Data Science and Automation

Lesson 17

PLC - Introduction

Who am I?

Andrea Bombarda

I'm a first year PhD student at the University of Bergamo (Formal Methods & Software Engineering Lab)

Tel. 035 2052022

andrea.bombarda@unibg.it

https://cs.unibg.it/bombarda/

Lessons and Lab

- 16 hours of «standard» lectures

- 12 hours of «lab»
    - I will introduce the main aspects of the SW that we can use to simulate a PLC on a regular PC
    - I will give you some exercices that you can solve using the PLC simulator, and we will analyze togheter the solutions

Topics

- Introduction to the Industrial Automation
- PLC
    - Introduction
    - Languages ot the IEC 61131 norm
        - Ladder
        - SFC
        - Structured Text

Introduction to the Industrial Automation

What does «automation» stands for...

(From Garzanti): «The introduction of mechanical production processes, mainly driven by electronic systems, in which the manual intervention of man is reduced to a minimum»

(From Treccani): «The use of a set of technical means and procedures which, acting suitably on particular devices or devices, ensure the automatic performance of a specific process, the automatic operation of an industrial plant, a public service, etc .;»

Why?

We introduce the industrial automation because it allows us to:

- Reduce production times and expenses
- Increase production quantities
- Increase the quality of a product
- Increase the flexibility of production plants
- Produce using the “Just in Time” technique (and reduce the size of the warehouse)

Some videos

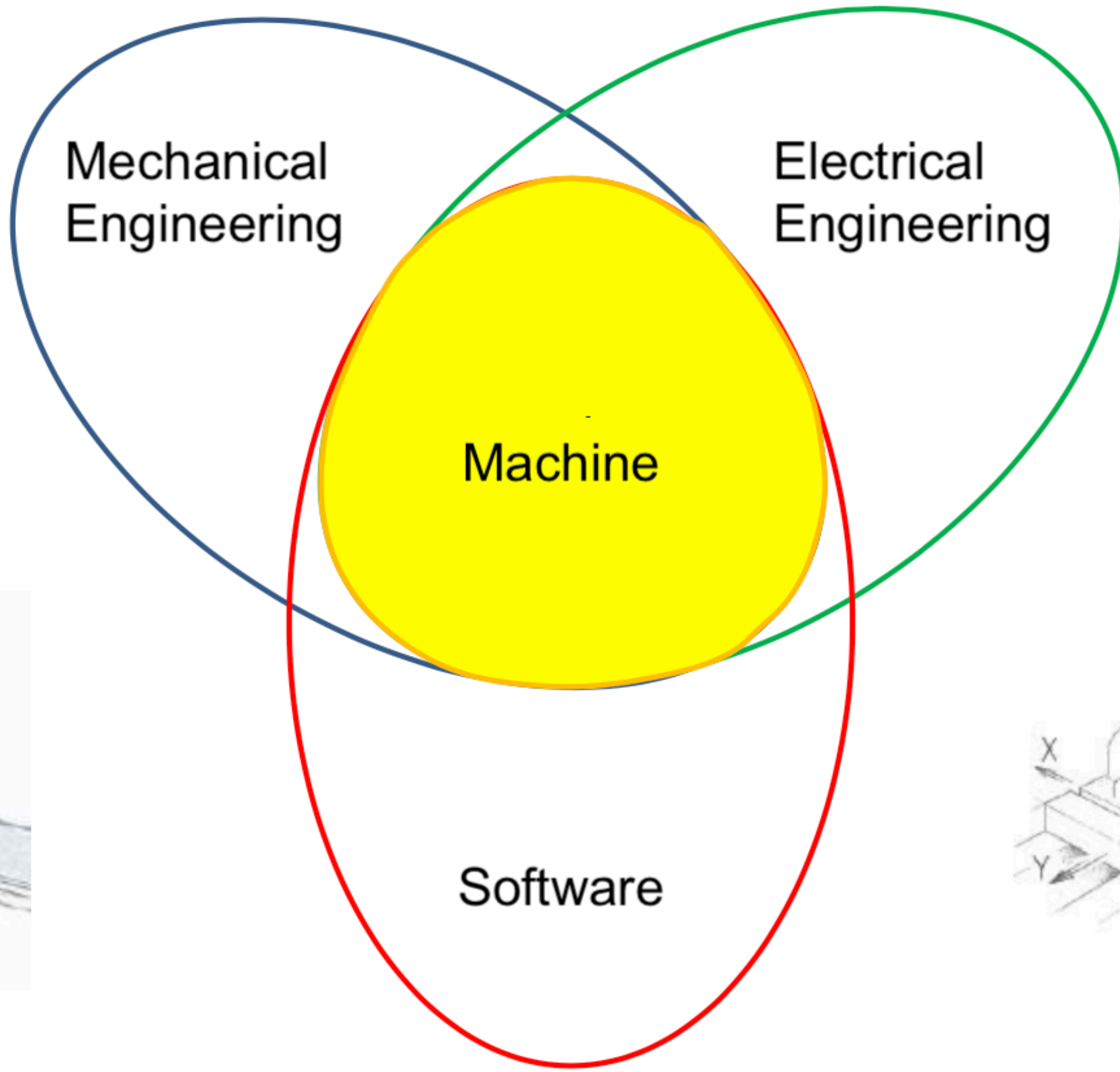
Ford production line:

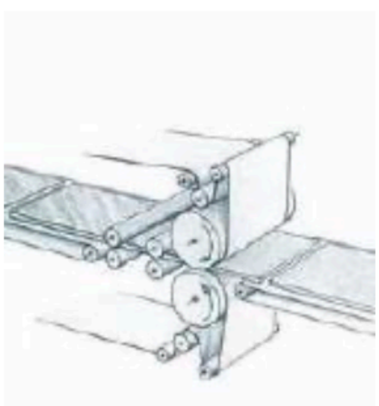
https://www.youtube.com/watch?v=mBUFD3C_YXI&t=605s

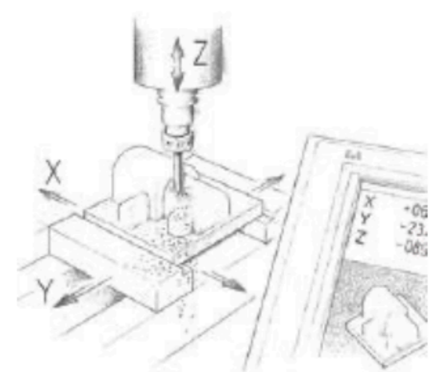
Industry 4.0:

https://www.youtube.com/watch?v=9jWH1YahNIY

Which topics are involved?



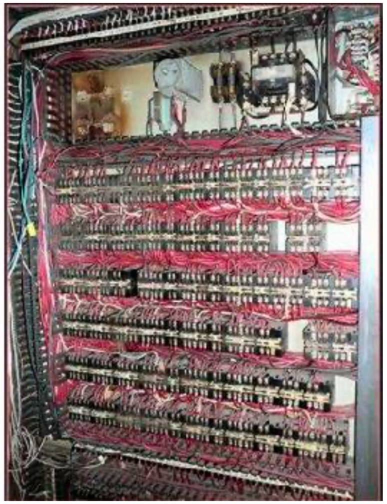




A little bit of history

First generation of controllers (1950):

- Controllers build with relè, coils, timers
- Slow computation
- Not adaptable (no flexibility)



A little bit of history

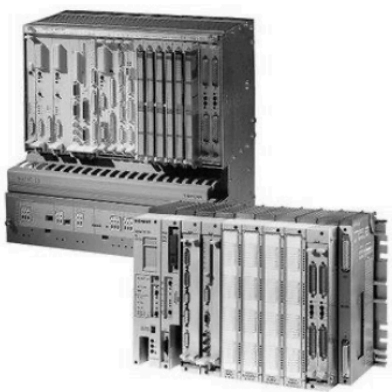
<u>Second generation of controllers (1960):</u>

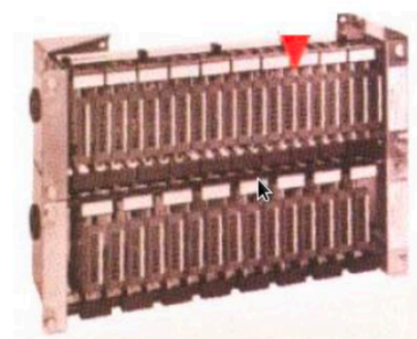
- Use of semiconductors
- Better performances
- Increase of the price
- Limited flexibility: the controllers were not programmable

A little bit of history

<u>Third generation of controllers (1960):</u>

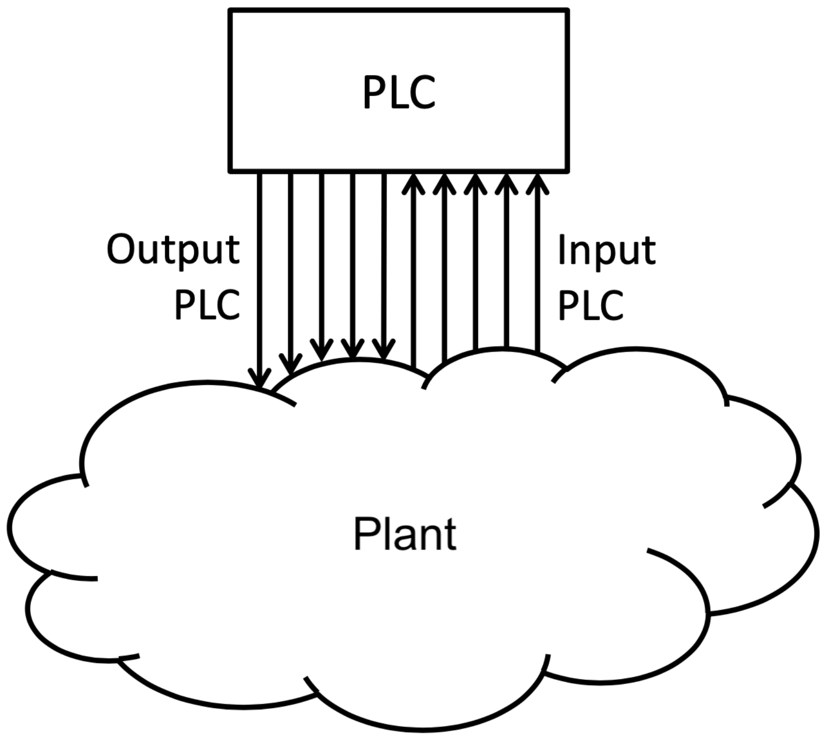
- Use of microprocessors
- The controllers are programmable
- The first PLC (Alleb Bradley 1968) was introduced





A little bit of history

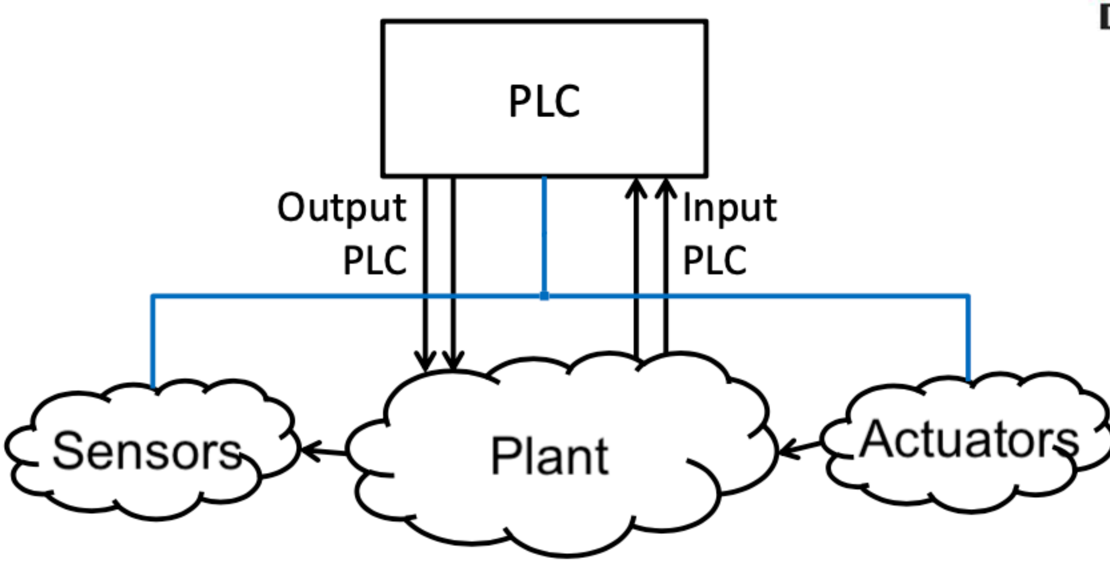
Centralized control (1960-1980):



A little bit of history

First scalable solutions (1980-1990):

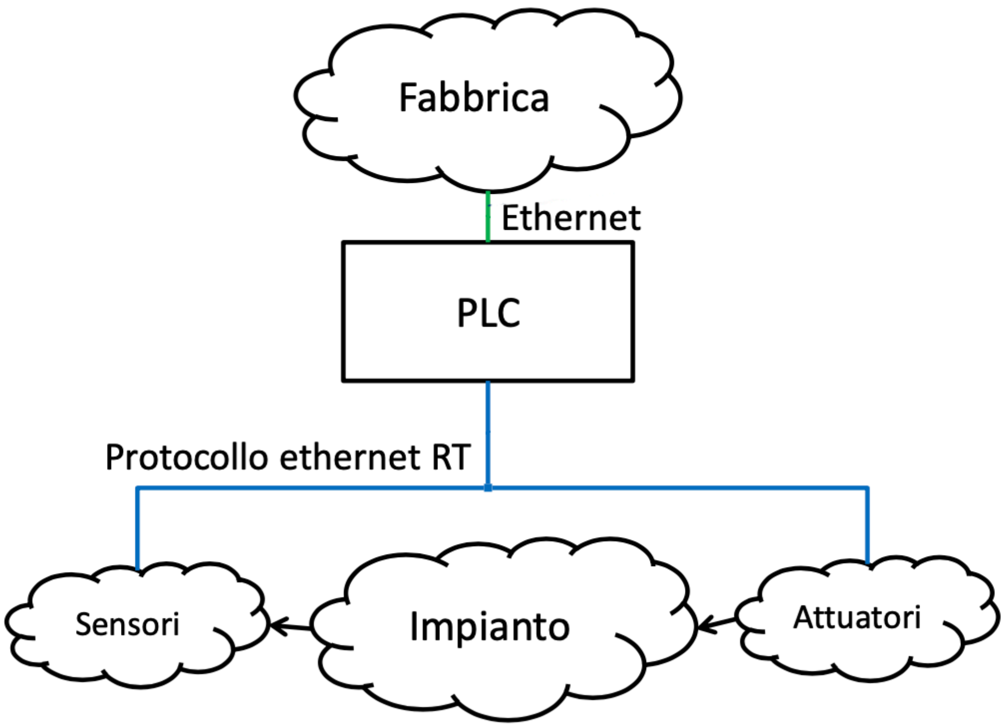
- First uses of Real-Time legacy network (Fieldbus)



A little bit of history

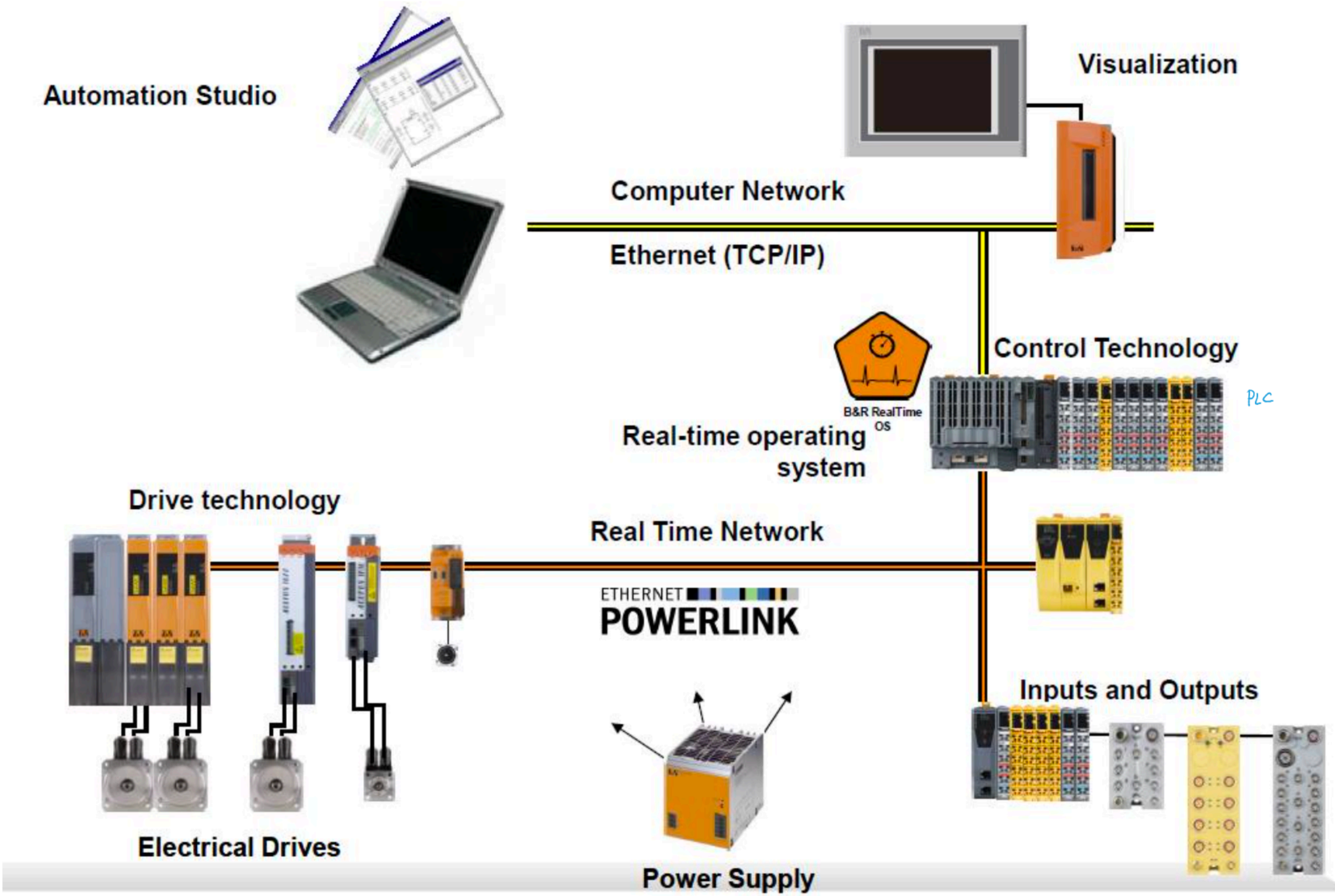
<u>Integration between plant and different levels (2000 ->):</u>

- Introduction of Real-Time networks based on Ethernet

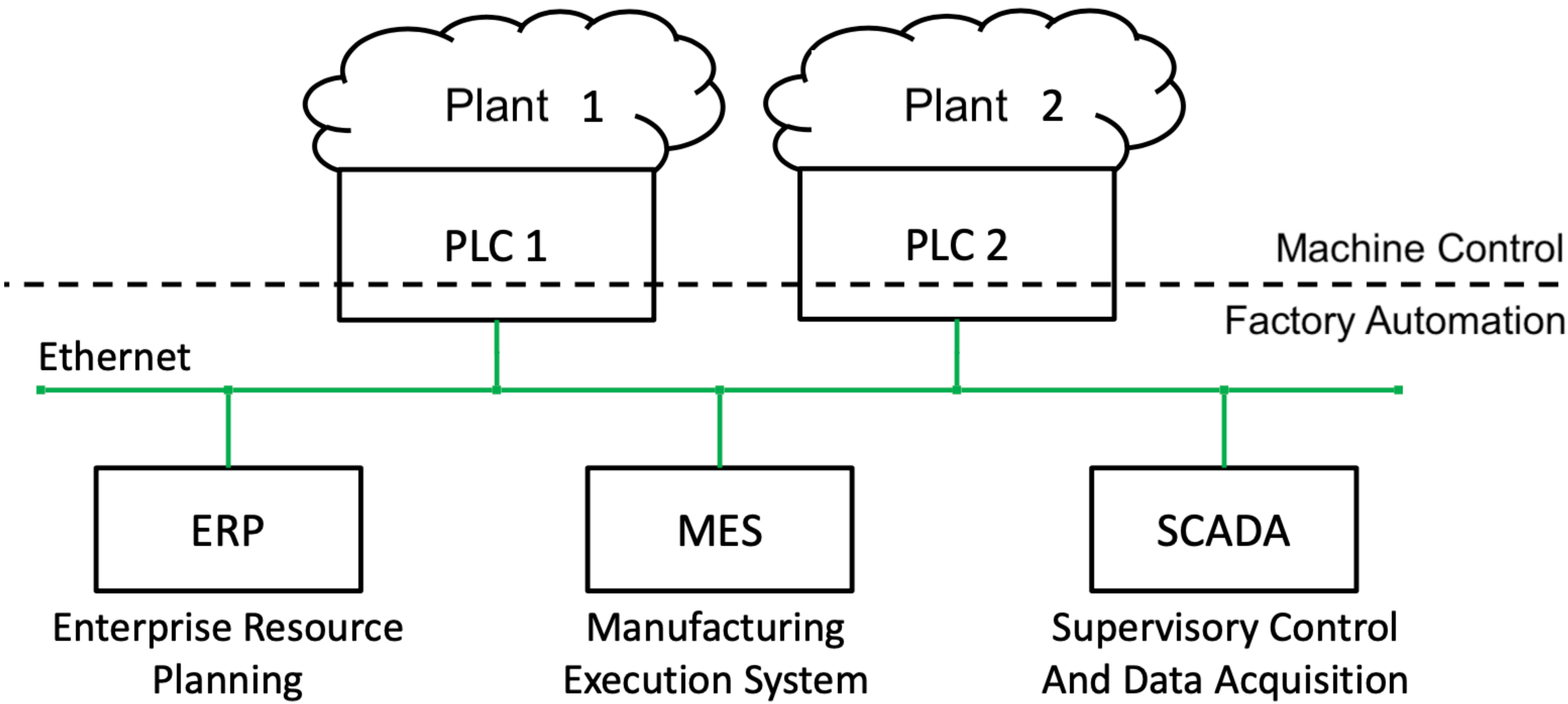


EtherNet/IP
PROFI NET
ETHERNET POWERLINK
EtherCAT
SERCOS interface
...

An example of the present Architecture



How does the Automation interacts with the management?



How does the Automation interacts with the management?

Enterprise Resource Planning

It is the company information system, including:

- Accounting
- Management control
- Management of:
    - Staff
    - Purchase
    - Warehouse
    - Production
    - Sales

How does the Automation interacts with the management?

Manufacturing Execution System

It is a software to schedule the production, the orders and monitor:

- Production
- Order status
- Production timing

How does the Automation interacts with the management?

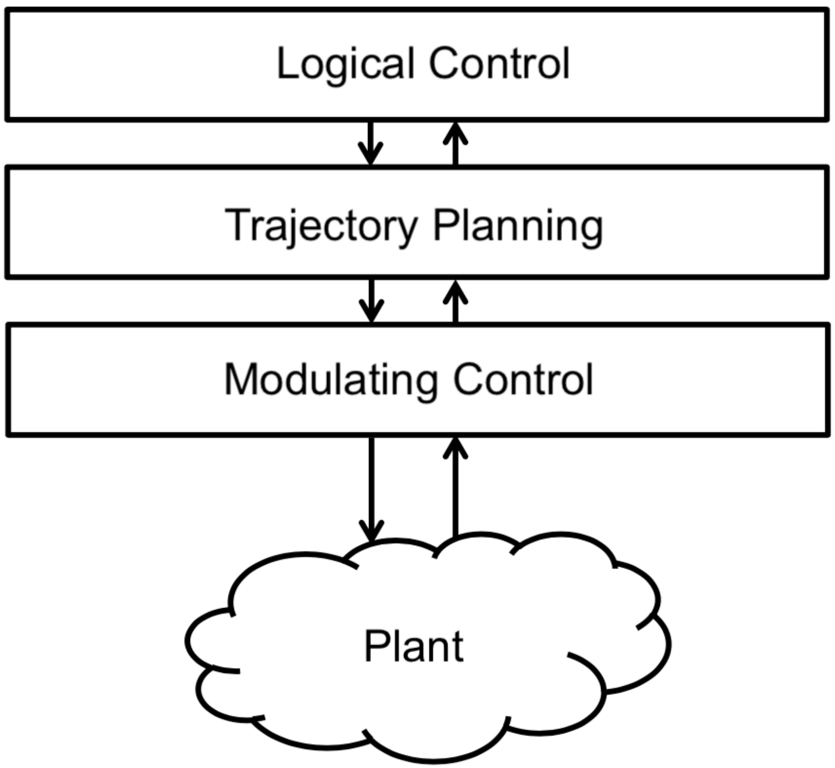
<u>SCADA</u>

It is the system used to control and monitor the production line (including the Human Machine Interface). It’s composed of:

- Production line control
- Data acquisition
- Data analysis
- Alarms management
- Data logging

Interaction between automation and plant

Conceptual scheme



PLC – Programmable Logic Controller

CNC – Computer Numerical Control

Inverter / Actuators / …

Interaction between automation and plant

Remarks:

- A single PLC can be used for a single machine, or more PLCs can be used for a single machine

- CNC operations can also be embedded into the PLC or in multi-axis inverters

- CNC me not be used in many cases

- For small plants, the PLC may do also the modulating control

Types of control

Modulating Control:

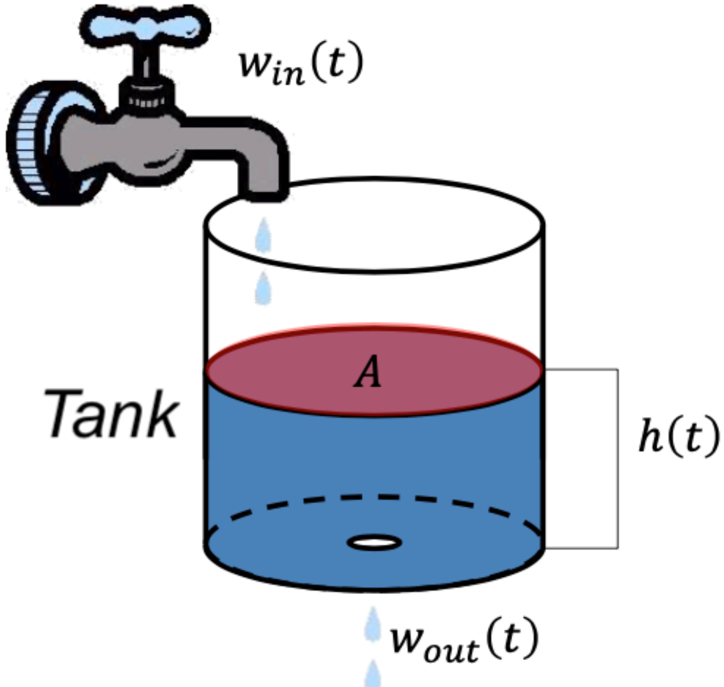
- Continuous
- Continuous control actions
- Modeled by differential equations

Logical Control:

- Event-based
- Discrete control actions
- Modeled by Petri Net or Finite State Automata

We will focus on this kind of control

Example of a (Continuous) System



Inputs

$w_{in}(t)$

$w_{out}(t)$ *(not adjustable)*

Output

$h(t)$

Model

$$A \cdot \frac{dh(t)}{dt} = w_{in}(t) - w_{out}(t)$$

Example of a (Continuous) System

<u>Logical control:</u>

Let's consider $w_{in}(t)$ as ON or OFF.

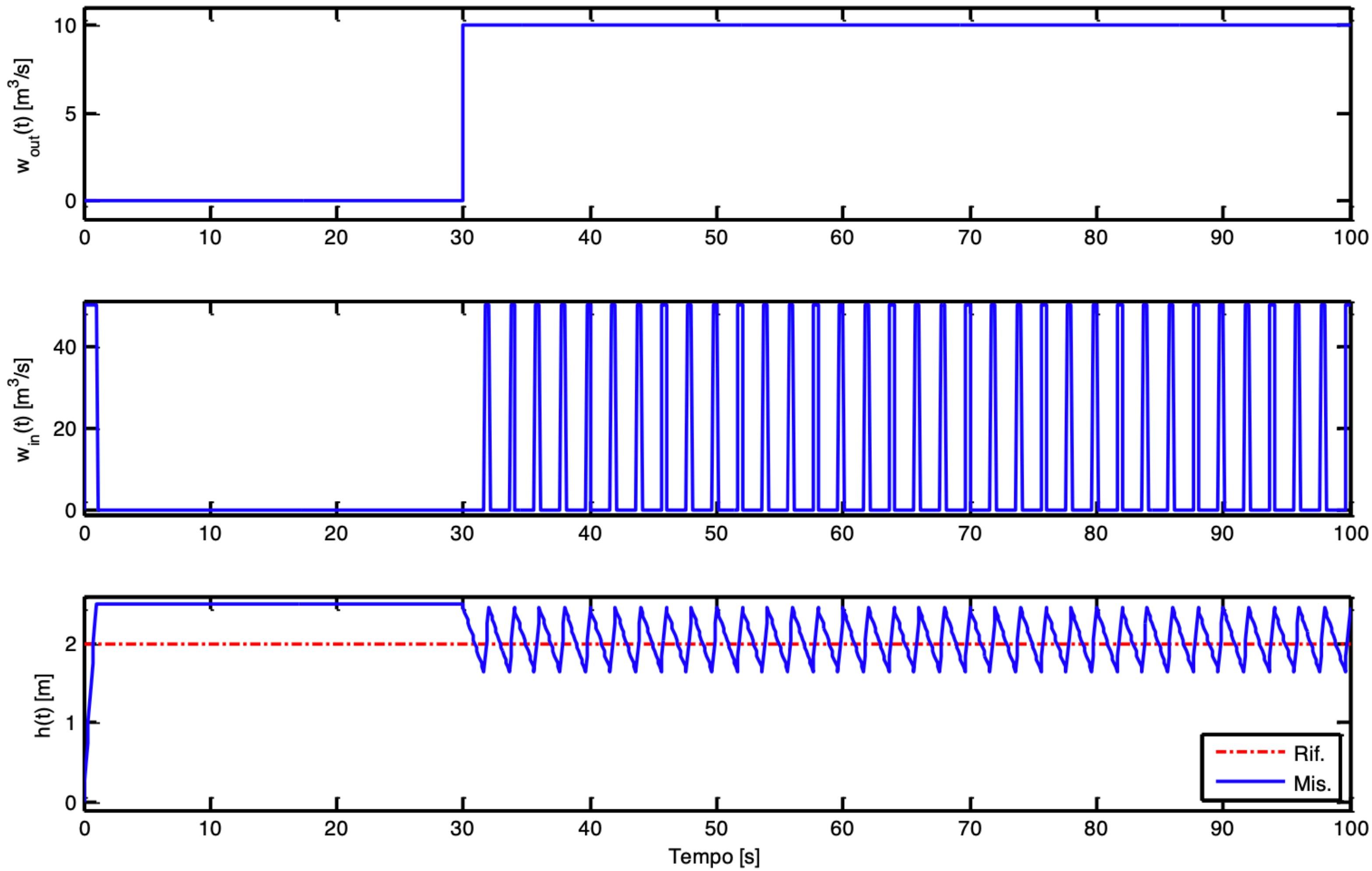
We can suppose that we want to keep h between h_{min} *and* h_{max}.

A simple control code can be:

```
if (h(t)>=h_max) {
        w_in(t) = OFF;
}
if (h(t)<=h_min) {
        w_in(t) = ON;
}
```

Example of a (Continuous) System

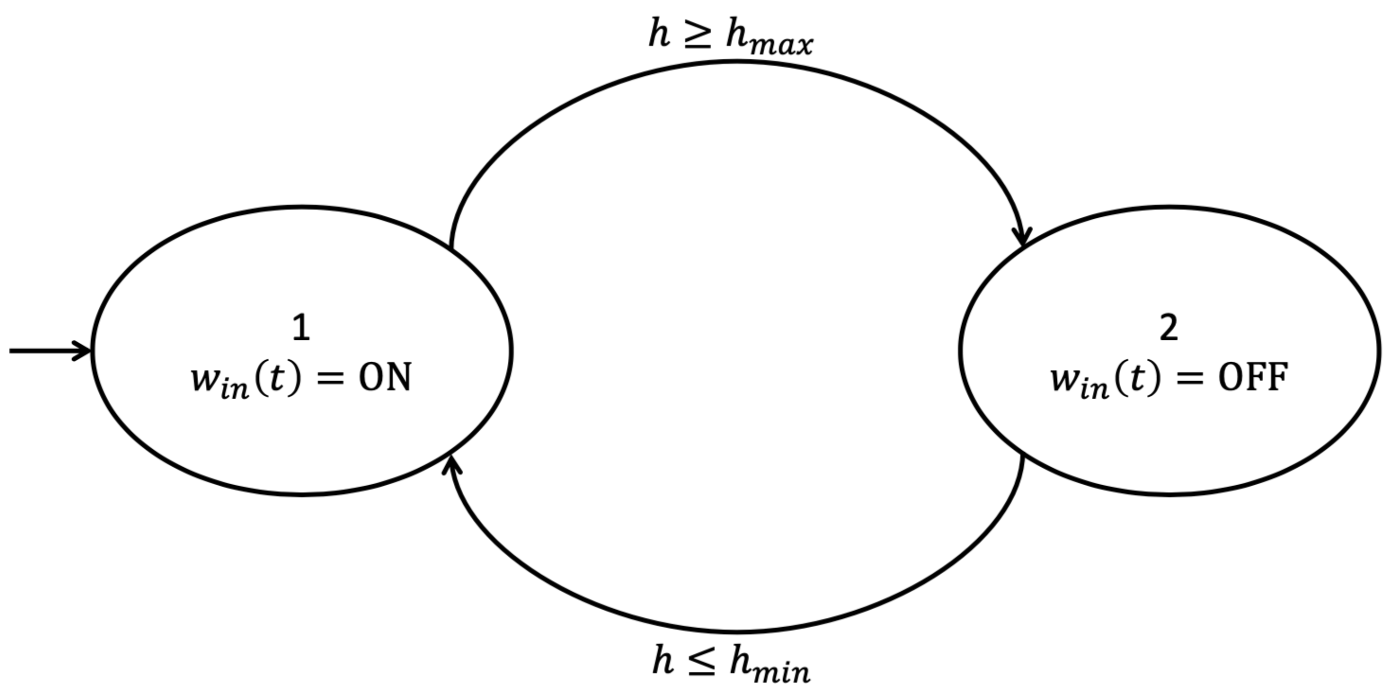
Results:



Example of a (Continuous) System

<u>Logical control:</u>

The same kind of control can be represented by a finite state automaton as follows:



Finite State Automata

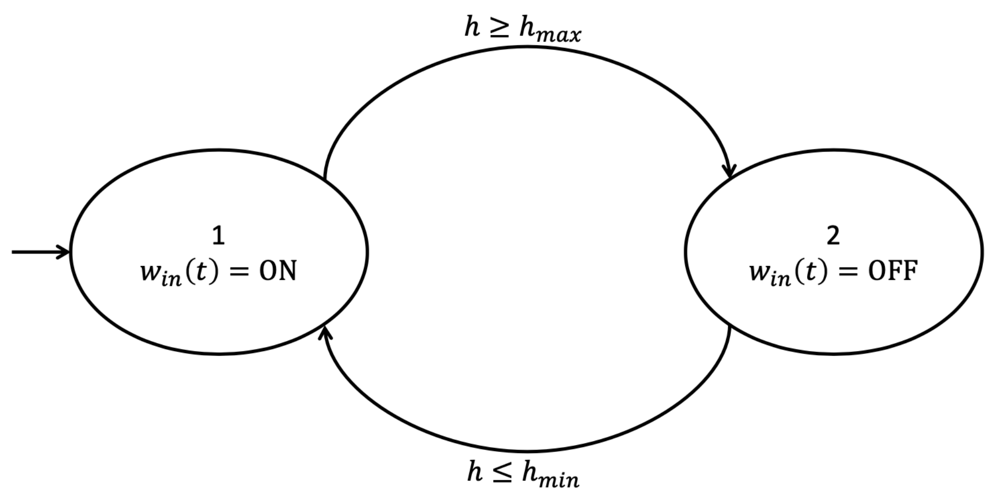
A finite state automata (with inputs and outputs) is a t-uple (U, X, Y, f, h, x_0) where:

- $U = \{u_1, u_2, u_3, \dots\}$ is the set of the *input events*
- $X = \{x_1, x_2, x_3, \dots\}$ is the *finite* set of the *states*
- Y is the *finite* set of the *outputs*
- $f: X \times U \rightarrow X$ is the *transition function*
- $h: X \times U \rightarrow Y$ is the *output update function*
- x_0 is the *initial state*

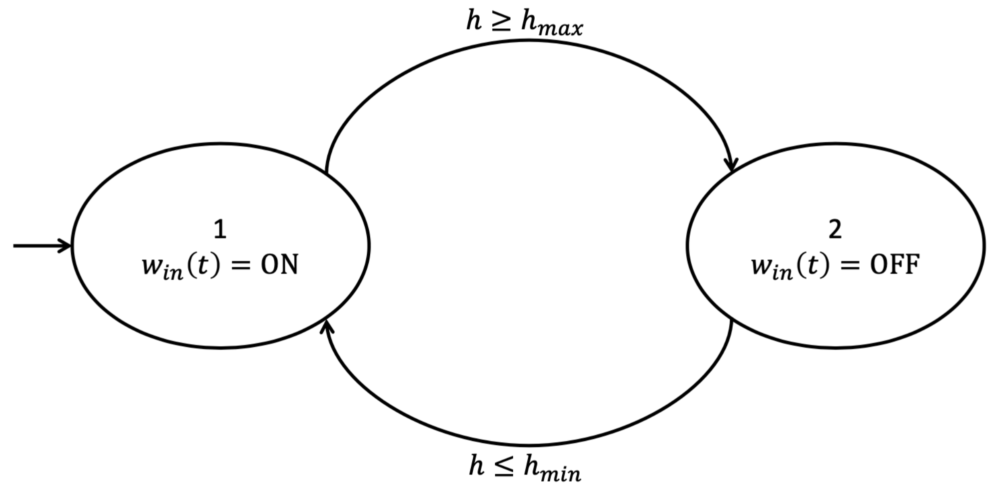
Finite State Automata

Considering the previous example, we have:

- $U = \{h \geq h_{max}, h \leq h_{min}\}$
- $X = \{1, 2\}$
- $Y = \{ON, OFF\}$
- $x_0 = 1$



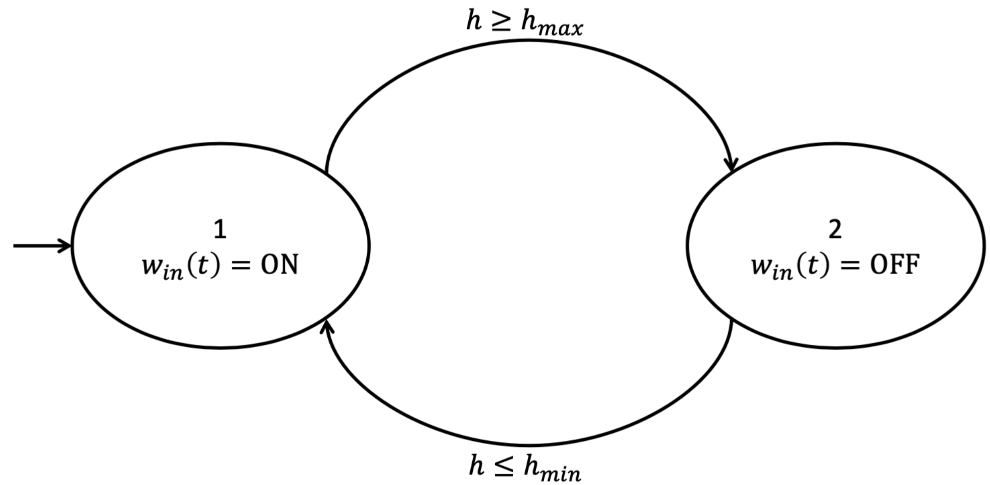
Finite State Automata



$$f(\cdot,\cdot): X \times U \to X$$

	$h \geq h_{max}$	$h \leq h_{min}$
1	2	-
2	-	1

Finite State Automata



$$h(\cdot,\cdot): X \times U \to Y$$

	$h \geq h_{max}$	$h \leq h_{min}$
1	OFF	ON
2	OFF	ON

Finite State Automata

Remarks:

- The output can be update also during the transition.

- Some inputs may not update the status.

Finite State Automata – A Simple Example

Our system executes the following activities:

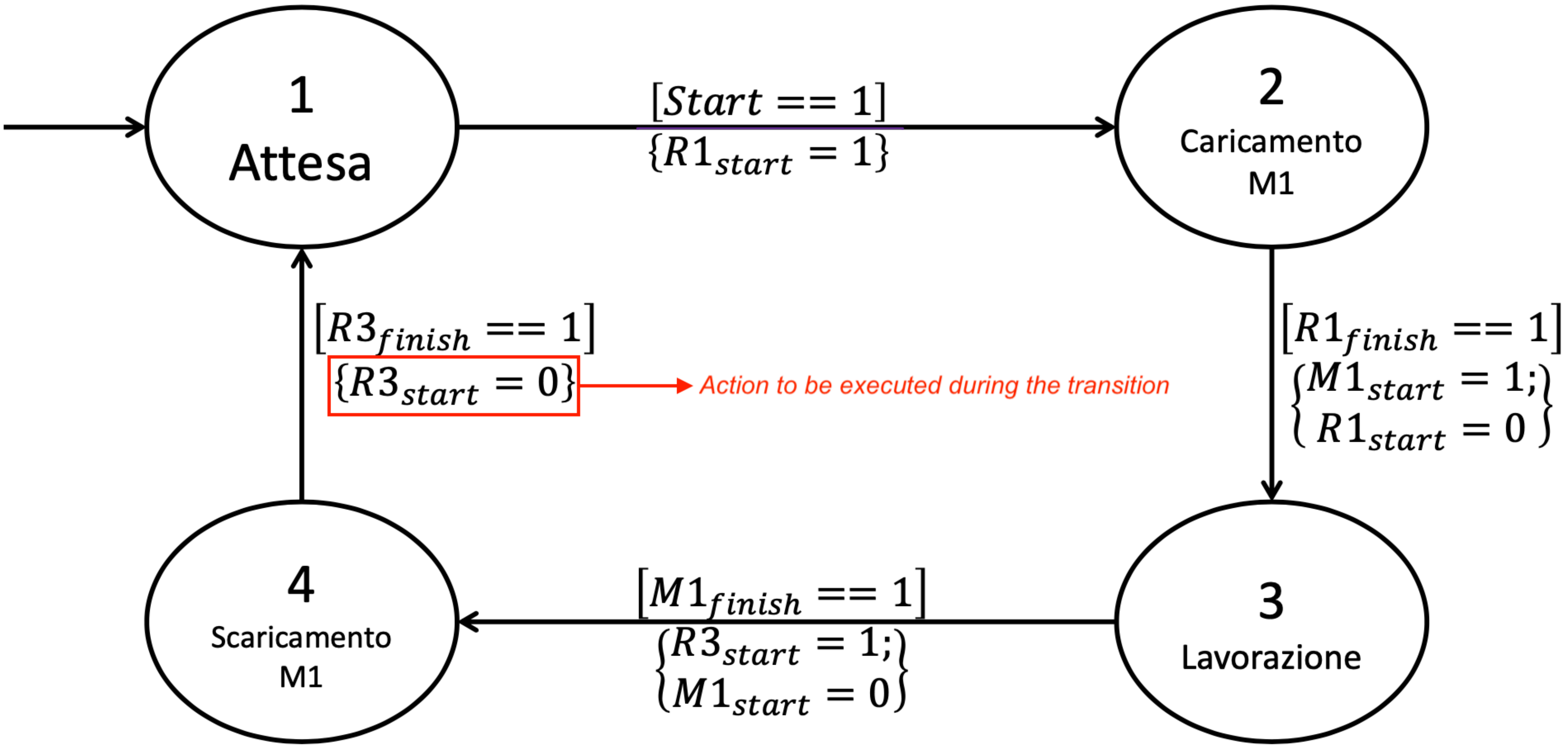
- Load a piece on M1 by using R1
- M1 process the piece
- Unload a piece from M1 by using R3

Finite State Automata – A Simple Example

The production line starts when $Start = 1$ and terminates when $R3_{finish} = 1$.

- $U = \left\{Start, R1_{finish}, M1_{finish}, R3_{finish}\right\}$
- $Y = \{R1_{start}, M1_{start}, R3_{start}\}$

Finite State Automata – A Simple Example



Finite State Automata – A Simple Example

Alternative representation

	$Start == 1$	$R1_{finish} == 1$	$M1_{finish} == 1$	$R3_{finish} == 1$	Y
1	2	-	-	-	$R1_{start} = 1$
2	-	3	-	-	$M1_{start} = 1;$ $R1_{start} = 0$
3	-	-	4	-	$R3_{start} = 1;$ $M1_{start} = 0$
4	-	-	-	1	$R3_{start} = 0$

Finite State Automata – A Simple Example

Remarks:

- In this case we don't have the function $h: X \times U \rightarrow Y$ because we do not vary the output when we are in a state, but only during a transition.