Improving the speed, reliability and utility of Model Predictive Control by use of dataflow techniques

Ruirui Gu, Shuvra Bhattacharyya, and William Levine

University of Maryland — The Institute for Systems Research

Motivation

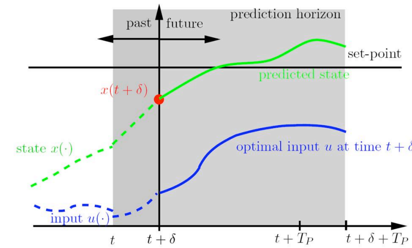
Model Predictive Control (MPC) is an advanced control technique

- Pros:
  - MPC provides an organized methodology for designing multi-input multi-output control systems
  - MPC has the ability to deal with saturation
  - MPC makes intuitive sense to controller designers
- Cons:
  - Computationally intensive
  - Not practical for fast real time applications

MPC is used in applications with slow dynamics. Can we expand the range to include applications with fast dynamics?

Motivation: The practical utility of MPC would be enhanced by faster computations

Model predictive control



Existing research on MPC:
optimization of MPC algorithms; offline look up table

Our method:
Solution to provide fast online implementation, which benefits real time applications

Dataflow modeling

Dataflow based methodology is well developed for embedded systems, which provides trade-offs among system performance and resource management

Reactive control integrated dataflow model is especially developed for MPC

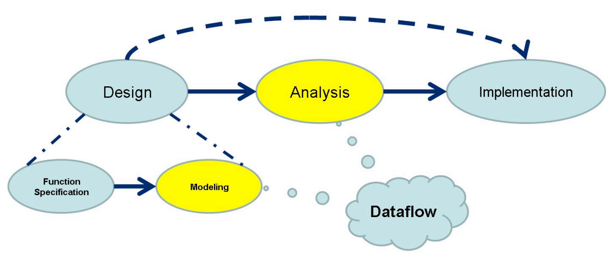
- Capable of describing control structures
- Mutually Exclusive Edge is key concept in RCDF model
  - Mutually Exclusive Production Edge (METP)

$$\sum_{i=1}^{m} I(prd(e_i, j)) = 1, \text{ where } I(x) = \begin{cases} 1, & x > 0 \\ 0, & x \le 0 \end{cases}$$

  - Mutually Exclusive Consumption Edge (METC)

$$\sum_{i=1}^{m} I(cns(e_i, j)) = 1, \quad \text{where } I(x) = \begin{cases} 1, & x > 0 \\ 0, & x \le 0 \end{cases}$$

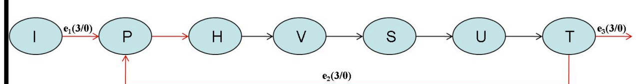
Optimization framework



Case study: Newton-KKT method to solve quadratic programming (QP)

$$\text{minimize } \frac{1}{2} < x, Qx > + < c, x >, \text{ s.t. } Ax \le b, x \in R^n$$

Newton-KKT in dataflow domain

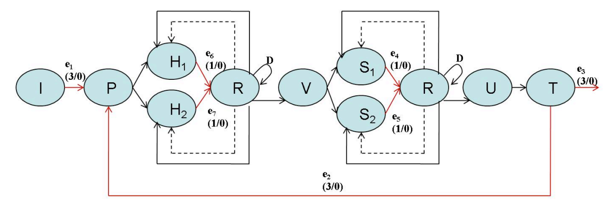


METP Edges: $\{e_2, e_3\}$; METC Edges: $\{e_1, e_2\}$.

I—InitializeValue; P—ParameterCalculation; H—HessianCalculation; V—MiddleValueCalculation; S—SearchDirection; U—Update; T—StoppingCriterion;

- Profiling and identifying bottlenecks:
  - H, S and U are bottlenecks to be resolved
- Troubleshooting:
  - H, S: Sequential version Vs. Multiple version Parallelism
  - U: Sequential version Vs Data parallelism transformation

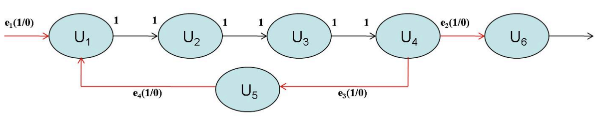
Multi version parallelism



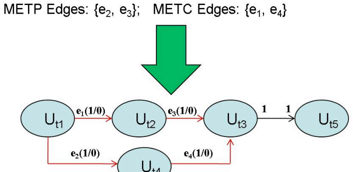
METP Edges: $\{e_2, e_3\}$;
METC Edges: $\{e_1, e_2\}$; $\{e_4, e_5\}$; $\{e_6, e_7\}$.

I—InitializeValue; P—ParameterCalculation; H_1—HessianCalculationOption1; H_2—HessianCalculationOption2; R—MVOS; V—MiddleValueCalculation; S_1—SearchDirectionOption1; S_2—SearchDirectionOption2; U—Update; T—StoppingCriterion;

Data parallelism transformation

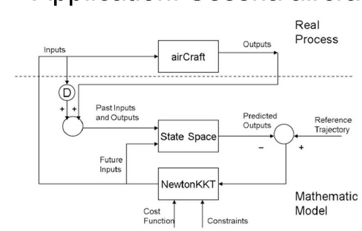


METP Edges: $\{e_2, e_3\}$; METC Edges: $\{e_1, e_4\}$



METP Edges: $\{e_1, e_2\}$; METC Edges: $\{e_3, e_4\}$.

Application: Cessna aircraft



$T_s(sec)$	original	PM+DPV	improve
0.5	0.0859375	0.0281250	67.2%
0.05	0.0945312	0.0291125	69.2%

Conclusions

In terms of control algorithm:

- An abstract model was proposed for the control algorithms used in MPC
- We analyzed the model for performance bottlenecks and methods to improve performance
- Different forms of parallelism were introduced to improve the system performance

In terms of MPC:

- Improved system performance
- Ensured the system reliability
- Made MPC practically useful for a larger class of real time applications