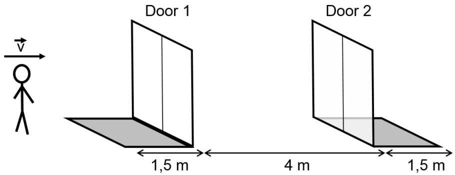
# Embedded systems
# Exercise session, 13/12
# Hybrid systems

# Guidelines

1. Read carefully the problem statement.
2. For unspecified details, make reasonable choices.
3. Decompose the problem into some number of processes.
4. Choose your variables and **document them**.
5. For each process, determine the number of control locations needed to represent the possible behaviors.
6. Organize the communication between processes.
7. Write the invariants, activities, guards and actions.

# Problem 1

A microcontroller controls two security doors at the entrance of a bank, located 4 meters apart. Both doors are equipped with a sensor that is able to detect people. The range of the sensors is depicted here:



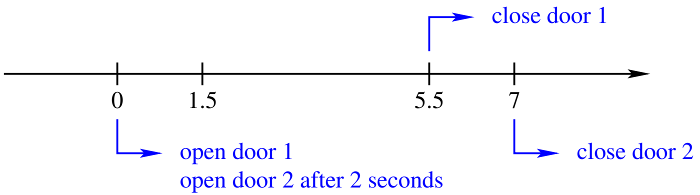

It is assumed that people always move from left to right, i.e., that they first go through door 1 and then through door 2. The microcontroller controls the doors as follows:

- Door 1 opens as soon as its sensor detects someone, at a rate of 67% per second (in other words, it takes about 1.5 s for it to become fully open). It closes, at a rate of 50% per second, when the sensor of door 2 picks up someone.
- Door 2 starts to open, at a rate of 50% per second, exactly 2 seconds after the sensor of door 1 has detected someone. It closes, at the same rate, when the sensor of door 2 stops sensing.

A person can pass through a door only if it is at least 50% open. Otherwise, he/she waits for the door to open enough. It is assumed that at most one person can use the doors at any time, in other words, a new person may approach door 1 only after the previous one has left the sensing area of door 2.

1. Describe a hybrid system modelling this situation.
2. Give the first 3 steps of the space-state exploration of this system, in the case of a person moving at a speed of 1.75 m/s.

Problem digest:




- At least 7 meters between two successive persons.
- No assumption on the speed of the person (other than $v > 0$).

Processes and variables:

- $P_1$: movement of the person, $x_1 =$ current position.
- $P_2$: door 1, $x_2 =$ percentage open (0–100).
- $P_3$: door 2, $x_3 =$ percentage open (0–100).
- $P_4$: 2-second timer, $x_4 =$ elapsed time.

Initial conditions:

- The person is out ($x_1 \leq 0$).
- The doors are closed.

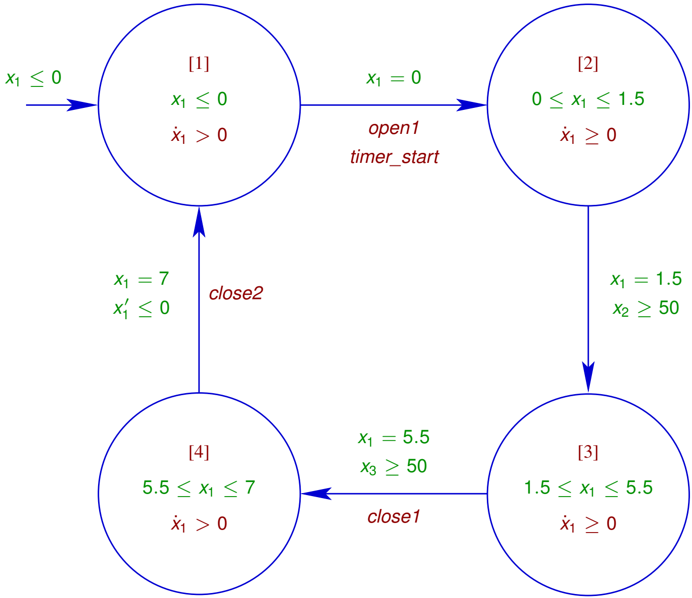
## Process 1

### Modes of operation:

- [1]: $x_1 \leq 0$ or $x_1 \geq 7$
- [2]: $0 \leq x_1 \leq 1.5$
- [3]: $1.5 \leq x_1 \leq 5.5$
- [4]: $5.5 \leq x_1 \leq 7$

### Communication:

- Values of $x_2$ and $x_3$ are read to pass doors.
- Synchronization labels *open1*, *close1* with $P_2$, *close2* with $P_3$, *timer_start* with $P_4$.



$x_1 \leq 0$
[1]
$x_1 \leq 0$
$\dot{x}_1 > 0$
$x_1 = 0$
open1
timer_start
[2]
$0 \leq x_1 \leq 1.5$
$\dot{x}_1 \geq 0$
$x_1 = 1.5$
$x_2 \geq 50$
[3]
$1.5 \leq x_1 \leq 5.5$
$\dot{x}_1 \geq 0$
$x_1 = 5.5$
$x_3 \geq 50$
close1
[4]
$5.5 \leq x_1 \leq 7$
$\dot{x}_1 > 0$
$x_1 = 7$
$x_1' \leq 0$
close2

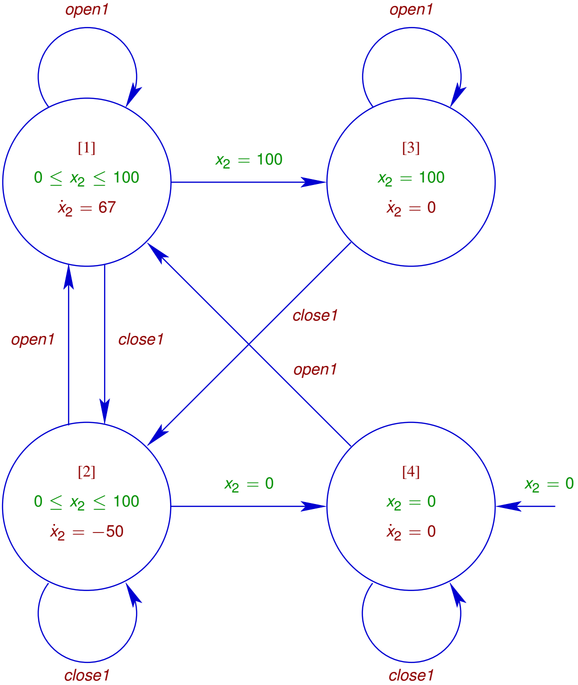
## Process 2

### Modes of operation:

- [1]: opening door
- [2]: closing door
- [3]: door open
- [4]: door closed

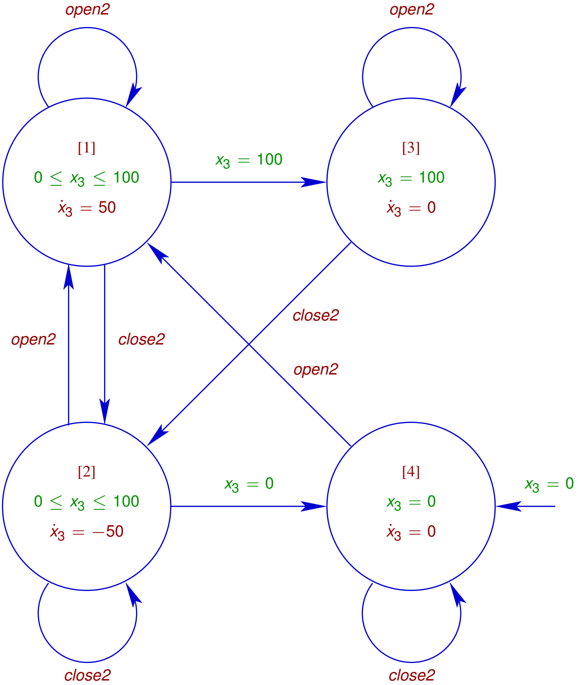
### Communication:

- Provides the value of $x_2$.
- Reacts to the labels *open1* and *close1*.


open1
open1
[1]
$0 \leq x_2 \leq 100$
$\dot{x}_2 = 67$
$x_2 = 100$
[3]
$x_2 = 100$
$\dot{x}_2 = 0$
close1
open1
close1
open1
[2]
$0 \leq x_2 \leq 100$
$\dot{x}_2 = -50$
$x_2 = 0$
[4]
$x_2 = 0$
$\dot{x}_2 = 0$
$x_2 = 0$
close1
close1

## Process 3

Similar to Process 2, with $x_2$ replaced by $x_3$ and *open1*/*close1* by *open2*/*close2*.



open2
open2
[1]
$0 \leq x_3 \leq 100$
$\dot{x}_3 = 50$
$x_3 = 100$
[3]
$x_3 = 100$
$\dot{x}_3 = 0$
close2
open2
close2
open2
[2]
$0 \leq x_3 \leq 100$
$\dot{x}_3 = -50$
$x_3 = 0$
[4]
$x_3 = 0$
$\dot{x}_3 = 0$
$x_3 = 0$
close2
close2

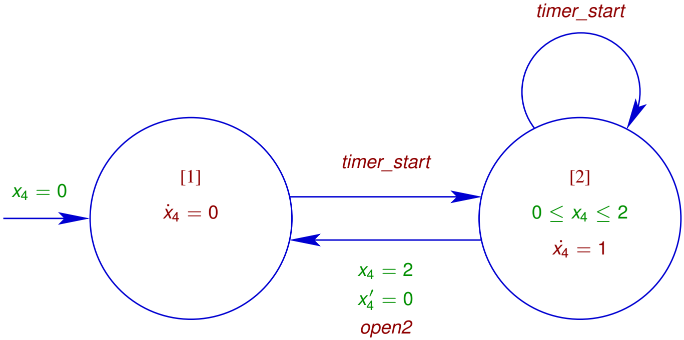
## Process 4

### Modes of operation:

- [1]: idle
- [2]: timer is active

### Communication:

- Reacts to *timer_start*
- Triggers *open2*.


timer_start
$x_4 = 0$
[1]
$\dot{x}_4 = 0$
timer_start
[2]
$0 \leq x_4 \leq 2$
$\dot{x}_4 = 1$
$x_4 = 2$
$x_4' = 0$
open2

State-space exploration:

$([1],[4],[4],[1]) : x_1 \leq 0,\ x_2 = 0,\ x_3 = 0, x_4 = 0.$

$$\Longrightarrow \quad ([1],[4],[4],[1]) \quad : \quad \begin{array}{l} x_1 \leq 0, x_2 = 0 \\ x_3 = 0,\ x_4 = 0. \end{array}$$

$$\xrightarrow[\textit{timer\_start}]{\textit{open1}} \quad ([2],[1],[4],[2]) \quad : \quad \begin{array}{l} x_1 = 0, x_2 = 0, \\ x_3 = 0,\ x_4 = 0. \end{array}$$

$$\overset{\leq \frac{1.5}{1.75}}{\Longrightarrow} \quad ([2],[1],[4],[2]) \quad : \quad \begin{array}{l} x_1 = 1.75\, t, x_2 = 67\, t \\ x_3 = 0,\ x_4 = t \\ 0 \leq t \leq \frac{1.5}{1.75} \end{array}$$

$$\longrightarrow \quad \ldots$$

# Problem 2

When the sluice gates of a dam are closed, the water level in the reservoir rises at the rate of 0.4 *m/h* if it is between 10 *m* and 20 *m*, and of 0.2 *m/h* between 20 *m* and 30 *m*. Above 30 *m*, a spillway drains the extra water. When the sluice gates are open, the water level decreases at the rate of 0.5 *m/h*, independently from the water level.

A sensor constantly measures the water level in the reservoir, and sends a signal each time that this level changes by 1 *m*. The sluice gates must be opened when the level exceeds 25 *m*, and closed when it drops below 15 *m*. Each opening or closing operation of the gates needs 120 seconds to complete.

1. Construct an hybrid system modelling this problem. Initially, the water level is at 17 *m*, and the sluice gates are open.
2. Give the first 3 steps of the state-space exploration of this system.

## Processes and variables:

- $P_1$: level of the water, $x_1 =$ current level.
- $P_2$: sensor, $x_2 =$ value of the level at the last signal.
- $P_3$: gates, $x_3 =$ percentage of closure (0–100).
- $P_4$: controller.

## Notes:

- We assume that the water level increases only when the gates are **fully** closed.
- If the water level goes outside the interval $[10, 30]$, the model may deadlock.

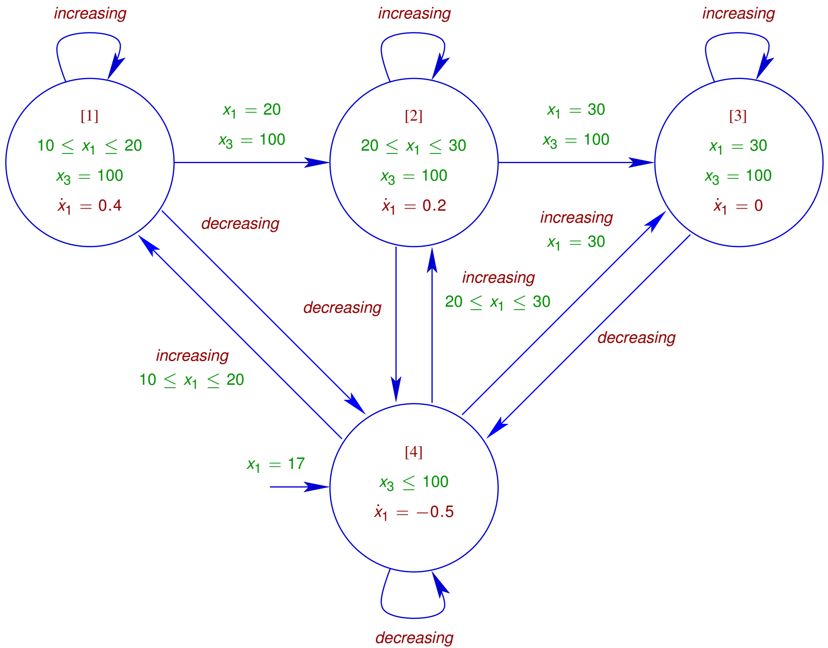
## Process 1

### Modes of operation:

- [1]: $10 \leq x_1 \leq 20$, gates fully closed, level increasing
- [2]: $20 \leq x_1 \leq 30$, gates fully closed, level increasing
- [3]: $x_1 = 30$, gates fully closed, level stable
- [4]: gates (at least partially) open, level decreasing

### Communication:

- Synchronize with Process 3 on labels *decreasing* and *increasing*.

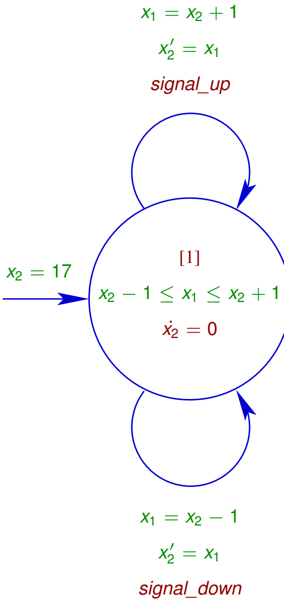

increasing
increasing
increasing
[1]
10 ≤ x1 ≤ 20
x3 = 100
ẋ1 = 0.4
x1 = 20
x3 = 100
[2]
20 ≤ x1 ≤ 30
x3 = 100
ẋ1 = 0.2
x1 = 30
x3 = 100
[3]
x1 = 30
x3 = 100
ẋ1 = 0
decreasing
increasing
x1 = 30
increasing
20 ≤ x1 ≤ 30
decreasing
decreasing
increasing
10 ≤ x1 ≤ 20
x1 = 17
[4]
x3 ≤ 100
ẋ1 = −0.5
decreasing

## Process 2

Modes of operation: A single mode from which we check the relative values of $x_1$ and $x_2$, and update $x_2$ whenever necessary.

Communication:

- Reads the current value of $x_1$.
- Synchronizes with $P_4$ on the labels *signal_up* and *signal_down*.



$x_1 = x_2 + 1$
$x_2' = x_1$
signal_up
$x_2 = 17$
[1]
$x_2 - 1 \leq x_1 \leq x_2 + 1$
$\dot{x_2} = 0$
$x_1 = x_2 - 1$
$x_2' = x_1$
signal_down

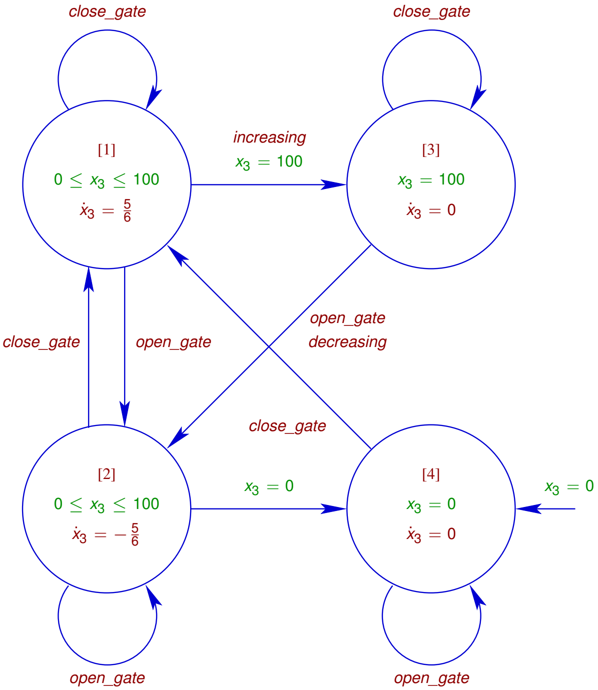
## Process 3

### Modes of operation:

- [1]: closing gates
- [2]: opening gates
- [3]: gates closed
- [4]: gates open

### Communication:

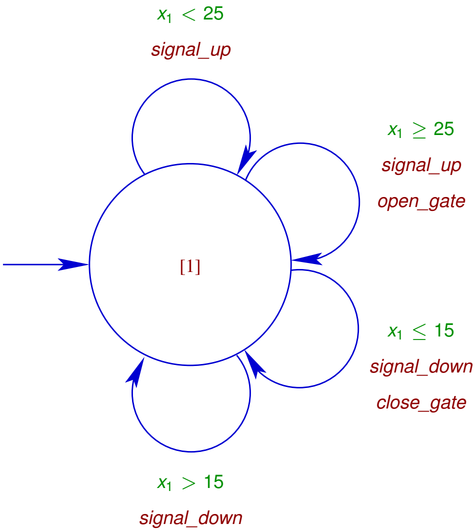
- Reacts to the labels *open_gate* and *close_gate*.
- Synchronizes with $P_1$ on *increasing* and *decreasing*.


close_gate
close_gate
[1]
$0 \leq x_3 \leq 100$
$\dot{x}_3 = \frac{5}{6}$
increasing
$x_3 = 100$
[3]
$x_3 = 100$
$\dot{x}_3 = 0$
open_gate
decreasing
close_gate
open_gate
close_gate
[2]
$0 \leq x_3 \leq 100$
$\dot{x}_3 = -\frac{5}{6}$
$x_3 = 0$
[4]
$x_3 = 0$
$\dot{x}_3 = 0$
$x_3 = 0$
open_gate
open_gate

## Process 4

Modes of operation: A single mode from which we check the value of $x_1$ when *signal_up* and *signal_down* are received, and emit *open_gate* and *close_gate* orders.

Note: We check $x_1$ and not $x_2$, since the value of $x_2$ is modified by the transitions of Process 2 that synchronize on *signal_up* and *signal_down*.



$x_1 < 25$
signal_up
$x_1 \geq 25$
signal_up
open_gate
[1]
$x_1 \leq 15$
signal_down
close_gate
$x_1 > 15$
signal_down

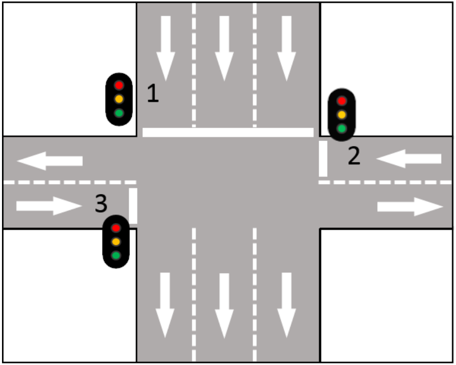
State-space exploration:

$([4],[1],[4],[1]) : x_1 = 17,\ x_2 = 17,\ x_3 = 0.$

$$
\begin{array}{lll}
\stackrel{\leq 2}{\Longrightarrow} & ([4],[1],[4],[1]) & : \quad 16 \leq x_1 \leq 17, x_2 = 17, \\
 & & \quad\;\; x_3 = 0. \\
\stackrel{signal\_down}{\longrightarrow} & ([4],[1],[4],[1]) & : \quad x_1 = 16, x_2 = 16, \\
 & & \quad\;\; x_3 = 0. \\
\stackrel{\leq 2}{\Longrightarrow} & ([4],[1],[4],[1]) & : \quad 15 \leq x_1 \leq 16, x_2 = 16, \\
 & & \quad\;\; x_3 = 0. \\
\longrightarrow & \ldots &
\end{array}
$$

# Problem 3

A smart traffic light system is installed at the intersection of two roads. There are three stop lights $\{1, 2, 3\}$ that can either be red, green, or orange. The state of 2 and 3 is identical at all times.



Traffic lights 2 and 3 are red and traffic light 1 is green as long as there are less than six cars waiting in front of 2 or 3. When this threshold is reached, stop light 1 becomes orange for 5 seconds before switching to red. At this time, stop lights 2 and 3 become green for 15 seconds. After that delay, they change to orange for 5 seconds and then switch to red as traffic light 1 becomes green again.

The incoming flows of cars at the three traffic lights are respectively $f_1 = 30$, $f_2 = 6$ and $f_3 = 3$ cars/minute. We also define the saturation flow of a traffic light as the rate of cars that are able to cross this light when it is green. The saturation flows of the three lights are respectively $s_1 = 1.5$, $s_2 = 0.5$ and $s_3 = 0.5$ cars/second.

1. Construct a hybrid system that models this traffic management system.
2. Give the first 3 steps of the space-state exploration of this system, when initially no cars are queueing in front of the traffic lights, and stop lights 2 and 3 are red while 1 is green.