Constrained trajectory optimization

Augmented Lagrangian trajectory optimization with ProxDDP

Note: to simplify presentation, all the h_t are now equality constraints.

The terminal stage value function looks like

$$\begin{aligned} V_N(x) &= \max_{\nu} \ell_N(x) + \nu^\top h_N(x) - \frac{\mu_k}{2}\|\nu - \nu^k\|^2 \\ &= \ell_N(x) + \frac{1}{2\mu_k}\|h_N(x) + \mu_k \nu^k\|^2 - \frac{\mu_k}{2}\|\nu^k\|^2. \end{aligned} \tag{27}$$

AGIMUS INNOVATIVE ROBOTICS FOR AGILE PRODUCTION

Note: to simplify presentation, all the h_t are now equality constraints.

The terminal stage value function looks like

$$\begin{aligned} V_N(x) &= \max_\nu \ell_N(x) + \nu^\top h_N(x) - \frac{\mu_k}{2}\|\nu - \nu^k\|^2 \\ &= \ell_N(x) + \frac{1}{2\mu_k}\|h_N(x) + \mu_k \nu^k\|^2 - \frac{\mu_k}{2}\|\nu^k\|^2. \end{aligned} \tag{27}$$

The proximal Bellman recursion is

$$V_t(x) = \min_{u,x'} \max_{\nu,\lambda} \left\{ \mathcal{Q}_t(x,u,\lambda,\nu,x') - \frac{\mu_k}{2}\|\lambda - \underbrace{\lambda^k}_{\text{prox. iteration}}\|^2 - \frac{\mu_k}{2}\|\nu - \nu^k\|^2 \right\} \tag{28}$$

where

$$\mathcal{Q}_t(x,u,\lambda,\nu,x') \overset{\text{def}}{=} \ell_t(x,u) + \nu^\top h_t(x,u) + \lambda^\top (f_t(x,u) - x') + V_{t+1}(x'). \tag{29}$$

General principle. Solve recursion DDP/iLQR-style with a quadratic model!

General principle. Solve recursion DDP/iLQR-style with a quadratic model!

Recursion hypothesis. We posit that the next-step value function variation is

$$\delta V_{t+1}(\delta x) \approx p_{t+1}^\top \delta x + \frac{1}{2}\delta x^\top P_{t+1}\delta x. \tag{30}$$

Goal. Close the recursion, by using Bellman's equation!

Then, solve for $(\delta u_t, \delta \nu_t, \delta \lambda_{t+1}, \delta x_{t+1})$ as functions of δx_t as follows:

$$\underbrace{\begin{bmatrix} R_t & D_t^\top & B_t^\top & \\ D_t & -\mu_k I & & \\ B_t & & -\mu_k I & -I \\ & & -I & P_{t+1} \end{bmatrix}}_{\stackrel{\text{def}}{=}\mathcal{K}_t} \begin{bmatrix} \delta u_t \\ \delta \nu_t \\ \delta \lambda_{t+1} \\ \delta x_{t+1} \end{bmatrix} = - \begin{bmatrix} r_t + S_t^\top \delta x_t \\ \bar{d}_t^k - \mu_k \nu_t + C_t \delta x_t \\ \bar{s}_t^k - \mu_k \lambda_{t+1} + A_t \delta x_t \\ p_{t+1} \end{bmatrix} \tag{31}$$

where, $\bar{d}_t^k = d_t + \mu_k \nu_t^k$, $\bar{s}_t^k = s_t + \mu_k \lambda_{t+1}^k$.

Then, solve for $(\delta u_t, \delta\nu_t, \delta\lambda_{t+1}, \delta x_{t+1})$ as functions of δx_t as follows:

$$\underbrace{\begin{bmatrix} R_t & D_t^\top & B_t^\top & \\ D_t & -\mu_k I & & \\ B_t & & -\mu_k I & -I \\ & & -I & P_{t+1} \end{bmatrix}}_{\stackrel{\text{def}}{=} \mathcal{K}_t} \begin{bmatrix} \delta u_t \\ \delta\nu_t \\ \delta\lambda_{t+1} \\ \delta x_{t+1} \end{bmatrix} = - \begin{bmatrix} r_t + S_t^\top \delta x_t \\ \bar{d}_t^k - \mu_k \nu_t + C_t \delta x_t \\ \bar{s}_t^k - \mu_k \lambda_{t+1} + A_t \delta x_t \\ p_{t+1} \end{bmatrix} \tag{31}$$

where, $\bar{d}_t^k = d_t + \mu_k \nu_t^k$, $\bar{s}_t^k = s_t + \mu_k \lambda_{t+1}^k$. As δx_t is unknown, we can (in DDP fashion) extract a *parametric* solution in feedforward/feedback form:

$$\begin{bmatrix} k_t & K_t \\ \zeta_t & Z_t \\ \xi_{t+1} & \Xi_{t+1} \\ m_t & M_t \end{bmatrix} = -\mathcal{K}_t^{-1} \begin{bmatrix} r_t & S_t^\top \\ \bar{d}_t^k - \mu_k \nu_t & C_t \\ \bar{s}_t^k - \mu_k \lambda_{t+1} & A_t \\ p_{t+1} & 0 \end{bmatrix} \tag{32}$$

The value function model update is given by

$$P_t = Q_t + S_t K_t + C_t^\top Z_t + B_t^\top \Xi_{t+1} \tag{33a}$$

$$p_t = q_t + S_t k_t + C_t^\top \zeta_t + B_t^\top \xi_{t+1} \tag{33b}$$

such that $\delta V_t(\delta x) \approx p_t^\top \delta x + \frac{1}{2}\delta x^\top P_t \delta x$.

Thereby closing the recursion.

Initial stage. The initial stage constraint is $\bar{x}_0 - x_0 = 0$.

The update $(\delta x_0, \delta\lambda_0)$ satisfies

$$\begin{bmatrix} P_0 & -I \\ -I & -\mu_k I \end{bmatrix} \begin{bmatrix} \delta x_0 \\ \delta\lambda_0 \end{bmatrix} = - \begin{bmatrix} p_0 \\ \bar{x}_0 \end{bmatrix} \tag{34}$$

This leaves the way open to some **extensions**, e.g. initial constraints $g_0(x_0) = 0$.

Linear vs. nonlinear rollouts

Once $(\delta x_0, \delta\lambda_0)$ is computed, we can reconstruct the update for the trajectory:

Linear rollout (a.k.a. SQP)

$$\delta u_t = k_t + K_t \delta x_t \tag{35a}$$

$$\delta \nu_t = \zeta_t + Z_t \delta x_t \tag{35b}$$

$$\delta \lambda_{t+1} = \xi_{t+1} + \Xi_{t+1} \delta x_t \tag{35c}$$

$$\delta x_{t+1} = m_t + M_t \delta x_t \tag{35d}$$

Nonlinear rollout (DDP-style)

$$u_t^+ = u_t + k_t + K_t \delta x_t \tag{36a}$$

$$\nu_t^+ = \nu_t + \zeta_t + Z_t \delta x_t \tag{36b}$$

$$\lambda_{t+1}^+ = \lambda_{t+1} + \xi_{t+1} + \Xi_{t+1} \delta x_t \tag{36c}$$

$$x_{t+1}^+ = f_t(x_t^+, u_t^+) - \mu_k \lambda_{t+1}^+ \tag{36d}$$

$$\delta x_{t+1} = x_{t+1}^+ - x_{t+1} \tag{36e}$$

Read the preprint!

ProxDDP: Proximal Constrained Trajectory Optimization

Wilson Jallet[*,1,2], Antoine Bambade[1], Etienne Arlaud[1], Sarah El-Kazdadi[1], Nicolas Mansard[2] and Justin Carpentier[1]

***Abstract*—Trajectory optimization (TO) has proven, over the last decade, to be a versatile and effective framework for robot control. Several numerical solvers have been demonstrated to be fast enough to allow recomputing full-dynamics trajectories for various systems at control time, enabling model predictive control (MPC) of complex robots. These first implementations of MPC in robotics predominantly utilize some differential dynamic programming (DDP) variant for its computational speed and ease of use in constraint-free settings. Nevertheless, many scenarios in robotics call for adding hard constraints in TO problems (e.g., torque limits, obstacle avoidance), which existing solvers, based on DDP, often struggle to handle. Effectively addressing path constraints still poses optimization challenges (e.g., numerical stability, efficiency, accuracy of constraint satisfaction) that we propose to solve by combining advances in numerical optimization with the foundational efficiency of DDP. In this article, we leverage proximal methods for constrained optimization and introduce a DDP-like method to achieve fast, constrained trajectory optimization with an efficient warm-starting strategy particularly suited for MPC applications. Compared to earlier solvers, our approach effectively manages hard constraints without warm-start limitations and exhibits commendable convergence accuracy. Additionally, we leverage the computational efficiency of DDP, enabling real-time resolution of complex problems such as whole-body quadruped locomotion. We provide a complete implementation as part of an open-source and flexible C++ trajectory optimization library called ALIGATOR. These algorithmic contributions are validated through several trajectory planning scenarios from the robotics literature and the real-time whole-body MPC of a quadruped robot.**

***Index Terms*—Optimization and Optimal Control, Legged Robots, Model-Predictive Control**

I. Introduction

TRAJECTORY OPTIMIZATION is an efficient and generic approach for controlling complex dynamical systems such as robots. It is a principled framework for describing desired behaviors and generating motion. A workhorse in modern robotics, it has become a crucial ingredient in both kinodynamic planning and model predictive control (MPC) over the past decade, enabled by the increasing performance of computer chips and algorithmic enhancements alleviating previous computational bottlenecks. Notably, recent progress in both software and hardware has enabled the real-time computation of numerical quantities commonly involved in

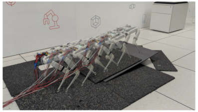

Fig. 1. **Solo-12 walking on an *unmodelled* slope** using the whole-body MPC framework based on primal-dual augmented Lagrangian techniques.

Optimal control problems (OCPs) are, by nature, infinite-dimensional optimization problems, which are largely not solvable in closed form. However, they can be solved numerically. On the one hand, there are *indirect* methods for OCPs, based on deriving their optimality conditions [4]. On the other hand, there are *direct* methods [5] which transcribe OCP problems into nonlinear programs (NLPs) of finite dimensions.

Direct methods, whichever the method of transcription, attempt resolution by utilizing a nonlinear programming approach, either leveraging general-purpose and off-the-shelf solvers such as IPOPT [6] or SNOPT [7], or a more tailored solution. Several approaches are considered in the literature to solve them in practice. We will argue why, in the robotics community, differential dynamic programming-based solvers are seen as a promising research direction, notably for deploying receding horizon control schemes for real-time robot control. One transcription method is *collocation*, which approximates the OC problem using a finite-dimensional basis functions such as polynomials. Another transcription method is *shooting methods*. They use a discretization of the system dynamics through numerical integration, which is generic, efficient, and easy to implement. For these shooting methods, there exist structure-exploiting solvers leveraging Riccati recursion [8]–[11], such as the differential dynamic programming (DDP) [12] algorithm. DDP is one of the earliest such methods and a reference in nonlinear trajectory optimization, known to have quadratic convergence [13], and has several variants such as the iterative

Library to be publicly released **soon** (multi-team effort, please contribute!).

Simple-Robotics / proxddp

Code · Issues 10 · Pull requests · Actions · Projects · Security · Insights · Settings

proxddp Private

Edit Pins · Unwatch 7 · Fork 7 · Starred 7

devel · 5 Branches · 5 Tags · Go to file · Code

ManifoldFR [core] alm-weights: move initMatrix out-... 470d893 · yesterday · 1,580 Commits

.github/workflows	[ci] update linux.yml	2 months ago
bench	[core|solvers] move proxddp files to solvers/...	yesterday
cmake @ 02719f3	cmake: sync submodule	2 weeks ago
doc	doc: add info on benchmarking	last year
examples	[core|solvers] move proxddp files to solvers/...	yesterday
include/proxddp	[core] alm-weights: move initMatrix out-of-line	yesterday
models	[models] add soccerball urdf and obj	6 months ago
python	[core] alm-weights: move initMatrix out-of-line	yesterday
scripts	scripts: add enable_perf.sh script	10 months ago
src	[core|solvers] move proxddp files to solvers/...	yesterday
tests	[core|solvers] move proxddp files to solvers/...	yesterday
.clang-format	pre-commit : run autoupdate	2 months ago
.cmake-format.yaml	config : add cmake formatting and linting	9 months ago
.gitignore	pre-commit : run autoupdate	2 months ago
.gitlab-ci.yml	gitlab-ci: update to build with croc support	last year
.gitmodules	add cmake submodule	last year
.pre-commit-config.yaml	Update pre-commit config (autoupdate)	2 weeks ago

About

A primal-dual augmented Lagrangian-type solver for trajectory optimization

Readme · Cite this repository · Activity · 7 stars · 7 watching · 7 forks

Releases

5 tags · Create a new release

Packages

No packages published · Publish your first package

Contributors 9

Languages

C++ 87.3% · Python 9.2% · CMake 3.5%

AGIMUS
INNOVATIVE ROBOTICS FOR AGILE PRODUCTION

[1] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, Jan. 12, 1997, 482 pp., ISBN: 978-0-691-01586-6. Google Books: 1Ti0ka9bx3sC.

[2] P. Gill, W. Murray, and M. Saunders, **“Snopt: An sqp algorithm for large-scale constrained optimization,”** *SIAM Journal on Optimization*, vol. 12, pp. 979–1006, Apr. 26, 2002. DOI: 10.2307/20453604.

[3] A. Wächter and L. T. Biegler, **“On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,”** *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar. 2006, ISSN: 0025-5610, 1436-4646. DOI: 10/c6j59p. [Online]. Available: http://link.springer.com/10.1007/s10107-004-0559-y (visited on 11/12/2021).

[4] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Lancelot: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, 1st ed. Springer Publishing Company, Incorporated, Nov. 2010, 330 pp., ISBN: 978-3-642-08139-2.

AGIMUS
INNOVATIVE ROBOTICS FOR AGILE PRODUCTION

[5] J. Nocedal and S. J. Wright, *Numerical Optimization* (Springer Series in Operations Research), 2nd ed. New York: Springer, 2006, 664 pp., ISBN: 978-0-387-30303-1.

[6] D. Goldfarb and A. Idnani, **"A numerically stable dual method for solving strictly convex quadratic programs,"** *Mathematical Programming*, vol. 27, no. 1, pp. 1–33, Sep. 1, 1983, ISSN: 1436-4646. DOI: 10.1007/BF02591962. [Online]. Available: https://doi.org/10.1007/BF02591962 (visited on 12/06/2023).

[7] A. Conn, N. Gould, and P. Toint, **"A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds,"** *SIAM Journal on Numerical Analysis*, vol. 28, Apr. 1, 1991. DOI: 10.1137/0728030.

AGIMUS
INNOVATIVE ROBOTICS FOR AGILE PRODUCTION

[8] J. Carpentier, R. Budhiraja, and N. Mansard, **"Proximal and sparse resolution of constrained dynamic equations,"** in *Robotics: Science and Systems XVII*, Robotics: Science and Systems Foundation, Jul. 12, 2021, ISBN: 978-0-9923747-7-8. DOI: 10.15607/RSS.2021.XVII.017. [Online]. Available: http://www.roboticsproceedings.org/rss17/p017.pdf (visited on 11/10/2022).

[9] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, **"Osqp: An operator splitting solver for quadratic programs,"** *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, Dec. 2020, ISSN: 1867-2949, 1867-2957. DOI: 10.1007/s12532-020-00179-2. [Online]. Available: http://link.springer.com/10.1007/s12532-020-00179-2 (visited on 01/23/2021).

[10] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, **"Qpoases: A parametric active-set algorithm for quadratic programming,"** *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, Dec. 1, 2014, ISSN: 1867-2957. DOI: 10.1007/s12532-014-0071-1. [Online]. Available: https://doi.org/10.1007/s12532-014-0071-1 (visited on 12/06/2023).

AGIMUS
INNOVATIVE ROBOTICS FOR AGILE PRODUCTION

[11] B. Hermans, A. Themelis, and P. Patrinos, **"Qpalm: A newton-type proximal augmented lagrangian method for quadratic programs,"** *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 4325–4330, Dec. 2019. DOI: 10.1109/CDC40024.2019.9030211. arXiv: 1911.02934. [Online]. Available: http://arxiv.org/abs/1911.02934 (visited on 04/01/2021).

[12] A. De Marchi, **"On a primal-dual newton proximal method for convex quadratic programs,"** *Computational Optimization and Applications*, vol. 81, no. 2, pp. 369–395, Mar. 2022, ISSN: 0926-6003, 1573-2894. DOI: 10.1007/s10589-021-00342-y. [Online]. Available: https://link.springer.com/10.1007/s10589-021-00342-y (visited on 02/18/2022).

[13] A. Bambade, F. Schramm, S. E. Kazdadi, S. Caron, A. Taylor, and J. Carpentier, **Proxqp: An efficient and versatile quadratic programming solver for real-time robotics applications and beyond,** Sep. 1, 2023. [Online]. Available: https://inria.hal.science/hal-04198663 (visited on 12/06/2023).

[14] R. T. Rockafellar, **“Augmented lagrangians and applications of the proximal point algorithm in convex programming,”** *Mathematics of Operations Research*, vol. 1, no. 2, pp. 97–116, 1976, ISSN: 0364-765X. JSTOR: 3689277. [Online]. Available: https://www.jstor.org/stable/3689277 (visited on 03/18/2021).

[15] W. Li and E. Todorov, **“Iterative linear quadratic regulator design for nonlinear biological movement systems,”** in *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*, Setúbal, Portugal: SciTePress - Science, 2004, pp. 222–229, ISBN: 978-972-8865-12-2. DOI: 10/fn7nnp. [Online]. Available: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0001143902220229 (visited on 01/07/2021).

[16] M. Giftthaler, M. Neunert, M. Stäuble, J. Buchli, and M. Diehl, **"A family of iterative gauss-newton shooting methods for nonlinear optimal control,"** *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018. DOI: 10.1109/IROS.2018.8593840.

[17] Y. Tassa, T. Erez, and E. Todorov, **"Synthesis and stabilization of complex behaviors through online trajectory optimization,"** in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems.*, Oct. 1, 2012, pp. 4906–4913, ISBN: 978-1-4673-1737-5. DOI: 10.1109/IROS.2012.6386025.

[18] R. Verschueren *et al.*, **"Acados—a modular open-source framework for fast embedded optimal control,"** *Mathematical Programming Computation*, vol. 14, no. 1, pp. 147–183, Mar. 1, 2022, ISSN: 1867-2957. DOI: 10.1007/s12532-021-00208-8. [Online]. Available: https://doi.org/10.1007/s12532-021-00208-8 (visited on 07/03/2023).

AGIMUS
INNOVATIVE ROBOTICS FOR AGILE PRODUCTION

[19] G. Frison and M. Diehl, **"Hpipm: A high-performance quadratic programming framework for model predictive control,"** *IFAC-PapersOnLine*, 21st IFAC World Congress, vol. 53, no. 2, pp. 6563–6569, Jan. 1, 2020, ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2020.12.073. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896320303293 (visited on 07/03/2023).

[20] C. Mastalli, R. Budhiraja, *et al.*, **"Crocoddyl: An efficient and versatile framework for multi-contact optimal control,"** *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2536–2542, May 2020. DOI: 10.1109/ICRA40945.2020.9196673. [Online]. Available: https://ieeexplore.ieee.org/document/9196673/ (visited on 07/02/2023).

[21] C. Mastalli, S. P. Chhatoi, T. Corbéres, S. Tonneau, and S. Vijayakumar, **"Inverse-dynamics mpc via nullspace resolution,"** *IEEE Transactions on Robotics*, pp. 1–20, 2023, ISSN: 1941-0468. DOI: 10.1109/TRO.2023.3262186.

[22] L. Vanroye, A. Sathya, J. De Schutter, and W. Decré, **"Fatrop : A fast constrained optimal control problem solver for robot trajectory optimization and control,"** arXiv: 2303.16746 [cs, math]. (Mar. 29, 2023), [Online]. Available: http://arxiv.org/abs/2303.16746 (visited on 07/02/2023), preprint.

[23] A. Jordana, S. Kleff, A. Meduri, J. Carpentier, N. Mansard, and L. Righetti, **"Stagewise implementations of sequential quadratic programming for model-predictive control,"** (Dec. 2023), preprint.