Research Statement

Brian Plancher
brian.plancher@gmail.com

Intelligent, robust, and dynamic field robots are a promising solution to many societal challenges from combating epidemics to providing home health care to the elderly [1, 2]. A major obstacle toward realizing their potential is the need for robust, dynamic motion planning and control algorithms that can be run online at real-time rates.

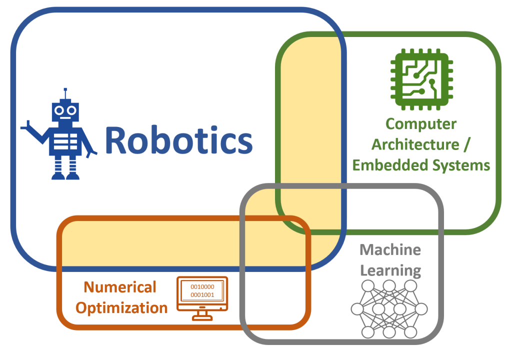

Figure 1: My research is at the intersection of robotics and adjacent fields

My research is focused on enabling these field robots by working at the intersection of robotics and computer architecture / embedded systems, numerical optimization, and machine learning. Inspired by the use of domain specific architecture in the field of machine learning, I approach robotics algorithm development through the lens of hardware-software co-design [3]—a foundational concept from the field of computer architecture / embedded systems—to target alternate computing platforms such as GPUs and FPGAs. I design planning and control algorithms with advantageous theoretical properties from the field of numerical optimization and use machine learning techniques to improve the overall robustness and efficacy of these algorithms. My hope is that through this intersectional approach, I can collaboratively develop robust algorithms that can be run at real-time rates, enabling safe human-robot collaboration and interaction in the real world.

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

Prior Work: Hardware-Software Co-Design for Robust Realtime Model Predictive Control

Model Predictive Control (MPC) transforms robot motion planning and control problems into (often nonlinear) optimization problems that are solved repeatedly online at very 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. Previous work has often relied on hierarchical controllers that leverage simplified robot dynamics models to both reduce the computational burden and improve the convergence properties of the solver [7]. The conservative assumptions used to derive these simplified models fundamentally limit the agility of the robots' behaviors. Much of my research has been focused on developing algorithms and implementations that address these issues.

Accelerating Computations through Co-Design

To overcome computational bottlenecks, I worked at the intersection of robotics and computer architecture, leveraging hardware-software co-design to target parallel computing platforms such as GPUs and FPGAs. MPC algorithms are composed of computations that have known sparsity patterns and exhibit both instruction-level and algorithm-level parallelism. GPUs, FPGAs, and other parallel architectures offer different computational models that can accelerate these computational patterns. However, careful co-design approaches are required to unlock this performance. As such, to enable their use for the broader robotics research community, there is a need for open-source libraries of both common and cutting-edge robotics kernels and algorithms that target these platforms. The development of these open-source libraries has driven much of my research progress to date.

For example, I have a published paper (and paper in preparation) developing co-design libraries and implementations of general-purpose trajectory optimization solvers (e.g., Differential Dynamic Programming, Conjugate Gradient Direct Methods) targeting GPUs. I have not only designed these algorithms but also deployed them for nonlinear MPC onto a 7-dof Kuka manipulator. Through these hardware experiments I have shown that GPU acceleration enables improved real-world tracking performance despite the presence of model discrepancies and communication delays between the robot and GPU [8, 9].

As another example, I have recently published (and have under review) at both robotics venues [10, 11] and computer architecture venues [12] GPU and FPGA libraries and proof-of-concept implementations of the spatial-algebra based formulations of rigid body dynamics. We targeted these kernels as they account for 30% to 90% of the total computational time of nonlinear MPC [5, 13, 14, 8]. We found that these parallel computing platforms could provide speedups of as much as an order of magnitude over a state-of-the-art multi-threaded CPU implementation.

Improving the Numerical Performance of Optimization (through Learning)

I also worked at the intersection of robotics and both numerical optimization and machine learning to develop algorithms with more robust convergence properties to enable more reliable real-world deployments. To remain robust to challenging, dynamic environments, my collaborators and I leveraged mathematical frameworks with good (global) convergence properties as well as sample efficient machine learning techniques to adapt and generalize solutions.

For example, my collaborators and I designed a variant of DDP that was both capable of satisfying constraints to higher degrees of accuracy through the use of an augmented Lagrangian, and had improved convergence properties by leveraging an unscented transform instead of traditional Taylor expansions [15]. This enabled our algorithm to successfully find valid solution paths despite bad initializations.

I also supervised an undergraduate thesis [16] that built on recent work leveraging offline reinforcement learning to add robustness to online MPC algorithms [17, 18]. We extended these approaches to enable complex robots to compute generative and flexible controllers for highly sparse environments in a sample efficient manner. This work has continued since the student graduated and we now have a paper in preparation.

Future Work

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

Enabling robotics research on high-performance parallel architectures

Many robotics researchers build their algorithms and implementations on top of core robotics kernel toolboxes such as Drake [19]. Unfortunately, these toolboxes only target CPUs for most operations. This fundamentally limits the larger robotics community's ability to target GPUs, FPGAs, and other higher performance parallel architectures. Researching at the intersection of robotics and computer architecture, I am developing a complete open-source robotics toolbox to enable robotics researchers to use these currently inaccessible high-performance architectures to address computational bottlenecks and unlock online use of powerful algorithms and approaches once deemed too computationally expensive.

While my current work, developing a GPU dynamics library, 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 to both implement additional core kernels such as collision detection and the computation of contact forces and normals, as well as explore alternate formulations of dynamics that may expose additional parallelism and performance through co-design [20, 21, 22, 23]. Finally, I want to make the toolbox user-friendly through front-end APIs in a high-level languages (e.g., Python, Julia). With these additional features, this toolbox should enable robotics researchers to easily access and build upon the performance improvements enabled by high-performance parallel architectures, enabling new creative solutions to dynamic robotics problems

Transforming offline solvers into realtime solutions

As previously mentioned, state-of-the-art CPU solutions for whole-body nonlinear trajectory optimization are, for the most part, not fast enough to run online today. This problem is only compounded when these solvers are also asked to reason about contact or to be robust to potential disturbances. This fundamentally limits our ability to deploy robots into many real world applications which require safe, dynamic interactions with humans. At the intersection of robotics and numerical optimization, I will research and develop a series of planning and control algorithms and implementations that are co-designed to expose additional hardware-friendly computational patterns to enable the acceleration needed to run these algorithms at real-time rates.

I will not only explore standard trajectory optimization approaches but also robust control techniques such as belief space planning [24], techniques that reason about contact online such as contact-implicit trajectory optimization [25], and large-scale globalization techniques to ensure better convergence to "good" local minima (if not to the global optima). These implementations would also be able to take advantage of the aforementioned toolbox to ensure that they were optimized down to the lowest level kernels.

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 [26]. At the same time, optimization-based robotics approaches often produce brittle solutions and require hours of careful tuning to ensure convergence. At the intersection of robotics and both machine learning and numerical optimization, I aim to develop methods that integrate structure from optimization-based techniques into machine learning algorithms to reduce their sample complexity and integrate the generalizability of machine learning approaches into optimization-based techniques to improve their generalizability and globalization properties.

Potential projects leveraging learning to improve optimization include using actor-critic methods to learn "good" trajectory initializations for MPC in dynamic environments and to learn objective function regularizers to improve convergence. Potential projects leveraging optimization to improve learning include developing novel MPC based differentiable model layers to reduce sample complexity.

To validate the real-world applicability of these approaches through contact, I aim to deploy them onto a low cost quadruped, e.g., the Unitree A1, and test them both in the lab and in the real world. Only once these approaches are stable in real, dynamic environments can they be deployed to help people globally.

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 environments where robots previously could not venture (e.g., crawling through small spaces during a search and rescue mission), but can also enable a larger percentage of the world to access cutting-edge robotics research. With the rapid growth of smart IoT devices leveraging TinyML [27], and the development of low cost, palm size robots like the Bittle, the time to explore new algorithmic and computational approaches to enable these tiny solutions is now.

At the intersection of robotics and embedded systems, I will explore the possibility of deploying sophisticated planning and control techniques onto microcontrollers. This will require the development of algorithms that consume far fewer computational resources and provide better numerical conditioning.

At the intersection of robotics and machine learning, I will also leverage autoencoders to reduce the dimensionality of these optimization problems to help them fit on tiny devices. 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 (e.g., Hybrid Zero Dynamics) [28, 29], improving their end-to-end performance.

Widening Access in STEM Education

I hope to continue to develop new and innovative courses and contribute to the field of STEM Education research. As mentioned in my other statements, I not only open-source my courses, but also publish academic papers describing the pedagogical approach and learnings from my courses [30, 31]. For example, I now have a paper in preparation detailing the learnings from scaling the project-based robotics summer program and working to improve diversity and inclusion in the student body. At your college or university, I hope to continue to explore, research, and develop new and innovative STEM learning models and outreach 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 graduate and undergraduate students in their applications for fellowships 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 Information Integration and Informatics, Robust Intelligence, Cyber-Physical Systems, and Improving Undergraduate STEM Education, as well as from industrial partners. I am also planning to actively pursue awards for early career research through programs such as the NSF CAREER program and through agencies such as DARPA and NASA.

References

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

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

[3] G. De Michell and R. K. Gupta, "Hardware/software co-design," *Proceedings of the IEEE*, 1997.

[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, *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, *et al.*, "Optimization-based locomotion planning, estimation, and control design for Atlas."

[7] J. Di Carlo, *et al.*, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.

[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] ——, "Realtime model predictive control using parallel ddp on a gpu," in *Toward Online Optimal Control of Dynamic Robots Workshop at the 2019 International Conference on Robotics and Automation (ICRA)*, Montreal, Canada, May. 2019.

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

[11] ——, "Grid: Gpu-accelerated rigid body dynamics with analytical gradients," 2021.

[12] S. M. Neuman, *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*, 2021.

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

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

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

[17] K. Lowrey, *et al.*, "Plan online, learn offline: Efficient learning and exploration via model-based control," in *International Conference on Learning Representations*, 2019.

[18] F. Farshidian, D. Hoeller, and M. Hutter, "Deep value model predictive control," 2019.

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

[20] 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.

[21] Y. Yang, Y. Wu, and J. Pan, "Parallel Dynamics Computation Using Prefix Sum Operations."

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

[23] J. Nganga and P. M. Wensing, "Accelerating second-order differential dynamic programming for rigid-body systems," *IEEE Robotics and Automation Letters*, 2021.

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

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

[26] 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.

[27] P. Warden and D. Situnayake, *Tinyml: Machine learning with tensorflow lite on arduino and ultra-low-power microcontrollers.* O'Reilly Media, 2019.

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

[29] K. Sreenath, *et al.*, "A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel," *The International Journal of Robotics Research*, 2011.

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

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