Complexity Bounds of Iterative Linear Quadratic Optimization Algorithms for Discrete Time Nonlinear Control

UNIVERSITY of WASHINGTON

Vincent Roulet, Siddhartha Srinivasa, Maryam Fazel, Zaid Harchaoui

University of Washington

Overview

- Nonlinear control is a non-convex problem with dynamical structure
- Yet, nonlinear control algo. may converge fast to optimal solution

$\rightarrow$ Identify sufficient conditions for global convergence

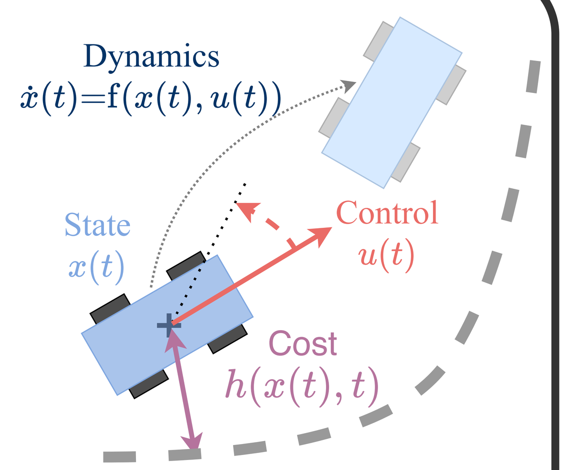
$\rightarrow$ Detail convergence rate

Nonlinear Control

Continuous Time

Trajectory $x(t)$ controlled by $u(t)$ via *dynamics* f to optimize *cost* h

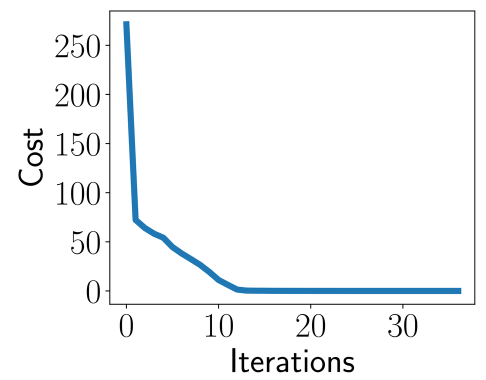
$$\min_{x(t), u(t)} \int_0^T \mathrm{h}(x(t), t) dt$$
$$\text{s.t.} \quad \dot{x}(t) = \mathrm{f}(x(t), u(t)), \ x(0) = \bar{x}_0$$



Discrete Time

Discretize dynamics and costs to get

$$\min_{\substack{x_0, \ldots, x_\tau \\ u_0, \ldots, u_{\tau-1}}} \sum_{t=1}^{\tau} h_t(x_t)$$
$$\text{s.t.} \quad x_{t+1} = f(x_t, u_t) \ x_0 = \bar{x}_0$$



Conv. algo. simple car model

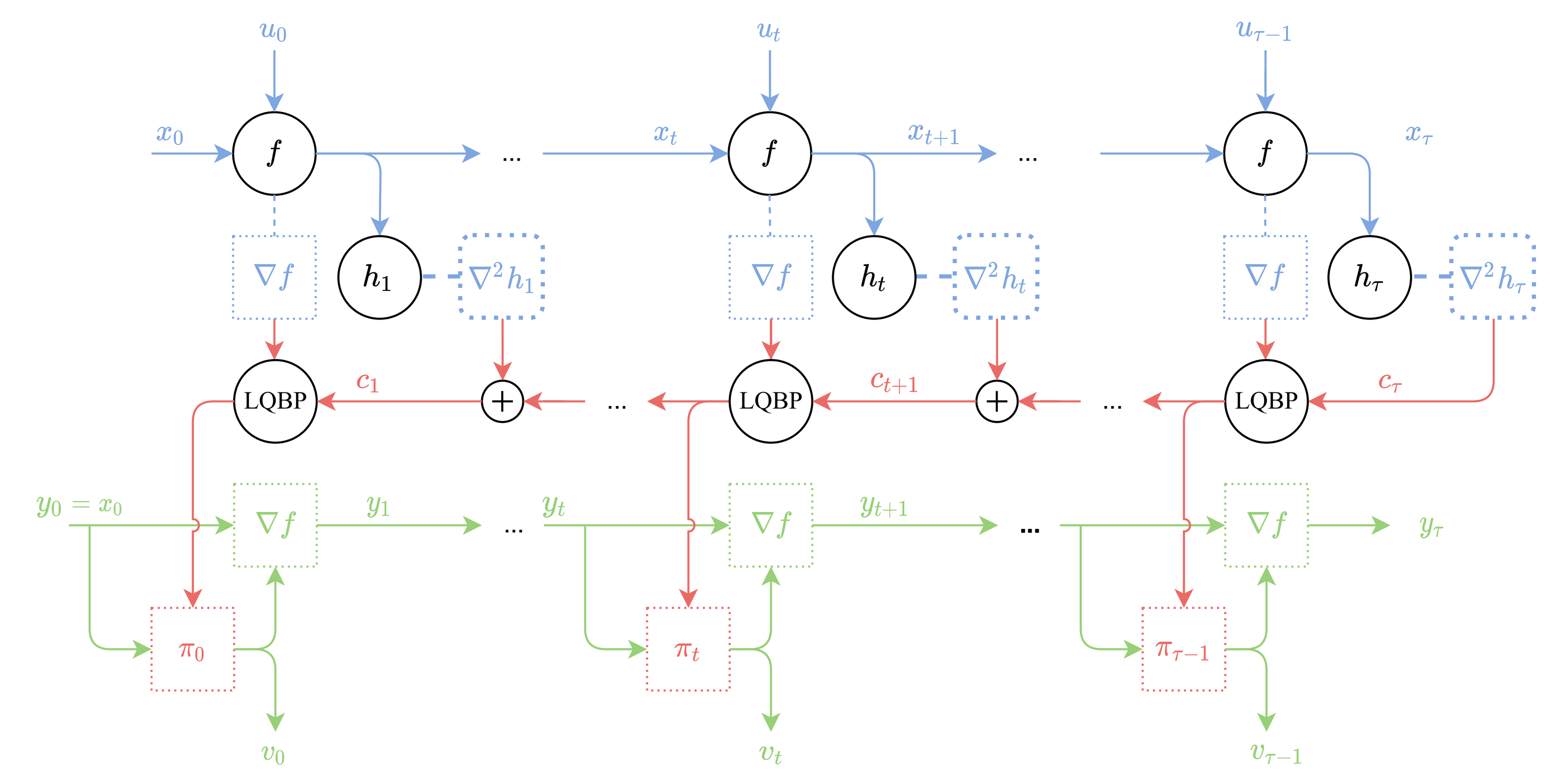
Iterative Linear Quadratic Regulator from current $u_0, \ldots, u_{\tau-1}$

1. Compute $x_{t+1} = f(x_t, u_t)$, for $t = 0, \ldots, \tau - 1$
   Record lin. approx. $\ell_f^{x_t, u_t}$ of f on x_t, u_t, quad. approx. $q_{h_t}^{x_t}$ of h_t on x_t
2. Define recursively min. cost of lin. quad. approx. from any y_t at time t

$$\mathrm{LQBP} : \ell_f^{x_t, u_t}, q_{h_t}^{x_t}, c_{t+1} \rightarrow \begin{cases} c_t : & y_t \mapsto q_{h_t}^{x_t}(y_t) + \min_{v_t} c_{t+1}(\ell_f^{x_t, u_t}(y_t, v_t)) \\ \pi_t : & y_t \mapsto \arg\min_{v_t} c_{t+1}(\ell_f^{x_t, u_t}(y_t, v_t)) \end{cases}$$

3. Roll-out optimal controls along the lin. dyn., update with $\gamma > 0$,

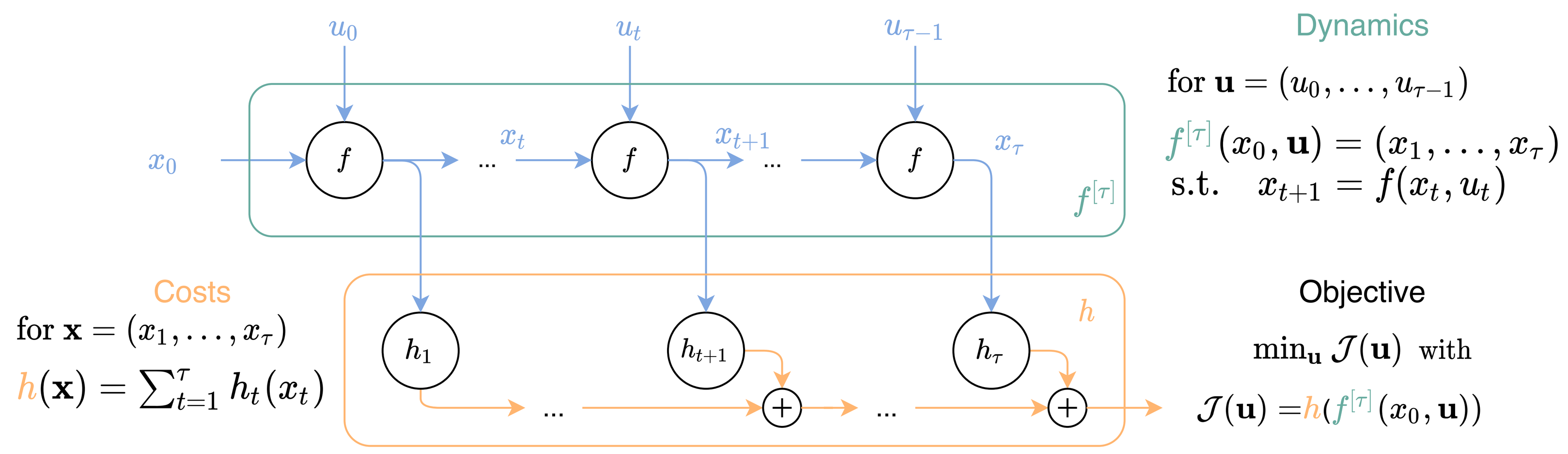
$$v_t = \pi_t(y_t), \quad y_{t+1} = \ell_f^{x_t, u_t}(y_t, v_t), \quad \text{update } u_t^{\text{next}} = u_t + \gamma v_t$$



A Global Convergence Condition

Optimization Viewpoint

$$\min_{\boldsymbol{u} = (u_0; \ldots; u_{\tau-1})} \{\mathcal{J}(\boldsymbol{u}) = h(f^{[\tau]}(x_0, \boldsymbol{u}))\}$$



Idea

- For h convex, *if we had access* to the inverse of $f^{[\tau]}$, we could reparameterize the problem to get a *convex problem*!
- The algorithm may only need the *possibility* to inverse $f^{[\tau]}$ through its linearized trajectories, namely we investigate whether

$$\forall x_0, \boldsymbol{u} \ \underline{\sigma}(\nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})) := \inf_{\boldsymbol{\lambda}} \|\nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u}) \boldsymbol{\lambda}\|_2 / \|\boldsymbol{\lambda}\|_2 \geq \boldsymbol{\sigma} > 0 \quad (S)$$

- For h $\boldsymbol{\mu}$ strongly cvx, this ensures that $\mathcal{J}$ is gradient dominated since

$$\|\nabla \mathcal{J}(\boldsymbol{u})\|_2^2 = \|\nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u}) \nabla h(\boldsymbol{x})\|_2^2 \geq \boldsymbol{\sigma}^2 \|\nabla h(\boldsymbol{x})\|_2^2 \geq \boldsymbol{\sigma}^2 \boldsymbol{\mu}(h(\boldsymbol{x}) - h^*) = \boldsymbol{\sigma}^2 \boldsymbol{\mu}(\mathcal{J}(\boldsymbol{u}) - \mathcal{J}^*)$$

  hence a gradient descent could converge globally for example
- $(S) \Leftrightarrow \boldsymbol{\lambda} \mapsto \nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u}) \boldsymbol{\lambda}$ injective $\Leftrightarrow \boldsymbol{v} \mapsto \nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})^\top \boldsymbol{v}$ surjective

Characterization in Terms of Dynamic

If the linearization, $v \mapsto \nabla_u f(x, u)^\top v$, of l_f-Lip. cont. dyn. f is surj.

$$\forall x, u, \quad \underline{\sigma}(\nabla_u f(x, u)) \geq \sigma_f > 0$$

then the linearization of the traj., $\boldsymbol{v} \mapsto \nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})^\top \boldsymbol{v}$, is surj,

$$\forall x_0, \boldsymbol{u}, \quad \underline{\sigma}(\nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})) \geq \sigma_f / (1 + l_f) > 0$$

$\rightarrow$ We can focus on f and decompose f according to discretization

Multi-step Discretization

Dyn. fractionated in k steps

$$f(x_t, u_t) = x_{t+1}$$
$$\text{s.t. } x_{t+(s+1)/k} = \phi(x_{t+s/k}, u_{t+s/k})$$

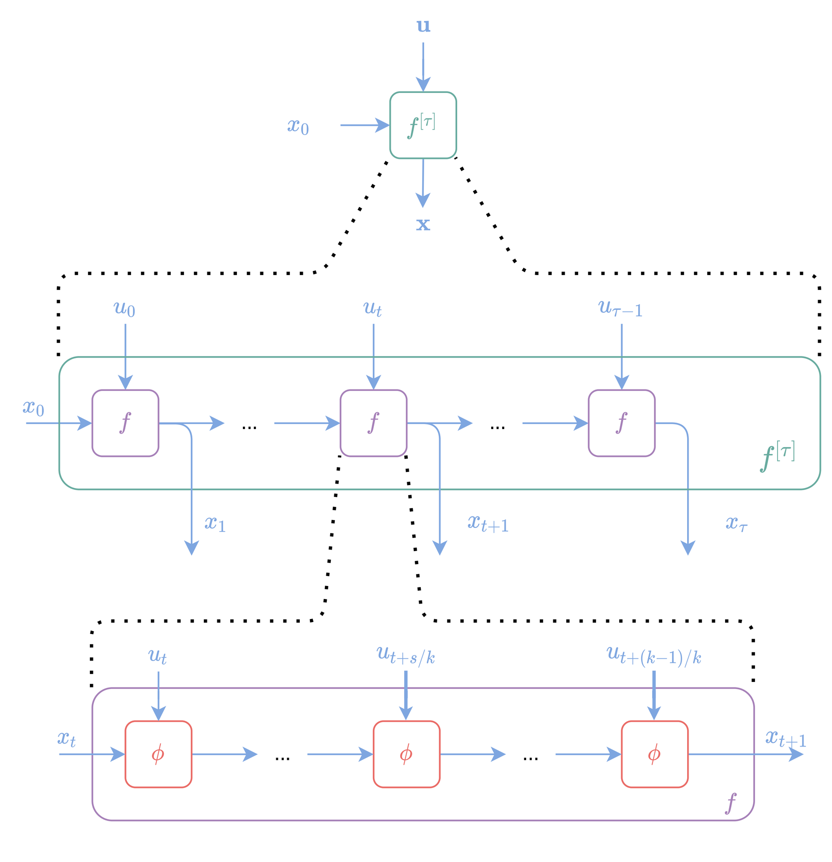
such as $\phi(y_t, v_t) = y_t + \Delta \mathrm{f}(y_t, v_t)$ for f continuous-time dynamic.

To satisfy (S), suffices that ϕ is *linearizable by static feedback*[1]

Example: for $x = (z, \dot{z})$,

$$z_{t+1} = z_t + \Delta \dot{z}_t$$
$$\dot{z}_{t+1} = \dot{z}_t + \Delta \psi(z_t, \dot{z}_t, v_t)$$

with $|\partial_v \psi(z_t, \dot{z}_t, v_t)| \neq 0$



Zooming in the dynamical structure

Convergence Analysis

Regularized Iterative Linear Quadratic Control (ILQR)

- Add $\nu \|v_t\|_2^2$ in computation of c_t, π_t in ILQR,
- Denote $\boldsymbol{v} = \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$ the output computed in roll-out phase

Generalized Gauss-Newton[3]

- ILQR minimizes a quad. approx. of h on top of a lin. approx. of g for $g(\boldsymbol{u}) = f^{[\tau]}(x_0, \boldsymbol{u})$, so it can be summarized as

$$\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u}) = \arg\min_{\boldsymbol{v}} q_h^{g(\boldsymbol{u})}(\ell_g^{\boldsymbol{u}}(\boldsymbol{v})) + \frac{\nu}{2} \|\boldsymbol{v}\|_2^2$$
$$= -(\nabla g(\boldsymbol{u}) \nabla^2 h(g(\boldsymbol{u})) \nabla g(\boldsymbol{u})^\top + \nu \mathrm{I})^{-1} \nabla g(\boldsymbol{u}) \nabla h(g(\boldsymbol{u}))$$

which is a *regularized generalized Gauss-Newton method*

Convergence Proof Idea

1. For large enough ν, $\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u}) \approx -\nu^{-1} \nabla g(\boldsymbol{u}) \nabla h(g(\boldsymbol{u}))$

$\rightarrow$ linear global convergence possible as for a gradient descent

2. Let $\boldsymbol{x}^{\text{next}} = g(\boldsymbol{u} + \boldsymbol{v})$, for $\boldsymbol{v} = \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$,

$$\boldsymbol{x}^{\text{next}} \approx g(\boldsymbol{u}) + \nabla g(\boldsymbol{u})^\top \boldsymbol{v} = \boldsymbol{x} - (\nabla^2 h(\boldsymbol{x}) + \nu (\nabla g(\boldsymbol{u})^\top \nabla g(\boldsymbol{u}))^{-1})^{-1} \nabla h(\boldsymbol{x})$$

so for small enough ν, we have $\boldsymbol{x}^{\text{next}} \approx \boldsymbol{x} - \nabla^2 h(\boldsymbol{x})^{-1} \nabla h(\boldsymbol{x})$

$\rightarrow$ local quadratic convergence possible as for a Newton method

3. Can show that a regularization $\nu \propto \|\nabla h(\boldsymbol{x})\|_2$ ensures both!

Complexity Bound

For g lip. cont., smooth, h strongly cvx, smooth, Hessian-smooth, if g satisfies $\forall \boldsymbol{u}, \underline{\sigma}(\nabla g(\boldsymbol{u})) \geq \sigma_g > 0$, taking $\nu(\boldsymbol{u}) = \bar{\nu} \|\nabla h(g(\boldsymbol{u}))\|_2$ for $\bar{\nu}$ large enough ILQR converges to accuracy ε in

$$\underbrace{4\theta_g(\sqrt{\delta_0} - \sqrt{\delta})}_{\text{1st phase}} + \underbrace{2\rho_h \ln\left(\frac{\delta_0}{\delta}\right) + 2\alpha \ln\left(\frac{\theta_g \sqrt{\delta_0} + \rho_g}{\theta_g \sqrt{\delta} + \rho_g}\right)}_{\text{2nd phase}} + \underbrace{O(\ln \ln(\varepsilon))}_{\text{3rd phase}}$$

iterations, each having a comput. complexity $O(\tau(\dim(x) + \dim(u))^3)$, where $\delta_0 = \mathcal{J}(\boldsymbol{u}^{(0)}) - \mathcal{J}^*$ is the initial gap, δ is the gap of quadratic conv., ρ_h, ρ_g, θ_h, θ_g α are condition numbers

Extensions[1]

- Analyzed Differential Dynamic Programming implementation
- Analyzed costs satisfying Łojasiewicz inequality or self-concordance

Code at `https://github.com/vroulet/ilqc`, **Experiments** in [2]

References

[1] Roulet, V., Srinivasa, S., Fazel, M., Harchaoui, Z. (2022). Complexity Bounds of Iterative Linear Quadratic Optimization Algorithms for Discrete Time Nonlinear Control. *arXiv preprint*

[2] Roulet, V., Srinivasa, S., Fazel, M., Harchaoui, Z. (2022). Iterative Linear Quadratic Optimization for Nonlinear Control: Differentiable Programming Algorithmic Templates. *arXiv preprint*

[3] Sideris, A., Bobrow, J. (2005). A fast sequential linear quadratic algorithm for solving unconstrained nonlinear optimal control problems. *American Control Conference.*