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

Vincent Roulet[†], Siddhartha Srinivasa[‡], Maryam Fazel[⋄], Zaid Harchaoui[†]

[†] Department of Statistics University of Washington, Seattle, USA

[‡] Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle

[⋄] Department of Electrical and Computer Engineering, University of Washington, Seattle

Abstract

A classical approach for solving discrete time nonlinear control on a finite horizon consists in repeatedly minimizing linear quadratic approximations of the original problem around current candidate solutions. While widely popular in many domains, such an approach has mainly been analyzed locally. We observe that global convergence guarantees can be ensured provided that the linearized discrete time dynamics are surjective and costs on the state variables are strongly convex. We present how the surjectivity of the linearized dynamics can be ensured by appropriate discretization schemes given the existence of a feedback linearization scheme. We present complexity bounds of algorithms based on linear quadratic approximations through the lens of generalized Gauss-Newton methods. Our analysis uncovers several convergence phases for regularized generalized Gauss-Newton algorithms.

Keywords: discrete time nonlinear control, generalized Gauss-Newton

1 Introduction

We consider nonlinear control problems in discrete time of the form

$$\begin{aligned} \min_{\substack{u_0,\dots,u_{\tau-1}\in\mathbb{R}^{n_u}\\ x_0,\dots,x_\tau\in\mathbb{R}^{n_x}}} \quad & \sum_{t=1}^{\tau} h_t(x_t) \\ \text{subject to} \quad & x_{t+1} = f(x_t, u_t) \quad \text{for } t \in \{0,\dots,\tau-1\}, \qquad x_0 = \bar{x}_0, \end{aligned} \tag{1}$$

where at the time index t, x_t is the state of the system, u_t is the control applied to the system, f is the discretized nonlinear dynamic, h_t is the cost applied to the system state and $\bar{x}_0$ is a given fixed initial state. We do not consider costs on the control variables in this work.

Problems of the form (1) have been tackled in various ways, from direct approaches using nonlinear optimization (Betts, 2010; Wright, 1990, 1991; Pantoja, 1988; Dunn and Bertsekas, 1989; Rao et al., 1998) to convex relaxations using semidefinite optimization (Boyd and Vandenberghe, 1997). A popular approach of the former category proceeds by computing at each iteration the linear quadratic regulator associated to a linear quadratic approximation of the problem around the current candidate solutions (Jacobson and Mayne, 1970; Li and Todorov, 2007; Sideris and Bobrow, 2005; Tassa et al., 2012). The resulting feedback policy is then applied either on the linearized dynamics as in the Iterative Linear Quadratic Regulator (ILQR) algorithm (Li and Todorov, 2007), or on the original dynamics, leading to an Iterative Differential Dynamic Programming (IDDP) algorithm (Tassa et al., 2012).

Empirically, these approaches often exhibit fast convergence to efficient or optimal controllers which explain their popularity in applied control (Tassa et al., 2012; Giftthaler et al., 2018) and the renewed interest for linear quadratic control in neuro-dynamic programming and reinforcement learning (Fazel et al., 2018; Kakade et al., 2020; Simchowitz and Foster, 2020; Recht, 2019). The theoretical analysis of these methods mainly focused on local convergence guarantees (Mayne and Polak, 1975; Murray and Yakowitz, 1984; Liao and Shoemaker, 1991) or convergence guarantees to points satisfying first-order necessary optimality conditions (Polak, 2011). Given the generic success of ILQR or IDDP, we investigate *global* convergence guarantees and the overall computational complexities of these methods.

Our analysis stems from observing that, for strongly convex costs, global convergence of the ILQR or IDDP algorithms is ensured if the linearized dynamics, i.e., the mappings $v \mapsto \nabla_u f(x,u)^\top v$, are surjective, where $\nabla_u f(x,u)^\top$ is the Jacobian of the dynamic with respect to the control variable on a state x for a given control u. To quantify the convergence of the ILQR and IDDP algorithms, we consider the existence of a parameter σ such that

$$\forall x, u \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}, \quad \sigma_{\min}(\nabla_u f(x,u)) \geq \sigma > 0, \tag{2}$$

where $\sigma_{\min}(\nabla_u f(x,u)) = \inf_{\lambda \in \mathbb{R}^{n_x}} \|\nabla_u f(x,u)\lambda\|_2 / \|\lambda\|_2$ is the minimal singular value of the gradient of the discrete time dynamics f w.r.t. the control variable. Eq. (2) ensures the injectivity of $\lambda \mapsto \nabla_u f(x,u)\lambda$ which is equivalent to the surjectivity of $v \mapsto \nabla_u f(x,u)^\top v$. Our main theorem is then stated below for strongly convex costs provided adequate smoothness assumptions on the costs and the dynamics.

Theorem 1.1. *Consider costs h_t that are strongly convex with Lipschitz-continuous gradients and Lipschitz-continuous Hessians and a dynamic f that is Lipschitz-continuous with Lipschitz-continuous gradients. If the linearized dynamics are surjective, i.e., f satisfies (2), then a regularized ILQR or IDDP algorithm converges globally with a local quadratic convergence rate.*

Our analysis is based on decomposing the problem at several scales. At the scale of of the trajectory, the objective can be seen as the composition of a total cost function and a function, which, given a sequence of controls, outputs the corresponding trajectory. From an optimization viewpoint, the ILQR or the IDDP algorithms, which use linear quadratic approximations of the objective, amount then to generalized Gauss-Newton algorithms (Sideris and Bobrow, 2005; Diehl and Messerer, 2019; Messerer et al., 2021). One contribution of this work is then to detail the convergence rates of regularized generalized Gauss-Newton algorithms for the composition of an outer strongly convex function and an inner function with non-singular transpose Jacobians.

Both algorithms take advantage of the dynamical structure of the problem to implement a step of a Gauss-Newton algorithm. Similarly, the convergence guarantees for the ILQR or IDDP algorithms can be detailed using the properties of the problem at the scale of a single time step. In particular, condition (2) entails a simple condition on the dynamic to ensure global convergence.

Finally, the dynamic itself can further be decomposed at the scale of the discretization method used to define the discrete time control problem. Condition (2) may then be ensured by considering a multi-rate sampling method, i.e., sampling the control variables at a higher rate than the sampling of the costs on the state variables. By combining all aforementioned scales, we obtain worst-case convergence guarantees to global optima for the ILQR and IDDP algorithms.

1.1 Related Work

Convergence analysis of Gauss-Newton methods. Regularized Gauss-Newton methods, a.k.a. Levenberg-Marquardt methods (Moré, 1978), have been extensively studied (Yamashita and Fukushima, 2001; Fan and Yuan, 2005; Dan et al., 2002; Zhao and Fan, 2016; Bergou et al., 2020). Global convergence to stationary points at a polynomial rate is established in, e.g., (Bergou et al., 2020, Theorem 3.1). The results could potentially be extended, provided that the non-linear mappings have surjective Jacobians (Ueda and Yamashita, 2010, Corollary 2.1). Our approach improves on previous results with polynomial rates and our complexity bounds provide clear and explicit dependencies on the initial gap and the region of quadratic convergence. We also depart from previous results using error bounds, such as the ones of Bergou et al. (2020, Assumption 4.2) and Yamashita and Fukushima (2001, Eq. (1.6)), in that our assumption on surjective Jacobians is stronger than an error bound.

Closer to our approach is the work (Nesterov, 2007) where the assumption of surjective Jacobians is used to provide global convergence guarantees of a *modified* Gauss-Newton method also known as the prox-linear method (Drusvyatskiy and Paquette, 2019). In (Nesterov, 2007) the author argues in favor of least un-squared norms methods, as opposed to least squared norms methods, by reasoning in terms of condition numbers irrespective of local subroutine computational complexity. In the context of nonlinear control problems, regularized Gauss-Newton oracles can be implemented efficiently by exploiting the dynamical structure of the problem while modified Gauss-Newton method oracles may require a computationally expensive line-search. Finally, our work considers generic twice differentiable outer functions, i.e., analyzes *generalized* Gauss-Newton methods, for which we provide global convergence guarantees while previous work focused on local convergence guarantees (Messerer et al., 2021, Theorem 2.2). Moreover our results are quantitative, relating the region of quadratic convergence to the smallest singular value of the transposed Jacobian.

Convergence analysis of iterative differential dynamic programming. Differential dynamic programming algorithms developed by Jacobson and Mayne (1970); Mayne and Polak (1975) appear to be one of the first approaches for nonlinear discrete time control problems. A modern account is provided in the companion report (Roulet et al., 2022) for reference; see also (Liao and Shoemaker, 1992; Tassa et al., 2014). Previous work on the local quadratic convergence analysis of Differential Dynamic Programming (DDP) is based on viewing DDP as an approximate Newton method (Pantoja, 1988; Di and Lamperski, 2019). An alternative proof of local quadratic convergence (Liao and Shoemaker, 1991) and an approach based on the method of strong variations (Mayne and Polak, 1975) are also worth mentioning. Previous work (Pantoja, 1988; Di and Lamperski, 2019) considers additional costs on the control variables and assumes that the Hessian of the overall objective (1) is invertible; see (Pantoja, 1988, Theorem 4.1) or (Di and Lamperski, 2019, Assumption 2.2). In contrast to previous work, we do not consider additional costs on the control variable and we consider IDDP algorithms using linear quadratic approximations developed recently (Tassa et al., 2012).

Sufficient conditions for global convergence. Discrete time nonlinear control problems of the form (1) stem from the time discretization of continuous time problems. Necessary optimality conditions for the continuous time control problems are characterized by Pontryagin's maximum principle (Pontryagin et al., 1963). However, these optimality conditions cannot be used for the discretized problems since Pontryagin variations in finite dimensional space do not exist (Polak, 2011). Necessary optimality conditions can be derived from the Karush-Kuhn-Tucker conditions for problem (1), which are equivalent to first order optimality conditions of the objective in terms of control variables. Sufficient optimality conditions for the continuous time nonlinear control problem were also derived by Mangasarian (1966); Arrow (1968); Kamien and Schwartz (1971). We translate these conditions for the discrete time nonlinear control problem in the companion technical report (Roulet et al., 2022). Unfortunately, such conditions require convexity assumptions of implicitly defined functions that seem difficult to verify in practice. We argue in Sec. 3.2 that our assumption (2) can be verified on simple instances.

Our assumption is based on analyzing the gradient dominating property of the objective of problem (1) in terms of the properties of the dynamic. The gradient dominating property was introduced by Polyak (1964); Łojasiewicz (1963) as a sufficient condition to ensure global convergence of gradient descent. Here, we exploit this property to ensure global and local quadratic convergence of a regularized generalized Gauss-Newton algorithm. From a nonlinear control viewpoint, our assumption translates as the controllability of the linearized trajectories recently used to analyzed MPC controllers by Na and Anitescu (2020, Assumption 2) following Xu and Anitescu (2019). However, Xu and Anitescu (2019); Na and Anitescu (2020) state the assumption without linking it to the actual properties of the discrete time dynamic of interest. We formally relate our condition to feedback linearization schemes well understood in continuous time (Isidori, 1995; Sontag, 2013) and further developed in discrete time by Jakubczyk and Sontag (1990); Jakubczyk (1987); Jayaraman and Chizeck (1993); Aranda-Bricaire et al. (1996); Belikov et al. (2017). We exploit the existence of a feedback linearization scheme by considering a multi-rate sampling scheme as proposed in the early work of Grizzle and Kokotovic (1988) on discrete time feedback linearization schemes.

1.2 Outline

We start by presenting classical nonlinear control algorithms for problem (1), i.e., the Iterative Linear Quadratic Regulator (ILQR) and the Iterative Differential Dynamic Programming algorithms (IDDP), in Sec. 2, and cast them as closed-box oracles. We analyze the properties of problem (1) with respect to the dynamics f in terms of smoothness and surjectivity of the linearized dynamics in Sec. 3.1. We further decompose the properties of the dynamic f with respect to the underlying discretization scheme in Sec. 3.2. We analyze the convergence of the ILQR and IDDP algorithms by relating them to a generalized Gauss-Newton method in Sec. 4. In particular, in Sec. 4.2, we demonstrate the *global convergence* of the ILQR algorithm provided that the costs are gradient dominated, the dynamics have surjective linearizations (2) and both costs and dynamics are smooth. We show the *local quadratic convergence* of the ILQR algorithm provided that the costs are self-concordant, the dynamics have surjective linearizations (2) and both costs and dynamics satisfy appropriate smoothness conditions in Sec. 4.3. Theorem 1.1 is detailed for the ILQR algorithm in Sec. 4.4 and convergence of the IDDP algorithm is analyzed in Sec. 4.5. Appendix A presents a summary of the constants used in the paper, Appendix B presents helpers lemmas and Appendix C details some technical computations

Numerical illustrations of the ILQR and IDDP algorithms can be found in the code repository `https://github.com/vroulet/ilqc`, along with a technical report detailing the implementations and additional technical lemmas.

1.3 Notations

For a sequence of vectors $x_1, \dots, x_\tau \in \mathbb{R}^{n_x}$, we denote by semi-colons their concatenation s.t. $\boldsymbol{x} = (x_1; \dots; x_\tau) \in \mathbb{R}^{\tau n_x}$. For a multivariate function $f : \mathbb{R}^d \to \mathbb{R}^n$, we denote $\nabla f(x) = (\partial_{x_i} f_j(x))_{i\in\{1,\dots,d\} j\in\{1,\dots,n\}} \in \mathbb{R}^{d\times n}$ the gradient of f, i.e., the transpose of the Jacobian of f on x. For $f : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^n$, $x \in \mathbb{R}^d$, $y \in \mathbb{R}^p$, we denote $\nabla_x f(x, y) = (\partial_{x_i} f_j(x, y))_{i\in\{1,\dots,d\} j\in\{1,\dots,n\}} \in \mathbb{R}^{d\times n}$ the partial gradient of f w.r.t. x on (x, y). For $f : \mathbb{R}^d \to \mathbb{R}$, we denote $f^* = \min_{x\in\mathbb{R}^d} f(x)$. For $f : \mathbb{R}^d \to \mathbb{R}^n$, $h : \mathbb{R}^d \to \mathbb{R}$, and $x \in \mathbb{R}^d$, we denote the linear expansion of f around x and the quadratic expansion of h around x as, respectively,

$$\ell_f^x : y \to \nabla f(x)^\top y, \quad q_h^x : y \to \nabla h(x)^\top y + \frac{1}{2} y^\top \nabla^2 h(x) y.$$

For $f : \mathbb{R}^d \to \mathbb{R}^n$, we denote the Lipschitz-continuity constant of f as

$$l_f = \sup_{x,y\in\mathbb{R}^d, x\neq y} \|f(x)-f(y)\|_2 / \|x-y\|_2.$$

For $A \in \mathbb{R}^{d\times n}$, we denote by $\|A\|_2 = \sigma_{\max}(A) = \sup_{\lambda\in\mathbb{R}^n} \|A\lambda\|_2/\|\lambda\|_2$ and $\sigma_{\min}(A) = \inf_{\lambda\in\mathbb{R}^n} \|A\lambda\|_2/\|\lambda\|_2$ the largest and smallest singular values of A respectively.

2 Nonlinear Control Algorithms

The objective in (1) only depends on the control variables and can be written, for $\boldsymbol{u} = (u_0; \dots; u_{\tau-1}) \in \mathbb{R}^{\tau n_u}$, as

$$\begin{aligned} \mathcal{J}(\boldsymbol{u}) = \quad & \sum_{t=1}^{\tau} h_t(x_t) \\ \text{s.t. } & x_{t+1} = f(x_t, u_t), \quad \text{for } t \in \{0, \dots, \tau-1\}, \qquad x_0 = \bar{x}_0, \end{aligned} \tag{3}$$

such that problem (1) consists in minimizing $\mathcal{J}$. We consider algorithms that, given a command $\boldsymbol{u}$, computes the next iterate as $\boldsymbol{u}_{\text{next}} = \boldsymbol{u} + \text{Oracle}_\nu(\mathcal{J})(\boldsymbol{u})$, where $\text{Oracle}_\nu(\mathcal{J})$ is an oracle, which, given a regularization parameter ν, outputs a direction $\text{Oracle}_\nu(\mathcal{J})(\boldsymbol{u})$. The original ILQR or IDDP algorithms did not incorporate an additional regularization (Li and Todorov, 2007; Tassa et al., 2012). Our implementation is a variant that leads to convergence guarantees of these algorithms (Roulet et al., 2019).

2.1 Iterative Linear Quadratic Regulator

Given a command $\boldsymbol{u} = (u_0; \dots; u_{\tau-1})$ with associated trajectory $x_1, \dots, x_\tau$, and a regularization $\nu > 0$, an Iterative Linear Quadratic Regulator (ILQR) algorithm computes the next command by computing the Linear Quadratic Regulator (LQR) associated with a quadratic approximation of the costs and a linear approximation of the dynamics around the current trajectory. Formally, the next iterate is computed as $\boldsymbol{u}_{\text{next}} = \boldsymbol{u} + \text{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$, where

$$\begin{aligned} \text{LQR}_\nu(\mathcal{J})(\boldsymbol{u}) = & \operatorname*{arg\,min}_{v_0,\dots,v_{\tau-1}\in\mathbb{R}^{n_u}} \min_{y_0,\dots,y_\tau\in\mathbb{R}^{n_x}} \sum_{t=1}^{\tau} \frac{1}{2} y_t^\top P_t y_t + p_t^\top y_t + \frac{\nu}{2} \sum_{t=0}^{\tau-1} \|v_t\|_2^2 \\ & \text{s.t.} \quad y_{t+1} = A_t y_t + B_t v_t, \text{ for } t \in \{0, \dots, \tau-1\}, y_0 = 0, \\ \text{with} \quad & P_t = \nabla^2 h_t(x_t), \; p_t = \nabla h_t(x_t), \qquad \text{for } t \in \{1, \dots, \tau\}, \\ & A_t = \nabla_{x_t} f(x_t, u_t)^\top, \; B_t = \nabla_{u_t} f(x_t, u_t)^\top, \quad \text{for } t \in \{0, \dots, \tau-1\}. \end{aligned} \tag{4}$$

The implementation of the LQR oracle is presented in Algo. 1. Problem (4) is first instantiated in a *forward pass* by collecting all first order or second order information on the dynamics and the costs necessary to pose problem (4).

Problem (4) is then solved by dynamic programming (Bertsekas, 2017). Namely, the cost-to-go $c_t(y_t)$ from a state y_t at time t is computed recursively in a *backward pass* as, starting from $c_\tau(y_\tau) = \frac{1}{2} y_\tau^\top P_\tau y_\tau + p_\tau^\top y_\tau$,

$$\begin{aligned} c_t(y_t) = & \min_{v_t,\dots,v_{\tau-1}\in\mathbb{R}^{n_u}} \min_{z_t,\dots,z_\tau\in\mathbb{R}^{n_x}} \sum_{s=t}^{\tau} \frac{1}{2} z_s^\top P_s z_s + p_s^\top z_s + \frac{\nu}{2} \sum_{s=t}^{\tau-1} \|v_s\|_2^2 \\ & \text{s.t.} \quad z_{s+1} = A_s z_s + B_s v_s, \text{ for } s \in \{t,\dots,\tau-1\}, \quad z_t = y_t, \\ = & \min_{v_t\in\mathbb{R}^{n_u}} \left\{ \frac{1}{2} y_t^\top P_t y_t + p_t^\top y_t + \frac{\nu}{2}\|v_t\|_2^2 + c_{t+1}(A_t y_t + B_t v_t) \right\} = \frac{1}{2} y_t^\top J_t y_t + y_t^\top j_t, \end{aligned} \tag{5}$$

where J_t, j_t are computed recursively in Algo. 1. The optimal control at time t from state y_t is then given by an affine policy

$$\pi_t(y_t) = \operatorname*{arg\,min}_{v_t\in\mathbb{R}^{n_u}} \left\{ \frac{1}{2} y_t^\top P_t y_t + p_t^\top y_t + \frac{\nu}{2}\|v_t\|_2^2 + c_{t+1}(A_t y_t + B_t v_t) \right\} = K_t y_t + k_t, \tag{6}$$

where K_t, k_t are given in Algo. 1. The solution of the LQR problem (4), is computed by *rolling-out* the policies along the linear trajectories of (4). Namely, the oracle is given as $\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u}) = (v_0;\dots;v_{\tau-1})$, where

$$v_t = \pi_t(y_t), \quad y_{t+1} = A_t y_t + B_t v_t \quad \text{for } t \in \{0,\dots,\tau-1\},$$

starting from $y_0 = 0$. Solving (4) by dynamic programming comes at a linear cost with respect to the length of the trajectory. Namely, in terms of elementary computations, the LQR oracle has a computational cost

$$\mathcal{C}(n_x, n_u, \tau) = O(\tau(n_x + n_u)^3). \tag{7}$$

Note that, in nonlinear control problems, the state and control dimensions are generally small. On the other hand, the horizon τ may be large if, for example, for a fixed continuous time horizon, a small discretization stepsize was used to get (1). The ILQR algorithm keeps then *a linear complexity with respect to the leading dimension τ of the problem*.

Overall an ILQR algorithm computes a sequence of iterates as

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

starting from a command $\boldsymbol{u}^{(0)}$, where ν_k are regularization parameters that may depend on the current iterate and LQR_ν is implemented by Algo. 1.

2.2 Iterative Differential Dynamic Programming

A Differential Dynamic Programming (DDP) approach considers computing approximate solutions of (1) around the current iterate by dynamic programming using approximations of the dynamics and the costs. We refer the reader to, e.g., Jacobson and Mayne (1970); Tassa et al. (2012); Roulet et al. (2022) for a detailed presentation. The original DDP approach uses quadratic approximations of the dynamics Jacobson and Mayne (1970). Here, we focus on the implementation using linear approximations of the dynamics and quadratic approximations of the costs as used in, e.g., Tassa et al. (2012). In this case, a DDP approach amounts to computing the same policies π_t as an ILQR algorithm but rolling-out the policies along the original dynamics rather than the linearized ones Roulet et al. (2022).

Namely, the oracle output by a DDP approach is given as

$$\begin{aligned} & \mathrm{DDP}_\nu(\mathcal{J})(\boldsymbol{u}) = (v_0;\dots;v_{\tau-1}) \\ & \quad \text{where } v_t = \pi_t(y_t),\ y_{t+1} = f(x_t + y_t, u_t + v_t) - f(x_t, u_t) \text{ for } t \in \{0,\dots,\tau-1\}, \end{aligned} \tag{8}$$

as presented in Algo. 1. The computational complexity of this approach is the same as the one of the ILQR approach. By iterating the above steps, starting from an initial command $\boldsymbol{u}^{(0)}$, we obtain the Iterative Differential Dynamic Programming (IDDP) algorithm, which computes iterates of the form

$$\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \mathrm{DDP}_{\nu_k}(\mathcal{J})(\boldsymbol{u}^{(k)}), \tag{IDDP}$$

where the regularization parameters ν_k may depend on the current iterate and DDP_ν is implemented by Algo. 1.

Algorithm 1 ILQR and IDDP steps for problem (1)

Inputs: Command $\boldsymbol{u} = (u_0; \dots; u_{\tau-1}) \in \mathbb{R}^{\tau n_u}$, regularization $\nu > 0$, initial state $\bar{x}_0 \in \mathbb{R}^{n_x}$, horizon τ, dynamic $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$, costs $(h_t)_{t=1}^{\tau}$, oracle type $\mathrm{Oracle} \in \{\mathrm{LQR}, \mathrm{DDP}\}$.

Forward pass ▷ *instantiate problem (4) for the given command*

Initialize $x_0 = \bar{x}_0$

for $t = 0, \dots, \tau - 1$ **do**

  Compute $x_{t+1} = f(x_t, u_t)$ and $h_t(x_t)$

  Store $A_t = \nabla_{x_t} f(x_t, u_t)^\top$, $B_t = \nabla_{u_t} f(x_t, u_t)^\top$, $p_t = \nabla h_t(x_t)$, $P_t = \nabla^2 h_t(x_t)$.

end for

Backward pass ▷ *compute optimal policies for problem (4)*

Initialize $J_\tau = P_\tau$, $j_\tau = p_\tau$

for $t = \tau - 1, \dots 0$ **do**

  Compute the cost-to-go functions $c_t : y_t \to \frac{1}{2} y_t^\top J_t y_t + j_t^\top y_t$ defined in (5) as

$$J_t = P_t + A_t^\top J_{t+1} A_t - A_t^\top J_{t+1} B_t (\nu \mathrm{I} + B_t^\top J_{t+1} B_t)^{-1} B_t^\top J_{t+1} A_t$$
$$j_t = p_t + A_t^\top j_{t+1} - A_t^\top J_{t+1} B_t (\nu \mathrm{I} + B_t^\top J_{t+1} B_t)^{-1} B_t^\top j_{t+1}$$

  Store the policies $\pi_t : y_t \to K_t y_t + k_t$ defined in (6) as

$$K_t = -(\nu \mathrm{I} + B_t^\top J_{t+1} B_t)^{-1} B_t^\top J_{t+1} A_t, \quad k_t = -(\nu \mathrm{I} + B_t^\top J_{t+1} B_t)^{-1} B_t^\top j_{t+1}$$

end for

Roll-out pass ▷ *apply the computed policies along the linearized or the exact dynamics*

Initialize $y_0 = 0$

for $t = 0, \dots, \tau - 1$ **do**

  if Oracle is LQR **then**

    Compute $v_t = \pi_t(y_t),\ y_{t+1} = A_t y_t + B_t v_t$

  else if Oracle is DDP **then**

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

  end if

end for

Output: Command direction $\boldsymbol{v} = (v_0; \dots; v_{\tau-1})$

3 Conditioning Analysis

3.1 Objective Decomposition

The objective $\mathcal{J}$, defined in (3), can be decomposed into (i) the costs associated to a given trajectory, and (ii) the function that, given an input command, outputs the corresponding trajectory, defined below.

Definition 3.1. *We define the control* of τ *steps of a discrete time dynamic* $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ *as the function* $f^{[\tau]} : \mathbb{R}^{n_x} \times \mathbb{R}^{\tau n_u} \to \mathbb{R}^{\tau n_x}$*, which, given an initial point* $x_0 \in \mathbb{R}^{n_x}$ *and a command* $\boldsymbol{u} = (u_0; \dots; u_{\tau-1}) \in \mathbb{R}^{\tau n_u}$*, outputs the corresponding trajectory* $x_1, \dots, x_\tau$*, i.e.,*

$$f^{[\tau]}(x_0, \boldsymbol{u}) = (x_1; \dots; x_\tau) \qquad (9)$$
$$\textit{s.t.} \quad x_{t+1} = f(x_t, u_t) \qquad \textit{for } t \in \{0, \dots, \tau - 1\}.$$

By summarizing the costs as the function

$$h : \boldsymbol{x} = (x_1; \dots; x_\tau) \to \sum_{t=1}^{\tau} h_t(x_t),$$

problem (1) amounts to solving

$$\min_{\boldsymbol{u} \in \mathbb{R}^{\tau n_u}} \left\{ \mathcal{J}(\boldsymbol{u}) = h(f^{[\tau]}(\bar{x}_0, \boldsymbol{u})) \right\}. \qquad (10)$$

For convex costs h, if the dynamic f is linear, then the function $f^{[\tau]}$ is also linear and the overall problem (10) is then convex, hence easily solvable from an optimization viewpoint using, e.g., a gradient descent.

For nonlinear dynamics, the problem is a priori not convex regardless of the convexity of the costs. Yet, global convergence guarantees of, e.g., first order methods, may still be obtained by considering whether the objective satisfies a gradient dominating property (Polyak, 1964; Łojasiewicz, 1963), i.e., whether there exists, for example $m > 0$, such that for any $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$, $\|\nabla \mathcal{J}(\boldsymbol{u})\|_2^2 \geq m\left(\mathcal{J}(\boldsymbol{u}) - \mathcal{J}^*\right)$. To focus on the properties on the nonlinear dynamic, we consider costs that are gradient dominated, e.g., such that for any $\boldsymbol{x} \in \mathbb{R}^{\tau n_x}$, we have $\|\nabla h(\boldsymbol{x})\|_2^2 \geq \mu(h(\boldsymbol{x}) - h^*)$ for some $\mu > 0$. In that case, a sufficient condition for the objective to satisfy a gradient dominating property is that the control of τ steps of the dynamic satisfies $\sigma_{\min}(\nabla_{\boldsymbol{u}} f^{[\tau]}(\bar{x}_0, \boldsymbol{u})) \geq \sigma > 0$ for any $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$, since then we have, for $\boldsymbol{x} = f^{[\tau]}(\bar{x}_0, \boldsymbol{u})$,

$$\|\nabla \mathcal{J}(\boldsymbol{u})\|_2^2 = \|\nabla_{\boldsymbol{u}} f^{[\tau]}(\bar{x}_0, \boldsymbol{u}) \nabla h(\boldsymbol{x})\|_2^2 \geq \sigma^2 \|\nabla h(\boldsymbol{x})\|_2^2 \geq \sigma^2 \mu (h(\boldsymbol{x}) - h^*).$$

Provided that the set of minimizers of the objective is not empty, one easily verifies (see beginning of the proof of Theorem 4.2) that the above equation implies that $h^* = \mathcal{J}^*$ and so that the overall objective satisfies a gradient dominating property since $h(\boldsymbol{x}) = \mathcal{J}(\boldsymbol{u})$. We investigate then whether the control of τ steps of a dynamic f can satisfy the aforementioned condition by considering the properties of the dynamic f.

The condition $\sigma_{\min}(\nabla_{\boldsymbol{u}} f^{[\tau]}(\bar{x}_0, \boldsymbol{u})) > 0$ can be interpreted as the surjectivity of the linearized control of τ steps, i.e., the mapping $\boldsymbol{v} = (v_0; \dots; v_{\tau-1}) \to \nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})^\top \boldsymbol{v} = (y_1; \dots; y_\tau)$ which can naturally be decomposed as

$$y_{t+1} = \nabla_{x_t} f(x_t, u_t)^\top y_t + \nabla_{u_t} f(x_t, u_t)^\top v_t \quad \text{for } t \in \{0, \dots, \tau - 1\}, \quad y_0 = 0.$$

We recognize here the linearized trajectories that are at the heart of the ILQR and IDDP algorithms. Our analysis stems from understanding that the surjectivity of the linearization of the control of τ steps, i.e, $\boldsymbol{v} \to \nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})^\top \boldsymbol{v}$, is inherited from the surjectivity of the linearization of a single step of the discrete dynamic, i.e., $v \to \nabla_u f(x, u)^\top v$ as formally stated in the following lemma.

Lemma 3.2. *If the linearized dynamics, $v \to \nabla_u f(x, u)^\top v$, of a Lipschitz continuous discrete time dynamic f are surjective in the sense that there exists $\sigma_f > 0$ s.t.*

$$\forall x, u \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}, \quad \sigma_{\min}(\nabla_u f(x, u)) \geq \sigma_f > 0, \tag{11}$$

then the linearizations, $\boldsymbol{v} \to \nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})^\top \boldsymbol{v}$, of the control of τ steps of the dynamic f is also surjective, namely,

$$\forall x_0, \boldsymbol{u} \in \mathbb{R}^{n_x} \times \mathbb{R}^{\tau n_u}, \quad \sigma_{\min}(\nabla_u f^{[\tau]}(x_0, \boldsymbol{u})) \geq \sigma_{f^{[\tau]}} := \frac{\sigma_f}{1 + l_f^x} > 0, \tag{12}$$

where $l_f^x = \sup_{u \in \mathbb{R}^{n_u}} l_{f(\cdot, u)}$ is the maximal Lipschitz-continuity constant of the functions $f(\cdot, u)$ for any $u \in \mathbb{R}^{n_u}$.

Proof. Fix $x_0 \in \mathbb{R}^{n_x}$. Given a sequence of controls $\boldsymbol{u} = (u_0; \dots; u_{\tau-1}) \in \mathbb{R}^{\tau n_u}$ with corresponding trajectory $\boldsymbol{x} = (x_1; \dots; x_\tau) = f^{[\tau]}(x_0, \boldsymbol{u}) \in \mathbb{R}^{\tau n_x}$, and $\boldsymbol{\mu} = (\mu_1; \dots; \mu_\tau) \in \mathbb{R}^{\tau n_x}$, the gradient vector product $\nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})\boldsymbol{\mu}$ is written

$$\begin{aligned} \nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})\boldsymbol{\mu} = (&\nabla_{u_0} f(x_0, u_0)\lambda_1; \dots; \nabla_{u_{\tau-1}} f(x_{\tau-1}, u_{\tau-1})\lambda_\tau) \\ \text{s.t.} \quad &\lambda_t = \nabla_{x_t} f(x_t, u_t)\lambda_{t+1} + \mu_t \quad \text{for } t \in \{1, \dots, \tau - 1\}, \quad \lambda_\tau = \mu_\tau. \end{aligned}$$

For $\boldsymbol{x}$=$(x_1; \dots; x_\tau)$, $\boldsymbol{u}$=$(u_0; \dots; u_{\tau-1})$, define the function $F(\boldsymbol{x}, \boldsymbol{u})$=$(f(x_0, u_0); \dots; f(x_{\tau-1}, u_{\tau-1}))$. By using the upper block diagonal structure of $\nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u})$, we have

$$\begin{aligned} (\mathrm{I} - \nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u}))^{-1}\boldsymbol{\mu} = &(\lambda_1; \dots; \lambda_\tau) \\ \text{s.t.} \quad &\lambda_t = \nabla f_{x_t}(x_t, u_t)\lambda_{t+1} + \mu_t \quad \text{for } t \in \{1, \dots, \tau - 1\}, \quad \lambda_\tau = \mu_\tau. \end{aligned}$$

The gradient vector product can then be written compactly as

$$\nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})\boldsymbol{\mu} = \nabla_{\boldsymbol{u}} F(\boldsymbol{x}, \boldsymbol{u})(\mathrm{I} - \nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u}))^{-1}\boldsymbol{\mu}.$$

Hence, for any command $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$ and any $x_0 \in \mathbb{R}^{n_x}$,

$$\sigma_{\min}(\nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})) \geq \frac{\sigma_{\min}(\nabla_{\boldsymbol{u}} F(\boldsymbol{x}, \boldsymbol{u}))}{\sigma_{\max}(\mathrm{I} - \nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u}))} \geq \frac{\sigma_f}{1 + l_f^x}.$$

□

Similarly, the smoothness properties of the control $f^{[\tau]}$ corresponding to dynamics f can be expressed in terms of the smoothness properties of the dynamics f (Roulet et al., 2022, Lemma 3.3). Namely, if f is Lipschitz-continuous with Lipschitz-continuous gradients, then the function $\boldsymbol{u} \to f^{[\tau]}(x_0, \boldsymbol{u})$ is $l_{f^{[\tau]}}$-Lipschitz-continuous and has $L_{f^{[\tau]}}$-Lipschitz-continuous gradients with

$$l_{f^{[\tau]}} \leq l_f^u S, \qquad L_{f^{[\tau]}} \leq S(L_f^{xx} S (l_{f^{[\tau]}})^2 + 2L_f^{xu} l_{f^{[\tau]}} + L_f^{uu}), \tag{13}$$

with $S = \sum_{t=0}^{\tau-1} (l_f^x)^t$, and $l_f^u = \sup_{x \in \mathbb{R}^{n_x}} l_{f(x,\cdot)}$, $L_f^{xx} = \sup_{u \in \mathbb{R}^{n_u}} l_{\nabla_x f(\cdot,u)}$, $L_f^{uu} = \sup_{x \in \mathbb{R}^{n_x}} l_{\nabla_u f(x,\cdot)}$, $L_f^{xu} = \sup_{x \in \mathbb{R}^{n_x}} l_{\nabla_u f(\cdot,u)}$ are maximal Lipschitz-continuity constants of partial functions or gradients of the dynamics.

3.2 Dynamic Decomposition

We have isolated condition (2) as a sufficient condition to ensure global convergence of, e.g., a gradient descent. It remains to consider whether this assumption can be satisfied on concrete examples. Note that assumption (2) requires $n_u \geq n_x$. While the underlying continuous control problem may have less control variables than state variables, by considering multiple steps of a simple Euler discretization method, we may still ensure the validity of assumption (2).

Formally, in this section, we assume that the discrete time dynamic f can be further decomposed as the control *in* k steps of some elementary discrete time dynamic ϕ as defined below. Concretely, ϕ may correspond to a single Euler discretization step of some continuous time dynamic. The discrete time dynamic f amounts then to k steps of such Euler discretization scheme and can be formulated as $f(x_t, u_t) = \phi^{\{k\}}(x_t, u_t)$, for some $k \geq 0$. On the other hand, we consider the costs to be computed only at the scale of the dynamic f, i.e., the sampling of the costs and the sampling of the dynamics differ, hence the terminology multi-rate sampling.

Definition 3.3. *We define the control* in k *steps of a discrete time dynamic* $\phi : \mathbb{R}^{n_x} \times \mathbb{R}^{m_u} \to \mathbb{R}^{n_x}$ *as the function* $\phi^{\{k\}} : \mathbb{R}^{n_x} \times \mathbb{R}^{k m_u} \to \mathbb{R}^{n_x}$*, which, given a state* y_0 *and a sequence of controls* $\boldsymbol{v} = (v_0; \dots; v_{k-1})$*, outputs the state computed after* k *steps, i.e.,*

$$\begin{aligned} &\phi^{\{k\}}(y_0, \boldsymbol{v}) = y_k \\ &\quad s.t. \quad y_{s+1} = \phi(y_s, v_s) \quad \text{for } s \in \{0, \dots, k-1\}. \end{aligned} \tag{14}$$

Our goal is then to know whether, by considering enough steps of ϕ, we can ensure the surjectivity of the linearized dynamic $\boldsymbol{w} \mapsto \nabla_{\boldsymbol{v}} \phi^{\{k\}}(y_0, \boldsymbol{v})^\top \boldsymbol{w}$. To build some intuition, consider a system driven by its acceleration such that the state of the system is determined by the position and the velocity ($n_x = 2$) and the control is a scalar force ($m_u = 1$) determining the acceleration, hence controlling effectively the speed at each time-step. For such system, the state of the system cannot be fully determined in one step of an Euler discretization scheme, as only the velocity is affected by the control. However, in two steps we can control both the position and the velocity, hence we may satisfy assumption (2). To formalize and generalize this intuition, we consider the availability of a feedback linearization scheme as defined below (adapted from Aranda-Bricaire et al. (1996)).

Definition 3.4. *A discrete time system defined by* $y_{t+1}{=}\phi(y_t, v_t)$ *with* $y_t \in \mathbb{R}^{n_x}, v_t \in \mathbb{R}^{m_u}$ *can be linearized by static feedback if there exist diffeomorphisms* $a : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ *and* $b(y, \cdot) : \mathbb{R}^{m_u} \to \mathbb{R}^{m_u}$ *such that the reparameterization of the system as* $z_t = a(x_t)$, $w_t = b(v_t, y_t)$ *is linear: there exists* m_u *indexes* $r_1, \dots, r_{m_u}$ *with* $\sum_{j=1}^{m_u} r_j = n_x$ *such that* z *can be decomposed as* $z_t = (\zeta_{t,1}; \dots; \zeta_{t,m_u})$ *with* $\zeta_{t,j} \in \mathbb{R}^{r_j}$ *decomposed as*

$$\zeta_{t+1,j}^{(i)} = \zeta_{t,j}^{(i+1)} \text{ for all } i \in \{1, \dots, r_j - 1\}, \quad \zeta_{t+1}^{(r_j)} = w_t^{(j)}, \text{ for all } j \in \{1, \dots, m_u\}.$$

For single-input system ($m_u = 1$, $r = n_x$), the reparameterized system takes the canonical Brunovsky form (Brunovskỳ, 1970)

$$z_{t+1}^{(i)} = z_t^{(i+1)} \text{ for all } i \in \{1, \dots, n_x - 1\}, \quad z_{t+1}^{(n_x)} = w_t, \tag{15}$$

i.e., $z_{t+1} = D z_t + w_t e$, where $D = \sum_{i=1}^{n_x - 1} e_i e_{i+1}^\top$ the upper shift matrix in $\mathbb{R}^{n_x}$ with e_i the i^{th} canonical vector in $\mathbb{R}^{n_x}$, such that $(Dz)^{(i)} = z^{(i+1)}$, and $e = e_{n_x}$.

As a concrete example, consider the Euler discretization with stepsize $\Delta > 0$ of a single input continuous time system driven by its $n_x{}^{\text{th}}$ derivative through a real differentiable function ψ

$$y_{t+1}^{(i)} = y_t^{(i)} + \Delta y_t^{(i+1)}, \text{ for all } i \in \{1, \dots, n_x - 1\}, \quad y_{t+1}^{(n_x)} = y_t^{(n_x)} + \Delta \psi(y_t, v_t). \tag{16}$$

As shown in Lemma B.1, such system can easily be reparameterized in the Brunovsky form (15) and provided that $|\partial_v \psi(y,v)| > 0$ for all $y \in \mathbb{R}^{n_x}, v \in \mathbb{R}$, we can have access to a feedback linearization scheme, i.e., we can reparameterize the system using diffeomorphisms.

The canonical representation (15) has the advantage to illuminate why the surjectivity of the linearized dynamics may hold by taking enough steps as it is clear that, in the representation (15), by controlling the system in n_x steps we directly control the output. Namely, we have that $z_{n_x}^{(i)} = w_{i-1}$ for all $i \in \{1, \dots, n_x\}$. So for the system (15), considering n_x steps ensures condition (2). The following theorem shows that this property is kept when considering the original system.

Theorem 3.5. *If a discrete time system $y_{t+1} = \phi(y_t, v_t)$ is linearizable by static feedback as defined in Def. 3.4, then $\phi^{\{k\}}$, the control in $k \geq r = \max\{r_1, \dots, r_{m_u}\}$ steps of ϕ, has surjective linearizations, i.e., it satisfies $\sigma_{\min}(\nabla_{\boldsymbol{v}} \phi^{\{k\}}(y_0, \boldsymbol{v})) > 0$ for any $y_0 \in \mathbb{R}^{n_x}, \boldsymbol{v} \in \mathbb{R}^{km_u}$.*

Quantitatively, if the system defined by $y_{t+1} = \phi(y_t, v_t)$ is linearizable by static feedback with transformations a and b that are Lipschitz-continuous and such that

$$\inf_{y \in \mathbb{R}^{n_x}} \sigma_{\min}(\nabla a(y)) \geq \sigma_a > 0, \qquad \inf_{y \in \mathbb{R}^{n_x}, v \in \mathbb{R}^{m_u}} \sigma_{\min}(\nabla_v b(y,v)) \geq \sigma_b > 0,$$

then the control in $k \geq r$ steps of the dynamic ϕ satisfies, for $l_b^y = \sup_{v \in \mathbb{R}^{m_u}} l_{b(\cdot,v)}$,

$$\inf_{y_0 \in \mathbb{R}^{n_x}, \boldsymbol{v} \in \mathbb{R}^{km_u}} \sigma_{\min}(\nabla_{\boldsymbol{v}} \phi^{\{k\}}(y_0, \boldsymbol{v})) \geq \frac{\sigma_b}{l_a} \frac{1}{1 + (r-1) l_b^y / \sigma_a} > 0.$$

Proof. We present the main steps of the proof, additional technical details are provided in the Appendix C. We detail first the single-input case described in (15), i.e., $m_u = 1$ and $r = n_x$. Moreover we consider first $k = n_x$. Let $\boldsymbol{v} = (v_0; \dots; v_{k-1}) \in \mathbb{R}^k$ and denote $y_k = \phi^{\{k\}}(y_0, \boldsymbol{v})$. In the reparameterization of the system in the form (15), we have that $z_k^{(i)} = w_{i-1}$ for all $i \in \{1, \dots, n_x\}$. Denoting for y_0 fixed, $\boldsymbol{y} = (y_1; \dots; y_k)$ and $\boldsymbol{v} = (v_0; \dots; v_{k-1})$, the function $B(\boldsymbol{y}, \boldsymbol{v}) = (b(y_0, v_0); \dots; b(y_{k-1}, v_{k-1})) \in \mathbb{R}^{n_x}$, we have $\phi^{\{k\}}(y_0, \boldsymbol{v}) = a^{-1}(B(\phi^{[k]}(y_0, \boldsymbol{v}), \boldsymbol{v}))$, where $\phi^{[k]}(y_0, \boldsymbol{v})$ denotes the control *of* k steps of ϕ. Hence, denoting $\boldsymbol{y} = (y_1; \dots; y_k) = \phi^{[k]}(y_0, \boldsymbol{v})$, we have

$$\nabla_{\boldsymbol{v}} \phi^{\{k\}}(y_0, \boldsymbol{v}) = \left(\nabla_{\boldsymbol{v}} B(\boldsymbol{y}, \boldsymbol{v}) + \nabla_{\boldsymbol{v}} \phi^{[k]}(y_0, \boldsymbol{v}) \nabla_{\boldsymbol{y}} B(\boldsymbol{y}, \boldsymbol{v}) \right) \nabla a(y_k)^{-1}.$$

Since $\nabla_{\boldsymbol{y}} B(\boldsymbol{y}, \boldsymbol{v})$ is strictly upper block block triangular, $\nabla_{\boldsymbol{v}} \phi^{[k]}(y_0, \boldsymbol{v})$ is upper block triangular, $\nabla_{\boldsymbol{v}} B(\boldsymbol{y}, \boldsymbol{v})$ is diagonal with non-zero entries, we have that $\nabla_{\boldsymbol{v}} \phi^{\{k\}}(y_0, \boldsymbol{v})$ is invertible

Now, consider $s = k - n_x > 0$. Denote $v_{a:b} = (v_a; \dots; v_b)$ for $a < b$. Let $y_0 \in \mathbb{R}^{n_x}$ and $\boldsymbol{v} = (v_0; \dots; v_{k-1}) \in \mathbb{R}^k$. We have $\phi^{\{k\}}(y_0, \boldsymbol{v}) = \phi^{\{n_x\}}(\phi^{\{s\}}(y_0, v_{0:s-1}), v_{s:k-1})$. Hence, denoting $y_s = \phi^{\{s\}}(y_0, v_{0:s-1})$, we have

$$\nabla_v \phi^{\{k\}}(y_0, \boldsymbol{v}) = \begin{pmatrix} \nabla_{v_{0:s-1}} \phi^{\{s\}}(y_0, v_{0:s-1}) \nabla_{y_s} \phi^{\{n_x\}}(y_s, v_{s:k-1}) \\ \nabla_{v_{s:k-1}} \phi^{\{n_x\}}(y_s, v_{s:k-1}) \end{pmatrix}. \tag{17}$$

The function $y_s, v_{s:k-1} \to \phi^{\{n_x\}}(y_s, v_{s:k-1})$ amounts to the control of ϕ in $k - s = n_x$ steps. Hence the matrix $\nabla_{v_{s:k-1}} \phi^{\{n_x\}}(y_s, v_{s:k-1})$ is invertible, so $\nabla_{\boldsymbol{v}} \phi^{\{k\}}(y_0, \boldsymbol{v})$ has full column rank. Overall, we showed the first part of the claim, i.e., that $\sigma_{\min}(\nabla_{\boldsymbol{v}} \phi^{\{k\}}(y_0, \boldsymbol{v})) > 0$ for any $y_0 \in \mathbb{R}^{n_x}, \boldsymbol{v} \in \mathbb{R}^{km_u}$, provided that $k \geq n_x$.

We consider now deriving quantitative bounds. We start with $k = n_x$ and focus on the single-input case. Define, for y_0 fixed, $\boldsymbol{y} = (y_1; \dots; y_k)$, $\boldsymbol{v} = (v_0; \dots; v_{k-1})$, the function $\Phi(\boldsymbol{y}, \boldsymbol{v}) = (\phi(y_0, v_0); \dots; \phi(y_{k-1}, v_{k-1}))$. By decomposing $\nabla_{\boldsymbol{v}} \phi^{[k]}(y_0, \boldsymbol{v})$ as in Lemma 3.2, we get

$$\nabla_{\boldsymbol{v}} \phi^{\{k\}}(y_0, \boldsymbol{v}) = \left(\nabla_{\boldsymbol{v}} B(\boldsymbol{y}, \boldsymbol{v}) + \nabla_{\boldsymbol{v}} \Phi(\boldsymbol{y}, \boldsymbol{v}) (\mathrm{I} - \nabla_{\boldsymbol{y}} \Phi(\boldsymbol{y}, \boldsymbol{v}))^{-1} \nabla_{\boldsymbol{y}} B(\boldsymbol{y}, \boldsymbol{v}) \right) \nabla a(y_k)^{-1}.$$

Given the feedback linearization scheme, the discrete time dynamic ϕ can be rewritten as $y_{t+1} = \phi(y_t, v_t) = a^{-1}(Da(y_t) + b(y_t, v_t)e)$, where D is the upper shift matrix in $\mathbb{R}^{n_x}$ and $e = e_{n_x}$ is the $n_x{}^{\text{th}}$ canonical vector in $\mathbb{R}^{n_x}$. Hence, we have for $t \in \{0, \dots, k-1\}$,

$$\begin{aligned} \nabla_{v_t} \phi(y_t, v_t) &= \partial_{v_t} b(y_t, v_t) e^\top \nabla a(y_{t+1})^{-1} \\ \nabla_{y_t} \phi(y_t, v_t) &= \left(\nabla a(y_t) D^\top + \nabla_{y_t} b(y_t, v_t) e^\top \right) \nabla a(y_{t+1})^{-1}. \end{aligned}$$

In the sequel, we denote the Kronecker product by $\otimes$ and for $R_1, \dots, R_n \in \mathbb{R}^{p\times q}$ we denote by $\mathbf{diag}((R_i)_{i=1}^n) = \sum_{i=1}^n e_i e_i^\top \otimes R_i \in \mathbb{R}^{np\times nq}$ the block diagonal matrix with blocks $R_1, \dots, R_n$, for e_i the i^{th} canonical vector in $\mathbb{R}^n$. Since $\nabla_{\boldsymbol{v}}\Phi(\boldsymbol{y},\boldsymbol{v}) = \mathbf{diag}((\nabla_{v_t}\phi(y_t,v_t))_{t=0}^{k-1})$, $\nabla_{\boldsymbol{y}}\Phi(\boldsymbol{y},\boldsymbol{v}) = (D\otimes \mathrm{I})\,\mathbf{diag}(\nabla_{y_t}\phi(y_t,v_t)_{t=0}^{k-1})$, we have that

$$\begin{aligned}
\nabla_{\boldsymbol{v}}\Phi(\boldsymbol{y},\boldsymbol{v}) &= \mathbf{diag}((\partial_{v_t}b(y_t,v_t))_{t=0}^{k-1})(\mathrm{I}\otimes e^\top)\,\mathbf{diag}((\nabla a(y_{t+1})^{-1})_{t=0}^{k-1})\\
\nabla_{\boldsymbol{y}}\Phi(\boldsymbol{y},\boldsymbol{v}) &= (D\otimes \mathrm{I})\,\mathbf{diag}((\nabla a(y_t))_{t=0}^{k-1})(\mathrm{I}\otimes D^\top)\,\mathbf{diag}((\nabla a(y_{t+1})^{-1})_{t=0}^{k-1})\\
&\quad + (D\otimes \mathrm{I})\,\mathbf{diag}((\nabla_{y_t}b(y_t,v_t))_{t=0}^{k-1})(\mathrm{I}\otimes e^\top)\,\mathbf{diag}((\nabla a(y_{t+1})^{-1})_{t=0}^{k-1}),
\end{aligned} \tag{18}$$

and, similarly, $\nabla_{\boldsymbol{v}}B(\boldsymbol{y},\boldsymbol{v}) = \mathbf{diag}((\partial_{v_t}b(y_t,v_t))_{t=0}^{k-1})$, $\nabla_{\boldsymbol{y}}B(\boldsymbol{y},\boldsymbol{v}) = (D\otimes \mathrm{I})\,\mathbf{diag}((\nabla_{y_t}b(y_t,v_t))_{t=0}^{k-1})$. Denoting $A = \mathbf{diag}((\nabla a(y_t))_{t=0}^{k-1})$, $C = \mathbf{diag}((\nabla a(y_{t+1}))_{t=0}^{k-1})$, $V = \mathbf{diag}((\partial_{v_t}b(y_t,v_t))_{t=0}^{k-1})$, $Y = \mathbf{diag}((\nabla_{y_t}b(y_t,v_t))_{t=0}^{k-1})$, $E = \mathrm{I}\otimes e^\top$, $F = D\otimes \mathrm{I}$ and $G = \mathrm{I}\otimes D^\top$, we get that

$$\begin{aligned}
\nabla_{\boldsymbol{v}}\phi^{\{k\}}(y_0,\boldsymbol{v})\nabla a(y_k) &= V(\mathrm{I} + EC^{-1}(\mathrm{I} - FAGC^{-1} - FYEC^{-1})^{-1}FY)\\
&\overset{(i)}{=} V(\mathrm{I} - EC^{-1}(\mathrm{I} - FAGC^{-1})^{-1}FY)^{-1},\\
&\overset{(ii)}{=} V(\mathrm{I} - E(\mathrm{I} - FG)^{-1}FA^{-1}Y)^{-1},\\
&\overset{(iii)}{=} V\left(\mathrm{I} - E\left(\sum_{i=1}^{k-1} D^i \otimes (D^\top)^{i-1}\right)A^{-1}Y\right)^{-1},
\end{aligned} \tag{19}$$

where in (i) we used the Sherman-Morrison-Woodbury identity, in (ii) we used that $FA = CF$ and $FA^{-1} = C^{-1}F$, in (iii) we used that $FG = D\otimes D^\top$ is nilpotent of order $k = n_x$ since $D^k = 0$. The result follows for $k = n_x$ from the assumptions. For $k > n_x$, we have from (17), that for any $\lambda \in \mathbb{R}^{n_x}$, $\|\nabla_v \phi^{\{k\}}(y_0,\boldsymbol{v})\lambda\|_2 \geq \|\nabla_{v_{s:k-1}}\phi^{\{n_x\}}(y_s, v_{s:k-1})\lambda\|_2$, hence the result follows.

For multi-input systems, let $r = \max\{r_1,\dots,r_{m_u}\}$. One easily verifies that for any $k \geq r$, the system in its linear representation can be written as $z_k = M\boldsymbol{w}$ for $\boldsymbol{w} = (w_0;\dots;w_{k-1})$ with $\sigma_{\min}(M^\top) = 1$. The first part of the claim follows then as in single input case. For the second part, the system can be decomposed by blocks and treated as in the single-input case. □

Overall, Theorem 3.5 shows that for, e.g., a dynamical system driven by its k^{th} derivative as in (16), by considering a dynamic f defined by k steps of an Euler discretization of the system, condition (2) can be ensured, which in turns can ensure a gradient dominating property for the objective. The ILQR and IDDP algorithms are not just gradient descent algorithms. It remains now to exploit assumption (2) to uncover the efficiency of the ILQR or IDDP algorithms.

4 Convergence Analysis

To analyze the convergence of the ILQR and the IDDP algorithms, we consider problem (1) at the scale of the whole trajectory and analyze problem (1) as a compositional problem of the form

$$\min_{\boldsymbol{u}\in\mathbb{R}^{\tau n_u}} \{\mathcal{J}(\boldsymbol{u}) = h(g(\boldsymbol{u}))\}, \text{ where } g(\boldsymbol{u}) = f^{[\tau]}(\bar{x}_0,\boldsymbol{u}) \text{ and } h(\boldsymbol{x}) = \sum_{t=1}^{\tau} h_t(x_t). \tag{20}$$

Note however that the dynamical structure of the problem revealed at the state scale is essential to the implementation of the ILQR and IDDP algorithms. We state our assumptions for global convergence at the state scale and translate them at the trajectory scale. A table of all constants introduced for the convergence analysis with their respective units is provided in Appendix A for ease of reference.

Hypothesis 4.1. *We consider convex costs h_t that have L_h-Lipschitz-continuous gradients and M_h-Lipschitz-continuous Hessians for all $t \in \{1,\dots,\tau\}$. We consider the dynamics to be Lipschitz-continuous with Lipschitz continuous gradients and satisfying (11).*

In consequence, the total cost h defined in (20) is convex, has L_h-Lipschitz-continuous gradients and M_h-Lipschitz-continuous Hessians. The function g defined in (20) is l_g-Lipschitz-continuous with L_g-Lipschitz-continuous gradients satisfying

$$\forall \boldsymbol{u} \in \mathbb{R}^{\tau n_u}, \quad \sigma_{\min}(\nabla g(\boldsymbol{u})) \geq \sigma_g > 0, \tag{21}$$

where $l_g = l_{f^{[\tau]}}$, $L_g = L_{f^{[\tau]}}$ are given in (13) and $\sigma_g = \sigma_{f^{[\tau]}}$ is given in (12).

4.1 Main ideas

The ILQR algorithm is a generalized Gauss-Newton algorithm. From a high-level perspective, the ILQR algorithm consists in linearizing the function $g : \boldsymbol{u} \to f^{[\tau]}(\bar{x}_0, \boldsymbol{u})$ that encapsulates the dynamics, taking a quadratic approximation of the costs h around the current trajectory $\boldsymbol{x} = g(\boldsymbol{u})$ and minimizing the resulting approximation with an additional regularization. Formally, as previously observed by Sideris and Bobrow (2005); Roulet et al. (2019), the ILQR algorithm is then computing

$$\begin{aligned}\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u}) &= \operatorname*{arg\,min}_{\boldsymbol{v}\in\mathbb{R}^{\tau n_u}} q_h^{g(\boldsymbol{u})}(\ell_g^{\boldsymbol{u}}(\boldsymbol{v})) + \frac{\nu}{2}\|\boldsymbol{v}\|_2^2 \\ &= -(\nabla g(\boldsymbol{u})\nabla^2 h(g(\boldsymbol{u}))\nabla g(\boldsymbol{u})^\top + \nu\,\mathrm{I})^{-1}\nabla g(\boldsymbol{u})\nabla h(g(\boldsymbol{u})),\end{aligned} \tag{22}$$

where $\ell_g^{\boldsymbol{u}}$ and $q_h^{g(\boldsymbol{u})}$ are the linear and quadratic approximations of, respectively, the control in τ steps around $\boldsymbol{u}$ and the total costs around $g(\boldsymbol{u})$ as defined in the notations. Equation 22 clearly reveals that the ILQR algorithm amounts to a regularized generalized Gauss-Newton algorithm Diehl and Messerer (2019). However, while a naive implementation of a generalized Gauss-Newton method would have a computational complexity per step of the order of $O((\tau n_u)^3)$, i.e., a cubic complexity with respect to τ the ILQR and the IDDP algorithms have a linear complexity with respect to the horizon τ as stated in (7).

Proof sketch of global convergence. Intuitively, assumption (21) ensures that the function g is invertible such that global convergence can be ensured as long as h satisfies sufficient conditions for global convergence and regularizations are chosen to ensure a sufficient decrease. Formally, to ensure global convergence, we consider taking a regularization ν that may depend on the current command $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$, s.t. for $\boldsymbol{v} = \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$,

$$\mathcal{J}\left(\boldsymbol{u}+\boldsymbol{v}\right) \leq h \circ g(\boldsymbol{u}) + q_h^{g(\boldsymbol{u})} \circ \ell_g^{\boldsymbol{u}}(\boldsymbol{v}) + \frac{\nu}{2}\|\boldsymbol{v}\|_2^2 = \mathcal{J}(\boldsymbol{u}) + \frac{1}{2}\nabla\mathcal{J}(\boldsymbol{u})^\top\boldsymbol{v}. \tag{23}$$

Given the analytic form of $\boldsymbol{v} = \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$ in (22), the above condition ensures that $\mathcal{J}\left(\boldsymbol{u}+\boldsymbol{v}\right) - \mathcal{J}(\boldsymbol{u}) \leq -\alpha\|\nabla h(g(\boldsymbol{u}))\|_2^2$, for some constant α that depends on the regularization ν and the properties of the objective. Hence, if h satisfies a gradient dominating property, i.e., there exists $\mu > 0, r \in [1/2, 1)$ s.t. $\|\nabla h(\boldsymbol{x})\|_2^2 \geq \mu^r(h(\boldsymbol{x}) - h^*)^r$ for any $\boldsymbol{x} \in \mathbb{R}^{\tau n_x}$, global convergence can be ensured given a constant regularization. For example, if $r = 1/2$, by taking a constant regularization ensuring (23), we get a global linear convergence rate. We further show that a regularization ensuring (23) can be chosen to scale as $\|\nabla h(g(\boldsymbol{u}))\|_2$, which leads to a computational complexity in two terms: a first term that illustrates a slow convergence phase when $\|\nabla h(g(\boldsymbol{u}))\|_2$ is large and large regularizations are chosen, a second term that amounts to a linear convergence rate as the iterates approach the solution. The second term is governed by the condition number of the cost and illustrates that, asymptotically, only the computational complexity of finding the minimum of the cost matters.

Proof sketch of local convergence. The global rate of convergence sketched above can be further improved by analyzing the local behavior of the algorithm around a solution, which amounts to an approximate Newton step on the total costs as explained below. Namely, if g satisfies (21), then the matrix $\nabla g(\boldsymbol{u})^\top\nabla g(\boldsymbol{u})$ is invertible. Denoting $\boldsymbol{x} = g(\boldsymbol{u})$, $G = \nabla g(\boldsymbol{u})$ and $H = \nabla^2 h(\boldsymbol{x})$, we then have by standard linear algebra manipulations, that the oracle returned by the ILQR algorithm satisfies

$$\begin{aligned}\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u}) &= -(GHG^\top + \nu\,\mathrm{I})^{-1}G\nabla h(\boldsymbol{x}) \\ &= -G(HG^\top G + \nu\,\mathrm{I})^{-1}\nabla h(\boldsymbol{x}) && \textit{(Push-through identity)} \\ &= -G(G^\top G)^{-1}(H + \nu(G^\top G)^{-1})^{-1}\nabla h(\boldsymbol{x}). && \textit{($G^\top G$ invertible)}\end{aligned}$$

Consider then the trajectory associated to a single step of the ILQR algorithm, i.e., for $\boldsymbol{v} = \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$,

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

For $\nu \ll 1$, the difference of the trajectories $\boldsymbol{y} - \boldsymbol{x}$ is close to a Newton direction on the total costs h. In other words, the ILQR algorithm may be analyzed as an approximate Newton method on the total costs. In particular, this suggests that the algorithm can have a local quadratic convergence rate if (i) the costs satisfy the assumptions required for a Newton method to converge locally quadratically, such as self-concordance, (ii) the regularization decreases fast enough.

As already mentioned, the smallest possible regularization ν that ensures condition (23) scales as $\|\nabla h(g(\boldsymbol{u}))\|_2$, such that the smallest possible regularization ensuring (23) will also ensure local quadratic convergence since $\|\nabla h(g(\boldsymbol{u}))\|_2$ converges to zero as we approach the solution. In total, for strongly convex functions, we then get a convergence in three phases: a slow convergence far from the solution as already explained, a linear convergence as we get closer to the solution and finally a quadratic convergence.

Proof sketch of convergence of the IDDP algorithm. The analysis of the IDDP algorithm follows from the analysis of the ILQR algorithm by noting that $\|\operatorname{DDP}_\nu(\mathcal{J})(\boldsymbol{u}) - \operatorname{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2 \leq \eta \|\operatorname{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2^2$ for some constant η independent of $\boldsymbol{u}$ and ν. Namely, the quadratic local convergence is maintained since the approximation error of LQR by DDP is quadratic and the global convergence can be ensured by choosing the same criterion for choosing the regularization as in the ILQR method.

4.2 Global convergence rate

We start by showing global convergence of the ILQR algorithm provided that the costs satisfy a sufficient condition for global convergence, namely gradient dominance, a.k.a. a Polyak-Łojasiewicz inequality Polak (2011); Łojasiewicz (1963); Bolte et al. (2017).

Theorem 4.2. *Given Hypothesis 4.1, the sufficient decrease condition (23) is satisfied for a regularization*

$$\nu(\boldsymbol{u}) = \frac{L_g \|\nabla h(g(\boldsymbol{u}))\|_2}{2} \gamma\left(\frac{L_g \|\nabla h(g(\boldsymbol{u}))\|_2}{4 l_g^2 (M_h l_h^2/(3L_g) + L_h)}\right),$$

where $\gamma(x) = 1 + \sqrt{1 + 1/x}$. In addition to Hypothesis 4.1, consider that the costs are dominated by their gradients, i.e., there exists $r \in [1/2, 1)$ and $\mu > 0$ such that $\|\nabla h_t(x)\|_2 \geq \mu^r (h_t(x) - h_t^*)^r$ for all $x \in \mathbb{R}^{n_x}, t \in \{1, \dots, \tau\}$, and the total cost satisfies then, for $\mu_h = \mu/\tau^{(2r-1)/2r}$,

$$\forall \boldsymbol{x} \in \mathbb{R}^{\tau n_x}, \quad \|\nabla h(\boldsymbol{x})\|_2 \geq \mu_h^r (h(\boldsymbol{x}) - h^*)^r. \tag{24}$$

If $r = 1/2$, the number of iterations of the ILQR algorithm to converge to an accuracy ε for problem (1) given regularizations $\nu_k = \nu(\boldsymbol{u}^{(k)})$ is at most

$$k \leq 2\rho_h \ln\left(\frac{\delta_0}{\varepsilon}\right) + 4\theta_g \sqrt{\delta_0} \gamma\left(\frac{\theta_g \sqrt{\delta_0}}{\alpha}\right),$$

and, if $1/2 < r < 1$, the number of iterations to converge to an accuracy ε is at most

$$k \leq \frac{2}{2r-1} \frac{\rho_h}{\varepsilon^{2r-1}} + \frac{2}{1-r} \theta_g \delta_0^{1-r} + \sqrt{2\theta_g \alpha} \frac{1}{1 - 3r/2} \left(\varepsilon^{1-3r/2} - \left(\frac{\alpha}{\theta_g}\right)^{1/r - 3/2}\right),$$

with $\rho_h = L_h/\mu_h^{2r}$, $\rho_g = l_g/\sigma_g$, $\theta_h = M_h/(2\mu_h^{3r})$, $\theta_g = L_g/(\sigma_g^2 \mu_h^r)$, $\alpha = 4\rho_g^2(2\rho_g^2 \theta_h/(3\theta_g) + \rho_h)$, $\delta_0 = \mathcal{J}(\boldsymbol{u}^{(0)}) - \mathcal{J}^*$ and the case $r = 2/3$ is to be understood limit-wise.

Before presenting the proof, a few remarks are in order.

Remark 4.3. Consider the case $r = 1/2$ in Theorem 4.2. The constants appearing in the bound are (i) the condition number $\rho_h = L_h/\mu_h$ of the total cost h, (ii) the condition number $\rho_g = l_g/\sigma_g$ of the gradient of g, $\nabla g(\boldsymbol{u})$, (iii) a constant $\theta_h = M_h/(2\mu_h^{3/2})$ that can be interpreted as a bound on the self-concordance parameter of the cost h if the total costs are strongly convex, (iv) a constant $\theta_g = L_g/(\sigma_g^2 \sqrt{\mu_h})$ whose dimension is the same as θ_h, i.e., the inverse of the squared root of the objective. Finally the term α in the bound is a dimension independent constant that incorporates the condition numbers of h and $\nabla g(\boldsymbol{u})$ and an additional condition number defined as the ratio θ_h/θ_g.

Remark 4.4. The rate of convergence in Theorem 4.2 for $r = 1/2$ entails a linear convergence rate for $\varepsilon \to 0$ but also a term that only depends on δ_0. This second term represents a slow convergence in the first iterations of the ILQR algorithm, in the sense that for low accuracy ε it dominates the computational complexity. Namely, the remaining number of iterations to reach accuracy ε after having done j iterations is of the order of $2\rho_h \ln(\delta_j/\varepsilon) + \theta_g \sqrt{\delta_j}$ so unless $\delta_j < 1/{\theta_g}^2$, the second term dominates the complexity. In other words, $1/{\theta_g}^2$ acts as a radius of linear convergence for the gaps δ_j.

Remark 4.5. Theorem 4.2 presents an implementation where we know the smoothness constants of the problem to compute $\nu(\boldsymbol{u}^{(k)})$. In practice, regularizations can be taken of the form $\nu_k = \bar{\nu}\|\nabla h(g(\boldsymbol{u}^{(k)}))\|_2\gamma(\|\nabla h(g(\boldsymbol{u}^{(k)}))\|_2)$. One easily sees that there exists $\bar{\nu}$ which ensure $\nu_k \geq \nu(\boldsymbol{u}^{(k)})$ for any $\boldsymbol{u}^{(k)}$ such that $\bar{\nu}$ can be found by a line-search procedure.

Remark 4.6. For $L_g = 0$, the terms depending on δ_0 uniquely vanish since $\theta_g = 0$ in this case. We then get the classical rates when minimizing a function h that satisfy (24) with a first-order method. The rates can be improved by analyzing the local behavior of the algorithm to take advantage of the quadratic approximations of the total costs h.

Proof of Theorem 4.2. The validity of the gradient dominating property for the total costs is presented in Lemma C.1. First, note that if h satisfies (24) and g satisfies (21), then for any $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$, we have $\|\nabla(h \circ g)(\boldsymbol{u})\|_2 \geq \sigma_g\mu_h^r(h(g(\boldsymbol{u}))-h^*)^r$. Hence for $\boldsymbol{u}^* \in \arg\min \mathcal{J}(\boldsymbol{u})$ with $\mathcal{J} = h\circ g$, we get $0 = \|\nabla\mathcal{J}(\boldsymbol{u}^*)\|_2 \geq \sigma_g\mu_h^r(h(g(\boldsymbol{u}^*))-h^*)^r \geq 0$, hence $\mathcal{J}^* = h^*$.

We have from Lemma B.2 that for any $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{\tau n_u}$, denoting $a_0 = M_h l_g^3/3+L_g L_h l_g$,

$$|(h \circ g)(\boldsymbol{u}+\boldsymbol{v})-(h \circ g)(\boldsymbol{u})-q_h^{g(\boldsymbol{u})} \circ \ell_g^{\boldsymbol{u}}(\boldsymbol{v})| \leq \frac{L_g\|\nabla h(g(\boldsymbol{u}))\|_2+a_0\|\boldsymbol{v}\|_2}{2}\|\boldsymbol{v}\|_2^2.$$

Since $\|\operatorname{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2 \leq l_g\|\nabla h(g(\boldsymbol{u}))\|_2/\nu$, condition (23) is satisfied for $\nu > 0$ s.t. $a_1 + a_2/\nu \leq \nu$, where $a_1 = L_g\|\nabla h(g(\boldsymbol{u}))\|_2$ $a_2 = a_0 l_g\|\nabla h(g(\boldsymbol{u}))\|_2$. Therefore, denoting $\gamma(x) = 1 + \sqrt{1+1/x}$, condition (23) is satisfied for any

$$\nu \geq \nu(\boldsymbol{u}) = \frac{a_1+\sqrt{a_1^2+4a_2}}{2} = \frac{L_g\|\nabla h(g(\boldsymbol{u}))\|_2}{2}\gamma\left(\frac{L_g^2\|\nabla h(g(\boldsymbol{u}))\|_2}{4a_0 l_g}\right).$$

We have then for $\boldsymbol{v} = \operatorname{LQR}_{\nu(\boldsymbol{u})}(\mathcal{J})(\boldsymbol{u})$, $G = \nabla g(\boldsymbol{u})$, $H = \nabla^2 h(g(\boldsymbol{u}))$, since condition (23) is satisfied,

$$\begin{aligned}
\mathcal{J}(\boldsymbol{u}+\boldsymbol{v}) - \mathcal{J}(\boldsymbol{u}) &\leq -\frac{1}{2}\nabla h(g(\boldsymbol{u}))^\top G^\top (GHG^\top + \nu(\boldsymbol{u})\,\mathrm{I})^{-1} G\nabla h(g(\boldsymbol{u})) \\
&= -\frac{1}{2}\nabla h(g(\boldsymbol{u}))^\top (H + \nu(\boldsymbol{u})(G^\top G)^{-1})^{-1}\nabla h(g(\boldsymbol{u})) \\
&\leq -\frac{1}{2}\frac{\sigma_g^2}{\sigma_g^2 L_h + \nu(\boldsymbol{u})}\|\nabla h(g(\boldsymbol{u}))\|_2^2 \leq -\frac{b_1 x^2}{\sqrt{b_2 x^2 + b_3 x} + b_4 x + b_5},
\end{aligned} \tag{25}$$

where $x = \|\nabla h(g(\boldsymbol{u}))\|_2$, $b_1 = \sigma_g^2$, $b_2 = L_g^2$, $b_3 = 4a_0 l_g$, $b_4 = L_g$, $b_5 = 2\sigma_g^2 L_h$.

The function $f_1 : x \to b_1 x^2/(\sqrt{b_2 x^2 + b_3 x} + b_4 x + b_5)$ is increasing for $x \geq 0$. Hence, denoting $\delta = h(g(\boldsymbol{u})) - h^* = \mathcal{J}(\boldsymbol{u}) - \mathcal{J}^*$, we have $f_1(x) \geq f_1((\mu_h\delta)^r)$ by hypothesis (24). Denoting $\delta_k = \mathcal{J}(\boldsymbol{u}^{(k)}) - \mathcal{J}^*$ for k the iteration of the ILQR algorithm, we then have $f_2'(\delta_k)(\delta_{k+1} - \delta_k) \leq -1$, with

$$f_2'(\delta) = \frac{1}{f_1((\mu_h\delta)^r)} = \frac{2\rho_h}{\delta^{2r}} + \frac{\theta_g}{\delta^r} + \frac{\theta_g\sqrt{\delta^{2r} + \alpha\delta^r/\theta_g}}{\delta^{2r}} = \frac{2\rho_h}{\delta^{2r}} + \frac{\theta_g\gamma(\theta_g\delta^r/\alpha)}{\delta^r},$$

with $\rho_h{=}L_h/\mu_h^{2r}$, $\rho_g{=}l_g/\sigma_g$, $\theta_h{=}M_h/(2\mu_h^{3r})$, $\theta_g{=}L_g/(\sigma_g^2\mu_h^r)$, $\alpha{=}4\rho_g^2(2\rho_g^2\theta_h/(3\theta_g){+}\rho_h)$.

Since f_2 is concave on $\mathbb{R}^+$, we deduce that $f_2(\delta_{k+1}){-}f_2(\delta_k) \leq -1$ and so $f_2(\delta_k) \leq -k{+}f_2(\delta_0)$. Note that f_2 is strictly decreasing, so we get that, for the algorithm to reach an accuracy ε, we need at most $k \leq f_2(\delta_0) - f_2(\varepsilon)$ iterations.

If $r = 1/2$, one can verify that $\delta \to a\ln(2a\sqrt{\delta}\gamma(\sqrt{\delta}/a) + a^2) + 2\sqrt{\delta}\gamma(\sqrt{\delta}/a)$ is an antiderivative of $\delta \to \gamma(\sqrt{\delta}/a)/\sqrt{\delta}$ for any $a > 0$. Hence, for $r = 1/2$, the number of iterations to converge to an accuracy ε is at most

$$\begin{aligned}
k &\leq 2\rho_h\ln\left(\frac{\delta_0}{\varepsilon}\right) + 2\theta_g\left(\sqrt{\delta_0}\gamma\left(\frac{\theta_g\sqrt{\delta_0}}{\alpha}\right) - \sqrt{\varepsilon}\gamma\left(\frac{\theta_g\sqrt{\varepsilon}}{\alpha}\right)\right) \\
&\quad + \alpha\ln\left(\frac{2\theta_g\sqrt{\delta_0}\gamma(\theta_g\sqrt{\delta_0}/\alpha)+\alpha}{2\theta_g\sqrt{\varepsilon}\gamma(\theta_g\sqrt{\varepsilon}/\alpha)+\alpha}\right) \\
&\leq 2\rho_h\ln\left(\frac{\delta_0}{\varepsilon}\right) + 2\theta_g\sqrt{\delta_0}\gamma\left(\frac{\theta_g\sqrt{\delta_0}}{\alpha}\right) + \alpha\ln\left(1{+}2\frac{\theta_g\sqrt{\delta_0}}{\alpha}\gamma\left(\frac{\theta_g\sqrt{\delta_0}}{\alpha}\right)\right).
\end{aligned}$$

By using that $\ln(1+x) \leq x$ for $x > -1$, we get the claimed bound in this case.

If $1/2 < r < 1$, by integrating f_2, the number of iterations to converge to an accuracy ε is at most

$$k \leq \frac{2\rho_h}{2r-1}\left(\frac{1}{\varepsilon^{2r-1}} - \frac{1}{\delta_0^{2r-1}}\right) + \frac{\theta_g}{(1-r)}\left(\delta_0^{1-r} - \varepsilon^{1-r}\right) + \int_\varepsilon^{\delta_0} \frac{\theta_g\sqrt{x^{2r} + \alpha x^r/\theta_g}}{x^{2r}} dx.$$

The bound follows in this case by using that, for $1/2 < r < 1$, and $a > 0$,

$$\int_\varepsilon^{\delta_0} \frac{\sqrt{x^{2r} + ax^r}}{x^{2r}} dx \leq \int_\varepsilon^{a^{1/r}} \frac{\sqrt{2a}}{x^{3r/2}} dx + \int_{a^{1/r}}^{\delta_0} \frac{1}{x^r} dx.$$

□

4.3 Local convergence rate

As we analyze the ILQR algorithm locally as an approximate Newton method on the costs, we use the notations and assumptions used to analyze a Newton method. Namely, we assume the costs h_t strictly convex and we define the norm induced by the Hessian at a point $\boldsymbol{x} \in \mathbb{R}^{\tau n_x}$ and its dual norm as, respectively, for $\boldsymbol{y} \in \mathbb{R}^{\tau n_x}$,

$$\|\boldsymbol{y}\|_{\boldsymbol{x}} = \sqrt{\boldsymbol{y}^\top \nabla^2 h(\boldsymbol{x}) \boldsymbol{y}}, \quad \|\boldsymbol{y}\|_{\boldsymbol{x}}^* = \sqrt{\boldsymbol{y}^\top \nabla^2 h(\boldsymbol{x})^{-1} \boldsymbol{y}}.$$

For a matrix $A \in \mathbb{R}^{\tau n_x \times \tau n_u}$, we denote $\|A\|_{\boldsymbol{x}} = \|\nabla^2 h(\boldsymbol{x})^{1/2} A\|_2$ the norm induced by the local geometry of h w.r.t. the Euclidean norm. Finally we denote the Newton decrement of the cost function, as, for $\boldsymbol{x} \in \mathbb{R}^{\tau n_u}$,

$$\lambda_h(\boldsymbol{x}) = \sqrt{\nabla h(\boldsymbol{x})^\top \nabla^2 h(\boldsymbol{x})^{-1} \nabla h(\boldsymbol{x})}.$$

To analyze the local convergence of the ILQR algorithm we consider the costs to be self-concordant as it is sufficient to ensure global convergence of a Newton method applied to the costs Nesterov (2018). In addition, we consider smoothness properties of the function g with respect to the geometry induced by the Hessian of the costs as presented in the assumptions below. Overall to guarantee local convergence, the following hypothesis is sufficient.

Hypothesis 4.7. *We consider that the costs h_t and so the total cost h are strictly convex and the following constants, defined for $g : \boldsymbol{u} \to f^{[\tau]}(\bar{x}_0, \boldsymbol{u})$ with $f^{[\tau]}$ the control in τ steps of the dynamic f defined in (9), are finite*

$$l = \sup_{\substack{\boldsymbol{u},\boldsymbol{v} \in \mathbb{R}^{\tau n_u} \\ \boldsymbol{v} \neq 0}} \frac{\|g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u})\|_{g(\boldsymbol{u})}}{\|\boldsymbol{v}\|_2}, \quad L = \sup_{\substack{\boldsymbol{u},\boldsymbol{v} \in \mathbb{R}^{\tau n_u} \\ \boldsymbol{v} \neq 0}} \frac{\|\nabla g(\boldsymbol{u}+\boldsymbol{v})^\top - \nabla g(\boldsymbol{u})^\top\|_{g(\boldsymbol{u})}}{\|\boldsymbol{v}\|_2}$$

$$\vartheta_h = \sup_{\substack{\boldsymbol{x},\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{y}_3 \in \mathbb{R}^{\tau n_x} \\ \boldsymbol{y}_1 \neq 0, \boldsymbol{y}_2 \neq 0, \boldsymbol{y}_3 \neq 0}} \frac{|\nabla^3 h(\boldsymbol{x})[\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3]|}{2\|\boldsymbol{y}_1\|_{\boldsymbol{x}} \|\boldsymbol{y}_2\|_{\boldsymbol{x}} \|\boldsymbol{y}_3\|_{\boldsymbol{x}}}, \quad \sigma = \inf_{\substack{\boldsymbol{u} \in \mathbb{R}^{\tau n_u}, \mu \in \mathbb{R}^{\tau n_x} \\ \mu \neq 0}} \frac{\|\nabla g(\boldsymbol{u})\boldsymbol{\mu}\|_2}{\|\boldsymbol{\mu}\|_{g(\boldsymbol{u})}^*}.$$

In consequence, h is ϑ_h-self concordant (Nesterov, 2018, Definition 5.1.1, Lemma 5.1.2) and we have that $\sigma \leq \sigma_{\min}(\nabla g(\boldsymbol{u}) \nabla^2 h(g(\boldsymbol{u}))^{1/2})$, $\sigma_{\max}(\nabla g(\boldsymbol{u}) \nabla^2 h(g(\boldsymbol{u}))^{1/2}) \leq l$, for any $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$.

In terms of the dynamic and the individual costs, Hypothesis 4.7 is satisfied if h_t is strongly convex for all t such that the total costs h are strongly convex and if Hypothesis 4.1 is also satisfied. In that case, we have

$$l \leq \sqrt{L_h} l_g, \quad L \leq \sqrt{L_h} L_g, \quad 2\vartheta_h \leq M_h/\mu_h^{3/2}, \quad \sigma \geq \sqrt{\mu_h}\sigma_g. \tag{26}$$

Equipped with a stepsize proportional to the Newton decrement, we can show a local quadratic convergence rate of the ILQR algorithm given Hypothesis 4.7.

Theorem 4.8. *Given Hypothesis 4.7, consider the ILQR for problem (1) with regularizations of the form $\nu_k = \bar{\nu}\lambda_h(g(\boldsymbol{u}^{(k)}))$ for some $\bar{\nu} \geq 0$. For $k \geq 0$ such that*

$$\lambda_h(g(\boldsymbol{u}^{(k)})) < \lambda = \frac{1}{\max\{4\vartheta_h + 3\vartheta_g + 2\bar{\nu}/\sigma^2, 2\varrho\vartheta_h\}}, \tag{27}$$

where $\varrho = l/\sigma$ and $\vartheta_g = L/\sigma^2$, we have $\lambda_h(g(\boldsymbol{u}^{(k+1)})) \leq \lambda^{-1}\lambda_h(g(\boldsymbol{u}^{(k)}))^2$, and the ILQR algorithm converges quadratically to the minimum value of problem (1).

Remark 4.9. If h is a quadratic, such that the algorithm reduces to a Gauss-Newton algorithm and $\vartheta_h = 0$, the radius of quadratic convergence reduces to $\lambda = 1/(3\vartheta_g + 2\bar{\nu})$. If in addition, no regularization is in effect, the radius of convergence reduces to $\lambda = 1/3\vartheta_g$, which can be expressed as $1/(3\theta_g\sqrt{\rho_h})$ if the total cost is μ_h strongly convex with θ_g, ρ_h defined as in Theorem 4.2 and σ, L expressed using Eq. (26). So up to $3\sqrt{\rho_h}$, the parameter $1/\theta_g$ acts again as a radius of fast convergence as in Theorem 4.2.

Remark 4.10. For better readability, we simplified the expression of the radius of convergence. A closer look at the proof shows that a non-zero regularization may lead to a larger radius of convergence than no regularization.

Proof of Theorem 4.8. Let $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$ $G{=}\nabla g(\boldsymbol{u})$, $H{=}\nabla^2 h(g(\boldsymbol{u}))$, $\boldsymbol{v}{=}\,\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$ with $\nu = \bar{\nu}\lambda_h(g(\boldsymbol{u}))$. Assume that

$$\lambda_h(g(\boldsymbol{u})) \leq 1/\max\{\sqrt{2\vartheta_h\vartheta_g}c_1, 2\vartheta_h\varrho c_2, 2\vartheta_h c_2\},$$

where $c_1{=}\max\{1{-}\bar{\nu}/(\sqrt{2\vartheta_h L}l), 0\}$, $c_2{=}\max\{1{-}\bar{\nu}/(2l^2\vartheta_h), 0\}$, $\varrho{=}l/\sigma$, $\vartheta_g{=}L/\sigma^2$. We have

$$\lambda_h(g(\boldsymbol{u}{+}\boldsymbol{v})){\leq}\underbrace{\|\nabla h(g(\boldsymbol{u}{+}\boldsymbol{v})){-}\nabla h(g(\boldsymbol{u}){+}G^\top\boldsymbol{v})\|^*_{g(\boldsymbol{u}+\boldsymbol{v})}}_{A} + \underbrace{\|\nabla h(g(\boldsymbol{u}){+}G^\top\boldsymbol{v})\|^*_{g(\boldsymbol{u}+\boldsymbol{v})}}_{B}. \tag{28}$$

Bounding A in (28). By definition of l in Hypothesis 4.7 and Lemma B.4, we have

$$\|g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u})\|_{g(\boldsymbol{u})} \leq l\|\boldsymbol{v}\|_2, \qquad \|\boldsymbol{v}\|_2 \leq \frac{l\lambda_h(g(\boldsymbol{u}))}{l\sigma + \bar{\nu}\lambda_h(g(\boldsymbol{u}))}. \tag{29}$$

One easily verifies that $x/(1+ax) \leq c$ if $0 \leq x \leq c/\max\{1-ca, 0\}$ for any $a, c > 0$. So for $\lambda_h(g(\boldsymbol{u}))) \leq 1/(2\vartheta_h\varrho c_2)$, we have $\|g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u})\|_{g(\boldsymbol{u})} \leq 1/(2\vartheta_h)$. Hence, using that h is ϑ_h-self-concordant, (Nesterov, 2018, Theorem 5.1.7) applies and by using the definition of L in Hypothesis 4.7, we have

$$\begin{aligned}\|g(\boldsymbol{u}{+}\boldsymbol{v}){-}g(\boldsymbol{u}){-}G^\top\boldsymbol{v}\|_{g(\boldsymbol{u}+\boldsymbol{v})} &\leq \frac{1}{1{-}\vartheta_h\|g(\boldsymbol{u}{+}\boldsymbol{v}){-}g(\boldsymbol{u})\|_{g(\boldsymbol{u})}}\|g(\boldsymbol{u}{+}\boldsymbol{v}){-}g(\boldsymbol{u}){-}G^\top\boldsymbol{v}\|_{g(\boldsymbol{u})} \\ &\leq 2\left\|\int_0^1 \nabla g(\boldsymbol{u}{+}t\boldsymbol{v})^\top\boldsymbol{v}dt - \nabla g(\boldsymbol{u})^\top\boldsymbol{v}\right\|_{g(\boldsymbol{u})} = L\|\boldsymbol{v}\|_2^2.\end{aligned}$$

Using (29), for $\lambda_h(g(\boldsymbol{u})) \leq 1/(\sqrt{2\vartheta_h\vartheta_g}c_1)$, we get $\|g(\boldsymbol{u}{+}\boldsymbol{v}){-}g(\boldsymbol{u}){-}G^\top\boldsymbol{v}\|_{g(\boldsymbol{u}+\boldsymbol{v})}{\leq}1/(2\vartheta_h)$. Since the total cost h is ϑ_h-self-concordant, we can then use Lemma B.5 to obtain

$$\begin{aligned}A &\leq \frac{1}{1 - \vartheta_h\|g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u}) - G^\top\boldsymbol{v}\|_{g(\boldsymbol{u}+\boldsymbol{v})}}\|g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u}) - G^\top\boldsymbol{v}\|_{g(\boldsymbol{u}+\boldsymbol{v})} \\ &\leq \frac{2Ll^2\lambda_h(g(\boldsymbol{u}))^2}{(l\sigma + \bar{\nu}\lambda_h(g(\boldsymbol{u})))^2}.\end{aligned} \tag{30}$$

Bounding B in (28). Recall that for $\lambda_h(g(\boldsymbol{u}))) \leq 1/(2\vartheta_h\varrho c_2)$, we have $\|g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u})\|_{g(\boldsymbol{u})} \leq 1/(2\vartheta_h)$. Since h is ϑ_h-self-concordant, we have then (Nesterov, 2018, Theorem 5.1.7),

$$B \leq \frac{1}{1{-}\vartheta_h\|g(\boldsymbol{u}{+}\boldsymbol{v}){-}g(\boldsymbol{u})\|_{g(\boldsymbol{u})}}\|\nabla h(g(\boldsymbol{u}){+}G^\top v)\|^*_{g(\boldsymbol{u})} \leq 2\|\nabla h(g(\boldsymbol{u}){+}G^\top v)\|^*_{g(\boldsymbol{u})}. \tag{31}$$

Denote $\nu = \bar{\nu}\lambda_h(g(\boldsymbol{u}))$ and define $\boldsymbol{n} = -(H + \nu(G^\top G)^{-1})^{-1}\nabla h(g(\boldsymbol{u}))$. Using that

$$\boldsymbol{v} = -G(G^\top G)^{-1}(H + \nu(G^\top G)^{-1})^{-1}\nabla h(g(\boldsymbol{u})),$$

and denoting $\boldsymbol{x} = g(\boldsymbol{u})$, we have then

$$\begin{aligned}\|\nabla h(g(\boldsymbol{u}) + G^\top\boldsymbol{v})\|^*_{g(\boldsymbol{u})} &= \|\nabla h(\boldsymbol{x}+\boldsymbol{n}) - \nabla h(\boldsymbol{x}) - (H + \nu(G^\top G)^{-1})\boldsymbol{n}\|^*_{\boldsymbol{x}} \\ &\leq \|\nabla h(\boldsymbol{x}+\boldsymbol{n}) - \nabla h(\boldsymbol{x}) - H\boldsymbol{n}\|^*_{\boldsymbol{x}} + \nu\|(G^\top G)^{-1}\boldsymbol{n}\|^*_{\boldsymbol{x}}.\end{aligned} \tag{32}$$

The first term can be bounded as in the proof of local convergence of a Newton method (Nesterov, 2018, Theorem 5.2.2). Namely, we have

$$\|\nabla h(\boldsymbol{x}+\boldsymbol{n}) - \nabla h(\boldsymbol{x}) - H\boldsymbol{n}\|^*_{\boldsymbol{x}} = \|\int_0^1 (\nabla^2 h(\boldsymbol{x}+t\boldsymbol{n}) - \nabla^2 h(\boldsymbol{x}))\boldsymbol{n}dt\|^*_{\boldsymbol{x}}.$$

Since $\sigma_{\max}(\nabla g(\boldsymbol{u})\nabla^2 h(g(\boldsymbol{u}))^{1/2}) \le l$, we have

$$\|\boldsymbol{n}\|_{\boldsymbol{x}} = \|(\mathrm{I}+\nu H^{-1/2}(G^\top G)^{-1}H^{-1/2})^{-1}H^{-1/2}\nabla h(g(\boldsymbol{u}))\|_2 \le \frac{\lambda_h(g(\boldsymbol{u}))}{1+\bar{\nu}l^{-2}\lambda_h(g(\boldsymbol{u}))}.$$

So if $\lambda_h(g(\boldsymbol{u})) \le 1/(2\vartheta_h c_2)$, we get $\|\boldsymbol{n}\|_{\boldsymbol{x}} \le 1/(2\vartheta_h)$ and, since h is self-concordant, by (Nesterov, 2018, Corollary 5.1.5), we have, denoting $J = \int_0^1 (\nabla^2 h(\boldsymbol{x}+t\boldsymbol{n}) - \nabla^2 h(\boldsymbol{x}))dt$,

$$(-\|\boldsymbol{n}\|_{\boldsymbol{x}}\vartheta_h + \|\boldsymbol{n}\|_{\boldsymbol{x}}^2\vartheta_h^2/3)\nabla^2 h(\boldsymbol{x}) \preceq J \preceq \frac{\|\boldsymbol{n}\|_{\boldsymbol{x}}\vartheta_h}{1-\|\boldsymbol{n}\|_{\boldsymbol{x}}\vartheta_h}\nabla^2 h(\boldsymbol{x}).$$

Moreover, since $\|\boldsymbol{n}\|_{\boldsymbol{x}} < 1/(2\vartheta_h)$, we have $\|\boldsymbol{n}\|_{\boldsymbol{x}}\vartheta_h - \|\boldsymbol{n}\|_{\boldsymbol{x}}^2\vartheta_h^2/3 \le \frac{\|\boldsymbol{n}\|_{\boldsymbol{x}}\vartheta_h}{1-\|\boldsymbol{n}\|_{\boldsymbol{x}}\vartheta_h}$. Hence we get

$$\|\nabla h(\boldsymbol{x}+\boldsymbol{n}) - \nabla h(\boldsymbol{x}) - H\boldsymbol{n}\|_{\boldsymbol{x}}^* \le \frac{\|\boldsymbol{n}\|_{\boldsymbol{x}}^2\vartheta_h}{1-\|\boldsymbol{n}\|_{\boldsymbol{x}}\vartheta_h} \le \frac{2\lambda_h(g(\boldsymbol{u}))^2\vartheta_h}{(1+\bar{\nu}l^{-2}\lambda_h(g(\boldsymbol{u})))^2}. \tag{33}$$

On the other hand, since $\sigma \le \sigma_{\min}(\nabla g(\boldsymbol{u})\nabla^2 h(g(\boldsymbol{u}))^{1/2})$, we have

$$\|(G^\top G)^{-1}\boldsymbol{n}\|_{g(\boldsymbol{u})}^* = \|(H^{1/2}G^\top G H^{1/2} + \nu\,\mathrm{I})^{-1}H^{-1/2}\nabla h(g(\boldsymbol{u}))\|_2 \le \frac{\lambda_h(g(\boldsymbol{u}))}{\sigma^2+\bar{\nu}\lambda_h(g(\boldsymbol{u}))}. \tag{34}$$

So combining (34) and (33) into (32) and then (31) we get

$$B \le 2\left(\frac{2\vartheta_h}{(1+\bar{\nu}l^{-2}\lambda_h(g(\boldsymbol{u})))^2} + \frac{\bar{\nu}}{\sigma^2+\bar{\nu}\lambda_h(g(\boldsymbol{u}))}\right)\lambda_h(g(\boldsymbol{u}))^2. \tag{35}$$

Local quadratic convergence rate. So combining (30) and (35) into (28), we get, as long as $\lambda_h(g(\boldsymbol{u})) \le 1/\max\{\sqrt{2\vartheta_h\vartheta_g}c_1, 2\vartheta_h\varrho c_2, 2\vartheta_h c_2\}$,

$$\lambda_h(g(\boldsymbol{u}+\boldsymbol{v})) \le \left(\frac{2Ll^2}{(l\sigma+\bar{\nu}\lambda_h(g(\boldsymbol{u})))^2} + \frac{4\vartheta_h}{(1+\bar{\nu}l^{-2}\lambda_h(g(\boldsymbol{u})))^2} + \frac{2\bar{\nu}}{\sigma^2+\bar{\nu}\lambda_h(g(\boldsymbol{u}))}\right)\lambda_h(g(\boldsymbol{u}))^2.$$

Note that $c_1, c_2 \le 1$ and that $2\vartheta_g + 4\vartheta_h + 2\bar{\nu}/\sigma^2 \ge \max\{2\vartheta_h, \sqrt{2\vartheta_h\vartheta_g}\}$, using the arithmetic-geometric mean inequality. Hence, for

$$\lambda_h(g(\boldsymbol{u})) < \lambda = 1/\max\{2\vartheta_g + 4\vartheta_h + 2\bar{\nu}/\sigma^2, 2\vartheta_h\varrho\},$$

we get $\lambda_h(g(\boldsymbol{u}+\boldsymbol{v})) \le \bar{\lambda}^{-1}\lambda_h(g(\boldsymbol{u}))^2 < \lambda_h(g(\boldsymbol{u}))$, that is, we reach the region of quadratic convergence for $g(\boldsymbol{u})$. □

4.4 Global complexity for strongly convex costs

Given Hypothesis 4.1, if the total cost is strongly convex then it satisfies the condition of Theorem 4.2 and Hypothesis 4.7 is satisfied with the estimates given in (26) . We can then bound the number of iterations to local quadratic convergence and obtain the total complexity bound in this case. The following theorem is the detailed version of Thm. 1.1.

Theorem 4.11. *Consider the costs h_t to be μ_h-strongly convex and Hypothesis 4.1 to be satisfied. Then condition (23) is satisfied for a regularization*

$$\nu(\boldsymbol{u}) = L_g\|\nabla h(g(\boldsymbol{u}))\|_2 + \frac{2l_g^2(M_h l_g^2/3 + L_g L_h)\|\nabla h(g(\boldsymbol{u}))\|_2}{L_g\|\nabla h(g(\boldsymbol{u}))\|_2 + \sigma_g l_g \mu_h}.$$

With such regularization, the number of iterations of the ILQR algorithm to reach an accuracy ε is at most

$$k \le 4\theta_g(\sqrt{\delta_0} - \sqrt{\delta}) + 2\rho_h \ln\left(\frac{\delta_0}{\delta}\right) + 2\alpha\ln\left(\frac{\theta_g\sqrt{\delta_0}+\rho_g}{\theta_g\sqrt{\delta}+\rho_g}\right) + O(\ln\ln(\varepsilon)),$$

where $\delta = 1/(32\rho_h(\theta_h(1+\sqrt{\rho_h}\rho_g^3/3) + \sqrt{\rho_h}\theta_g(1+\rho_g\rho_h))^2)$ is the value of the gap determining the quadratic convergence phase, $\rho_h = L_h/\mu_h$, $\rho_g = l_g/\sigma_g$, $\theta_g = L_g/(\sigma_g^2\sqrt{\mu_h})$, $\theta_h = M_h/(2\mu_h^{3/2})$, $\alpha = 4\rho_g^2(2\rho_g^2\theta_h/(3\theta_g) + \rho_h)$ and $\delta_0 = \mathcal{J}(\boldsymbol{u}^{(0)}) - \mathcal{J}^*$.

The total computational complexity of the algorithm in terms of basic operations is then of the order of $k \times \mathcal{C}(n_x, n_u, \tau)$ with k defined above and $\mathcal{C}(n_x, n_u, \tau)$ defined in Eq. (7).

Remark 4.12. The rate of convergence can now be separated between three phases, (i) the number of iterations $4\theta_g\left(\sqrt{\delta_0}-\sqrt{\delta}\right)$ to reach some linear convergence as explained in Remark 4.4, (ii) the number of iterations to reach the quadratic convergence rate that is captured by the logarithmic terms in the complexity bound , (iii) the quadratic convergence phase.

Remark 4.13. Although the constants μ_h and σ_g may not be available, the ideal regularization $\nu(\boldsymbol{u})$ informs us on an efficient parameterization of the regularizations. Namely, we can choose regularizations of the form $\nu_k = \bar{\nu}\|\nabla h(g(\boldsymbol{u}^{(k)}))\|_2$, where $\bar{\nu}$ can be found by a line-search procedure. A line-search on $\bar{\nu}$ starting from an initial guess $\bar{\nu}_0$ that doubles $\bar{\nu}$ until condition (23) is satisfied, induces then an additional computational cost of at most $\log_2(L_g(1+2\theta_h/\theta_g\rho_g^3+2\rho_h\rho_g)/\bar{\nu}_0)\mathcal{C}(n_x,n_u,\tau)$ elementary computations as can be seen from the expression of $\nu(\boldsymbol{u})$ in Theorem 4.11.

Proof of Theorem 4.11. By using the strong convexity of the costs h, we can refine the choice of the regularization to ensure (23). The validity of the proposed regularization to ensure condition (23) is shown in Lemma B.3. With the proposed regularization, we can show, following the same reasoning as in the proof of Theorem 4.2, that the number of iterations of the ILQR algorithm needed to reach an accuracy ε is bounded by (see Appendix C)

$$k \leq 2\rho_h \ln\left(\frac{\delta_0}{\varepsilon}\right) + 4\theta_g\left(\sqrt{\delta_0}-\sqrt{\varepsilon}\right) + 2\alpha\ln\left(\frac{\theta_g\sqrt{\delta_0}+\rho_g}{\theta_g\sqrt{\varepsilon}+\rho_g}\right), \tag{36}$$

with ρ_h, ρ_g, θ_h, θ_g, α defined as in Theorem 4.2.

For the local convergence, the constants in Theorem 4.8 can be expressed in terms of the constants in Theorem 4.2 as $\sigma=\sqrt{\mu_h}\sigma_g, \vartheta_h=\theta_h, \vartheta_g=\sqrt{\rho_h}\theta_g, \varrho=\sqrt{\rho_h}\rho_g$. From the proof of Theorem 4.8, if $\lambda_h(g(\boldsymbol{u}^{(k)}))\leq 1/\max\{\sqrt{2\vartheta_h\vartheta_g}, 2\vartheta_h\varrho, 2\vartheta_h\}$, then

$$\lambda_h(g(\boldsymbol{u}^{(k+1)})) \leq \left(2\vartheta_g + 4\vartheta_h + \frac{2\bar{\nu}_k}{\sigma^2}\right)\lambda_h(g(\boldsymbol{u}^{(k)}))^2,$$

where $\bar{\nu}_k=\nu(\boldsymbol{u}^{(k)})/\lambda_h(g(\boldsymbol{u}^{(k)}))\leq\sqrt{L_h}(L_g+2l_g(M_h l_g^2/3+L_gL_h)/(\sigma_g\mu_h))$. Define then

$$\lambda = \frac{1}{4(\theta_h(1+\sqrt{\rho_h}\rho_g^3/3)+\sqrt{\rho_h}\theta_g(1+\rho_g\rho_h))}.$$

We have that $\lambda\leq 1/\max\{\sqrt{2\vartheta_h\vartheta_g}, 2\vartheta_h\varrho, 2\vartheta_h\}$. So, if $\lambda_h(g(\boldsymbol{u}^{(k)}))\leq\lambda$, quadratic convergence is ensured.

It remains to link the objective gap to the Newton decrement. By considering a gradient step with step-size $1/L_h$, we have $\|\nabla h(\boldsymbol{x})\|^2\leq 2L_h(h(\boldsymbol{x})-h^*)$ for any $\boldsymbol{x}$, hence $\lambda_h(\boldsymbol{x})\leq\sqrt{2\rho_h(h(\boldsymbol{x})-h^*)}$. So, the number of iterations to reach quadratic convergence is bounded by the number of iterations to get an accuracy $\delta=\lambda^2/(2\rho_h)$. Once quadratic convergence is reached the remaining number of iterations is of the order of $O(\ln\ln\varepsilon^{-1})$. □

4.5 Convergence of Iterative Differential Dynamic Programming

For the IDDP algorithm, we consider selecting the stepsize such that

$$\mathcal{J}(\boldsymbol{u}+\mathrm{DDP}_\nu(\mathcal{J})(\boldsymbol{u})) \leq \mathcal{J}(\boldsymbol{u}) + \frac{1}{2}\nabla\mathcal{J}(\boldsymbol{u})^\top \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u}), \tag{37}$$

i.e., we use the same criterion as for the ILQR algorithm (23) to ensure a sufficient decrease. This choice of regularization is motivated by the implementation of the ILQR and IDDP algorithms which both compute $\frac{1}{2}\nabla\mathcal{J}(\boldsymbol{u})^\top\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$ by dynamic programming; see (Roulet et al., 2022) for more details. The global and local convergence analysis of the IDDP algorithm follow then from the convergence analysis of the ILQR algorithm after observing that for μ_h-strongly convex costs h_t, given Hypothesis 4.1, we have $\|\mathrm{DDP}_\nu(\mathcal{J})(\boldsymbol{u})-\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2\leq\eta\|\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2^2$ for some constant η independent of $\boldsymbol{u}$ and ν.

Theorem 4.14. *Consider h to be μ_h-strongly convex and Hypothesis 4.1 to be satisfied. Then the constant $\eta=\sup_{\boldsymbol{u}\in\mathbb{R}^{\tau n_u},\nu>0}\|\mathrm{DDP}_\nu(\mathcal{J})(\boldsymbol{u})-\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2/\|\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2^2$ is finite. Condition (37) is satisfied for a regularization*

$$\nu(\boldsymbol{u}) = L_g\beta\|\nabla h(g(\boldsymbol{u}))\|_2 + \rho_h\sigma_g^2{\theta_g}^2\chi^2\|\nabla h(g(\boldsymbol{u}))\|_2^2,$$

where $\beta=(1+\rho_h\rho_g)(1+2\chi)+\rho_g^3(2\theta_h)/(3\theta_g)$, $\chi=l_g\eta/L_g$ and ρ_h, ρ_g, θ_h, θ_g are defined in Theorem 4.11.

With such regularization, the number of iterations of the IDDP algorithm 8 to reach an accuracy ε is at most

$$k \leq 2\rho_h \ln\left(\frac{\delta_0}{\delta}\right) + 4\theta_g\beta(\sqrt{\delta_0} - \sqrt{\delta}) + 2\rho_h{\theta_g}^2\chi^2(\delta_0 - \delta) + O(\ln\ln(\varepsilon)),$$

where $\delta = 1/(32\rho_h(\theta_g\sqrt{\rho_h}(2 + 2\beta + \sqrt{\rho_h}\chi) + 4\theta_h)^2)$ *is the value of the gap determining the quadratic convergence phase.*

Remark 4.15. The complexity bounds for the IDDP algorithm in Theorem 4.14 takes then the same form as the complexity bounds obtained for the ILQR algorithm in Theorem 4.11 up to some additional multiplicative factors. Our proof is built on considering IDDP to approximate ILQR. In practice, IDDP appears more efficient than ILQR Roulet et al. (2022); Liao and Shoemaker (1992) and alternative proofs may better explain this phenomenon. On the other hand, our implementation and analysis provide global convergence guarantees that have not been considered before.

Proof of Theorem 4.14. We sketch the proof of the first part of the claim, whose technical details can be found in Lemma B.6. Given a command $\boldsymbol{u} = (u_0; \dots; u_{\tau-1})$ with associated trajectory $\boldsymbol{x} = g(\boldsymbol{u}) = (x_1; \dots; x_\tau)$, denote $\pi_t : y_t \to K_t y_t + k_t$ for $t \in \{0, \dots, \tau-1\}$ the affine policies computed in Algo. 1 and define for $\boldsymbol{y} = (y_1; \dots; y_\tau)$, $\pi(\boldsymbol{y}) = (\pi_0(0); \pi_1(y_1); \dots; \pi_{\tau-1}(y_{\tau-1}))$. Denoting then $\boldsymbol{v} = \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$, $\boldsymbol{w} = \mathrm{DDP}_\nu(\mathcal{J})(\boldsymbol{u})$, we have, after close inspection of the roll-outs,

$$\boldsymbol{v} = \pi(\nabla g(\boldsymbol{u})^\top \boldsymbol{v}), \qquad \boldsymbol{w} = \pi(g(\boldsymbol{u} + \boldsymbol{w}) - g(\boldsymbol{u})).$$

Denoting, for e_i the i$^{\text{th}}$ canonical vector in $\mathbb{R}^\tau$, $K = \sum_{i=2}^{\tau} e_i e_{i-1}^\top \otimes K_{i-1} \in \mathbb{R}^{\tau n_u \times \tau n_x}$, $k = (k_0; \dots; k_{\tau-1})$, $G = \nabla g(\boldsymbol{u})$, we get that $\boldsymbol{v} = k + KG^\top \boldsymbol{v}$. Since $G^\top$ is lower block triangular and K is strictly lower block triangular, $KG^\top$ is strictly lower block triangular and so $\mathrm{I} - KG^\top$ is invertible. Therefore, we can express the LQR oracle as $\boldsymbol{v} = (\mathrm{I} - KG^\top)^{-1}k$. For the IDDP oracle, a similar expression can be found by using the mean value theorem as formally shown in Lemma B.6. Informally, there exists a matrix D which can be thought as $\nabla g(\boldsymbol{u} + \zeta)$ for some $\|\zeta\|_2 \leq \|\boldsymbol{w}\|_2$ such that $\boldsymbol{w} = (\mathrm{I} - KD^\top)^{-1}k$. The difference $\boldsymbol{v} - \boldsymbol{w}$ can be bounded by $c_0\|k\|_2\|C^\top - D^\top\|_2$ for some constant c and $\|C^\top - D^\top\|_2$ can be bounded as $c_1\|\boldsymbol{w}\|_2$ such that we get in total a quadratic error bound in $\|k\|_2$ which can easily be converted in a quadratic bound in terms of $\|\boldsymbol{v}\|_2$.

For $\boldsymbol{u} \in^{\tau n_u}$ denote $\boldsymbol{v} = \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$, $\boldsymbol{w} = \mathrm{DDP}_\nu(\mathcal{J})(\boldsymbol{u})$. By definition of $\boldsymbol{v}$, condition (37) is satisfied if

$$\mathcal{J}(\boldsymbol{u} + \boldsymbol{w}) \leq \mathcal{J}(\boldsymbol{u}) + q_h^{g(\boldsymbol{u})} \circ \ell_g^{\boldsymbol{u}}(\boldsymbol{v}) + \frac{\nu}{2}\|\boldsymbol{v}\|_2^2.$$

We proceed by first observing that, by Lipschitz continuity of the gradients of h,

$$\mathcal{J}(\boldsymbol{u}+\boldsymbol{w}) - \mathcal{J}(\boldsymbol{u}+\boldsymbol{v}) \leq \nabla h(g(\boldsymbol{u}+\boldsymbol{v}))^\top (g(\boldsymbol{u}+\boldsymbol{w}) - g(\boldsymbol{u}+\boldsymbol{v})) + L_h\|g(\boldsymbol{u}+\boldsymbol{w}) - g(\boldsymbol{u}+\boldsymbol{v})\|_2^2/2,$$

and

$$\|\nabla h(g(\boldsymbol{u}+\boldsymbol{v}))\|_2 \leq \|\nabla h(g(\boldsymbol{u}))\|_2 + L_h\|g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u})\|_2.$$

Hence, using the Lipschitz continuity of g and the definition of η, we have

$$\mathcal{J}(\boldsymbol{u}+\boldsymbol{w}) - \mathcal{J}(\boldsymbol{u}+\boldsymbol{v}) \leq (\|\nabla h(g(\boldsymbol{u}))\|_2 + L_h l_g\|\boldsymbol{v}\|_2) l_g\eta\|\boldsymbol{v}\|_2^2 + L_h l_g^2\eta^2\|\boldsymbol{v}\|_2^4/2.$$

On the other hand, the term $\mathcal{J}(\boldsymbol{u}+\boldsymbol{v}) - \mathcal{J}(\boldsymbol{u}) - q_h^{g(\boldsymbol{u})} \circ \ell_g^{\boldsymbol{u}}(\boldsymbol{v})$ can be bounded using Lemma B.2. Hence, using that $\|\boldsymbol{v}\|_2 \leq \|\nabla h(g(\boldsymbol{u}))\|_2/(\mu_h\sigma_g)$ (see the first paragraph of the proof of Theorem 4.11), we get that condition (37) is satisfied for

$$\nu(\boldsymbol{u}) = L_g\beta\|\nabla h(g(\boldsymbol{u}))\|_2 + \rho_h\sigma_g^2{\theta_g}^2\chi^2\|\nabla h(g(\boldsymbol{u}))\|_2^2,$$

for $\beta = (1 + \rho_h\rho_g)(1 + 2\chi) + \rho_g^3(2\theta_h)/(3\theta_g)$, $\chi = l_g\eta/L_g$, where ρ_h, ρ_g, θ_h, θ_g are defined in Theorem 4.11.

With such regularization choice, the convergence of the IDDP method follows from the proof of Theorem 4.2 by using that condition (37) is satisfied. Namely, we get that the number of iterations of an IDDP algorithm with regularizations $\nu_k = \nu(\boldsymbol{u}^{(k)})$ to ensure an objective less than ε is at most (see Appendix C for the detailed derivation)

$$k \leq 2\rho_h \ln(\delta_0/\varepsilon) + 4\theta_g\beta(\sqrt{\delta_0} - \sqrt{\varepsilon}) + 2\rho_h{\theta_g}^2\chi^2(\delta_0 - \varepsilon). \tag{38}$$

For the local convergence, define $l, \sigma, L, \vartheta_h, \vartheta_g$ as in the proof of Theorem 4.11. We have

$$\lambda_h(g(\boldsymbol{u} + \boldsymbol{w})) \leq \|\nabla h(g(\boldsymbol{u} + \boldsymbol{w})) - \nabla h(g(\boldsymbol{u} + \boldsymbol{v}))\|_{g(\boldsymbol{u}+\boldsymbol{w})}^* + \|\nabla h(g(\boldsymbol{u} + \boldsymbol{v}))\|_{g(\boldsymbol{u}+\boldsymbol{w})}^*.$$

If $\lambda_h(g(\boldsymbol{u})) \leq \sigma/\sqrt{2\vartheta_h l\eta}$, then,

$$\|g(\boldsymbol{u}+\boldsymbol{w}) - g(\boldsymbol{u}+\boldsymbol{v})\|_{g(\boldsymbol{u}+\boldsymbol{w})} \leq l\|\boldsymbol{v}-\boldsymbol{w}\|_2 \leq l\eta\|\boldsymbol{v}\|_2^2 \leq l\eta\lambda_h(g(\boldsymbol{u}))^2/\sigma^2 \leq 1/(2\vartheta_h),$$

where we used that $\|\boldsymbol{v}\|_2 \leq \lambda_h(g(\boldsymbol{u}))/\sigma$ as shown in the second paragraph of the proof of Theorem 4.8. Hence, using Lemma B.5, we have that $\|\nabla h(g(\boldsymbol{u}+\boldsymbol{w})) - \nabla h(g(\boldsymbol{u}+\boldsymbol{v}))\|^*_{g(\boldsymbol{u}+\boldsymbol{w})} \leq 2l\eta\lambda_h(g(\boldsymbol{u}))^2/\sigma^2$ and using (Nesterov, 2018, Theorem 5.1.7), we have that $\|\nabla h(g(\boldsymbol{u}+\boldsymbol{v}))\|^*_{g(\boldsymbol{u}+\boldsymbol{w})} \leq 2\|\nabla h(g(\boldsymbol{u}+\boldsymbol{v}))\|^*_{g(\boldsymbol{u}+\boldsymbol{v})}$. We conclude that if $\lambda_h(g(\boldsymbol{u})) \leq 1/\sqrt{2\vartheta_h\vartheta_g\chi}$,

$$\lambda_h(g(\boldsymbol{u}+\boldsymbol{w})) \leq 2\chi\vartheta_g\lambda_h(g(\boldsymbol{u}))^2 + 2\lambda_h(g(\boldsymbol{u}+\boldsymbol{v})).$$

Hence, using the bound derived in Theorem 4.8 for $\lambda_h(g(\boldsymbol{u}+\boldsymbol{v}))$, we conclude that for

$$\lambda_h(g(\boldsymbol{u})) \leq 1/\max\{\sqrt{2\vartheta_h\vartheta_g}, \sqrt{2\vartheta_h\vartheta_g\chi}, 2\varrho\vartheta_h, 2\vartheta_h\},$$

we have that

$$\begin{aligned}\lambda_h(g(\boldsymbol{u}+\boldsymbol{w})) &\leq \left(2(2+\chi)\vartheta_g + 8\vartheta_h + 4\bar{\nu}\sigma^{-2}\right)\lambda_h(g(\boldsymbol{u}))^2 \\ &\leq \left(2\theta_g\sqrt{\rho_h}(2+2\beta+\chi) + 8\theta_h + 4\rho_h^2\chi^2{\theta_g}^2\lambda_h(g(\boldsymbol{u}))\right)\lambda_h(g(\boldsymbol{u}))^2,\end{aligned}$$

where we used that $\bar{\nu} = \nu(\boldsymbol{u})/\lambda_h(g(\boldsymbol{u})) \leq L_g\sqrt{L_h}\beta + L_h\rho_h\sigma_g^2{\theta_g}^2\chi^2\lambda_h(g(\boldsymbol{u}))$. Denote

$$\lambda = 1/(4(\theta_g\sqrt{\rho_h}(2+2\beta+\sqrt{\rho_h}\chi) + 4\theta_h)).$$

s.t. $\lambda \leq 1/\max\{\sqrt{2\vartheta_h\vartheta_g}, \sqrt{2\vartheta_h\vartheta_g\chi}, 2\varrho\vartheta_h, 2\vartheta_h\}$. For $\lambda_h(g(\boldsymbol{u})) < \lambda$, quadratic convergence is ensured, i.e., $\lambda_h(g(\boldsymbol{u}+\boldsymbol{w})) \leq \lambda^{-1}\lambda_h(g(\boldsymbol{u}))^2 < \lambda_h(g(\boldsymbol{u}))$. The conclusion follows as in the proof of Theorem 4.11. □

5 Conclusion

We have detailed global computational complexities of the ILQR and IDDP algorithms for problems of the form (1). Our analysis decomposes at several scales. At the scale of the whole trajectory, the problem can be summarized to a compositional objective and analyzed as a Gauss-Newton type algorithm. The trajectories can be detailed at the scale of the dynamic, which reveals the low computational cost of the optimization oracles. Finally, the dynamics can further be detailed in terms of the discretization scheme in order to ensure sufficient conditions for global convergence of the algorithms.

As future work, we may analyze constraints on the control variables while ensuring a gradient dominating-like property on the objective. Analyzing further the links between feedback linearization schemes and sufficient conditions for global optimality may also reveal the impact of the discretization stepsize on the overall condition number of the problem.

Acknowledgments. This work was supported by NSF DMS-1839371, DMS-2134012, CCF-2019844, CIFAR-LMB, NSF TRIPODS II DMS-2023166 and faculty research awards. The authors thank Dmitriy Drusvyatskiy, Alexander Liniger, Krishna Pillutla and John Thickstun for fruitful discussions on the paper and their help to develop the numerical experiments.

Notation	Definition	Interpretation	Unit
σ_f	$\inf_{x,u} \sigma_{\min}(\nabla_u f(x,u))$	Surj. param. of $v \to \nabla_u f(x,u)^\top v$	x/u
l_f^x	$\sup_u l_{f(\cdot,u)}$	Lip. cont. of $f(\cdot,u)$ for any u	1
l_f^u	$\sup_x l_{f(x,\cdot)}$	Lip. cont. of $f(x,\cdot)$ for any x	x/u
L_f^{xx}	$\sup_u l_{\nabla_x f(\cdot,u)}$	Bound on $\|\nabla_{xx} f^2(x,u)\|$	$1/x$
L_f^{uu}	$\sup_x l_{\nabla_u f(x,\cdot)}$	Bound on $\|\nabla_{uu} f^2(x,u)\|$	x/u^2
L_f^{xu}	$\sup_x l_{\nabla_u f(\cdot,u)}$	Bound on $\|\nabla_{xu} f^2(x,u)\|$	$1/u$
$\sigma_g, \sigma_{f^{[\tau]}}$	$\sigma_f/(1+l_f^x)$	Lower bound on $\sigma_{\min}(\nabla f^{[\tau]}(x_0, \boldsymbol{u}))$	x/u
$l_g, l_{f^{[\tau]}}$	$l_f^u S$	Lip. cont. of $f^{[\tau]}(x_0,u)$	x/u
$L_g, L_{f^{[\tau]}}$	$L_f^{xx}(l_f^u S)^2 + 2L_f^{xu} l_f^u S + L_f^{uu} S$	Smoothness param. of $f^{[\tau]}(x_0,u)$	x/u^2
S	$\sum_{t=0}^{\tau-1} (l_f^x)^t$	Auxiliary constant	1
μ_h	$\inf_x \sigma_{\min}(\nabla^2 h_t(x))$	Strong convexity param. of the costs	h/x^2
L_h	$\sup_x \sigma_{\max}(\nabla^2 h_t(x))$	Smoothness param. of the costs	h/x^2
M_h	$l_{\nabla^2 h_t}$	Lip. cont. of Hessians of the costs	h/x^3
ρ_g	l_g/σ_g	Cond. nb of $\nabla g(\boldsymbol{u})$	1
ρ_h	L_h/μ_h	Cond. nb of the costs	1
θ_h, ϑ_h	$M_h/(2\mu_h^{3/2})$	Self-concordance of the costs	$1/\sqrt{h}$
θ_g	$L_g/(\sigma_g^2\sqrt{\mu_h})$	Scaling param. for g	$1/\sqrt{h}$
α	$4\rho_g^2(2\rho_g^2\theta_h/(3\theta_g)+\rho_h)$	Cond. nb for global conv. of ILQR	1
l	$\sqrt{L_h} l_g$	Lip. cont. of g w.r.t. h in Hyp. 4.1	$\sqrt{h}/u$
L	$\sqrt{L_h} L_g$	Smoothness of g w.r.t. h in Hyp. 4.1	$\sqrt{h}/u^2$
σ	$\sqrt{\mu_h}\sigma_g$	Surj. param. of g w.r.t. h	$\sqrt{h}/u$
ϱ	$l/\sigma = \sqrt{\rho_h}\rho_g$	Cond. nb of g w.r.t. h	1
ϑ_g	$L/\sigma^2 = \sqrt{\rho_h}\theta_g$	Scaling param. of g w.r.t. h	$1/\sqrt{h}$
η	See Lemma B.6	Relative bound btw DDP & LQR: $\frac{\|\mathrm{DDP}_\nu(\mathcal{J})(\boldsymbol{u}) - \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2}{\|\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2^2} \leq \eta$	$1/u$
χ	$l_g\eta/L_g$	Factor of smoothness for IDDP	1

Table 1: Index of constants used in the paper.

A Index of constants

Table 1 presents an index of the constants used in the paper with their units. We denote the unit of the control variables, the states and the costs as, respectively, u, x and h and use 1 if the constant has no dimension.

Note that all constants are rooted in assumptions about the dynamic f and the individual costs h_t of problem (1). In particular, constants governing the compositional problem (20) defined by the total cost h and the control g in τ steps of f for fixed initial state (see (20) and Def. 3.1), are all explicitly given in terms of the constants of f, h_t. Moreover, note that the constants governing the dynamic f can be further decomposed by considering the dynamic as the control in k of a dynamic as presented in Sec. 3.2.

For simplicity, we present only the strongly convex case. For the gradient dominating case with exponent $r \neq 1/2$ we refer the reader to Theorem 4.2. For the local convergence, constants σ, l, L, θ_h can be defined without strong convexity as presented in Hypothesis 4.7.

B Helper Lemmas

B.1 Reparameterization in Brunovsky form

Lemma B.1 shows how a system driven by its k^{th} derivative can be expressed in the canonical Brunovsky form (15) (Brunovskỳ, 1970).

Lemma B.1. *Consider the Euler discretization of a single-input continuous-time system driven by its $n_x{}^{th}$ derivative as presented in (16). If $|\partial_v \psi(y,v)| > 0$ for all $y \in \mathbb{R}^{n_x}, v \in \mathbb{R}$ then the dynamical system (16) can be linearized by static feedback into the canonical form (15).*

Proof. Denoting $A = \mathrm{I} + \Delta D$, with D the upper-shift matrix in $\mathbb{R}^{n_x}$, the original dynamical system (16) can be written as $y_{t+1} = Ay_t + \Delta\psi(y_t, v_t)e$, with $e = e_{n_x}$ the $n_x{}^{\text{th}}$ canonical vector in $\mathbb{R}^{n_x}$. It suffices to note that the matrix A is similar to a matrix of the form $B = D + ec^\top$ for some vector c. Namely, denoting $P_{n_x} = (p_{n_x,1}, \dots, p_{n_x,n_x})^\top$ the $n_x{}^{\text{th}}$ lower triangular Pascal matrix defined by rows $p_{n_x,i} = (\binom{i-1}{j-1})_{j=1}^{n_x}$ with the convention $\binom{i}{j} = 0$ if $i < j$ and $Q = P_{n_x}\,\mathbf{diag}((\Delta^{i-n_x})_{i=1}^{n_x})$, we get that $BQ = QA$ for $B = D + ec^\top$ with $c = ((-1)^{n_x-i}\binom{n_x}{i-1})_{i=1}^{n_x}$.

Hence by considering the change of variable $z_t = a(y_t) = Qy_t$, we get that

$$z_{t+1} = Bz_t + \Delta\psi(y_t, v_t)Qe = Dz_t + c^\top z_t e + \Delta\psi(y_t, v_t)e,$$

using that $Qe = e$. By defining $w_t = b(y_t, v_t) = c^\top Qy_t + \Delta\psi(y_t, v_t)$ we get the desired form (15). The transformation a is a diffeomorphism since Q is invertible. The transformations $b(y_t, \cdot)$ are also diffeomorphisms since $|\partial_v\psi(y, v)| > 0$ for all $y \in \mathbb{R}^{n_x}, v \in \mathbb{R}$. □

B.2 Global convergence analysis of ILQR

Lemma B.2 states that a linear quadratic approximation of the compositional objective in (20) approximates the objective up to a cubic error.

Lemma B.2. *Given Hypothesis 4.1, we have, for problem (20),*

$$|(h\circ g)(\boldsymbol{u}+\boldsymbol{v}) - (h\circ g)(\boldsymbol{u}) - q_h^{g(\boldsymbol{u})}\circ \ell_g^{\boldsymbol{u}}(\boldsymbol{v})| \leq \frac{L_g\|\nabla h(g(\boldsymbol{u}))\|_2 + (M_h l_g^3/3 + L_g L_h l_g)\|\boldsymbol{v}\|_2}{2}\|\boldsymbol{v}\|_2^2.$$

Proof. We have for any $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{\tau n_u}$,

$$\begin{aligned}|h(g(\boldsymbol{u}+\boldsymbol{v})) - h(g(\boldsymbol{u})) - q_h^{g(\boldsymbol{u})}(\ell_g^{\boldsymbol{u}}(\boldsymbol{v}))| &\leq |h(g(\boldsymbol{u}+\boldsymbol{v})) - h(g(\boldsymbol{u})) - q_h^{g(\boldsymbol{u})}(g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u}))| \\ &\quad + |q_h^{g(\boldsymbol{u})}(g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u})) - q_h^{g(\boldsymbol{u})}(\ell_g^{\boldsymbol{u}}(\boldsymbol{v}))|.\end{aligned}$$

On one hand, we have, by Taylor-Lagrange inequality,

$$|h(g(\boldsymbol{u}+\boldsymbol{v})) - h(g(\boldsymbol{u})) - q_h^{g(\boldsymbol{u})}(g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u}))| \leq \frac{M_h}{6}\|g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u})\|_2^3 \leq \frac{M_h l_g^3}{6}\|\boldsymbol{v}\|_2^3.$$

On the other hand, we have,

$$\begin{aligned}|q_h^{g(\boldsymbol{u})}(g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u})) - q_h^{g(\boldsymbol{u})}(\ell_g^u(\boldsymbol{v}))| &= \Big|(g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u}) - \nabla g(\boldsymbol{u})^\top \boldsymbol{v})^\top \nabla h(g(\boldsymbol{u})) \\ &\quad + \frac{1}{2}(g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u}) - \nabla g(\boldsymbol{u})^\top \boldsymbol{v})^\top \nabla^2 h(g(\boldsymbol{u}))(g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u}) + \nabla g(\boldsymbol{u})^\top \boldsymbol{v})\Big| \\ &\leq \frac{L_g\|\nabla h(g(\boldsymbol{u}))\|_2}{2}\|v\|_2^2 + \frac{L_h L_g l_g}{2}\|\boldsymbol{v}\|_2^3.\end{aligned}$$

□

Lemma B.3 refines the regularization choice of Theorem 4.2 by exploiting an additional assumption of strong convexity of the costs.

Lemma B.3. *Consider h to be μ_h-strongly convex and Hypothesis 4.1 to be satisfied. Condition (23) is satisfied by choosing a regularization*

$$\nu(\boldsymbol{u}) = L_g\|\nabla h(g(\boldsymbol{u}))\|_2 + \frac{2l_g^2(M_h l_g^2/3 + L_g L_h)\|\nabla h(g(\boldsymbol{u}))\|_2}{L_g\|\nabla h(g(\boldsymbol{u}))\|_2 + \sigma_g l_g \mu_h}.$$

Proof. Let $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$, $G = \nabla g(\boldsymbol{u})$, $H = \nabla^2 h(g(\boldsymbol{u}))$. We have

$$\begin{aligned}\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u}) &= -G(G^\top G)^{-1}(H + \nu(G^\top G)^{-1})^{-1}\nabla h(g(\boldsymbol{u})) \\ &= -G(G^\top G)^{-1/2}((G^\top G)^{1/2} H (G^\top G)^{1/2} + \nu\,\mathrm{I})^{-1}(G^\top G)^{1/2}\nabla h(g(\boldsymbol{u})).\end{aligned}$$

We have then

$$\begin{aligned}\|\operatorname{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2/\|\nabla h(g(\boldsymbol{u}))\|_2 &\leq \min\{l_g^2/(\mu_h\sigma_g l_g^2+\nu\sigma_g), l_g/(\nu+\mu_h\sigma_g^2)\}\\ &\leq 2l_g/(\nu(1+\sigma_g/l_g)+\mu_h\sigma_g(\sigma_g+l_g))\\ &\leq 2l_g/(\nu+\mu_h\sigma_g l_g),\end{aligned}$$

where we used that $\min\{a,b\} \leq 2/(1/a+1/b)$. Hence condition (23) is satisfied if ν satisfies $a_1 + a_2/(a_3+\nu) \leq \nu$ with $a_1 = L_g\|\nabla h(g(\boldsymbol{u}))\|_2$, $a_2 = 2a_0 l_g\|\nabla h(g(\boldsymbol{u}))\|_2$, $a_3 = \sigma_g l_g \mu_h$, $a_0 = M_h l_g^3/3 + L_g L_h l_g$. Hence condition (23) is satisfied for $\nu \geq \nu_0 = (a_1 - a_3 + (a_1+a_3)\sqrt{1+4a_2(a_1+a_3)^{-2}})/2$. The result follows using that since $\sqrt{1+2x} \leq 1+x$, we have $\nu_0 \leq a_1 + a_2/(a_1+a_3)$. □

B.3 Local convergence analysis of ILQR

Lemma B.4 provides a bound on the oracle returned by an ILQR method in terms of the constants introduced in Theorem 4.8.

Lemma B.4. *Given Hypothesis 4.7 on problem (20), we have for any $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$, $\nu \geq 0$,*

$$\|\operatorname{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2 \leq \frac{l}{l\sigma+\nu}\|\nabla h(g(\boldsymbol{u}))\|^*_{g(\boldsymbol{u})}.$$

Proof. For $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$, $\nu \geq 0$, denoting $\nabla^2 h(g(\boldsymbol{u})) = H$, $\nabla g(\boldsymbol{u}) = G$, we have

$$\operatorname{LQR}_\nu(\mathcal{J})(\boldsymbol{u}) = -GH^{1/2}(H^{1/2}G^\top G H^{1/2} + \nu\,\mathrm{I})^{-1}H^{-1/2}\nabla h(g(\boldsymbol{u})).$$

Recall that by definition of σ and l, we have $\sigma \leq \sigma_{\min}(GH^{1/2}), \sigma_{\max}(GH^{1/2}) \leq l$. By considering the singular value decomposition of $GH^{1/2}$, we then have

$$\|GH^{1/2}(H^{1/2}G^\top G H^{1/2} + \nu\,\mathrm{I})^{-1}\|_2 \leq \max_{x\in[\sigma,l]} \frac{x}{\nu+x^2} = \begin{cases} \frac{\sigma}{\sigma^2+\nu} & \text{if } \nu \leq \sigma^2 \\ \frac{1}{2\sqrt{\nu}} & \text{if } \sigma^2 \leq \nu \leq l^2 \\ \frac{l}{l^2+\nu} & \text{if } \nu \geq l^2 \end{cases}.$$

By analyzing each case, we get the claimed inequality. □

Lemma B.5 provides a bound on the differences of gradients of a self-concordant function. It replaces the classical bound we can have for Lipschitz-continuous gradients.

Lemma B.5. *For a ϑ_h-self-concordant strictly convex function h (Nesterov, 2018, Definition 5.1.1) and y, x such that $\|y-x\|_x < 1/\vartheta_h$, we have,*

$$\|\nabla h(y) - \nabla h(x)\|^*_x \leq \frac{1}{1-\vartheta_h\|y-x\|_x}\|y-x\|_x.$$

Proof. Denote $J = \int_0^1 \nabla^2 h(x+t(y-x))dt$ and $H = \nabla^2 h(x)$, we have $\|\nabla h(y) - \nabla h(x)\|^*_x = \|J(y-x)\|^*_x = \|H^{-1/2}JH^{-1/2}\|_2\|y-x\|_x$. Now $H^{-1/2}JH^{-1/2} \succeq 0$ since h is strictly convex and by (Nesterov, 2018, Corollary 5.1.5), we have $J \preceq \nabla^2 h(x)/(1-\vartheta_h\|y-x\|_x)$ hence $\|H^{-1/2}JH^{-1/2}\|_2 \leq 1/(1-\vartheta_h\|y-x\|_x)$. □

B.4 Convergence analysis of IDDP

Lemma B.6 shows that the oracle returned by a IDDP algorithm is close to the oracle returned by an ILQR algorithm up to a square error.

Lemma B.6. *For $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$ and $\nu > 0$, denote $\boldsymbol{v} = \operatorname{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$, $\boldsymbol{w} = \operatorname{DDP}_\nu(\mathcal{J})(\boldsymbol{u})$. Given Hypothesis 4.1 and μ_h-strongly convex costs h_t, we have that $\|\boldsymbol{v}-\boldsymbol{w}\|_2 \leq \eta\|\boldsymbol{v}\|_2^2$ for η a constant independent of ν and $\boldsymbol{u}$.*

Proof. Let $\boldsymbol{u} = (u_0; \ldots; u_{\tau-1})$, $\nu > 0$, denote $\boldsymbol{w} = \operatorname{DDP}_\nu(\boldsymbol{u})$. Denote $\boldsymbol{z} = (z_1; \ldots; z_\tau) = g(\boldsymbol{u}+\boldsymbol{w}) - g(\boldsymbol{u})$ s.t. $z_{t+1} = f(x_t+z_t, u_t+w_t) - f(x_t, u_t)$ for $t \in \{0, \ldots, \tau-1\}$, with $z_0 = 0$. By the mean value theorem, for all

$t \in \{0, \dots, \tau-1\}$, there exists $\zeta_{t,1}, \dots, \zeta_{t,n_x} \in \mathbb{R}^{n_x}$, $\eta_{t,1}, \dots, \eta_{t,n_x} \in \mathbb{R}^{n_u}$ s.t. for all $i \in \{1, \dots, , n_x\}$, denoting f_i the i$^{\text{th}}$ coordinate of f, we have

$$f_i(x_t + z_t, u_t + w_t) - f_i(x_t, u_t + w_t) = \nabla_{x_t+\xi_{t,i}} f_i(x_t + \xi_{t,i}, u_t + w_t)^\top z_t$$
$$f_i(x_t, u_t + w_t) - f_i(x_t, u_t) = \nabla_{u_t+\eta_{t,i}} f_i(x_t, u_t + \eta_{t,i})^\top z_t,$$

with $\|\zeta_{t,i}\|_2 \leq \|z_t\|_2$ and $\|\eta_{t,i}\|_2 \leq \|w_t\|_2$. We can then write the dynamics of z_t as

$$z_{t+1} = Z_t^\top z_t + W_t^\top w_t \quad \text{for } t \in \{0, \dots, \tau - 1\}$$
$$\text{where } Z_t = \sum_{i=1}^{n_x} e_i^\top \otimes \nabla_{x_t+\xi_{t,i}} f_i(x_t + \xi_{t,i}, u_t + w_t) \qquad W_t = \sum_{i=1}^{n_x} e_i^\top \otimes \nabla_{u_t+\eta_{t,i}} f_i(x_t, u_t + \eta_{t,i}).$$

Denoting $Z = \sum_{t=1}^{\tau-1} e_t e_{t+1} \otimes Z_t$, $W = \sum_{t=1}^{\tau} e_t e_t^\top \otimes W_{t-1}$ and $D = W(\mathrm{I} - Z)^{-1}$ we get that $g(\boldsymbol{u}+\boldsymbol{w}) - g(\boldsymbol{u}) = D^\top \boldsymbol{w}$. Since $\boldsymbol{w} = \pi(g(\boldsymbol{u}+\boldsymbol{w}) - g(\boldsymbol{u}))$, we get that $\boldsymbol{w} = k + KD^\top \boldsymbol{w}$. Since $D^\top$ is lower block triangular, we have that $\mathrm{I} - KD^\top$ is invertible and so the IDDP oracle can be written $\boldsymbol{w} = (\mathrm{I} - KD^\top)^{-1} k$.

Combining the expressions of $\boldsymbol{v}$ presented in the beginning of the proof of Theorem 4.14 and the expression of $\boldsymbol{w}$ presented above, we then get that

$$\|\boldsymbol{v} - \boldsymbol{w}\|_2 \leq \|(\mathrm{I} - KG^\top)^{-1}\|_2 \|(\mathrm{I} - KD^\top)^{-1}\|_2 \|K\|_2 \|G^\top - D^\top\|_2 \|k\|_2.$$

With the notations of Lemma 3.2, we have $G = U(\mathrm{I} - X)^{-1}$ with $U = \nabla_{\boldsymbol{u}} F(\boldsymbol{x}, \boldsymbol{u})$ and $X = \nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u})$. Hence we get that

$$G - D = (U - W)(\mathrm{I} - X)^{-1} + W(\mathrm{I} - X)^{-1}(X - Z)(\mathrm{I} - Z)^{-1}.$$

Using the block structures of the matrices, we get $\|W\|_2 \leq \sqrt{n_x} l_f^u$, $\|Z\|_2 \leq \sqrt{n_x} l_f^x$, $\|U - W\|_{2,2} \leq \sqrt{n_x} L_f^{uu} \|\boldsymbol{w}\|_2$ since $\|\eta_{t,i}\|_2 \leq \|w_t\|_2$, $\|X - Z\|_{2,2} \leq 2\sqrt{n_x}(L_f^{xx} \|\boldsymbol{z}\|_2 + L_f^{xu} \|\boldsymbol{w}\|_2)$ since $\|\xi_{t,i}\|_2 \leq \|z_t\|_2$. We have that $(\mathrm{I} - X)^{-1} = \sum_{t=0}^{\tau-1} X^t$ since $X^\tau = 0$ and similarly $(\mathrm{I} - Z)^{-1} = \sum_{t=0}^{\tau-1} Z^t$. So, using that $\|\boldsymbol{z}\|_2 \leq l_g \|\boldsymbol{w}\|_2$ with $l_g = l_f^u S$ for $S = \sum_{t=0}^{\tau-1} (l_f^x)^t$, we get

$$\|G - D\|_2 \leq \beta_0 \|\boldsymbol{w}\|_2, \quad \text{where } \beta_0 = \sqrt{n_x} S(L_f^{uu} + 2\sqrt{n_x} l_f^u S'(L_f^{xx} l_g + L_f^{xu})),$$

and $S' = \sum_{t=0}^{\tau-1} (\sqrt{n_x} l_f^x)^t$. In addition, we have that $\|\boldsymbol{w}\|_2 \leq \|(\mathrm{I} - KD^\top)^{-1}\|_2 \|k\|_2$, hence we have

$$\|\boldsymbol{v} - \boldsymbol{w}\|_2 \leq \beta_0 \|(\mathrm{I} - KG^\top)^{-1}\|_2 \|(\mathrm{I} - KD^\top)^{-1}\|_2^2 \|K\|_2 \|k\|_2^2.$$

Using Lemma B.7 with simplified bounds, we have that

$$\|K\|_2 \leq \rho_h \rho_f \qquad \|k\|_2 \leq \left(\sum_{t=0}^{\tau-1} (l_f^x \rho_h)^t \right) \|\nabla h(\boldsymbol{x})\|_2 / (\sigma_f \mu_h (1 + \nu / ((l_f^u)^2 \mu_h))),$$

where $\rho_h = L_h / \mu_h$ and $\rho_f = l_f^x / \sigma_f$. In addition, one can verify that $\|\nabla h(\boldsymbol{x})\|_2 \leq l_g (L_h + \nu \sigma_g^{-2}) \|\boldsymbol{v}\|_2$ by using the expression of the LQR step derived in Sec. 4.1. So we get after simplifications

$$\frac{\|k\|_2}{\|\boldsymbol{v}\|_2} \leq \frac{l_g}{\sigma_f} \frac{L_h}{\mu_h} \frac{l_f^2}{\sigma_g^2} \left(\sum_{t=0}^{\tau-1} (l_f \rho_h)^t \right) = \rho_h \rho_f \rho_g \frac{l_f}{\sigma_g} \left(\sum_{t=0}^{\tau-1} (l_f \rho_h)^t \right).$$

Plugging the bound on $\|\nabla h(\boldsymbol{x})\|_2$ and bounding the remaining terms, we get that

$$\|\boldsymbol{v} - \boldsymbol{w}\|_2 \leq \eta \|\boldsymbol{v}\|_2^2 \qquad \text{where } \eta \leq \beta_0 \rho_h^3 \rho_g^2 \rho_f \frac{(l_f^u)^4}{\sigma_f^2 \sigma_g^2} S_1 S_2^2 S_3^2,$$

with $S_1 = \sum_{t=0}^{\tau-1} \left(\rho_h \rho_f l_f^u \sum_{s=1}^{\tau} (l_f^x)^s \right)^t$, $S_2 = \sum_{t=0}^{\tau-1} \left(\rho_h \rho_f \sqrt{n_x} l_f^u \sum_{s=0}^{\tau-1} (\sqrt{n_x} l_f^x)^s \right)^t$, $S_3 = \sum_{t=0}^{\tau-1} (l_f^x \rho_h)^t$. □

Lemma B.7 estimates the smoothness properties of the policies computed by Algo. 1 in terms of the smoothness constants of the dynamics and the costs.

Lemma B.7. *Define $\pi : (y_1; \dots; y_\tau) \to (\pi_0(0); \pi_1(y_1); \dots; \pi_{\tau-1}(y_{\tau-1}))$ for π_t the policies computed in Algo. 1 and let K, k be such that $\pi(\boldsymbol{y}) = K\boldsymbol{y} + k$ for $\boldsymbol{y} \in \mathbb{R}^{\tau n_x}$. Given Hypothesis 4.1 and μ_h-strongly convex costs h_t, we have*

$$\|K\|_2 \leq \frac{l_f^x L_h}{\sigma_f \mu_h} \frac{1}{1+\nu/((l_f^u)^2\mu_h)}, \|k\|_2 \leq \frac{\|\nabla h(\boldsymbol{x})\|_2}{\sigma_f \mu_h (1+\nu/((l_f^u)^2\mu_h))} \sum_{t=0}^{\tau-1} \left(\frac{L_h}{\mu_h} \frac{l_f^x}{1+\nu^{-1}\sigma^2 L_h} \right)^t.$$

Proof. Consider a K_t, k_t, J_t, j_t defined in Algo. 1 for a command $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$ and a regularization $\nu > 0$. By recursion, we have that J_t is positive definite, since $J_t = P_t + A_t^\top J_{t+1}^{1/2}(\mathrm{I}+\nu^{-1}J_{t+1}^{1/2}B_tB_t^\top)J_{t+1}^{1/2}A_t$ and P_t is positive definite.

Denote $J = \sum_{t=1}^\tau e_t e_t^\top \otimes J_t$, $P = \sum_{t=1}^\tau e_t e_t^\top \otimes P_t$, $K = \sum_{t=2}^\tau e_t e_{t-1}^\top \otimes K_{t-1}$, $X = \sum_{t=1}^{\tau-1} e_t e_{t+1}^\top \otimes A_t^\top$, $U = \sum_{t=1}^\tau e_t e_t^\top \otimes B_{t-1}^\top$, $j = (j_1; \dots; j_\tau)$, $k = (k_0; \dots; k_{\tau-1})$, $p = (p_1; \dots p_\tau)$. We have then in matrix form

$$K = -(\nu \mathrm{I} + UJU^\top)^{-1}UJX^\top, \quad J = P + XJX^\top - XJU^\top(\nu \mathrm{I} + UJU^\top)^{-1}UJX^\top$$
$$k = -(\nu \mathrm{I} + UJU^\top)^{-1}Uj, \qquad j = p + X(\mathrm{I} - JU^\top(\nu \mathrm{I} + UJU^\top)^{-1}U)j.$$

Using that $U^\top U$ and J are invertible, we get by standard linear algebra manipulations that

$$K = -U(U^\top U)^{-1}(\nu(U^\top U)^{-1} + J)^{-1}JX^\top, \quad J = P + X(J^{-1} + \nu^{-1}U^\top U)^{-1}X^\top,$$
$$k = -U(U^\top U)^{-1}(\nu(U^\top U)^{-1} + J)^{-1}j, \quad j = (\mathrm{I} - X(J^{-1} + \nu^{-1}U^\top U)^{-1}J^{-1})^{-1}p.$$

We have then $J \succeq P \succeq \mu_h \mathrm{I}$. Moreover, $\|J\|_2$ satisfies $\|J\|_2 \leq L_h + (l_f^x)^2/(1/\|J\|_2 + \nu^{-1}\sigma_f^2)$. So

$$\|J\|_2 \leq \left((l_f^x)^2 + L_h\nu^{-1}\sigma_f^2 - 1 + \sqrt{(L_h\nu^{-1}\sigma_f^2 + (l_f^x)^2 + 1)^2 - 4(l_f^x)^2} \right) / (\nu^{-1}\sigma_f^2).$$

Using that $\sqrt{1+x} \leq 1 + x/2$, we conclude that $\|J\|_2 \leq L_h$. The final bounds follow from the above decompositions using that $(\mathrm{I} - C)^{-1} = \sum_{t=0}^{\tau-1} C^t$ for $C = X(J^{-1} + \nu^{-1}U^\top U)^{-1}J^{-1}$, since C is nilpotent, i.e., $C^\tau = 0$. □

C Detailed Computations

In this Appendix, we detail some of the technical computations done in the paper.

C.1 Details on the Proof of Theorem 3.5

Details on Eq. (18). Note that we naturally have $\nabla_{\boldsymbol{v}}\Phi(\boldsymbol{y}, \boldsymbol{v}) = \mathbf{diag}((\nabla_{v_t}\phi(y_t, v_t))_{t=0}^{k-1})$, such that, by definition of the operator $\mathbf{diag}$,

$$\begin{aligned}
\nabla_{\boldsymbol{v}}\Phi(\boldsymbol{y}, \boldsymbol{v}) &= \sum_{t=1}^k e_t e_t^\top \otimes \nabla_{v_{t-1}}\phi(y_{t-1}, v_{t-1}) \\
&= \sum_{t=1}^k e_t e_t^\top \otimes \partial_{v_{t-1}} b(y_{t-1}, v_{t-1}) e^\top \nabla a(y_{t+1})^{-1} \\
&= \left(\sum_{t=1}^k \partial_{v_{t-1}} b(y_{t-1}, v_{t-1}) e_t e_t^\top \otimes 1 \right) (\mathrm{I} \otimes e^\top) \left(\sum_{t=1}^k e_t e_t^\top \otimes \nabla a(y_{t+1})^{-1} \right), \\
&= \mathbf{diag}((\partial_{v_t} b(y_t, v_t))_{t=0}^{k-1})(\mathrm{I} \otimes e^\top)\, \mathbf{diag}((\nabla a(y_{t+1})^{-1})_{t=0}^{k-1})
\end{aligned}$$

where $k = n_x$ and we use that $(A \otimes B)(C \otimes D) = (AC \otimes BD)$ for A, B, C, D of appropriate sizes and 1 is the identity in $\mathbb{R}^1$. Similarly, one has that

$$\begin{aligned}
\nabla_{\boldsymbol{y}}\Phi(\boldsymbol{y}, \boldsymbol{v}) &= \sum_{t=1}^{k-1} e_t e_{t+1}^\top \otimes \nabla_{y_t}\phi(y_t, v_t) \\
&= \left(\sum_{t=1}^{k-1} e_t e_{t+1}^\top \otimes \mathrm{I} \right) \left(\sum_{t=1}^k e_t e_t^\top \otimes \nabla_{y_{t-1}}\phi(y_{t-1}, v_{t-1}) \right) = (D \otimes \mathrm{I})\, \mathbf{diag}(\nabla_{y_t}\phi(y_t, v_t)_{t=0}^{k-1})
\end{aligned}$$

for $D = \sum_{t=1}^{k-1} e_t e_{t+1}^\top$ the upper-shift matrix in $\mathbb{R}^k = \mathbb{R}^{n_x}$. On the other hand, we have

$$\begin{aligned}
\mathbf{diag}(\nabla_{y_t}\phi(y_t, v_t)_{t=0}^{k-1}) &= \sum_{t=0}^{k-1} e_{t+1}e_{t+1}^\top \otimes \left(\nabla a(y_t) D^\top + \nabla_{y_t} b(y_t, v_t) e^\top\right) \nabla a(y_{t+1})^{-1} \\
&= \underbrace{\sum_{t=0}^{k-1} e_{t+1}e_{t+1}^\top \otimes \nabla a(y_t) D^\top \nabla a(y_{t+1})^{-1}}_{A} + \underbrace{\sum_{t=0}^{k-1} e_{t+1}e_{t+1}^\top \otimes \nabla_{y_t} b(y_t, v_t) e^\top \nabla a(y_{t+1})^{-1}}_{B} \\
A &= \left(\sum_{t=0}^{k-1} e_{t+1}e_{t+1}^\top \otimes \nabla a(y_t)\right) (\mathrm{I} \otimes D^\top) \left(\sum_{t=0}^{k-1} e_{t+1}e_{t+1}^\top \otimes \nabla a(y_{t+1})^{-1}\right) \\
&= \mathbf{diag}((\nabla a(y_t))_{t=0}^{k-1})(\mathrm{I} \otimes D^\top)\, \mathbf{diag}((\nabla a(y_{t+1})^{-1})_{t=0}^{k-1}) \\
B &= \left(\sum_{t=0}^{k-1} e_{t+1}e_{t+1}^\top \otimes \nabla_{y_t} b(y_t, v_t)\right) (\mathrm{I} \otimes e^\top) \left(\sum_{t=0}^{k-1} e_{t+1}e_{t+1}^\top \otimes \nabla a(y_{t+1})^{-1}\right) \\
&= \mathbf{diag}((\nabla_{y_t} b(y_t, v_t))_{t=0}^{k-1})(\mathrm{I} \otimes e^\top)\, \mathbf{diag}((\nabla a(y_{t+1})^{-1})_{t=0}^{k-1}).
\end{aligned}$$

□

Details on line (ii) in Eq. (19). Denoting $K_t = \nabla a(y_t)$, we have, using that $De_t = e_{t-1}$, $FA = (D \otimes \mathrm{I})(\sum_{t=1}^n e_t e_t^\top \otimes K_{t-1}) = \sum_{t=2}^n e_{t-1}e_t^\top \otimes K_{t-1}) = \sum_{t=1}^{n-1} e_t e_{t+1}^\top K_t$ and, using that $e_t^\top D = e_{t+1}^\top$, $CF = (\sum_{t=1}^n e_t e_t^\top \otimes K_t)(D \otimes \mathrm{I}) = \sum_{t=1}^{n-1} e_t e_{t+1}^\top \otimes K_t$. Therefore, we have $FA = CF$ and similarly we can show that $FA^{-1} = C^{-1}F$. □

Details on line (iii) in Eq. (19). Since $D = \sum_{t=1}^{k-1} e_t e_{t+1}^\top$, we have $D^j = \sum_{t=1}^{k-j} e_t e_{t+j}^\top$ hence $D^k = 0$. Therefore $F \otimes G = D \otimes D^\top$ is nilpotent of order k. Hence $(\mathrm{I} - F \otimes G)^{-1} = \sum_{t=0}^{k-1} D^t \otimes (D^\top)^t$ and so, for $F = D \otimes \mathrm{I}$, we have $(\mathrm{I} - F \otimes G)^{-1} F = (\sum_{t=0}^{k-1} D^t \otimes (D^\top)^t)(D \otimes \mathrm{I}) = \sum_{t=1}^{k-1} D^t \otimes (D^\top)^{t-1}$. □

Details on the extension to multi inputs. Consider the multi-input case as described in Def. 3.4. For any $k \geq r$, $j \in \{1, \ldots, m_u\}$, $i \in \{1, \ldots, r_j\}$, we have $\zeta_{k,j}^{(i)} = w_{k+i-r_j-1}^{(j)}$. Denote $T = \sum_{i=1}^k \sum_{j=1}^{m_u} e_j e_i^\top \otimes e_i e_j^\top$ for e_i, e_j canonical vectors of, respectively, $\mathbb{R}^k$ and $\mathbb{R}^{m_u}$. For $\boldsymbol{w} = (w_0; \ldots; w_{k-1})$, we have that $\boldsymbol{\omega} = T\boldsymbol{w}$ reorders the coordinates of $\boldsymbol{w}$ such that $\boldsymbol{\omega} = (\omega_1; \ldots; \omega_{m_u})$ with $\omega_j^{(i)} = w_{i-1}^{(j)}$ for $i \in \{1, \ldots, k\}$, $j \in \{1, \ldots, m_u\}$. Hence we have for any $k \geq r$, $j \in \{1, \ldots, m_u\}$, denoting here e_i the ith canonical vector in $\mathbb{R}^{r_j}$, D_{r_j} the upper-shift matrix in $\mathbb{R}^{r_j}$,

$$\begin{aligned}
\zeta_{k,j} &= (D_{r_j}^{k-1} e_{r_j}, \ldots, D_{r_j} e_{r_j}, e_{r_j})\omega_j \\
&= (\underbrace{0_{r_j}, \ldots, 0_{r_j}}_{k-r_j}, e_1, \ldots, e_{r_j})\omega_j = (0_{r_j}, \ldots, 0_{r_j}, \mathrm{I}_{r_j})\omega_j := C_j \omega_j
\end{aligned}$$

where 0_{r_j} is the null vector in $\mathbb{R}^{r_j}$ and I_{r_j} is the identity matrix in $\mathbb{R}^{r_j}$. So we get that

$$z_k = \left(\sum_{j=1}^{m_u} e_j e_j^\top \otimes C_j\right) \boldsymbol{\omega} = \left(\sum_{j=1}^{m_u} e_j e_j^\top \otimes C_j\right) T\boldsymbol{w},$$

i.e., $z_k = M\boldsymbol{w}$ with $\sigma_{\min}(M^\top) = 1$.

Consider $k = r$. We have, with the notations of the proof of Theorem 3.5, $y_{t+1} = a^{-1}(MB(\phi^{[k]}(y_0, \boldsymbol{v}), \boldsymbol{v}))$. Hence,

$$\nabla_{\boldsymbol{v}} \phi^{\{k\}}(y_0, \boldsymbol{v}) = \left(\nabla_{\boldsymbol{v}} B(\boldsymbol{y}, \boldsymbol{v}) + \nabla_{\boldsymbol{v}} \Phi(\boldsymbol{y}, \boldsymbol{v})(\mathrm{I} - \nabla_{\boldsymbol{y}} \Phi(\boldsymbol{y}, \boldsymbol{v}))^{-1} \nabla_{\boldsymbol{y}} B(\boldsymbol{y}, \boldsymbol{v})\right) M^\top \nabla a(y_k)^{-1}.$$

The discrete time dynamic can be written

$$y_{t+1} = a^{-1}(Ja(y_t) + Kb(y_t, v_t))$$

with, denoting $e_{j,i}$ the i^{th} canonical vector in $\mathbb{R}^{r_j}$ and e_{ℓ_j} the ${\ell_j}^{\text{th}}$ canonical vector in $\mathbb{R}^{n_x}$.

$$J = \begin{pmatrix} D_{r_1} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & D_{r_{m_u}} \end{pmatrix}, \quad D_{r_j} = \sum_{i=1}^{r_j} e_{j,i} e_{j,i+1}^\top, \quad K = \sum_{j=1}^{m_u} e_{\ell_j} e_j^\top, \quad \ell_j = \sum_{s=1}^{j} r_s.$$

Hence, we have for $t \in \{0, \dots, \tau - 1\}$,

$$\begin{aligned} \nabla_{v_t} \phi(y_t, v_t) &= \nabla_{v_t} b(y_t, v_t) K^\top \nabla a(y_{t+1})^{-1} \\ \nabla_{y_t} \phi(y_t, v_t) &= \left(\nabla a(y_t) J^\top + \nabla_{y_t} b(y_t, v_t) K^\top \right) \nabla a(y_{t+1})^{-1}. \end{aligned}$$

The rest of the proof follows as in the proof of Theorem 3.5 by redefining $E = \mathrm{I} \otimes K^\top$, $G = \mathrm{I} \otimes J^\top$, $F = D_k \otimes \mathrm{I}$ for D_k the upper-shift matrix in $\mathbb{R}^k = \mathbb{R}^r$, $V = \mathbf{diag}((\nabla_{v_t} b(y_t, v_t))_{t=0}^{k-1})$. We then get

$$\nabla_{\boldsymbol{v}} \phi^{\{k\}}(y_0, \boldsymbol{v}) \nabla a(y_k) = V \left(\mathrm{I} - \left(\sum_{i=1}^{k-1} D_k^i \otimes K^\top (J^\top)^i \right) A^{-1} Y \right)^{-1} M^\top.$$

The result follows for $k = r$ and for $k > r$ the same reasoning as in the single input case applies. □

C.2 Details on Theorem 4.2

In the statement of Theorem 4.2, we used Lemma C.1 with $\mu_t = \mu$ for all t such that $\mu_h = \|\boldsymbol{\mu}^{-1}\|_q^{-1} = \mu \tau^{-q} = \mu / \tau^{2r/(2r-1)}$.

Lemma C.1. *Let $h_1, \dots, h_\tau$ be differentiable functions from $\mathbb{R}^{n_x} \to \mathbb{R}$ such that*

$$\|\nabla h_t(x_t)\|_2 \geq \mu_t^r (h_t(x_t) - h_t^*)^r \quad \text{for } t \in \{1, \dots, \tau\},$$

for some constants $\mu_t \geq 0$, $r \in [1/2, 1)$. The function $h : \boldsymbol{x} = (x_1; \dots; x_\tau) \to \sum_{t=1}^\tau h_t(x_t)$ satisfies

$$\|\nabla h(\boldsymbol{x})\|_2 \geq \mu_h^r (h(\boldsymbol{x}) - h^*)^r \quad \text{for } \mu_h = \|\boldsymbol{\mu}^{-1}\|_q^{-1},$$

for $q = 2r/(2r-1)$ and $\boldsymbol{\mu}^{-1} = (\mu_1^{-1}, \dots, \mu_\tau^{-1})^\top$ with $\|\boldsymbol{\mu}^{-1}\|_{+\infty}^{-1} = \min_{t \in \{1, \dots, \tau\}} \mu_t$ if $r = 1/2$.

Proof. Denoting for simplicity $\delta_t = h_t(x_t) - h^*$, we have

$$\|\nabla h(\boldsymbol{x})\|_2^2 = \sum_{t=1}^\tau \|\nabla h_t(x_t)\|_2^2 \geq \sum_{t=1}^\tau (\mu_t \delta_t)^{2r} = \|\boldsymbol{\mu} \odot \boldsymbol{\delta}\|_{2r}^{2r} \geq \frac{1}{\|\boldsymbol{\mu}^{-1}\|_q^{2r}} (\boldsymbol{\delta}^\top \mathbf{1})^{2r},$$

for $q = 2r/(2r-1)$, where $\boldsymbol{\mu} = (\mu_1, \dots, \mu_\tau)^\top$, $\boldsymbol{\delta} = (\delta_1, \dots, \delta_\tau)^\top$, $\odot$ denotes the element-wise product and we used Hölder's inequality $\|x\|_p \|y\|_q \leq |x^\top y|$ for $p = 2r$, $q = p/(p-1) = 2r/(2r-1)$, $x = \boldsymbol{\mu} \odot \boldsymbol{\delta}$ and $y = \boldsymbol{\mu}^{-1}$. Plugging the values of δ in the inequality above, we get

$$\|\nabla h(\boldsymbol{x})\|_2 \geq \|\boldsymbol{\mu}^{-1}\|_q^{-r} \left(\sum_{t=1}^\tau h_t(x_t) - h_t^* \right)^r = \mu_h^r (h(\boldsymbol{x}) - h^*)^r,$$

where we used that, since h is decomposable in the variables x_t, $h^* = \sum_{t=1}^\tau h_t^*$. □

C.3 Details on Theorem 4.11

Details on Eq. (36). Let $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$ and $\boldsymbol{v} = \mathrm{LQR}_{\nu(\boldsymbol{u})}(\mathcal{J})(\boldsymbol{u})$ for

$$\nu(\boldsymbol{u}) = L_g \|\nabla h(g(\boldsymbol{u}))\|_2 + \frac{2 l_g^2 (M_h l_g^2 / 3 + L_g L_h) \|\nabla h(g(\boldsymbol{u}))\|_2}{L_g \|\nabla h(g(\boldsymbol{u}))\|_2 + \sigma_g l_g \mu_h}.$$

As in Eq. (25), in the proof of Theorem 4.2, we get that

$$\mathcal{J}(\boldsymbol{u}+\boldsymbol{v})-\mathcal{J}(\boldsymbol{u}) \leq -\frac{1}{2}\frac{\sigma_g^2}{\sigma_g^2 L_h+\nu(\boldsymbol{u})}\|\nabla h(g(\boldsymbol{u}))\|_2^2 = -\frac{b_1x^3+b_2x^2}{b_3x^2+b_4x+1},$$

where $x=\|\nabla h(g(\boldsymbol{u}))\|_2$, $b_1=L_g/(2l_g\mu_h L_h\sigma_g)$, $b_2=1/(2L_h)$, $b_3=L_g^2/(\sigma_g^3 l_g\mu_h L_h)$, $b_4=L_g/(\sigma_g l_g\mu_h)+L_g/(\sigma_g^2 L_h)+2a_0/(\sigma_g^3\mu_h L_h)$. The function $f_1(x)=(b_1x^3+b_2x^2)/(b_3x^2+b_4x+1)$ is increasing and since h is strongly convex, we have that $\|\nabla h(g(\boldsymbol{u}))\|_2^2 \geq \mu_h(h(g(\boldsymbol{u}))-h^*)=\mu_h\delta$ for $\delta=\mathcal{J}(\boldsymbol{u})-\mathcal{J}^*$. Hence, as in the proof of Theorem 4.2, we get that the total number of iterations to reach an accuracy ε is at most $k\leq f_2(\delta_0)-f_2(\varepsilon)$ where

$$f_2'(\delta)=\frac{1}{f_1(\sqrt{\mu_h\delta})}=\frac{1+c_1\delta^{1/2}+c_2\delta}{c_3\delta+c_4\delta^{3/2}},$$

where $c_1=\theta_g(\rho_g^{-1}+2\rho_g+\rho_h^{-1})+4\rho_g^3\theta_h/(3\rho_h)$, $c_2={\theta_g}^2/(\rho_g\rho_h)$, $c_3=1/(2\rho_h)$, $c_4=\theta_g/(2\rho_g\rho_h)$. By standard integration, we have that an antiderivative of f_2' is

$$\begin{aligned} f_2(x) &= \frac{\ln(\delta)}{c_3}+\frac{2c_2}{c_4}\sqrt{\delta}-2\frac{(c_2c_3^2-c_4c_1c_3+c_4^2)}{c_3c_4^2}\ln(c_4\sqrt{\delta}+c_3) \\ &= 2\rho_h\ln(\delta)+4\theta_g\sqrt{\delta}+8\rho_g^2(\rho_h+2\rho_g^2\theta_h/(3\theta_g))\ln(\theta_g\sqrt{\delta}/(2\rho_h\rho_g)+1/(2\rho_h)). \end{aligned}$$

The result in Eq. (36) follows. □

C.4 Details on Theorem 4.14

Detaitls on Eq. (38). With the notations of Theorem 4.2, we have that

$$\delta_{k+1}-\delta_k \leq -\frac{1}{2}\frac{\sigma_g^2x^2}{\sigma_g^2L_h+\beta x+\rho_h\chi^2x^2}$$

with $x=\|\nabla h(g(\boldsymbol{u}^{(k)}))\|_2$ and $\delta_k=\mathcal{J}(\boldsymbol{u}^{(k)})-\mathcal{J}^*$. The function $f_1: x\to\sigma_g^2x^2/(2(\sigma_g^2L_h+\beta x+\rho_h\chi^2x^2))$ is strictly increasing, so we can follow the steps of the proof Theorem 4.2 and obtain that $f_2(\delta_k)(\delta_{k+1}-\delta_k)\leq -1$ with

$$f_2'(\delta)=\frac{1}{f_1(\sqrt{\mu_h\delta})}=2\rho_h\frac{1}{\delta}+2\theta_g\beta\frac{1}{\sqrt{\delta}}+2{\theta_g}^2\rho_h\chi^2$$

The result follows by integrating f_2' and, as in the proof Theorem 4.2, we have that convergence to an accuracy ε is ensured after at most $k\leq f_2(\delta_0)-f_2(\varepsilon)$. □

References

Aranda-Bricaire, E., Kotta, Ü., and Moog, C. (1996). Linearization of discrete-time systems. *SIAM Journal on Control and Optimization*, 34(6):1999–2023.

Arrow, K. (1968). Applications of control theory to economic growth. In *Lectures on Applied Mathematics*, volume 12 (Mathematic of the Decision Science, Part 2).

Belikov, J., Kaldmäe, A., and Kotta, Ü. (2017). Global linearization approach to nonlinear control systems: a brief tutorial overview. In *Proceedings of the Estonian Academy of Sciences*, volume 66, pages 243–263.

Bergou, E. H., Diouane, Y., and 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.

Bertsekas, D. (2017). *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, 4th edition.

Betts, J. (2010). *Practical methods for optimal control and estimation using nonlinear programming*. SIAM.

Bolte, J., Nguyen, T. P., Peypouquet, J., and Suter, B. W. (2017). From error bounds to the complexity of first-order descent methods for convex functions. *Mathematical Programming*, 165(2):471–507.

Boyd, S. and Vandenberghe, L. (1997). Semidefinite programming relaxations of non-convex problems in control and combinatorial optimization. In *Communications, Computation, Control, and Signal Processing*, pages 279–287. Springer.

Brunovskỳ, P. (1970). A classification of linear controllable systems. *Kybernetika*, 6(3):173–188.

Dan, H., Yamashita, N., and Fukushima, M. (2002). Convergence properties of the inexact Levenberg-Marquardt method under local error bound conditions. *Optimization methods and software*, 17(4):605–626.

Di, B. and Lamperski, A. (2019). Newton's method and differential dynamic programming for unconstrained nonlinear dynamic games. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 4073–4078.

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

Drusvyatskiy, D. and Paquette, C. (2019). Efficiency of minimizing compositions of convex functions and smooth maps. *Mathematical Programming*, 178(1):503–558.

Dunn, J. and 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.

Fan, J.-Y. and Yuan, Y.-X. (2005). On the quadratic convergence of the Levenberg-Marquardt method without nonsingularity assumption. *Computing*, 74(1):23–39.

Fazel, M., Ge, R., Kakade, S., and Mesbahi, M. (2018). Global convergence of policy gradient methods for the linear quadratic regulator. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1467–1476.

Giftthaler, M., Neunert, M., Stäuble, M., Buchli, J., and Diehl, M. (2018). A family of iterative Gauss-Newton shooting methods for nonlinear optimal control. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9.

Grizzle, J. and Kokotovic, P. (1988). Feedback linearization of sampled-data systems. *IEEE Transactions on Automatic Control*, 33(9):857–859.

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

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

Jakubczyk, B. (1987). Feedback linearization of discrete-time systems. *Systems & Control Letters*, 9(5):411–416.

Jakubczyk, B. and Sontag, E. (1990). Controllability of nonlinear discrete-time systems: A Lie-algebraic approach. *SIAM Journal on Control and Optimization*, 28(1):1–33.

Jayaraman, G. and Chizeck, H. (1993). Feedback linearization of discrete-time systems. In *Proceedings of the 32nd IEEE Conference on Decision and Control (CDC)*, pages 2972–2977.

Kakade, S., Krishnamurthy, A., Lowrey, K., Ohnishi, M., and Sun, W. (2020). Information theoretic regret bounds for online nonlinear control. *arXiv preprint arXiv:2006.12466*.

Kamien, M. and Schwartz, N. (1971). Sufficient conditions in optimal control theory. *Journal of Economic Theory*, 3(2):207–214.

Li, W. and 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. and Shoemaker, C. (1991). Convergence in unconstrained discrete-time differential dynamic programming. *IEEE Transactions on Automatic Control*, 36(6):692–706.

Liao, L.-Z. and Shoemaker, C. A. (1992). Advantages of differential dynamic programming over Newton's method for discrete-time optimal control problems. Technical report, Cornell University.

Łojasiewicz, S. (1963). A topological property of real analytic subsets. *Coll. du CNRS, Les équations aux dérivées partielles*, 117(87-89):2.

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

Mayne, D. and Polak, E. (1975). First-order strong variation algorithms for optimal control. *Journal of Optimization Theory and Applications*, 16(3):277–301.

Messerer, F., Baumgärtner, K., and Diehl, M. (2021). Survey of sequential convex programming and generalized Gauss-Newton methods. *ESAIM. Proceedings and Surveys*, 71:64.

Moré, J. (1978). The Levenberg-Marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer.

Murray, D. and 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.

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

Nesterov, Y. (2007). Modified Gauss-Newton scheme with worst case guarantees for global performance. *Optimisation methods and software*, 22(3):469–483.

Nesterov, Y. (2018). *Lectures on convex optimization*. Springer.

Pantoja, J. (1988). Differential dynamic programming and Newton's method. *International Journal of Control*, 47(5):1539–1553.

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

Polyak, B. (1964). Some methods of speeding up the convergence of iteration methods. *USSR computational mathematics and mathematical physics*, 4(5):1–17.

Pontryagin, L., Boltyansky, V., Gamkrelidze, R., and Mischenko, E. (1963). The mathematical theory of optimal processes. *Wiley-Interscience*.

Rao, C., Wright, S., and Rawlings, J. (1998). Application of interior-point methods to model predictive control. *Journal of optimization theory and applications*, 99(3):723–757.

Recht, B. (2019). A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:253–279.

Roulet, V., Srinivasa, S., Drusvyatskiy, D., and Harchaoui, Z. (2019). Iterative linearized control: stable algorithms and complexity guarantees. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5518–5527.

Roulet, V., Srinivasa, S., Fazel, M., and Harchaoui, Z. (2022). Iterative linear quadratic optimization for nonlinear control: Differentiable programming algorithmic templates. Technical report, Department of Statistics, University of Washington. http://staff.washington.edu/vroulet/ilqc_techreport.pdf.

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

Simchowitz, M. and Foster, D. (2020). Naive exploration is optimal for online LQR. In *Proceedings of the 37th International Conference on Machine Learning*, pages 8937–8948.

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

Tassa, Y., Erez, T., and Todorov, E. (2012). Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913.

Tassa, Y., Mansard, N., and Todorov, E. (2014). Control-limited differential dynamic programming. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1168–1175.

Ueda, K. and Yamashita, N. (2010). On a global complexity bound of the Levenberg-Marquardt method. *Journal of optimization theory and applications*, 147(3):443–453.

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

Wright, S. (1991). Partitioned dynamic programming for optimal control. *SIAM Journal on optimization*, 1(4):620–642.

Xu, W. and Anitescu, M. (2019). Exponentially convergent receding horizon strategy for constrained optimal control. *Vietnam Journal of Mathematics*, 47(4):897–929.

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

Zhao, R. and Fan, J. (2016). Global complexity bound of the Levenberg-Marquardt method. *Optimization Methods and Software*, 31(4):805–814.