Journal of Machine Learning Research X (2025) XX-XX Submitted X/XX; Published XX/XX

# On Global and Local Convergence of Iterative Linear Quadratic Optimization Algorithms for Discrete Time Nonlinear Control

**Vincent Roulet** VROULET@GOOGLE.COM
*Google DeepMind*
*Seattle, WA, USA*

**Siddhartha Srinivasa** SIDDH@CS.UW.EDU
*Paul G. Allen School of Computer Science and Engineering*
*University of Washington*
*Seattle, WA, USA*

**Maryam Fazel** MFAZEL@ECE.UW.EDU
*Department of Electrical and Computer Engineering*
*University of Washington*
*Seattle, WA, USA*

**Zaid Harchaoui** ZAID@UW.EDU
*Department of Statistics*
*University of Washington*
*Seattle, WA, USA*

**Editor:**

## 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 provide detailed convergence guarantees to stationary points as well as local linear convergence rates for the Iterative Linear Quadratic Regulator (ILQR) algorithm and its differential dynamic programming variant (iLQR). For problems without costs on control variables, we observe that global convergence to minima can be ensured provided that the linearized discrete time dynamics are surjective, costs on the state variables are gradient dominated. We further detail quadratic local convergence when the costs are self-concordant. We show that surjectivity of the linearized dynamics hold for 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, Differentiable Dynamic Programming, Gradient Dominance, Feedback Linearization.

## 1. Introduction

We consider nonlinear control problems in discrete time of the form

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

where at the time index $t$, $x_t$ is the state of the system, $u_t$ is the control applied to the system, $f_t$ is the discretized nonlinear dynamic, $h_t$ is the cost applied to the system state and the control variable, and $\bar{x}_0$ is a given fixed initial state.

Problems of the form (1) have been tackled in various ways, from direct approaches using nonlinear optimization (Jacobson and Mayne, 1970; Bock and Plitt, 1984; Pantoja, 1988; Dunn and Bertsekas, 1989; Wright, 1990, 1991a; Rao et al., 1998; Betts, 2010) to convex relaxations using semidefinite optimization (Boyd and Vandenberghe, 1997). Numerous packages exist for such problems such as CasAdi (Andersson et al., 2018), Pyomo (Bynum et al., 2021), JumP (Dunning et al., 2017), IPOPT (Wächter and Biegler, 2006), or SNOPT (Gill et al., 2005), Crocoddyl (Jallet et al., 2023), acados (Verschueren et al., 2021). 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, 2004; Sideris and Bobrow, 2005; Tassa et al., 2012). The resulting feedback policy can then be applied on the linearized dynamics as in the Iterative Linear Quadratic Regulator (ILQR) algorithm (Rawlings et al., 2017, Section 8.8.5), (Li and Todorov, 2004; Sideris and Bobrow, 2005). Alternatively, the feedback policy can be applied on the original dynamics, as in the iterative Linear Quadratic Regulator (iLQR) algorithm (Rawlings et al., 2017, Sectin 8.8.6), (Tassa et al., 2012), akin to a Differential Dynamic Programming (DDP) approach (Mayne, 1966). To avoid confusion, we name this second approach Iterative Dynamic Differentiable Programming (IDDP).

**Motivation.** 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; Recht, 2019; Kakade et al., 2020; Simchowitz and Foster, 2020; Westenbroek et al., 2021). The empirical performance of the ILQR and IDDP algorithms are illustrated in Fig. 1. The first problem considered in Fig. 1 consists in swinging up a pendulum to a vertical position in finite time, the second problem consists in controlling a simple model of a car to be at predefined positions at given times. The detailed experimental setting is presented in Appendix I. Most importantly, the costs consists in quadratic state costs bounded below by 0, i.e., of the form $h_t(x_t,u_t) = (x_t-\hat{x}_t)^\top Q_t(x_t-\hat{x}_t)$ for $Q_t$ positive definite and $\hat{x}_t$ a reference state. In Fig. 1, we plot $c^{(k)}/c^{(0)}$ in log-scale, where $c^{(k)}\geq 0$ denotes the total cost at iteration $k$ computed by means of a gradient descent, an ILQR algorithm or an IDPP algorithm, and $c^{(0)}$ denotes an initial cost given by initializing the control variables at 0. We observe that both the ILQR and the IDDP algorithms converge to an optimal cost, i.e., $c^{(k)}\to 0$. Moreover, both algorithms outperform a simple gradient descent and appear to exhibit a fast convergence

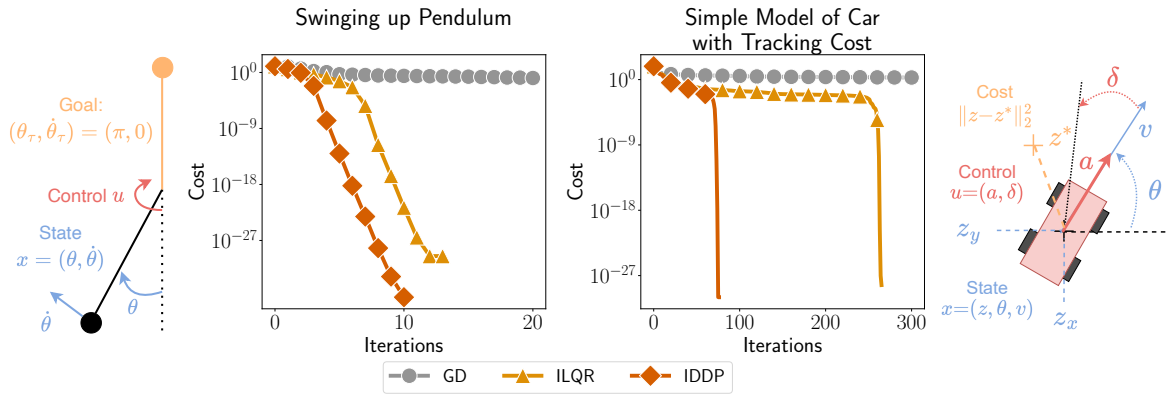


Figure 1: Cost along iterations of ILQR, IDDP and Gradient Descent (GD) on two discrete time nonlinear control problems detailed in Section 5.

after some iterations. The empirical behavior illustrated in Fig. 1 does not hold for any nonlinear control problem as illustrated in Appendix I with a more realistic model of a car taken from Liniger et al. (2015). Yet, the examples presented in Fig. 1 are surprising from an optimization viewpoint as the problems considered escape the usual paradigm of convex or linear optimization.

The empirical efficiency of ILQR and IDDP on some nonlinear control problems as the ones illustrated in Fig. 1 motivates then the following questions.

1. *What conditions on a discrete time nonlinear control problem ensure algorithms such as ILQR and IDDP converge to a globally optimal solution?*

2. *What convergence behaviors can we expect from these algorithms?*

We first present generic convergence results for the ILQR and IDDP algorithms that ensure their global convergence to stationary points and local convergence to minima in Theorems 2 and 3. However, the aforementioned convergence results do not explain the convergence to *global minima* observed in Fig. 1.

We then turn our attention to nonlinear control problems without control costs and with time-invariant dynamics $f$, i.e., problems of the form

$$\min_{\substack{u_0,\dots,u_{\tau-1}\in\mathbb{R}^{n_u}\\ x_0,\dots,x_\tau\in\mathbb{R}^{n_x}}} \quad \sum_{t=1}^{\tau} h_t(x_t) \tag{2}$$
$$\text{subject to} \quad x_{t+1} = f(x_t, u_t) \quad \text{for } t \in \{0,\dots,\tau-1\}, \qquad x_0 = \bar{x}_0.$$

Considering time-invariant dynamics make clearer the relationship with the underlying continuous dynamical system. Generalizations to time-variant systems are pointed out when applicable. However, problems of the form (2) conserve the main challenge of generic discrete time control problems (1), that is, the nonlinearity of the dynamics, which prevent us from using classical results from convex analysis even if the state costs are convex. The

nonlinearity of the dynamics distinguish problems (2) from the linear quadratic settings recently studied by, e.g., (Fazel et al., 2018; Zhang et al., 2019, 2020; Sun and Fazel, 2021; Lin et al., 2021) for which global convergence of policy methods have been shown by means of algebraic considerations. The absence of costs on controls variables restrict the problem class compared to problems of the form (1). However, this also allows focussing on the properties of the dynamics to understand the properties of non-convex problems (2) and pave the way to analyze generic problems of the form (1).

**Approach.** 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 $\nabla_u f(x,u)^\top \in \mathbb{R}^{n_x \times n_u}$ 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 $\sigma$ such that

$$\forall x, u \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}, \quad \sigma_{\min}(\nabla_u f(x,u)) \geq \sigma > 0, \tag{3}$$

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. (3) 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** *In problem (2), 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 (3), then a regularized ILQR or IDDP algorithm converges to a* global minimum *with a* local quadratic convergence rate.

Our analysis is based on decomposing the problem at several scales. At the scale 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 (3) 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 (3) 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.

**Outline.** We start by presenting classical nonlinear control algorithms for problem (2), i.e., the Iterative Linear Quadratic Regulator (ILQR) and the iterative Linear Quadratic Regulator (IDDP) algorithms, in Sec. 2.1 and Sec. 2.2, and cast them as closed-box oracles. We provide convergence guarantees to stationary points of both algorithms for generic problems of the form (1), as well as linear local convergence guarantees in Sec. 2.3. We analyze the properties of problem (2) 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 in, respectively, Sec. 4.2, Sec. 4.3. In particular, in Sec. 4.2.1, we demonstrate the *global convergence* of the ILQR algorithm provided that the costs are gradient dominated, the dynamics have surjective linearizations (3) 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 (3) and both costs and dynamics satisfy appropriate smoothness conditions in Sec. 4.2.2. Theorem 1 is detailed for the ILQR algorithm in Sec. 4.2.3 and convergence of the IDDP algorithm is analyzed in Sec. 4.3. Numerical experiments are presented in Section 5 to assess the theoretical findings. We discuss related work in Section 6.

Additional numerical illustrations of the ILQR and IDDP algorithms can be found in the companion report (Roulet et al., 2022) and reproduced or further explored by using the companion toolbox `https://github.com/vroulet/ilqc`.

**Summary of contributions.** For problems of the form (1), we demonstrate global convergence to stationary points and local linear convergence to minima of both ILQR and IDDP algorithms under usual regularity assumptions (Theorems 2, 3). For problems of the form (2), we make the following contributions.

1. We present sufficient conditions for global convergence to a minimum of the problem through the lens of a *gradient-dominating property* of the objective. Namely, we show that a gradient-dominating property of the objective can be decomposed into the properties of the discrete time dynamic and ensured for appropriate discretization schemes (Lemma 6, Theorem 10).
2. We prove that the ILQR algorithm *converges globally to a minimum* if the cost is smooth, gradient dominated, and if the dynamic is smooth with non-singular transpose Jacobians (3) (Theorem 13).
3. We prove that the ILQR algorithm *converges locally with a quadratic rate* if the cost is smooth and self-concordant, and if the dynamic is smooth with non-singular transpose Jacobians (3) (Theorem 18).
4. We show and detail the *global and local convergence* to minima of both ILQR and IDDP algorithms for smooth and strongly convex costs and smooth dynamic with non-singular transpose Jacobians (3) (Theorems 21 and 24).
5. Inspired from the theoretical findings, we also present a line-search variant of the ILQR algorithm that keep the same global and local convergence guarantees to minima, while not requiring any knowledge of problems constants (Corollary 23).

**Notations.** For a sequence of vectors $x_1, \dots, x_\tau \in \mathbb{R}^{n_x}$, we denote by semicolons 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 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 transpose Jacobian 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 $f : \mathbb{R}^d \to \mathbb{R}^n$, $h : \mathbb{R}^n \to \mathbb{R}$, 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 $l_f = \sup_{x,y\in\mathbb{R}^d, x\neq y} \|f(x)-f(y)\|_2/\|x-y\|_2$ the Lipschitz-continuity constant of $f$. For a matrix $A \in \mathbb{R}^{d\times n}$, 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 $\boldsymbol{u} = (u_0; \dots; u_{\tau-1}) \in \mathbb{R}^{\tau n_u}$ and can be written as

$$\begin{aligned} \mathcal{J}(\boldsymbol{u}) = \quad & \sum_{t=0}^{\tau-1} h_t(x_t, u_t) + h_\tau(x_\tau) \\ \text{s.t.} \quad & x_{t+1} = f_t(x_t, u_t), \quad \text{for } t \in \{0, \dots, \tau-1\}, \qquad x_0 = \bar{x}_0. \end{aligned} \tag{4}$$

Problem (1) consists then in minimizing $\mathcal{J}$. In the following, we always *asumme that* $\mathcal{J}$ *has at least one minimizer* $\boldsymbol{u}^*$. The classical ILQR ((Li and Todorov, 2004; Sideris and Bobrow, 2005)), and IDDP algorithms (Tassa et al., 2012) compute the next iterate as $\boldsymbol{u}_{\text{next}} = \boldsymbol{u} + \text{Oracle}_\nu(\mathcal{J})(\boldsymbol{u})$, for given control variables $\boldsymbol{u}$. Here, $\text{Oracle}_\nu(\mathcal{J})$ is an oracle, which, given a regularization parameter $\nu$ and control variables $\boldsymbol{u}$, 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, 2004; Tassa et al., 2012). Our implementation is a variant that leads to non-asymptotic convergence guarantees of these algorithms (Roulet et al., 2019).

### 2.1 Iterative Linear Quadratic Regulator

Given control variables $\boldsymbol{u} = (u_0; \dots; u_{\tau-1})$ 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

$$\begin{aligned}\text{LQR}_\nu(\mathcal{J})(\boldsymbol{u}) = \underset{v_0,\dots,v_{\tau-1}\in\mathbb{R}^{n_u}}{\arg\min} & \sum_{t=0}^{\tau-1} \begin{pmatrix} p_t \\ q_t \end{pmatrix}^\top \begin{pmatrix} y_t \\ v_t \end{pmatrix} + \frac{1}{2}\begin{pmatrix} y_t \\ v_t \end{pmatrix}^\top \begin{pmatrix} P_t & R_t \\ R_t^\top & Q_t \end{pmatrix} \begin{pmatrix} y_t \\ v_t \end{pmatrix} + \frac{\nu}{2}\|v_t\|_2^2 \\ & + p_\tau^\top y_\tau + \frac{1}{2} y_\tau^\top P_\tau y_\tau \\ \text{s.t.} \quad & y_{t+1} = A_t y_t + B_t v_t, \text{ for } t \in \{0,\dots,\tau-1\}, y_0 = 0, \end{aligned} \tag{5}$$

$$\begin{aligned}\text{with} \quad & P_\tau = \nabla^2_{x_\tau x_\tau} h_\tau(x_\tau),\ p_\tau = \nabla_{x_\tau} h_\tau(x_\tau), \\ & P_t = \nabla^2_{x_t x_t} h_t(x_t,u_t),\ p_t = \nabla_{x_t} h_t(x_t,u_t), && \text{for } t \in \{0,\dots,\tau-1\}, \\ & Q_t = \nabla^2_{u_t u_t} h_t(x_t,u_t),\ q_t = \nabla_{u_t} h_t(x_t,u_t), && \text{for } t \in \{0,\dots,\tau-1\}, \\ & R_t = \nabla^2_{u_t x_t} h_t(x_t,u_t) && \text{for } t \in \{0,\dots,\tau-1\}, \\ & A_t = \nabla_{x_t} f_t(x_t,u_t)^\top,\ B_t = \nabla_{u_t} f_t(x_t,u_t)^\top, && \text{for } t \in \{0,\dots,\tau-1\}. \end{aligned}$$

The minimum above is well-defined as long as either the costs are convex, or the regularization $\nu$ is large enough. The implementation of the ILQR oracle is presented in Algo. 1. Its computational scheme is illustrated in Fig. 2.

Problem (5) 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 (5).

Problem (5) 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 $c_\tau(y_\tau) = \frac{1}{2} y_\tau^\top P_\tau y_\tau + p_\tau^\top y_\tau$,

$$\begin{aligned} c_t(y_t) = \min_{v_t,\dots v_{\tau-1}\in\mathbb{R}^{n_u}} & \sum_{s=t}^{\tau-1} \begin{pmatrix} p_s \\ q_s \end{pmatrix}^\top \begin{pmatrix} y_s \\ v_s \end{pmatrix} + \frac{1}{2}\begin{pmatrix} y_s \\ v_s \end{pmatrix}^\top \begin{pmatrix} P_s & R_s \\ R_s^\top & Q_s \end{pmatrix} \begin{pmatrix} y_s \\ v_s \end{pmatrix} + \frac{\nu}{2}\|v_s\|_2^2 \\ & + p_\tau^\top y_\tau + \frac{1}{2} y_\tau^\top P_\tau y_\tau \\ \text{s.t.} \quad & y_{s+1} = A_s y_s + B_s v_s, \text{ for } s \in \{t,\dots,\tau-1\}, \\ = \min_{v_t\in\mathbb{R}^{n_u}} & \left\{ \begin{pmatrix} p_t \\ q_t \end{pmatrix}^\top \begin{pmatrix} y_t \\ v_t \end{pmatrix} + \frac{1}{2}\begin{pmatrix} y_t \\ v_t \end{pmatrix}^\top \begin{pmatrix} P_t & R_t \\ R_t^\top & Q_t \end{pmatrix} \begin{pmatrix} y_t \\ v_t \end{pmatrix} + \frac{\nu}{2}\|v_t\|_2^2 + c_{t+1}(A_t y_t + B_t v_t) \right\} \end{aligned} \tag{6}$$

$$= \frac{1}{2} y_t^\top J_t y_t + y_t^\top j_t, \tag{7}$$

where $J_t$, $j_t$ are computed recursively in line 10 of Algo. 1. The optimal control at time $t$ from state $y_t$ is then given by an affine policy

$$\begin{aligned} \pi_t(y_t) &= \underset{v_t\in\mathbb{R}^{n_u}}{\arg\min} \left\{ \begin{pmatrix} p_t \\ q_t \end{pmatrix}^\top \begin{pmatrix} y_t \\ v_t \end{pmatrix} + \frac{1}{2}\begin{pmatrix} y_t \\ v_t \end{pmatrix}^\top \begin{pmatrix} P_t & R_t \\ R_t^\top & Q_t \end{pmatrix} \begin{pmatrix} y_t \\ v_t \end{pmatrix} + \frac{\nu}{2}\|v_t\|_2^2 + c_{t+1}(A_t y_t + B_t v_t) \right\} \\ &= K_t y_t + k_t, \end{aligned} \tag{8}$$

where $K_t, k_t$ are computed in line 11 of Algo. 1. The cost-to-go functions and policies are well-defined as long as all costs $h_t$ are convex or if the regularization $\nu$ is large enough (see e.g. (Roulet et al., 2022)).

The solution of the LQR problem (5), is given by *rolling-out* the policies along the linear trajectories of (5). The oracle is then $\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u}) = (v_0; \dots; v_{\tau-1})$, 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 (5) by dynamic programming comes at a linear cost with respect to the length of the trajectory. Namely, in terms of elementary computations, the ILQR oracle has a computational cost

$$\mathcal{C}(n_x, n_u, \tau) = O(\tau(n_x + n_u)^3). \tag{9}$$

Note that, in nonlinear control problems, the state and control dimensions are generally small. On the other hand, the horizon $\tau$ may be large if, for example, for a fixed continuous time horizon, a small discretization stepsize was used to define (2). The ILQR algorithm keeps then a linear complexity with respect to the leading dimension $\tau$ of the problem. The linear quadratic problem (5) can also be solved by alternative linear algebra subroutines ranging from matrix-free solvers that take advantage of differentiable programming framework, or by introducing Lagrange multipliers. We refer to, e.g. Wright (1991a), for more details.

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 control variables $\boldsymbol{u}^{(0)}$, where $\nu_k$ are regularization parameters that may depend on the current iterate and $\mathrm{LQR}_\nu$ is implemented by Algo. 1.

### 2.2 Iterative Differential Dynamic Programming

The IDDP algorithm is an instance of a Differential Dynamic Programming (DDP) approach. A DDP approach considers computing approximate solutions of (2) 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 by, e.g., Tassa et al. (2012). In this case, a DDP approach amounts to computing the same policies $\pi_t$ as an ILQR algorithm but rolling-out the policies along the original dynamics rather than the linearized ones.

Namely, the oracle output by IDDP is given as

$$\begin{aligned} &\mathrm{DDP}_\nu(\mathcal{J})(\boldsymbol{u}) = (v_0; \dots; v_{\tau-1}) \\ &\quad \text{where } v_t = \pi_t(y_t),\ y_{t+1} = f_t(x_t + y_t, u_t + v_t) - f_t(x_t, u_t) \text{ for } t \in \{0, \dots, \tau - 1\}, \end{aligned} \tag{10}$$

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 initial control variables $\boldsymbol{u}^{(0)}$, we obtain the iterative Linear Quadratic Regulator (IDDP) algorithm, which computes iterates of the form

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

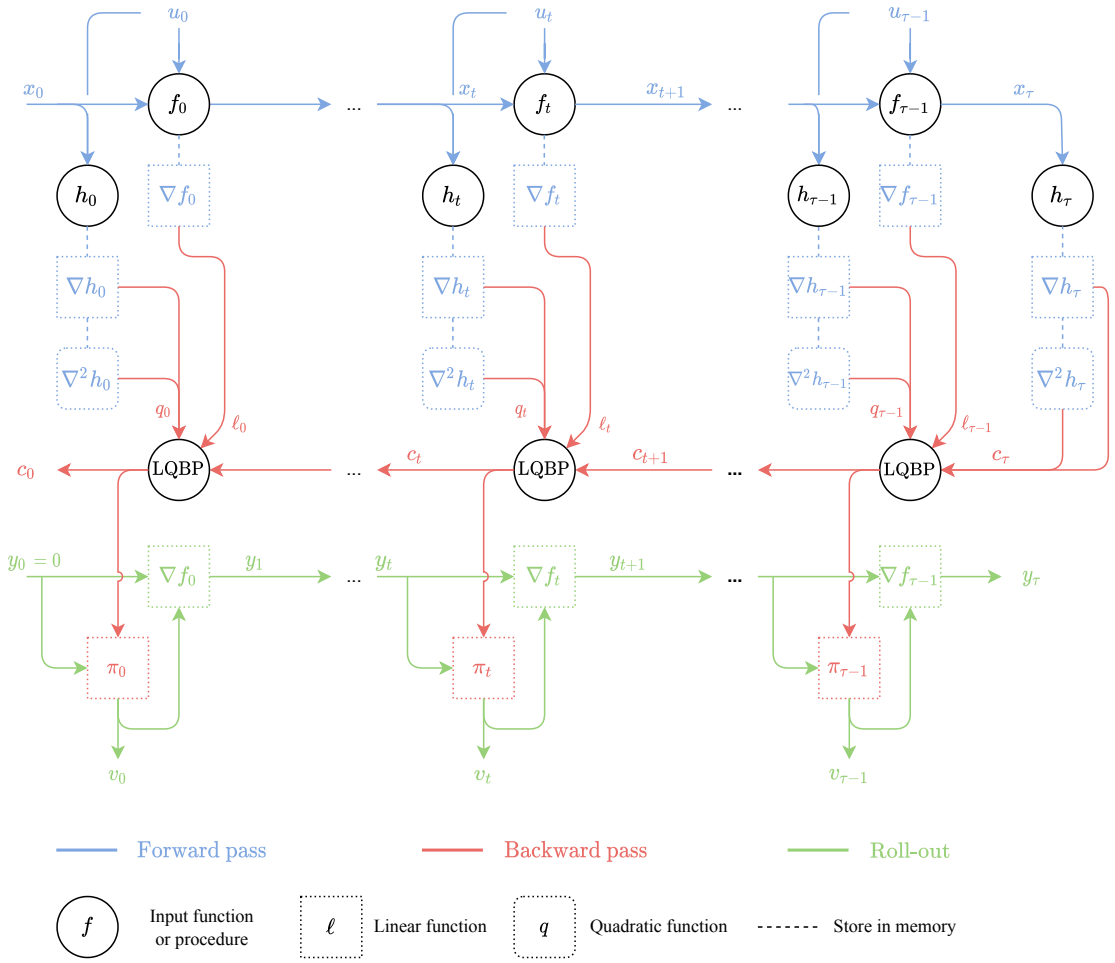


Figure 2: Computational scheme of the ILQR algorithm. The algorithm proceeds in three phases. In the *forward pass*, the first derivatives of the dynamics as well as the first and second derivatives of the costs are stored in memory (or the inputs are checkpointed to access these derivatives). During the *backward pass* the cost-to-go functions are back-propagated at each time step through matrix products and inversions, denoted simply LQBP for linear-quadratic backpropagation. The policies computed in the backward pass are used in a final *roll-out* phase through the linearized dynamics to output a candidate sequence of control inputs.

**Algorithm 1** ILQR and IDDP steps for problem (2)

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

Forward pass ▷ *instantiate problem (5) for the given control variables*

Initialize $x_0 = \bar{x}_0$

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

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

  Compute and store

$$A_t = \nabla_{x_t} f(x_t, u_t)^\top, \; B_t = \nabla_{u_t} f(x_t, u_t)^\top,$$
$$p_t = \nabla_{x_t} h_t(x_t, u_t), \; q_t = \nabla_{u_t} h_t(x_t, u_t),$$
$$P_t = \nabla^2_{x_t x_t} h_t(x_t, u_t), \; Q_t = \nabla^2_{u_t u_t} h_t(x_t, u_t), \; R_t = \nabla^2_{x_t u_t} h_t(x_t, u_t)$$

**end for**

Compute $h_\tau(x_\tau)$, $p_\tau = \nabla_{x_\tau} h_\tau(x_\tau)$ $P_\tau = \nabla^2_{x_\tau x_\tau} h_\tau(x_\tau)$

Backward pass ▷ *compute optimal policies for problem (5)*

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 (7) as

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

  Store the policies $\pi_t : y_t \to K_t y_t + k_t$ defined in (8) as

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

**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:** Control directions $\boldsymbol{v} = (v_0; \dots; v_{\tau-1})$

where the regularization parameters $\nu_k$ may depend on the current iterate and $\text{DDP}_\nu$ is implemented by Algo. 1.

### 2.3 Generic Convergence Guarantees

We start by presenting convergence guarantees of the ILQR and IDDP algorithm for generic problems of the form (1). First, with an appropriate choice of regularization both algorithms can converge globally to a stationary point at a polynomial rate (Theorem 2). Such a stationary point of $\mathcal{J}$ satisfies naturally necessary optimality conditions for problem (1) as recalled in Appendix B. Note that necessary optimality conditions in discrete time control problem differ from their continuous time counterpart as discussed in detail in Appendix B.

**Theorem 2** *For problem (1), assume that the dynamics $f_t$ are Lipschitz continuous with Lipschitz continuous gradients and that the costs $h_t$ are Lipschitz continuous with Lipschitz continuous gradients and Lipschitz continuous Hessians. Then, provided that the regularization $\nu$ is larger than some $c_1 > 0$, the iterates of the ILQR or the IDDP algorithms satisfy*

$$\min_{k\in\{0,\dots,K\}} \|\nabla \mathcal{J}(\boldsymbol{u}^{(k)})\|_2 \leq \sqrt{\frac{2(c_2+\nu)\left(\mathcal{J}(\boldsymbol{u}^{(0)}) - \min_{\boldsymbol{u}\in\mathbb{R}^{\tau n_u}} \mathcal{J}(\boldsymbol{u})\right)}{K+1}},$$

*for $c_1, c_2$ depending on the smoothness properties of the dynamics and the costs.*

**Proof** Detailed statements and proofs are presented in Lemma 31 and Lemma 33 for the ILQR and the IDDP algorithms, respectively. ■

We can also demonstrate local linear convergence of both algorithms towards a minimum under regular assumptions.

**Theorem 3** *For problem (1), assume that the dynamics $f_t$ and the costs $h_t$ are Lipschitz continuous with Lipschitz continuous gradients and Lipschitz continuous Hessians. Let $\boldsymbol{u}^{(k)}$ denote the $k^{th}$ iterate of the ILQR or the IDDP algorithms. Assume $\boldsymbol{u}^{(k)}$ to be close to a minimum $\boldsymbol{u}^*$ of $\mathcal{J}$ with positive definite Hessian. If the regularization $\nu$ is larger than some $c_1 > 0$, then the iterations of the ILQR or the IDDP algorithm converge linearly to $\boldsymbol{u}^*$ as*

$$\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^*\|_2 \leq \left(1 - \frac{c_2}{\nu}\right) \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2,$$

*for $c_1, c_2$ depending on the smoothness properties of the dynamics and the costs.*

**Proof** Detailed statements and proofs are presented in Lemma 32 and Lemma 34 for the ILQR and the IDDP algorithms, respectively. ■

**Remark 4** *Compared to a Newton method that can converge locally at a quadratic rate on problems of the form (1) (Nocedal and Wright, 2006; Pantoja, 1988; Dunn and Bertsekas, 1989), the ILQR and IDDP algorithms converge locally only at linear rate a priori (see also Baumgärtner et al. (2023)). Similarly, the original Differential Dynamic Programming (DDP) approach of Jacobson and Mayne (1970) can converge locally at a quadratic rate (Murray and Yakowitz, 1984; Liao and Shoemaker, 1991; Di and Lamperski, 2019). However, the local linear convergence rates presented in Theorem 3 do not match the superlinear rates observed in practice in Fig. 8 (see also Roulet et al. (2022)). Hence, we consider in the following additional properties of the problem that can uncover both the global convergence of the ILQR and IDDP algorithms as well as their fast local convergence.*

## 3. Conditioning Analysis

To understand the convergence behavior of the ILQR and IDDP algorithms displayed in Fig. 1, we consider a restricted class of control problems without control costs of the form (2). Namely, from now on, we consider objectives of the form

$$\mathcal{J}(\boldsymbol{u}) = \sum_{t=1}^{\tau} h_t(x_t) \qquad (11)$$
$$\text{s.t.} \quad x_{t+1} = f(x_t, u_t), \quad \text{for } t \in \{0, \dots, \tau - 1\}, \qquad x_0 = \bar{x}_0,$$

for $\boldsymbol{u} = (u_0; \dots; u_{\tau-1}) \in \mathbb{R}^{\tau n_u}$. Such objectives keep the main difficulty of nonlinear control problems: for nonlinear dynamics $f$, the overall objective $\mathcal{J}$ is non-convex such that convergence to global minima is a priori not guaranteed by even a simple gradient descent. Nevertheless, by decomposing the objective at the scale of the dynamics, and further decomposing the dynamics by an appropriate discretization scheme, we can identify sufficient conditions for global convergence to minima linked to usual notions in nonlinear control. We can then further show the convergence of the ILQR and IDDP algorithms to a global minimum, and detail the several phases of convergence (Sec. 4.1, 4.2, 4.3).

### 3.1 Objective Decomposition

The objective $\mathcal{J}$, defined in (11), 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 5** *We define the control* of $\tau$ *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.,*

$$f^{[\tau]}(x_0, \boldsymbol{u}) = (x_1; \dots; x_\tau) \qquad (12)$$
$$\textit{s.t.} \quad x_{t+1} = f(x_t, u_t) \qquad \textit{for } t \in \{0, \dots, \tau - 1\}.$$

By defining the cost $h(\boldsymbol{x})$ of a trajectory $\boldsymbol{x} = (x_1, \dots, x_\tau)$ as the sum of the cost of the states, problem (2) amounts to solving

$$\min_{\boldsymbol{u} \in \mathbb{R}^{\tau n_u}} \left\{ \mathcal{J}(\boldsymbol{u}) = h(f^{[\tau]}(\bar{x}_0, \boldsymbol{u})) \right\}, \text{for } f^{[\tau]}(x_0, \boldsymbol{u}) \text{ given in (12)}, \; h(\boldsymbol{x}) = \sum_{t=1}^{\tau} h_t(x_t). \qquad (13)$$

For convex costs $h$, if the dynamic $f$ is linear, then the function $f^{[\tau]}$ is also linear and the overall problem (13) 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 $\tau$ 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 $\boldsymbol{x} = f^{[\tau]}(\bar{x}_0, \boldsymbol{u})$,

$$\|\nabla \mathcal{J}(\boldsymbol{u})\|_2^2 = \|\nabla_{\boldsymbol{u}} f^{[\tau]}(\bar{x}_0, \boldsymbol{u}) \nabla h(\boldsymbol{x})\|_2^2 \geq \sigma^2 \|\nabla h(\boldsymbol{x})\|_2^2 \geq \sigma^2 \mu (h(\boldsymbol{x}) - h^*), \tag{14}$$

where $h^* = \min_{\boldsymbol{x} \in \mathbb{R}^{\tau n_u}} h(\boldsymbol{x})$. Since the set $\{\boldsymbol{u} \in \mathbb{R}^{\tau n_u} : \nabla \mathcal{J}(\boldsymbol{u}) = 0\}$ is not empty as we assumed that the problem has a minimizer, the above equation implies that $h^* = \mathcal{J}^*$ and so that the overall objective satisfies a gradient dominating property. We investigate then whether the control of $\tau$ 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 $\tau$ 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 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 $\tau$ 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. Note that Lemma 6 and the subsequent analysis of the algorithms presented in Sec. 2 can be extended to time-varying discrete time dynamics as presented in Lemma 37.

**Lemma 6** *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.*

$$\forall x, u \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}, \quad \sigma_{\min}(\nabla_u f(x, u)) \geq \sigma_f > 0, \tag{15}$$

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

$$\forall x_0, \boldsymbol{u} \in \mathbb{R}^{n_x} \times \mathbb{R}^{\tau n_u}, \quad \sigma_{\min}(\nabla_u f^{[\tau]}(x_0, \boldsymbol{u})) \geq \sigma_{f^{[\tau]}} := \frac{\sigma_f}{1 + l_f^x} > 0, \tag{16}$$

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

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

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 $f$ as shown in the following lemma.

**Lemma 7** *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*

$$l_{f^{[\tau]}} \leq l_f^u S, \qquad L_{f^{[\tau]}} \leq S(L_f^{xx} l_{f^{[\tau]}}^2 + 2L_f^{xu} l_{f^{[\tau]}} + L_f^{uu}) \qquad S = \sum_{t=0}^{\tau-1} (l_f^x)^t, \tag{17}$$

*where the constants $l_f^u = \sup_{x \in \mathbb{R}^{n_x}} l_{f(x,\cdot)}$, $L_f^{xx} = \sup_{u \in \mathbb{R}^{n_u}} l_{\nabla_x f(\cdot,u)}$, $L_f^{uu} = \sup_{x \in \mathbb{R}^{n_x}} l_{\nabla_u f(x,\cdot)}$, $L_f^{xu} = \sup_{x \in \mathbb{R}^{n_x}} l_{\nabla_u f(\cdot,u)}$ are maximal Lipschitz continuity constants of partial functions or gradients of the dynamics.*

**Proof** This is a direct corollary of the time-varying version presented in Lemma 36. ■

At first glance the Lipschitz continuity constant of the function $f^{[\tau]}$ and its gradients appear to depend exponentially on the horizon $\tau$ through the constant $S$ defined above. However, recall that problems of the form (2) stem from the discretization of a continuous problem on a finite time interval $[0, T]$. The Lipschitz continuity constant of the discretized dynamics depend then on the discretization step $\Delta$, which depends itself on the discrete time horizon $\tau$ as $\Delta = T/\tau$. Hence, the dependency of the smoothness constants of the problem may not depend exponentially on $\tau$.

For example, if the continuous time dynamics of the problem are given by a function f and an Euler discretization scheme is used, then the discretized dynamic take the form $f(x_t, u_t) = x_t + \Delta \mathrm{f}(x_t, u_t)$ with $\Delta = T/\tau$ and the Lipschitz continuity parameter of the discretized dynamics is then $l_f^x \leq 1 + \Delta l_{\mathrm{f}}^x$. Hence, the constant $S$ defined above can be upper bounded as $S \leq \sum_{t=0}^{\tau-1} (1 + l_{\mathrm{f}}^x T/\tau)^t \leq (\exp(T l_{\mathrm{f}}^x) - 1)\tau/(T l_{\mathrm{f}}^x)$ and since $l_f^u, L_f^{xx}, L_f^{xu}, L_f^{uu}$ are all proportional to $\Delta = T/\tau$, the smoothness constants derived in (17) are independent of $\tau$ in this case and only depends on the length $T$ of the continuous time problem.

### 3.2 Dynamic Decomposition

We have isolated condition (3) 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 (3) 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 (3) as illustrated in Example 1.

**Example 1** *Consider the continuous time evolution of a pendulum*

$$\begin{aligned} \dot{\theta}(t) &= \omega(t), \\ ml^2\dot{\omega}(t) &= p(\theta(t), \omega(t), u(t)) \coloneqq -mlg\sin\theta(t) - \mu\omega(t) + u(t), \end{aligned}$$

*where $\theta$ is the angle with the vertical axis, $\omega$ is the angular speed, $u$ is a torque applied to the pendulum which defines the control we have on the system, and $m$, $l$, $\mu$, $g$ are physical constants of the problem described in Section 5. The state is defined by $x = (\theta, \omega)$. Using a simple Euler scheme, the discretized dynamics cannot satisfy (15), since we would have only one variable $u_t$ to control two elements, $\theta_t, \omega_t$, at each time step $t$.*

*On the other hand, one can consider a two-step discretization scheme such that the controls are divided in two variables $u_t = (v_t, v_{t+1/2})$. The dynamics read then*

$$\begin{aligned} \theta_{t+1/2} &= \theta_t + \Delta\omega_t, & \theta_{t+1} &= \theta_{t+1/2} + \Delta\omega_{t+1/2}, \\ ml^2\omega_{t+1/2} &= \omega_t + \Delta p(\theta_t, \omega_t, v_t), & ml^2\omega_{t+1} &= \omega_{t+1/2} + \Delta p(\theta_{t+1/2}, \omega_{t+1/2}, v_{t+1/2}), \end{aligned}$$

*where $\Delta$ is some discretization step. Intuitively, the variable $v_{t+1/2}$ fully controls $\omega_{t+1}$, while the variable $v_t$ fully controls $\theta_{t+1}$. One can verify that the Jacobian of the discretized dynamics $x_{t+1} = f(x_t, u_t)$ for $u_t = (v_t, v_{t+1/2})$ are then surjective which then ensure the surjectivity of the overall control of the pendulum in $\tau$ steps and the efficiency of the ILQR and IDDP algorithms as observed in Fig. 1 and further justify in Sec. 4.*

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 $\phi$ as defined below. Concretely, $\phi$ 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 8** *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 $\phi^{\{k\}} : \mathbb{R}^{n_x} \times \mathbb{R}^{km_u} \to \mathbb{R}^{n_x}$, 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.,*

$$\begin{aligned} &\phi^{\{k\}}(y_0, \boldsymbol{v}) = y_k \\ &\quad s.t. \quad y_{s+1} = \phi(y_s, v_s) \quad \text{for } s \in \{0, \dots, k-1\}. \end{aligned} \tag{18}$$

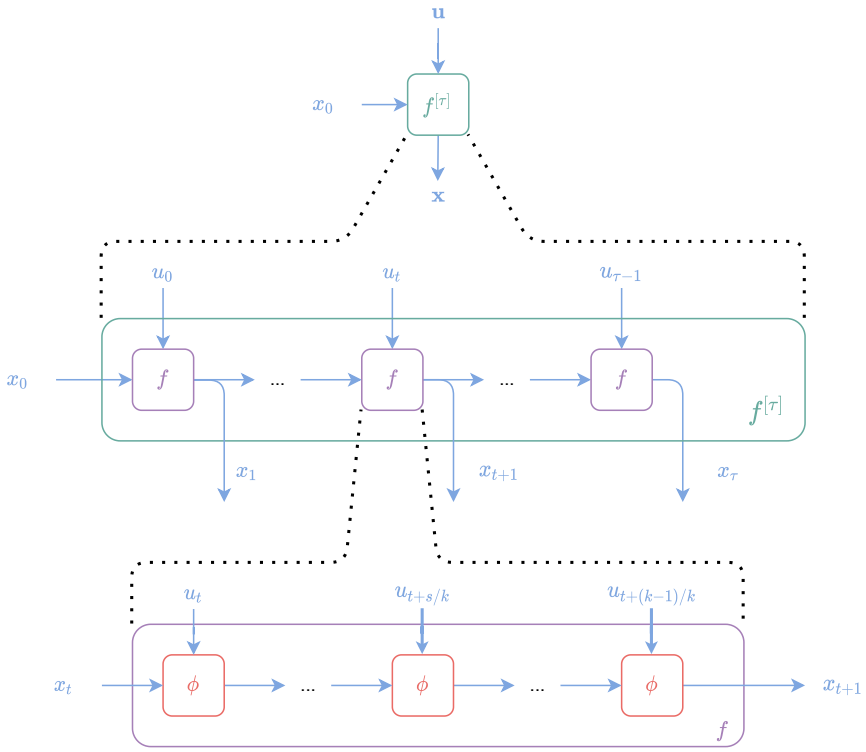


Figure 3: Zooming into the properties of the dynamics. The overall objective can be split at the scale of each step of the dynamics. Each step can be further decomposed at the scale of the discretization scheme to link properties of the underlying dynamic to global properties of the objective.

Our overall approach is illustrated in Fig. 3. Our goal is then to know whether, by considering enough steps of $\phi$, 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 (3) as illustrated in Example 1. 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)). A brief exposition of static feedback linearization schemes in continuous time and the associated Brunovsky's form are presented in Appendix E.

**Definition 9** *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 some 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 $z_t = a(y_t)$,*

*$w_t = b(v_t, y_t)$ is linear. Namely, 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*

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

*where $\zeta^{(i)}$ denotes the $i^{th}$ coordinate of $\zeta$.*

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

$$z_{t+1}^{(i)} = z_t^{(i+1)} \text{ for all } i \in \{1, \dots, n_x - 1\}, \quad z_{t+1}^{(n_x)} = w_t, \tag{19}$$

i.e., $z_{t+1} = Dz_t + w_t e$, where $D = \sum_{i=1}^{n_x - 1} e_i e_{i+1}^\top$ is the upper shift matrix in $\mathbb{R}^{n_x}$ with $e_i$ the $i^{\text{th}}$ 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 differentiable function $\psi$, that is,

$$y_{t+1}^{(i)} = y_t^{(i)} + \Delta y_t^{(i+1)}, \text{ for all } i \in \{1, \dots, n_x - 1\}, \quad y_{t+1}^{(n_x)} = y_t^{(n_x)} + \Delta \psi(y_t, v_t). \tag{20}$$

As shown in Lemma 38 in Appendix E, such system can easily be reparameterized in Brunovsky's form (19) 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 in a linear form using diffeomorphisms.

The canonical representation (19) clarifies why the surjectivity of the linearized dynamics may hold by taking enough steps as it is clear that, in the representation (19), 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 (19), considering $n_x$ steps ensures condition (3). The following theorem shows that this property is kept when considering the original system.

**Theorem 10** *If a discrete time system $y_{t+1} = \phi(y_t, v_t)$ is linearizable by static feedback as defined in Def. 9, then $\phi^{\{k\}}$, the control in $k \geq r = \max\{r_1, \dots, r_{m_u}\}$ steps of $\phi$, 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 $\phi$ 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 H. We detail first the single-input case described in (19), 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 (19), we have that $z_k^{(i)} = w_{i-1}$ for all $i \in \{1, \dots, n_x\}$. By defining, 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 $\phi$. Hence, denoting $\boldsymbol{y} = (y_1; \dots; y_k) = \phi^{[k]}(y_0, \boldsymbol{v})$, we have

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

Since $\nabla_{\boldsymbol{y}} B(\boldsymbol{y}, \boldsymbol{v})$ is strictly upper 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 $k > n_x$ and $s = k - n_x > 0$. 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

$$\nabla_{\boldsymbol{v}} \phi^{\{k\}}(y_0, \boldsymbol{v}) = \begin{pmatrix} \nabla_{v_{0:s-1}} \phi^{\{s\}}(y_0, v_{0:s-1}) \nabla_{y_s} \phi^{\{n_x\}}(y_s, v_{s:k-1}) \\ \nabla_{v_{s:k-1}} \phi^{\{n_x\}}(y_s, v_{s:k-1}) \end{pmatrix}. \tag{21}$$

The function $y_s, v_{s:k-1} \to \phi^{\{n_x\}}(y_s, v_{s:k-1})$ amounts to the control of $\phi$ 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 $y_0 \in \mathbb{R}^{n_x}, \boldsymbol{v} \in \mathbb{R}^{k m_u}$, provided that $k \geq n_x$.

We consider now deriving quantitative bounds. We focus on the single-input case and start with $k = n_x$. 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 6, we get

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

Given the feedback linearization scheme, the discrete time dynamic $\phi$ 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 $t \in \{0, \dots, k-1\}$,

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

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 $i^{\text{th}}$ 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 (see Appendix H for more details)

$$\begin{aligned} \nabla_{\boldsymbol{v}} \Phi(\boldsymbol{y}, \boldsymbol{v}) &= \mathbf{diag}((\partial_{v_t} b(y_t, v_t))_{t=0}^{k-1}) (\mathrm{I} \otimes e^\top) \, \mathbf{diag}((\nabla a(y_{t+1})^{-1})_{t=0}^{k-1}) \\ \nabla_{\boldsymbol{y}} \Phi(\boldsymbol{y}, \boldsymbol{v}) &= (D \otimes \mathrm{I}) \, \mathbf{diag}((\nabla a(y_t))_{t=0}^{k-1}) (\mathrm{I} \otimes D^\top) \, \mathbf{diag}((\nabla a(y_{t+1})^{-1})_{t=0}^{k-1}) \\ &\quad + (D \otimes \mathrm{I}) \, \mathbf{diag}((\nabla_{y_t} b(y_t, v_t))_{t=0}^{k-1}) (\mathrm{I} \otimes e^\top) \, \mathbf{diag}((\nabla a(y_{t+1})^{-1})_{t=0}^{k-1}), \end{aligned} \tag{22}$$

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

$$\begin{aligned}
\nabla_{\boldsymbol{v}} \phi^{\{k\}}(y_0,\boldsymbol{v})\nabla a(y_k) &= V(\mathrm{I}+EC^{-1}(\mathrm{I}-FAGC^{-1}-FYEC^{-1})^{-1}FY) \\
&\overset{(i)}{=} V(\mathrm{I}-EC^{-1}(\mathrm{I}-FAGC^{-1})^{-1}FY)^{-1}, \\
&\overset{(ii)}{=} V(\mathrm{I}-E(\mathrm{I}-FG)^{-1}FA^{-1}Y)^{-1}, \\
&\overset{(iii)}{=} V\left(\mathrm{I}-E\left(\sum_{i=1}^{k-1} D^i\otimes (D^\top)^{i-1}\right)A^{-1}Y\right)^{-1}. \qquad (23)
\end{aligned}$$

Above, in $(i)$ we used the Sherman-Morrison-Woodbury identity, in $(ii)$ we used that $FA = CF$ and $FA^{-1} = C^{-1}F$ (see Appendix H), in $(iii)$ we used that $FG = D\otimes D^\top$ is nilpotent of order $k = n_x$ since $D^k = 0$ (see Appendix H). The result follows for $k = n_x$ from the assumptions of Lipschitz continuity and non-singularity of the gradients of the diffeomorphisms, and from the fact that $\|E\|_2 \le 1$ and $\|D\|_2 \le 1$. For $k > n_x$, we have from (21), that for any $\lambda \in \mathbb{R}^{n_x}$, $\|\nabla_v \phi^{\{k\}}(y_0,\boldsymbol{v})\lambda\|_2 \ge \|\nabla_{v_{s:k-1}} \phi^{\{n_x\}}(y_s, v_{s:k-1})\lambda\|_2$, hence the result follows.

For multi-input systems, let $r = \max\{r_1,\dots,r_{m_u}\}$. One easily verifies that for any $k \ge r$, 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, see Appendix H for more details. ■

Overall, Theorem 10 shows that for, e.g., a dynamical system driven by its $k^{\text{th}}$ derivative as in (20), by considering a dynamic $f$ defined by $k$ steps of an Euler discretization of the system, condition (3) 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 (3) 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 (2) at the scale of the whole trajectory and analyze problem (2) as a compositional problem of the form

$$\min_{\boldsymbol{u}\in\mathbb{R}^{\tau n_u}} \{\mathcal{J}(\boldsymbol{u}) = h(g(\boldsymbol{u}))\}, \text{ where } g(\boldsymbol{u}) = f^{[\tau]}(\bar{x}_0,\boldsymbol{u}) \text{ and } h(\boldsymbol{x}) = \sum_{t=1}^{\tau} h_t(x_t). \qquad (24)$$

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.

**Assumption 11** *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 (15).*

*In consequence, the total cost $h$ defined in (24) is convex, has $L_h$-Lipschitz-continuous gradients and $M_h$-Lipschitz-continuous Hessians. The function $g$ defined in (24) is $l_g$-Lipschitz-continuous with $L_g$-Lipschitz-continuous gradients satisfying*

$$\forall \boldsymbol{u} \in \mathbb{R}^{\tau n_u}, \quad \sigma_{\min}(\nabla g(\boldsymbol{u})) \geq \sigma_g > 0, \tag{25}$$

*where $l_g = l_{f^{[\tau]}}$, $L_g = L_{f^{[\tau]}}$ are given in (17) and $\sigma_g = \sigma_{f^{[\tau]}}$ is given in (16).*

### 4.1 Convergence Proof Sketches

**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

$$\begin{aligned} \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u}) &= \arg\min_{\boldsymbol{v} \in \mathbb{R}^{\tau n_u}} q_h^{g(\boldsymbol{u})}(\ell_g^{\boldsymbol{u}}(\boldsymbol{v})) + \frac{\nu}{2}\|\boldsymbol{v}\|_2^2 \\ &= -(\nabla g(\boldsymbol{u}) \nabla^2 h(g(\boldsymbol{u})) \nabla g(\boldsymbol{u})^\top + \nu \, \mathrm{I})^{-1} \nabla g(\boldsymbol{u}) \nabla h(g(\boldsymbol{u})), \end{aligned} \tag{26}$$

where $\ell_g^{\boldsymbol{u}}$ and $q_h^{g(\boldsymbol{u})}$ are the linear and quadratic approximations of, respectively, the control in $\tau$ steps around $\boldsymbol{u}$ and the total costs around $g(\boldsymbol{u})$ as defined in the notations. Equation 26 clearly reveals that the ILQR algorithm amounts to a regularized generalized Gauss-Newton algorithm (Diehl and Messerer, 2019) implemented by a dynamic programming procedure exploiting the structure of the problem.

**Proof sketch of global convergence.** By choosing a large enough regularization, the updates of the ILQR algorithm approach the ones of a gradient descent as we have from the expression of $\mathrm{LQR}_\nu$ in (26) that for $\nu \gg 1$, $\boldsymbol{u} + \mathrm{LQR}_\nu(\mathcal{J}(\boldsymbol{u})) \approx \boldsymbol{u} - \nu^{-1} \nabla \mathcal{J}(\boldsymbol{u})$. This suggests that the ILQR algorithm can converge globally just as a gradient descent given a gradient dominating property such as (14) (Polyak, 1964; Bolte et al., 2017).

Formally, to ensure global convergence, we consider taking a regularization $\nu$ that may depend on the current command $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$, s.t. for $\boldsymbol{v} = \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$,

$$\mathcal{J}\left(\boldsymbol{u} + \boldsymbol{v}\right) \leq h \circ g(\boldsymbol{u}) + q_h^{g(\boldsymbol{u})} \circ \ell_g^{\boldsymbol{u}}(\boldsymbol{v}) + \frac{\nu}{2}\|\boldsymbol{v}\|_2^2 = \mathcal{J}(\boldsymbol{u}) + \frac{1}{2} \nabla \mathcal{J}(\boldsymbol{u})^\top \boldsymbol{v}. \tag{27}$$

Given the analytic form of $\boldsymbol{v} = \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$ in (26), 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 $\alpha$ that depends on the regularization $\nu$ 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 (27), we get a global linear convergence rate.

We further show that a regularization ensuring (27) can be chosen to scale as a function of $\|\nabla h(g(\boldsymbol{u}))\|_2$, which helps decompose the computational complexity in (i) the complexity of solving $\min_{\boldsymbol{x}\in\mathbb{R}^{\tau n_x}} h(\boldsymbol{x})$ given an assumption on its gradient dominance and the smoothness properties of the costs, (ii) a term that depends on the initial gap and condition numbers associated to the approximation of a gradient descent by a Gauss-Newton method though the smoothness properties of the cost, the dynamics and the surjectivity of the dynamics.

**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. Namely, if $g$ satisfies (25), 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

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

Consider then the trajectory associated to a single step of ILQR, 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.

**Proof sketch of total complexity.** To blend global convergence and local quadratic convergence, we observe that if the costs are strongly convex then they satisfy a gradient dominating property and are self-concordant. To satisfy condition (27), the regularization can then be chosen to be proportional to the norm of the gradient of the costs at the current iterate, i.e., $\nu_k = \bar{\nu}_k \|\nabla h(g(\boldsymbol{u}^{(k)}))\|_2$ for $\bar{\nu}_k$ bounded above by a constant which ensures that $\nu_k$ tends to 0 with the iterations $k$. By satisfying condition (27), we can ensure global convergence, while by having $\nu_k \to 0$, we can ensure local quadratic convergence.

**Proof sketch of convergence of the IDDP algorithm.** The IDDP algorithm cannot be simply analyzed as an instance of a classical optimization algorithm. However, a careful analysis of the difference in the updates of the ILQR and IDDP algorithms for strongly convex costs reveal that the difference in their oracles can be bounded as $\|\,\mathrm{DDP}_\nu(\mathcal{J})(\boldsymbol{u}) - \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2 \leq \eta \|\,\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2^2$ for some constant $\eta$ independent of $\boldsymbol{u}$ and $\nu$. This observation enables us to derive an appropriate rule for selecting the regularization for the IDDP algorithm and to ensure that the quadratic local convergence is maintained since the approximation error of LQR by DDP is quadratic.

**Remark 12** *If the function $g$ is surjective and satisfies Assumption (25), then local quadratic convergence of e.g. a Gauss-Newton method or a Levenberg-Marquardt method (for $h$*

*quadratic) is known, see Björck (2024, Chapter 9.2.2), Bergou et al. (2020). For $h$ non-quadratic, local quadratic convergence of generalized Gauss-Newton methods has also been shown in some special cases by Messerer et al. (2021, Section 2.2). Compared to these results, we consider deriving an entire global convergence rate decomposed into a first slow convergence phase and a fast local convergence phase. Moreover, we consider quantitative bounds involving the constants in (25) and additional assumptions (self-concordance or gradient dominant assumptions).*

## 4.2 Convergence Analysis of ILQR

### 4.2.1 Global Convergence Rate to Minima

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 (Łojasiewicz, 1963; Polyak, 1964; Bolte et al., 2017).

**Theorem 13** *Given Assumption 11, the sufficient decrease condition (27) is satisfied for a regularization*

$$\nu(\boldsymbol{u}) = \frac{L_g\|\nabla h(g(\boldsymbol{u}))\|_2}{2}\gamma\left(\frac{L_g\|\nabla h(g(\boldsymbol{u}))\|_2}{4l_g^2L_h(\beta+1)}\right),$$

*where $\gamma(x) = 1+\sqrt{1+1/x}$ and $\beta = M_h l_g^2/(3L_gL_h)$. In addition to Assumption 11, 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 $x \in \mathbb{R}^{n_x}, t \in \{1, \dots, \tau\}$. The total cost satisfies then, for $\mu_h = \mu/\tau^{(2r-1)/2r}$,*

$$\forall \boldsymbol{x} \in \mathbb{R}^{\tau n_x}, \quad \|\nabla h(\boldsymbol{x})\|_2 \geq \mu_h^r(h(\boldsymbol{x}) - h^*)^r. \tag{28}$$

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

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

*and, if $1/2 < r < 1$, the number of iterations to converge to an accuracy $\varepsilon$ 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 $\rho_h = L_h/\mu_h^{2r}$, $\rho_g = l_g/\sigma_g$, $\theta_h = M_h/(2\mu_h^{3r})$, $\theta_g = L_g/(\sigma_g^2\mu_h^r)$, $\alpha = 4\rho_g^2\rho_h(\beta+1)$, $\delta_0 = \mathcal{J}(\boldsymbol{u}^{(0)}) - \mathcal{J}^*$ and the case $r = 2/3$ is to be understood limit-wise.*

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

**Remark 14** *Consider the case $r = 1/2$ in Theorem 13. 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 $\theta_h$, i.e., the inverse of the squared root of the objective. Finally, the terms $\beta$ and $\alpha$ are additional dimension independent constants that act as additional condition numbers.*

**Remark 15** *The rate of convergence in Theorem 13 for $r = 1/2$ is composed of (i) a term $\rho_h \ln(\delta_0/\varepsilon)$ that is the linear complexity associated to the computation of $\min_{\boldsymbol{x}\in\mathbb{R}^{\tau n_u}} h(\boldsymbol{x})$ by a gradient descent on a function $h$ that has Lipschitz-continuous gradients with a gradient dominance property and (ii) a term $4\theta_g\sqrt{\delta_0}\gamma\left(\theta_g\sqrt{\delta_0}/\alpha\right)$ that depends on the initial gap and appropriate condition numbers on the problem. To understand the effect of this second term, consider computing the value of the gap $\delta_j$ after $j$ iterations such that the complexity of reducing the gap further by a factor $1/e \approx 1/2$ is dominated by the logarithmic term such that we enter a linear phase of convergence. Formally, after $j$ iterations of the algorithm, the remaining number of iterations to reduce the gap further by a factor $1/e$, i.e., reach an accuracy $\varepsilon = \delta_j/e$, is $4\theta_g\sqrt{\delta_j}\gamma\left(\theta_g\sqrt{\delta_j}/\alpha\right) + 2\rho_h$. To neglect the first term in favor of the second term we need $\tilde{\gamma}(\theta_g\sqrt{\delta_j}/\alpha) \leq \rho_h/(2\alpha) \leq 1$ for $\tilde{\gamma}(x) = x + \sqrt{x^2+x}$, which is satisfied for $\delta_j \leq c^2/{\theta_g}^2$ with $c = \rho_h/(16\rho_g^2(1+\beta))$. So up to a multiplicative factor $c$, the parameter $1/{\theta_g}^2$ plays the role of a gap determining a linear convergence phase.*

**Remark 16** *For $L_g = 0$, the terms depending on $\delta_0$ uniquely vanish since $\theta_g = 0$ in this case. We then get the classical rates when minimizing a function $h$ that satisfy (28) 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 $h$ as shown in Sec. 4.2.2.*

**Proof** [Proof of Theorem 13] The validity of the gradient dominating property for the total costs is presented in Lemma 39 in Appendix F. Note that if $h$ satisfies (28) and $g$ satisfies (25), 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 $\boldsymbol{u}^* \in \arg\min_{\boldsymbol{u}\in\mathbb{R}^{\tau n_u}} \mathcal{J}(\boldsymbol{u})$ with $\mathcal{J} = h\circ g$, we get $0 = \|\nabla\mathcal{J}(\boldsymbol{u}^*)\|_2 \geq \sigma_g\mu_h^r(h(g(\boldsymbol{u}^*)) - h^*)^r \geq 0$, such that we have $\mathcal{J}^* = h^*$.

We have from Lemma 40 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 (27) 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 (27) is satisfied for any

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

with $a_0 = l_g L_g L_h(\beta+1)$ for $\beta = M_h l_g^2/(3L_g L_h)$. 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 (27) is satisfied,

$$\begin{aligned}
\mathcal{J}(\boldsymbol{u}+\boldsymbol{v}) - \mathcal{J}(\boldsymbol{u}) &\leq -\frac{1}{2}\nabla h(g(\boldsymbol{u}))^\top G^\top (GHG^\top + \nu(\boldsymbol{u})\,\mathrm{I})^{-1} G\nabla h(g(\boldsymbol{u})) \\
&= -\frac{1}{2}\nabla h(g(\boldsymbol{u}))^\top (H + \nu(\boldsymbol{u})(G^\top G)^{-1})^{-1}\nabla h(g(\boldsymbol{u})) \\
&\leq -\frac{1}{2}\frac{\sigma_g^2}{\sigma_g^2 L_h + \nu(\boldsymbol{u})}\|\nabla h(g(\boldsymbol{u}))\|_2^2 \leq -\frac{b_1 x^2}{\sqrt{b_2 x^2 + b_3 x} + b_4 x + b_5}, \qquad (29)
\end{aligned}$$

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 $f_1(\|\nabla h(g(\boldsymbol{u}))\|_2) \geq f_1((\mu_h \delta)^r)$ by assumption (28). 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 $\rho_h = L_h/\mu_h^{2r}$, $\rho_g = l_g/\sigma_g$, $\theta_h = M_h/(2\mu_h^{3r})$, $\theta_g = L_g/(\sigma_g^2 \mu_h^r)$, $\alpha = 4\rho_g^2 \rho_h(\beta + 1)$, $\beta = M_h l_g^2/(3L_g L_h)$.

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 $\varepsilon$, 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 $\varepsilon$ 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 $\varepsilon$ 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 $a > 0$,

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

■

#### 4.2.2 Local Convergence Rate to Minima

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 $h_t$ to be 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 (Nesterov, 2018, Definition 5.1.1). 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.

**Assumption 17** *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 $\tau$ steps of the dynamic $f$ defined in (12), are finite*

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

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

*In consequence, $h$ is $\vartheta_h$-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, Assumption 17 is satisfied if $h_t$ is strongly convex for all $t$ such that the total costs $h$ are strongly convex and if Assumption 11 is also satisfied. In that case, we have

$$l \leq \sqrt{L_h} l_g, \quad L \leq \sqrt{L_h} L_g, \quad 2\vartheta_h \leq M_h / \mu_h^{3/2}, \quad \sigma \geq \sqrt{\mu_h} \sigma_g. \tag{30}$$

Given Assumption 17 and equipped with a stepsize proportional to the Newton decrement, we can show a local quadratic convergence rate of the ILQR algorithm.

**Theorem 18** *Given Assumption 17, consider the ILQR algorithm for problem (2) 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*

$$\lambda_h(g(\boldsymbol{u}^{(k)})) < \lambda = \frac{1}{\max\{4\vartheta_h + 3\vartheta_g + 2\bar{\nu}/\sigma^2, 2\varrho\vartheta_h\}}, \tag{31}$$

*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 (2).*

**Remark 19** *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 quadratic 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 $\mu_h$ strongly convex with $\theta_g, \rho_h$ defined as in Theorem 13 and $\sigma, L$ expressed using (30). So up to $3\sqrt{\rho_h}$, the parameter $1/\theta_g$ acts again as a radius of fast convergence as in Theorem 13.*

**Remark 20** *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** [Proof of Theorem 18] 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 $c_1 = \max\{1-\bar{\nu}/(\sqrt{2\vartheta_h}Ll), 0\}$, $c_2 = \max\{1-\bar{\nu}/(2l^2\vartheta_h), 0\}$, $\varrho = l/\sigma$, $\vartheta_g = L/\sigma^2$. We have

$$\lambda_h(g(\boldsymbol{u}+\boldsymbol{v})) \leq \underbrace{\|\nabla h(g(\boldsymbol{u}+\boldsymbol{v})) - \nabla h(g(\boldsymbol{u}) + G^\top \boldsymbol{v})\|^*_{g(\boldsymbol{u}+\boldsymbol{v})}}_{A} + \underbrace{\|\nabla h(g(\boldsymbol{u}) + G^\top \boldsymbol{v})\|^*_{g(\boldsymbol{u}+\boldsymbol{v})}}_{B}. \tag{32}$$

**Bounding $A$ in (32).** By definition of $l$ in Assumption 17 and Lemma 41, we have

$$\|g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u})\|_{g(\boldsymbol{u})} \leq l\|\boldsymbol{v}\|_2, \qquad \|\boldsymbol{v}\|_2 \leq \frac{l\lambda_h(g(\boldsymbol{u}))}{l\sigma + \bar{\nu}\lambda_h(g(\boldsymbol{u}))}. \tag{33}$$

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 $\vartheta_h$-self-concordant, Theorem 5.1.7 of Nesterov (2018) applies and by using the definition of $L$ in Assumption 17, 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 (33), 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 $\vartheta_h$-self-concordant, we can then use Lemma 42 to obtain

$$\begin{aligned}A &\leq \frac{1}{1-\vartheta_h\|g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u}) - G^\top \boldsymbol{v}\|_{g(\boldsymbol{u}+\boldsymbol{v})}}\|g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u}) - G^\top \boldsymbol{v}\|_{g(\boldsymbol{u}+\boldsymbol{v})} \\ &\leq \frac{2Ll^2\lambda_h(g(\boldsymbol{u}))^2}{(l\sigma + \bar{\nu}\lambda_h(g(\boldsymbol{u})))^2}.\end{aligned} \tag{34}$$

**Bounding B in (32).** 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 $\vartheta_h$-self-concordant, we have then (Nesterov, 2018, Theorem 5.1.7),

$$B \leq \frac{1}{1-\vartheta_h\|g(\boldsymbol{u}+\boldsymbol{v}) - g(\boldsymbol{u})\|_{g(\boldsymbol{u})}}\|\nabla h(g(\boldsymbol{u}) + G^\top v)\|^*_{g(\boldsymbol{u})} \leq 2\|\nabla h(g(\boldsymbol{u}) + G^\top v)\|^*_{g(\boldsymbol{u})}. \tag{35}$$

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

$$\begin{aligned}\|\nabla h(g(\boldsymbol{u}) + G^\top \boldsymbol{v})\|^*_{g(\boldsymbol{u})} &= \|\nabla h(\boldsymbol{x}+\boldsymbol{n}) - \nabla h(\boldsymbol{x}) - (H + \nu(G^\top G)^{-1})\boldsymbol{n}\|^*_{\boldsymbol{x}} \\ &\leq \|\nabla h(\boldsymbol{x}+\boldsymbol{n}) - \nabla h(\boldsymbol{x}) - H\boldsymbol{n}\|^*_{\boldsymbol{x}} + \nu\|(G^\top G)^{-1}\boldsymbol{n}\|^*_{\boldsymbol{x}}. \end{aligned} \tag{36}$$

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

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

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

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

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

$$\|\nabla h(\boldsymbol{x}+\boldsymbol{n}) - \nabla h(\boldsymbol{x}) - H\boldsymbol{n}\|^*_{\boldsymbol{x}} \leq \frac{\|\boldsymbol{n}\|^2_{\boldsymbol{x}}\vartheta_h}{1-\|\boldsymbol{n}\|_{\boldsymbol{x}}\vartheta_h} \leq \frac{2\lambda_h(g(\boldsymbol{u}))^2\vartheta_h}{(1+\bar{\nu}l^{-2}\lambda_h(g(\boldsymbol{u})))^2}. \tag{37}$$

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

$$\|(G^\top G)^{-1}\boldsymbol{n}\|^*_{g(\boldsymbol{u})} = \|(H^{1/2}G^\top G H^{1/2} + \nu\,\mathrm{I})^{-1}H^{-1/2}\nabla h(g(\boldsymbol{u}))\|_2 \leq \frac{\lambda_h(g(\boldsymbol{u}))}{\sigma^2 + \bar{\nu}\lambda_h(g(\boldsymbol{u}))}. \tag{38}$$

So combining (38) and (37) into (36) and then (35) we get

$$B \leq 2\left(\frac{2\vartheta_h}{(1+\bar{\nu}l^{-2}\lambda_h(g(\boldsymbol{u})))^2} + \frac{\bar{\nu}}{\sigma^2+\bar{\nu}\lambda_h(g(\boldsymbol{u}))}\right)\lambda_h(g(\boldsymbol{u}))^2. \tag{39}$$

**Local quadratic convergence rate.** By combining (34) and (39) into (32), we get, as long as $\lambda_h(g(\boldsymbol{u})) \leq 1/\max\{\sqrt{2\vartheta_h\vartheta_g}c_1, 2\vartheta_h\varrho c_2, 2\vartheta_h c_2\}$,

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

Note that $c_1, c_2 \leq 1$ and that $2\vartheta_g + 4\vartheta_h + 2\bar{\nu}/\sigma^2 \geq \max\{2\vartheta_h, \sqrt{2\vartheta_h\vartheta_g}\}$, 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 $\lambda_h(g(\boldsymbol{u}+\boldsymbol{v})) \leq \bar{\lambda}^{-1}\lambda_h(g(\boldsymbol{u}))^2 < \lambda_h(g(\boldsymbol{u}))$, that is, we reach the region of quadratic convergence for $g(\boldsymbol{u})$. ■

#### 4.2.3 Global Complexity

Given Assumption 11, if the total cost is strongly convex then it satisfies the condition of Theorem 13 and Assumption 17 is satisfied with the estimates given in (30). 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.

**Theorem 21** *Consider the costs $h_t$ to be $\mu_h$-strongly convex and Assumption 11 to be satisfied. Then condition (27) is satisfied for a regularization*

$$\nu(\boldsymbol{u}) = \left(1 + \frac{\alpha}{2(1+\theta_g\|\nabla h(g(\boldsymbol{u}))\|_2/(\sqrt{\mu_h}\rho_g))}\right) L_g\|\nabla h(g(\boldsymbol{u}))\|_2$$

*With such regularization, the number of iterations of the ILQR algorithm to reach an accuracy $\varepsilon$ is at most*

$$k(\delta_0, \varepsilon) = 4\theta_g(\sqrt{\delta_0} - \sqrt{\varepsilon}) + 2\rho_h \ln\left(\frac{\delta_0}{\varepsilon}\right) + 2\alpha \ln\left(\frac{\theta_g\sqrt{\delta_0} + \rho_g}{\theta_g\sqrt{\varepsilon} + \rho_g}\right), \tag{40}$$

*where $\rho_h = L_h/\mu_h$, $\rho_g = l_g/\sigma_g$, $\theta_g = L_g/(\sigma_g^2\sqrt{\mu_h})$, $\theta_h = M_h/(2\mu_h^{3/2})$, $\alpha = 4\rho_g^2\rho_h(\beta+1)$, $\beta = M_h l_g^2/(3L_gL_h)$, and $\delta_0 = \mathcal{J}(\boldsymbol{u}^{(0)}) - \mathcal{J}^*$.*

*If in addition the target accuracy $\varepsilon$ is smaller than $\delta = 1/(32\rho_h(\theta_h(1+\sqrt{\rho_h}\rho_g^3/3) + \sqrt{\rho_h}\theta_g(1+\rho_g\rho_h))^2)$ which determines a quadratic convergence phase, the number of iterations of an ILQR algorithm to reach the accuracy $\varepsilon$ is at most $k(\delta_0, \delta) + O(\ln\ln(\varepsilon^{-1}))$.*

*The total computational complexity of the algorithm in terms of basic operations is then of the order of $(k(\delta_0, \delta) + O(\ln\ln(\varepsilon^{-1}))) \times \mathcal{C}(n_x, n_u, \tau)$ with $\mathcal{C}(n_x, n_u, \tau)$ defined in (9).*

**Remark 22** *The rate of convergence can now be separated between three phases, (i) the number of iterations to reach some linear convergence determined by the first term in the complexity bound (40), (ii) the number of iterations to reach the quadratic convergence rate that is captured by the logarithmic terms in the complexity bound (40), (iii) the quadratic convergence phase once $\delta_k$ is smaller than the gap of local quadratic convergence $\delta$.*

**Proof** [Proof of Theorem 21] By using the strong convexity of the costs $h$, we can refine the choice of the regularization to ensure (27). The validity of the proposed regularization to ensure condition (27) is shown in Lemma 43 in Appendix F. With the proposed regularization, Lemma 44 in Appendix F shows, following the same reasoning as in the proof of Theorem 13, that the number of iterations of the ILQR algorithm needed to reach an accuracy $\varepsilon$ is bounded by

$$k \leq 2\rho_h \ln\left(\frac{\delta_0}{\varepsilon}\right) + 4\theta_g\left(\sqrt{\delta_0} - \sqrt{\varepsilon}\right) + 2\alpha \ln\left(\frac{\theta_g\sqrt{\delta_0} + \rho_g}{\theta_g\sqrt{\varepsilon} + \rho_g}\right), \tag{41}$$

with $\rho_h$, $\rho_g$, $\theta_h$, $\theta_g$, $\alpha$ defined as in Theorem 13.

For the local convergence, the constants in Theorem 18 can be expressed in terms of the constants in Theorem 13 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 18, 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 $\bar{\nu}_k = \nu(\boldsymbol{u}^{(k)})/\lambda_h(g(\boldsymbol{u}^{(k)})) \leq \sqrt{L_h}(L_g + 2l_g(M_h l_g^2/3 + L_g L_h)/(\sigma_g \mu_h))$. 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 $O(\ln\ln\varepsilon^{-1})$. ■

Theorem 21 presents an ideal implementation of the ILQR algorithm given the knowledge of all constants to define the regularizations. This ideal implementation informs us on an appropriate line search strategy for the regularization, namely searching over $\bar{\nu}$ for regularizations of the form $\nu_k = \bar{\nu}\|\nabla h(g(\boldsymbol{u}^{(k)}))\|_2$. We present in Algo. 2 an implementation of the ILQR algorithm with an adequate line-search procedure that is guaranteed to terminate and maintain the complexity bounds presented in Theorem 21 as formally stated in Corollary 23, whose proof is given in Appendix F.

**Corollary 23** *Consider the assumptions and notations of Theorem 21 on problem (2) and Algo. 2 with an initial scaled regularization guess* $\bar{\nu}_{-1} \leq \left(1 + \alpha/(2 + 2\theta_g\sqrt{\delta_0}/\rho_g)\right) L_g$. *The total number of calls to ILQR oracles of Algo. 2 to reach an accuracy* $\varepsilon$ *is at most* $2k(\delta_0, \delta') + \ln\ln(\varepsilon^{-1}) + \lceil \log_2((1+\alpha/2)L_g/\bar{\nu}_{-1})\rceil$, *where* $k(\delta_0, \delta')$ *is defined as in Theorem 21 and* $\delta' = 1/(32\rho_h(\theta_h(1 + 2\sqrt{\rho_h}\rho_g^3/3) + \sqrt{\rho_h}\theta_g(1 + 2\rho_g\rho_h))^2)$ *is a gap of quadratic convergence for Algo. 2.*

**Algorithm 2** ILQR with Line-Search

**Inputs:** Initial point $\boldsymbol{u}_0$, initial scaled regularization $\bar{\nu}_{-1} > 0$, costs and dynamics summarized as $h$ and $g$ as in (24), $\mathrm{LQR}_\nu(\mathcal{J})$ oracle for $\mathcal{J} = h \circ g$.
**for** $k = 0, \ldots$ **do**
  Set $\bar{\nu}_k = \bar{\nu}_{k-1}$, $\nu_k = \bar{\nu}_k\|\nabla h(g(\boldsymbol{u}^{(k)}))\|_2$
  Compute $\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \mathrm{LQR}_{\nu_k}(\mathcal{J})(\boldsymbol{u}^{(k)})$
  **while** $\mathcal{J}\left(\boldsymbol{u}^{(k+1)}\right) > \mathcal{J}(\boldsymbol{u}^{(k)}) + \nabla\mathcal{J}(\boldsymbol{u})^\top(\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)})/2$ **do**
    Set $\bar{\nu}_k \leftarrow 2\bar{\nu}_k$, $\nu_k \leftarrow \bar{\nu}_k\|\nabla h(g(\boldsymbol{u}^{(k)}))\|_2$
    Set $\boldsymbol{u}^{(k+1)} \leftarrow \boldsymbol{u}^{(k)} + \mathrm{LQR}_{\nu_k}(\mathcal{J})(\boldsymbol{u}^{(k)})$
  **end while**
**end for**

### 4.3 Convergence Analysis of IDDP

The IDDP algorithm departs from the implementation of usual optimization algorithms for compositional problems as it cannot be formulated as the minimization of an approximation of the objective but rather as an approximate minimization of the objective by

dynamic programming, see e.g. Roulet et al. (2022) for a detailed overview. Its analysis can nevertheless be pursued by analogy of its implementation with the ILQR algorithm. Namely, the technical Lemmas 47 and 50 in Appendix G decompose the implementation of the IDDP algorithm into the dynamical structure of the problem to quantify an approximation bound between the oracles returned by the ILQR and IDDP algorithm of the form $\|\operatorname{DDP}_\nu(\mathcal{J})(\boldsymbol{u}) - \operatorname{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2 \leq \eta \|\operatorname{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2^2$ for some constant $\eta$ independent of $\boldsymbol{u}$ and $\nu$, provided that the costs are strongly convex.

Equipped with this approximation bound, we consider selecting the regularization of the IDDP algorithm such that

$$\mathcal{J}(\boldsymbol{u} + \operatorname{DDP}_\nu(\mathcal{J})(\boldsymbol{u})) \leq \mathcal{J}(\boldsymbol{u}) + \frac{1}{2}\nabla\mathcal{J}(\boldsymbol{u})^\top \operatorname{LQR}_\nu(\mathcal{J})(\boldsymbol{u}), \tag{42}$$

i.e., we use the same criterion as for the ILQR algorithm (27) to ensure a sufficient decrease. This choice of regularization is motivated by the implementation of the ILQR and IDDP algorithms which both compute $\frac{1}{2}\nabla\mathcal{J}(\boldsymbol{u})^\top \operatorname{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$ by dynamic programming; see Roulet et al. (2022) for more details. For strongly convex costs, the rule provided in (42) to select the stepsize together with the quadratic approximation bound between the oracles of the ILQR and IDDP algorithm enable us to state a global convergence result for the IDDP algorithm.

**Theorem 24** *Consider the costs to be $\mu_h$-strongly convex and Assumption 11 to be satisfied. Then the constant $\eta = \sup_{\boldsymbol{u}\in\mathbb{R}^{\tau n_u}, \nu>0} \|\operatorname{DDP}_\nu(\mathcal{J})(\boldsymbol{u}) - \operatorname{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2 / \|\operatorname{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2^2$ is finite. Condition (42) is satisfied for a regularization*

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

*where $\xi = (1+\rho_h\rho_g)(1+2\chi) + \rho_g^3(2\theta_h)/(3\theta_g)$, $\chi = l_g\eta/L_g$ and $\rho_h$, $\rho_g$, $\theta_h$, $\theta_g$ are defined in Theorem 21.*

*With such regularization, the number of iterations of the IDDP algorithm to reach an accuracy $\varepsilon$ is at most*

$$k \leq 2\rho_h \ln\left(\frac{\delta_0}{\delta}\right) + 4\theta_g\xi(\sqrt{\delta_0} - \sqrt{\delta}) + 2\rho_h{\theta_g}^2\chi^2(\delta_0 - \delta) + O(\ln\ln(\varepsilon^{-1})),$$

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

**Remark 25** *The complexity bounds for the IDDP algorithm in Theorem 24 take then the same form as the complexity bounds obtained for the ILQR algorithm in Theorem 21 up to some additional multiplicative factors. Our proof is built on considering IDDP to approximate ILQR. In practice, IDDP appears more efficient than ILQR as illustrated in Fig. 1 and other works (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 theoretical global convergence guarantees.*

**Proof** [Proof of Theorem 24] We sketch the proof of the first part of the claim, whose technical details can be found in Lemma 47. 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 $\mathrm{i}^{\mathrm{th}}$ 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 47. 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 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 (42) 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 $\eta$, 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 40. 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 21), we get that condition (42) is satisfied for

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

for $\xi = (1 + \rho_h\rho_g)(1 + 2\chi) + \rho_g^3(2\theta_h)/(3\theta_g)$, $\chi = l_g\eta/L_g$, where $\rho_h$, $\rho_g$, $\theta_h$, $\theta_g$ are defined in Theorem 21.

With such regularization choice, the convergence of the IDDP method follows from the proof of Theorem 13 by using that condition (42) 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 $\varepsilon$ is at most (see Appendix H for the detailed derivation)

$$k \leq 2\rho_h \ln(\delta_0/\varepsilon) + 4\theta_g \xi(\sqrt{\delta_0} - \sqrt{\varepsilon}) + 2\rho_h {\theta_g}^2 \chi^2 (\delta_0 - \varepsilon). \tag{43}$$

For the local convergence, define $l, \sigma, L, \vartheta_h, \vartheta_g$ as in the proof of Theorem 21. 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 18. Hence, using Lemma 42, 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 Theorem 5.1.7 of Nesterov (2018), 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 18 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

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

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

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

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 21. ■

## 5. Numerical Evaluations

We illustrate numerically the theoretical findings to examine their relevance. In all experiments, we implemented gradient descent (GD), ILQR, IDDP, with a line-search on either the stepsize for GD or the scaled regularization for ILQR and IDDP as in Algorithm 2.

### 5.1 Settings considered

We consider two simple synthetic control environments: swinging up a pendulum, and controlling a simplified model of a car. Experiments on a more realistic model of a car are presented in Appendix I. The code is publicly available at `https://github.com/vroulet/ilqc`. In all experiments we consider only a cost on the state variables, i.e., $h_t(x_t, u_t) = h_t(x_t)$. See Roulet et al. (2022) for additional experiments with costs on the control variables and other settings.

**Swinging up pendulum.** We consider swinging up a pendulum vertically through the control of a torque. The state $x = (\theta, \omega)$ consists in the angle $\theta$ with the vertical axis and the angular speed $\omega$ as illustrated in Fig 1. The dynamics in continuous time are

$$\dot{\theta}(t) = \omega(t), \qquad ml^2\dot{\omega}(t) = -mlg\sin\theta(t) - \mu\omega(t) + u(t), \tag{44}$$

where $m = 1$ is the mass of the blob, $l = 1$ is the length of the blob, $\mu = 0.01$ is a friction coefficient, $g = 10$ is the gravitational constant. The system is controlled through a torque, $u(t)$, applied to the pendulum. We use an Euler discretization scheme (Gautschi, 2011, Chapter 4) for the continuous dynamics (44) with a discretization step $\Delta = T/\tau$ for a total time $T = 2$ and a number of discretization steps $\tau = 100$.

For Fig. 1, we consider a single cost on the last state. Namely, the objective is to swing up the pendulum to be vertical with

$$h_\tau(x_\tau) = (\theta_\tau - \pi)^2 + \omega_\tau^2, \qquad h_t(x_t) = 0 \text{ for } t \in \{1, \dots, \tau - 1\}$$

for $x_\tau = (\theta_\tau, \omega_\tau)$. In other words, we target $\theta(T) = \pi, \omega(T) = 0$ for some time horizon $T$, given $\theta(0) = 0$, $\omega(0) = 0$. In some experiments below, we consider variations of the costs, such as considering a cost for each time step or a subsampled cost.

**Simple Model of a Car with Tracking Costs.** We consider a simple model of the car, illustrated in Fig. 1. The state consists in $x = (z_x, z_y, \theta, v)$, where $z = (z_x, z_y)$ is the position of the car, $\theta$ is the angle between the orientation of the car and the horizontal axis, a.k.a., the yaw, and $v$ is the longitudinal speed. The controls $u = (a, \delta)$ consist of the longitudinal acceleration $a$ of the car, and the steering angle $\delta$. For a car of length $l = 1$, the continuous time dynamics of this simplified model of the car are

$$\begin{aligned} \dot{z}_x(t) &= v(t)\cos\theta(t) \qquad & \dot{\theta}(t) &= v(t)\tan\delta(t)/l \\ \dot{z}_y(t) &= v(t)\sin\theta(t) \qquad & \dot{v}(t) &= a(t). \end{aligned}$$

We use a Runge-Kutta method of order 4 (Gautschi, 2011, Chapter 4), a discretization step $\Delta = T/\tau$ for a total time $T = 2$, and a number of discretization steps $\tau = 25$.

The objective consists in minimizing the distance between the position of the car and a reference position on a track. We define a reference track $z^*(t)$ as a continuous spline using a simple track presented in (Roulet et al., 2022, Figure 13). The discrete time reference positions are defined as $z_t^* = z^*(\Delta t)$. The costs consist then

$$h_t(x_t) = \|z_t - z_t^*\|_2^2 \quad \text{for } t \in \{1, \dots, \tau\},$$

For Fig. 1, we consider a subsampled cost equivalent to consider a multistep discretization strategy detailed in Section 3.2. Namely, we subsample the cost every $k = 3$ steps such that the costs are then

$$h_t(x_t) = \begin{cases} \|z_t - z_t^*\|_2^2 & \text{if } t \bmod k = 0 \\ 0 & \text{otherwise} \end{cases} \tag{45}$$

with $\Delta = T/(k\tau)$. Below, we consider also costs on every time-step, i.e., $k = 1$.

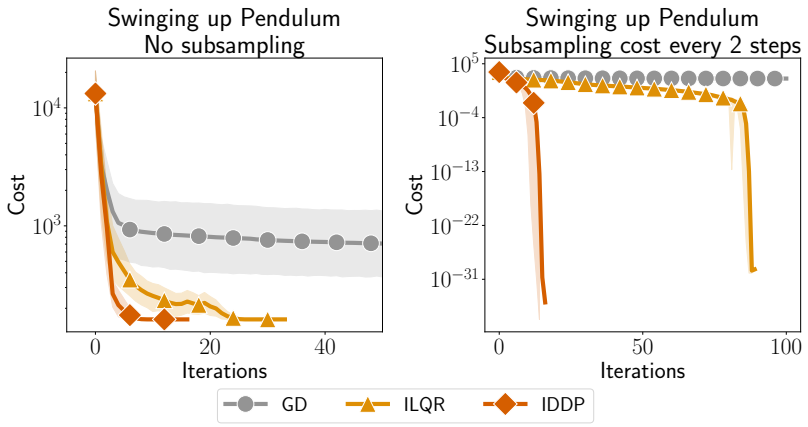


Figure 4: Cost along iterations of ILQR, IDDP and Gradient Descent (GD) on the pendulum problem using a cost at each time step or every two time steps. Shaded areas correspond to a 95% confidence intervals over 10 random initializations of the control sequences.

### 5.2 Evaluations

**Costs along iterations for the pendulum.** For a single final cost, the problem of swinging up the pendulum is equivalent to minimizing the composition of a strongly convex cost with the control in $\tau$ steps of the discrete dynamics of the pendulum. With an Euler discretization of the continuous dynamics of the pendulum (44), one easily observes that the control in any $k \geq 2$ steps of the discrete dynamics satisfies the sufficient condition for global convergence outlined in Section 3. Hence, with a single final cost, this problem falls under the assumptions of Section 4. The convergence of both ILQR and IDDP algorithms towards a global minimum cost, namely a null cost, is observed in Fig. 1.

In Fig. 4, we consider a cost every $k$ steps, that is

$$h_t(x_t) = \begin{cases} (\theta_t - \pi)^2 + \omega_t^2 & \text{if } t \bmod k = 0 \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } t \in \{1, \dots, \tau\},$$

for $k \in \{1, 2\}$. We also consider 10 random initial sequence of control variables, i.e., $u_t^{(0)} \sim \mathcal{N}(0, \sigma)$, for $\sigma = 1/\tau = 100$, $t \in \{0, \dots, \tau - 1\}$.

By taking $k = 2$, we observe that ILQR and IDDP both converge to a 0 cost, hence a global minimum, across random initializations. As mentioned above, by taking $k > 1$ convergence to a global minimal cost is predicted by the theory in Section 3 and 4.

For $k = 1$, none algorithm converges to 0. However, this does not mean that they do not converge to a global minimum. In fact, one observes that accross random initializations, both ILQR and IDDP converge to the same cost. Namely, the standard deviation of the minimum cost computed by these algorithms across random initializations is $10^{-14}$. This suggests a global convergence behavior to the same point. While the theory developped in Section 3 and 4 explains the behavior for $k = 2$, the results for $k = 1$ suggest that global convergence of these algorithms may be ensured beyond the sufficient condition (3).

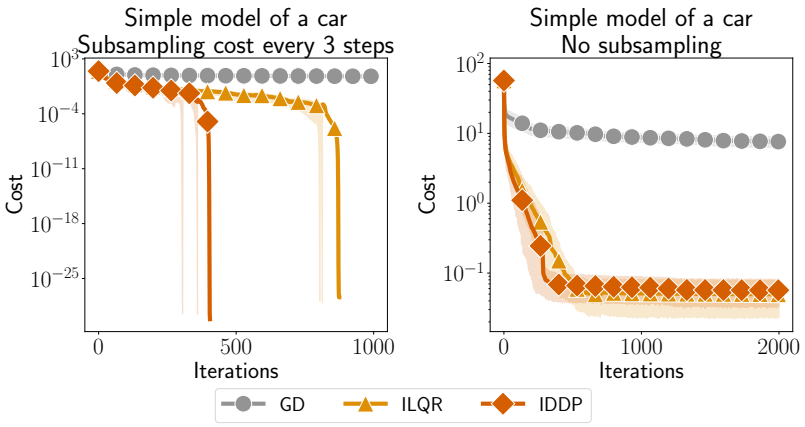


Figure 5: Cost along iterations of ILQR, IDDP and Gradient Descent (GD) on the car problem using a cost at each time step or every three time steps with varying initial controls. Shaded areas correspond to a 95% confidence intervals over 10 random initializations of the control sequences.

We also note that, in any case, IDDP converges faster than ILQR. This observation was also made by (Liao and Shoemaker, 1991) and further explored in the companion report (Roulet et al., 2022).

**Costs along iteration for the simple model of a car.** In Fig. 1, we considered a subsampled cost, such that a sufficient condition for global convergence outlined in Section 3.2 may be satisfied. We observe in Fig.1 convergence to a global minimal cost, namely a null cost, for both ILQR and IDDP algorithms.

In Fig. 5, we consider a cost at each time step (no subsampling of the costs, i.e., $k = 1$ in (45)) with 10 random initial control sequences, i.e., $u_t^{(0)} \sim \mathcal{N}(0, \sigma)$ for $\sigma = 1/\Delta = 25$, $t \in \{0, \dots, \tau - 1\}$. We also repeat the experiment with costs subsampled every 3 time steps with the same random initializations schemes.

For subsampled costs, i.e., $k = 3$ in (45), we observe convergence to global minimal costs (null costs) for both IDDP and ILQR algorithms across random initializations.

For non-subsampled costs, i.e., $k = 1$ in (45), the costs do not converge to 0. Contrarily to the pendulum case, we observed a discrepancy in the minimal cost reached after 2000 iterations. ILQR and IDDP reach, on average across initializations, costs of, respectively, $4.61 \cdot 10^{-2}$ and $5.68 \cdot 10^{-2}$ with standard deviations across initializations of, respectively, $3.93 \cdot 10^{-2}$ and $3.75 \cdot 10^{-2}$.

**Instantaneous rate of convergence.** The theoretical findings of Sec. 4 outline a priori three phases of convergence, sublinear, linear and quadratic. Convergence rates of ILQR and IDDP can be assessed through convergence rates in function values $\rho^{(k)} = (c^{(k+1)} - c^*)/(c^{(k)} - c^*)$ for $c^*$ the minimal cost as done in Appendix I, or by considering convergence in iterates through $\kappa^{(k)} = \|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)}\|_2 / \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^{(k-1)}\|_2$ as done in Fig. 6.

For the simple model of a car, in Fig. 6, we observe that the convergence rate in iterations of these algorithms remain close to 1 for many iterations (the x-axis in Fig. 6 is in reverted

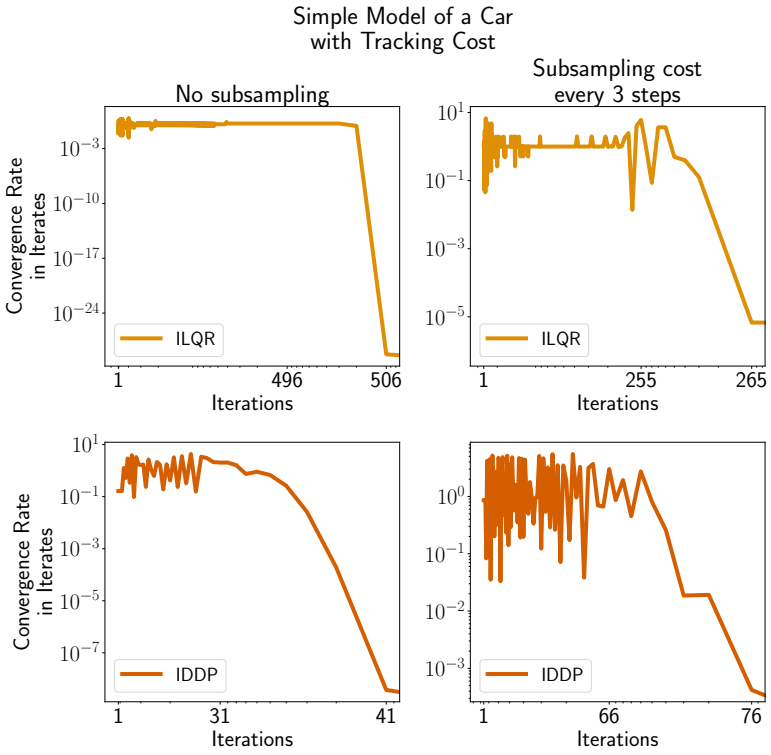


Figure 6: Convergence rate in iterates, $\kappa^{(k)} = \|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)}\|_2 / \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^{(k-1)}\|_2$, along iterations of ILQR and IDDP algorithms for the simple model of a car with or without subsampling the costs.

log-scale). This rate suddenly drops close to convergence akin to a local quadratic local convergence. This shows that the main difficulty of the problem arises for a long first phase of slow convergence.

**Surjectivity of the Jacobian.** The sufficient condition for global convergence (3) can be assessed by computing the minimal singular value $\sigma_{\min}(\nabla f^{[\tau]}(\boldsymbol{u}^{(k)}))$ of the transpose Jacobian of the control of $\tau$ steps of the discrete dynamics. In Fig. 7, we plot this minimal singular value along the iterations of the ILQR and IDDP algorithms. We consider discrete dynamics defined as the control in $k = 2$ and $k = 3$ steps of the discretization of the continuous dynamics of, respectively, the pendulum and the simple model of a car. Considering discrete dynamics in multiple steps amount to the subsampling of the costs presented in previous experiments.

We observe in Fig. 7 that $\sigma_{\min}(\nabla f^{[\tau]}(\boldsymbol{u}^{(k)}))$ is small yet bounded away from 0 along the iterations. This result concurs with the convergence to global minimal costs of these algorithms observed in the right panels of Fig. 4 and Fig. 5.

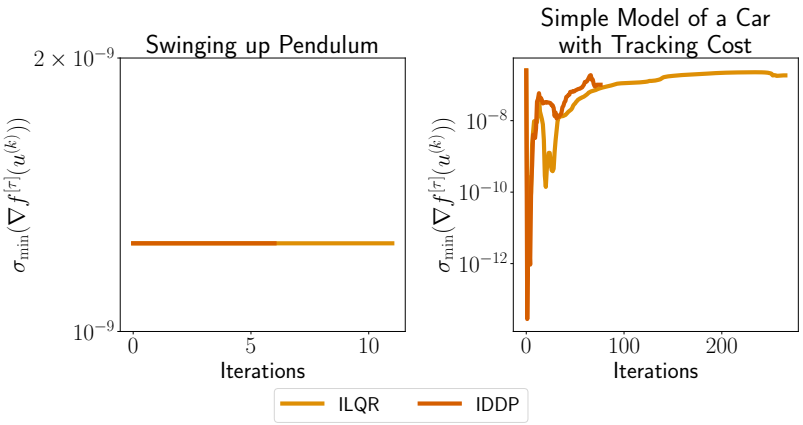


Figure 7: Minimal singular value of the transpose Jacobian of the control in $\tau$ steps of the discrete dynamics along the iterations of ILQR and IDDP algorithms. We consider discrete dynamics of a pendulum or of a simple model of a car stemming from the control in 2 and 3 steps respectively of the original discretization of the dynamics.

## 6. Related Work

**Nonlinear control algorithms based on local approximations and iterative refinements.** Nonlinear control problems of the form (1) stem from the discretization of generic optimal control problems in continuous time of the form

$$\begin{aligned} \min_{x(\cdot),u(\cdot)} \quad & \int_0^T h(x(t),u(T)) + h_T(x(T)) \\ \text{subject to} \quad & \dot{x}(t) = f(x(t),u(t)), \quad x(0) = \bar{x}_0, \end{aligned} \tag{46}$$

Continuous optimal control problems of the form (46) can be tackled in various ways (Diehl et al., 2006). One can approach the problem from a *dynamic programming* perspective to derive the Hamilton-Jacobi-Bellman equation, a partial differential equation in state space (Lions, 1982). Alternatively, one can derive necessary optimality conditions for (46) to derive a boundary value problem. Such a method is referred to as an *indirect method* and amounts to a "optimize then discretize" approach (Farshidian et al., 2017). Finally, problem (46) can be tackled by *direct methods* that consider finite dimensional approximations of the original infinite dimensional problem (46). Direct methods amount to a "discretize then optimize" approach (Diehl et al., 2006), they can further be split into different approaches. First, one may consider a finite representation of the continuous control $u(t)$ as piecewise constant functions whose values $q_1, \dots, q_\tau$ at each piece define the finite number of degrees of freedom. The problem still involves an ODE in the state variable, $\dot{x}(t) = f(x(t), u_{q_{1:\tau}}(t))$, albeit a simpler one. Tackling the problem with such a partial discretization is referred to as a *single shooting* method (Diehl et al., 2006; Bock and Plitt, 1984). *Collocation methods* (Von Stryk, 1993) consider discretizing both the states and controls, leading to a formulation like (1), that can benefit from advanced numerical in-

tegration methods. Finally, *multiple shooting* (Diehl et al., 2006; Bock and Plitt, 1984) combines both approaches. The system is split in multiple windows and for each window a single shooting method is used. We focus solely on the resulting discrete time nonlinear control problems (1) and refer the interested reader to, e.g., Rawlings et al. (2017, Section 8.5) for an overview of the approaches mentioned above.

One of the first approaches for nonlinear discrete time control problems (1) appear to be the Differential Dynamic Programming (DDP) methods developed by Mayne (1966); Jacobson and Mayne (1970); Mayne and Polak (1975). Its principle is to apply a dynamic programming procedure to the nonlinear system. The associated Bellman equation is approximately solved by considering its quadratic approximation around the current trajectory. A set of policies is computed along this process and applied to the original dynamics as if the true solutions of the Bellman equations were found. A modern account is provided in the companion report (Roulet et al., 2022) for reference; see also (Liao and Shoemaker, 1992). Numerous variants of DDP have been developed to account for constraints or noise in the dynamics (Li and Todorov, 2004; Tassa et al., 2007, 2014; Giftthaler et al., 2018). Among those, the IDDP algorithm can be seen to follow the same principle as DDP except that *linear-quadratic* approximations à la Gauss-Newton are used in place of the quadratic approximations of the Bellman equation akin to Newton's method.

DDP approaches differ from the implementation of classical optimization algorithms such as a Newton, quasi-Newton or Gauss-Newton method for discrete nonlinear control problems. Bock (1983); Bock and Plitt (1984) first presented such approaches referred to as *direct multiple shooting*. Detailed and efficient implementations of Newton's method exploiting the dynamical structure of the problem were presented by Pantoja (1988); Dunn and Bertsekas (1989). A linear algebraic viewpoint on these implementations was presented by Wright (1990, 1991a), that enabled the use of fast linear solvers exploiting the structure of nonlinear control problems (Wright, 1991b; Jerez et al., 2011; Rao et al., 1998). In particular, Wright (1991a) presents alternative resolutions of the linear quadratic subproblem using a "Riccati-like" recursion that slightly differs from the resolution by dynamic programming presented here. Wright (1991a) further developed parallel implementations of algorithms solving the LQR problems. We do not delve into the specific implementations of the oracles used in ILQR or IDDP and rather focus on the global behavior of the algorithms.

This viewpoint was further generalized to handle nonlinear inequalities in model predictive control (Diehl et al., 2009) or even generic graphs of computations (Srinivasan and Todorov, 2015). The ILQR algorithm can be seen as an instance of direct multiple shooting, namely, an instance of a generalized Gauss-Newton method (Sideris and Bobrow, 2005) which uses *linear-quadratic* approximations of the problem decomposed along the dynamics.

Detailed implementations of DDP approach (quadratic approximation of Bellman equation), IDDP approach (linear-quadratic approximation of Bellman equation), Newton (quadratic approximation of the objective) and ILQR (linear-quadratic approximation of the objective) are presented in the companion report (Roulet et al., 2022) to highlight their common points and differences.

The decomposition of the problem at several scales by means of some quadratic approximations have also been developed and studied by Messerer et al. (2021); Frasch et al. (2015); Verschueren et al. (2016); Houska and Diehl (2013).

**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 by, 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 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 of 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) for nonlinear fitting. Nesterov (2007) 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 contrast, we consider twice differentiable costs, for which we build a quadratic model, leading to *generalized* Gauss-Newton methods. In the context of nonlinear control problems, generalized 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. Messerer et al. (2021) considered also convergence of generalized Gauss-Newton methods. However, Messerer et al. (2021) analyzes such algorithms without regularization, nor line-search or trust-region techniques, resulting in possibly divergent algorithms or only local convergence guarantees that fail to explain global phenomena presented in the introduction. By adding a regularization scheme, we are able to ensure global convergence, and to provide practical guidance on the choice of regularization (Algorithm 2). Baumgärtner et al. (2023) also considered the local convergence properties of ILQR, IDDP to determine that they share the same linear convergence rate locally. We considered here more general convergence properties either towards stationary points or global minima given additional assumptions. Finally, our results are quantitative, relating the region of quadratic convergence to the smallest singular value of the transposed Jacobian.

As mentioned earlier, a Newton's method could just as well be implemented to exploit the dynamical structure of the problem (Dunn and Bertsekas, 1989). Several caveats lend still in favor of a Gauss-Newton method. First, a Newton's method (or a DDP approach) requires computing and storing the second order information associated to the dynamics at the intermediate states, although the storage issue can be mitigated by an adequate implementation in a differentiable programming framework (Nganga and Wensing, 2021; Roulet et al., 2022). Second, Newton's method does not compute a priori descent directions if the Hessian is not positive definite. Hessian modifications (Nocedal and Wright, 2006, Section 3.4) may be necessary to ensure a descent direction such that a linesearch can be used. On the other hand, for generic functions, Newton's method is known to converge locally at a quadratic rate (Nesterov, 2018), which is a priori not true for a generalized Gauss-Newton method. Our analysis shows that in some nonlinear control problems generalized

Gauss-Newton methods can converge with such a local quadratic rate, just as observed empirically. Our analysis stems in fact from considering a generalized Gauss-Newton method as an approximate Newton method in the space of the trajectories which enable us to recover the fast local rate of convergence of Newton's method by appropriately controlling the approximation error.

**Convergence analysis of differentiable dynamic programming methods.** DDP 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. Their principle is to apply a dynamic programming procedure to the nonlinear system for which the associated Bellman equation is approximately solved by considering its quadratic approximation around the current trajectory. A set of policies is computed along this process and applied to the original dynamics as if the true solutions of the Bellman equations were found. The IDDP algorithm can be seen to follow the same principle except that linear-quadratic approximations à la Gauss-Newton are used in place of the quadratic approximations of the Bellman equation. 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 mainly focused on local convergence guarantees (Mayne and Polak, 1975; Murray and Yakowitz, 1984; Liao and Shoemaker, 1991) or convergence guarantees towards controls satisfying first-order necessary optimality conditions (Polak, 2011). The local quadratic convergence analysis of 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 (2) 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 the IDDP algorithm which uses linear-quadratic approximations developed by Tassa et al. (2012) and extended by Giftthaler et al. (2018). The IDDP algorithm benefits from a smaller per-iteration cost compared to DDP, as IDDP does not require computing intermediate second-order information associated to the dynamics, see Roulet et al. (2022) for more details.

**Sufficient conditions for global convergence.** Discrete time nonlinear control problems of the form (2) 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 (2), 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 (3) can be verified on simple instances.

Our assumption is based on analyzing the gradient dominating property of the objective of problem (2) 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 discrete linearized trajectories in one step. In a similar spirit, a controllability assumption on the discrete linearized trajectories in *several* steps was considered to analyze the local convergence of MPC controllers by Na and Anitescu (2020, Assumption 2) following Xu and Anitescu (2019). Compared to Xu and Anitescu (2019); Na and Anitescu (2020), we consider *global* convergence results to minimizers, which justifies a stronger assumption. In addition, compared to Xu and Anitescu (2019); Na and Anitescu (2020), 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). In particular, we exploit the existence of a feedback linearization scheme by considering a multi-rate sampling scheme to ensure our sufficient condition. Using multi-rate sampling was proposed in the early work of Grizzle and Kokotovic (1988) on discrete time feedback linearization schemes.

We consider only understanding the performance of two popular algorithms, ILQR and IDDP, and derive sufficient conditions for global convergence adapted to these algorithms. Several variants can be considered. In particular, given the surjectivity of the Jacobian of the dynamics (3), the problem may also be rephrased as a feasibility problem and tackled differently. Namely, the minimizers $x_t^*$ of the costs $h_t$ could be computed offline, and the problem would reduce to fit a nonlinear model of the states, here described by the trajectories given by the dynamics, to the minimizers $x_t^*$. Such feasibility problems may be tackled for example by penalty method as done by Kim and Wright (2016). However, such penalty methods may dismiss the dynamical structure of the problem. Moreover, ILQR or IDDP methods can tackle the original problem at once, rather than deriving a two-stage method consisting in computing first the minimizers of the costs.

## 7. Conclusion

We have detailed global computational complexities of the ILQR and IDDP algorithms for problems of the form (2). Our analysis decomposes at several scales. At the scale of the whole trajectory, the problem can be summarized as 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.

The sufficient conditions for global convergence are restricted to problems without costs or constraints on the control variables. Moreover, they may not be applicable in usual scenarios with costs that are not subsampled. As future work, one 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 and Disclosure of Funding

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.

## References

Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 2018.

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

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

Aram V Arutyunov and Richard B Vinter. A simple 'finite approximations' proof of the Pontryagin maximum principle under reduced differentiability hypotheses. *Set-valued analysis*, 12(1):5–24, 2004.

Katrin Baumgärtner, Florian Messerer, and Moritz Diehl. Local convergence behaviour of generalized gauss-newton multiple shooting, single shooting and differential dynamic programming. *arXiv preprint arXiv:2301.04047*, 2023.

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

El Houcine Bergou, Youssef Diouane, and Vyacheslav Kungurtsev. Convergence and complexity analysis of a Levenberg-Marquardt algorithm for inverse problems. *Journal of Optimization Theory and Applications*, 185(3):927–944, 2020.

Dimitri Bertsekas. *Dynamic Programming and Stochastic Control*. Academic Press, 1976.

Dimitri Bertsekas. *Nonlinear Programming*, volume 4. Athena Scientific, 2016.

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

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

Åke Björck. *Numerical methods for least squares problems*. SIAM, 2024.

Hans Georg Bock. Recent advances in parameteridentification techniques for ode. In *Numerical Treatment of Inverse Problems in Differential and Integral Equations: Proceedings of an International Workshop, Heidelberg, Fed. Rep. of Germany, August 30—September 3, 1982*, pages 95–121. Springer, 1983.

Hans Georg Bock and Karl-Josef Plitt. A multiple shooting algorithm for direct solution of optimal control problems. *IFAC Proceedings Volumes*, 17(2):1603–1608, 1984.

Jérôme Bolte, Trong Phong Nguyen, Juan Peypouquet, and Bruce W Suter. From error bounds to the complexity of first-order descent methods for convex functions. *Mathematical Programming*, 165(2):471–507, 2017.

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

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

Krishna Busawon and Mohamed Djemaï. Algorithms for transformation into the extended Jordan controllable and observable forms. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC)*, pages 1764–1769, 2009.

Michael L. Bynum, Gabriel A. Hackebeil, William E. Hart, Carl D. Laird, Bethany L. Nicholson, John D. Siirola, Jean-Paul Watson, and David L. Woodruff. *Pyomo–optimization modeling in python*, volume 67. Springer Science & Business Media, third edition, 2021.

Frank Clarke. Optimal control and the true hamiltonian. *SIAM Review*, 21(2):157–166, 1979.

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

Bolei Di and Andrew Lamperski. 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, 2019.

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

Moritz Diehl, Hans Georg Bock, Holger Diedam, and P-B Wieber. Fast direct multiple shooting algorithms for optimal robot control. *Fast motions in biomechanics and robotics: optimization and feedback control*, pages 65–93, 2006.

Moritz Diehl, Hans Joachim Ferreau, and Niels Haverbeke. Efficient numerical methods for nonlinear mpc and moving horizon estimation. *Nonlinear model predictive control: towards new challenging applications*, pages 391–417, 2009.

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

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

Iain Dunning, Joey Huchette, and Miles Lubin. Jump: A modeling language for mathematical optimization. *SIAM review*, 59(2):295–320, 2017.

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

Farbod Farshidian, Michael Neunert, Alexander W Winkler, Gonzalo Rey, and Jonas Buchli. An efficient optimal planning and control framework for quadrupedal locomotion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 93–100. IEEE, 2017.

Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1467–1476, 2018.

Janick V Frasch, Sebastian Sager, and Moritz Diehl. A parallel quadratic programming method for dynamic optimization problems. *Mathematical programming computation*, 7 (3):289–329, 2015.

Walter Gautschi. *Numerical analysis*. Springer Science & Business Media, 2011.

Markus Giftthaler, Michael Neunert, Markus Stäuble, Jonas Buchli, and Moritz Diehl. 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, 2018.

Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.

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

Boris Houska and Moritz Diehl. A quadratically convergent inexact sqp method for optimal control of differential algebraic equations. *Optimal Control Applications and Methods*, 34 (4):396–414, 2013.

Alberto Isidori. *Nonlinear Control Systems*. Springer-Verlag, 3rd edition, 1995.

David Jacobson and David Mayne. *Differential Dynamic Programming*. Elsevier, 1970.

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

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

Wilson Jallet, Antoine Bambade, Etienne Arlaud, Sarah El-Kazdadi, Nicolas Mansard, and Justin Carpentier. Proxddp: Proximal constrained trajectory optimization. 2023.

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

Juan L Jerez, Eric C Kerrigan, and George A Constantinides. A condensed and sparse qp formulation for predictive control. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 5217–5222. IEEE, 2011.

Sham Kakade, Akshay Krishnamurthy, Kendall Lowrey, Motoya Ohnishi, and Wen Sun. Information theoretic regret bounds for online nonlinear control. *arXiv preprint arXiv:2006.12466*, 2020.

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

Taedong Kim and Stephen J Wright. An s $\ell$ 1 lp-active set approach for feasibility restoration in power systems. *Optimization and Engineering*, 17(2):385–419, 2016.

Andrew D Lewis. The maximum principle of Pontryagin in control and in optimal control. *Handouts for the course taught at the Universitat Politecnica de Catalunya*, 2006.

Weiwei Li and Emanuel Todorov. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *First International Conference on Informatics in Control, Automation and Robotics*, volume 2, pages 222–229. SciTePress, 2004.

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

Li-Zhi Liao and Christine A Shoemaker. Advantages of differential dynamic programming over Newton's method for discrete-time optimal control problems. Technical report, Cornell University, 1992.

Yiheng Lin, Yang Hu, Guanya Shi, Haoyuan Sun, Guannan Qu, and Adam Wierman. Perturbation-based regret analysis of predictive control in linear time varying systems. *Advances in Neural Information Processing Systems*, 34:5174–5185, 2021.

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

Pierre-Louis Lions. *Generalized Solutions of Hamilton-Jacobi Equations*. Pitman, 1982.

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

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

David Mayne. A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems. *International Journal of Control*, 3(1):85–95, 1966.

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

Florian Messerer, Katrin Baumgärtner, and Moritz Diehl. Survey of sequential convex programming and generalized Gauss-Newton methods. *ESAIM. Proceedings and Surveys*, 71:64, 2021.

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

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

Sen Na and Mihai Anitescu. Superconvergence of online optimization for model predictive control. *arXiv preprint arXiv:2001.03707*, 2020.

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

Yurii Nesterov. *Lectures on convex optimization*. Springer, 2018.

John Nganga and Patrick Wensing. Accelerating second-order differential dynamic programming for rigid-body systems. *IEEE Robotics and Automation Letters*, 6(4):7659–7666, 2021.

Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

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

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

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

Lev Pontryagin, Vladimir Boltyansky, Revaz Gamkrelidze, and Evgenii Mischenko. The mathematical theory of optimal processes. *Wiley-Interscience*, 1963.

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

James Blake Rawlings, David Q Mayne, Moritz Diehl, et al. *Model predictive control: theory, computation, and design*, volume 2. Nob Hill Publishing Madison, WI, 2017.

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

Vincent Roulet, Siddhartha Srinivasa, Dmitriy Drusvyatskiy, and Zaid Harchaoui. Iterative linearized control: stable algorithms and complexity guarantees. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5518–5527, 2019.

Vincent Roulet, Siddhartha Srinivasa, Maryam Fazel, and Zaid Harchaoui. Iterative linear quadratic optimization for nonlinear control: Differentiable programming algorithmic templates. *arXiv preprint arXiv:2207.06362*, 2022.

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

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

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

Akshay Srinivasan and Emanuel Todorov. Graphical newton. *arXiv preprint arXiv:1508.00952*, 2015.

Yue Sun and Maryam Fazel. Learning optimal controllers by policy gradient: Global optimality via convex parameterization. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 4576–4581. IEEE, 2021.

Yuval Tassa, Tom Erez, and William Smart. Receding horizon differential dynamic programming. *Advances in neural information processing systems*, 20, 2007.

Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913, 2012.

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

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

Robin Verschueren, Niels van Duijkeren, Rien Quirynen, and Moritz Diehl. Exploiting convexity in direct optimal control: a sequential convex quadratic programming method. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 1099–1104. IEEE, 2016.

Robin Verschueren, Gianluca Frison, Dimitris Kouzoupis, Jonathan Frey, Niels van Duijkeren, Andrea Zanelli, Branimir Novoselnik, Thivaharan Albin, Rien Quirynen, and Moritz Diehl. acados – a modular open-source framework for fast embedded optimal control. *Mathematical Programming Computation*, 2021.

Oskar Von Stryk. *Numerical solution of optimal control problems by direct collocation*. Springer, 1993.

Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106:25–57, 2006.

Tyler Westenbroek, Max Simchowitz, Michael Jordan, and Shankar Sastry. On the stability of nonlinear receding horizon control: A geometric perspective. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 742–749, 2021.

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

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

Stephen Wright. Structured interior point methods for optimal control. In *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*, pages 1711–1716. IEEE, 1991b.

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

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

Kaiqing Zhang, Zhuoran Yang, and Tamer Basar. Policy optimization provably converges to nash equilibria in zero-sum linear quadratic games. *Advances in Neural Information Processing Systems*, 32, 2019.

Kaiqing Zhang, Bin Hu, and Tamer Basar. Policy optimization for $\mathcal{H}_2$ linear control with $\mathcal{H}_\infty$ robustness guarantee: Implicit regularization and global convergence. In *Learning for Dynamics and Control*, pages 179–190. PMLR, 2020.

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

# Appendix

## Table of Contents



## A. Index of Constants

Table 1 presents an index of the constants used in the main results of the paper in Sec. 4 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 (2). In particular, constants governing the compositional problem (24) defined by the total cost $h$ and the control $g$ in $\tau$ steps of $f$ for fixed initial state

| Notation | Definition | Interpretation | Unit |
|---|---|---|---|
| $\sigma_f$ | $\inf_{x,u} \sigma_{\min}(\nabla_u f(x,u))$ | 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 $f^{[\tau]}(x_0,u)$ | $x/u$ |
| $L_g, L_{f^{[\tau]}}$ | $L_f^{xx}(l_f^u S)^2 + 2L_f^{xu} l_f^u S + L_f^{uu} S$ | Lip. cont. of $\nabla_{\boldsymbol{u}} f^{[\tau]}(x_0,u)$ | $x/u^2$ |
| $S$ | $\sum_{t=0}^{\tau-1} (l_f^x)^t$ | Auxiliary constant | 1 |
| $\mu_h$ | $\inf_x \sigma_{\min}(\nabla^2 h_t(x))$ | Strong convexity param. of the costs | $h/x^2$ |
| $L_h$ | $\sup_x \sigma_{\max}(\nabla^2 h_t(x))$ | Lip. cont. of gradients of the costs | $h/x^2$ |
| $M_h$ | $l_{\nabla^2 h_t}$ | Lip. cont. of Hessians of the costs | $h/x^3$ |
| $\rho_g$ | $l_g/\sigma_g$ | Cond. nb of $\nabla g(\boldsymbol{u})$ | 1 |
| $\rho_h$ | $L_h/\mu_h$ | Cond. nb of the costs | 1 |
| $\theta_h$, $\vartheta_h$ | $M_h/(2\mu_h^{3/2})$ | Self-concordance of the costs | $1/\sqrt{h}$ |
| $\theta_g$ | $L_g/(\sigma_g^2\sqrt{\mu_h})$ | Scaling param. for $g$ | $1/\sqrt{h}$ |
| $\beta$ | $M_h l_g^2/(3L_g L_h)$ | Cond. nb for global conv. of ILQR | 1 |
| $\alpha$ | $4\rho_g^2\rho_h(\beta+1)$ | Cond. nb for global conv. of ILQR | 1 |
| $l$ | $\sqrt{L_h} l_g$ | Lip. cont. of $g$ w.r.t. $h$ in Hyp. 11 | $\sqrt{h}/u$ |
| $L$ | $\sqrt{L_h} L_g$ | Lip. cont. of grad. of $g$ w.r.t. $h$ in Hyp. 11 | $\sqrt{h}/u^2$ |
| $\sigma$ | $\sqrt{\mu_h}\sigma_g$ | Surj. param. of $g$ w.r.t. $h$ | $\sqrt{h}/u$ |
| $\varrho$ | $l/\sigma = \sqrt{\rho_h}\rho_g$ | Cond. nb of $g$ w.r.t. $h$ | 1 |
| $\vartheta_g$ | $L/\sigma^2 = \sqrt{\rho_h}\theta_g$ | Scaling param. of $g$ w.r.t. $h$ | $1/\sqrt{h}$ |
| $\eta$ | See Corollary 51 | Relative bound btw DDP & LQR: $\frac{\|\mathrm{DDP}_\nu(\mathcal{J})(\boldsymbol{u}) - \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2}{\|\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})\|_2^2} \leq \eta$ | $1/u$ |
| $\chi$ | $l_g\eta/L_g$ | Factor of smoothness for IDDP | 1 |

Table 1: Index of constants used in the paper.

(see (24) and Def. 5), 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 13. For the local convergence, constants $\sigma, l, L, \theta_h$ can be defined without strong convexity as presented in Assumption 17.

## B. Optimality Conditions

### B.1 Necessary Optimality Conditions

We recall necessary optimality conditions for nonlinear control problems in continuous and discrete time to underline their discrepancies. The problem we consider in continuous time is

$$\begin{aligned} &\min_{\substack{x\in\mathcal{C}^1([0,1],\mathbb{R}^{n_x})\\ u\in\mathcal{C}([0,1],\mathbb{R}^{n_u})}} \int_0^1 h(x(t),u(t),t)dt + h(x(1),1) \\ &\text{subject to} \quad \dot{x}(t) = \psi(x(t),u(t),t), \quad \text{for } t\in[0,1] \quad x(0)=\bar{x}_0, \end{aligned} \tag{47}$$

where $\mathcal{C}([0,1],\mathbb{R}^d)$ and $\mathcal{C}^1([0,1],\mathbb{R}^d)$ denote the set of continuous and continuously differentiable functions from $[0,1]$ onto $\mathbb{R}^d$ respectively, and we assume $\psi$ and $h$ to be continuously differentiable. By using an Euler discretization scheme with discretization stepsize $\Delta = 1/\tau$, we get the discrete time control problem

$$\begin{aligned} &\min_{\substack{x_0,\dots,x_\tau\in\mathbb{R}^{n_x}\\ u_0\dots,u_{\tau-1}\in\mathbb{R}^{n_u}}} \sum_{t=0}^{\tau-1} h_t(x_t,u_t) + h_\tau(x_\tau) \\ &\text{subject to} \quad x_{t+1} = x_t + \psi_t(x_t,u_t), \quad \text{for } t\in\{0,\dots,\tau-1\}, \quad x_0=\bar{x}_0, \end{aligned} \tag{48}$$

where $x_t = x(\Delta t)$, $u_t = u(\Delta t)$, $h_t = \Delta h(\cdot,\cdot,\Delta t)$, $h_\tau = h(\cdot,1)$, $\psi_t = \Delta\psi(\cdot,\cdot,\Delta t)$. Compared to problem (1), we have $x_t + \psi_t(x_t,u_t) = f_t(x_t,u_t)$.

**Continuous Time Necessary Optimality Condition.** Necessary optimality conditions for the continuous time control problem are known as Pontryagin's maximum principle, recalled below. See Arutyunov and Vinter (2004) for a recent proof and Lewis (2006) for a comprehensive overview.

**Theorem 26 (Pontryagin's maximum principle (Pontryagin et al., 1963))** *Define the Hamiltonian associated with problem (47) as*

$$H(x(t),u(t),\lambda(t),t) = \lambda(t)^\top\psi(x(t),u(t),t) - h(x(t),u(t),t).$$

*A trajectory $x\in\mathcal{C}^1([0,1],\mathbb{R}^{n_x})$ and a control function $u\in\mathcal{C}([0,1],\mathbb{R}^{n_u})$ are optimal if there exists $\lambda\in\mathcal{C}^1([0,1],\mathbb{R}^{n_x})$ such that*

$$\dot{x}(t) = \nabla_{\lambda(t)}H(x(t),u(t),\lambda(t),t) \quad \text{for all } t\in[0,1], \tag{C1}$$

$$\text{with } x(0)=\bar{x}_0$$

$$\dot{\lambda}(t) = -\nabla_{x(t)}H(x(t),u(t),\lambda(t),t) \text{ for all } t\in[0,1], \tag{C2}$$

$$\text{with } \lambda(1) = -\nabla_{x(1)}h(x(1),1)$$

$$H(x(t),u(t),\lambda(t),t) = \max_{u\in\mathbb{R}^{n_u}} H(x(t),u,\lambda(t),t) \qquad \text{for all } t\in[0,1]. \tag{C3}$$

**Discrete Time Necessary Optimality Conditions.** In comparison, necessary optimality conditions for the discretized problem (48) are given by considering the Karush-Kuhn-Tucker conditions of the problem, or equivalently by considering a sequence of controls such that the gradient of the objective is null (Bertsekas, 2016).

**Lemma 27** *Define the Hamiltonian associated with problem* (48) *as*

$$H_t(x_t, u_t, \lambda_{t+1}) = \lambda_{t+1}^\top \psi_t(x_t, u_t) - h_t(x_t, u_t)$$

*A trajectory* $x_0, \dots, x_\tau \in \mathbb{R}^{n_x}$ *and a sequence of controls* $u_0, \dots, u_{\tau-1} \in \mathbb{R}^{n_u}$ *are optimal if there exists* $\lambda_1, \dots, \lambda_\tau \in \mathbb{R}^{n_x}$ *such that*

$$x_{t+1} - x_t = \nabla_{\lambda_{t+1}} H_t(x_t, u_t, \lambda_{t+1}) \quad \text{for all } t \in \{0, \dots, \tau-1\}, \text{ with } x_0 = \bar{x}_0 \tag{D1}$$

$$\lambda_{t+1} - \lambda_t = -\nabla_{x_t} H_t(x_t, u_t, \lambda_{t+1}) \quad \text{for all } t \in \{1, \dots, \tau-1\}, \text{ with } \lambda_\tau = -\nabla h_\tau(x_\tau) \tag{D2}$$

$$0 = \nabla_{u_t} H_t(x_t, u_t, \lambda_{t+1}) \quad \text{for all } t \in \{0, \dots, \tau-1\}. \tag{D3}$$

**Proof** Necessary optimality conditions are given by considering stationary points of the Lagrangian (Bertsekas, 1976). The Lagrangian of problem (48) is given for $\boldsymbol{\lambda} = (\lambda_1; \dots; \lambda_\tau)^\top$, $\boldsymbol{x} = (x_1; \dots; x_\tau)$, $\boldsymbol{u} = (u_0; \dots; u_{\tau-1})$ as, for $x_0 = \bar{x}_0$ fixed,

$$\begin{aligned} L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) &= \sum_{t=0}^{\tau-1} h_t(x_t, u_t) + \sum_{t=0}^{\tau-1} \lambda_{t+1}^\top (x_{t+1} - x_t - \psi_t(x_t, u_t)) + h_\tau(x_\tau) \\ &= \sum_{t=0}^{\tau-1} h_t(x_t, u_t) + \sum_{t=1}^{\tau-1} \left( x_t^\top (\lambda_t - \lambda_{t+1}) - \lambda_{t+1}^\top \psi_t(x_t, u_t) \right) \\ &\quad + h_\tau(x_\tau) + \lambda_\tau^\top x_\tau - \lambda_1^\top (x_0 + \psi_0(x_0, u_0)). \end{aligned}$$

We have then, for $t \in \{0, \dots, \tau-1\}$,

$$\begin{aligned} \nabla_{\lambda_{t+1}} L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = 0 \iff x_{t+1} - x_t &= \psi_t(x_t, u_t) \\ &= \nabla_{\lambda_{t+1}} H_t(x_t, u_t, \lambda_{t+1}), \end{aligned}$$

$$\begin{aligned} \nabla_{u_t} L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = 0 \iff 0 &= -\nabla_{u_t} \psi_t(x_t, u_t) \lambda_{t+1} + \nabla_{u_t} h_t(x_t, u_t) \\ &= -\nabla_{u_t} H_t(x_t, u_t, \lambda_{t+1}), \end{aligned}$$

We have, for $t \in \{1, \dots, \tau-1\}$,

$$\begin{aligned} \nabla_{x_t} L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = 0 \iff \lambda_{t+1} - \lambda_t &= -\nabla_{x_t} \psi_t(x_t, u_t) \lambda_{t+1} + \nabla_{x_t} h_t(x_t, u_t) \\ &= -\nabla_{x_t} H_t(x_t, u_t, \lambda_{t+1}), \end{aligned}$$

Finally, for $t = \tau$, we have $\nabla_{x_\tau} L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = 0 \iff \nabla h_\tau(x_\tau) + \lambda_\tau = 0$. ■

**Common Points and Discrepancies between Continuous and Discrete Time.** The first two necessary optimality conditions (D1) and (D2) for the discretized problem correspond to the discretizations of the first two necessary optimality conditions (C1) and (C2) for the continuous time problem. The third condition differs since, in discrete time, the control variables only need to be stationary points of the Hamiltonian. One may wonder whether condition (D3) could be replaced by a stronger necessary optimality condition of the form

$$u_t \in \arg\max_{u \in \mathbb{R}^{n_u}} H_t(x_t, u_t, \lambda_{t+1}). \tag{D4}$$

If the Hamiltonian is convex w.r.t. to the control variable, i.e., $H_t(x_t, \cdot, \lambda_{t+1})$ is concave. If, e.g., the costs $h_t(x_t, \cdot)$ are convex and if the dynamics are affine input of the form $\psi_t(x_t, u_t) = a_t(x_t) + B_t(x_t)u_t$, then condition (D3) is equivalent to condition (D4). However, generally, condition (D4) is not a necessary optimality condition for the discrete-time control problem as shown in the counter-example 2.

**Example 2** *Consider the continuous time control problem*

$$\min_{x(t), u(t) \in \mathcal{C}([0,1], \mathbb{R})} \int_0^1 (ax(t)^2 - u(t)^2)dt + ax(1)^2$$
$$\text{subject to} \quad \dot{x}(t) = u(t), \quad x(0) = 0,$$

*for some $a > 0$ and the associated discrete time control problem, for an Euler scheme with discretization $\Delta = 1/\tau$,*

$$\min_{\substack{x_0, \dots, x_\tau \in \mathbb{R} \\ u_0, \dots, u_{\tau-1} \in \mathbb{R}}} \sum_{t=0}^{\tau-1} \Delta(ax_t^2 - u_t^2) + ax_\tau^2$$
$$\text{subject to} \quad x_{t+1} = x_t + \Delta u_t, \quad x_0 = 0.$$

*The Hamiltonians in continuous time, $H(x(t), u, \lambda(t)) = \lambda(t)^\top u + u^2 - ax(t)^2$, and in discrete time, $H_t(x_t, u_t, \lambda_{t+1}) = \Delta\lambda_{t+1}^\top u + u^2 - ax_t^2$, are both strongly convex in $u$ such that neither condition (C3) nor (D4) can be satisfied.*

*According to Theorem 26, this means that the continuous time control problem has no solution. This can be verified by expressing the continuous time control problem uniquely in terms of the trajectory $x(t)$ as*

$$\min_{x(t): x(0)=0} \left\{ C(x) = \int_0^1 (ax(t)^2 - \dot{x}(t)^2)dt + ax(1)^2 \right\}.$$

*By considering functions of the form $x_k(t) = \exp(t^k) - 1$, we observe that the corresponding costs are unbounded below, namely, $C(x_k) \leq 2a(\exp(1) - 1)^2 - k^2/(2k-1) \underset{k \to +\infty}{\to} -\infty$ which shows that the problem is unbounded below and has no minimizer.*

*On the other hand, the discrete time control problem can be expressed in terms of the control variables as*

$$\min_{\boldsymbol{u} \in \mathbb{R}^{\tau n_u}} a\Delta^2 \boldsymbol{u}^\top D^{-\top} J D^{-1} \boldsymbol{u} - \Delta \|\boldsymbol{u}\|_2^2,$$

*where* $J = \mathbf{diag}(\Delta, \dots, \Delta, 1)$, $D = \mathrm{I} - \sum_{t=1}^{\tau-1} e_{t+1} e_t^\top$. *We have, using that* $\Delta < 1$ *for the first inequality,* $\boldsymbol{u}^\top D^{-\top} J D^{-1} \boldsymbol{u} \geq \Delta \|D^{-1}\boldsymbol{u}\|_2^2 \geq \Delta \sigma_{\min}(D^{-1})^2 \|\boldsymbol{u}\|_2^2 = \Delta \|\boldsymbol{u}\|_2^2 / \|D\|_2^2 \geq \Delta \|\boldsymbol{u}\|_2^2/4$. *Hence, for any* $a$ *such that* $a\Delta^2/4 > 1$, *the above problem is strongly convex and has a unique solution. Yet, if condition* (D4) *was necessary the discrete control problem should not have a solution since condition* (D4) *cannot be satisfied.*

**Alternative derivation.** Necessary optimality conditions for the discretized problem (1) can be derived from Lemma 27 using the correspondence $x_t + \psi_t(x_t, u_t) = f_t(x_t, u_t)$. We can also derive the necessary optimality conditions simply by considering a sequence of control variables $\boldsymbol{u} = (u_0; \dots; u_{\tau-1})$ that minimize the objective $\mathcal{J}$ defined in (4). Namely, the gradient of the objective $\mathcal{J}$ on $\boldsymbol{u} = (u_0; \dots; u_{\tau-1})$ can be obtained by gradient back-propagation as follows. First the states corresponding to the control variables are computed in a forward pass

$$x_{t+1} = f_t(x_t, u_t), \quad \text{for } t \in \{0, \dots, \tau - 1\}$$

starting from $x_0 = \bar{x}_0$. Then the gradients $\nabla \mathcal{J}(\boldsymbol{u}) = (g_0; \dots; g_{\tau-1})$ are computed in a backward pass as

$$\begin{aligned}
\lambda_\tau &= \nabla h_\tau(x_\tau), \\
\lambda_t &= \nabla_{x_t} f_t(x_t, u_t)^\top \lambda_{t+1} + \nabla_{x_t} h_t(x_t, u_t), \quad \text{for } t \in \{0, \dots, \tau - 1\}, \\
g_t &= \nabla_{u_t} f_t(x_t, u_t)^\top \lambda_{t+1} + \nabla_{u_t} h_t(x_t, u_t), \quad \text{for } t \in \{0, \dots, \tau - 1\}.
\end{aligned}$$

One easily verifies then that having $g_t = 0$ for all $t \in \{0, \dots, \tau - 1\}$ correspond to the optimality conditions presented in Lemma 27 with the correspondence $x_t + \psi_t(x_t, u_t) = f_t(x_t, u_t)$.

### B.2 Sufficient Optimality Conditions

Sufficient optimality conditions can also be derived following sufficent optimality conditions in continuous time presented by Mangasarian (1966); Arrow (1968); Kamien and Schwartz (1971). We start by rewriting problem (1) as

$$\begin{aligned}
&\min_{\substack{x_0, \dots, x_\tau \in \mathbb{R}^{n_x} \\ \delta_0, \dots, \delta_{\tau-1} \in \mathbb{R}^{n_x}}} \sum_{t=0}^{\tau-1} m_t(x_t, \delta_t) + h_\tau(x_\tau), \quad \text{where } m_t(x_t, \delta_t) = \inf_{\substack{u \in \mathbb{R}^{n_u} \\ \delta_t = f(x_t, u_t) - x_t}} h_t(x_t, u) \qquad (49) \\
&\text{subject to} \quad \delta_t = x_{t+1} - x_t, \; x_0 = \bar{x}_0.
\end{aligned}$$

Sufficient conditions can be expressed through the true Hamiltonian, presented by Clarke (1979), and defined as the convex conjugate of $m_t(x_t, \cdot)$, i.e., for $x_t, \lambda_{t+1} \in \mathbb{R}^{n_x}$,

$$\begin{aligned}
\bar{H}_t(x_t, \lambda_{t+1}) &= \sup_{\delta \in \mathbb{R}^{n_x}} \lambda_{t+1}^\top \delta - m_t(x_t, \delta) \\
&= \sup_{u \in \mathbb{R}^{n_u}} \lambda_{t+1}^\top (f(x_t, u_t) - x_t) - h_t(x_t, u) \\
&= \sup_{u \in \mathbb{R}^{n_u}} H_t(x_t, u, \lambda_{t+1}),
\end{aligned}$$

where

$$H_t(x_t, u_t, \lambda_{t+1}) = \lambda_{t+1}^\top (f_t(x_t, u_t) - x_t) - h_t(x_t, u_t)$$

is the Hamiltonian associated with problem (1).

**Theorem 28** *Assume that $m_t$ defined in (49) is such that $m_t(x_t, \cdot)$ is convex for any $x_t$ and $h_\tau$ is convex. If there exist $x_0^*, \dots, x_\tau^*$ and $\lambda_1^*, \dots, \lambda_\tau^*$ such that $\bar{H}_t(\cdot, \lambda_{t+1}^*)$ is concave and*

$$\lambda_t^* - \lambda_{t+1}^* \in \partial_{x_t} \bar{H}_t(x_t^*, \lambda_{t+1}^*) \quad \text{for } t \in \{1, \dots, \tau - 1\}, \qquad \lambda_\tau^* = \nabla h_\tau(x_\tau^*) \tag{50}$$

$$x_{t+1}^* - x_t^* \in \partial_{\lambda_{t+1}} \bar{H}_t(x_t^*, \lambda_{t+1}^*) \quad \text{for } t \in \{0, \dots, \tau - 1\}, \qquad x_0^* = \bar{x}_0, \tag{51}$$

*then $x_0^*, \dots, x_\tau^*$ is an optimal trajectory for (49). Conditions (50) and (51) amount to the existence of $u_t^* \in \arg\max_{u \in \mathbb{R}^{n_u}} \lambda_{t+1}^\top (f(x_t, u_t) - x_t) - h_t(x_t, u)$, $v_t^* \in \arg\max_{v \in \mathbb{R}^{n_u}} \lambda_{t+1}^\top \psi_t(x_t, v) - h_t(x_t, v)$ such that*

$$\lambda_t^* - \lambda_{t+1}^* = \nabla_{x_t} \psi_t(x_t^*, v_t^*) \lambda_{t+1}^* - \nabla_{x_t} h_t(x_t^*, v_t^*), \qquad x_{t+1}^* - x_t^* = \psi_t(x_t^*, u_t^*).$$

**Proof** Since $m_t(x_t, \cdot)$ is convex for any $x_t$, problem (49) can be rewritten

$$\min_{\substack{x_1, \dots, x_\tau \in \mathbb{R}^{n_x} \\ x_0 = \hat{x}_0}} \sup_{\lambda_1, \dots, \lambda_\tau \in \mathbb{R}^{n_x}} \sum_{t=0}^{\tau-1} \Big( {\lambda_{t+1}}^\top (x_{t+1} - x_t) - \bar{H}_t(x_t, \lambda_{t+1}) \Big) + h_\tau(x_\tau). \tag{52}$$

The above problem can be written as $\min_{\boldsymbol{x} \in \mathbb{R}^{\tau n_x}} \sup_{\boldsymbol{\lambda} \in \mathbb{R}^{\tau n_x}} c(\boldsymbol{x}, \boldsymbol{\lambda})$ with $c(\boldsymbol{x}, \cdot)$ concave for any $\boldsymbol{x}$. The assumptions amount to consider $\boldsymbol{x}^*, \boldsymbol{\lambda}^*$ such that (i) $0 \in \partial_{\boldsymbol{\lambda}^*} c(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$, (ii) $c(\cdot, \boldsymbol{\lambda}^*)$ convex and $0 \in \partial_{\boldsymbol{x}^*} c(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$. Then for any $\boldsymbol{x} \in \mathbb{R}^{\tau n_x}$,

$$\sup_{\boldsymbol{\lambda} \in \mathbb{R}^{\tau n_x}} c(\boldsymbol{x}, \boldsymbol{\lambda}) \geq c(\boldsymbol{x}, \boldsymbol{\lambda}^*) \overset{(ii)}{\geq} c(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) \overset{(i)}{=} \sup_{\boldsymbol{\lambda} \in \mathbb{R}^{\tau n_x}} c(\boldsymbol{x}^*, \boldsymbol{\lambda}).$$

Hence, $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x} \in \mathbb{R}^{\tau n_x}} \sup_{\boldsymbol{\lambda} \in \mathbb{R}^{\tau n_x}} c(\boldsymbol{x}, \boldsymbol{\lambda})$, that is, $x_0^*, \dots, x_\tau^*$ is an optimal trajectory. ■

Theorem 28 provides generic sufficient optimality conditions for problem of the form (1) inspired by the continuous time viewpoint. However, as noted by (Polak, 2011), optimality conditions in continuous time may not be informative for discrete time counterparts. This is illustrated here by the difficulty to verify convexity of $m_t(x_t, \cdot)$ or the concavity of $\bar{H}_t(\cdot, \lambda_{t+1}^*)$.

## C. Generic Convergence Results

In this section, we present the convergence analysis for generic problems of the form (1), recalled below.

$$\min_{\substack{u_0, \dots, u_{\tau-1} \in \mathbb{R}^{n_u} \\ x_0, \dots, x_\tau \in \mathbb{R}^{n_x}}} \sum_{t=0}^{\tau-1} h_t(x_t, u_t) + h_\tau(x_\tau) \tag{53}$$

$$\text{subject to} \quad x_{t+1} = f_t(x_t, u_t) \quad \text{for } t \in \{0, \dots, \tau - 1\}, \qquad x_0 = \bar{x}_0.$$

We decompose the problem in a composition by defining first the control of $\tau$ discrete dynamics below.

**Definition 29** *We define the control of $\tau$ discrete time dynamics $(f_t : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x})_{t=0}^{\tau-1}$ 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 sequence of controls $\boldsymbol{u} = (u_0; \dots; u_{\tau-1}) \in \mathbb{R}^{\tau n_u}$, outputs the corresponding trajectory $x_1, \dots, x_\tau$, i.e.,*

$$\begin{aligned} & f^{[\tau]}(x_0, \boldsymbol{u}) = (x_1; \dots; x_\tau) \\ & s.t. \quad x_{t+1} = f_t(x_t, u_t) \quad \text{for } t \in \{0, \dots, \tau - 1\}. \end{aligned} \tag{54}$$

We consider then the cost of a sequence of control variables $\boldsymbol{u} = (u_0; \dots; u_{\tau-1}) \in \mathbb{R}^{\tau n_u}$ and associated trajectory $\boldsymbol{x} = (x_1; \dots; x_\tau) \in \mathbb{R}^{\tau n_x}$, as

$$h(\boldsymbol{x}, \boldsymbol{u}) = \sum_{t=0}^{\tau-1} h_t(x_t, u_t) + h_\tau(x_\tau).$$

Problem (53) amounts then to a compositional problem of the form

$$\min_{\boldsymbol{u} \in \mathbb{R}^{\tau n_u}} \mathcal{J}(\boldsymbol{u}), \text{ for } \mathcal{J}(\boldsymbol{u}) = h(g(\boldsymbol{u})), \; g(\boldsymbol{u}) = (f^{[\tau]}(\boldsymbol{u}, \bar{x}_0), \boldsymbol{u}) \tag{55}$$

We make the following regular smoothness assumptions.

**Assumption 30** *We consider all costs $h_t$ to be $l_h$ Lipschitz continuous, with $L_h$ Lipschitz continuous gradients and $M_h$ Lipschitz continuous Hessians, then the cost function $h$ is $l_h$ Lipschitz continuous, with $L_h$ Lipschitz continuous gradients and $M_h$ Lipschitz continuous Hessians.*

*We consider the dynamics to be Lipschitz-continuous with Lipschitz-continuous gradients such that the control $f^{[\tau]}$ of these dynamics is $l_{f^{[\tau]}}$ Lipschitz continuous with $L_{f^{[\tau]}}$ Lipschitz continuous gradients as detailed in Lemma 36. The augmented function $g$ is then $l_g = \sqrt{l_{f^{[\tau]}}^2 + 1}$ Lipschitz continuous with $L_g = \sqrt{L_{f^{[\tau]}}^2 + 1}$ Lipschitz continuous gradients.*

**Note:** The notations $h, \mathcal{J}, g$ used in Appendix C for problems of the form (53) pertain only to Appendix C, Lemma 48 and Corollary 49. In particular, the definition of $g$ and its smoothness properties differ here than from the main text (see (24)).

### C.1 Generic Convergence Results for ILQR

As explained in the main text for the problem without control costs (Sec. 4), the ILQR algorithm amounts to linearizing $g$ and taking a quadratic approximation of $h$ such that, provided that the minimum exists,

$$\begin{aligned} \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u}) &= \arg\min_{\boldsymbol{v} \in \mathbb{R}^{\tau n_u}} q_h^{g(\boldsymbol{u})}(\ell_g^{\boldsymbol{u}}(\boldsymbol{v})) + \frac{\nu}{2}\|\boldsymbol{v}\|_2^2 \\ &= -(\nabla g(\boldsymbol{u}) \nabla^2 h(g(\boldsymbol{u})) \nabla g(\boldsymbol{u})^\top + \nu \, \mathrm{I})^{-1} \nabla g(\boldsymbol{u}) \nabla h(g(\boldsymbol{u})), \end{aligned} \tag{56}$$

where $\ell_g^{\boldsymbol{u}}$ and $q_h^{g(\boldsymbol{u})}$ are the linear and quadratic expansions of, respectively, the control in $\tau$ steps around $\boldsymbol{u}$ and the total costs around $g(\boldsymbol{u})$. In other words the ILQR algorithm is a generalized Gauss-Newton algorithm that exploits the compositional structure of the problem. Lemma 31 below presents then the convergence to stationary point of the ILQR algorithm from the lens of a generalized Gauss-Newton algorithm. Lemma 32 presents local convergence guarantees.

**Lemma 31** *Under assumption 30, provided that the regularization $\nu$ satisfies*

$$\nu \geq \max\left\{2l_g^2 L_h, \frac{L_g l_h}{2}\gamma\left(\frac{L_g l_h}{4L_h l_g(1+\beta)}\right)\right\},$$

*for $\gamma(x) = 1 + \sqrt{1+1/x}$ and $\beta = M_h l_g^2/(3L_g L_h)$, the iterations of the ILQR algorithm satisfy*

$$\min_{k\in\{0,\dots,K\}} \|\nabla \mathcal{J}(\boldsymbol{u}^{(k)})\|_2 \leq \sqrt{\frac{2(l_g^2 L_h + \nu)\left(\mathcal{J}(\boldsymbol{u}^{(0)}) - \min_{\boldsymbol{u}\in\mathbb{R}^{\tau n_u}} \mathcal{J}(\boldsymbol{u})\right)}{K+1}},$$

*for $\mathcal{J}, h, g$ defining the objective in (55).*

**Proof** Using Lemma 40 adapted to Assumption 30, we have that for any $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{\tau n_u}$,

$$(h\circ g)(\boldsymbol{u}+\boldsymbol{v}) \leq (h\circ g)(\boldsymbol{u}) + q_h^{g(\boldsymbol{u})} \circ \ell_g^{\boldsymbol{u}}(\boldsymbol{v}) + \frac{a_1 + a_2\|\boldsymbol{v}\|_2}{2}\|\boldsymbol{v}\|_2^2,$$

for $a_1 = L_g l_h, a_2 = M_h l_g^3/3 + L_g L_h l_g$. For $\nu > l_g^2 L_h$ the minimizer in (56) is uniquely defined. The oracle $\boldsymbol{v} = \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$ satisfies then $\|\boldsymbol{v}\|_2 \leq l_g l_h/\nu$. For $\nu \geq a_1(1 + \sqrt{1+4a_2 l_g l_h/a_1^2})/2$, we have $a_1 + a_2 l_g l_h/\nu \leq \nu$. Expanding $a_1, a_2$, the condition $\nu \geq a_1(1+\sqrt{1+4a_2 l_g l_h/a_1^2})/2$ reads

$$\nu \geq \frac{L_g l_h}{2}\gamma\left(\frac{L_g l_h}{4L_h l_g(1+\beta)}\right)$$

for $\gamma(x) = 1 + \sqrt{1+1/x}$ and $\beta = M_h l_g^2/(3L_g L_h)$. Then, for $\boldsymbol{v} = \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$, we have

$$\begin{aligned}
(h\circ g)(\boldsymbol{u}+\boldsymbol{v}) &\leq (h\circ g)(\boldsymbol{u}) + q_h^{g(\boldsymbol{u})} \circ \ell_g^{\boldsymbol{u}}(\boldsymbol{v}) + \frac{\nu}{2}\|\boldsymbol{v}\|_2^2 \\
&= (h\circ g)(\boldsymbol{u}) - \frac{1}{2}\nabla(h\circ g)(\boldsymbol{u})^\top(\nabla g(\boldsymbol{u})\nabla^2 h(g(\boldsymbol{u}))\nabla g(\boldsymbol{u})^\top + \nu\,\mathrm{I})^{-1}\nabla(h\circ g)(\boldsymbol{u}) \\
&\leq (h\circ g)(\boldsymbol{u}) - \frac{1}{2(l_g^2 L_h + \nu)}\|\nabla(h\circ g)(\boldsymbol{u})\|_2^2.
\end{aligned}$$

We have then in terms of the ILQR iterations,

$$\frac{1}{2(l_g^2 L_h + \nu)}\|\nabla(h\circ g)(\boldsymbol{u}^{(k)})\|_2^2 \leq (h\circ g)(\boldsymbol{u}^{(k)}) - (h\circ g)(\boldsymbol{u}^{(k+1)}).$$

Summing over $k = 0, \dots, K-1$ and taking the minimum on the left-hand side gives the result. ■

**Lemma 32** *Consider Assumption 30. Assume in addition that the dynamics are twice differentiable with Lipschitz-continuous Hessian such that $\mathcal{J} = h\circ g$ has $M_{\mathcal{J}}$-Lipschitz continuous Hessians. Consider $\boldsymbol{u}^{(k)}$ to be close to a minimum $\boldsymbol{u}^*$ of $\mathcal{J} = h\circ g$ satisfying $\mu_{\mathcal{J}} = \lambda_{\min}(\nabla^2\mathcal{J}(\boldsymbol{u}^*)) > 0$. If $\|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2 \leq \mu_{\mathcal{J}}/M_{\mathcal{J}}$, and the regularization satisfies*

$$\nu \geq \kappa_{\mathcal{J}} \max\{5L_h l_g^2, 8L_{\mathcal{J}}\},$$

*for $\kappa_{\mathcal{J}} = L_{\mathcal{J}}/\mu_{\mathcal{J}}$, then the iterations of the ILQR algorithm converge linearly to $\boldsymbol{u}^*$ as*

$$\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^*\|_2 \leq \left(1 - \frac{\mu_{\mathcal{J}}}{16\nu}\right)\|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2.$$

**Proof** Denote $R = \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2$ and $H = \int_0^1 \nabla^2 \mathcal{J}(\boldsymbol{u}^* + t(\boldsymbol{u}^{(k)} - \boldsymbol{u}^*))dt$ such that $\nabla \mathcal{J}(\boldsymbol{u}^{(k)}) = \nabla \mathcal{J}(\boldsymbol{u}^{(k)}) - \nabla \mathcal{J}(\boldsymbol{u}^*) = H(\boldsymbol{u}^{(k)} - \boldsymbol{u}^*)$. Note that $(\mu_{\mathcal{J}}/2)\,\mathrm{I} \leq (\mu_{\mathcal{J}} - M_{\mathcal{J}} R/2)\,\mathrm{I} \preceq H \preceq L_{\mathcal{J}}\,\mathrm{I}$, where $L_{\mathcal{J}} \leq L_h l_g^2 + L_g l_h$ is the Lipschitz continuity parameter of $\mathcal{J} = h \circ g$.

Denote $P = \nabla g(\boldsymbol{u}) \nabla^2 h(g(\boldsymbol{u})) \nabla g(\boldsymbol{u})^\top + \nu\,\mathrm{I}$. We have

$$\begin{aligned}
\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^*\|_2^2 &\leq \|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)}\|_2^2 + (\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)})^\top (\boldsymbol{u}^{(k)} - \boldsymbol{u}^*) + \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2^2 \\
&\leq \|P^{-1} H (\boldsymbol{u}^{(k)} - \boldsymbol{u}^*)\|_2^2 - (\boldsymbol{u}^{(k)} - \boldsymbol{u}^*)^\top H P^{-1} (\boldsymbol{u}^{(k)} - \boldsymbol{u}^*) + \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2^2 \\
&\leq (1 + L_{\mathcal{J}}^2 \nu^{-2}) \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2^2 - (\boldsymbol{u}^{(k)} - \boldsymbol{u}^*)^\top H P^{-1} (\boldsymbol{u}^{(k)} - \boldsymbol{u}^*). \qquad (57)
\end{aligned}$$

Denote $C = \nabla g(\boldsymbol{u}) \nabla^2 h(g(\boldsymbol{u})) \nabla g(\boldsymbol{u})^\top$. For $\nu$ such that $\|C\|_2/\nu < 1$, we have

$$P^{-1} = \nu^{-1}(\mathrm{I} + \nu^{-1} C)^{-1} = \nu^{-1}\,\mathrm{I} + \nu^{-1} \sum_{j=1}^{+\infty} \nu^{-j} C^j$$

Denoting $G = \sum_{j=1}^{+\infty} \nu^{-j} C^j$, for $\|C\|_2/\nu \leq (\kappa^{-1}/4)/(1 + \kappa^{-1}/4)$ with $\kappa = L_{\mathcal{J}}/\mu_{\mathcal{J}} \geq 1$, we have $\|G\| \leq \kappa^{-1}/4$. We then have

$$\begin{aligned}
-(\boldsymbol{u}^{(k)} - \boldsymbol{u}^*)^\top H P^{-1} (\boldsymbol{u}^{(k)} - \boldsymbol{u}^*) &= -\nu^{-1} (\boldsymbol{u}^{(k)} - \boldsymbol{u}^*)^\top H (\boldsymbol{u}^{(k)} \\
&\quad - \boldsymbol{u}^*) - \nu^{-1} (\boldsymbol{u}^{(k)} - \boldsymbol{u}^*)^\top G H (\boldsymbol{u}^{(k)} - \boldsymbol{u}^*) \\
&\leq -\frac{\nu^{-1} \mu_{\mathcal{J}}}{2} \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2^2 + \frac{\nu^{-1} L_{\mathcal{J}} \kappa^{-1}}{4} \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2^2 \\
&= -\frac{\nu^{-1} \mu_{\mathcal{J}}}{4} \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2^2.
\end{aligned}$$

Plugging the above equation into (57), we have that if $\nu$ satisfies in addition $\nu \geq 8 L_{\mathcal{J}} \kappa_{\mathcal{J}}$, the iterates of the ILQR algorithm converge linearly to $\boldsymbol{u}^*$ as

$$\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^*\|_2^2 \leq (1 + L_{\mathcal{J}}^2 \nu^{-2} - \nu^{-1} \mu_{\mathcal{J}}/4) \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2^2 \leq \left(1 - \frac{\mu_h}{8\nu}\right) \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2^2.$$

■

### C.2 Generic Convergence Results for IDDP

We analyze the convergence of the IDDP through the lens of the ILQR algorithm. As these two algorithms differ simply by the roll-out procedure, Lemmas 47 and 48, summarized in Corollary 49, show that their oracles differ by at most

$$\|\boldsymbol{w} - \boldsymbol{v}\|_2 \leq \xi \|\boldsymbol{v}\|_2^2$$

for $\xi$ independent of $\nu$ provided that $\nu$ is sufficiently large.

We can then show the convergence of IDDP to stationary points in Lemma 33, as well as its local convergence behavior in Lemma 34

**Lemma 33** *Under Assumption 30, provided that the regularization $\nu$ is larger than $2l_g^2 L_h$ the oracle $\boldsymbol{v} = \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$ returned by the ILQR algorithm and the oracle $\boldsymbol{w} = \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$ returned by the IDDP algorithm differ as*

$$\|\boldsymbol{v} - \boldsymbol{w}\|_2 \leq \xi \|\boldsymbol{v}\|_2^2,$$

*for $\xi$ a constant independent of $\nu$. Moreover, if the regularization $\nu$ satisfies*

$$\nu \geq \max\left\{2l_g^2 L_h, \frac{L_g l_h}{2}\gamma\left(\frac{L_g l_h}{4L_h l_g(1+\beta+\beta')}\right)\right\}$$

*for $\gamma(x) = 1 + \sqrt{1+1/x}$ and $\beta = M_h l_g^2/(3L_g L_h)$, $\beta' = 2l_h\xi/(L_h L_g)$, the iterations of the IDDP algorithm satisfy*

$$\min_{k\in\{0,\dots,K\}} \|\nabla\mathcal{J}(\boldsymbol{u}^{(k)})\|_2 \leq \sqrt{\frac{2(l_g^2 L_h + \nu)\left(\mathcal{J}(\boldsymbol{u}^{(0)}) - \min_{\boldsymbol{u}\in\mathbb{R}^{\tau n_u}}\mathcal{J}(\boldsymbol{u})\right)}{K+1}},$$

*for $\mathcal{J}, h, g$ defining the objective in (55).*

**Proof** To show the convergence of the IDDP, we consider selecting $\nu$ such that

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

for $\boldsymbol{w} = \mathrm{DDP}_\nu(\mathcal{J})(\boldsymbol{u})$ and $\boldsymbol{v} = \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$. Using that costs and dynamics are Lipschitz continuous, we have

$$|\mathcal{J}(\boldsymbol{u}+\boldsymbol{w}) - \mathcal{J}(\boldsymbol{u}+\boldsymbol{v})| \leq l_h l_g \|\boldsymbol{w}-\boldsymbol{v}\|_2.$$

On the other hand, by Corollary 49 for $\nu \geq 2L_h l_g^2$, there exists a constant $\xi$ independent of $\nu$ such that $\|\boldsymbol{w}-\boldsymbol{v}\|_2 \leq \xi\|\boldsymbol{v}\|_2^2$ and so

$$|\mathcal{J}(\boldsymbol{u}+\boldsymbol{w}) - \mathcal{J}(\boldsymbol{u}+\boldsymbol{v})| \leq l_h l_g \xi \|\boldsymbol{v}\|_2^2.$$

Now using Lemma 40 adapted to Assumption 30, we have that for any $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{\tau n_u}$,

$$\mathcal{J}(\boldsymbol{u}+\boldsymbol{v}) \leq \mathcal{J}(\boldsymbol{u}) + q_h^{g(\boldsymbol{u})} \circ \ell_g^{\boldsymbol{u}}(\boldsymbol{v}) + \frac{a_1 + a_2\|\boldsymbol{v}\|_2}{2}\|\boldsymbol{v}\|_2^2,$$

for $a_1 = L_g l_h, a_2 = M_h l_g^3/3 + L_g L_h l_g$. Hence, we have

$$\mathcal{J}(\boldsymbol{u}+\boldsymbol{w}) \leq \mathcal{J}(\boldsymbol{u}) + q_h^{g(\boldsymbol{u})}\ell_g^{\boldsymbol{u}}(\boldsymbol{v}) + \frac{a_1 + a_3\|\boldsymbol{v}\|_2}{2}\|\boldsymbol{v}\|_2^2$$

for $a_3 = a_2 + 2l_h l_g \xi$. Selecting $\nu \geq a_1(1+\sqrt{1+4a_3 l_g l_h/a_1^2})/2$, that is,

$$\nu \geq \frac{L_g l_h}{2}\gamma\left(\frac{L_g l_h}{4L_h l_g(1+\beta+\beta')}\right)$$

for $\gamma, \beta$ defined as in Lemma 31 and $\beta' = 2l_h\xi/(L_h L_g)$ ensures that $a_1 + a_3\|\boldsymbol{v}\|_2 \leq \nu$. So we get that

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

The rest of the proof follows exactly the proof of Lemma 31. ■

**Lemma 34** *Consider Assumption 30. Assume in addition that the dynamics are twice differentiable with Lipschitz-continuous Hessian such that $\mathcal{J} = h \circ g$ has $M_{\mathcal{J}}$-Lipschitz continuous Hessians. Consider $\boldsymbol{u}^{(k)}$ to be close to a minimum $\boldsymbol{u}^*$ of $\mathcal{J} = h \circ g$ satisfying $\mu_{\mathcal{J}} = \lambda_{\min}(\nabla^2 \mathcal{J}(\boldsymbol{u}^*)) > 0$. If $\|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2 \leq \mu_{\mathcal{J}}/M_{\mathcal{J}}$, and the regularization satisfies*

$$\nu \geq \max\{2l_g^2 L_h, 5\kappa_{\mathcal{J}} L_h l_g^2, 8\kappa_{\mathcal{J}} L_{\mathcal{J}}, 32\xi \frac{L_{\mathcal{J}}^2}{M_{\mathcal{J}}}\},$$

*for $\kappa_{\mathcal{J}} = L_{\mathcal{J}}/\mu_{\mathcal{J}}$, then the iterations of the IDDP algorithm converge linearly to $\boldsymbol{u}^*$ as*

$$\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^*\|_2 \leq \left(1 - \frac{\mu_{\mathcal{J}}}{32\nu}\right) \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2.$$

**Proof** Given the $k^{\text{th}}$ iteration $\boldsymbol{u}^{(k)}$ of the IDDP, denote

$$\boldsymbol{u}_{\text{IDDP}}^{(k+1)} = \boldsymbol{u}^{(k)} + \text{DDP}(\mathcal{J})(\boldsymbol{u}^{(k)}), \quad \boldsymbol{u}_{\text{ILQR}}^{(k+1)} = \boldsymbol{u}^{(k)} + \text{LQR}(\mathcal{J})(\boldsymbol{u}^{(k)}),$$

the next iteration if the LQR or the DDP oracles are used respectively. We have using Lemma 31 and Lemma 47,

$$\begin{aligned}\|\boldsymbol{u}_{\text{IDDP}}^{(k+1)} - \boldsymbol{u}^*\|_2 &\leq \|\boldsymbol{u}_{\text{ILQR}}^{(k+1)} - \boldsymbol{u}^*\|_2 + \xi \|\text{LQR}(\mathcal{J})(\boldsymbol{u}^{(k)})\|_2^2 \\ &\leq \left(1 - \frac{\mu_{\mathcal{J}}}{16\nu} + \xi \frac{L_{\mathcal{J}}^2}{\nu^2} \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2\right) \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2.\end{aligned}$$

The result follows by using that $\|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2 \leq \mu_{\mathcal{J}}/M_{\mathcal{J}}$ and taking

$$\nu \geq \frac{32\xi L_{\mathcal{J}}}{M_{\mathcal{J}}}.$$

■

## D. Conditioning Analysis

### D.1 Smoothness Estimations

To derive simple bounds on the Lipschitz-continuity constants of the trajectory function $f^{[\tau]}$, we present first a compact formulation of the first and second order information of $f^{[\tau]}$ with respect to the first and second order information of the dynamics $(f_t)_{t=0}^{\tau-1}$ in Lemma 35. We require the following tensor notations in this subsection.

A tensor $\mathcal{A} = (a_{i,j,k})_{1 \leq i \leq d, 1 \leq j \leq p, 1 \leq k \leq n} \in \mathbb{R}^{d \times p \times n}$ is represented as a list of matrices $\mathcal{A} = (A_1, \dots, A_n)$ where $A_k = (a_{i,j,k})_{1 \leq i \leq d, 1 \leq j \leq p} \in \mathbb{R}^{d \times p}$ for $k \in \{1, \dots n\}$. Given $\mathcal{A} \in \mathbb{R}^{d \times p \times n}$ and $P \in \mathbb{R}^{d \times d'}, Q \in \mathbb{R}^{p \times p'}, R \in \mathbb{R}^{n \times n'}$, we denote

$$\mathcal{A}[P, Q, R] = \left(\sum_{k=1}^n R_{k,1} P^\top A_k Q, \dots, \sum_{k=1}^n R_{k,n'} P^\top A_k Q\right) \in \mathbb{R}^{d' \times p' \times n'}.$$

For $\mathcal{A}_0 \in \mathbb{R}^{d_0 \times p_0 \times n_0}$, $P \in \mathbb{R}^{d_0 \times d_1}, Q \in \mathbb{R}^{p_0 \times p_1}, R \in \mathbb{R}^{n_0 \times n_1}$ denote $\mathcal{A}_1 = \mathcal{A}_0[P, Q, R] \in \mathbb{R}^{d_1 \times p_1 \times n_1}$. Then, for $S \in \mathbb{R}^{d_1 \times d_2}, T \in \mathbb{R}^{p_1 \times p_2}, U \in \mathbb{R}^{n_1 \times n_2}$, we have

$$\mathcal{A}_1[S, T, U] = \mathcal{A}_0[PS, QT, RU] \in \mathbb{R}^{d_2 \times p_2 \times n_2}.$$

If $P, Q$ or $R$ are identity matrices, we use the symbol " $\cdot$ " in place of the identity matrix. For example, we denote $\mathcal{A}[P, Q, \mathrm{I}_n] = \mathcal{A}[P, Q, \cdot] = \left(P^\top A_1 Q, \dots, P^\top A_n Q\right)$. If $P, Q$ or $R$ are vectors we consider the flattened object. In particular, for $x \in \mathbb{R}^d, y \in \mathbb{R}^p$, we denote

$$\mathcal{A}[x, y, \cdot] = \left(x^\top A_1 y, \dots, x^\top A_n y\right)^\top \in \mathbb{R}^n,$$

rather than having $\mathcal{A}[x, y, \cdot] \in \mathbb{R}^{1 \times 1 \times n}$. Similarly, for $z \in \mathbb{R}^n$, we denote

$$\mathcal{A}[\cdot, \cdot, z] = \sum_{k=1}^{n} z_k A_k \in \mathbb{R}^{d \times p}.$$

We denote $\|a\|_2$ the Euclidean norm for $a \in \mathbb{R}^d$, $\|A\|_{2,2}$ the spectral norm of a matrix $A \in \mathbb{R}^{d \times p}$, and we define the norm of a tensor $\mathcal{A}$ induced by the Euclidean norm as $\|\mathcal{A}\|_{2,2,2} = \sup_{x \neq 0, y \neq 0, z \neq 0} \mathcal{A}[x, y, z] / (\|x\|_2 \|y\|_2 \|z\|_2)$.

**Lemma 35** *Consider the control $f^{[\tau]}$ of $\tau$ dynamics $(f_t)_{t=0}^{\tau-1}$ as defined in Def. 29 and an initial point $x_0 \in \mathbb{R}^{n_x}$. For $\boldsymbol{x} = (x_1; \dots; x_\tau)$ and $\boldsymbol{u} = (u_0; \dots; u_{\tau-1})$, define*

$$F(\boldsymbol{x}, \boldsymbol{u}) = (f_0(x_0, u_0); \dots; f_{\tau-1}(x_{\tau-1}, u_{\tau-1})),$$

*such that $\boldsymbol{x} = f^{[\tau]}(x_0, \boldsymbol{u})$ is the unique solution of the implicit equation $\boldsymbol{x} = F(\boldsymbol{x}, \boldsymbol{u})$. The gradient of the control $f^{[\tau]}$ of the dynamics $(f_t)_{t=0}^{\tau-1}$ on $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$ can be written*

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

*The Hessian of the control $f^{[\tau]}$ of the dynamics $(f_t)_{t=0}^{\tau-1}$ on $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$ can be written*

$$\begin{aligned} \nabla^2_{\boldsymbol{u}\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u}) = {} & \nabla^2_{\boldsymbol{x}\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u})[N, N, M] + \nabla^2_{\boldsymbol{u}\boldsymbol{u}} F(\boldsymbol{x}, \boldsymbol{u})[\cdot, \cdot, M] \\ & + \nabla^2_{\boldsymbol{x}\boldsymbol{u}} F(\boldsymbol{x}, \boldsymbol{u})[N, \cdot, M] + \nabla^2_{\boldsymbol{u}\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u})[\cdot, N, M], \end{aligned}$$

*where $M = (\mathrm{I} - \nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u}))^{-1}$ and $N = \nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})^\top$.*

**Proof** Denote simply, for $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$, $\varphi(\boldsymbol{u}) = f^{[\tau]}(x_0, \boldsymbol{u})$ with $x_0$ a fixed initial state. By definition, the function $\varphi$ can be decomposed, for $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$, as $\varphi(\boldsymbol{u}) = (\varphi_1(\boldsymbol{u}); \dots; \varphi_\tau(\boldsymbol{u}))$, such that

$$\varphi_{t+1}(\boldsymbol{u}) = f_t(\varphi_t(\boldsymbol{u}), E_t^\top \boldsymbol{u}) \quad \text{for } t \in \{0, \dots, \tau - 1\}, \tag{58}$$

with $\varphi_0(\boldsymbol{u}) = x_0$ and for $t \in \{0, \dots, \tau - 1\}$, $E_t = e_t \otimes \mathrm{I}_{n_u}$ is such that $E_t^\top \boldsymbol{u} = u_t$, with $e_t$ the $t + 1^{\text{th}}$ canonical vector in $\mathbb{R}^\tau$, $\otimes$ the Kronecker product and $\mathrm{I}_{n_u} \in \mathbb{R}^{n_u \times n_u}$ the identity matrix. By taking the derivative of (58), we get, denoting $x_t = \varphi_t(\boldsymbol{u})$ for $t \in \{0, \dots, \tau\}$ and using that $E_t^\top \boldsymbol{u} = u_t$,

$$\nabla \varphi_{t+1}(\boldsymbol{u}) = \nabla \varphi_t(\boldsymbol{u}) \nabla_{x_t} f_t(x_t, u_t) + E_t \nabla_{u_t} f_t(x_t, u_t) \quad \text{for } t \in \{0, \dots, \tau - 1\}.$$

So, for $\boldsymbol{v} = (v_0; \dots; v_{\tau-1}) \in \mathbb{R}^{\tau n_u}$, denoting $\nabla\varphi(\boldsymbol{u})^\top \boldsymbol{v} = (y_1; \dots; y_\tau)$ s.t. $\nabla\varphi_t(\boldsymbol{u})^\top \boldsymbol{v} = y_t$ for $t \in \{1, \dots, \tau\}$, we have, with $y_0 = 0$,

$$y_{t+1} = \nabla_{x_t} f_t(x_t, u_t)^\top y_t + \nabla_{u_t} f_t(x_t, u_t)^\top v_t \quad \text{for } t \in \{0, \dots, \tau-1\}. \tag{59}$$

Denoting $\boldsymbol{y} = (y_1; \dots; y_\tau)$, we have then

$$(\mathrm{I} - A)\boldsymbol{y} = B\boldsymbol{v}, \quad \text{i.e.,} \quad \nabla\varphi(\boldsymbol{u})^\top \boldsymbol{v} = (\mathrm{I} - A)^{-1} B\boldsymbol{v},$$

where $A = \sum_{t=1}^{\tau-1} e_t e_{t+1}^\top \otimes A_t$ with $A_t = \nabla_{x_t} f_t(x_t, u_t)^\top$ for $t \in \{1, \dots, \tau-1\}$ and $B = \sum_{t=1}^{\tau} e_t e_t^\top \otimes B_{t-1}$ with $B_t = \nabla_{u_t} f_t(x_t, u_t)^\top$ for $t \in \{0, \dots, \tau-1\}$, i.e.

$$A = \begin{pmatrix} 0 & \dots & & \dots & 0 \\ A_1 & \ddots & & & \vdots \\ 0 & \ddots & & & \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & A_{\tau-1} & 0 \end{pmatrix}, \quad B = \begin{pmatrix} B_0 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & B_{\tau-1} \end{pmatrix}.$$

By definition of $F$ in the claim, one easily check that $A = \nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u})^\top$ and $B = \nabla_{\boldsymbol{u}} F(\boldsymbol{x}, \boldsymbol{u})^\top$. Therefore, we get

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

For the Hessian, note that for $g : \mathbb{R}^d \to \mathbb{R}^p$, $f : \mathbb{R}^p \to \mathbb{R}$, $x \in \mathbb{R}^d$, we have $\nabla^2(f \circ g)(x) = \nabla g(x) \nabla^2 f(x) \nabla g(x)^\top + \nabla^2 g(x)[\cdot, \cdot, \nabla f(x)] \in \mathbb{R}^{d \times d}$. If $f : \mathbb{R}^p \to \mathbb{R}^n$, we have $\nabla^2(f \circ g)(x) = \nabla^2 f(x)[\nabla g(x)^\top, \nabla g(x)^\top, \cdot] + \nabla^2 g(x)[\cdot, \cdot, \nabla f(x)] \in \mathbb{R}^{d \times d \times n}$. Applying this on $f_t \circ g_t$ for $g_t(\boldsymbol{u}) = (\varphi_t(\boldsymbol{u}), E_t^\top \boldsymbol{u})$, we get from (58), using that $\nabla g_t(\boldsymbol{u}) = (\nabla\varphi_t(\boldsymbol{u}), E_t)$,

$$\begin{aligned} \nabla^2\varphi_{t+1}(\boldsymbol{u}) = {} & \nabla^2\varphi_t(\boldsymbol{u})[\cdot, \cdot, \nabla_{x_t} f_t(x_t, u_t)] \\ & + \nabla^2_{x_t x_t} f_t(x_t, u_t)[\nabla\varphi_t(\boldsymbol{u})^\top, \nabla\varphi_t(\boldsymbol{u})^\top, \cdot] + \nabla^2_{u_t u_t} f_t(x_t, u_t)[E_t^\top, E_t^\top, \cdot] \\ & + \nabla^2_{x_t u_t} f_t(x_t, u_t)[\nabla\varphi_t(\boldsymbol{u})^\top, E_t^\top, \cdot] + \nabla^2_{u_t x_t} f_t(x_t, u_t)[E_t^\top, \nabla\varphi_t(\boldsymbol{u})^\top, \cdot], \end{aligned}$$

for $t \in \{0, \dots, \tau-1\}$, with $\nabla^2\varphi_0(\boldsymbol{u}) = 0$. Therefore, for $\boldsymbol{v} = (v_0; \dots; v_{\tau-1}), \boldsymbol{w} = (w_0; \dots; w_{\tau-1}) \in \mathbb{R}^{\tau n_u}$, $\boldsymbol{\mu} = (\mu_1; \dots; \mu_\tau) \in \mathbb{R}^{\tau n_x}$, we get

$$\begin{aligned} \nabla^2\varphi(\boldsymbol{u})[\boldsymbol{v}, \boldsymbol{w}, \boldsymbol{\mu}] &= \sum_{t=0}^{\tau-1} \nabla^2\varphi_{t+1}(\boldsymbol{u})[\boldsymbol{v}, \boldsymbol{w}, \mu_{t+1}] \\ &= \sum_{t=0}^{\tau-1} \Big( \nabla^2_{x_t x_t} f_t(x_t, u_t)[y_t, z_t, \lambda_{t+1}] + \nabla^2_{u_t u_t} f_t(x_t, u_t)[v_t, w_t, \lambda_{t+1}] \\ &\qquad + \nabla^2_{x_t u_t} f_t(x_t, u_t)[y_t, w_t, \lambda_{t+1}] + \nabla^2_{u_t x_t} f_t(x_t, u_t)[v_t, z_t, \lambda_{t+1}] \Big), \end{aligned} \tag{60}$$

where $\boldsymbol{y} = (y_1; \dots; y_\tau) = \nabla\varphi(\boldsymbol{u})^\top \boldsymbol{v}$, $\boldsymbol{z} = (z_1; \dots; z_\tau) = \nabla\varphi(\boldsymbol{u})^\top \boldsymbol{w}$, with $y_0 = z_0 = 0$ and $\boldsymbol{\lambda} = (\lambda_1; \dots; \lambda_\tau) \in \mathbb{R}^{\tau n_x}$ is defined by

$$\lambda_t = \nabla_{x_t} f_t(x_t, u_t)\lambda_{t+1} + \mu_t \qquad \text{for } t \in \{1, \dots, \tau-1\}, \quad \lambda_\tau = \mu_\tau.$$

On the other hand, denoting $F_t(\boldsymbol{x}, \boldsymbol{u}) = f_t(x_t, u_t)$ for $t \in \{0, \dots, \tau - 1\}$, the Hessian of $F$ with respect to the variables $\boldsymbol{u}$ can be decomposed as

$$\nabla^2_{\boldsymbol{uu}} F(\boldsymbol{x}, \boldsymbol{u})[\boldsymbol{v}, \boldsymbol{w}, \boldsymbol{\lambda}] = \sum_{t=0}^{\tau-1} \nabla^2_{\boldsymbol{uu}} F_t(\boldsymbol{x}, \boldsymbol{u})[\boldsymbol{v}, \boldsymbol{w}, \lambda_{t+1}] = \sum_{t=0}^{\tau-1} \nabla^2_{u_t u_t} f_t(x_t, u_t)[v_t, w_t, \lambda_{t+1}].$$

The Hessian of $F$ with respect to the variable $\boldsymbol{x}$ can be decomposed as

$$\nabla^2_{\boldsymbol{xx}} F(\boldsymbol{x}, \boldsymbol{u})[\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\lambda}] = \sum_{t=0}^{\tau-1} \nabla^2_{\boldsymbol{xx}} F_t(\boldsymbol{x}, \boldsymbol{u})[\boldsymbol{y}, \boldsymbol{z}, \lambda_{t+1}] = \sum_{t=1}^{\tau-1} \nabla^2_{x_t x_t} f_t(x_t, u_t)[y_t, z_t, \lambda_{t+1}].$$

A similar decomposition can be done for $\nabla^2_{\boldsymbol{xu}} F(\boldsymbol{x}, \boldsymbol{u})$. From (60), we then get

$$\begin{aligned}\nabla^2 \varphi(\boldsymbol{u})[\boldsymbol{v}, \boldsymbol{w}, \boldsymbol{\mu}] = \nabla^2_{\boldsymbol{xx}} F(\boldsymbol{x}, \boldsymbol{u})[\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\lambda}] + \nabla^2_{\boldsymbol{uu}} F(\boldsymbol{x}, \boldsymbol{u})[\boldsymbol{v}, \boldsymbol{w}, \boldsymbol{\lambda}] \\ + \nabla^2_{\boldsymbol{xu}} F(\boldsymbol{x}, \boldsymbol{u})[\boldsymbol{y}, \boldsymbol{w}, \boldsymbol{\lambda}] + \nabla^2_{\boldsymbol{ux}} F(\boldsymbol{x}, \boldsymbol{u})[\boldsymbol{v}, \boldsymbol{z}, \boldsymbol{\lambda}].\end{aligned}$$

Finally, by noting that

$$\begin{aligned}\boldsymbol{y} &= (\nabla_{\boldsymbol{u}} F(\boldsymbol{x}, \boldsymbol{u})(\mathrm{I} - \nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u}))^{-1})^\top \boldsymbol{v}, \\ \boldsymbol{z} &= (\nabla_{\boldsymbol{u}} F(\boldsymbol{x}, \boldsymbol{u})(\mathrm{I} - \nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u}))^{-1})^\top \boldsymbol{w} \\ \boldsymbol{\lambda} &= (\mathrm{I} - \nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u}))^{-1} \boldsymbol{\mu}\end{aligned}$$

the claim is shown. ■

Lemma 35 can be used to get estimates on the smoothness properties of the control of $\tau$ dynamics given the smoothness properties of each individual dynamics.

**Lemma 36** *If $\tau$ dynamics $(f_t)_{t=0}^{\tau-1}$ are Lipschitz continuous with Lipschitz continuous gradients, then the function $\boldsymbol{u} \to f^{[\tau]}(x_0, \boldsymbol{u})$, with $f^{[\tau]}$ the control of the $\tau$ dynamics $(f_t)_{t=0}^{\tau-1}$, is $l_{f^{[\tau]}}$-Lipschitz continuous and has $L_{f^{[\tau]}}$-Lipschitz continuous gradients with*

$$l_{f^{[\tau]}} \leq l^u_f S, \qquad L_{f^{[\tau]}} \leq S(L^{xx}_f l^2_{f^{[\tau]}} + 2L^{xu}_f l_{f^{[\tau]}} + L^{uu}_f), \tag{61}$$

*where $l^u_{f_t} = \sup_{x,u} \|\nabla_u f(x,u)\|_{2,2}$, $l^x_{f_t} = \sup_{x,u} \|\nabla_x f(x,u)\|_{2,2}$, $L^{xx}_{f_t} = \sup_{x,u} \|\nabla^2_{xx} f(x,u)\|_{2,2,2}$, $L^{uu}_{f_t} = \sup_{x,u} \|\nabla^2_{uu} f(x,u)\|_{2,2,2}$, $L^{xu}_{f_t} = \sup_{x,u} \|\nabla^2_{xu} f(x,u)\|_{2,2,2}$, $S = \sum_{t=0}^{\tau-1} (l^x_f)^t$, and we drop the index $t$ to denote the maximum over all dynamics such as $l^x_f = \max_{t \in \{0,\dots,\tau-1\}} l^x_{f_t}$.*

**Proof** The Lipschitz continuity constant of $\boldsymbol{u} \to f^{[\tau]}(x_0, \boldsymbol{u})$ and its gradients can be estimated by upper bounding the norm of the gradients and the Hessians. With the notations of Lemma 35, $\nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u})$ is nilpotent of degree $\tau$ since it can be written $\nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u}) = \sum_{t=1}^{\tau-1} e_{t+1} e_t^\top \otimes \nabla_{x_t} f_t(x_t, u_t)$ and $(A \otimes B)(C \otimes D) = (AC \otimes BD)$. Hence, we have

$$(\mathrm{I} - \nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u}))^{-1} = \sum_{t=0}^{\tau-1} \nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u})^t.$$

The Lipschitz continuity constant of $f^{[\tau]}$ is then estimated by

$$\|\nabla_{\boldsymbol{u}} f^{[\tau]}(x_0, \boldsymbol{u})\|_{2,2} \leq \|\nabla_{\boldsymbol{u}} F(\boldsymbol{x}, \boldsymbol{u})\|_{2,2} \|(\mathrm{I} - \nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u}))^{-1}\|_{2,2} \leq l_f^u \sum_{t=0}^{\tau-1} (l_f^x)^t.$$

As shown in Lemma 35, the Hessian of $\boldsymbol{u} \to f^{[\tau]}(x_0, \boldsymbol{u})$ can be decomposed as

$$\begin{aligned}\nabla^2_{\boldsymbol{uu}} f^{[\tau]}(x_0, \boldsymbol{u}) = \nabla^2_{\boldsymbol{xx}} F(\boldsymbol{x}, \boldsymbol{u})[N, N, M] + \nabla^2_{\boldsymbol{uu}} F(\boldsymbol{x}, \boldsymbol{u})[\cdot, \cdot, M] \\ + \nabla^2_{\boldsymbol{xu}} F(\boldsymbol{x}, \boldsymbol{u})[N, \cdot, M] + \nabla^2_{\boldsymbol{ux}} F(\boldsymbol{x}, \boldsymbol{u})[\cdot, N, M],\end{aligned}$$

where $M = (\mathrm{I} - \nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u}))^{-1}$ and $N = \nabla_{\boldsymbol{u}} f^{[\tau]}(\bar{x}_0, \boldsymbol{u})^\top$. Given the structure of $F$, bounds on the Hessians are $\|\nabla^2_{ab} F(\boldsymbol{x}, \boldsymbol{u})\|_{2,2,2} \leq L_f^{ab}$ for $a, b \in \{\boldsymbol{x}, \boldsymbol{u}\}$, where $\|\mathcal{A}\|_{2,2,2}$ is the norm of a tensor $\mathcal{A}$ w.r.t. the Euclidean norm as defined in the notations. Note that for a given tensor $\mathcal{A} \in \mathbb{R}^{d \times p \times n}$ and $P, Q, R$ of appropriate sizes, we have $\|\mathcal{A}[P, Q, R]\|_{2,2,2} \leq \|\mathcal{A}\|_{2,2,2} \|P\|_{2,2} \|Q\|_{2,2} \|R\|_{2,2}$. We then get

$$||\nabla^2_{\boldsymbol{uu}} f^{[\tau]}(x_0, \boldsymbol{u})||_{2,2,2} \leq L_f^{xx} \|N\|_{2,2}^2 \|M\|_{2,2} + L_f^{uu} \|M\|_{2,2} + 2 L_f^{xu} \|M\|_{2,2} \|N\|_{2,2},$$

where for twice differentiable functions we used that $L_f^{xu} = L_f^{ux}$. ■

### D.2 Sufficient Condition for Global Convergence of Time-varying Dynamics

Lemma 37 presents a simple extension of Lemma 6 for time-varying dynamics. Note that provided that condition (62) is satisfied, the analysis of the ILQR and IDDP algorithms remain essentially unchanged, up to different constants.

**Lemma 37** *Consider the control of $\tau$ discrete time dynamics $(f_t : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x})_{t=0}^{\tau-1}$ as defined in Definition 29. If the dynamics $f_t$ are Lipschitz continuous and satisfy*

$$\forall x, u \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}, \quad \sigma_{\min}(\nabla_u f_t(x, u)) \geq \sigma_{f_t} > 0, \tag{62}$$

*then the control $f^{[\tau]}$ of these dynamics satisfy for all $t \in \{0, \ldots, \tau - 1\}$,*

$$\forall x_0, \boldsymbol{u} \in \mathbb{R}^{n_x} \times \mathbb{R}^{\tau n_u}, \quad \sigma_{\min}(\nabla_u f^{[\tau]}(x_0, \boldsymbol{u})) \geq \sigma_{f^{[\tau]}} := \frac{\min_{t \in \{0, \ldots, \tau-1\}} \sigma_{f_t}}{1 + \max_{t \in \{0, \ldots, \tau-1\}} l_{f_t}^x} > 0,$$

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

**Proof** With the notations of the proof of Lemma 6 we have

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

where here $F(\boldsymbol{x}, \boldsymbol{u}) = (f_0(x_0, u_0); \ldots; f_{\tau-1}(x_{\tau-1}, u_{\tau-1}))$. Noting that $\sigma_{\min}(\nabla_{\boldsymbol{u}} F(\boldsymbol{x}, \boldsymbol{u})) \geq \min_{t \in \{0, \ldots, \tau-1\}} \sigma_{f_t}$ and $\sigma_{\max}(\mathrm{I} - \nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u})) \leq 1 + \max_{t \in \{0, \ldots, \tau-1\}} l_{f_t}^x$ concludes the proof. ■

## E. Linearization by Static Feedback and Brunovsky's Form

We briefly recall here the rationale behind the parameterization of a system in Brunovsky's form and the theory underlying the existence of static feedback linearization in continuous time, see, e.g. Isidori (1995); Busawon and Djemaï (2009) for more details on these subjects and Aranda-Bricaire et al. (1996) for an analysis of feedback linearization in discrete time.

### E.1 Brunovsky's Form

We start by understanding the relevance of the parameterization in Brunovsky's form for discrete time linear systems of the form

$$x_{t+1} = Ax_t + Bu_t \quad \text{for } t = 0, 1, \dots \tag{63}$$

for $x_t \in \mathbb{R}^{n_x}$, $u_t \in \mathbb{R}^{n_u}$ with $n_u = 1$ for ease of presentation, where $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times 1}$ and we denote $n_x = n$ for more readability. An important property that can be investigated for such system is its controllability, i.e., whether, from any initial state $x_0$, we can reach any state $x^*$ after a sufficient number of steps of the discrete dynamical system and appropriate control variables. This question can be answered by examining the controllability matrix $C = [B, AB, \dots A^{n-1}B]$ associated to the system (63). If $C$ has full row rank, i.e., $\text{rank}(C) = n$, then the system is controllable in at most $n$ steps, as observed from standard linear algebra considerations. For a controllable system (63), we can investigate whether there exists a reparameterization of the system in variables $y_t = Mx_t, v_t = Nu_t + Py_t$, in which the notion of controllability is transparent in the reparameterized system $y_{t+1} = Fy_t + Gv_t$, for $F, G$ defined appropriately from $A, B, M, N, P$. One ideal reparameterization is given by Brunovsky's form,

$$\begin{aligned} y_{t+1}^{(1)} &= y_t^{(2)} \\ y_{t+1}^{(2)} &= y_t^{(3)} \\ &\vdots \qquad \text{that is} \quad y_{t+1} = Dy_t + Ev_t \\ y_{t+1}^{(n-1)} &= y_t^{(n)} \\ y_{t+1}^{(n)} &= v_t \end{aligned}$$

for $D = \sum_{i=1}^{n-1} e_i e_{i+1}^\top$ the upper-shift matrix in $\mathbb{R}^n$ and $e_i$ the $i^{\text{th}}$ canonical vector in $\mathbb{R}^n$ with $E = e_n$. In this reparameterized system of equations, after $n$ steps of the linear system we naturally have that $y_n^{(i)} = v_{i-1}$ for $i \in \{1, \dots, n\}$, that is, the operator that, at $n$ control variables associates the state of the system after $n$ steps is just the identity operator, which clearly satisfies the definition of controllability.

To get such a parameterization, consider state variables of the form $y_t = Mx_t$ for $M$ invertible. The resulting linear system has the form $y_{t+1} = MAM^{-1}y_t + MBu_t$. We then need to choose $M$ such that $MAM^{-1} = D + EJ$ for some $J \in \mathbb{R}^{1 \times n}$ and $FB = E$ such that by defining $v_t = EJy_t + Eu_t$, we get that $y_{t+1} = Dy_t + Ev_t$. Such invertible matrix $M$ can be computed in closed form from the expressions of $A$, $B$, $D$ and $E$ as $CK^{-1}$ for $C$ the controllability matrix associated to the pair $(A, B)$ defining the linear system (63) and $K$ the controllability matrix associated to the pair $D, E$ defining the linear system in Brunovsky's form. This is essentially the approach taken in Lemma 38.

### E.2 Static Feedback Linearization for Continuous Time Systems.

Consider a continuous dynamical system of the form

$$\dot{x} = f(x) + g(x)u \tag{64}$$

for $f : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ and $g : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ with $u \in \mathbb{R}$, and we denote here $n_x = n$ for more readability. Static feedback linearization schemes aim to find a reparameterization of this system around an initial state $x_0$ such that the system is linear and controllable in the reparameterized variables under suitable assumptions on $f$ and $g$. The point of departure of the analysis of conditions for the existence of a static feedback linearization scheme (Isidori, 1995, Section 4) is to consider a function $h$ which defines the output of the system (64) as $z = h(x)$. We may then analyze the influence of the control on this output through the derivatives of $z$. Namely, we have that $\frac{\partial z}{\partial t} = \frac{\partial h}{\partial x}\frac{\partial x}{\partial t} = \frac{\partial h}{\partial x}(f(x) + g(x)u) = L_f h(x) + L_g h(x)u$, where we defined the derivative of $h : \mathbb{R}^n \to \mathbb{R}$ along $f : \mathbb{R}^n \to \mathbb{R}^n$ as $L_f h(x) = \sum_{i=1}^n \frac{\partial h}{\partial x_i}(x) f_i(x)$ for $f_i(x)$ the $i^{\text{th}}$ coordinate of $f(x)$. If $L_g h(x) = 0$ in a neighborhood of the initial state $x_0$, i.e., the derivative of the output function $h$ along $g$ is zero, then the control has no effect on the first derivative of the output for $t$ small enough, i.e., $\frac{\partial z}{\partial t}(t) = L_f h(x(t))$. Analyzing the second derivative of the output, we have $\frac{\partial^2 z}{\partial t^2} = L_f^2 h(x) + L_g L_f h(x)u$, where $L_g L_f h(x) = \sum_{i=1}^n \frac{\partial L_f h}{\partial x_i}(x) g_i(x)$ and $L_f^2 h(x) = L_f L_f h(x)$. If $L_g L_f h(x) = 0$ in a neighborhood of the initial state, then the control variable has no effect on the second derivative of the output for $t$ small enough. Continuing this way, if for any $k \in \{0, \dots, n-1\}$, $L_g L_f^k h(x) = 0$ around $x_0$ and $L_g L_f^n h(x_0) \neq 0$ then the derivatives of the output satisfy $\frac{\partial^k z}{\partial t^k} = L_f^k h(t)$ for $t$ small enough and $\frac{\partial^n z}{\partial t^n}(0) = L_f^n h(x_0) + L_g L_f^{n-1} h(x_0) u(0)$. In other words, under the aforementioned conditions, the output of the system can be seen as a dynamical system driven by its $n^{\text{th}}$ derivative. Given an output function $h$ satisfying the aforementioned conditions, we can then consider the reparameterization $y_i = L_f^{i-1} h(x)$ (which corresponds to consider a system whose coordinates are defined by the $i^{\text{th}}$ derivative of the output) and define $v = L_f^n h(x) + L_g L_f^{n-1} h(x) u$ such that the reparameterized system takes the form

$$\begin{aligned}
\dot{y}_1 &= y_2 \\
\dot{y}_2 &= y_3 \\
&\vdots \\
\dot{y}_{n-1} &= y_n \\
\dot{y}_n &= L_f^n h(x) + L_g L_f^{n-1} h(x) u = v.
\end{aligned}$$

We recognize here again a parameterization in Brunovsky's form, here for the continuous time system considered. Existence of an output function satisfying the aforementioned assumptions and such that the reparameterization is a diffeomorphism around the initial point can be verified by considering the involutivity and regularity of the vector field defined by repeated Lie brackets of the function $f$ on the function $g$, see, e.g. Isidori (1995, Lemma 4.2.2).

### E.3 Reparameterization in Brunovsky Form

Lemma 38 shows how a discrete time system driven by its $k^{\text{th}}$ derivative can be expressed in Brunovsky's form (19) (Brunovskỳ, 1970).

**Lemma 38** *Consider the Euler discretization of a single-input continuous-time system driven by its $n_x{}^{th}$ derivative as presented in (20). If $|\partial_v \psi(y,v)| > 0$ for all $y \in \mathbb{R}^{n_x}, v \in \mathbb{R}$ then the dynamical system (20) can be linearized by static feedback into the canonical form (19).*

**Proof** Denoting $A = \mathrm{I} + \Delta D$, with $D$ the upper-shift matrix in $\mathbb{R}^{n_x}$, the original dynamical system (20) 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 $w_t = b(y_t, v_t) = c^\top Q y_t + \Delta\psi(y_t, v_t)$ we get the desired form (19). 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 $y \in \mathbb{R}^{n_x}, v \in \mathbb{R}$. ■

## F. Convergence Analysis of ILQR

### F.1 Global Convergence Analysis

In the statement of Theorem 13, we used Lemma 39 to relate the constants associated to gradient dominance properties of the costs on the sates to the constant associated to the gradient dominance property of the cost on the trajectory with, in Theorem 13, compared to Lemma 39, we used $\mu_t = \mu$ for all $t$ such that $\mu_h = \|\boldsymbol{\mu}^{-1}\|_q^{-1} = \mu\tau^{-q} = \mu/\tau^{2r/(2r-1)}$.

**Lemma 39** *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 $\delta$ 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 $x_t$, $h^* = \sum_{t=1}^{\tau} h_t^*$. ■

Lemma 40 states that a linear quadratic approximation of the compositional objective in (24) approximates the objective up to a cubic error.

**Lemma 40** *Given Assumption 11, we have, for problem (24),*

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

**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,

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

■

### F.2 Local Convergence Analysis

Lemma 41 provides a bound on the oracle returned by an ILQR method in terms of the constants introduced in Theorem 18.

**Lemma 41** *Given Assumption 17 on problem (24), 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

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

Recall that by definition of $\sigma$ 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 42 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 42** *For a $\vartheta_h$-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 $\|\nabla h(y) - \nabla h(x)\|_x^* = \|J(y - x)\|_x^* = \|H^{-1/2} J H^{-1/2}\|_2 \|y - x\|_x$. Now $H^{-1/2} J H^{-1/2} \succeq 0$ 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 $\|H^{-1/2} J H^{-1/2}\|_2 \leq 1/(1 - \vartheta_h \|y - x\|_x)$. ■

### F.3 Global Complexity Bound

Lemma 43 refines the regularization choice of Theorem 13 by exploiting an additional assumption of strong convexity of the costs.

**Lemma 43** *Consider $h$ to be $\mu_h$-strongly convex and Assumption 11 to be satisfied. Condition (27) is satisfied by choosing a regularization*

$$\nu \geq \nu(\boldsymbol{u}) = \left(1 + \frac{\alpha}{2(1 + \theta_g \|\nabla h(g(\boldsymbol{u}))\|_2 / (\sqrt{\mu_h} \rho_g))}\right) L_g \|\nabla h(g(\boldsymbol{u}))\|_2,$$

*for $\rho_h = L_h/\mu_h$, $\rho_g = l_g/\sigma_g$, $\theta_g = L_g/(\sigma_g^2 \sqrt{\mu_h})$, $\alpha = 4\rho_g^2 \rho_h(\beta + 1)$, $\beta = M_h l_g^2/(3 L_h L_g)$.*

**Proof** Let $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$, $G = \nabla g(\boldsymbol{u})$, $H = \nabla^2 h(g(\boldsymbol{u}))$. We have using that $G^\top G \succeq \sigma_g^2 \mathrm{I}$, i.e., $G^\top G$ invertible,

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

By bounding each formulation, using $\|G(G^\top G)^{-1}\| \leq 1/\sigma_g$ for the first formulation, and, $\|G(G^\top G)^{-1/2}\|_2 \leq 1$, $(G^\top G)^{1/2} H (G^\top G)^{1/2} \succeq \mu_h \sigma_g^2$ for the second formulation,

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

where we used that $\min\{a,b\} \leq 2/(1/a+1/b)$. Hence, condition (27) is satisfied if $\nu$ 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 (27) is satisfied for $\nu \geq \nu_0 = (a_1 - a_3 + (a_1 + a_3)\sqrt{1 + 4a_2(a_1+a_3)^{-2}})/2$. Since $\sqrt{1+2x} \leq 1 + x$, we have $\nu_0 \leq a_1 + a_2/(a_1+a_3)$, so it suffices to take a regularization larger than or equal to

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

■

Lemma 44 details the computations of the complexity bounds of the ILQR algorithm in the case of strongly convex costs, used in (41) before taking into account the local quadratic convergence.

**Lemma 44** *Consider the notations and assumptions of Theorem 21. The number of iterations of the ILQR algorithm with regularizations*

$$\nu_k = \left(1 + \frac{\alpha}{2(1 + \theta_g\|\nabla h(g(\boldsymbol{u}^{(k)}))\|_2/(\sqrt{\mu_h}\rho_g))}\right) L_g\|\nabla h(g(\boldsymbol{u}^{(k)}))\|_2,$$

*needed to reach an accuracy $\varepsilon$ is at most*

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

*where $\rho_h = L_h/\mu_h$, $\rho_g = l_g/\sigma_g$, $\theta_g = L_g/(\sigma_g^2\sqrt{\mu_h})$, $\theta_h = M_h/(2\mu_h^{3/2})$, $\alpha = 4\rho_g^2\rho_h(\beta+1)$, $\beta = M_h l_g^2/(3L_h L_g)$*

**Proof** Let $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$ and $\boldsymbol{v} = \operatorname{LQR}_{\nu(\boldsymbol{u})}(\mathcal{J})(\boldsymbol{u})$ for

$$\nu(\boldsymbol{u}) = \left(1 + \frac{\alpha}{2(1 + \theta_g\|\nabla h(g(\boldsymbol{u}))\|_2/(\sqrt{\mu_h}\rho_g))}\right) L_g\|\nabla h(g(\boldsymbol{u}))\|_2.$$

As shown in Lemma 43, the chosen regularization ensures the sufficient decrease (27). As in (29), in the proof of Theorem 13, we get that

$$\mathcal{J}(\boldsymbol{u}+\boldsymbol{v}) - \mathcal{J}(\boldsymbol{u}) \leq -\frac{1}{2}\frac{\sigma_g^2}{\sigma_g^2 L_h + \nu(\boldsymbol{u})}\|\nabla h(g(\boldsymbol{u}))\|_2^2 = -\frac{b_1 x^3 + b_2 x^2}{b_3 x^2 + b_4 x + 1},$$

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 13, we get that the total number of iterations to reach an accuracy $\varepsilon$ 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 follows. ■

We present below the proof of Corollary 23 that ensures the validity of the line-search procedure presented in Algo. 2.

**Corollary 45** *Consider the assumptions and notations of Theorem 21 on problem (2) and Algo. 2 with an initial scaled regularization guess $\bar{\nu}_{-1} \leq \left(1+\alpha/(2+2\theta_g\sqrt{\delta_0}/\rho_g)\right)L_g$. The total number of calls to ILQR oracles of Algo. 2 to reach an accuracy $\varepsilon$ is at most $2k(\delta_0,\delta')+\ln\ln(\varepsilon^{-1})+\lceil\log_2\left((1+\alpha/2)L_g/\bar{\nu}_{-1}\right)\rceil$, where $k(\delta_0,\delta')$ is defined as in Theorem 21 and $\delta' = 1/(32\rho_h(\theta_h(1+2\sqrt{\rho_h}\rho_g^3/3)+\sqrt{\rho_h}\theta_g(1+2\rho_g\rho_h))^2)$ is a gap of quadratic convergence for Algo. 2.*

**Proof** Define for $\boldsymbol{u}\in\mathbb{R}^{\tau n_u}$,

$$\bar{\nu}(\boldsymbol{u}) = \left(1+\frac{\alpha}{2(1+\theta_g\sqrt{\mathcal{J}(\boldsymbol{u})-\min_{\boldsymbol{v}\in\mathbb{R}^{\tau n_u}}\mathcal{J}(\boldsymbol{v})}/\rho_g))}\right)L_g$$

Since $h$ is strongly convex, we have that $\|\nabla h(g(\boldsymbol{u}))\|_2 \geq \sqrt{\mu_h}(h(g(\boldsymbol{u}))-\min_{\boldsymbol{y}\in\mathbb{R}^{\tau n_x}}h(\boldsymbol{y})) = \sqrt{\mu_h}(\mathcal{J}(\boldsymbol{u})-\min_{\boldsymbol{v}\in\mathbb{R}^{\tau n_x}}\mathcal{J}(\boldsymbol{v}))$, where we recall that $\min_{\boldsymbol{y}\in\mathbb{R}^{\tau n_x}}h(\boldsymbol{y}) = \min_{\boldsymbol{v}\in\mathbb{R}^{\tau n_x}}\mathcal{J}(\boldsymbol{v})$ as shown in Theorem 13. Hence, we have that $\bar{\nu}(\boldsymbol{u})\|\nabla h(g(\boldsymbol{u}))\|_2 \geq \nu(\boldsymbol{u})$ for $\nu(\boldsymbol{u})$ defined in Lemma 43. Therefore, by Lemma 43, the line-search procedure of Algo. 2 at the $k^{\text{th}}$ iteration necessarily terminates with a scaled regularization $\bar{\nu}_k \leq 2\bar{\nu}(\boldsymbol{u}_k)$ since we chose $\bar{\nu}_{-1} \leq \bar{\nu}(\boldsymbol{u}_0)$ and since $\bar{\nu}(\boldsymbol{u}_k)$ necessarily increases over the iterations as $\mathcal{J}(\boldsymbol{u}_k)$ decreases when condition (27) is satisfied.

Moreover, since $\bar{\nu}(\boldsymbol{u})$ is upper bounded by $(1+\alpha/2)L_g$ the total number of calls to oracles made by the line-search inner loop to satisfy the decrease condition after $k$ iterations is at most

$$k + \left\lceil \log_2\left(\frac{(1+\alpha/2)L_g}{\bar{\nu}_{-1}}\right)\right\rceil.$$

Since the line-search ensures the decrease condition (27), we have, as in Theorem 21 that for $\nu_k = \bar{\nu}_k \|\nabla h(g(\boldsymbol{u}_k))\|_2$,

$$\begin{aligned}\mathcal{J}(\boldsymbol{u}_{k+1}) - \mathcal{J}(\boldsymbol{u}_k) &\leq -\frac{1}{2}\frac{\sigma_g^2}{\sigma_g^2 L_h + \nu_k}\|\nabla h(g(\boldsymbol{u}))\|_2^2 \\ &\leq -\frac{1}{2}\frac{\sigma_g^2}{\sigma_g^2 L_h + 2\bar{\nu}(\boldsymbol{u}_k)\|\nabla h(g(\boldsymbol{u}))\|_2}\|\nabla h(g(\boldsymbol{u}))\|_2^2 \\ &\leq -\frac{1}{4}\frac{\sigma_g^2}{\sigma_g^2 L_h + \bar{\nu}(\boldsymbol{u}_k)\|\nabla h(g(\boldsymbol{u}))\|_2}\|\nabla h(g(\boldsymbol{u}))\|_2^2.\end{aligned}$$

The rest of the proof of Lemma 44 follows, and we get that the number of iterations of Algo. 2 to reach an accuracy $\varepsilon$ is at most $2k(\delta_0, \varepsilon)$ for $k(\delta_0, \varepsilon)$ defined as in Theorem 21.

For the quadratic convergence rate, we have, with the notations of the proof of Theorem 21, that

$$\nu_k/\lambda_h(g(\boldsymbol{u}_k)) \leq 2\bar{\nu}(\boldsymbol{u}_k)\|\nabla h(g(\boldsymbol{u}_k))\|_2^2/\lambda_h(g(\boldsymbol{u}_k)) \leq 2\sqrt{L_h}(L_g + 2l_g(M_h l_g^2/3 + L_g L_h)/(\sigma_g \mu_h)).$$

The rest of the proof follows with a slightly modified quadratic convergence gap. ■

## G. Convergence Analysis of IDDP

The ILQR and iLQR algorithms differ only by the rolling-out phase. The former uses the linearized dynamics, while the latter uses the original shifted dynamics. We formalize the roll-out phase in Definition 46.

**Definition 46** *We define the roll-out of $\tau$ policies $\pi_t : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ along $\tau$ dynamics $\phi_t : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ from $x_0$ as*

$$\begin{aligned}\text{rollout} : x_0, (\phi_t)_{t=0}^{\tau-1}, (\pi_t)_{t=0}^{\tau-1} &\mapsto (u_0, \dots, u_{\tau-1}) \\ &\text{s.t. } u_t = \pi_t(x_t), \quad x_{t+1} = \phi_t(x_t, u_t), \text{ for } t \in \{0, \dots, \tau-1\}.\end{aligned}$$

With the notations of Definition 46, denoting

$$\begin{aligned}\Phi(\boldsymbol{x}, \boldsymbol{u}) &= (\phi_0(x_0, u_0); \dots; \phi_{\tau-1}(x_{\tau-1}, u_{\tau-1})) \\ \pi(\boldsymbol{x}) &= (\pi(x_0); \dots; \pi(x_{\tau-1}))\end{aligned}$$

for $\boldsymbol{x} = (x_1; \dots; x_\tau), \boldsymbol{u} = (u_0; \dots; u_{\tau-1})$, the trajectory $\boldsymbol{x} = \phi^{[\tau]}(x_0, \boldsymbol{u})$ associated to $\boldsymbol{u}$ is the unique solution of $\boldsymbol{x} = \Phi(\boldsymbol{x}, \boldsymbol{u})$ and the roll-out is the unique solution of

$$\boldsymbol{u} = \pi(\boldsymbol{x}), \qquad \boldsymbol{x} = \Phi(\boldsymbol{x}, \boldsymbol{u}). \tag{65}$$

Given a trajectory $(x_1; \dots; x_\tau) = f^{[\tau]}(x_0, \boldsymbol{u})$ computed from $\boldsymbol{u} = (u_0; \dots; u_{\tau-1})$, and $\tau$ policies $(\pi_t)_{t=0}^{\tau-1}$ computed in the backward pass of Algo. 1, the ILQR and iLQR algorithms

can be expressed as

$$\begin{aligned}
\mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u}) &= \mathrm{rollout}(0, (\ell_t)_{t=0}^{\tau-1}, (\pi_t)_{t=0}^{\tau-1}) \\
\text{for } \ell_t(y_t, v_t) &= \ell_{f_t}^{x_t,u_t}(y_t, v_t) = \nabla_{x_t} f_t(x_t, u_t)^\top y_t + \nabla_{u_t} f_t(x_t, u_t)^\top v_t \qquad (66) \\
\mathrm{DDP}_\nu(\mathcal{J})(\boldsymbol{u}) &= \mathrm{rollout}(0, (\delta_t)_{t=0}^{\tau-1}, (\pi_t)_{t=0}^{\tau-1}) \\
\text{for } \delta_t(y_t, v_t) &= \delta_{f_t}^{x_t,u_t}(y_t, v_t) = f_t(x_t + y_t, u_t + v_t) - f_t(x_t, u_t) \qquad (67)
\end{aligned}$$

To analyze the convergence of the iLQR algorithm, we consider how close it is from the ILQR algorithm. This can be traced as measuring how the roll-out phase differ between using $\ell_{f_t}^{x_t,u_t}(y_t, v_t)$ or $\delta_{f_t}^{x_t,u_t}(y_t, v_t)$ as done in Lemma 47.

**Lemma 47** *Given $\tau$ discrete dynamics $f_t : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ and $\tau$ policies $\pi_t : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ for $t = 0, \dots, \tau - 1$, denote*

$$\boldsymbol{v} = \mathrm{rollout}(0, (\ell_t)_{t=0}^{\tau-1}, (\pi_t)_{t=0}^{\tau-1}), \qquad \boldsymbol{w} = \mathrm{rollout}(0, (\delta_t)_{t=0}^{\tau-1}, (\pi_t)_{t=0}^{\tau-1}),$$

*for $\ell_t$, $\delta_t$ defined as in (66) and (67) from $(x_1; \dots; x_\tau) = f^{[\tau]}(x_0, \boldsymbol{u})$ and $\boldsymbol{u} = (u_0; \dots; u_{\tau-1})$. Suppose that the policies are affine of the form are $\pi_t(x_t) = K_t x_t + k_t$, and that all dynamics are Lipschitz continous with Lipschitz-continuous gradients. Then the directions $\boldsymbol{v}$ and $\boldsymbol{w}$ differ by*

$$\|\boldsymbol{w} - \boldsymbol{v}\|_2 \leq \eta(K) \|\boldsymbol{v}\|_2^2$$

*for $\eta(K)$ an increasing function of $\|K\|_2$ detailed in the proof.*

**Proof** In this proof, we ignore the dependency w.r.t. $\bar{x}_0$ and denote simply $f^{[\tau]}(\boldsymbol{u}) = f^{[\tau]}(\bar{x}_0, \boldsymbol{u})$. Similarly, we denote $\ell^{[\tau]}(\boldsymbol{v}) = \ell^{[\tau]}(0, \boldsymbol{v})$ and $\delta^{[\tau]}(\boldsymbol{w}) = \delta^{[\tau]}(0, \boldsymbol{w})$ the trajectories associated to the linearized and shifted dynamics starting from 0. For $\boldsymbol{y} = (y_1; \dots; y_\tau)$, denote $\pi(\boldsymbol{y}) = (\pi_0(0); \pi_1(y_1); \dots; \pi_{\tau-1}(y_{\tau-1}))$. Denoting $K = \sum_{t=2}^{\tau} e_t e_{t-1}^\top \otimes K_{t-1} \in \mathbb{R}^{\tau n_u \times \tau n_x}$, $k = (k_0; \dots; k_{\tau-1}) \in \mathbb{R}^{\tau n_x}$ we have that $\pi(\boldsymbol{v}) = K\boldsymbol{v} + \boldsymbol{k}$ with

$$\begin{pmatrix}
0 & \dots & & \dots & 0 \\
K_1 & \ddots & & & \vdots \\
0 & \ddots & & & \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \dots & 0 & K_{\tau-1} & 0
\end{pmatrix}.$$

The roll-outs are defined as the solutions of

$$\boldsymbol{v} = \pi(\ell^{[\tau]}(\boldsymbol{v})), \quad \boldsymbol{w} = \pi(\delta^{[\tau]}(\boldsymbol{w}))$$

From Lemma 35, the linearized trajectories can be expressed as

$$\ell^{[\tau]}(\boldsymbol{v}) = (\mathrm{I} - A)^{-1} B \boldsymbol{v}$$

for $B = \nabla_{\boldsymbol{u}} F(\boldsymbol{x}, \boldsymbol{u})^\top$, $A = \nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{u})^\top$, with $F(\boldsymbol{x}, \boldsymbol{u}) = (f_0(x_0, u_0); \dots; f_{\tau-1}(x_{\tau-1}, u_{\tau-1}))$ for $\boldsymbol{x} = (x_1; \dots; x_\tau)$, $\boldsymbol{u} = (u_0; \dots; u_{\tau-1})$. We have that $\boldsymbol{v}$ satisfy

$$\boldsymbol{v} = K(\mathrm{I} - A)^{-1} B \boldsymbol{v} + \boldsymbol{k}.$$

Note that $A + BK = \sum_{t=2}^{\tau} e_t e_{t-1}^\top \otimes (A_{t-1} + B_{t-1}K_{t-1})$ for $B_t = \nabla_{u_t} f_t(x_t, u_t)^\top, A_t = \nabla_{x_t} f_t(x_t, u_t)^\top$, that is

$$BK + A = \begin{pmatrix} 0 & \dots & & \dots & 0 \\ U_1 & \ddots & & & \vdots \\ 0 & \ddots & & & \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & U_{\tau-1} & 0 \end{pmatrix},$$

for $U_t = A_t + B_t K_t$. So $\mathrm{I} - A - BK$ is invertible by solving an autoregressive problem. We then have that $(\mathrm{I} - K(\mathrm{I} - A)^{-1}B)(\mathrm{I} + K(\mathrm{I} - A - BK)^{-1}B) = \mathrm{I}$ such that the solution of $\boldsymbol{v} = \pi(\ell^{[\tau]}(\boldsymbol{v}))$ is

$$\boldsymbol{v} = \boldsymbol{k} + K(\mathrm{I} - A - BK)^{-1}B\boldsymbol{k}.$$

For $\boldsymbol{w} = (w_0; \dots; w_{\tau-1})$, denote $\boldsymbol{z} = (z_1; \dots; z_\tau) = \delta^{[\tau]}(\boldsymbol{w})$ s.t. $z_{t+1} = f_t(x_t + z_t, u_t + w_t) - f_t(x_t, u_t)$ for $t \in \{0, \dots, \tau - 1\}$, 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 $\mathrm{i}^{\text{th}}$ coordinate of $f$, we have

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

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

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

Denoting $C = \sum_{t=2}^{\tau} e_t e_{t-1} \otimes C_{t-1}$, $D = \sum_{t=1}^{\tau} e_t e_t^\top \otimes D_{t-1}$, we get that $\delta^{[\tau]}(\boldsymbol{u}) = (\mathrm{I} - C)^{-1} D\boldsymbol{v}$. Since $\boldsymbol{w} = \pi(\delta^{[\tau]}(\boldsymbol{u}))$, we get that $\boldsymbol{w}$ satisfies

$$\boldsymbol{w} = K(\mathrm{I} - C)^{-1} D\boldsymbol{w} + \boldsymbol{k}.$$

The solution of this system can be found as before as

$$\boldsymbol{w} = \boldsymbol{k} + K(\mathrm{I} - C - DK)^{-1} D\boldsymbol{k}.$$

We then have

$$\|\boldsymbol{w} - \boldsymbol{v}\|_2 \leq \|K\|_2 \|(\mathrm{I} - C - DK)^{-1} D - (\mathrm{I} - A - BK)^{-1} B\|_2 \|\boldsymbol{k}\|_2$$

Then, we decompose the middle term as

$$\begin{aligned} &(\mathrm{I} - C - DK)^{-1} D - (\mathrm{I} - A - BK)^{-1} B \\ &= ((\mathrm{I} - C - DK)^{-1} - (\mathrm{I} - A - BK)^{-1}) D - (\mathrm{I} - A - BK)^{-1}(B - D) \\ &= (\mathrm{I} - C - DK)^{-1}(C - A + (D - B)K)(\mathrm{I} - A - BK)^{-1} D \\ &\quad - (\mathrm{I} - A - BK)^{-1}(B - D). \end{aligned}$$

We have $(\mathrm{I} - A - BK)^{-1} = \sum_{t=0}^{\tau-1}(A+BK)^t$ since $(A+BK)^\tau = 0$. So we get

$$\|(\mathrm{I} - A - BK)^{-1}\|_2 \leq \sum_{t=0}^{\tau-1}\|A+BK\|_2^t \leq \sum_{t=0}^{\tau-1}(l_f^x + l_f^u\|K\|_2)^t := S_1(K),$$

for $l_f^x, l_f^u$ defined as in Lemma 36. Similarly, we have

$$\|(\mathrm{I} - C - DK)^{-1}\|_2 \leq \sum_{t=0}^{\tau-1}(n_x l_f^x + n_x l_f^u\|K\|_2)^t := S_2(K).$$

Using the block structure of the matrices, we have, using that $\|\eta_{t,i}\| \leq \|w_t\|_2$, $\|\zeta_{t,i}\| \leq \|z_t\|_2$,

$$\begin{aligned}\|B - D\|_2 &\leq n_x L_f^{uu}\|\boldsymbol{w}\|_2\\ \|A - C\|_2 &\leq n_x(L_f^{xx}\|\boldsymbol{z}\|_2 + L_f^{xu}\|\boldsymbol{w}\|_2),\\ \|D\|_2 &\leq n_x l_f^u,\end{aligned}$$

for $L_f^{uu}, L_f^{xx}, L_f^{xu}$ defined as in Lemma 36. In addition, we have that $\|\boldsymbol{z}\|_2 \leq l_{f^{[\tau]}}\|\boldsymbol{w}\|_2$, where $l_{f^{[\tau]}}$ is the Lipschitz-constant of $f^{[\tau]}$ computed in Lemma 36.

So in total we get that

$$\begin{aligned}\|\boldsymbol{w} - \boldsymbol{v}\|_2 &\leq \|K\|_2(n_x{}^2 S_1(K)S_2(K)l_f^u(L_f^{xx}l_{f^{[\tau]}} + L_f^{xu} + L_f^{uu}\|K\|_2) + n_x S_1(K)L_f^{uu})\|\boldsymbol{w}\|_2\|\boldsymbol{k}\|_2\\ &:= \eta_1(K)\|\boldsymbol{w}\|_2\|\boldsymbol{k}\|_2\end{aligned}$$

Now since $\boldsymbol{w} = \boldsymbol{k} + K(\mathrm{I} - C - DK)^{-1}D\boldsymbol{k}$ and $\boldsymbol{k} = \boldsymbol{v} - K(\mathrm{I} - A)^{-1}B\boldsymbol{v}$, we have

$$\begin{aligned}\|\boldsymbol{w}\|_2 &\leq (1 + n_x\|K\|_2 S_2(K)l_f^u)\|\boldsymbol{k}\|_2,\\ \|\boldsymbol{k}\|_2 &\leq (1 + \|K\|_2 l_{f^{[\tau]}})\|\boldsymbol{v}\|_2.\end{aligned}$$

Hence, we get

$$\begin{aligned}\|\boldsymbol{w} - \boldsymbol{v}\|_2 &\leq \eta_1(K)(1 + n_x\|K\|_2 S_2(K)l_f^u)(1 + \|K\|_2 l_{f^{[\tau]}})^2\|\boldsymbol{v}\|_2^2\\ &:= \eta(K)\|\boldsymbol{v}\|_2^2.\end{aligned}$$

■

It remains to bound the Lipschitz continuity constant of the policies derived in the backward pass of the ILQR and iLQR algorithms. In the general case, i.e., problem (1), Lemma 48 shows that the policies are Lipschitz continuous with a Lipschitz continuity parameter independent of $\nu$ provided that $\nu$ is sufficiently large. For the restricted problem (2), the policies are Lipschitz continuous with a Lipschitz continuous parameter independent of $\nu$ unconditionally, as shown in Lemma 50.

**Lemma 48** *Consider problem (1) with dynamics and costs Lipschitz continuous with Lipschitz-continuous gradients as in Assumption 30. For any $\nu \geq 2l_g L_h$, the policies $\pi_t : y_t \mapsto K_t y_t + k_t$ computed in Algo. 1 are well-defined and Lipschitz continuous with $\|K_t\|_2 \leq c$ for $c$ independent of $\nu$.*

**Proof** For $t \in \{0, \dots, \tau-1\}$, denote $f^{[t:\tau]}(x_t, \boldsymbol{u}_{[t:\tau-1]}) = (x_{t+1}; \dots, x_\tau)$ the control of the dynamics $f_t, \dots, f_{\tau-1}$ starting from $x_t$ with control variables $\boldsymbol{u}_{[t:\tau-1]} = (u_t; \dots; u_{\tau-1})$. For $t = 0$, denoting $[0:\tau] = [\tau]$, we retrieve Definition 29. Define similarly $g^{[t:\tau]}(\boldsymbol{u}_{[t:\tau-1]}) = (f^{[t:\tau]}(x_t, \boldsymbol{u}_{[t:\tau-1]}), \boldsymbol{u}_{[t:\tau-1]})$ and $h^{[t:\tau]}(x_{[t+1;\tau]}, \boldsymbol{u}_{[t:\tau-1]}) = \sum_{s=t}^{\tau-1} h_s(x_s, u_s) + h_\tau(x_\tau)$. The $t^{\text{th}}$ policy $\pi_t(y_t)$ is formally equal to $v_t^*(y_t)$ for

$$\begin{aligned} v_t^*(y_t), \dots v_{\tau-1}^*(y_t) &= \underset{u_t, \dots, u_{\tau-1}}{\arg\min}\ q_{h^{[t:\tau]}}^{f^{[t:\tau]}(x_t, \boldsymbol{u}_{[t:\tau-1]})}(\ell_{f^{[t:\tau]}}^{x_t, \boldsymbol{u}_{t:\tau-1}}(y_t, \boldsymbol{v}_{[t:\tau-1]}), \boldsymbol{v}_{[t:\tau-1]}) + \frac{\nu}{2}\|\boldsymbol{v}_{[t:\tau-1]}\|_2^2 \\ &= (\nu\,\mathrm{I} + \nabla g^{[t:\tau]}(\boldsymbol{u}_{[t:\tau]}) \nabla^2 h^{[t:\tau]}(g^{[t:\tau]}(\boldsymbol{u}_{[t:\tau]})) \nabla g^{[t:\tau]}(\boldsymbol{u}_{[t:\tau]})^\top)^{-1}(Ay_t + a) \end{aligned}$$

for some $A$, $a$ independent of $\nu$. For $\nu > L_{h^{[t:\tau]}} l_{g^{[t:\tau]}}^2$ the policies are well-defined. Since $L_{h^{[t:\tau]}} = L_{h^{[\tau]}} = L_h$ and $l_{g^{[t:\tau]}}^2 \leq l_{g^{[0:\tau]}}^2 = l_g^2$, the policies are well-defined for any $\nu \geq 2 l_g L_h$. Moreover, for any $\nu \geq 2 l_g L_h$,

$$\|(\nu\,\mathrm{I} + \nabla g^{[t:\tau]}(\boldsymbol{u}_{[t:\tau]}) \nabla^2 h^{[t:\tau]}(g^{[t:\tau]}(\boldsymbol{u}_{[t:\tau]})))^{-1}\|_2 \leq \frac{1}{l_g L_h}.$$

Hence, the associated policy $\pi_t$ is at most $1/(l_g L_h)$ Lipschitz-continuous. ■

**Corollary 49** *Consider problem (1) with dynamics and costs Lipschitz continuous with Lipschitz-continuous gradients as in Assumption 30. For any $\nu \geq 2 l_g L_h$, there exists a constant $\eta$ independent of $\nu$, such that the ILQR or iLQR directions $\boldsymbol{v} = \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$ and $\boldsymbol{w} = \mathrm{DDP}_\nu(\mathcal{J})(\boldsymbol{u})$ on any control variables $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$ differ by*

$$\|\boldsymbol{w} - \boldsymbol{v}\|_2 \leq \eta \|\boldsymbol{v}\|_2^2.$$

**Proof** The result follows from Lemma 47 and 48. ■

**Lemma 50** *Consider Algo. 1 applied to problems of the form (2), that is, such that $R_t = 0, Q_t = 0, q_t = 0$. Assume in addition that the costs are strongly convex, the dynamics are surjective and both costs and dynamics are smooth as described in Assumption 11. The policies $\pi_t : y_t \to K_t y_t + k_t$ computed in Algo. 1 are always well-defined and such that $\|K_t\|_2 \leq c$ for some $c$ independent of $\nu$.*

**Proof** Consider $K_t, J_t$ defined in Algo. 1 for a command $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$, a regularization $\nu > 0$ and no control costs ($R_t = 0, Q_t = 0, q_t = 0$). By recursion, we have that $J_t$ is positive definite, since

$$J_t = P_t + A_t^\top J_{t+1}^{1/2}(\mathrm{I} + \nu^{-1} J_{t+1}^{1/2} B_t B_t^\top J_{t+1}^{1/2})^{-1} J_{t+1}^{1/2} A_t,$$

and $J_\tau = P_\tau$, and $P_t$ are positive definite.

In particular, $J_t \succeq P_t \succeq \mu_h \,\mathrm{I}$ and for any $t \in \{1, \dots, \tau-1\}$,

$$\begin{aligned} \|J_t\|_2 &\leq L_h + (l_f^x)^2 \|J_{t+1}\|_2 \\ &\leq \sum_{s=t}^{\tau} (\prod_{j=t}^{s-1} (l_f^x)^2) L_h, \end{aligned}$$

where here and in the following we use $l_f^x, l_f^u$ defined in Lemma 36. Therefore, we have

$$\sup_{t\in\{1,\dots,\tau\}} \|J_t\|_2 \leq L_h \sum_{t=1}^{\tau} (l_f^x)^{2(s-t)}.$$

On the other hand, we have

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

The spectral norm of the matrix $(\nu \mathrm{I} + B_t^\top J_{t+1} B_t)^{-1} B_t^\top$ can be bounded just as in Lemma 43 given the assumptions. Namely, we have,

$$\|(\nu \mathrm{I} + B_t^\top J_{t+1} B_t)^{-1} B_t^\top\|_2 \leq \frac{2 l_f^u}{\nu + \mu_h \sigma_f l_f^u}.$$

Hence, we have

$$\begin{aligned}\sup_{t\in\{0,\dots,\tau-1\}} \|K_t\|_2 &\leq \frac{2 l_f^u l_f^x}{\nu + \mu_h \sigma_f l_f^u} L_h \sum_{t=1}^{\tau} (l_f^x)^{2(s-t)} \\ &\leq \frac{2 l_f^x}{\mu_h \sigma_f} L_h \sum_{t=1}^{\tau} (l_f^x)^{2(s-t)}.\end{aligned}$$

■

**Corollary 51** *Consider Algo. 1 applied to problems of the form (2), that is, such that $R_t = 0, Q_t = 0, q_t = 0$. Assume in addition that the costs are strongly convex, the dynamics are surjective and both costs and dynamics are smooth as described in Assumption 11. Then there exists a constant $\eta$ independent of $\nu$, such that the ILQR or iLQR directions $\boldsymbol{v} = \mathrm{LQR}_\nu(\mathcal{J})(\boldsymbol{u})$ and $\boldsymbol{w} = \mathrm{DDP}_\nu(\mathcal{J})(\boldsymbol{u})$ on any control variables $\boldsymbol{u} \in \mathbb{R}^{\tau n_u}$ differ by*

$$\|\boldsymbol{w} - \boldsymbol{v}\|_2 \leq \eta \|\boldsymbol{v}\|_2^2.$$

**Proof** Follows from Lemma 47 and 50. ■

## H. Detailed Computations

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

### H.1 Details on Theorem 10

**Proof** [Details on Eq. (22)] Note that $\nabla_{\boldsymbol{v}}\Phi(\boldsymbol{y},\boldsymbol{v}) = \mathbf{diag}((\nabla_{v_t}\phi(y_t,v_t))_{t=0}^{k-1})$, such that, by definition of **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, for $D = \sum_{t=1}^{k-1} e_t e_{t+1}^\top$ the upper-shift matrix in $\mathbb{R}^k = \mathbb{R}^{n_x}$.

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

On the other hand, we have

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

■

**Proof** [Details on line (ii) in Eq. (23)] Denote $K_t = \nabla a(y_t)$. Using that $De_t{=}e_{t-1}$, we have $FA{=}(D\otimes \mathrm{I})(\sum_{t=1}^n e_t e_t^\top \otimes K_{t-1}) = \sum_{t=2}^n e_{t-1}e_t^\top \otimes K_{t-1}) = \sum_{t=1}^{n-1} e_t e_{t+1}^\top K_t$ and, using that $e_t^\top D = e_{t+1}^\top$, we have $CF = (\sum_{t=1}^n e_t e_t^\top \otimes K_t)(D\otimes \mathrm{I}) = \sum_{t=1}^{n-1} e_t e_{t+1}^\top \otimes K_t$. Therefore, we have $FA = CF$, and similarly we can show that $FA^{-1} = C^{-1}F$. ■

**Proof** [Details on line (iii) in Eq. (23)] 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 $(\mathrm{I} - F\otimes G)^{-1}F = (\sum_{t=0}^{k-1} D^t\otimes (D^\top)^t)(D\otimes \mathrm{I}) = \sum_{t=1}^{k-1} D^t \otimes (D^\top)^{t-1}$. ■

**Proof** [Details on the extension to multi inputs] Consider the multi-input case as described in Def. 9. For any $k \geq r$, $j\in\{1,\dots,m_u\}$, $i\in\{1,\dots,r_j\}$, 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 $\boldsymbol{w} = (w_0;\dots;w_{k-1})$, we have that $\boldsymbol{\omega} = T\boldsymbol{w}$ reorders the coordinates of $\boldsymbol{w}$ such that $\boldsymbol{\omega} = (\omega_1;\dots;\omega_{m_u})$ with $\omega_j^{(i)} = w_{i-1}^{(j)}$ for $i\in\{1,\dots,k\}$, $j\in\{1,\dots,m_u\}$. Hence we have for any $k\geq r$, $j\in\{1,\dots,m_u\}$, denoting here $e_i$ the ith canonical vector in $\mathbb{R}^{r_j}$, $D_{r_j}$ the upper-shift matrix in $\mathbb{R}^{r_j}$,

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

where $0_{r_j}$ is the null vector in $\mathbb{R}^{r_j}$ and $\mathrm{I}_{r_j}$ 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 $k = r$ and the notations of the proof of Theorem 10. We can write that $y_{t+1} = a^{-1}(MB(\phi^{[k]}(y_0,\boldsymbol{v}),\boldsymbol{v}))$. 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 $i^{\text{th}}$ canonical vector in $\mathbb{R}^{r_j}$ and $e_{\ell_j}$ 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 $t \in \{0, \dots, \tau - 1\}$,

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

The rest of the proof follows as in the proof of Theorem 10 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. ■

### H.2 Details on Theorem 24

**Proof** [Details on Eq. (43)] With the notations of Theorem 13, we have that

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

with $x = \|\nabla h(g(\boldsymbol{u}^{(k)}))\|_2$ and $\delta_k = \mathcal{J}(\boldsymbol{u}^{(k)}) - \mathcal{J}^*$. The function $f_1 : x \to \sigma_g^2 x^2 / (2(\sigma_g^2 L_h + \xi x + \rho_h \chi^2 x^2))$ is strictly increasing, so we can follow the steps of the proof Theorem 13 and obtain that $f_2(\delta_k)(\delta_{k+1} - \delta_k) \leq -1$ with

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

The result follows by integrating $f_2'$ and, as in the proof Theorem 13, we have that convergence to an accuracy $\varepsilon$ is ensured after at most $k \leq f_2(\delta_0) - f_2(\varepsilon)$. ■

## I. Additional Numerical Evaluations

**Realistic model of a car with tracking cost.** On Fig. 8, we consider the same setting as for the simple model of a car except that we replace the simple model of the dynamics of a car by a bicycle model driven by tire forces taken from Liniger et al. (2015), also detailed by Roulet et al. (2022, Section 10.3.1). We considered a fourth order Runge Kutta discretization scheme of the continuous dynamics of the bicycle model of the car. We keep a tracking cost as explained for the simple model of a car in Section 5. We use a discretization step $\Delta = T/\tau$ for a total time $T = 2$, and a number of discretization steps $\tau = 25$. We use random initial control sequences $u_t^{(0)} \sim \mathcal{N}(0, \sigma)$ for $\sigma = 1/\tau = 25$.

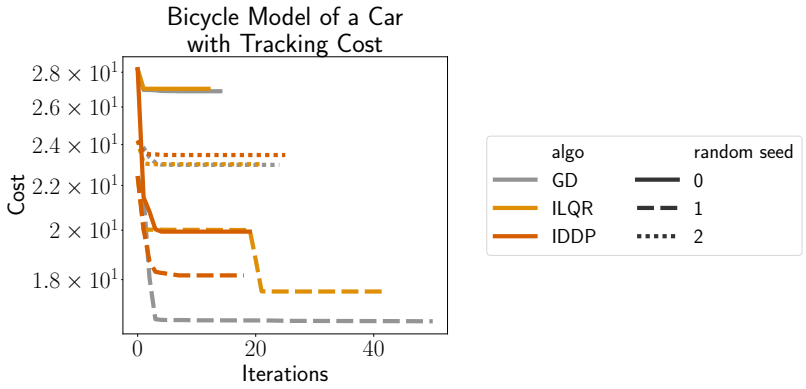


Figure 8: Convergence of gradient descent (GD), ILQR and ILQC to control a bicycle model of a car for a tracking cost

In this case, the ILQR and IDDP algorithms do not appear to converge to the same value across random initial control sequences. This suggests no global convergence phenomenon in this example.

Contouring costs and model predictive controllers can circumvent the difficulty of this task as presented by Liniger et al. (2015); Roulet et al. (2022). However, the bottleneck of, e.g., model predictive controllers remain an algorithm such as ILQR or IDDP to compute the short term policies. Understanding the behavior of these algorithms may then help the design of model predictive controllers.

**Convergence rates.** In Fig. 9, we plot convergence in iterates for the pendulum example. We retrieve a similar superlinear rate of convergence after some number of iterations.

In Fig. 10 and Fig. 11, we also consider convergence rates in function values, that is, $\rho^{(k)} = (c^{(k)} - c^*)/(c^{(k-1)} - c^*)$ for $c^{(k)}$ the cost at iterate $k$ and $c^*$ the minimal cost. To plot this rate, we consider $c^* = 0$ when subsampling the costs. For non-subsampled costs, we consider $c^*$ as the minimum reached by ILQR and IDDP algorithms across random initializations for the pendulum example. For the simple model of a car, we consider $c^*$ as the minimum reached by each algorithm on the given instance. In that case, $c^*$ may be a local minimum.

We observe generally a long phase where the convergence rate is close to one, followed by a sudden phase of superlinear convergence where the rate drops to 0. The second phase of convergence outlined in the theory of Section 4 appears transient. The algorithms appear to mostly show a phase of sublinear convergence followed by a phase of superlinear convergence.

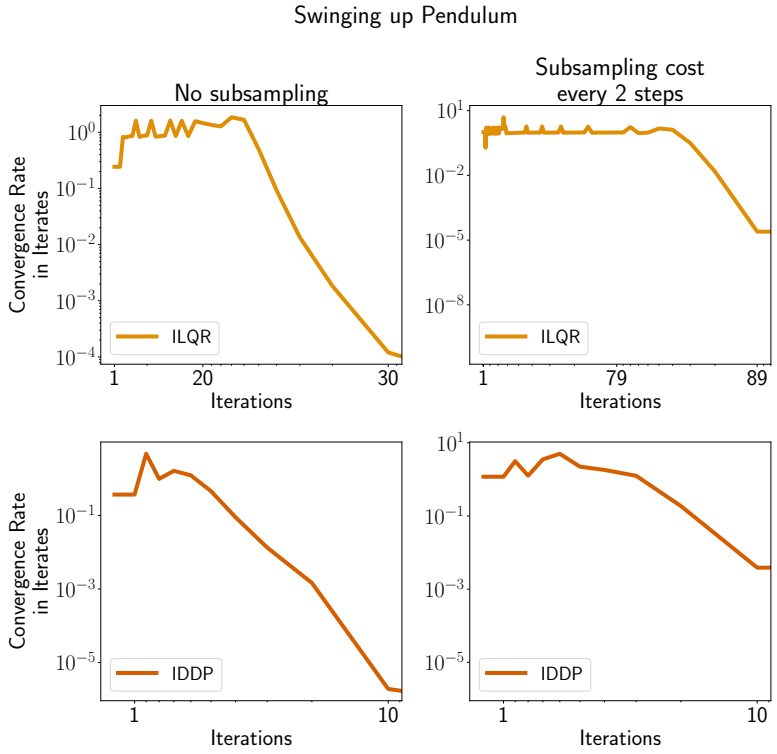


Figure 9: Convergence rate in iterates, $\kappa^{(k)} = \|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)}\|_2 / \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^{(k-1)}\|_2$, along iterations of ILQR and IDDP algorithms for the pendulum example with or without subsampling the costs.

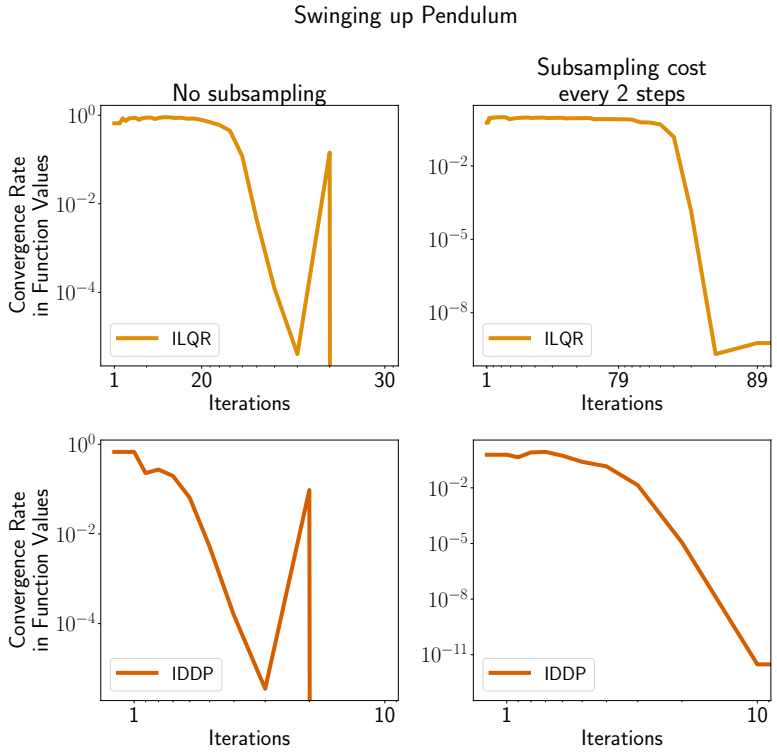


Figure 10: Convergence rate in function values, $\rho^{(k)} = (c^{(k)} - c^*)/(c^{(k)} - c^*)$, along iterations of ILQR and IDDP algorithms for the pendulum example with or without subsampling the costs. When subsampling, the minimal cost is set to $c^* = 0$. Without subsampling, the minimal cost is set to the minimal cost found by all algorithms across random initializations.

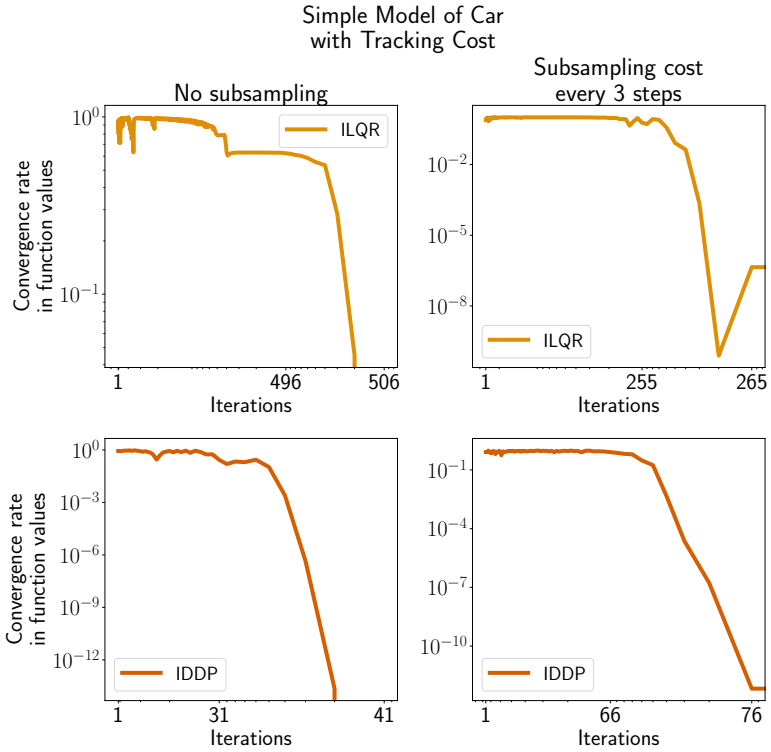


Figure 11: Convergence rate in function values, $\rho^{(k)} = (c^{(k)} - c^*)/(c^{(k)} - c^*)$, along iterations of ILQR and IDDP algorithms for the simple model of a car with or without subsampling the costs. When subsampling, the minimal cost is set to $c^* = 0$. Without subsampling, the minimal cost is set to the minimal cost found by the particular algorithm on this instance.