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

Vincent Roulet
Co-authors: Siddhartha Srinivasa, Maryam Fazel, Zaid Harchaoui

paper: `https://arxiv.org/abs/2204.02322`
code: `https://github.com/vroulet/ilqc`

International Conference on Continuous Optimization

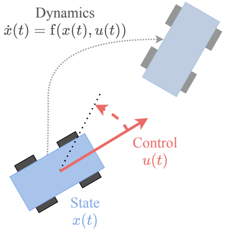
07/27/2022

Nonlinear Control Problems

Continuous Time

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

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



Dynamics of a car

Discrete Time

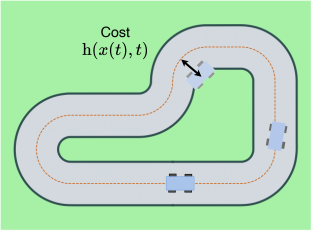
Discretize dynamics and costs with e.g. Euler scheme
Optimize over controls $u_0, \ldots, u_{\tau-1}$

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

Algorithms Principle (Jacobson & Mayne 1970, Li & Todorov 2007)

Current controls $u_0, \ldots, u_{\tau-1}$ with trajectory $x_0, \ldots, x_\tau$

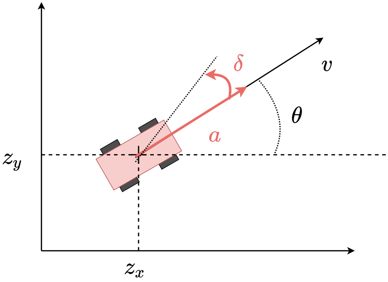
1. Linearize dynamics f around x_t, u_t
2. Take quadratic approx. of the costs h_t around x_t
3. Solve resulting lin. quad. problem
4. Repeat from 1.



Tracking objective

Autonomous Car Racing

Simple model of a car

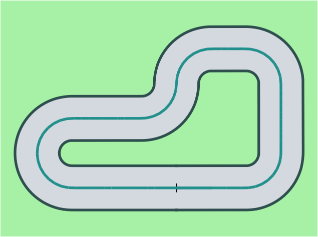


$$x = (z_x, z_y, \theta, v), \qquad u = (\delta, a)$$

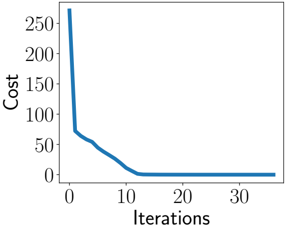
$$\dot{z}_x = v \cos\theta \qquad \dot{\theta} = v \tan(\delta)$$

$$\dot{z}_y = v \sin\theta \qquad \dot{v} = a$$

Algo. converges *fast* to *optimal trajectory*



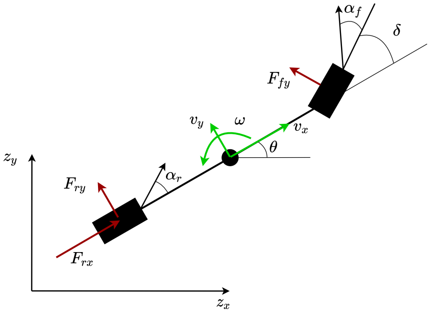
Optimized trajectory horizon $\tau = 100$



Convergence of the algorithm

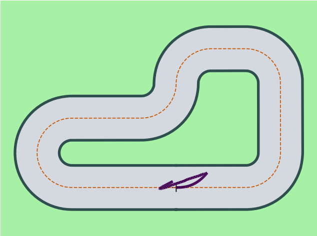
Autonomous Car Racing

Bicycle model of a car (Liniger et al. 2015)

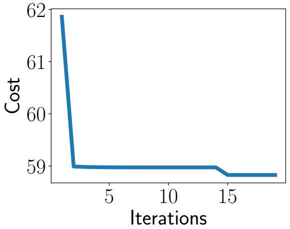


Models tire forces (highly non-linear)

Unclear whether the algorithm succeeded...



Optimized trajectory horizon $\tau = 100$



Convergence of the algorithm

Outline

Outline

A Sufficient Condition for Global Convergence

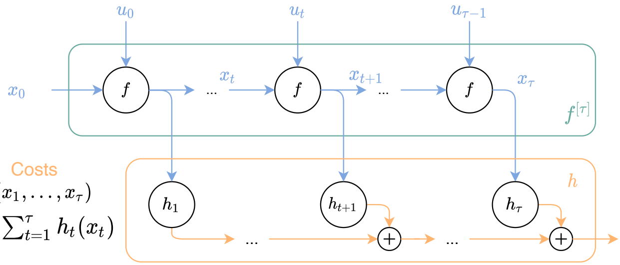
Implementation and Convergence Analysis

Optimization Viewpoint

Compositional Problem for $\boldsymbol{u} = (u_0, \ldots, u_{\tau-1})$, with $x_0 = \bar{x}_0$

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

- $f^{[\tau]}$ take sequence of controls outputs sequence of states
- h total cost on the states



Dynamics

for $\mathbf{u} = (u_0, \ldots, u_{\tau-1})$

$f^{[\tau]}(x_0, \mathbf{u}) = (x_1, \ldots, x_\tau)$
s.t. $\quad x_{t+1} = f(x_t, u_t)$

Costs

for $\mathbf{x} = (x_1, \ldots, x_\tau)$

$h(\mathbf{x}) = \sum_{t=1}^{\tau} h_t(x_t)$

Objective

$\mathcal{J}(\mathbf{u}) = h(f^{[\tau]}(x_0, \mathbf{u}))$

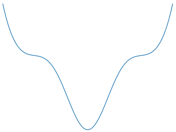
A Sufficient Condition for Global Convergence

Idea

- Prove sufficient condition for global conv. of 1st order methods, such as, for $c > 0$,

$$\|\nabla \mathcal{J}(\boldsymbol{u})\|_2^2 \geq c(\mathcal{J}(\boldsymbol{u}) - \mathcal{J}^*)$$

Gradient dominating objective $\mathcal{J}$



Non-convex, gradient dominating function

Derivation

- Consider the total cost h to be μ-strongly cvx s.t. $\|\nabla h(\boldsymbol{x})\|_2^2 \geq \boldsymbol{\mu}(h(\boldsymbol{x}) - h^*)$
- We have $\mathcal{J}(\boldsymbol{u}) = h(f^{[\tau]}(x_0, \boldsymbol{u}))$ so $\|\nabla \mathcal{J}(\boldsymbol{u})\|_2^2 = \|\nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u}) \nabla h(\boldsymbol{x})\|_2^2$ where $\nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u}) \in \mathbb{R}^{\dim(\boldsymbol{u}) \times \dim(\boldsymbol{x})}$ (transpose Jacobian)
- So if $f^{[\tau]}(x_0, \boldsymbol{u})$ satisfies

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

where $\underline{\sigma}(A)$ is the minimal singular value of A, then

$$\|\nabla \mathcal{J}(\boldsymbol{u})\|_2^2 \geq \boldsymbol{\sigma}^2 \|\nabla h(\boldsymbol{x})\|_2^2 \geq \boldsymbol{\sigma}^2 \boldsymbol{\mu}(h(\boldsymbol{x}) - h^*) = \boldsymbol{\sigma}^2 \boldsymbol{\mu}(\mathcal{J}(\boldsymbol{u}) - \mathcal{J}^*) \quad \checkmark$$

Interpretation of a Sufficient Condition for Global Convergence

Interpretation

$$\underline{\sigma}(\nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})) > 0$$

$$\iff \text{Reverse mode of auto-diff } \boldsymbol{\lambda} \mapsto \nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})\boldsymbol{\lambda} \text{ is injective}$$

$$\iff \text{Forward mode of auto-diff } \boldsymbol{v} \mapsto \nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})^\top \boldsymbol{v} \text{ is surjective}$$

Here $\boldsymbol{y} = \nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})^\top \boldsymbol{v}$ is the linearization of the trajectories given as

$$y_{t+1} = \nabla_{x_t} f(x_t, u_t)^\top y_t + \nabla_{u_t} f(x_t, u_t)^\top v_t, \quad y_0 = 0$$

So $\underline{\sigma}(\nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})) > 0$ if the linearization of the trajectories are *surjective*

How to verify this condition from f only?

Previous work

Sufficient optimality conditions in continuous time done by Mangasarian (1966)

$\rightarrow$ Translatable in discrete time but requires convexity of implicitly defined functions...

Characterization of a Sufficient Condition for Global Convergence

Lemma (R. et al. (2022))

If the linearization, $v \to \nabla_u f(x, u)^\top v$*, of* l_f*-Lip. cont. dynamics* f *is surjective,*

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

then the linearization of the trajectories, $\boldsymbol{v} \to \nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})^\top \boldsymbol{v}$*, is surjective,*

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

Problem:

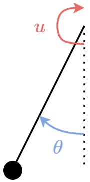
- Usually less control variables than state variables $\dim(u(t)) < \dim(x(t))$
  So $\underline{\sigma}(\nabla_u f(x(t), u(t))) > 0$ impossible *when looking in the continuous time*

→ But we can use multistep discretization schemes

Intuition for a Sufficient Condition for Global Convergence

Pendulum dynamics

$$m\ddot{\theta}(t) = -mg\sin\theta(t) - \mu\dot{\theta}(t) + u(t)$$



One step Euler scheme
$f(x_t, u_t) = x_{t+1}$ for $x_t = (\theta_t, \omega_t)$ with $\omega = \dot{\theta}$

$$\begin{aligned} \text{angle} \quad & \theta_{t+1} = \theta_t + \Delta\omega_t \\ \text{angle speed} \quad & \omega_{t+1} = \omega_t - \Delta(g\sin\theta_t - \mu\omega_t) + \Delta u_t \end{aligned}$$

Linearization surjective? ✗

Two steps Euler scheme $f(x_t, u_t) = x_{t+1}$ with $u_t = (v_t, v_{t+1/2})$

$$\begin{aligned} \theta_{t+1/2} &= \theta_t + \Delta\omega_t & \theta_{t+1} &= \theta_t + \ldots + \Delta^2 v_t \\ \omega_{t+1/2} &= \omega_t - \Delta(g\sin\theta_t - \mu\omega_t) + \Delta v_t & \omega_{t+1} &= \omega_t + \ldots + \Delta v_{t+1/2} \end{aligned}$$

Linearization surjective w.r.t. $u_t = (v_t, v_{t+1/2})$? ✓

Overall Analysis

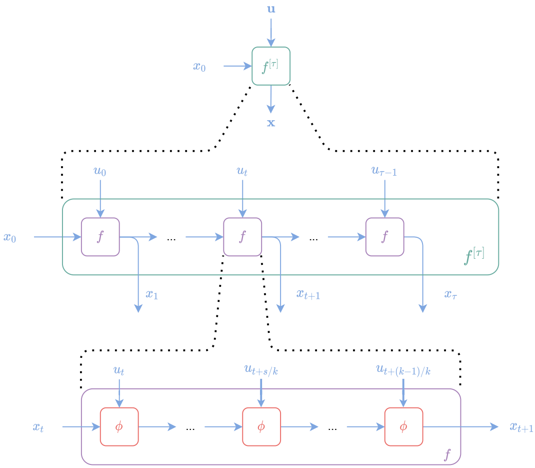
Trajectory decomposed in τ steps

$$f^{[\tau]}(x_0, \boldsymbol{u}) = (x_1; \ldots; x_\tau)$$
$$\text{s.t.} \quad x_{t+1} = f(x_t, u_t)$$

Dynamic 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.



Zooming in the dynamical structure

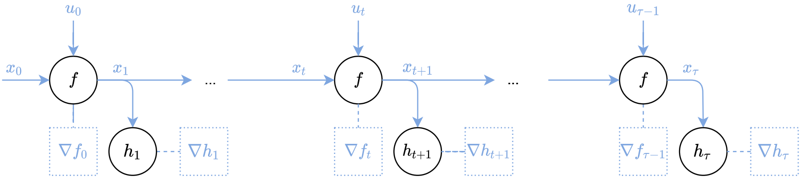
Sufficient condition for global convergence can be verified by analyzing whether ϕ can be *linearized by static feedback*, *see R. et al. (2022)*

Outline

Implementation

Gradient oracle

- Linear approx. of dynamics, costs,
- Gradients of objective computed through dynamics

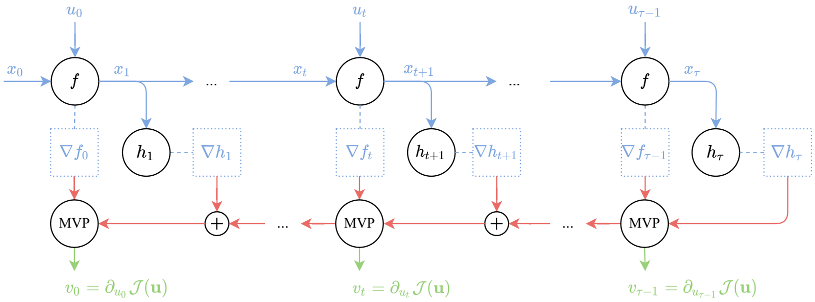


Forward pass
Compute objective and linear approx.

Implementation

Gradient oracle

- Linear approx. of dynamics, costs,
- Gradients of objective computed through dynamics



Backward pass

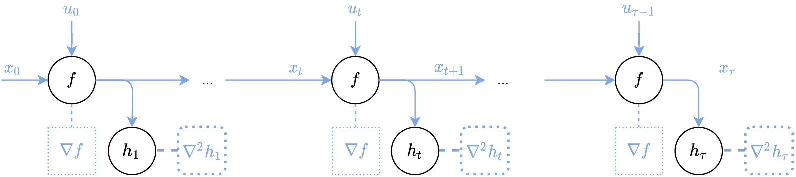
Backpropagate gradients through Matrix Vector Products (MVP)

Output gradients of objective w.r.t. control variables

Implementation

Linear Quadratic Regulator oracle

- Linear approx. of dynamics, quadratic approx. of costs
- Regularized linear quadratic approx. of objective minimized through dynamics



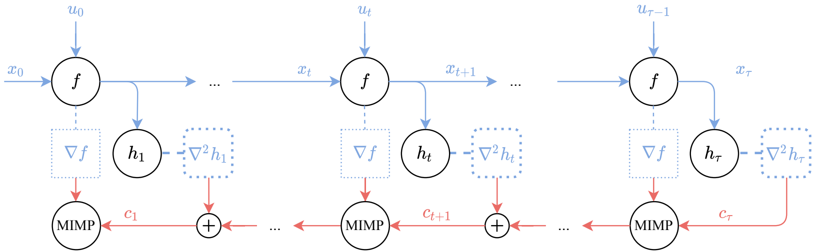
Forward pass

Compute objective, linear approx. of dynamics, *quad. approx. of costs*

Implementation

Linear Quadratic Regulator oracle

- Linear approx. of dynamics, quadratic approx. of costs
- Regularized linear quadratic approx. of objective minimized through dynamics



Backward pass

Define recursively minimum cost of reg. lin. quad. approx. starting from **any** y_t at time t

$$c_t : y_t \mapsto q_{h_t}^{x_t}(y_t) + \min_{v_t} \left\{ \nu \|v_t\|_2^2 + c_{t+1}(\ell_f^{x_t,u_t}(y_t, v_t)) \right\}$$

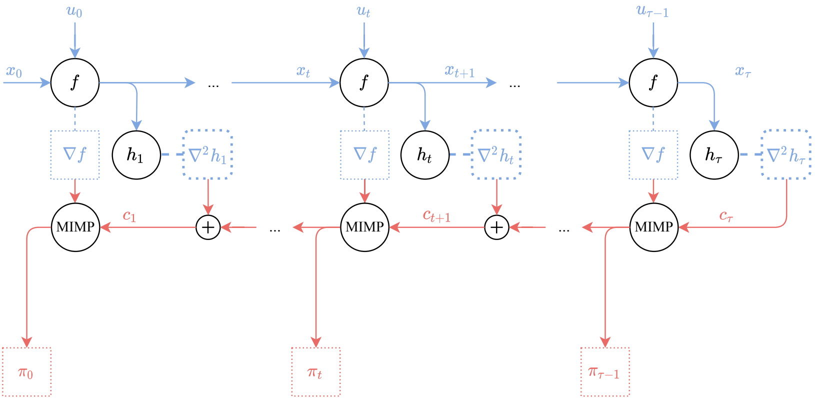
$q_{h_t}^{x_t}$ quad. approx of h_t on x_t $\qquad$ $\ell_f^{x_t,u_t}$ lin. approx of f on x_t, u_t

where c_t is a quad. function param. by Matrices Inverse & Matrices Products (MIMP)

Implementation

Linear Quadratic Regulator oracle

- Linear approx. of dynamics, quadratic approx. of costs
- Regularized linear quadratic approx. of objective minimized through dynamics



Backward pass

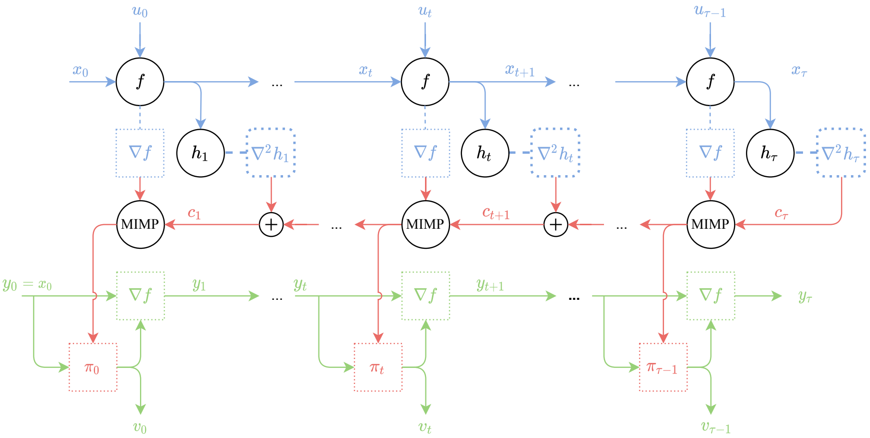
Record optimal control of reg. lin. quad. approx. starting from **any** y_t at time t

$$\pi_t : y_t \mapsto \arg\min_{v_t} \left\{ \nu \|v_t\|_2^2 + c_{t+1}(\ell_f^{x_t,u_t}(y_t, v_t)) \right\}$$

Implementation

Linear Quadratic Regulator oracle

- Linear approx. of dynamics, quadratic approx. of costs
- Regularized linear quadratic approx. of objective minimized through dynamics



Roll-out pass

Roll-out optimal controls along the linear dynamics

$$v_t = \pi_t(y_t), \qquad y_{t+1} = \ell_f^{x_t,u_t}(y_t, v_t)$$

Convergence Analysis

Problem

$$\min_{\boldsymbol{u}} \left\{ \mathcal{J}(\boldsymbol{u}) = h(g(\boldsymbol{u})) \right\}, \text{ where } g(\boldsymbol{u}) = f^{[\tau]}(\bar{x}_0, \boldsymbol{u}), \quad h(\boldsymbol{x}) = \sum_{t=1}^{\tau} h_t(x_t)$$

Algorithm (Li & Todorov 2007)

$$\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \mathrm{LQR}_{\nu_k}(\mathcal{J})(\boldsymbol{u}^{(k)}) \qquad \text{(ILQR)}$$

where $\mathrm{LQR}_{\nu_k}(\mathcal{J})(\boldsymbol{u}^{(k)})$ is the oracle returning a direction computed by dynamic programming with a regularization ν_k

Assumptions

• costs h_t: μ_h-strongly convex, L_h-smooth, M_h-smooth Hessian
$\rightarrow$ same for overall cost h

• dynamic f: l_f-Lip. continuous, L_f smooth with $\underline{\sigma}(\nabla_u f(x, u)) \geq \sigma_f > 0$
$\rightarrow$ mapping g: l_g-Lip.continous, L_g-smooth with $\underline{\sigma}(\nabla g(\boldsymbol{u})) \geq \sigma_g > 0$
with l_g, L_g, σ_g estimable from l_f, L_f, σ_f

Convergence Analysis Viewpoint

ILQR as a generalized Gauss-Newton (Sideris & Bobrow 2005, Wright 1990)

- Overall ILQR minimizes a quadratic approx. of h on top of a linear approx. of g
- So it can be summarized as

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

which is a *regularized generalized Gauss-Newton method*

Convergence proof idea

1. For large enough regularization, $\mathrm{LQR}_{\nu}(\mathcal{J})(\boldsymbol{u}) \approx -\nu^{-1}\nabla g(\boldsymbol{u})\nabla h(g(\boldsymbol{u}))$
$\rightarrow$ linear global convergence possible as for a gradient descent

2. Denoting $\boldsymbol{x}^{\text{next}} = g(\boldsymbol{u}+\boldsymbol{v})$ for $\boldsymbol{v} = \mathrm{LQR}_{\nu}(\mathcal{J})(\boldsymbol{u})$, with simple linear algebra,
$$\boldsymbol{x}^{\text{next}} \approx g(\boldsymbol{u}) + \nabla g(\boldsymbol{u})^\top \boldsymbol{v} = \boldsymbol{x} - (\nabla^2 h(\boldsymbol{x}) + \nu(\nabla g(\boldsymbol{u})^\top \nabla g(\boldsymbol{u}))^{-1})^{-1}\nabla h(\boldsymbol{x}).$$
so for small enough regularization $\boldsymbol{x}^{\text{next}} \approx \boldsymbol{x} - \nabla^2 h(\boldsymbol{x})^{-1}\nabla h(\boldsymbol{x})$
$\rightarrow$ local quadratic convergence possible as for a Newton method

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

Previous work

Global convergence of regularized Gauss-Newton a.k.a. Levenberg-Marquardt e.g. (Bergou et al. 2020)

Local convergence of generalized Gauss-Newton (Yamashita & Fukushima 2001, Diehl & Messerer 2019)

Complexity Bound for ILQR

Theorem (R. et al. (2022)[1])

Under the aforementioned assumptions, the ILQR algorithm equipped with $\nu(\boldsymbol{u}) = \bar{\nu}\|\nabla h(g(\boldsymbol{u}))\|_2$ for $\bar{\nu}$ large enough 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. :* $\delta_0 \leq \delta \implies$ *3rd phase*
- $\rho_h = L_h/\mu_h$ *is the* condition number *of the costs*
- $\rho_g = l_g/\sigma_g$ *is the* condition number *of the linearized traj.*
- $\theta_h = M_h/\mu_h^{3/2}$ *is the param. of* self-concordance *of the costs*
- $\theta_g = L_g/(\sigma_g^2\sqrt{\mu_h})$ *acts as* self-concordance *param. for the linear-quadratic decomp.*
- α *is another cond. nb*

[1] Extensions to self-concordant or gradient dominated costs, differential dynamic programming algorithms available

Code Example from Toolbox ILQC

```
import torch
from envs.car import Car
from envs.backward import lin_quad_backward, quad_backward
from envs.rollout import roll_out_lin


# Define control problem and candidate control variables
env = Car(model='simple', discretization='euler', cost='exact',
          horizon=50, dt=0.02)
ctrls = torch.randn(env.horizon, env.dim_ctrl, requires_grad=True)

# ILQR/Gauss-Newton step
traj, costs = env.forward(ctrls, approx='linquad')
policies = lin_quad_backward(traj, costs, reg_ctrl=1.)[0]
gauss_newton_dir = roll_out_lin(traj, policies)
gauss_newton_step = ctrls + gauss_newton_dir

# Newton and Differentiable Dynamic Programming also available
```

Conclusion

Summary

- Conv. guarantees for canonical noncvx pb

$\rightarrow$ analyze problem at elementary scale
as done in a diff. prog. implementation

- Complexity bounds for ILQR

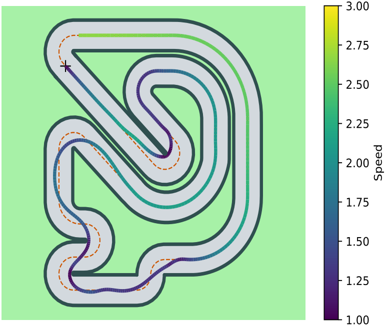
$\rightarrow$ quad. convergence at low iteration cost
by using a diff. prog. implementation

Future directions

- Quantify cond. number w.r.t. discretization step

$\rightarrow$ Optimal window for Model Predictive Control (MPC)?

- Use similar global convergence condition to analyze MPC (Na & Anitescu 2020)



Model Predictive Control
& contouring objective

Thank you for your attention!

Bergou, E. H., Diouane, Y. & Kungurtsev, V. (2020), 'Convergence and complexity analysis of a Levenberg-Marquardt algorithm for inverse problems', *Journal of Optimization Theory and Applications* **185**(3), 927–944.

Diehl, M. & Messerer, F. (2019), Local convergence of generalized Gauss-Newton and sequential convex programming, *in* '2019 IEEE 58th Conference on Decision and Control (CDC)', pp. 3942–3947.

Jacobson, D. & Mayne, D. (1970), *Differential Dynamic Programming*, Elsevier.

Li, W. & Todorov, E. (2007), 'Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system', *International Journal of Control* **80**(9), 1439–1453.

Liniger, A., Domahidi, A. & Morari, M. (2015), 'Optimization-based autonomous racing of 1: 43 scale RC cars', *Optimal Control Applications and Methods* **36**(5), 628–647.

Mangasarian, O. (1966), 'Sufficient conditions for the optimal control of nonlinear systems', *SIAM Journal on Control* **4**(1), 139–152.

Na, S. & Anitescu, M. (2020), 'Superconvergence of online optimization for model predictive control', *arXiv preprint arXiv:2001.03707* .

R., V., Srinivasa, S., Fazel, M. & Harchaoui, Z. (2022), 'Complexity bounds of iterative linear quadratic optimization algorithms for discrete time nonlinear control', *arXiv preprint arXiv:2204.02322* .

Sideris, A. & Bobrow, J. (2005), An efficient sequential linear quadratic algorithm for solving nonlinear optimal control problems, *in* 'Proceedings of the 2005 American Control Conference', pp. 2275–2280.

Wright, S. (1990), 'Solution of discrete-time optimal control problems on parallel computers', *Parallel Computing* **16**(2-3), 221–237.

Yamashita, N. & Fukushima, M. (2001), On the rate of convergence of the Levenberg-Marquardt method, *in* 'Topics in numerical analysis', Springer, pp. 239–249.