Realtime Model Predictive Control using Parallel DDP on a GPU

Brian Plancher and Scott Kuindersma

I. Introduction

Trajectory optimization algorithms are a powerful set of tools for synthesizing dynamic motions for complex robots. Recent work using variants of Differential Dynamic Programming (DDP) [1] has shown that online planning for model predictive control (MPC) is possible for a variety of practical (and sometimes high-dimensional) robots [2], [3], [4], [5], [6], [7]. In our own recent work [8], we conducted a variety of simulation experiments with the goal of understanding whether computational benefits could be gained by implementing a parallelized variant of the iterative Linear Quadratic Regulator (iLQR) [9] on a GPU. We found that GPU-based solvers can offer increased per-iteration computation time and faster convergence in some cases, but in general tradeoffs exist between convergence behavior and degree of algorithm-level parallelism. Given these promising observations, we continued this line of work using our implementation for MPC on a physical Kuka arm to demonstrate the feasibility of this approach in the presence of model discrepancies and communication delays between the robot and GPU. We found that higher control rates generally lead to better tracking performance across a range of parallelization options.

II. Background

Implementing a parallelized variant of iLQR requires the exploitation of *instruction-level parallelism*, (e.g., running all possible line search iterations in parallel) and *algorithm-level parallelism* (modifying iLQR for multiple shooting and using a block based backward pass). Care must be taken to minimize memory latency and kernel/process launch overhead. Our solver implementations and examples can be found at `http://bit.ly/ParallelDDP`.

III. Simulation Experiments

Our initial simulation experiments with the Kuka arm showed that parallelism greatly increased the speed of the fully parallelizable Taylor approximations of the dynamics and cost functions, and increased the speed, with diminishing returns, of the forward simulation and backward pass. We found that on the CPU, the backward pass does not improve beyond where M, the amount of algorithm level parallelism, is equal to the number of CPU cores. We also found that increased algorithm level parallelism led to a decreased rate of convergence caused by increased average line search depth which led to non-monotonic forward simulation times on the CPU. Overall, this tradeoff allowed the GPU to outperform the CPU as shown in Figure 1.

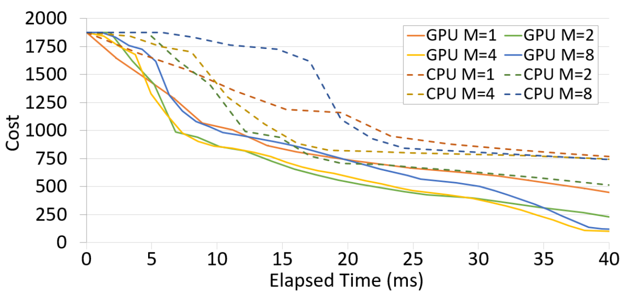

Fig. 1. Median cost for the first 40 milliseconds of each solve. M indicates the number of parallel backward pass blocks / multiple shooting intervals.

IV. Hardware Experiments

We ran a figure eight goal tracking experiment with the physical Kuka arm sweeping control step duration and amount of algorithm level parallelism. We warm started the iLQR algorithm by shifting all variables from the previous solve by the control step duration and then rolling out a new initial state trajectory starting from the current measured state (with a gravity compensating input in the trailing time steps). Simultaneously, we had another thread executing the previously computed feedback controller. To initialize the experiment, we held the first goal pose constant until the 2-norm of the end effector pose error and joint velocity were both less than 0.05 at which point the goal began moving along the figure eight path. Figure 2 shows the Kuka arm during one of these experiments.

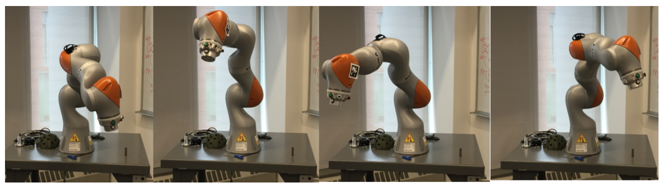

Fig. 2. The Kuka arm during a figure eight goal tracking experiment.

Figure 3 shows the average tracking error plotted against control step duration. We found that good tracking performance is possible for a wide range of solvers, and a faster control step duration generally had better tracking performance. We also found that solvers start to fail when they had about as many (or less) iterations as the amount of algorithm level parallelism (e.g., $M = 4$ with 3 or less iterations).

School of Engineering and Applied Sciences, Harvard University, Cambridge, MA. `brian_plancher@g.harvard.edu`, `scottk@seas.harvard.edu`

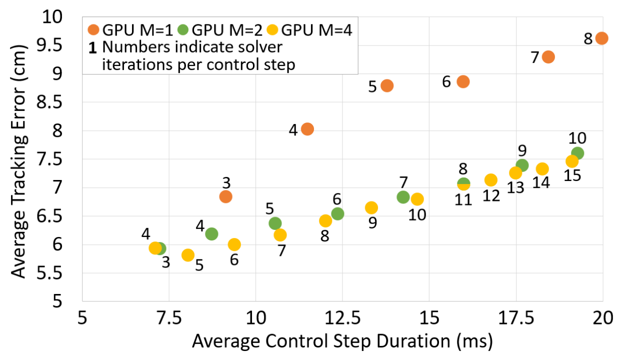

Fig. 3. Tracking error for a range of solvers vs. control step duration.

Acknowledgments

This work was supported by a Draper Internal Research and Development grant and by the National Science Foundation Graduate Research Fellowship (under grant DGE1745303). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the funding organizations.

References

[1] D. H. Jacobson and D. Q. Mayne. *Differential Dynamic Programming*. Elsevier.

[2] F. Farshidian, E. Jelavic, A. Satapathy, M. Giftthaler, and J. Buchli. Real-time motion planning of legged robots: A model predictive control approach. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics*.

[3] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and Stabilization of Complex Behaviors through Online Trajectory Optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[4] T. Erez, K. Lowrey, Y. Tassa, V. Kumar, S. Kolev, and E. Todorov. An integrated system for real-time model predictive control of humanoid robots. In *2013 13th IEEE-RAS International Conference on Humanoid Robots*.

[5] Jonas Koenemann, Andrea Del Prete, Yuval Tassa, Emanuel Todorov, Olivier Stasse, Maren Bennewitz, and Nicolas Mansard. Whole-body Model-Predictive Control applied to the HRP-2 Humanoid. In *Proceedings of the IEEERAS Conference on Intelligent Robots*.

[6] M. Neunert, C. de Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli. Fast nonlinear Model Predictive Control for unified trajectory optimization and tracking. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1398–1404.

[7] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli. Trajectory Optimization Through Contacts and Automatic Gait Discovery for Quadrupeds. 2(3):1502–1509.

[8] Brian Plancher and Scott Kuindersma. A Performance Analysis of Parallel Differential Dynamic Programming on a GPU. In *International Workshop on the Algorithmic Foundations of Robotics (WAFR)*.

[9] Weiwie Li and Emanuel Todorov. Iterative Linear Quadratic Regulator Design for Nonlinear Biological Movement Systems. In *Proceedings of the 1st International Conference on Informatics in Control, Automation and Robotics*.