CoreRobotics: An object-oriented C++ library with cross-language wrappers for cross-platform robot control

Parker Owan, Cameron Devine, and W. Tony Piaskowy
University of Washington, Department of Mechanical Engineering

November 21, 2017

Library Page: http://www.CoreRobotics.org
Software Repository: https://gitlab.com/powan/CoreRobotics

Summary

Real-time controllers for robot manipulators are typically developed and implemented on a custom basis due to 1) the complexity of control in real application, and 2) requirement of only incremental changes in hardware and software as the design matures. Modular approaches to actuators have made it easier to quickly assemble custom robot designs to address the increasing thrust for physical automation in society. However, development of controllers for such robot platforms is often performed in resource-intensive software suites such as Robot Operating System (ROS) [1]. This approach requires unnecessarily high processor performance when the controller does not fully utilize the suite.

The CoreRobotics libraries were developed in an effort to provide generalized implementations of algorithms facilitating rapid development of real-time robot control. CoreRobotics utilizes an object-oriented approach in C++ to implement fast cross-platform thread management and timing, core math solvers [2], manipulator control [3, 4, 5, 6, 7], and trajectory shaping [8], and modeling for state estimation [9, 10, 11].

CoreRobotics has been compiled on Windows 8.1, 10, Linux, and MacOS on various hardware architectures. Linear algebra is handled with Eigen [12]. CMake [13] is used to unify the compile process across multi-platform developer environments, and an option is provided to compile Python and MATLAB wrappers using SWIG [14]. The CoreRobotics library is used in several research projects at the University of Washington, Seattle.

Acknowledgments

The authors would like to thank Professor Santosh Devasia and Professor Joseph Garbini for their support and direction during development of the software library.

References

[1] The Open Source Robotics Foundation, "Robot Operating System (ROS)," 2017. http://www.ros.org/

[2] E. Kreyszig, *Advanced Engineering Mathematics*, ed.9, John Wiley & Sons, 2011.

[3] J. Craig, *Introduction to Robotics: Mechanics and Control*, ed. 3, Pearson, 2004.

[4] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robot Manipulation*, ed. 1, CRC Press, 1993. http://www.cds.caltech.edu/~murray/mlswiki/index.php/Main_Page

[5] A. Aristidou and J. Lasenby, "Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver," University of Cambridge, Technical Report, 2009. http://publications.eng.cam.ac.uk/324095/

[6] O. Khatib, "Advanced Robotic Manipulation," Lecture Notes (CS327A) 2005. http://www.in.tum.de/fileadmin/user_upload/Lehrstuehle/Lehrstuhl_XXIII/AdvancedRoboticManipulation.pdf

[7] A. Hourtash, "The Kinematic Hessian and Higher Derivatives," IEEE International Symposium on Computational Intelligence in Robotics and Automation, 169-174, 2005. http://doi.org/10.1109/CIRA.2005.1554272

[8] N. Hogan, "Adaptive control of mechanical impedance by coactivation of antagonist muscles," IEEE Trans. on Automatic Control AC-29: 681-690, 1984.

[9] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press, 2006.

[10] J. Crassidis and J. Junkins, *Optimal Estimation of Dynamic Systems*, ed. 2, CRC Press, 2012.

[11] M.S. Arulampalam et al., "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," IEEE Transactions on Signal Processing, vol. 50, no. 2, pp. 174-188, 2002.

[12] Eigen (C++ Library), 2017. http://eigen.tuxfamily.org/

[13] T. Lovelace, "CMake: The Cross Platform Build System," Linux Magazine, July 2006.

[14] Simplified Wrapper and Interface Generator (SWIG), C/C++ Library, http://www.swig.org/