Time Testing of Cyber-Physical Systems

Aviral Shrivastava

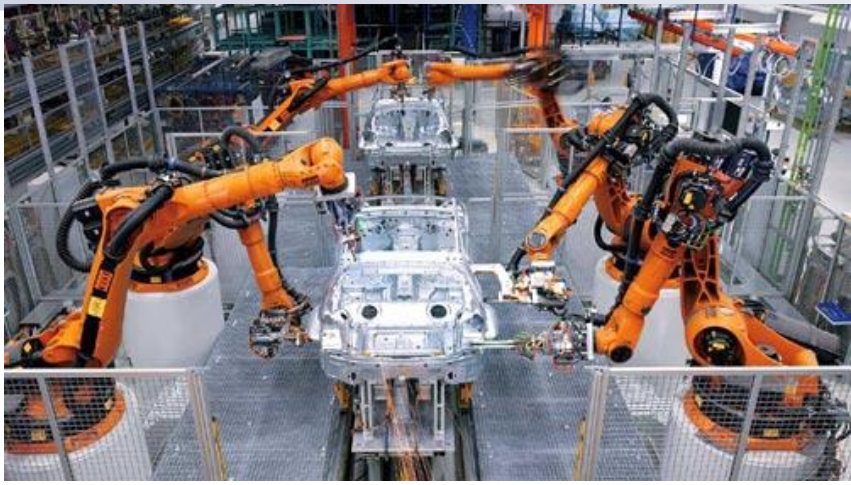
Compiler Microarchitecture Lab

Arizona State University

CML

Cyber-Physical Systems (CPS)

- Control is in Software
  - Arbitrarily complex control systems
  - smart cities, smart buildings, smart defense, etc.
- Cyber-Physical Systems
  - *networked* embedded systems
  - sensor networks with *actuation*



- *Hard-real-time* CPS
  - correctness depends on functionality as well as correct timing
  - autonomous cars
- *Safety-critical* CPS
  - failure of timing can lead to a catastrophe

Achieving right timing in CPS is hard

- Modern computing systems are designed to improve performance at the cost of timing predictability
  - Architecture: Cache, Branch prediction
    - Improves performance, but the latency of instructions becomes unpredictable
  - Operating system: Unbounded pre-emption
    - Time it takes to serve an interrupt is unbounded

Timing Accuracy:
Cycle: ns
Instruction: 100 of ns
Program: > ms

Achieving right timing is hard!!

- Programming languages have no sense of time
  - Cannot specify, "execute this store operation at 4:00 pm tomorrow."
  - Cannot specify, "execute this loop every 100 ms."
- We use OS facilities to get some handle on time
  - getTimeOfTheDay()
  - delay()
- Programming language cannot guarantee any timing
  - unlike functionality – programming language guarantees the functionality irrespective of OS, other tasks on the system, or even hardware – makes guaranteeing functionality easier.

Timing must be tested!!

Testing the timing of single-node CPS is hard

- Heterogeneity of CPS components
- Temporal behavior of signals makes observation difficult
  - Digital signals: rise time can change the time of event
  - Analog signals: a system with high frequency components can effect the threshold detection of a signal
  - Noise, cross talk , etc. have effect on the event detection time
- The existing method and equipment are sophisticated
  - Oscilloscopes and digital/signal/frequency analyzers

The time testing methods are customized, and can be hard to reason about!

Testing the timing of distributed CPS is even harder!

- When a CPS is geographically distributed how we can assure that measurements are taken at the same time?
  - Clock drift
  - Synchronization accuracy
  - Synchronization frequency

- How do we combine and make sense of data measured with different monitoring equipment, each with it's own clock, precision, and latency properties.

What we have been doing?

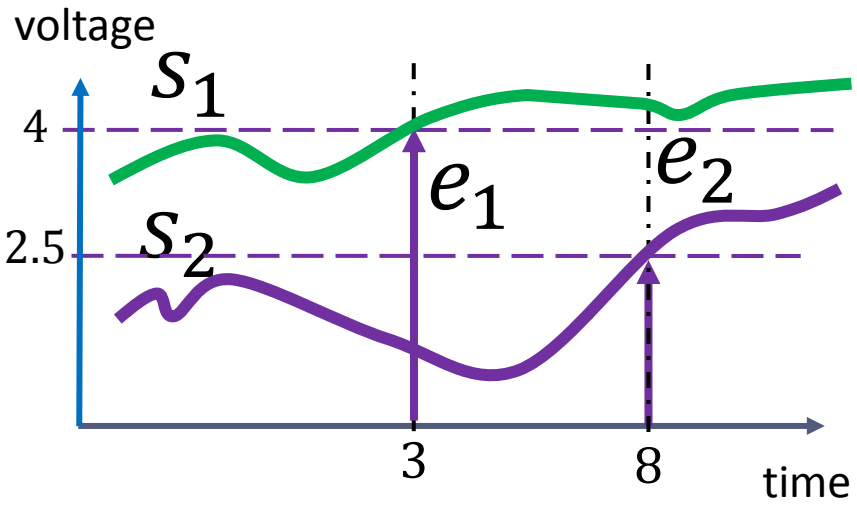
- Define a language to express the timing constraints on distributed CPS
  - Traditional constraint specification languages like CTL, MTL, LTL do not work
    - since they are for state-based systems and digital signals
    - *but* CPS may have continuous systems and analog signals
  - STL does not work
    - since STL is level triggered, so you can specify
      - Globally if signal A is greater than 5V, then eventually signal B will be less than 1 volt.
    - but, timing constraints are most often between edges/events, so need level-triggered logic.
      - Globally, whenever signal A rises above 4V, then within 5 seconds, signal B also rises above 4V.

1. Develop a way to express the timing requirements of CPS

2. Design a testbed to validate the timing requirements of CPS

Timing Constraints in CPS

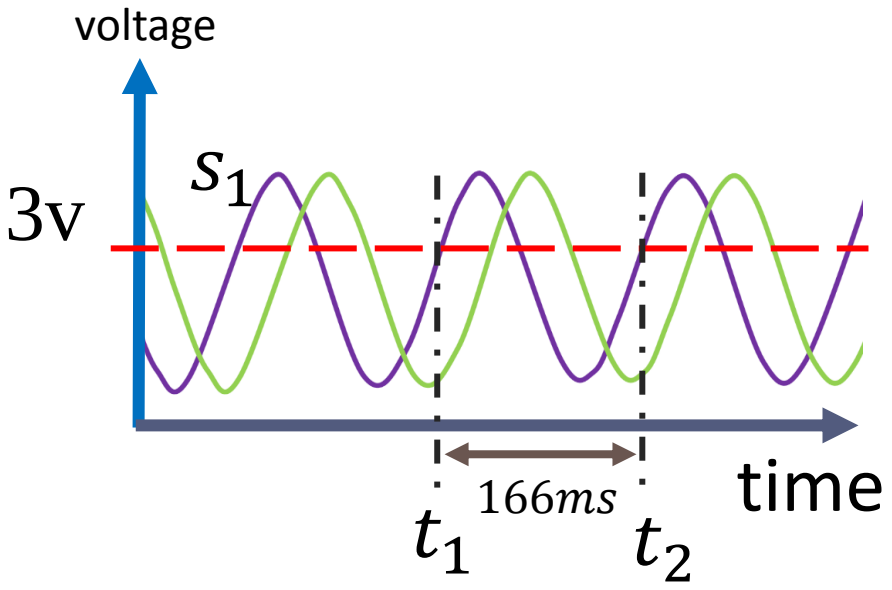
- Different types of timing constraints
  - Latency between two events.
    - The time interval of two events is the concern
    - $\Delta t = 8 - 3 = 5$
    - LCE$(e_1, e_2, 5)$
  - Simultaneity of several events
    - All signals cross the threshold at the same time
  - The events are chronological
    - The order of events is the matter



The latency between s_1 when it crosses 4v and s_2 goes above 2.5v is 5s

Timing Constraints in CPS (repetitive)

- Frequency of events is a certain amount
  - The frequencies of an events is measured
  - FC(e_1,60Hz,0.06Hz)
  - Frequency=$\frac{1}{t_2-t_1} \pm 0.06s$
  - $16.65ms \leq \frac{1}{t_2-t_1} \leq 16.68ms$
- Phase of two signals with the same frequency is the a certain value
  - The time difference between crossing the threshold in two signals in each period
- Two events are sporadic with a minimum time interval
  - When a signal crosses its threshold, it should not be crossed again for a minimum time
- A burst of event is shown up
  - The event should happen for 'd' times then rest for a minimum time 'm'



The latecy between two consecutive events on s_1 is between 16.65ms and 16.68ms

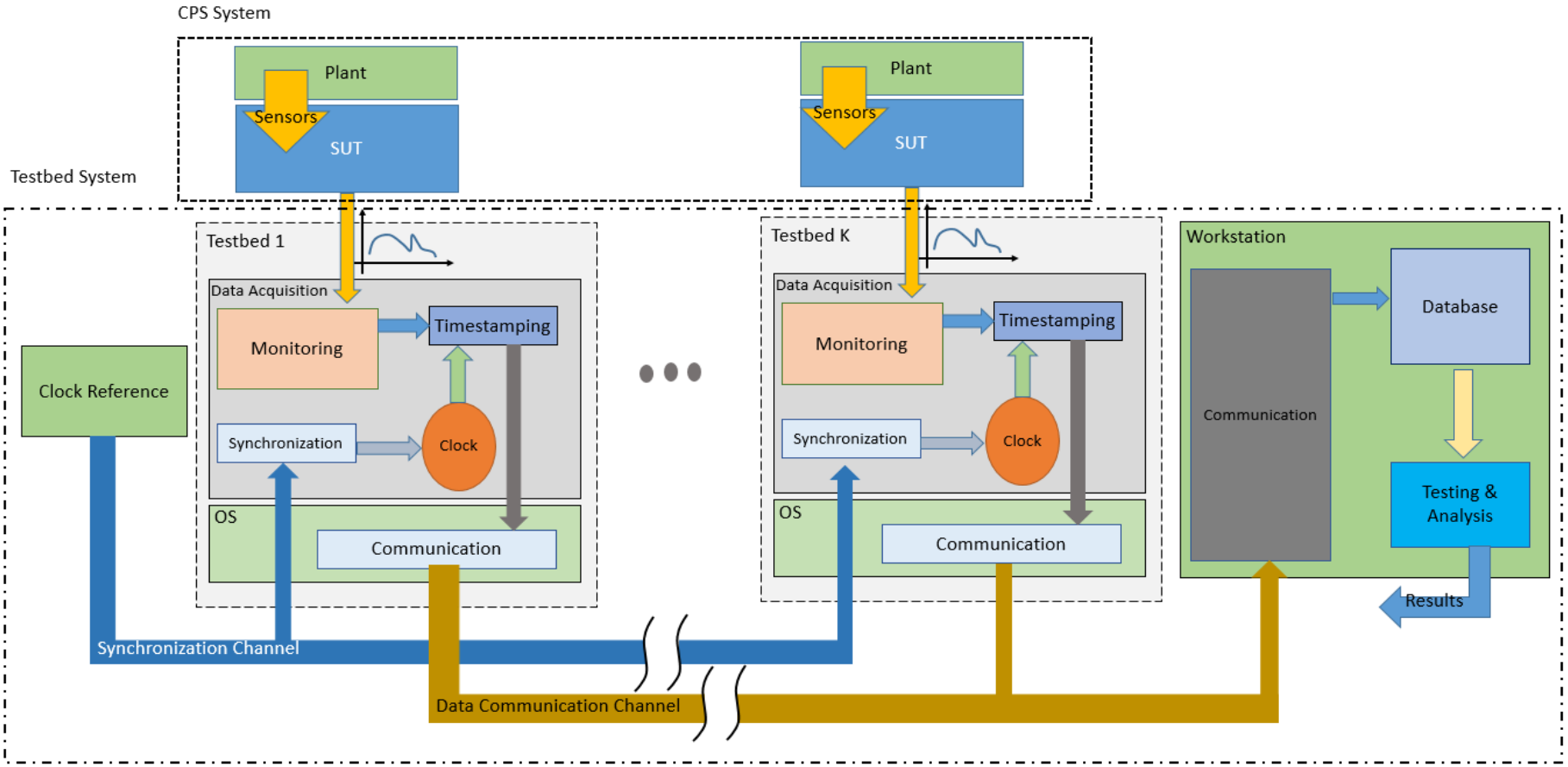
What we have been doing?

- Define a language to express the timing constraints on distributed CPS
  - Traditional constraint specification languages like CTL, MTL, LTL do not work
    - since they are for state-based systems and digital signals
    - *but* CPS may have continuous systems and analog signals
  - STL does not work
    - since STL is level triggered, so you can specify
      - Globally if signal A is greater than 5V, then eventually signal B will be less than 1 volt.
    - but, timing constraints are most often between edges/events, so need level-triggered logic.
      - Globally, whenever signal A rises above 4V, then within 5 seconds, signal B also rises above 4V.

1. Develop a way to express the timing requirements of CPS

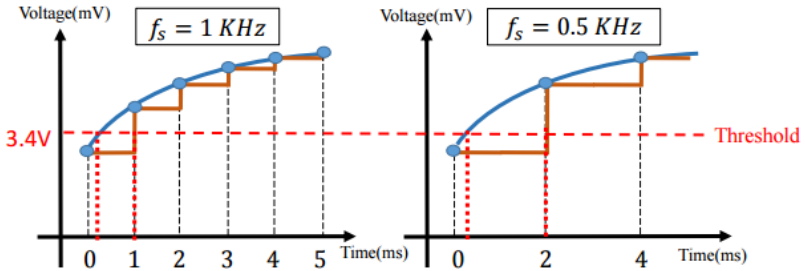
2. Design a testbed to validate the timing requirements of CPS

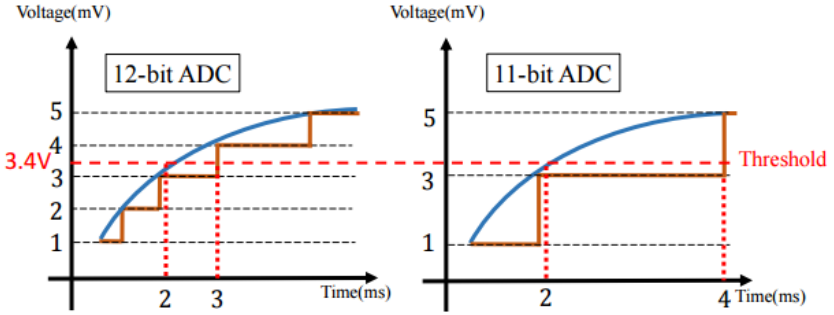
Testbed Structure



Testbed Specs affect measurement

- Data acquisition sampling rate
  - Signal sampling granularity (e.g. 20KS/s)
- ADC resolution
  - Signal amplitude granularity (e.g. 12-bit)





- Clock accuracy
  - Internal clock drift (e.g. 40 ppm)
  - synchronization accuracy (e.g., 1 ms)
  - Synchronization frequency (every 1 second)
- Internal impedance
  - to avoid loading effect (e.g. 10 MΩ)

Testbed Implementation

- Two cRIOs (NI-9067 and NI-9035)
- Synchronized with IEEE 1588 (PTP)
- Digital/Analog input
- On board FPGA

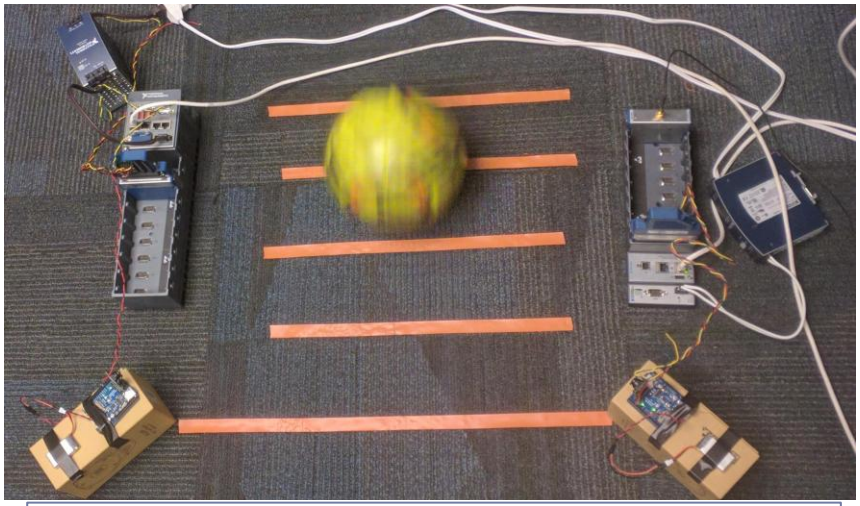
Testing Setup Specifications

- ADC: 12-bit

- Synchronize every second,
- PTP synchronization,

Case Study 1: Simultaneous Image Capture

- Images from cameras are reconstructed to create a 3-D image
  - May not be able to reconstruct if cameras click at different times, and there is a fast moving object, e.g., soccer ball
  - Maximum delay between the time of the clicks = 100µs.
- The ArduCAM ESP8266 UNO boards
  - includes a 2MP CMOS camera.
  - built-in ESP8266 Module for wireless communication
- An HTTP web-server is used to send the capture command to both cameras.
- Upon capturing, each ArduCAM board generates a trigger signal on one of the digital I/O pins.



Timing Requirement:
Need to capture image of an object within 20 ms.
(S(, ,20ms)

$\varepsilon wcco$ = 5µs + 100ns = 5.1µs.
$\varepsilon wcco + \varepsilon ADC$ < 100µs.
Output impedance = 470Ω << input impedance of cRIO = 1$M\Omega$

Can validate timing constraint

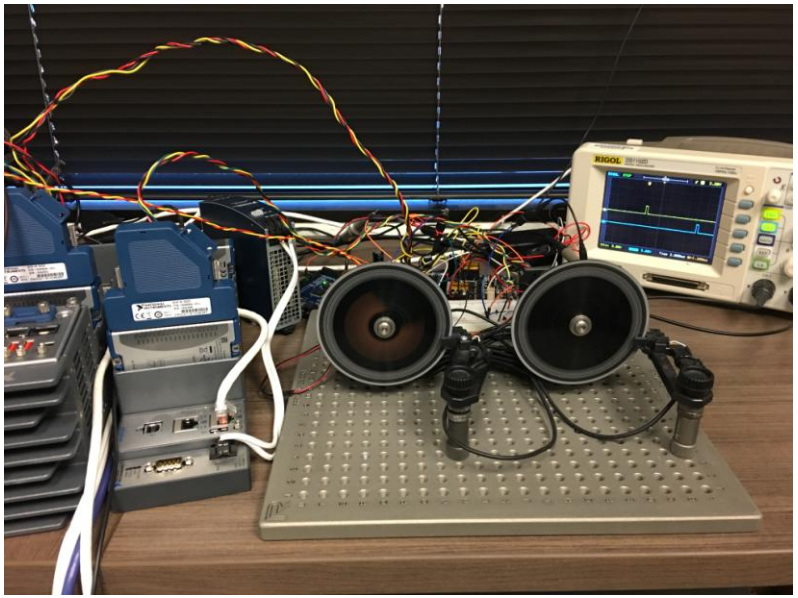
Case Study 2: Generator Synchronization

- 2 motors controlled by different controllers, connected to each other via internet
- All generators must operate at 60Hz±0.1%
- Phase cannot be more than 10° different

- two DC motors to
- represent two small generators.

- Master motor sends its rotation frequency and its phase to the otherArduino boards are synchronized with each other using two wireless modules (NRF24L01+, 2.4GHz).

- Power grid case study, required accuracy is 33μs for frequency constraint and 463μs for phase constraint.



Timing

or
Same phase ➔

meets the timing requirements. Can validate!