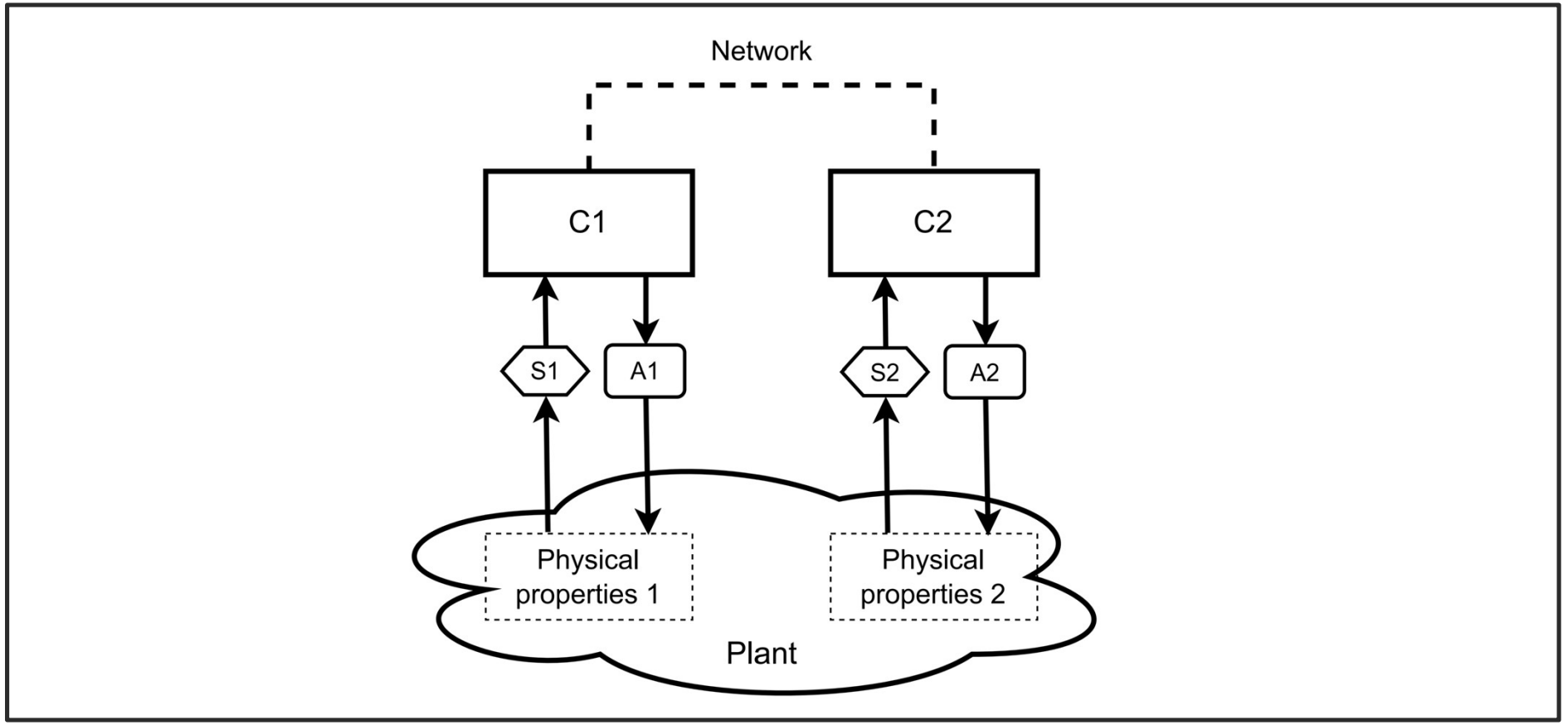
Provable Adversarial Safety in Cyber-Physical Systems

John H. Castellanos (CISPA), Mohamed Maghenem (CNRS), Alvaro A. Cardenas (UCSC), Ricardo G. Sanfelice (UCSC), Jianying Zhou (SUTD)

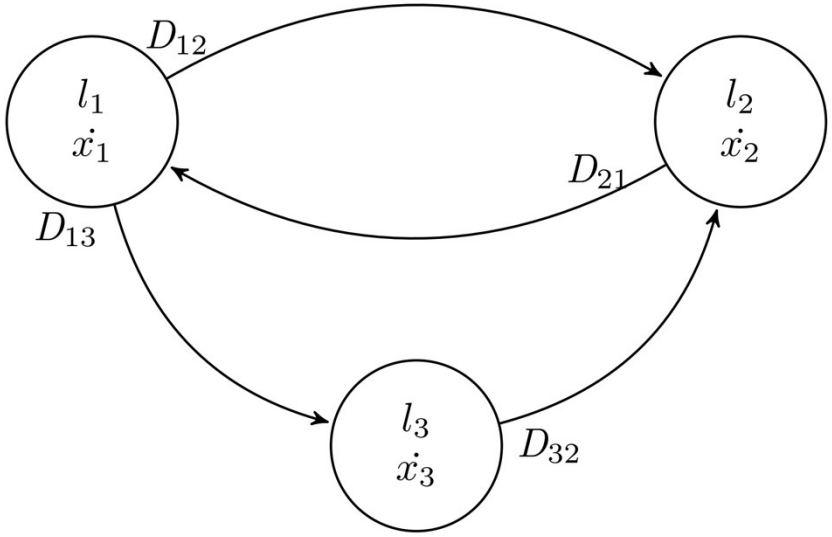
EuroS&P'23 | 06-July-2023

Cyber-physical systems

Cyber-Physical systems

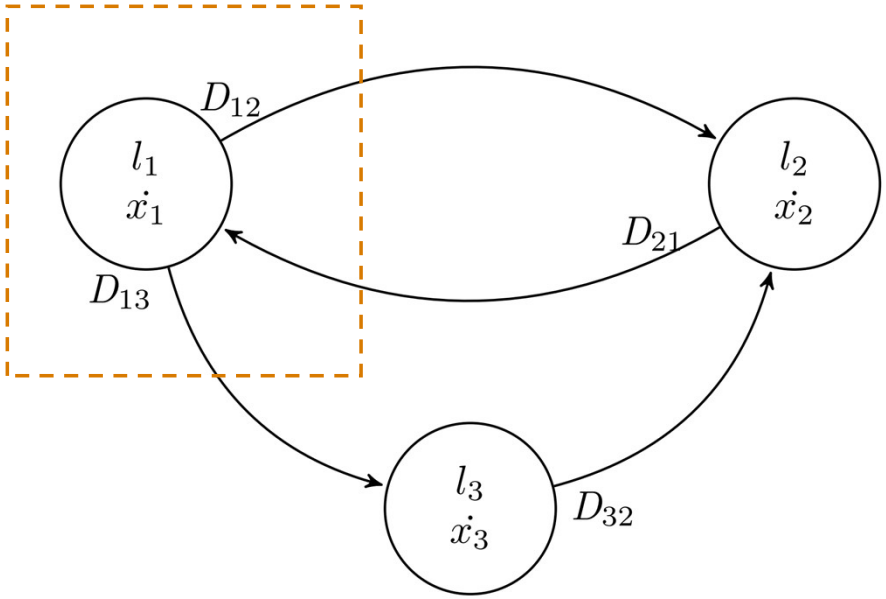


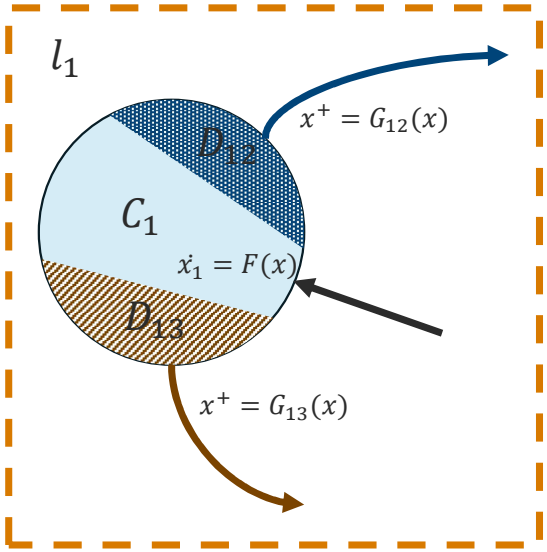
CPS as a hybrid system



$$\mathcal{H} : \begin{cases} \dot{x} = F(x) & x \in C \\ x^+ = G(x) & x \in D, \end{cases}$$

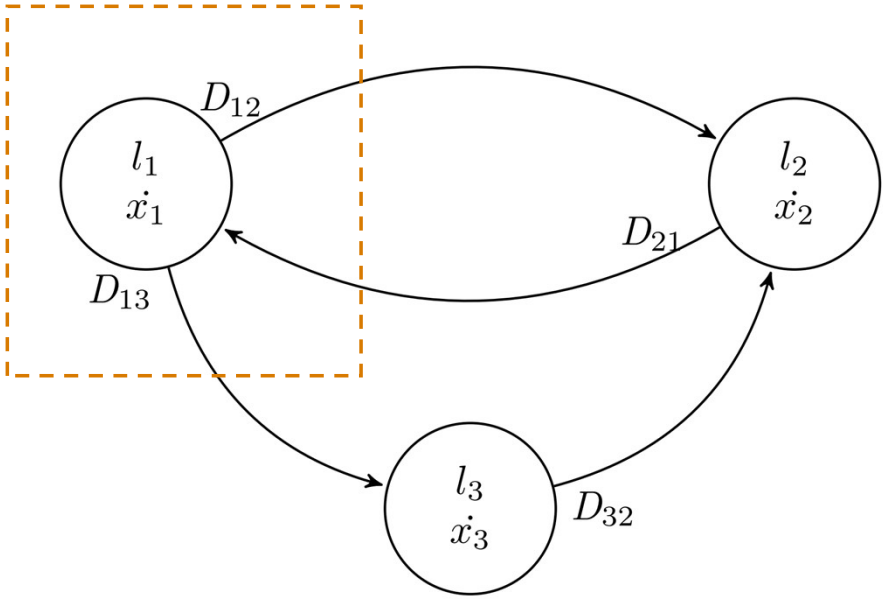
CPS as a hybrid system

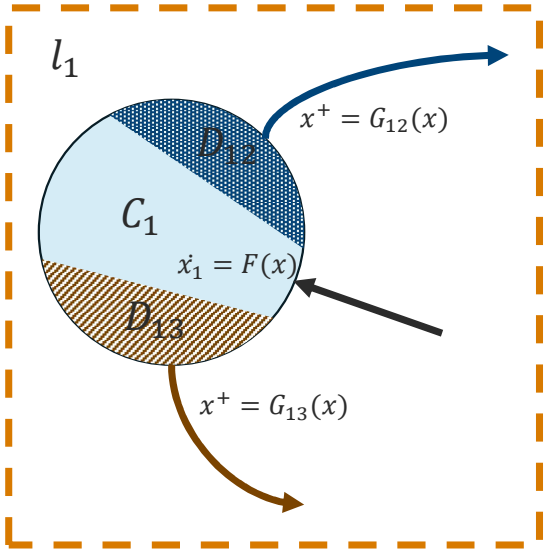




$$\mathcal{H} : \begin{cases} \dot{x} = F(x) & x \in C \\ x^+ = G(x) & x \in D, \end{cases}$$

CPS as a hybrid system

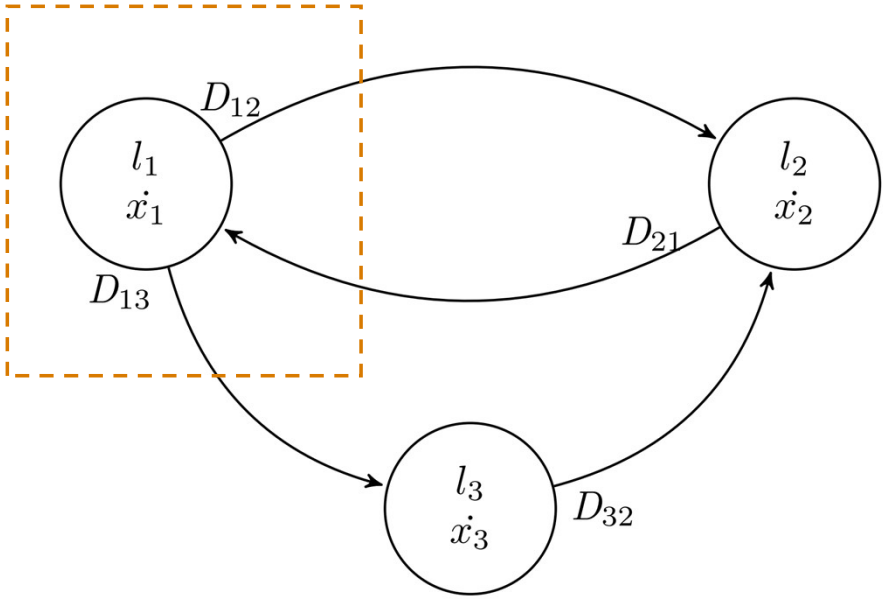


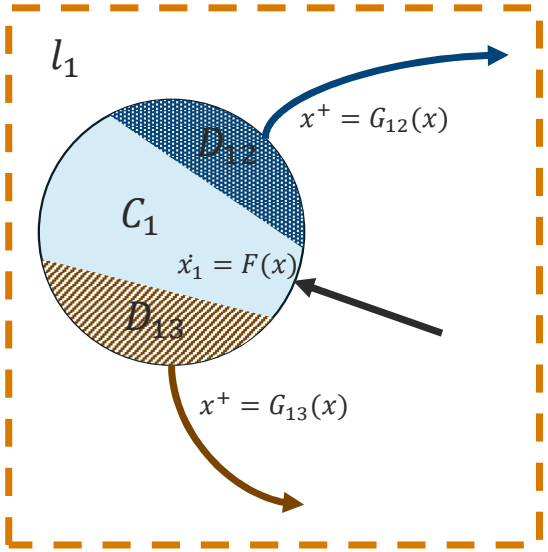


The system flows when $x \in C$ (continuous time)

$$\mathcal{H}: \begin{cases} \dot{x} = F(x) & x \in C \\ x^+ = G(x) & x \in D, \end{cases}$$

CPS as a hybrid system

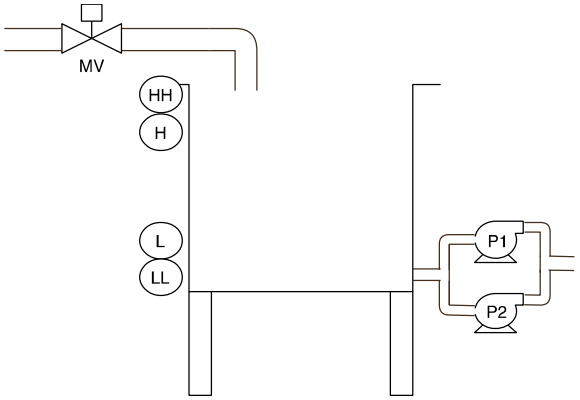


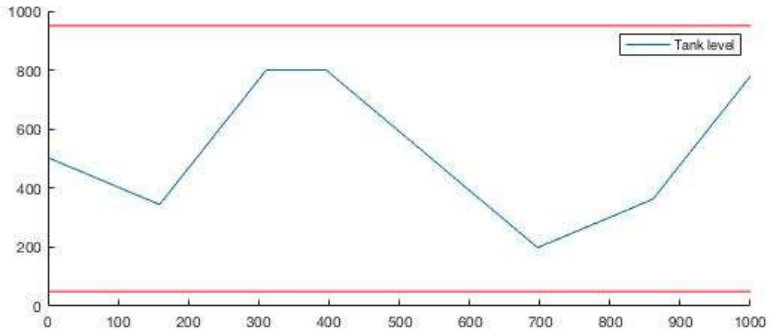


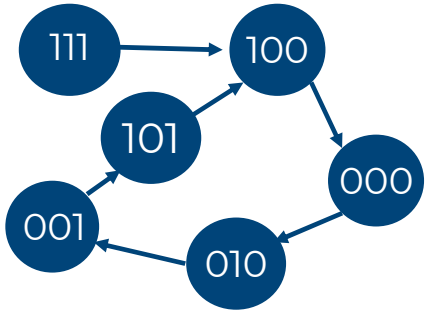
The system jumps when $x \in D$ (discrete time)

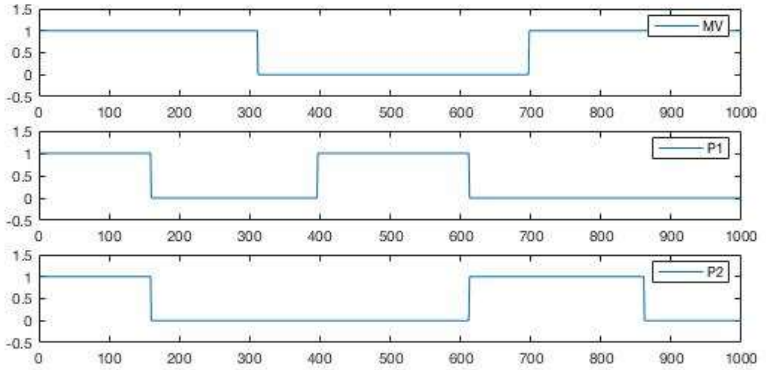
$$\mathcal{H}: \begin{cases} \dot{x} = F(x) & x \in C \\ x^+ = G(x) & x \in D, \end{cases}$$

Simple tank example

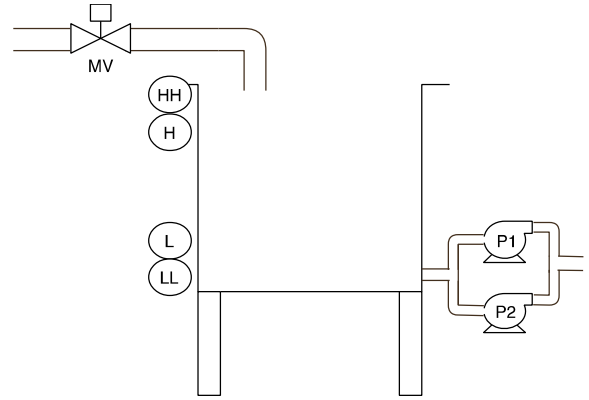


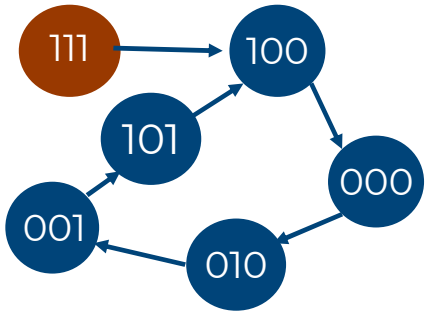


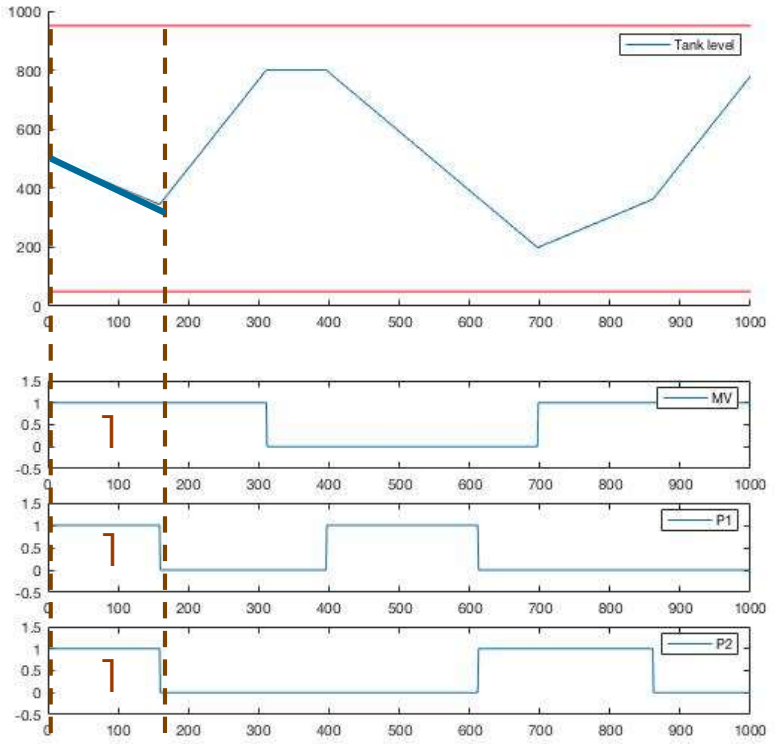




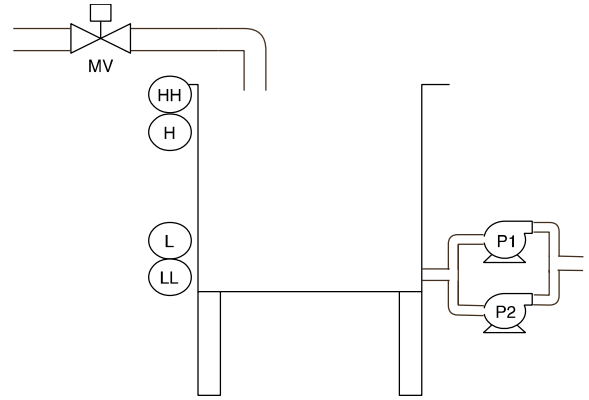
Simple tank example

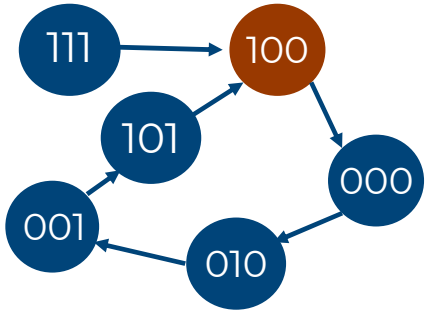


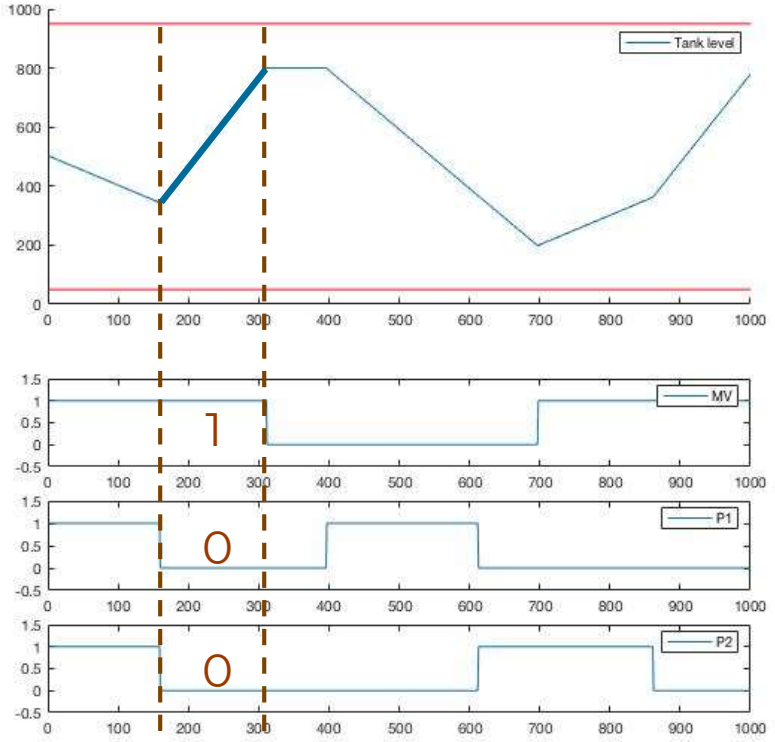




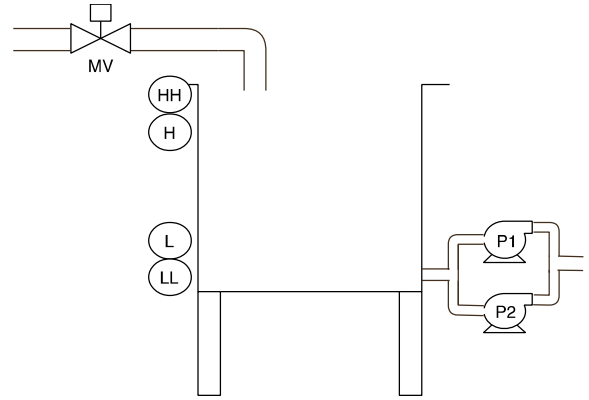
Simple tank example

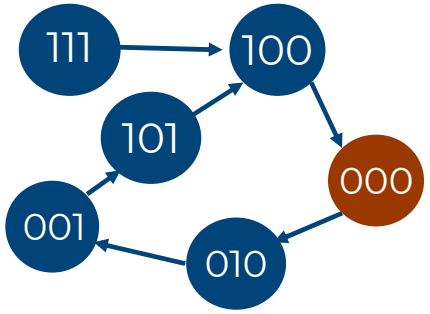


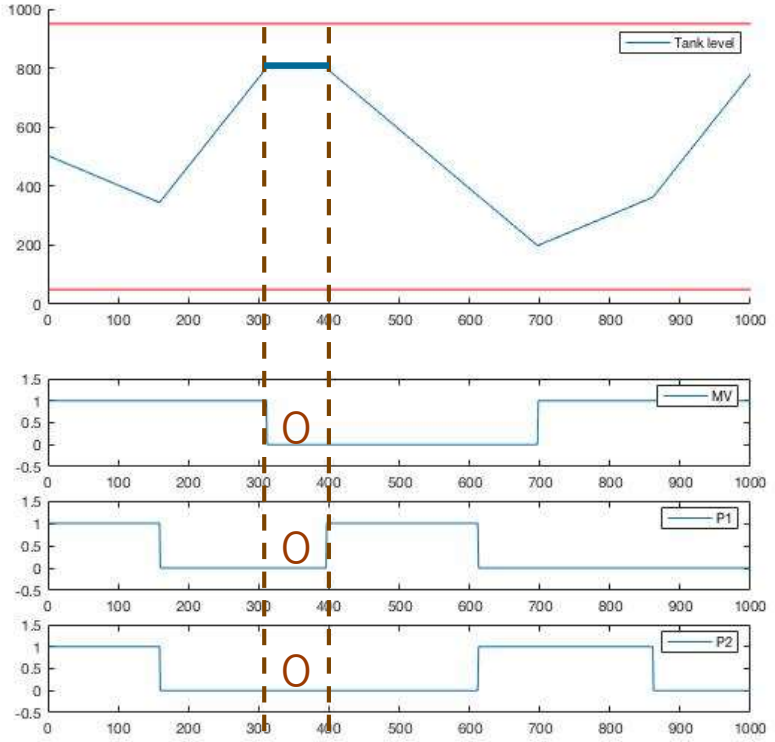




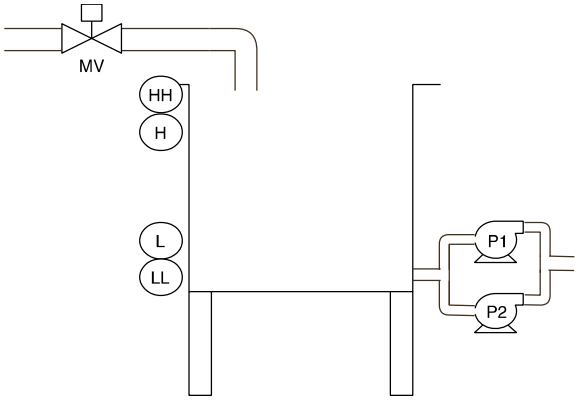
Simple tank example

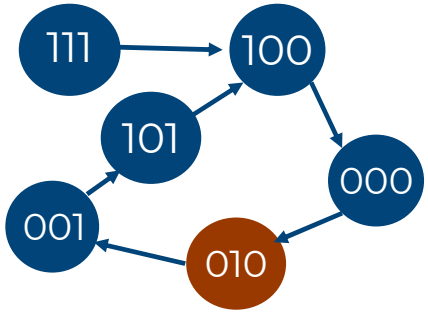


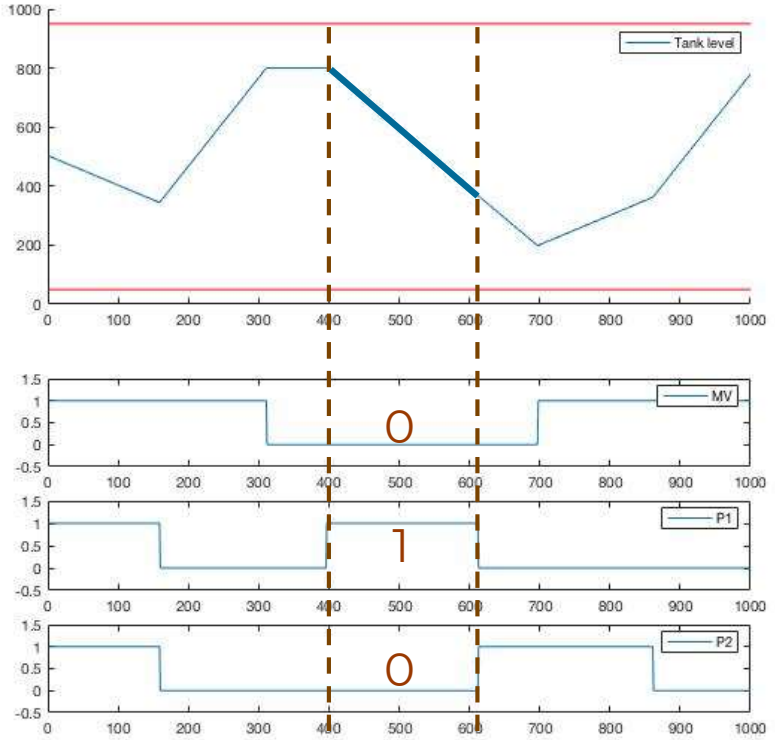




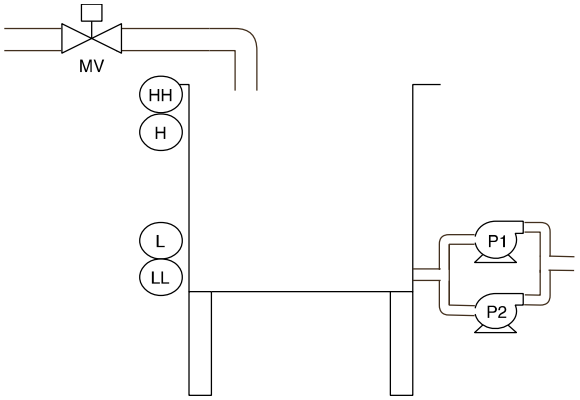
Simple tank example

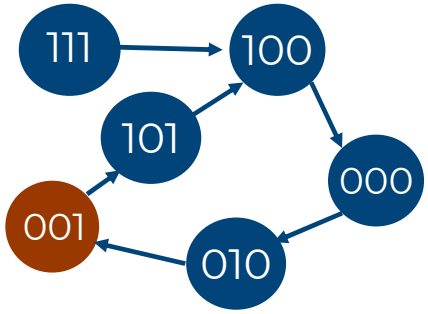


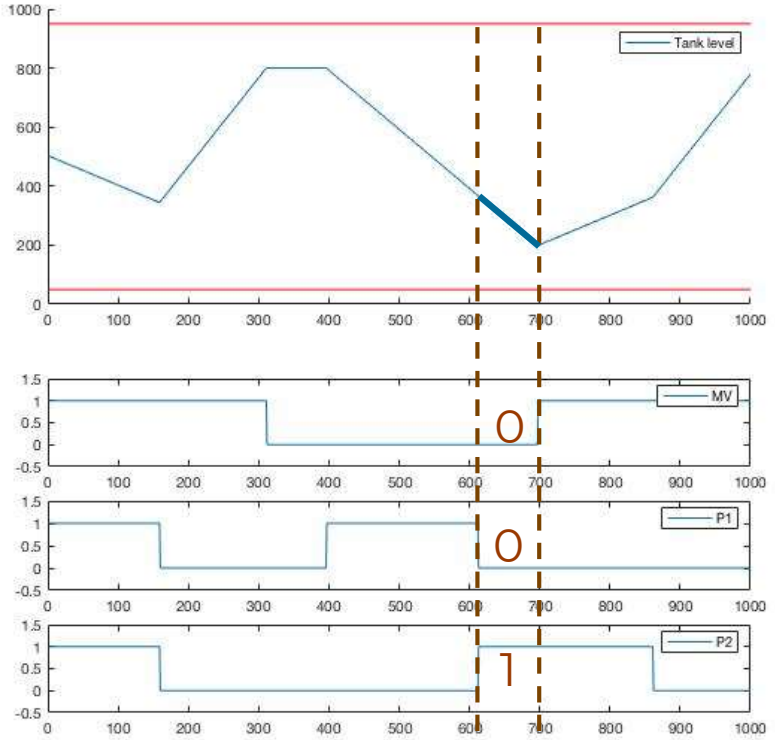




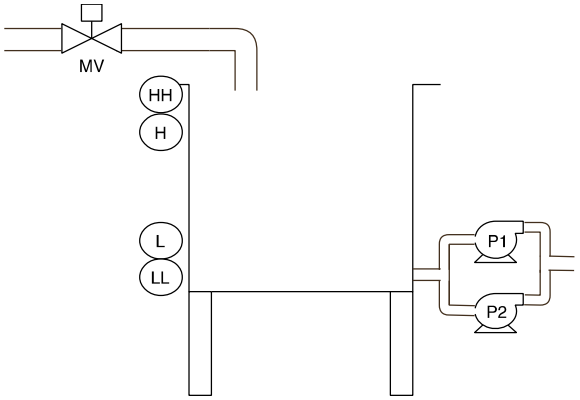
Simple tank example

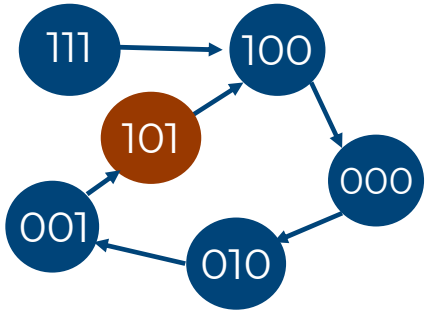


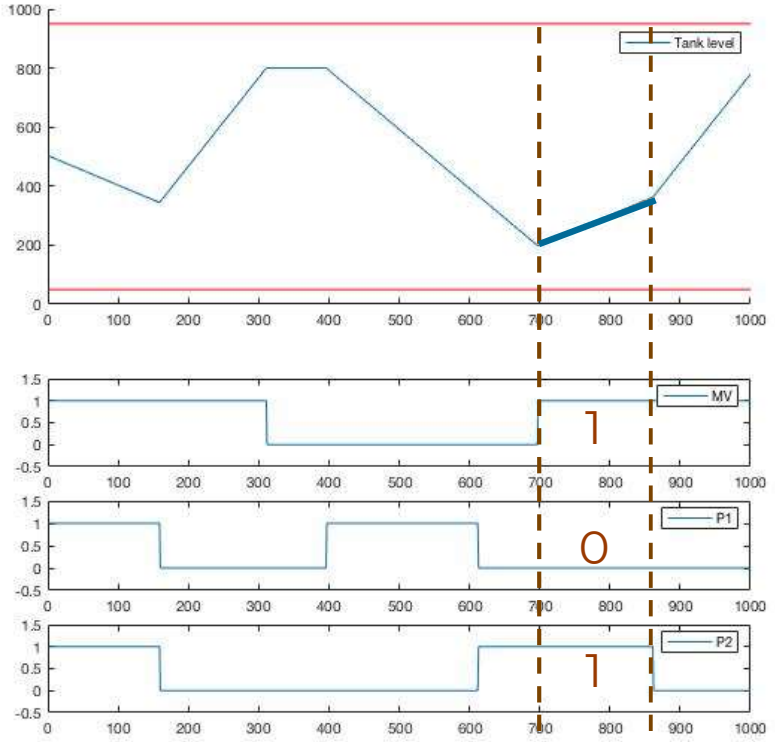




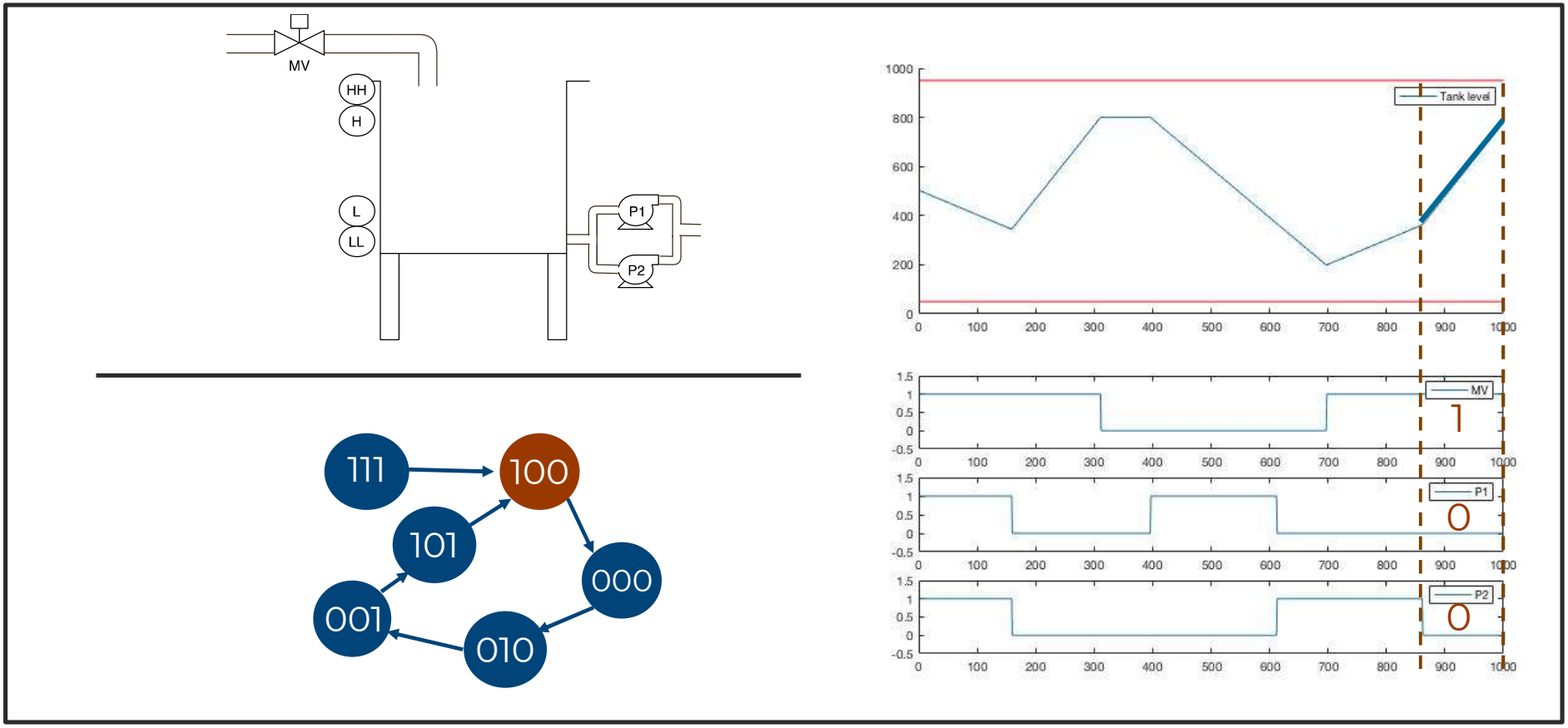
Simple tank example



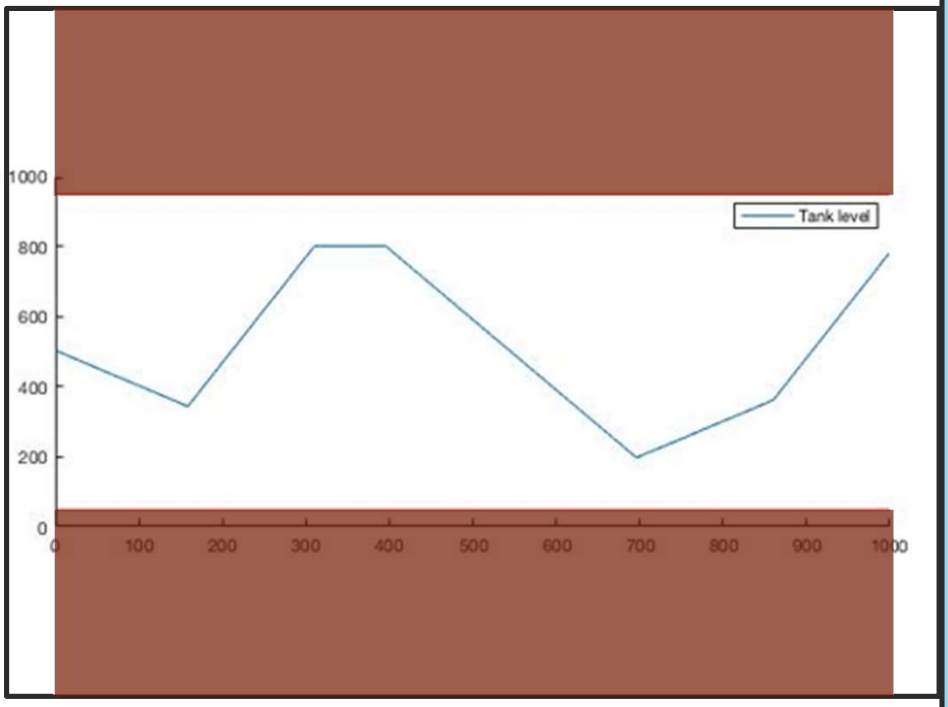




Simple tank example



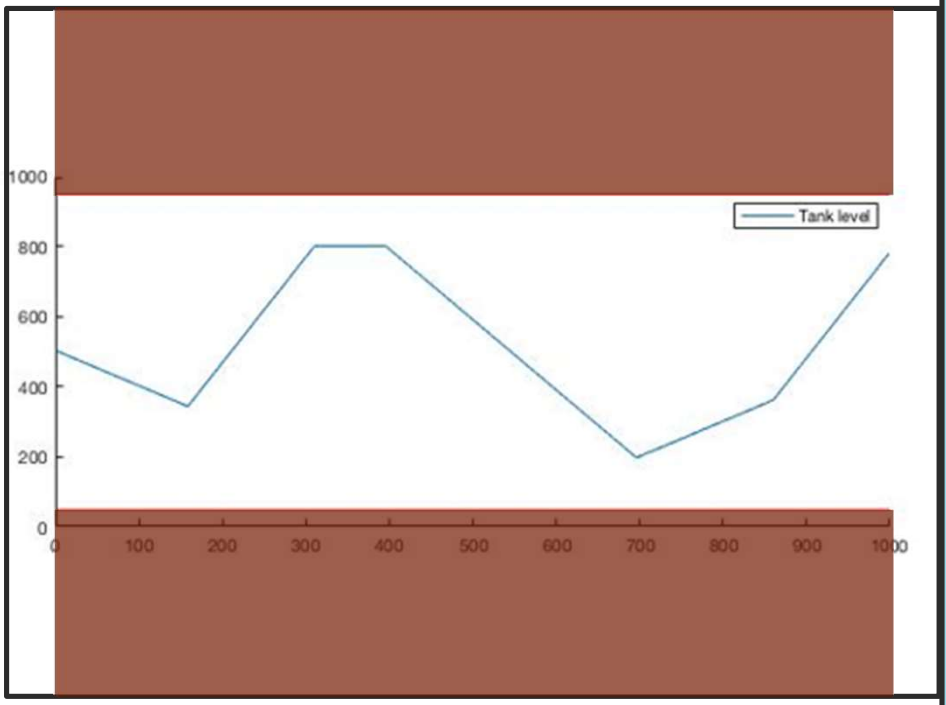
Barrier function candidate



Safe region

Unsafe region

Barrier function candidate

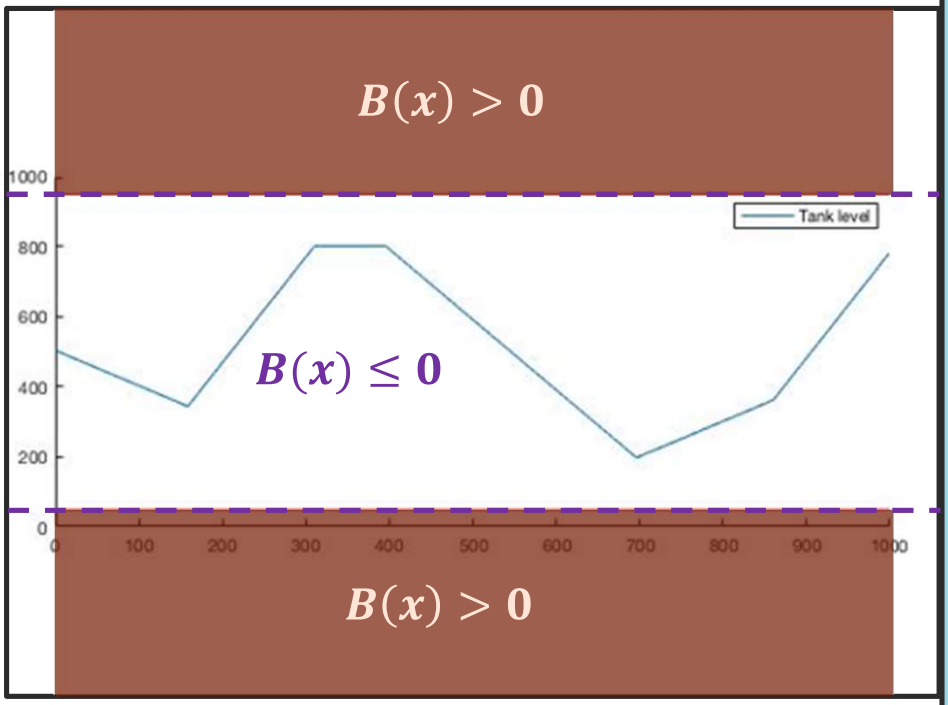


Safe region

Unsafe region

- Barrier function ($B(x)$) takes a positive value when the system is at unsafe region.

Barrier function candidate

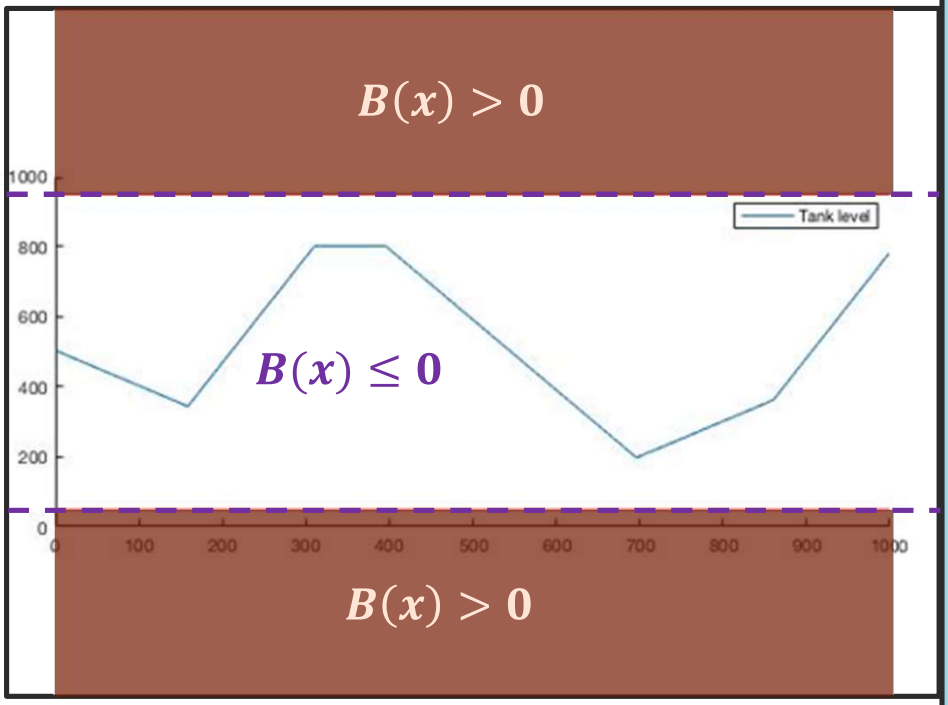


Safe region

Unsafe region

- Barrier function ($B(x)$) takes a positive value when the system is at unsafe region.

Safety analysis

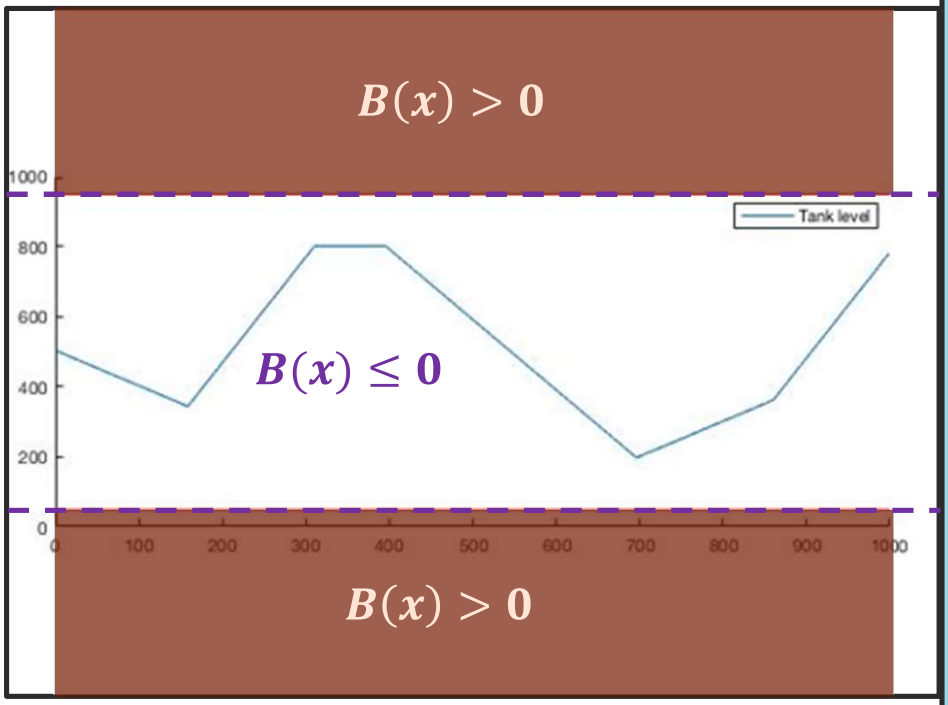


Safe region

Unsafe region

- How to guarantee safety using the barrier certificate $B(x)$?

Safety analysis



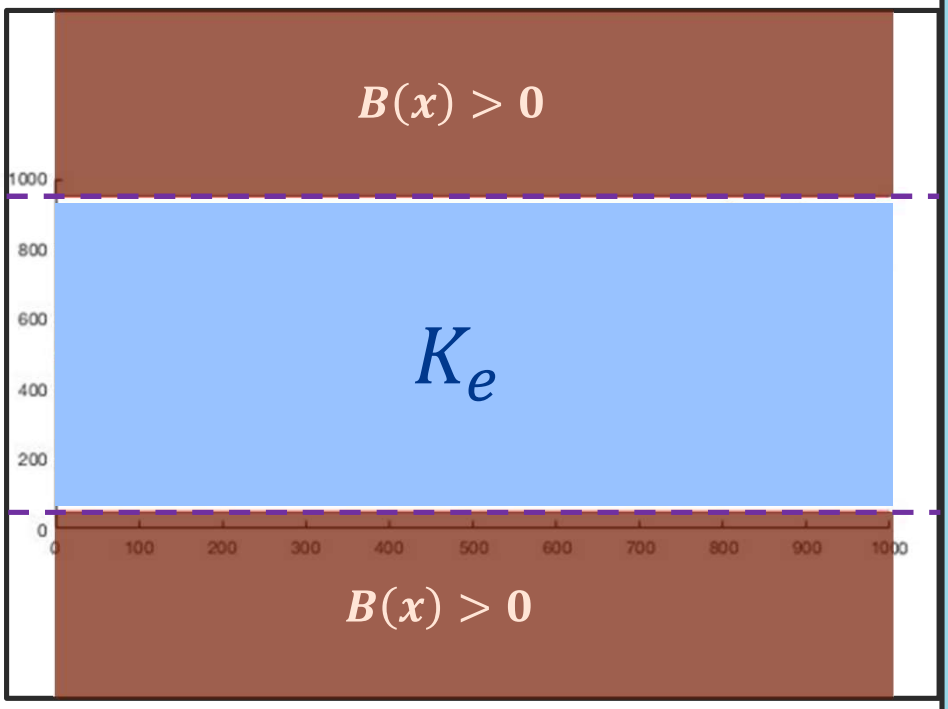
Safe region

Unsafe region

- How to guarantee safety using the barrier certificate $B(x)$?
  - Define the set K_e.

$$K_e := \{x \in X : B(x) \leq 0\}.$$

Safety analysis



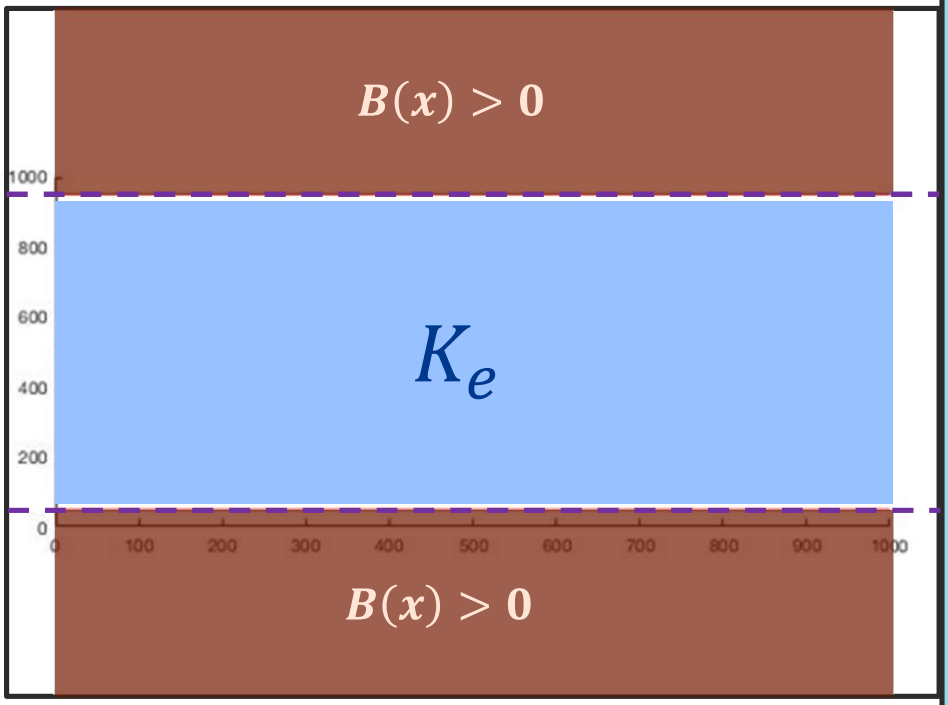
Safe region

Unsafe region

- How to guarantee safety using the barrier certificate $B(x)$?
  - Define the set K_e.

$$K_e := \{x \in X : B(x) \leq 0\}.$$

Safety analysis



Safe region

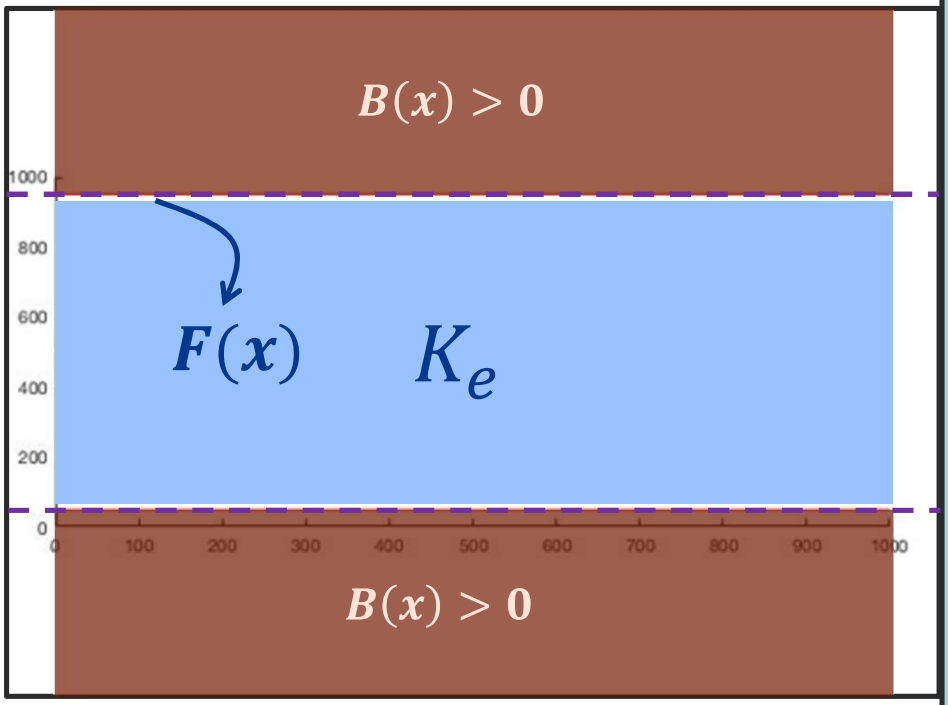
Unsafe region

- How to guarantee safety using the barrier certificate $B(x)$?
  - Define the set K_e.

$$K_e := \{x \in X : B(x) \leq 0\}.$$

  - $\dot{x} = F(x)$ always flows to a safe region on the edge of K_e.

Safety analysis



Safe region

Unsafe region

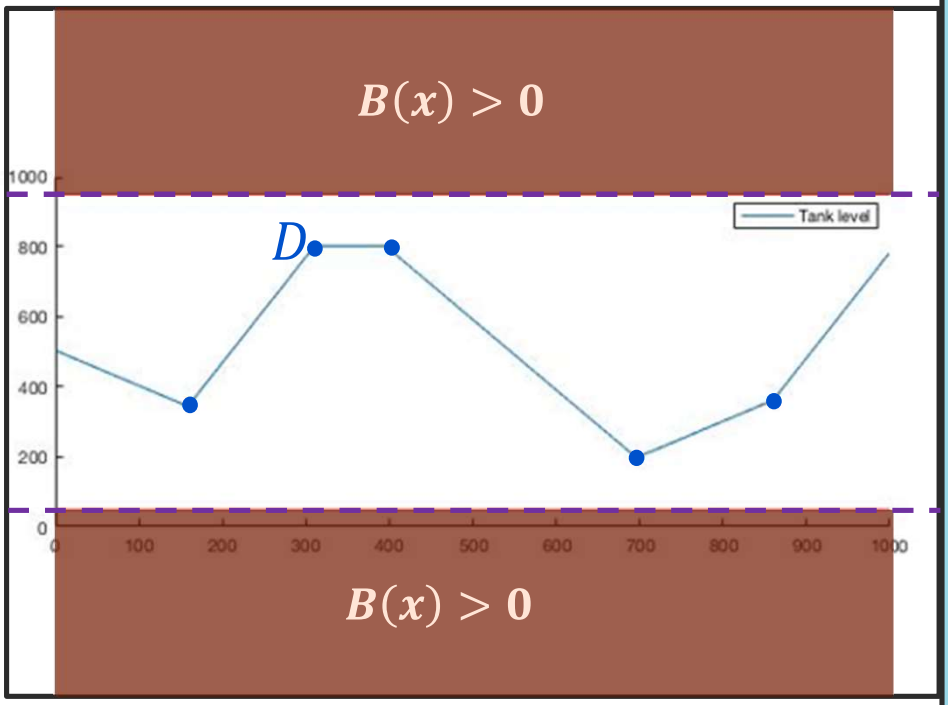
- How to guarantee safety using the barrier certificate $B(x)$?
  - Define the set K_e.

$$K_e := \{x \in X : B(x) \leq 0\}.$$

  - $\dot{x} = F(x)$ always flows to a safe region on the edge of K_e.

$$\langle \nabla B(x), F(x) \rangle \leq 0 \qquad \forall x \in (U(\partial K_e) \backslash K_e) \cap C,$$

Safety analysis



Safe region

Unsafe region

- How to guarantee safety using the barrier certificate $B(x)$?
  - Define the set K_e.

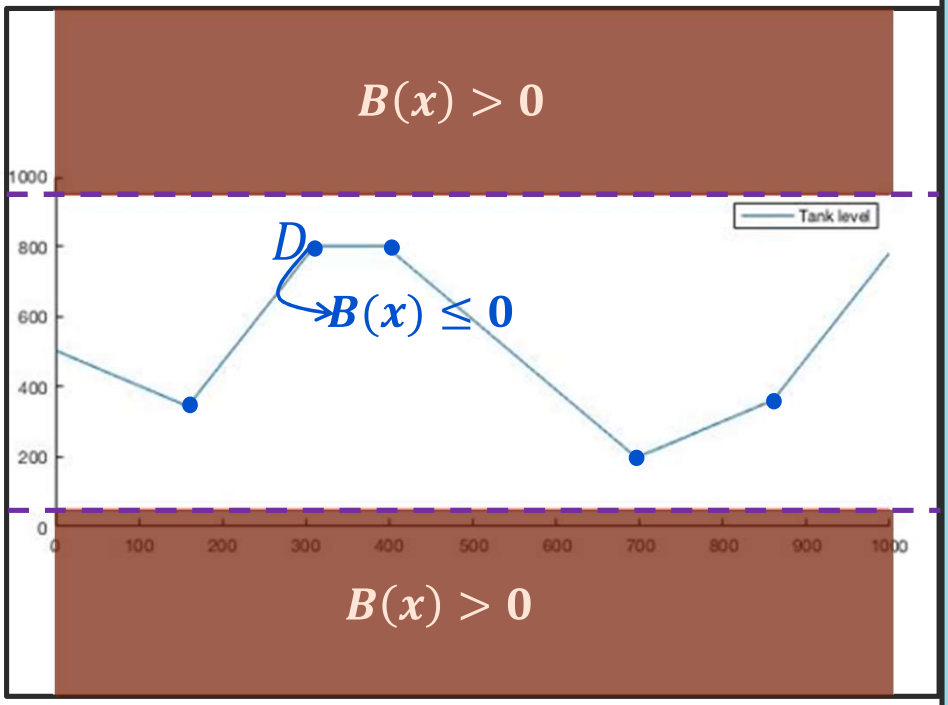
$$K_e := \{x \in X : B(x) \leq 0\}.$$

  - $\dot{x} = F(x)$ always flows to a safe region on the edge of K_e.

$$\langle \nabla B(x), F(x) \rangle \leq 0 \qquad \forall x \in (U(\partial K_e) \backslash K_e) \cap C,$$

  - $B(x)$ is nonpositive in transition D.

Safety analysis



Safe region

Unsafe region

- How to guarantee safety using the barrier certificate $B(x)$?
  - Define the set K_e.

$$K_e := \{x \in X : B(x) \leq 0\}.$$

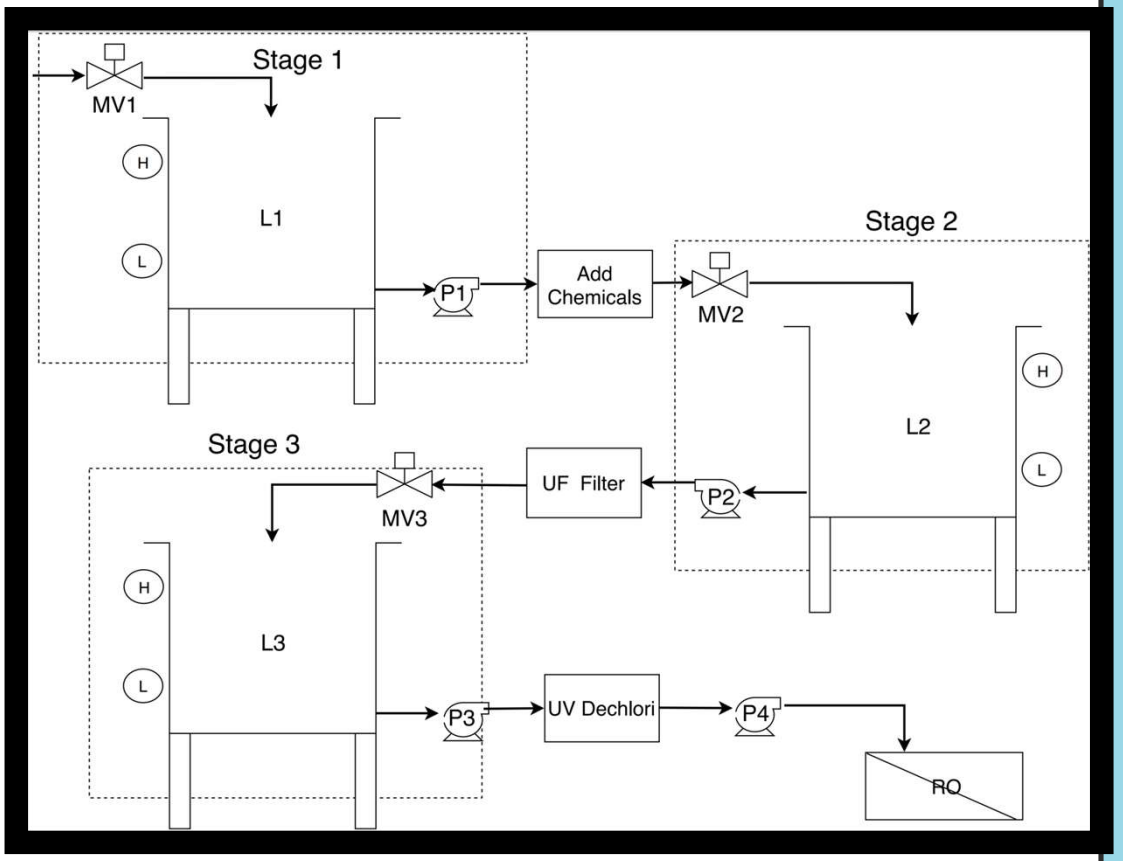
  - $\dot{x} = F(x)$ always flows to a safe region on the edge of K_e.

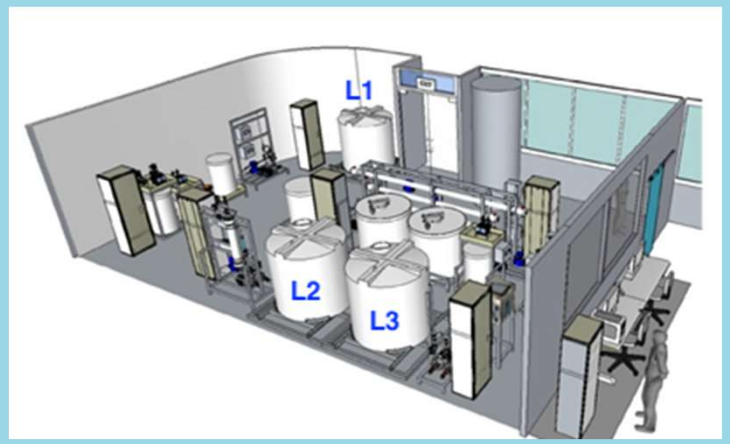
$$\langle \nabla B(x), F(x) \rangle \leq 0 \qquad \forall x \in (U(\partial K_e) \backslash K_e) \cap C,$$

  - $B(x)$ is nonpositive in transition D.

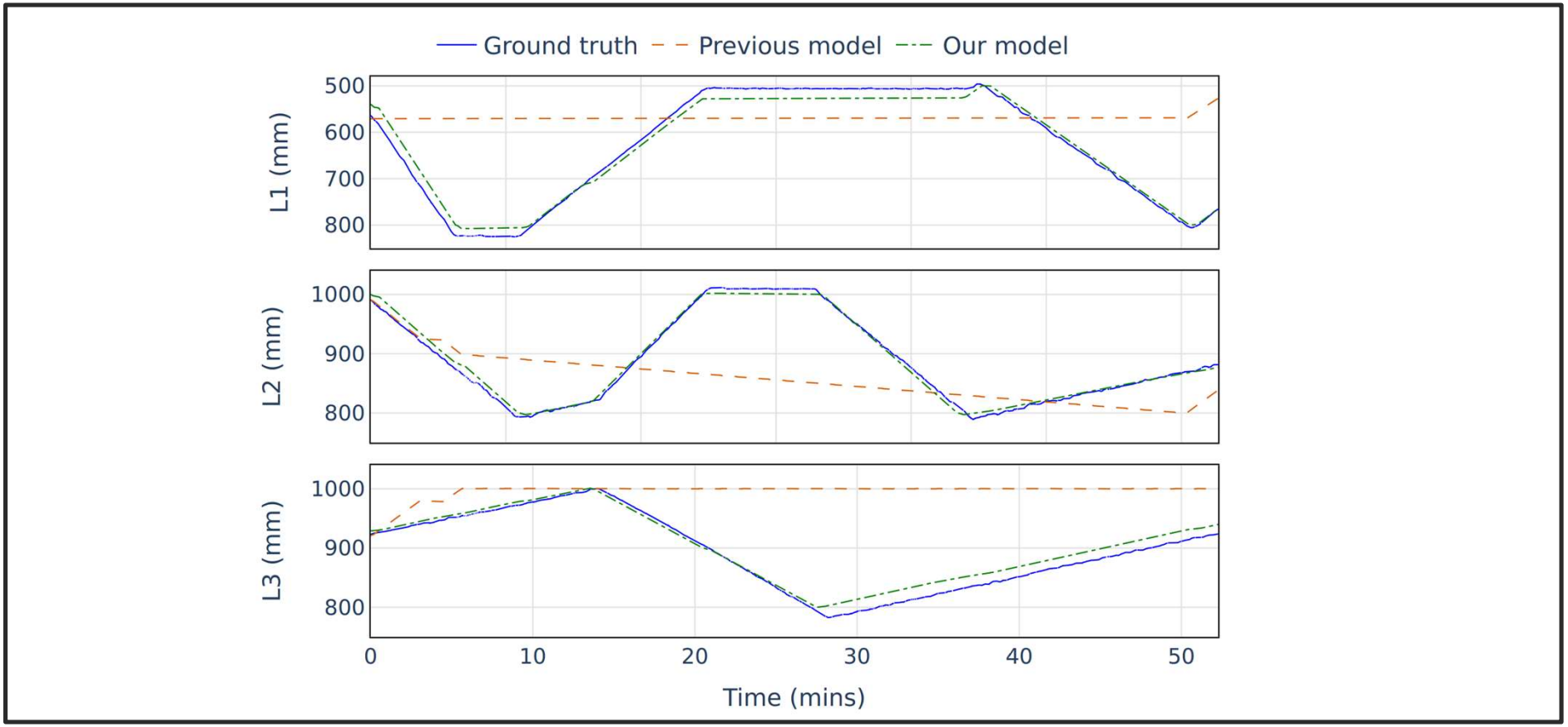
$$B(G(x)) \leq 0 \qquad \forall x \in D \cap K_e,$$

Use case: SWaT

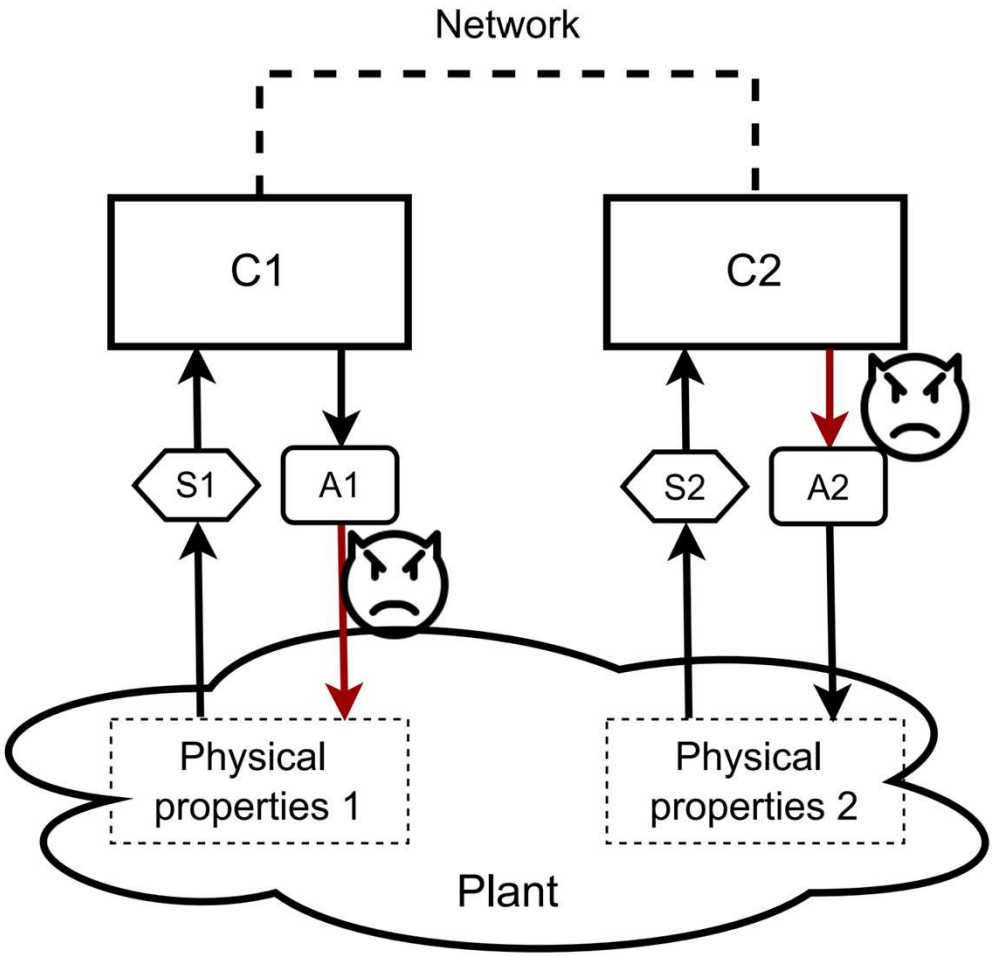




Modeling SWaT

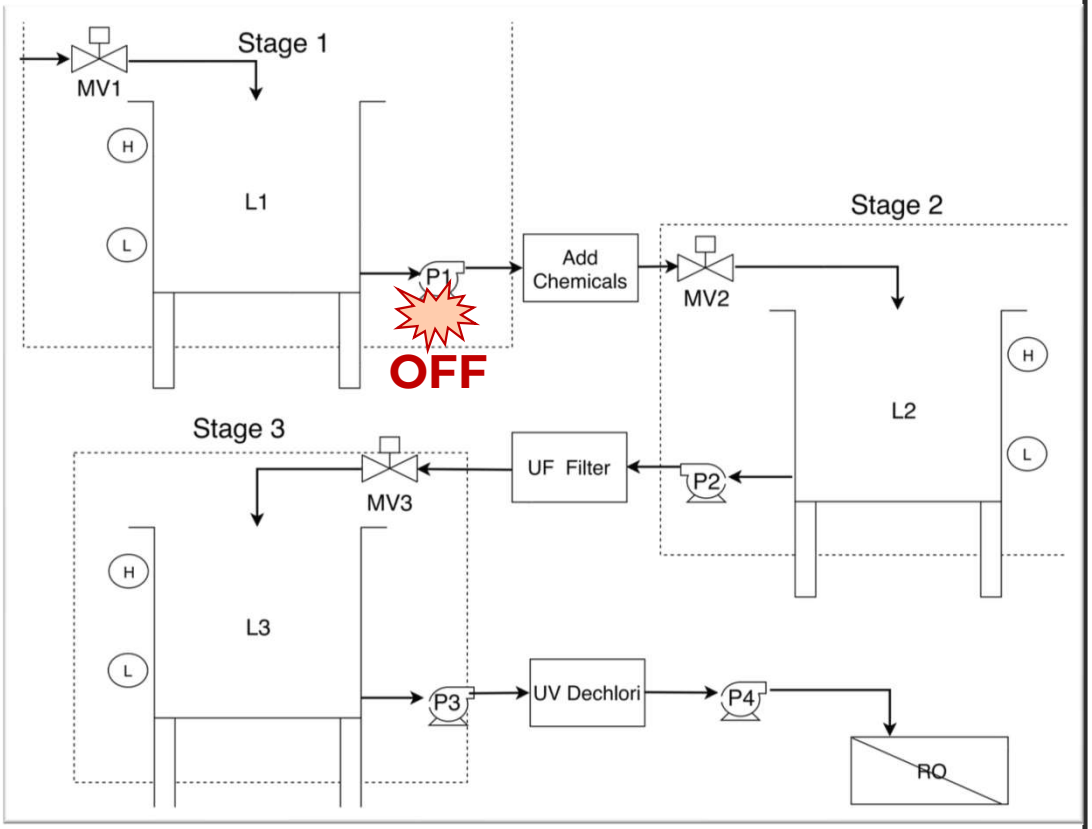


Threat model

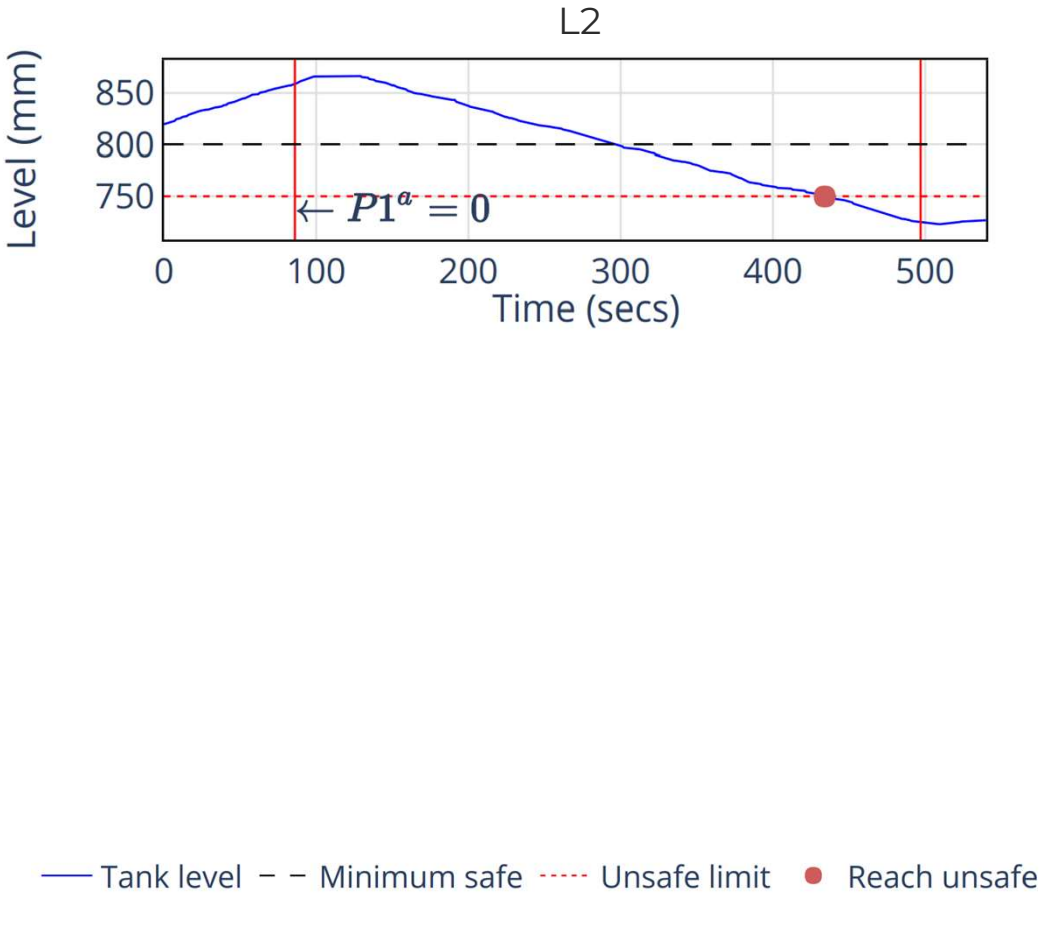


- Attackers affect only actuators.
- Attackers compromise one actuator at a time.

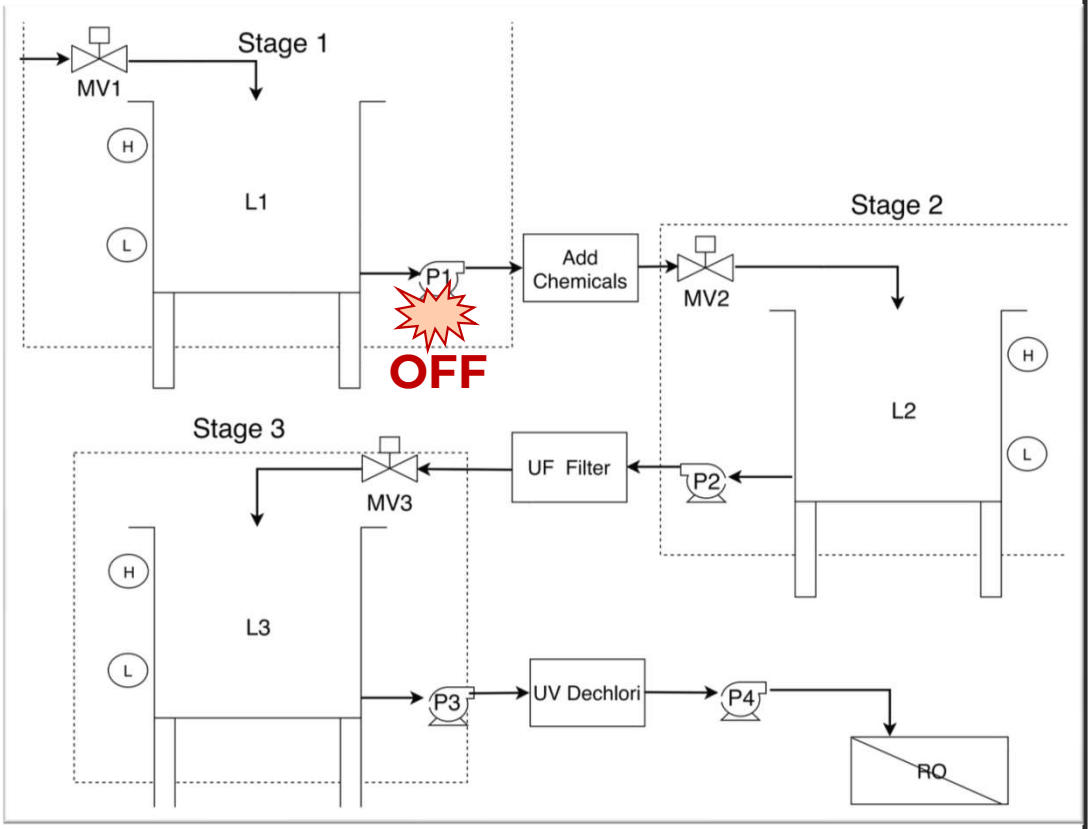
Attacking P1



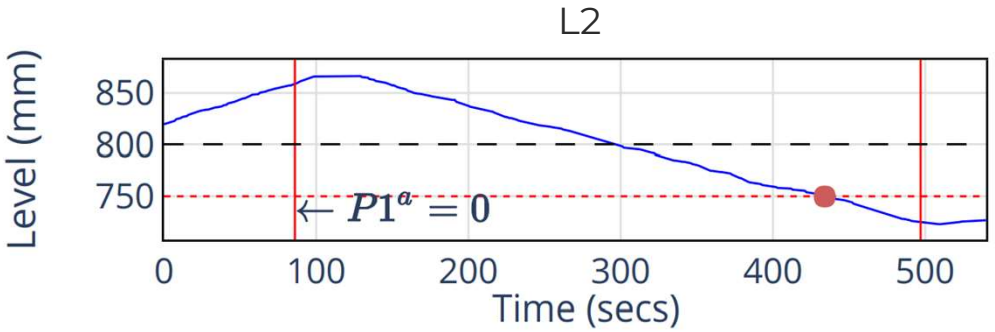
Unsafe controller



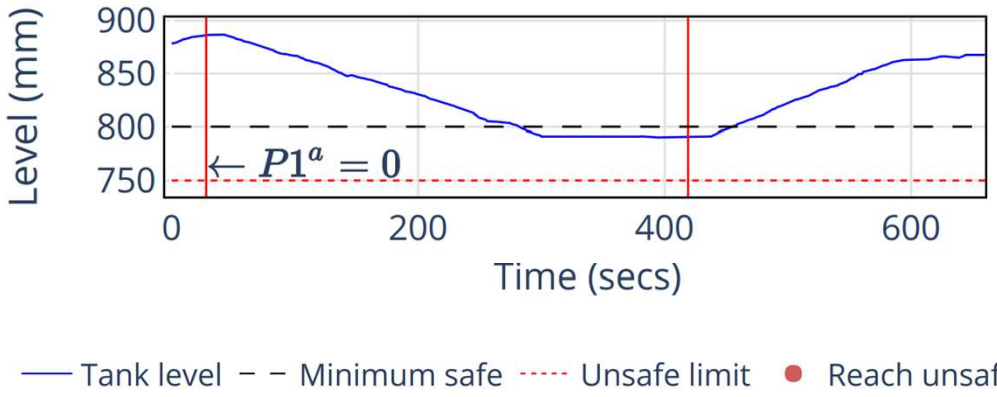
Attacking P1



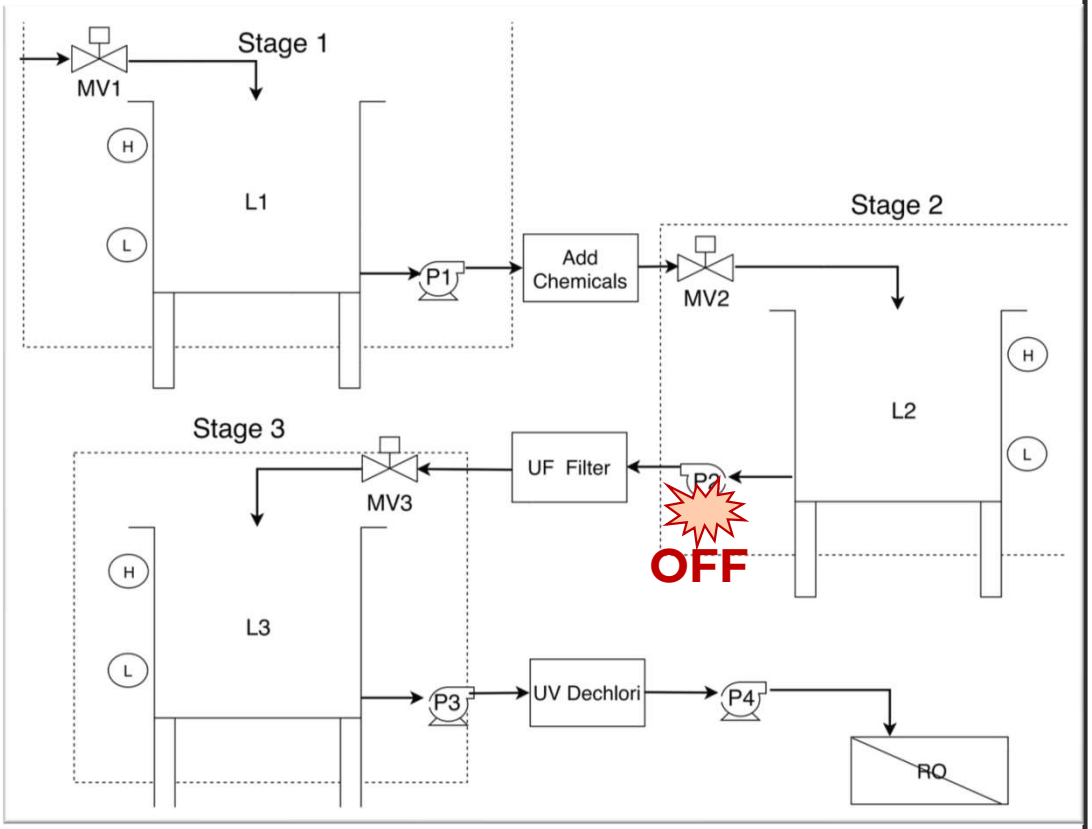
Unsafe controller



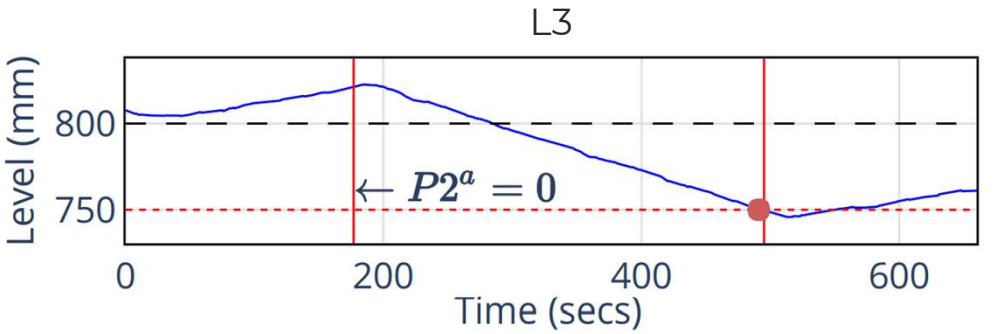
Harden controller



Attacking P2

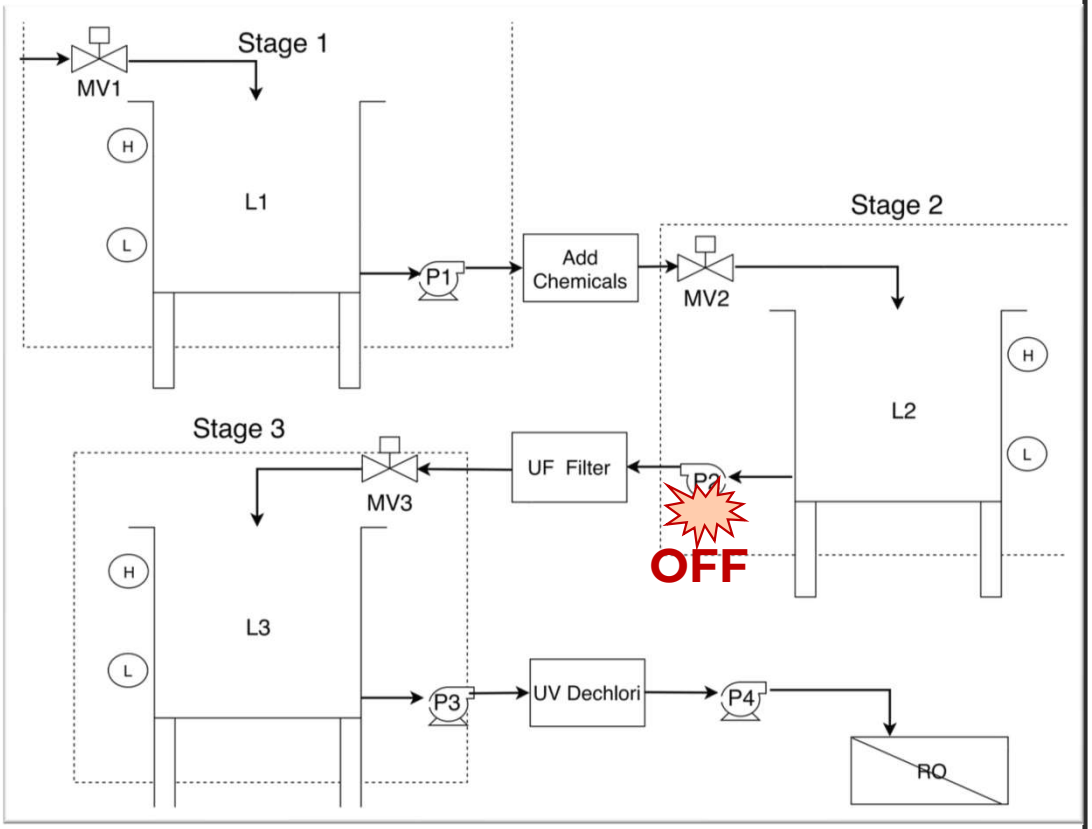


Unsafe controller

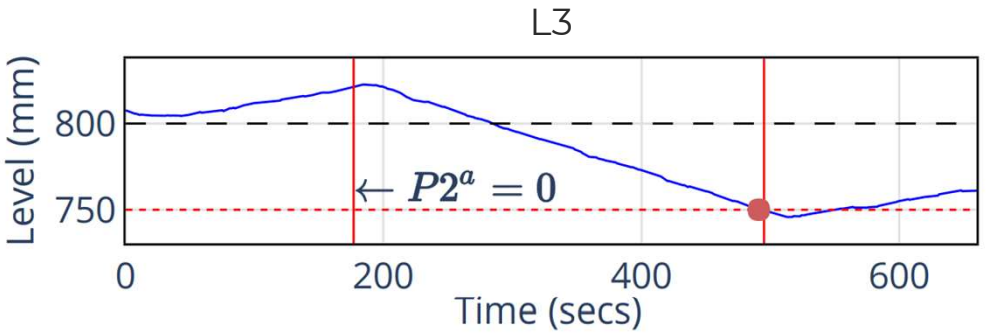


Tank level Minimum safe Unsafe limit Reach unsafe

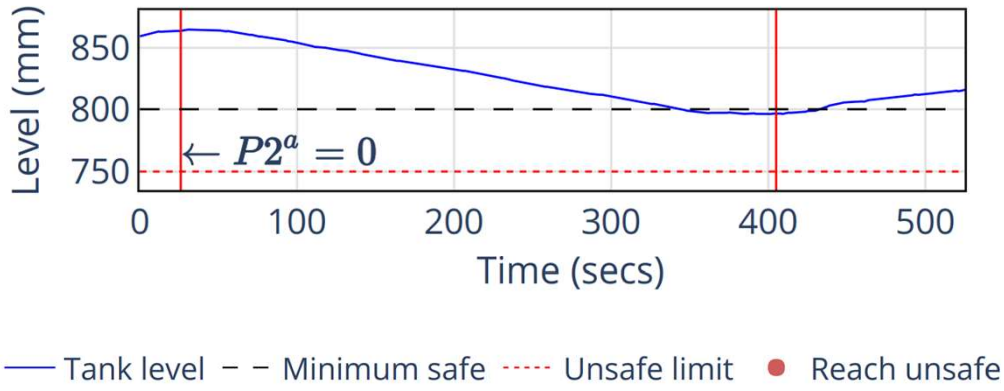
Attacking P2



Unsafe controller



Harden controller



Conclusion