$= \boldsymbol{\Phi}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\boldsymbol{B}\epsilon_k$$

Now given that $\nabla_{\boldsymbol{\theta}} log(p(\boldsymbol{u}_i|\boldsymbol{x}_i;\boldsymbol{\theta})) = \boldsymbol{\Phi}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\boldsymbol{B}\epsilon_k$ we turn can rewrite our gradient policy:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = E_{p(\boldsymbol{\tau})}\Bigg[\sum_{i=0}^{N-1}\nabla_{\boldsymbol{\theta}} log(p(\boldsymbol{u}_i|\boldsymbol{x}_i;\boldsymbol{\theta}))R(\boldsymbol{\tau})\Bigg]$$

$$= E_{p(\boldsymbol{\tau})}\Bigg[R(\boldsymbol{\tau})\sum_{i=0}^{N-1}\boldsymbol{\Phi}^T\Big(\boldsymbol{B}\boldsymbol{B}^T\Big)^{-1}\boldsymbol{B}\epsilon_k\Bigg]$$

If we paramtrize the policy such that $\Phi = \boldsymbol{B}$ then the final form of the REINFORCE Gradient can be written as:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = E_{p(\boldsymbol{\tau})}\left[R(\boldsymbol{\tau})\sum_{i=0}^{N-1}\boldsymbol{B}^T\left(\boldsymbol{B}\boldsymbol{B}^T\right)^{-1}\boldsymbol{B}\epsilon_i\right]$$

For this implementation our policy controller takes the following form where $\Phi = \boldsymbol{B} = \boldsymbol{x}_i$

$$\boldsymbol{u}_i = \boldsymbol{\theta}\boldsymbol{x}_i + \epsilon_i \boldsymbol{x}_i$$

Therefore the gradient takes the following form:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = E_{p(\boldsymbol{\tau})}\left[R(\boldsymbol{\tau})\sum_{k=0}^{N-1}\boldsymbol{x}_i^T\left(\boldsymbol{x}_i\boldsymbol{x}_i^T\right)^{-1}\boldsymbol{x}_i\epsilon_k\right]$$

$$= E_{p(\boldsymbol{\tau})}\left[R(\boldsymbol{\tau})\sum_{i=0}^{N-1}\epsilon_i\right]$$

$$= \frac{1}{M}\sum_{m=0}^{M}\left(\left(\sum_{i=0}^{N-1}\epsilon_{m,i}\right)\left(\sum_{i=0}^{N-1}r(\boldsymbol{x}_{m,i},\boldsymbol{u}_{m,i},i)\right)\right)$$

**2.2**) For our experiment we formulate the same system as in Problem 1 which results in the same optimal LQR gain K = 0.3964.

We formulate the same Reinforcement Learning optimization problem as before and solve it using gradient ascent and REINFORCE policy gradient estimation.

This results in an optimal $\boldsymbol{\theta}^\star = 0.3968$. Our convergence criterion fulfills that the gradient is sufficiently small for some $\epsilon$ or that our Reward begins to decrease after increasing for a certain number of times.

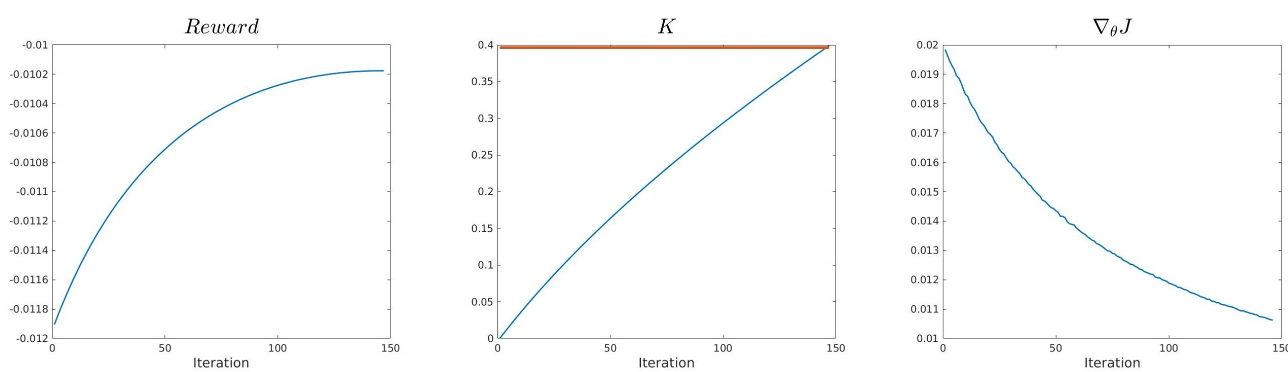


Figure 2: Results using REINFORCE policy gradient estimation.