Research Statement

Brian Plancher
brian.plancher@gmail.com

Intelligent field robots are a promising solution to many societal challenges from combating epidemics, to scaling global supply chains, to providing home health care to the elderly [1, 2, 3]. Realizing this vision requires the development of planning and control systems that can respond to real-world dynamic constraints with high speed and reliability.

I work in robotics, at the intersection of computer architecture and numerical optimization, to design state-of-the-art robot planning and control algorithms in the pursuit of this vision. My research focuses on using principles of hardware-software co-design to design, implement, and evaluate numerical optimal control algorithms on modern GPUs, FPGAs, and custom ASICs. This interdisciplinary approach leads to algorithms that robustly solve real-world planning and control problems at orders-of-magnitude faster rates, which will enable safer and more effective human-robot collaboration and interaction.

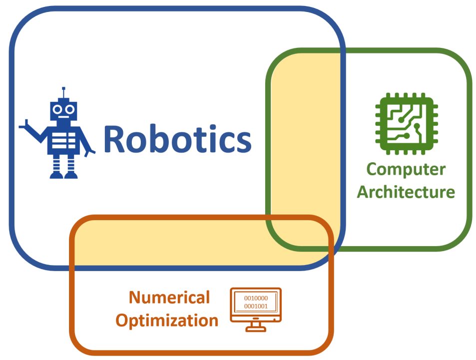

Figure 1: My research is at the intersection of robotics and adjacent fields

I am excited to develop motion planning and control techniques to power the next generation of dynamic and useful robots at your college or university, and look forward to grounding my research in collaboration and innovation at the intersection of robotics and adjacent fields.

Prior Work: Hardware Acceleration for Reliable Realtime Optimal Control

Model Predictive Control (MPC) transforms robot motion planning and control problems into (often nonlinear) optimization problems that are solved repeatedly online at high rates. This approach has been shown to generate highly dynamic motions for complex robots [4, 5, 6], but suffers from two fundamental problems: high computational complexity, and a lack of robust convergence mainly driven by non-convexity in the underlying optimization problem. Much of my research has focused on developing algorithms and implementations that address these issues through improved numerical methods and through hardware acceleration on modern GPUs, FPGAs, and custom ASICs.

GPU Accelerated Trajectory Optimization

My research has shown that the computational performance of MPC can be improved by leveraging the high throughput of GPUs. I designed new variants of general-purpose trajectory optimization solvers (e.g., Differential Dynamic Programming, Conjugate Gradient Direct Methods) that expose and exploit GPU-friendly sparsity patterns as well as parallelism at both the instruction-level and algorithm-level. Implementations of these co-designed algorithms outperformed state-of-the-art multi-threaded CPU based solvers both in the quality (cost) of the resulting trajectory and in the time required to compute that trajectory.

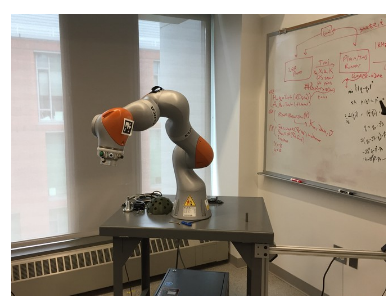

Figure 2: The Kuka manipulator using GPU-accelerated DDP [7].

To validate how well these approaches translate to the real-world, particularly in the presence of model discrepancies and communication delays between the robot and GPU, I deployed these implementations onto a 7-dof Kuka manipulator for nonlinear MPC (see Figure 2). These hardware experiments showed that GPU acceleration enables the MPC control loop to run at over 100Hz, improving overall system performance [8, 7].

Custom Hardware Accelerators for Rigid Body Dynamics

In order for algorithms to run efficiently on computer hardware all of their core computational kernels must also be accelerated. Rigid body dynamics kernels are key bottlenecks in many robotics algorithms, including 30% to 90% of the total computational time of MPC [5, 9, 10, 8]. To alleviate this bottleneck, I designed implementations of these algorithms for GPUs, FPGAs, and custom ASICs.

To enable the wider robotics community to leverage GPUs, I created GRiD, an open-source, GPU-accelerated library for rigid body dynamics and their gradients. By carefully taking advantage of sparsity and parallelism through code-generation, GRiD's throughput outperforms a state-of-the-art, code-generated, and multi-threaded CPU library [11] by as much as 7x [12, 13], translating to as much as a 25% to 77% reduction in the total computation time of nonlinear MPC.

For further acceleration, I also collaboratively developed hardware design methodologies to automate the design of robotics hardware accelerator chips. These design flows leverage robot morphology to generate sparsity and parallelism exploiting accelerators. We synthesized a proof-of-concept accelerator for the gradient of forward dynamics, which provided as much as a 58x and 519x latency speedup over the CPU and GPU respectively [12, 14]. A full open-source library leveraging these approaches is currently under development.

Improving the Numerical Performance of Optimization

Many existing state-of-the-art optimization algorithms do not support realistic real-world applications in challenging, dynamic environments. I develop variants of these algorithms that overcome these issues.

Differential Dynamic Programming (DDP) does not natively handle state constraints that commonly appear in the real world (e.g., avoiding obstacles) as it implicitly defines the states from the inputs and dynamics. My collaborators and I designed a variant of DDP that was both capable of satisfying state (and input) constraints through the use of an augmented Lagrangian, and had improved convergence properties by leveraging an unscented transform instead of traditional Taylor expansions. This enabled our algorithm to successfully find valid solution paths for a torque-limited Kuka manipulator in a simulated cluttered environment while satisfying constraints to high precision ($5e^{-5}$) from a range of problem initializations [15].

Future Work

At your college or university, I aim to pursue interdisciplinary and collaborative robotics research at the intersection of computer architecture, numerical optimization, embedded systems, and machine learning, to develop open-source algorithms and implementations that the larger robotics community can leverage to power the next generation of dynamic and useful robots.

Enabling robotics research on high-performance parallel architectures

Many robotics researchers develop their algorithms and implementations on top of core robotics toolboxes (e.g., Drake [16]) which only target CPUs for most operations. I am developing an open-source robotics toolbox to enable other researchers to accelerate their algorithms by leveraging high-performance parallel architectures (e.g., GPUs, FPGAs).

While my current work, GRiD, provides a starting point, it only represents a small fraction of the need that could be filled by this type of toolbox. For example, I would like support other parallel architectures (e.g., FPGAs), implement additional core kernels such as collision detection and the computation of contact forces and normals, as well as implement alternate formulations of dynamics that may enable higher performance through co-design [17, 18]. I also want to make the toolbox user-friendly and accessible to the broader robotics community through front-end APIs in high-level languages (e.g., Python, Julia).

Transforming offline solvers into realtime solutions

As previously mentioned, state-of-the-art solutions for whole-body nonlinear trajectory optimization are, for the most part, not fast enough to run online. This problem is only compounded when these solvers are also asked to reason about contact or to be robust to potential disturbances, limiting our ability to deploy robots into many real world applications. I aim to develop planning and control algorithms that are co-designed to expose hardware-friendly computational patterns to enable their acceleration to real-time rates.

I will not only explore standard trajectory optimization approaches but also robust control techniques such as belief space planning [19], techniques that reason about contact online such as contact-implicit trajectory optimization [20], and globalization techniques to enable convergence to “better” local minima [21]. These implementations would also be able to take advantage of the aforementioned toolbox for further optimization.

Developing deployable and data efficient learning and generalizable optimal control

Machine learning systems suffer from long training times and huge data costs, and often the results cannot be transferred to hardware [22]. At the same time, optimization-based approaches often produce brittle solutions and require hours of careful tuning to ensure convergence. By combining optimization-based approaches with machine learning, I will develop new algorithms with low sample complexity and high generalizability.

I recently supervised an undergraduate thesis that developed generative and flexible optimal controllers for highly sparse environments through the use of sample efficient learning [23]. Building on this work, I will develop actor-critic methods that learn “good” trajectory initializations and objective function regularizers to improve the convergence of optimization-based techniques. I will also develop novel MPC based differentiable model layers to reduce the sample complexity of machine learning. I will deploy and test these approaches on a low cost quadruped (e.g., the Unitree A1) to validate their real-world applicability in dynamic environments.

Tiny Robotics: lowering the cost of robotics for widespread deployment

I believe that much of the future of robotics is tiny. Small, low-cost, yet sophisticated systems can not only be deployed into more environments (e.g., crawling through small spaces for search and rescue), but can also increase global access to cutting-edge robotics research. With the growth of smart, ML-powered IoT devices, and the development of low cost, small robots like the Bittle, the time to explore these tiny solutions is now.

I will explore the possibility of reducing the computational requirements and improving the numerical conditioning of sophisticated planning and control techniques to enable their deployment onto microcontrollers. For example, I believe that well-designed autoencoders can provide a compressed latent representation that captures the full nonlinear dynamics better than template based models [24].

Widening Access in STEM Education

I will also contribute to the field of STEM Education research. As mentioned in my other statements, I not only open-source my courses, but also publish papers describing their pedagogical approach and lessons learned [25, 26, 27]. For example, I have a paper in preparation detailing the learnings from scaling the project-based robotics summer program with a focus on diversity, inclusion, and belonging. I aim to continue to research and develop new STEM learning models and programs to improve student access and outcomes.

Anticipated Funding Sources

Most of my research has been supported by the NSF GRFP fellowship and I have assisted multiple other students in their fellowship applications including the NSF GRFP, NDSEG, NVIDIA Research, and Microsoft Research. I have also assisted in building collaborations with industrial partners such as NVIDIA, Xillinx (now AMD), Google, Facebook, and Boston Dynamics. I anticipate my funding will come from a combination of government grants including NSF Robust Intelligence, Cyber-Physical Systems, and Improving Undergraduate STEM Education, as well as from industrial partners. I am also planning to pursue awards for early career research (e.g., NSF CAREER) and through agencies such as DARPA and NASA.

References

[1] G.-Z. Yang, B. J. Nelson, *et al.*, "Combating covid-19," *Science Robotics*, 2020.

[2] P. Palandrani, "Global supply chain disruptions driving robotics adoption," *Robotics Business Review*, 2020.

[3] J. Shaw, "The coming eldercare tsunami," *Harvard Magazine*, 2020.

[4] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and Stabilization of Complex Behaviors through Online Trajectory Optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[5] J. Koenemann, A. Del Prete, *et al.*, "Whole-body Model-Predictive Control applied to the HRP-2 Humanoid," in *Proceedings of the IEEERAS Conference on Intelligent Robots*.

[6] S. Kuindersma, R. Deits, *et al.*, "Optimization-based locomotion planning, estimation, and control design for Atlas."

[7] **B. Plancher** and S. Kuindersma, "Realtime model predictive control using parallel ddp on a gpu," in *Toward Online Optimal Control of Dynamic Robots Workshop at ICRA 2019*, Montreal, Canada, May. 2019.

[8] **B. Plancher** and S. Kuindersma, "A Performance Analysis of Parallel Differential Dynamic Programming on a GPU," in *International Workshop on the Algorithmic Foundations of Robotics (WAFR)*.

[9] M. Neunert, C. de Crousaz, *et al.*, "Fast nonlinear Model Predictive Control for unified trajectory optimization and tracking," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*.

[10] J. Carpentier and N. Mansard, "Analytical derivatives of rigid body dynamics algorithms," in *Robotics: Science and Systems*, 2018.

[11] J. Carpentier, G. Saurel, *et al.*, "The pinocchio c++ library : A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *2019 International Symposium on System Integration (SII)*, 2019.

[12] **B. Plancher**, S. M. Neuman, *et al.*, "Accelerating robot dynamics gradients on a cpu, gpu, and fpga," *IEEE Robotics and Automation Letters*, 2021.

[13] **B. Plancher**, S. M. Neuman, *et al.*, "Grid: Gpu-accelerated rigid body dynamics with analytical gradients," 2021.

[14] S. M. Neuman, **B. Plancher**, *et al.*, "Robomorphic computing: A design methodology for domain-specific accelerators parameterized by robot morphology," in *Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2021.

[15] **B. Plancher**, Z. Manchester, and S. Kuindersma, "Constrained Unscented Dynamic Programming," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[16] R. Tedrake, "Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems."

[17] R. Featherstone, "A divide-and-conquer articulated-body algorithm for parallel o (log (n)) calculation of rigid-body dynamics. part 1: Basic algorithm," *The International Journal of Robotics Research*, 1999.

[18] J. Brüdigam and Z. Manchester, "Linear-time variational integrators in maximal coordinates," 2020.

[19] R. Platt, L. Kaelbling, *et al.*, "Non-gaussian belief space planning: Correctness and complexity," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

[20] Z. Manchester and S. Kuindersma, "Variational contact-implicit trajectory optimization," in *Robotics Research*, 2020.

[21] E. Jelavic, F. Farshidian, and M. Hutter, "Combined sampling and optimization based planning for legged-wheeled robots," *arXiv preprint arXiv:2104.04247*, 2021.

[22] P. Varin, L. Grossman, and S. Kuindersma, "A comparison of action spaces for learning manipulation tasks," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.

[23] L. Grossman, "Reinforcement learning to enable robust robotic model predictive control," Cambridge, MA, May. 2020.

[24] R. J. Full and D. E. Koditschek, "Templates and anchors: neuromechanical hypotheses of legged locomotion on land," *Journal of experimental biology*, 1999.

[25] S. Karaman, A. Anders, *et al.*, "Project-based, collaborative, algorithmic robotics for high school students: Programming self-driving race cars at mit," in *IEEE integrated STEM education conference (ISEC)*, 2017.

[26] V. J. Reddi, **B. Plancher**, *et al.*, "Widening access to applied machine learning with tinyml," 2021.

[27] **B. Plancher** and V. Janapa Reddi, "Tinymledu: the tiny machine learning open education initiative," in *ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '21, 2022.