Ingegneria e Tecnologie dei Sistemi di Controllo T
Control Systems Technologies

Architectures and Technologies
in Control Systems

Ing. Christian Conficoni (prof. Andrea Tilli)
DEI - Alma Mater Studiorum Università di Bologna
E-mail: christian.conficoni@unibo.it
https://www.unibo.it/sitoweb/christian.conficoni3

Objectives

- **Introduction to technological architectures in control systems**
  - **Model/Scheme to be used as “general guideline”**

- **Define basic components and introduce: main features and problems**

- **To define (and understand) a <u>general</u> *technological architecture*, we need <u>general</u> *functional architecture* before**
  - **First “functions”, then “implementations”**

- **Some consideration on mapping from “functions space” to “technological space”**

Contents

- **Definition of a general functional scheme for control systems**
  - **Quite general (even if qualitative)**
  - **It helps in defining technological scheme**

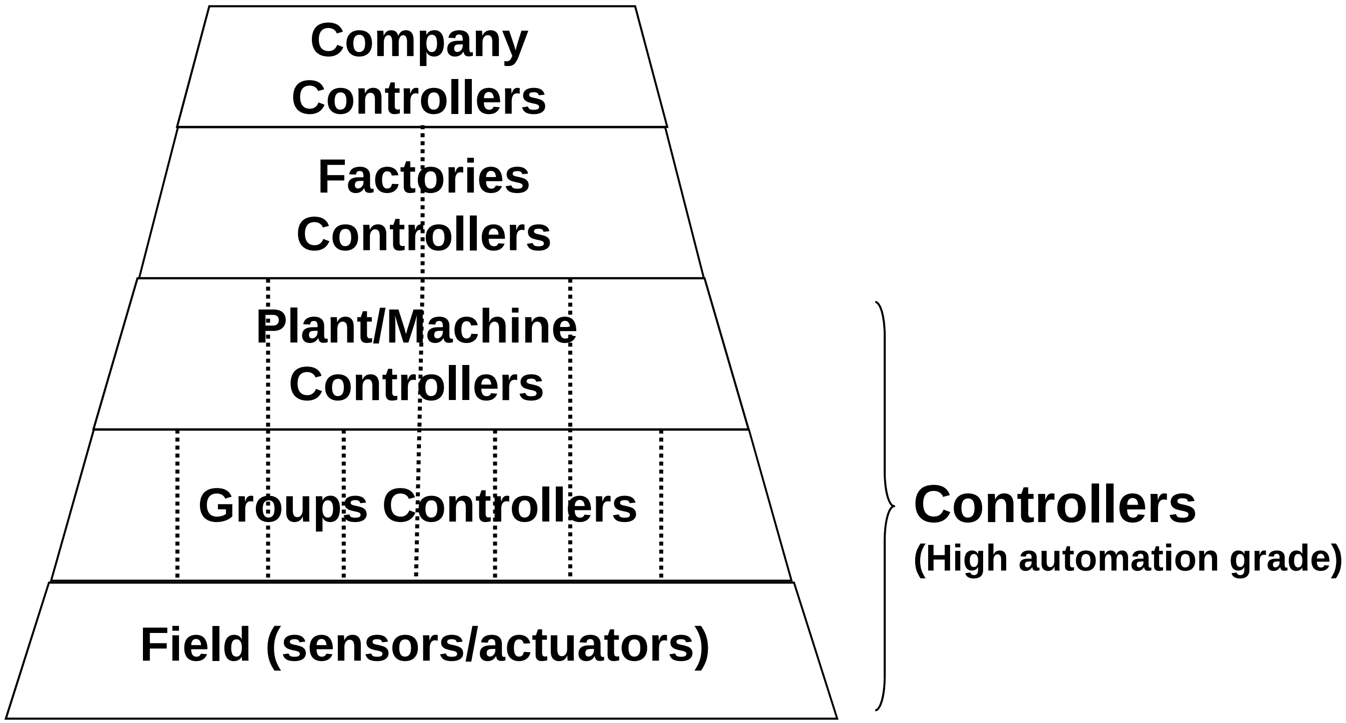
- **Definition of a general technological scheme**
  - **Quite general (even if qualitative)**

- **Basics on Components**
  - **Adopted technologies**
  - **Potential problems to remember**

- **Guidelines to map a functional scheme on a technological scheme**

GENERAL FUNCTIONAL SCHEME OF A CONTROL SYSTEM

General functional scheme



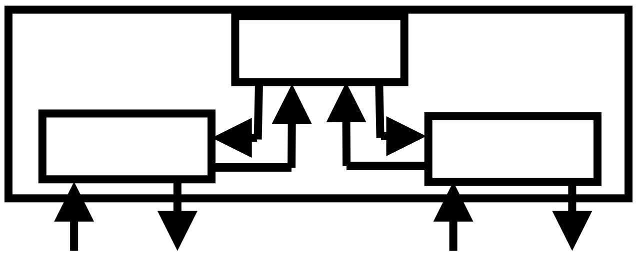
Focus on “controllers” level:

Time-dependent system control (“classic”) and logic control

General Functional Scheme

MIMO Schemes

- **In general functional control schemes are MIMO**
  - **“Classic” time-dependent-systems controllers can be MIMO**
    - **State-space approach**
    - **Cascaded SISO structures**
  - **Logic controllers are usually MIMO**
    - **Many logic I/O**
    - **Finite-states automata**
  - **Anyway, according to general organization depicted in Automation Pyramids overall controller is usually MIMO**
    - **“Controllers” level is a sum of subcontrollers (SISO or MIMO)**
    - **The results is MIMO**
      - **Sometimes hundreds of I/O**



Typical relationships between logic controllers and time-dependent-systems controllers

- **Usually, *Logic Controllers* trigger *Time-dependent-system Controllers***
  - **Give consensus to start working**
  - **Change references**

- **Usually, *logic controllers are hierarchically higher than time-driven-systems controllers***

Typical relationships between logic controllers and time-driven-systems controllers

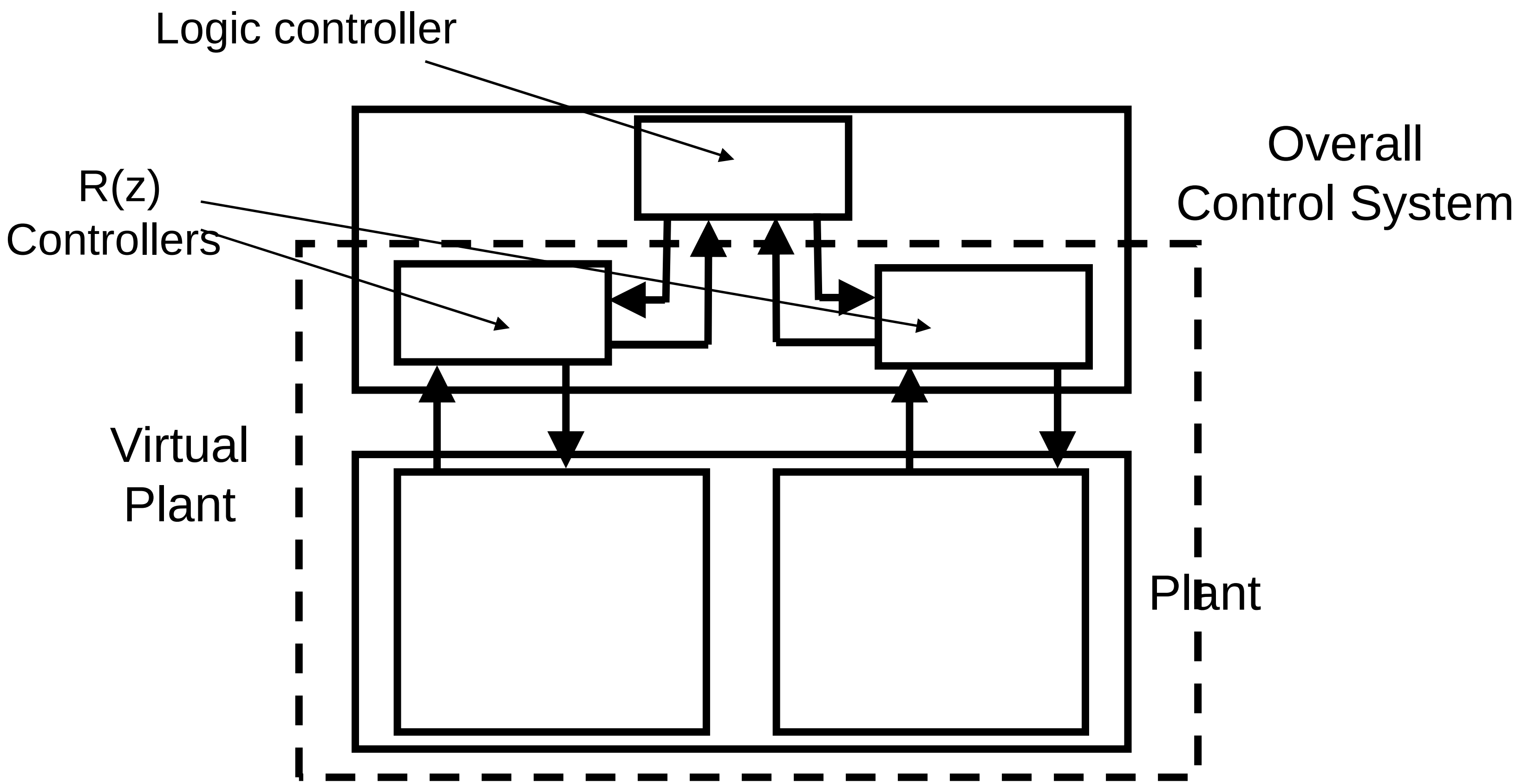
➢ **Summing up:**

- **IN AUTOMATION, IN GENERAL, “CONTROLLERS” HAVE TO MANAGE DIFFERENT WORKING PHASES, NOT ONLY REGULATE SOME TIME-DRIVEN-OUTPUT VARIABLES**

  - **The overall controller will be a sum of “state-machines” (possibly very complex) which will give commands to activate-deactivate or change something in time-driven-systems controllers.**

Typical relationships between logic controllers and time-driven-systems controllers

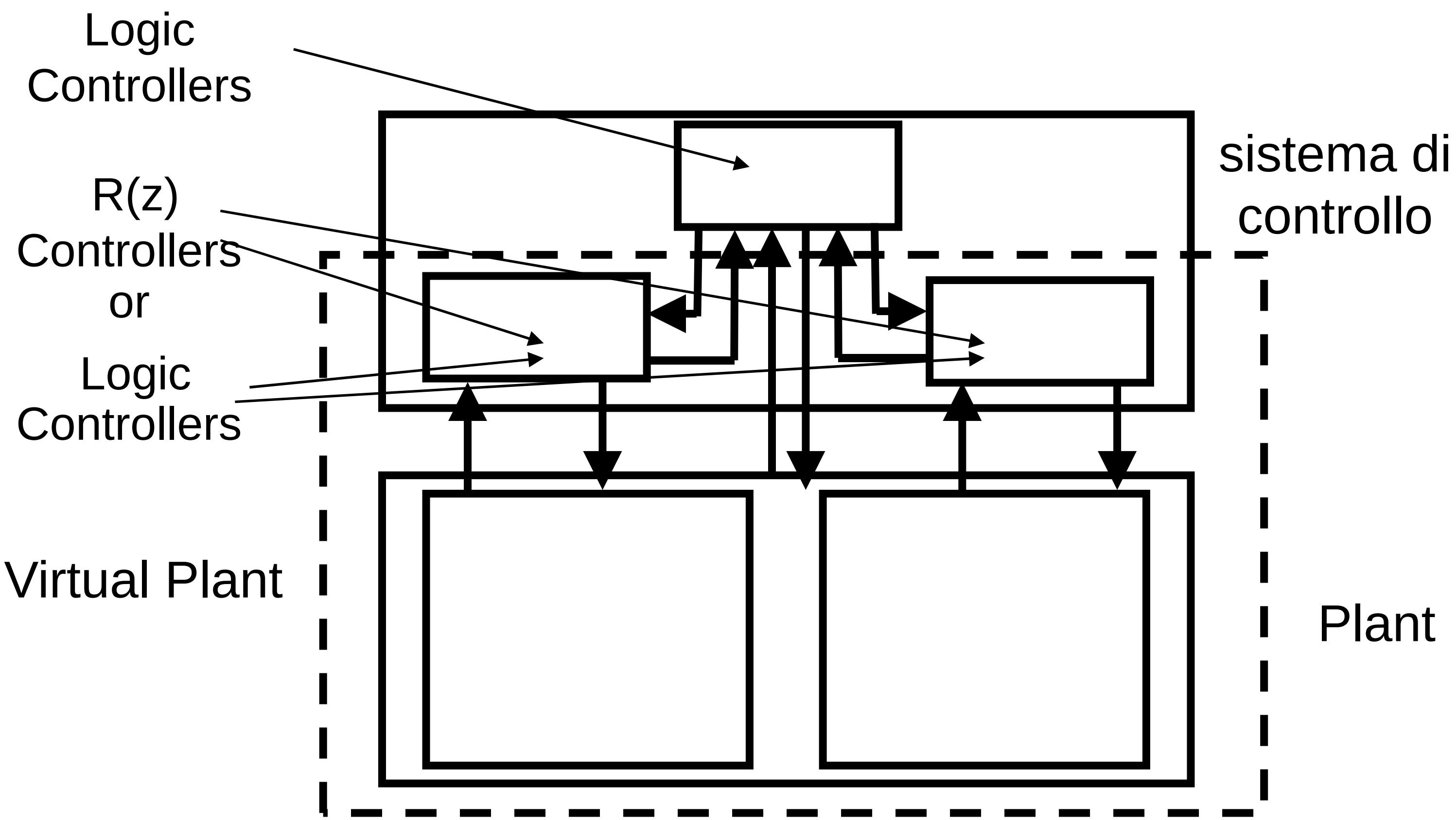
- **Indicating time-driven-system controllers as R(z):**



General Functional Scheme

Typical relationships between logic controllers and time-driven-systems controllers

- **Actually hierarchical relation among R(z) and logic controllers is not so "pure and clean"…**



General Functional Scheme

Typical relationships between logic controllers and time-driven-systems controllers

- **Logic Controllers and R(z) Controllers: “non-pure” hierarchy**
  - **Logic Controllers could act and sense directly to/from the plant**
  - **“Non-pure” hierarchies among controllers of the same type**
    - **R(z): Cascade controllers**
    - **Logic Controllers: see previous slide**
    - **<u>USUALLY NO Ctrl R(z) commands Logic Controllers.</u>**
  - **Sometimes, R(z) and logic controllers are considered at the same level and hierarchy is treated as “horizontal collaboration”**
    - **But it is better to consider “non-pure” hierarchy and use “horizontal collaboration” for controllers actually on the same level (really cooperating controllers, not so common in industrial automation, considered in other systems)**

General Functional Scheme

Additional consideration

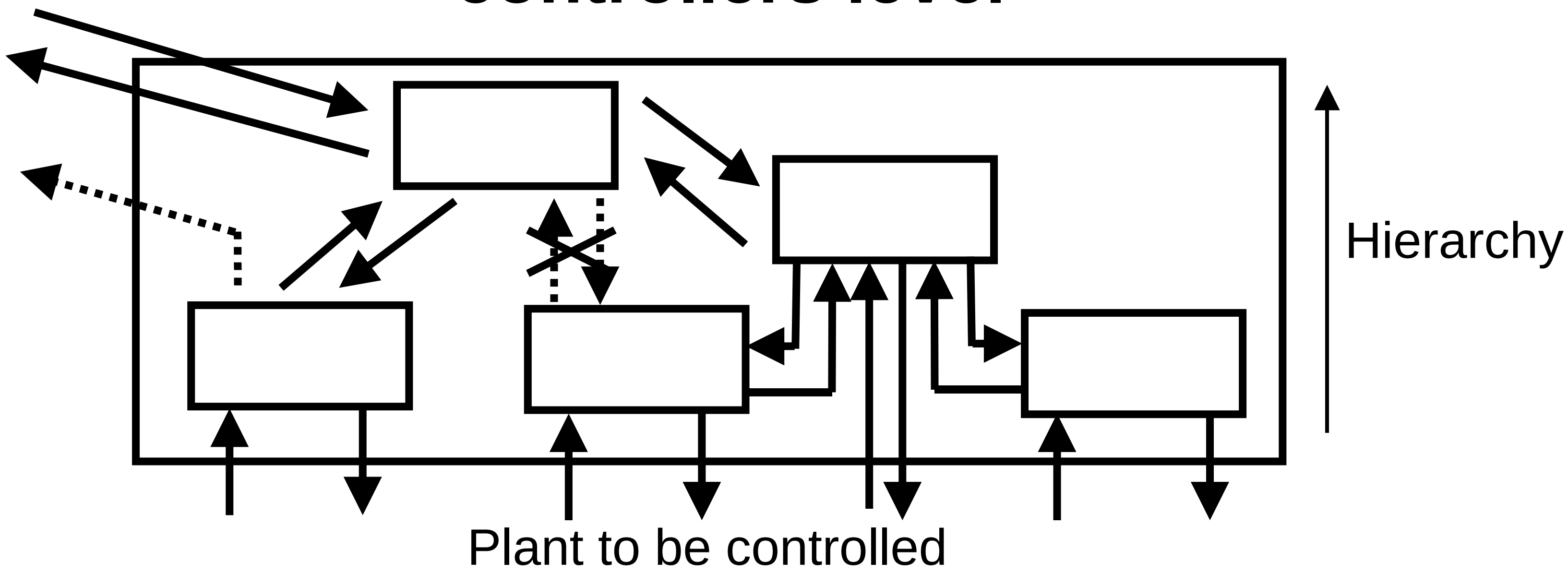
- **At higher levels of automation pyramid non-pure hierarchies are not so common**

- **Anyway at higher levels no direct interactions with plant**

  - **They interact with controllers level to give commands and get measurements**
  - **(With Industrial Ethernet we could …. Complex!)**

- **Remember: focus of the course on “controllers level”**

General Model/Architecture to represent the "controllers level"

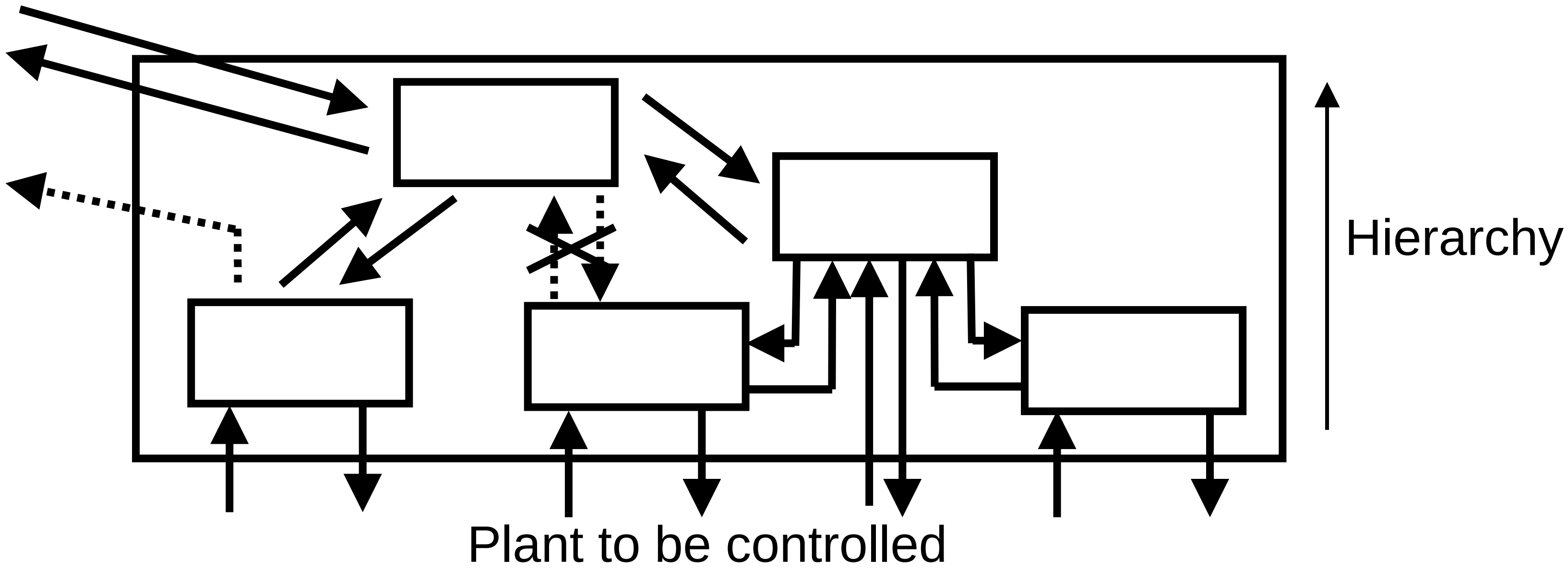


Blocks: representing logic controllers or time-dependent-system controllers

Hierarchical constraints:
- logic ctrl cannot be commanded by a time-driven-system ctrl
- a ctrl cannot be commanded by more than one ctrl

External communication:
- set point or other from higher AP level, humans better to higher controllers only
- with lower controllers just monitoring, possibly

General Model/Architecture to represent the "controllers level"



Every functional project of a complex control system
(logic ctrl + R(z)/R(s))

must (should) comply with the just-defined general functional architecture

GENERAL TECHNOLOGICAL SCHEME FOR A CONTROL SYSTEM

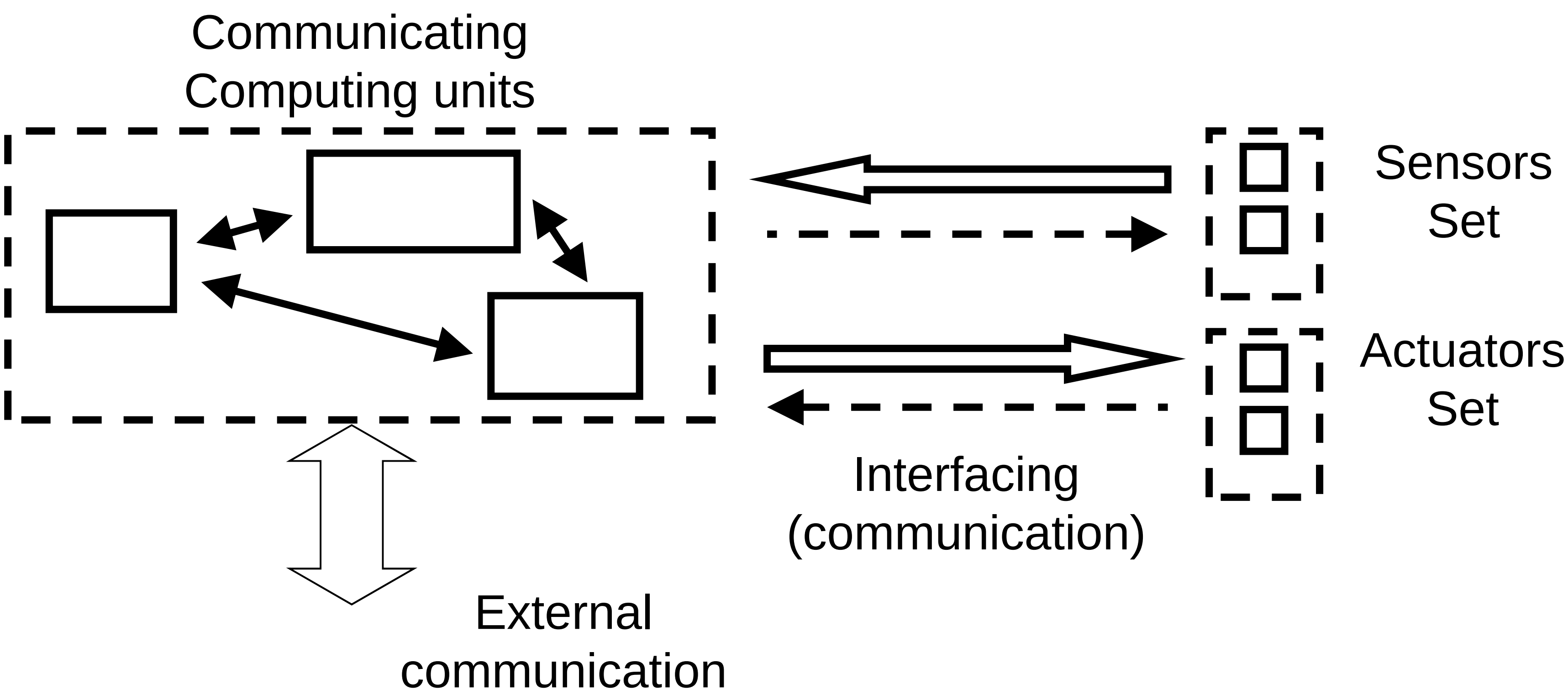
General Technological Scheme

Introduction

- **Similarly to functional scheme:**
  definition of a general technological scheme

- **For a given project, on this general technological basis, the actual technological solution will be defined starting from the specific functional scheme**
- **Remark: the choice of the specific components should be driven by the functional/behavioural requirements**
    - **Remember: some additional troubles can come from implementation**
        - **E.g.: communication delays not considered at functional levels**

General Technological Scheme
Model/Architecture



General Technological Scheme

General Technological Model Architecture: Components

- **Computing Units:**
  - Different technologies
  - Remote location
- **Communication among computing units**
  - Depending on comp.units techn. (tipically electronics)
  - Different technologies in the same project
  - Can mimic the functional architecture
- **Sensors and Interfacing**
  - Field measures
  - “Intelligent” sensors
    - Configurability
    - Integration in computing units communication systems (remove dedicated channels)

General Technological Model Architecture: Components (cont'd)

- **Actuators and interfacing**
  - **Actions on the field**
  - **"Power" devices**
  - **"Intelligent" actuators**
    - **Local control given by the manufacturer (e.g.: electric drive)**
    - **Configurability**
    - **Integration in computing units communication systems (remove dedicated channels)**
- **External communication**
  - **HMI**
  - **Other levels of the automation pyramid**
  - **Integrate in communication among computing units**

COMPONENTS OF THE GENERAL TECHNOLOGICAL SCHEME

Computing Units

Technologies:

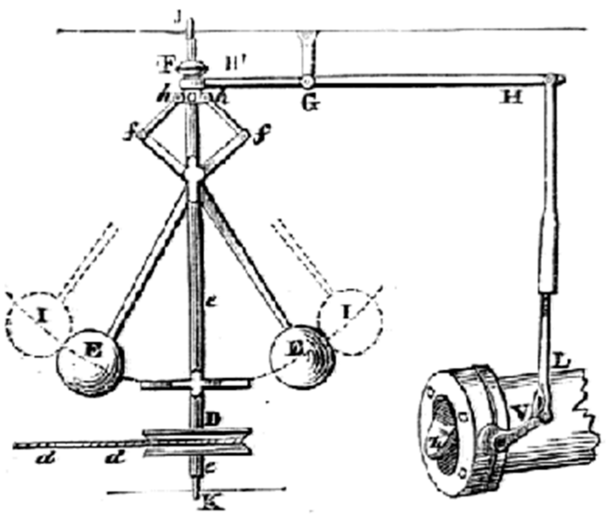
- **Programmable Digital Electronics + Informatics**
  - **Different kind: PC, microcontrollers, microprocessors, DSP, etc.**
    - **Performance, characteristics**
  - **<u>Dominant</u> (as already anticipated implicitly)**
    - **But "not to be used in any case"**

- **In the past controllers were "technologically homogeneous" with the plant**
  - **E.g.: Watt's speed regulator**
  - **No clear separation between controller and plant (functional concepts as control and feedback were not clear)**

Computing units

Tecnology:

- **Watt's speed regulator**



Computing units

Technologies:

- Advantages of Informatics – digital electronics for complex control systems:
  - COMPUTING POWER / RELIABILITY
  - FLEXIBILITY
  - INTERACTION WITH USERS
  - EXTERNAL COMMUNICATION
- Problems of Informatics – Digital Electronics for complex control systems:
  - SAMPLING
  - QUANTIZATION

(see “Digital Control” in “Automatic Control 2” and “Controlli Automatici T2”)

Computing Units

Technologies:

- **Informatics – Digital Electronics:**
  Electronics and Informatics "customized" for control application
  - **Time constraints**
  - **Solution:**
    - "Ad hoc" electronics
    - Real-Time informatics

(problem/solutions for real-time programming will be considered later on)

Computing Units

Technologies:

- **Warning: for simple systems and without frequent reconfiguration, controllers which are homogeneous with the plant are still adopted (and also the most reasonable)**

  - **E.g.: pressure regulators for fuel injection systems in IC engines**

  - **Level, pressure or flow regulators adopted in hydraulic applications.**

Components of the general technological scheme

Computing units

Technologies:

- Other solutions: Analog Elettronics e Electromechanics
- Computing units based on analog elettronics or electromechanics (switches and relays), largely adopted in the past, nowadays are declining.
  - Former “classic” continuous-time ctrl ➔ analog electronics
  - Former logic controllers ➔ elettromechanics
    - Remark: “switches and relays” allows to reproduce all the combinatorial and sequential logics
      - Considered later on…

Computing units

Technologies:

- Computing units based on analog electronics or electromechanics (switches and relays) are practically "endangered"
- Relevant exceptions:
  1) time-dependent-system controller where cost/sampling time trade-off is critical ➔ analog electronics
  - or non-programmable digital electronics or PLD/FPGA
  - E.g.: current controllers for electrovalves

Computing units

Technologies:

- Computing units based on analog electronics or electromechanics (switches and ralays) are practically "endangered"

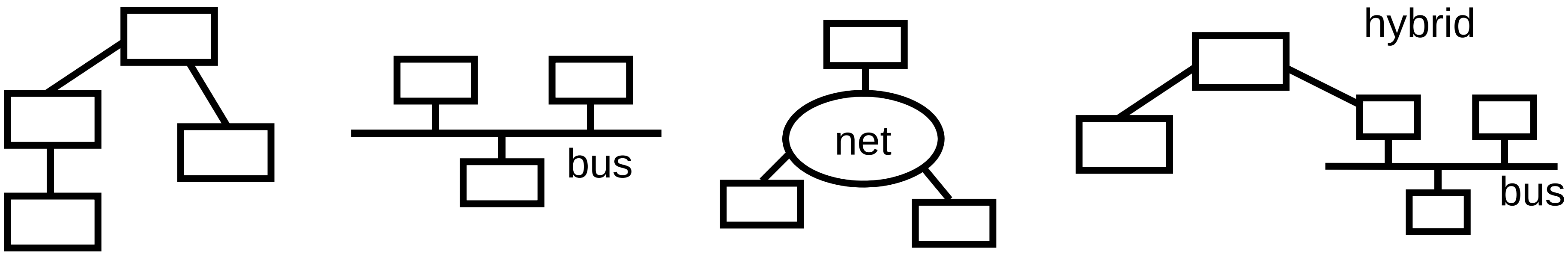
- Relevant exceptions:
  2) Simple logic controllers with no flexibility reqs. ➔ electromechanics

  - often legacy from past solutions
  - e.g.: relay rack for lift control (old…)

Components of the general technological scheme

Computing Units Communication System

- **For Electronic computing units**
  - **Mainly digital**
- **Characteristics:**
  - **Digital (rarely analog, legacy of past solutions)**
  - **Point-to-point (often mimics the functional scheme)**
    - **Specific sizing of the channel according to functional traffic**
    - **Sometimes custom solution**
  - **Bus/Net: Standard (FieldBuses, variants of Ethernet)**
    - **Simplified cabling**



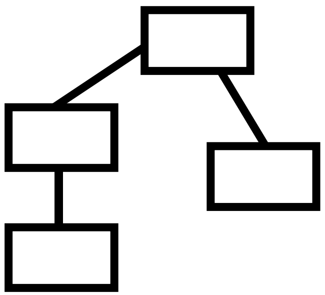
Components of the general technological scheme

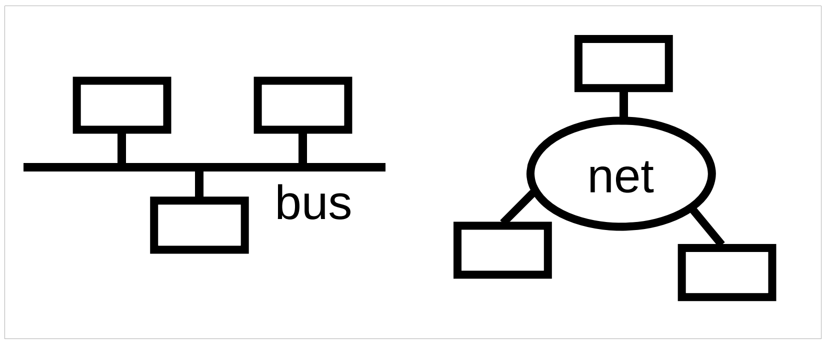
Computing Units Communication System

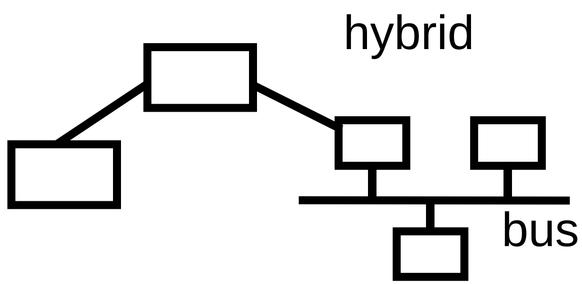
- **<u>Most promising</u>: <u>Standard Bus/Net</u>**
  - **Reusability, independence on functional architecture**
  - **Simple expansion/modularity/interoperability**
    - **Problem: standardization at application level…**

- **Problems for control:**
  - delay
  - determinism (collisions)

  ← depending on number of nodes and their traffic ➔ pay attention in reusing







Components of the general technological scheme
Sensors and Interfacing

- **Sensors: transfer information from the plant physical domain to the computing unit domain**
  - **Typically "final domain" is electric (electronics computing units)**

- **Interfacing**
  - **Common sensors: analog output**
  - **Common comp. unit: digital input**

  ⇨ Level conditioning + A/D Conversion

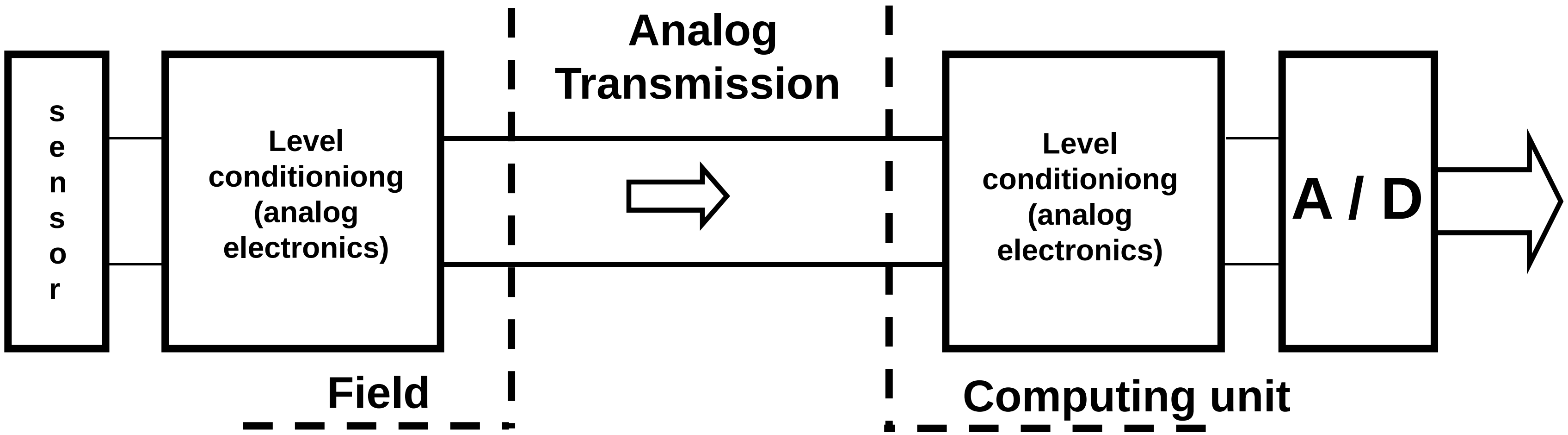
  - **Sensors far from Computing Unit**

  ⇨ Transmission

Components of the general technological scheme

Sensors and Interfacing

- **Interfacing architectures:**

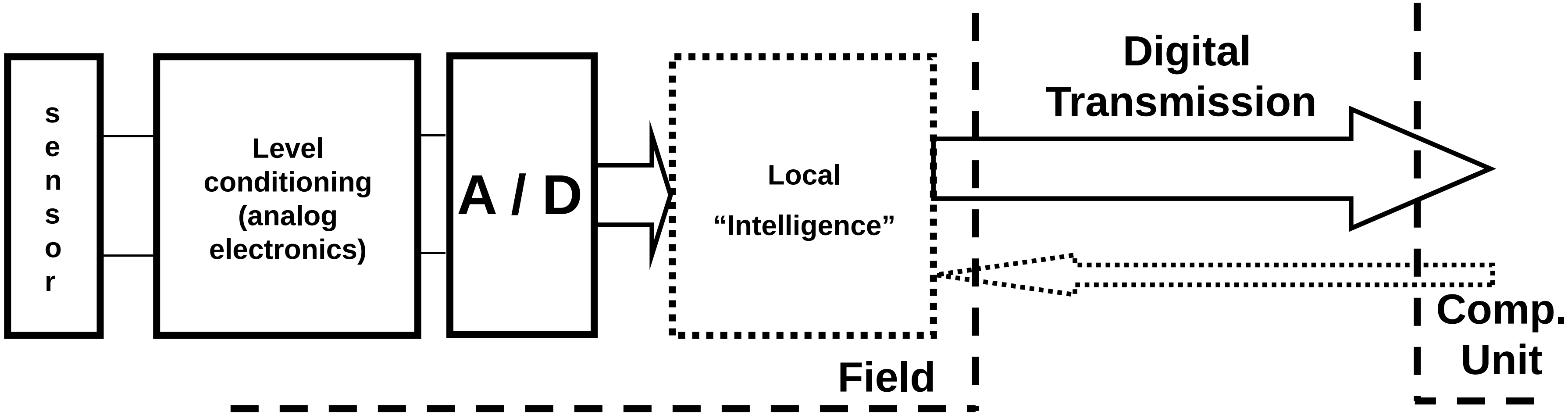


- **Usually: sensor signal (current or voltage), level adaptation and/or converted (voltage/current) and transmitted**
  - **No "particular" techniques (modulation etc.)**
  - **Low frequency signal: no propagation phenomena**
- **Point-to-point transmission (seldom more receivers in parallel)**
- **Pros: Simple; Cons: EMC; Cabling with many sensors**

Components of the general technological scheme

Sensors and Interfacing

- **Interfacing architectures:**



- **Pros:**
  EMC robustness; Flexibilty; Bus or Net (simpler cabling, integration with computing unit communication system)
- **Cons:**
  "Complexity" on field (supply, logics, hostile environment); Communication delay; Determinism! (NOT TRIVIAL)

Components of the general technological scheme

Actuators and Interfacing

- **Actuators: convert control commands in actions on physical plant (domain and power)**
  - **Different physical domains**
  - **“Signal” and “Power”**

- **Interfacing:**
  - **Dual w.r.t. sensors**
  - **Similar solutions… but something more:**

    Often “local intelligence” in actuator ➔ complete feedback controller
    - **E.g.: Electric Drives (later on…)**

      - **A curiosity: Digital Ctrl, Digital drive, but analog interface! (legacy of past solutions)**

External Communication

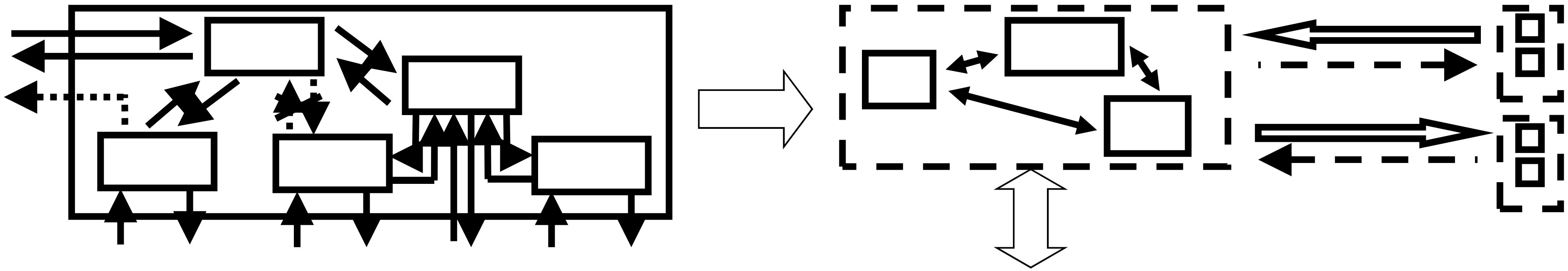
- Again referred to digital computing units

➢ DIGITAL STANDARD BUS OR NET

- **Application:**
  - HMI
  - Communication with higher Automation Pyramid(supervision/managment)
- **NO hard real-time constraints**
- **Standard**
  - Ethernet-TCP/IP
- **<u>Avoid relevant computational load for computing units</u>**
  - Control is the main objective (time constraints)!
  - Decoupling

GUIDE-LINES TO MAP A SPECIFIC FUNCTIONAL SOLUTION IN A TECHNOLOGICAL IMPLEMENTATION

Objective



- Given the functional project of a <u>specific</u> control system
  - Usually compliant with the general functional scheme
- Design the corresponding technological scheme
  - Should be compliant with the general technological scheme
    - Otherwise: hard to be realized, not standard
  - Possibly optimal: best trade-off performance/costs
    - Performance: also flexibility, expandibility…

Proposed Guidelines

1a) For each control function/algorithm, define:

- the type of necessary computing unit

- Digital electronics + informatics: PC, PLC, microcontroller
  - Different “computing power”
- Analog electronics and/or FPGA-PLD-non programmable logics
- Electro-mechanics (relays etc.)
- ….. (often a-priori constraints)

- detailed implementation requirements

- Sampling time
- Numeric precision
- …

Proposed guidelines

1b) Define typology and specs for sensors and actuators to be used

- Range / size
- Accuracy
- …..

2) Group control functions/algorithms depending on:
- similarity in the type of the required computing unit
- hierarchical closeness of functions or closeness of the systems to be controlled

➔ define the set of computing units to be used

- Minimize numbers / costs
- Many functions/algorithms on 1 unit: multitasking…
- Save margin…

Proposed Guidelines

3) **Define communication systems among computing units and interfacing with sensors/actuators**
    - Possible integration
    - Remark:
      Make explicit and satisfy speed and determinism reqs coming from functional design
      Reserved channels for critical communication/interfacing
4) **Define external communication system**
    - Minimize impacts on other functions

Remark: obviously some steps could cause review of previous ones

Proposed Guidelines

Comments:

We have seen “what to do”, not “how to do”.

For “how to do” deep knowledge of typical components of technological platform is crucial.

Proposed Guidelines

u

Comments:

Once the technological architecture has been defined,

Detailed designs (or buying) begin for all components

- **Different designers (complex systems)**

Problems in tech architecture should be found ASAP

⇩

Better spending time at the beginning in overall system analysis, before starting detailed design.

(generally true…)

In the following…

Next topics:

- **Deeper analysis of technological components**
  - **Design issues for some of them**

- **Mapping is not deeply considered**
  - **Customized for each case**
- **We will start with electronics and informatics for control-oriented computing units.**

Then:

Fundamental digression: functional design of logic control

Then:

Sensors and Actuators (elec. drives) for industrial automation

Ingegneria e Tecnologie dei Sistemi di Controllo T
Control Systems Technologies

Architectures and Technologies in Control Systems

The End

Ing. Christian Conficoni (prof. Andrea Tilli)
DEI - Alma Mater Studiorum Università di Bologna
E-mail: christian.conficoni@unibo.it
https://www.unibo.it/sitoweb/christian.conficoni3