UNIVERSITY of WASHINGTON

Iterative Linearized Control: Stable Algorithms and Complexity Guarantees

Vincent Roulet, Siddhartha Srinavasa, Dmitry Drusviatskiy, Zaid Harchaoui

University of Washington

Overview

Algorithms for nonlinear discrete time control are heuristics

- We provide **regularized** and **accelerated** variants of those heuristics with proven convergence to a stationary point
- We characterize optimization oracles as **dynamic programming** procedures and present implementations with **automatic-differentiation**

Nonlinear control

Discrete time nonlinear control problem with finite horizon τ

$$\min_{\substack{x_0,\ldots,x_\tau\\u_0,\ldots,u_{\tau-1}}} \sum_{t=1}^{\tau} h_t(x_t) + \sum_{t=0}^{\tau-1} g_t(u_t) \qquad \text{(Cost)}$$

$$\text{s.t.} \quad x_{t+1} = \phi_t(x_t, u_t) \qquad x_0 = \hat{x}_0 \qquad \text{(Dyn)}$$

- state $x_t \in \mathbb{R}^d$, control $u_t \in \mathbb{R}^p$
- h_t convex state cost, e.g., $h_t(x_t) = \frac{1}{2}(x_t - \hat{x}_t)^\top H_t(x_t - \hat{x}_t)$
- g_t convex control penalty, e.g., $g_t(u_t) = \frac{1}{2}u_t^\top G_t u_t$
- ϕ_t nonlinear dynamic given by physics laws, e.g., pendulum movement

Iterative Linear Quadratic Regulator

1. Given current iterate u_t with x_t given by (Dyn), solve by dynamic programming

$$\min_{\substack{y_0,\ldots,y_\tau\\v_0,\ldots,v_{\tau-1}}} \sum_{t=1}^{\tau} q_{h_t}(x_t + y_t) + \sum_{t=0}^{\tau-1} q_{g_t}(u_t + v_t) \qquad \text{(LQR)}$$

$$\text{s.t.} \quad y_{t+1} = \ell_{\phi_t}(y_t, v_t) \qquad y_0 = 0$$

- q_{h_t} quad. approx. of h_t around x_t
- q_{g_t} quad. approx. of g_t around u_t
- ℓ_{ϕ_t} linear approx. of ϕ_t around (x_t, u_t)

2. Get next iterate $u_t^+ = u_t + \alpha v_t^*$ by line-search

Optimization of compositions

Trajectory $\bar{x}(\bar{u}) = (\bar{x}_1(\bar{u}); \ldots; \bar{x}_\tau(\bar{u}))$ function of $\bar{u} = (u_0; \ldots; u_{\tau-1})$ as

$$\bar{x}_{t+1}(\bar{u}) = \phi_t(\bar{x}_t(\bar{u}), u_t) \qquad \bar{x}_1(\bar{u}) = \phi_0(\hat{x}_0, u_0) \qquad \text{(Traj)}$$

Reformulation as optimization of compositions

$$\min_{\bar{u}} \quad h(\bar{x}(\bar{u})) + g(\bar{u})$$

with $h(\bar{x}) = \sum_{t=1}^{\tau} h_t(x_t)$, $g(\bar{u}) = \sum_{t=0}^{\tau-1} g_t(u_t)$

Oracles

Model-minimization steps

1. Model the objective at iterate $\bar{u}$ by
   - linearizing the trajectory $\bar{x}(\bar{u} + \bar{v}) \approx \bar{x}(\bar{u}) + \nabla\bar{x}(\bar{u})^\top\bar{v}$
   - approx. cost $h(\bar{x} + \bar{y}) \approx m_h(\bar{x} + \bar{y})$, penalty $g(\bar{u} + \bar{v}) \approx m_g(\bar{u} + \bar{v})$
2. Minimize model with proximal term to get next iterate $\bar{u}^+$

$$\bar{u}^+ = \bar{u} + \arg\min_{\bar{v}} \left\{ m_h\left(\bar{x}(\bar{u}) + \nabla\bar{x}(\bar{u})^\top\bar{v}\right) + m_g(\bar{u} + \bar{v}) + \tfrac{1}{2\gamma}\|\bar{v}\|_2^2 \right\} \qquad \text{(Oracle)}$$

Examples

- Gradient step: $m_h(\bar{x} + \bar{y}) = h(\bar{x}) + \nabla h(\bar{x})^\top\bar{y}$, same for m_g
- Regularized Gauss-Newton:
  $m_h(\bar{x} + \bar{y}) = h(\bar{x}) + \nabla h(\bar{x})^\top\bar{y} + \frac{1}{2}y^\top\nabla^2 h(\bar{x})y$, same for m_g

Optimization steps by dynamic programming

- Subproblems (Oracle) are given by $\bar{u}^+ = \bar{u} + \bar{v}^*$ where $\bar{v}^*$ solves

$$\min_{\substack{y_0,\ldots,y_\tau\\v_0,\ldots,v_{\tau-1}}} \sum_{t=1}^{\tau} m_{h_t}(x_t + y_t) + \sum_{t=0}^{\tau-1} m_{g_t}(u_t + v_t) + \tfrac{1}{2\gamma}\|v_t\|^2$$

$$\text{s.t.} \quad y_{t+1} = \Phi_{t,x}^\top y_t + \Phi_{t,u}^\top v_t \qquad y_0 = 0$$

  where $\Phi_{t,x} = \nabla_x\phi_t(x_t, u_t)$, $\Phi_{t,u} = \nabla_u\phi_t(x_t, u_t)$ and $x_t = \bar{x}_t(\bar{u})$.
- For m_{h_t}, m_{g_t} linear or quadratics this is solved in linear time w.r.t. τ by dynamic programming

Consequences

→ Classical gradient back-propagation is a dynamic programming method

→ Gradient and regularized Gauss-Newton steps have **same cost** w.r.t. τ

→ ILQR is Gauss-Newton: needs **regularization** from optim. viewpoint

Optimization steps by automatic differentiation oracle

Define automatic differentiation oracle as any procedure that computes

$$z \to \nabla\bar{x}(\bar{u})z$$

for $\bar{x} : \mathbb{R}^{\tau p} \to \mathbb{R}^{\tau d}$ defined as in (Traj), $\bar{u} \in \mathbb{R}^{\tau p}$ and $z \in \mathbb{R}^{\tau d}$.

For final-state cost, $h = h_\tau$, dual of optimization step (Oracle) reads

$$\min_{z\in\mathbb{R}^d} \quad m_{h_\tau}^*(z) + \left(m_g + \tfrac{1}{2\gamma}\|\cdot\|_2^2\right)^*(-\nabla\bar{x}_\tau(\bar{u})z)$$

Consequence

→ For m_h, m_g quadratics, step (Oracle) given by $2d + 1$ calls to an automatic differentiation oracle using conjugate gradient method to solve the dual problem

Regularized and Accelerated ILQR

Denote $\bar{u}^+ = \text{oracle}(\bar{u}, \gamma)$ the step in (Oracle) for m_h, m_g quadratics

Regularized ILQR

Regularized ILQR reads

$$\bar{u}^{(k+1)} = \text{oracle}(\bar{u}^{(k)}, \gamma_k)$$

where step-size γ_k is chosen such that

$$f(\bar{u}^{(k+1)}) \leq m_f(\bar{u}^{(k+1)}) + \tfrac{1}{2\gamma_k}\|\bar{u}^{(k+1)} - \bar{u}^{(k)}\|_2^2$$

with f the objective, m_f the model around $\bar{u}^{(k)}$

For h, g quadratics, such step-sizes exist and convergence to an ε stationary point is guaranteed after $\mathcal{O}(\varepsilon^2)$ iterations

Accelerated Regularized ILQR

Use extrapolation steps

$$\bar{v}^{(k)} = \bar{u}^{(k)} + \theta_k(\bar{u}^{(k)} - \bar{u}^{(k-1)})$$
$$\bar{u}^{(k+1)} = \text{oracle}(\bar{v}^{(k)}, \delta_k)$$

For h, g quadratics, if $m_f(\bar{u} + \bar{v}) \leq f(\bar{u} + \bar{v})$ then convergence to an ε-minimum is guaranteed after $\mathcal{O}(\sqrt{\varepsilon})$

Synthetic experiments

Compare ILQR, Regularized ILQR and Accelerated Regularized ILQR on synthetic control experiments

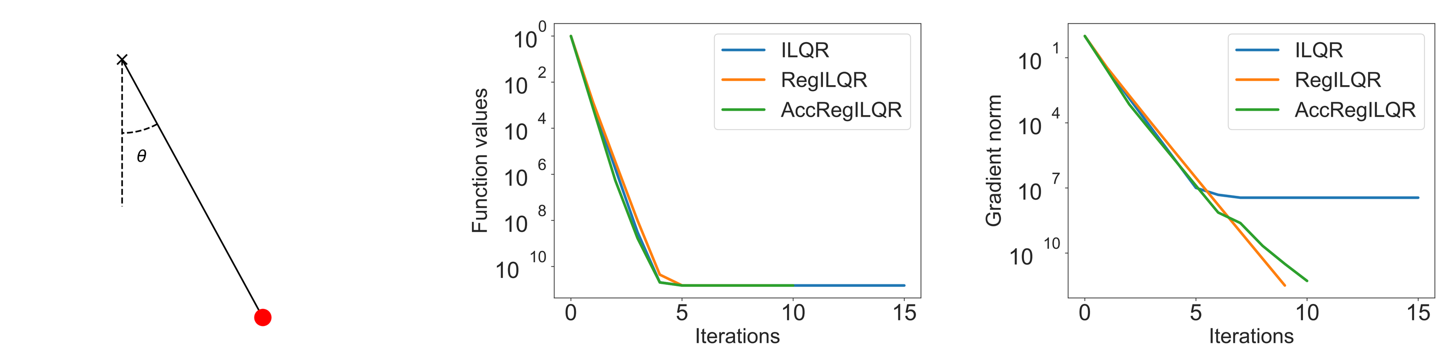

Figure 1: Swing-up a pendulum

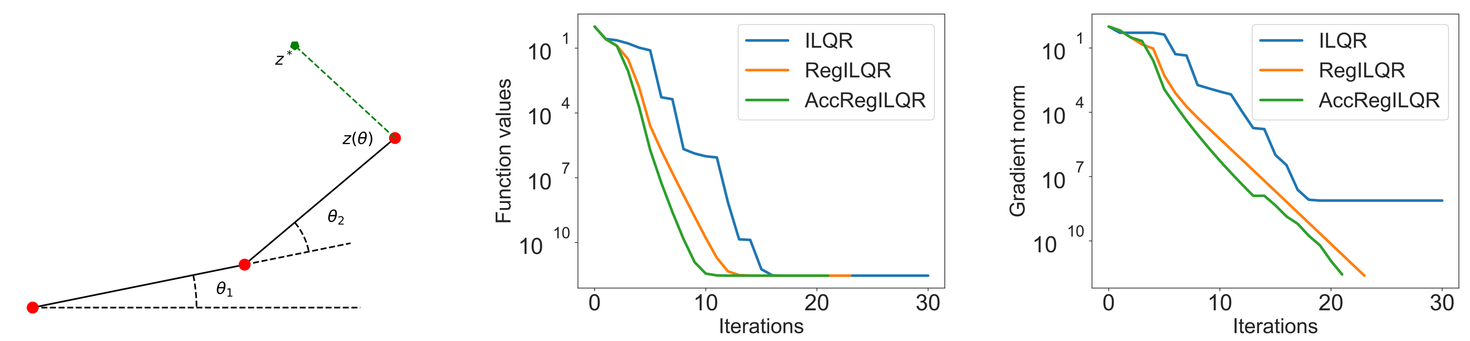

Figure 2: Moving two-link arm robot

Code available at https://github.com/vroulet/ilqc

References

Li, W. and Todorov, E. [2007], 'Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic systems', *International Journal of Control*