Learning Convex Optimization Control Policies

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

*Stanford University  †MIT/Princeton University

¹Alphabetical 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
- \( x_t \in \mathbb{R}^n \) is state
- \( u_t \in \mathbb{R}^m \) is input or action
- \( w_t \in \mathcal{W} \) is the (random) disturbance
- \( f : \mathbb{R}^n \times \mathbb{R}^m \times \mathcal{W} \to \mathbb{R}^n \) is state transition function
Convex optimization control policy

Convex optimization control policy (COCP):

\[
\phi(x) = \arg\min_u f_0(x, u; \theta)
\]
subject to

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

- \(f_i\) are convex in \(u\) and \(g_i\) are affine in \(u\)
- \(\theta \in \Theta \subseteq \mathbb{R}^p\) are parameters
- e.g.: LQR, ADP, MPC
Judging a COCP

- Consider length-$T$ trajectories

\[
X = (x_0, x_1, \ldots, x_T) \in \mathbb{R}^{(T+1)n}
\]

\[
U = (u_0, u_1, \ldots, u_{T-1}) \in \mathbb{R}^{Tm}
\]

\[
W = (w_0, w_1, \ldots, w_{T-1}) \in \mathcal{W}^T
\]

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

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

\[ \phi(x) = \arg\min_u E (g(x, u, w) + V(f(x, u, w))) \]

- \( V : \mathbb{R}^n \to \mathbb{R} \) 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

$$\phi(x) = \arg\min_u \mathbb{E} \left( g(x, u, w) + \hat{V}(f(x, u, w)) \right)$$

- 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

\[ \begin{align*}
\text{minimize} & \quad J(\theta) \\
\text{subject to} & \quad \theta \in \Theta
\end{align*} \]

▶ Nonconvex and difficult to solve exactly
▶ Possible to use derivative-free methods, but slow
A gradient-based method

- COCP often differentiable in $x$ and $\theta$ [ABB$^+$19; Amo19]
- If cost and dynamics differentiable, can compute $\nabla_\theta J(\theta)$
- Use projected gradient method
  $$\theta^{k+1} = \Pi_\Theta(\theta^k - \alpha^k g^k), \quad k = 0, \ldots, n_{\text{iter}}$$
  - $g^k$ is stochastic gradient of $J(\theta)$, computed through Monte Carlo
  - $\alpha^k$ is step size
- When COCP non-differentiable, often still get descent direction
Implementation

- CVXPY layers package\(^2\) to define COCPs [AAB\(^+\)19]

\[ x^*(\theta) = \arg\min_x f(x; \theta) \]
subject to \( g(x; \theta) \leq 0 \)
\( h(x; \theta) = 0 \)

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

- **ADP policy**
  \[ \phi(x) = \arg\min_u u^T R u + \|\theta(Ax + Bu)\|_2^2 \]
  subject to \( \|u\|_\infty \leq u_{\text{max}} \).

- **Compare to LMI-based upper- and lower-bound** [WB09]
Box-constrained LQR

![Graph showing the convergence of cost over iterations with COCP and upper/lower bounds.](image)
single-good supply chain over \( n \) nodes
\[
\begin{align*}
\mathbf{x}_t &= (h_t, p_t, d_t); \ h_t \in \mathbb{R}^n \text{ is quantity held, } p_t \in \mathbb{R}^{k} \text{ is supplier price, } d_t \in \mathbb{R}^{c} \text{ is consumer demand} \\
\mathbf{u}_t &= (b_t, s_t, z_t); \ z_t \in \mathbb{R}^{m-k-c} \text{ is quantity shipped, } b_t \in \mathbb{R}^{k} \text{ is quantity bought, } s_t \in \mathbb{R}^{c} \text{ is quantity sold} \\
r &\in \mathbb{R}^{c} \text{ is consumer price}
\end{align*}
\]
Supply chain

- Dynamics
  \[ h_{t+1} = h_t + (A^{\text{in}} - A^{\text{out}})u_t \]

- \( A_{ij}^{\text{in(out)}} \) 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
  \[
  0 \leq u_t \leq u_{\text{max}}, \quad 0 \leq h_t \leq h_{\text{max}}, \quad A^{\text{out}} u_t \leq h_t, \quad s \leq d_t
  \]
Supply chain

► COCP

\[
\phi(h_t, p_t, d_t) = \arg\min_{b, s, z} \quad p_t^T b - r^T s + \tau^T z + \|Sh^+\|_2^2 + q^T h^+
\]
subject to

\[
h^+ = h_t + (A^{in} - A^{out})(b, s, z)
\]
\[
0 \leq h^+ \leq h_{\text{max}}, \quad 0 \leq (b, s, z) \leq u_{\text{max}},
\]
\[
A^{out}(b, s, z) \leq h_t, \quad s \leq d_t
\]

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:


