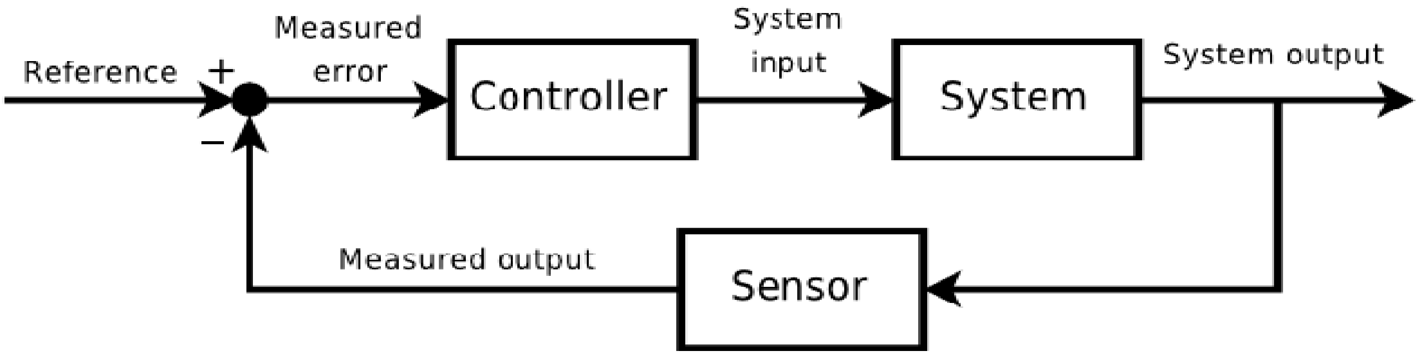
Analysis Methods for Data-driven Feedback Control

Alexey Ershov
Institute for Synchrotron Radiation, Imaging Group

UFO Meeting, Karlsruhe, March 22 2012

Feedback Loop

- **Sensor**: Gets an input from Detector, analyzes it and provides a Measure for Controller
- **Controller**: Based on input from Sensor and control parameters manipulates the System
- **System**: Using commands from Controller adjusts process under study and executes a measuring procedure



- Terminology could be discussed and changed
- Sensor **is not** a Detector
- The general model is **independent** of computational architecture and physical local

Sensor

All Sensor methods could be classified according to:

Response time (processing time) of the Sensor:

- Real-time (e.g. fast reject)
- Interactive (e.g. auto-focusing)
- Offline (e.g. general optical flow computation)

Priority (How often such functionality is required?):

- High Priority: Should be used routinely
- Middle Priority: Could be used sometimes
- Low Priority: Very rare or application is not clear

Measure (output) type:

- Scalar value (e.g. changes rate, maximum velocity)
- Vector (e.g. object tracking, shift adjustment)

Sensor Methods I

- Automated focusing
  - Idea: Evaluate image sharpness to adjust a focus
  - Response: Interactive.
  - Priority: High
  - Applications: All samples
  - Implementation example: Gradient based

- Automated frame-rate adjustment
  - Idea: Analyze displacements magnitudes to adjust acquisition rate.
  - Response: Interactive.
  - Priority: Middle
  - Applications: All samples
  - Implementation example: Normalized-Cross Correlation algorithm

Sensor Methods II

- Fast rejection
  - Idea: Evaluate amount of changes between frames to reject unchanged data
  - Response: Real-time.
  - Priority: Middle
  - Applications: very high frame rate, process cannot be triggered
  - Implementation example: Difference analysis, Histogram analysis

- Automated sample alignment
  - Idea: Perform object recognition, identify sample position to align it
  - Response: Interactive
  - Priority: Middle (or High?)
  - Applications: samples with distinct shapes (e.g. circular)
  - Implementation example : Feature detection, shape recognition

Sensor Methods III

- Automated alignment of axis of rotation
  - Idea: Evaluate projections taken from different rotation angles to adjust AOR
  - Response: Interactive
  - Priority: High
  - Applications: For all tomographic experiments
  - Implementation example: Difference analysis

- Global shift alignment (or Tracking)
  - Idea: Compute global shift to track an object or process
  - Response: Real-time
  - Priority: Low. But could be
  - Applications: follow foam expansion, follow flow front
  - Implementation example: Normalized-Cross Correlation algorithm

Sensor Methods IV

- General Optical flow
  - Idea: Compute correspondences between all the pixels of the subsequent image frames
  - Response: Offline
  - Priority: High
  - Applications: All samples, but not suitable as a Sensor method
  - Implementation example: Variational Optical flow methods

All motion-related Sensor methods could be derived from general OF method.

Could be used to test motion-related methods on realistic image sequences.

Simulated Sensor

Idea: Implement a simulated Sensor for development purposes.

Aspects to consider:

- Simulated output from different types of Sensor algorithms
- Communication between Sensor and Controller
  - Frequency of events (measures from Sensor to Controller)
  - Latency
  - Synchronization
  - Reliability (e.g. Acknowledgments between Sensor and Controller)
- Errors and exceptions handling

For Controller:

- Architecture: centralized, distributed, combined
- Multiple Controlling processes
- Common open-source libraries or frameworks