Final Demo

- Final demo (in person)
  - 4/28, 1pm-3:15pm
  - Busch Hall 100

- 10 min per team
  - 9-min demo + 1-min Q&A
  - All should attend the entire session. It’ll be fun!

- Check out the new classroom and test your demo
  - 4/26, 1pm-2:15pm
  - Busch Hall 100

- Submit before class: slides, backup video if needed
Towards Dependable Internet of Things

Chenyang Lu

Cyber-Physical Systems Laboratory
IoT we can depend on

- From best-effort consumer applications to critical domains
  - Industry 4.0
  - Smart healthcare

- Must achieve new forms of **dependability**
Dependability Challenges

- Resilient control of physical systems
- Robust predictions with noisy data
- Real-time and reliable edge platforms

Clinical decisions and intervention
Dependability Challenges

- Resilient control of physical systems
- Robust predictions with noisy data
- Real-time and reliable edge platforms
Resilient Control of Physical Systems

- Enabling technologies for Industry 4.0
  - Wireless networks: flexibility and easy deployment
  - Edge computing: on-premise computing resources

- Challenges
  - Exploit edge computing to enhance **control performance**
  - Maintain **stability** despite unreliable wireless networks
  - **Resiliency** under wireless and physical disturbances
Local Control vs. Edge Control

- Wired connection → Stability guarantee
- Wireless network → Varying data loss
- Computation capacity → Sophisticated control
Tradeoff between Local and Edge Control

Robotic Joint Position Control

- Edge may improve control performance
- Edge may suffer from data loss
- Choice between local and edge depends on
  - network reliability
  - physical plant states

Metric: mean absolute error (MAE)
Switching Multi-tier Control (SMC)

- **Objectives**
  - Optimize control performance
  - Guarantee stability

- **SMC: Switch between local and edge controllers**
  - **Optimal Platform Classifier**: select local or edge controller for performance
  - **Stability Switch**: switch to local controller to guarantee stability

- **Based on cyber-physical states**
  - network reliability
  - physical states

---

**Switching Logic**

**Stability Switch** guarantees stability

Sha, L., Using simplicity to control complexity. IEEE Software, 2001]

$x \notin PR$: switch to local controller to guarantee stability

$x \in PR$: OPC selects the optimal controller based on network conditions and physical states

**Optimal Platform Classifier** selects the optimal control platform
Optimal Platform Classifier

- Theoretical analyses of wireless control performance is challenging
- **Learning**-based Optimal Platform Classifier

- Overcome restrictions of analytical modeling
- Applicable to a wide range of control and network technologies
Training the Optimal Platform Classifier

- Physical plant: PUMA 560
- Local: state feedback controller
- Edge: model predictive controller
- Wireless network: two-state Markov chain loss model

The Gilbert-Elliott loss link model

Optimal Platform Classifier

- $x_e$ (state error)
- $\alpha$ over $T_c$
- $\beta$ over $T_c$
Each data point represents a simulation run.
- Label each data point with the optimal controller.
- 26,000 simulations, 40 GB data
- When $x_e$ and $\beta$ are low, and $\alpha$ is high, edge control has smaller MAE
- Train a model to classify optimal controller
Optimal Platform Classifier

- SVM model learns the non-linear boundary between the controllers
- When $x_e$ and $\beta$ are low, and $\alpha$ is high, OPC chooses edge control
- Testing accuracy: 90.98%
- Cost of misclassification is low
Case Study: Robotic Joint Position Control

- SMC dynamically optimizes control performance with stability guarantees.
- SMC outperforms local and edge control under changing network reliability.
- **Holistic control** based on physical and network states → **resiliency**!
Dependability Challenges

- **Resilient control** of physical systems
- **Robust predictions with noisy data**
- **Real-time and reliable** edge platforms
Internet of Medical Things

- IoMT is poised to transform healthcare
  - Wearables: continuous and unobtrusive monitoring
  - Connectivity: real-time monitoring and intervention
  - Cloud: scalable storage and processing
  - Analytics: predict outcomes and support decisions

- Wearable data: fine-grained, noisy, lossy
  - Sampling period: 1 min (step, heart rate); 1 day (sleep)
  - Small patient cohorts → avoid deep models

- How to extract robust and predictive features from wearable data?
Predict Clinical Outcomes with Fitbit

- **Predict clinical outcomes**
  - Surgical complications of patients undergoing pancreas surgery
  - Readmissions of congestive heart failure patients
  - Recovery after spine surgery
  - Dynamic cognitive function in youth with type-1 diabetes
  - Depression of older adults undergoing digital behavioral therapy

- **Robust pipelines for data collection and feature engineering for Fitbit**

Data Collection

Feature Engineering

- *Predict Clinical Outcomes with Fitbit*

- *Robust pipelines for data collection and feature engineering for Fitbit*
Two-Level Feature Extraction

### Daily Features

- **Statistical features**
  - 1\textsuperscript{st} order: mean, max, min, skewness, kurtosis
  - 2\textsuperscript{nd} order: energy, entropy, correlation, inertia and local homogeneity

- **Detrended Fluctuation Analysis**
  - Determine statistical self-affinity of time series
  - The fluctuation is then used as feature

- **Sedentary behavior**

### High-Level Features

- Robust singular spectrum analysis of noisy and incomplete daily features

![Trend Extraction with Missing Components](chart.png)

**Trend Extraction from Incomplete DFA-Sleep-20 Daily Feature**
Predictive Performance of Surgical Complications

**Machine learning** models outperform traditional surgical risk scores.

(ACS NSQIP: American College of Surgeons National Surgical Quality Improvement Program)

<table>
<thead>
<tr>
<th>Data Source</th>
<th>AUROC</th>
<th>AUPRC</th>
<th>Sensitivity</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random weighted classifier</td>
<td>0.5097</td>
<td>0.4322</td>
<td>0.1520</td>
<td>0.8583</td>
</tr>
<tr>
<td>NSQIP with Clinical Characteristics</td>
<td>0.6114</td>
<td>0.4075</td>
<td>0.2800</td>
<td>0.8571</td>
</tr>
<tr>
<td>ML with Clinical Characteristics</td>
<td>0.7632</td>
<td>0.7374</td>
<td>0.5800</td>
<td>0.8583</td>
</tr>
<tr>
<td>Wearable Data</td>
<td>0.7326</td>
<td>0.7192</td>
<td>0.5480</td>
<td>0.8583</td>
</tr>
<tr>
<td>Clinical Characteristics + Wearable Data</td>
<td>0.8802</td>
<td>0.8871</td>
<td>0.8320</td>
<td>0.8583</td>
</tr>
</tbody>
</table>

Prospective clinical study with 61 patients undergoing pancreatic surgery
Wearable features complement clinical data to enhance predictive performance.

<table>
<thead>
<tr>
<th>Data Source</th>
<th>AUROC</th>
<th>AUPRC</th>
<th>Sensitivity</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random weighted classifier</td>
<td>0.5097 (0.0585)</td>
<td>0.4322 (0.0469)</td>
<td>0.1520 (0.0854)</td>
<td>0.8583 (0.0504)</td>
</tr>
<tr>
<td>NSQIP with Clinical Characteristics</td>
<td>0.6114 (0.0000)</td>
<td>0.4075 (0.0000)</td>
<td>0.2800 (0.0000)</td>
<td>0.8571 (0.0000)</td>
</tr>
<tr>
<td>ML with Clinical Characteristics</td>
<td>0.7632 (0.0085)</td>
<td>0.7374 (0.0206)</td>
<td>0.5800 (0.0699)</td>
<td>0.8583 (0.0083)</td>
</tr>
<tr>
<td>Wearable Data</td>
<td>0.7326 (0.0074)</td>
<td>0.7192 (0.0154)</td>
<td>0.5480 (0.0440)</td>
<td>0.8583 (0.0083)</td>
</tr>
<tr>
<td>Clinical Characteristics + Wearable Data</td>
<td>0.8802 (0.0050)</td>
<td>0.8871 (0.0087)</td>
<td>0.8320 (0.0160)</td>
<td>0.8583 (0.0083)</td>
</tr>
</tbody>
</table>

Prospective clinical study with 61 patients undergoing pancreatic surgery
Dependability Challenges

- Resilient control of physical systems
- Robust predictions with noisy data
- Real-time and reliable edge platforms
Coordination Service for Distributed Applications

- Distributed applications need **coordination**
  - Apache Kafka, Flink, Spark, Hadoop…
  - Peers exchange, share and persist states

- Fault-tolerant coordination service
  - Maintain global-accessible states for distributed applications
  - Replicate data store for fault tolerance
  - Synchronize data across data stores
ZooKeeper: a Prevailing Coordination Service

- **Tolerate crash failures** of data stores through replication.
- **Data consistency**: keep data consistent across replicas.
- **Leader-Followers**: One server works as the leader.
- **Leader Election**: ZooKeeper Ensemble will elect a new leader if the old one fails.

**failure recovery ➔ service latency**
Real-time Coordination on Edge Clouds

- **Edge** clouds for *time-sensitive* embedded applications
  - Challenge: recover from failures in a timely manner
  - Opportunity: exploit localized networked platforms

- **RT-ZooKeeper**: novel protocols for fast failure recovery on edge platforms
  - Fast Converging Election: fast leader selection on asynchronous networks
  - Quorum Channel Reuse: reduce delays for establishing communication channels
  - Distributed Epoch Persistence: persist epoch value through replication over the network

Leader Election on a Synchronous Network

- OptFloodMax: elect the node with the largest vote

**How about an asynchronous network?**

- Synchronous: Nodes communicate and compute in synchronous rounds.
- Asynchronous: Messages arrive at arbitrary time; tasks runs at its own pace.
ZooKeeper: Adapt to Asynchronous Network

Epoch: 1
Phase: 1

Phase-1: Election
ZkSrv 2
V = 4
ZkSrv 1
V = 5
ZkSrv 0

Phase-2: Recovery

Phase-3: Service

"I am still following my old leader."

W
W
W

Change Vote
l = 2,
V = 5

"This notification is obsolete, just ignore it."

Waiting an extra round for convergence → delay!
RT-Zookeeper: Fast-Converging Election

Resend the notification

Epoch:  
1 = 1  
Phase:  
Phase-3: Service  
1 = 2  
Phase-1: Election  
Phase-2: Recovery

ZkSrv 2  
V = 4  

ZkSrv 1  
V = 5  

ZkSrv 0

Change Vote  
l = 2,  
V = 5

A single round of wait before convergence!
Micro-Benchmark: Shorten the Recovery Latency

![Diagram showing ZooKeeper Ensemble and recovery latency reduction]
RT-ZooKeeper: Real-Time Coordination on Edge

- **Root causes** of failure recovery delays in ZooKeeper
- **New protocols** for fast failure recovery on edge platforms
- **Microbenchmarks**: 91% reduction in maximum recovery latency
- **Case studies with Kafka**: impact of fast failure recovery on messaging latency

*Importance of revisiting the fundamental designs of cloud services for real-time edge computing*
Towards Dependable CPS

- Resilient control of physical systems
- **Holistic control across local and edge platforms**
- Robust predictions with noisy data
- *Clinical prediction pipeline for wearable data*
- Real-time and reliable edge platforms
- *Redesign of cloud services for edge platforms*