Research Statement

Brian Plancher
brian.plancher@gmail.com

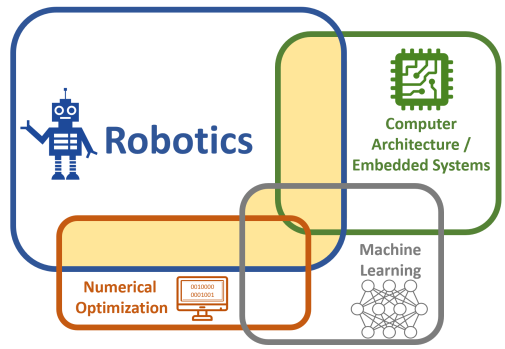

Figure 1: My research is at the intersection of robotics and adjacent fields

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 cost-efficient, dynamic motion planning and control algorithms that can be run online at real-time rates.

My research is focused on enabling these field robots by working at the intersection of robotics and numerical optimization, computer architecture / embedded systems, 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 Model Predictive 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.

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

Hardware-Software Co-Design for Realtime Robust Model Predictive Control

Model Predictive Control (MPC) transforms robot motion planning and control problems into (often non-linear) 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 PhD research has been focused on developing algorithms and implementations that address these issues.

High Computational Cost

To overcome computational issues, 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.

I developed a co-designed implementation of the state-of-the-art Differential Dynamic Programming (DDP) algorithm to target GPUs. In doing so, I explored the benefits and tradeoffs of utilizing both instruction-level and algorithm-level parallelism [8]. I then deployed this approach onto a 7-dof Kuka manipulator, which improved tracking performance despite the presence of model discrepancies and communication delays between the robot and GPU [9]. I also currently have a paper in preparation on a further accelerated GPU solver that exposes more parallelism by using a preconditioned conjugate gradient direct method.

This work revealed that the development of additional co-designed robotics algorithms was hindered by a lack of core robotics kernels for GPUs, FPGAs, and other alternative hardware platforms. This inspired recent collaborative work currently under review, in which we developed a proof-of-concept co-designed implementation of the gradient of rigid body dynamics on a GPU and FPGA. We targeted this kernel as it is present in many robotics algorithms—e.g., it accounts for 30% to 90% of the total computational time of MPC [5,8,10,11].

Non-Robust Convergence

I also worked at the intersection of robotics and both numerical optimization and machine learning to develop algorithms with more robust convergence properties.

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 of the dynamics and cost functions [12].

I also recently supervised an undergraduate thesis that built on recent work leveraging offline reinforcement learning to add robustness to online MPC algorithms [13,14]. 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

I would be excited to build an intersectional robotics program, and I see many opportunities for collaborations on parallel algorithms and architectures.

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 [15]. Unfortunately, these toolboxes only target CPUs for most operations.

Researching at the intersection of robotics and computer architecture, I will develop a complete robotics toolbox for GPUs and FPGAs. This will enable robotics researchers to use these currently inaccessible high-performance parallel architectures to address computational bottlenecks. While my current work 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 explore alternate formulations of dynamics that may expose additional parallelism and performance through co-design [16,17,18,19], and implement additional core kernels (e.g., contact forces/normals, collision detection). Finally, I want to make the toolbox user-friendly through an easy-to-use front-end API in a high-level language (e.g., Python, Julia), backed by code generation tools.

At the intersection of robotics and numerical optimization, I would research and develop a series of MPC algorithms and implementations that that are co-designed to expose additional hardware-friendly computational patterns and are built on top of the toolbox's core kernels. I would also broaden the support of my current implementations to additional objectives and constraints through sparsity exploiting code generation.

I believe that the research needed to develop this toolbox could be tackled by motivated undergraduates, as it can be approached through a series of bite size projects. For example, I recently advised an undergraduate thesis that implemented a state-of-the-art motion planning algorithm on an FPGA for a mobile manipulator, and would be excited to advise similar projects in the future.

Developing deployable and data efficient learning

Machine learning systems suffer from long training times and huge data costs, and often the results cannot be transferred to hardware [20]. 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 leverage their respective strengths. Potential projects include using actor-critic methods to learn hyperparameters, trajectory initializations, and objective function regularizers that improve MPC convergence and avoid "bad" local minima, as well as learning high level task parameters for execution by MPC agents.

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.

All of these projects would require some machine learning background, but students can learn much of the prerequisite knowledge in lab, as these approaches leverage extensions of classical techniques.

Tiny Robotics: lowering the cost of robotics for widespread deployment

Inspired by the recent growth in TinyML and the development of low cost, palm size robots like the Bittle, I would be excited to develop a Tiny Robotics research program at the intersection of robotics and embedded systems. This program will explore the possibility of deploying these planning and control techniques onto microcontrollers and will require the development of algorithms that provide better numerical conditioning to support reduced precision fixed point operations.

At the intersection of robotics and machine learning, I would also like to leverage autoencoders to reduce the dimensionality of these optimization problems to help them fit on tiny devices. I believe that well designed autoencoders would provide a compressed latent representation that captures the full nonlinear dynamics better than template based models [21,22], improving end-to-end performance.

I believe that Tiny Robotics is particularly well suited for both undergraduate research projects, class assignments / labs, and outreach programs due to its low cost and low barriers to entry. I would be excited to explore using this research in all of those settings.

Scaling for Diversity in STEM Education

Finally, I hope to continue to develop STEM Education research. I helped launch a tuition-free, project-based robotics summer program for high school students in 2016 [23]. Since then we have expanded the program from one course with 40 students, centered around self driving cars, to over 10 courses with over 250 total students, focused on a range of topics such as UAVs, 3D-Printing, and Cyber Security. We now have a paper in preparation detailing the learnings from scaling the program and working to improve diversity and inclusion in the student body. I am also working with researchers at UCL to explore using TinyML to develop low-cost tools to monitor student engagement and improve learning outcomes globally. At Wiliams, I hope to continue to explore new and innovative STEM learning models and outreach programs to improve student access and outcomes.

References

[1] G. Z. Yang, et al., "Combating covid-19—the role of robotics in managing public health and infectious diseases," 2020.

[2] J. Shaw, "The coming eldercare tsunami," Harvard Magazine, Jan 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 2012 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 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems.

[6] S. Kuindersma, et al., "Optimization-based locomotion planning, estimation, and control design for Atlas." Technical Report, 2017.

[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).

[8] B. Plancher and S. Kuindersma, "A Performance Analysis of Parallel Differential Dynamic Programming on a GPU," in 2018 International Workshop on the Algorithmic Foundations of Robotics (WAFR).

[9] 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 the 2019 International Conference on Robotics and Automation (ICRA)

[10] 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).

[11] J. Carpentier and N. Mansard, “Analytical derivatives of rigid body dynamics algorithms,” in Robotics: Science and Systems, 2018.

[12] B. Plancher, Z. Manchester, and S. Kuindersma, “Constrained Unscented Dynamic Programming,” in 2017 Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

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

[14] F. Farshidian, D. Hoeller, and M. Hutter, “Deep value model predictive control,” 2019.

[15] R. Tedrake and the Drake Development Team, “Drake: A planning, control, and analysis toolbox fornonlinear dynamical systems.” http://drake.mit.edu

[16] R. Featherstone, “A divide-and-conquer articulated-body algorithm for parallel O(log(n)) calculation of rigid-body dynamics,” in The International Journal of Robotics Research, 1999.

[17] K. Yamane and Y. Nakamura, “Comparative Study on Serial and Parallel Forward Dynamics Algorithms for Kinematic Chains.” in The International Journal of Robotics Research, 2009.

[18] Y. Yang, Y. Wu, and J. Pan, “Parallel Dynamics Computation Using Prefix Sum Operations.” in IEEE Robotics and Automation Letters, 2017.

[19] J. Brudigam and Z. Manchester, “Linear-time variational integrators in maximal coordinates,” in 2020 International Workshop on the Algorithmic Foundations of Robotics (WAFR).

[20] 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).

[21] R. J. Full and D. E. Koditschek, “Templates and anchors: neuromechanical hypotheses of legged locomotion on land,” Journal of Experimental Biology, 1999.

[22] 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.

[23] 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).