`proxddp`: a primal-dual solver for constrained trajectory optimization

Journées Nationales de la Robotique Humanoïde, 2022

Wilson Jallet[a,b,*], Antoine Bambade[b,c], Nicolas Mansard[a] and Justin Carpentier[b]

***Abstract*— Trajectory optimization are problems are a fundamental component of solving optimal control problems (OCPs) on high-dimensional, complex robotic systems. It relies on two key components: 1/ the transcription into a sparse nonlinear program, and 2/ an appropriate algorithm which iteratively computes a solution within a desired convergence threshold. On one hand, differential dynamic programming (DDP) [1] is an efficient framework for transcription and resolution of OCPs as a finite-dimensional nonlinear program, which fully exploits the sparsity structure induced by time. On the other hand, augmented Lagrangian methods [2]–[4] provide a framework for efficient algorithms that allow advanced constraint-satisfaction strategies. In the sequel, we discuss our work on building efficient algorithms for constrained trajectory optimization, based on DDP and augmented Lagrangian methods, as well as our upcoming `C++` package for modelling and solving such problems. The features of our methods include: targeting problems with both equality and inequality constraints, being able to solve them within good accuracy, and being able to handle multiple-shooting formulations naturally (including the use of advanced implicit integrators e.g. variational integrators [5]).**

I. Introduction

In recent work [6]–[8], new variants of DDP were introduced; firstly for equality-constrained problems [6], [7] and then with multiple-shooting capabilities for use with implicit integrators (e.g. variational [5]) [7], based on augmented Lagrangians. Then, further extensions to our framework were made to include inequality-constrained problems [8], based on a novel primal-dual augmented Lagrangian formulation for constrained programming which has successfully been used in quadratic programming [9].

a) Problem setting: A typical trajectory optimization problem in continuous time can be formulated as the following infinite-dimensional problem:

$$\begin{aligned} \min_{x,u} \ & \int_0^T \ell(t, x(t), u(t)) \, \mathrm{d}t + \ell_{\mathrm{f}}(x(T)) \\ \text{s.t. } & \dot{x}(t) = f(x(t), u(t)), \ x(0) = \bar{x}_0 \\ & h(t, x(t), u(t)) \leqslant 0 \text{ (path constraints)} \\ & h_T(x(T)) \leqslant 0 \end{aligned} \tag{1}$$

[a] LAAS-CNRS, 7 Avenue du Colonel Roche, F-31400 Toulouse, France
[b] Inria, D.I. ENS, CNRS, PSL Research University, Paris, France
[c] ENPC, France, *corresponding author: wjallet@laas.fr

This work was supported in part by the HPC resources from GENCI-IDRIS (Grant AD011011342), the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute) and ANR-19- P3IA-000 (ANITI 3IA Institute), Louis Vuitton ENS Chair on Artificial Intelligence, and the European project MEMMO (Grant 780684).

b) Transcription: In [7], we argue that for stiff systems (involving contact dynamics, for instance), it is desirable to get increased numerical stability in integration rules, which can be achieved by leveraging *implicit* numerical integrators (such as the midpoint rule, the class of implicit Runge-Kutta methods, or variational integrators like [5]). In general, assuming the discrete sequence of states and controls are $\{x_k\}_{k=0}^{N}, \{u_k\}_{k=0}^{N-1}$ respectively, an implicit integration rule can be written in the form $f^{\mathrm{d}}(x_k, u_k, x_{k+1}) = 0$ where the superscript "d" means "discretised" (it will dropped in the sequel). The equation above can be solved (using Newton-Raphson for instance) to obtain the next state x_{k+1} from (x_k, u_k). Some authors such as [10] solve for x_{k+1} and apply the implicit function theorem to obtain the derivatives of the mapping $(x_k, u_k) \mapsto x_{k+1}$, within a standard single-shooting framework. In our approach, outlined below, we go for a multiple-shooting formulation where dynamical feasibility is only asked for at convergence of the algorithm.

The corresponding transcription of the OCP (1), including the path constraints, reads:

$$\min_{\mathbf{x},\mathbf{u}} \ J(\mathbf{x}, \mathbf{u}) = \sum_{k=0}^{N-1} \ell_k(x_k, u_k) + \ell_{\mathrm{f}}(x_N) \tag{2a}$$

$$\text{s.t. } f^{\mathrm{d}}(x_k, u_k, x_{k+1}) = 0, \ x_0 = \bar{x}_0 \tag{2b}$$

$$h_k(x_k, u_k) \leqslant 0 \tag{2c}$$

$$h_N(x_N) \leqslant 0. \tag{2d}$$

II. Proposed method

a) Backward sweep and Bellman equation: Our main idea is to consider the dynamic programming principle or *Bellman equation* for (2), which reads

$$\begin{aligned} V_i(x) = \min_{u,y} \ & \ell(x, u) + V_{i+1}(y) \\ & f(x, u, y) = 0, \ h(x, u) \leqslant 0 \end{aligned} \tag{3}$$

with boundary condition $V_N(x) = \ell_{\mathrm{f}}(x)$. In the sequel, we introduce the Q-function associated with the Lagrangian of (3), where (λ, ν) are the dual variables:

$$Q(x, u, y, \lambda, \nu) = \ell(x, u) + V_{i+1}(y) + \lambda^\top f(x, u, y) + \nu^\top h(x, u). \tag{4}$$

The Bellman equation then simply becomes the min-max problem

$$V_i(x) = \min_{u,y} \max_{\lambda, \nu : \nu \geqslant 0} Q(x, u, y, \lambda, \nu). \tag{5}$$

This Bellman equation can be relaxed by using the classical augmented Lagrangian method or the primal-dual augmented Lagrangian method of Gill & Robinson [11], extended to inequality constraints in [8] and [9]. This relaxation can be seen as a proximal-point iteration on the dual problem to (3), or in the dual variables in (5), as studied in [4].

We start with the (*primal*) augmented Lagrangian method, that of Powell, Hestenes and Rockafellar [4]. At step k of the overall algorithm, the Bellman iteration now depends on external multiplier estimates λ_k, ν_k for the constraints,

$$\min_{u,y} \ell(x,u) + V_{i+1}(y) + \frac{1}{2\mu} \left\| \begin{bmatrix} f(x,u,y) + \mu\lambda_k \\ [h(x,u) + \mu\nu_k]_+ \end{bmatrix} \right\|^2 . \quad (6)$$

The minimand is the augmented Lagrangian $\mathcal{L}_k(u,y;x,\mu)$ associated with the Bellman equation (3) which depends on the current state x. This Bellman equation, just like in standard DDP, is handled by a single (quasi)-Newton step – by the dynamic programming principle, this backward sweep recovers a Newton step over the entire state-control trajectory $(\mathbf{x},\mathbf{u})$, with respect to the *augmented Lagrangian of the initial problem (2)*.

For the *primal-dual* augmented Lagrangian, we consider a primal-dual step which includes both (u,y) and multipliers (λ,ν).

$$\min_{u,y,\lambda,\nu} \mathcal{L}_k(u,y;x,\mu) + \frac{1}{2\mu} \left\| \begin{bmatrix} f(x,u,y) + \mu(\lambda_k - \lambda) \\ [h(x,u) + \mu\nu_k]_+ - \mu\nu \end{bmatrix} \right\|^2 . \quad (7)$$

b) The KKT system and feedback gains: To compute the Newton or quasi-Newton step $(\delta u, \delta y, \delta\lambda, \delta\nu)$ in the primal and dual variables in either (6) or (7), the following system can be solved (using e.g. an indefinite Cholesky factorization):

$$\mathcal{K}_\mu \begin{bmatrix} \delta u \\ \delta y \\ \delta\lambda \\ \delta\nu \end{bmatrix} = - \begin{bmatrix} Q_u + Q_{ux}\delta x \\ Q_y + Q_{yx}\delta x \\ f + f_x\delta x + \mu(\lambda_l - \lambda) \\ [h + h_x\delta x + \mu\nu_l]_+ - \mu\nu \end{bmatrix}, \quad (8)$$

where the matrix $\mathcal{K}_\mu$ is the "KKT matrix" of the backward sweep is

$$\mathcal{K}_\mu \overset{\text{def}}{=} \begin{bmatrix} Q_{uu} & Q_{uy} & f_u^\top & S_\mathcal{A} h_u^\top \\ Q_{yu} & Q_{yy} & f_y^\top & S_\mathcal{A} h_y^\top \\ f_u & f_y & -\mu I & \\ S_\mathcal{A} h_u & S_\mathcal{A} h_y & & -\mu S_\mathcal{A} \end{bmatrix}, \quad (9)$$

and $S_\mathcal{A}$ is simply a selection matrix for the active set $\mathcal{A}$ of constraints in the augmented Lagrangian method (the indices j such as $[h(x,u) + \mu\nu_k]_j \geqslant 0$, see [9] for further details): $S_\mathcal{A}$ is diagonal with $[S_\mathcal{A}]_{jj} = 1$ if the constraint is active ($j \in \mathcal{A}$), 0 otherwise.

III. Experimental showcase

The aim of our work is to go towards a full-fledged implementation of our method, named ProxDDP, in C++ for fast and efficient resolution of constrained trajectory optimization problems, as a successor to the Crocoddyl [12] software library. For this talk, we have included two experiments from previous works, figures 1 and 2, which show satisfactory behaviour of the solutions to some constrained problems.

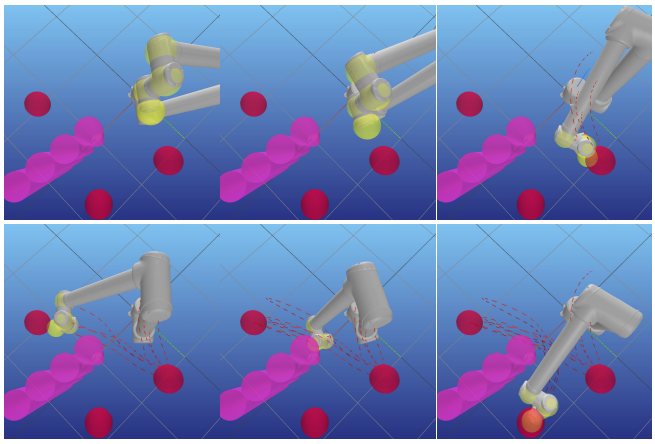

Fig. 1. UR10 reach task. The yellow spheres around the end-effector and wrist links do not collide with the purple cylinders, and the waypoints are reached at the specified times. Joint and torque limits (specified by the robot URDF file) have been imposed. The no-collision constraints are described using a closed-form expression for the distance between a sphere and a cylinder.

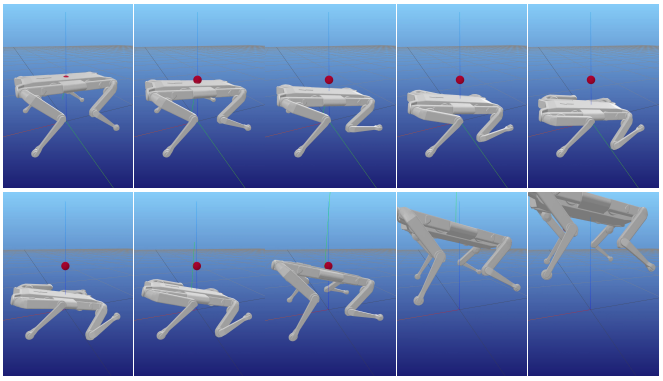

Fig. 2. Illustration of the Solo-12 robot performing a jump using variational integrator discrete dynamics from [5], at a time-step of $\Delta t = 30\,\text{ms}$. These dynamics include no Baumgarte stabilization term and work to enforce exactly the bilateral contact constraints of the form $\phi(q) = 0$.

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, Jan. 1966.

[2] M. R. Hestenes, "Multiplier and gradient methods," *Journal of Optimization Theory and Applications*, vol. 4, no. 5, Nov. 1969.

[3] M. J. D. Powell, "Algorithms for nonlinear constraints that use lagrangian functions," *Mathematical Programming*, vol. 14, no. 1, Dec. 1978.

[4] R. T. Rockafellar, "Augmented Lagrangians and Applications of the Proximal Point Algorithm in Convex Programming," *Mathematics of Operations Research*, vol. 1, no. 2, 1976.

[5] E. R. Johnson and T. D. Murphey, "Scalable Variational Integrators for Constrained Mechanical Systems in Generalized Coordinates," *IEEE Transactions on Robotics*, vol. 25, no. 6, Dec. 2009.

[6] S. Kazdadi, J. Carpentier, and J. Ponce, "Equality Constrained Differential Dynamic Programming," in *ICRA 2021 - IEEE International Conference on Robotics and Automation*, May 2021.

[7] W. Jallet, N. Mansard, and J. Carpentier, "Implicit Differential Dynamic Programming," in *International Conference on Robotics and Automation (ICRA 2022)*. Philadelphia, United States: IEEE Robotics and Automation Society, May 2022.

[8] W. Jallet, A. Bambade, N. Mansard, and J. Carpentier, "Constrained Differential Dynamic Programming: A primal-dual augmented Lagrangian approach," Mar. 2022.

[9] A. Bambade, S. El-Kazdadi, A. Taylor, and J. Carpentier, "ProxQP: Yet another Quadratic Programming Solver for Robotics and beyond."
[10] I. Chatzinikolaidis and Z. Li, "Trajectory Optimization of Contact-Rich Motions Using Implicit Differential Dynamic Programming," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, Apr. 2021.
[11] P. E. Gill and D. P. Robinson, "A primal-dual augmented Lagrangian," *Computational Optimization and Applications*, vol. 51, no. 1, Jan. 2012.
[12] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocoddyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control," *arXiv:1909.04947 [cs, math]*, Mar. 2020.