# The Nuts and Bolts of Deploying Process-Level IDS in Industrial Control Systems

Magnus Almgren
Chalmers University

Wissam Aoudi
Chalmers University

Robert Gustafsson
Chalmers University

Robin Krahl
University of Freiburg

Andreas Lindhé
Combitech

CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

## Industrial Control Systems (ICS)

- control industrial processes;
- typically operate on critical infrastructures.

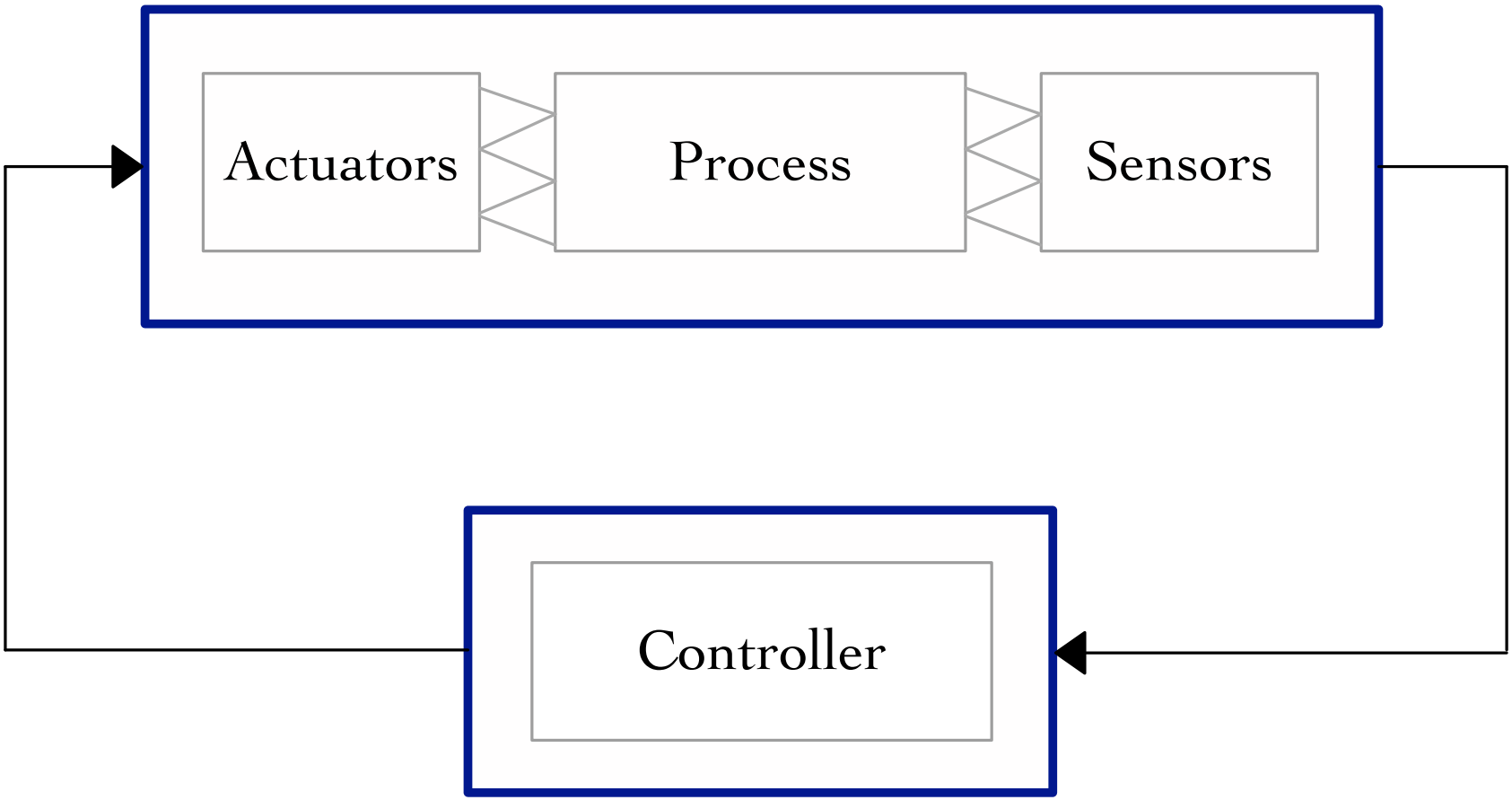
## The Problem

- Attacks on ICS are increasing.
- Successful attacks on ICS
  - can be highly rewarding for attackers;
  - may have far-reaching consequences on society at large.
- Classical IT-based security is not sufficient.

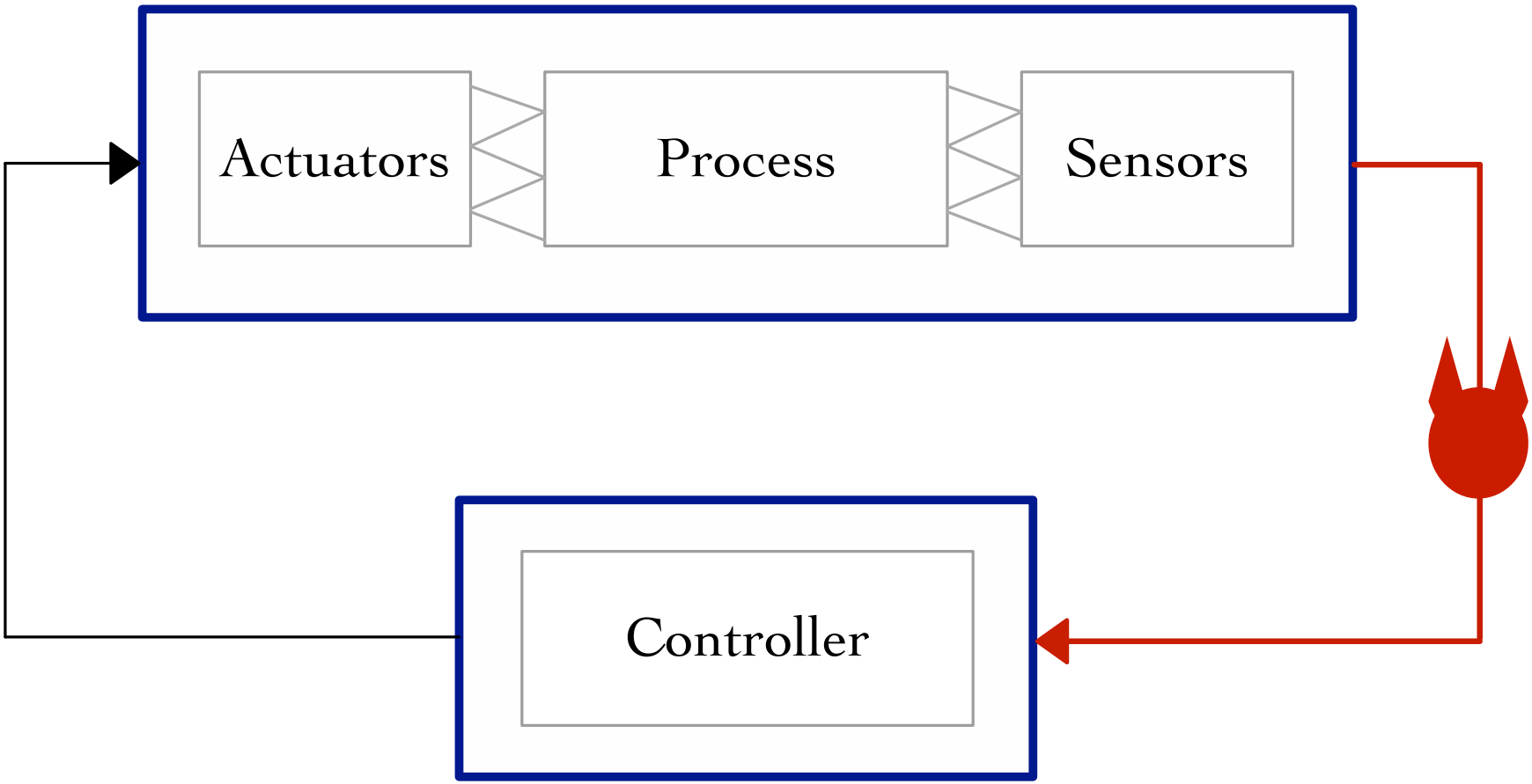
## Process-Level Attack Detection

| | |
|---|---|
| **Why?** | Because ICS combine both IT and OT technologies. |
| **What?** | Check if **physical process** deviates from **the norm**. |
| **How?** | By monitoring **process output** in real time. |

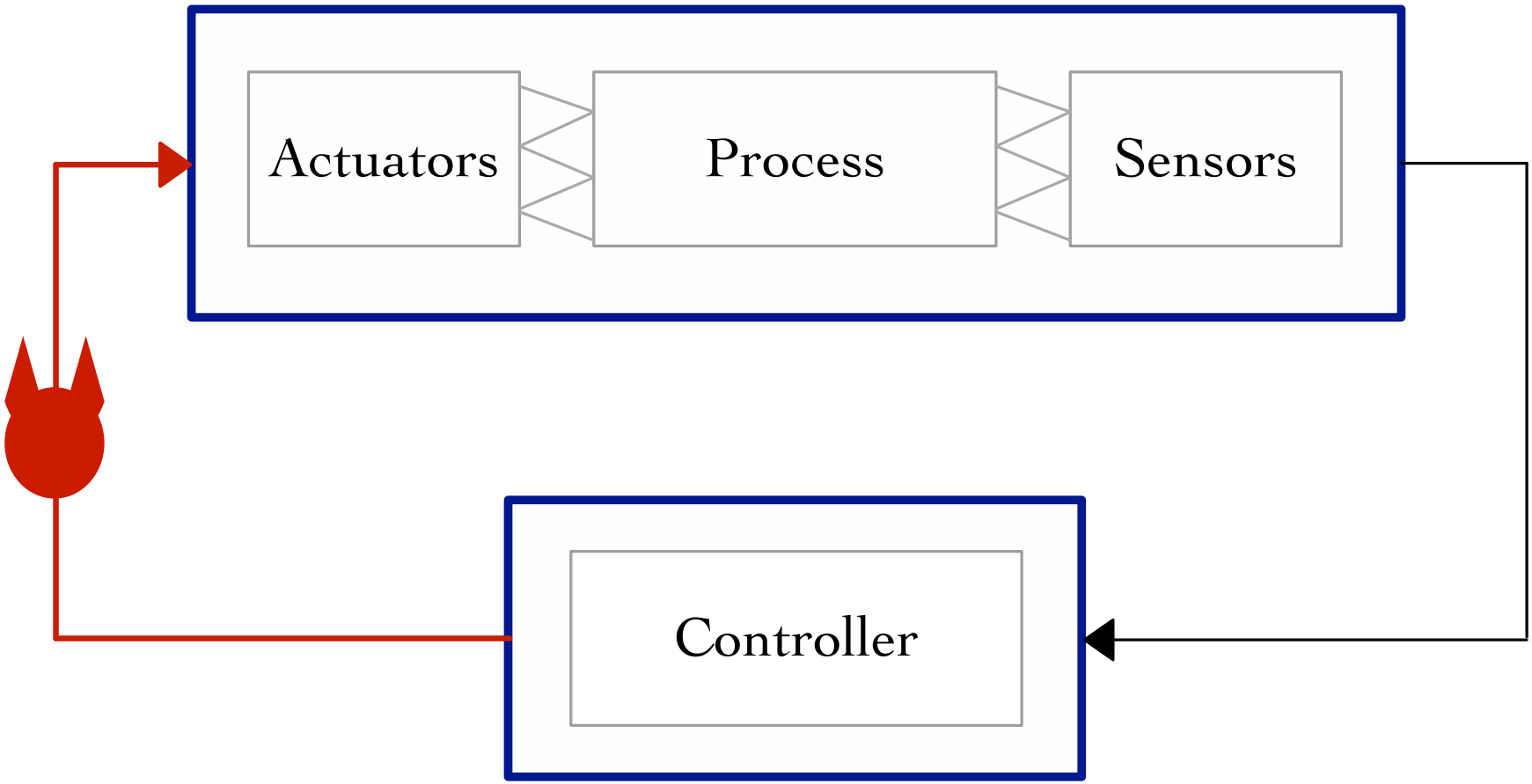
# Control Loop and Attacker Model



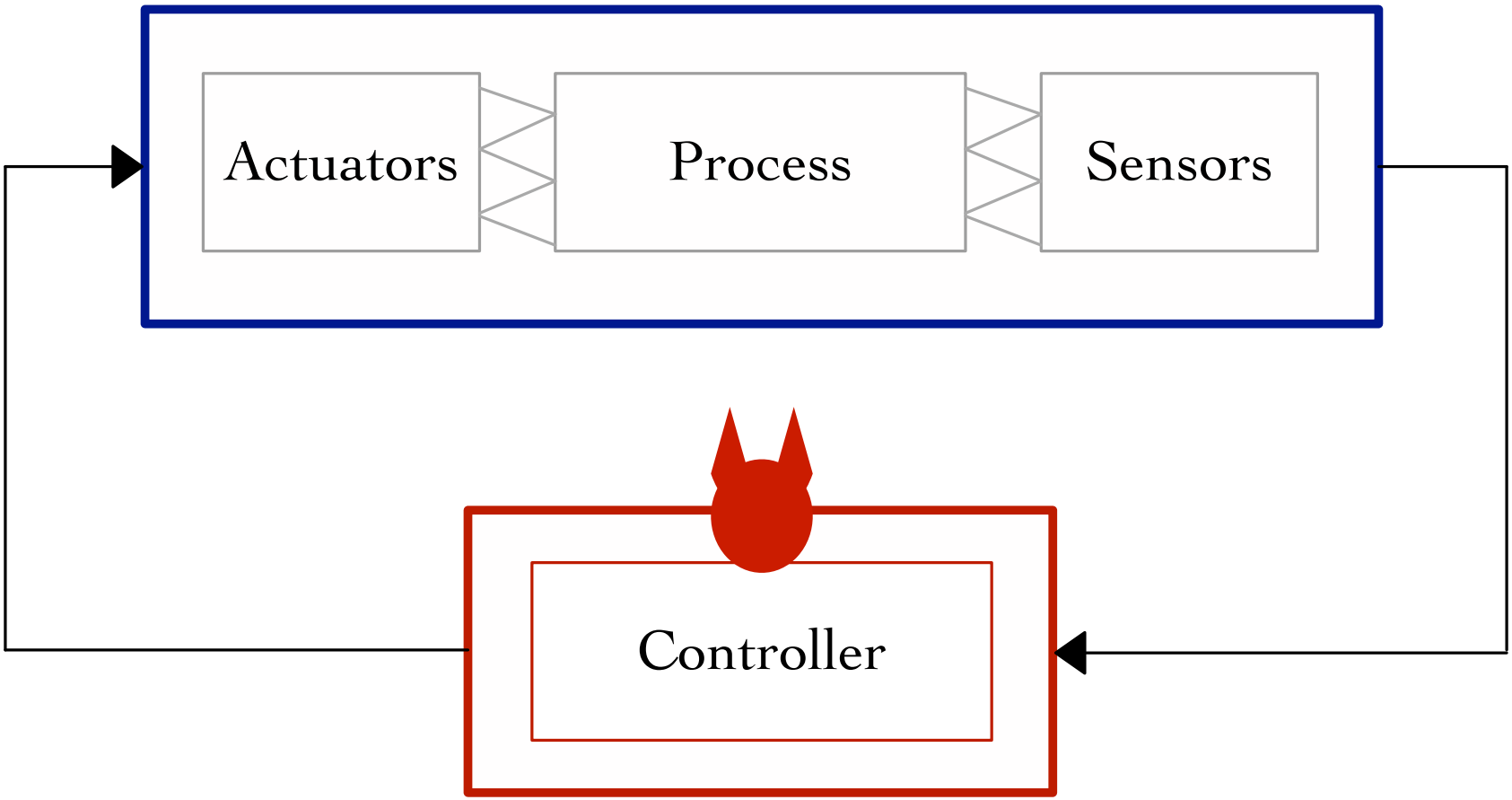
# Control Loop and Attacker Model



# Control Loop and Attacker Model



# Control Loop and Attacker Model



# ICS behavior is deterministic

## ICS-Specific Features

- Controllers (e.g., PLCs) operate in a **cyclic** manner.
- Signals repeat $\Rightarrow$ level of **determinism** is relatively high.
- Normal behavior can be **learned** or **modeled**.

# ICS behavior is deterministic

## ICS-Specific Features

- Controllers (e.g., PLCs) operate in a **cyclic** manner.
- **Regularity of ICS behavior enables data-driven approaches.**
- Normal behavior can be **learned** or **modeled**.

## Classical Approach

**Build** a model of the physical process

↓

Use the model to **predict** future system behavior

↓

Monitor residuals: Is $|$**observed** $-$ **predicted**$|$ too large?

---

Urbina, David I., et al. "Limiting the Impact of Stealthy Attacks on Industrial Control Systems." 2016 ACM Conference on Computer and Communications Security.

## Classical Approach

**Build** a model of the physical process

↓

**Use the model to predict future system behavior**

*Solving a more general problem as an intermediate step!*

↓

Monitor residuals: Is $|\textbf{observed} - \textbf{predicted}|$ too large?

---

Urbina, David I., et al. "Limiting the Impact of Stealthy Attacks on Industrial Control Systems." 2016 ACM Conference on Computer and Communications Security.

CHALMERS

## PASAD

1. solves an easier problem;
2. requires limited knowledge of system dynamics;
3. is capable of detecting subtle changes in system behavior.

---

Wissam Aoudi, Mikel Iturbe, and Magnus Almgren. "Truth Will Out: Departure-Based Process-Level Detection of Stealthy Attacks on Control Systems." 2018 ACM SIGSAC Conference on Computer and Communications Security.

# PASAD

1. solves an easier problem:

**Learns** normal behavior from historical data

↓

Measures to what extent **present** readings **conform** with the estimated dynamics.

---

Wissam Aoudi, Mikel Iturbe, and Magnus Almgren. "Truth Will Out: Departure-Based Process-Level Detection of Stealthy Attacks on Control Systems." 2018 ACM SIGSAC Conference on Computer and Communications Security.

# PASAD

1. solves an easier problem:

**Learns** normal behavior from historical data

**No need to predict the future!**

Measures to what extent **present** readings **conform** with the estimated dynamics.

Wissam Aoudi, Mikel Iturbe, and Magnus Almgren. "Truth Will Out: Departure-Based Process-Level Detection of Stealthy Attacks on Control Systems." 2018 ACM SIGSAC Conference on Computer and Communications Security.

CHALMERS

## PASAD

2. requires limited knowledge of system dynamics:

- It is entirely data-driven.
- Uses only **raw** sensor readings.
- It is model-free.

---

Wissam Aoudi, Mikel Iturbe, and Magnus Almgren. "Truth Will Out: Departure-Based Process-Level Detection of Stealthy Attacks on Control Systems." 2018 ACM SIGSAC Conference on Computer and Communications Security.

CHALMERS

# PASAD

2. requires limited knowledge of system dynamics:

- It is entirely
- Uses only
- It is model-free.

**PASAD is specification-agnostic.**
**Applicable to various systems.**

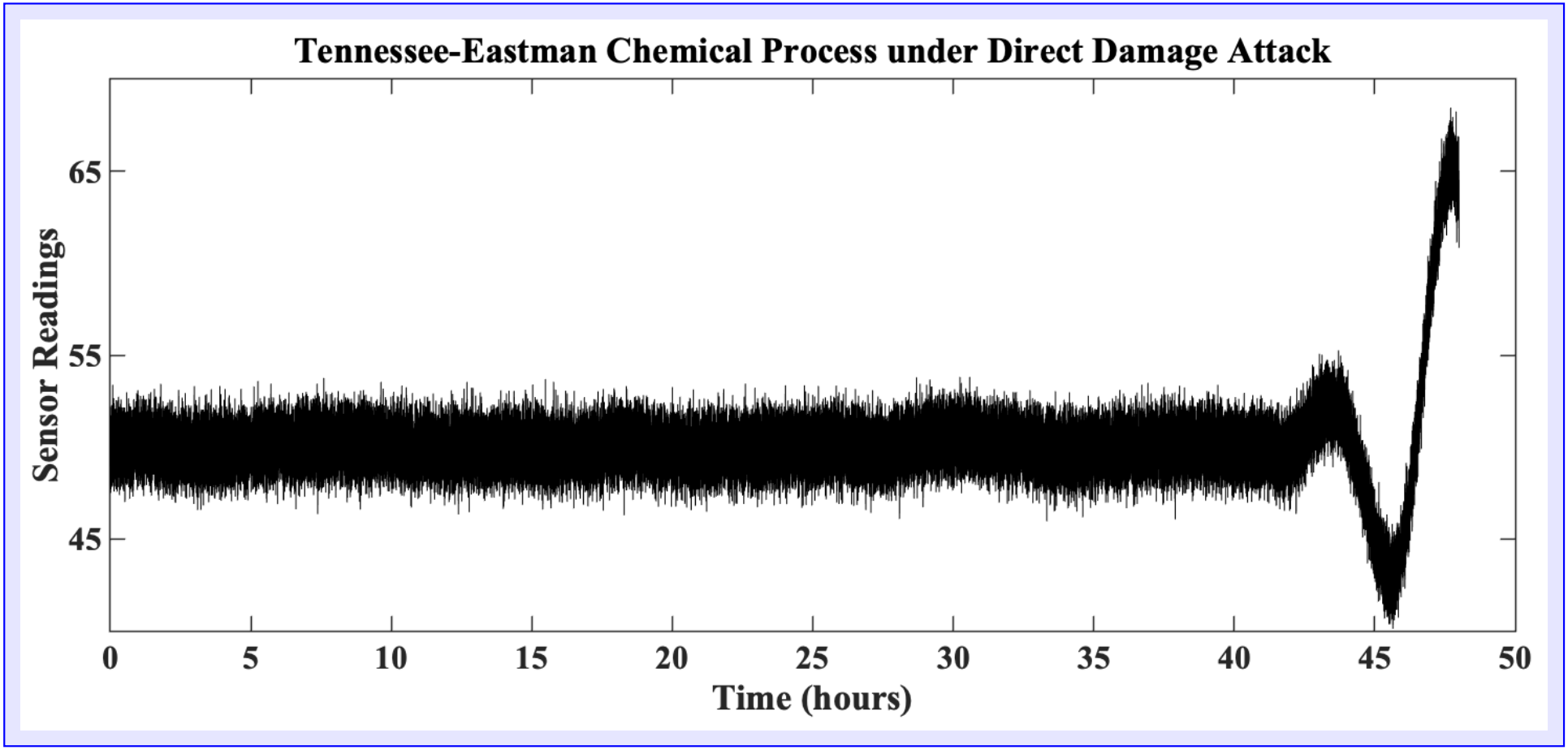
---

Wissam Aoudi, Mikel Iturbe, and Magnus Almgren. "Truth Will Out: Departure-Based Process-Level Detection of Stealthy Attacks on Control Systems." 2018 ACM SIGSAC Conference on Computer and Communications Security.

CHALMERS

# PASAD

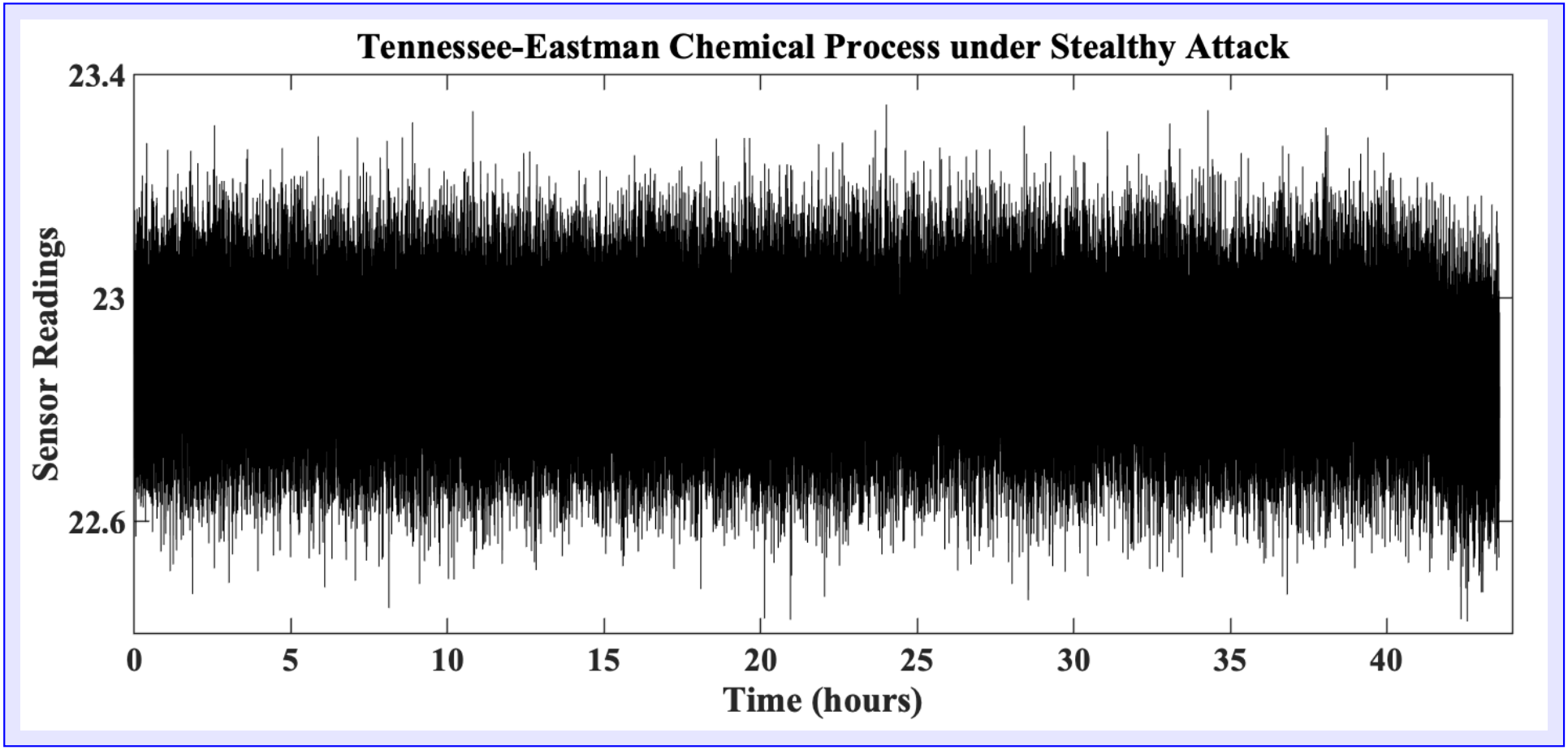
3. is capable of detecting subtle changes in system behavior:



CHALMERS

# PASAD

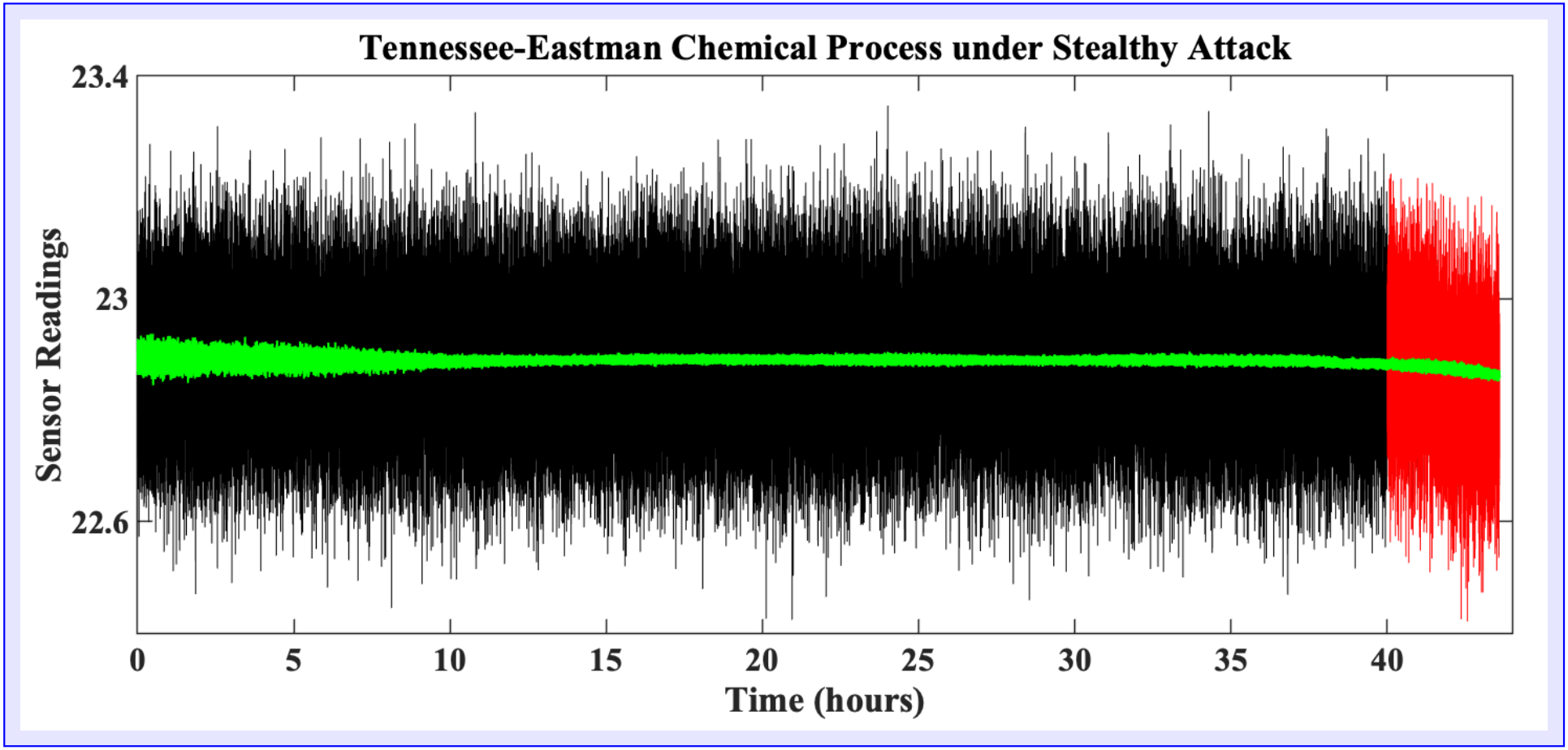
3. is capable of detecting subtle changes in system behavior:



CHALMERS

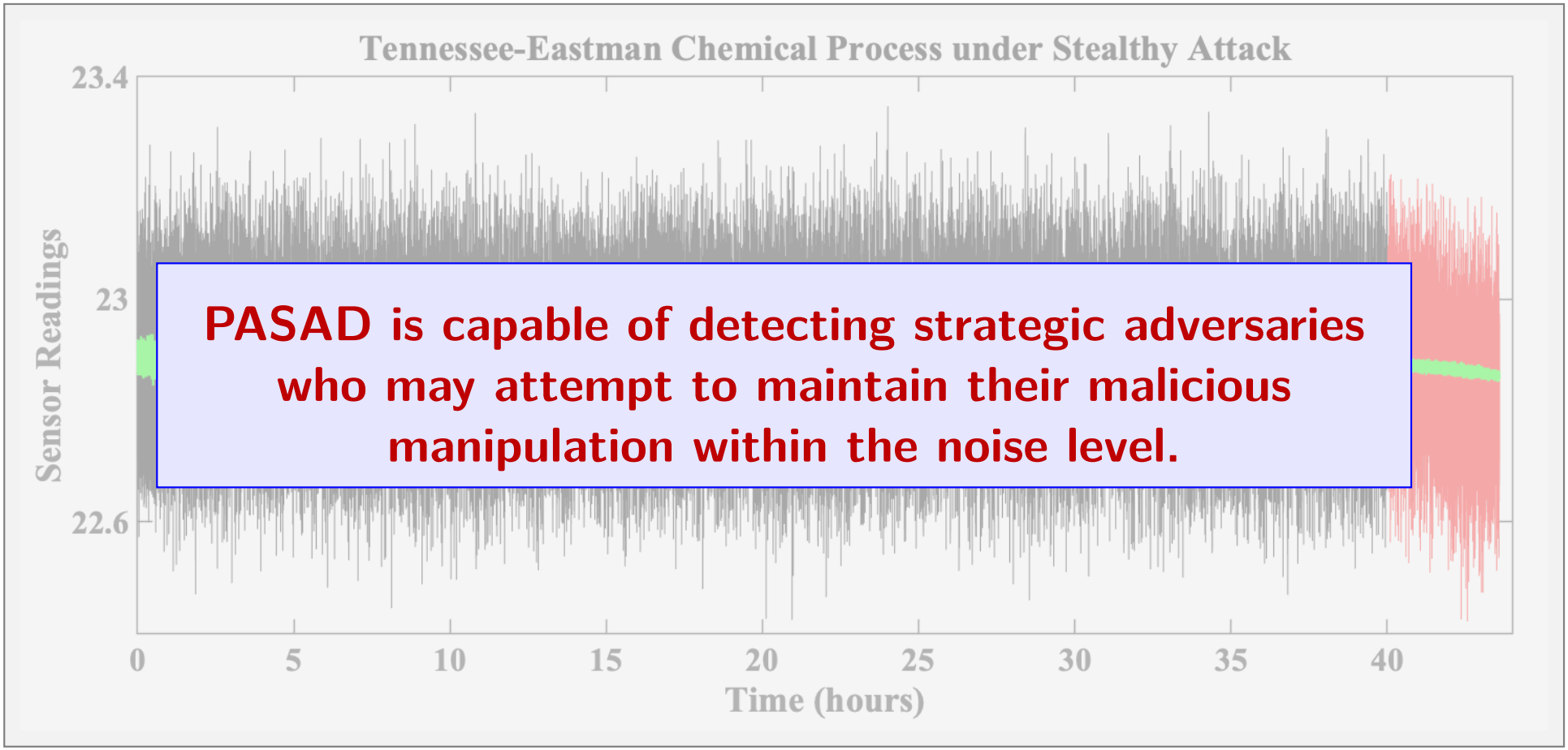
# PASAD

3. is capable of detecting subtle changes in system behavior:



# PASAD

3. is capable of detecting subtle changes in system behavior:



# PASAD — Process-Aware Stealth-Attack Detection

**Rationale:** Detect attacks on ICS by monitoring sensor measurements for unusual behavior.

PASAD works in two phases: *Offline learning* and *online detection*.

# PASAD — Process-Aware Stealth-Attack Detection

**Rationale:** Detect attacks on ICS by monitoring sensor measurements for unusual behavior.

PASAD works in two phases: *Offline learning* and *online detection*.

**Learning Phase:** **Create a mathematical representation of the *norm***

- Extract noise-reduced signal information from noisy time series of sensor readings.
- Construct *Signal Subspace* and project training vectors.
- Compute centroid of the cluster formed by training vectors.

# PASAD — Process-Aware Stealth-Attack Detection

**Rationale:** Detect attacks on ICS by monitoring sensor measurements for unusual behavior.

PASAD works in two phases: *Offline learning* and *online detection*.

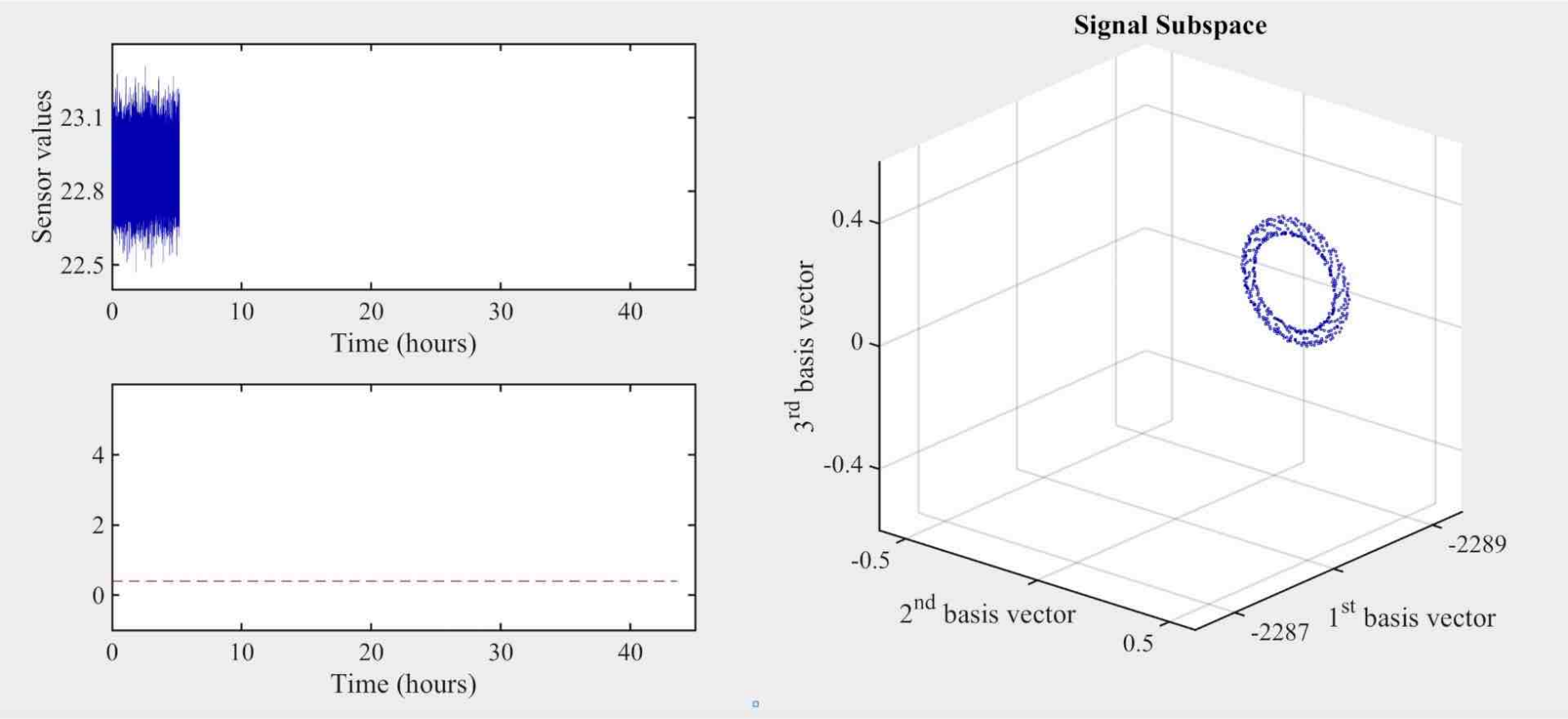
**Learning Phase:** **Create a mathematical representation of the *norm***

- Extract noise-reduced signal information from noisy time series of sensor readings.
- Construct *Signal Subspace* and project training vectors.
- Compute centroid of the cluster formed by training vectors.

**Detection Phase:** **Track distance from the centroid**

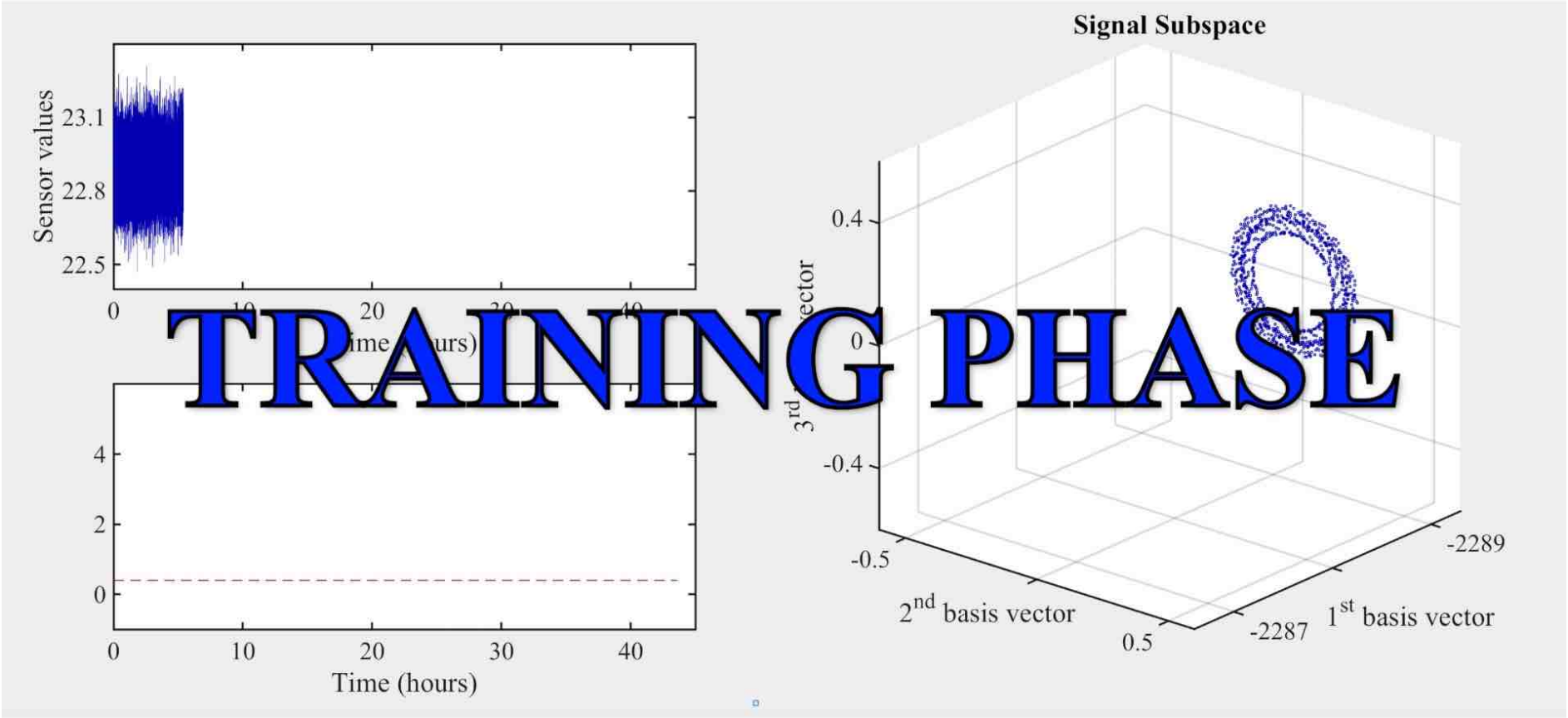
- Project most recent measurement vector onto the subspace.
- Compute a *departure score*: distance from the centroid.
- Raise an alarm if a certain threshold is crossed.

# Validation — Visualizing the Departure



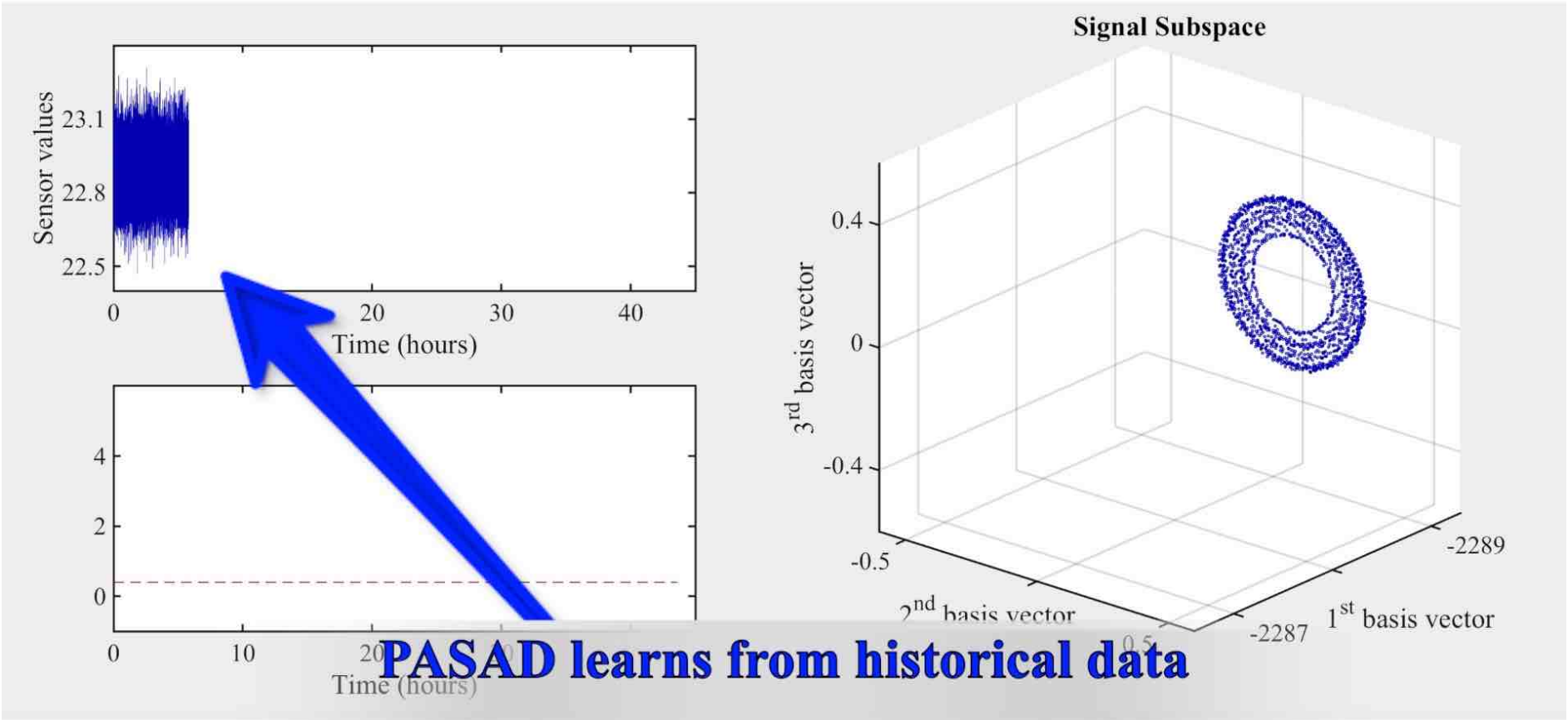

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



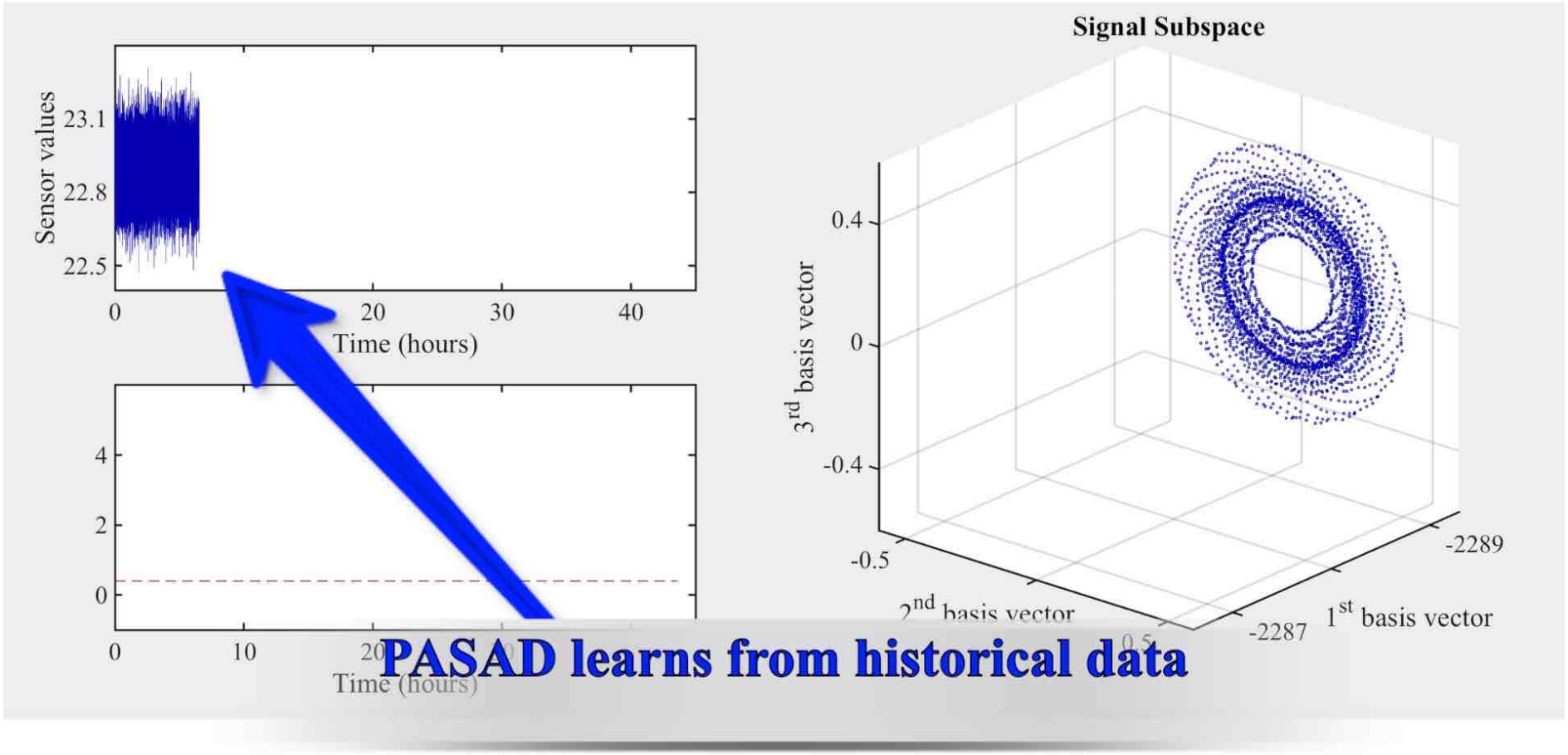

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



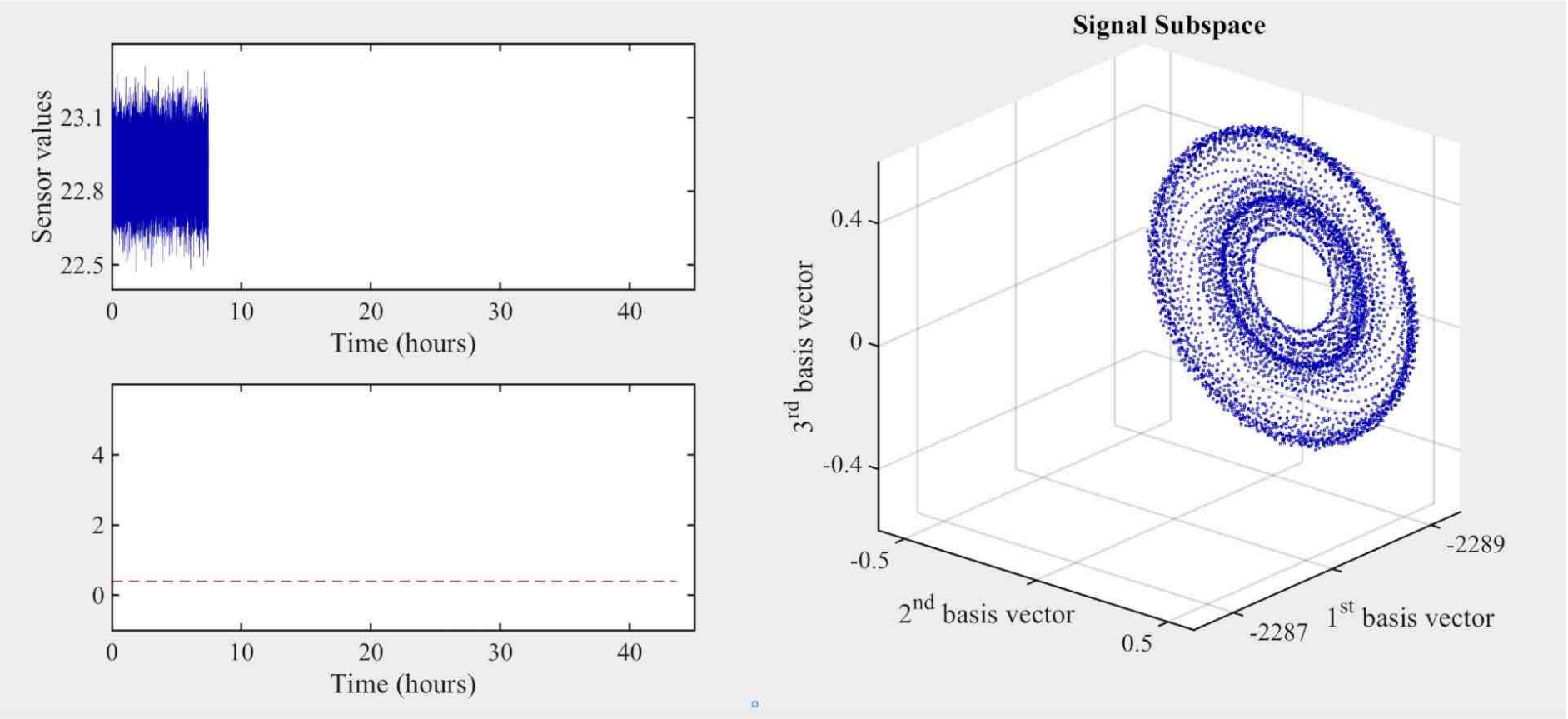

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



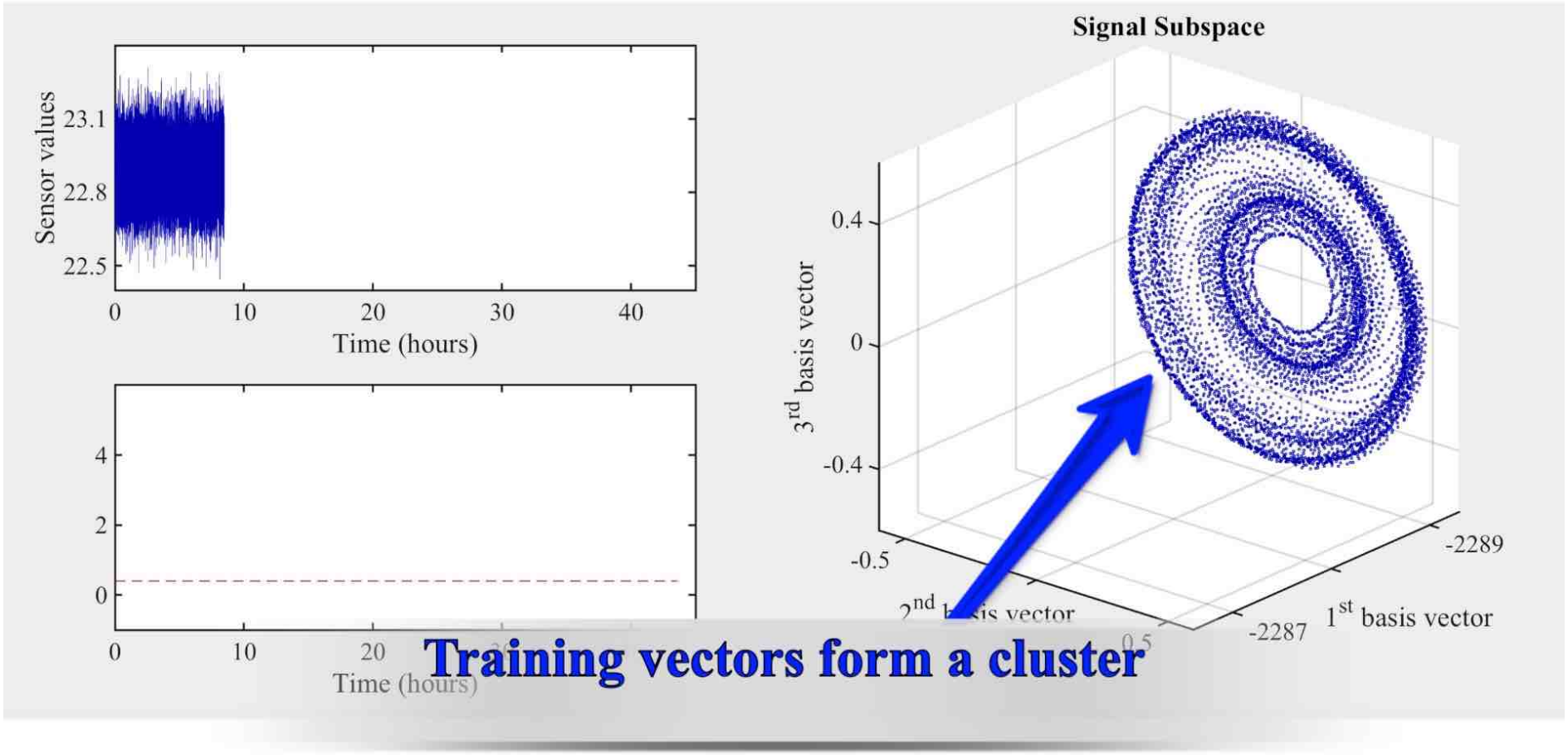

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



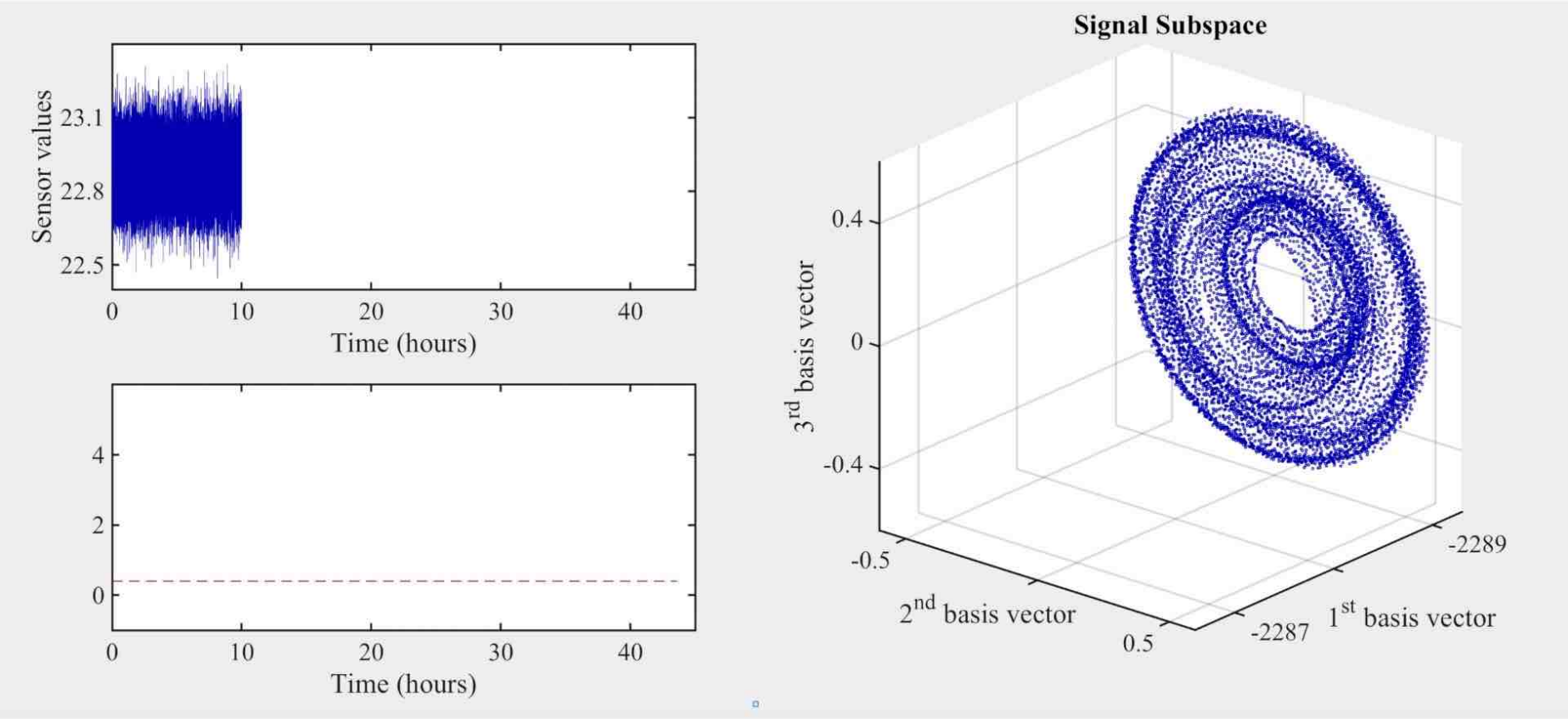

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



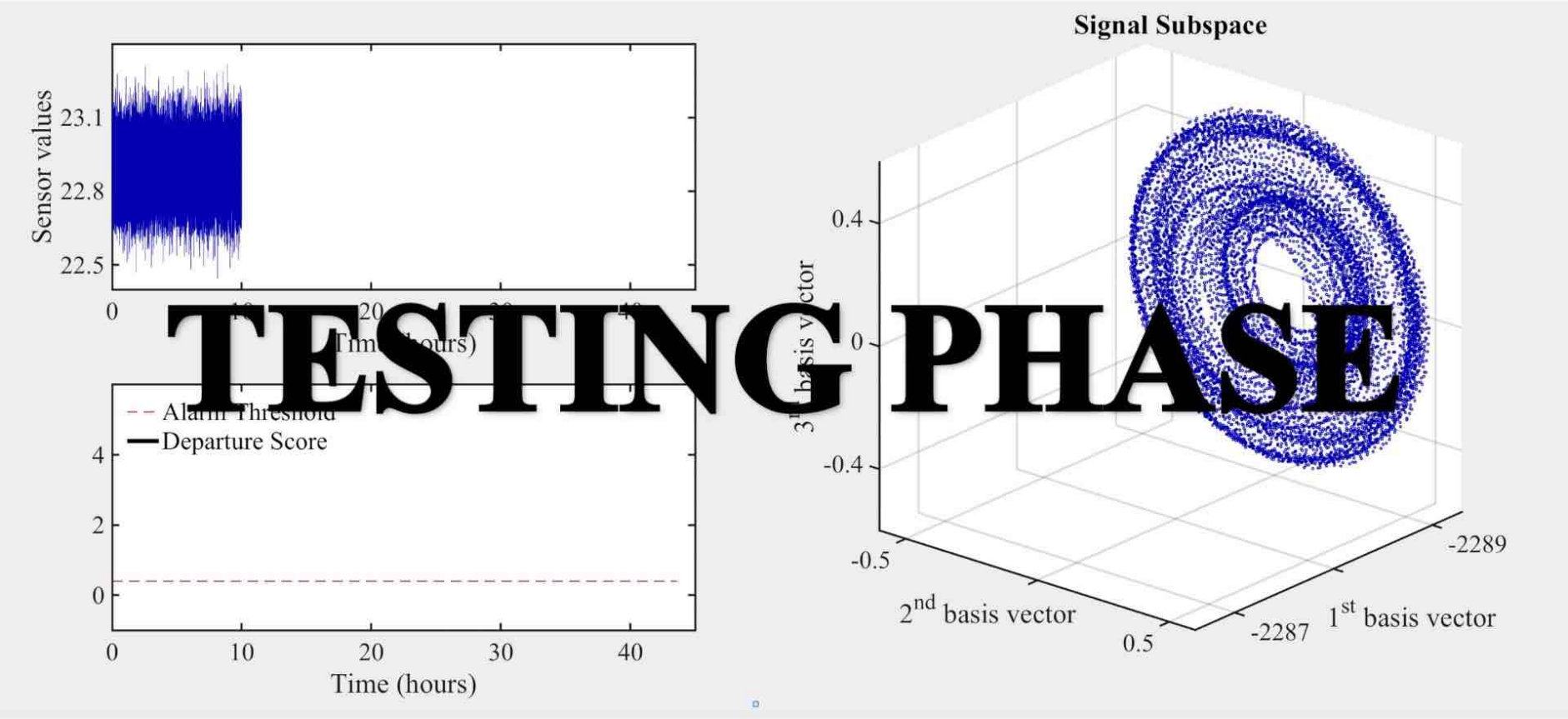

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



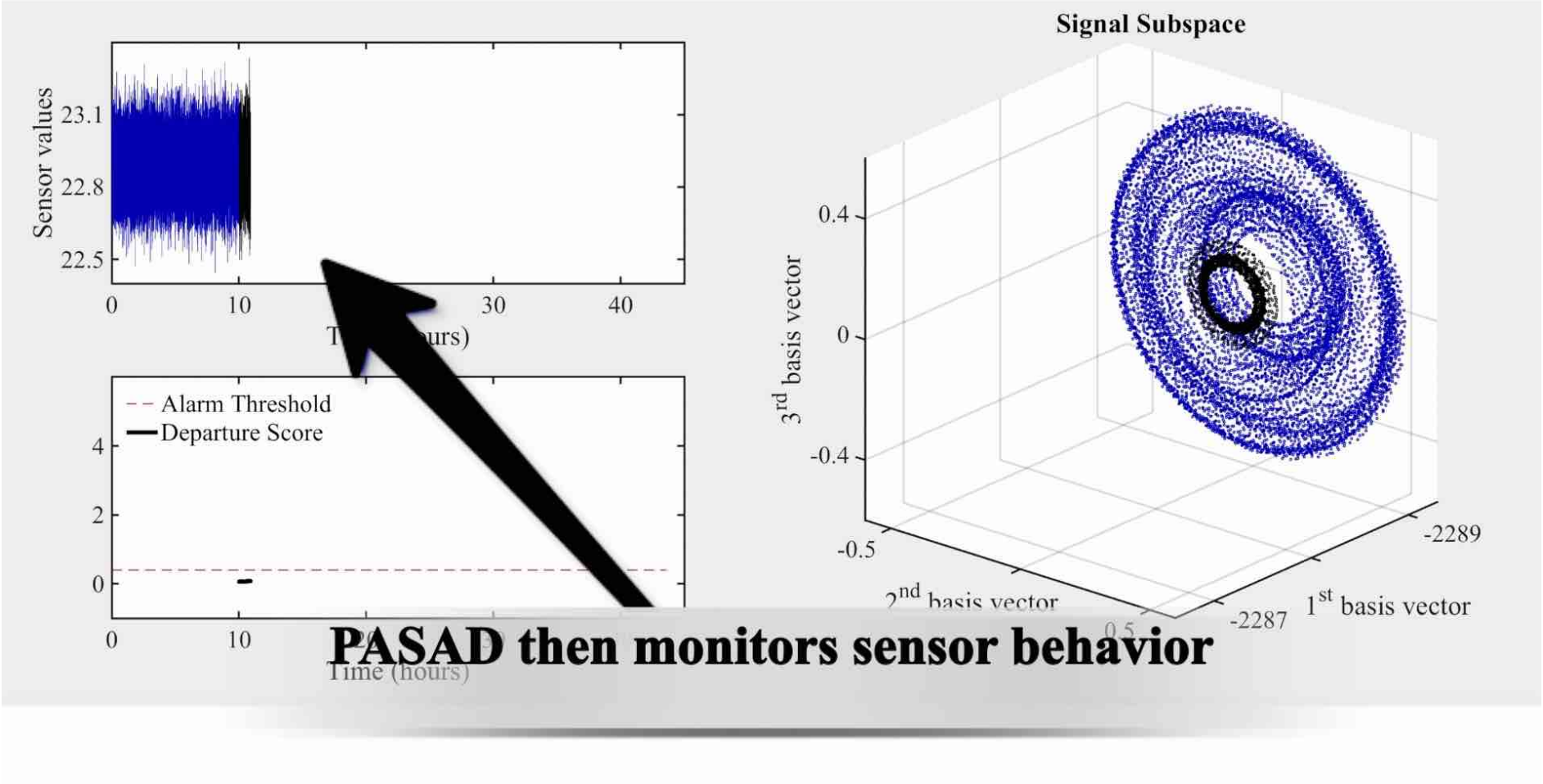

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



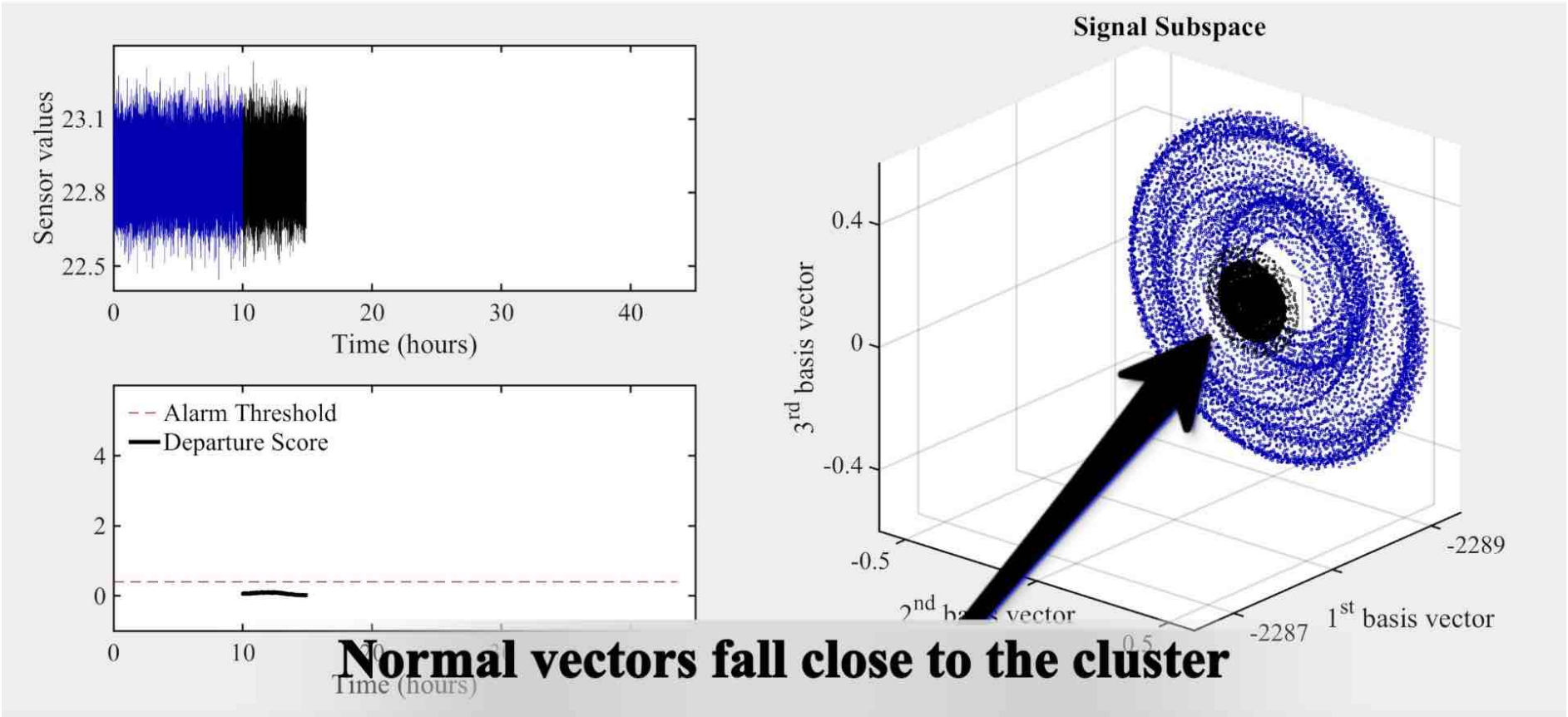

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



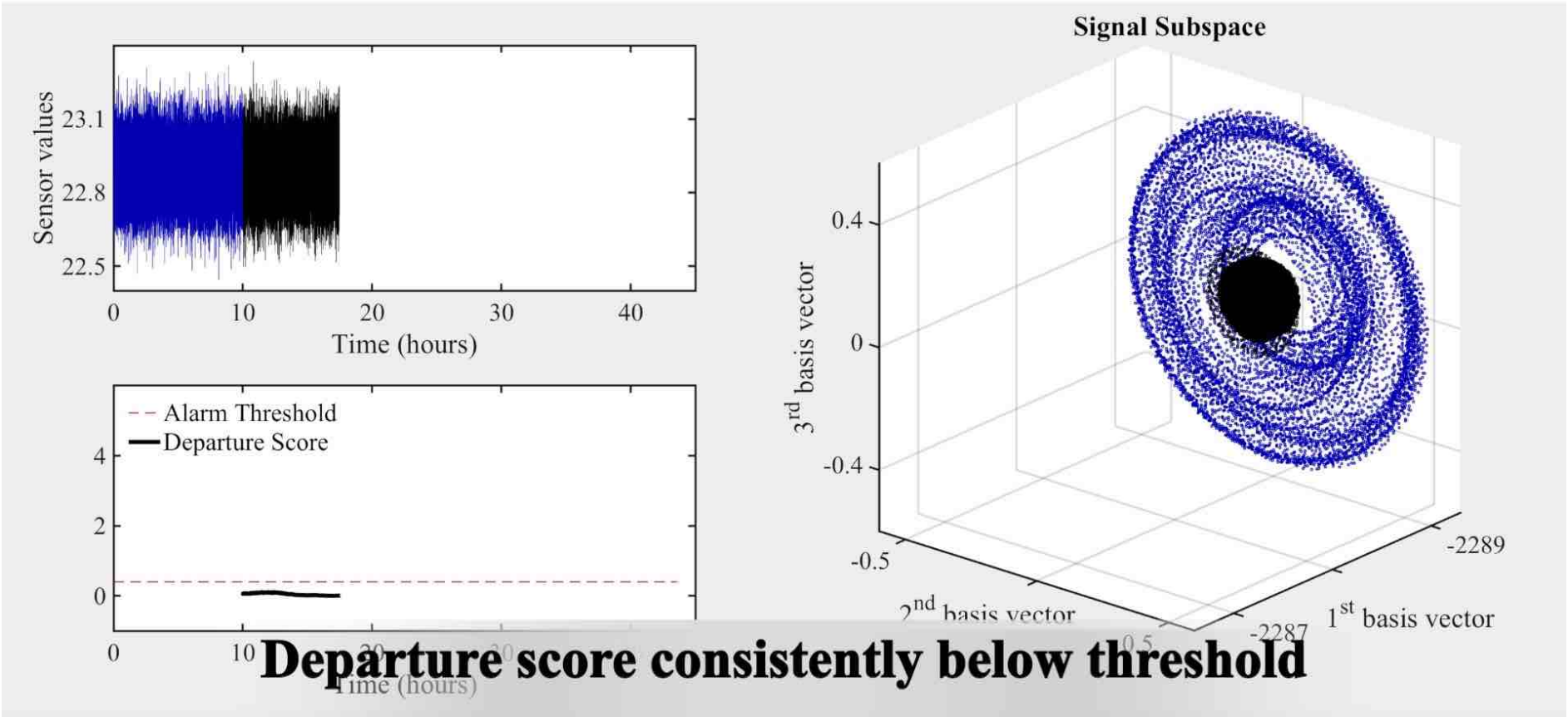

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



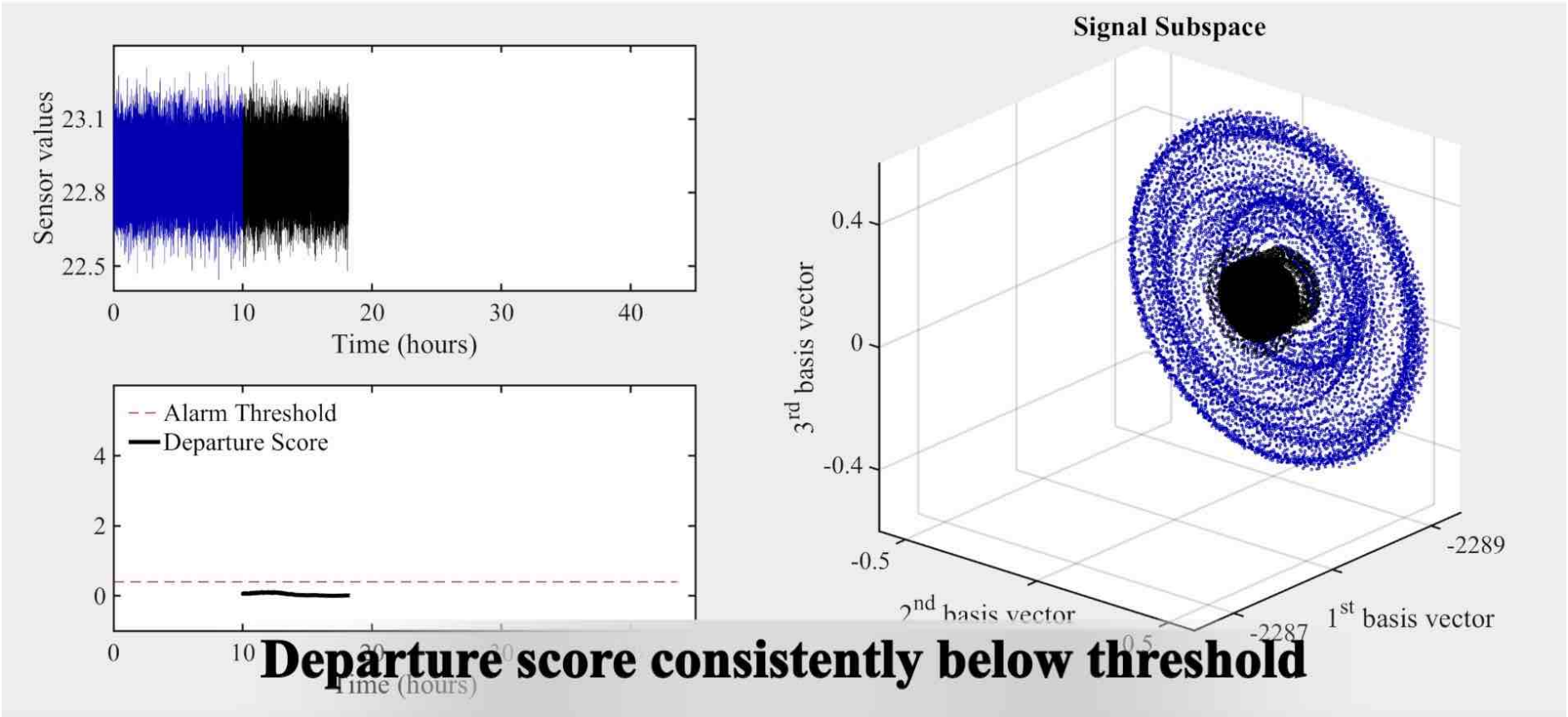

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



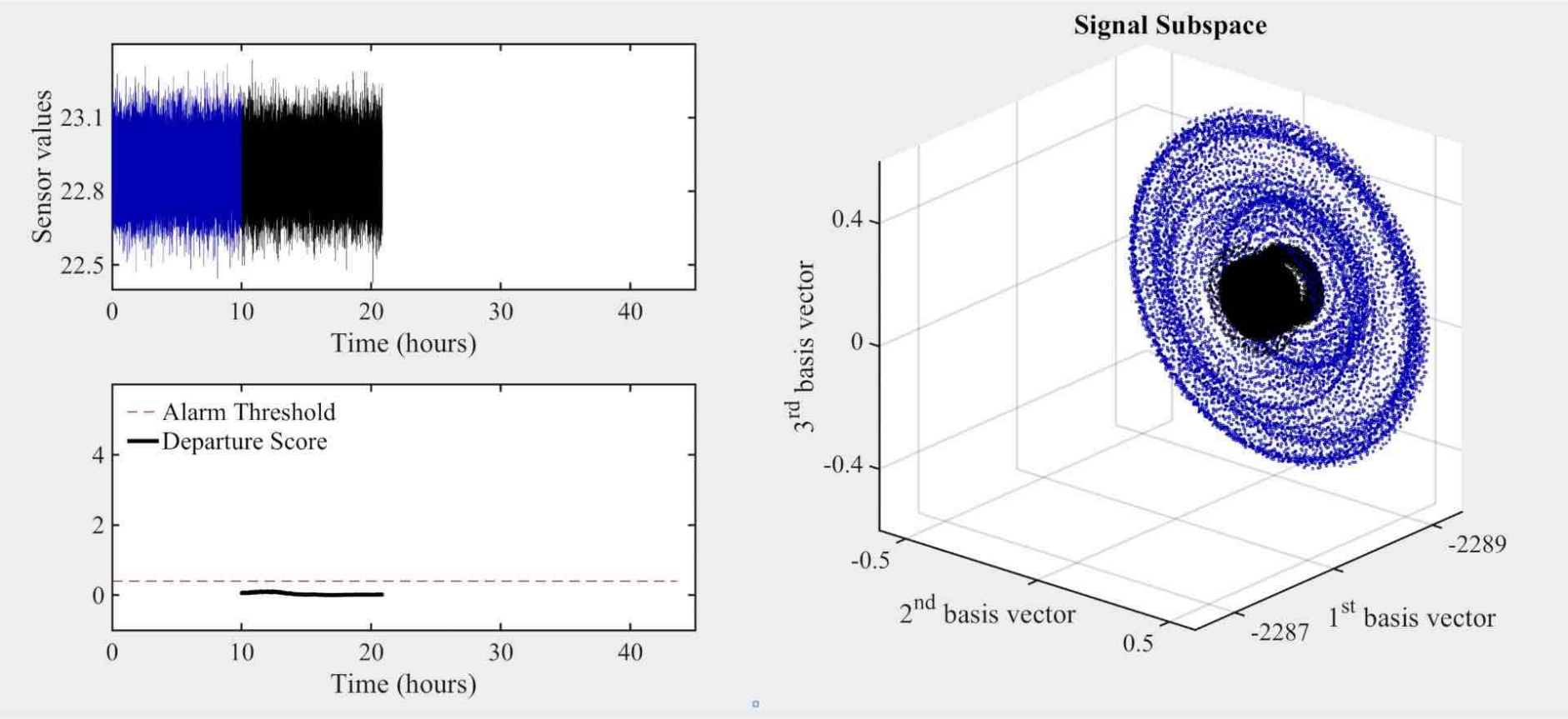

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



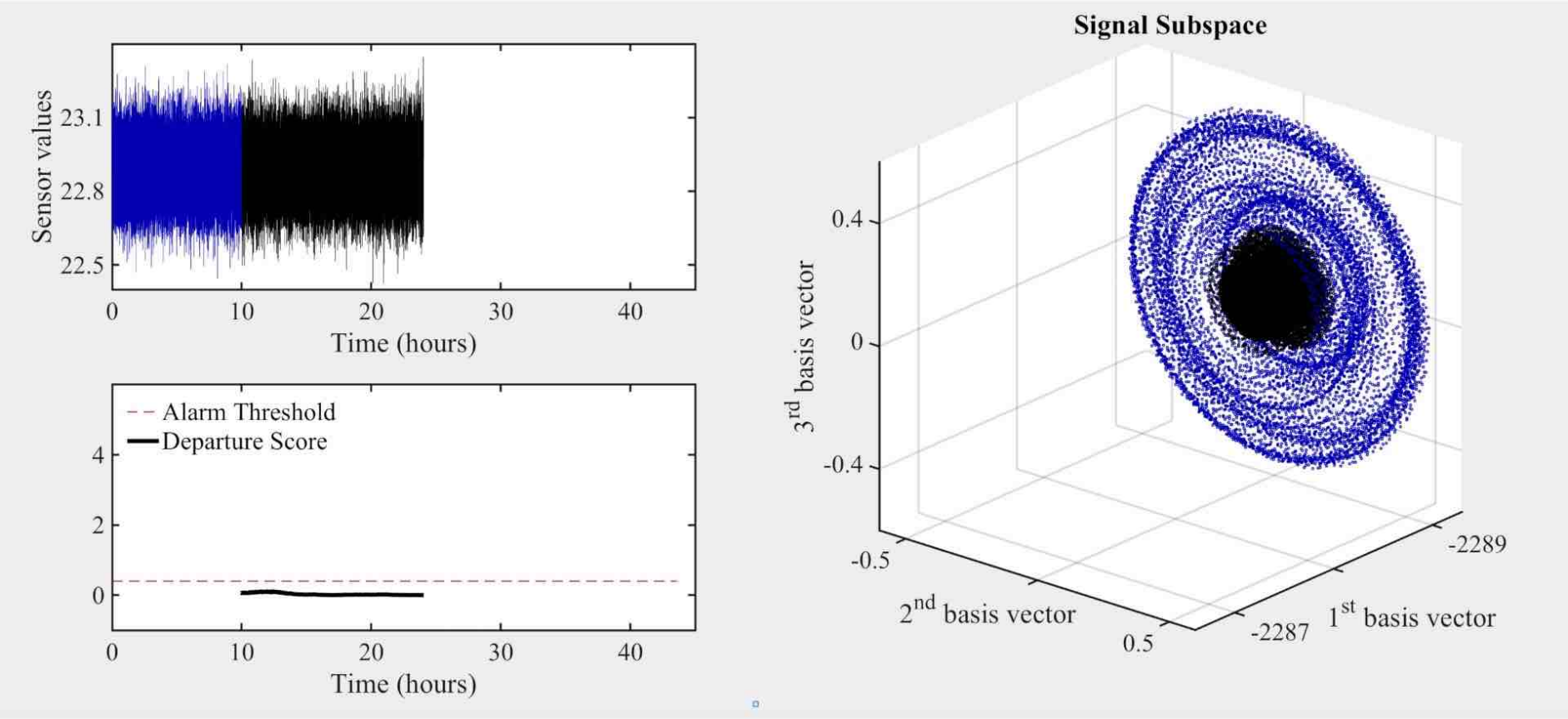

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



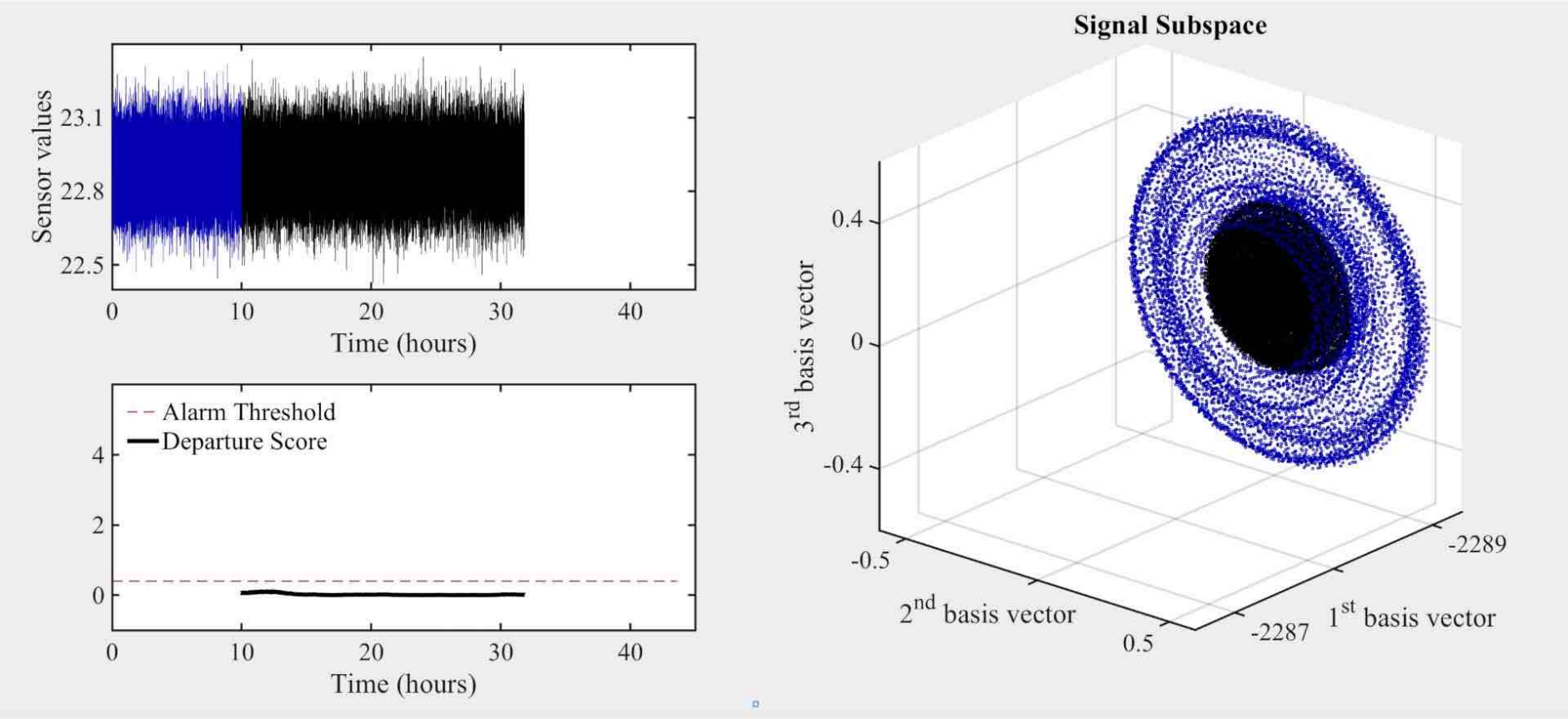

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



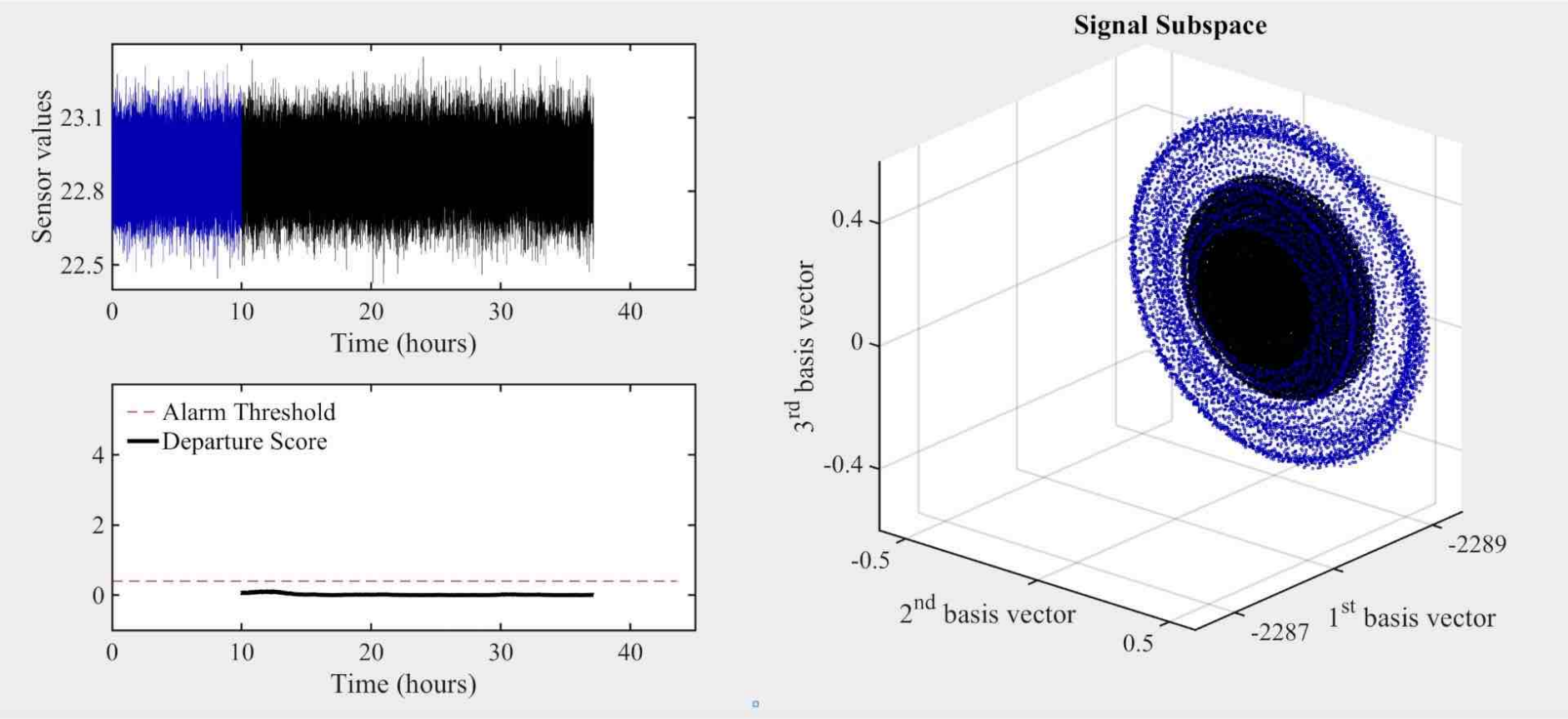

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



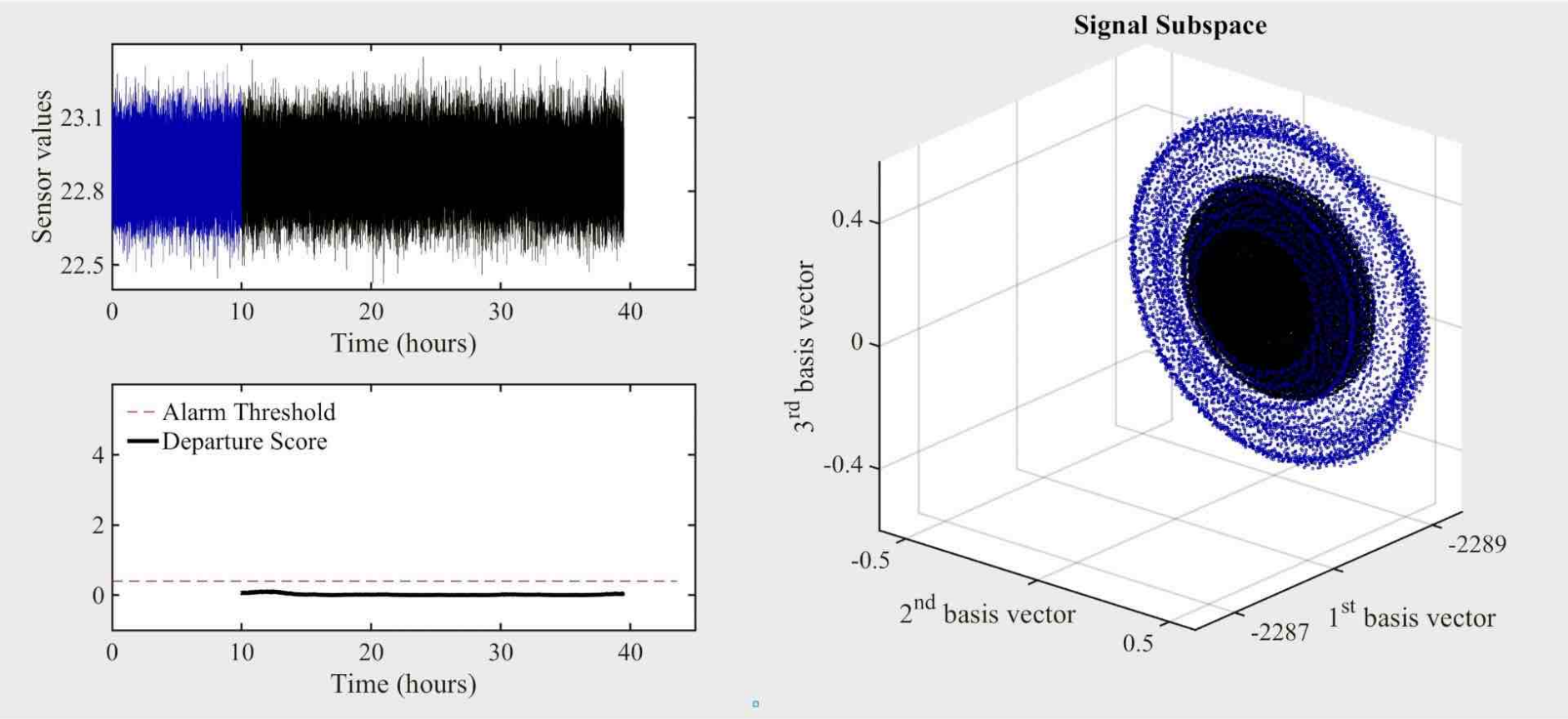

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



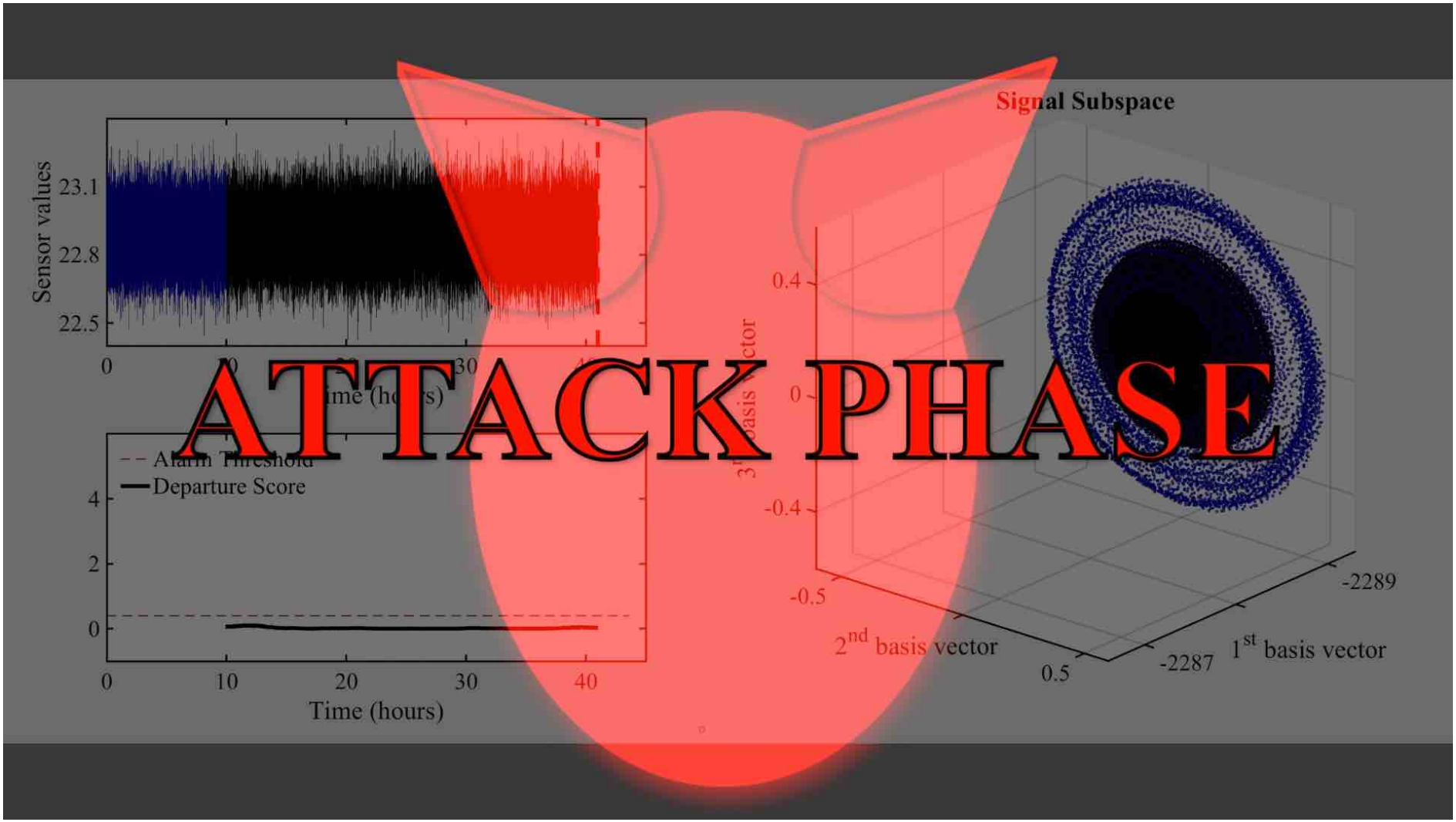

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



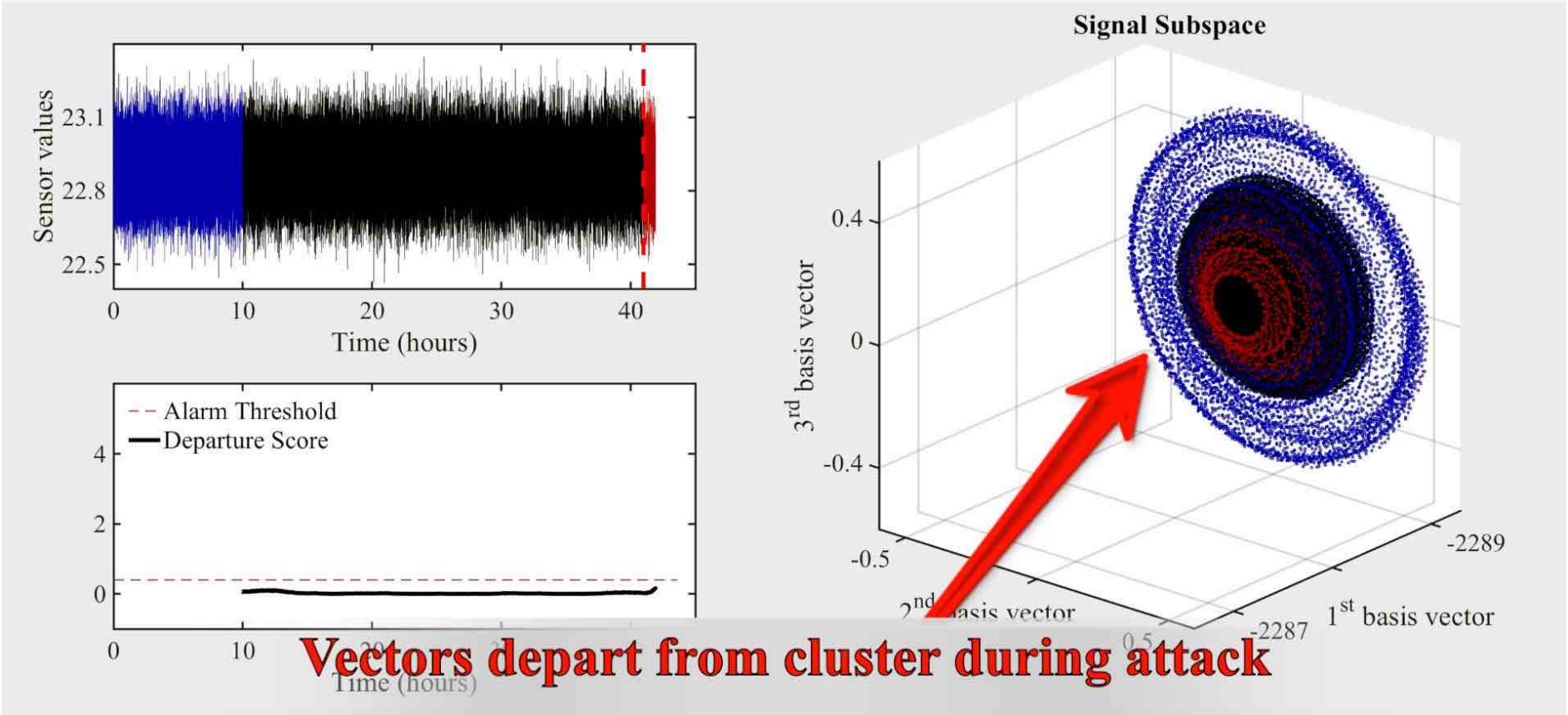

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



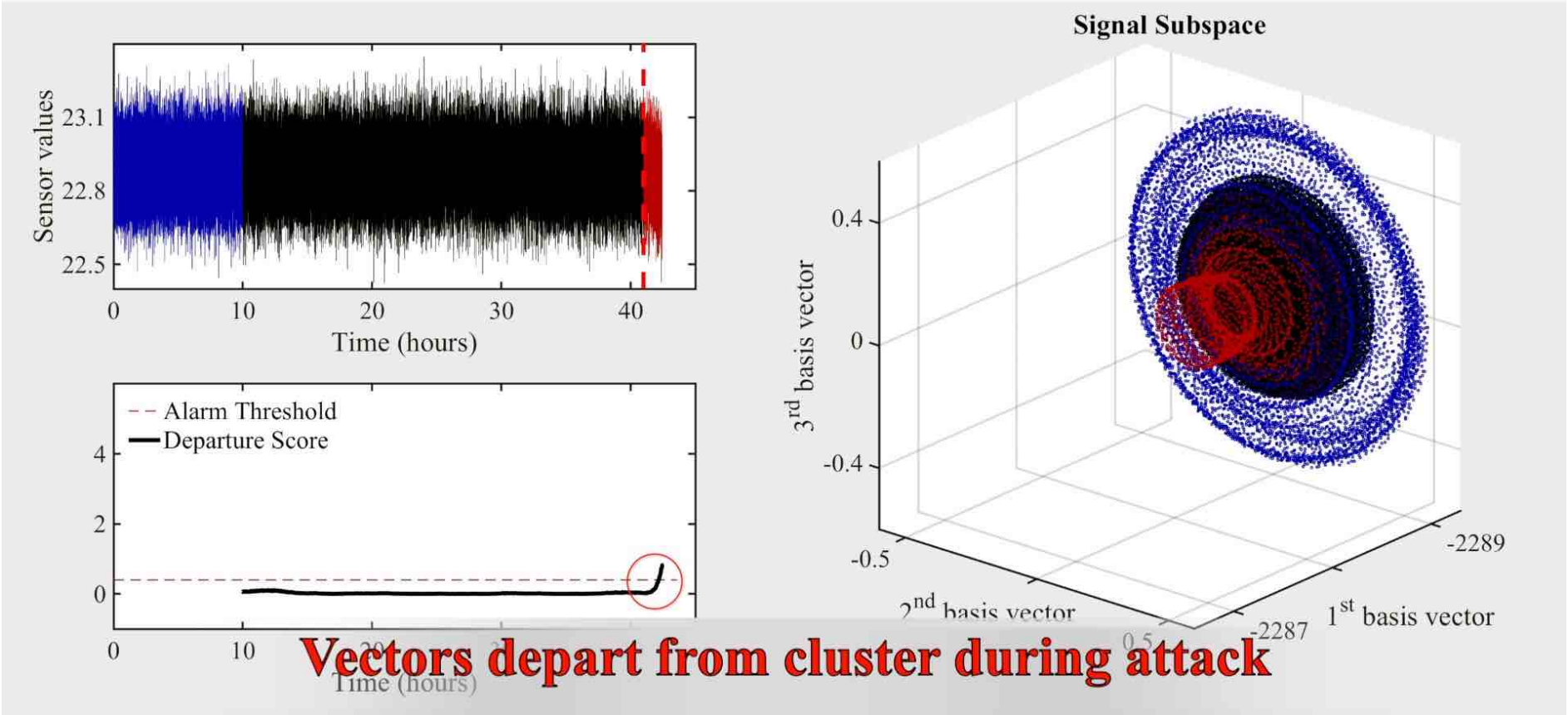

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



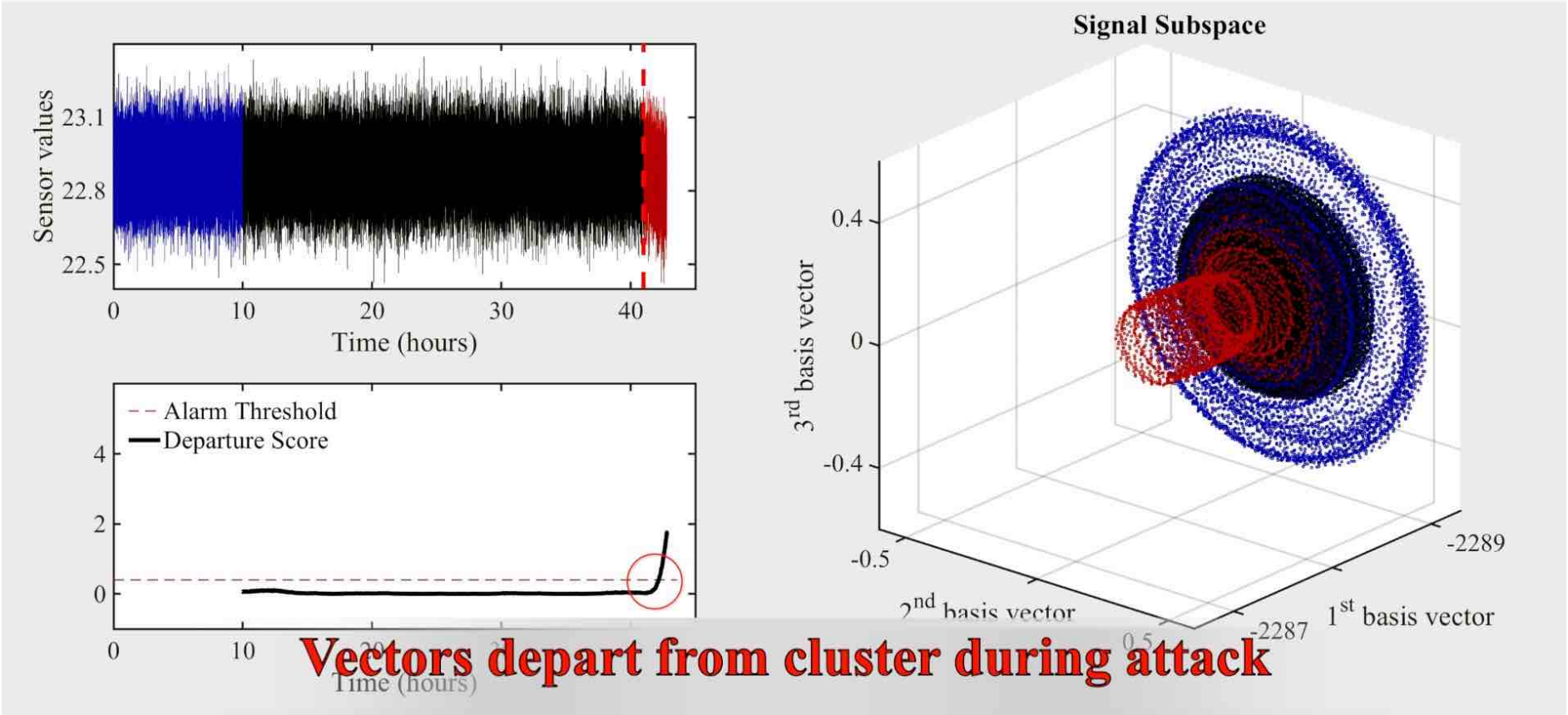

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



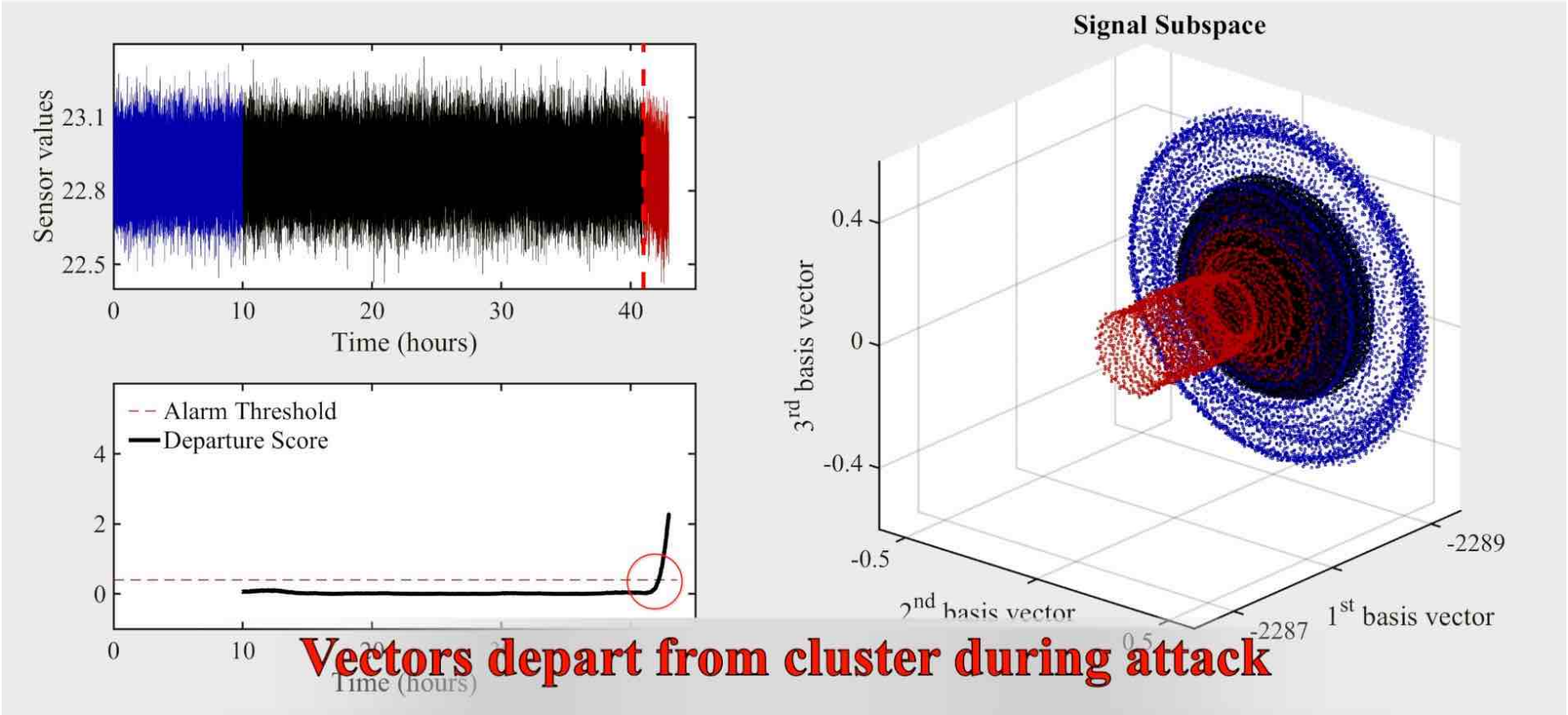

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



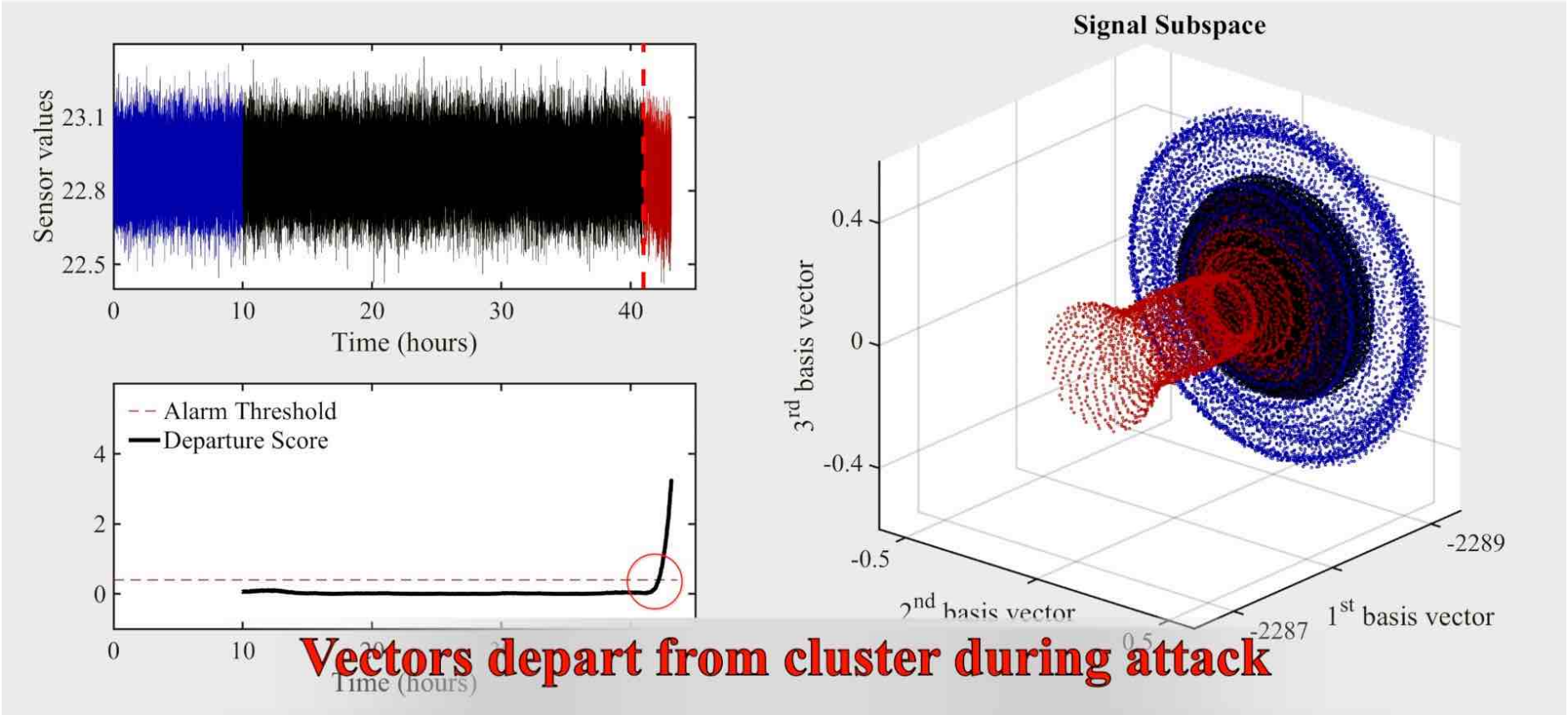

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



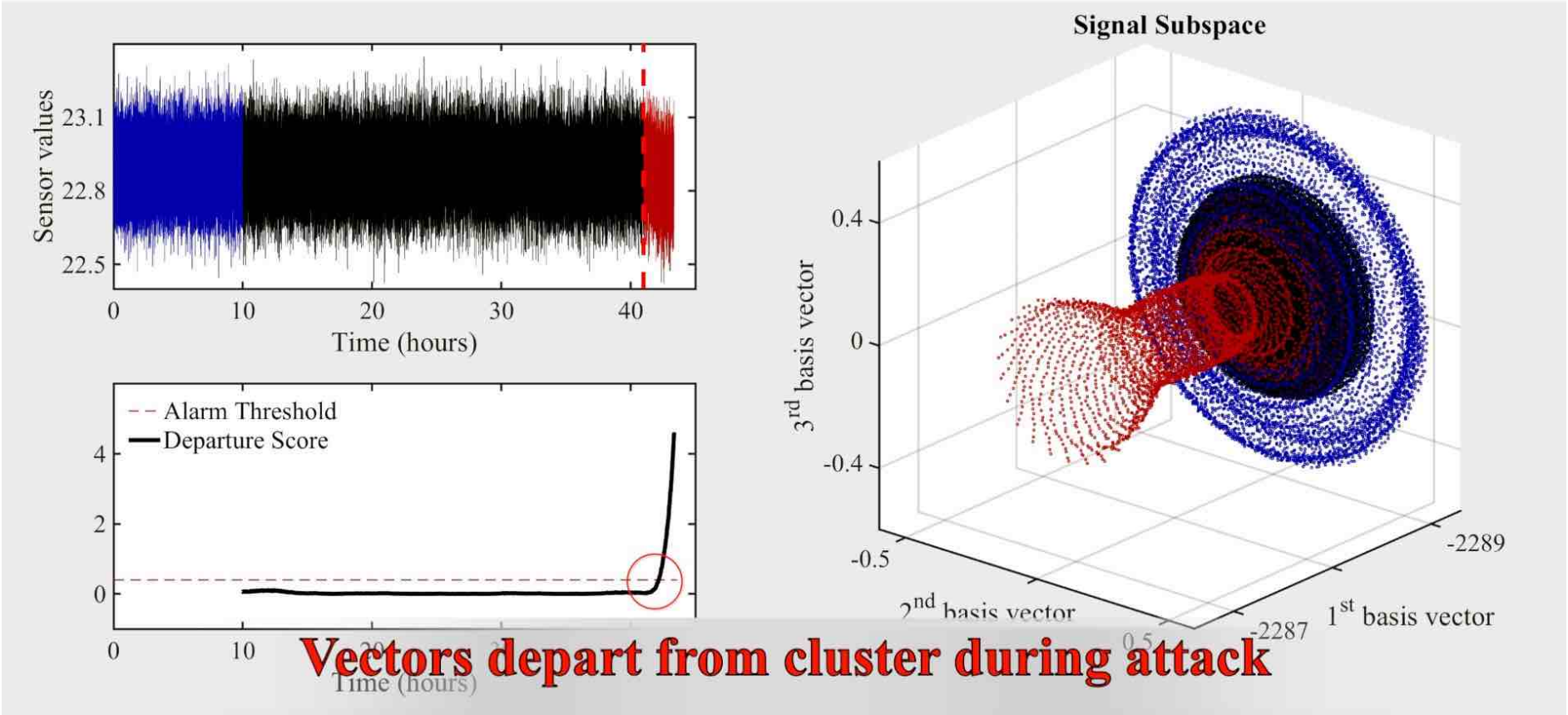

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



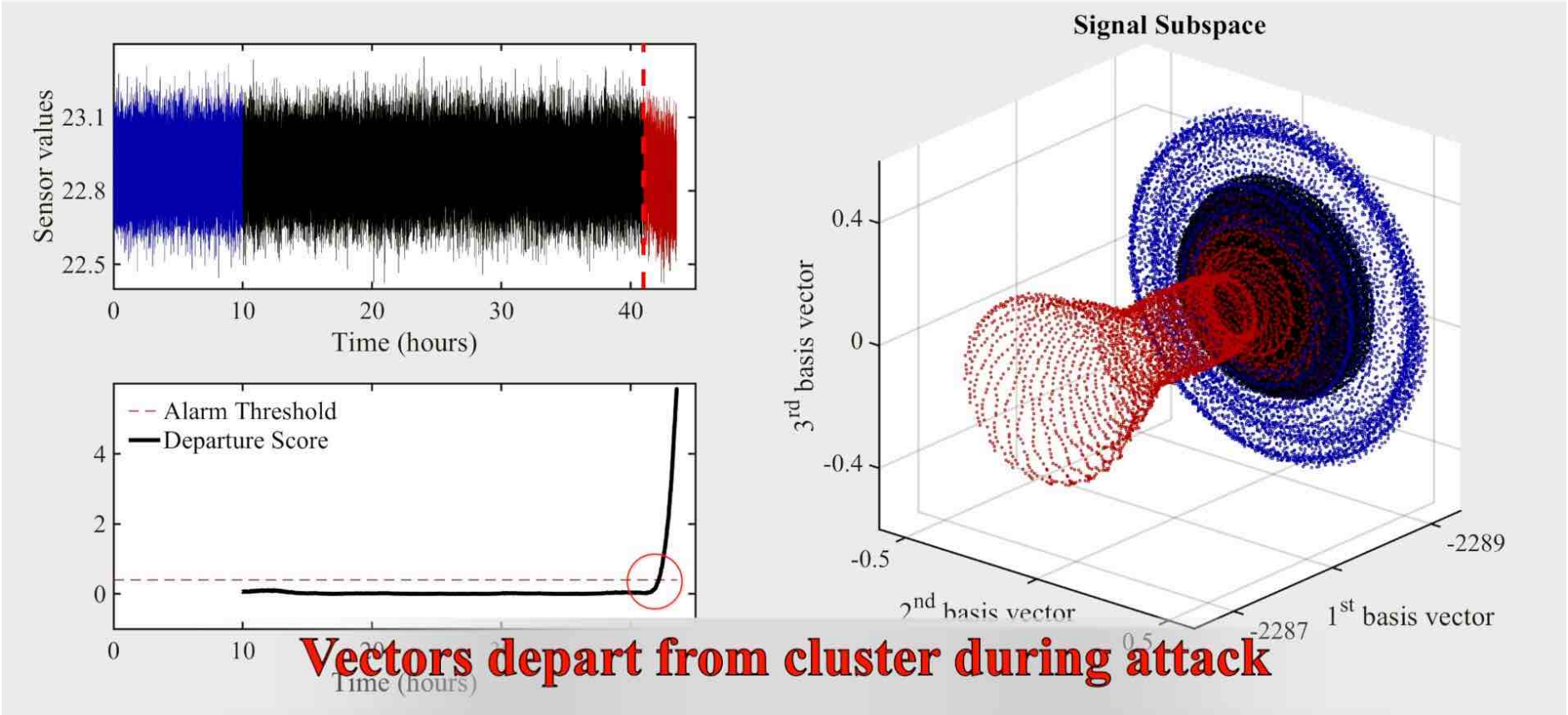

Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Visualizing the Departure



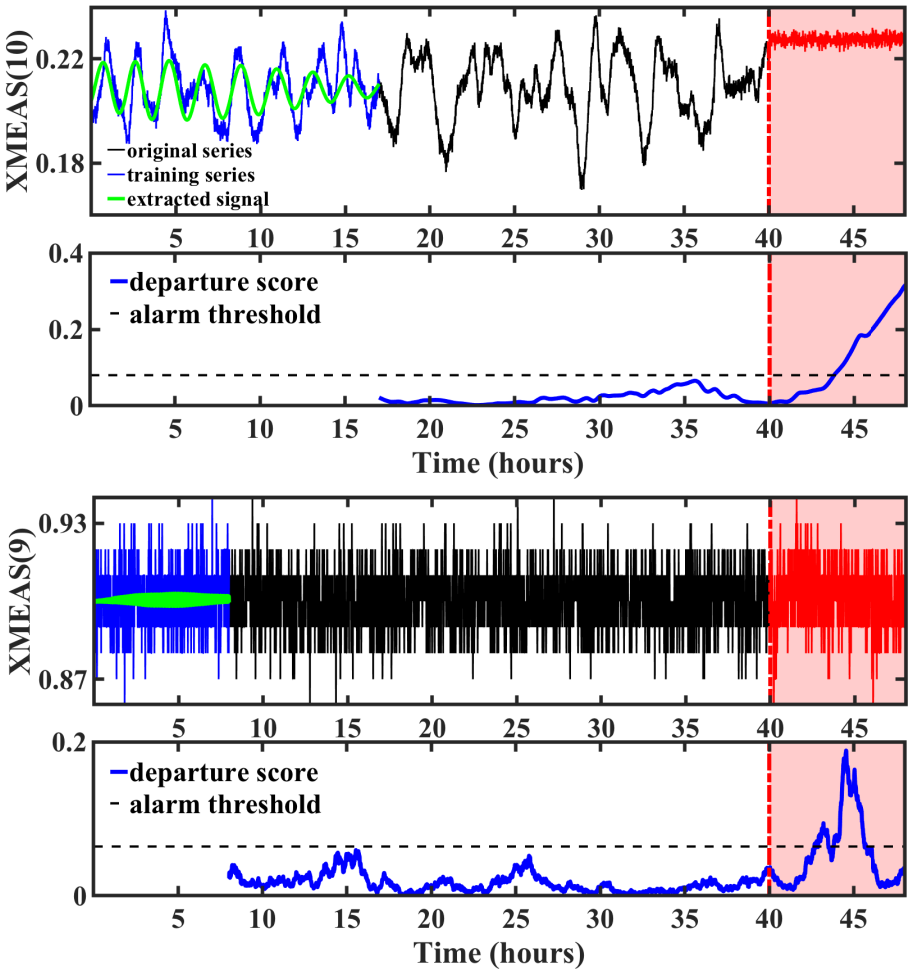

Watch the full video at https://youtu.be/SSs4IeM2MOs.

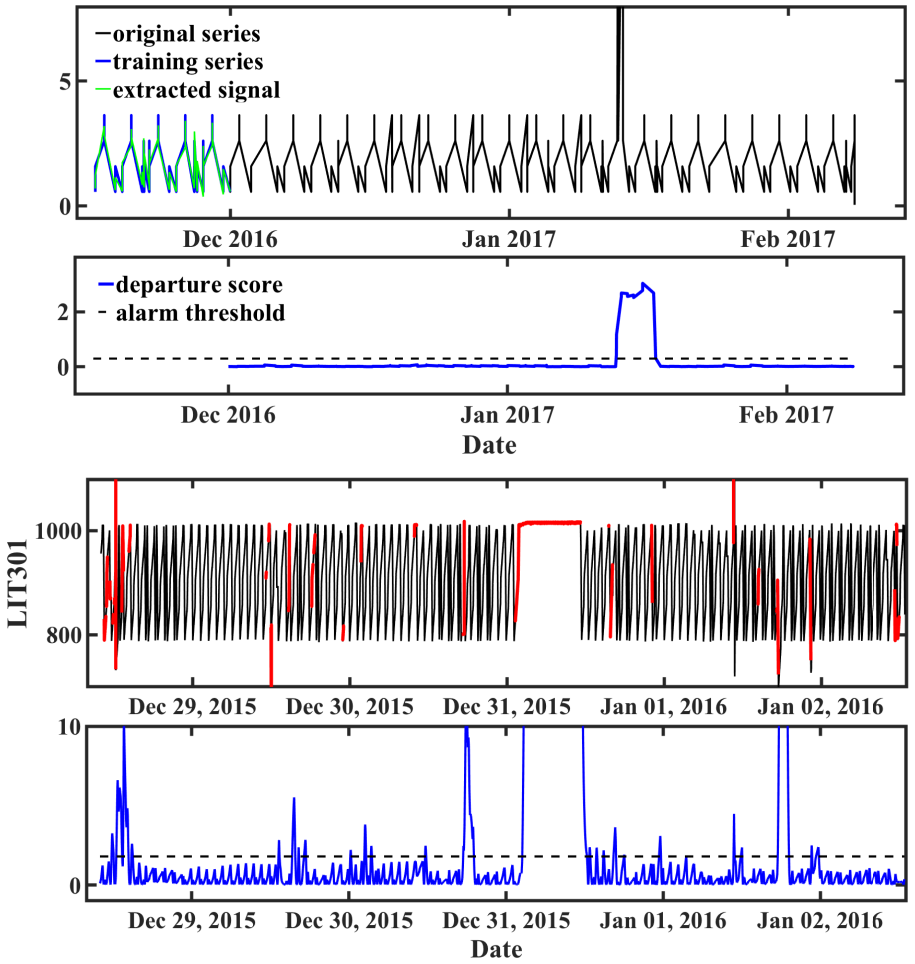
# Validation — Visualizing the Departure




Watch the full video at https://youtu.be/SSs4IeM2MOs.

# Validation — Evaluation on Various Systems



# Deploying a Prototype in a Real Environment

CHALMERS

## Deployment in a real control system

- A full-fledged PASAD prototype was deployed in a paper factory in Sweden.
- System operation was monitored for 75 days.

# Challenges

## Challenges include

- dealing with an unknown environment;
- achieving high performance and low footprint;
- maintaining system stability;
- trust and data access issues.

## Overview

- Bro for network parsing
- Parsing of Modbus data
- Buffer for event synchronization
- PASAD for analysis

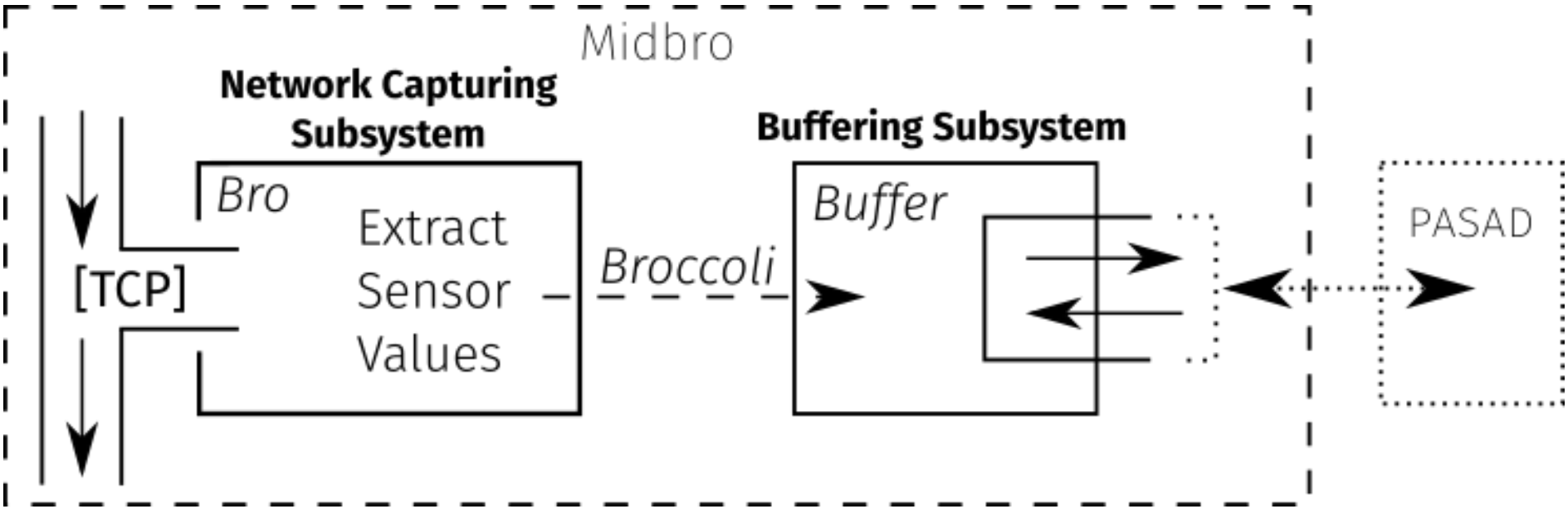



Figure: System overview

## Network Capturing Subsystem

### Handling Modbus Traffic with Bro

- There exist several variants of the Modbus protocol.
- The communication is Client-Server based.
- Nodes are identified using UID and IP addresses.
- Transaction ID (TID) is needed to match requests and responses.

## Network Capturing Subsystem

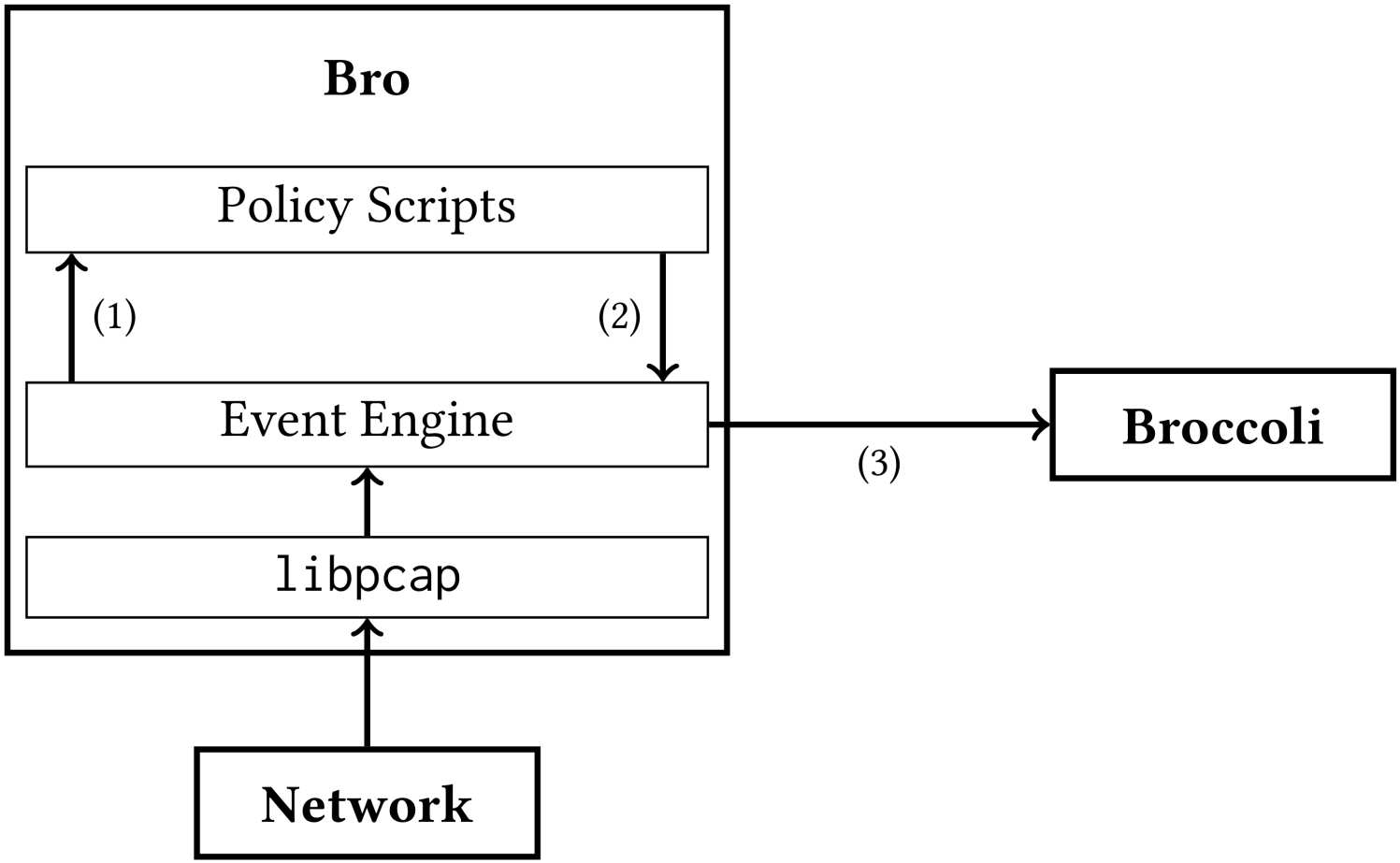



Figure: The Bro framework.

# Midbro System Architecture

## Buffering Subsystem

- Synchronizes Bro events with PASAD.
- Receives Bro events via a socket (broccoli library).
- Stores values in a bounded buffer.
- Provides interface to PASAD.

# Experiments

## The key to success: a local testbed

- Used real Modbus/TCP traffic from water distribution plant.
- Captured traffic was replayed over a small network.
- Stress testing the prototype.
- Emulating packet drops.
- Identifying unexpected behaviours.

CHALMERS

# The Factory

## The Factory

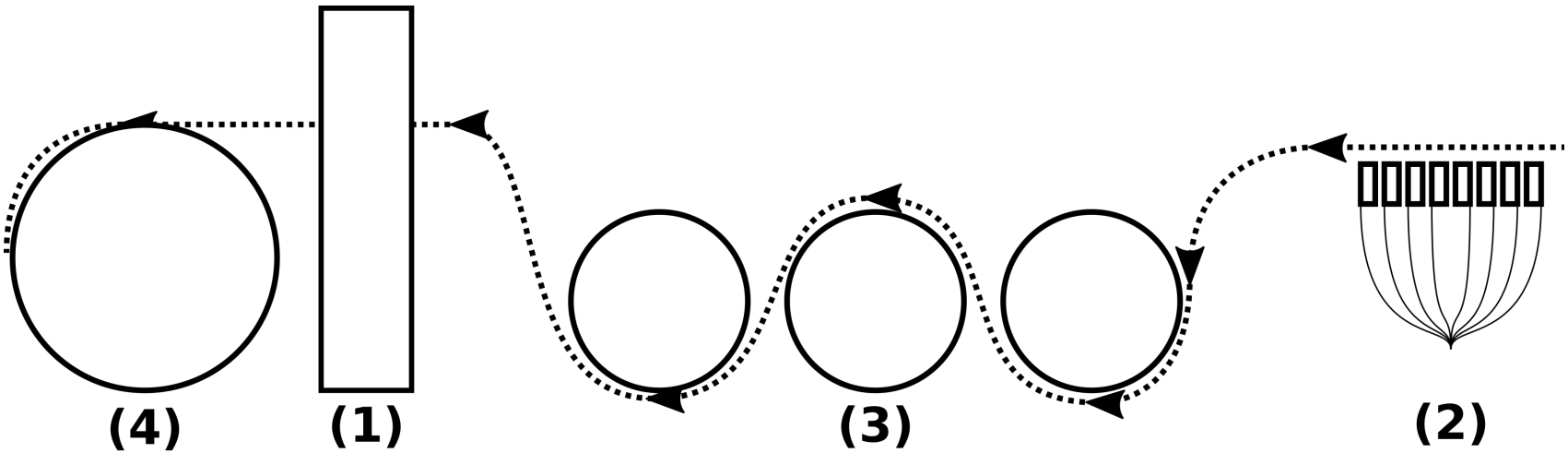



Figure: Production line.

## The Factory


Figure: **(1)** The control frame with sensors.

## The Factory

Figure: **(2)** The actuators for the water valves.

## The Factory

Figure: **(3)** The drying process.

## The Factory


Figure: **(4)** The paper roll.

## Deployment

- Raspberry Pi 3+
- Gigabit Ethernet
- 1.4 GHz quad-core processor
- 1 GB RAM
- Rasbian OS

# Process Monitoring

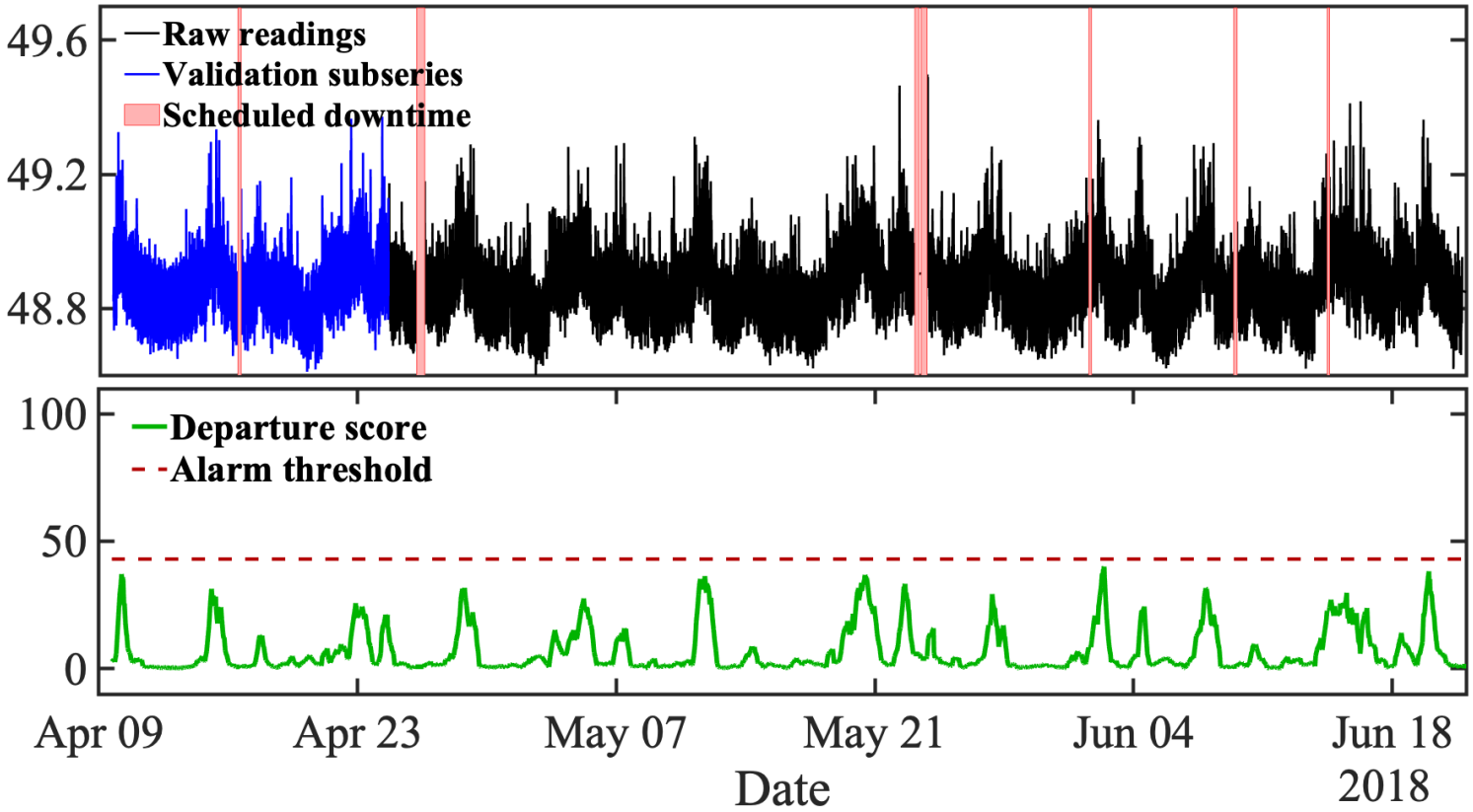



Figure: Sensor readings and departure score from PASAD.

## System Load

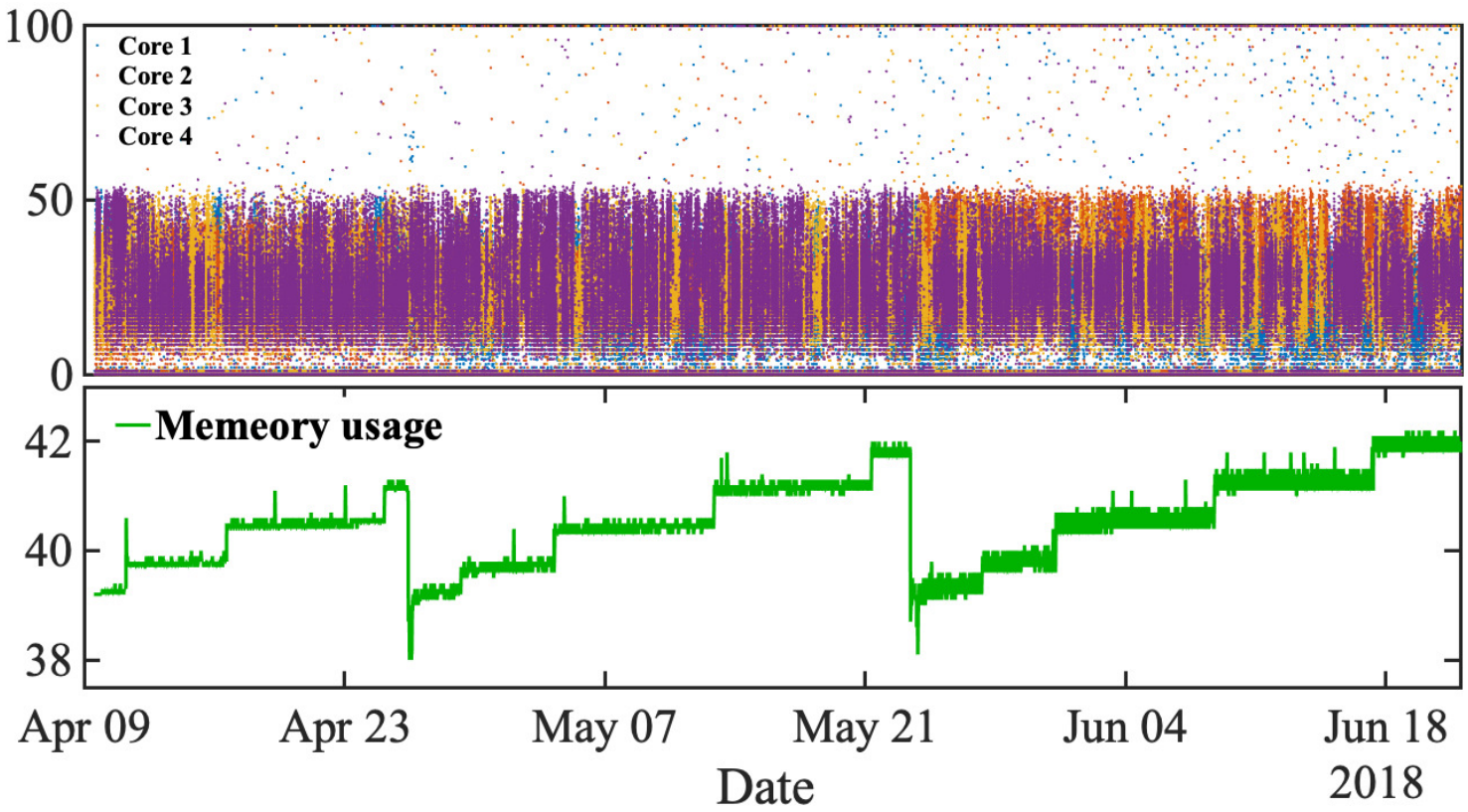



Figure: Processor and memory usage.

# Conclusions

## Lessons Learned

- Process agnostic does not mean plug-and-play.
- Signal data interruptions are tolerable.
- Bro was easy and versatile for Modbus parsing.
- Buffering is valuable, especially with Bro.