Acknowledgements
Acknowledgements

- I am here because of Jack Stankovic
Acknowledgements

- I am here because of Jack Stankovic

In this community
It started in 1990…

- I got my undergrad degree…
  and started looking for a Masters…
1990

- I got my undergrad degree...
- and started looking for a Masters...
- in *automatic control*
1990

- I got my undergrad degree...
  and started looking for a Masters...
  in *automatic control*
1990

- I got my undergrad degree...
  and started looking for a Masters...
  in *automatic control*
1990

- I got my undergrad degree...
  and started looking for a Masters...
  in *automatic control*
1990

- I got my undergrad degree...
  and started looking for a Masters...
  in *automatic control*  
  Real-time systems
1990

- I got my undergrad degree...
  and started looking for a Masters...
  in *automatic control*
  Real-time systems
I got my undergrad degree... and started looking for a Masters... in *automatic control* — Real-time systems.
1990

- I got my undergrad degree... and started looking for a Masters...
  in *automatic control*
  Real-time systems

Misconceptions in real-time computing (1988)
How to do Research?
Tree of Research
Misconception #1: To advance research, find a niche and extend a branch
Tree of Research

Jack’s practice:
Divert the trunk!
Render the tree obsolete
How to do Research?

- Identify fundamental trends (*those that have a reason to exist... so far*)
- Predict inflexion points (*points where a trend becomes unsustainable/obsolete*)
- Position yourself to capitalize on the aftermath (*before others do*)!!!
Example: Moore’s Law

Exponential growth in the number of transistors on a chip... until thermal limits were reached, heralding the emergence of massively parallel/distributed frameworks.
Example: Moore’s Law

Exponential growth in the number of transistors on a chip... until thermal limits were reached, heralding the emergence of massively parallel/distributed frameworks.
Example: Moore’s Law

Exponential growth in the number of transistors on a chip... until thermal limits were reached, heralding the emergence of massively parallel/distributed frameworks
Misconception #2:
Future CPS Research is Driven by Embedded Tech, Everywhere
Manifestation: The Internet of Things

Source: Texas Instruments

Wearables
- Entertainment
- Fitness
- Smart watch
- Location and tracking

Building & Home Automation
- Access control
- Light & temp control
- Energy optimization
- Predictive maintenance
- Connected appliances

Smart Cities
- Residential E-meters
- Smart street lights
- Pipeline leak detection
- Traffic control
- Surveillance cameras
- Centralized and integrated system control

Smart Manufacturing
- Flow optimization
- Real time inventory
- Asset tracking
- Employee safety
- Predictive maintenance
- Firmware updates

Health Care
- Remote monitoring
- Ambulance telemetry
- Drugs tracking
- Hospital asset tracking
- Access control
- Predictive maintenance

Automotive
- Infotainment
- Wire replacement
- Telemetry
- Predictive maintenance
- C2C and C2I
A Bigger Trend:
Democratized Broadcast
Democratized Broadcast

- Broadcast in the 20\textsuperscript{th} century:
Democratized Broadcast

- Broadcast in the 20th century:
  - Image of a broadcast tower

- Broadcast today:
  - Image of a smartphone taking a photo of a meal
Democratized Broadcast
Fundamental Implications

The Past

O(n)  O(n)

The Present

O(n)  O(n^2) →  O(n^2)
A Deluge of Information!!!
Real-time data grows much faster than our cognitive ability to consume it.

Fast data growth!

Slow Evolution!
Real-time data grows much faster than our cognitive ability to consume it.

Services to Bridge the Gap:

**Autonomous/Cyber-physical Information Processing and Distillation Services**

**Fast data growth!**

**Slow Evolution!**

- Embedded Everywhere
- Democratized Broadcast
Example: Smart City Data-to-Decision Pipelines

Events
- Traffic jams
- Disasters
- Civil unrest

News and Public Sources

People

Sensors

Data

Autonomic Service Stack

Decision Support
Misconception #3

- CPS is about cyber components meeting physical/engineered artifacts
It’s All About People

CPS Challenges are People Centric

- **Medical**
  Insulin pumps, pacemakers, glucose monitors, ...

- **Energy**

- **Disaster Recovery**

- **Sustainability**

- **Transportation**
Today’s Smart Sensor
Today’s Smart Sensor
Interface to the Most Versatile Data Source

The Human “API”
An Analogy: Broadcast Media are “Materials” Propagating “Signals”

Physical target

Response of physical propagation medium (e.g., acoustic, vibration, optical, ...)

Received signature (energy in multiple signal frequency bands)

Physical event

Response of social propagation medium (e.g., tweets)

Received signature (energy in multiple keyword frequency bands)
The Social Signal Layer

Social Sensing Signal, $Signal (k)$

Frequency counts

Keywords

Event Detection and Tracking

Event Data (Tweets Associated with the Event)

Event Map

Events and Trajectories

Social Medium

Observers of Physical Events
Misconception #4

- CPS is embedded systems on steroids...
A Future CPS Skill: Language-agnostic Processing of Unstructured Text (Input to CPS)

- A wealth of data will be generated globally by human sources (social media, blogs, articles, news, print, ...)
- How can a machine ingest such unstructured data and convert it into a structured format amenable to data fusion in CPS applications?
- Text presents a particular challenge: many languages, rich semantics, special dialects, new jargon, grammatical mistakes, sarcasm, ...
Misconception #5

- CPS is based on understanding and exploiting physical world/artifact models
Conter-example:
Deep Learning (Tensor Flow)

Can learn nonlinear functions provided each perceptron has a differentiable nonlinearity

\[
g(t) = \frac{1}{1 + e^{-t}}
\]
The Power of Deep Learning

Enable human-like interaction modalities with physical things

Speech Recognition

Context/Activity Recognition

Vision/Object Detection
Now The Ugly Side...

- Cyber-physical computing (and computing in general) is a victim of its own success!
Rate of Innovation and Development Time Issues

- Near the turn of the 20th century products had a 20-30 year life-span before new “versions” were developed.

- At present, a software product is obsolete in less than 2 years.
  - No time to discover and “debug” all possible problems.
  - New problems introduced in new versions.
  - Component reuse generates additional problems.
Typical Software Engineering Practice

- Abstraction
- Separation of concerns

Abstraction

- Transport
- Network
- Link
- Physical

Separate virtual machines or protection domains

Kernel

Virtualization
Abstraction → Specialization

- Complexity
  - More levels of abstraction
  - Narrower specialization
    - More details are “abstracted away”
    - Myopic view. Less knowledge of possible adverse interactions
      - More potential for interaction or incompatibility errors
Abstraction → Specialization

- Complexity
  - More levels of abstraction
  - Narrower specialization
    - More details are “abstracted away”
    - Myopic view. Less knowledge of possible adverse interactions
  - More potential for interaction or incompatibility errors
Challenge: Building Reliable Systems on Unreliable Components

- The reliability problem will get worse
- The classical research directions in CPS design and analysis that ensure correctness of temporal and logical system behavior will remain an important component of the research roadmap!
Conclusion

- Cyber-physical systems are ubiquitous
- Information distillation challenges will become more dominant
- Research frontier:
  - Tools for ingesting unstructured data and converting it into structured form
  - Tools for learning (take the human out of the loop)
  - Approaches for resilience: producing reliable outputs using unreliable components
- Goal: Elevate the human to progressively higher-level “supervisory” roles in CPS systems; bridge the gap between a growing data volume and our cognitive limits to consume it.