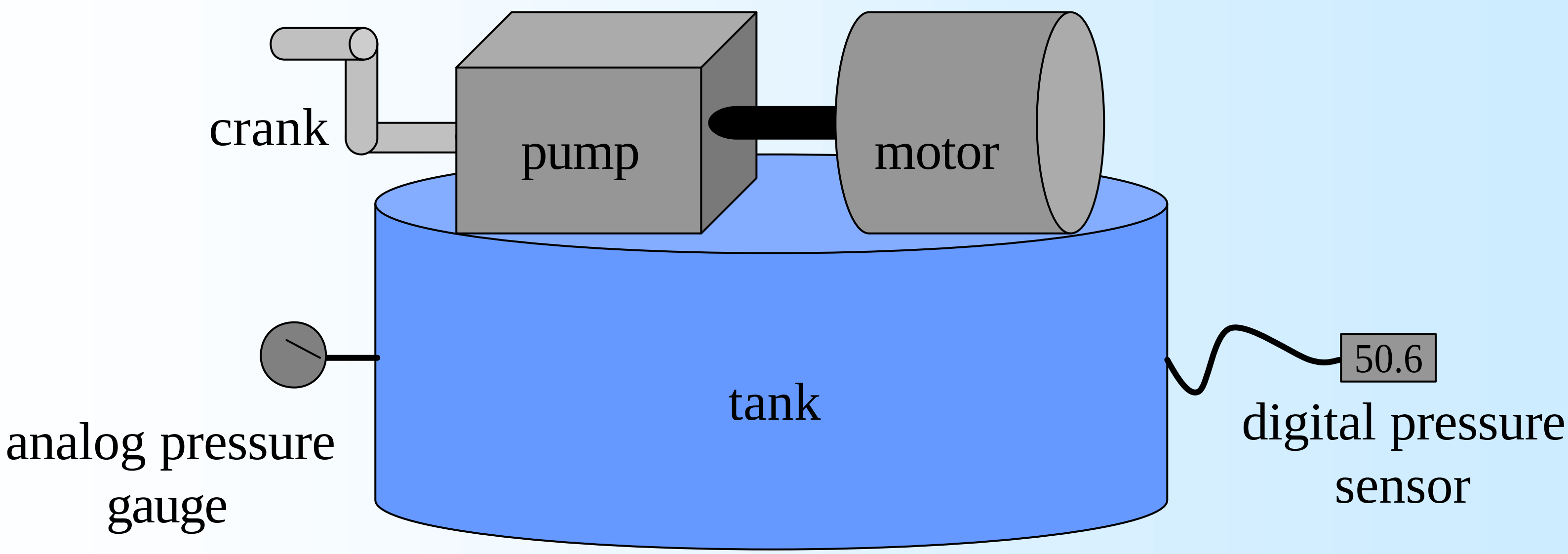
Design of AA systems

- Strive for full autonomy where possible.
- Build in appropriate sensing, even for situations in which the human is in control
- Build in capabilities that enable multiple methods for humans and system to achieve goals
- Plan for changes in autonomy as much as possible
  - system-planned changes in autonomy
  - system-initiated changes in autonomy
  - human-initiated changes in autonomy

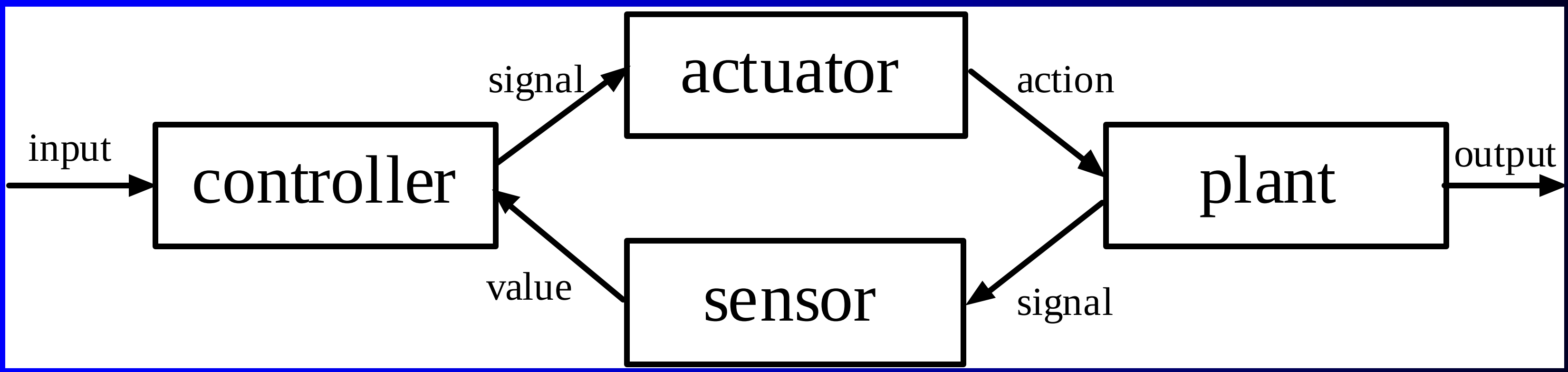
A simple example

- Simple example that will be discussed throughout this tutorial



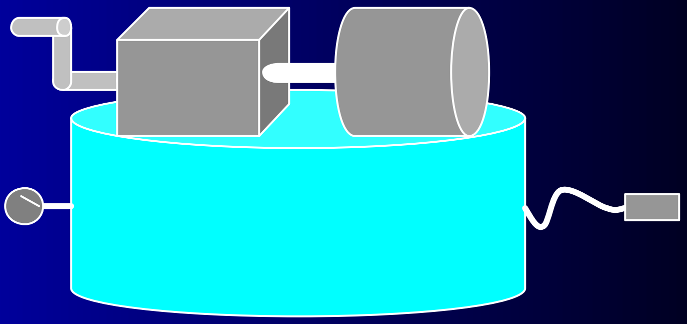
Generic control system

- Closed-loop control
  – Sensed output of plant (known via sensors) is compared to desired state of plant (input) and control of the plant through actuators is adjusted by the controller



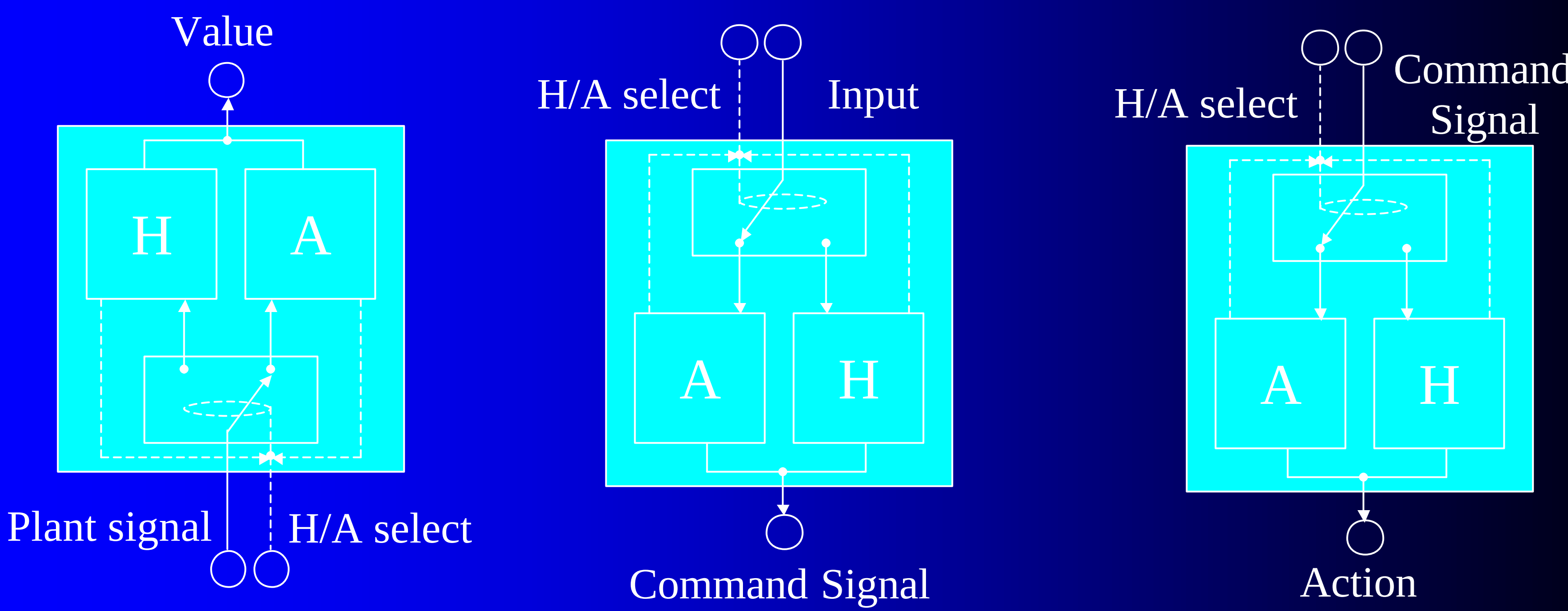
Our running example

- Controller
  - human who decides whether pressure should be increased
  - computer that decides whether pressure should be increased
- Actuator
  - pump that human cranks
  - pump that motor activates



- Sensor
  - digital pressure sensor that computer reads
  - analog pressure gauge that human reads

AA control system components

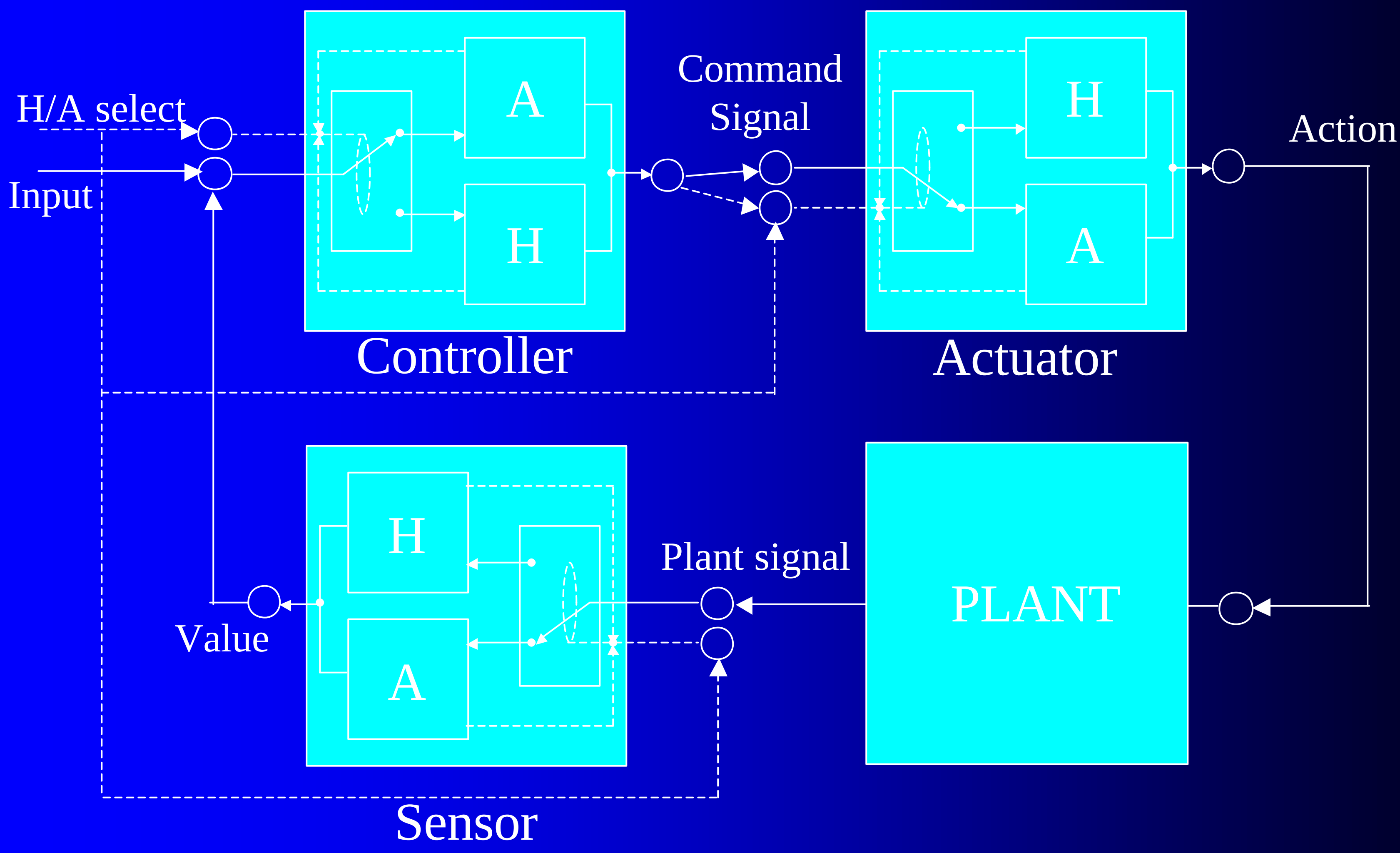


Sensor

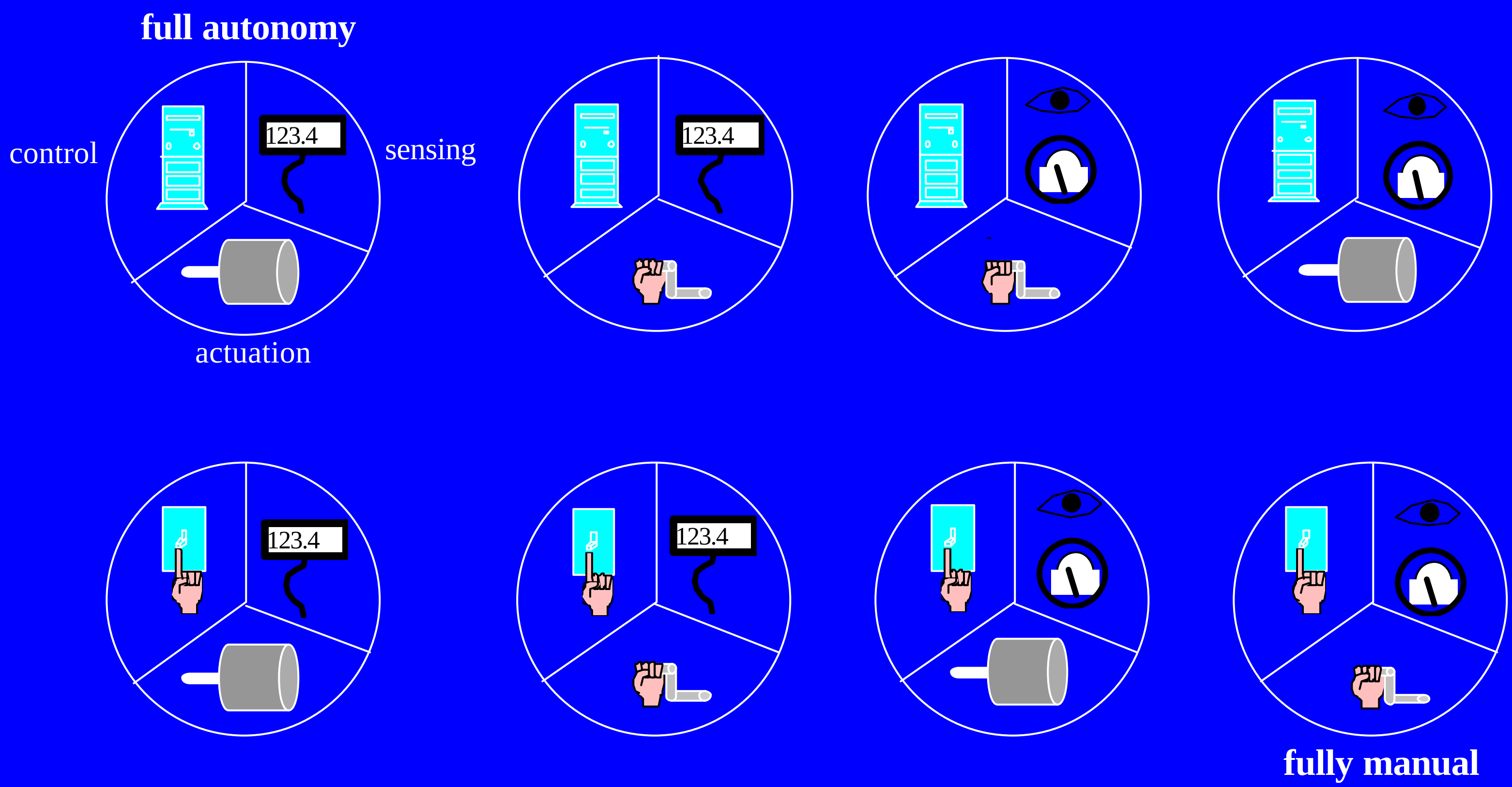
Controller

Actuator

A Simple AA Control System



Different autonomy modes



Adjusting Autonomy Example

- Control system believes motor is broken and is no longer responding to commands
  - Belief is either by internal processes or external command
- Autonomy is switched from autonomous (motor) to crank (human) for pump
- Autonomous control system still decides when pump needs to be activated
  - Human is only an actuator, not a controller

Example RAP Code

```
(define-rap increase-pressure ?maximum-pressure
  (context (equal actuator-autonomy-level ‘A)
    (task1 ‘(turn-on-pump))
    (wait-for (pressure-at ?maximum-pressure))
    (task2 ‘(turn-off-pump)))
  (context (equal actuator-autonomy-level ‘H)
   (task1 ‘(tell-user “Start cranking pump”)
   (wait-for (pressure-at ?maximum-pressure))
   (task2 ‘(tell-user “Stop cranking pump”)))
```

Labeling autonomy levels

- Difficult for human operator to know or set level of autonomy
- Design labels to help
  - tele-operated
  - supervisory
  - autonomous
  - …
- Labels can be tied to specific tasks

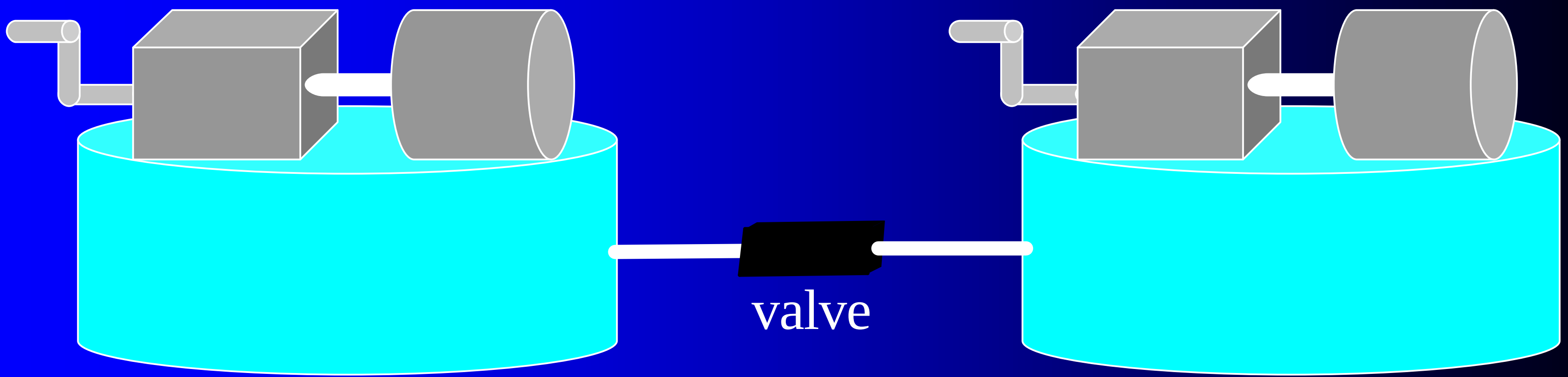
Communicating autonomy level

- Human operator needs to know the current state of the system, including autonomy level
- Must be in a form that is easy to understand "at a glance"
- Autonomy level of subsystems may be different
  - how to represent global autonomy level

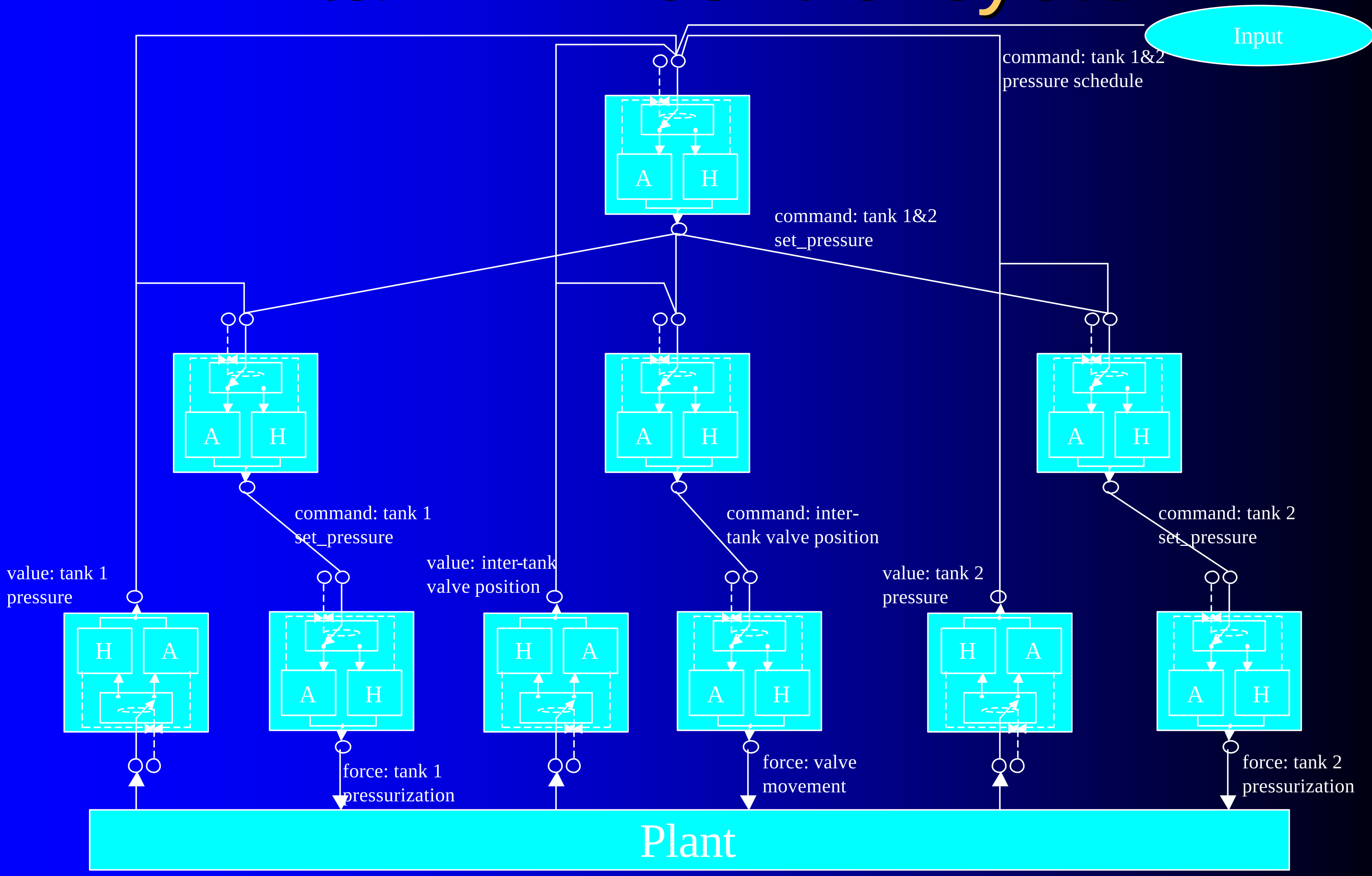
AA Hierarchies

- Multiple sensors, actuators and controllers make the combinatorics of adjustable autonomy nasty
- Arranging adjustable autonomy in a hierarchy can help alleviate these problems
  - setting level of autonomy at top of hierarchy and have it apply to all systems below
- Significant design decisions in arranging hierarchy
  - what level(s) can talk directly to sensors/actuators
  - is it a tree or a graph?

Expanding our example

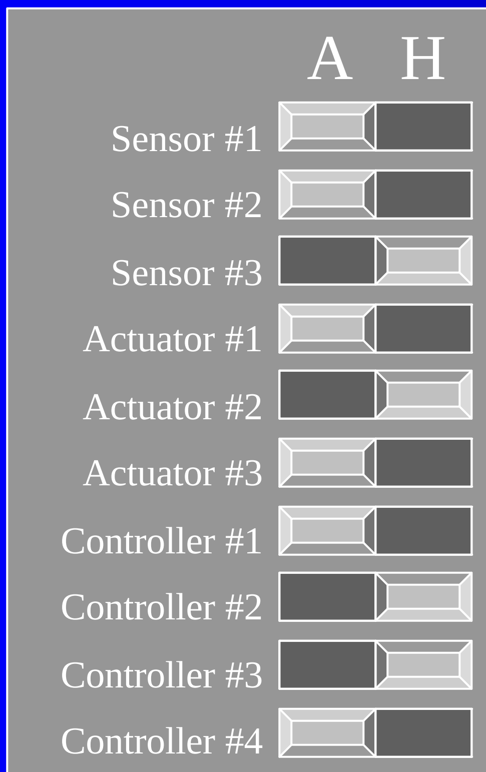


A 2-tank AA control system



Autonomy Level Selection

Autonomy Level Selector



For a system with:

s sensors
a actuators
c controllers

The possible number of autonomy levels = $2^{(s+a+c)}$

In this case, $2^{10} = 1024$

Autonomy Level Considerations

- Not all autonomy levels valid, e.g., Actuator #2 must always be set to H and at least one sensor must always be set to H.
- It may not be permitted to directly switch from one valid autonomy level to another valid autonomy level, you may be required to transition to one or more intermediate autonomy levels. The number of possible transitions: $(n^2-n)/2$ where n = number of autonomy levels. When n=1024, the number of transitions is 523,776.

Autonomy Level Considerations cont.

- If a transition may be initiated only by a human (with the proper clearance level), or only by the autonomous system or either, then the number of possible transitions doubles to n^2-n.
- Undesirable cycles may form, e.g., human selects an autonomy level and the autonomous system reselects the previous autonomy level.
- The set of valid autonomy levels and transitions may change over time, e.g., change of personnel shifts, equipment maintenance, or failure.

Design decisions

- What tasks can be done only by humans? Only by automation? By both?
  - Are there certain times or situations when a task should only be done by a human or automation?
- Who can set the level of autonomy for a task?
  - Can the level of autonomy change at any time or only under certain circumstances
- How will autonomy level(s) be set?
- How will autonomy level(s) be recognized?
- What will the system hierarchy look like?