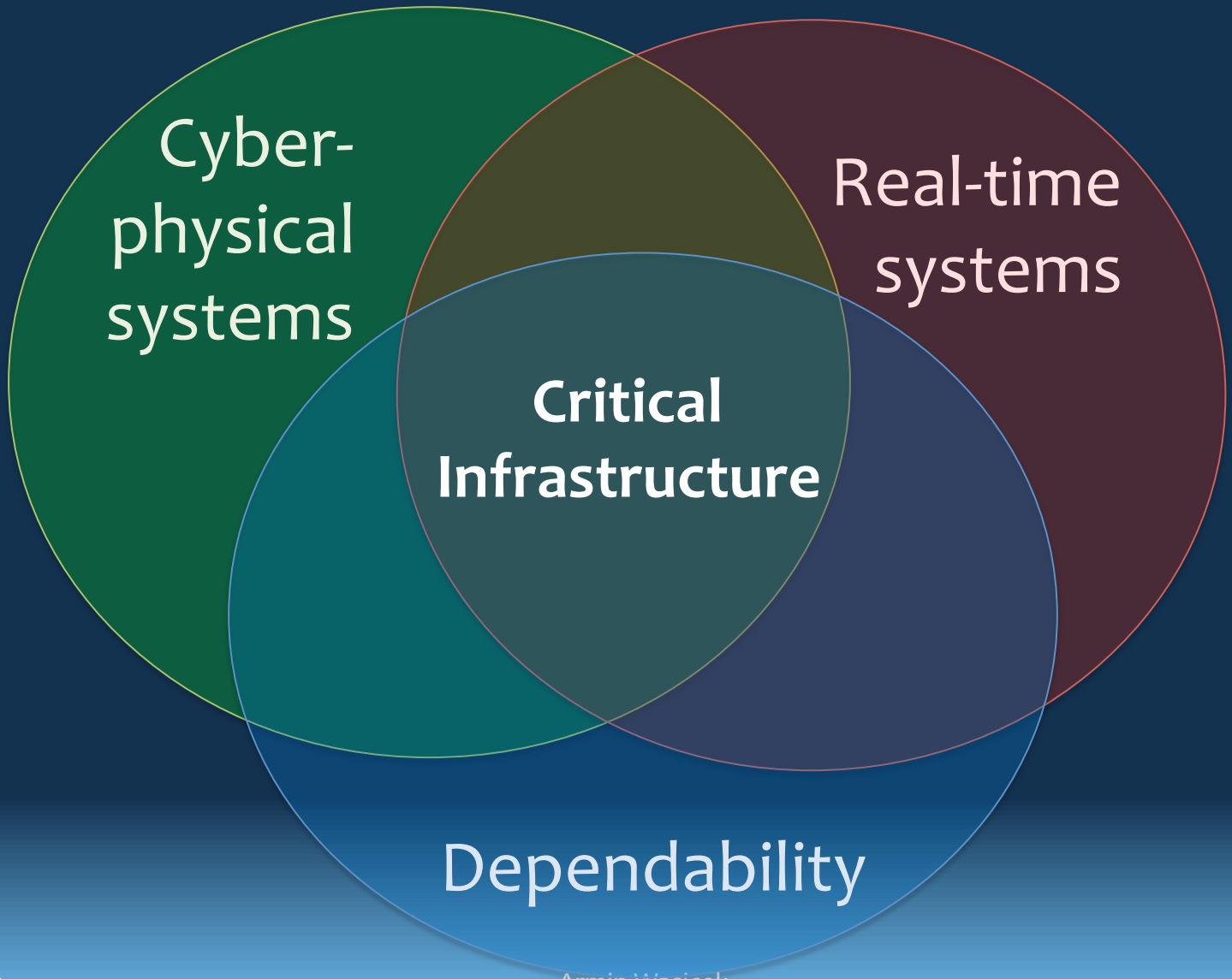
Attack Modeling in Ptolemy: Towards a Secure Design for Cyber-Physical Systems

Armin Wasicek
UC Berkeley

10th Biennial Ptolemy Miniconference

Critical Infrastructure Protection



Why Security Fails in CPS

- **Security is no or a minimal concern in CPS**
  - **Systems operated in isolation**
  - **No trained engineers**
- Common sources of vulnerabilities:
  - No appropriate security mechanisms
  - Misconfiguration of security mechanisms
  - Security is implemented as ‘add-on’

► **Engineering approach to secure CPS**

Security Engineering

GOAL: Establish security properties in a system

- E.g., Top-down approach:
  - Threat / Attack Model → Understand
  - Security Policy → Design
  - Security Mechanism → Implement
  - Security Assessment → Review

► **Defines a structured way of working**

Model-based Design

Apply models to improve software engineering

- Provides a **common design environment**
- Enables early **location and correction of faults**
- Promotes **design reuse**

► **“No total, but reasonable automation“**

Attack Modeling

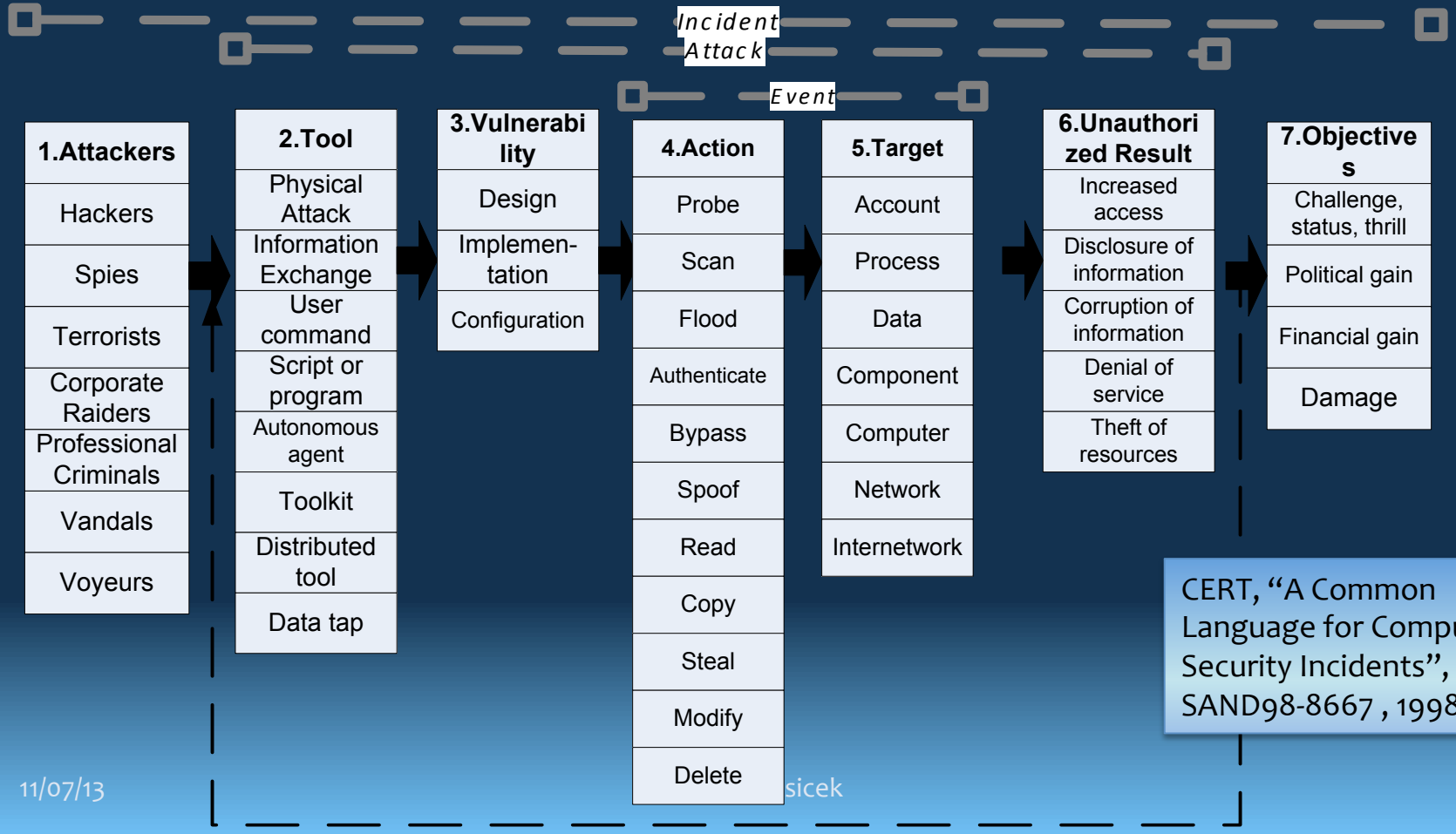
An adversary is usually described by its

- Capabilities
- Behaviors

- **Identifies weaknesses**
- **Enables search for mitigations**
- **Promotes understanding of attack vectors**

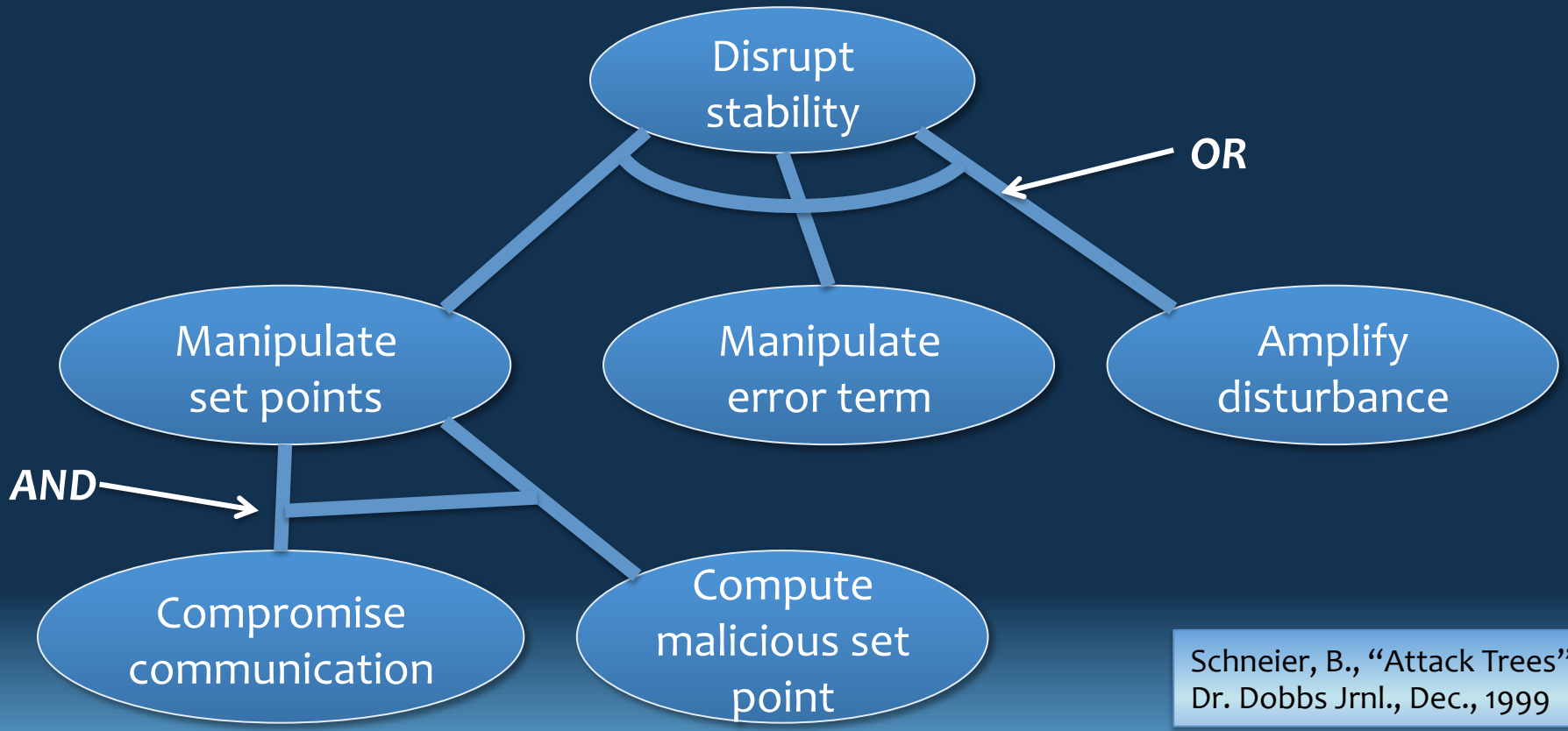
Textual Attack Models

- CERT Security Incident Taxonomy



Graph-based Attack Models

- Attack Trees describe paths to a target



Schneier, B., "Attack Trees", Dr. Dobbs Jrnl., Dec., 1999

Formal Attack Models

- Measurements as input for control system

$$y_i(k) \in \mathcal{Y}_i \ \ldots \text{sensor values } y_i \text{ at time } k$$

- Received measurements $\tilde{y}_i$, potentially tampered by a_i during attack interval $k \in \mathcal{K}_a$

$$\tilde{y}_i(k) = \begin{cases} y_i(k) & \text{for } k \notin \mathcal{K}_a, \\ a_i(k) & \text{for } k \in \mathcal{K}_a, a_i(k) \in \mathcal{Y}_i \end{cases}$$

Cardenas, A., et al., "Attacks against Process control systems", AsiaCCS, 2011

Attack Modeling in Ptolemy

Goals:

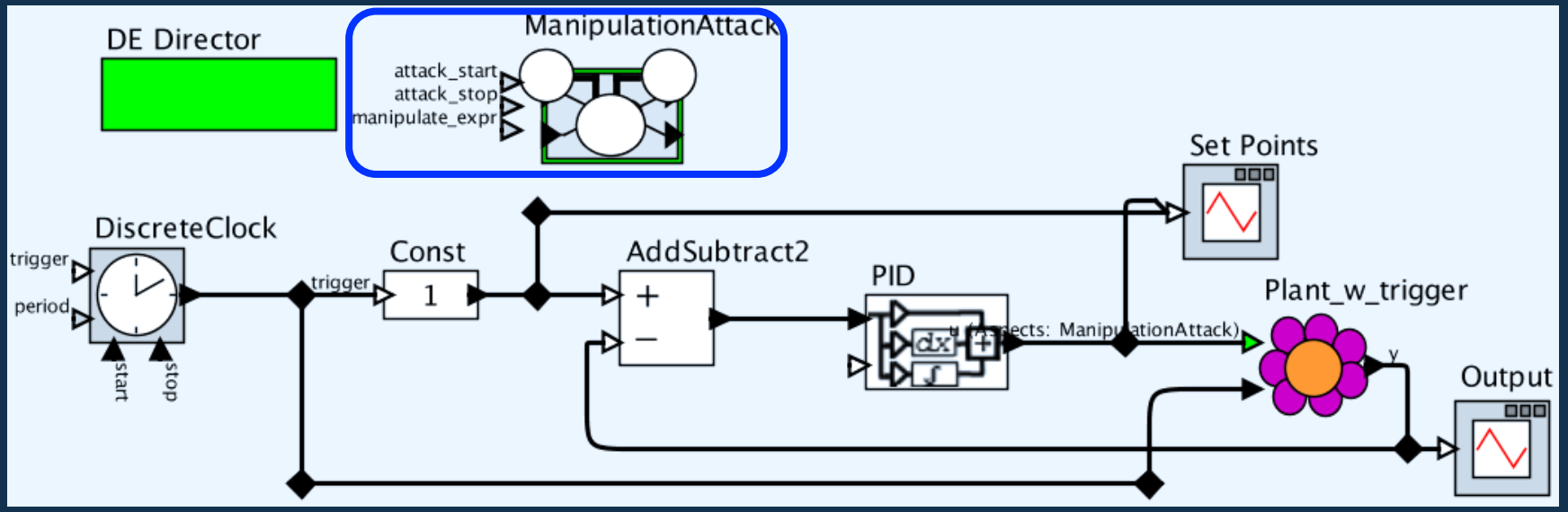
- Enable a design space exploration
- Help system designers to understand threats
- Educate

Means:

- Aspect-oriented modeling to reason about systems and attacks
- Enables separation of concerns

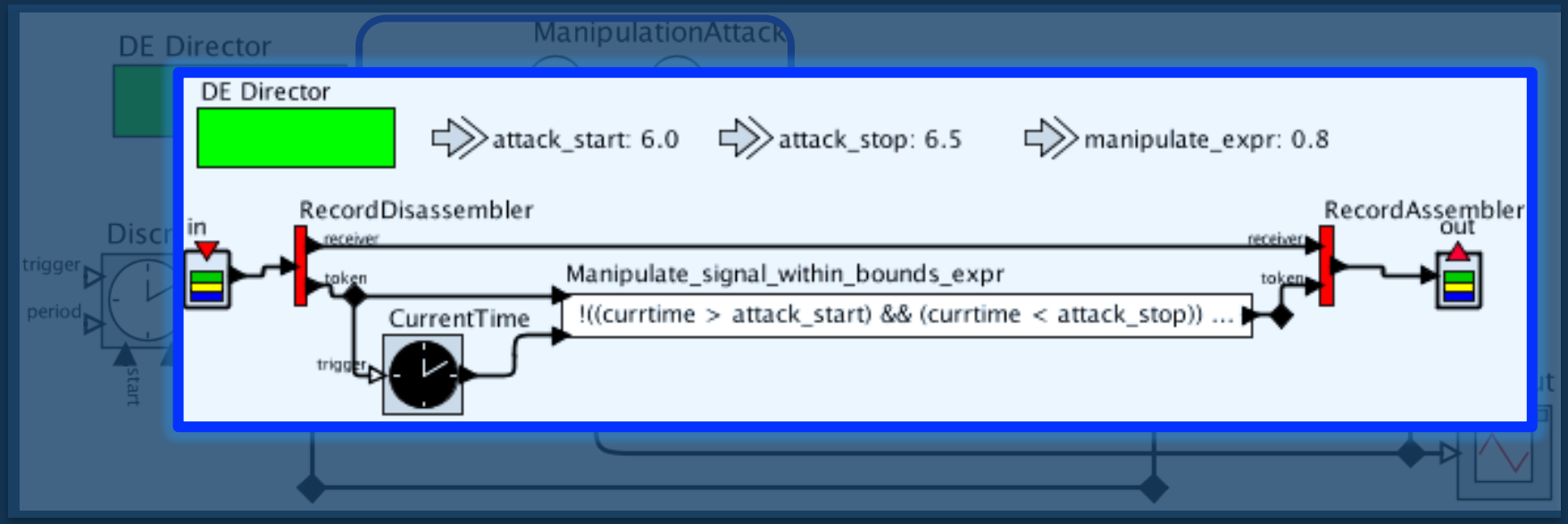
Case Study: Inverted Pendulum

Insert Attack Model via a CommunicationAspect



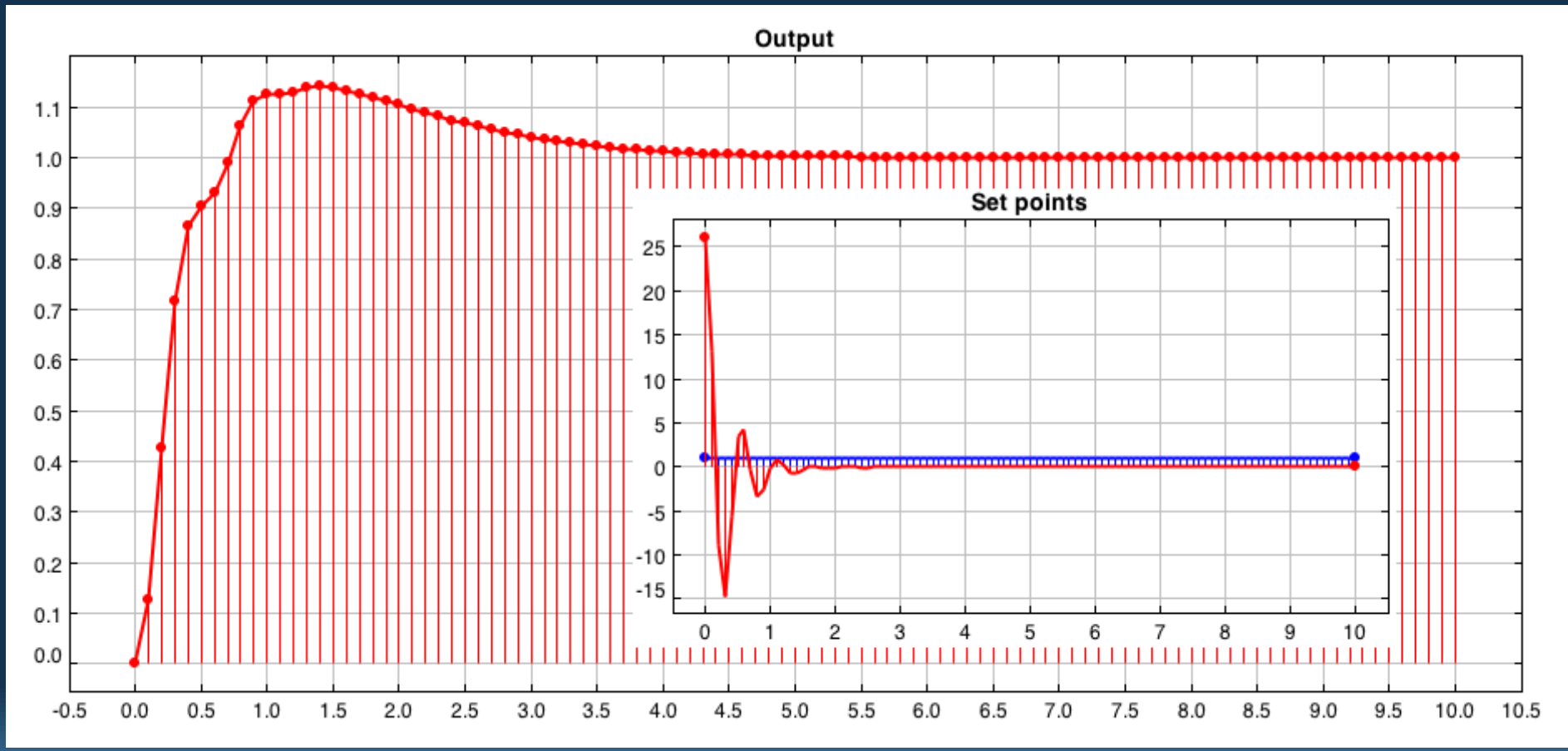
Case Study: Inverted Pendulum

Insert Attack Model via a CommunicationAspect

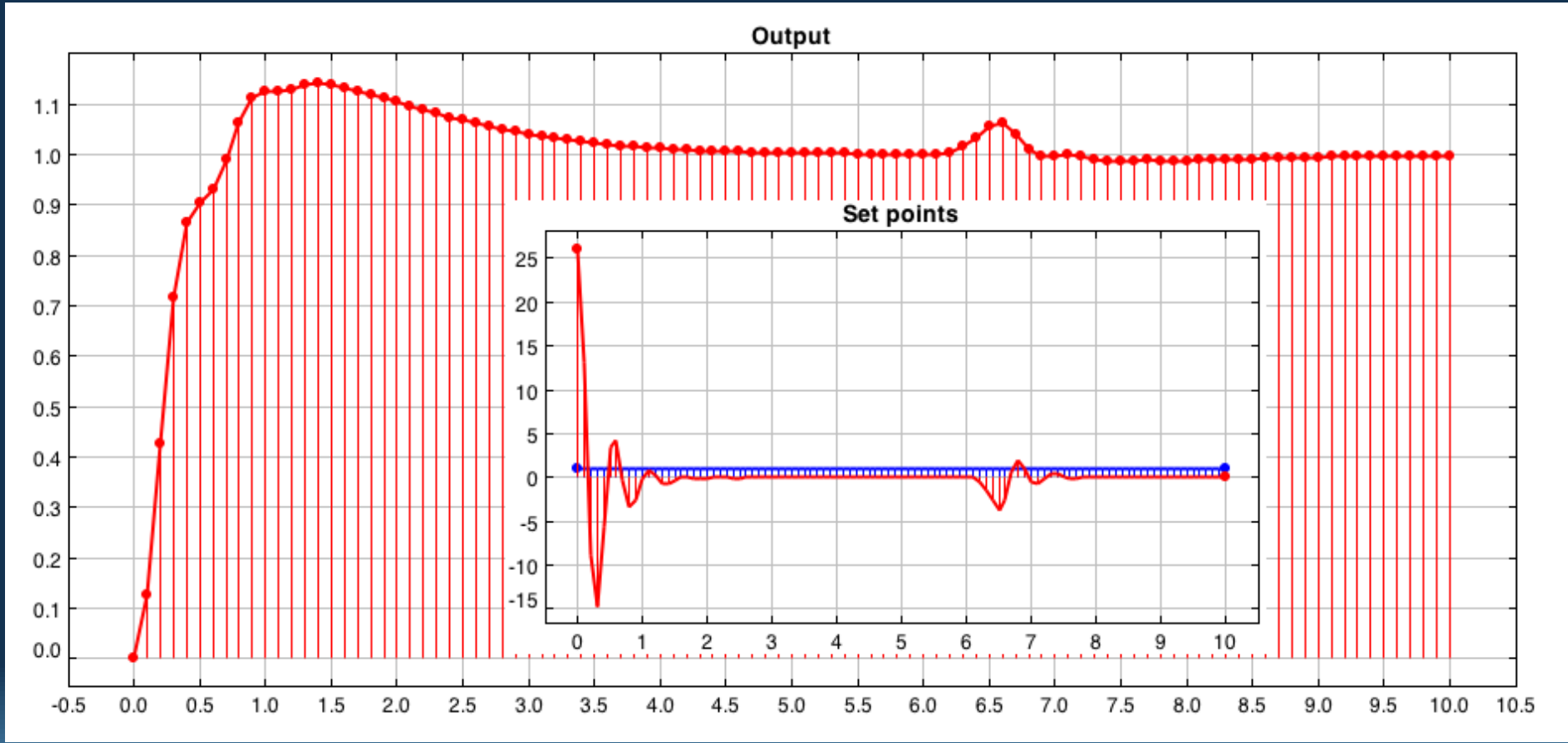


- **Heterogeneous MoC enables any attack model**

Plant under normal operation

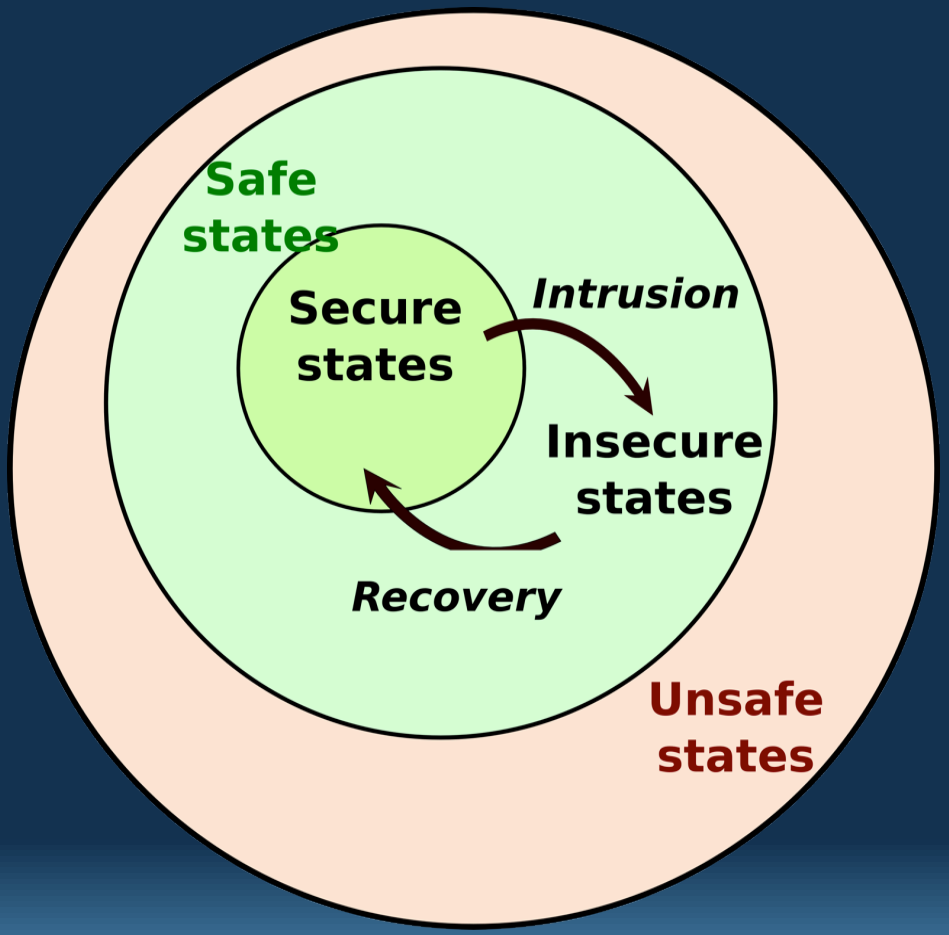


Plant under manipulation attack



Exploring the Design Space

- **Fault tolerance** mediates between safe and unsafe states in a system
  - Sharp boundary
- **Security is more subtle:**
  - Differentiation of secure and insecure is hard



Conclusion

- Aspect-oriented modeling used to **reason about systems and attacks**
- Weaving domain models enables **separation of concerns** between functionality and security
- **Heterogeneous MoC** enable the composition of virtually any attack with any system model


Thanks for your attention!