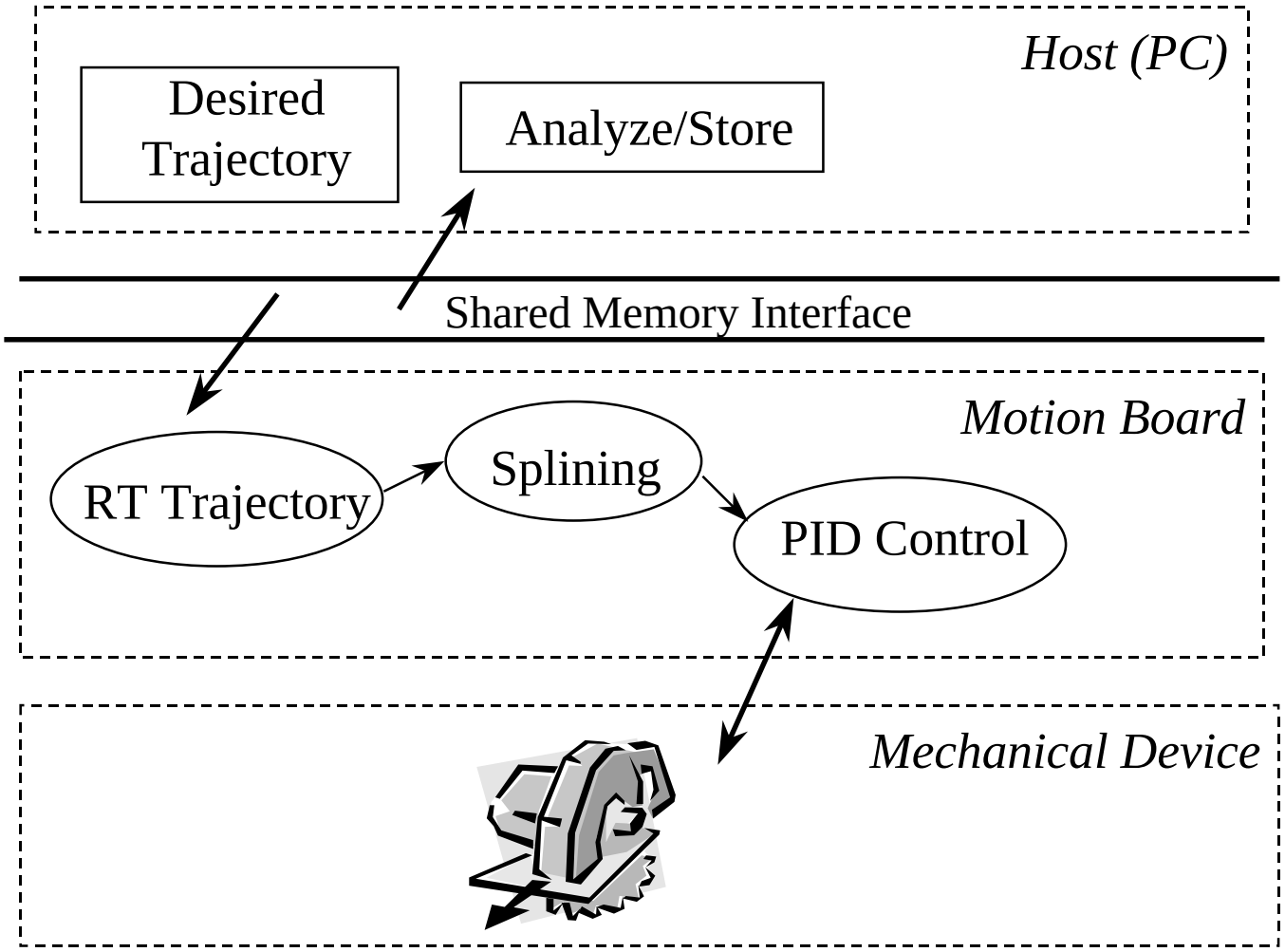
LVRT Based Motion Control

Sadia Malik

Ram Rajagopal

May 2, 2000

Motion Control Technologies



Problem Statement

- Hard Coded Control Loops
- Simulation environment very different from actual environment
- Limited trajectories
- Custom designed solution often needed

Possible Solutions

- Provide common simulation and implementation environment
  - Adjust input parameters dynamically
  - Update control algorithm with system changes
- Auto identify and present the most suitable model for the system
  - Allow user defined algorithms
  - Select optimal control design

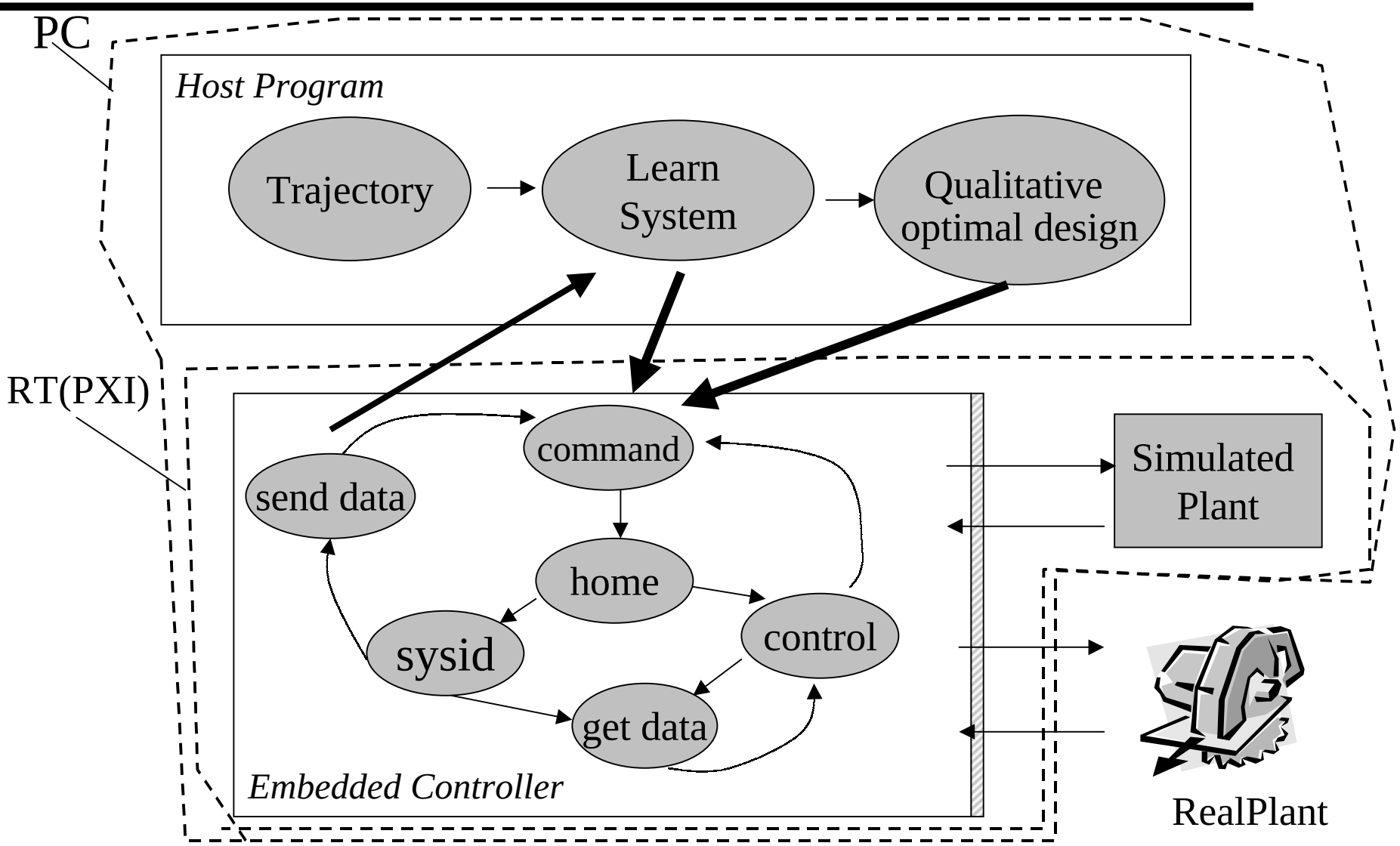
LabVIEW Real Time

- Graphical Programming Environment
  - Virtual Instruments as components
  - Built-in debug environment
- Real time execution environment
- Code portable to any LVRT series hardware

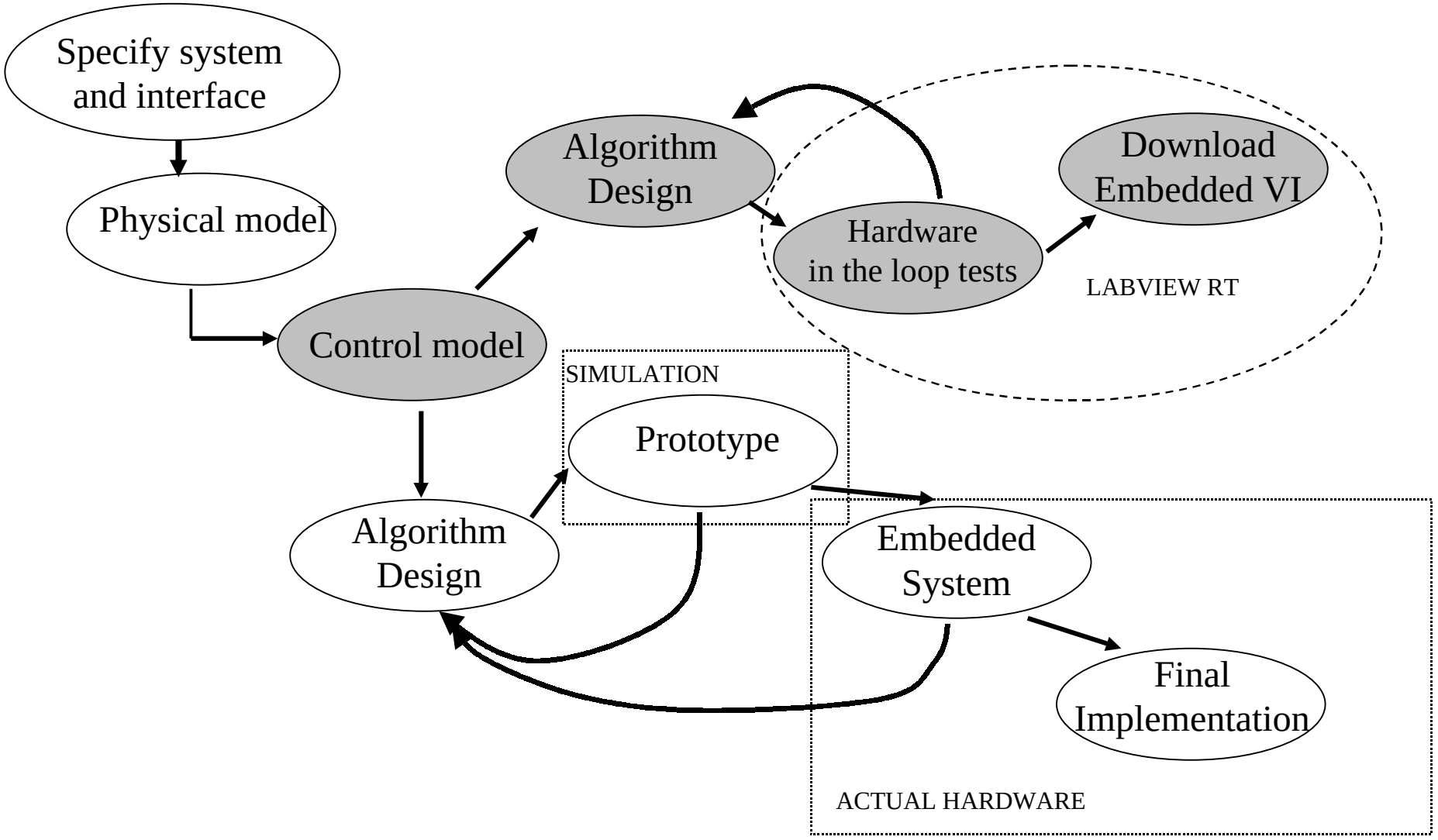
Our Solution

- Used LVRT platform as the development and execution environment
- Identified system and control design prior to generating trajectory
- Adjusted control design according to system needs
- Combined G Dataflow with FSM model

Next Generation Motion



Control Design Cycle



Accomplishments

- Proved that
  - G is strictly bounded in memory
  - G schedule guarantees complete execution
  - Graph determinism can be verified ($O(|actors|^3)$)
- Implemented
  - Proposed structure (PID, MIMO)
  - Transparent Hardware in the Loop simulation
  - Novel LQR based qualitative design strategy
  - Oversampled simulation

Conclusion

- G is a well behaved language that shares several features with SDF, PN, and BDF models
- Current motion control products can be made more flexible and responsive
- RT based design adds flexibility and reduces development time
- Future developments
  - Include on-the-fly host based system identification
  - Incorporate other control and optimal design techniques