CSE 520S
Real-Time Systems

Prof. Chenyang Lu
TAs: Ruixuan Dai, Jiangnan Liu
Real-Time Systems

- Systems operating under timing constraints

- Safety-critical systems
  - Automobiles.
  - Airplanes.
  - Mars rovers.
  - Factory automation.
  - Air traffic control.

- Time-sensitive systems
  - Game console, Google Stadia.
  - Stock trading.

- >95% of microprocessors are used for embedded systems.
Embedding a Computer

CPU

output

input

mem

actuators

sensors

embedded computer
Anti-lock Brake System

- Pumps brakes to reduce skidding: real-time $\rightarrow$ safety
**Highway Autopilot**

**General Motors Super Cruise: John Capp, Jeremy Salingar, Eric Raphael, and team**

The self-driving car started as a science-fiction fantasy—then DARPA and Google turned it into a real-world experiment. Now GM is working to make automotive autonomy part of an option package. The automaker has combined two existing technologies—adaptive cruise control and lane centering—into its Super Cruise system, which allows for hands-free driving at highway speeds. It could be available in a production vehicle as soon as 2018.

**How It Works**

**Collision Avoidance**

A long-distance radar system detects vehicles more than 300 feet ahead. The vehicle will automatically accelerate or apply the brakes to maintain a preset following distance.

**Lane Centering**

Using a combination of GPS and infrared optical cameras, the Super Cruise system watches the road ahead and adjusts steering to keep the car in the middle of its lane.
A Distributed Real-Time System
More on a Car

~100 microprocessors:
- 4-bit microcontroller checks seat belt;
- microcontrollers run dashboard devices;
- 16/32-bit microprocessor controls engine;
- In-Vehicle Infotainment (IVI): audio/video, navigation, communication…
Real-Time Applications in a Car

- Soft real-time: Infotainment on Linux or Android
- Hard real-time: Safety-critical control on AUTOSAR

WU/Purdue Real-Time Hybrid Simulation

- Enabled by real-time parallel computing
- Expand to larger-scale, multi-specimen experiments (bridge spanning a river, different ground motions on each end)
- Towards cloud-based multi-site experiments
Internet of Things

Convergence of

- **Miniaturized devices**: processor+sensors+radio, embedded OS.
- **Low-power wireless**: connect millions of devices to the Internet.
- **Data analytics**: make sense of sensor data.
- **Cloud and edge computing**: scalable real-time data processing.

Large-scale IoT-driven control

- **Smart manufacturing, transportation, power grid, healthcare…**
- **Closed-loop control requires real-time performance!**
Clinical Warning

IoT-driven Control

- Smart manufacturing, transportation, grid, healthcare...
- Closed-loop control $\rightarrow$ latency bounds
Real-Time IoT

End-to-End Real-Time Performance

- Miniaturized devices $\rightarrow$ real-time embedded systems
- Low-power wireless $\rightarrow$ real-time wireless
- Data analytics $\rightarrow$ real-time analytics
- Cloud $\rightarrow$ real-time data processing
Real-Time Cloud

- IoT → large-scale sensing and **control** of physical world
  - Smart manufacturing, smart transportation, smart grid…
  - Feedback control demands real-time performance guarantees.

- Example: Intelligent Transportation
  - Cloud collects data from cameras and roadside detectors.
  - Control the traffic signals and message signs in real-time.
  - Transportation information feed to drivers.
  - SCATS @ Sydney: controlling 3,400 signals at 1s round-trip latency.

- **Cloud needs to be** real-time and predictable!
  - URL: [https://youtu.be/CluvnRaVhqA](https://youtu.be/CluvnRaVhqA)
Real-Time Cloud

- Support real-time applications in the cloud.
  - Latency guarantees for tasks running in virtual machines (VMs).
  - Real-time performance isolation between VMs.
  - Resource sharing between real-time and non-real-time VMs.

- Real-time cloud stack.
  - RT-Xen -> real-time VM scheduling
  - VATC -> real-time network I/O on a virtualized host.
  - RT-OpenStack -> real-time cloud resource management.
Example: RT-Xen

- Real-time schedulers in the Xen hypervisor.
- Provide real-time guarantees to tasks in VMs.
- Incorporated in **Xen 4.5** as the rtds scheduler.

[https://sites.google.com/site/realtimexen/](https://sites.google.com/site/realtimexen/)

Challenges

Must meet non-functional constraints

- Real-time
- Memory
- Battery lifetime
- Reliability, safety and certification
- Cost

Correct output is NOT enough!
Real-time Requirements

- **Period**: release a job every $T$ sec
  - Playback 30 video frames per second

- **Deadline**: complete a job within $D$ sec
  - Anti-lock brake must start within 10 ms after skidding starts
Hard vs. Soft Real-Time

- **Hard:** violating timing constraints $\rightarrow$ failure
  - Automobile: active safety features, autonomous driving
  - Air traffic control

- **Soft:** violating timing constraints $\rightarrow$ inconvenience
  - Video
  - Audio ("harder" than video)
  - Stock trading
Topics

1. Real-Time Operating Systems
2. Real-Time Scheduling
3. Real-Time Edge Computing
4. Real-Time Parallel Computing
5. Real-Time Virtualization and Cloud Computing
6. Real-Time End-to-End Scheduling
7. Adaptive Quality of Service Control
8. Industrial Wireless Control
9. Project: Cloud Middleware for IoT
   - Based on Amazon Web Services (AWS)
Grading

- Projects 60%
  - Cloud warm-up homework: 1%
  - Proposal and presentation: 10%
  - Demo 1: 5%
  - Demo 2: 5%
  - Final demo & report: 39%

- Critiques 35%

- Participation 5%
Critiques

- 1/2 page critiques of research papers
- Submit by 10am before class
- Back-of-envelop comments - NOT whole essays
- See guidelines on class web site
  - [http://www.cs.wustl.edu/~7Elu/cse521s/critique.html](http://www.cs.wustl.edu/~7Elu/cse521s/critique.html)
Project

- Three students per team

- Build IoT systems based on cloud
  - Front end: smart watch, wristband, Raspberry Pi
  - Cloud backend: storage, analytics, Alexa, notification
  - Write a paper
  - Demo to the class
Smartwatch as a Healthcare Tool

- **Open, programmable platform**
  - Wear OS, Research Kit, onboard analytics

- **Two-way communication**
  - Ecological momentary assessments

- **Continuous, passive measurements**
  - Activity, heart rate, sleep, location…
Raspberry Pi

https://cse.wustl.edu/Pages/default.aspx
Amazon Web Services (AWS) IoT

United: Connect + Communication

Smart: Other Cloud Service
Data Storage
Machine Learning

Source: https://aws.amazon.com/iot-platform/
Timed Up and Go with Smartwatch

- **Watch app**
  - Remind participants to take the assessment
  - Automatically upload the data to the cloud for analysis
  - Analyze gait and motion features
  - Feedback to physicians and participants

- Assess physical health and fall risk during prehabilitation.
  - 20 participants undergoing neoadjuvant radiotherapy followed by surgery
  - Patients will complete TUG at home with the smartwatch for 90 days.

[https://www.cse.wustl.edu/~lu/TUG.mp4](https://www.cse.wustl.edu/~lu/TUG.mp4)

Joint work with Matthew Spraker (Radiation Oncology), Ruixuan Dai (CSE)
Voice-based Smart Medicine Dispenser

https://www.cse.wustl.edu/~lu/cse521s/Videos/medicine_dispenser.mp4
Steps

1. Choose your favorite topic
2. Form a team
3. Propose a plan
4. Implement
5. Measure and analyze
6. Demo: 1, 2, final
7. Write a technical report
Start Early and Work Often!

- Choose topics
- Put together a team
- Meet every week to coordinate
- Lots of development and experiments throughout the semester!
Pointers

- [http://www.cse.wustl.edu/~lu/cse520s/](http://www.cse.wustl.edu/~lu/cse520s/)

- Email for appointment
  - Chenyang (Jolley 213)
  - Ruixuan Dai (Jolley 219A): Projects
  - Jiangnan Liu (Jolley 219A): Critiques