Detection is not enough: Low-Cost Attack Recovery for Robotic Vehicle Systems

Pritam Dash, Zitao Chen, Guanpeng Li, Mehdi Karimibiuki,

Karthik Pattabiraman

THE UNIVERSITY
OF BRITISH COLUMBIA

Autonomous Systems

Increasingly used in real-world safety-critical contexts

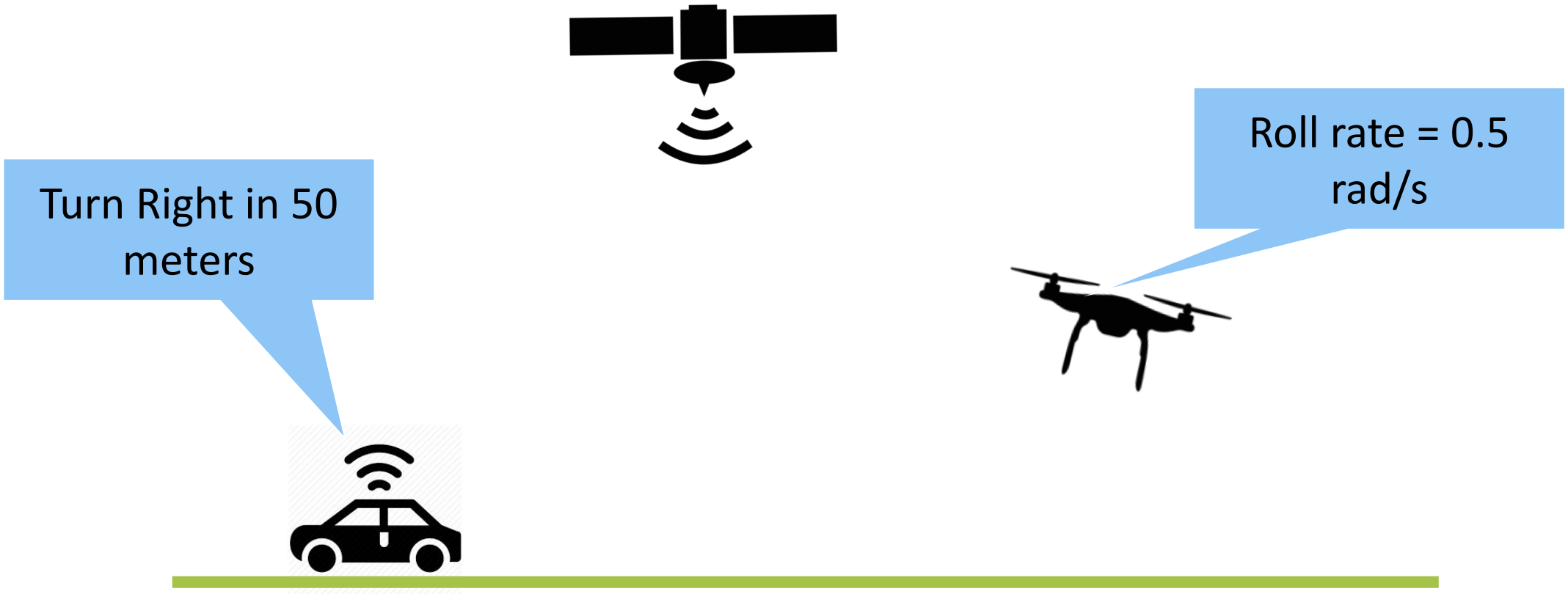
Autonomous Systems: Reliability and Security

Robotic Vehicles (RV): Motivation

Robotic Vehicles (RV) are becoming popular in many industrial sectors.

Safeguard RVs, Safe missions.

Perception in Robotic Vehicles (RV)



Sensor Attacks Against Robotic Vehicles (RV)

GPS Spoofing.
Transmit malicious GPS Signals

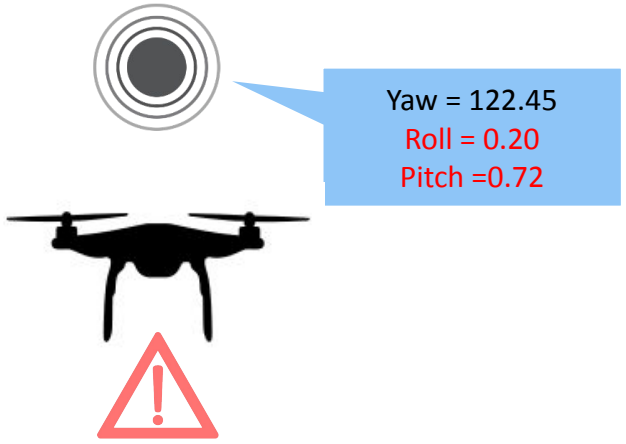
Actual Position

Spoofed Position

Tippenhauer et. al. On the requirements for successful GPS spoofing attacks. CCS'11

Sensor Attacks Against Robotic Vehicles (RV)

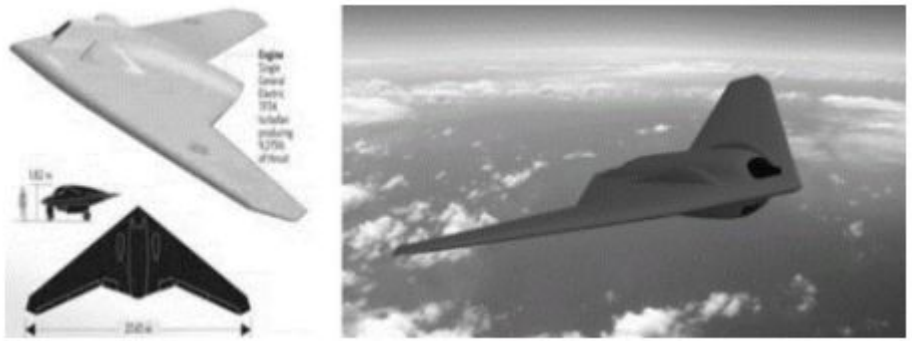
Signal Injection.
Optical, Magnetic or Acoustic noise



Son et. al. Rocking Drones with Intentional Sound Noise on Gyroscopic Sensors. Usenix Security'2015

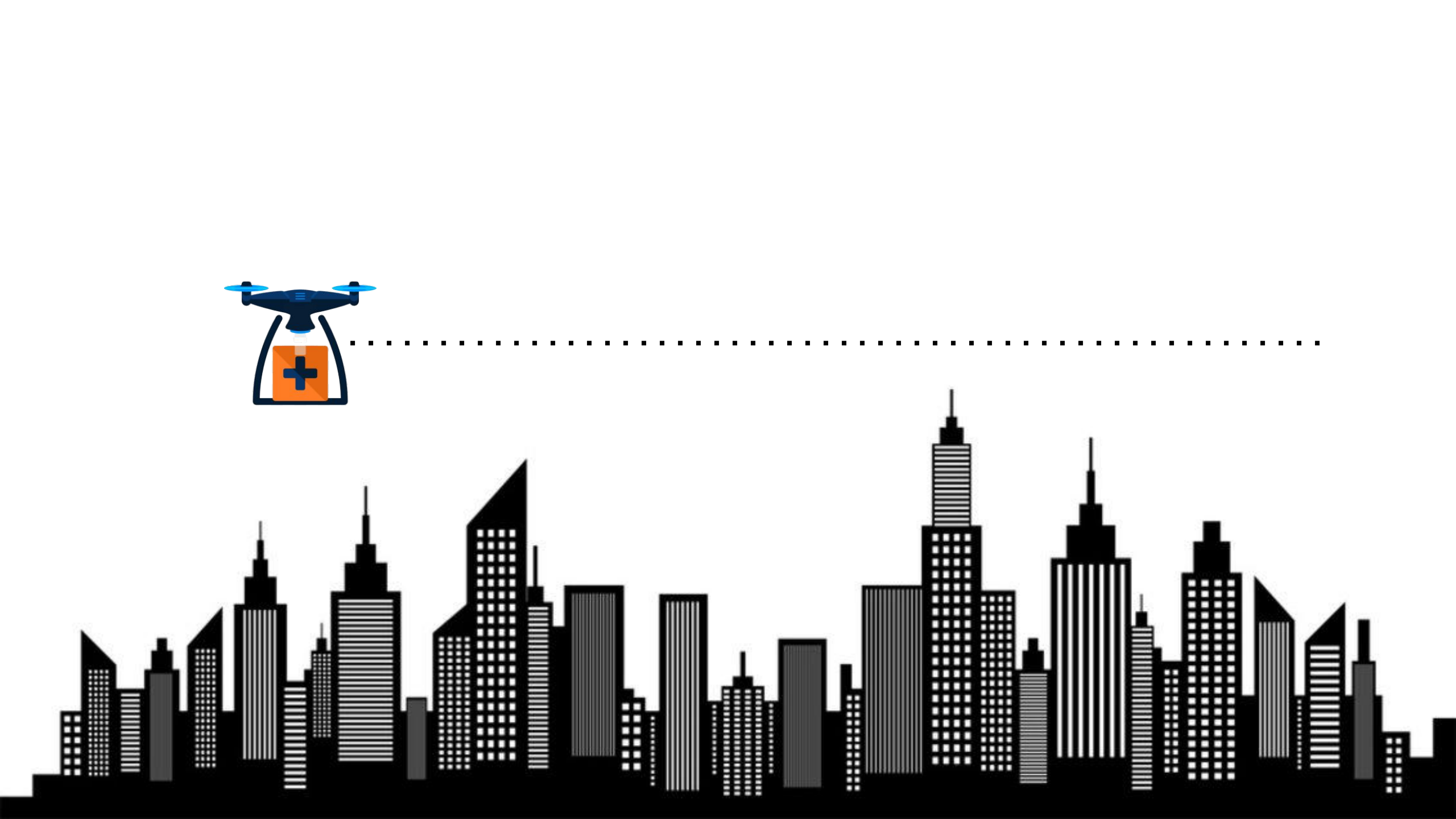
Sensor Attacks and Consequences

Iran–U.S. RQ-170 incident



ACM TECHNEWS

GPS Cyberattack Falsely Placed U.K. Warship Near Russian Naval Base



Prior work

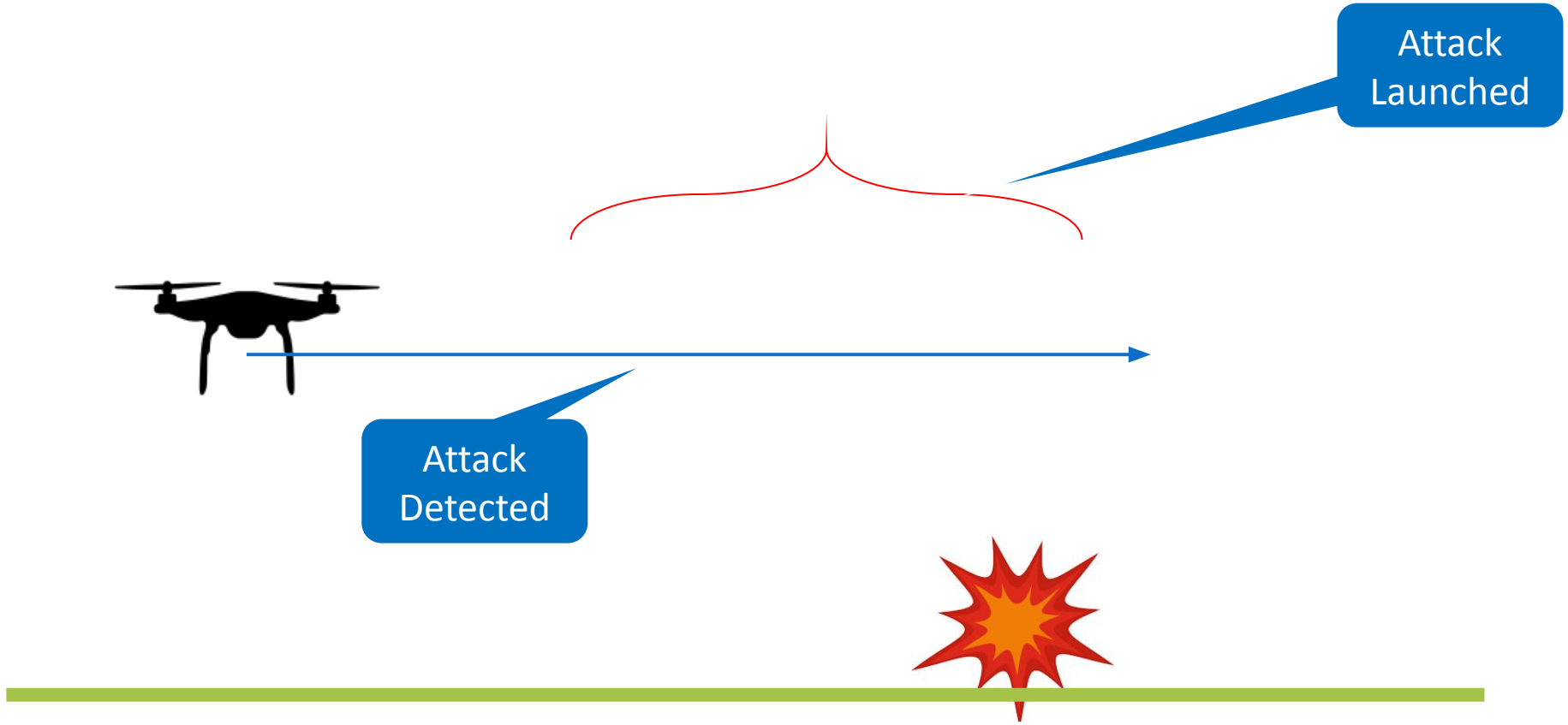
Invariant Based Detection

Model based Detection

"Very Effective in Detecting Attacks"

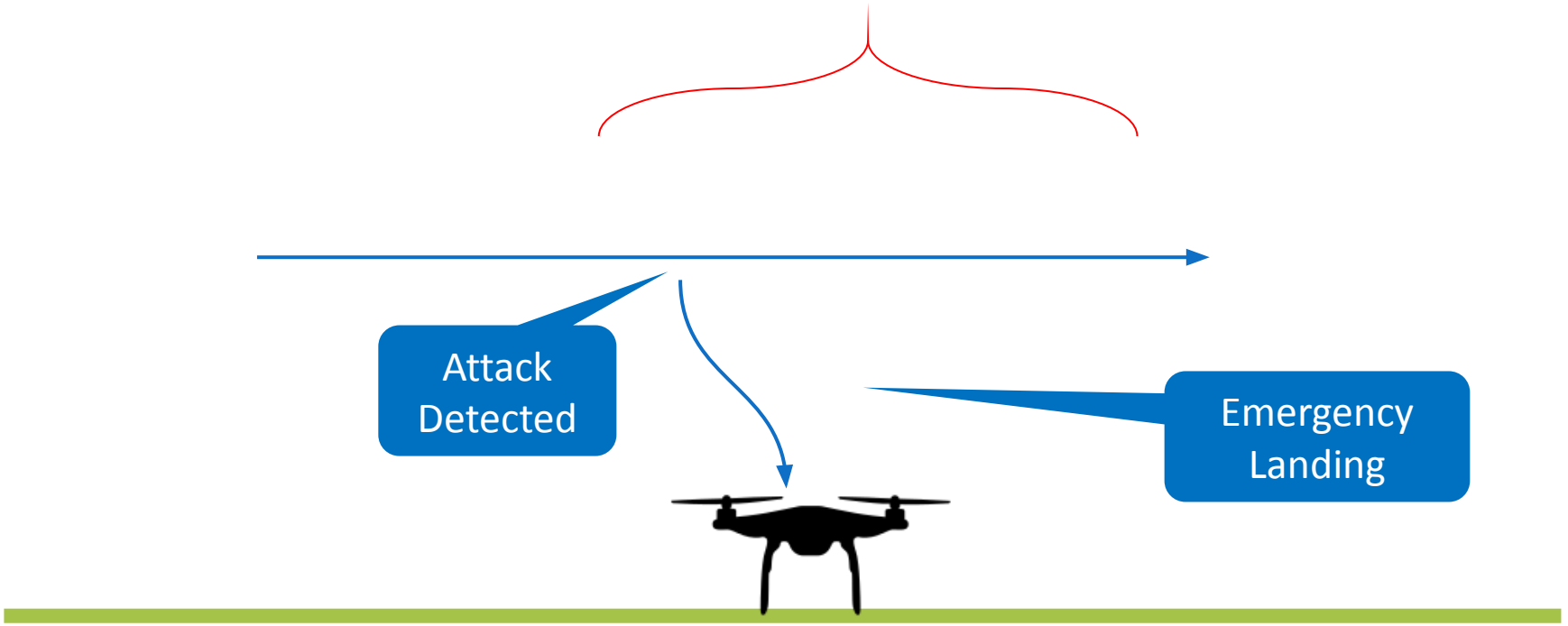
Choi et. al., Detecting Attacks against Robotic Vehicles: a Control Invariant Approach, CCS'18
Quinonez et. al., SAVIOR: Securing Autonomous Vehicles with Robust Physical Invariants, Usenix Security'20

Detection is not Enough …

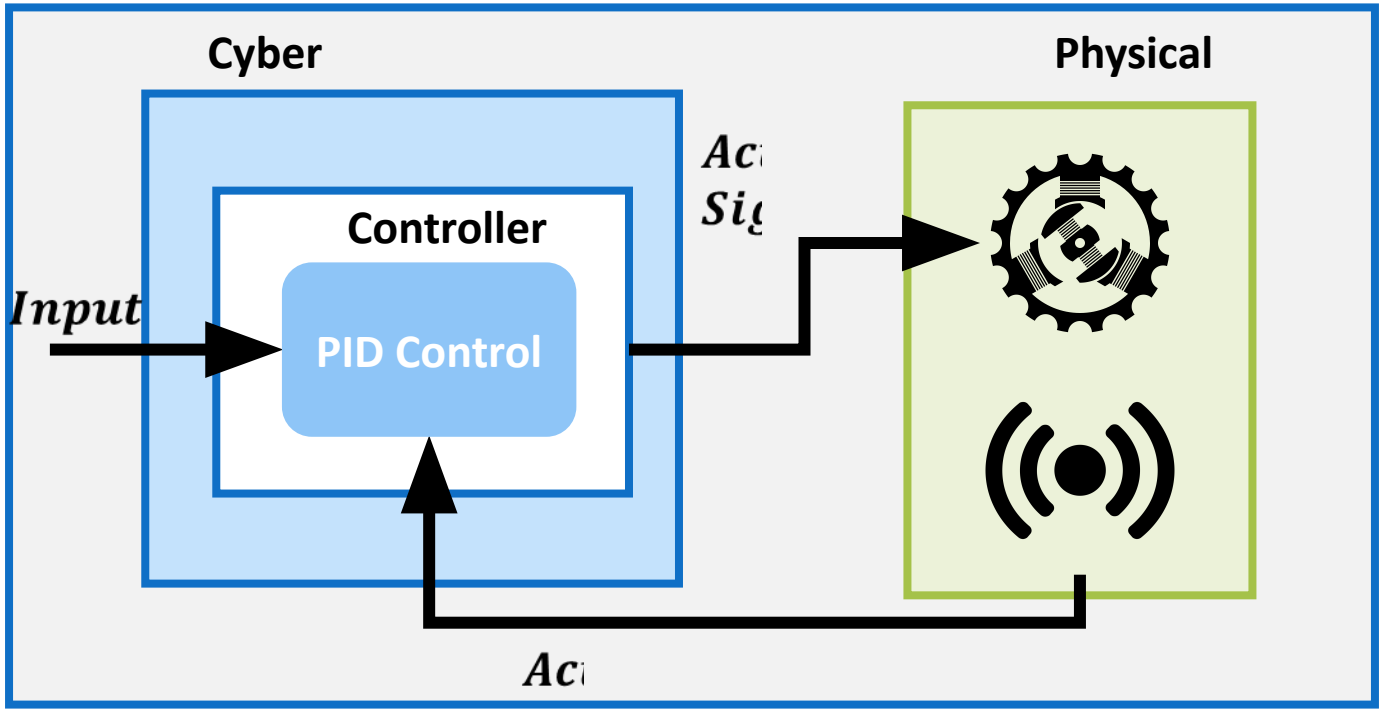


Choi et. al., Detecting Attacks against Robotic Vehicles: a Control Invariant Approach, CCS'18
Quinonez et. al., SAVIOR: Securing Autonomous Vehicles with Robust Physical Invariants, Usenix Security'20

Failsafe is not enough either…

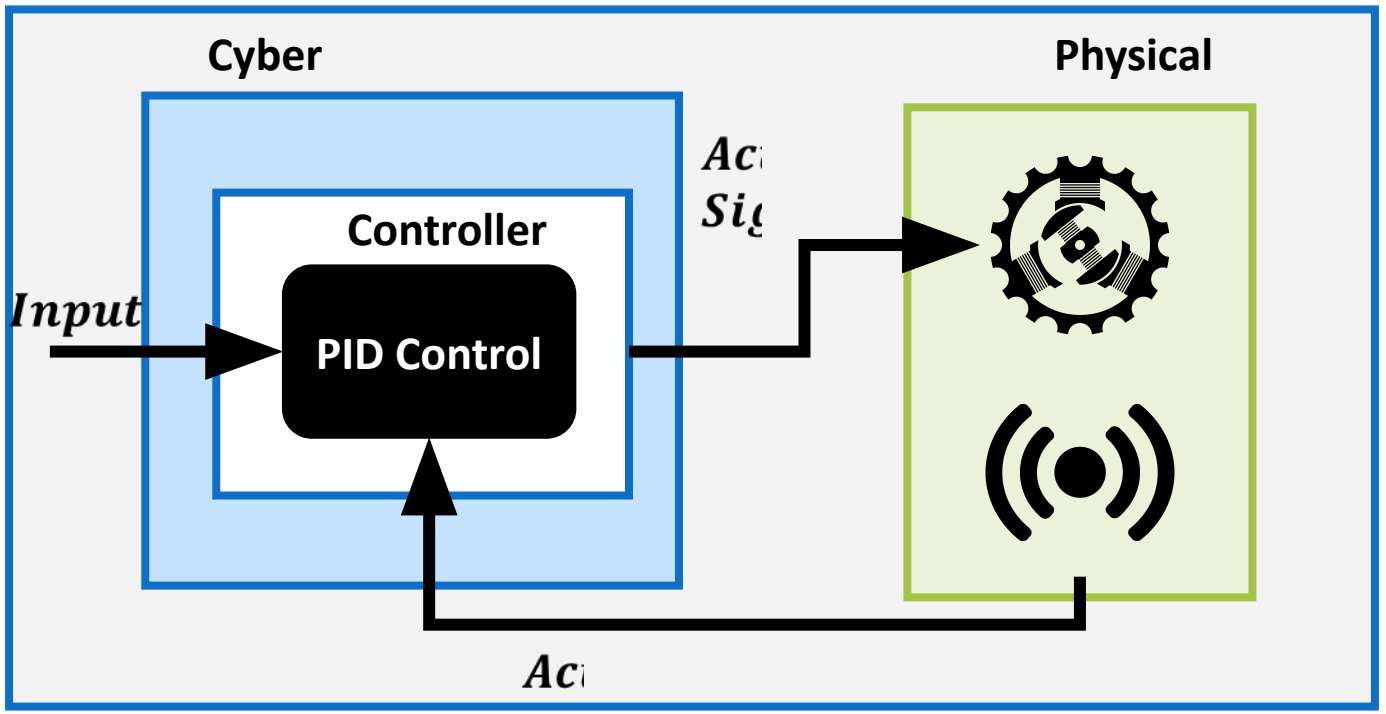


Sensor □ PID Control □ Actuator Signal

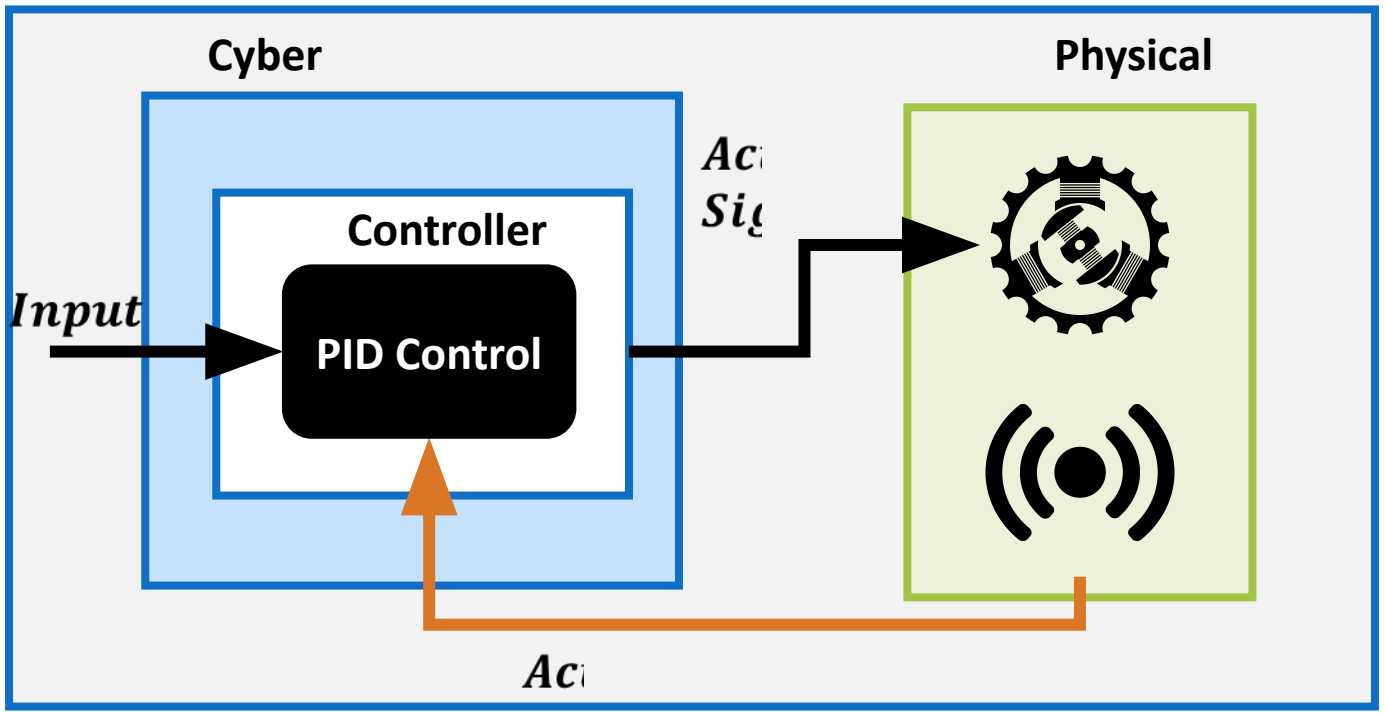


PID Control (Proportional Integral Derivative)

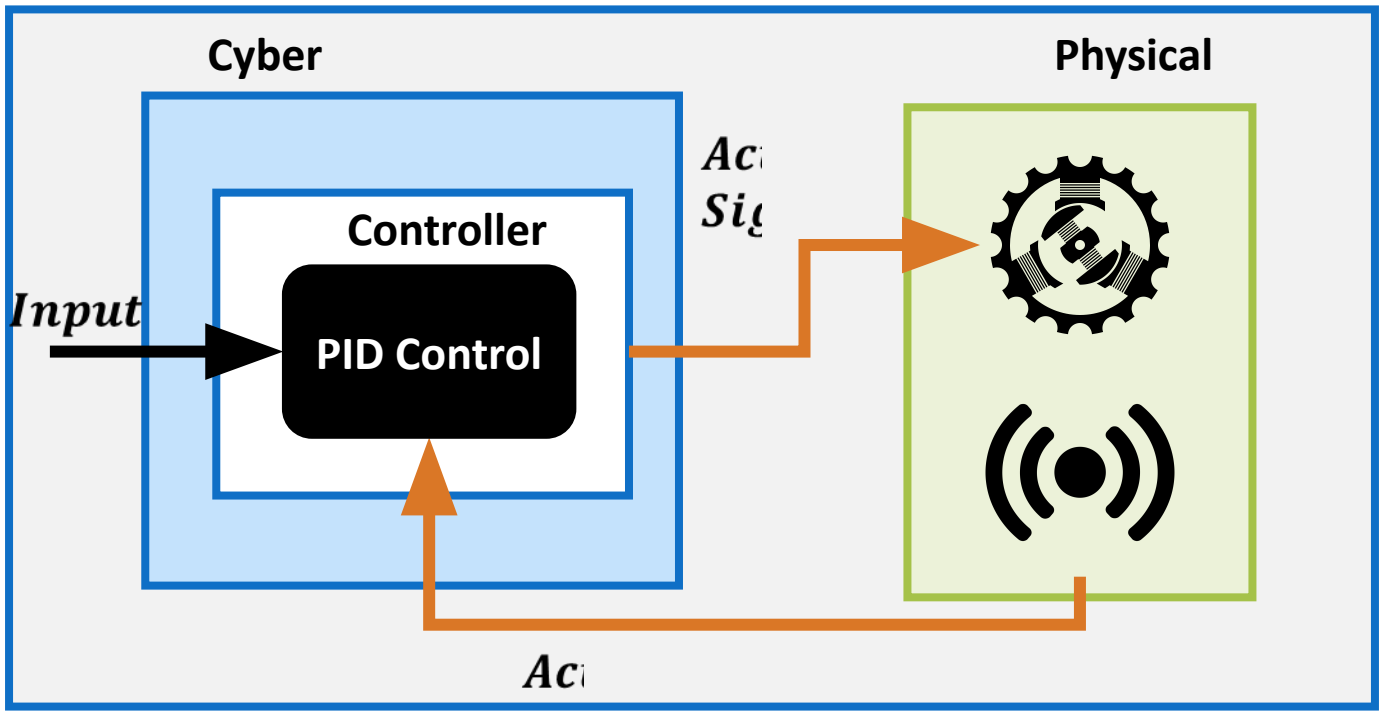
Sensor □ PID Control □ Actuator Signal



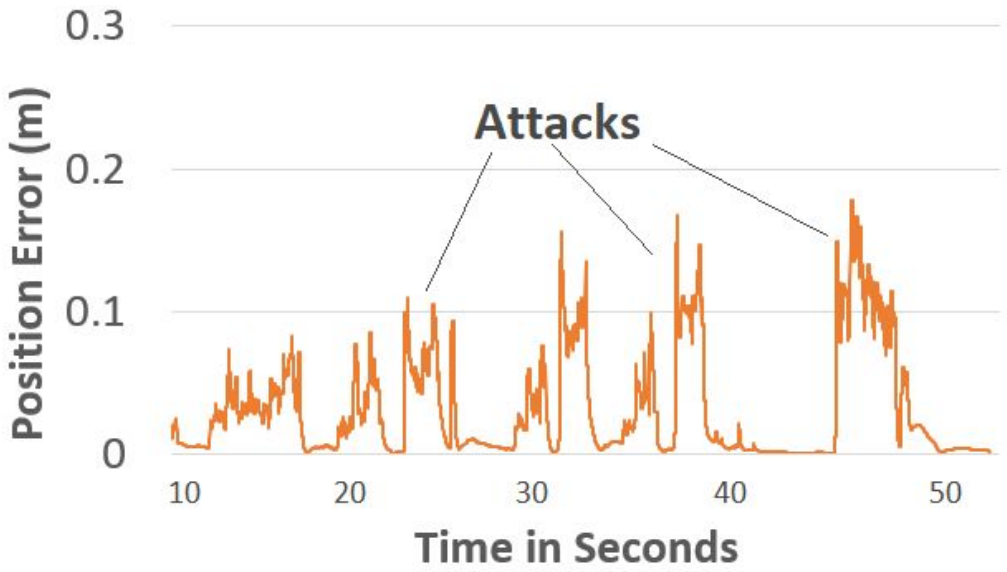
Sensor □ PID Control □ Actuator Signal



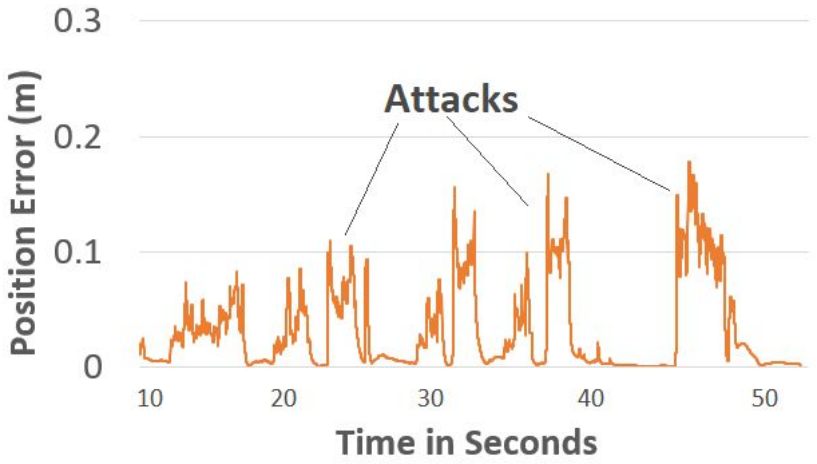
Sensor □ PID Control □ Actuator Signal



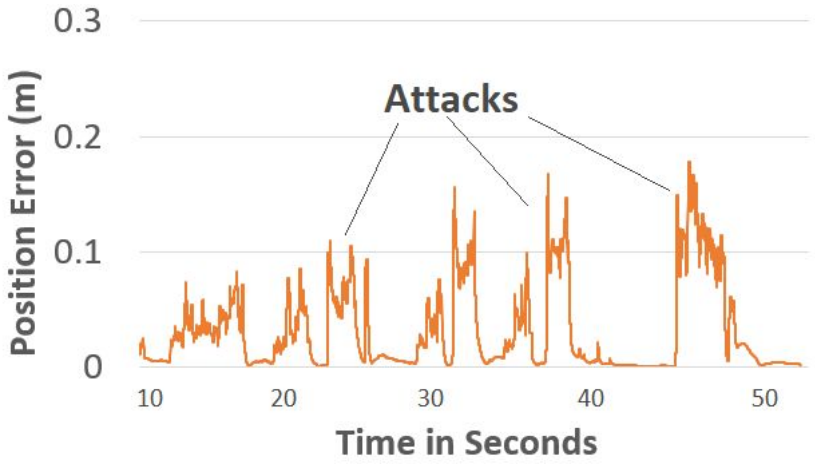
RV under Attack

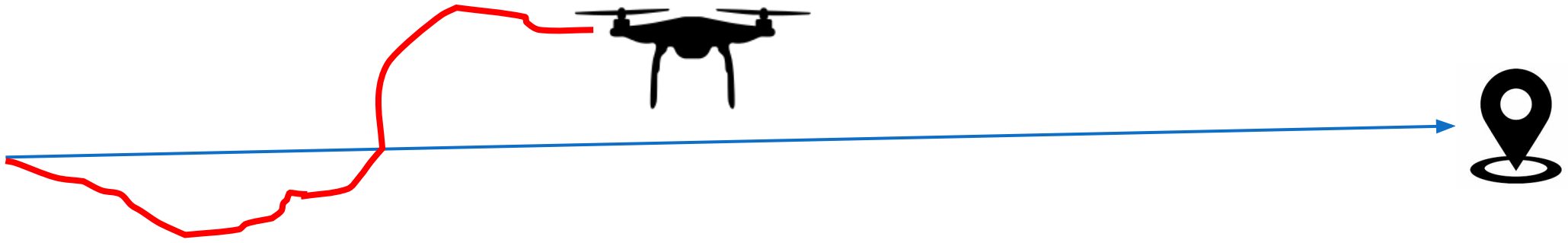


PID Over-Compensates under Attacks

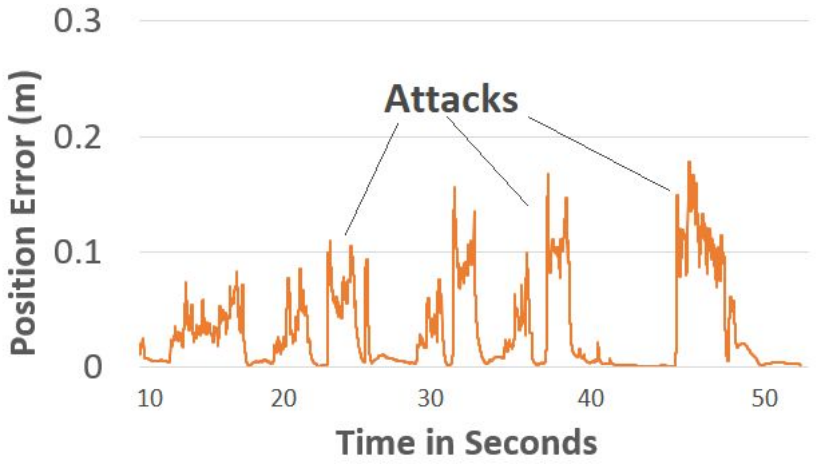


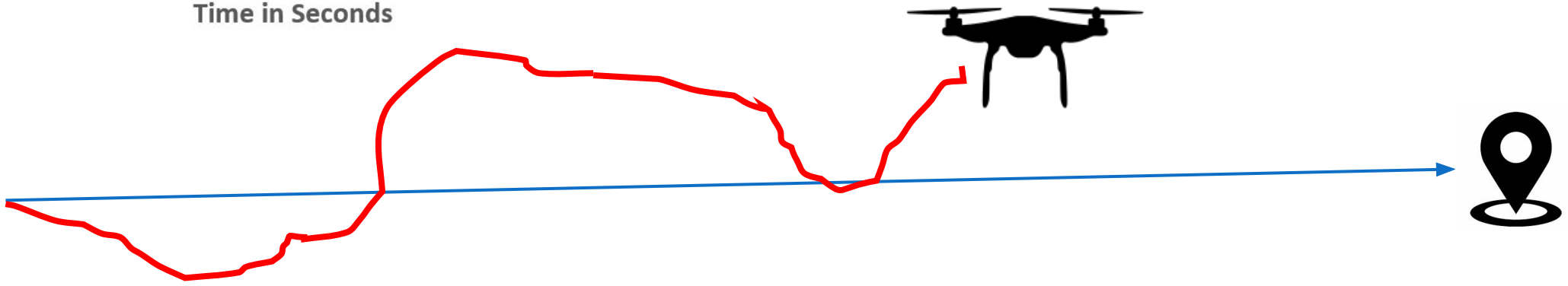
PID Over-Compensates under Attacks



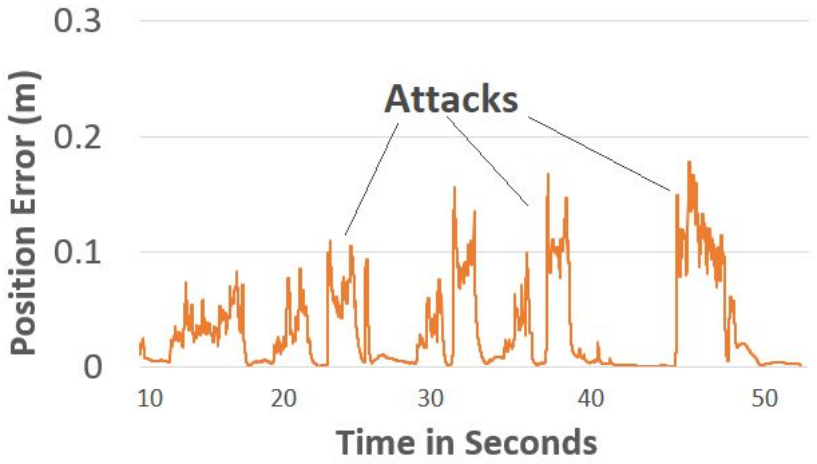


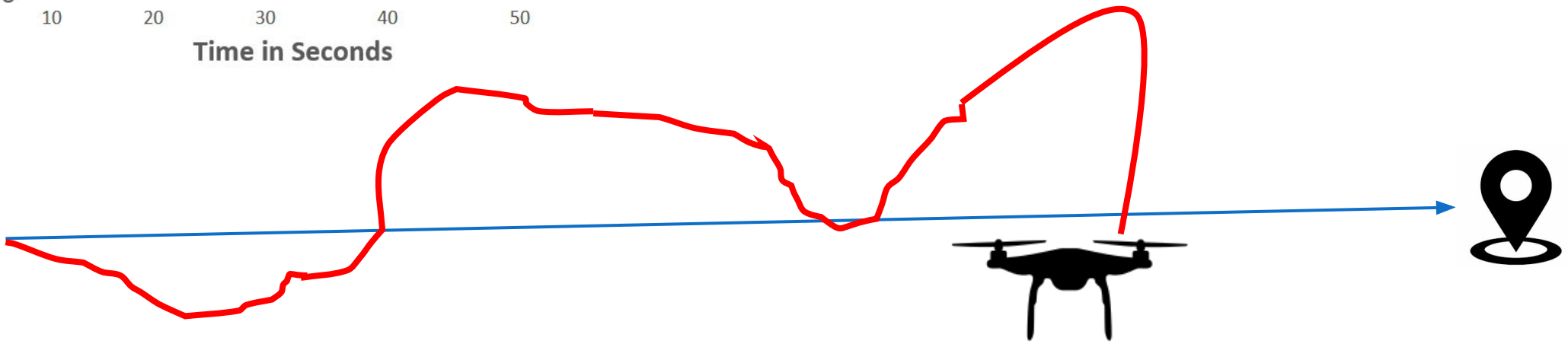
PID Over-Compensates under Attacks



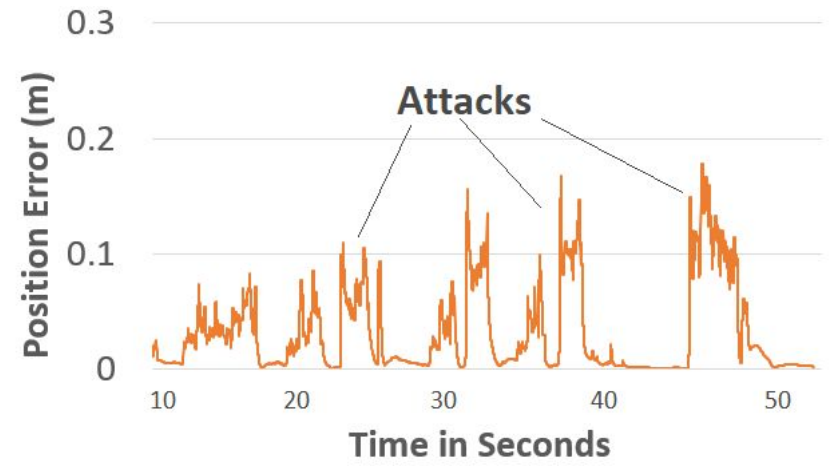


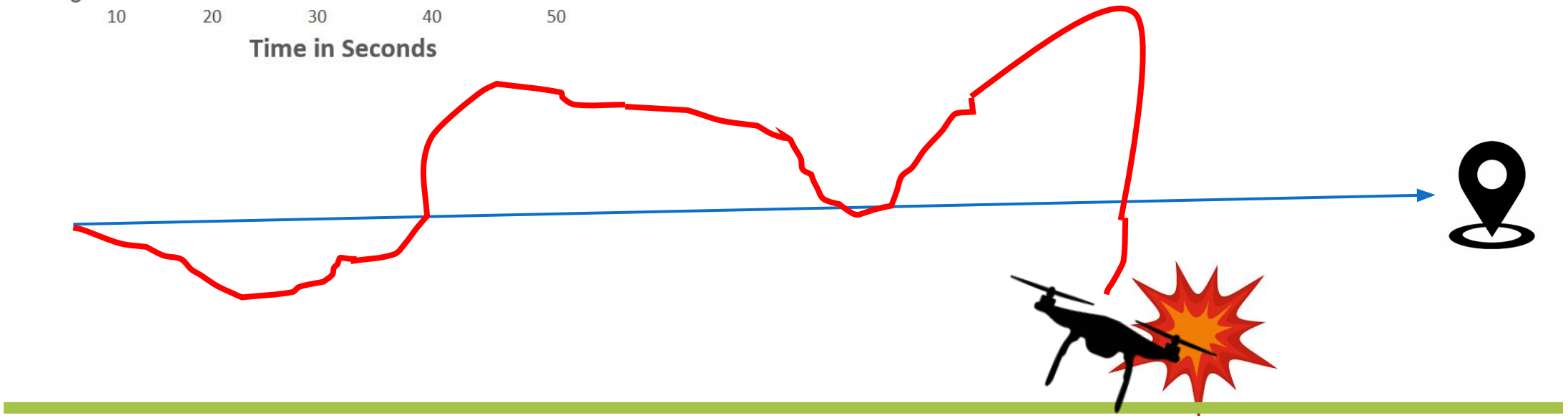
PID Over-Compensates under Attacks



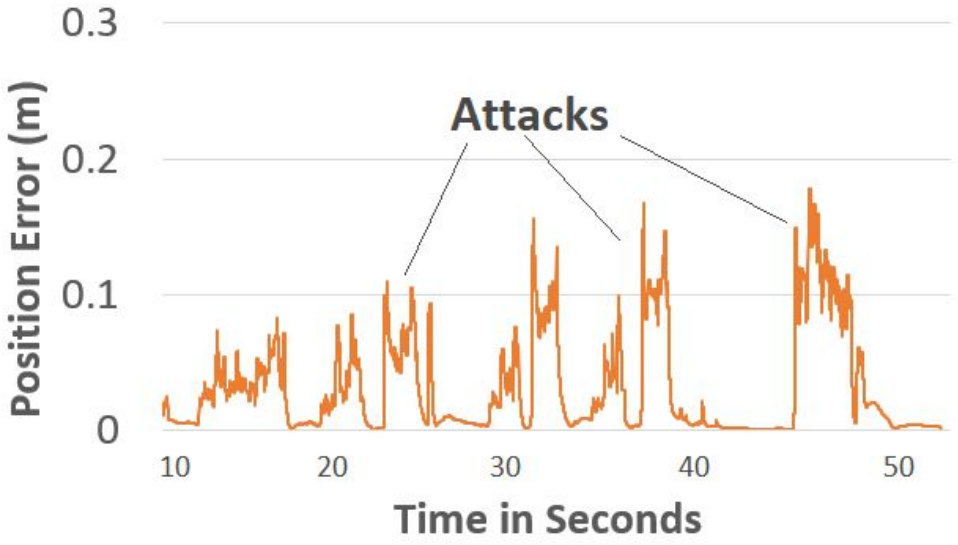


PID Over-Compensates under Attacks





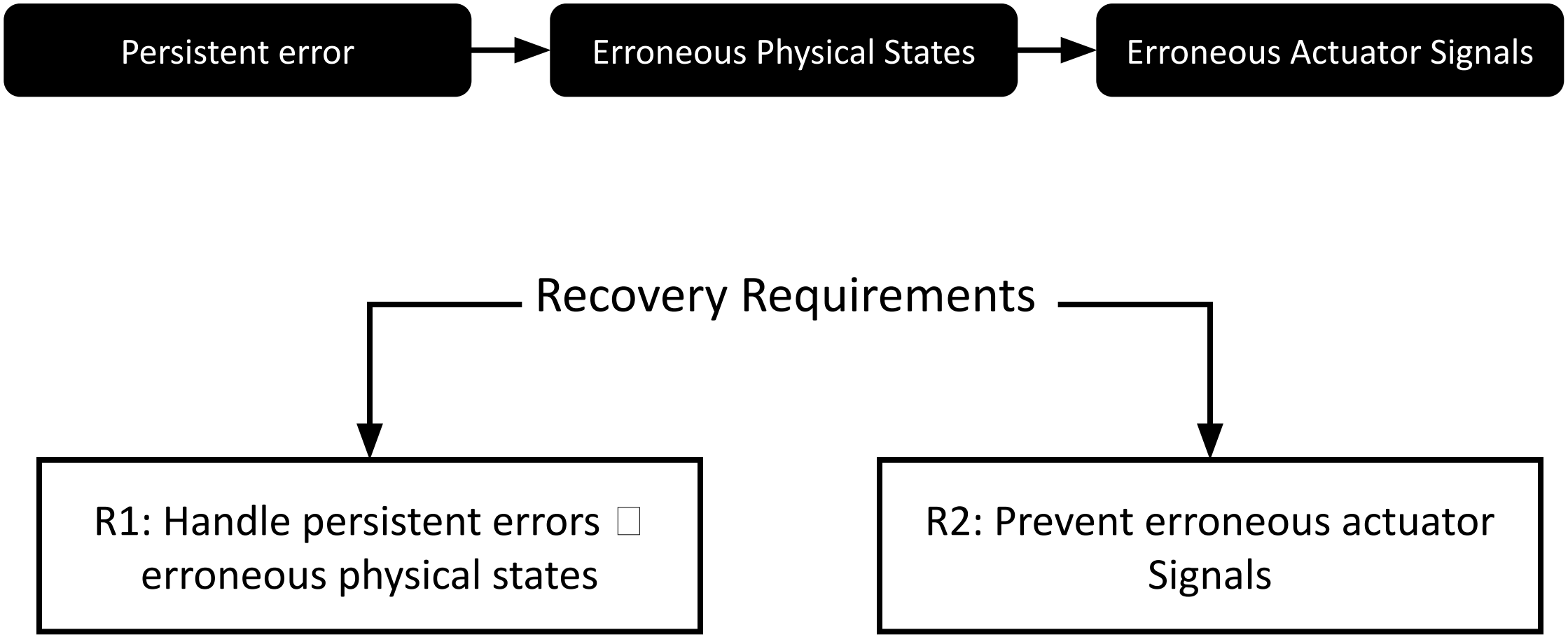
PID Over-Compensates under Attacks



PID compensation □ handling faults ✔

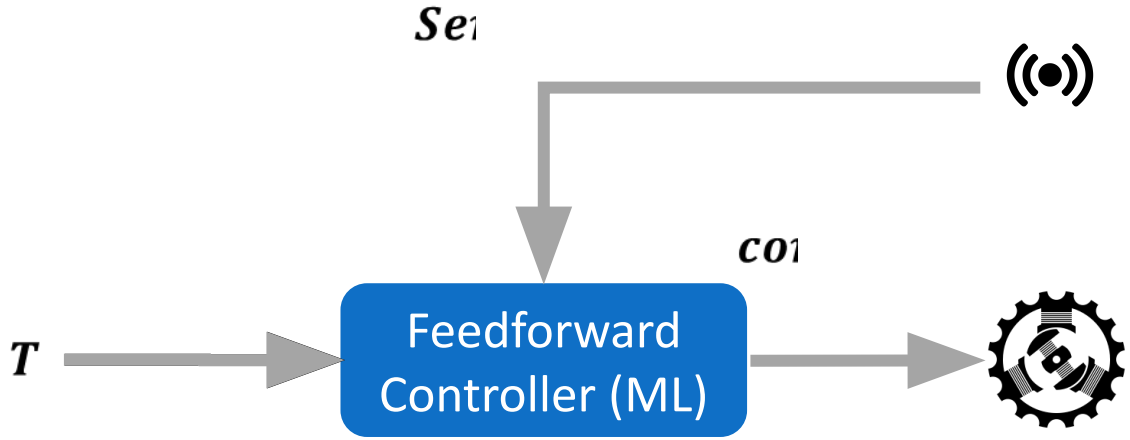
under attacks ✘

Approach to design Recovery Techniques



Feedforward Controller (FFC) Design

Feedforward Control



Recovery Requirements

R1: Prevent erroneous physical states

R2: Prevent erroneous actuator signals

FFC design using LSTM Model

Feedforward Control (FFC) design

$y(t) \rightarrow f(x(t), u(t))$

$u \rightarrow$ target waypoints

$x \rightarrow$ {gyro, mag, baro, gps, accelerometer, coefficients, ….., } 44 parameters

Reduced Feature set: 24 parameters

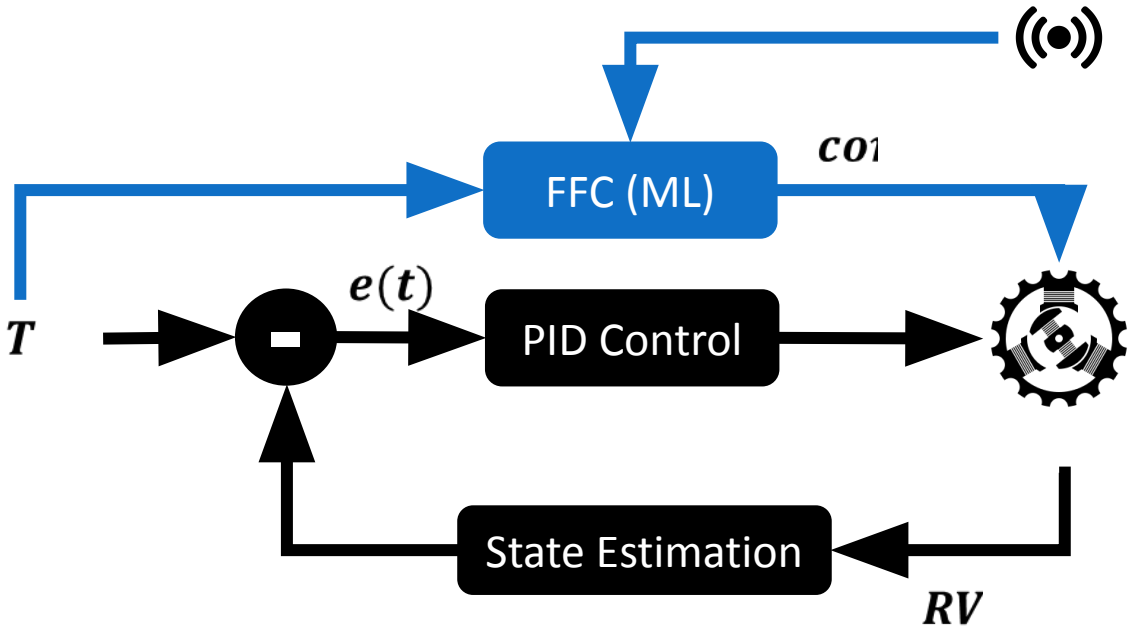
LSTM design

Correlate past and present sensors → Reject sensor perturbations

Recovery Framework

Feedforward Control

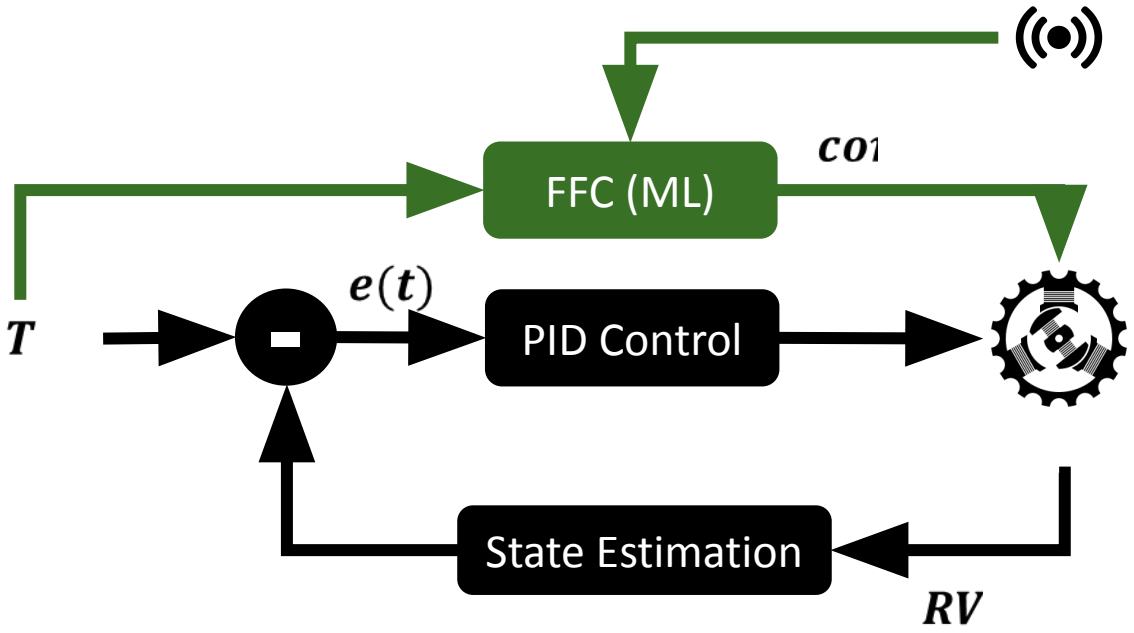
Feedback Control



Recovery Framework

Feedforward Control

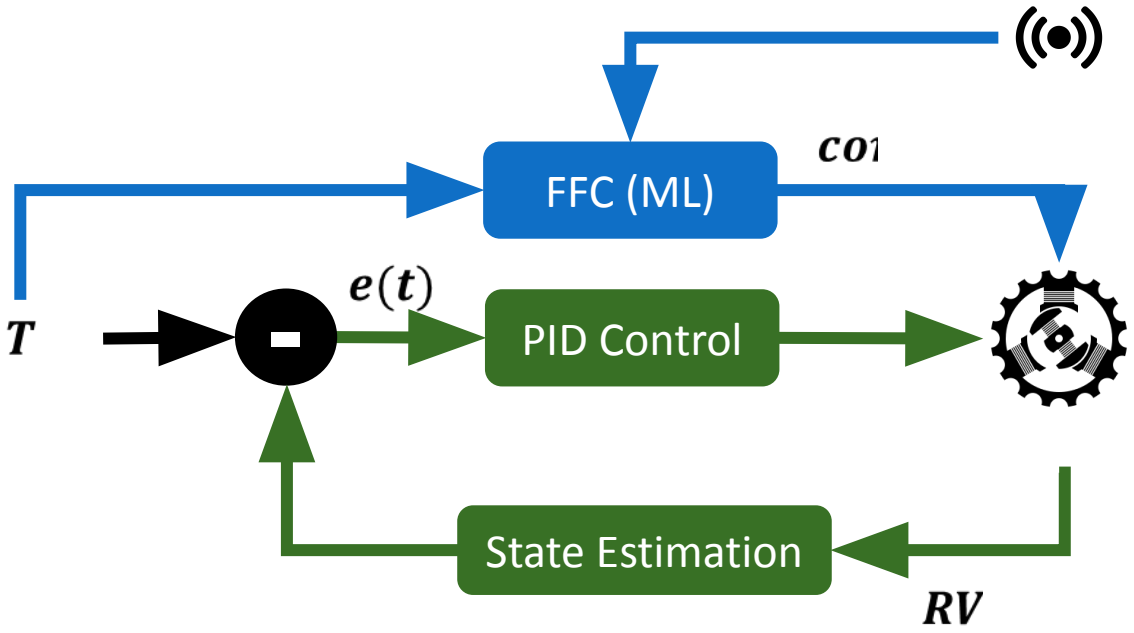
Feedback Control



Recovery Framework

Feedforward Control

Feedback Control



Experimental Setup

PID-Piper Implementation

- FFC built using LSTM model (Python)
- Trained (Python)
- Plugged into Autopilot Firmware (C++)

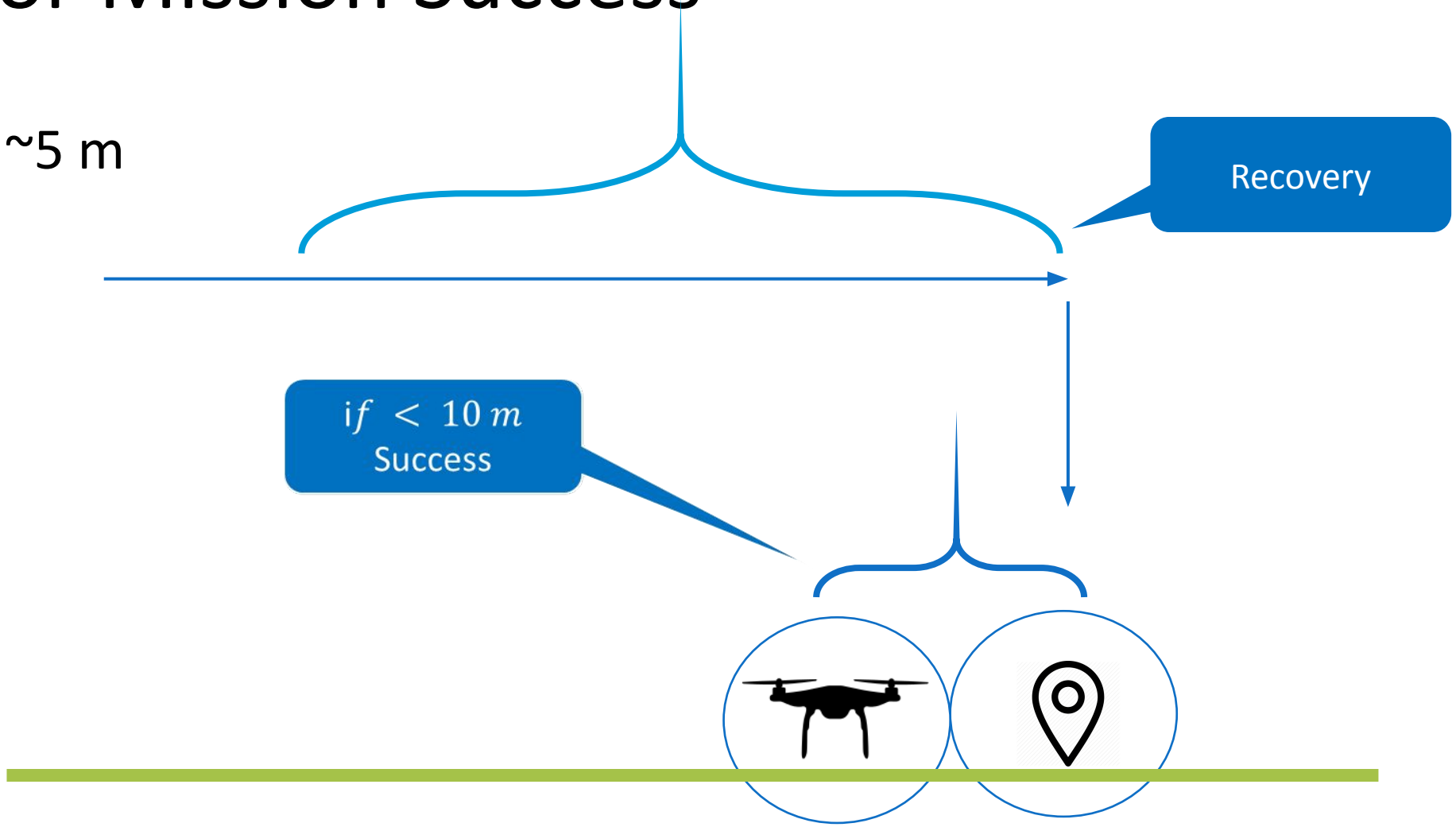
Training

- 30 RV mission profile data
- Circular, Polygonal, Straight line.

Experimental Setup

Metric for Mission Success

- GPS Offset ~5 m



PID-Piper: False Positives

Analysis Type	SRR [RAID'20]	PID-Piper [This work]
Recovery Activated	20%	10%
Missions Failed	50%	0%
FPR	**10%**	**0%**

$$FPR = \frac{Number\ of\ missions\ failed}{Total\ number\ of\ missions}$$

PID-Piper: False Positives

Analysis Type	SRR [RAID'20]	PID-Piper [This work]
Recovery Activated	20%	10%
Missions Failed	50%	0%
FPR	**10%**	**0%**

$$FPR = \frac{Number\ of\ missions\ failed}{Total\ number\ of\ missions}$$

PID-Piper: Recovery under Attacks

Analysis Type	SRR [RAID'20]	PID-Piper [This work]
Mission Success	13%	83%
Mission Failed (no Crash)	50%	17%
Crash/Stall	**37%**	**0%**

$$Mission\ Success = \frac{No.\ of\ missions\ with\ deviation < 10\ meters}{Total\ number\ of\ missions}$$

PID-Piper: Recovery under Attacks

Analysis Type	SRR [RAID'20]	PID-Piper [This work]
Mission Success	13%	83%
Mission Failed (no Crash)	50%	17%
Crash/Stall	**37%**	**0%**

$$Mission\ Success = \frac{No.\ of\ missions\ with\ deviation < 10\ meters}{Total\ number\ of\ missions}$$

PID-Piper: Recovery under Attacks

Analysis Type	SRR [RAID'20]	PID-Piper [This work]
Mission Success	13%	83%
Mission Failed (no Crash)	50%	17%
Crash/Stall	**37%**	**0%**

Recovery Successful in 83% of the cases with 0 crashes.

PID-Piper: Overheads

Analysis Type	PID-Piper [This work]
CPU Overhead	~7%
Energy Overhead	~0.9%
Mission delays	Negligible

Ongoing Work

PID-Piper cannot handle simultaneous, multiple sensor attacks

- Example: Both GPS and Gyrometer are attacked simultaneously

Our approach: **DeLorean**

- Online diagnosis using factor graphs to identify attacked sensor
-
- Historic state replay to override faulty sensor inputs
-
- Switch back to real sensor after attack subsides

Summary

- Prevents crashes – no crashes
- No false-positives
- Ensure mission success despite attacks
- ~7% performance overhead.

PID-Piper: Recovering Robotic Vehicles from Physical Attacks,

Pritam Dash, Guanpeng Li, Zitao Chen, Mehdi Karimibiuki, and Karthik Pattabiraman, IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2021. **Best Paper Award (1 of 300 submissions)**

https://github.com/DependableSystemsLab/pid-piper

PID-Piper
Recovery Videos