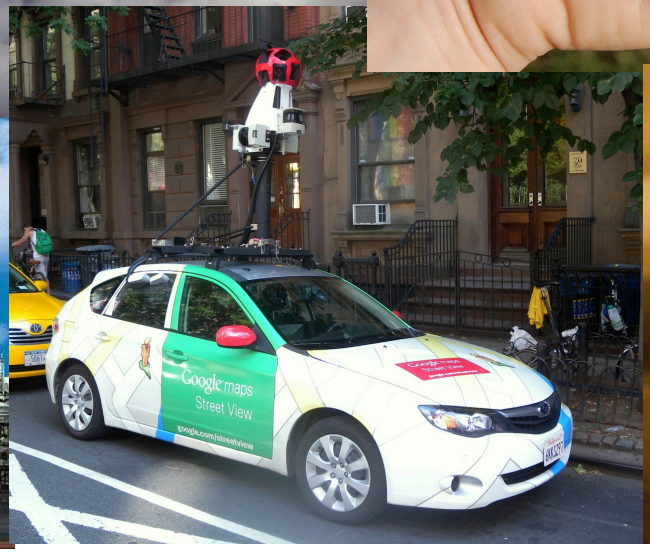
Security of Cyber-Physical Systems: Challenges and Approaches

Insup Lee and James Weimer
PRECISE Center
Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania

HotSoS
Hanover, MD
4 April 2017

Cyber-Physical Systems

We are heading towards (living in?) a sensor-driven world



need control systems capable of operating in malicious environments

Cyber-Physical Systems Security

BBC News Sport Weather Capital Culture Autos

NEWS TECHNOLOGY

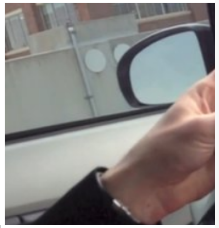
Home US & Canada Latin America UK Africa Asia Europe Mid-East Business Health Sci/Environme

25 July 2013 Last updated at 19:04 ET

Share

Car hackers use laptop to control standard car

By Zoe Kleinman
Technology reporter, BBC News

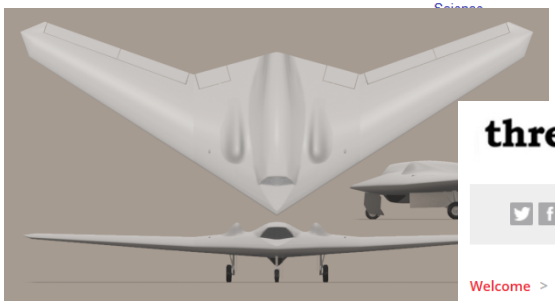


The New York Times

WORLD U.S. N.Y. / REGION BUSI
AFRICA AMERICAS ASIA PACIFIC

Worm Was Perfect f

By WILLIAM J. BROAD and DAVID E. SANG
Published: November 18, 2010



The Washington Post PostTV Politics Opinions Local Sports National World Bu

National Security

Home > Collections > Surveillance

Iran says it downed U.S. stealth drone; Pentago acknowledges aircraft downing

By Greg Jaffe and Thomas Erdbrink, December 04, 2011

A secret U.S. surveillance drone that went missing last week in western Afghanistan appear have crashed in Iran, in what may be the first case of such an aircraft ending up in the hand an adversary.

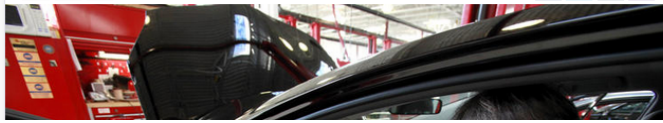
Iran's news agencies asserted that the nation's defense force Iranian reports said was an RQ-170 stealth aircraft. It is des

YAHOO! NEWS

Search News Search Web

News Home
U.S.
World
Politics
Tech

Hackers find weaknesses in car computer systems



threatpost CATEGORIES FEATURED PODCASTS VIDEOS

08/22/14 6:13 NIST Releases Secure Shell Guidance Document http://t.co/rHpQXdZojI

Welcome > Blog Home > Hacks > Researchers Hack GPS, $80M Yacht Veers Off Course

WIRED

Hackers Remotely Kill a Jeep on the Highway—With Me in It

BUSINESS DESIGN ENTERTAINMENT GEAR SCIENCE SECURITY

Obama Order Sped Up Wave of Cyberattacks Against Iran

By DAVID E. SANGER
Published: June 1, 2012

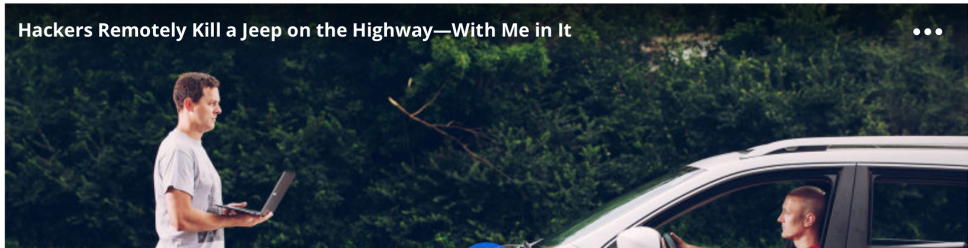
WASHINGTON — From his first months in office, President Obama secretly ordered increasingly sophisticated attacks on the computer systems that run Iran's main nuclear enrichment facilities, significantly expanding America's first sustained use of cyberweapons, according to participants in the program.

Mr. Obama decided to accelerate the attacks — begun in the Bush administration and code-named Olympic Games — even after an

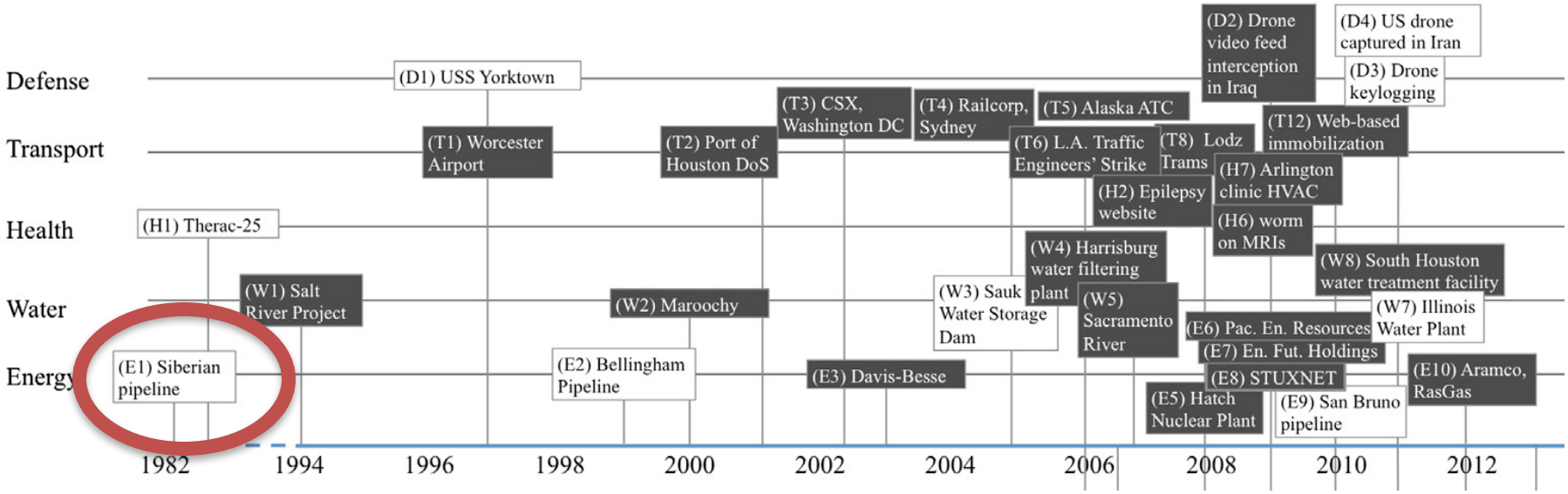
FACEBOOK
TWITTER
GOOGLE+
EMAIL
SHARE
PRINT
SINGLE PAGE
REPRINTS

ANDY GREENBERG SECURITY 07.21.15 6:00 AM

HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY—WITH ME IN IT



CPS security incidents

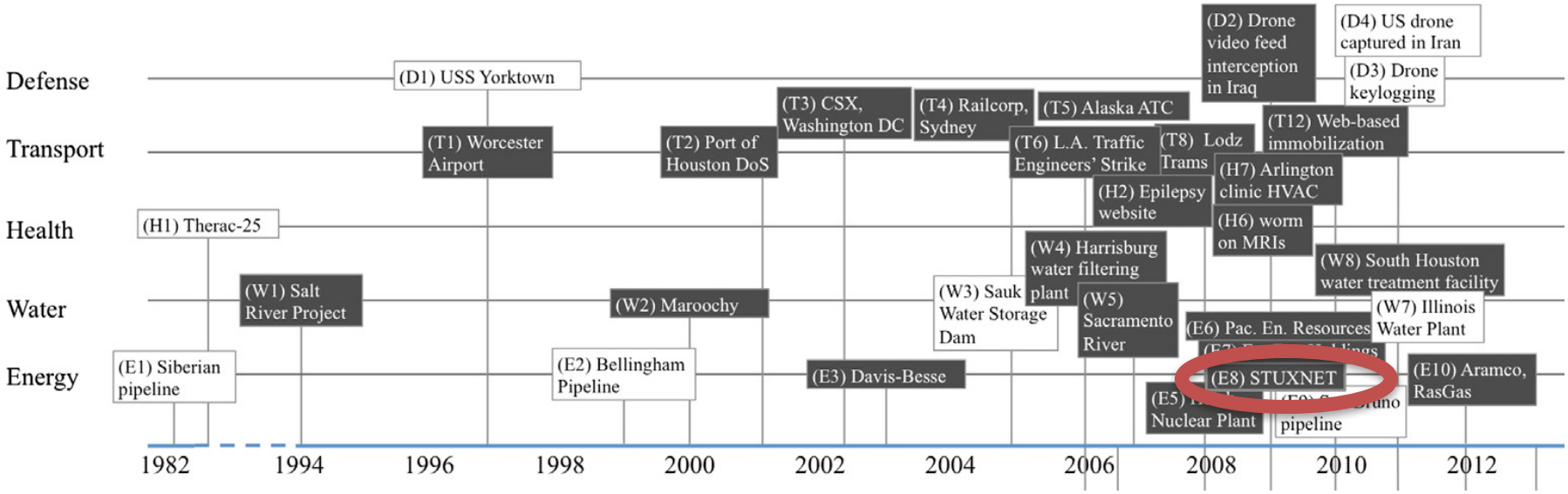


– Siberian pipeline: June 1982:

- Soviets stole control software from Canadian company.
- US influence Canadian company to alter code such that pipeline pressures would build up.
- explosion could be seen from space.

cyber-physical attacks: a growing invisible threat: George Loukas, 2015.

CPS security incidents

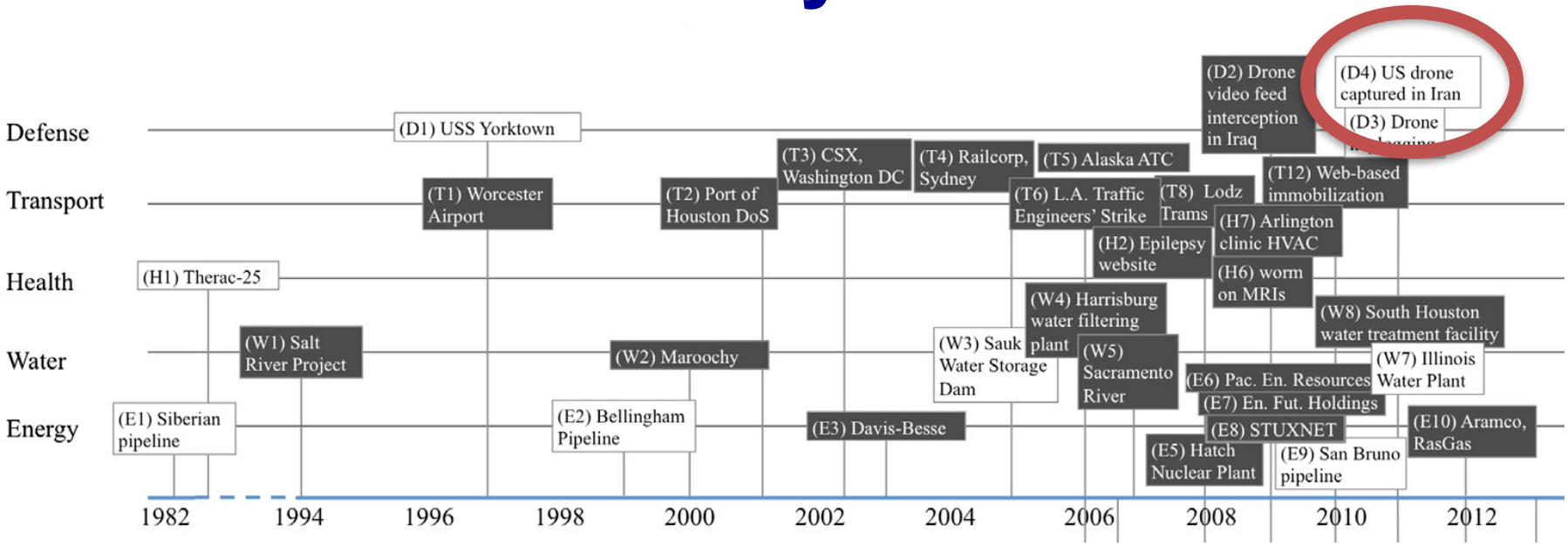


– Stuxnet: 2009:

- Attack on Iranian nuclear facility
- Used 4 undiscovered exploits targeting control

cyber-physical attacks: a growing invisible threat: George Loukas, 2015.

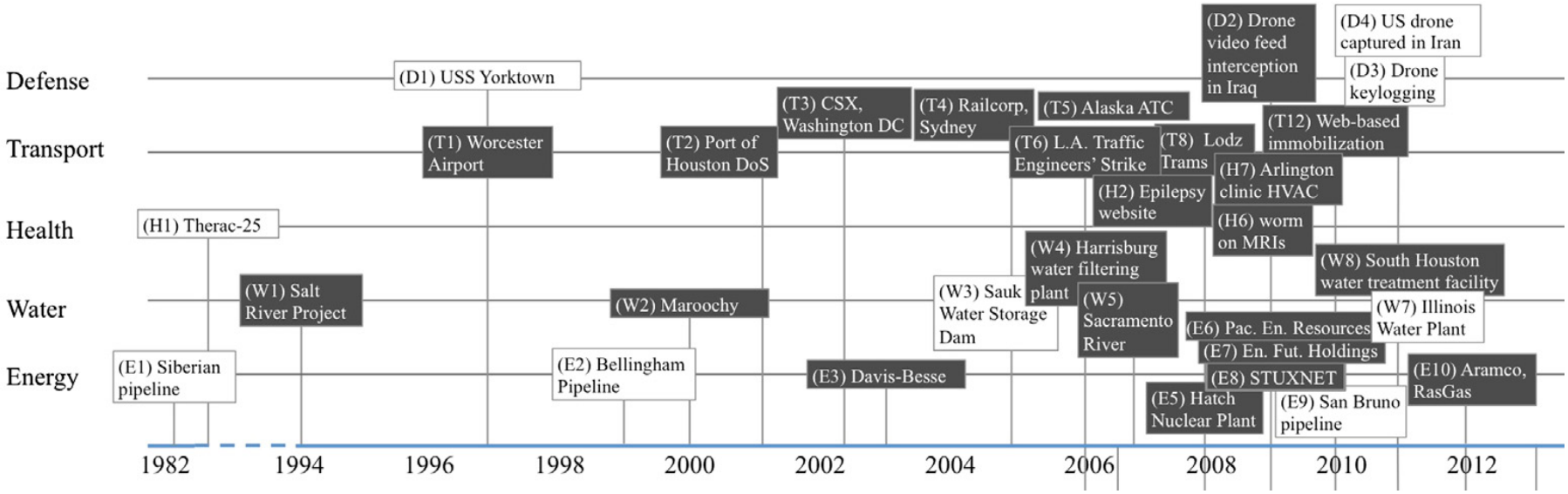
CPS security incidents



- US Drone captured: 2011:
  - Iran captured predator drone that landed in the wrong area.
  - GPS spoofing
  - "System" worked perfectly
    - sensor measurements where wrong

cyber-physical attacks: a growing invisible threat: George Loukas, 2015.

CPS security incidents

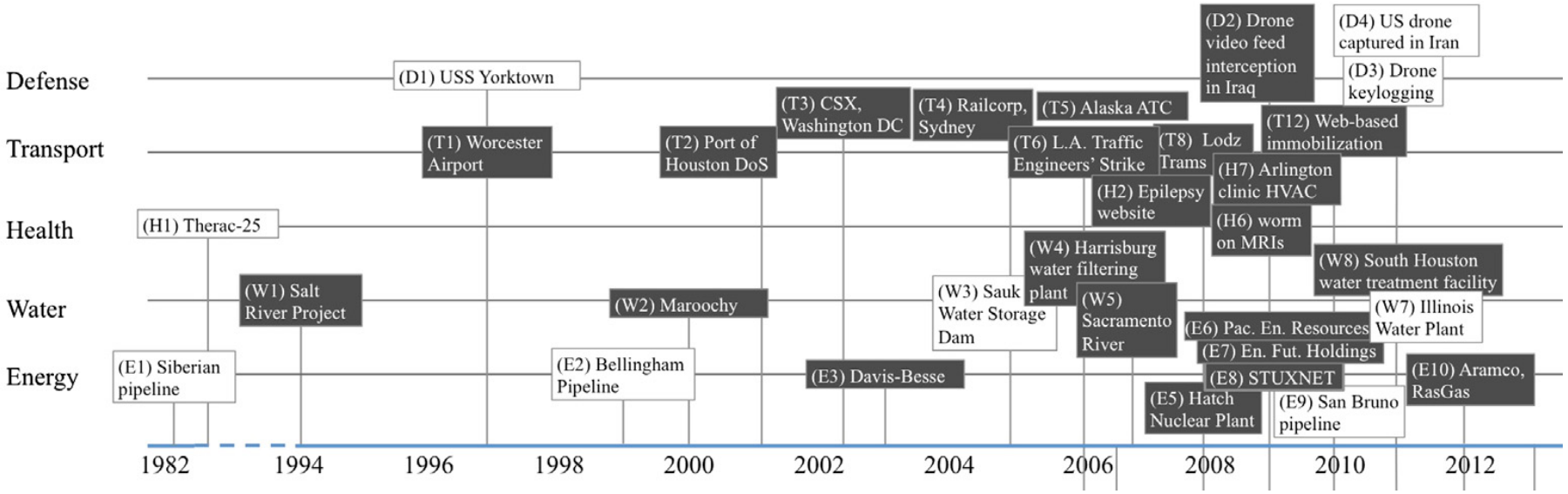


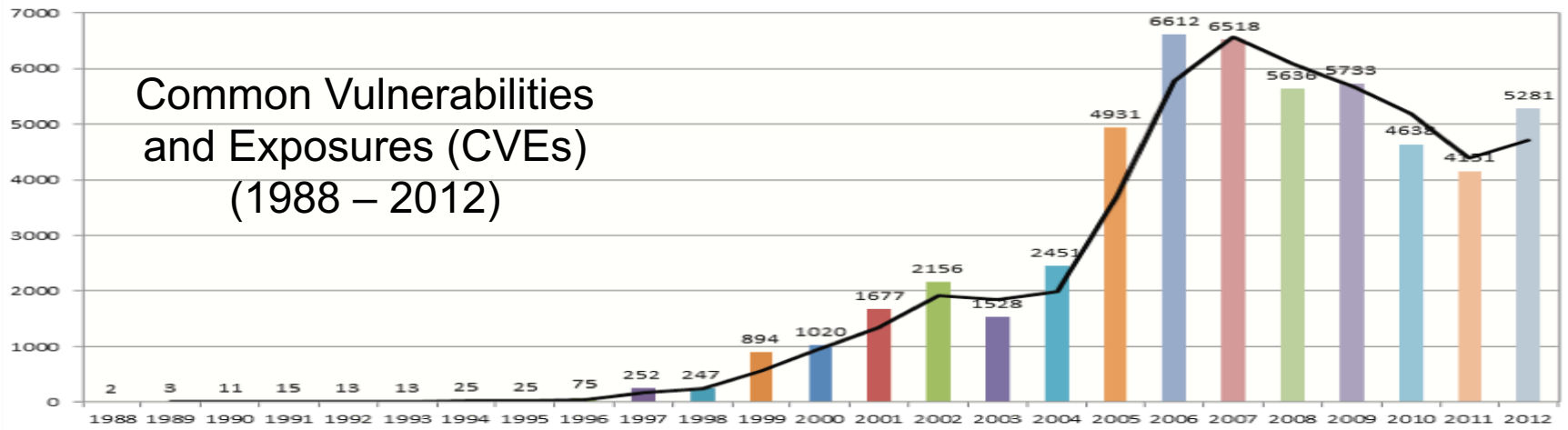
– IoT DDoS : October 21, 2016

- thousands of devices overtaken using default passwords
- organized into botnet to flood DNS provider
- took down many major websites
  - $17 Billion cost to economy (0.1% of GDP)

cyber-physical attacks: a growing invisible threat: George Loukas, 2015.

CPS security incidents

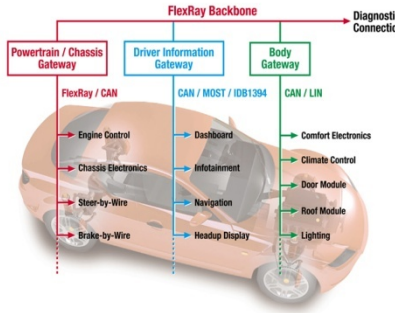




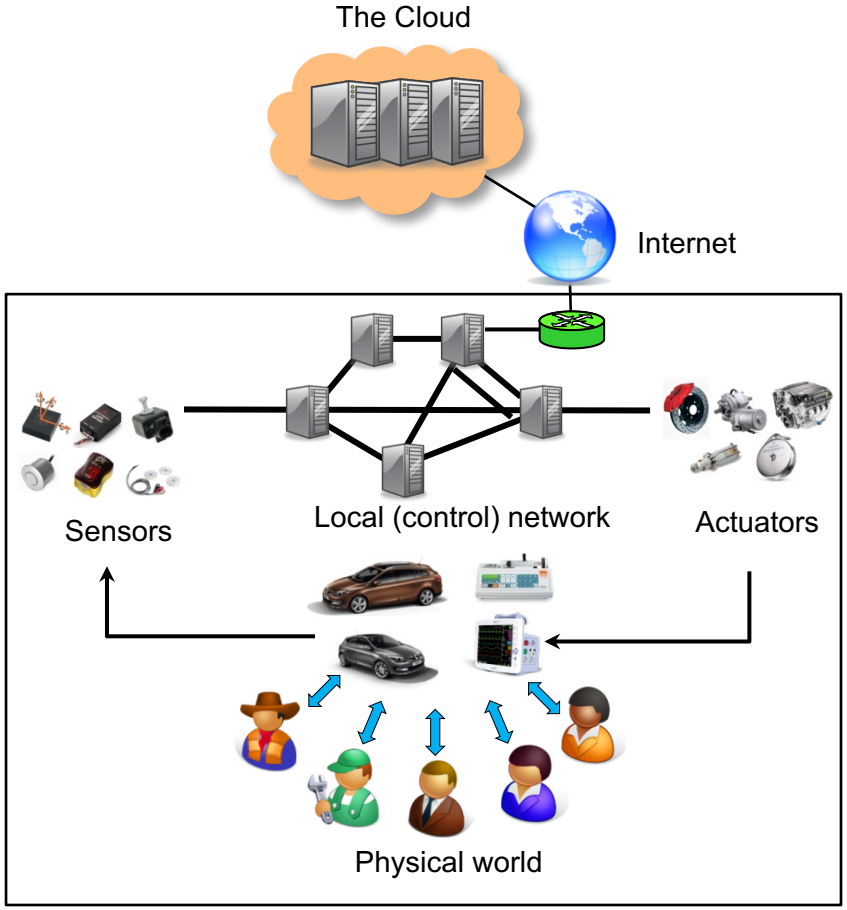
cyber-physical attacks: a growing invisible threat: George Loukas, 2015.

25-years of vulnerabilities, 1988-2012. Yves Younan.

Typical CPS Architecture

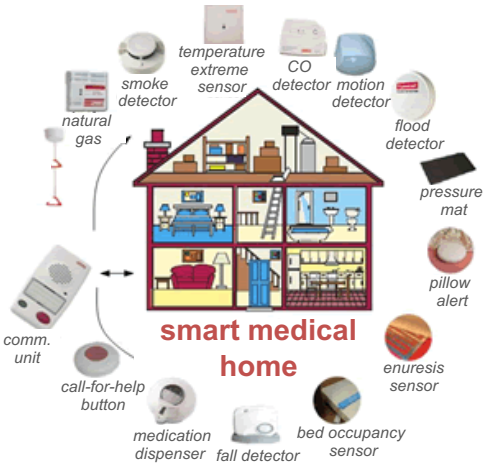


complex platform architecture



Internet-connected car

medical devices

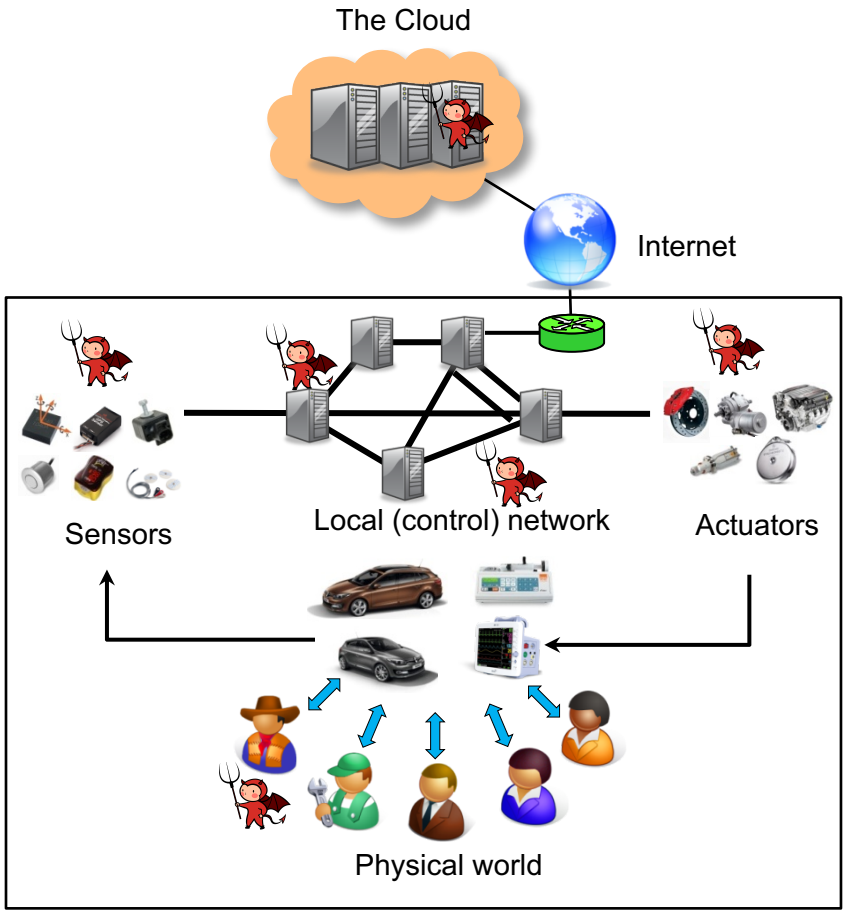


What is CPS Security?

- A CPS attack whose goal is to (negatively) affect the interaction between a CPS and the physical world
  - Originates through any attack surface
    - cyber, physical, or any combination of cyber/physical
- CPS security concerns the development of technologies for defending against CPS attacks
  - e.g., discovering new vulnerabilities, techniques for detection/mitigation/recovery, …

CPS Attack Surfaces

- Cyber attack surfaces
  - e.g., communication, networks, computers, databases, ...
- Physical attack surfaces
  - e.g., locks, casings, cables, ...
- Environmental attack surfaces
  - e.g., GPS signal, electro-magnetic interference, battery draining/cycling/heating, …
- Human attack surfaces
  - e.g., phishing, bribing, blackmail, etc.



CPS Security Challenges

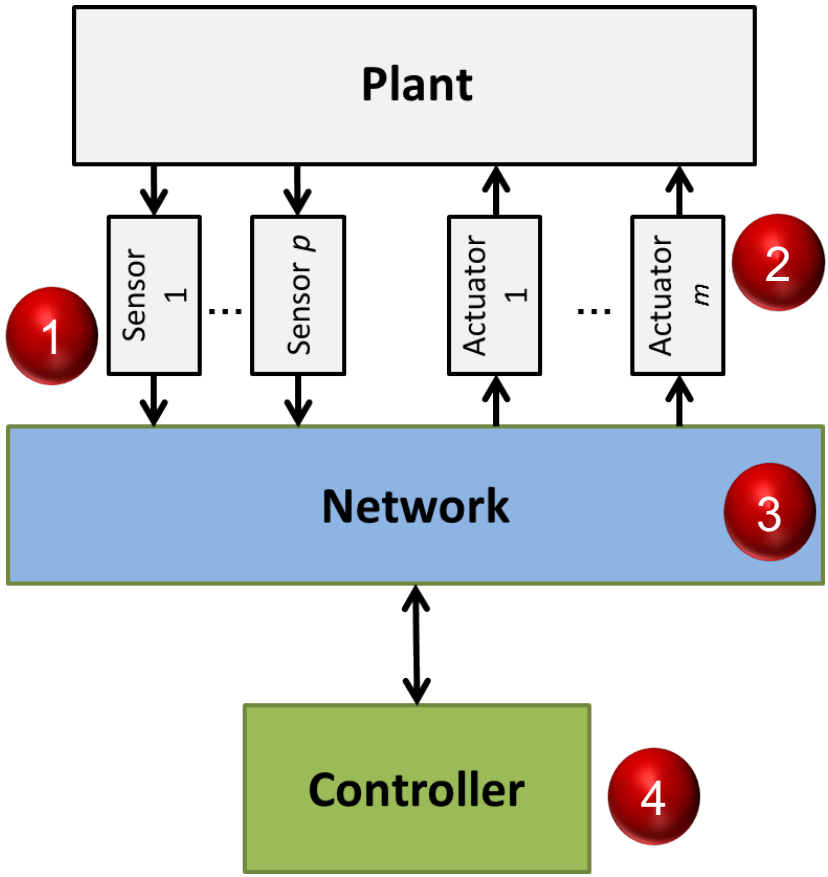
- Foundational Challenges
  - How to build an ideal resilient CPS?
  - Quantifying CPS attacks effectiveness
    - wide variability in metrics for CPS security
    - concerns depend on the CPS mission
  - System evolution
    - operate in many different physical environments
    - adapt to physical surroundings
  - Operating scenarios restrict defensive capabilities
    - patching and frequent updates, are not well suited for control systems
    - real-time availability provides a stricter operational environment than most traditional IT systems.
    - legacy systems may not be updated
- Social and Legal Challenges
  - What solutions will be accepted by practitioners?
  - Who/what is liable when such a system fails due to security and privacy attacks?

Improving CPS security

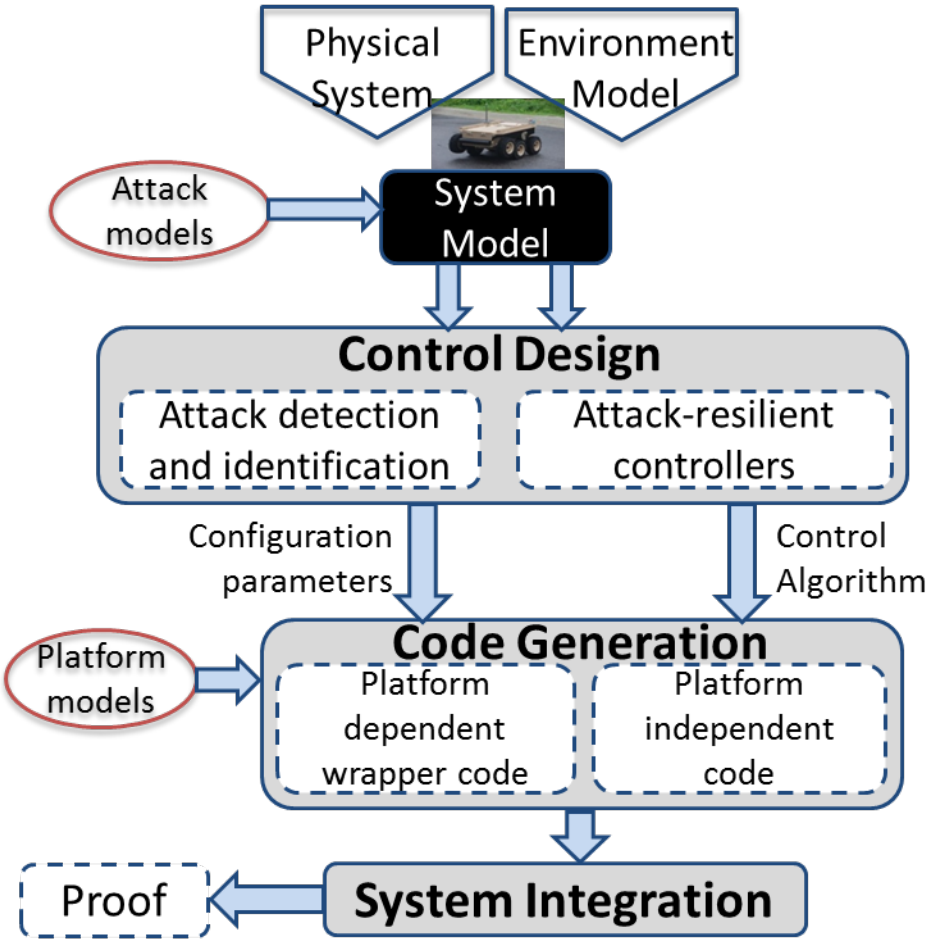
- Apply suitable best (cyber) security practices
- CPS can provide additional information
  - CPS architecture / physical-world interface
    - e.g., multiple sensors, actuators, controllers
  - Environmental context
    - e.g., operating conditions (rain/snow), geographic location
  - Physical constraints and guarantees
    - e.g., laws of physics, bounds on power, CPU speed, network bandwidth
- How to leverage additional information to improve CPS security?

Attacks on Control Systems

1. Sensor attacks
   - The attacker can arbitrarily change sensor measurements.
2. Actuator attacks
   - The attacker can arbitrarily change actuator values.
3. Communication attacks
   - The attacker can change messages sensors->controllers, and controllers->actuators.
4. Controller attacks
   - The attacker can change the controllers' parameters (e.g., execution model) or even the controllers' code.



Platform-Aware CPS Design Framework



- Control-level techniques
  - Attack detection and identification using redundant sensing and model of the system's dynamics
  - Attack-resilient control architectures
- Code-level techniques
  - Ensure that the control code is correctly implemented and integrated
  - Preventing malicious code injection into the controller

Goal: Ensure that a CPS maintain a degree of control even when the system is under ***cyber*** and/or ***physical*** attack

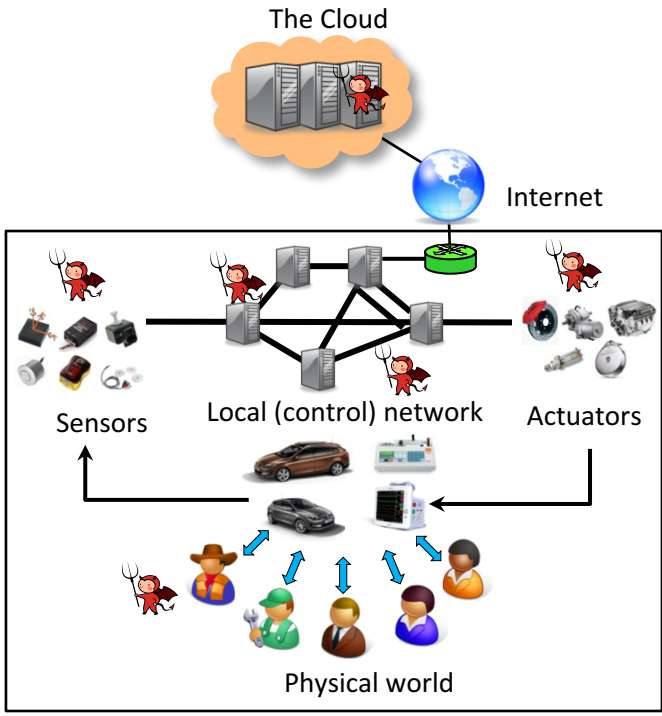
Security-Aware Control Design

- Physical world abides by the laws of physics!
- Physical interfaces introduce new attack vectors!

- How can we exploit ***limited*** knowledge of laws of physics (system model) for control and attack detection/identification
- Approach
  - Analyze the difference between observed measurements and `expected' system behavior over a time window for different attack models

Attack-Detection and Identification (ADI)

- Problem: How can we ***detect*** and ***identify*** which system sensors have been compromised
-
- Approach: Exploit *spatial and temporal redundancy*
  - sensor fusion
  - resilient state estimator



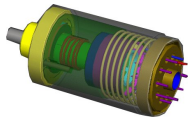
The rest of the talk

- Dealing with sensor attacks
  - Sensor fusion based on abstract sensor models
  - Attacks vs. transient faults
  - Resilient state estimator
- Security-aware CPS architecture
  - Human-in-the-loop
  - Checkpointing and recovery

DEALING WITH SENSOR ATTACKS

Motivation

- Modern CPS are equipped with multiple sensors (e.g., GPS, encoder, camera, IMU)
  - Can separately estimate the same physical variable (e.g., velocity)
  - This redundancy can be used to improve system performance



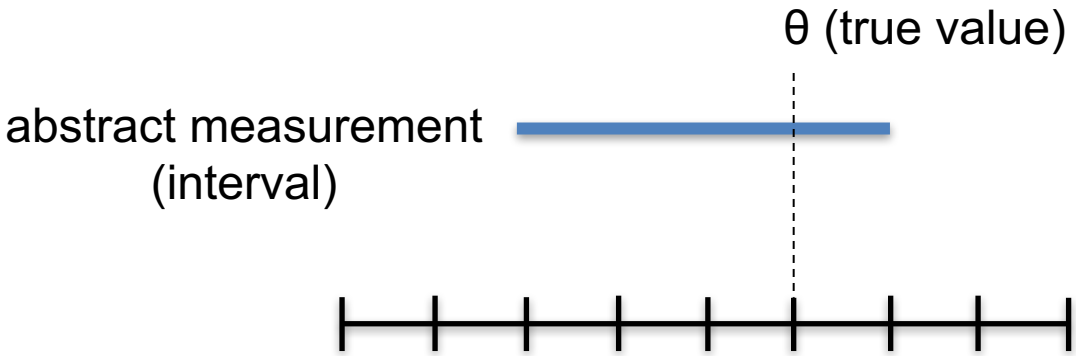
- Some sensors may be vulnerable to sensor attacks (e.g., GPS spoofing)

- Low-precision sensors can be used to improve attack detection and identification (e.g., ambient FM signals)

- How can redundancy improve system resilience?

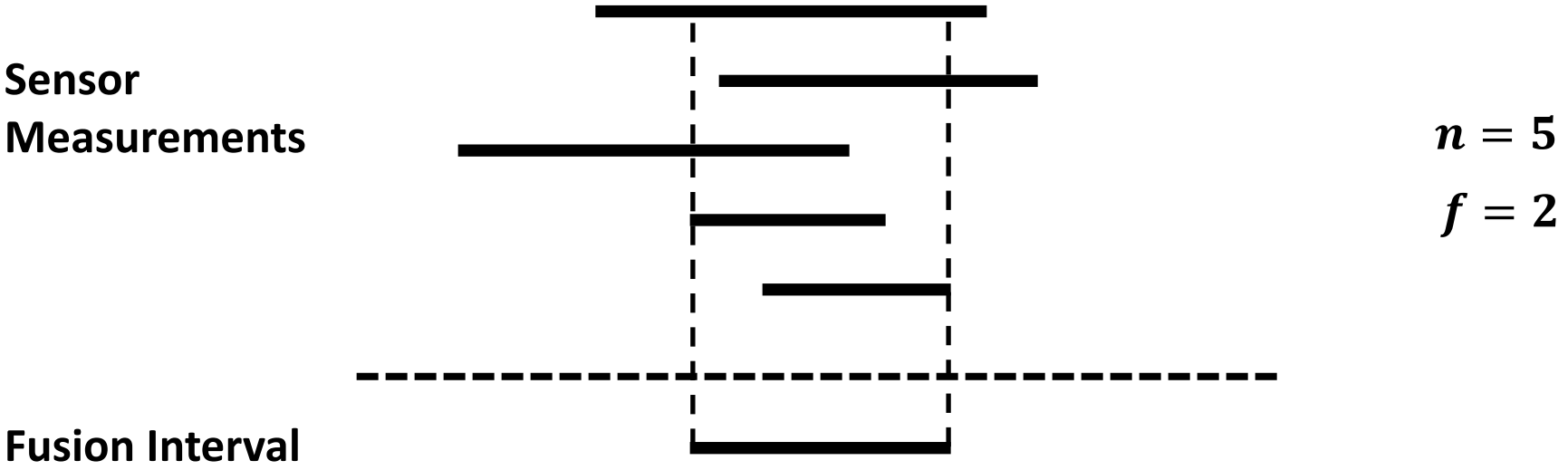
Abstract Sensor Model

- Most sensor models assume probabilistic noise
  - Used to argue about expected operation
  - Not applicable to analyzing rare events (e.g., attacks)
- Interval containing all points that may be the true value
- No assumption on noise distribution
- The size of the interval reflects the accuracy of the sensor
- Well-suited for worst-case analysis



Fusion Algorithm

- Based on algorithm developed by Marzullo*
- Input are n real intervals and a number f
- At most f sensors under attack ($f < \lceil \frac{n}{2} \rceil$)
- Output is a "fusion interval"
  - Smallest to largest point contained in $n - f$ intervals



* Marzullo, K., "Tolerating Failures Of Continuous-Valued Sensors." ACM Transactions on Computer Systems, vol. 8, (no. 4), pp. 284-304, Nov. 1990

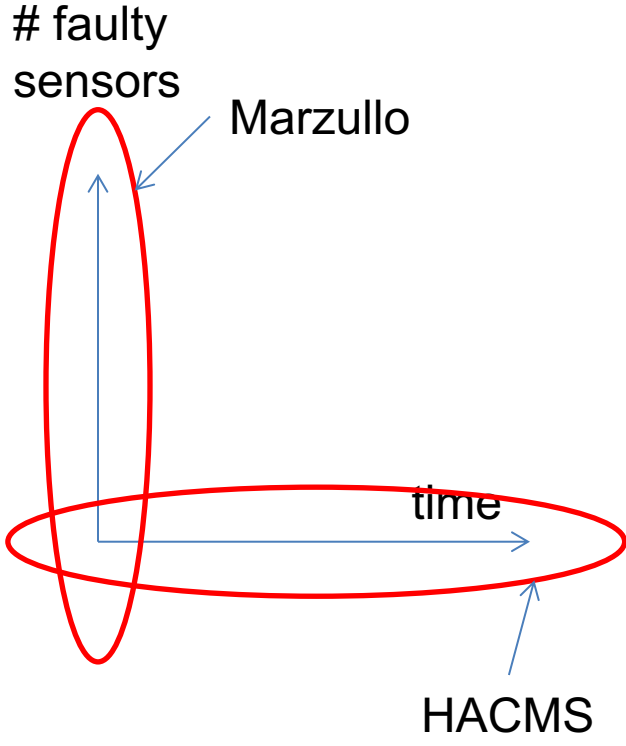
Sensor Fusion under Attack over Time

- Our approach
  - Extend Marzullo's work to attacks and over time
  - Use a dynamic system

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + w$$

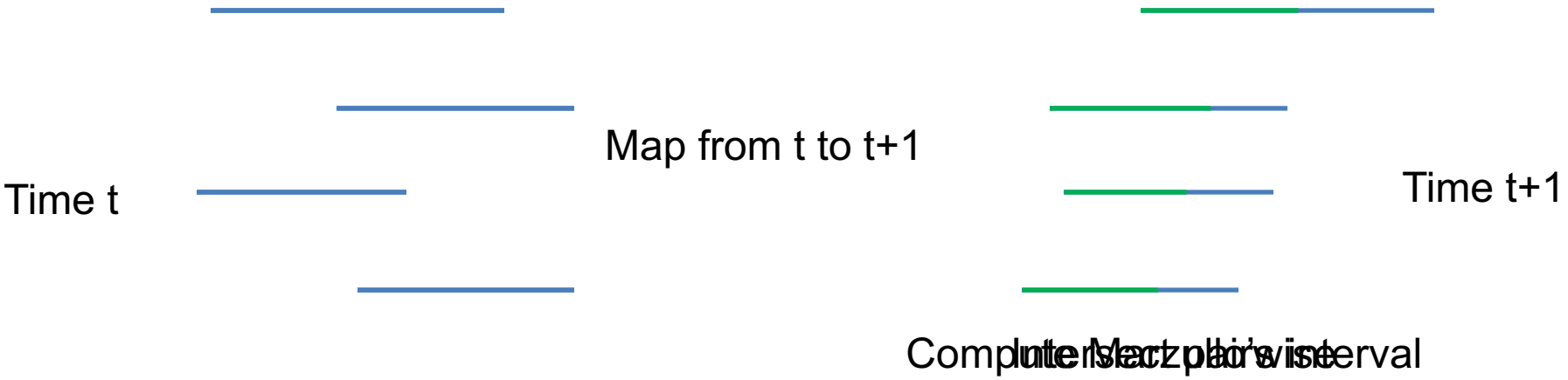
    - where w is a disturbance, and
    - we know **A** with some uncertainty

- Using time will help us isolate malicious sensors
  - Attacker no longer able to give unreasonable measurements



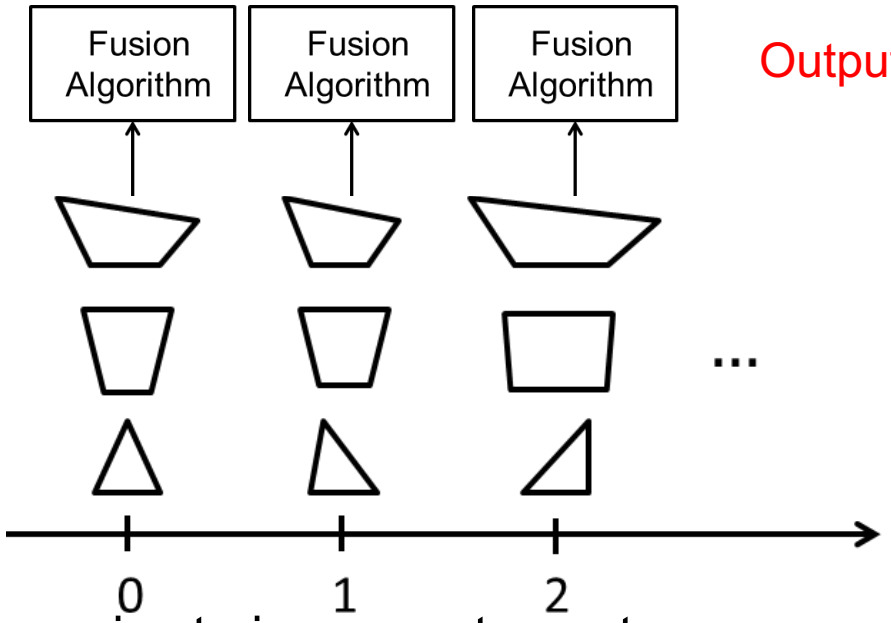
Our Results

- Identified a mapping method: is this optimal?



- Identified an “optimal” attack strategy for attacker
- In addition:
  - Analyze this optimal attack policy with the above mapping
  - App to Integrate sensor measurements from smartphones with On-Board Diagnostic (OBD) system (for American Built Car)

Measurement History

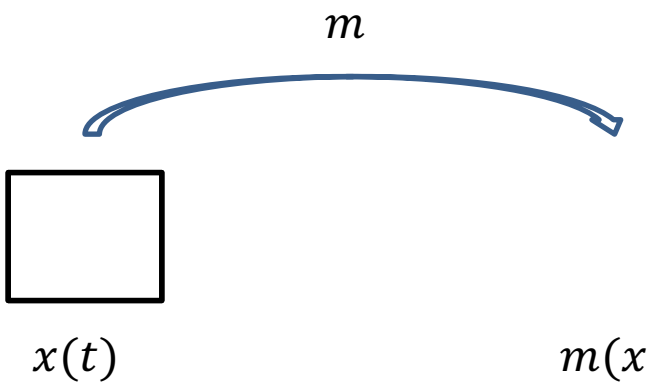


Outputs a fusion polyhedron!

- Use system dynamics to incorporate past measurements
  - Map x(t) to time t+1 using system dynamics:

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + w$$

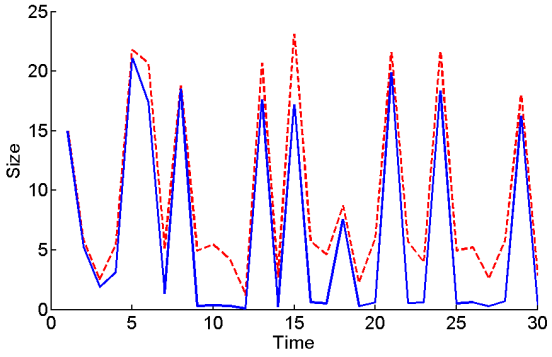
Identified 5 algorithms of using past measurements



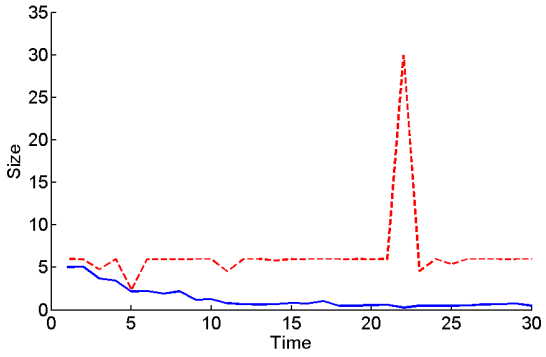
Results

- **Theorem**: *pairwise_intersect* is the best of all five methods*

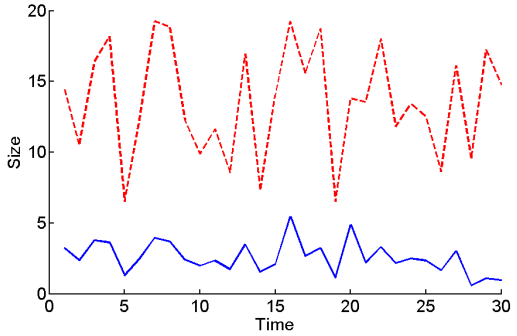
- Simulations – estimate velocity/position
  - Camera, GPS and two encoders
  - One sensor always under attack
  - Red: Volume of fusion polyhedron with no history
  - Blue: Volume of fusion polyhedron with *pairwise_intersect*



Camera under attack



Encoder under attack

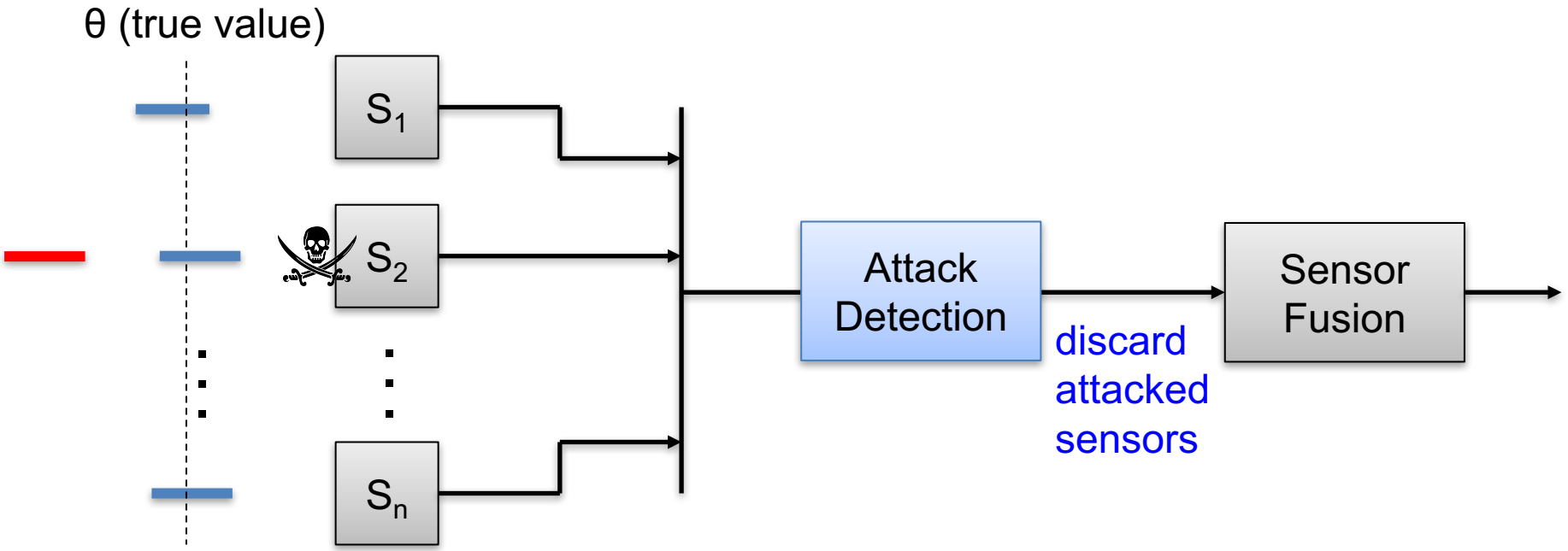


GPS under attack

*Assuming the transition matrix A is full rank.

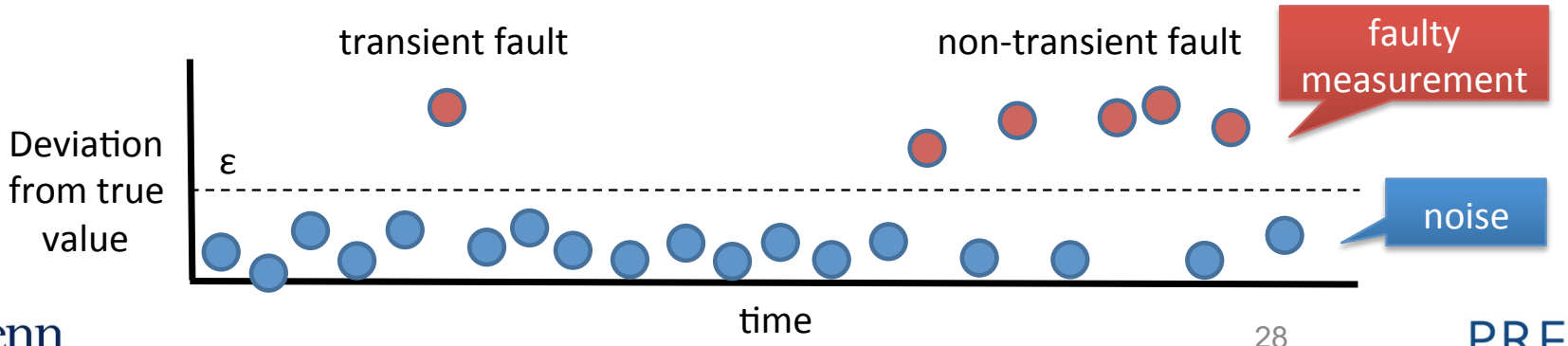
Attack Detection

- Sensor fusion
  - produce a better measurement → improve performance
- Attack detection
  - identify and discard attacked sensors → improve resilience

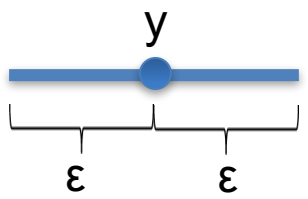


Attacks vs. Faults

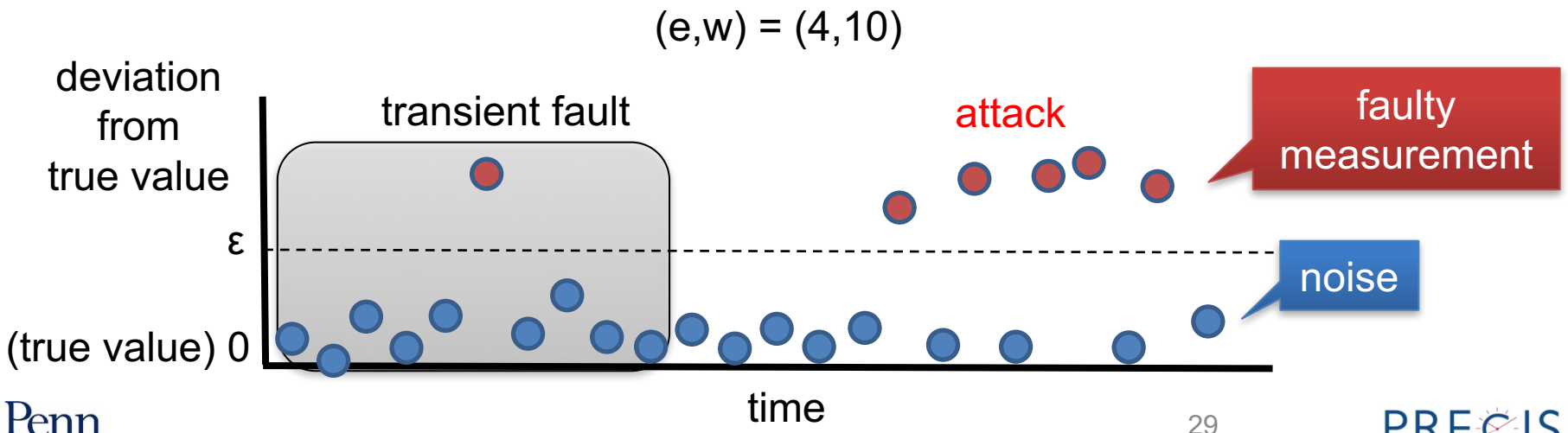
- The sensor-fusion approach is too conservative in that they treat faults and attacks in the same way
- Types of sensor measurements and faults
  - Noise vs. faulty measurement
  - Transient faults: occur shortly and disappear
  - Non-transient (permanent) faults: persist for a longer period of time
  - This work: focus on attacks that manifest as permanent faults



Transient Fault Model (ε, e, w)



- Error bound ε signifies the worst case noise threshold
- Transient threshold (e, w)
  - allows at most e faulty measurements within window size w
  - If exceeded, fault is non-transient
- Not conforming to transient fault model → considered as attacked

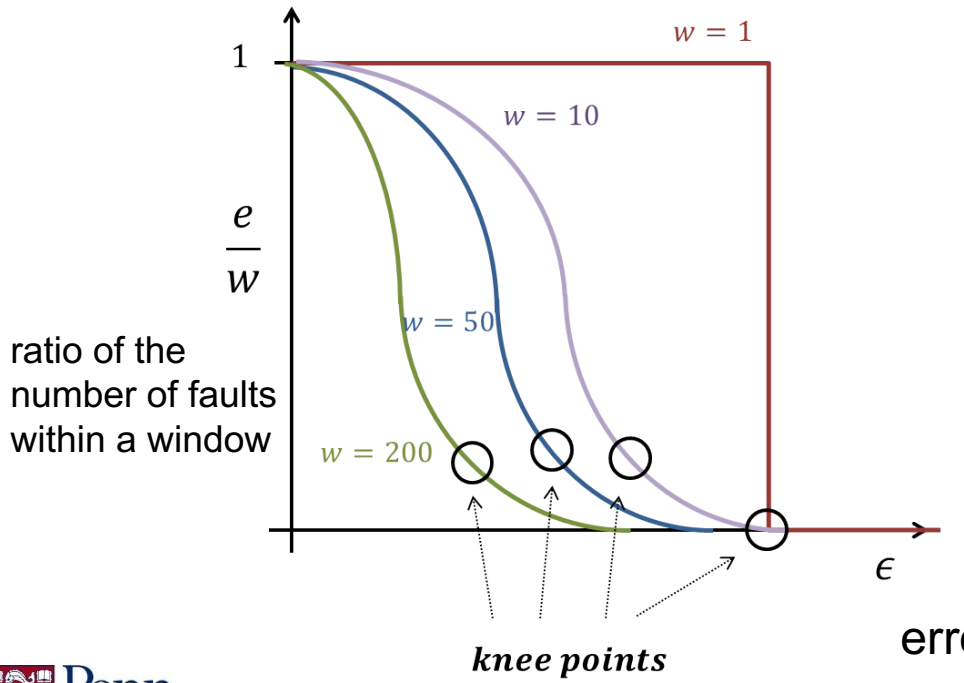


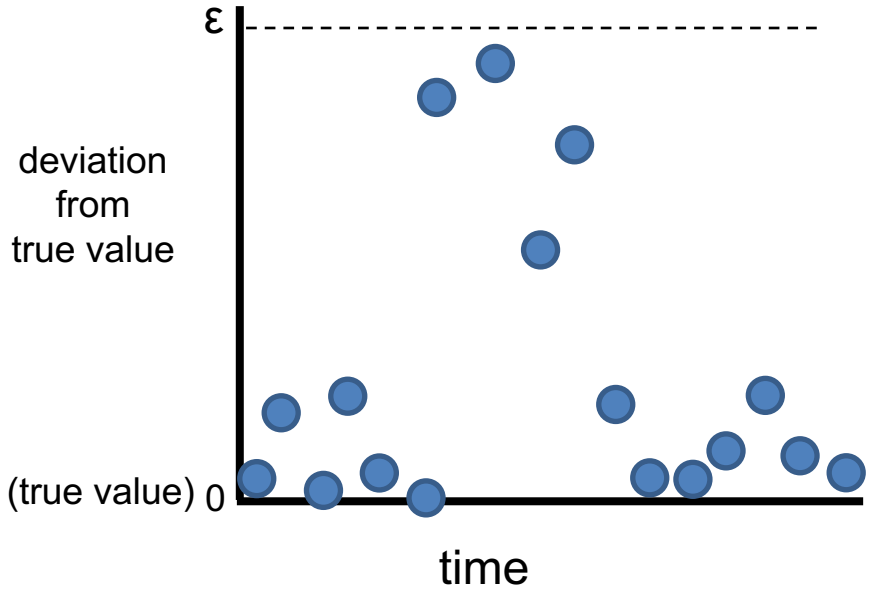
Two Problem Statements

- Problem 1 (Transient Fault Modeling)
  - Develop a transient fault model for each sensor from training data
- Problem 2 (Detection and Identification)
  - Given transient fault model (ε,e,w) for each sensor, develop an algorithm to detect and identify sensor attacks

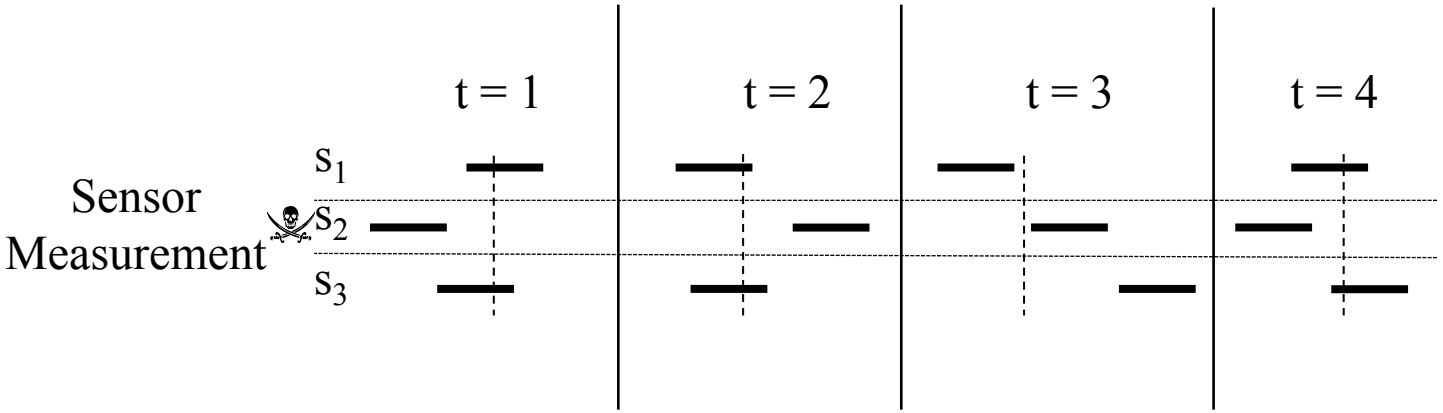
Problem 1: Transient Fault Modeling

- Sometimes provided by manufacturer
  - E.g., Bosch
- Otherwise, have to choose them based on data
- ***e*** : the number of faults within time window size of **w**
- ***ε*** : error bound (the worst case noise bound)
  - Choose ε small enough: observe faulty measurements
  - Choose ε big enough: do not treat noise as faults





Problem 2: Attack Detection and Identification



Transient Fault Model

	(e_i, w_i)
s1	(1,3)
s2	(1,3)
s3	(1,3)

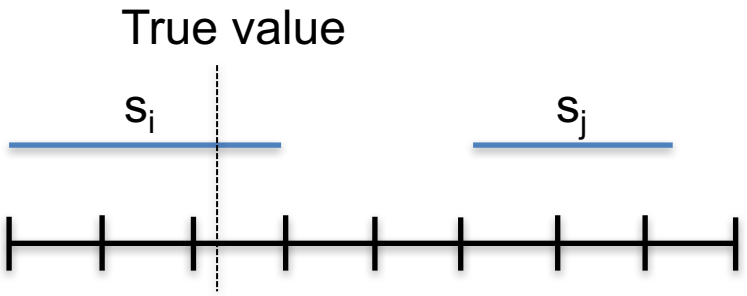
(ε is the half of interval size)

- Pairwise inconsistency of sensor measurements
  - unknown true value → unknown whether faulty or not
- Weak inconsistency
  - Two sensors are too far from each other at a certain time
- Strong inconsistency
  - Two sensors are frequently inconsistent over a time window

Attack Detection: Weak Inconsistency

- Two sensors s_i and s_j are weakly inconsistent at time t
  - iff one of them provides a faulty measurement
  - That is, $WI(i,j,t) \equiv F(i,t) \vee F(j,t)$
    where *F(i,t)* signifies that s_i provides a faulty measurement at t
- Cannot decide in general – true value not known
  - Sufficient condition exists: if the two sensors' intervals do not overlap. That is:

$$|y_i^{(t)} - y_j^{(t)}| > \epsilon_i + \epsilon_j \implies WI(i,j,t)$$

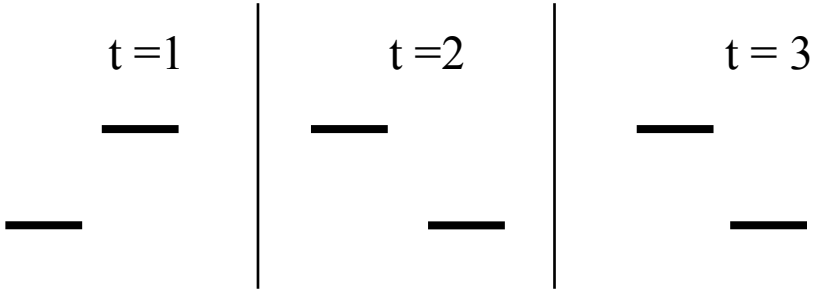


Attack Detection: Strong Inconsistency

- Two sensors s_i and s_j are strongly inconsistent
  - iff one of them is non-transiently faulty
  - That is, $SI(i,j,t) \equiv NTF(i,t) \vee NTF(j,t)$
    where *NTF(i,t)* signifies that s_i is non-transiently faulty at time t
- Again, cannot decide in general
  - Sufficient condition exists: the sensors are weakly inconsistent frequently. That is,

$$\sum_{t-\min(w_i,w_j)+1}^{t} WI_1(i,j,t') > e_i + e_j \implies SI(i,j,t)$$

$e_1 = e_2 = 1$
$w_1 = w_2 = 3$



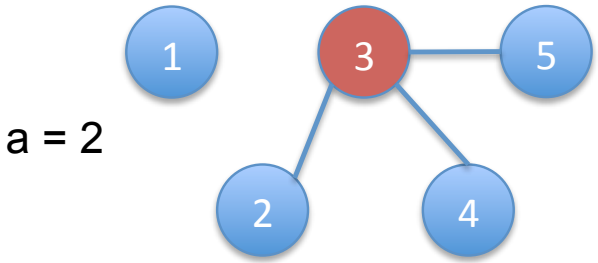
3 inconsistencies in a window of 3

Attack Identification

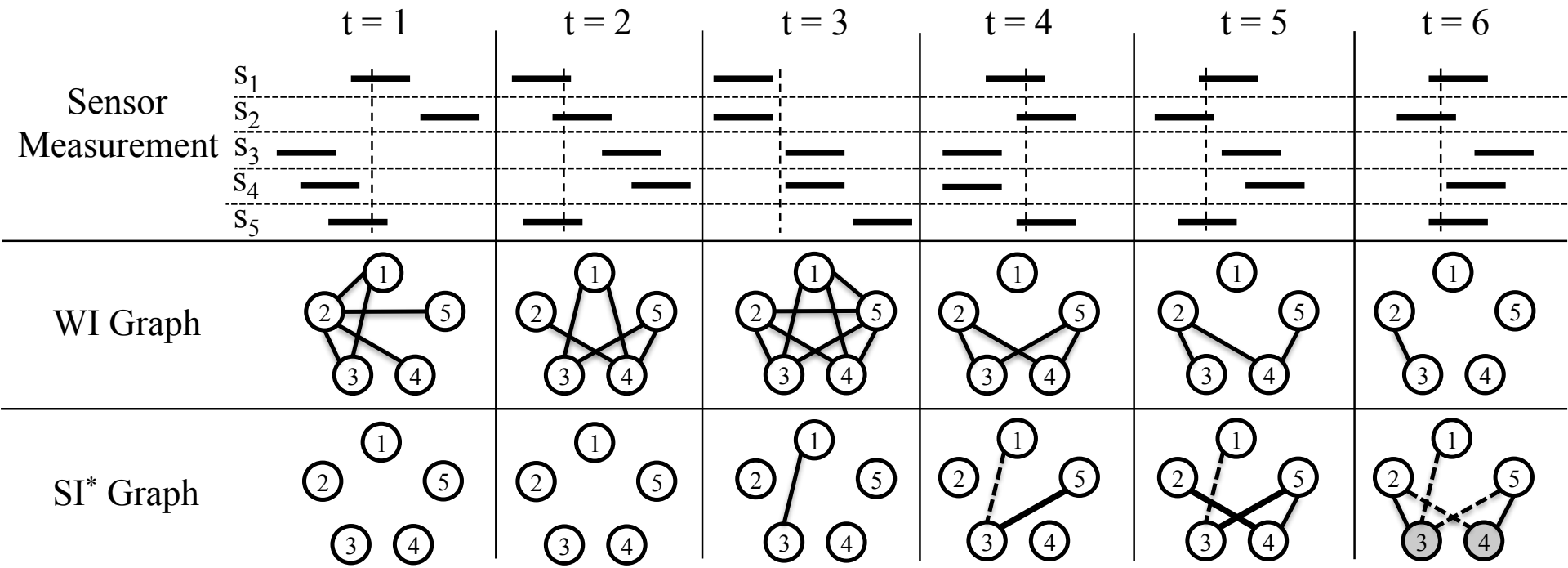
- For identification, it is necessary to assume that there exist at most '*a*' attacked sensors (where *a* < *n*-1).

- If sensor s_i is strongly inconsistent with '*a*' other sensors, then s_i is attacked, i.e., given i,

$$\mathbf{deg}(i) > a \implies A(i)$$



Example



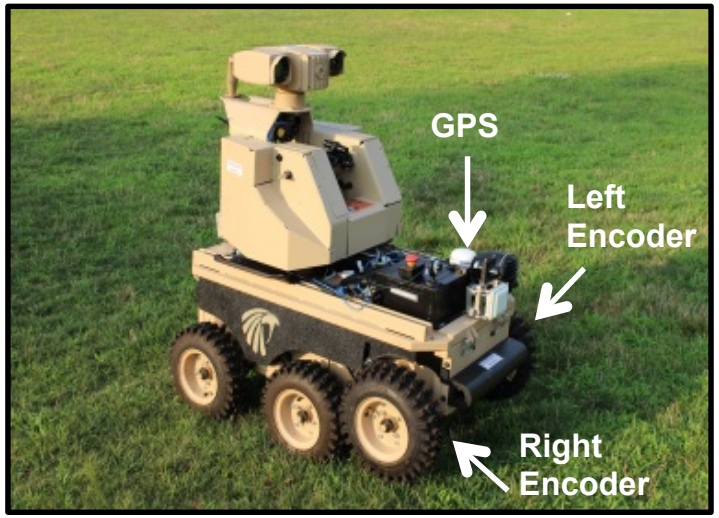
$n = 5$
$a = 2$

	$(\varepsilon_i, e_i, w_i)$
s1	(1,1,6)
s2	(1,2,5)
s3	(1,1,4)
s4	(1,2,6)
s5	(1,1,5)

- The attack is detected at time 3
- The attacked sensors s_3 and s_4 are identified at time 6

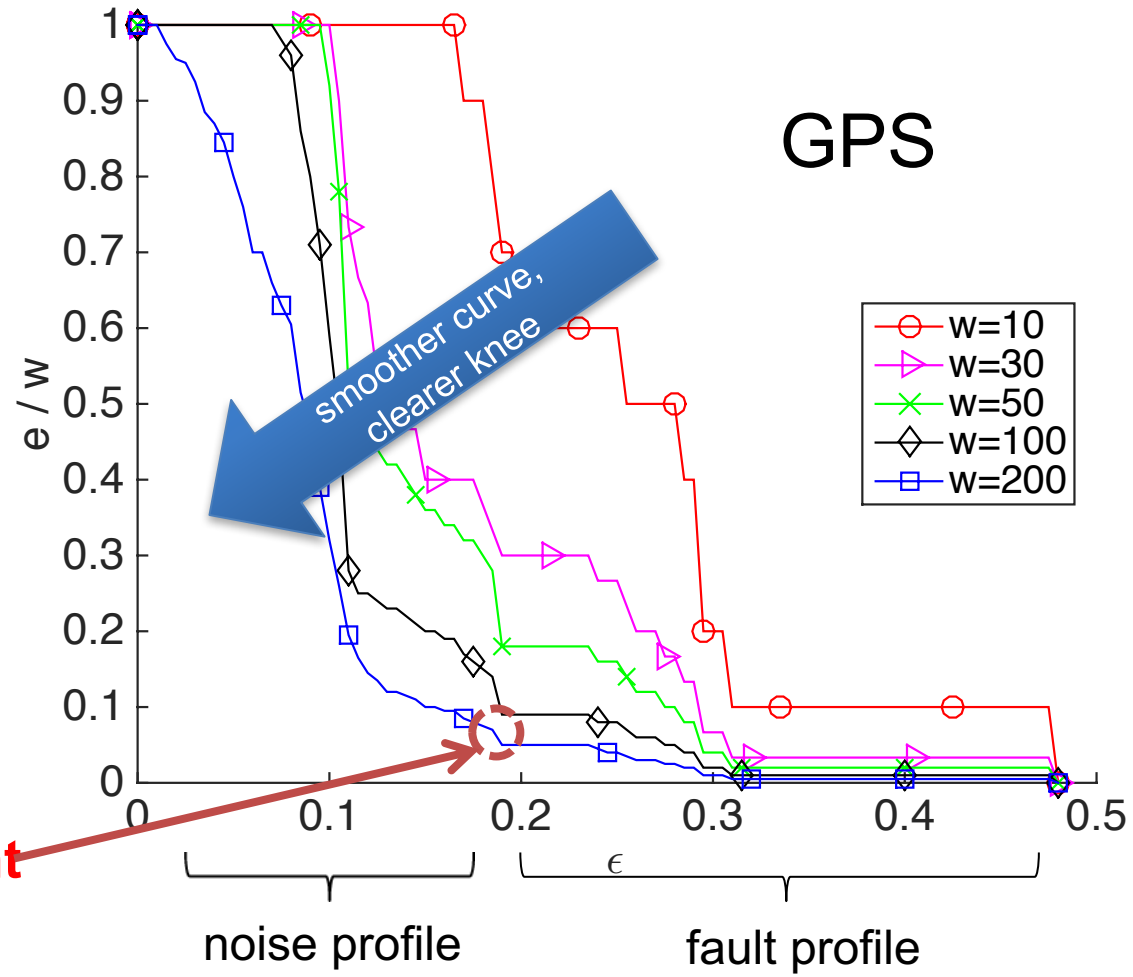
Case Study: Experiment Setup

- Driving an unmanned ground vehicle (called LandShark) in a straight line
- Gathering velocity measurements
  - Separately from left wheel encoder, right wheel encoder and GPS unit
  - At a rate of 10 Hz



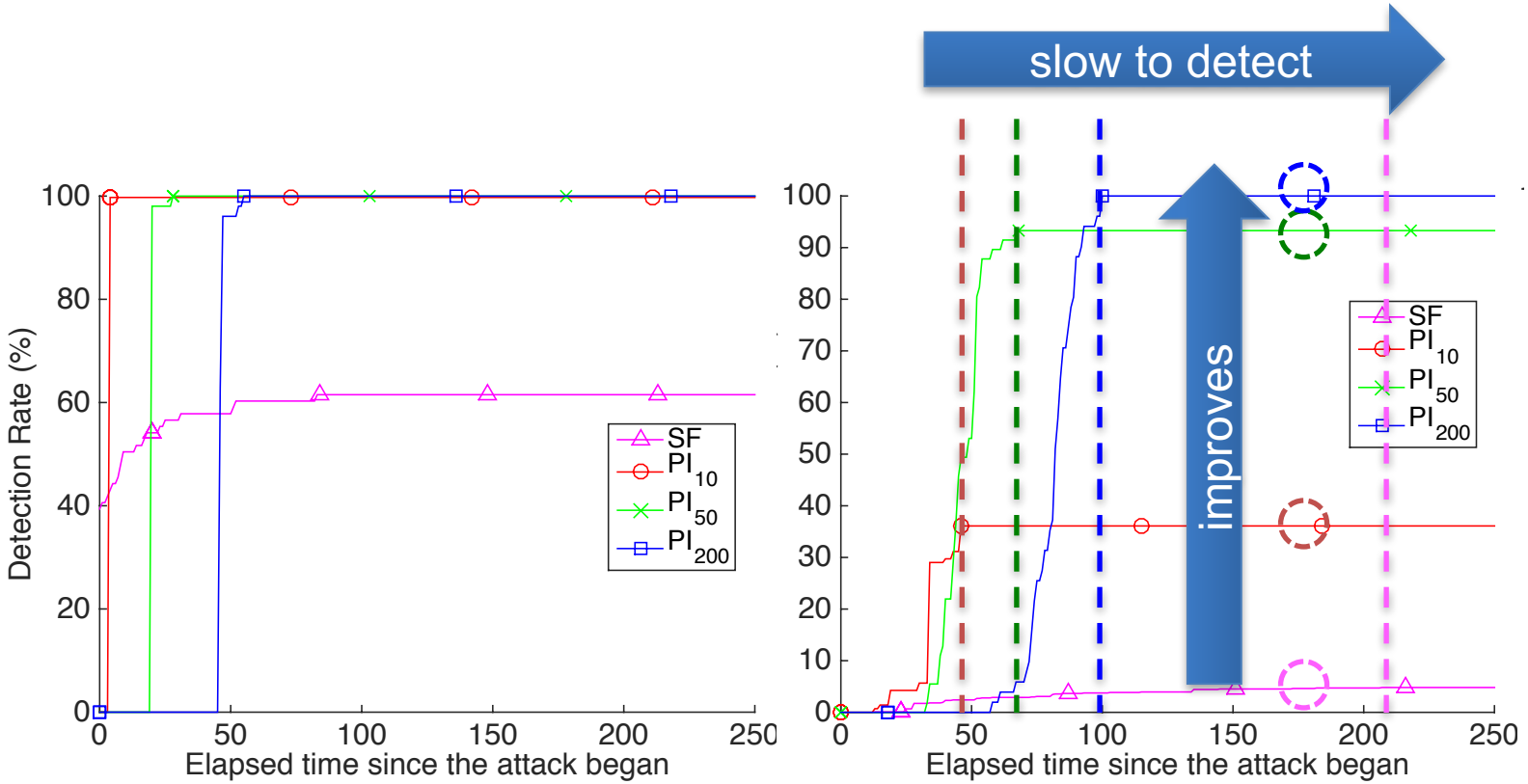
Transient Fault Model Parameter Selection

ratio of the number of faults within a window



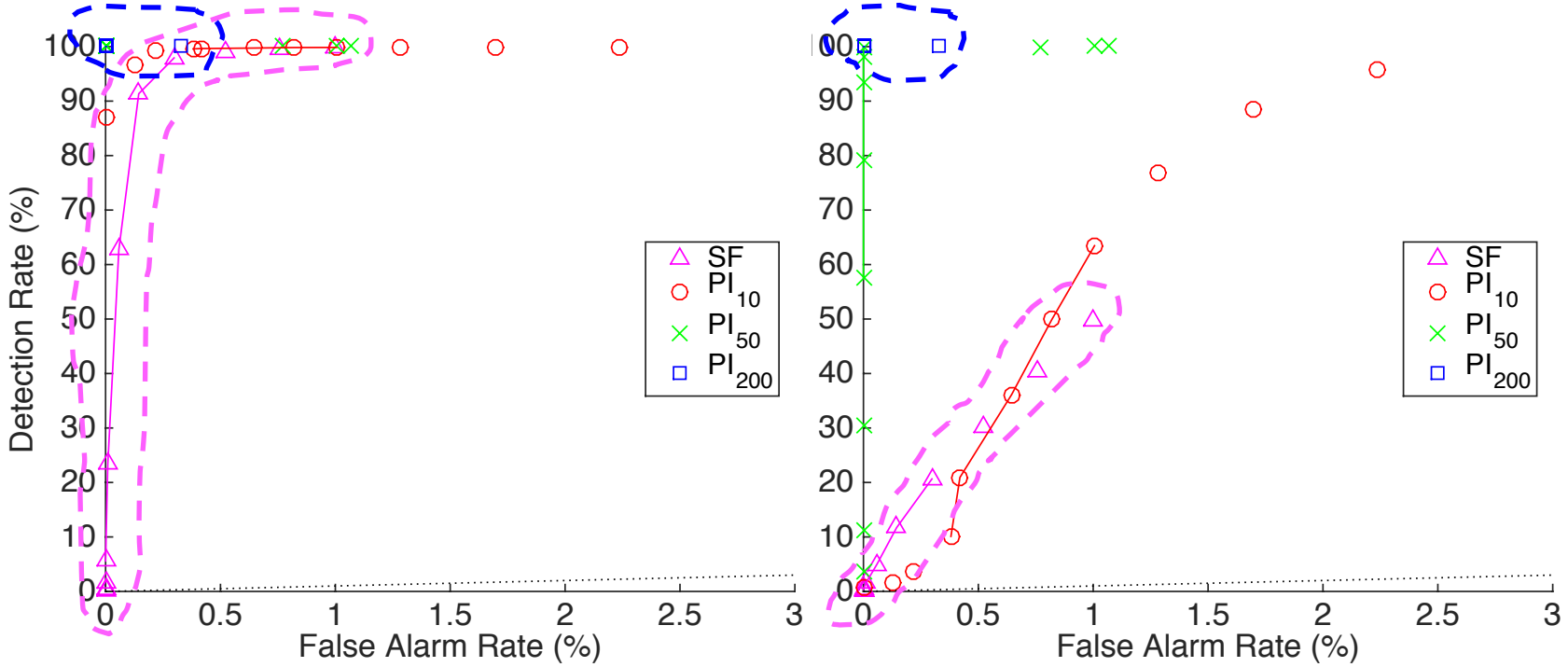
Detection Performance

- Detection rate vs. elapsed time



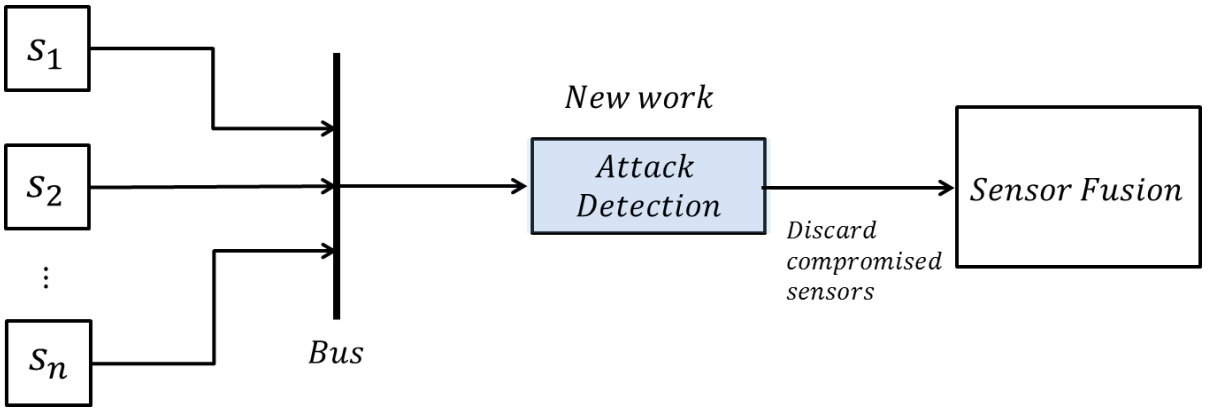
Detection Performance (cont.)

- Detection rate vs. false alarm rate when error bounds are varied



Summary

- Sensor fusion based on abstract sensors using spatial and temporal redundancy and dynamics
- Attack detection in the presence of transient faults
  - Transient fault model / modeling
  - Detection algorithms based on pairwise inconsistencies (PI)
- Case study with an unmanned ground vehicle
  - PI-based detectors outperform SF-based detector
  - PI-based detector with a bigger window size
    - pros: higher detection rate, lower false alarm rate, more robust
    - cons: slightly slower to detect



Extensions

- Can this be adapted to dynamically changing environment?
  - Adjusting parameters (ε, e, w) based on context information; e.g., vehicle speed
- Learning transient fault models (ε, e, w) at run-time

RESILIENT STATE ESTIMATOR

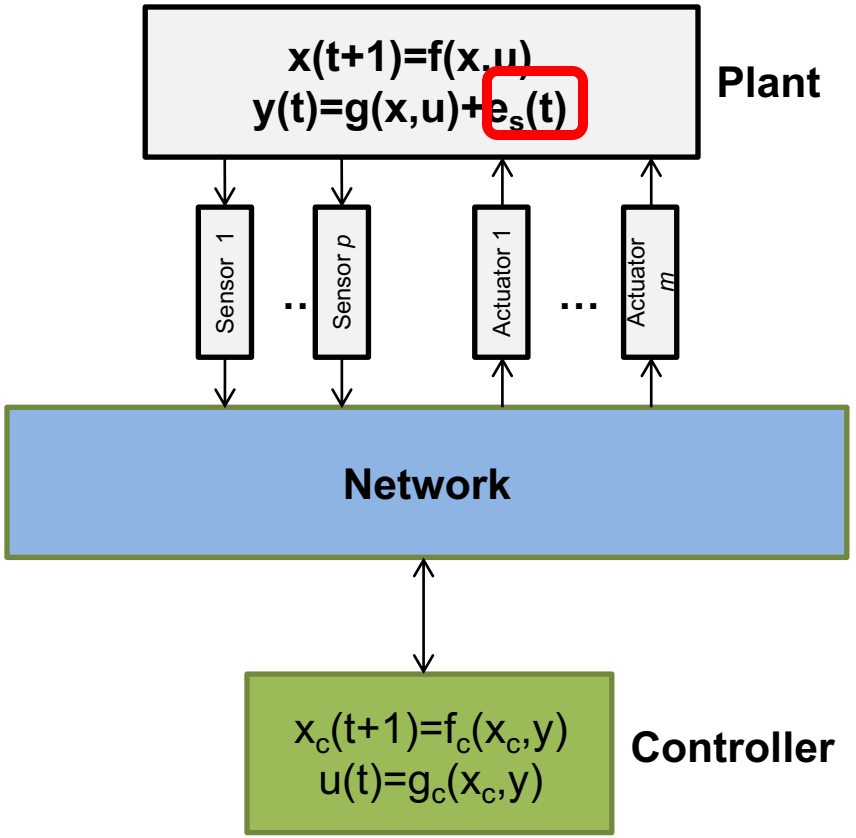
Attacks on Control Systems: Attack Space

1. Sensor attacks

- The attacker can arbitrarily change sensor measurements

$e_s^i(t)$ - the value injected by the attacker in sensor *i*

If no attack is injected $e_s^i(t)=0$



Sensor Attacks

Goal: Design a decoder $x_{t-N+1} = D_N(y_{t-N+1},\ldots,y_t,u_{t-N+1},\ldots,u_{t-1})$

Approach: Formulate the problem as an optimization problem

$$\min_x = \left\| \tilde{Y}_N - \Phi_N x \right\|_{l_0}$$

$$\tilde{Y}_N = \left[y_{t-N+1} \middle| y_{t-N+2} - CBu_{t-N+1} \middle| \ldots y_t - \sum_{i=0}^{N-2} CA^i Bu_{t-1-i} \right]$$

Sensor and actuator history

$$\Phi_N x = \left[Cx \middle| CAx \middle| \ldots CA^{N-1}x \right]$$

System dynamics

$$\min_x = \left\| \tilde{Y}_N - \Phi_N x \right\|_{l_1/l_r}$$

The history of attacks e(t-1), e(t-2)…

$$\tilde{Y}_N - \Phi_N x^* = \left[e_{t-N+1} + v_{t-N+1} \middle| \ldots e_{t-1} + v_{t-1} \right]$$

Identify attacked sensors for low-level measurement noise

Intrusion Detection for Sensor and Actuator Attacks

- ***Intrusion Detection*** **for** ***actuator attacks*** can be handled in a similar manner!

How many attacked sensors and actuators that can be tolerated?

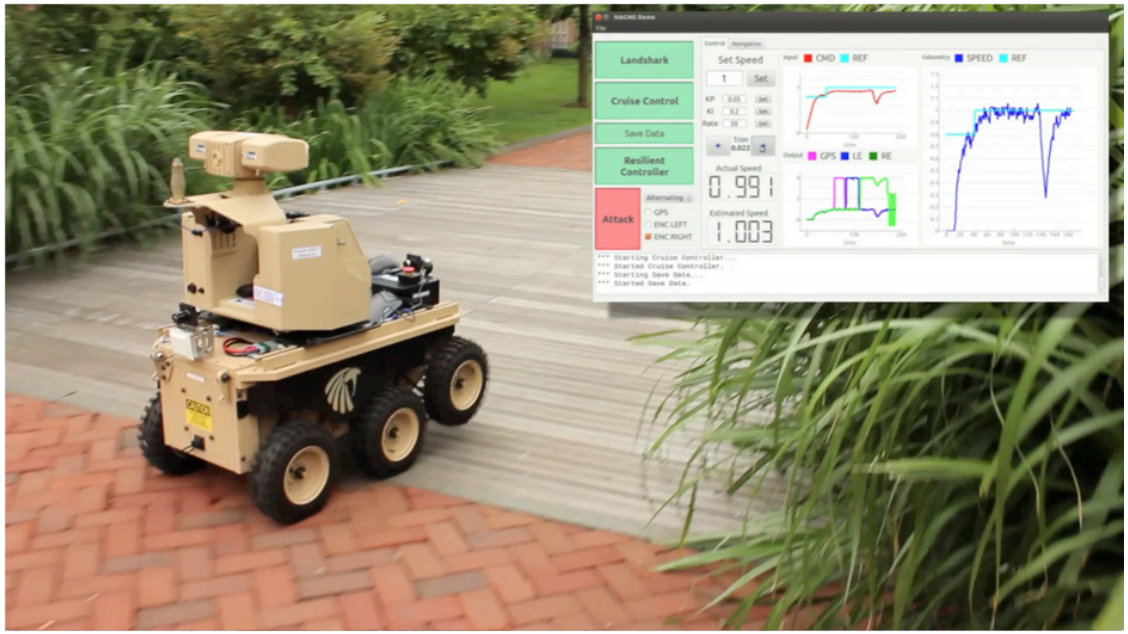
- p – number of sensors, q – number of attacked sensors and actuators
- In the best case, we can deal with $\lceil p/2 - 1 \rceil$ attacked sensors

$$q \geq \frac{p}{2} \quad \Rightarrow \quad \textbf{impossible to detect an attack!}$$

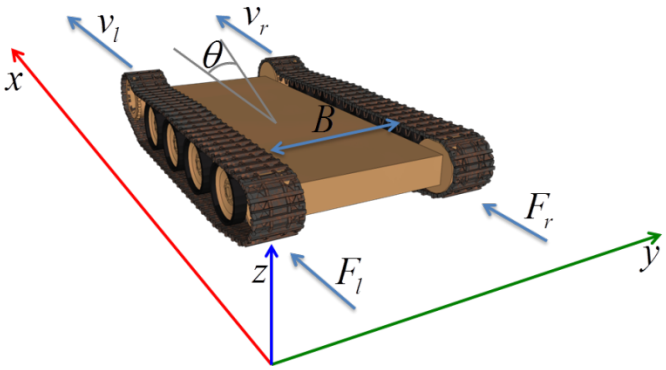
$$q < \frac{p}{2} \quad \Rightarrow \quad \textbf{detection depends on the system dynamics (i.e., matrices A, B and C)}$$

Case Study

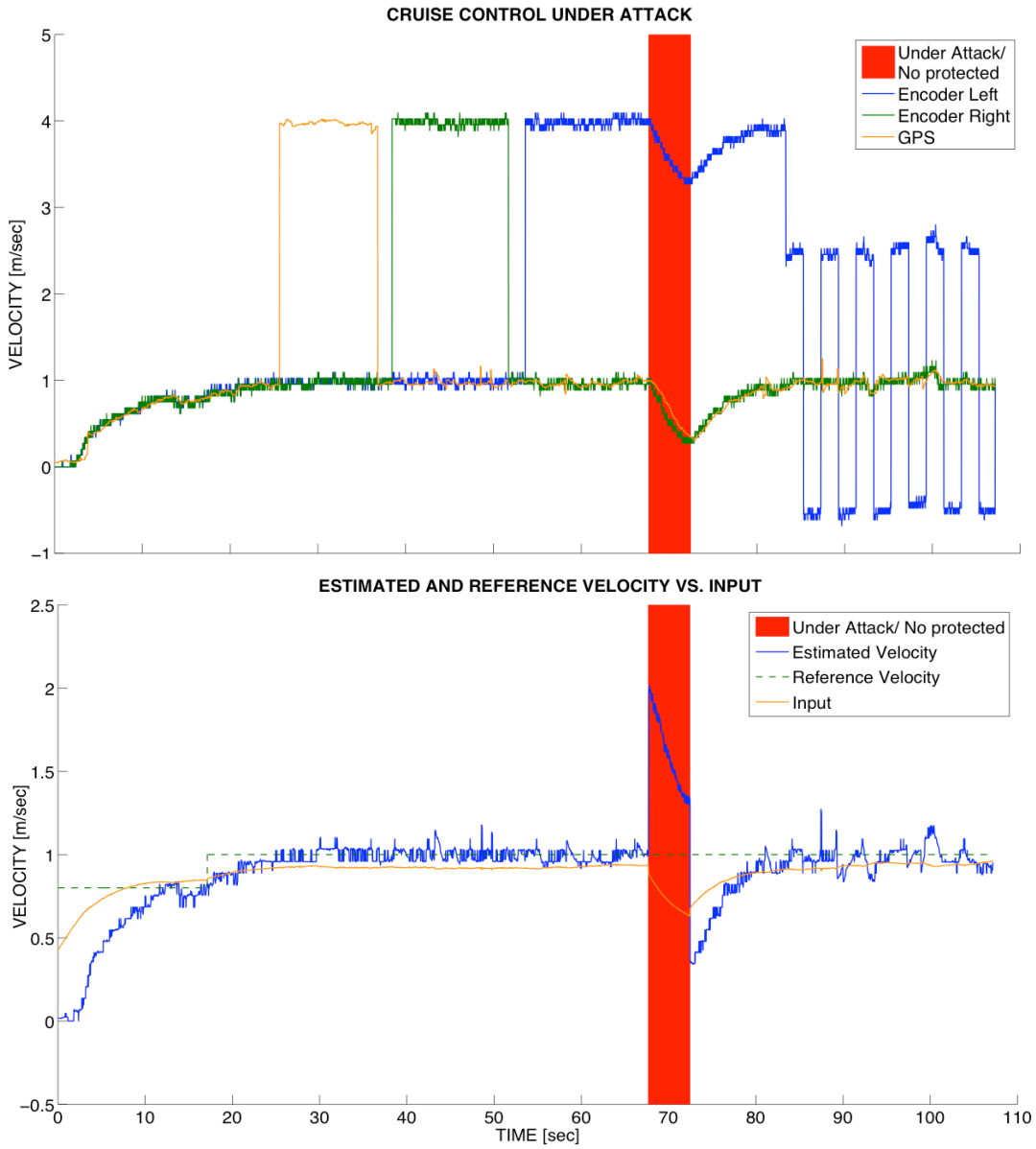
- Constant-speed cruise control for LandShark
  - Ensure that the vehicle can maintain speed when some of the sensors are under attacked



7th order skid steering model



Attack-Resilient Cruise Control Demo



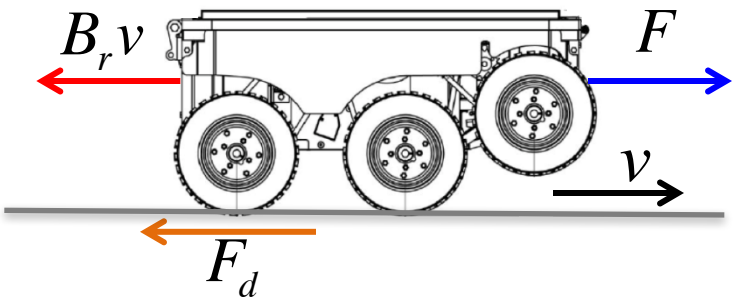
Penn Engineering

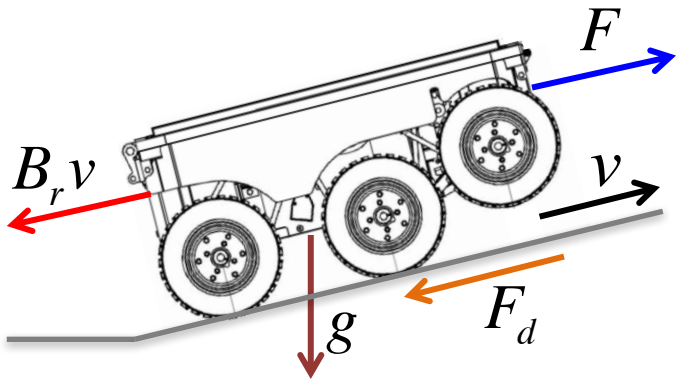
PRECISE

Attack-Resilient Cruise Control Demo

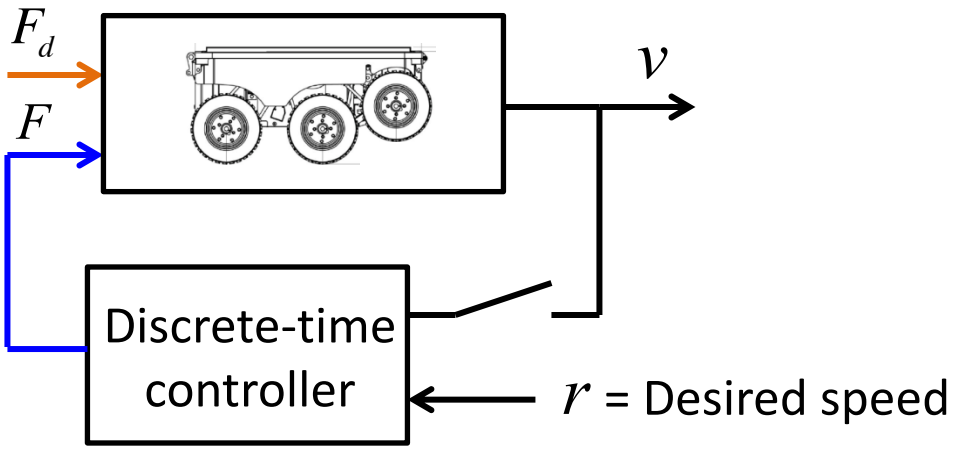
Challenge Problem – Cruise Control

- Case studies under analysis



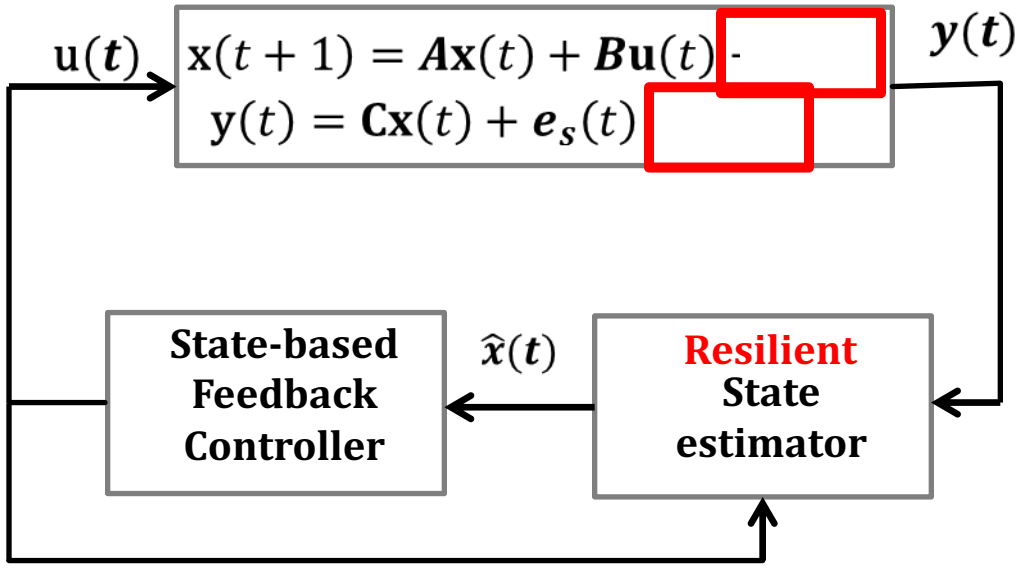


- Goal: Maintaining a driver set speed of the vehicle



Attack-resilient State Estimator

- State estimation from sensor measurement history
- Requires accurate model



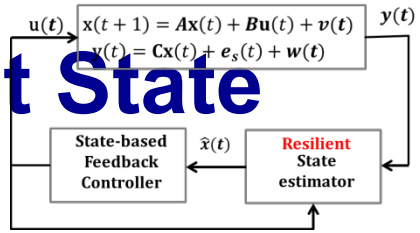
- In practice, we have process and measurement noise, and modeling errors (including jitter, latencies, etc)

Problems

- Can we still use the same detector?
- Can the attacker exploit the noise to destabilize the system?
- Can we bound the error of the state estimation?

M. Pajic, J. Weimer, N. Bezzo, P. Tabuada, O. Sokolsky, I. Lee, and G. J. Pappas, "Robustness of Attack-resilient State Estimators", ICCPS 2014. **(Best Paper Award)**

Robustness of the Attack-resilient State Estimator



$$P_0(\mathbf{Y}): \quad \min_{\mathbf{x}\in\mathbb{R}^n} ||\mathbf{E}||_{l_0}$$
$$\mathbf{Y} - \mathbf{\Phi x} = \mathbf{E}$$

$$(\mathbf{x}_0, \mathbf{E}) = \text{argmin } P_0(\mathbf{Y})$$

$$P_{0,\Delta}(\mathbf{Y}): \quad \min_{\mathbf{x}\in\mathbb{R}^n} ||\mathbf{E}||_{l_0}$$
$$-\mathbf{\Delta} \preccurlyeq \mathbf{Y} - \mathbf{\Phi x} - \mathbf{E} \preccurlyeq \mathbf{\Delta}$$

$$(\mathbf{x}_{0,\Delta}, \mathbf{E}) = \text{argmin } P_{0,\Delta}(\mathbf{Y})$$

If the state-feedback controller utilizes the state estimate for input control

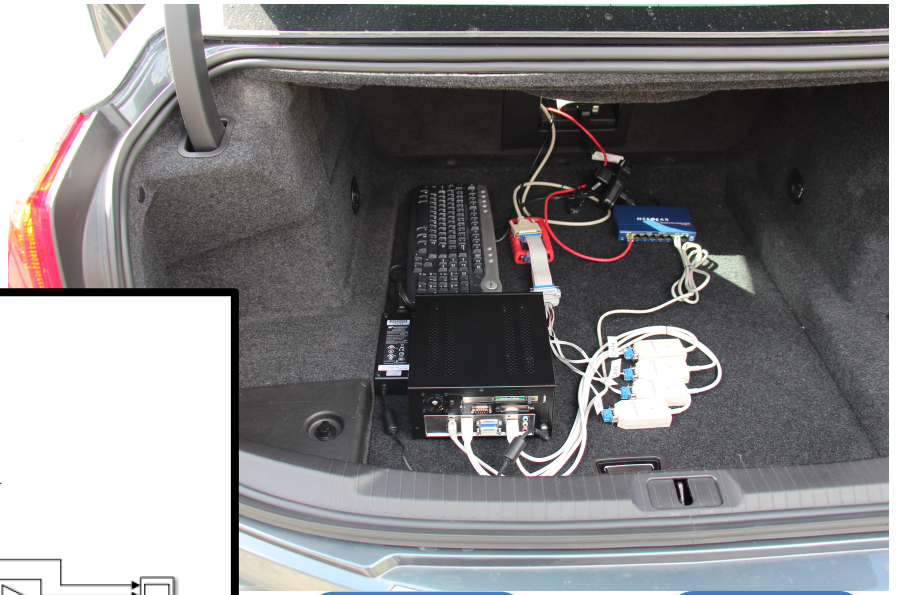
- Then the closed-loop system will remain stable when at most q_{max} sensors have been compromised.

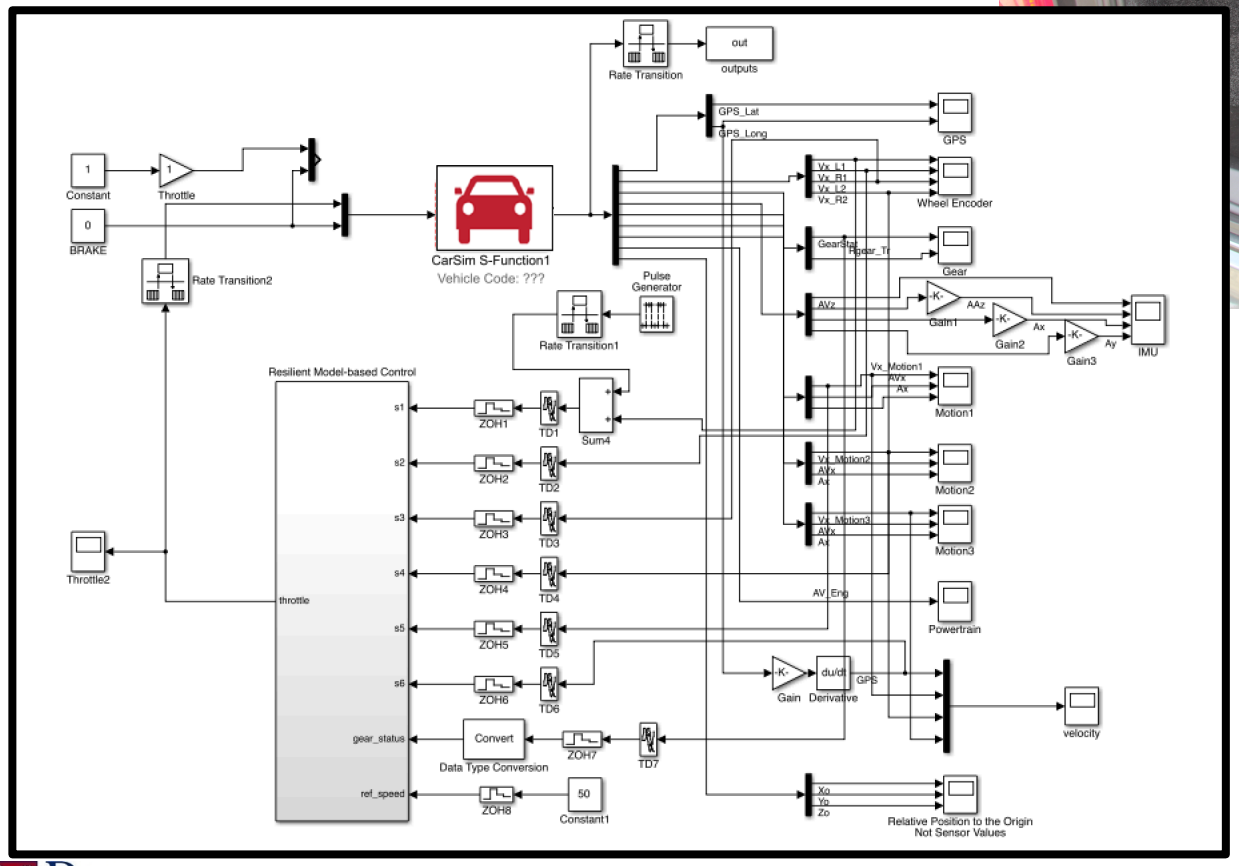
We have derived a design-time procedure to calculate an upper bound of the estimation error

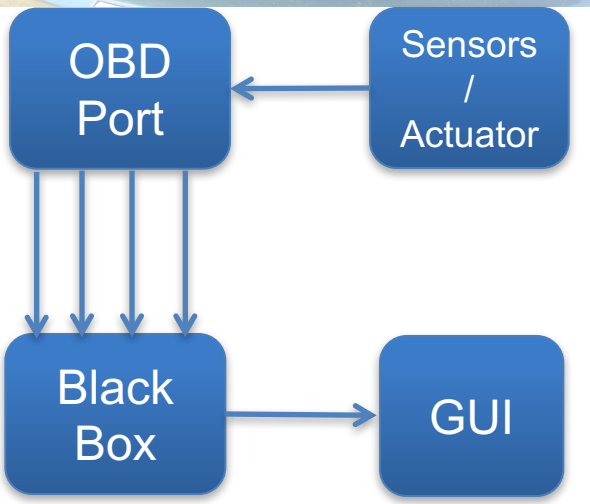
$$||\mathbf{x}_{0,\Delta} - \mathbf{x}_0||_2$$

Attack-resilient state estimator for American Built Car

- CarSim Simulation to obtain the model
- In-Car Implementation







Penn Engineering

PRECISE

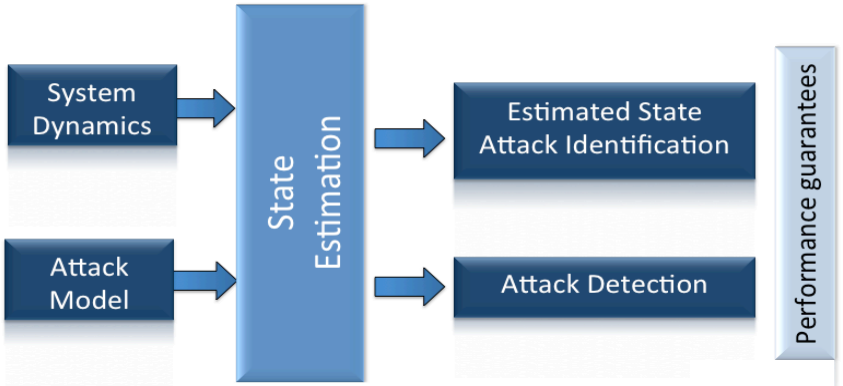
Attack-resilient state estimator for American Built Car

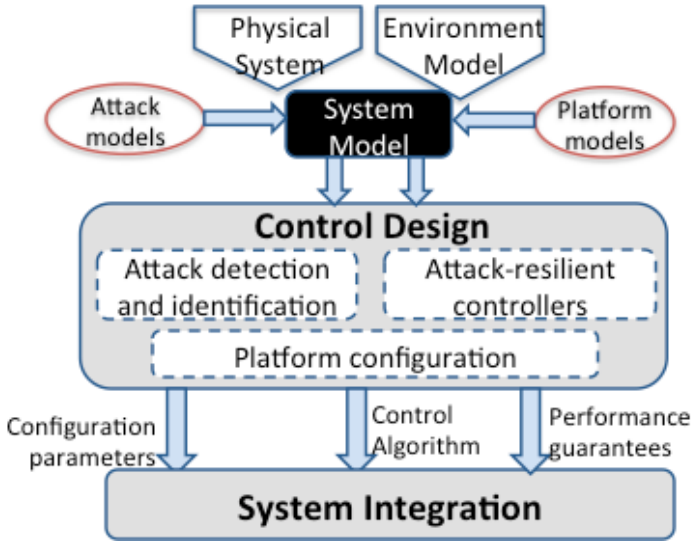
Penn Engineering

PRECISE
PENN RESEARCH IN EMBEDDED COMPUTING AND INTEGRATED SYSTEMS ENGINEERING

Extensions to Resilient State Estimators

- Challenges:
  - non-linear dynamics
  - realistic fault models
  - Impact of execution platform (computation & communication) on attack-resilient control
- Improve ADI when different types of sensors are used (continuous & discrete-events sensors) → sensor fusion with *context*
- Develop data-driven methods to handle non-linear dynamics and to derive better fault models
- Develop a framework for cross-layer analysis of platform effects on our resilient algorithms
  - Timing/scheduling effects
  - Resource constraints
  - Adaptation of attack-resilient control

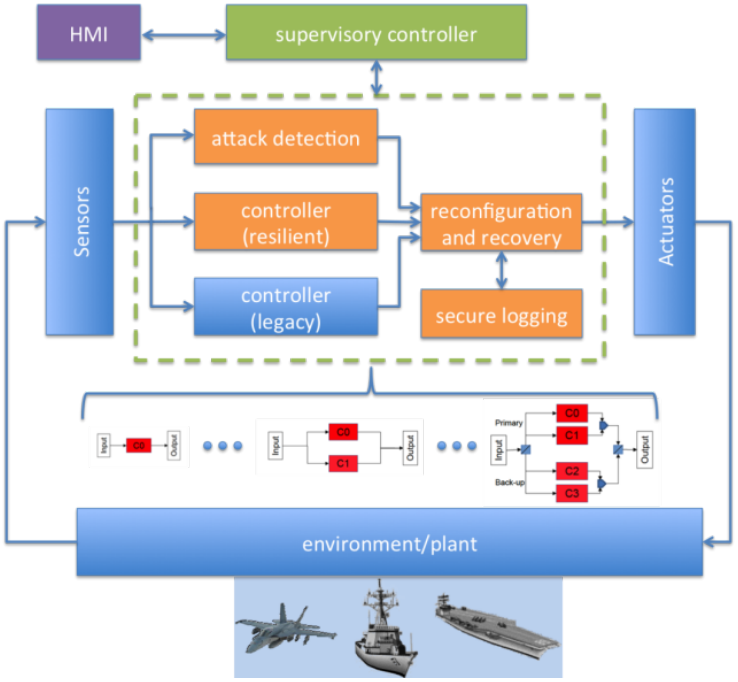




SECURITY-AWARE CPS ARCHITECTURE

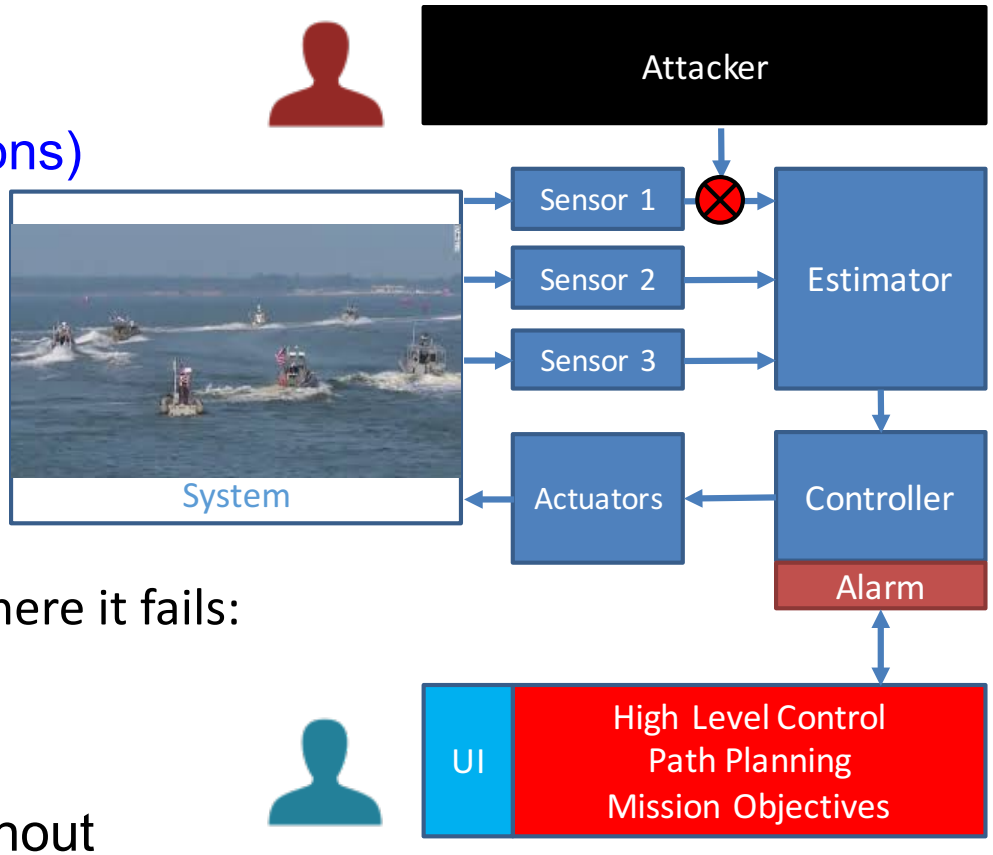
Attack-Resilient Architecture

- Goal: develop an architecture that
  - Leverages multiple security techniques to provide stronger guarantees
  - Enables adjusting the level of resilience to match the changing environment
  - Applicable to both legacy and clean-slate CPS
- Approach
  - Combine high-performance low-resilience techniques with high-resilience lower-performance techniques
  - Switch between techniques using attack detectors
  - Capture assumptions and guarantees of each technique to enable architecture-level analysis of system security
  - Human-on-the-loop, exploit the role of supervisors
- Challenge: how to balance
  - available systems resources
  - desired control performance
  - resiliency guarantees



Human-on-the Loop

- Autonomous CPS system
  - Read sensors
  - Process data (making decisions)
  - Command actuators
- Human-on-the-loop
  - Setting/Updating objectives
  - Intermittent monitoring
  - Active (vs. passive) monitoring
  - Complementing the autonomy where it fails: ***Uncertainty***
- Challenges
  - Increase system resiliency, without information overload
  - Ensuring system objectives are met



CPS Checkpointing and Recovery

- Detection algorithms and control architecture provide detection guarentees
  - time-to-detection
  - detection rate

- Cyber-Physical Checkpointing
  - Checkpoint generation: when/where/what to log
  - Use property of controller software and physical process to reduce amount of logged data
  - Secure logger: tamper-proof logging

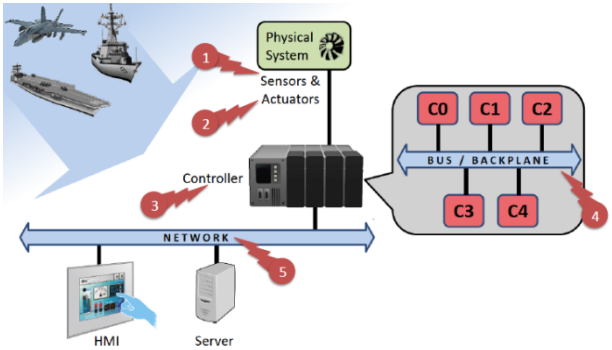
- Safe Recovery of Controllers
  - When attack is detected, the control system may need to be reset to a safe state with respect to the control physical process
  - Develop (formal) techniques to guarantee safety of recovery process

Checkpoint Generation Issues

- Single-loop control scenario
  - log subset inputs/outputs
  - exploit physical dynamics to reduce amount of logging necessary
- Distributed control scenario
  - conservative logging of all inputs and outputs is impractical
  - find minimal sets of data and when to store them
  - exploit concepts from distributed control system monitoring
- Logging always happens, but recovery is rare
  - balance tradeoff between recovery and logging costs/requirements
  - require secure logging capability

Safe Recovery of Controllers

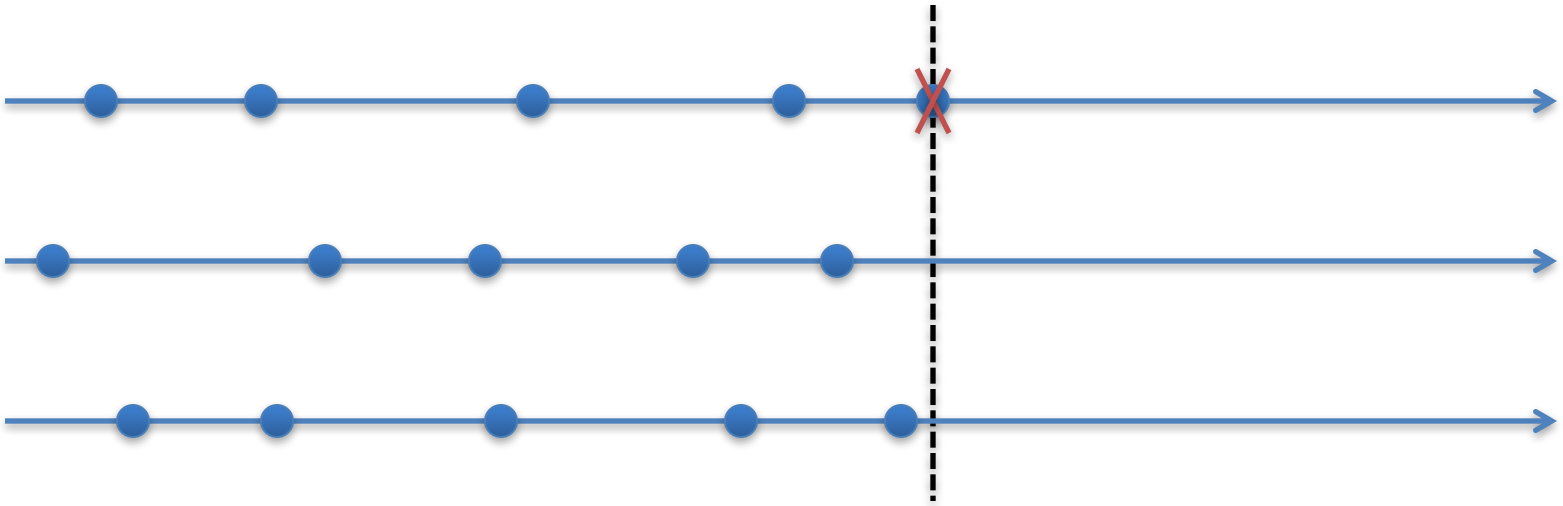
- Problem: After an attack is detected, how can we perform controller recovery while guaranteeing system safety
- Goal: How to ensure consistency between the control mode and state of the physical plant?
- Challenges
  - Safe recovery
    - Ensure system recovery to a correct state
    - Guarantee real-time recovery w/o loss of control functionality
      - Bounded recovery time



Some Problems in CPS Recovery

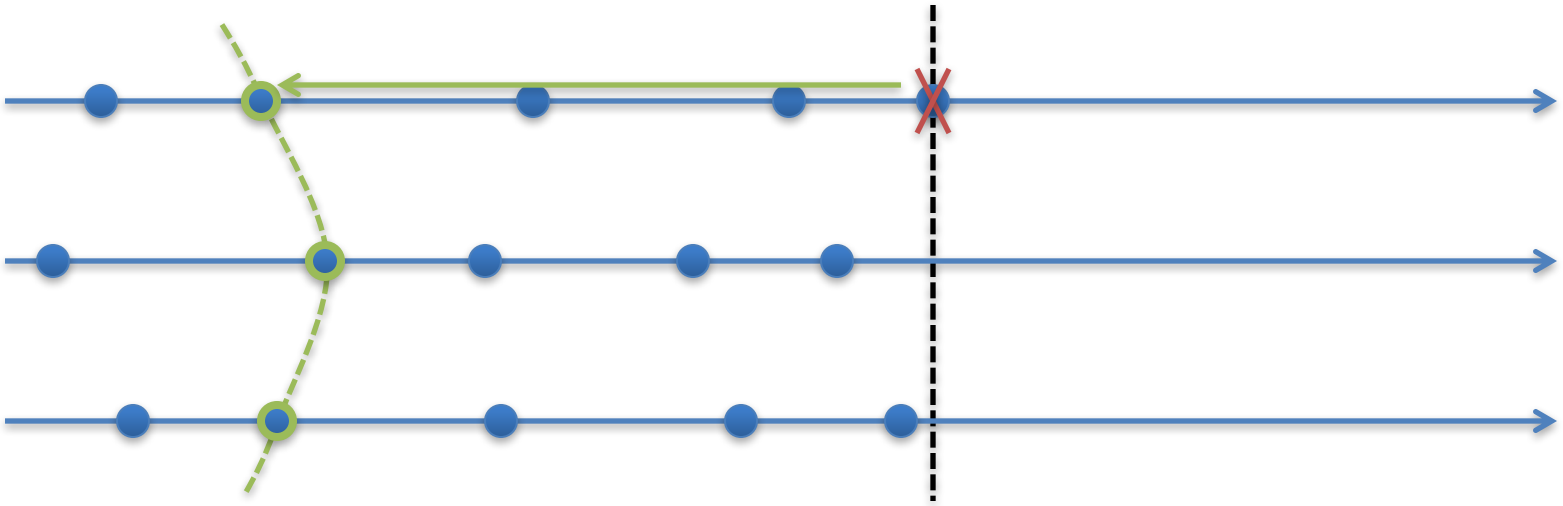
- Developing the right notion of consistent global state in CPS
- Determining when to roll-back and how far to roll-back
- Developing strategies for roll-forward

CPS Checkpointing



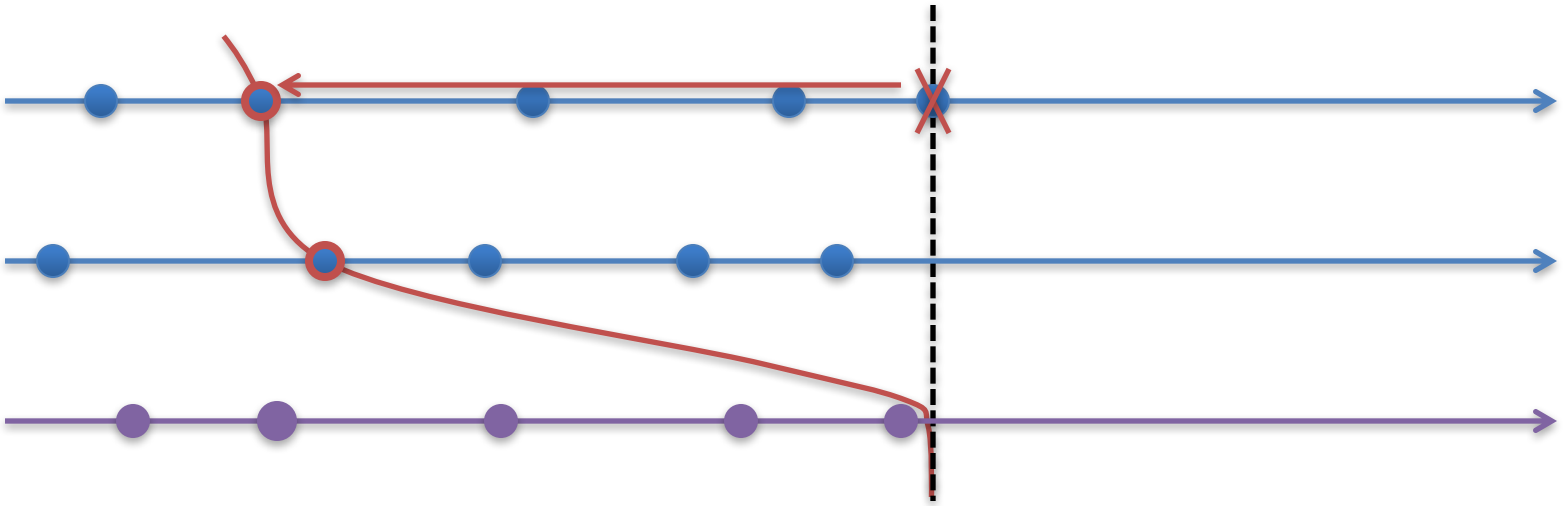
- A system with checkpointing discovers an error …

CPS Checkpointing



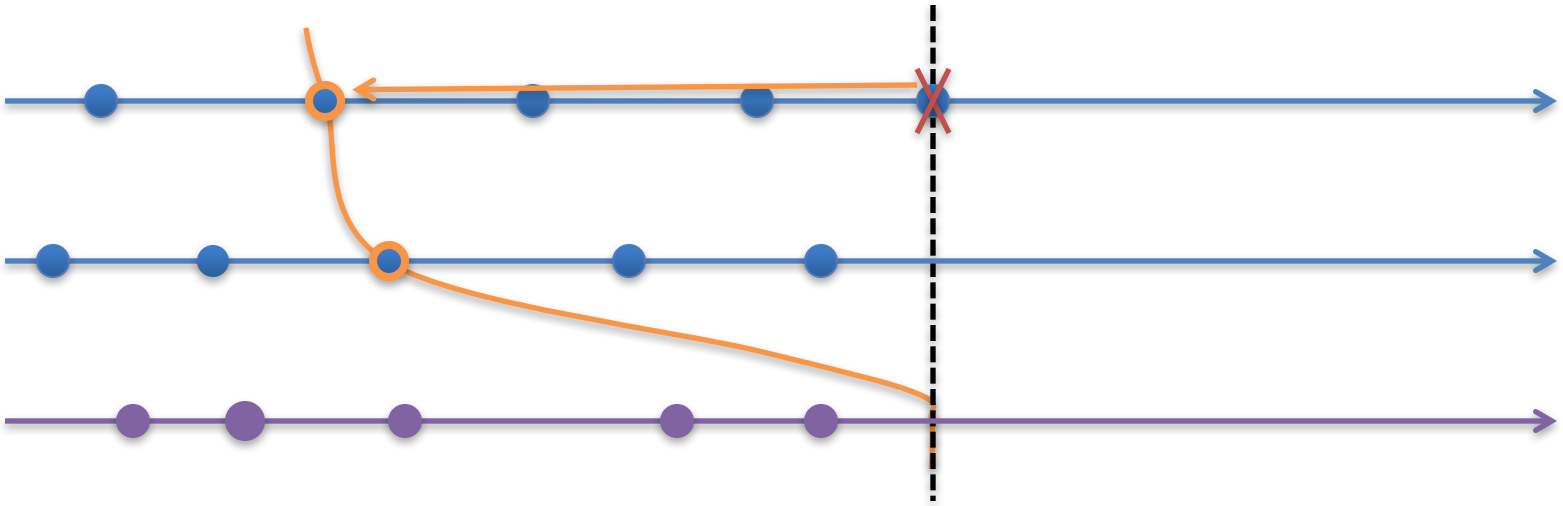
- Classical checkpointing rolls back the entire system to a logically consistent state

CPS Checkpointing



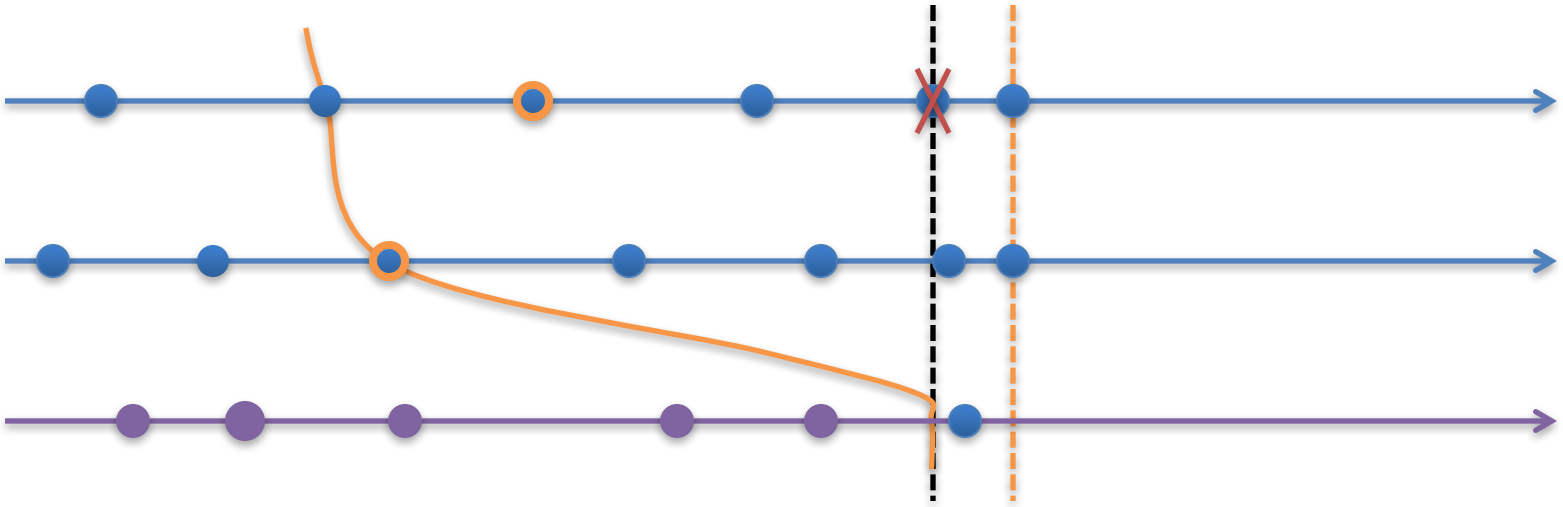
- In CPS, it may not be possible to roll back all states
  - e.g., physical states such as “position”
- Rolling back only a subset of the states may not be consistent/safe

CPS Checkpointing



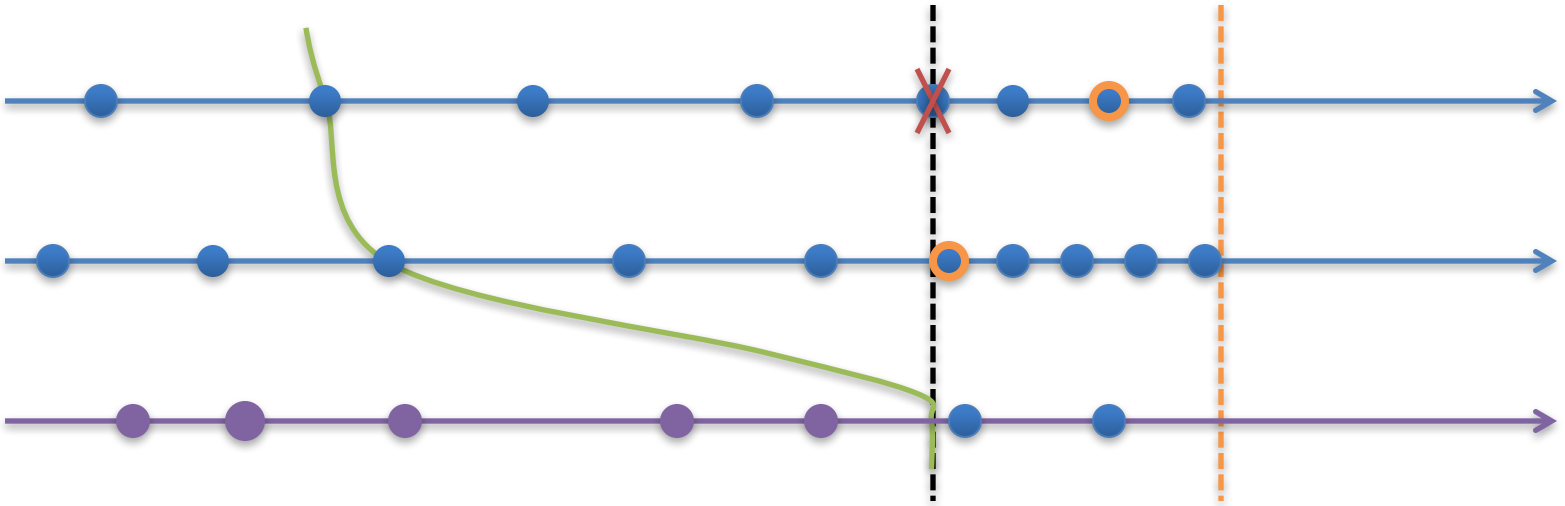
- In CPS, it may not be possible to roll back all states
  - e.g., physical states such as “position”
- Rolling back only a subset of the states may not be consistent/safe
- Challenge: how to checkpoint states that can be rolled back to ensure “safe rollback”
  - “safe” but may have reduced operational capabilities

CPS Recovery



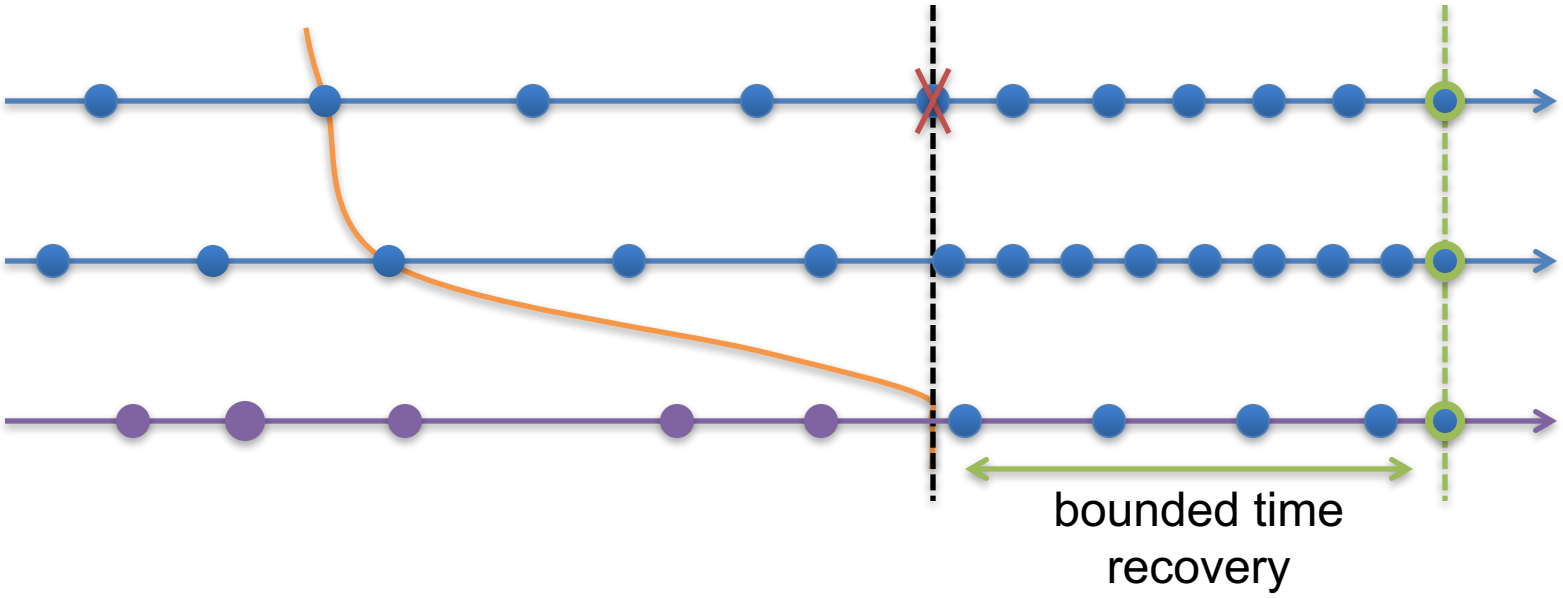
- Challenge: How to ensure recovery to a safe state
  - Guarantee real-time recovery to ensure robust system operation

CPS Recovery



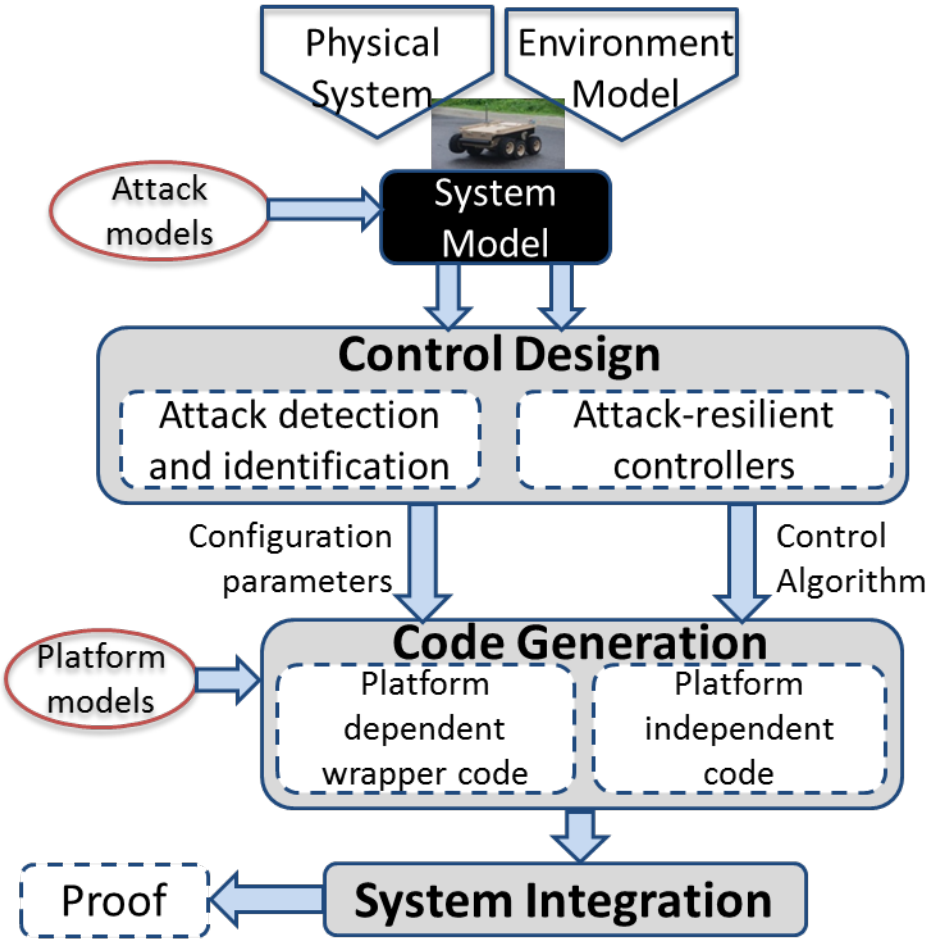
- Challenge: How to ensure recovery to a safe state
  - Guarantee real-time recovery to ensure robust system operation

CPS Recovery



- Challenge: How to ensure recovery to a safe state
  - Guarantee real-time recovery to ensure robust system operation

Platform-Aware CPS Design Framework



- Control-level techniques
  - Attack detection and identification using redundant sensing and model of the system's dynamics
  - Attack-resilient control architectures
- Code-level techniques
  - Ensure that the control code is correctly implemented and integrated
  - Preventing malicious code injection into the controller

Goal: Ensure that a CPS maintain a degree of control even when the system is under ***cyber*** and/or ***physical*** attack

Additional CPS Security Challenges

- Data-driven CPS
  - Attacks on training data
- How to retrofit legacy systems to be resilient to newly discovered attacks
- Human-in-the-loop CPS
- Privacy
- Assurance cases for security (and safety)
- Which solutions will be accepted by practitioners?
- Who/what is liable when such a system fails due to security and privacy attacks?

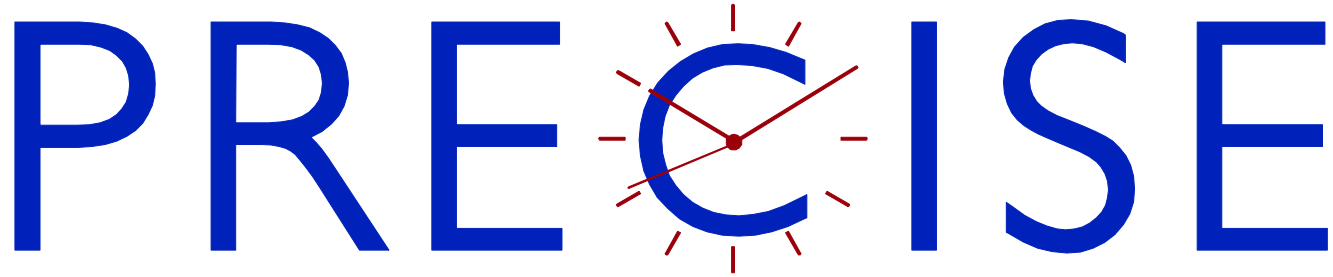
References

- S. Park, J. Weimer, I. Lee. Resilient Linear Classification: An Approach to Deal with Attacks on Training Data. ICCPS, 2017. (to appear)
- M. Jo, J. Park, Y. Baek, R. Ivanov, J. Weimer, S.H. Son and I. Lee, "Adaptive Transient Fault Model for Sensor Attack Detection", 4th International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA), 2016 (Best Paper Award).
- R. Ivanov, N. Atanasov, M. Pajic, G. Pappas and I. Lee, "Robust Estimation Using Context-Aware Filtering", 53rd Annual Allerton Conference on Communication, Control, and Computing, 2015.
- J. Park, R. Ivanov, J. Weimer, M. Pajic, and I. Lee, Sensor attack detection in the presence of transient faults, ICCPS, 2015.
- N. Bezzo, J. Weimer, M. Pajic, Y. Shoukry, P. Tabuada, O. Sokolsky and I. Lee, *Attack Resilience State Estimation for Autonomous Robotic Systems,* iROS, 2014
- M. Pajic, J. Weimer, N. Bezzo, P. Tabuada, O. Sokolsky, I. Lee and G.J. Pappas, *Robustness of Attack-resilient State Estimators*, ICCPS, 2014 (Best Paper Award)
- Radoslav Ivanov, Miroslav Pajic and Insup Lee, Resilient Multidimensional Sensor Fusion Using Measurement History. HiCoNS, 2014
- R. Ivanov, M. Pajic, and I. Lee, *Attack-Resilient Sensor Fusion,* DATE 2014
- J. Weimer, N. Bezzo, M. Pajic, O. Sokolsky, and I. Lee, *Attack-Resilient Minimum-Variance Estimation*, ACC, 2014
- M. Pajic, N. Bezzo, J. Weimer, O. Sokolsky, R. Alur, R. Mangharam, N. Michael, G. J. Pappas, P. Tabuada, S. Weirich and I. Lee, *Towards Synthesis of Platform-aware Attack-Resilient Control Systems*, HiCoNS 2013
- J. Weimer, N. Bezzo, M. Pajic, G. J. Pappas, O. Sokolsky, and I. Lee, *Resilient Parameter-Invariant Control with Application to Vehicle Cruise Control*, Control of Cyber-Physical Systems, LNCIS 2013

Acknowledgements

- Collaborators:
  - Rado Ivanov, George Pappas, Junkil Park, Oleg Sokolsky, Nicola Bezzo (UVA), Miroslave Pajic (Duke)
- Research supported in part by
  - DARPA FA8750-12-2-0247 (HACMS)
  - ONR N000141712012
  - NSF CNS-1505799 and the Intel-NSF Partnership for Cyber-Physical Systems Security and Privacy.

THANK YOU!



http://precise.seas.upenn.edu