Optimizing Fault Injection in FMI Co-Simulation through Sensitivity Partitioning

Mehrdad Moradi, Cláudio Gomes,
Bentley James Oakes and Joachim Denil

Summersim 2019
July 22, 2019
Berlin, Germany

Outline

- Introduction
  - Context and fault injection
- Power window use case
  - Model and specifications
- Co-simulation
- Technique
  - Interval partitioning and sensitivity analysis
- Simulation
- Summary and next steps

Introduction

Cyber-Physical System (CPS)

- Increasing complexity
- Increasing application

Error prone and critical

- Test and simulate system

Introduction

Why?

- Understand system behavioral
- Safety
- Robustness
- Detect failure mode and errors
- Repaired

Fault Injection

What?

Testing method which aids in understanding how
[virtual/real] system behaves when stressed in unusual ways

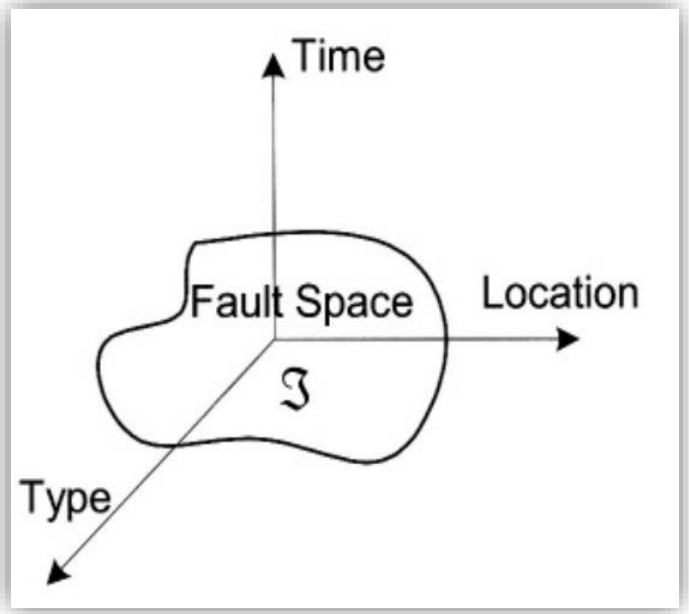
Fault injection

How?

- In the level of model
- Library of faults considering fault's nature
  - latency
  - Stuck to value

Research problem

- Faults are uncertain
- Limit fault space
  - Using sensitivity analysis
-



Benso, A. and Prinetto, P. eds., 2003. Fault injection techniques and tools for embedded systems reliability evaluation (Vol. 23). Springer Science & Business Media.

Motivating example

Power window

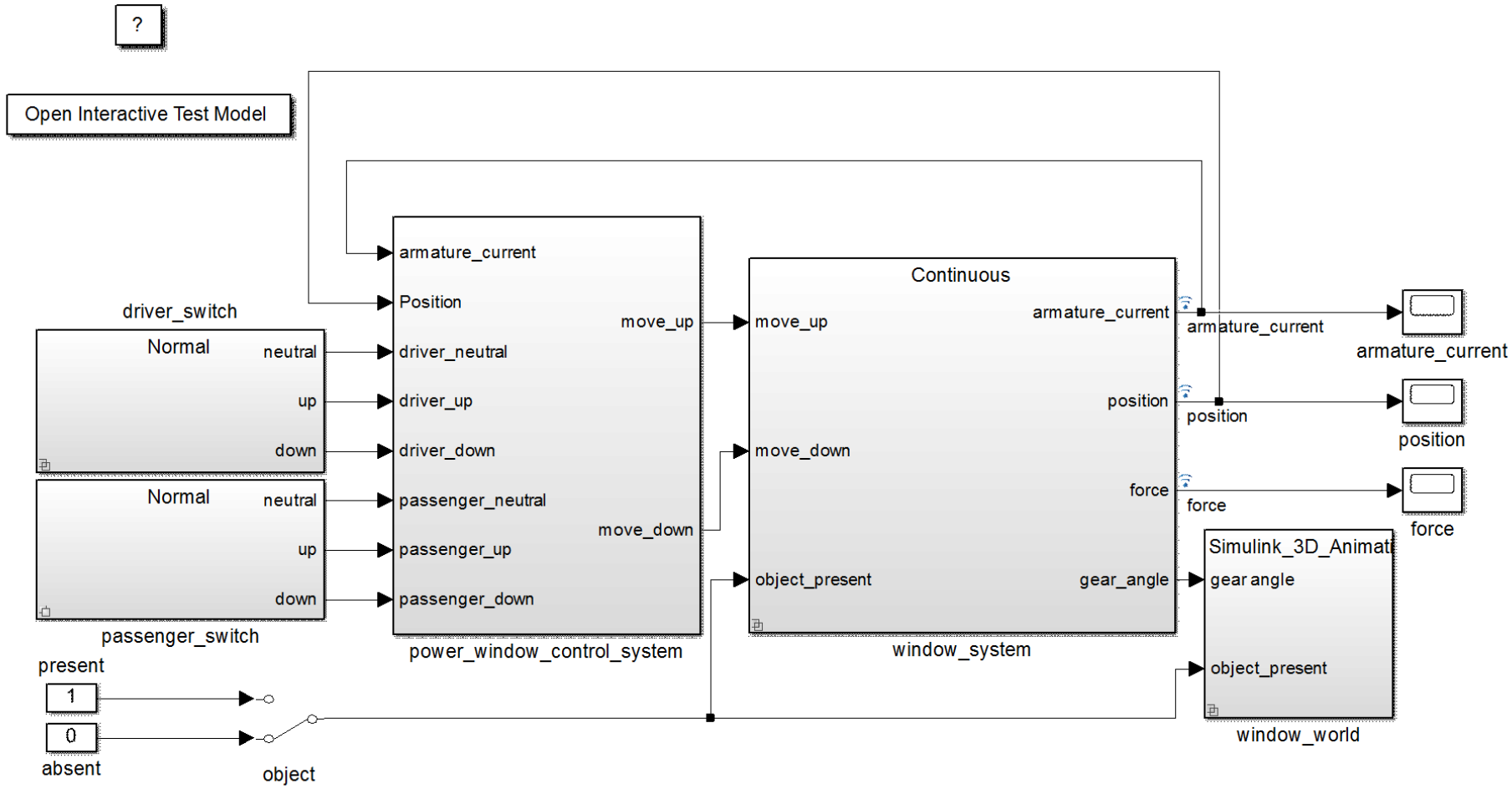
- Hybrid system

Specifications:

1. Fully opened/closed within 4s
2. Force to detect when an object is present should be less than 100 [N]

Safety-critical system due to possibility of object being crushed

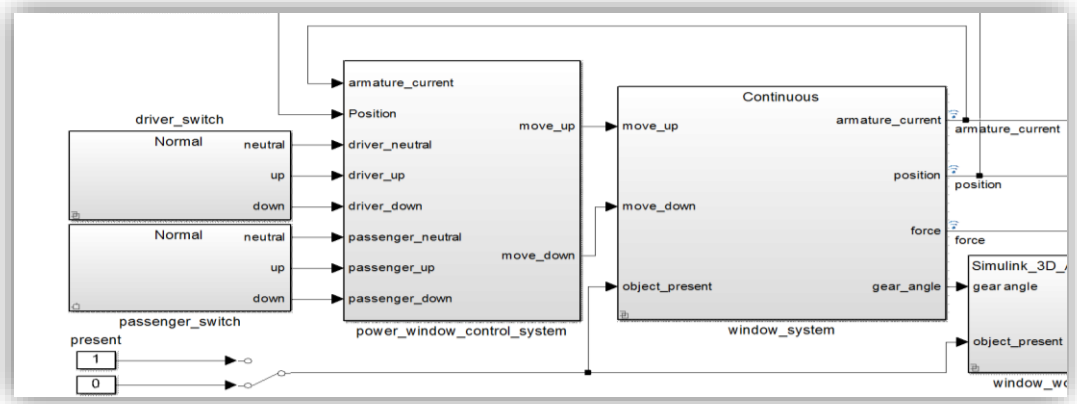
Motivating example - Structure



https://www.mathworks.com/help/simulink/ug/power-window-example-case-study.html

Motivating example – Normal behavior

Holding the up switch

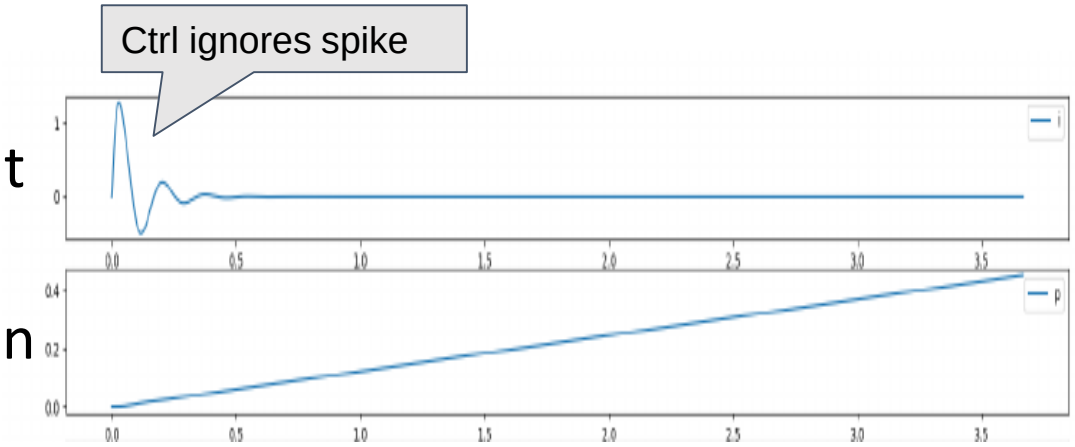


Specifications:

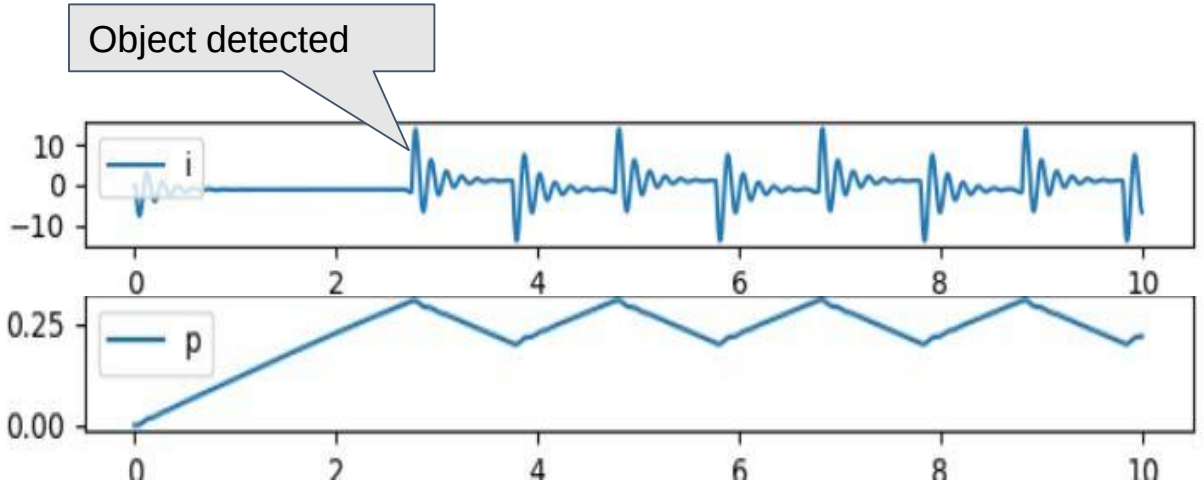
1. Fully opened/closed within 4s
2. Force to detect when an object is present should be less than 100 [N]

Current

Position



Without Object



With Object

Motivating Example – Fault

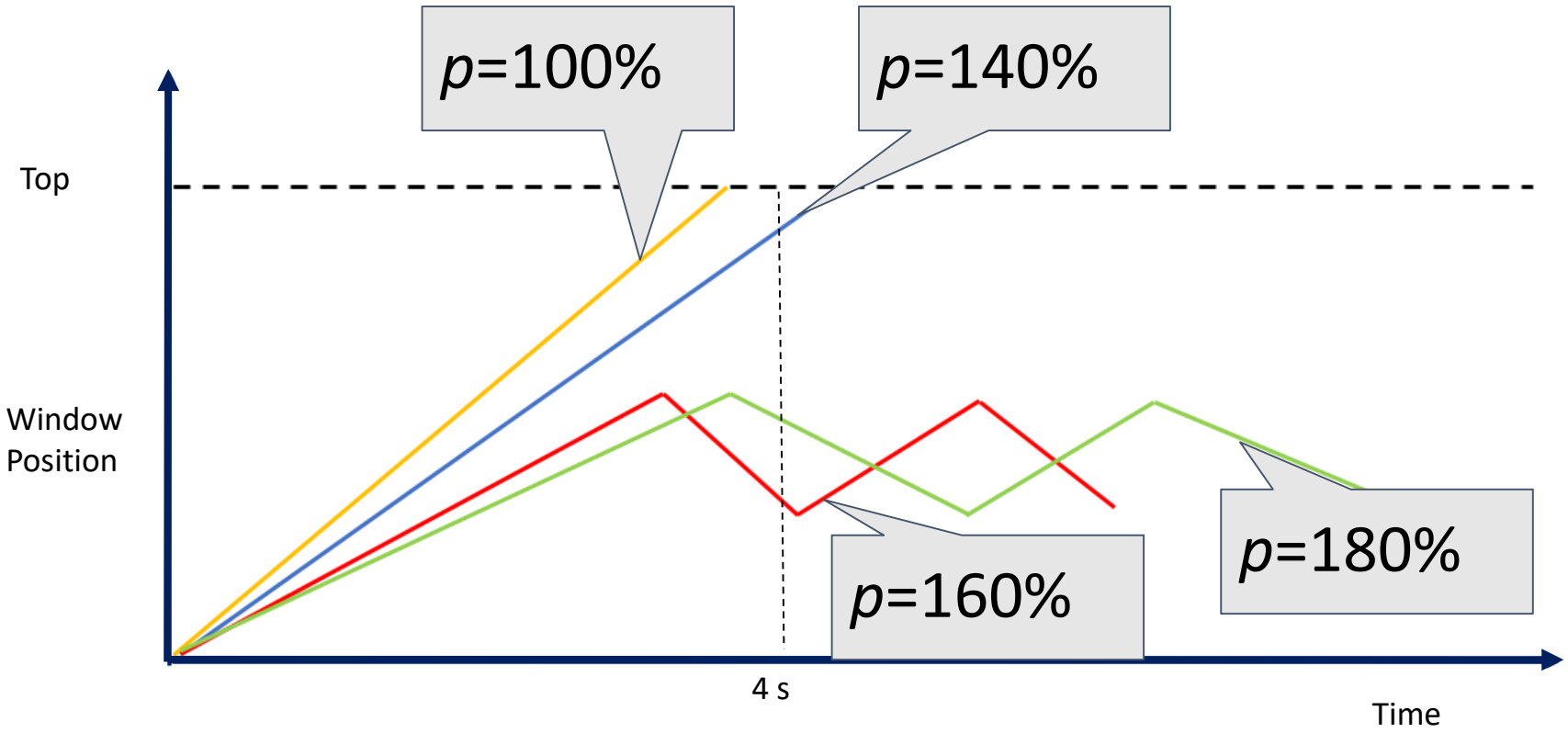
A power window is affected when the rubber slips into the window

Uncertainty in: when the rubber falls; and how much friction there is.

Example fault - Rubber friction

Friction parameter *p* is in [100%, 200%]

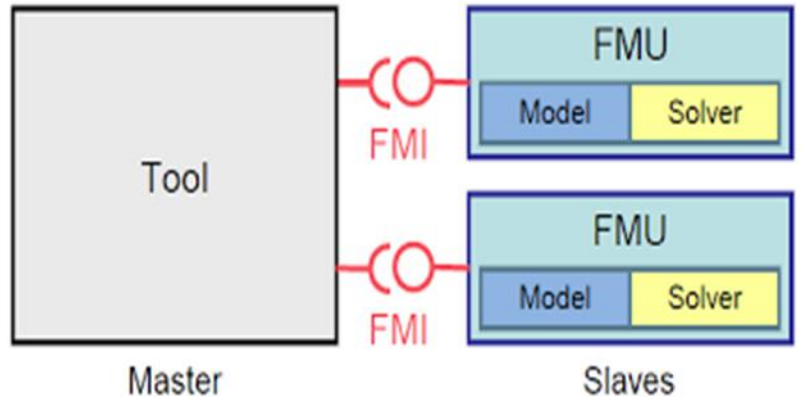
Different values of *p* lead to different behaviors of the window



Co-simulation

https://fmi-standard.org/

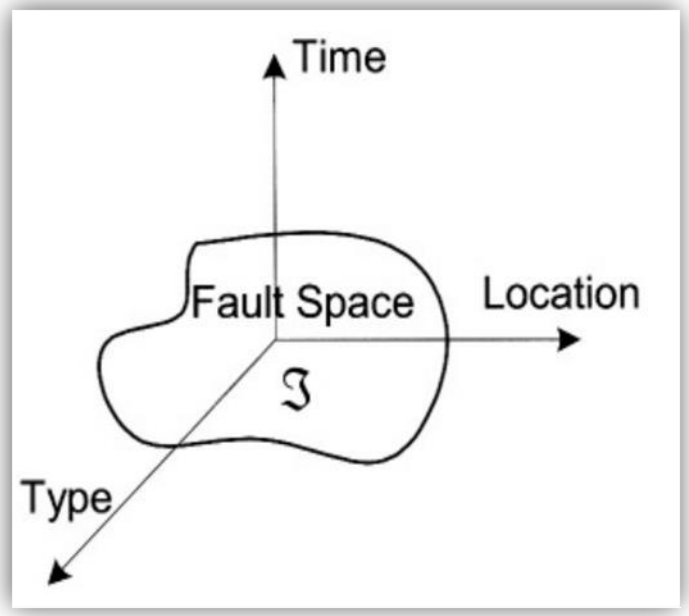
Co-simulation - defined in Functional Mock-up Interface (FMI)



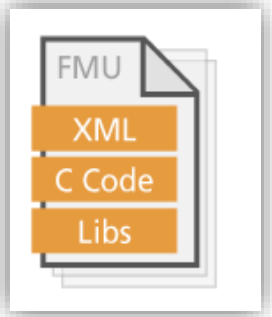
- Heterogeneous system
- High level of fidelity
- Free from IP concerns
- Increasing standardization and adoption by industry

Research problem in the fault injection

- Given a co-simulation, a library of faults, and system specifications
- Provide user with an **intuition** about how the fault parameters affect the behavior
  - Possibly violating specifications

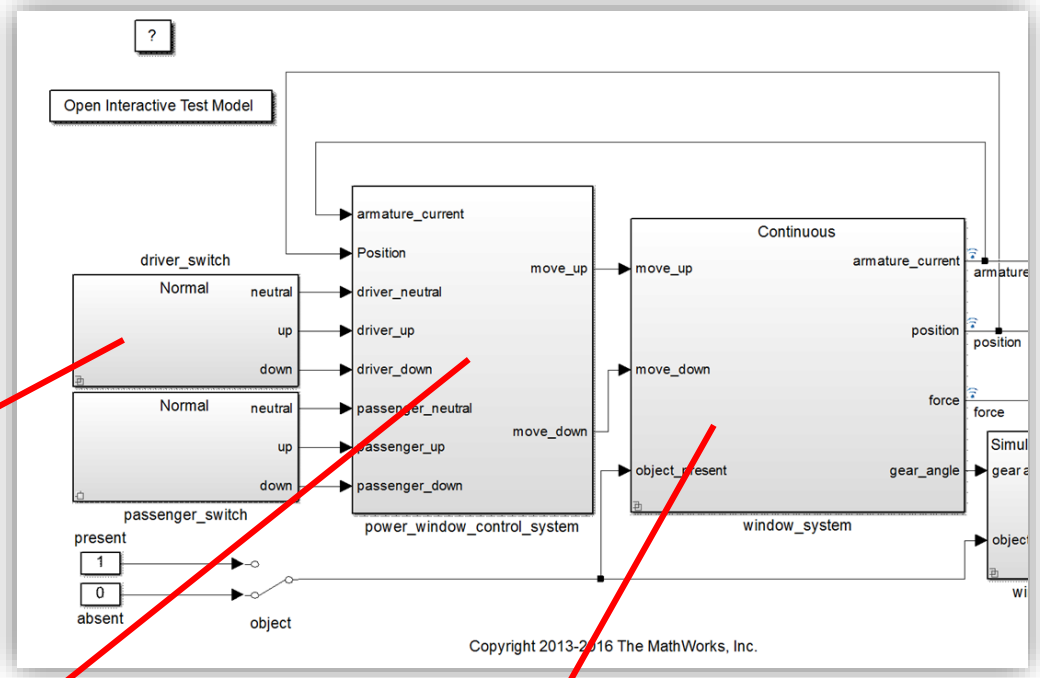


Injecting faults into co-simulation

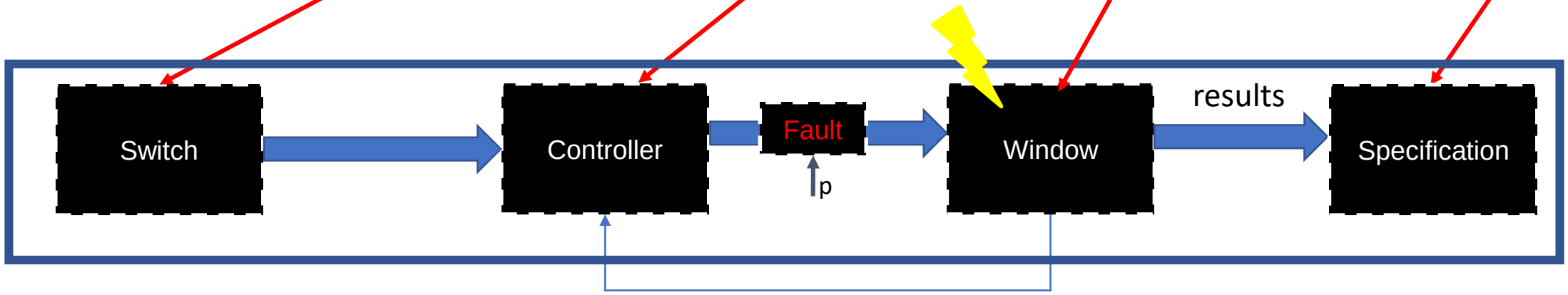


Virtual FMU

doStop()
setReal()
getReal()



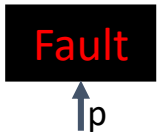
2. Force to detect when an object is present should be less than 100 [N]



Assumption: Stable and accurate master algorithm

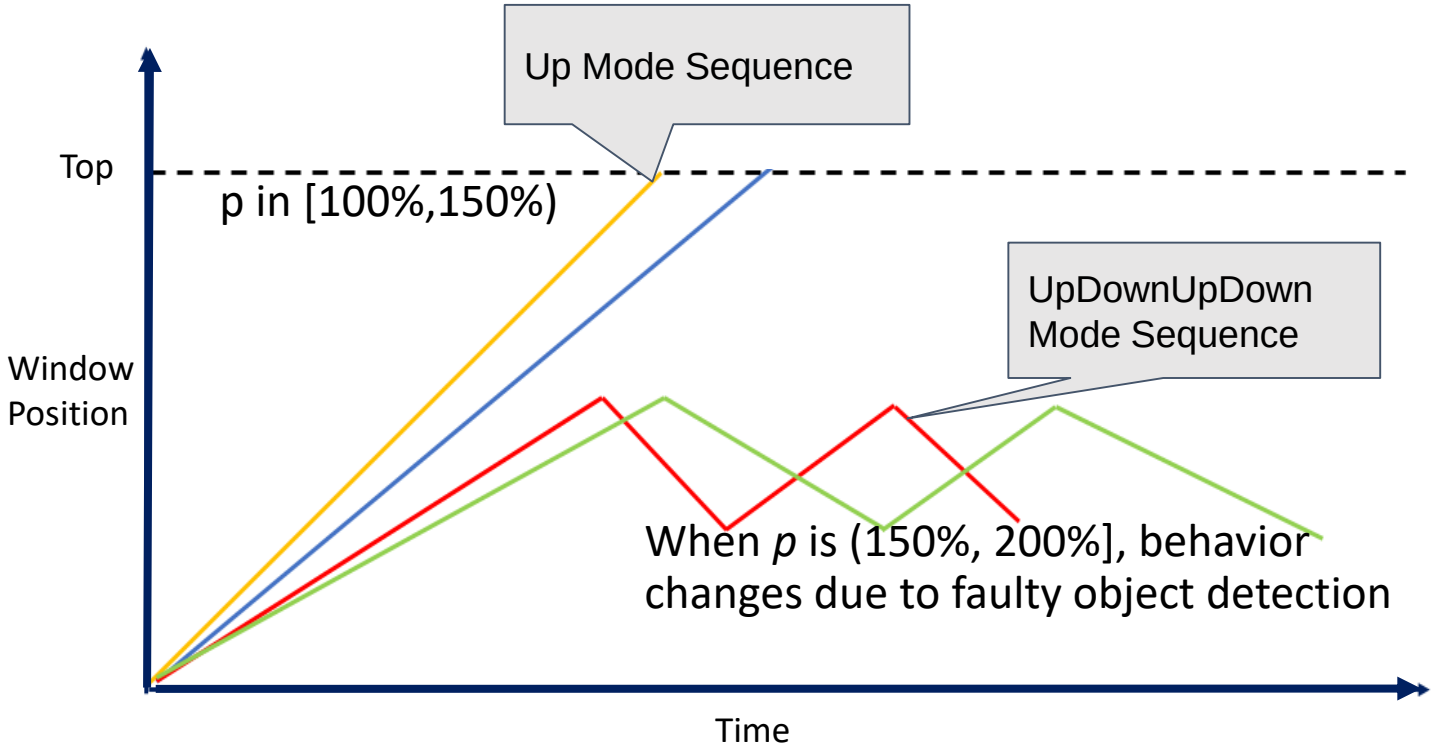
Interval partitioning → Sensitivity analysis

Parameter interval partitioning



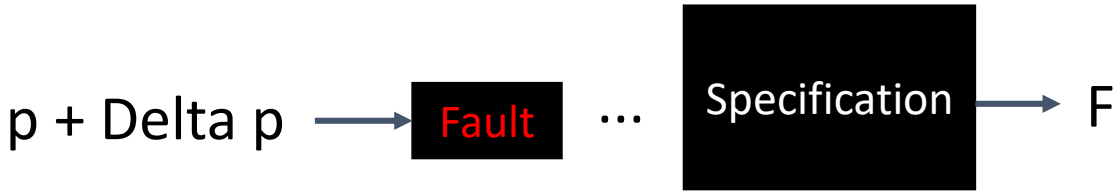
Example naive procedure:

1. Let fault parameter *p* in interval [100%,200%]
2. Partition the interval into N points,
   - run a co-sim per point,
   - store sequence of modes
3. Group adjacent points with same sequence of modes to form equivalence classes of continuity
4.

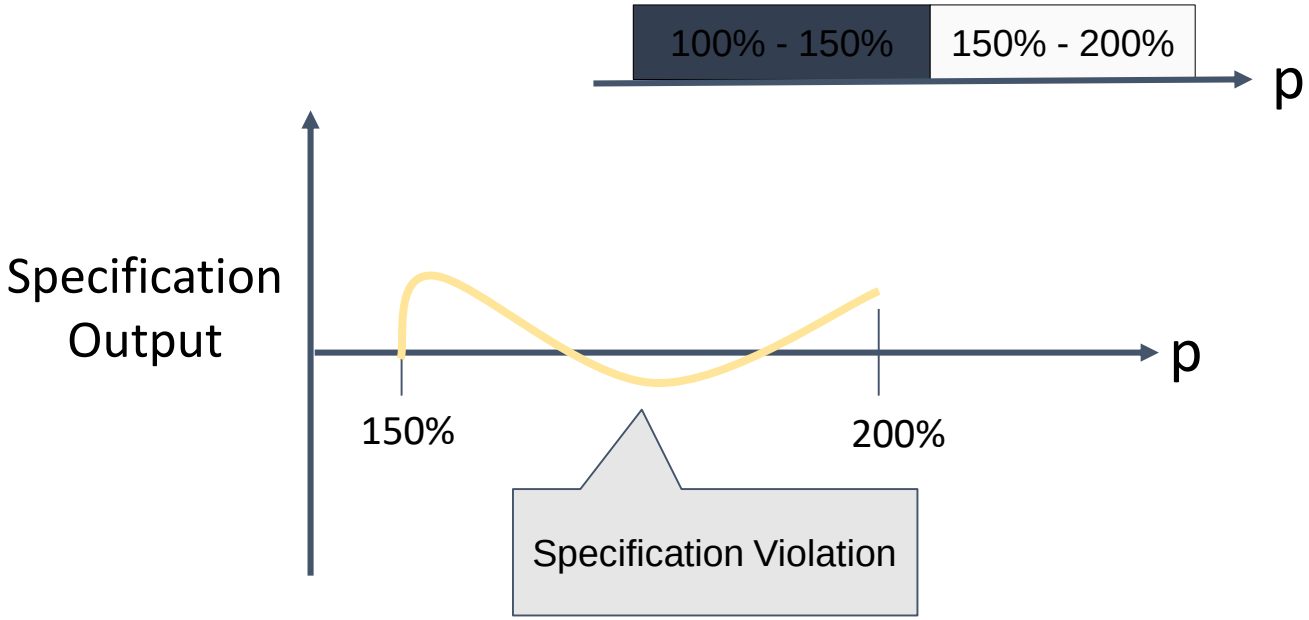


Sensitivity analysis

Want to determine how a change in our fault parameter affects the output of the specification



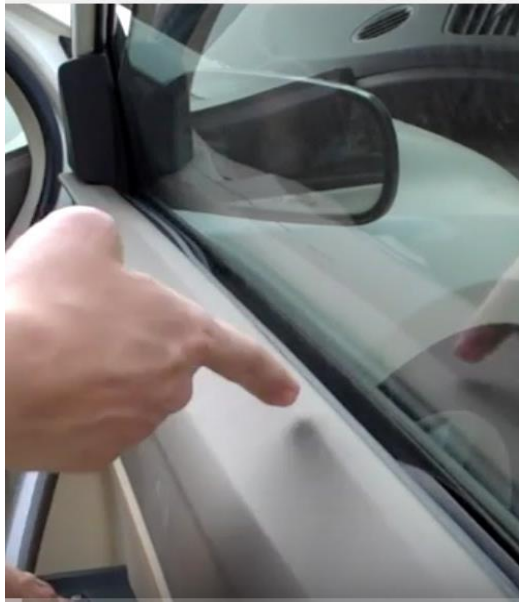
Within the same equivalence interval



Simulation

- Object is present
- Change of the rubber slip out’s time

- Causes the window to:
  To crush the object

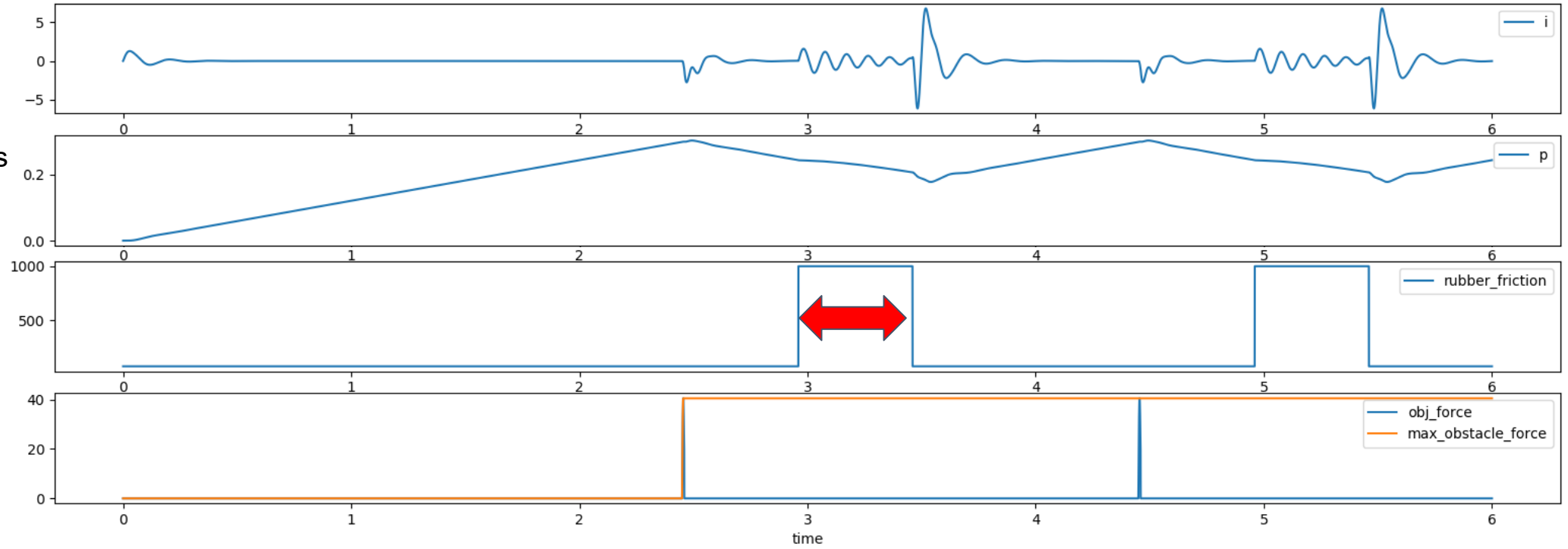


Armature current

Window's position

Rubber friction

Force

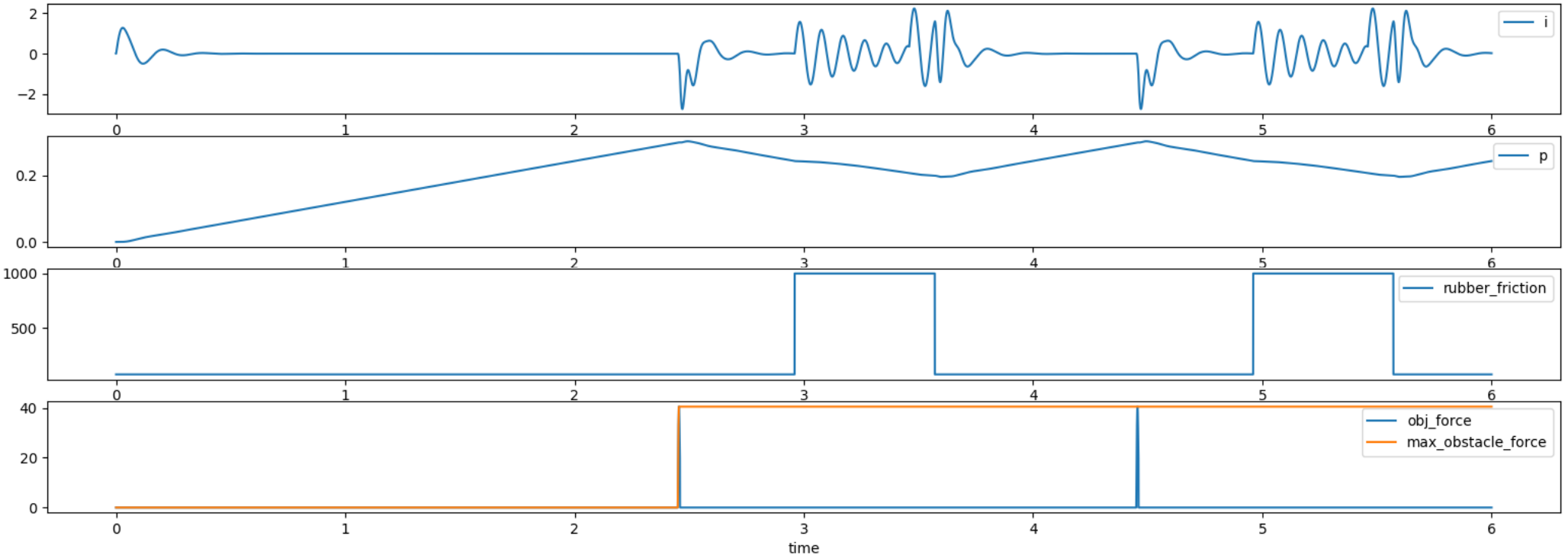


Armature current

Window's position

Rubber friction

Force

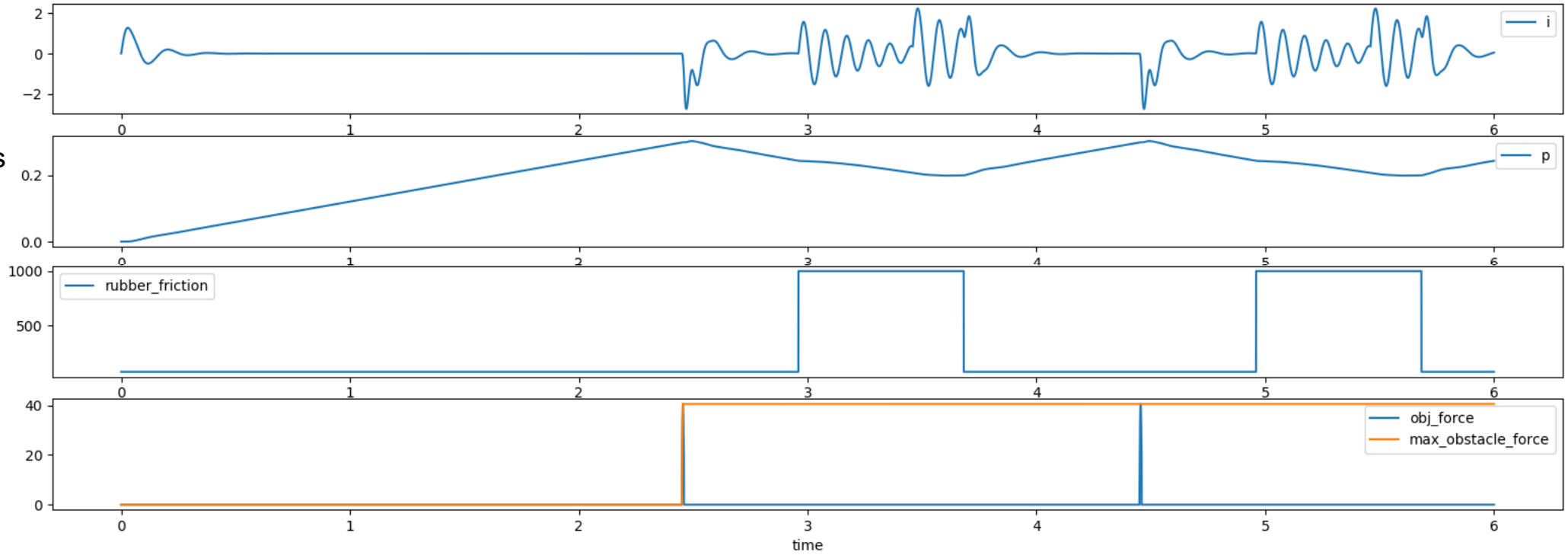


Armature current

Window's position

Rubber friction

Force

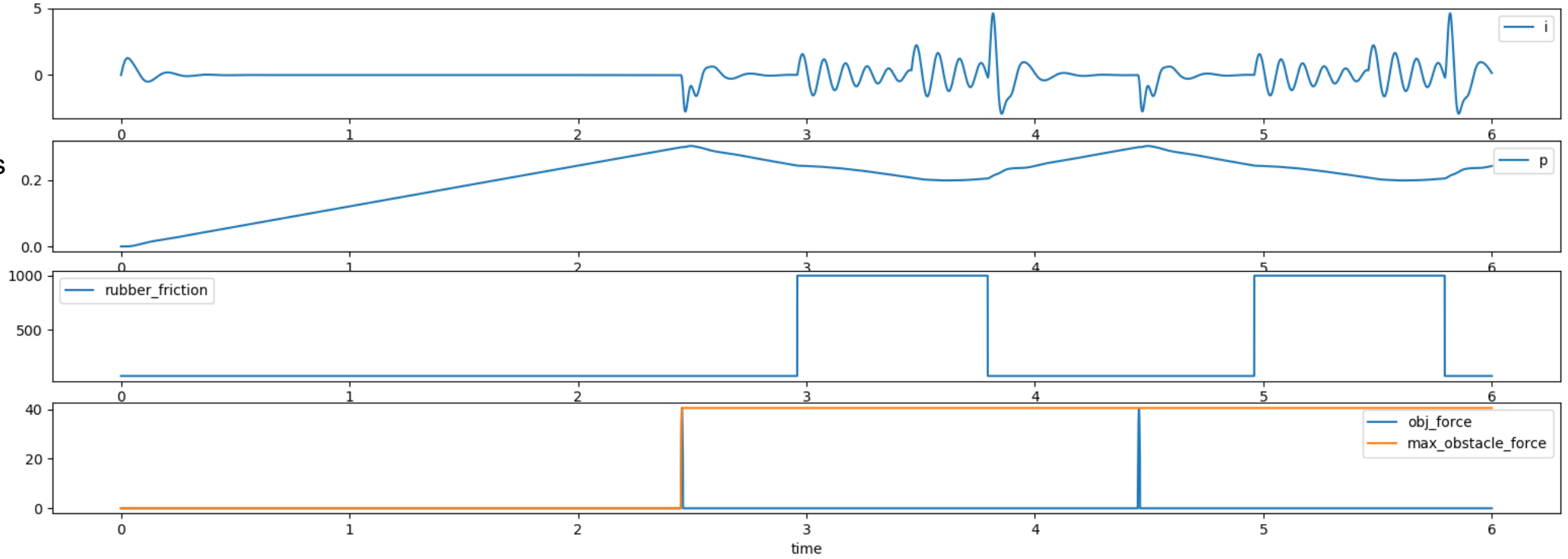


Armature current

Window's position

Rubber friction

Force

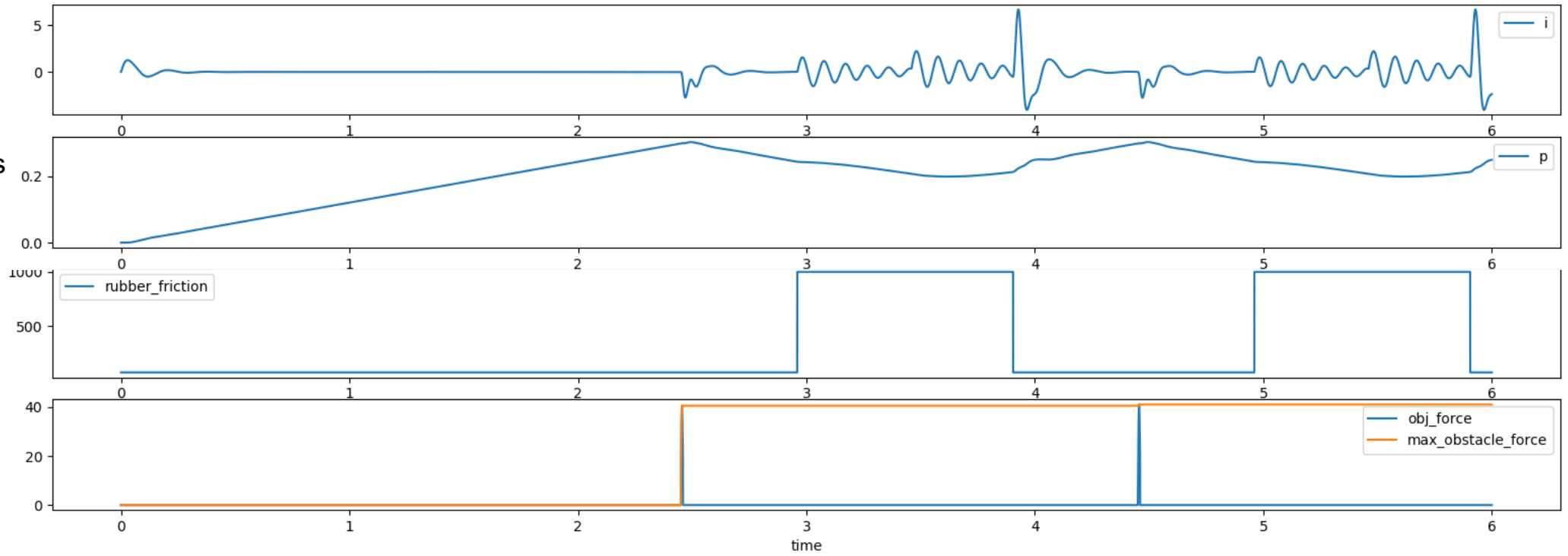


Armature current

Window's position

Rubber friction

Force

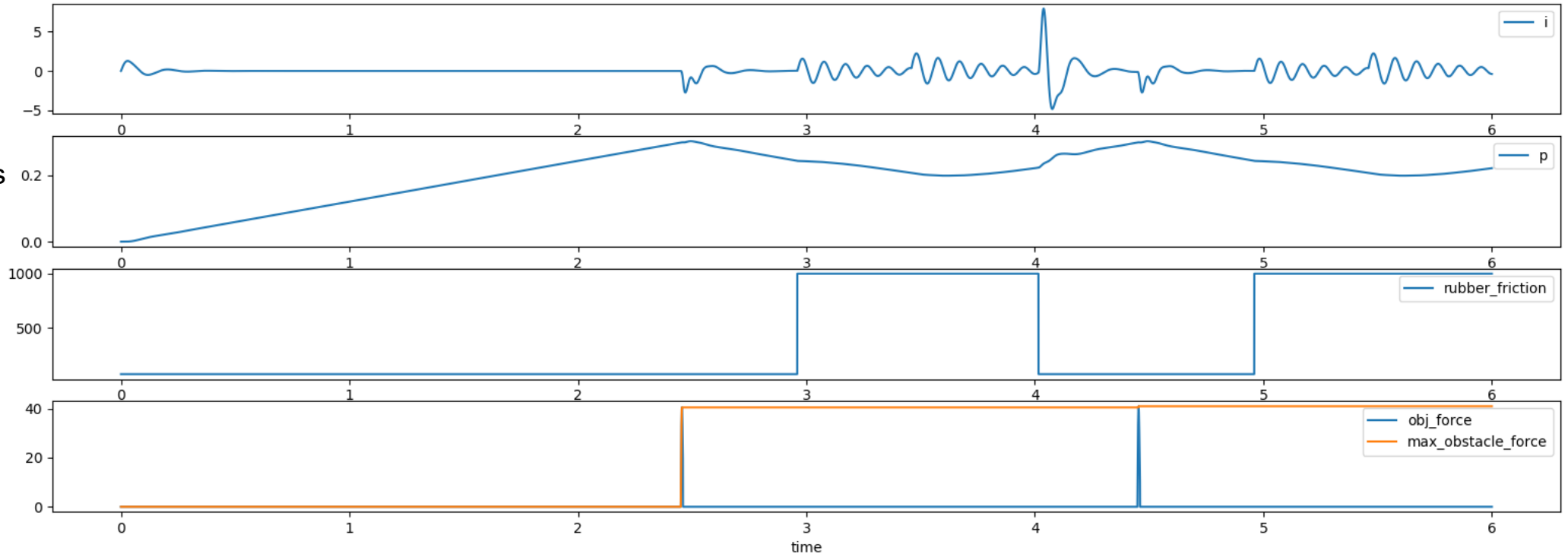


Armature current

Window's position

Rubber friction

Force

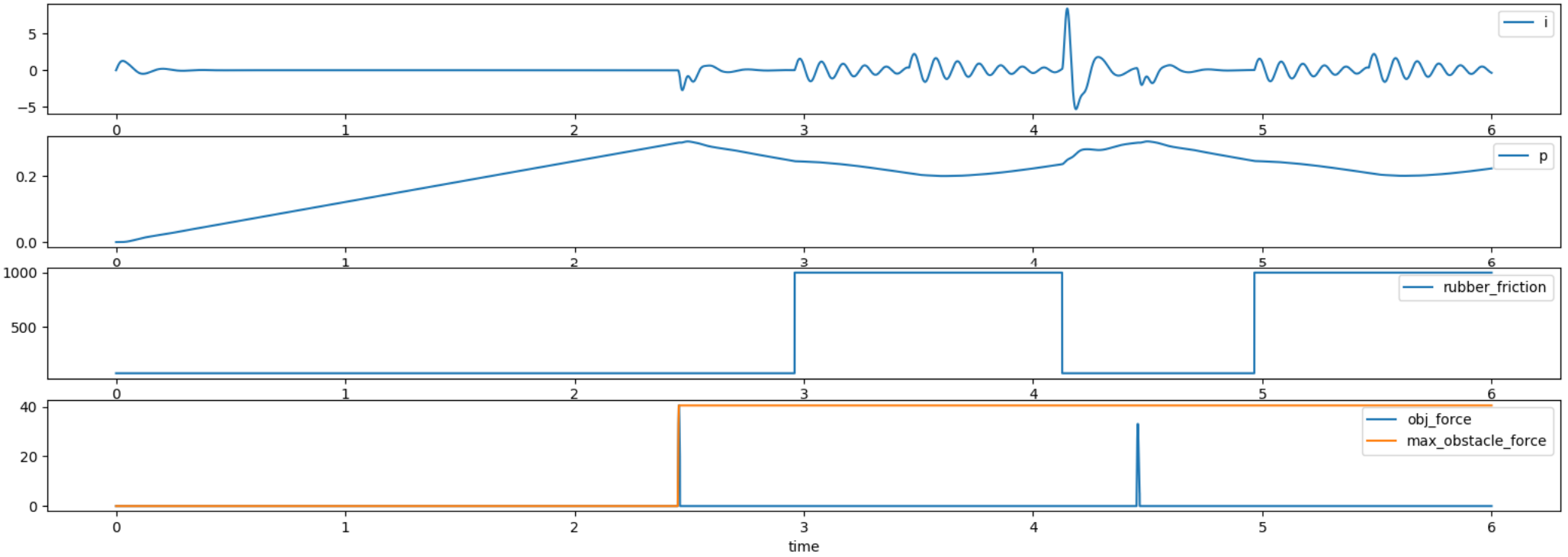


Armature current

Window's position

Rubber friction

Force

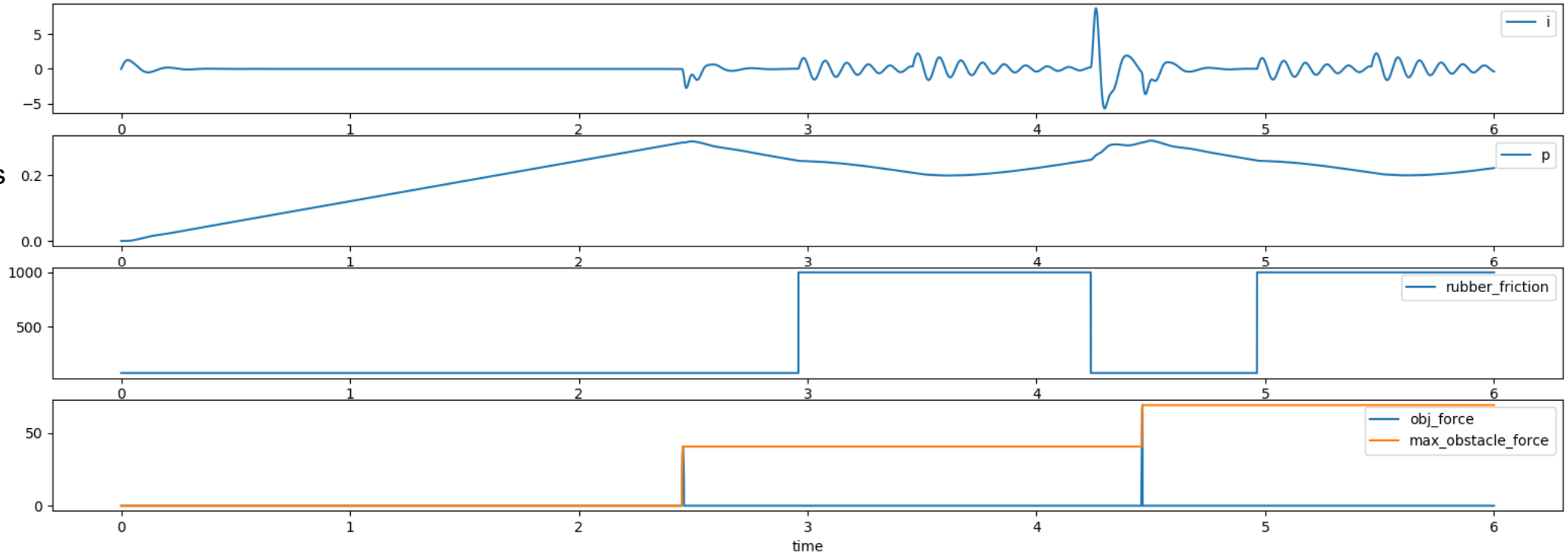


Armature current

Window's position

Rubber friction

Force

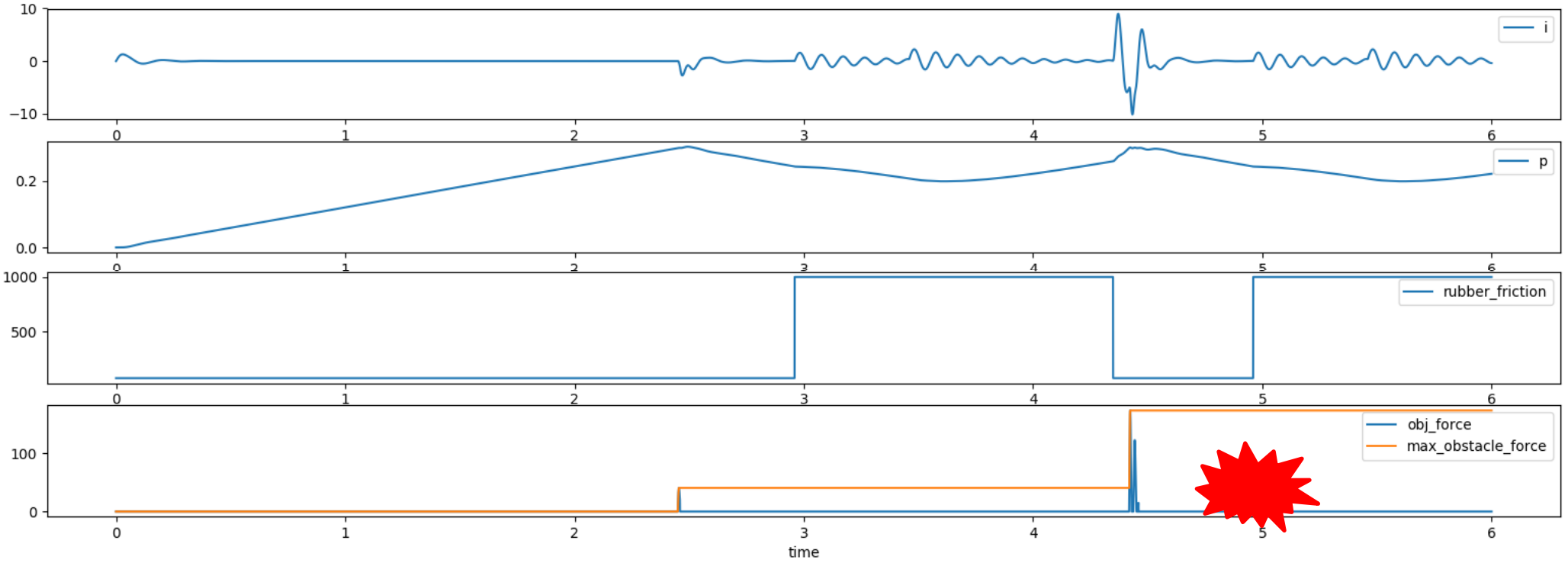


Armature current

Window's position

Rubber friction

Force

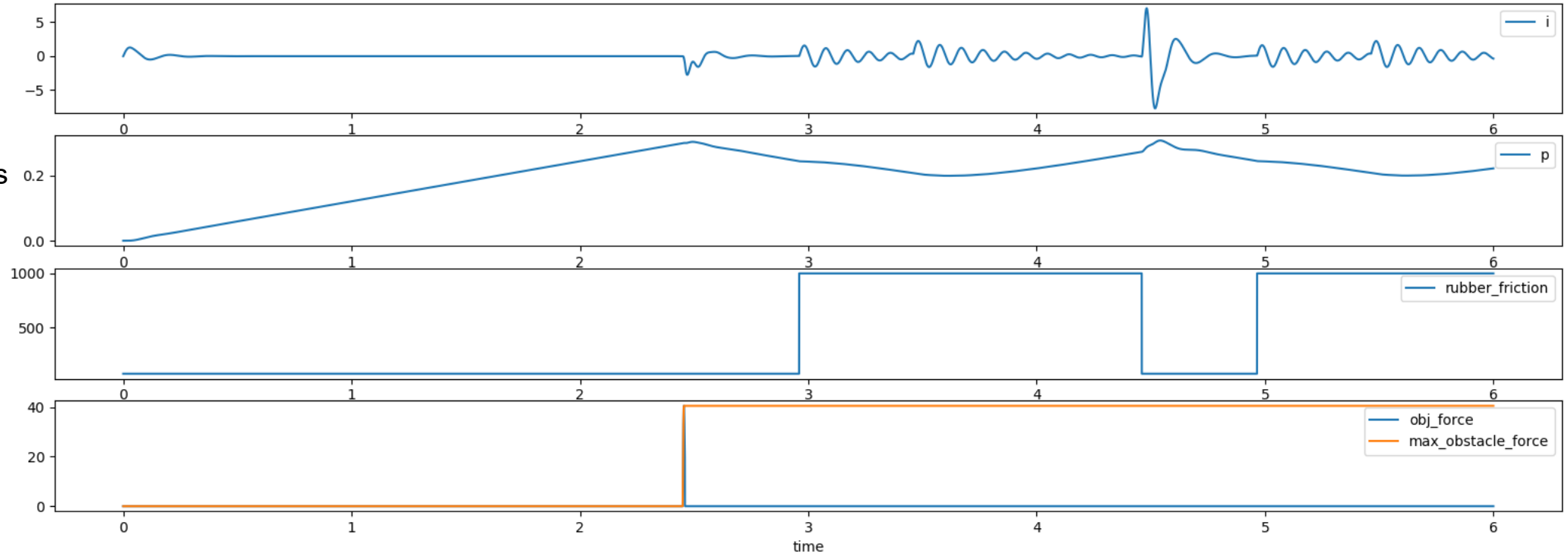


Armature current

Window's position

Rubber friction

Force



Summary

- Applying fault injection to FMI
- Using sensitivity analysis to relate system trajectories to fault parameters

Next steps

- Co-simulation effect
  - Delay due to step-size
- How to **observe discrete** modes of the hybrid system in FMI standard
- How to **compute intervals** more effectively
  - Sensitivity equations

Thank you for your attention