Interior Point Differential Dynamic Programming

Andrei Pavlov, Student Member, IEEE, Iman Shames, Member, IEEE, and Chris Manzie, Senior Member, IEEE.

Abstract—This paper introduces a new Differential Dynamic Programming (DDP) algorithm for solving discrete-time finite-horizon optimal control problems with inequality constraints. Two variants, namely Feasible- and Infeasible-IPDDP algorithms, are developed using primal-dual interior-point methodology, and their local quadratic convergence properties are characterised. We show that the stationary points of the algorithms are the perturbed KKT points, and thus can be moved arbitrarily close to a locally optimal solution. Being free from the burden of the active-set methods, it can handle nonlinear state and input inequality constraints without a discernible increase in its computational complexity relative to the unconstrained case. The performance of the proposed algorithms is demonstrated using numerical experiments on three different problems: control-limited inverted pendulum, car-parking, and 3-link planar manipulator.

Index Terms—Finite Horizon Optimal Control, Differential Dynamic Programming, Interior Point Methods, Numerical Methods.

I. INTRODUCTION

Among optimisation algorithms that can effectively utilise the structure of the optimal control problems, a distinct place is held by Differential Dynamic Programming (DDP) algorithm introduced by Mayne [1]. Its advantages include optimisation over the space of control inputs with linear complexity in the length of prediction horizon and locally optimal feedback policies as an extra output. Being a second-order method, it has provable local quadratic convergence [2], while global convergence with inexact line-search can also be established [3]. Moreover, DDP can address minimax-type problems [4] [5] or be used for Model Predictive Control applications [6]. However, the DDP approach requires second-order derivatives of system dynamics, which makes it prohibitively expensive for solving large problems. As a possible remedy, one can omit the second-order information as in iterative-Linear-Quadratic Regulator (iLQR) [7], or estimate it using Quasi-Newton DDP [8], Sampled DDP [9] or Unscented DDP [10].

One major shortcoming of DDP is a lack of an elegant generalisation for the problems with general inequality constraints. Methods reported in the literature usually fall into one of two distinct categories: penalty, barrier and Augmented Lagrangian (AL) methods; or active sets methods. The first family utilises penalty and barrier functions (or augments the Lagrangian function) to convert constrained problems into unconstrained, e.g., see [11] [12]. Algorithms of the first family potentially suffer from major drawbacks, such as ill-conditioning, time-consuming hand-tuning, slow convergence and/or appearance of saddle-points. The second family is based on active-set methodology, e.g., see [13]–[15]. In contrast to the former family, these methods require the constraints to be explicitly dependent on the control variables, and are characterised by presence of an extra routine aimed at identification of active/inactive constraints. This approach is known to have combinatorial complexity in the number of constraints in the worst case. To circumvent potential computational difficulties Tassa et. al. [16] propose Control-Limited Differential Dynamic Programming (CLDDP) algorithm where only box constraints on input are considered. A hybrid algorithm which uses ideas from both families was reported in [17]. To the best of our knowledge there no rigorous proofs of convergence for constrained DDP algorithms.

While general purpose primal-dual interior-point methods are widely known for their polynomial time complexity and quadratic convergence, there were no extension of DDP algorithm accommodating primal-dual interior-point techniques reported so far. Our aim is to fill the gap and explain the benefits of this approach. As demonstrated in the following, the Interior-Point DDP algorithm seems to be one of the most natural extensions to DDP. It requires neither modifying the objective function nor identifying active/inactive constraints by a separate procedure. Importantly, it has provable local quadratic convergence for problems with state and control constraints, which is a new result in the DDP framework.

The paper is organised as follows. In Section II we introduce the finite-time optimal control problem, and state a system of (perturbed) first-order conditions for local optimality. In Section III, the main contribution of the paper, i.e., two extensions for the Differential Dynamic Programming (DDP) algorithm, namely Feasible- and Infeasible-Interior-Point DDP algorithms are described and their local quadratic convergence properties are established. In Section IV we consider numerical examples to demonstrate advantages of the proposed algorithms.

II. PRELIMINARIES

Consider a discrete-time system of the following form:

\[ x_{t+1} = f(x_t, u_t), \]

subject to constraints \( c(x_t, u_t) \leq 0 \), where \( x_t \in \mathbb{R}^n \) and \( u_t \in \mathbb{R}^m \) are state and control vectors at time \( t \), functions \( f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n \) and \( c : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^l \) are twice continuously differentiable.

Consider a finite-time constrained optimal control problem for the above system at a known state \( \bar{x}_0 \):

\[
J_N^* (\bar{x}_0) = \min_{x, u} \sum_{t=0}^{N-1} q(x_t, u_t) + p(x_N) \\
\text{s.t.} \quad x_0 = \bar{x}_0, \quad \text{and for } t \in \{0, \ldots, N-1\} : \\
x_{t+1} = f(x_t, u_t), \quad c(x_t, u_t) \leq 0, \\
(1)
\]
where \( q : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^2 \) and \( p : \mathbb{R}^n \rightarrow \mathbb{R}^2 \) are twice continuously differentiable stage cost and terminal cost, respectively. Moreover, \( N \) is a positive integer that denotes the prediction horizon length, and vectors \( x \) and \( u \) are concatenations of the corresponding decision variables, i.e., \( x = [x_0, \ldots, x_N] \) and \( u = [u_0, \ldots, u_{N-1}] \).

The Lagrangian function for this optimisation problem is

\[
L(x, u, \lambda, s) = \sum_{t=0}^{N-1} q(x_t, u_t) + \langle s_t, c(x_t, u_t) \rangle + p(x_N)
\]

\[
+ \langle \lambda_0, x_0 - x_0 \rangle + \sum_{t=1}^{N} \langle \lambda_t, f(x_{t-1}, u_{t-1}) - x_t \rangle,
\]

where \( s_t \in \mathbb{R}^l \) and \( \lambda_t \in \mathbb{R}^m \) are the dual variables.

Define vectors of dual variables \( \lambda = [\lambda_0, \ldots, \lambda_N] \) and \( s = [s_0, \ldots, s_{N-1}] \), and let \( c(x, u) \) be the vector of \( c(x_t, u_t) \), i.e., \( c(x, u) = [c(x_0, u_0), \ldots, c(x_{N-1}, u_{N-1})] \).

Denote \( C_t = \text{diag}[c(x_t, u_t)] \), \( S_t = \text{diag}[s_t] \) and \( S = \text{diag}[s] \).

The perturbed KKT system is defined as

\[
\nabla_u L(x, u, \lambda, s) = 0, \quad \nabla_\lambda L(x, u, \lambda, s) = 0, \quad \nabla_\lambda L(x, u, \lambda, s) = 0,
\]

\[Sc(x, u) + \mu = 0, \quad c(x, u) \leq 0, \quad s \geq 0, \]

where \( \nabla \) is the gradient operator, inequalities are understood to be element-wise, and \( \mu \) is a vector of perturbation \( \mu > 0 \) of an appropriate dimension. The KKT system, i.e., system (3) with \( \mu = 0 \), defines a set of the first order necessary conditions for a local constrained minimiser.

**Assumption 1.** The KKT system has a solution \((x^*, u^*, \lambda^*, s^*)\), which satisfies the following conditions:

1. Strict complementarity holds at the solution, i.e., \( c(x_t^*, u_t^*) < s_t^* \) for \( t \in \{0, \ldots, N-1\} \).
2. The standard second-order constrained optimality conditions hold at \((x^*, u^*, \lambda^*, s^*)\).

Under Assumption 1 and some additional regularity requirements, the optimal solution of (3) can be obtained as the limit point of the solutions to the perturbed KKT system for decaying perturbation, e.g., see [13].

**Remark 1** (Iteration Index Convention). For the clarity of notation, we drop variable iteration indices. Instead we use the superscript \( t \) to denote the value of a variable at the next iteration. A variable’s value at the current iteration is denoted by the variable name without this superscript.

### III. PRIMAL-DUAL INTERIOR-POINT DIFFERENTIAL DYNAMIC PROGRAMMING

In this section we introduce the primal-dual Interior-Point Differential Dynamic Programming (IPDDP) algorithms for solving constrained optimal control problems.

Starting from now we make no distinction between decision variable \( x_0 \) and initial state \( x_0 \). This cause no ambiguity since we always operate with dynamically feasible state trajectories.

To illustrate the underlying idea behind the algorithm, we first resolve the system’s states in (1), and apply the Bellman’s optimality principle to transform (1) into

\[
\min_{u_0 \text{ s.t.}} \left[q(x_0, u_0) + \min_{u_1 \text{ s.t.}} \left[q(f(x_0, u_0), u_1) + \ldots \right]\right].
\]

Defining \( J_0 \): \( p(x) \) and, for \( k \in \{1, \ldots, N\} \),

\[
J_k(x) := \min_{u \text{ s.t.}} [q(x, u) + J_{k-1}(f(x, u))],
\]

where \( x \in \mathbb{R}^n \) and \( u \in \mathbb{R}^m \), equation (4) can be written as

\[
J_N(x_0) = \min_{u_0 \text{ s.t.}} [q(x_0, u_0) + J_{N-1}(f(x_0, u_0))].
\]

This enables one to apply the dynamic programming principle to solve the optimisation problem as a series of nested optimisation problems, which is the main idea behind any DDP method. However, in this paper, instead of solving the constrained nested problems sequentially, we replace the constrained minimisation problem with their min-max primal-dual counter part. With a mild abuse of notation we define

\[
J_k(x) := \min_{u \text{ s.t.}} [\ell(x, u, s) + J_{k-1}(f(x, u))],
\]

where \( \ell(x, u, s) := q(x, u) + (s, c(x, u)) \). In turn, this allows us to state the problem of interest in this paper as:

\[
J_N(x_0) = \min_{u_0 \text{ s.t.}} \left[\ell(x_0, u_0, s_0) + J_{N-1}(f(x_0, u_0))\right].
\]

**A. PRIMAL-DUAL FEASIBLE-INTERIOR-POINT DDP**

Let \( \mu \) be a strictly positive constant and assume that an initial solution estimate for \((3)\) is given by tuple \((x, u, s)\), such that \( x_{t+1} = f(x_t, u_t), c(x_t, u_t) < 0 \) and \( s_t > 0 \) for \( t \in \{0, \ldots, N-1\} \). The solution approach aims to improve the initial trajectory by calculating updates for control inputs that minimise a local quadratic model of the value function in the vicinity of this trajectory through a Backward Pass, and computing a new trajectory after a Forward Pass. The backward and forward passes are carried out repeatedly until a convergence criterion is satisfied.

1) **Backward pass:** Define quadratic functions

\[
V^t(x) := V_0^t + (V_0^t)^T (x - x_t) + (x - x_t)^T V_{xx}^t (x - x_t),
\]

where \( x \in \mathbb{R}^n \) and \( t \in \{0, \ldots, N\} \). Let \( V_0^N = p(x_N) \), \( V_x^N = p_x(x_N) \) and \( V_{xx}^N = p_{xx}(x_N) \) are the value, gradient and Hessian of function \( p(x) \) at \( x_N \) correspondingly, while the remaining coefficients \( V_0^t, V_x^t \) and \( V_{xx}^t \) are to be defined recursively later in the following.

Define \( Q^t(x, u, s) := \ell(x, u, s) + V^{t+1}(f(x, u)) \) for \( t \in \{0, \ldots, N-1\} \), and consider its second-order Taylor expansion \( \delta Q^t(\delta x, \delta u, \delta s) \) around \((x_t, u_t, s_t)\):

\[
\delta Q^t(\delta x, \delta u, \delta s) := Q_0^t + \begin{pmatrix} Q_x^t\
Q_u^t\
Q_s^t\end{pmatrix}^T \begin{pmatrix} \delta x \\ \delta u \\ \delta s \end{pmatrix} + \frac{1}{2} \begin{pmatrix} \delta x \\ \delta u \\ \delta s \end{pmatrix}^T \begin{pmatrix} Q_{xx}^t & Q_{xu}^t & Q_{xs}^t \\ Q_{ux}^t & Q_{uu}^t & Q_{us}^t \\ Q_{sx}^t & Q_{su}^t & Q_{ss}^t \end{pmatrix} \begin{pmatrix} \delta x \\ \delta u \\ \delta s \end{pmatrix},
\]

where
with the derivatives being evaluated at \((x_t, u_t, s_t)\):
\[
\begin{align*}
Q^t_0 &= Q'(x_t, u_t, s_t),
Q^t_x &= \ell_x, 
Q^t_{ux} &= \ell_u, 
Q^t_\delta &= 0,
Q^t_x &= \ell_x + f^*_x V^{t+1},
Q^t_u &= \ell_u + f^*_u V^{t+1}, 
Q^t_\delta &= \ell_\delta + f^*_\delta V^{t+1},
\end{align*}
\]
(6)
where \(\cdot \) denotes tensor contraction along an appropriate dimension.

**Assumption 2.** Matrices \(Q^t_{uu}, \) defined in (6), are positive definite for all \(t \in \{0, \ldots, N-1\}\).

The aim is to construct a quadratic model of \(J^{N-t}_N(x)\) in the vicinity of \(x_t\) by providing suitable expressions for \(V^t(x)\), and obtain local improvements to the decision variables. To this aim, we consider a solution to
\[
\begin{align*}
\min_{du} \max_{s, \delta_u, \delta_x, \delta_s \geq 0} \delta Q^t(\delta x, \delta u, \delta s),
\end{align*}
\]
where this problem is a local approximation to
\[
\begin{align*}
\min_{u} \max_{s \geq 0} \left[ \ell(x, u, s) + V^{t+1}(f(x, u)) \right].
\end{align*}
\]
Local minimisation of (7) with respect to \(\delta u\) yields
\[
Q^t_u + Q^t_{uu} \delta x + Q^t_{ux} \delta u + Q^t_{u\delta} \delta s = 0,
\]
while the maximiser is addressed by setting the first-order expansion for the residual of the perturbed complementarity condition
\[
r_t := S_t c(x_t, u_t) + \mu \to 0,
\]
\[
rt + C_t \delta s + S_t c_x \delta x + S_t C_u \delta u = 0.
\]
The resulting parametric system of equations is given by
\[
\begin{align*}
\begin{bmatrix} Q^t_{uu} & Q^t_{us} \\
S_t Q^t_{su} & C_t \end{bmatrix} \begin{bmatrix} \delta u \\
\delta s \end{bmatrix} &= - \begin{bmatrix} Q^t_u \\
S_t Q^t_{sx} \end{bmatrix} \frac{\partial \ell}{\partial x} + \begin{bmatrix} Q^t_t \\
S_t Q^t_{tx} \end{bmatrix} \delta x.
\end{align*}
\]
(8)
The parametric solution of (8) as a function of \(\delta x\) is
\[
\begin{align*}
\begin{bmatrix} \delta u \\
\delta s \end{bmatrix} &= \begin{bmatrix} \alpha_t \\
\eta_t \end{bmatrix} + \begin{bmatrix} \beta_t \\
\theta_t \end{bmatrix} \delta x,
\end{align*}
\]
where the coefficients are given by
\[
\begin{align*}
\begin{bmatrix} \alpha_t & \beta_t \\
\eta_t & \theta_t \end{bmatrix} &= - \begin{bmatrix} Q^t_{uu} & Q^t_{us} \\
S_t Q^t_{su} & C_t \end{bmatrix}^{-1} \begin{bmatrix} Q^t_u \\
S_t Q^t_{sx} \end{bmatrix} = \begin{bmatrix} Q^t_{uu} & Q^t_{us} \\
S_t Q^t_{su} & C_t \end{bmatrix}^{-1} \begin{bmatrix} Q^t_u \\
S_t Q^t_{sx} \end{bmatrix}.
\end{align*}
\]
(10)
Note that (10) can be computed for \(t = N-1\) as \(V^N(x)\) is fully determined by function \(p(x)\). Next, the expressions for the coefficients of \(V^t(x)\) for \(t \in \{0, \ldots, N-1\}\) are
\[
\begin{align*}
V^0_t &= Q^0_t + \alpha_t^T \bar{Q}^t_{tu} + \frac{1}{2} \alpha_t^T \bar{Q}^t_{uu} \alpha_t,
V^t_x &= Q^t_x + \beta_t^T \bar{Q}^t_{tu} + (\bar{Q}^t_{ux} + \bar{Q}^t_{uu} \beta_t)^T \alpha_t,
V^t_{xx} &= \bar{Q}^t_{xx} + \beta_t^T \bar{Q}^t_{tu} + \beta_t^T \bar{Q}^t_{uu} \beta_t,
V^t_{ux} &= \bar{Q}^t_{ux} + \bar{Q}^t_{uu} \beta_t,
\end{align*}
\]
(11)
where
\[
\begin{align*}
\begin{bmatrix} Q^t_t & Q^t_{xx} \\
S_t Q^t_{sx} & C_t \end{bmatrix} &= \frac{1}{2} \begin{bmatrix} Q^t_{uu} & Q^t_{us} \\
S_t Q^t_{su} & C_t \end{bmatrix}^{-1},
\end{align*}
\]
(12)
\[
\begin{align*}
\bar{Q}^t_{tu} &= Q^t_{tu} - Q^t_{ux} C^{-1}_t r_t,
\bar{Q}^t_{uu} &= Q^t_{uu} - Q^t_{ux} C^{-1}_t C^{-1}_t S_t Q^t_{sx},
\bar{Q}^t_{ux} &= Q^t_{ux} - Q^t_{xx} C^{-1}_t S_t Q^t_{sx},
\bar{Q}^t_{xu} &= Q^t_{xu} - Q^t_{xx} C^{-1}_t S_t Q^t_{sx}.
\end{align*}
\]

**Remark 2.** Equivalently, system of equations (8) can be obtained by modifying coefficients \(Q^t_u\) and \(Q^t_{uu}\) in (6) as
\[
\begin{align*}
Q^t_u &= S^{-1}_t r_t, 
Q^t_{uu} &= S^{-1}_t C_t,
\end{align*}
\]
(13)
and solving
\[
\begin{align*}
\min_{\delta u} \max_{\delta s} \delta Q^t(\delta x, \delta u, \delta s),
\end{align*}
\]
with respect to both \(\delta u\) and (now unconstrained) \(\delta s\). The expressions in (11) are derived by considering
\[
V^t(x_t + \delta x) = \delta Q^t(\delta x, \alpha_t + \beta_t \delta x, \eta_t + \theta_t \delta x),
\]
while using modifications (13). Note that if just (6) is used it will give a different (and incorrect) result.

2) **Forward pass:** Define the update functions
\[
\begin{align*}
\phi_t(x) := u_t + \alpha_t + \beta_t (x - x_t),
\psi_t(x) := s_t + \eta_t + \theta_t (x - x_t),
\end{align*}
\]
(14)
and denote a new iterate by \((x^+, u^+, s^+)\), where
\[
\begin{align*}
x^+ &= [x_t^+, \ldots, x_{N-1}^+],
\end{align*}
\]
(11)
and
\[
\begin{align*}
u^+ &= [u_0^+, \ldots, u_{N-1}^+]
\end{align*}
\]
and
\[
\begin{align*}
s^+ &= [s_0^+, \ldots, s_{N-1}^+].
\end{align*}
\]
(15)

**Remark 3.** Note that the strict feasibility of a new iterate should be ensured by some means, i.e., the iterate should satisfy \(c(x^+, u^+) < 0\) and \(s^+_t > 0\) for \(t \in \{0, \ldots, N-1\}\). A common approach is to employ line-search procedure by including a step length \(\gamma \in [0; 1]\) in the definition of functions
\[
\begin{align*}
\phi_t(x, \gamma) &= u_t + \gamma \alpha_t + \beta_t (x - x_t),
\psi_t(x, \gamma) &= s_t + \gamma \eta_t + \theta_t (x - x_t),
\end{align*}
\]
B. **Properties of Feasible-Interior-Point DDP**

Let \(\mu\) be a strictly positive constant and denote \(w = [x, u, s]\), where \(x = [x_0, \ldots, x_N]\), \(u = [u_0, \ldots, u_{N-1}]\) and \(s = [s_0, \ldots, s_{N-1}]\) as before. Here we treat \(x_0\) as a given initial state, while all other states as functions of the initial state and the control inputs, i.e., \(x_t = f(x_{t-1}, u_{t-1})\) for \(t \in \{0, \ldots, N\}\).

**Definition 1.** We call \(w = [x, u, s]\) feasible if it satisfies \(c_t(x_t, u_t) \leq 0\) and \(s_t \geq 0\) for all \(t \in \{0, \ldots, N-1\}\), or strictly feasible if the inequalities are satisfied strictly.

**Lemma 1.** If \(w = [x, u, s]\) is strictly feasible then the linear operators
\[
P_t = \begin{bmatrix} Q^t_{uu} & Q^t_{us} \\
S_t Q^t_{su} & C_t \end{bmatrix}^{-1},
\]
(11)
are continuous in \(w\).

**Proof.** Note that \(Q^t_{uu}\) are positive definite by assumption and \(C_t\) are negative definite by strict feasibility of \(w\). Thus, matrices \(C_t\) and \(Q^t_{uu} = Q^t_{uu} C^{-1}_t S_t Q^t_{sx}\) are invertible. This is a sufficient condition for existence of \(P_t\), which can be found using a block-wise matrix inverse formula.
The matrix inverse (when exists) is continuous in \( w \) when its matrix components are continuous function of \( w \). As the continuity of \( Q_{uu}^{-1}, Q_{us}^{-1} \) and \( C_{N-1} \) follows directly from \( q(x,u), c(x,u) \) and \( p(x) \) being twice continuously differentiable, we readily conclude continuity of \( P_{N-1} \). Next, since \( f(x,u) \) is twice continuously differentiable, we establish continuity of \( Q_{NN}^{-2} \) and \( P_{NN-2} \). Following the recursion we can conclude continuity of \( P_t \) for \( N = 3 \) to \( t = 0 \). 

Now we show that \( \mathbf{(11)} \) can be further simplified. With this aim we resolve system \( \mathbf{(8)} \) for \( \delta s \)

\[
\delta s = -C_1^{-1}(r_t + S_t Q_{uu}^T \delta x + S_t Q_{us}^T \delta u),
\]

and eliminate it from the equations for \( \delta u \). Using hatted notations from \( \mathbf{(12)} \), this yields

\[
\hat{Q}_{uu}^T \delta u = -\hat{Q}_{uu}^T - \hat{Q}_{ux}^T \delta x.
\]

With the explicit expressions \( \alpha_t = -(\hat{Q}_{uu}^T)^{-1} \hat{Q}_{u}^T \) and \( \beta_t = -(\hat{Q}_{uu}^T)^{-1} \hat{Q}_{us}^T \) in \( \mathbf{(11)} \) finally we have

\[
V_0 = Q_0^T + \frac{1}{2} \alpha_t Q_0^T \alpha_t \quad V_x^T = \hat{Q}_{uu} + \frac{1}{2} \alpha_t \hat{Q}_{uu} \alpha_t \quad V_{xx} = \hat{Q}_{uu} + \frac{1}{2} \alpha_t \hat{Q}_{uu} \alpha_t,
\]

Next we consider a vector-valued function \( F(w, \mu) \) defined as

\[
F(w, \mu) := [Q_0^T, \ldots, Q_{N-1}^T, r_0, \ldots, r_{N-1}],
\]

where vectors \( Q_t \) are defined as in \( \mathbf{(3)} \) and \( r_t = S_t c(x_t, u_t) + \mu \). Note that, if \( w \) is a solution of \( F(w, \mu) = 0 \) for a given \( \mu > 0 \), then it is a stationary point of the IPDDP algorithm, since all \( \alpha_t \) and \( \eta_t \) in \( \mathbf{(10)} \) are zero in this case. Moreover, such \( w \) is a solution of the perturbed KKT system, as it is established in the following theorem.

**Theorem 1.** Let \( w = [x, u, s] \) be strictly feasible. If it is a solution of \( F(w, \mu) = 0 \) for a given \( \mu > 0 \), then there exists \( \lambda = [\lambda_0, \ldots, \lambda_N] \) such that \( (x, u, \lambda, s) \) satisfies \( \mathbf{(5)} \).

**Proof.** Note that satisfaction of \( c(x, u) \leq 0 \) and \( s \geq 0 \) in \( \mathbf{(3)} \) follows directly from the strict feasibility hypothesis. We also note that \( \nabla \lambda c(x, u, \lambda, s) = f(x_{t-1}, u_{t-1}) - x_t = 0 \) is satisfied for all \( t \in \{0, \ldots, N\} \) due to \( \mathbf{(11)} \).

Now assume \( w \) is a solution of \( F(w, \mu) = 0 \), meaning that \( Q_t^T = 0 \) and \( r_t = S_t c(x_t, u_t) + \mu = 0 \) for all \( t = 0, \ldots, N-1 \), and note that it ensures satisfaction of \( Sc(x, u) + \mu = 0 \) in \( \mathbf{(5)} \). Also note that \( \nabla \phi L(x, u, \lambda, s) = 0 \) is satisfied as soon as \( \lambda_N = V_x^N, \) where \( V_x^N \) is a derivative of \( p(x) \) evaluated at \( x_N \). Next, we show that the remaining equations in \( \mathbf{(3)} \) are satisfied, starting with \( t = N-1 \) and proceeding by recursion:

\[
Q_{xx}^{-1} = \ell_x + f^T x V_{xx}^{-1} \quad Q_{uu}^{-1} = e_u + f^T u V_{uu}^{-1}
\]

\[
\nabla_{x_{N-1}} - L = \ell_x + f^T \lambda_N - \lambda_{N-1} = Q_{xx}^{-1} - \lambda_{N-1},
\]

\[
\nabla_{u_{N-1}} - L = e_u + f^T u \lambda_N = Q_{uu}^{-1},
\]

where the derivatives are evaluated at \( (x_{N-1}, u_{N-1}, s_{N-1}) \), and we choose \( \lambda_{N-1} = Q_{xx}^{-1} = V_{xx}^{-1} \). Following this procedure, we construct \( \lambda = [\lambda_0, \ldots, \lambda_N] \), where \( \lambda_t = V_x^t \) for \( 0 = 1, \ldots, N \), such that \( (x, u, \lambda, s) \) is a (perturbed) KKT point, i.e., it satisfies system \( \mathbf{(5)} \).

Moreover, once \( w \) is sufficiently close to a solution of \( F(w, \mu) = 0 \), the IPDDP iterates converge to the solution at a quadratic rate. This is demonstrated in the following theorem.

**Theorem 2.** Let \( w^* = [x^*, u^*, s^*] \) be a feasible solution of \( F(w, \mu) = 0 \) for a given \( \mu > 0 \). Assume \( w = [x, u, s] \) and \( w^+ = [x^+, u^+, s^+] \), defined by \( \mathbf{(15)} \), are strictly feasible.

There exist \( M > 0 \) and \( \varepsilon > 0 \) such that if \( \|w - w^*\| \leq \varepsilon \), then

\[
\|w^+ - w^*\| = M\|w - w^*\|^2, \quad \|w^+ - w^*\| < \|w - w^*\|.
\]

**Proof.** Consider functions defined by \( Q_t^T \) and \( r_t \), and note that they are differentiable, as all the terms involved are differentiable. Then, by the Taylor theorem there exist functions \( h_t(w, w^*) \) and \( g_t(w, w^*) \) such that

\[
Q_t^T u_t = Q_t^T u + Q_{uu}^T (x_t^* - x_t) + Q_{us}^T (u_t^* - u_t) \quad + Q_{us}^T (s_t^* - s_t) + h_t(w, w^*) = 0,
\]

\[
r_t = r_t + S_t Q_{xx}^T (x_t^* - x_t) + S_t Q_{u}^T (u_t^* - u_t) \quad + C_t (s_t^* - s_t) + g_t(w, w^*) = 0,
\]

where we used that \( w^* \) is a solution of \( F(w, \mu) = 0 \), and the norms of the residual functions are bounded:

\[
\|h_t(w, w^*)\| \leq M\|w - w^*\|^2, \quad \|g_t(w, w^*)\| \leq M\|w - w^*\|^2,
\]

where \( h_t \) and \( g_t \) for \( t \in \{0, \ldots, N-1\} \) are some constants. Denote \( \Delta x_t = x_t^* - x_t, \Delta u_t = u_t^* - u_t, \Delta s_t = s_t^* - s_t \) for \( t \in \{0, \ldots, N-1\} \), and notice that \( (\Delta x_t, \Delta u_t, \Delta s_t) \) is a solution of system \( \mathbf{(8)} \), as it belongs to parametric family \( \mathbf{(9)} \):

\[
\Delta u_t = u_t^* - u_t = \phi(x_t^*) - u_t = \alpha_t + \beta_t \Delta x_t, \quad \Delta s_t = s_t^* - s_t = \psi(x_t^*) - s_t = \eta_t + \theta_t \Delta x_t.
\]

Now, by adding and subtracting \( x_t^+, u_t^+, s_t^+ \) in the appropriate parentheses of \( \mathbf{(17)} \), and using the fact that \( (\Delta x_t, \Delta u_t, \Delta s_t) \) is a solution of \( \mathbf{(8)} \), we have

\[
Q_{xx}^T (x_t^* - x_t^+) = Q_{uu}^T (u_t^* - u_t^+) + Q_{us}^T (s_t^* - s_t^+) \quad + h_t(w, w^*) = 0,
\]

\[
S_t Q_{xx}^T (x_t^* - x_t^+) + S_t Q_{u}^T (u_t^* - u_t^+) + C_t (s_t^* - s_t^+) \quad + g_t(w, w^*) = 0,
\]

which can be rewritten using operator \( P_t \) as

\[
\left[ u_t^* - u_t^+ \right] = -P_t \left[ h_t(w, w^*) \right] - P_t \left[ Q_{uu}^T (u_t^* - u_t^+) \right] + C_t (s_t^* - s_t^+) \quad + g_t(w, w^*) = 0.
\]

Since \( P_t \) is a continuous function of \( w \), and \( x_0^* = x_0 = x_0 \), for \( t = 0 \) this gives

\[
\|u_0^* - u_0^+, s_0^* - s_0^+\| \leq \|P_0\| \|h_0(w, w^*)\| \|g_0(w, w^*)\| \leq M_0 \|w - w^*\|^2,
\]

where \( M_0 = (H_0 + G_0) \max_{w \in \Omega} \|P_0\|, \) and the operator norm \( \|P_0\| \) (induced by Euclidian vector norm) of continuous linear operator \( P_0 \) is bounded on a compact subset \( \Omega \) of all strictly feasible \( w \), which can be chosen to be sufficiently big, i.e.,
such that it contains set of all strictly feasible \( w : \| w - w^* \| \leq \varepsilon \) for any \( \varepsilon \leq \bar{\varepsilon} \), where \( \bar{\varepsilon} > 0 \) is a constant. By using Taylor formula for \( f(x_0^*, u_0^*) \), and bounding the residuals in \( \| w - w^* \| \leq \varepsilon \) we get:

\[
\| x_1^* - x_1^+ \| = \| f(x_0^*, u_0^*) - f(x_1^*, u_1^*) \| \leq \| f'(x_0^*)^T (x_1^* - x_0^+) + f''_{uu}(u_0^*) (u_1^* - u_0^+) \| + K_1 \| x_1^* - x_0^+, u_1^* - u_0^+ \|^2 \leq K_2 \| x_1^* - x_0^+, u_1^* - u_0^+ \| \leq K_3 \| w - w^* \|^2,
\]

for some constants \( K_1, K_2, K_3 \geq 0 \). Continuing by induction, there are some constants \( M_t \geq 0 \) for \( t \in \{0, \ldots, N - 1\} \), such that for a positive \( \varepsilon \leq \bar{\varepsilon} \) and all strictly feasible \( w : \| w - w^* \| \leq \varepsilon \) we have:

\[
\| x_t^* - x_t^+ \|, u_t^* - u_t^+, s_t^* - s_t^+ \| \leq M_t \| w - w^* \| ^2,
\]

which proves the first inequality in (16) by choosing \( M = M_1 + M_2 + \ldots + M_N \):

\[
\| w^* - w^+ \| \leq \sum_{t=1}^{N} \| x_t^* - x_t^+, u_t^* - u_t^+, s_t^* - s_t^+ \| \leq M \| w - w^* \| ^2.
\]

The second inequality in (16) is obtained from the first one by pickings \( \varepsilon \) sufficiently small, i.e., such that \( \varepsilon < 1/M \) in addition to \( \varepsilon \leq \bar{\varepsilon} \). Now, for all strictly feasible \( w \) such that \( \| w - w^* \| \leq \varepsilon \) we additionally have:

\[
\| w^+ - w^* \| \leq M \| w - w^* \| ^2 \leq \varepsilon M \| w - w^* \| < \| w - w^* \|.
\]

C. Primal-Dual Infeasible-Interior-Point DDP

In the previous sections we assumed that one has access to a feasible initial solution guess, which might be quite restrictive in practice. Here we propose an extension, where feasibility requirement in the sense of constraints satisfaction can be avoided, but still ensured in the limit. In this section we explain what should be modified in the IPDDP algorithm, and how the theoretical results translate to a new algorithm.

1) Backward pass: We use the same definitions for \( V^t(x) \) and \( Q^t(x) \), and follow the same steps to obtain the first equation of system (8). Next, however, we introduce slack variables \( y_t \geq 0 \) for \( t \in \{0, \ldots, N - 1\} \) to transform the inequality constraints into equalities

\[
c(x_t, u_t) + y_t = 0,
\]

and use the perturbed complementarity condition, formulated in terms of the dual and slack variables

\[
S_t y_t - \mu = 0.
\]

Finally, we proceed by setting the first order Taylor expansions for \( c(x_t, u_t) + y_t \) and \( S_t y_t - \mu \) to zero. In this way, the parametric system of equations becomes

\[
\begin{bmatrix}
Q_{uu}^t & Q_{us}^t & 0 \\
Q_{su}^t & 0 & I \\
0 & Y_t & S_t
\end{bmatrix}
\begin{bmatrix}
\delta u \\
\delta s \\
\delta y
\end{bmatrix} = - \begin{bmatrix}
Q_u^t \\
0 \\
0
\end{bmatrix}
\begin{bmatrix}
\gamma_u^t \\
\gamma_s^t \\
\gamma_d^t
\end{bmatrix} \delta x, \tag{18}
\]

where \( r_u^t := c(x_t, u_t) + y_t \) and \( r_d^t := S_t y_t - \mu \) are the primal and dual residuals correspondingly, and \( Y_t := \text{diag}([y_t]) \). Note that elimination of \( dy \) leads to a reduced system similar to (3)

\[
\begin{bmatrix}
Q_{uu}^t & Q_{us}^t \\
Q_{su}^t & -Y_t
\end{bmatrix}
\begin{bmatrix}
\delta u \\
\delta s
\end{bmatrix} = - \begin{bmatrix}
Q_u^t \\
Y_t
\end{bmatrix} \begin{bmatrix}
\gamma_u^t \\
\gamma_s^t
\end{bmatrix} \delta x,
\]

with \( \tilde{r}_t := S_t r_u^t - r_d^t \), and an extra equation for slack variables:

\( \delta y = -r_u^t - Q_{su} \delta u - Q_{sx} \delta x \). This observation allows to skip repeating derivation and modify expressions in (13) directly.

Now we solve the parametric system by computing

\[
\begin{bmatrix}
\alpha_t & \beta_t \\
\eta_t & \theta_t
\end{bmatrix} = - \begin{bmatrix}
Q_{uu}^t & Q_{us}^t \\
Q_{su}^t & 0
\end{bmatrix}^{-1} \begin{bmatrix}
Q_u^t \\
0
\end{bmatrix} \begin{bmatrix}
\gamma_u^t \\
\gamma_s^t
\end{bmatrix}
\begin{bmatrix}
r_u^t \\
r_d^t
\end{bmatrix},
\]

and establish new coefficients

\[
Q_u^t = Q_{uu}^t + Q_{us}^t Y_t^{-1} \tilde{r}_t,
\]

\[
Q_u^t = Q_{uu}^t + Q_{us}^t Y_t^{-1} \tilde{r}_t,
\]

\[
Q_{uu}^t = Q_{uu}^t + Q_{us}^t Y_t^{-1} S_t Q_{uu}^t,
\]

\[
Q_{xx}^t = Q_{xx}^t + Q_{xs}^t Y_t^{-1} S_t Q_{xx}^t,
\]

\[
Q_{uu}^t = Q_{uu}^t + Q_{us}^t Y_t^{-1} S_t Q_{uu}^t,
\]

2) Forward pass: Define the update functions

\[
\phi_t(x) = u_t + \alpha_t + \beta_t (x - x_t),
\]

\[
\psi_t(x) = s_t + \eta_t + \theta_t (x - x_t),
\]

\[
\xi_t(x) = y_t + \chi_t + \zeta_t (x - x_t),
\]

and denote a new solution guess by \( (x^+, u^+, s^+, y^+) \), where \( x^+ = [x_0^+ \ldots x_N^+] \), \( u^+ = [u_0^+ \ldots u_{N-1}^+] \), \( s^+ = [s_0^+ \ldots s_{N-1}^+] \) and \( y^+ = [y_0^+ \ldots y_{N-1}^+] \). Let \( x_0^+ = x_0 \) and for \( t \in \{0, \ldots, N - 1\} \) we compute

\[
\begin{align*}
 u_t^+ &= \phi_t(x_t^+), \\
 s_t^+ &= \psi_t(x_t^+), \\
 y_t^+ &= \xi_t(x_t^+), \\
 x_{t+1}^+ &= f(x_t^+, u_t^+).
\end{align*}
\]

Remark 4. Note that now only positivity of \( s_t^+ \) and \( y_t^+ \) must be ensured, while inequality constraints \( c(x_t^+, u_t^+) \leq 0 \) will be satisfied upon convergence. This potentially facilitates algorithm implementation.

D. Properties of Infeasible-Interior-Point DDP

Now we add slack variables \( y = [y_0, \ldots, y_{N-1}] \) into the iteration vector \( w = [x, u, s, y] \), and modify the definition of iteration feasibility to mean the case when \( s \geq 0 \) and \( y \geq 0 \) (and strict feasibility with strict inequalities). Consider a vector-valued function \( F(w, \mu) \) defined as

\[
F(w, \mu) := [Q_u^0, \ldots, Q_u^{N-1}, r_u^0, \ldots, r_u^{N-1}, r_d^0, \ldots, r_d^{N-1}],
\]

where \( r_u^t := c(x_t, u_t) + y_t \) and \( r_d^t := S_t y_t - \mu \).

Lemma 2. If \( w = [x, u, s, y] \) is strictly feasible then the linear operators

\[
P_t = \begin{bmatrix}
Q_{uu}^t & Q_{us}^t & 0 \\
Q_{su}^t & 0 & I \\
0 & Y_t & S_t
\end{bmatrix}^{-1}, \quad t \in \{0, \ldots, N - 1\},
\]
are continuous in \(w\).

Proof. The proof is similar to that of Lemma 1.

Theorem 3. Let \(w = [x, u, s, y]\) be strictly feasible. If it is a solution of \(F(w, \mu) = 0\) for a given \(\mu > 0\), then there exists \(\lambda = [\lambda_1, \ldots, \lambda_N]\) such that \((x, u, \lambda, s)\) satisfies (3).

Proof. To show validity of the result it is enough to notice that \(r^d_l = c(x_l, u_l) + y_l = 0\) ensures \(Y_l = -C_l\) for \(t \in \{0, \ldots, N\}\). With this in mind one can follow the same steps as in the proof of Theorem 1.

Theorem 4. Let \(w^* = [x^*, u^*, s^*, y^*]\) be a feasible solution of \(F(w, \mu) = 0\) for a given \(\mu > 0\). Assume \(w = [x, u, s, y]\) and \(w^+ = [x^+, u^+, s^+, y^+]\), defined by (19), are strictly feasible. There exist \(M \geq 0\) and \(\epsilon > 0\) such that if \(\|w - w^*\| \leq \epsilon\), then

\[
\|w^+ - w^*\| \leq M\|w - w^*\|^2, \\
\|w^+ - w^*\| < \|w - w^*\|.
\]

Proof. The proof follows the same steps as the proof of Theorem 1. We fix \(\mu > 0\) and consider differentiable functions defined by \(Q_{ux}(x, u, s, y)\) and \(r^d_l\), and use its first-order Taylor expansion at \(w^*:\)

\[
Q^l_{ux}(x) = Q^l_{ux}(x^* - x_l) + Q^l_{uu}(u^* - u_l) + Q^l_{us}(s^* - s_l) + h_l(w, w^*) = 0,
\]

\[
r^d_l = r^d_l + Q^l_{sx}(s^* - s_l) + Q^l_{su}(u^* - u_l) + (y^* - y_l) + g_l(w, w^*) = 0,
\]

and the norms of the residuals are bounded

\[
\|h_l(w, w^*)\| \leq H_l\|w - w^*\|^2, \\
\|g_l(w, w^*)\| \leq G_l\|w - w^*\|^2, \\
\|k_l(w, w^*)\| \leq K_l\|w - w^*\|^2,
\]

where \(H_l, G_l, K_l\) for \(t \in \{0, \ldots, N - 1\}\) are some constants. As before, we use \(\Delta x_l = x^+_l - x^*_l, \Delta u_l = u^+_l - u^*_l, \Delta s_l = s^+_l - s^*_l, \Delta y_l = y^+_l - y^*_l\) for \(t \in \{0, \ldots, N\}\), and note that \((\Delta x_l, \Delta u_l, \Delta s_l, \Delta y_l)\) is a solution of system (18). Thus,

\[
Q^l_{ux}(x_l^* - x_l^+) + Q^l_{uu}(u_l^* - u_l^+) + Q^l_{us}(s_l^* - s_l^+) + h_l(w, w^*) = 0,
\]

\[
Q^l_{sx}(s_l^* - s_l^+) + Q^l_{su}(u_l^* - u_l^+) + (y_l^* - y_l^+) + g_l(w, w^*) = 0,
\]

\[
Y_l(s_l^* - s_l^+) + S_l(y_l^* - y_l^+) + k_l(w, w^*) = 0,
\]

and using the new definition of operators \(P^l:\)

\[
\begin{pmatrix}
    u_l - u_l^+ \\
    s_l - s_l^+ \\
    y_l - y_l^+
\end{pmatrix} = -P^l \begin{pmatrix} h_l(w, w^*) \\ g_l(w, w^*) \\ k_l(w, w^*) \end{pmatrix} - P^l \begin{pmatrix} Q^l_{ux} \\ Q^l_{sx} \end{pmatrix} (x_l^* - x_l^+).
\]

From here the remaining proof is straightforward: we start with \(t = 0\) and proceed iteratively to \(t = N - 1\) as it is described in the proof of Theorem 1.

IV. Numerical simulations

Here we compare implementations of the proposed Interior-Point DDP algorithms (IPDDP) [19] with an implementation of Control-Limited Differential Dynamic Programming (CLDDP) [20].

The wall-clock time (physical time) required for solving problems using these methods depends mainly on the number of iterations and time necessary to complete each iteration. While we focus on the total number of iterations required for algorithms to converge to a solution, we briefly note that the per-iteration complexity for each algorithm, due to their nature, is different. Specifically, CLDDP require solving \(N\) box-QPs (of size \(m + l\) constraints) per iteration, while IPDDP require solving \(N\) linear systems of equations (of size \(m + l\)) per iteration. Thus, one expects that each iteration of the CLDDP takes longer than that of the IPDDP variants. However, a rigorous and fair wall-clock time comparison between the two families of methods is beyond the scope of this paper.

To ensure global convergence of the proposed algorithms we rely on three main ingredients: line-search (see Remark 3), steps filter [21] and regularisation of type \(Q_{ux}^\mu + \gamma I\). Perturbation \(\mu\) is initialised as \(\mu \rightarrow \min(0.2\mu, \mu^1 2)\) every time \(\|F(w, \mu)\|_\infty\) is less than \(0.2\mu\). In all simulations we use random initial solution guesses, which are obtained by uniform sampling from interval \([-0.01, 0.01]\), and conduct 50 trials for each experiment.

We use the following definition for the logarithmic cost error

\[
E_J := \log_{10} \left[ J(x, u) - J(x^*, u^*) \right],
\]

where \(J(\cdot, \cdot)\) is the objective function evaluated at a current solution guess \((x, u)\) or at a (numerically obtained) locally optimal solution \((x^*, u^*)\).

A. Inverted pendulum

Consider a task of stabilising the inverted pendulum with state \(x = [\phi, \omega]\), where \(\phi\) is angle and \(\omega\) is angular velocity, and control input \(u\):

\[
f(x, u) = \begin{bmatrix} \phi + h\omega \\ \omega + h\sin(\phi) + hu \end{bmatrix},
\]

where \(h = 0.05\) is a time step; initial state is \(x_0 = [-\pi; 0]\), control constraints are \(-0.25 \leq u \leq 0.25\), and \(N = 500\). We choose the stage and terminal cost functions as

\[
q(x, u) = \frac{h}{2}(\phi^2 + \omega^2 + u^2), p(x) = \frac{10}{2}(\phi^2 + \omega^2).
\]

Fig. 1 shows that both Feasible- and Infeasible-IPDDP consistently demonstrate rapid convergence to the optimal solution with the total number of iterations being less that 200 for all trials. On the other hand, most runs of Control-Limited DDP (CLDDP) take about or more than 200 iterations to converge (here we limited the maximum number of iteration to 300 and dropped all unsuccessful trials).
Given by 

\[ \text{solve constrained QPs complicates the correct identification of } \]

\[ \text{bang nature of the optimal control and the need to (exactly)} \]

\[ \text{proaches zero. We believe that the interplay between the bang-} \]

\[ \text{active and inactive constraints for CLDPP algorithm. This po-} \]

\[ \text{tential issue is mitigated for IPDDP algorithms by sufficien} \]

\[ \text{tly perturbing the problem.} \]

B. Car parking problem

Next we consider a car parking problem as in [16]. The dy-

\[ \text{namic of a car with four-dimensional state } x = [r_x, r_y, \varphi, v] \]

\[ \text{(r_x and r_y are the x- and y-coordinates, \varphi is the car’s heading and v is its velocity) and input } u = [w, a] (w and a are the} \]

\[ \text{front wheels’ steering angle and acceleration respectively) is} \]

\[ \text{given by} \]

\[ f(x, u) = \begin{bmatrix} r_x + b(v, w) \cos(\varphi) \\
 r_y + b(v, w) \sin(\varphi) \\
 \varphi + \sin^{-1}(h_d v \sin(w)) \\
 v + ha \end{bmatrix}, \]

where \( b(v, w) = d + hv \cos(\omega) - \sqrt{d^2 - h^2 v^2 \sin^2(w)}, d \) is a constant (distance between the front and back axles of a car), \( h_d = \frac{h}{d} \) and \( h = 0.03 \) is a time step. The cost functions are chosen as

\[ q(x, u) = 0.01(H(r_x, 0.1) + H(r_y, 0.1) + w^2 + 0.01a^2), \]

\[ p(x) = H(r_x, 0.1) + H(r_y, 0.1) + H(\varphi, 0.01) + H(v, 0.1), \]

where \( H(y, z) = \sqrt{y^2 + z^2} - z; \) initial state is \( x_0 = [1; 1; 3\pi/2; 0], \) control constraints are \(-0.5 \leq w \leq 0.5 \) and \(-2 \leq a \leq 2, \) and \( N = 500. \)

In many cases CLDPP converges faster than IPDDP (see Fig. 3), but presence of very distinct locally optimal solutions greatly influences its performance, while IPDDP algorithms follow the central path, i.e., a trace of stationary points defined for different values of perturbations, and thus behave in a more predictable manner. After we apply state constraints \(-2 \leq r_x \leq 2 \) and \(-2 \leq r_y \leq 2 \) in addition to the existing input constraints the problem becomes unsolvable with CLDPP. However, IPDDP algorithms are shown to solve the new problem with a relatively small increase in the number of iterations (see Fig. 4).
C. Planar 3-link manipulator

Finally we consider a 3-link planar manipulator (with unit length for all links) modeled as three integrators:

\[ f(x, u) = x + hu, \]

where state vector \( x = [\varphi_1, \varphi_2, \varphi_3] \) consists of three joint’s angles, input vector \( u = [\omega_1, \omega_2, \omega_3] \) is commanded angular velocities, and \( h = 0.1 \) is a time step. We set initial state \( x_0 = [-\pi/2; 0; 0] \), \( N = 500 \), and let

\[
q(x, u) = \frac{h}{2} u^T u, \quad p(x) = \frac{100}{2} (\varphi_1 - \pi/2)^2 + 10 \varphi_2^2 + 1 \varphi_3^2.
\]

We use linear constraints on input \(-0.1 \leq u \leq 0.1\), and nonlinear (in decision variables) constraints on y-coordinates of the joint preceding the end-effector \( r_{y2} = \cos(\varphi_1) + \cos(\varphi_1 + \varphi_2) \) and the end-effector itself \( r_{y3} = r_{y2} + \cos(\varphi_1 + \varphi_2 + \varphi_3) \):

\[-1 \leq r_{y2} \leq 1 \quad \text{and} \quad -1 \leq r_{y3} \leq 1.\]

Note that IPDDP requires the second-order derivatives of the state constraint, and without this information the quadratic convergence cannot be ensured. Since CLDDP cannot handle state constraints we can proceed only with IPDDP algorithms, which have similar performance on this problem (see Fig 5).

V. CONCLUSIONS

Two variants of Primal-Dual Interior-Point DDP algorithms, namely Feasible- and Infeasible-IPDDP have been proposed in this paper. The former operates with strictly feasible iterations, while the latter allows to start with infeasible solution guess, and ensures constraint satisfaction upon convergence. We prove the local quadratic convergence of the algorithms, and show that stationary points of both algorithms satisfy the (perturbed) system of first-order conditions for optimality. The numerical simulations with a globally convergent implementation (without proofs provided) on three examples demonstrate ability of the proposed algorithms to converge at superlinear rate, often being superior to its closest competitor (Control-Limited DDP) in the total number (and variability) of iterations required to converge.

We conjecture that the proposed algorithms have global convergence properties. A future research direction is to prove this conjecture. Also, we believe that the numerical performance can be further improved by implementing adaptive schemes for selecting the perturbation parameter. Other promising extensions include generalisation to manifolds, and constrained game-theoretic scenarios.

REFERENCES

[1] D. Mayne, “A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems,” International Journal of Control, vol. 3, no. 1, pp. 85–95, 1966.
[2] J. F. A. De O. Pantoja, “Differential dynamic programming and newton’s method,” International Journal of Control, vol. 47, no. 5, pp. 1539–1553, 1988.
[3] L.-Z. Liao and C. A. Shoemaker, “Convergence in unconstrained discrete-time differential dynamic programming,” IEEE Transactions on Automatic Control, vol. 36, no. 6, pp. 692–706, 1991.
[4] J. Morimoto, G. Zeglin, and C. G. Atkeson, “Minimax differential dynamic programming: Application to a biped walking robot,” in Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) Cat. No. 03CH37453), vol. 2. IEEE, 2003, pp. 1927–1932.
[5] W. Sun, E. A. Theodorou, and P. Tsiotras, “Game theoretic continuous time differential dynamic programming,” in 2015 American Control Conference (ACC). IEEE, 2015, pp. 5593–5598.
[6] Y. Tassa, T. Erez, and W. D. Smart, “Receding horizon differential dynamic programming,” in Advances in neural information processing systems, 2008, pp. 1465–1472.
[7] W. Li and E. Todorov, “Iterative linear quadratic regulator design for nonlinear biological movement systems,” in ICINCO (1), 2004, pp. 222–229.
[8] S. Sen and S. J. Yakowitz, “A quasi-Newton differential dynamic programming algorithm for discrete-time optimal control,” Automatica, vol. 23, no. 6, pp. 749–752, 1987.
[9] J. Rajamäki, K. Naderi, V. Kyrki, and P. Hamäläinen, “Sampled differential dynamic programming,” in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016, pp. 1402–1409.
[10] Z. Manchester and S. Kuijpersma, “Derivative-free trajectory optimization with unscented dynamic programming,” in 2016 IEEE 55th Conference on Decision and Control (CDC). IEEE, 2016, pp. 3642–3647.
[11] E. Pellegrini and R. P. Russell, “Applications of the multiple-shooting differential dynamic programming algorithm with path and terminal constraints,” in AAS/AIAA Astrodynamics Specialist Conference, Stevenson, WA, 2017.
[12] B. Plancher, Z. Manchester, and S. Kuijpersma, “Constrained unscented dynamic programming,” in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017, pp. 5674–5680.
[13] D. M. Murray and S. J. Yakowitz, “Constrained differential dynamic programming and its application to multireservoir control,” Water Resources Research, vol. 15, no. 5, pp. 1017–1027, 1979.
[14] T. Lin and J. Arora, “Differential dynamic programming technique for constrained optimal control,” Computational Mechanics, vol. 9, no. 1, pp. 27–40, 1991.
[15] Z. Xie, C. K. Liu, and K. Hauser, “Differential dynamic programming with nonlinear constraints,” in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 695–702.
[16] Y. Tassa, N. Mansard, and E. Todorov, “Control-limited differential dynamic programming,” in 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014, pp. 1168–1175.
[17] G. Lantoine and R. P. Russell, “A hybrid differential dynamic programming algorithm for constrained optimal control problems. part 1: theory,” Journal of Optimization Theory and Applications, vol. 154, no. 2, pp. 382–417, 2012.
[18] A. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang, “On the formulation and theory of the newton interior-point method for nonlinear programming,” Journal of Optimization Theory and Applications, vol. 89, no. 3, pp. 507–541, 1996.
[19] A. Pavlov, “Interior point differential dynamic programming repository,” 2020, Available at: https://github.com/xapavlov/odpp.
[20] Y. Tassa, “iLQG/DDP trajectory optimization,” 2015, Available at: https://www.mathworks.com/matlabcentral/fileexchange/52069-ilqg-ddp-trajectory-optimization.
[21] R. Fletcher and S. Leyffer, “Nonlinear programming without a penalty function,” Mathematical programming, vol. 91, no. 2, pp. 239–269, 2002.