Nonlinear Control Algorithms as Globally Convergent Generalized Gauss-Newton Methods in a Differentiable Programming Framework

Vincent Roulet
Co-authors: Siddhartha Srinivasa, Maryam Fazel, Zaid Harchaoui
paper: https://arxiv.org/abs/2204.02322
code: https://github.com/vroulet/ilqc

04/13/2022

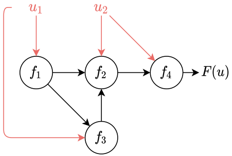
UNIVERSITY of WASHINGTON

Optimization in a Differentiable Programming Framework

Differentiable programming framework

- To solve $\min_u F(u)$, needs oracle as $\nabla F(u)$

1. Record gradients of elementary computations
$\rightarrow$ Needs differentiable programming framework
2. Use chain-rule along graph of computations
$\rightarrow$ Back-propagate gradients



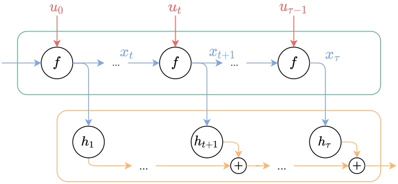
Generic graph of computations

Today's problem

- Simple dynamical structure $x_{t+1} = f(x_t, u_t)$
- Canonical example: nonlinear control

Why?

- Algorithms used are not just a gradient descent
- Surprising empirical performance
- May serve as a starting point to extend differentiable programming methods

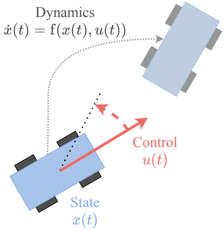


Graph of computations in nonlinear control

Nonlinear Control Problems

Continuous Time Control problem

- System driven by dynamics $\dot{x}(t) = \mathrm{f}(x(t), u(t))$
- Minimize cost $\mathrm{h}(x(t), t)$ over $t \in [0, T]$ for $x(0)$ fixed



Dynamics of a car

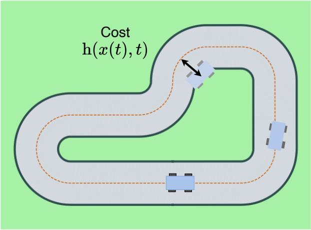
Discrete Time Control Problem

- Discretize dynamics as $x_{t+1} = f(x_t, u_t)$
- Minimize costs $h_t(x_t)$ over $t \in \{0, \ldots, \tau\}$ for x_0 fixed

Algorithms Principle

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

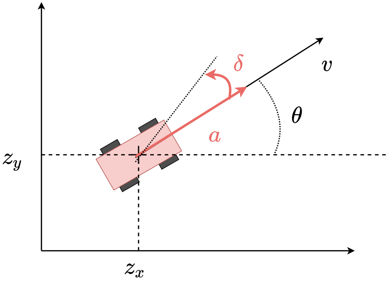
1. Linearize dynamics f around x_t, u_t
2. Take quadratic approx. of the costs h_t around x_t
3. Solve resulting lin. quad. problem
4. Repeat from 1.



Tracking objective

Autonomous Car Racing

Simple model of a car

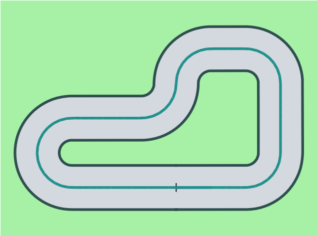


$$x = (z_x, z_y, \theta, v), \qquad u = (\delta, a)$$

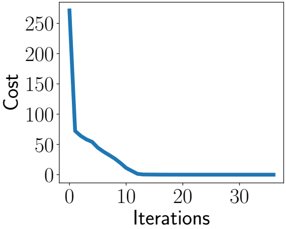
$$\dot{z}_x = v \cos\theta \qquad \dot{\theta} = v \tan(\delta)$$

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

Algo. converges *fast* to *optimal trajectory*



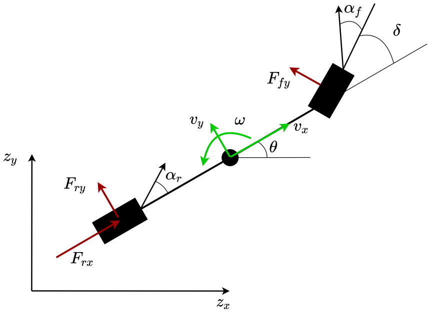
Optimized trajectory horizon $\tau = 100$



Convergence of the algorithm

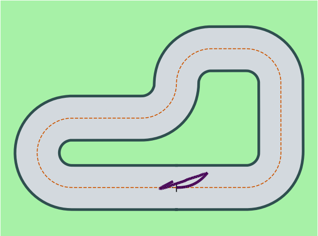
Autonomous Car Racing

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

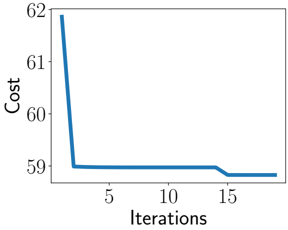


Models tire forces (highly non-linear)

Unclear whether the algorithm succeeded...



Optimized trajectory horizon $\tau = 100$



Convergence of the algorithm

Objectives

Questions

1. What are sufficient conditions to ensure global convergence?
2. How can we understand these algorithms from an optimization viewpoint?
3. What are the worst-case complexity bounds of these algorithms?

Related work

- Sufficient optimality conditions in continuous time (Mangasarian 1966)
  $\rightarrow$ Translatable in discrete time, requires convexity of implicitly defined functions
- Local convergence of Differential Dynamic Programming or generalized Gauss-Newton
  (Polak 2011, Murray & Yakowitz 1984, Liao & Shoemaker 1991, Yamashita & Fukushima 2001, Diehl & Messerer 2019)
- (Unregularized) Gauss-Newton, Newton methods for nonlinear control
  (Sideris & Bobrow 2005, Dunn & Bertsekas 1989, Wright 1990)

Outline

Iterative Linear Quadratic Optimization Algorithms for Nonlinear Control

A Sufficient Condition for Global Convergence

Convergence Analysis of ILQR and IDDP

Discrete Time Control Problems

Continuous Time Control problem

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

Discrete Time Control Problem

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

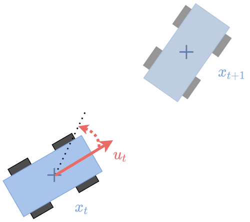
Discretization schemes: (time-step Δ)

Euler: $f(x_t, u_t) = x_t + \Delta \mathrm{f}(x_t, u_t)$

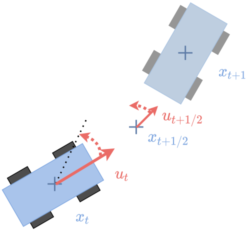
Multi-step: $f(x_t, u_t) = x_{t+1}$

s.t. $x_{t+(s+1)/k} = x_s + \Delta \mathrm{f}(x_{t+s/k}, u_{t+s/k})$

$\dim(u_t) = k \dim(u(t))$ for k steps



Euler discretization



2-step discretization

Nonlinear Control Algorithms for Discrete Time Control Problems

Forward Given a sequence of controls $u_0, \ldots, u_{\tau-1}$

a. Compute associated trajectory $x_{t+1} = f(x_t, u_t)$

b. Record linear expansions $\ell_f^{x_t, u_t}$ of the dynamics f around x_t, u_t

c. Record quadratic expansions $q_{h_t}^{x_t}$ of the costs around x_t

Backward Solve the associated regularized linear-quadratic control problem

$$\min_{\substack{y_0, \ldots, y_\tau \\ v_0, \ldots, v_{\tau-1}}} \quad \sum_{t=1}^{\tau} q_{h_t}^{x_t}(y_t) + \frac{\nu}{2} \sum_{t=0}^{\tau-1} \|v_t\|_2^2$$

$$\text{s.t} \quad y_{t+1} = \ell_f^{x_t, u_t}(y_t, v_t), \quad y_0 = 0$$

by computing recursively the cost-to-go from y_t at time t, from $c_\tau = q_{h_\tau}$,

$$c_t : y_t \mapsto \underbrace{q_{h_t}^{x_t}(y_t)}_{\text{current cost}} + \underbrace{\min_{v_t} \left\{ \frac{\nu}{2} \|v_t\|_2^2 + c_{t+1}(\ell_f^{x_t, u_t}(y_t, v_t)) \right\}}_{\substack{\text{optimal move at time } t \\ \text{associated policy } \pi_t \,:\, y_t \mapsto v_t^*}}$$

$\begin{pmatrix} \text{lin. quad. problem} \\ \to \text{closed form sol.} \end{pmatrix}$

Roll-out Update the iterates as $u_t^{\text{next}} = u_t + v_t$

where v_t are computed by rolling-out the policies π_t along either

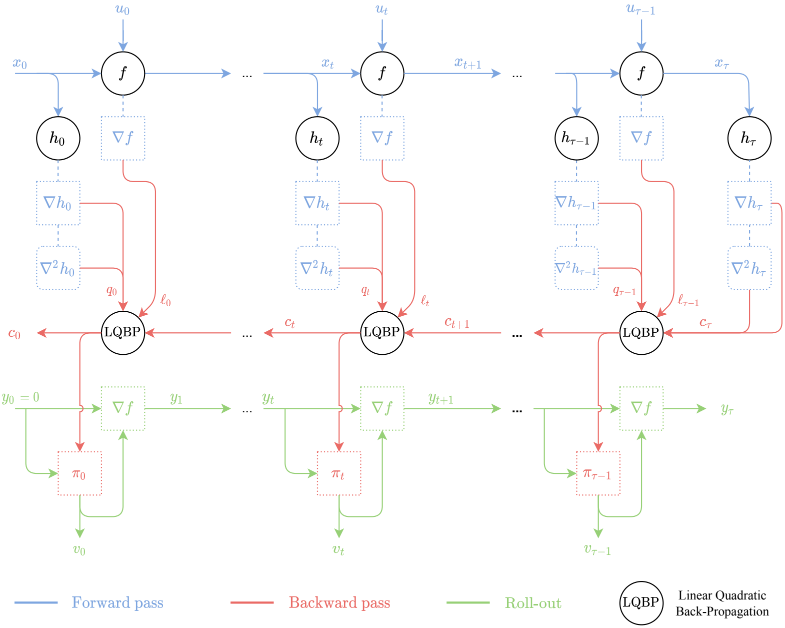
- the linearized dynamics → Iterative Linear Quadratic Regulator (ILQR) (Li & Todorov 2007)

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

- the original dynamics → Iterative Differential Dynamic Programming (IDDP) (Tassa et al. 2012)

$$v_t = \pi_t(y_t) \quad y_{t+1} = f(x_t + y_t, u_t + v_t) - f(x_t, u_t)$$

ILQR Computational Scheme



Outline

Objective Decomposition

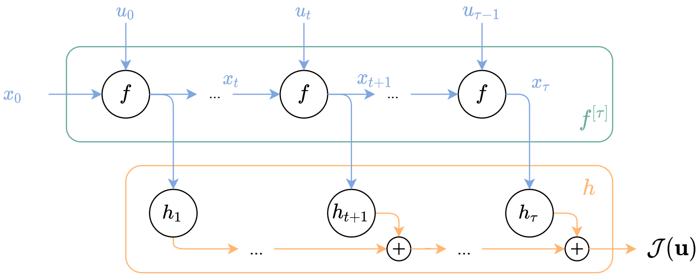
Control of τ steps of f for $\boldsymbol{u} = (u_0; \ldots; u_{\tau-1})$

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

Total cost for $\boldsymbol{x} = (x_1, \ldots, x_\tau)$ $h(\boldsymbol{x}) = \sum_{t=1}^{\tau} h_t(x_t)$

Composite objective for $x_0 = \bar{x}_0$

$$\mathcal{J}(\boldsymbol{u}) = h(f^{[\tau]}(x_0, \boldsymbol{u})) = \sum_{t=1}^{\tau} h_t(x_t)$$
$$\text{s.t.} \quad x_{t+1} = f(x_t, u_t)$$



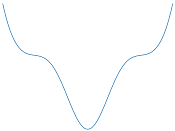
A Sufficient Condition for Global Convergence

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 dominated objective $\mathcal{J}$



Non-convex, gradient dominated function

Derivation:

- Here consider that the total cost h is e.g. μ-strongly convex s.t.

$$\|\nabla h(\boldsymbol{x})\|_2^2 \geq \mu(h(\boldsymbol{x}) - h^*)$$

- We have $\mathcal{J}(\boldsymbol{u}) = h(f^{[\tau]}(x_0, \boldsymbol{u}))$ so $\|\nabla \mathcal{J}(\boldsymbol{u})\|_2^2 = \|\nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u}) \nabla h(\boldsymbol{x})\|_2^2$
- So if $f^{[\tau]}(x_0, \boldsymbol{u})$ satisfies

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

then

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

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?

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,

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

→ Simply need to check that the dynamic has surj. linearizations

Problem:

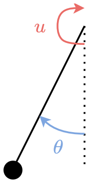
- Usually less control variables than state variables $\dim(u(t)) < \dim(x(t))$
  So $\sigma_{\min}(\nabla_u f(x(t), u(t)) > 0$ impossible

→ Use multistep schemes s.t. $\dim(u_t) = k \dim(u(t))$

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 $\omega = \dot{\theta}$

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

Linearization surjective? ✗

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

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

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

Overall Analysis

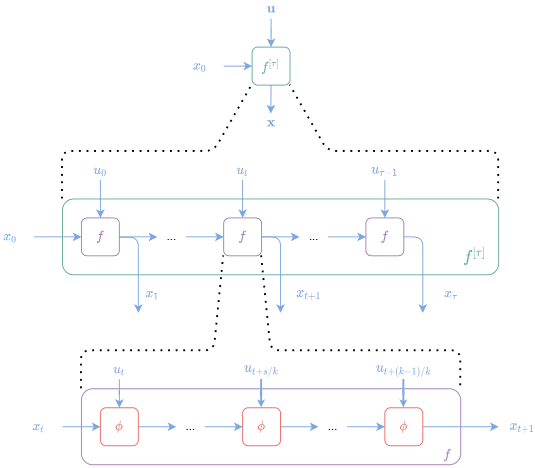
Multistep scheme

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

$\rightarrow$ study dynamical struct. of f itself

Control of a dynamic ϕ in k steps
for $\boldsymbol{v} = (v_0; \dots; v_{k-1})$,

$$\phi^{\{k\}}(y_0, \boldsymbol{v}) = y_k$$
$$\text{s.t.} \quad y_{s+1} = \phi(y_s, v_s)$$



Zooming in the dynamical structure

Sufficient condition for global convergence can be verified by analyzing whether ϕ can be *linearized by static feedback*

Linearization Scheme General Idea

Definition (Simplified see e.g. (Isidori 1995, Sontag 2013))

A dynamical system $y_{s+1} = \phi(y_s, v_s)$ is linearizable by static feedback if there exists some diffeomorphisms a and $b(y, \cdot)$ such that the reparam. system $z_s = a(y_s)$, $w_s = b(y_s, v_s)$ is linear, i.e., $z_{s+1} = Az_s + Bw_s$

Simple Example

- System driven by its d^{th} derivative (like acceleration in the pendulum example)

$$y_{t+1}^{(i)} = y_t^{(i)} + \Delta y_t^{(i+1)} \text{ for } i \in \{1, \ldots, d-1\}, \quad y_{t+1}^{(d)} = y_t^{(d)} + \Delta\psi(y_t, v_t)$$

s.t. $|\psi(y_t, v_t)| \neq 0$ for all y_t, v_t, with Δ the time step

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

If a d-dimensional system defined by $y_{t+1} = \phi(y_t, v_t)$ is linearizable by static feedback then its control in d steps $\phi^{\{d\}}(y, \boldsymbol{v})$ has surjective linearizations. Hence a control problem with dynamic $f = \phi^{\{d\}}$ and strongly convex costs h satisfy a gradient dominating property.

The problem could be solved by gradient descent
But the algorithms are not a gradient descent!

[1] Quantitative results available

Outline

Setup

Problem

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

Algorithm

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

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)

- 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!

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*
- $\delta = 1/(32\rho_h(\theta_h(1 + \sqrt{\rho_h}{\rho_g}^3/3) + \sqrt{\rho_h}\theta_g(1 + \rho_g\rho_h))^2)$ *is the gap of quadratic conv.*
- $\rho_h = L_h/\mu_h$ *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.*
- $\alpha = 4{\rho_g}^2(2{\rho_g}^2\theta_h/(3\theta_g) + \rho_h)$ *is another cond. nb*

[1] Extensions to self-concordant costs or gradient dominated costs available

Complexity Bound for IDDP

Idea
Analyze IDDP as an approximate ILQR similar as (Murray & Yakowitz 1984) for local conv.

Lemma (R. et al. (2022))

Under the aforementioned assumptions, denoting $\mathrm{DDP}_\nu(\mathcal{J})(\boldsymbol{u})$, $\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$ *the oracles returned by IDDP and ILQR resp., there exists* $\eta > 0$ *s.t.*

$$\forall \boldsymbol{u}, \nu \quad \| \mathrm{DDP}_\nu(\mathcal{J})(\boldsymbol{u}) - \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u}) \|_2 \leq \eta \| \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u}) \|_2^2$$

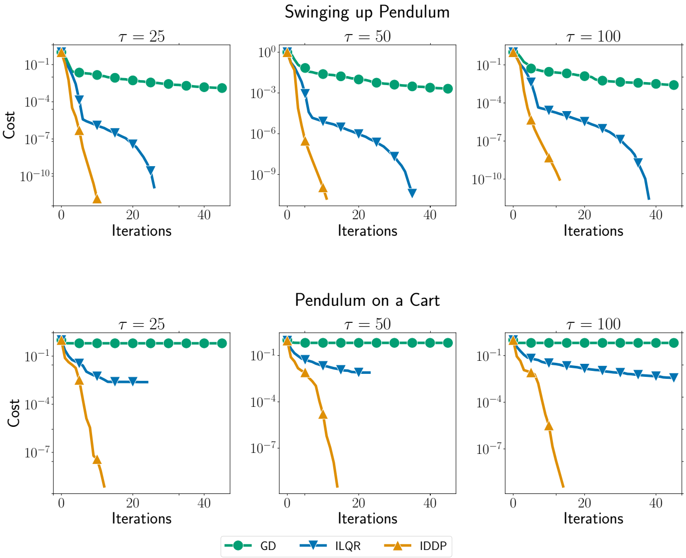
Theorem (R. et al. (2022))

Under the aforementioned assumptions, the IDDP algorithm equipped with appropriate regularization converges globally with a local quadratic rate.

Code Example from Toolbox ILQC

```python
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

# IDDP step
iddp_dir = roll_out_exact(traj, policies)
iddp_step = ctrls + iddp_dir

# Newton and DDP with quad. approx. also available
```

Numerical Illustrations

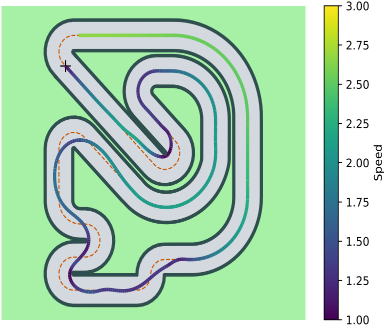


IDDP exploits differentiable programming but is not a classical GN method
Can we derive similar algorithms that exploit the problem structure?

Conclusion

Summary

- Conv. guarantees for canonical noncvx pb
  → analyze problem at elementary scale
  as done in a diff. prog. implementation
- Complexity bounds for ILQR and IDDP
  → quad. convergence at low iteration cost
  by using a diff. prog. implementation
- Generalized back-propagation as in IDDP
  → consider alternate sol. for oracle subpbs,
  use similar graph of computations



Model Predictive Control
& contouring objective

Thank you for your attention!

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.

Dunn, J. & Bertsekas, D. (1989), 'Efficient dynamic programming implementations of Newton's method for unconstrained optimal control problems', *Journal of Optimization Theory and Applications* **63**(1), 23–38.

Isidori, A. (1995), *Nonlinear Control Systems*, 3rd edn, Springer-Verlag.

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.

Liao, L.-Z. & Shoemaker, C. (1991), 'Convergence in unconstrained discrete-time differential dynamic programming', *IEEE Transactions on Automatic Control* **36**(6), 692–706.

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.

Murray, D. & Yakowitz, S. (1984), 'Differential dynamic programming and Newton's method for discrete optimal control problems', *Journal of Optimization Theory and Applications* **43**(3), 395–414.

Polak, E. (2011), 'On the role of optimality functions in numerical optimal control', *Annual Reviews in Control* **35**(2), 247–253.

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.

Sontag, E. (2013), *Mathematical control theory: deterministic finite dimensional systems*, Vol. 6, Springer Science & Business Media.

Tassa, Y., Erez, T. & Todorov, E. (2012), Synthesis and stabilization of complex behaviors through online trajectory optimization, *in* '2012 IEEE/RSJ International Conference on Intelligent Robots and Systems', pp. 4906–4913.

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.