A Survey of the Implementation of Linear Model Predictive Control on FPGAs

Ian McInerney[1*], George A. Constantinides[1], Eric C. Kerrigan[1,2]

[1]Department of Electrical and Electronic Engineering, Imperial College London
[2]Department of Aeronautics, Imperial College London

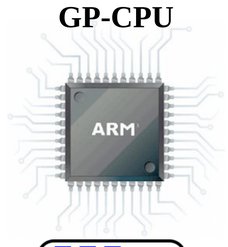
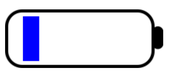
Optimal Control Problem

- Constrained Linear Quadratic Regulator

$$\begin{aligned} \underset{u,x}{\text{minimize}}\ & \|x_N\|_P^2 + \sum_{k=0}^{N-1}\left(\|x_k\|_Q^2 + \|u_k\|_R^2\right) && \}\text{Quadratic cost function} \\ \text{subject to}\ & x_{k+1} = Ax_k + Bu_k,\ k = 0,\dots N-1 && \}\text{Linear prediction model} \\ & Fu_k \leq c_u,\ k = 0,\dots N-1 && \}\text{Input constraints} \\ & Dx_k \leq c_x,\ k = 1,\dots N && \}\text{State constraints} \end{aligned}$$

- Solution found by solving a convex quadratic program

Computing Devices

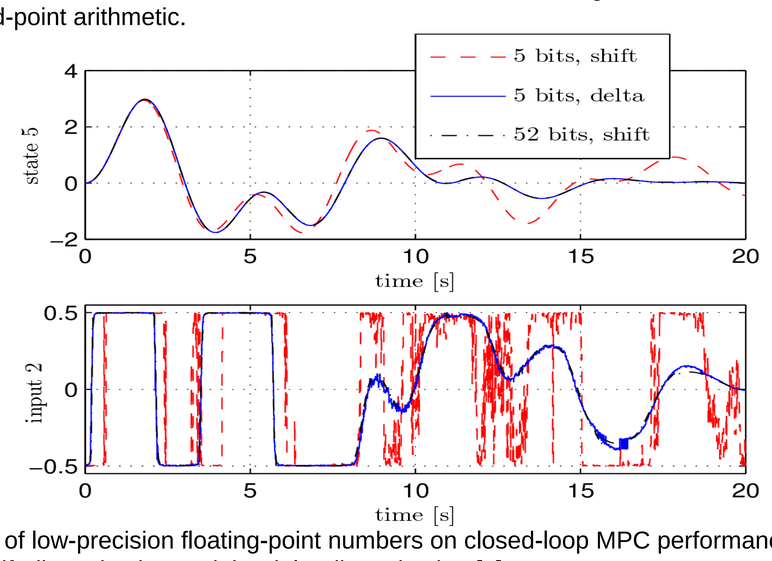
	GP-CPU	GPU	FPGA
			
Power Requirements			
Architecture	Sequential	Parallel	Customizable
Peak FLOPS	100M-100G	100G-3T	10T
MPC Research	Popular	Limited	Popular

Number Representation

- The closed-loop performance of MPC may degrade in reduced-precision
- Algorithms must be stable under reduced-precision
- Prefer algorithms with bounds on variable values to reduce likelihood of overflow in fixed-point arithmetic

	Fast Gradient Method	Alternating Direction Method of Multipliers	Dual Gradient Projection	Proximal Newton Method
Stable Under Roundoff Errors	✓	✓	✓	✓
Data-Type Size Rules	✓	✓	✓	✓
Upper Iteration Bound	✓	✗	✓	✓

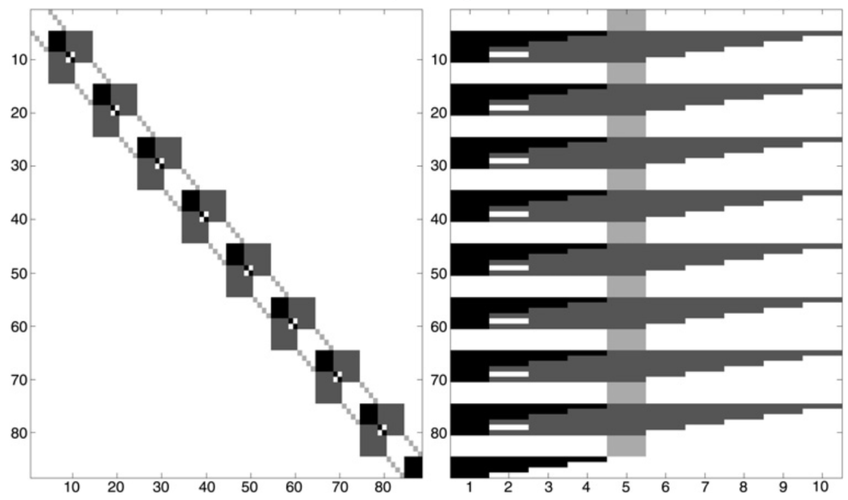
The existence of theoretical results for various first-order algorithms when used in fixed-point arithmetic.



Effect of low-precision floating-point numbers on closed-loop MPC performance using the shift discretization and the delta discretization [1]

Memory Architectures

- Exploit structure to reduce memory usage and improve scalability

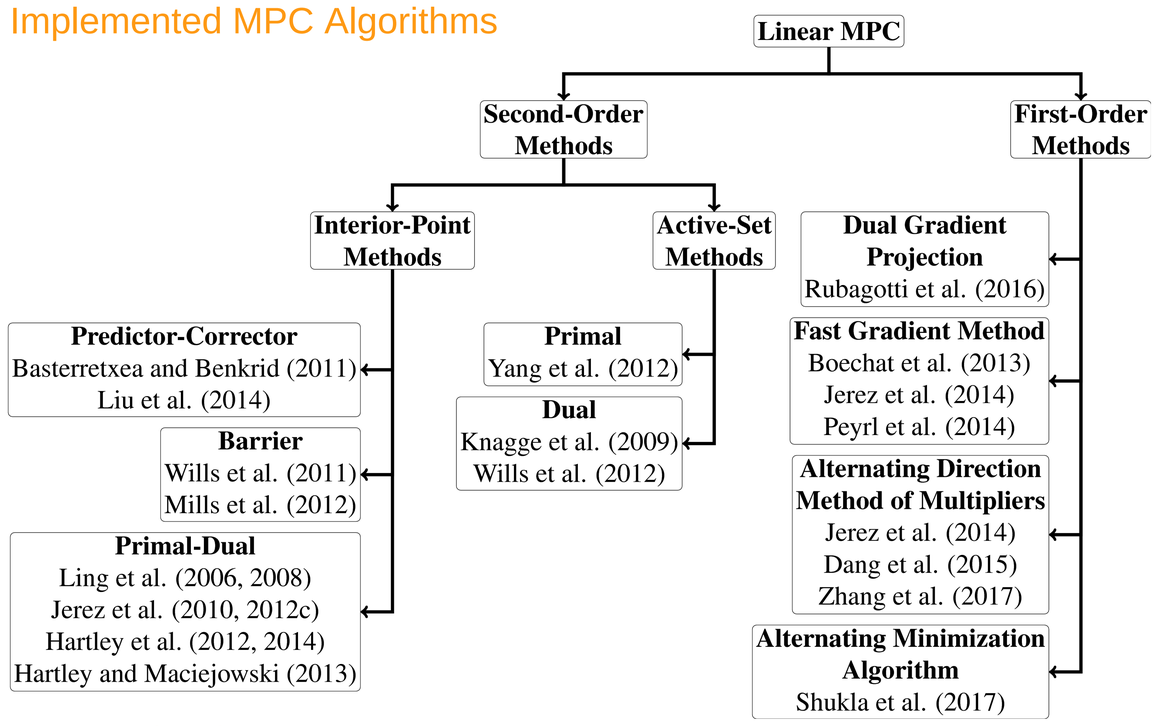


Sparse KKT matrix (left) and a possible storage pattern (right) that only stores the non-zero diagonals [3]

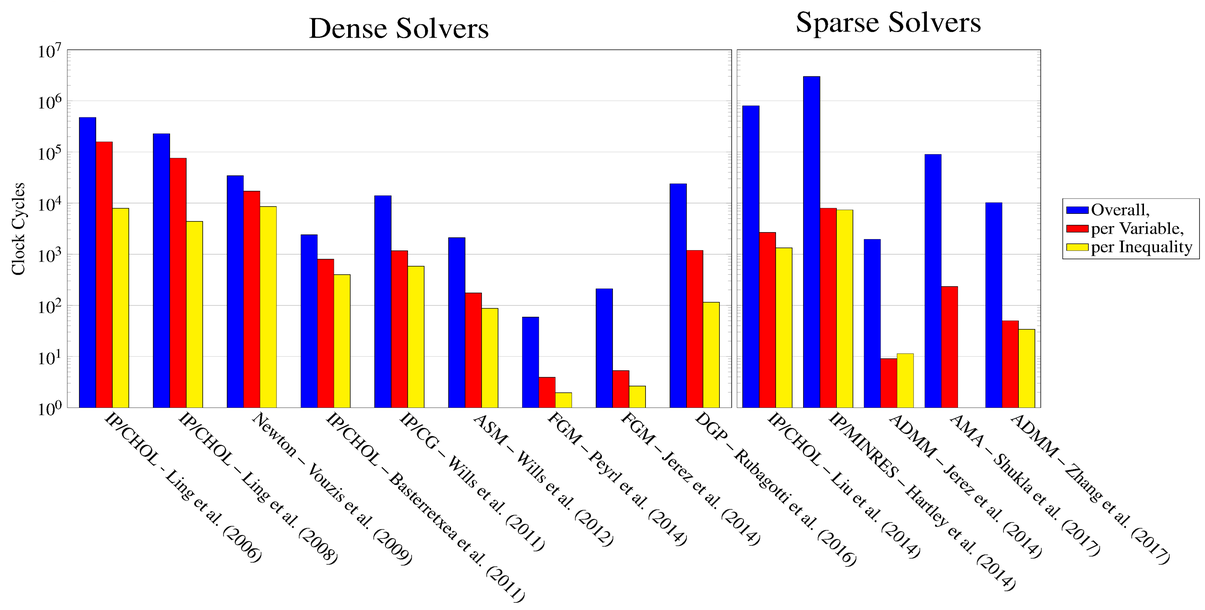
ACKNOWLEDGEMENTS

The support of the EPSRC Centre for Doctoral Training in High Performance Embedded and Distributed Systems (HiPEDS, Grant Reference EP/L016796/1) is gratefully acknowledged.

Implemented MPC Algorithms

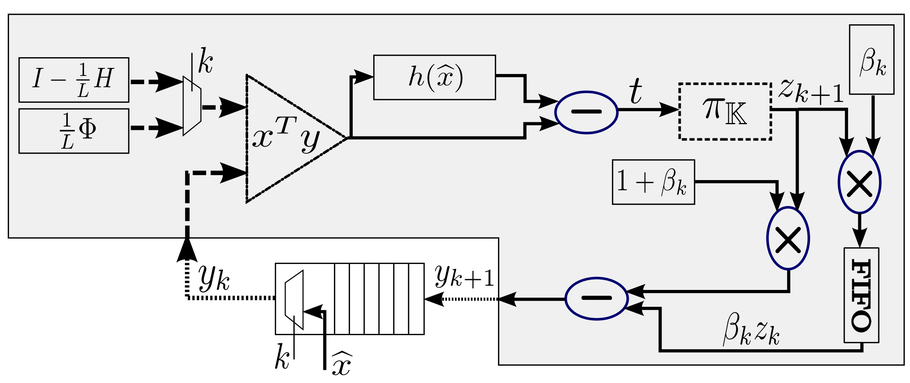


Reported Solver Performance



Computational Units

- Low-level linear algebra easily parallelized for both first-order and second-order methods
- Algorithm-level parallelizations implemented primarily for first-order methods



Computational unit for a single variable in a Fast Gradient Method Implementation [2]. Multiple units can be used to process multiple variables in parallel.

Future Research Directions

- User-friendly code-generation utilities for FPGAs
- Using High-Level Synthesis tools for implementations
- Hardware-software co-design optimization
- Heterogeneous processor systems & computational resource allocation

Open questions for nonlinear MPC on FPGAs

- What algorithmic parallelizations can be exploited?
- How does reduced-precision affect nonlinear discretization schemes?
- How can mesh-refinement be efficiently handled on embedded systems?
- How can nonlinear functions be implemented efficiently?
- How should the matrices be structured in memory to improve performance and scalability?

REFERENCES

[1] S. Longo, E. C. Kerrigan, and G. A. Constantinides, “Constrained LQR for low-precision data representation,” Automatica, vol. 50, no. 1, pp. 162–168, 2014.

[2] J. L. Jerez, P. J. Goulart, S. Richter, G. A. Constantinides, E. C. Kerrigan, and M. Morari, “Embedded Online Optimization for Model Predictive Control at Megahertz Rates,” IEEE Transactions on Automatic Control, vol. 59, no. 12, pp. 3238–3251, 2014.

[3] J. L. Jerez, K. V. Ling, G. A. Constantinides, and E. C. Kerrigan, “Model predictive control for deeply pipelined field-programmable gate array implementation: algorithms and circuitry,” IET Control Theory and Applications., vol. 6, no. 8, pp. 1029–1041, 2012.