The Journal of Open Source Software

opty: Software for trajectory optimization and parameter identification using direct collocation

Jason K. Moore[1] **and Antonie van den Bogert**[2]

1 University of California, Davis **2** Cleveland State University

DOI: 10.21105/joss.00300

Software

- Review
- Repository
- Archive

Submitted: 04 June 2017
Published: 29 January 2018

Summary

opty is a tool for describing and solving trajectory optimization and parameter identification problems based on symbolic descriptions of ordinary differential equations and differential algebriac equations that describe a dynamical system. The motivation for its development resides in the need to solve optimal control problems of biomechanical systems. The target audience is engineers and scientists interested in solving nonlinear optimal control and parameter identification problems with minimal computational overhead.

A user of opty is responsible for specifying the system's dynamics (the ODE/DAEs), the cost or fitness function, bounds on the solution, and the initial guess for the solution. opty uses this problem specification to derive the constraints needed to solve the optimization problem using the direct collocation method (Betts 2010). This method maps the problem to a non-linear programming problem and the result is then solved numerically with an interior point optimizer IPOPT (Wächter and Biegler 2006) which is wrapped by cyipopt (cyipopt Developers 2017) for use in Python. The software allows the user to describe the dynamical system of interest at a high level in symbolic form without needing to concern themselves with the numerical computation details. This is made possible by utilizing SymPy (Meurer et al. 2017) and Cython (Behnel et al. 2011) for code generation and just-in-time compilation to obtain wrap optimized C functions that are accessible in Python.

Direct collocation methods have been especially successful in the field of human movement (Ackermann and Bogert 2010, A. J. van den Bogert et al. (2012)) because those systems are highly nonlinear, dynamically stiff, and unstable with open loop control. Typically, closed-source tools were used for multibody dynamics (e.g. SD/Fast, Autolev), for optimization (SNOPT), and for the programming environment (Matlab). Recently, promising work has been done with the Opensim/Simbody dynamics engine (Lee and Umberger 2016, Lin and Pandy (2017)), but this requires that Jacobian matrices are approximated by finite differences. In contrast, opty provides symbolic differentiation which makes the code faster and prevents poor convergence when the severe nonlinearity causes finite differences to be inaccurate. Furthermore, opty allows the system to be formulated using an implicit differential equation, which often results in far simpler equations and better numerical conditioning. The first application of opty was in the identification of feedback control parameters for human standing (Moore and Bogert 2015). It should be noted that opty can use any dynamic system model and is not limited to human movement.

Presently, opty only implements first order (backward Euler) and second order (midpoint Euler) approximations of the dynamics. Higher accuracy and/or larger time steps can be achieved with higher order polynomials (Patterson and Rao 2014), and opty could be

extended towards such capabilities. In our experience, however, the low order discretizations provide more robust convergence to the globally optimal trajectory in a nonlinear system when a good initial guess is not available (Zarei 2016).

There are existing software packages that can solve optimal control problems and have have similarities to opty. Below, is a feature comparison of those we are aware of:

Name	Citation	Language	License	Derivatives	Discretization	Implicit Dynamics	Solvers	Project Website
Casadi[1]	(Andersson 2013)	C++, Python, Octave,	LGPL	Automatic differentiation	None	Yes	IPOPT, WORHP, SNOPT, KNITRO	Casadi Website
DIDO	(Ross and Fahroo 2002)	Matlab	Commercial	Analytic	Pseudospectral	Yes	built-in	DIDO Website
DIRCOL	(Stryk and Bulirsch 1992)	Fortran	Non-commercial	Finite differences	Piecewise linear/cubic	Yes	NPSOL, SNOPT	DIRCOL Website
DYNOPT	(Cizniar et al. 2005)	Matlab	Custom Open Source, Non-commercial	Must be supplied by user	Pseudospectral	Mass matrix	fmincon	DYNOPT Code and Documentation
FROST	(Hereid and Ames 2017)	Matlab, Mathematica	BSD 3-Clause	Analytic	?	?	IPOPT, fmincon	FROST Documentation
GESOP	(Gath and Well 2001)	Matlab, C, Fortan, Ada	Commercial	?	Pseudospectral	No	SLLSQP, SNOPT, SOCS	Astos Solutions Gmbh
GPOPS	(Patterson and Rao 2014)	Matlab	Commercial	Automatic differentiation	Pseudospectral	No	SNOPT, IPOPT	GPOPS Website
opty	NA	Python	BSD 2-Clause	Analytic	Euler, Midpoint	Yes	IPOPT	opty Documentation
OTIS	(Hargraves and Paris 1987)	Fortran	US Export Controlled	?	Gauss-Labatto, Pseudospectral	Yes	SNOPT	OTIS Website
PROPT	(Rutquist and Edvall 2010)	Matlab	Commercial	Analytic	Pseudospectral	Yes	SNOPT, KNITRO	TOMDYN Website
PSOPT	(Becerra 2010)	C++	GPL	Automatic differentiation, Sparse finite differences	Pseudospectral, RK	Yes	IPOPT, SNOPT	PSOPT Website
SOCS	(Betts 2010)	Fortran	Commercial	Finite differences	Euler, RK, & others	Yes	built-in	SOCS Documentation

Each of these software packages offer a different combination of attributes and features

[1] Casadi does not have a built in direct collocation transcription but includes examples which show how to do so for specific problems.

that make it useful for different problems. opty is the only package that has a liberal open source license for itself and its dependencies, following precedent set by other core scientific Python packages. This allows anyone to use and modify the code without having to share the source of their application. opty also is the only package, open source or not, that allows (in fact forces) the user to describe their problem via a high level symbolic mathematical description using the API of a widely used computer algebra system instead of a domain specific language. This relieves the user from having to translate the much simpler continuous problem definition into a discretize NLP problem. opty leverages the popular Scientific Python core tools like NumPy, SymPy, Cython, and matplotlib allowing users to include opty code into Python programs. Lastly, the numeric code generated by opty to evaluate the NLP constraints is optimized providing extremely efficient parallel evaluation of the contraints. This becomes very valuable for high dimensional dynamics. opty currently does not offer a wide range of discretization methods nor support for solvers other than IPOPT, but those could relatively easily be added based on user need.

References

Ackermann, Marko, and Antonie J. van den Bogert. 2010. "Optimality Principles for Model-Based Prediction of Human Gait." *Journal of Biomechanics* 43 (6):1055–60. https://doi.org/https://doi.org/10.1016/j.jbiomech.2009.12.012.

Andersson, Joel. 2013. "A General-Purpose Software Framework for Dynamic Optimization." PhD thesis, Department of Electrical Engineering (ESAT/SCD); Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium: Arenberg Doctoral School, KU Leuven.

Becerra, V. M. 2010. "Solving Complex Optimal Control Problems at No Cost with PSOPT." In *2010 IEEE International Symposium on Computer-Aided Control System Design*, 1391–6. https://doi.org/10.1109/CACSD.2010.5612676.

Behnel, S., R. Bradshaw, C. Citro, L. Dalcin, D.S. Seljebotn, and K. Smith. 2011. "Cython: The Best of Both Worlds." *Computing in Science Engineering* 13 (2):31–39. https://doi.org/10.1109/MCSE.2010.118.

Betts, J. 2010. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Advances in Design and Control. Society for Industrial; Applied Mathematics. https://doi.org/10.1137/1.9780898718577.

Bogert, Antonie J van den, Maarten Hupperets, Heiko Schlarb, and Berthold Krabbe. 2012. "Predictive Musculoskeletal Simulation Using Optimal Control: Effects of Added Limb Mass on Energy Cost and Kinematics of Walking and Running." *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology* 226 (2):123–33. https://doi.org/10.1177/1754337112440644.

cyipopt Developers. 2017. "Cyipopt: Cython Interface for the Interior Point Optimzer IPOPT."

Čižniar, M., D. Salhi, M. Fikar, and M. A. Latifi. 2005. "DYNOPT - Dynamic Optimisation Code for Matlab." In *13th Annual Conference Proceedings of Technical Computing Prague 2005*. Praha: Humusoft. http://www.kirp.chtf.stuba.sk/publication_info.php?id_pub=255.

Gath, Peter, and Klaus Well. 2001. "Trajectory Optimization Using a Combination of Direct Multiple Shooting and Collocation." In *AIAA Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics; Astronautics. https://doi.org/10.2514/6.2001-4047.

Hargraves, C. R., and S. W. Paris. 1987. "Direct Trajectory Optimization Using Nonlinear Programming and Collocation." *AIAA Journal of Guidance, Control, and Dynamics* 10 (4):338–42.

Hereid, Ayonga, and Aaron D. Ames. 2017. "FROST: Fast Robot Optimization and Simulation Toolkit." In *IEEE/Rsj International Conference on Intelligent Robots and Systems (Iros)*. Vancouver, BC, Canada: IEEE/RSJ.

Lee, Leng-Feng, and Brian R. Umberger. 2016. "Generating Optimal Control Simulations of Musculoskeletal Movement Using OpenSim and MATLAB." *PeerJ* 4 (January):e1638. https://doi.org/10.7717/peerj.1638.

Lin, Yi-Chung, and Marcus G. Pandy. 2017. "Three-Dimensional Data-Tracking Dynamic Optimization Simulations of Human Locomotion Generated by Direct Collocation." *Journal of Biomechanics* 59 (Supplement C):1–8. https://doi.org/https://doi.org/10.1016/j.jbiomech.2017.04.038.

Meurer, Aaron, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, et al. 2017. "SymPy: symbolic Computing in Python." *PeerJ Computer Science* 3 (January):e103. https://doi.org/10.7717/peerj-cs.103.

Moore, Jason K., and Antonie J. van den Bogert. 2015. "Quiet Standing Control Parameter Identification with Direct Collocation." In *15th International Symposium on Computer Simulation in Biomechanics*, 21–22. Glasgow, U.K.

Patterson, Michael A., and Anil V. Rao. 2014. "GPOPS-Ii: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using Hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming." *ACM Trans. Math. Softw.* 41 (1). New York, NY, USA: ACM:1:1–1:37. https://doi.org/10.1145/2558904.

Ross, I. Michael, and Fariba Fahroo. 2002. "A Direct Method for Solving Nonsmooth Optimal Control Problems." *IFAC Proceedings Volumes* 35 (1):479–84. https://doi.org/https://doi.org/10.3182/20020721-6-ES-1901.00329.

Rutquist, Per E, and Marcus M Edvall. 2010. "Propt-Matlab Optimal Control Software." 1. Vol. 260. Tomlab Optimization Inc.

Stryk, O. von, and R. Bulirsch. 1992. "Direct and Indirect Methods for Trajectory Optimization." *Annals of Operations Research* 37 (1):357–73. https://doi.org/10.1007/BF02071065.

Wächter, Andreas, and Lorenz T. Biegler. 2006. "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming." *Mathematical Programming* 106 (1):25–57. https://doi.org/10.1007/s10107-004-0559-y.

Zarei, Milad. 2016. "Predictive Simulation of Rowing Exercise." Master's thesis, Cleveland State University; Cleveland State University. http://rave.ohiolink.edu/etdc/view?acc_num=csu1472557492.