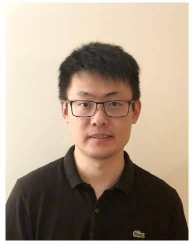
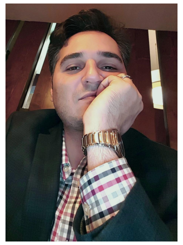
Detection is not enough: Low-Cost Attack Recovery for Robotic Vehicle Systems

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

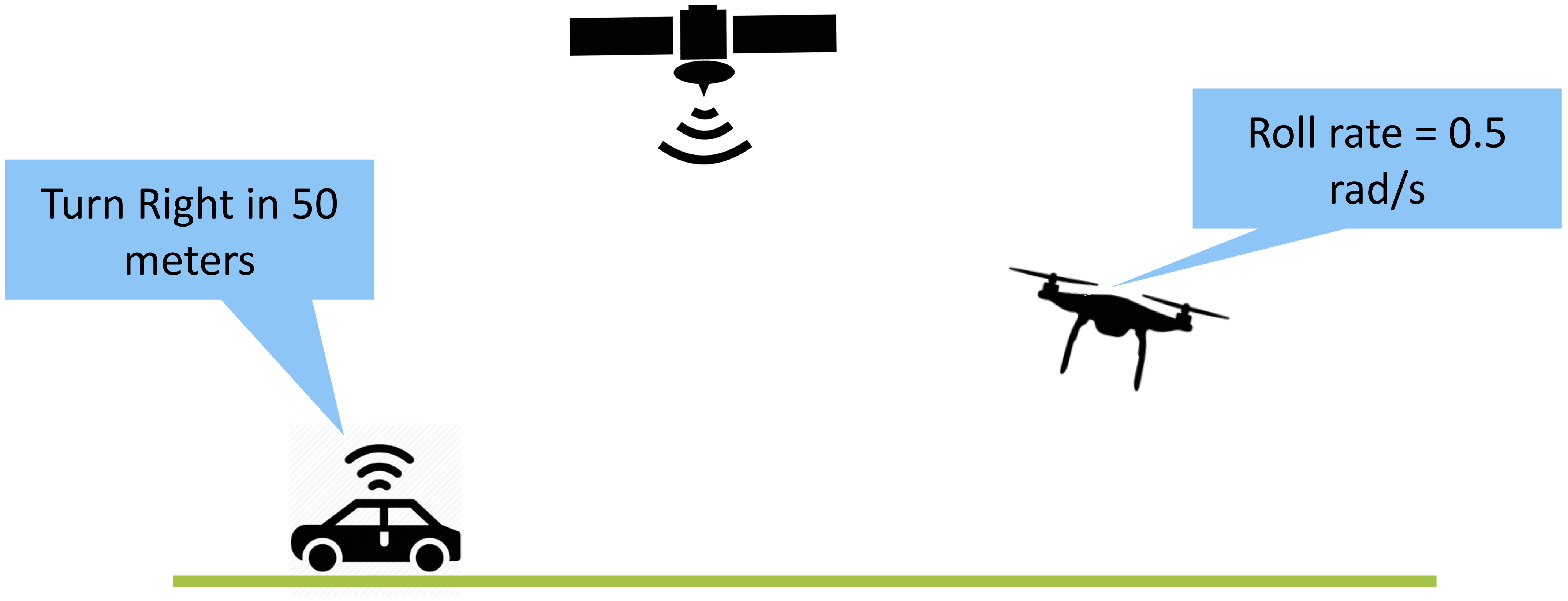
Karthik Pattabiraman

THE UNIVERSITY OF BRITISH COLUMBIA

Robotic Vehicles (RV): Motivation

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

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

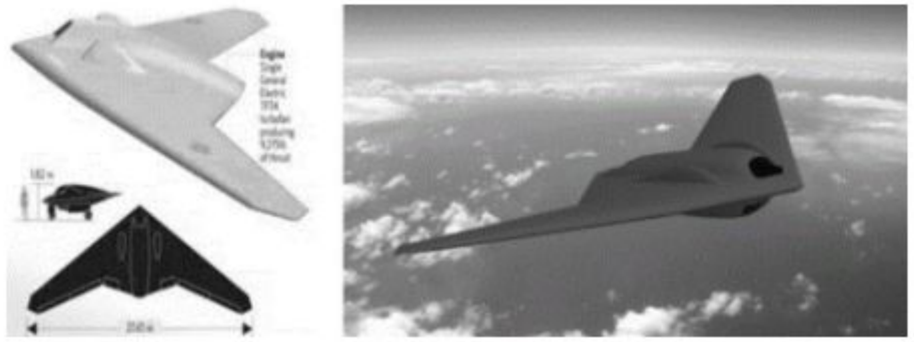
Sensor Attacks Against Robotic Vehicles (RV)

Signal Injection.
Optical, Magnetic or Acoustic noise

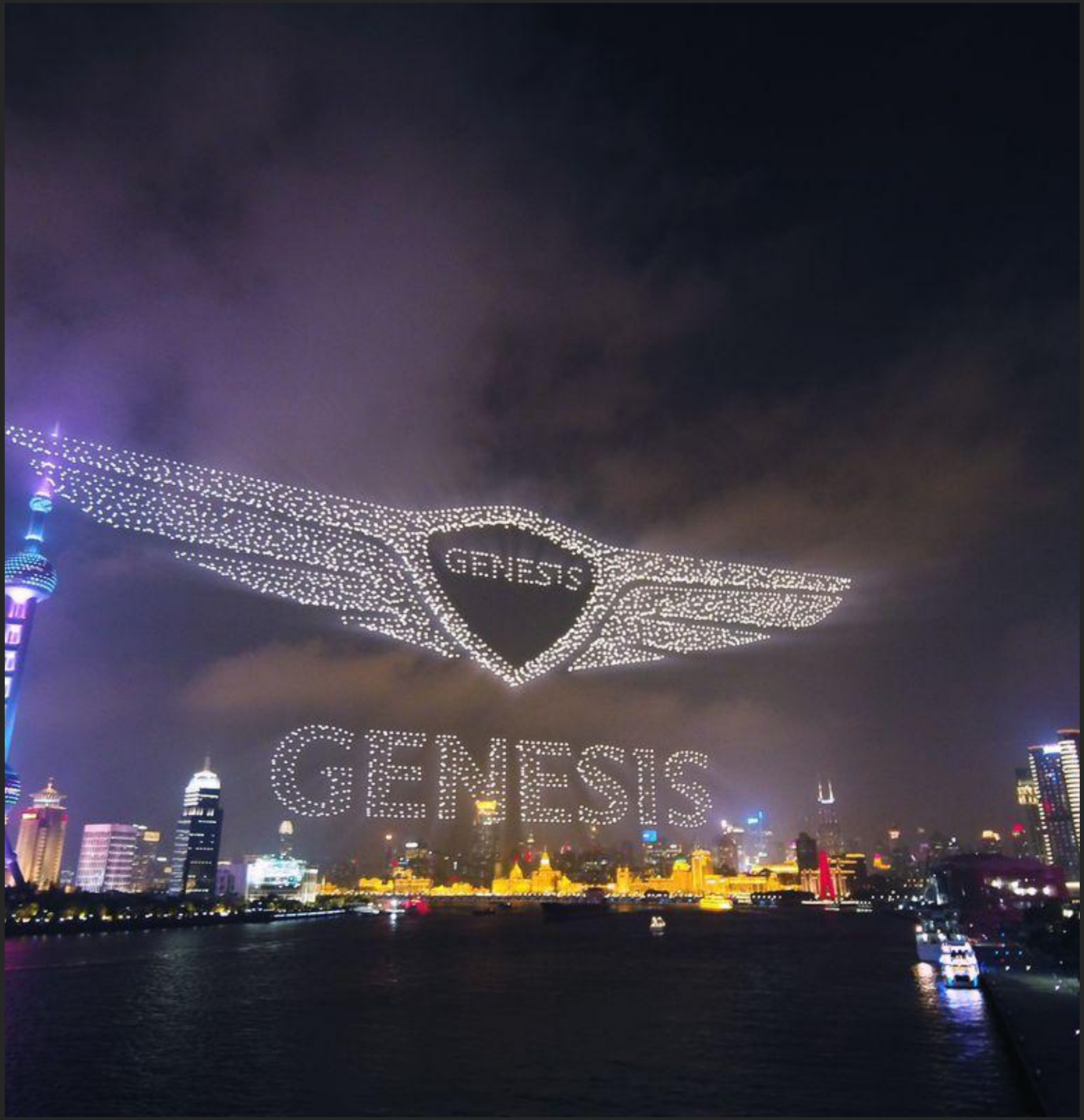
Yaw = 122.45
Roll = 0.20
Pitch =0.72

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

Sensor Attacks in the Real World


Iran–U.S. RQ-170 incident

UK Warship falsely pleased near Russian Naval Base by a GPS Cyber-attack


GENESIS
GENESIS

Prior work

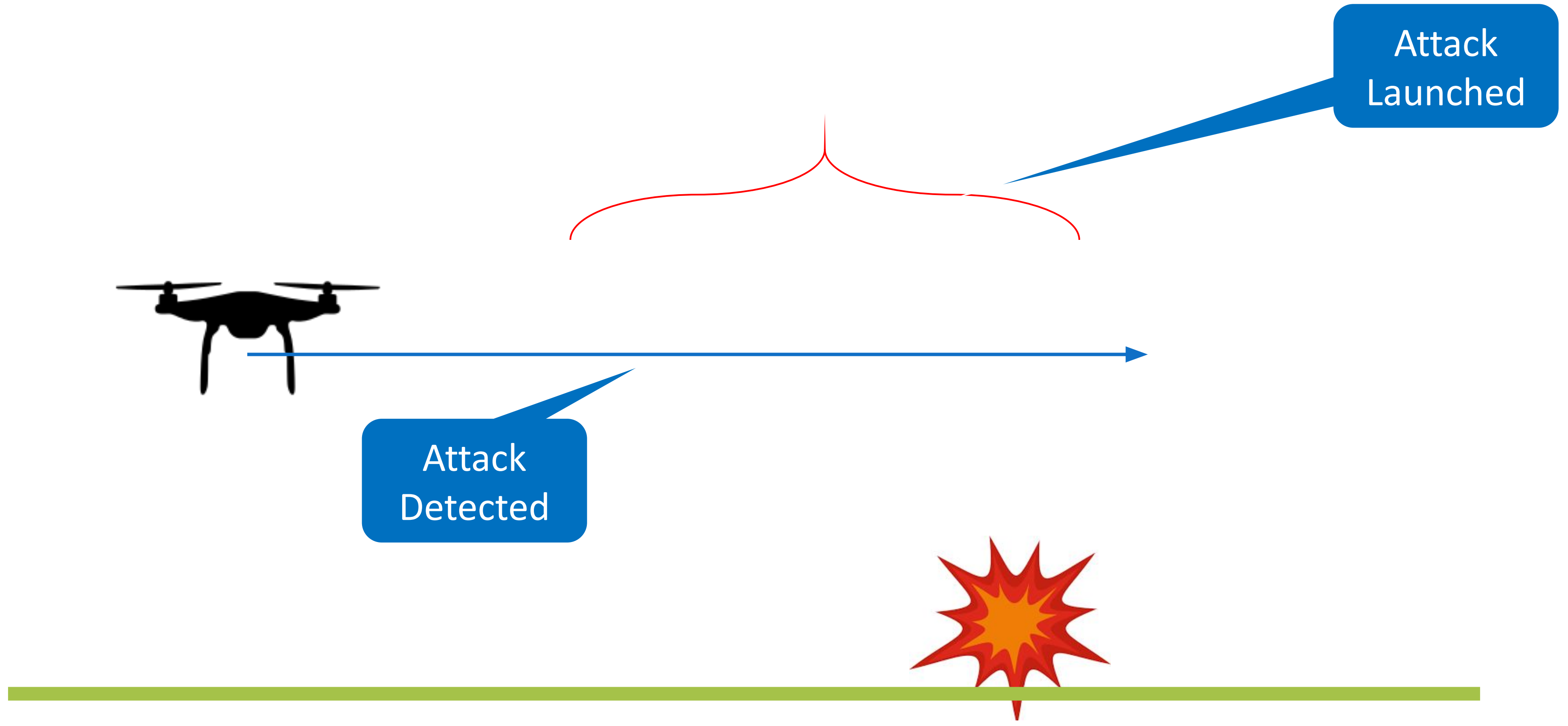
Invariant Based Detection

Model based Detection

“Very Effective in Detecting Attacks”

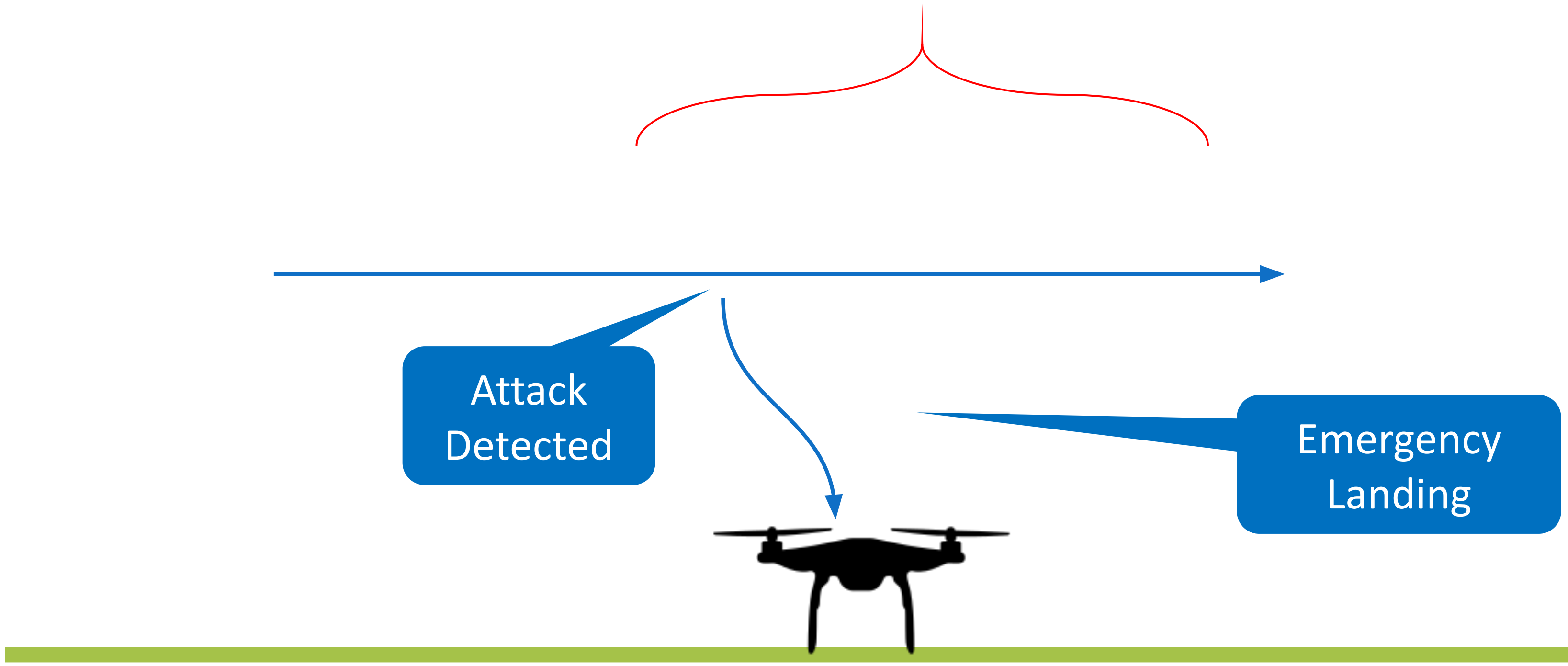
Choi et. al., Detecting Attacks against Robotic Vehicles: a Control Invariant Approach, CCS’18
Quinonez et. al., SAVIOR: Securing Autonomous Vehicles with Robust Physical Invariants, Usenix Security’20

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…



Our Goal

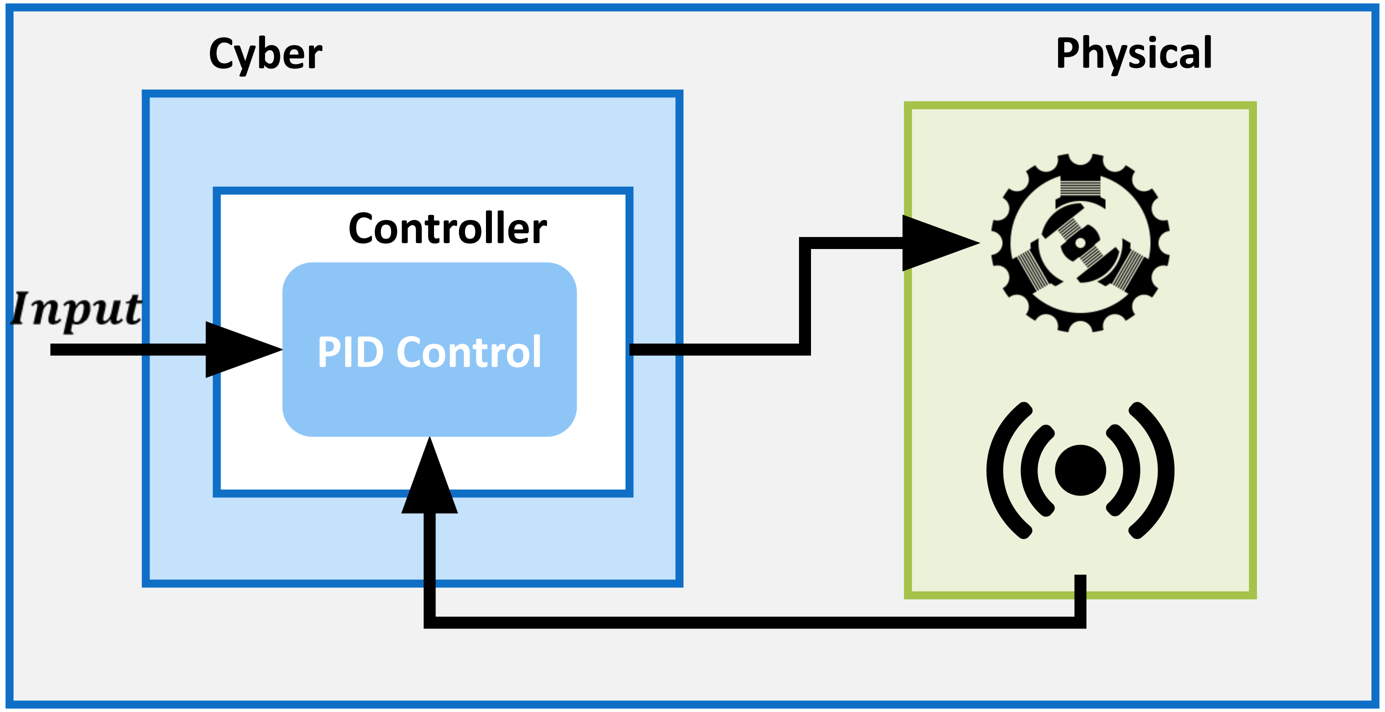
Recover from attacks and complete the mission without crashing the RV

Two Techniques for Attack Recovery:

1. **PID-Piper [DSN’21 - Best paper award]**

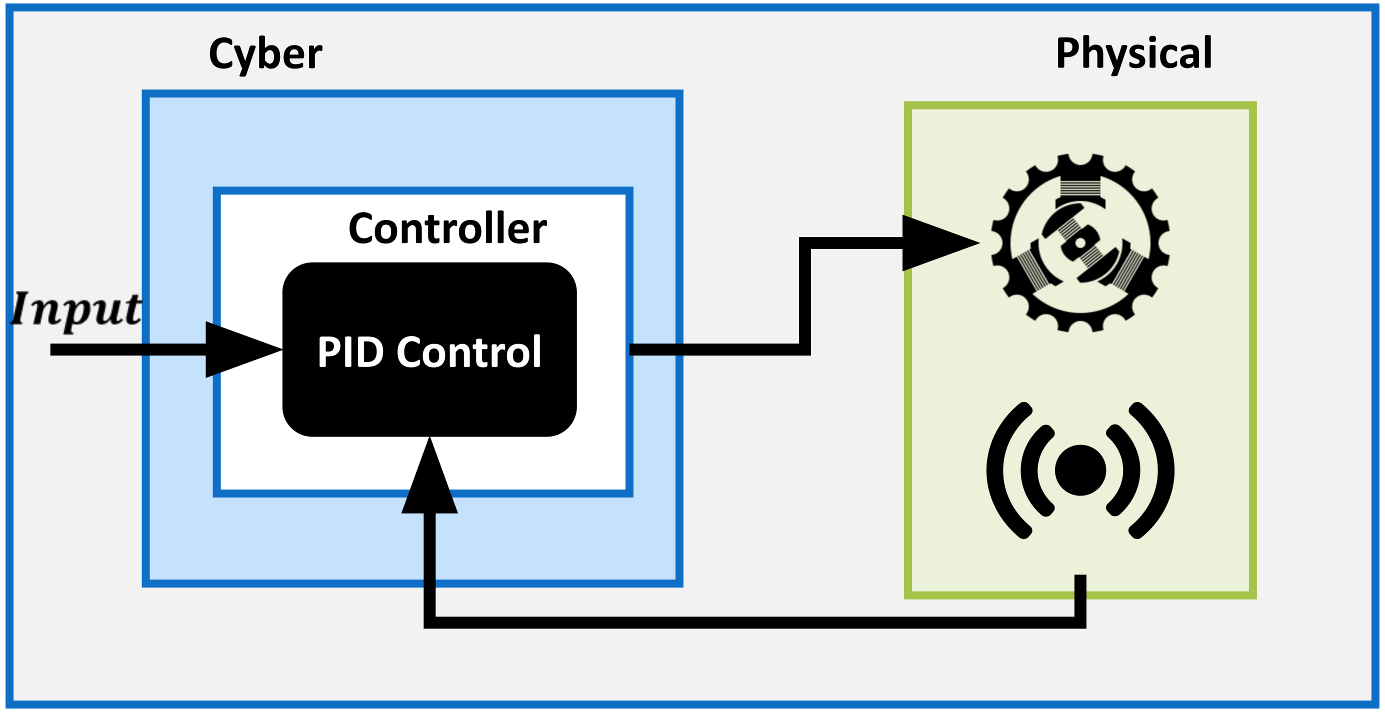
2. **DeLorean [Under submission]**

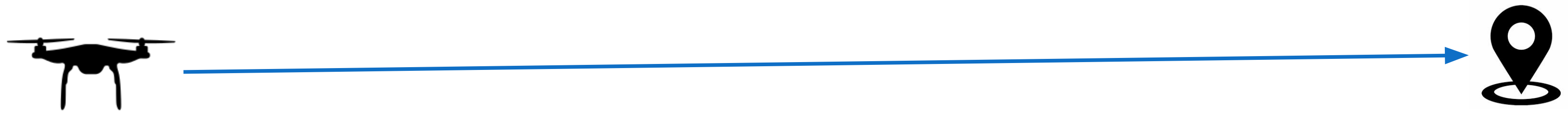
Sensor □ PID Control □ Actuator Signal



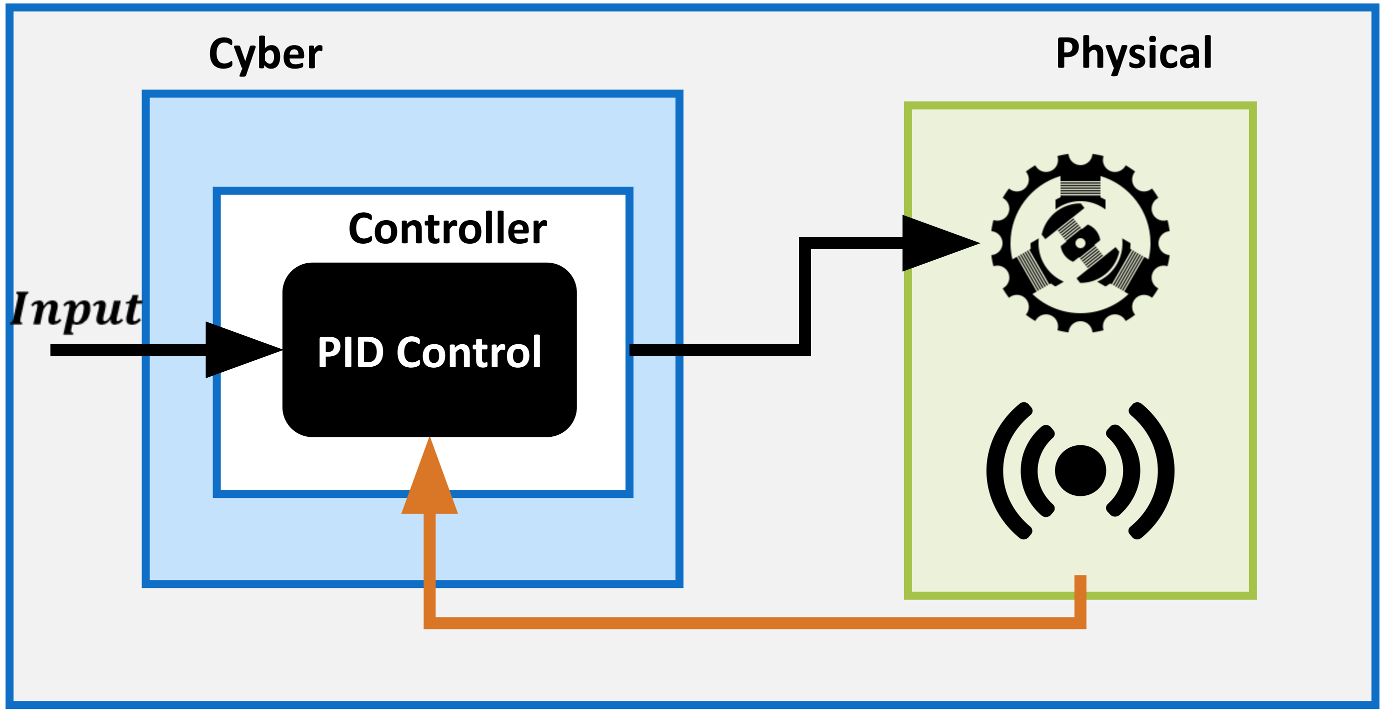
PID Control (Proportional Integral Derivative)

Sensor PID Control Actuator Signal

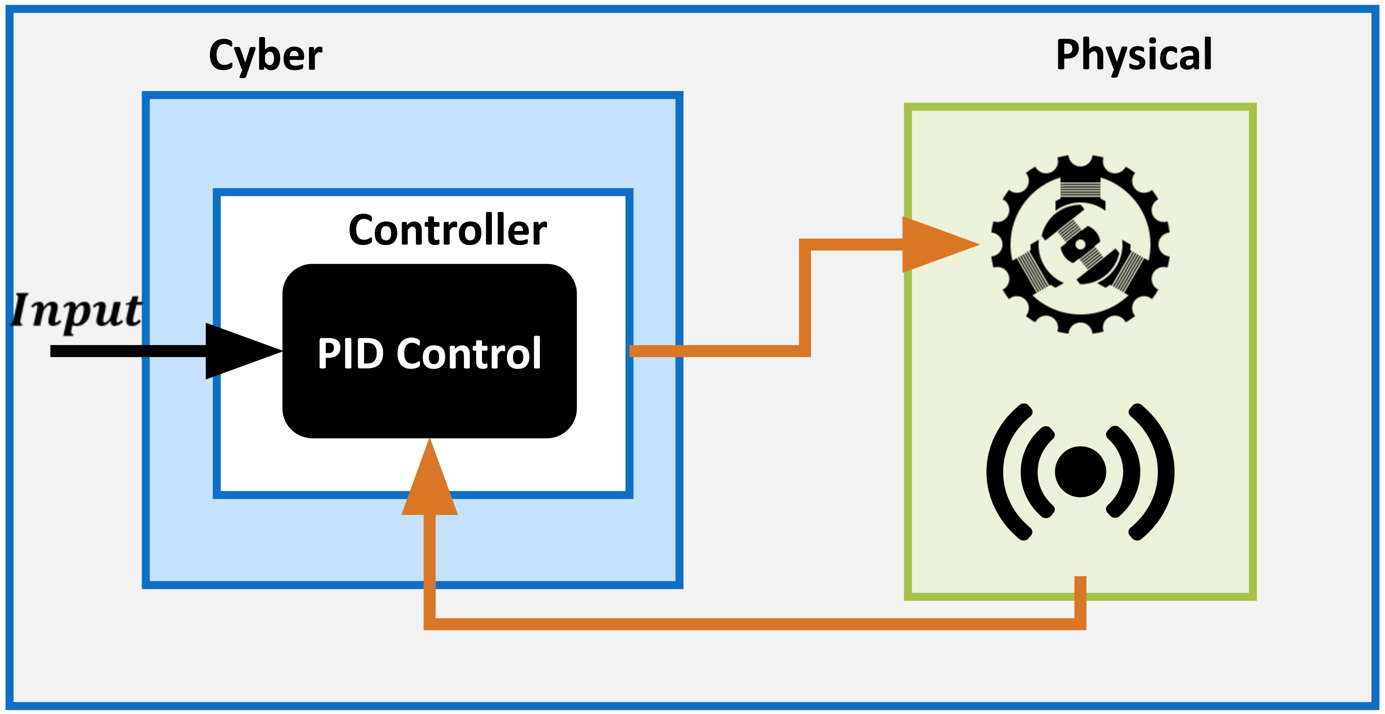




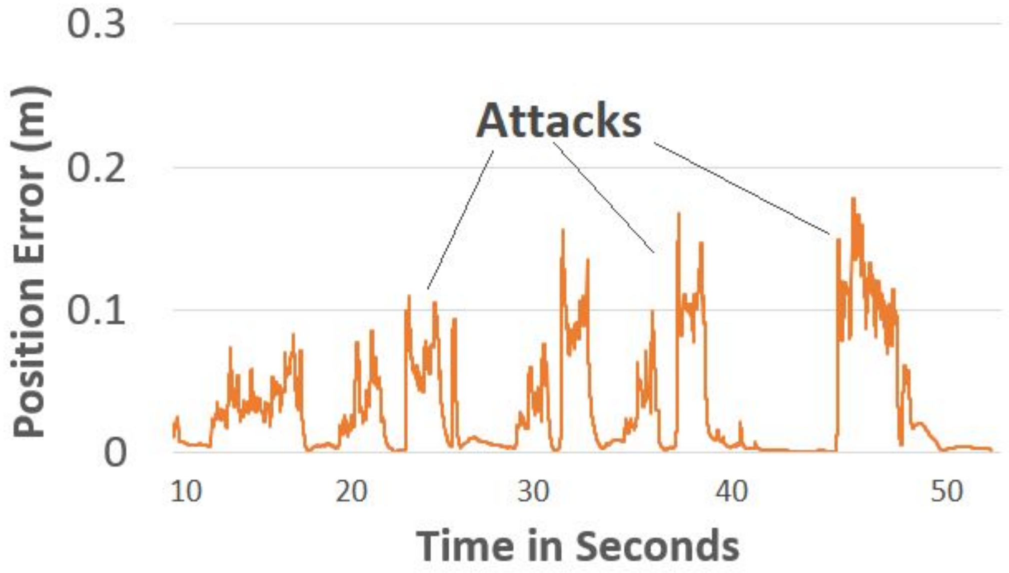
Sensor PID Control Actuator Signal



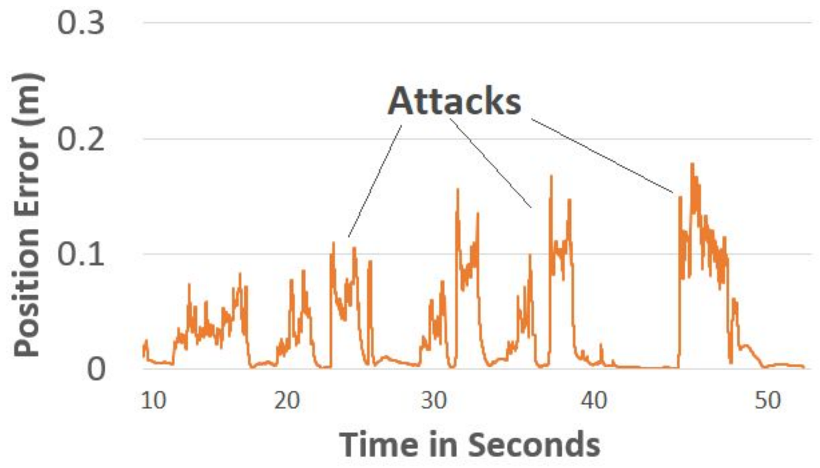
Sensor PID Control Actuator Signal

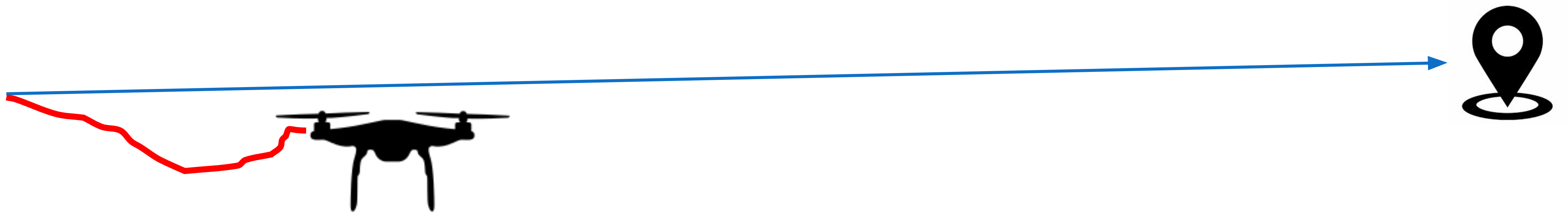


RV under Attack

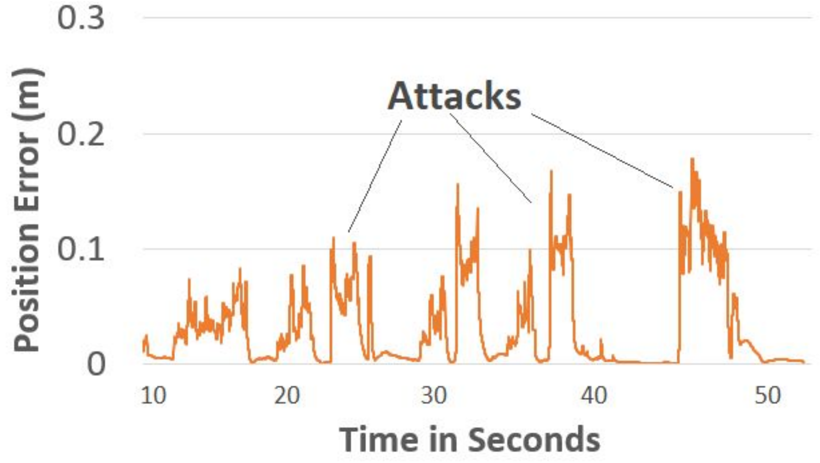


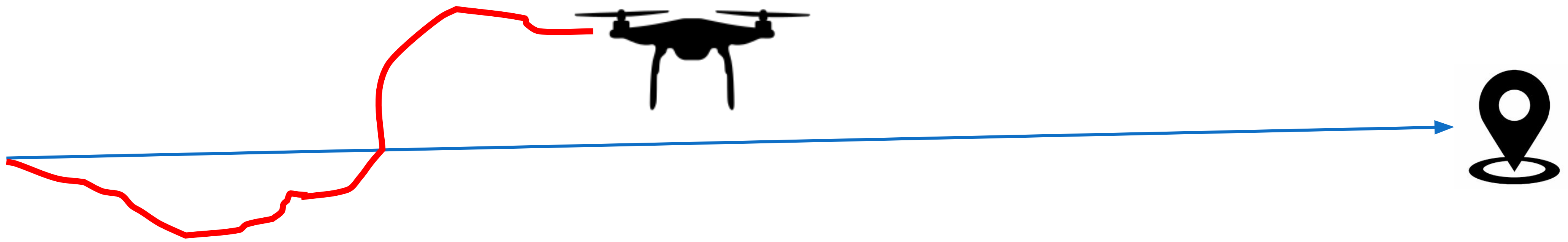
PID Over-Compensates under Attacks



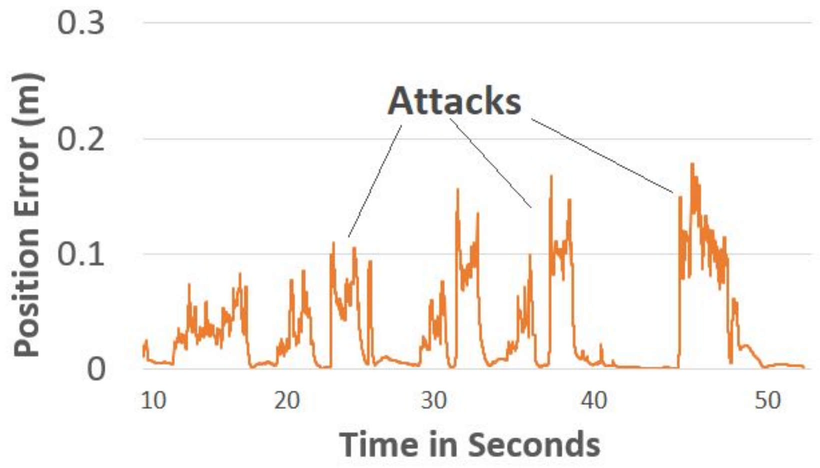


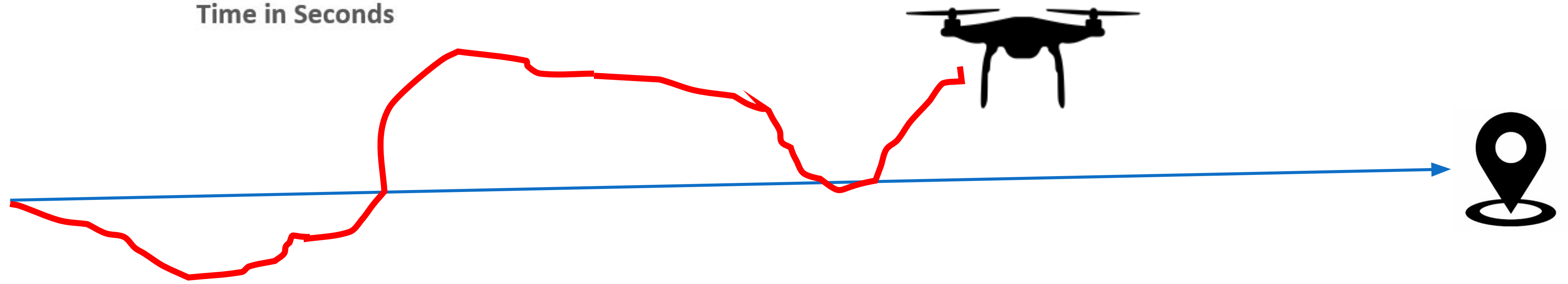
PID Over-Compensates under Attacks



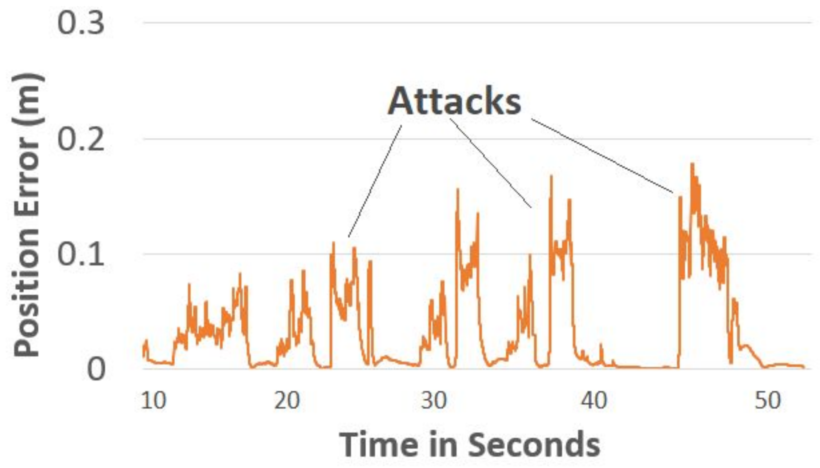


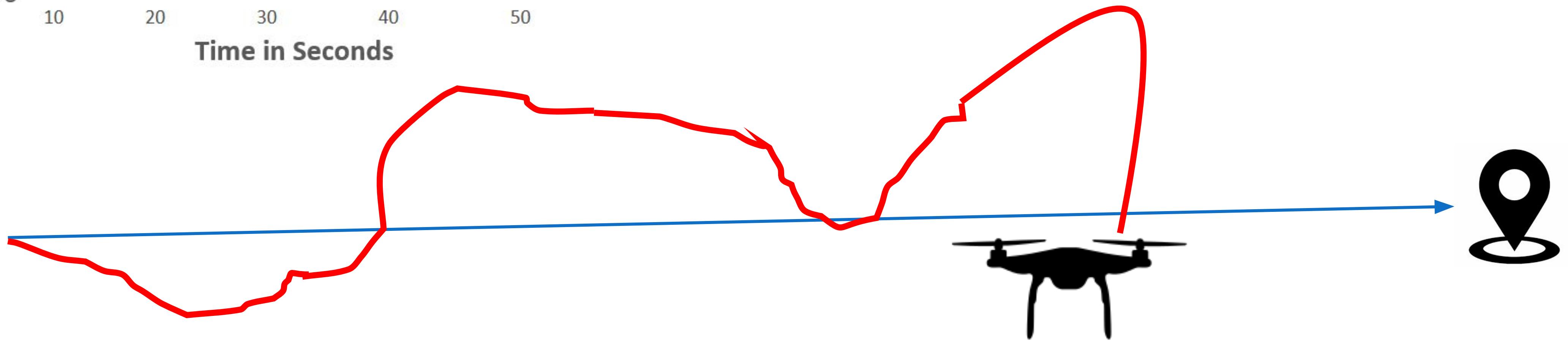
PID Over-Compensates under Attacks



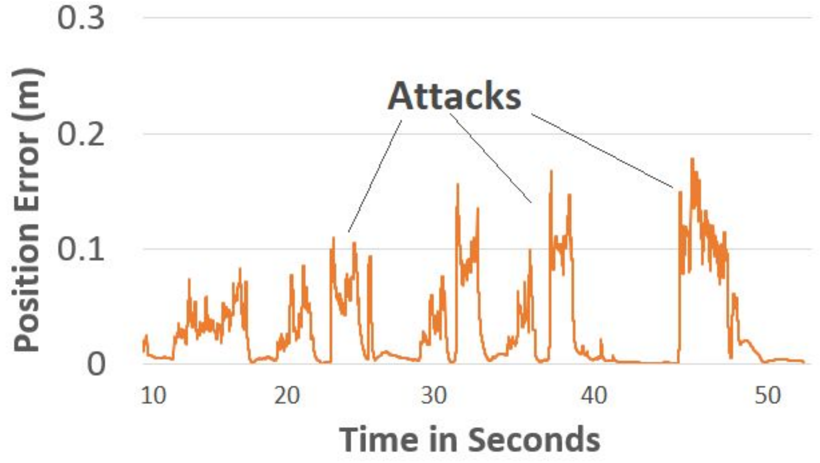


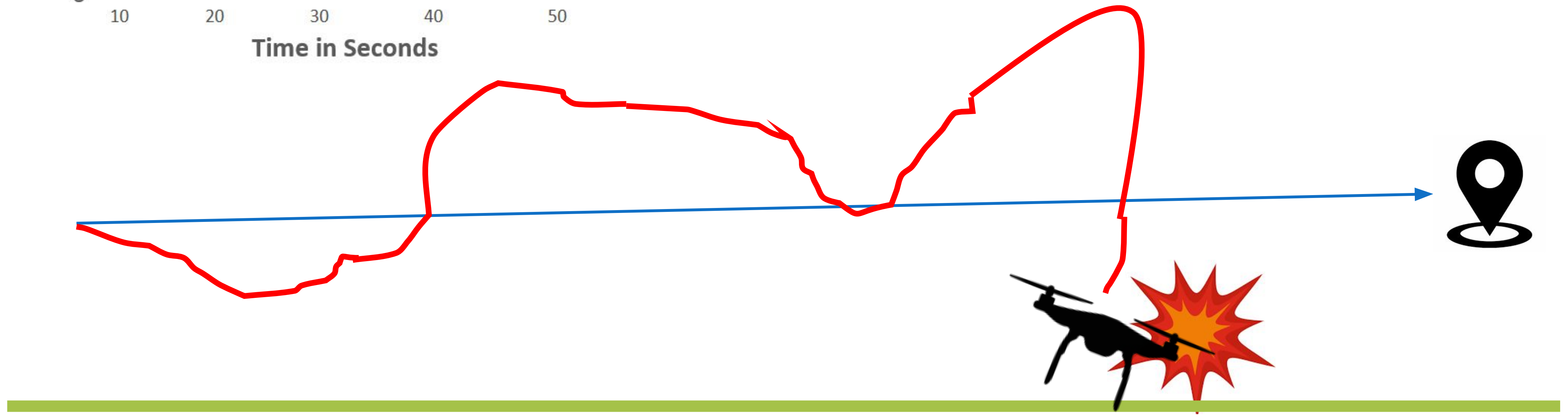
PID Over-Compensates under Attacks



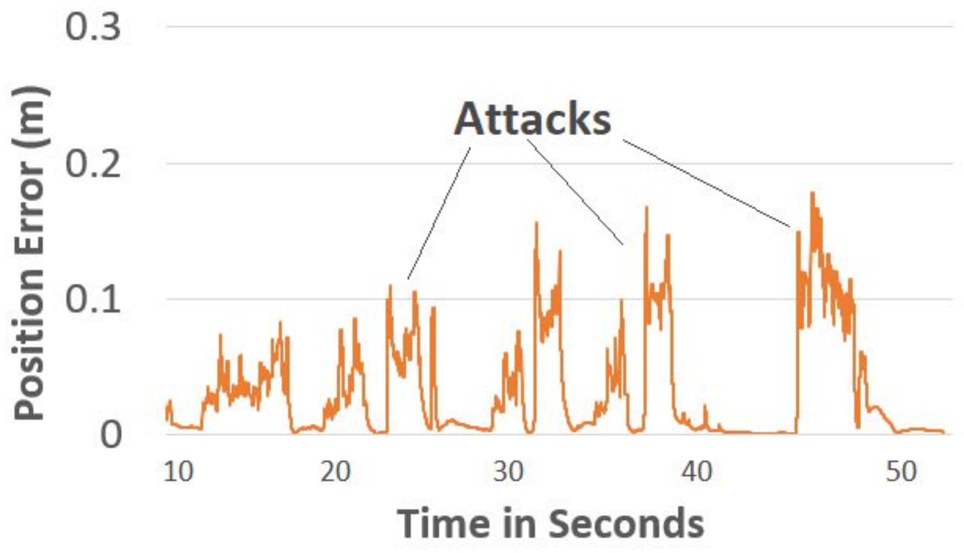


PID Over-Compensates under Attacks





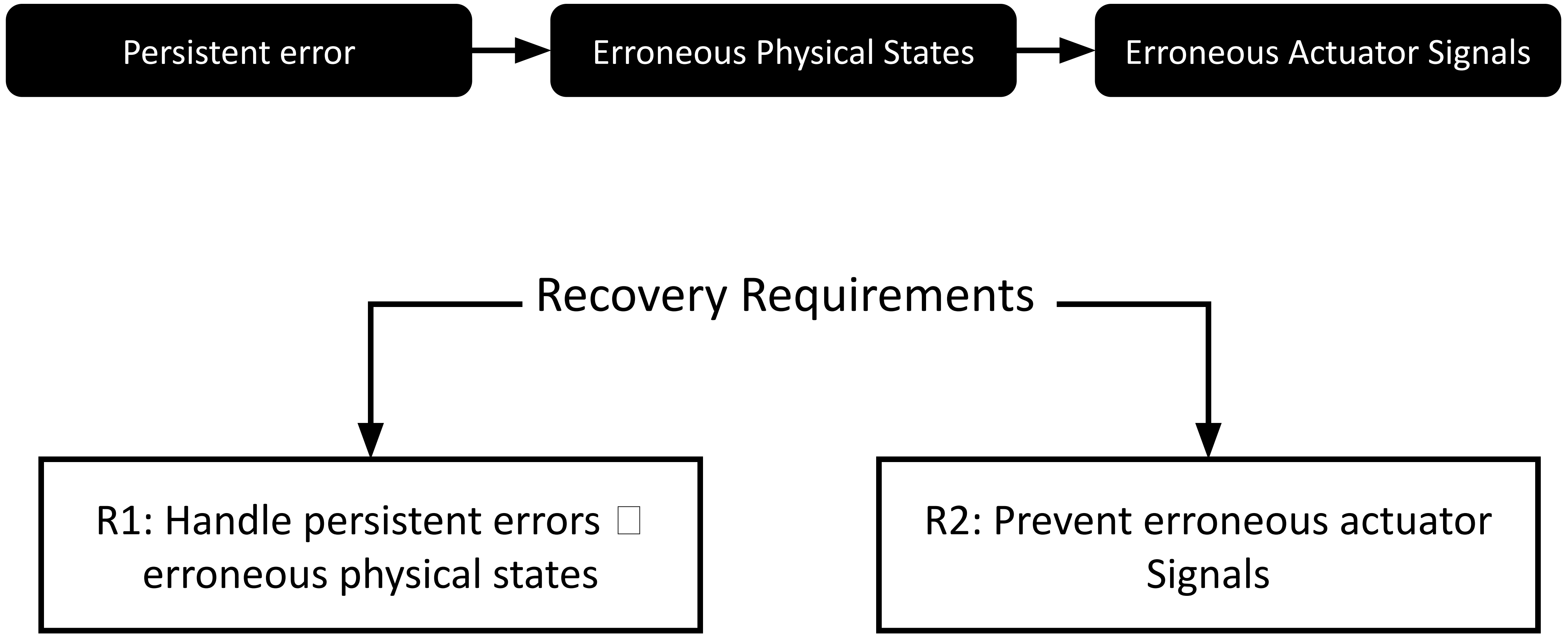
PID Over-Compensates under Attacks



PID compensation   handling faults ✓

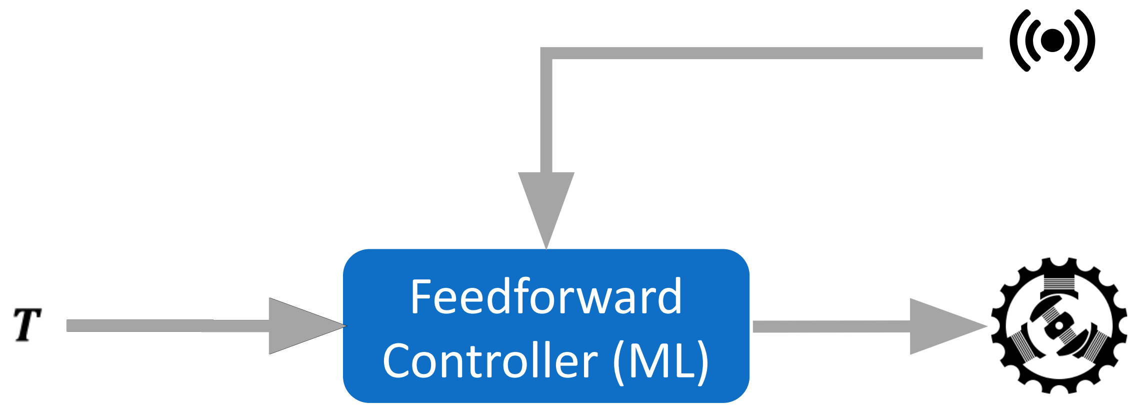
under attacks ✗

Approach to design Recovery Techniques


Persistent error
Erroneous Physical States
Erroneous Actuator Signals
Recovery Requirements
R1: Handle persistent errors erroneous physical states
R2: Prevent erroneous actuator Signals

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

$u(t) \longleftarrow f(x(t), w(t))$

$w \longrightarrow$ waypoints

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

Feature Engineering $\longrightarrow$ Reduced Feature set: 24 parameters

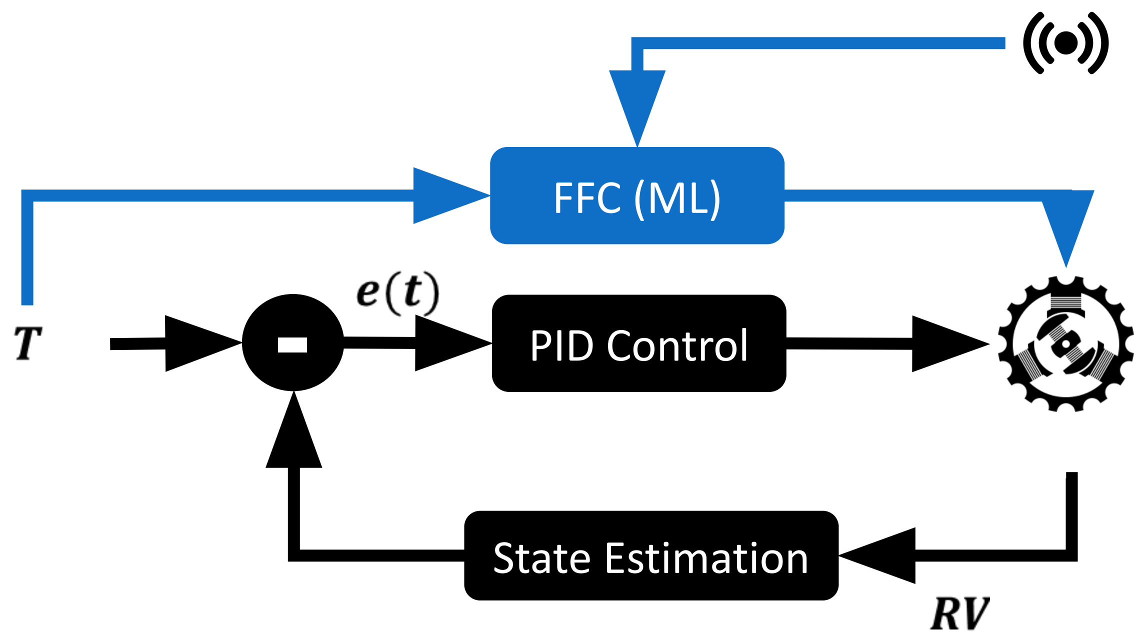
LSTM design

Correlate past and present sensors $\longrightarrow$ Reject sensor perturbations

PID-Piper: Recovery Framework

Feedforward Control

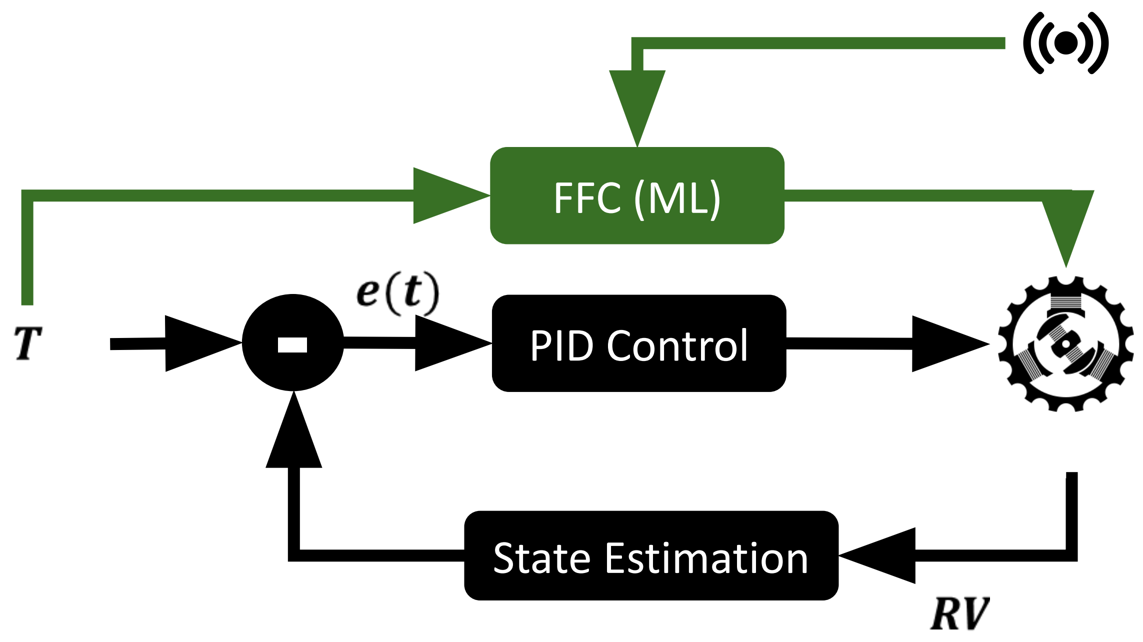
Feedback Control



PID-Piper: Recovery Framework

Feedforward Control

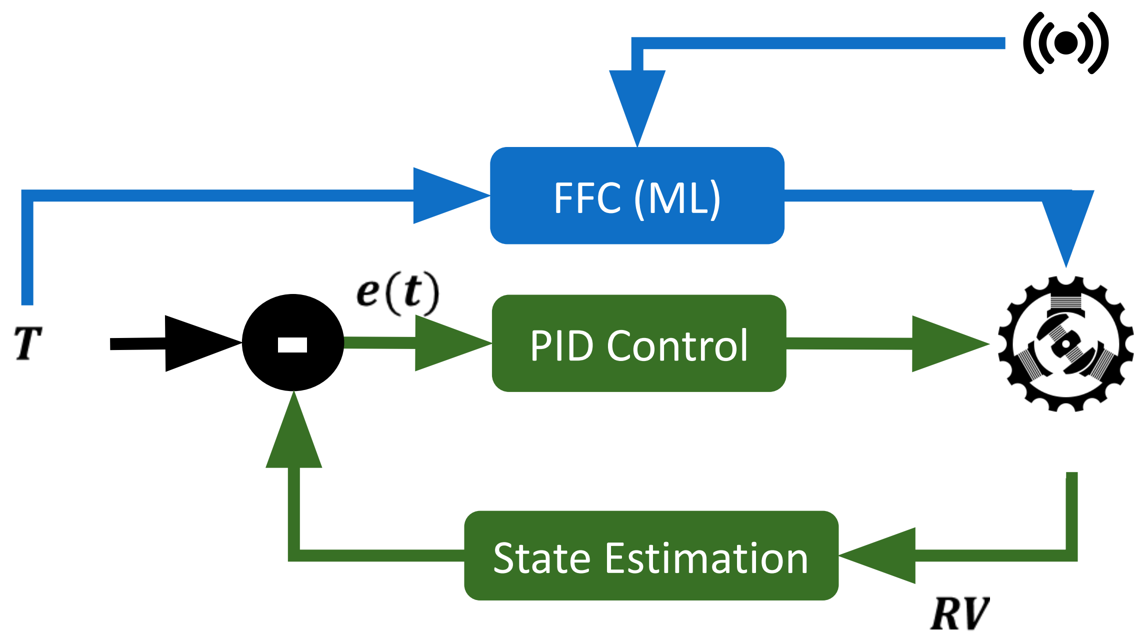
Feedback Control



PID-Piper: 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.

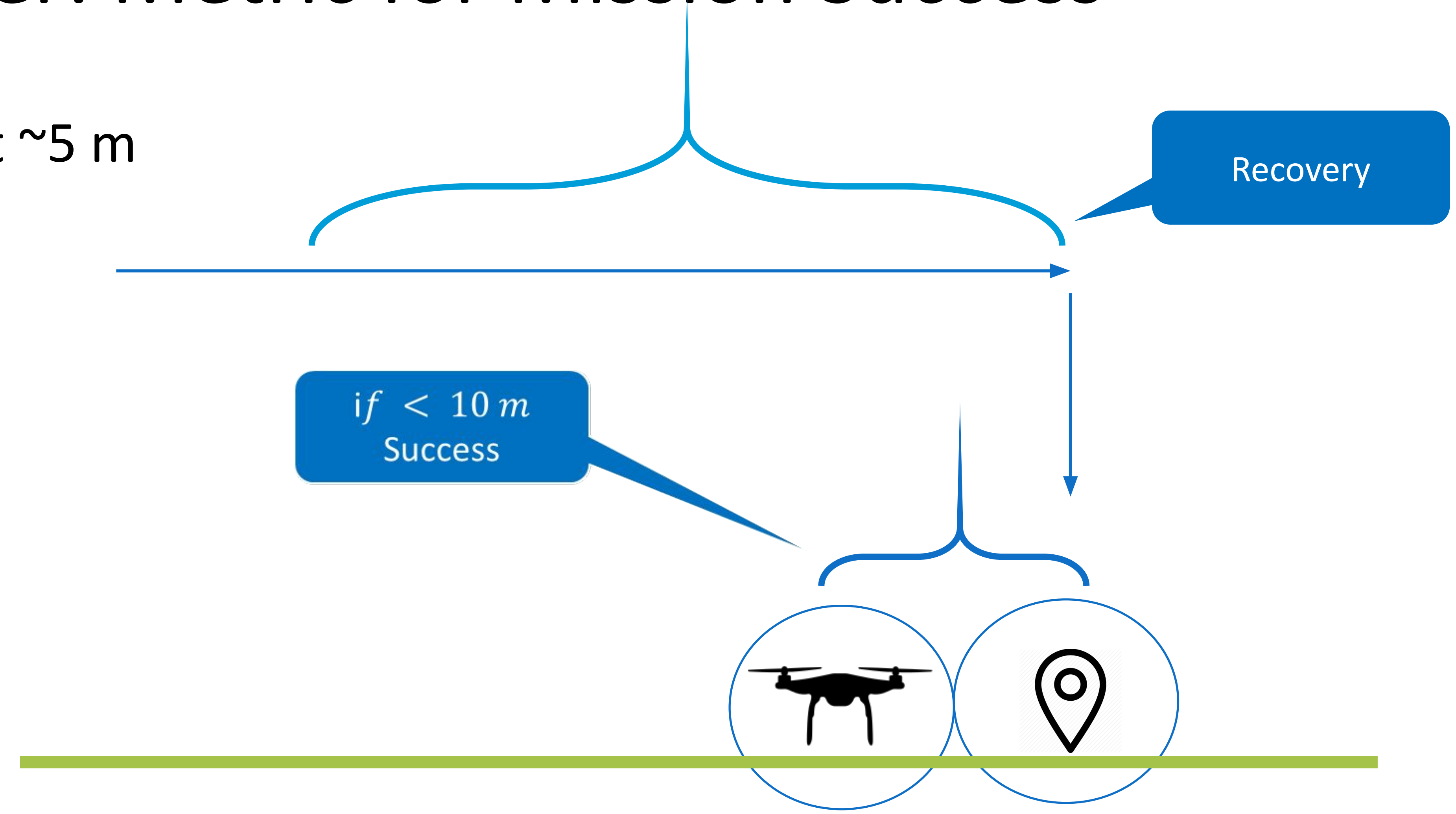
Experimental Setup

ARDUPILOT

PX4 autopilot

PID-Piper: 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: 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%**

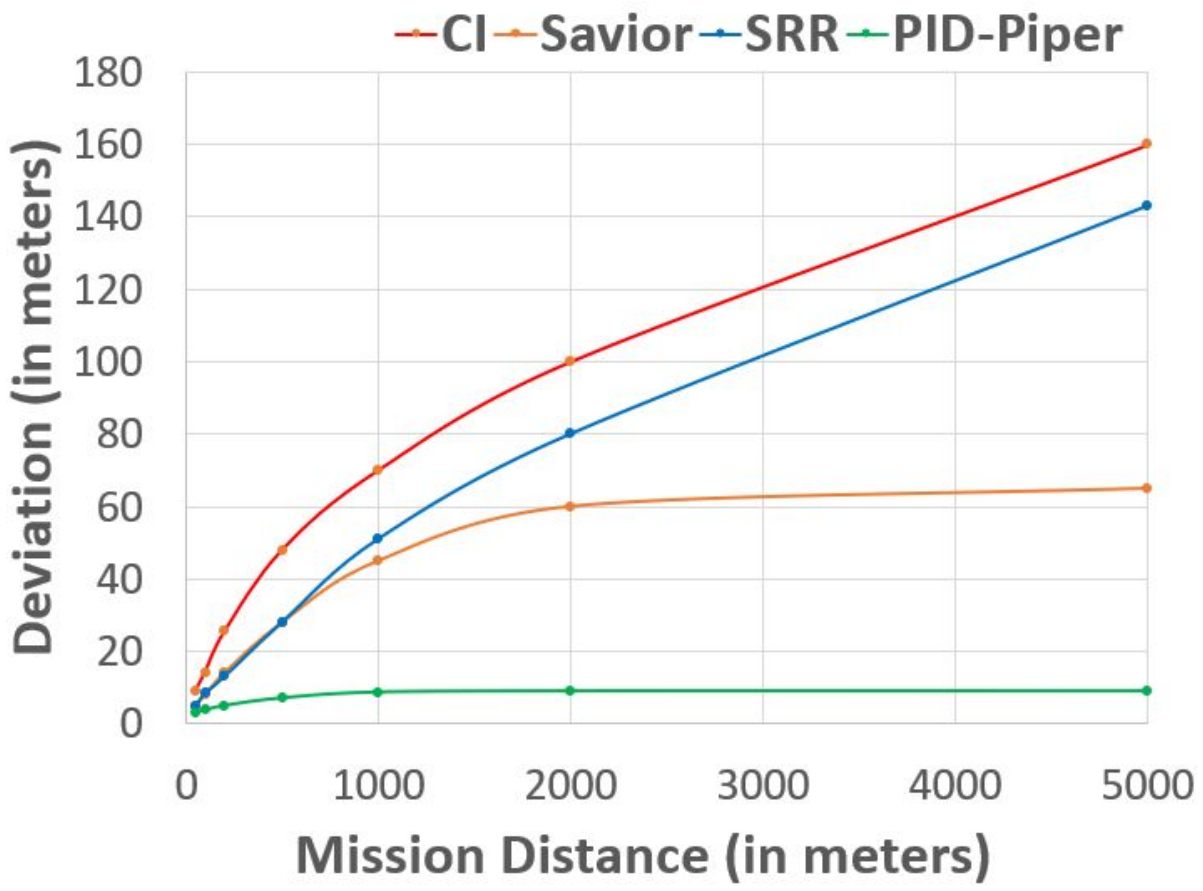
$$Mission\ Success = \frac{No.of\ missions\ with\ deviation < 10\ meters}{Total\ number\ of\ missions}$$

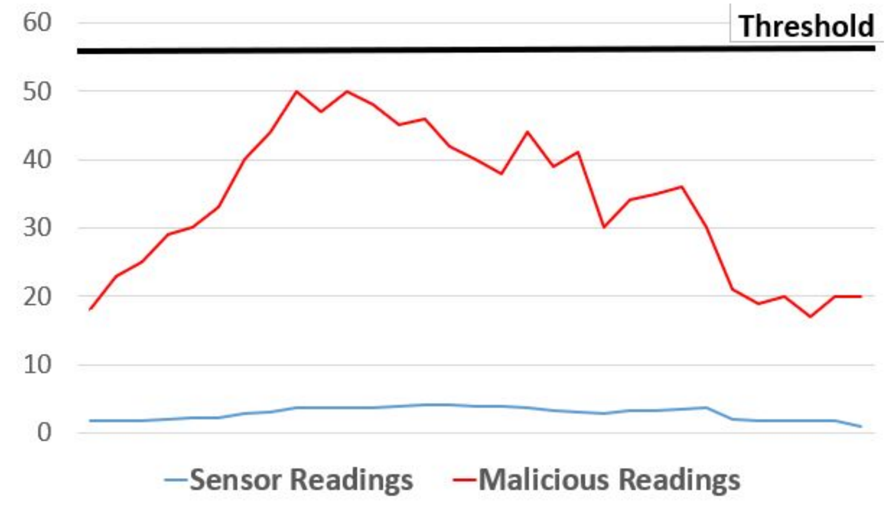
PID-Piper: Recovery under Attacks

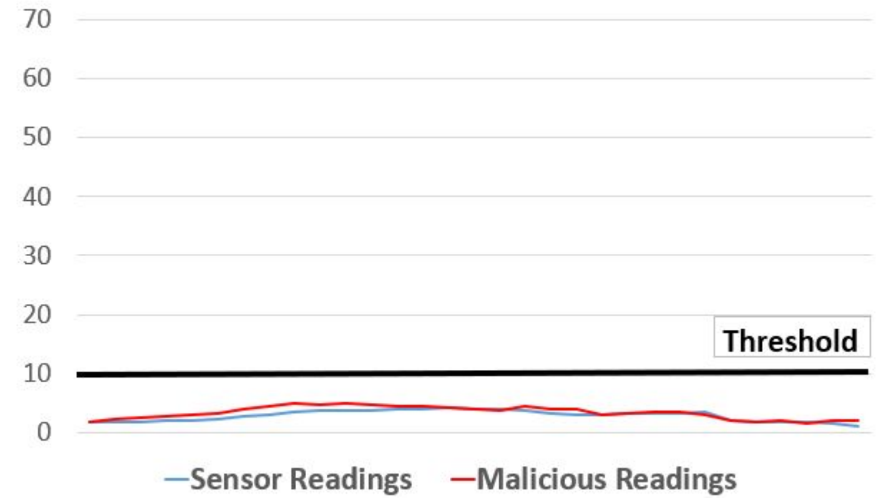
Analysis Type	SRR [RAID’20]	PID-Piper [This work]
Mission Success	13%	83%
Mission Failed (no Crash)	50%	17%
Crash/Stall	**37%**	**0%**

Recovery was successful in 83% of the cases with 0 crashes.

PID-Piper under Stealthy Attacks







PID-Piper: Overheads

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

PID-Piper: Summary

- **PID-Piper: A framework to recover Robotic Vehicles from attacks**
- Feed-forward Control to address overcompensation.
- 3 real and 3 simulated RV systems.
- 83% mission success from attacks, 0% false positives, limit stealthy attacks

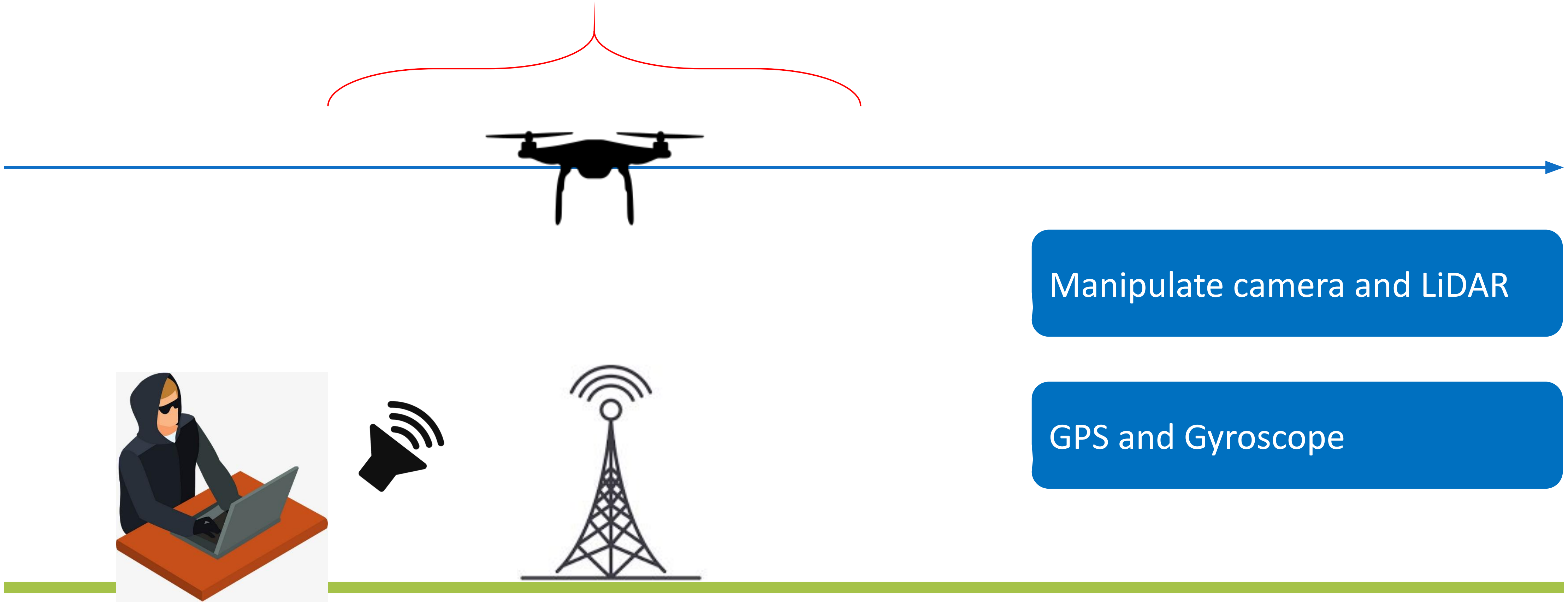
Videos

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

Pritam Dash, Guanpeng Li, Zitao Chen, Mehdi Karimibiuki, Karthik Pattabiraman, *PID-Piper: Recovering Robotic Vehicles from Physical Attacks,* DSN, 2021.

Best Paper Award.

DeLorean: Multiple Sensors under Attack

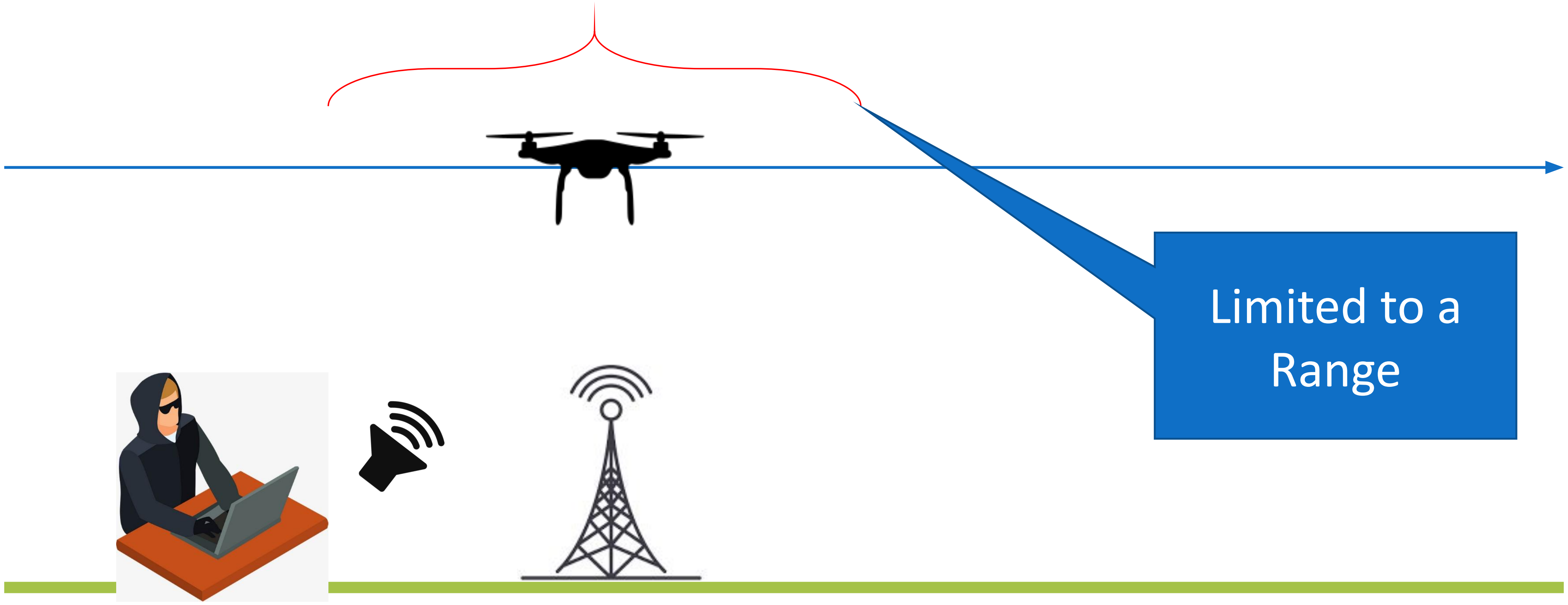


Manipulate camera and LiDAR

GPS and Gyroscope

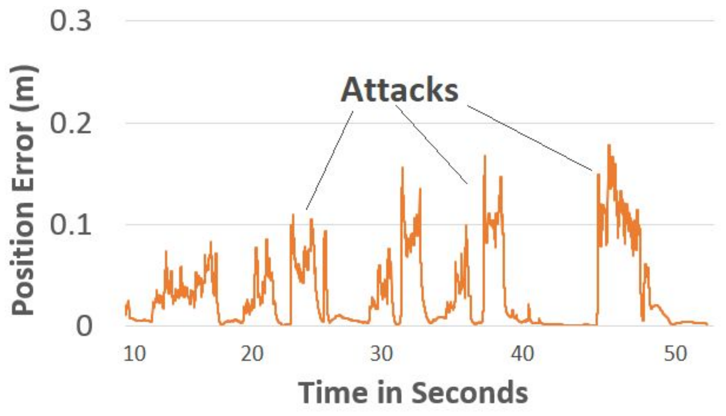
Cao et. al., Invisible to both Camera and Lidar, IEEE S&P 2021

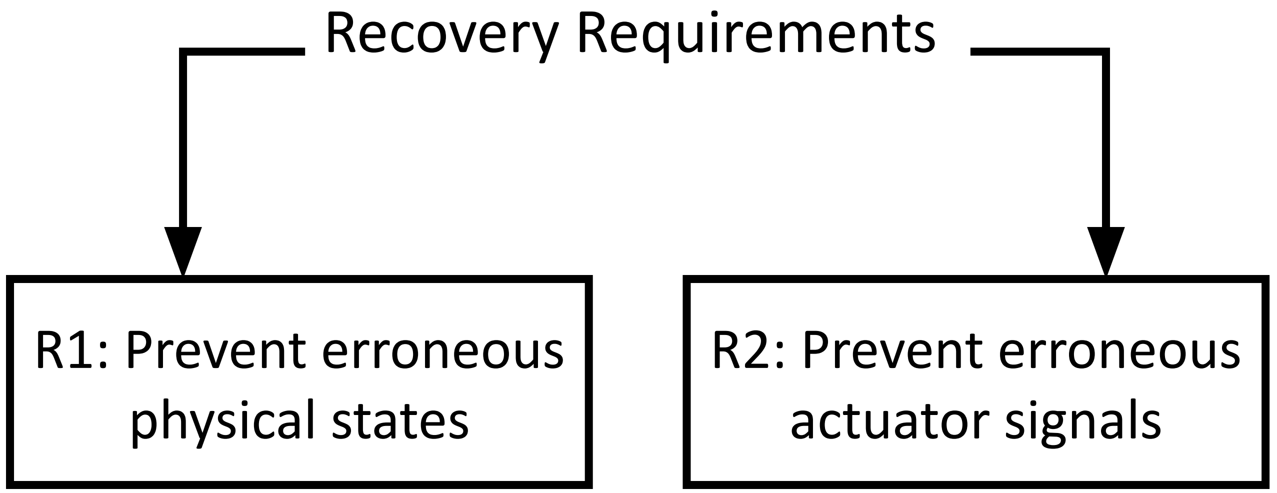
DeLorean: Threat Model



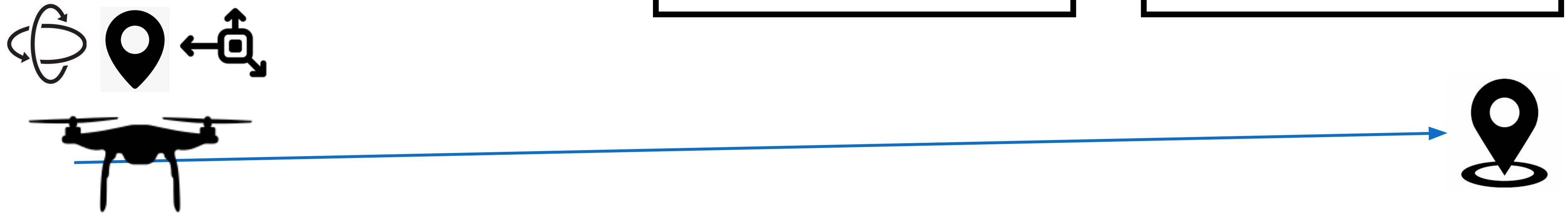
Cao et. al., Invisible to both Camera and Lidar, IEEE S&P 2021

DeLorean: Goal

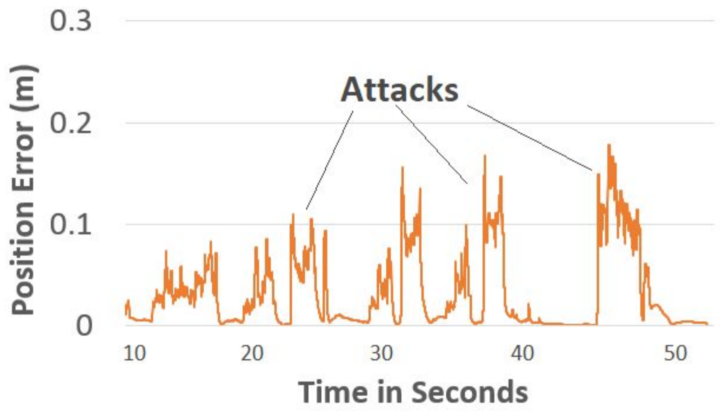


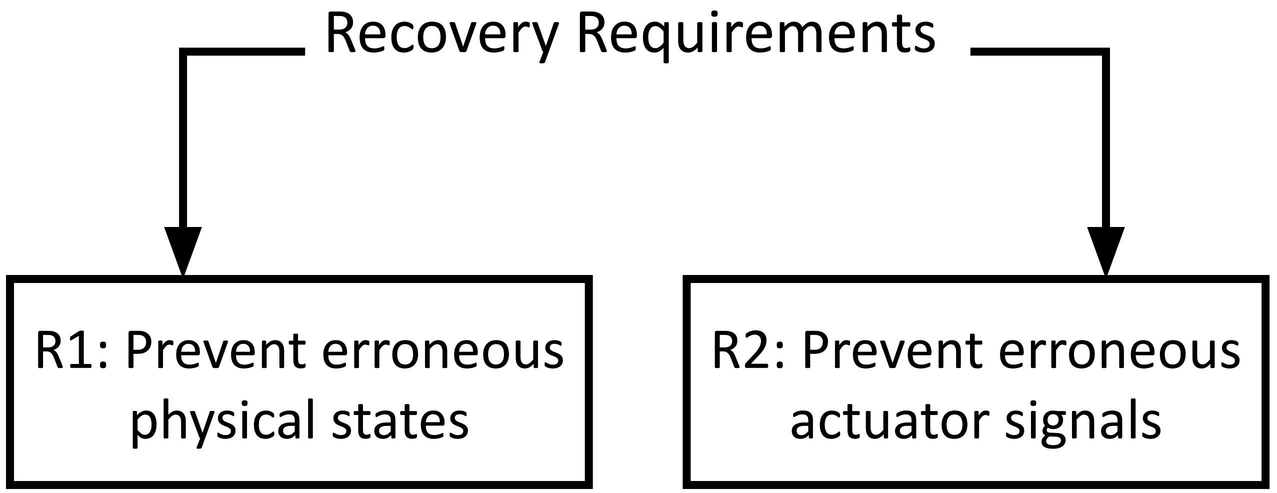


Recovery Requirements
R1: Prevent erroneous physical states
R2: Prevent erroneous actuator signals



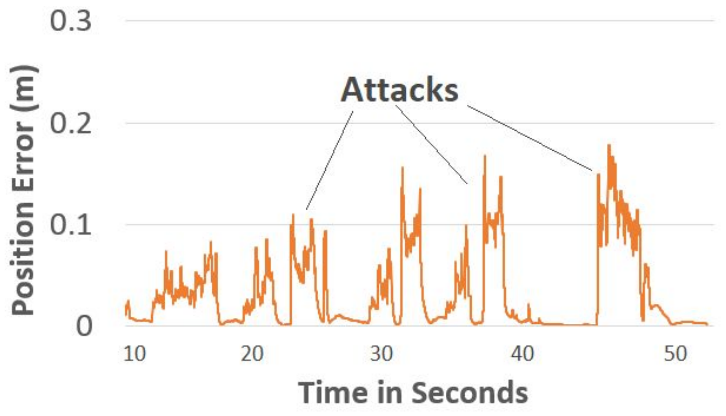
DeLorean: Identify the Sensor(s) under attack





Recovery Requirements
R1: Prevent erroneous physical states
R2: Prevent erroneous actuator signals

DeLorean: Isolate Sensor(s) from Control Process

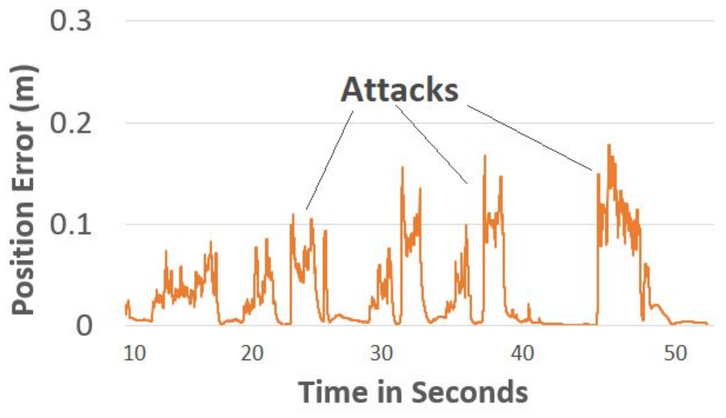


Recovery Requirements

R1: Prevent erroneous physical states

R2: Prevent erroneous actuator signals

DeLorean: Substitute Input Sequence

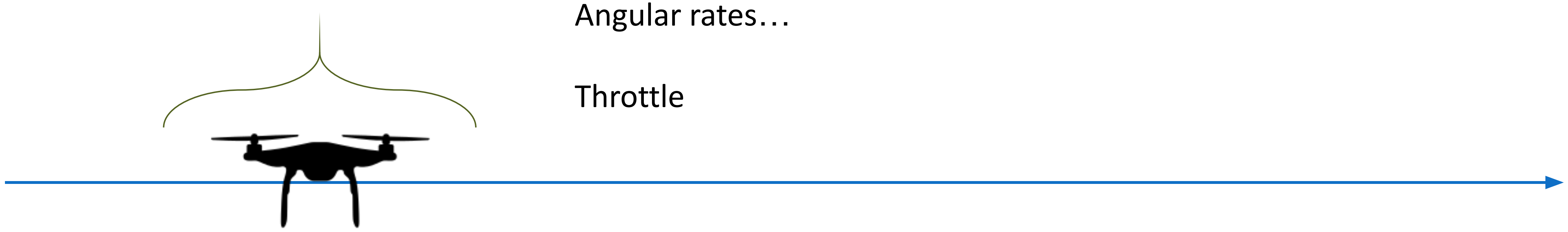


Recovery Requirements

R1: Prevent erroneous physical states

R2: Prevent erroneous actuator signals

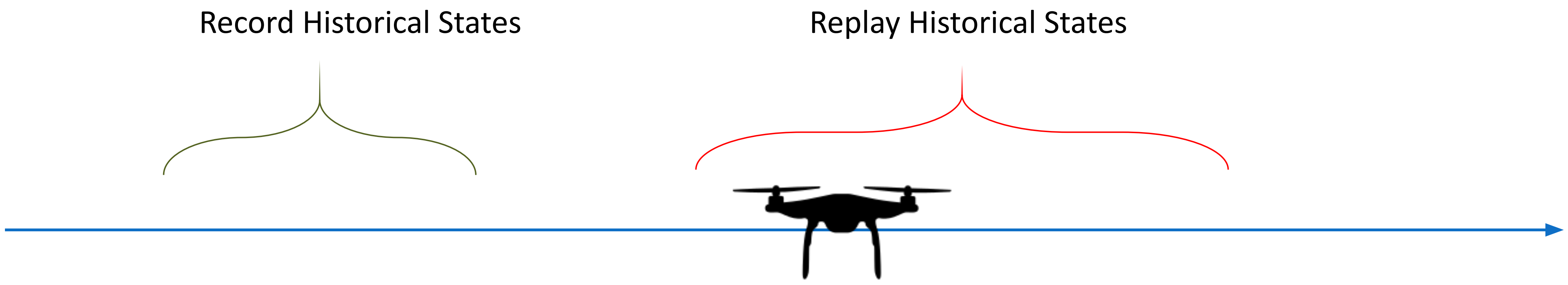
DeLorean: Substitute Input Sequence



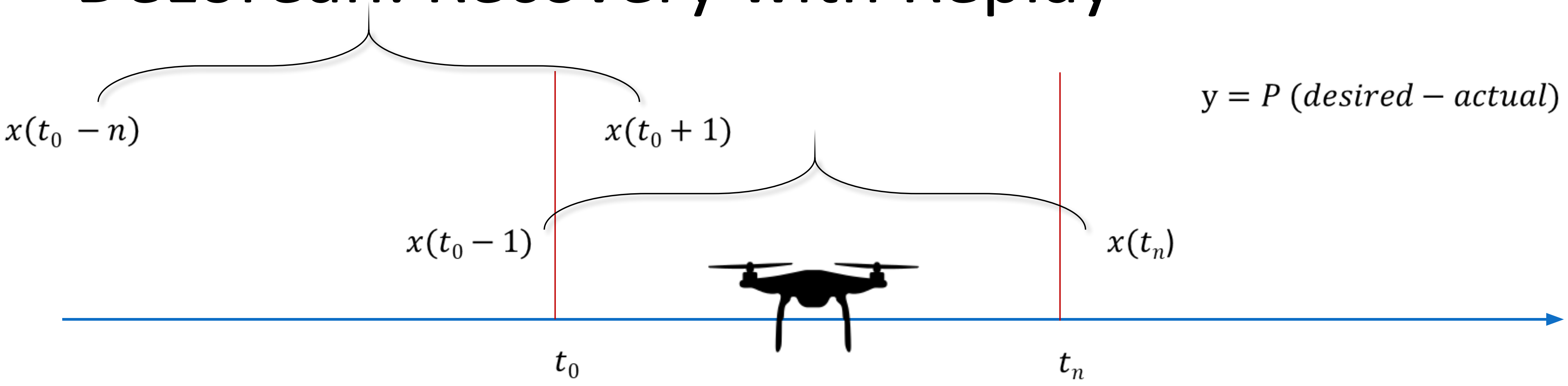
Position,
Velocity,
Angular rates…

Throttle

DeLorean: Substitute Input Sequence

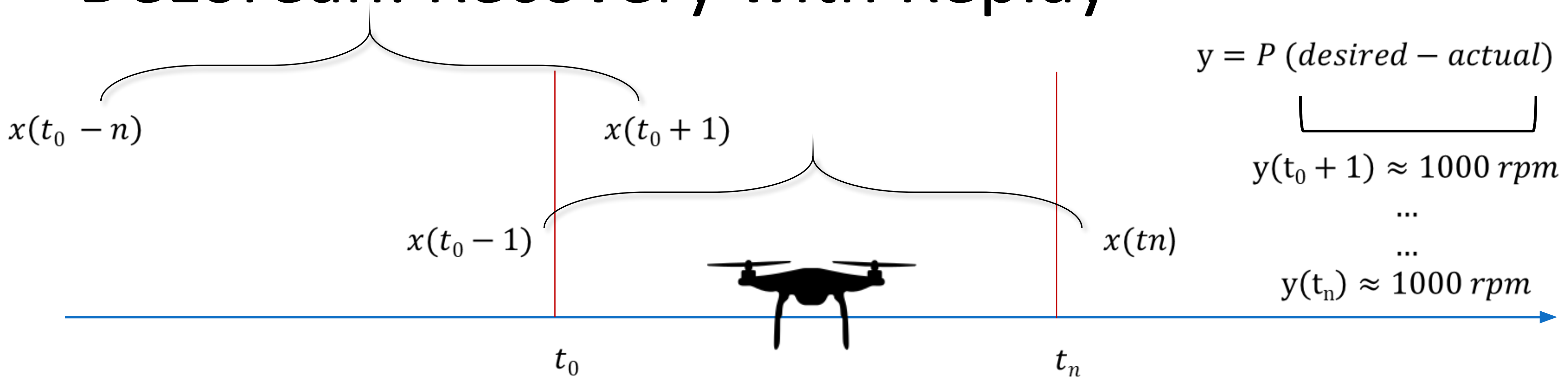


DeLorean: Recovery with Replay



Replay Historical States

DeLorean: Recovery with Replay



$x(t_0 - n)$
$x(t_0 + 1)$
$x(t_0 - 1)$
$x(tn)$
t_0
t_n
$y = P\ (\underbrace{desired - actual})$
$y(t_0 + 1) \approx 1000\ rpm$
...
...
$y(t_n) \approx 1000\ rpm$

Replay Historical States

Experimental Setup

ARDUPILOT

DeLorean: Mission Success Under Attacks (Percentage)

Nos. of attacked Sensors	SRR [RAID'20]	PID-Piper	DeLorean
1	64	100	100
2	20	20	100
3	0	0	100
4	0	0	88
5	0	0	82

DeLorean recovers the RVs in 94% of the cases overall (0 crashes).
82% mission success even under attacks targeting all the sensors.

DeLorean: Summary

DeLorean: A framework to recover RVs from multi-sensor attack.

- Replays historic states to recover from attacks: single & multi-sensor
- Evaluated in 4 real RVs, and 2 simulated RVs
- **94% mission success, 82% when all the sensors are under attack**
  - No other technique is able to recover from multi-sensor attacks beyond 2
- **Performance overhead: 7.5%, Energy overhead: 19%**

Under submission

Conclusion

Robotic Vehicles (RV) security is an important problem

- Used in many mission-critical and safety-critical settings
- Sensors can be modified/spoofed by attackers
- Need to ensure **mission success** despite attacks on RV

Two Techniques for recovering RVs from sensor attacks

- PID-Piper [DSN’21]   : Single-sensor, but persistent attacks
- DeLoRean[submitted]: Multiple-sensor, but localized attacks
- Future work: Recovering RV platoons/drone swarms from attacks