Learning Convex Optimization Control Policies

Akshay Agrawal* **Shane Barratt*** Stephen Boyd* Bartolomeo Stellato†[1]

*Stanford University †MIT/Princeton University

[1] Alphabetical order.

Convex optimization control policies

Dynamics

- Known dynamical system

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

- $t = 0, 1, \ldots$ is time period
- $x_t \in \mathbf{R}^n$ is state
- $u_t \in \mathbf{R}^m$ is input or action
- $w_t \in \mathcal{W}$ is the (random) disturbance
- $f : \mathbf{R}^n \times \mathbf{R}^m \times \mathcal{W} \to \mathbf{R}^n$ is state transition function

Convex optimization control policy

- Convex optimization control policy (COCP):

$$\begin{array}{rcll} \phi(x) & = & \underset{u}{\text{argmin}} & f_0(x, u; \theta) \\ & & \text{subject to} & f_i(x, u; \theta) \leq 0, \quad i = 1, \ldots, k \\ & & & g_i(x, u; \theta) = 0, \quad i = 1, \ldots, \ell \end{array}$$

- f_i are convex in u and g_i are affine in u
- $\theta \in \Theta \subseteq \mathbf{R}^p$ are parameters
- *e.g.*: LQR, ADP, MPC

Judging a COCP

- Consider length-T trajectories

$$\begin{aligned} X &= (x_0, x_1, \ldots, x_T) \in \mathbf{R}^{(T+1)n} \\ U &= (u_0, u_1, \ldots, u_{T-1}) \in \mathbf{R}^{Tm} \\ W &= (w_0, w_1, \ldots, w_{T-1}) \in \mathcal{W}^T \end{aligned}$$

- Judge control policy by average of cost $\psi : \mathbf{R}^{(T+1)n} \times \mathbf{R}^{Tm} \times \mathcal{W}^T \to \mathbf{R}$:

$$J(\theta) = \mathbf{E}\, \psi(X, U, W)$$

Examples of COCPs

Dynamic programming policy

- Time-separable cost:

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

- Optimal policy as $T \to \infty$ is

$$\phi(x) = \text{argmin}_u \, \mathbf{E} \left(g(x, u, w) + V(f(x, u, w)) \right)$$

- $V : \mathbf{R}^n \to \mathbf{R}$ is cost-to-go function
- COCP when f is affine in x and u and g is convex in x and u

Approximate dynamic programming policy

- Replace cost-to-go V with approximate cost-to-go $\hat{V}$
- ADP policy has the form

$$\phi(x) = \mathrm{argmin}_u \, \mathbf{E} \left(g(x, u, w) + \hat{V}(f(x, u, w)) \right)$$

- This is a COCP when g is convex in u, f is affine in u, and $\hat{V}$ is convex

Learning method

Controller tuning problem

- Controller tuning problem

$$\begin{array}{ll} \text{minimize} & J(\theta) \\ \text{subject to} & \theta \in \Theta \end{array}$$

- Nonconvex and difficult to solve exactly
- Possible to use derivative-free methods, but slow

A gradient-based method

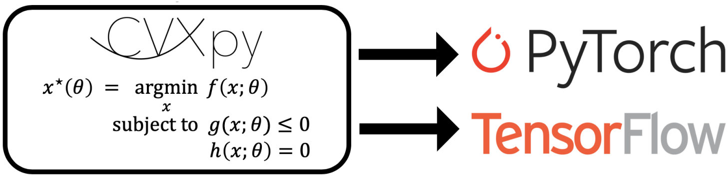
- COCP often differentiable in x and θ [ABB+19; Amo19]
- If cost and dynamics differentiable, can compute $\nabla_\theta J(\theta)$
- Use projected gradient method

$$\theta^{k+1} = \Pi_\Theta(\theta^k - \alpha^k g^k), \quad k = 0, \ldots, n_{\mathrm{iter}}$$

- g^k is stochastic gradient of $J(\theta)$, computed through Monte Carlo
- α^k is step size
- When COCP non-differentiable, often still get descent direction

Implementation

- CVXPY layers package[2] to define COCPs [AAB+19]



- PyTorch for the chain rule

[2] www.github.com/cvxgrp/cvxpylayers

Numerical examples

Box-constrained LQR

- Dynamics

$$x_{t+1} = Ax_t + Bu_t + w_t$$

  w_t is Gaussian

- Cost

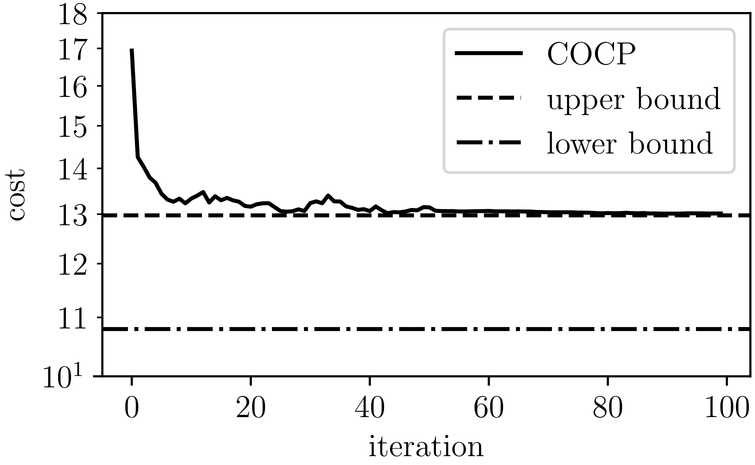
$$\psi(X, U, W) = \begin{cases} \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t + x_T^T Q x_T & \|u_t\|_\infty \leq u_{\max} \\ +\infty & \text{otherwise} \end{cases}$$

- ADP policy

$$\begin{array}{rcll} \phi(x) & = & \underset{u}{\text{argmin}} & u^T R u + \|\theta(Ax + Bu)\|_2^2 \\ & & \text{subject to} & \|u\|_\infty \leq u_{\max}. \end{array}$$

- Compare to LMI-based upper- and lower-bound [WB09]

Box-constrained LQR



Supply chain

- single-good supply chain over n nodes
- $x_t = (h_t, p_t, d_t)$; $h_t \in \mathbf{R}^n$ is quantity held, $p_t \in \mathbf{R}^k$ is supplier price, $d_t \in \mathbf{R}^c$ is consumer demand
- $u_t = (b_t, s_t, z_t)$; $z_t \in \mathbf{R}^{m-k-c}$ is quantity shipped, $b_t \in \mathbf{R}^k$ is quantity bought, $s_t \in \mathbf{R}^c$ is quantity sold
- $r \in \mathbf{R}^c$ is consumer price

Supply chain

- Dynamics

$$h_{t+1} = h_t + (A^{\mathrm{in}} - A^{\mathrm{out}})u_t$$

- $A_{ij}^{\mathrm{in(out)}}$ is 1 if link j enters (exist) node i and 0 otherwise
- p_{t+1} and d_{t+1} are IID log-normal
- Cost:

$$\psi(X, U, W) = \frac{1}{T}\sum_{t=0}^{T-1} p_t^T b_t - r^T s_t + \tau^T z_t + \alpha^T h_t + \beta^T h_t^2 + I(x_t, u_t)$$

From left to right: payment to suppliers, sale revenues, shipment cost, storage cost, constraints

- Constraints are

$$0 \le u_t \le u_{\max}, \quad 0 \le h_t \le h^{\max}, \quad A^{\mathrm{out}}u_t \le h_t, \quad s \le d_t$$

Supply chain

- COCP

$$
\begin{array}{rcll}
\phi(h_t, p_t, d_t) & = & \underset{b,s,z}{\text{argmin}} & p_t^T b - r^T s + \tau^T z + \|Sh^+\|_2^2 + q^T h^+ \\
 & & \text{subject to} & h^+ = h_t + (A^{\mathrm{in}} - A^{\mathrm{out}})(b, s, z) \\
 & & & 0 \leq h^+ \leq h_{\mathrm{max}}, \quad 0 \leq (b, s, z) \leq u_{\mathrm{max}}, \\
 & & & A^{\mathrm{out}}(b, s, z) \leq h_t, \quad s \leq d_t
\end{array}
$$

Parameters S and q

Supply chain

Simulated example with 4 nodes, 4 links, 2 supply links, 2 consumer links

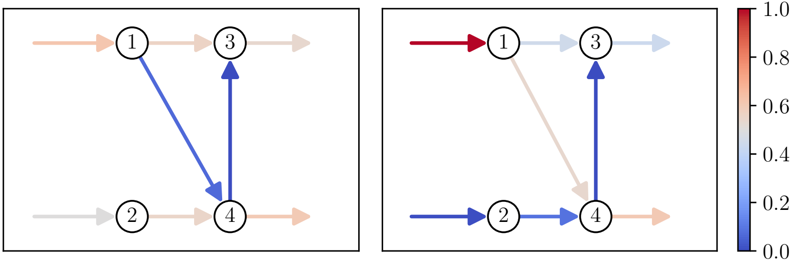


Figure: Normalized shipments (0-1). Left: untrained. Right: trained.

Summary

- Can learn COCPs efficiently w/ gradient descent
- Easy to enforce constraints; hard with neural networks
- Applications to vehicle control and finance in our paper

Learning Convex Optimization Control Policies

Software:

- https://github.com/cvxgrp/cvxpylayers
- https://github.com/cvxgrp/cocp

References:

[AAB+19] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and Z. Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems*. 2019.

[ABB+19] A. Agrawal, S. Barratt, S. Boyd, E. Busseti, and W. Moursi. Differentiating through a cone program. *Journal of Applied and Numerical Optimization* 1.2 (2019), pp. 107–115.

[Amo19] B. Amos. Differentiable optimization-based modeling for machine learning. PhD thesis. Carnegie Mellon University, 2019.

[WB09] Y. Wang and S. Boyd. Performance bounds for linear stochastic control. *Systems & Control Letters* 58.3 (2009), pp. 178–182.