

AE 4803 Robotics and Autonomy
Professor Evangelos Theodorou
Homework 3

Luis Pimentel Jackson Crandell
lpimentel3@gatech.edu 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:

$$\underset{\theta}{\text{maximize}} \quad \mathbb{E}[R(\tau)]$$

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

$$r(\mathbf{x}_t, \mathbf{u}_t, t) = -\mathbf{x}^T \mathbf{Q} \mathbf{x} - \mathbf{u}^T \mathbf{R} \mathbf{u}$$

$$\mathbf{u} = -\theta \mathbf{x}$$

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

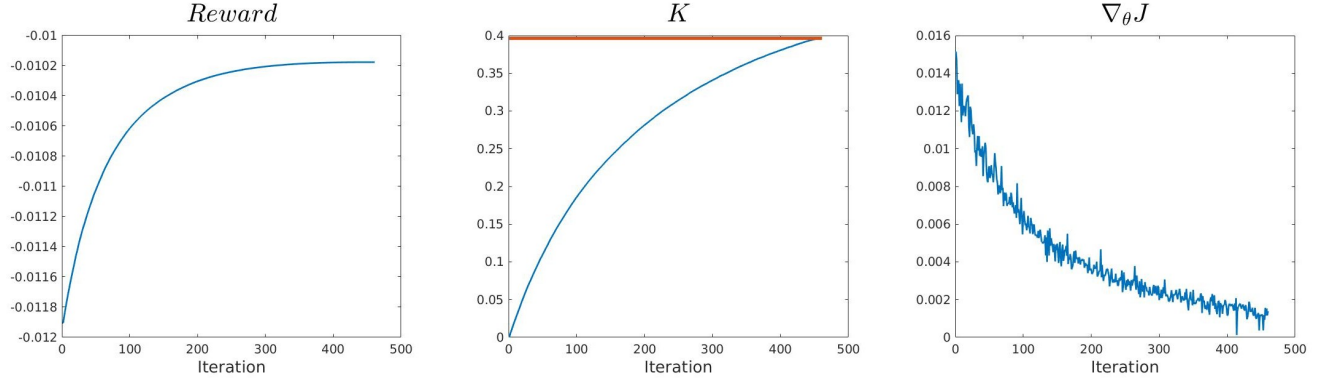


Figure 1: Results using Finite Differencing policy gradient estimation.

Problem 2.

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

$$J(\theta) = \int p(\tau) R(\tau) d\tau$$

A trajectory can be expressed as $\tau = (\mathbf{x}_0, \mathbf{u}_0, \dots, \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N)$ with states $\mathbf{x} \in \mathbb{R}^\ell$ and controls $\mathbf{u} \in \mathbb{R}^p$ over the time horizon $T = Nd$. $R(\tau)$ is the accumulated cost over a trajectory and $p(\tau)$ represents the path probability of the trajectory, which using Bayesian and Markov properties can be expressed as:

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

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

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

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

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

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:

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

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

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

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

From this we rewrite our policy gradient as

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

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

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

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

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

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

$$\begin{aligned}
\log(p(\mathbf{u}_i|\mathbf{x}_i;\boldsymbol{\theta})) &= \log\left(\frac{1}{(2\pi)^{m/2}|\mathbf{B}\mathbf{B}^T|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{u}-\boldsymbol{\Phi}\boldsymbol{\theta})^T(\mathbf{B}\mathbf{B}^T)^{-1}(\mathbf{u}-\boldsymbol{\Phi}\boldsymbol{\theta})\right)\right) \\
&= \log\left(\frac{1}{(2\pi)^{m/2}|\mathbf{B}\mathbf{B}^T|^{1/2}}\right) + \log\left(\exp\left(-\frac{1}{2}(\mathbf{u}-\boldsymbol{\Phi}\boldsymbol{\theta})^T(\mathbf{B}\mathbf{B}^T)^{-1}(\mathbf{u}-\boldsymbol{\Phi}\boldsymbol{\theta})\right)\right) \\
&= -\log\left((2\pi)^{m/2}|\mathbf{B}\mathbf{B}^T|^{1/2}\right) - \frac{1}{2}(\mathbf{u}-\boldsymbol{\Phi}\boldsymbol{\theta})^T(\mathbf{B}\mathbf{B}^T)^{-1}(\mathbf{u}-\boldsymbol{\Phi}\boldsymbol{\theta}) \\
&= -\log\left((2\pi)^{m/2}|\mathbf{B}\mathbf{B}^T|^{1/2}\right) - \frac{1}{2}(\mathbf{u}^T - \boldsymbol{\theta}^T\boldsymbol{\Phi}^T)(\mathbf{B}\mathbf{B}^T)^{-1}(\mathbf{u}-\boldsymbol{\Phi}\boldsymbol{\theta}) \\
&= -\log\left((2\pi)^{m/2}|\mathbf{B}\mathbf{B}^T|^{1/2}\right) + \left(-\frac{1}{2}\mathbf{u}^T(\mathbf{B}\mathbf{B}^T)^{-1} + \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\Phi}^T(\mathbf{B}\mathbf{B}^T)^{-1}\right)(\mathbf{u}-\boldsymbol{\Phi}\boldsymbol{\theta}) \\
&= -\log\left((2\pi)^{m/2}|\mathbf{B}\mathbf{B}^T|^{1/2}\right) - \frac{1}{2}\mathbf{u}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{u} + \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\Phi}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{u} + \frac{1}{2}\mathbf{u}^T(\mathbf{B}\mathbf{B}^T)^{-1}\boldsymbol{\Phi}\boldsymbol{\theta} \\
&\quad - \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\Phi}^T(\mathbf{B}\mathbf{B}^T)^{-1}\boldsymbol{\Phi}\boldsymbol{\theta} \\
&= -\log\left((2\pi)^{m/2}|\mathbf{B}\mathbf{B}^T|^{1/2}\right) - \frac{1}{2}\mathbf{u}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{u} + \boldsymbol{\theta}^T\boldsymbol{\Phi}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{u} - \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\Phi}^T(\mathbf{B}\mathbf{B}^T)^{-1}\boldsymbol{\Phi}\boldsymbol{\theta}
\end{aligned}$$

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

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}}\log(p(\mathbf{u}_i|\mathbf{x}_i;\boldsymbol{\theta})) &= \boldsymbol{\Phi}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{u} - \boldsymbol{\Phi}^T(\mathbf{B}\mathbf{B}^T)^{-1}\boldsymbol{\Phi}\boldsymbol{\theta} \\
&= \boldsymbol{\Phi}^T(\mathbf{B}\mathbf{B}^T)^{-1}\left(\boldsymbol{\Phi}\boldsymbol{\theta} + \mathbf{B}\epsilon_k\right) - \boldsymbol{\Phi}^T(\mathbf{B}\mathbf{B}^T)^{-1}\boldsymbol{\Phi}\boldsymbol{\theta} \\
&= \boldsymbol{\Phi}^T(\mathbf{B}\mathbf{B}^T)^{-1}\boldsymbol{\Phi}\boldsymbol{\theta} + \boldsymbol{\Phi}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{B}\epsilon_k - \boldsymbol{\Phi}^T(\mathbf{B}\mathbf{B}^T)^{-1}\boldsymbol{\Phi}\boldsymbol{\theta} \\
&= \boldsymbol{\Phi}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{B}\epsilon_k
\end{aligned}$$

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

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}}J(\boldsymbol{\theta}) &= E_{p(\boldsymbol{\tau})}\left[\sum_{i=0}^{N-1}\nabla_{\boldsymbol{\theta}}\log(p(\mathbf{u}_i|\mathbf{x}_i;\boldsymbol{\theta}))R(\boldsymbol{\tau})\right] \\
&= E_{p(\boldsymbol{\tau})}\left[R(\boldsymbol{\tau})\sum_{i=0}^{N-1}\boldsymbol{\Phi}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{B}\epsilon_k\right]
\end{aligned}$$

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

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

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

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

Therefore the gradient takes the following form:

$$\begin{aligned} \nabla_{\theta} J(\theta) &= E_{p(\tau)} \left[R(\tau) \sum_{k=0}^{N-1} \mathbf{x}_i^T (\mathbf{x}_i \mathbf{x}_i^T)^{-1} \mathbf{x}_i \epsilon_k \right] \\ &= E_{p(\tau)} \left[R(\tau) \sum_{i=0}^{N-1} \epsilon_i \right] \\ &= \frac{1}{M} \sum_{m=0}^M \left(\left(\sum_{i=0}^{N-1} \epsilon_{m,i} \right) \left(\sum_{i=0}^{N-1} r(\mathbf{x}_{m,i}, \mathbf{u}_{m,i}, i) \right) \right) \end{aligned}$$

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 $\theta^* = 0.3968$. Our convergence criterion fulfills that the gradient is sufficiently small for some ϵ or that our Reward begins to decrease after increasing for a certain number of times.

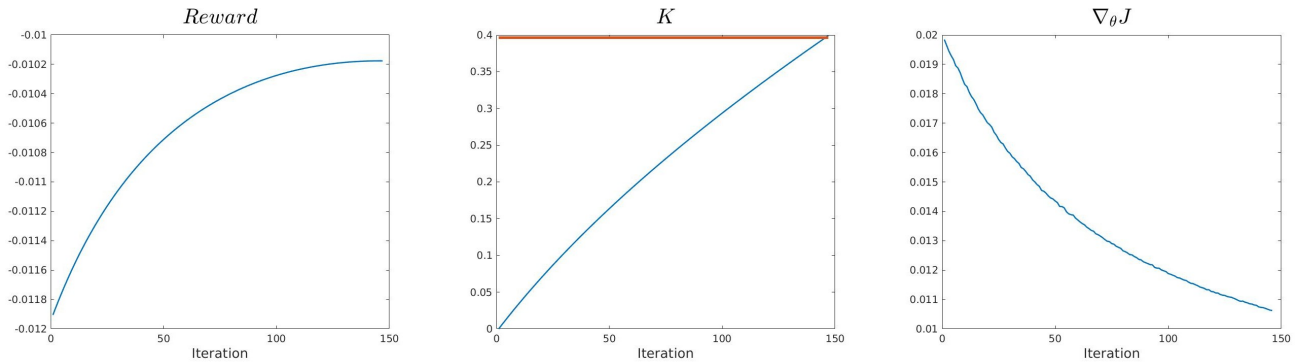


Figure 2: Results using REINFORCE policy gradient estimation.