Accelerating Predictive Model Control Applications on Manycores

Koji Inoue

Kyushu University

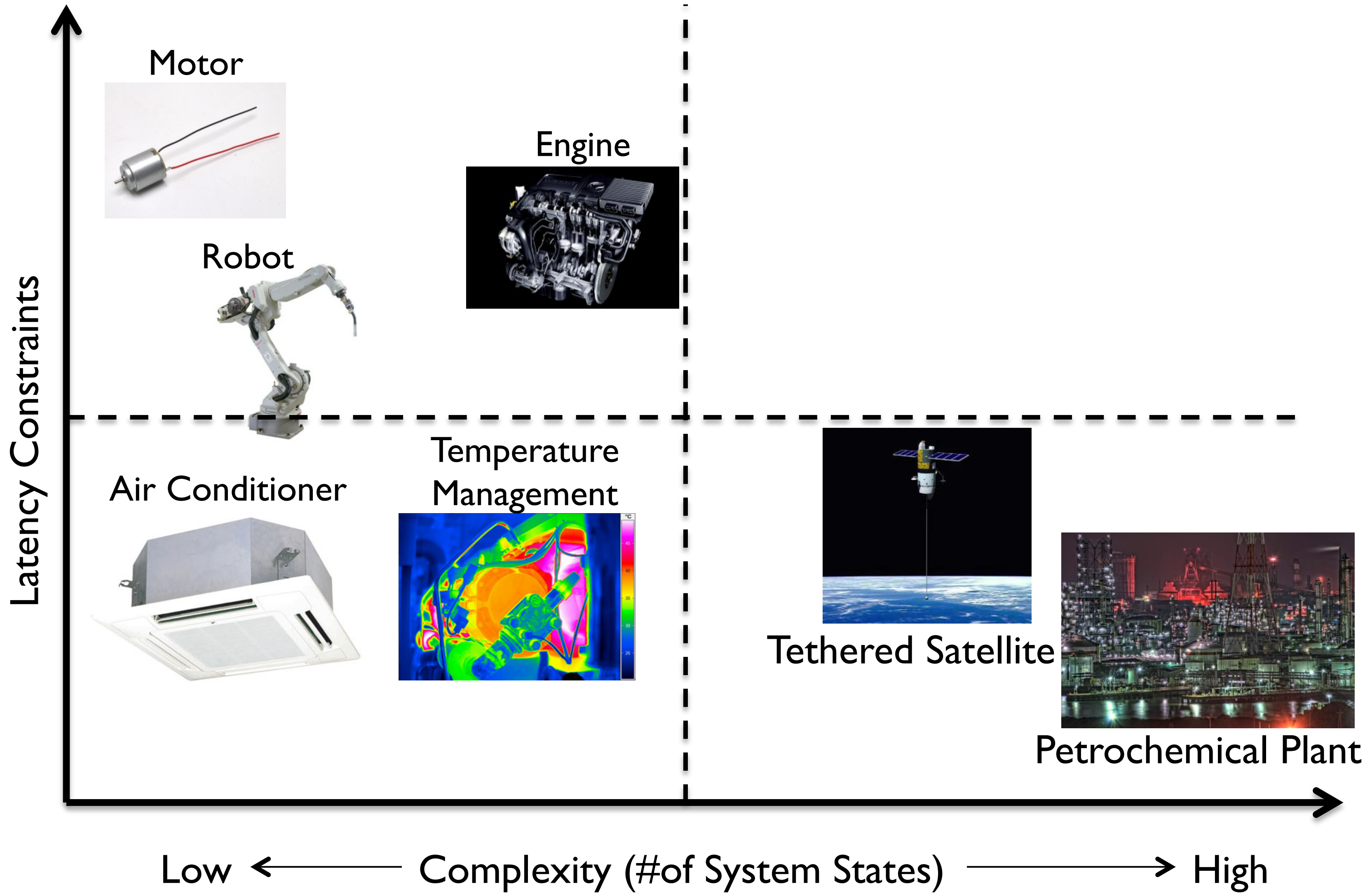
Outline

- Background
- Our Target
- Speculative MPC Execution w/ Input Value Prediction
- Preliminary Evaluation
- Conclusions

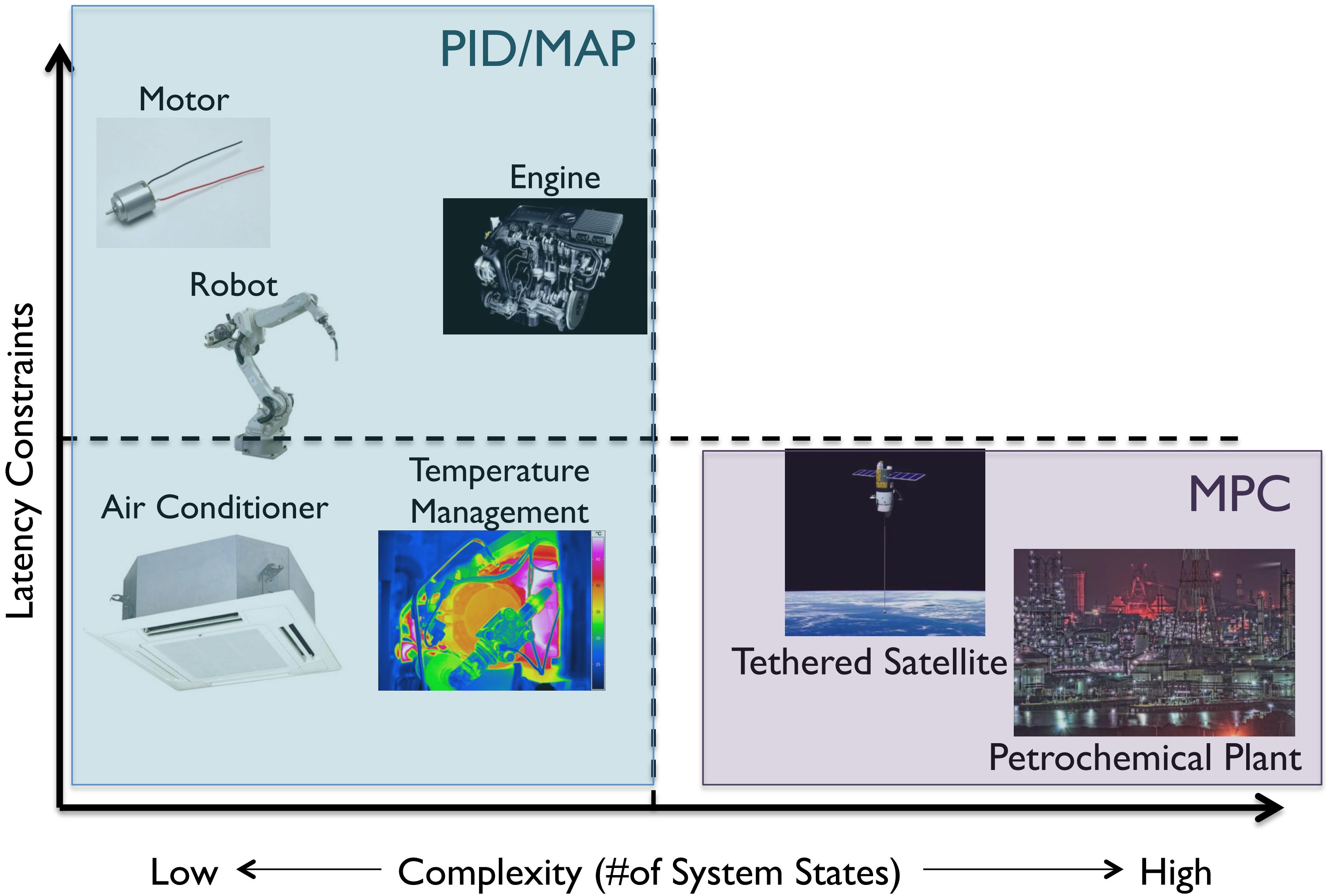
Outline

- Background
- Our Target
- Speculative MPC Execution w/ Input Value Prediction
- Preliminary Evaluation
- Conclusions

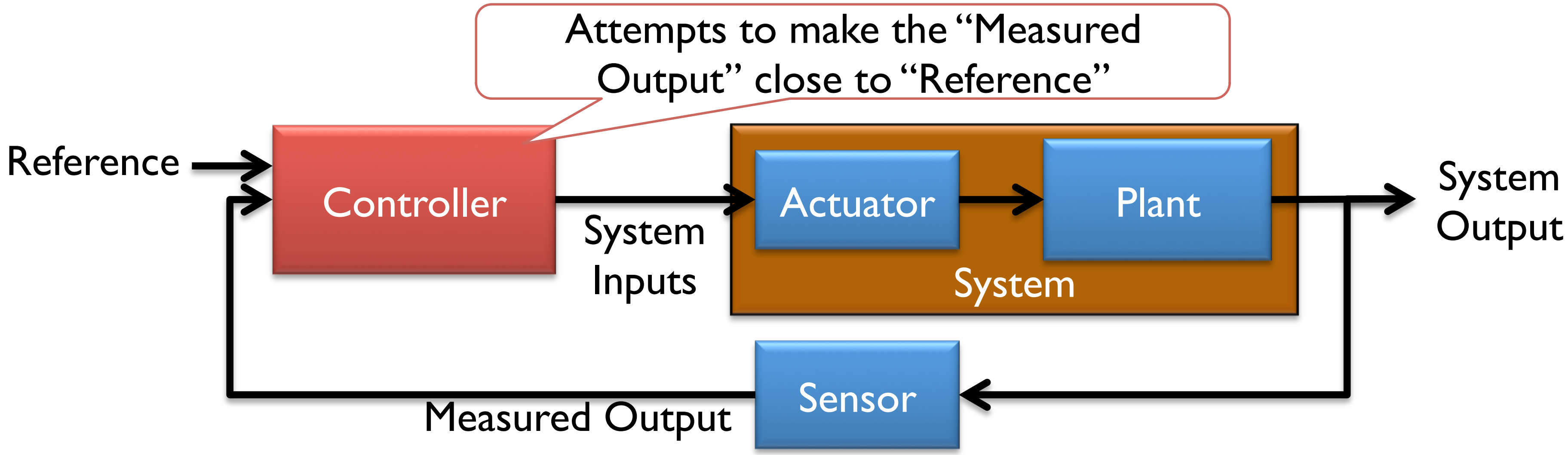
Real-Time Control Systems



Real-Time Control Systems



Feedback Control Systems (1/2)



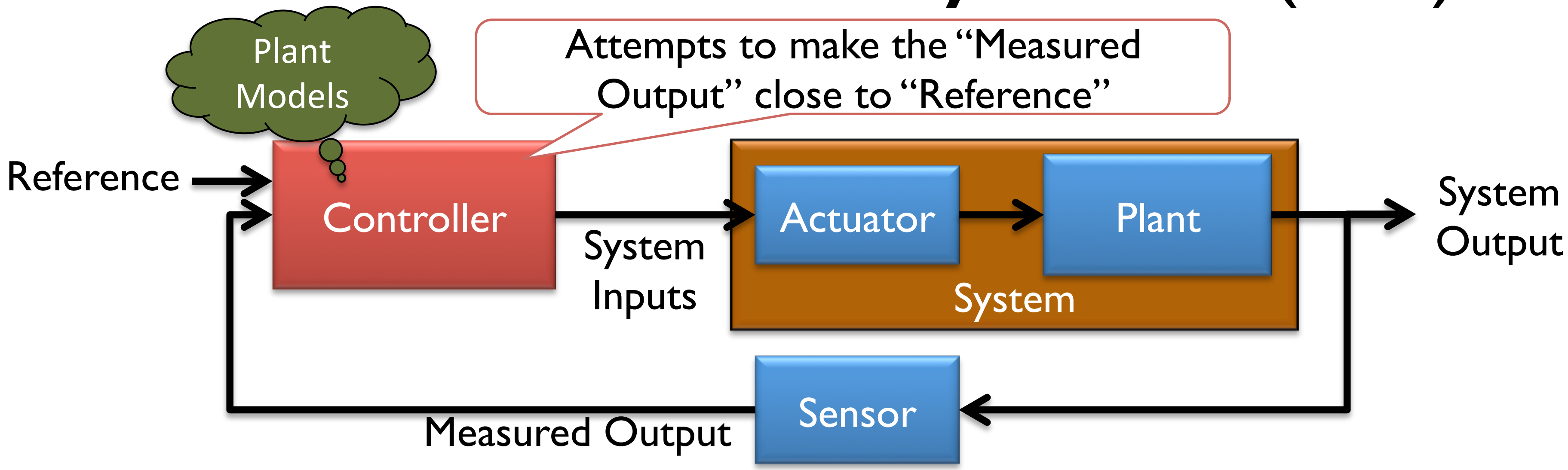
PID Control

- Features
  - Calculation-based Control
  - Involves three separate parameters, the Proportional, Integral, and Derivative values
  - Widely used in industrial control systems
- Problem
  - Can be used only for one-input one-output simple systems

MAP Control

- Features
  - Look-up based Control
  - Obtain an appropriate manipulation for each sequence of system inputs, disturbance, and system outputs, and prepare look-up tables
  - System identification methods with plant test results can be used
- Problem
  - Require huge memory space

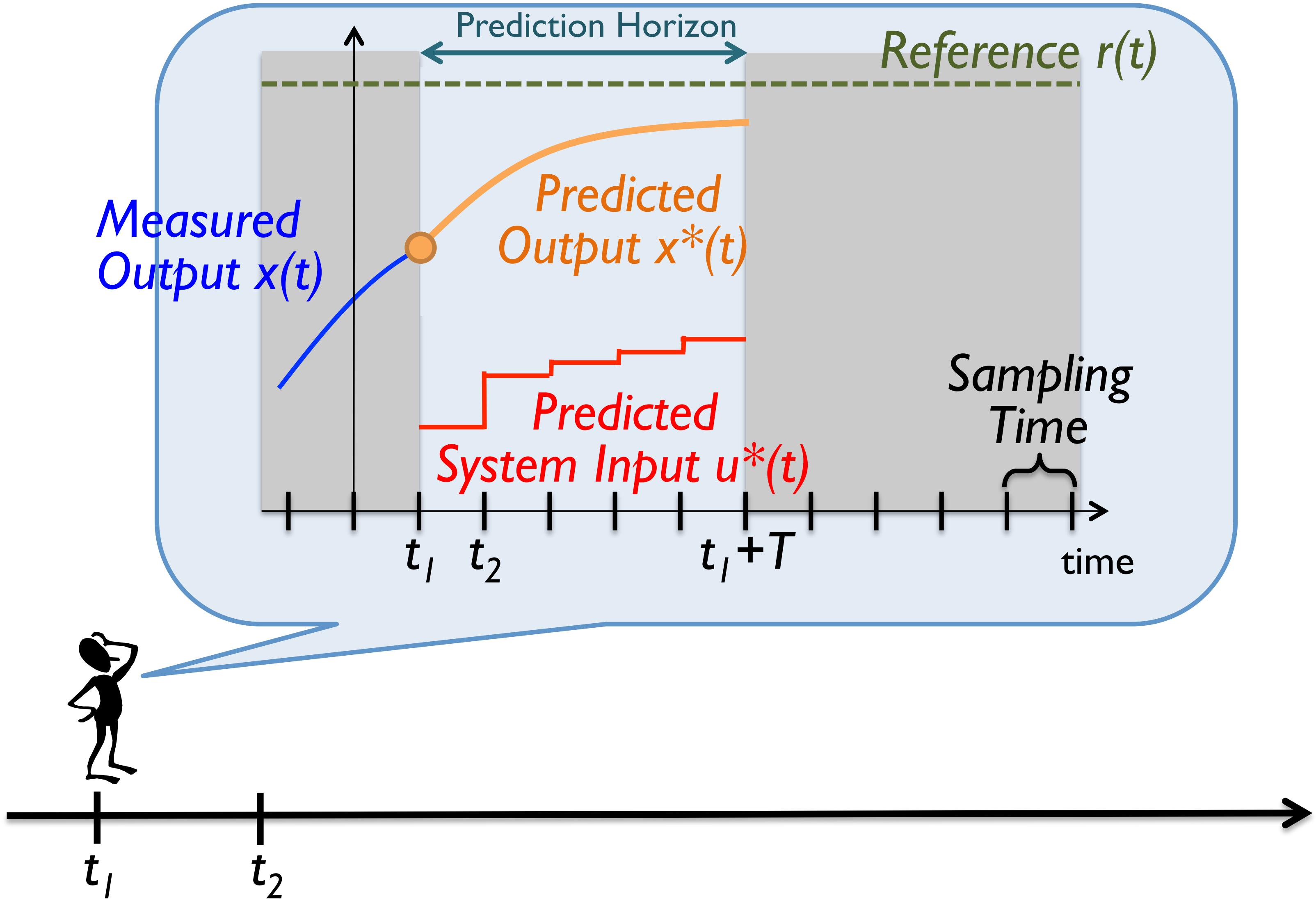
Feedback Control Systems (2/2)



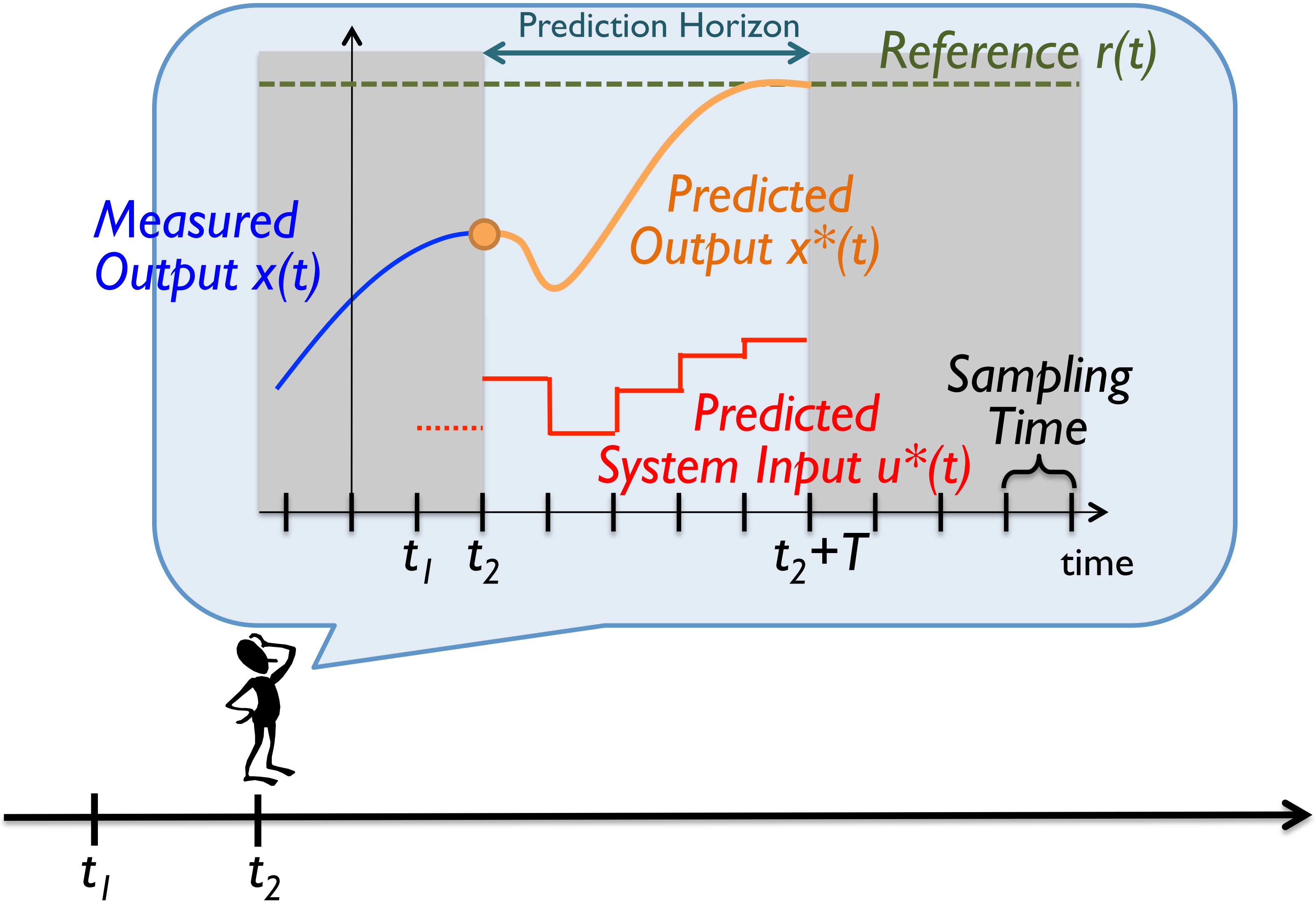
Real-Time Model Predictive Control (MPC)

- Features
  - Involves state equations (or plant models) in the controller
  - Decide how to manipulate the actuator based on predicted future plant behavior by solving optimal control problems
  - Needs to satisfy real-time periodical operations
- Problem
  - $O(N^3)$ computational complexity

Overview of Model Predictive Control



Overview of Model Predictive Control

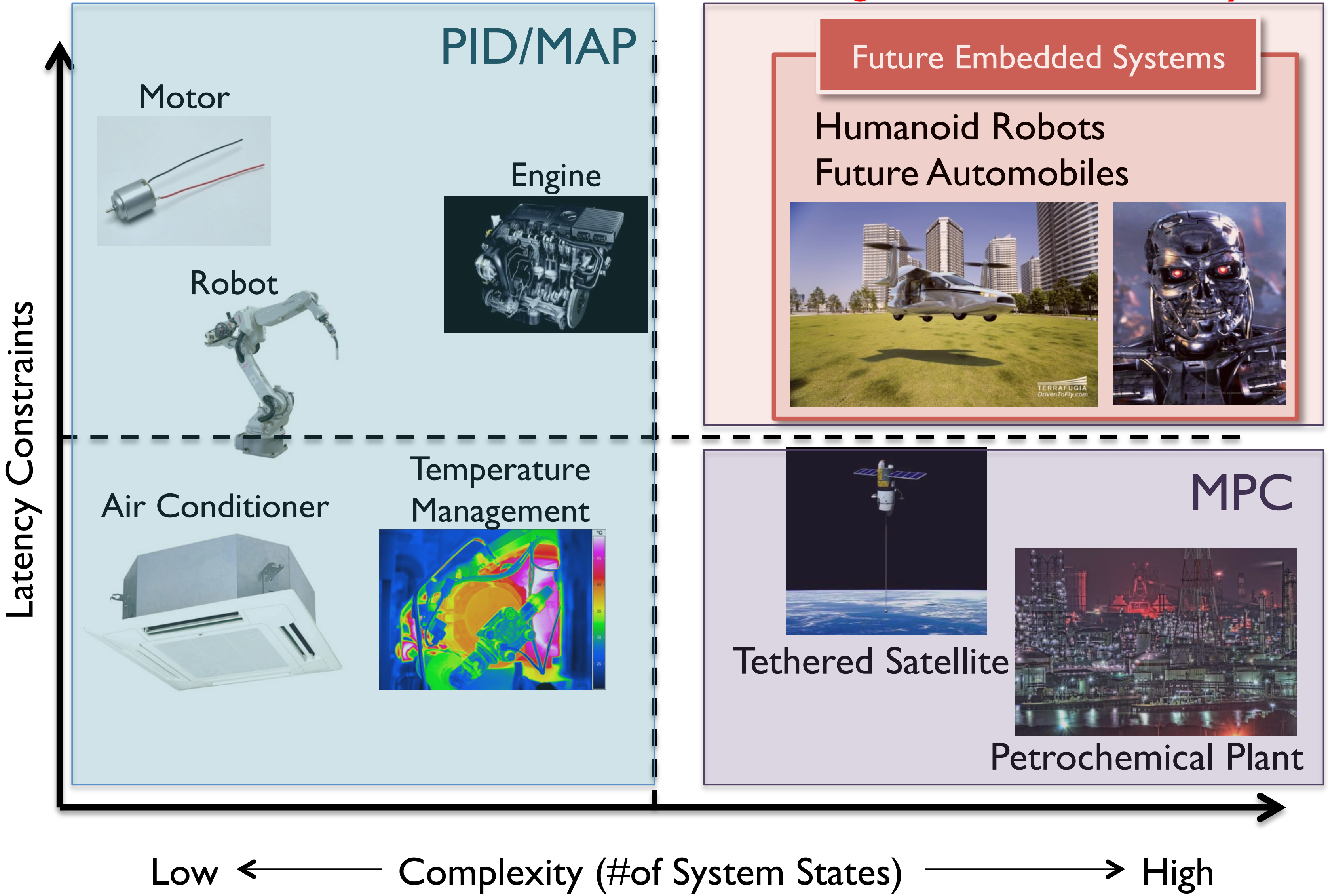


Outline

- Background
- Our Target
- Speculative MPC Execution w/ Input Value Prediction
- Preliminary Evaluation
- Conclusions

Trends in Real-Time Control Systems

Our Target : MPC on Manycore!



Outline

- Background
- Our Target
- Speculative MPC Execution w/ Input Value Prediction
- Preliminary Evaluation
- Conclusions

Do Conventional Parallel Approaches Work Well?

Data-Level Parallelism

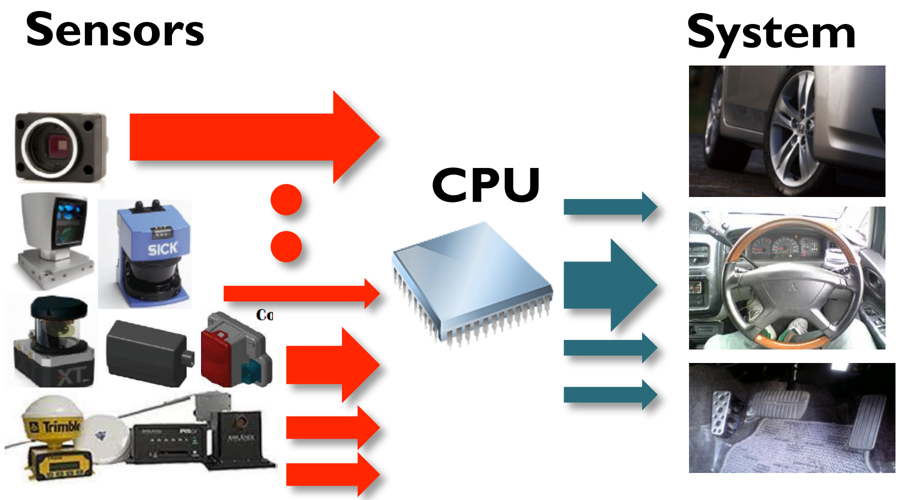
→NO!

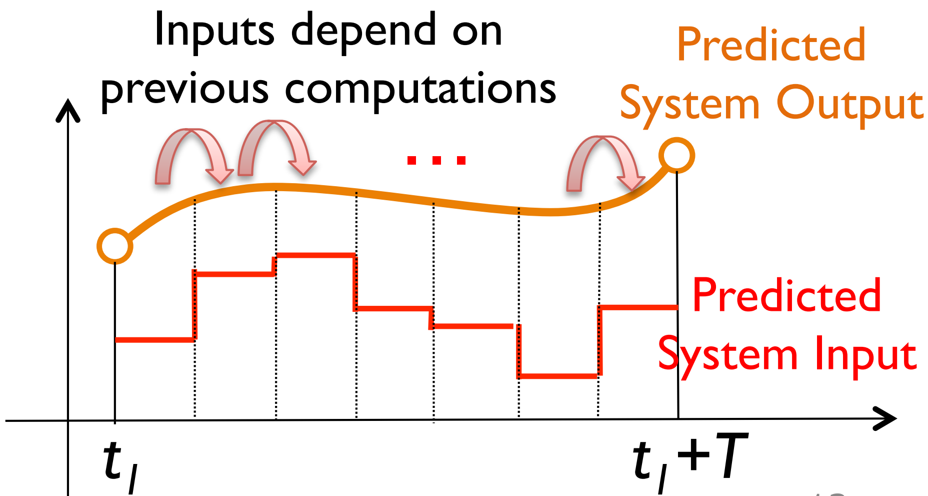
Small size data are fed sequentially!

Thread-Level Parallelism

→NO!

Each step is executed sequentially!

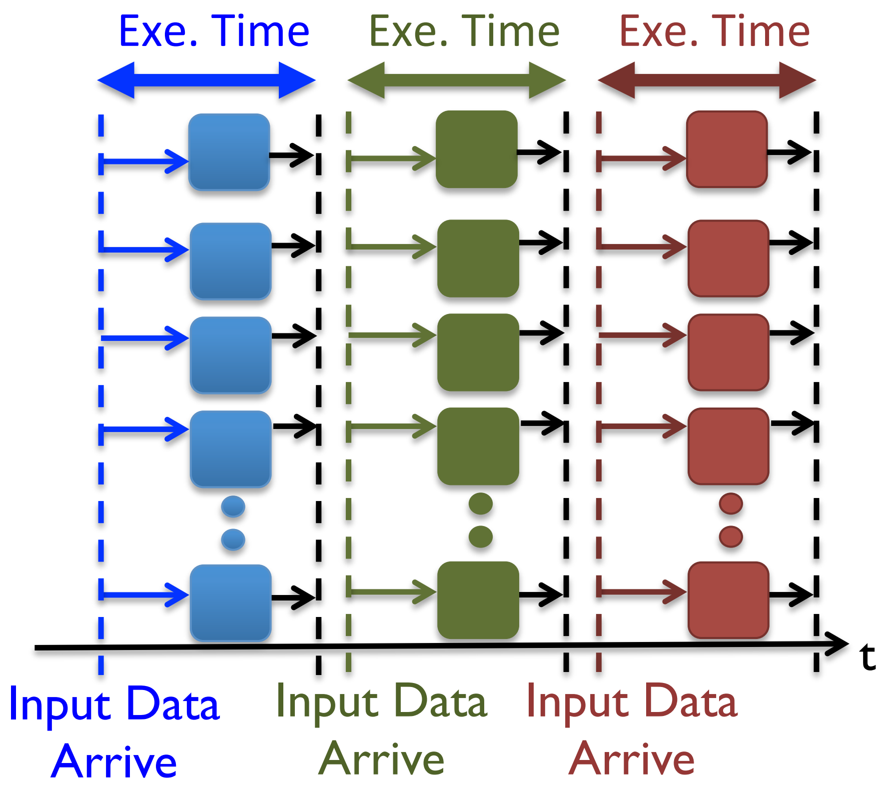




Our Approach
~ Speculative Execution on Manycore ~

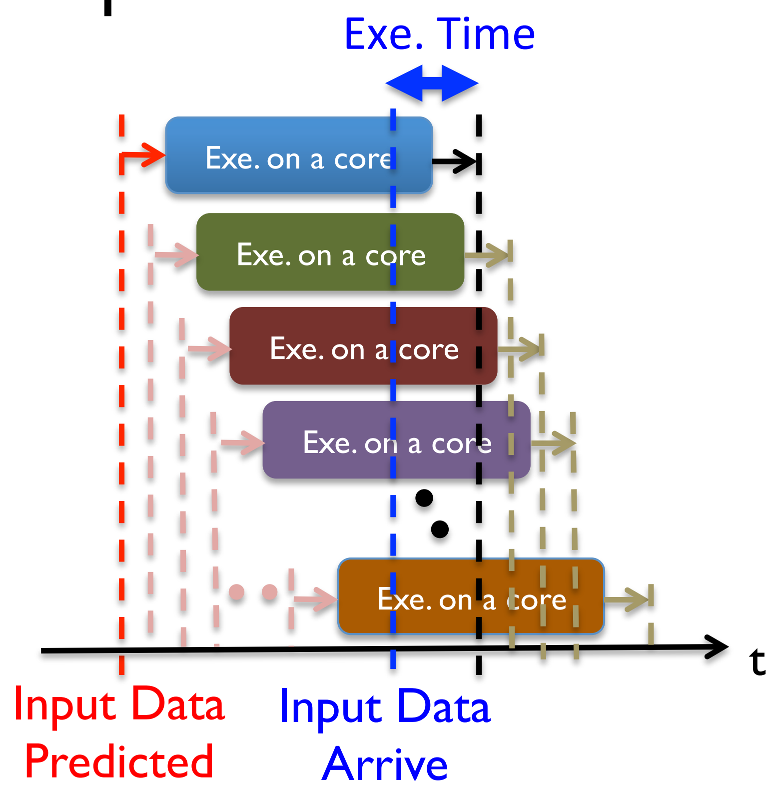
[Conventional]

Spatial Parallel Exe.



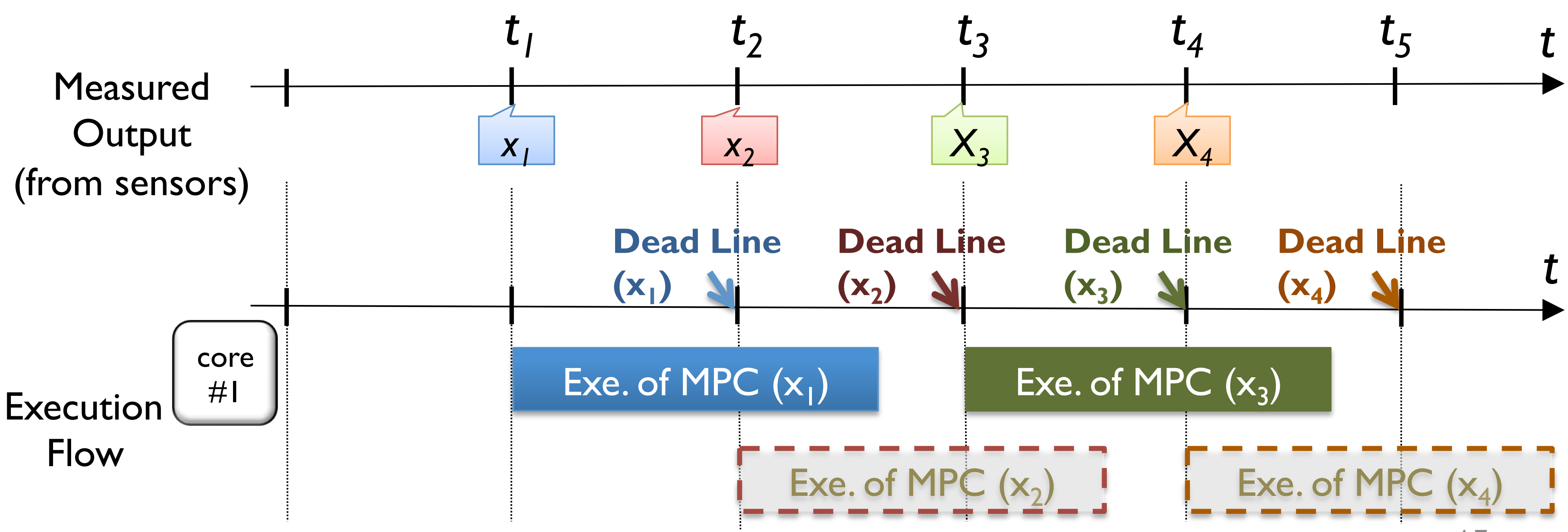
[Proposed]

Temporal Parallel Exe.
w/ Input Value Prediction

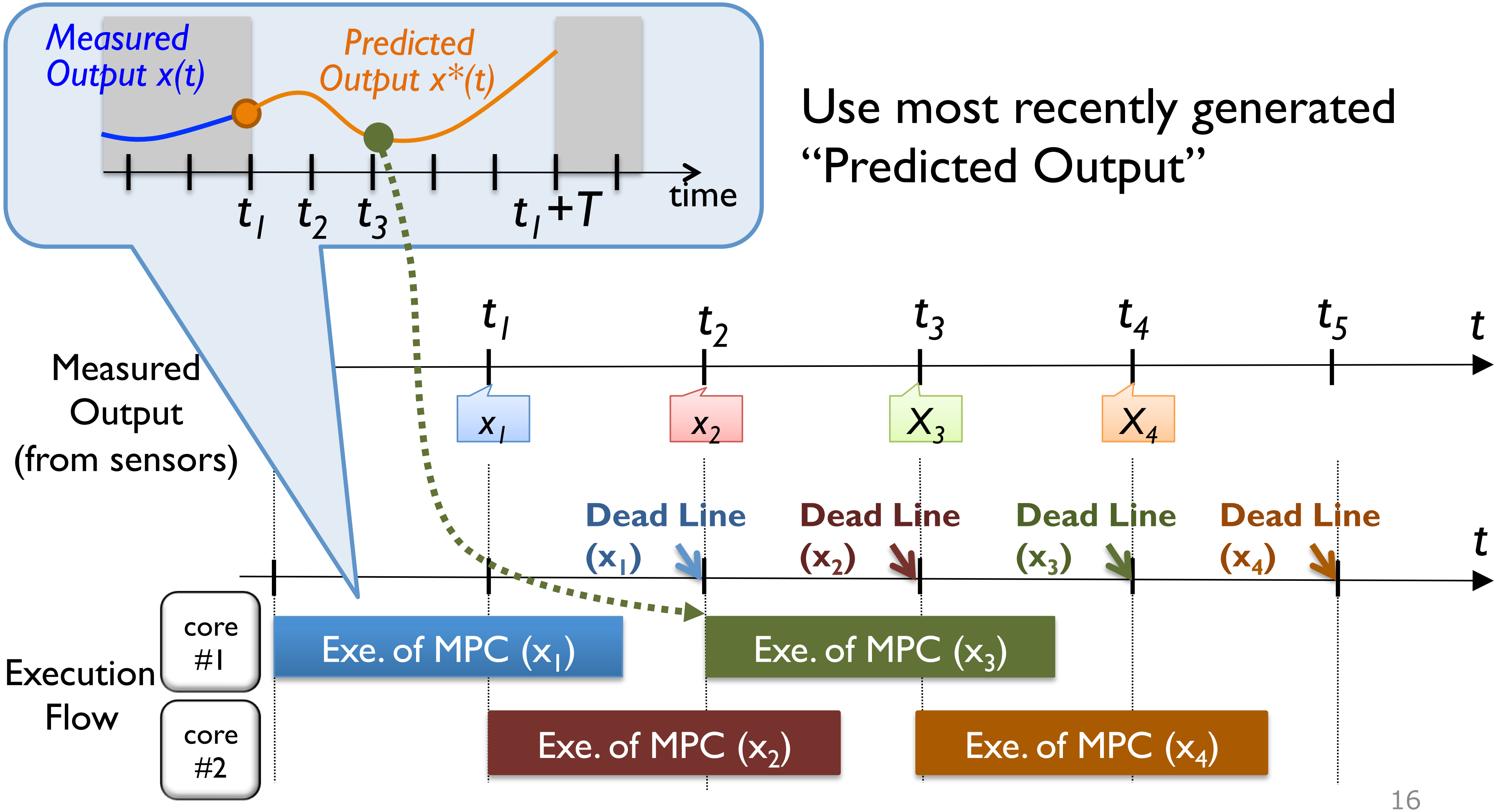


Conv. Single-Core Execution Fails!

Can NOT satisfy RT requirement



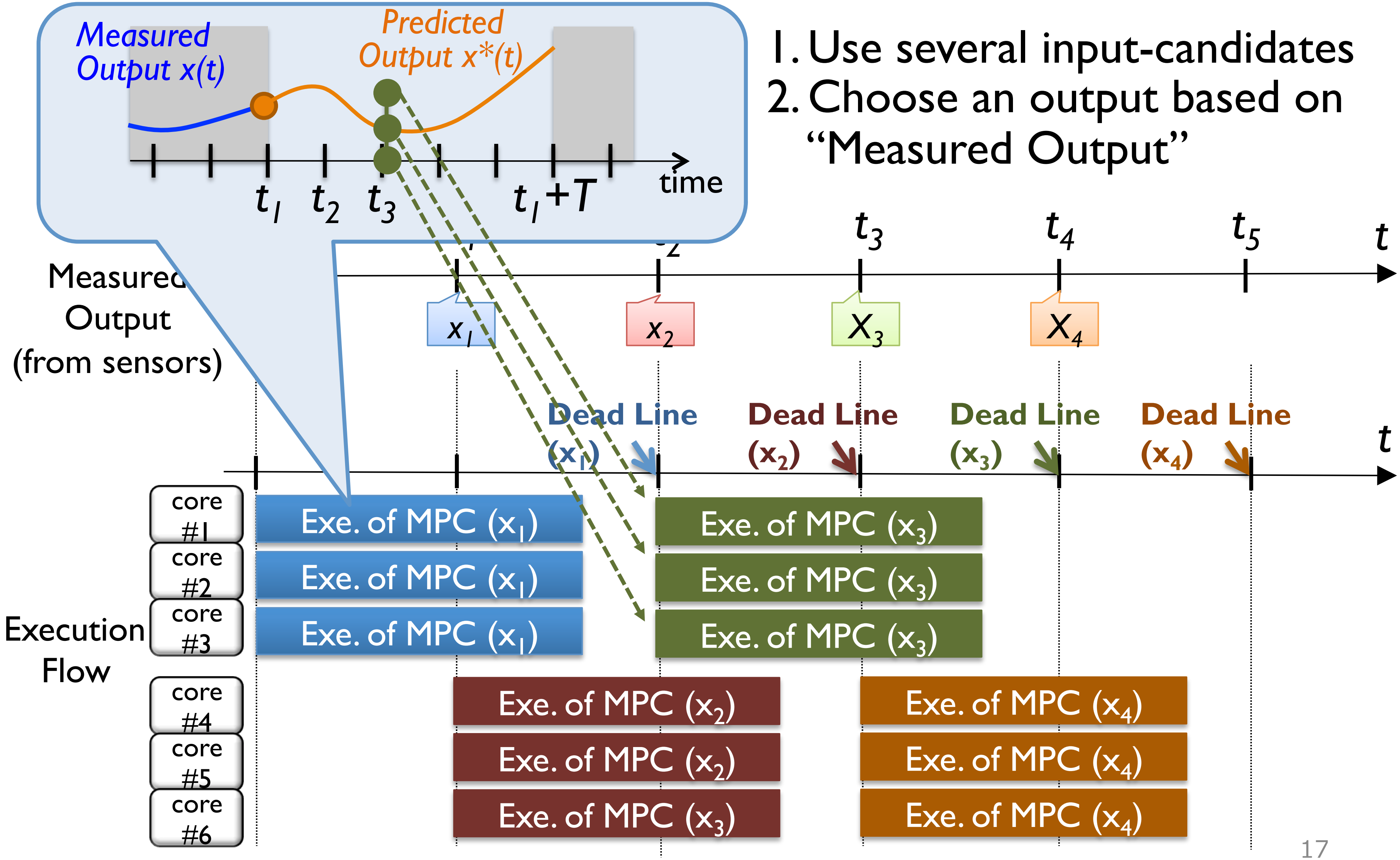
How Can We Predict Input Data for Speculative Executions?



Use most recently generated "Predicted Output"

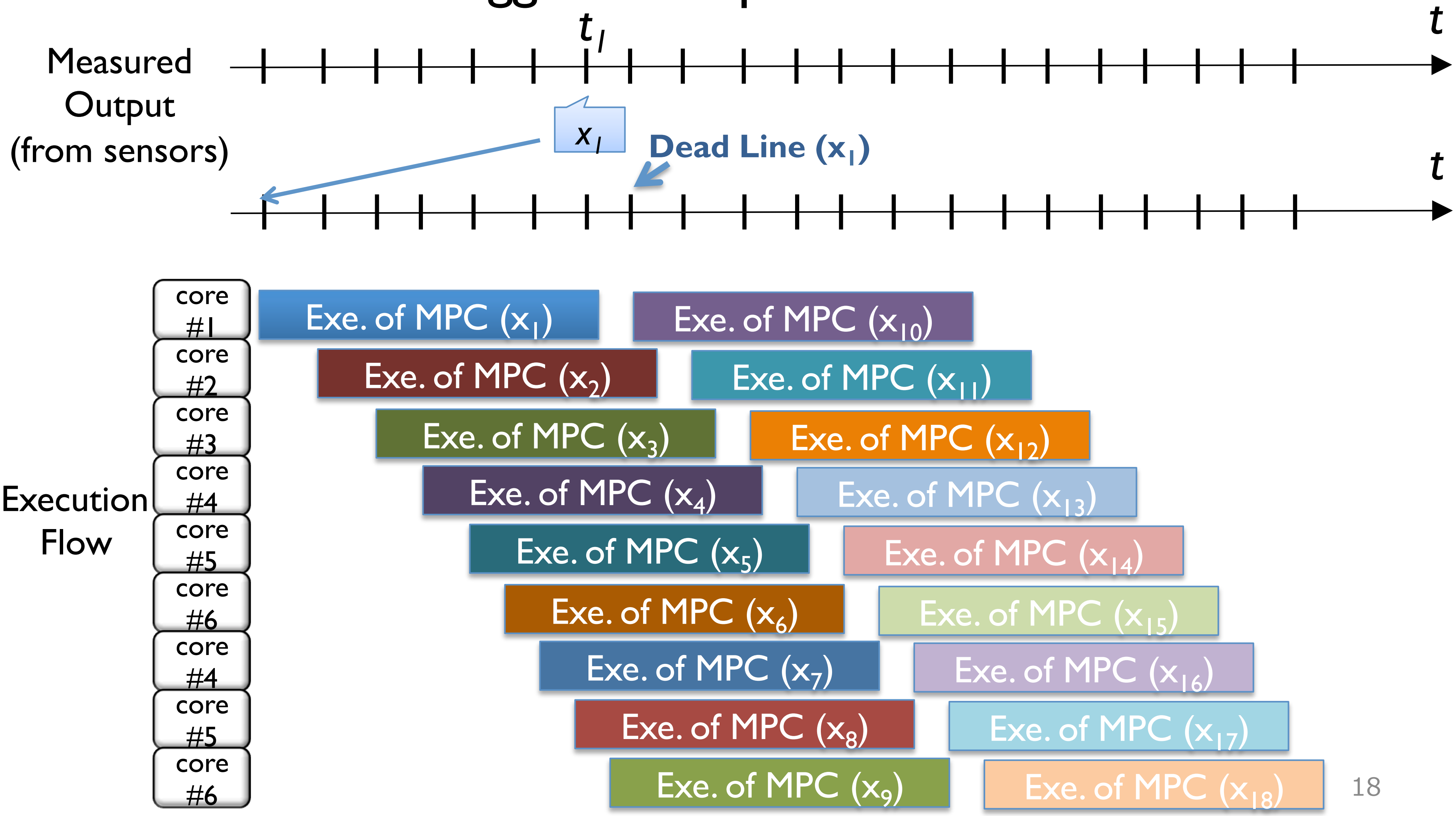
How Can We Improve Prediction Accuracy?

1. Use several input-candidates
2. Choose an output based on "Measured Output"



How Can We Obtain More Performance?

Aggressive Speculation!

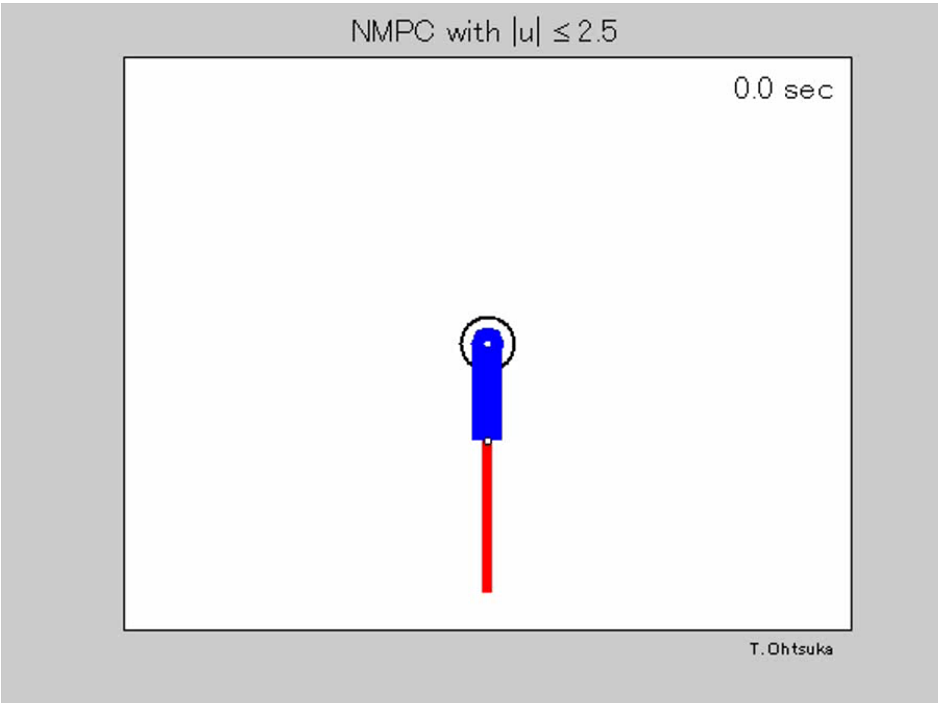


Outline

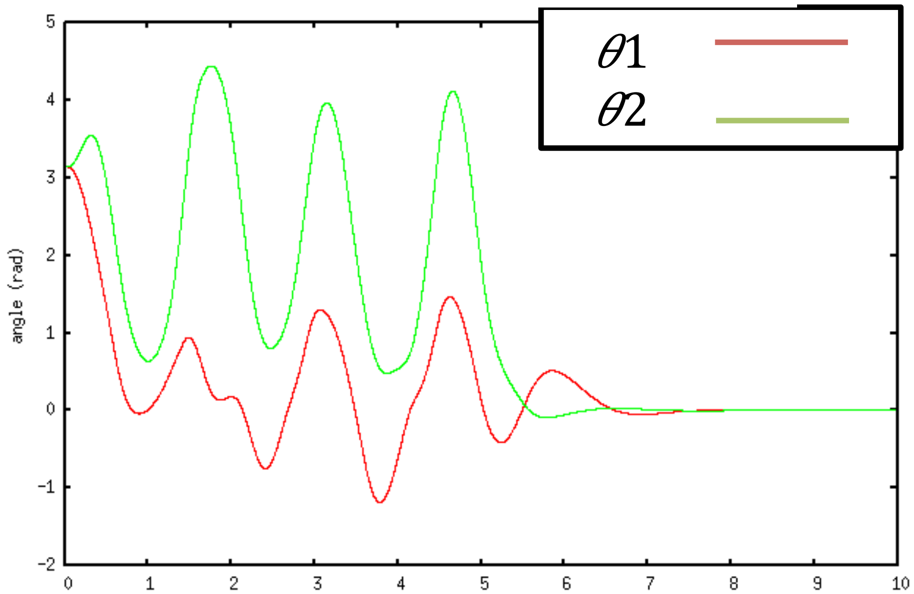
- Background
- Our Target
- Speculative MPC Execution w/ Input Value Prediction
- Preliminary Evaluation
- Conclusions

Experimental Setup

- How many cores are required to achieve 100% accurate input-value prediction?
- Arm-typed Pendulum Swing-up Control
  - Total Simulation Time: 10[s]
  - Prediction Horizon: 0.5[s]
  - Sampling Time : 0.001[s]

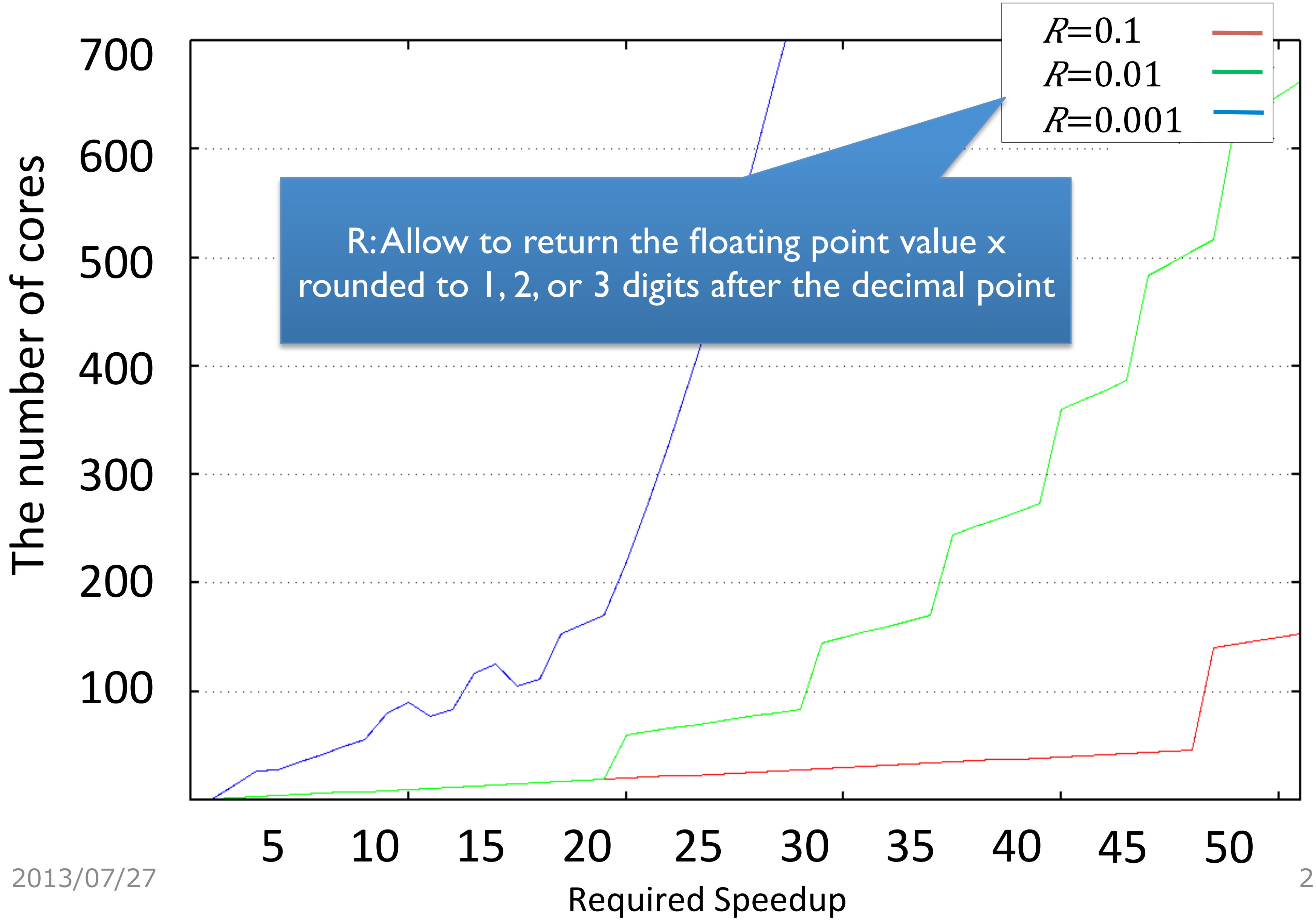


Simulation Result

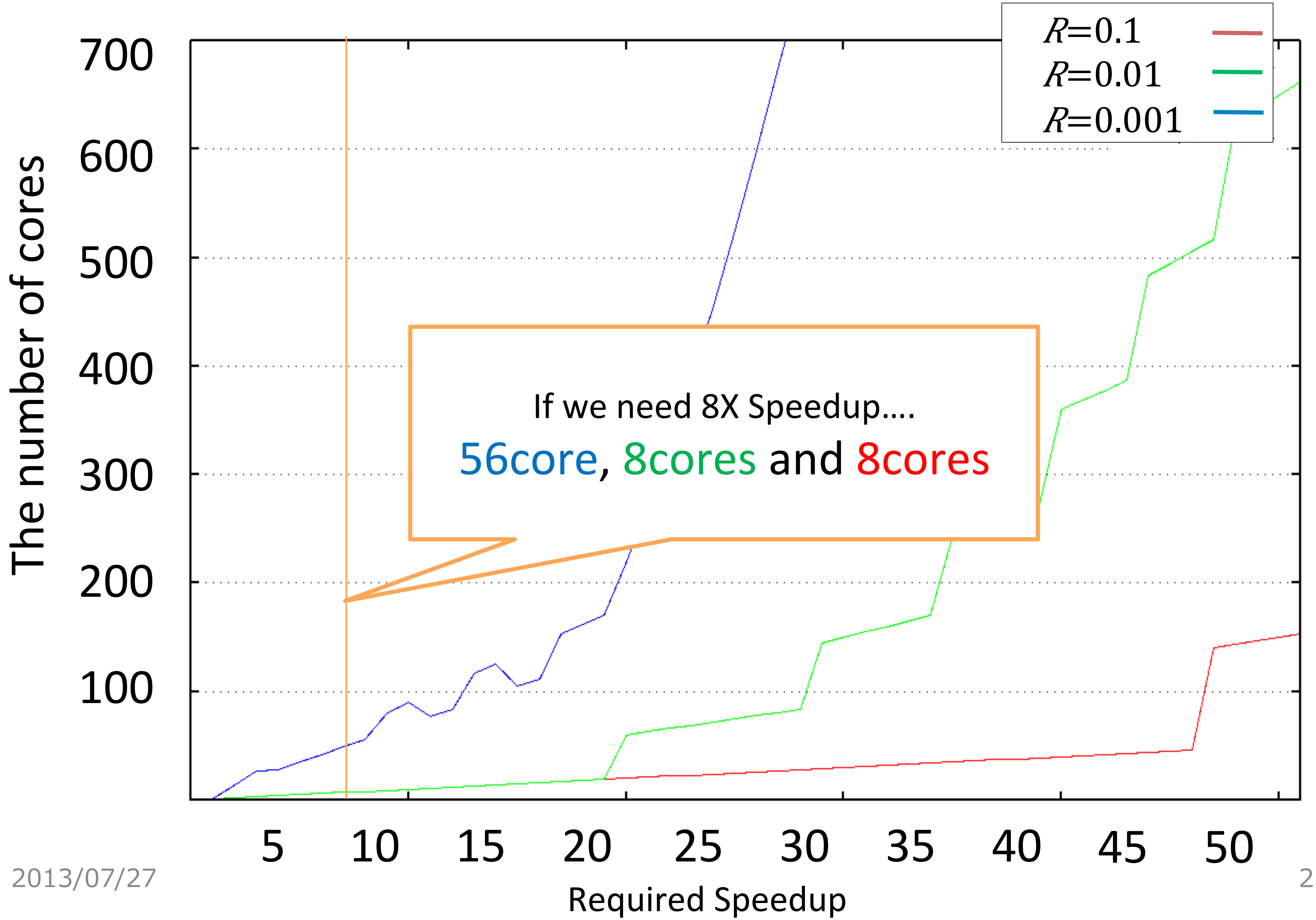


Time

Correlation of the number of cores and Speedup ～A Case Study: Arm-typed Pendulum



Correlation of the number of cores and Speedup ～A Case Study: Arm-typed Pendulum～



Outline

- Background
- Our Target
- Speculative MPC Execution w/ Input Value Prediction
- Preliminary Evaluation
- Conclusions

Conclusions

- Summary
  - Manycore acceleration for RT MPC applications
  - Speculative execution w/ input-value prediction

- Ongoing Work
  - Performance modeling to clarify the potential of proposed approach
  - Consider how to recover from negative impacts of miss-predictions
  - Detailed evaluation on a real manycore chip w/ other applications