Evaluation of Learning Convex Optimization Control Policies

Shashi Kanth Koppala
Department of EECS
Oregon State University
Corvallis, OR, 57331
koppalas@oregonstate.edu

Abstract

Control policies are defined by parameters that govern the state inputs of a closed-loop control problem. Tuning these parameters is typically done through methods like grid search or other simpler optimization techniques. For convex-optimized control policies (COCP), the learnable policy algorithm leverages first-order differential methods to estimate these parameters more efficiently. This paper explores this approach and implements it across various scenarios, including the novel case of the Mars soft landing problem, among others.

1 Introduction

The control of a dynamic system is governed by its state and the corresponding input. The system's behavior changes as we modify the input. When this input is determined using a convex optimization problem, aimed at achieving an optimal or reasonable output, it is referred to as Convex Optimization Control Policies (COCPs). A policy is essentially a mapping from a state space to a specific value. To obtain an optimal policy, the parameters are usually tuned manually, through grid search, or based on past simulations.

Agrawal et al. (1) proposed a new automated method for tuning these parameters. Their approach involves simulating the system with the policy in the loop, computing a stochastic gradient of the expected performance with respect to the parameters, and updating them using a projected stochastic gradient method. This approach takes advantage of the fact that the solution map for convex programs is often differentiable, and its derivative can be efficiently computed. Although this method doesn't guarantee global optimality, the authors argue that, in real-world applications, initializing the parameters with a reasonable guess and using this method significantly improves performance

2 Problem Formulation

2.1 Background

Control System dynamics are given by the following equation:

$$x_{t+1} = f(x_t, u_t, w_t), \quad t = 0, 1, 2, \dots \tag{1}$$

where, x_t is the current state, u_t is the input given to the system at time t,and w_t is the noise at time t. The input at time u_t is calculated using the policy function ϕ:

$$u_t = \phi(x_t), \tag{2}$$

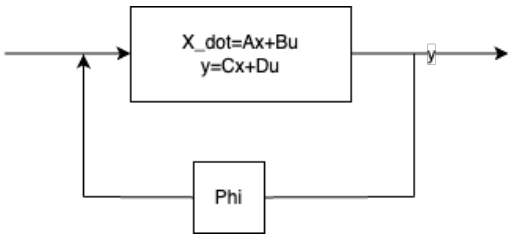

Figure 1: depicts a general control theory problem. ϕ is the control policy which we will learn by tuning the parameters

2.2 Authors Contribution

Agrawal et al. (1) consider COCPs of the form:

$$\phi(x) = \arg\min_{u} \; f_0(x, u; \theta), \tag{3}$$

subject to the constraints:

$$f_i(x, u; \theta) \leq 0, \quad i = 1, \ldots, k,$$
$$g_i(x, u; \theta) = 0, \quad i = 1, \ldots, \ell,$$

where f_i is convex in u and g_i is affine in u. The problem addressed by the authors is the choice of parameter θ, which specifies a particular problem instance.

The authors define the metric to evaluate a policy as the average value of a cost over trajectories of length T.

$$X = (x_0, x_1, \ldots, x_T) \in \mathbb{R}^N, \quad U = (u_0, u_1, \ldots, u_T) \in \mathbb{R}^M, \quad W = (w_0, w_1, \ldots, w_T) \in \mathbb{W}^{T+1}$$

are defined as the state, input, and disturbance trajectories t.

The cost is provided by a function

$$\psi(X, U, W) = \frac{1}{T+1} \sum_{t=0}^{T} g(x_t, u_t, w_t), \tag{4}$$

where $\psi : \mathbb{R}^N \times \mathbb{R}^M \times W^{T+1} \to \mathbb{R} \cup \{+\infty\}$, where $g : \mathbb{R}^n \times \mathbb{R}^m \times W \to \mathbb{R} \cup \{+\infty\}$ is a stage cost function.

The policy is evaluated on the expected value of the cost, which is defined as:

$$J(\theta) = \mathbb{E}[\psi(X, U, W)],$$

But as solving for J is not possible , they take the trajectory over K to come up with a Monte Carlo approximation.

$$\hat{J}(\theta) = \frac{1}{K} \sum_{i=1}^{K} \psi(X_i, U_i, W_i). \tag{5}$$

One interesting thing to observe is as K is large, we can get a better approximation as the variance of the approximation goes towards zero.

Therefore the final Convex Optimzation problem which the authors used to evaluate their algorithm is

$$\min_{\theta} J(\theta) \quad \text{subject to.} \quad \theta \in \Theta \tag{6}$$

3 Solution

The proposed solution takes an approximation approach, as solving equation 6 directly is challenging. Agrawal et al. (1) utilize projected stochastic gradient descent, in contrast to existing derivative-free methods like evolutionary algorithms or random search, which are typically slow to converge.

The authors first initialize the parameter θ_0 and iterative compute the cost $\hat{J}(\theta_k)$ subject to the update rule as

$$\theta_{k+1} = \Pi_\Theta \left(\theta_k - \alpha_k g_k\right),$$

where Π_Θ denotes the Euclidean projection onto the feasible set Θ, and α_k is the step size.

For applying a derivative-based approach, differentiability is essential, and this is an assumption made by the authors. In cases where the functions are non-differentiable, they suggest using heuristic approximations, much like how neural networks are trained using stochastic gradient descent despite being non-differentiable in some instances.

4 Examples

I have implemented the algorithm proposed by the authors in three different scenarios. Each subsection of this section presents a different scenario. The results obtained from these scenarios are discussed in the "Results" section of this report.

4.1 Linear Quadratic Regulator (LQR)

The first scenario is that of LQR (2). In LQR the constraints are affine and the cost fucntion is Quadratic in nature hence the name Linear Quadratic Regulator.

$$f(x,u,w) = Ax + Bu + w, \quad \psi(X,U,W) = \frac{1}{T+1}\sum_{t=0}^{T}\left(x_t^T Q x_t + u_t^T R u_t\right), \tag{7}$$

$$\text{where} \quad A \in \mathbb{R}^{n\times n},\, B \in \mathbb{R}^{n\times m},\, Q \in \mathbb{S}_n^+,\, R \in \mathbb{S}_m^{++}, \quad w \sim \mathcal{N}(0,\Sigma). \tag{8}$$

$$\text{The COCP is defined as} \quad \phi(x) = \arg\min_u \left(u^T R u + \|\theta(Ax+Bu)\|_2^2\right). \tag{9}$$

The matrix Q penalizes the state error, whereas R penalizes the input error. The parameter θ is to be found out and if it satsisfies a certain condition of the algebraic Riccati Equation, the solution found will be optimal.

My Experiment I have taken a 2-D novel experiment (Toy Problem) where an UFO is to be positioned such that its alignment is appropriate as it is initially upside down. We consider the angle and angular velocity to be the states of the UFO and penalize the angular error, angular rate Q and thruster effort R .

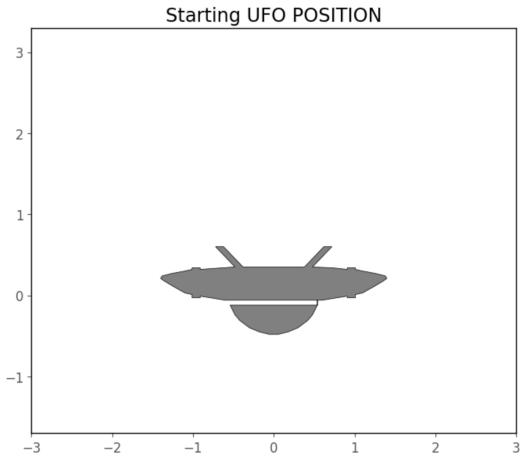

Figure 2: UFO Initial Position (Upside Down)

4.2 Vehicle Controller on Curved Path

As mentioned by the Agrawal et al. (1) I considered vehicle moving relative to a smooth curved path, with state and input

$$\mathbf{x}_t = (e_t, \Delta\psi_t, v_t, v_{des_t}, \kappa_t)\,, \quad \mathbf{u}_t = (a_t, z_t)\,.$$

At time t, e_t is the lateral path deviation (m), $\Delta\psi_t$ is the heading deviation from the path (rad), v_t is the velocity (m/s), v_{des_t} is the desired velocity (m/s), κ_t is the current curvature (i.e., inverse radius) of the path (1/m), a_t is the acceleration (m/s^2), and

$$z_t := \tan(\delta_t) - L\kappa_t,$$

where δ_t is the wheel angle (rad) and L is the vehicle's wheelbase (m).

The COCP hat computes (a_t, z_t) as

$$\phi(\mathbf{x}_t) = \arg\min_{a,z} \lambda_3|a| + \lambda_4 z^2 + \|Sy\|_2^2 + q^T y$$

subject to

$$y = \begin{bmatrix} e_t + hv_t \sin(\Delta\psi_t) \\ \Delta\psi_t + hv_t\left(\kappa_t + \frac{z}{L} - \frac{\kappa_t}{1-e_t\kappa_t}\cos(\Delta\psi_t)\right) \\ v_t + ha - (0.98)v_{des_t} - (0.02)4.5 \\ y_1 + hv_t \sin(y_2 - hv_t\frac{z}{L}) + h\frac{1}{2}v_t^2 \end{bmatrix}$$

$$|a| \le a_{max}, \quad |z + L\kappa_t| \le \tan(\delta_{max}),$$

where h is the time step, λ_3 and λ_4 are weights, and the terms y_1, y_2 correspond to elements of the state vector. This implementation is that of Approximate Dynamic Programming, which replaces the value function of standard Dynamic Programming with an approximation. The approximation here is given as $\|Sx\|_2^2 + q^T x$.

Requesting to go through the original work of (3) or the vehicle controller subsection of Agrawal et al. (1) for a thorough clarification on the equations, the intializations as I have not mentioned it due to space constraints.

My Experiment I have considered a hypothetical case (Toy Problem) of a Delivery Robot delivering food from a dining center to a desired location . Since this is a ADP (Approximate Dynamic Programming) problem a closed form solution is absent. Hence I compare the tuned policy with that of an untuned one

4.3 Mars Soft Landing Control Problem

The Mars Soft Landing Control Problem is a novel application of the policy Algorithm discussed in this report. This is an interesting problem where the objective is to land a spacecraft at a specific position on Mars. Açkmeşe et al. (4) who first introduced this problem formulation referred to it as a "soft landing. The objective is to minimize landing error (the distance from the target landing point) while simultaneously minimizing fuel consumption.

Unfortunately, the problem is non-convex in nature, but Açkmeşe et al. (4) devised a lossless convexification of the original problem. The COCP devised is:

$$\min_{\theta} J(\theta) = \theta_1 \left|Ex_{N-1} - q\right|_2 + \theta_2 \sum_{t=0}^{N-1} \gamma_t \Delta t + \theta_3 \sum_{t=0}^{N-1} |u_t|_2 \quad \text{subject to } \theta \in \Theta$$

where:

$$\theta \in \Theta = \left\{\theta \in \mathbb{R}_+^3 : \theta_1 + \theta_2 + \theta_3 = 1\right\}$$

Here:

- θ_1 weights the landing error term.

- θ_2 weights the fuel consumption term.
- θ_3 weights the thrust magnitude term.

The variables are as follows.

- $Ex_{N-1} - q$ is the final landing error.
- γ_t relates to fuel consumption at time t.
- u_t is the thrust control input at time t.

The original paper is very detailed and descriptive. Due to space constraints, I have omitted the initial formulation and the reasoning behind these equations in this report.

My Experiment I have taken a toy problem where the spacecraft has to land at the origin in mars. The data and variable inputs are similar to the one proposed in the original paper.

5 Results

The results are split into 3 subsections, where each subsection corresponds to a different experiment result as mentioned in the previous section.

5.1 UFO-LQR

As we can see in Figure 3, the proposed COCP Algorithm performs well with respect to LQR in this scenario, as it almost all converges to the optimal value in just 15-20 iterations. I have used a trajectory of K=6 over 50 iterations. A minute error of 0.01-0.02 is present.Below is the image which depicts the comparison of LQR and the proposed algorithm.

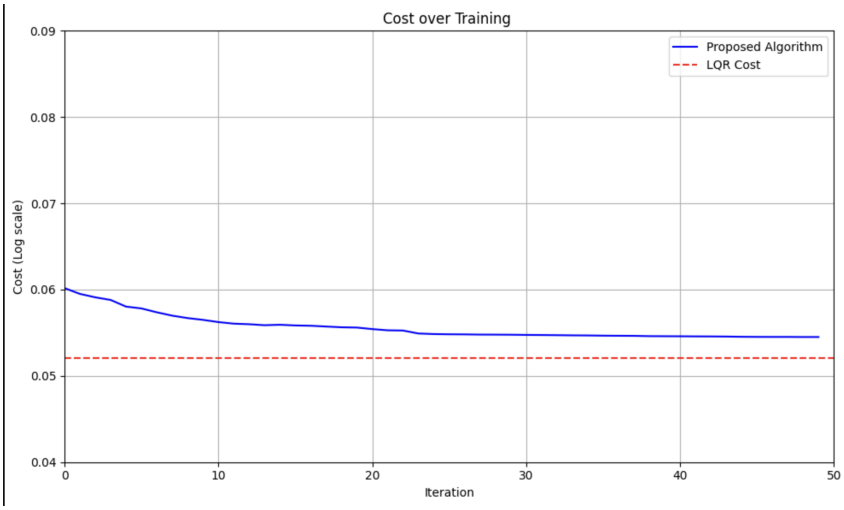

Figure 3: Convergence of the proposed Algorithm

5.2 Delivery Robot Vehicular Controller

The Proposed COCP Algorithm fares well as compared to an untuned policy. Figure 4 clearly shows the path followed by the tuned proposed policy as opposed to the untuned one.

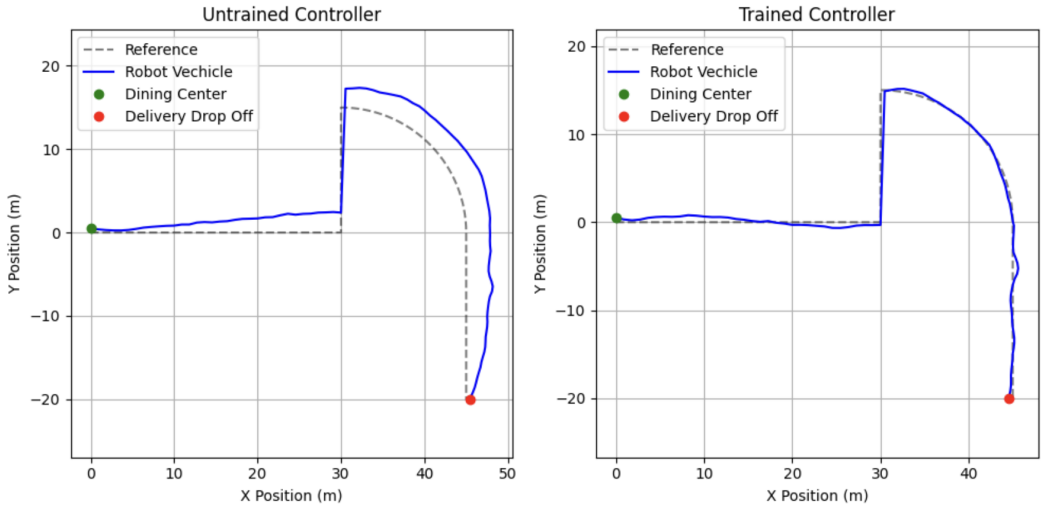

Figure 4: Convergence of the proposed Algorithm

As we can see, the tuned COCP policy does a better job and is almost on its original track with minimal loss.

5.3 Mars Soft Landing

The proposed algorithm performs reasonably well when applied to this novel use case. It successfully reaches the target, and while the resulting 3D path is slightly different from those obtained by the authors, it still closely approximates the desired trajectory.

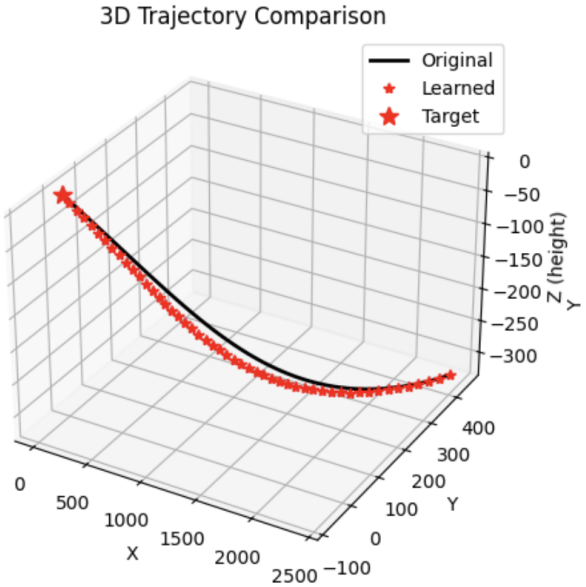

Figure 5: Comparision of Learning COCP on Soft Landing in Mars

6 Conclusion

I chose to read this paper because I found the application of convex optimization in control theory particularly intriguing. In conclusion, the use of a learnable Convex Optimization Control Policy (COCP) presents a promising approach. Although it does not guarantee an optimal solution, it offers a significant advantage in terms of speed and practicality. Unlike derivative-free methods, the learnable COCP benefits from its differentiability, making it an effective heuristic for many real-world applications.

References

[1] Agrawal, A., Barratt, S., Boyd, S., & Stellato, B. (2020, July 31). Learning Convex Optimization Control Policies. *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*. PMLR.

[2] Kalman, R. E. (1960). Contributions to the theory of optimal control. *Boletín de la Sociedad Matemática Mexicana*, 5(2), 102–119.

[3] Christian Gerdes. ME 227 vehicle dynamics and control course notes, 2019. Lectures 1 and 2.

[4] Açıkmeşe, B., Carson III, J. M., & Blackmore, L. (2013). Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem. *IEEE Transactions on Control Systems Technology*, 21(6), 2110–2118.