Control System Design

Control Lab

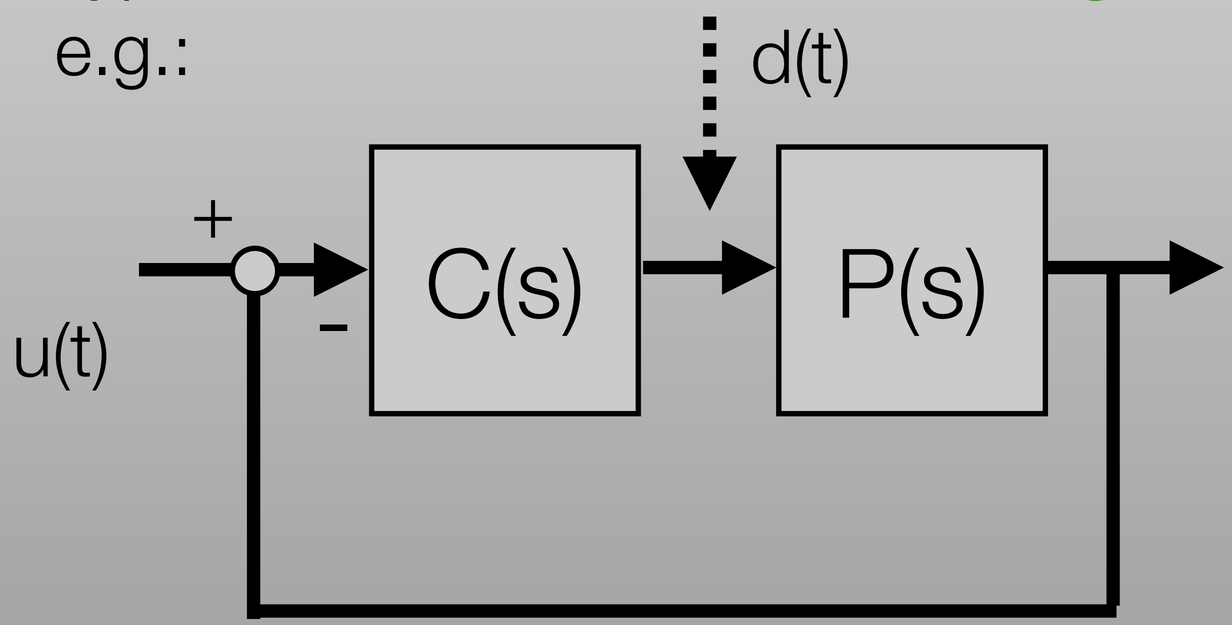
Instructor: Francesco Ticozzi

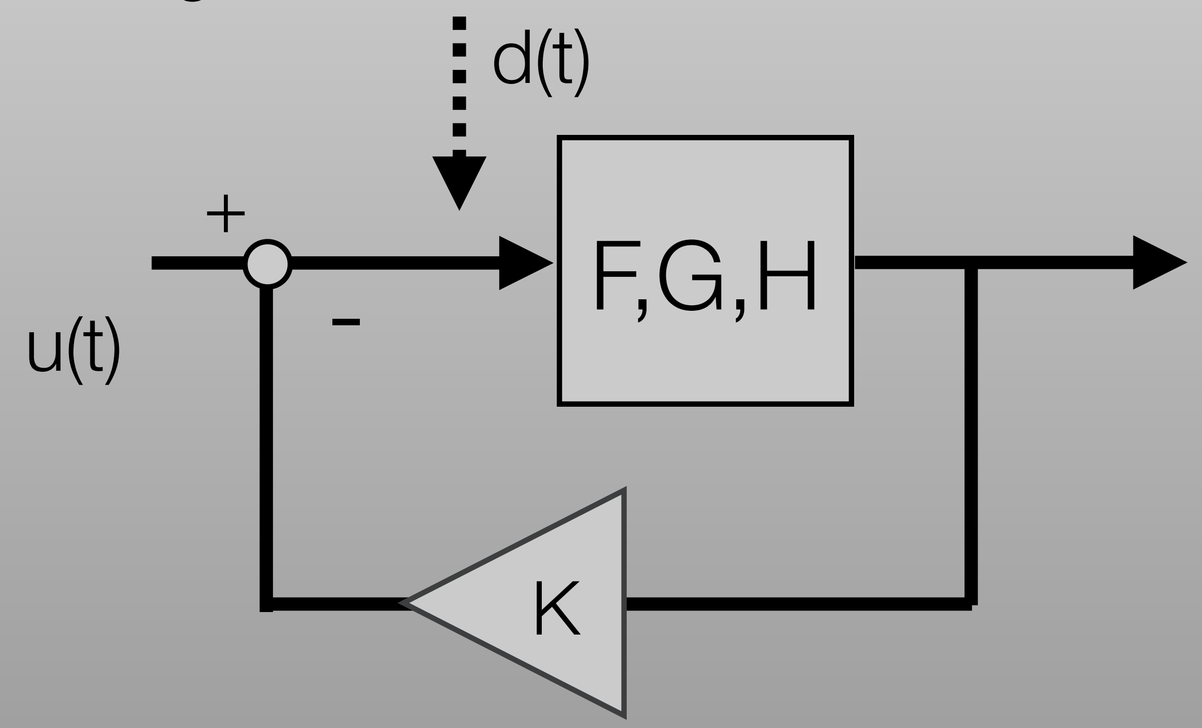
What have you been trained to do?

- In the courses of classical (transfer-function based) and modern (state-space based) control, control design is typically tackled in a very "mechanical" way, following a rigid recipe.

- **We assume to be given a model:** linear, lumped-parameter ODE; LTI, SISO, convolutional and causal TF; or a state-space representation.

- Typical Task: to allocate **poles/eigenvalues** given a controller structure, e.g.:





- **When I face a real-world problem, I have to consider...**

... for the System / Plant:

- The model could not be available (in the needed form):
  Modeling from basic laws or **learning from data (Black-Box)**;

- The process is typically non-linear:
  Linearization, Limits on the validity of the linear approximation;
  Idealization and refinement by trial-and-error;

- Time-varying processes (even if typically slowly so):
  Determining whether it can be considered TI on time intervals (how small?)

- Linear Process with NON-RATIONAL transfer function *P(s)*,
  e.g. DELAY in the feedback loop (= exp(Ts)):
  Approximation with rational ones (they are "dense" subset),
  use of more general tools (e.g. Small gain theorem);

... for the System / Plant (continued):

- Presence of disturbances:
  Disturbance Decoupling,
  Disturbance Rejection (for some classes),
  Shaping of the Filtering Properties (if decoupled from output),
  Feedforward,
  Robust Control;

- Uncertainty in the plant model:
  Identification,
  Adaptive Control,
  Robust Control;

... for the Controller:

- Deciding the controller structure is a delicate step:
  Different type of controllers have different advantages
  (e.g. PID are well-known and easy to re-tune);
  All components/blocks have to be carefully designed, realized and interfaced in the real system;

- **The choice of structure must be part of the design process:**
  Tradeoff between flexibility, cost and simplicity.
  If it is not there it won't break or create problems…
  Start with a simple one and move to more complex if needed;

- Input and output signals to/from different blocks are not typically compatible:
  Choose and build models for Converters, Interfaces, Actuators, Sensors, etc;

Design of Control Systems

Let’s review the fundamental steps (sketch of an algorithm):

1. **Qualitative description and analysis of the system we want to control:**

   1.1. **Which are the physical components?**

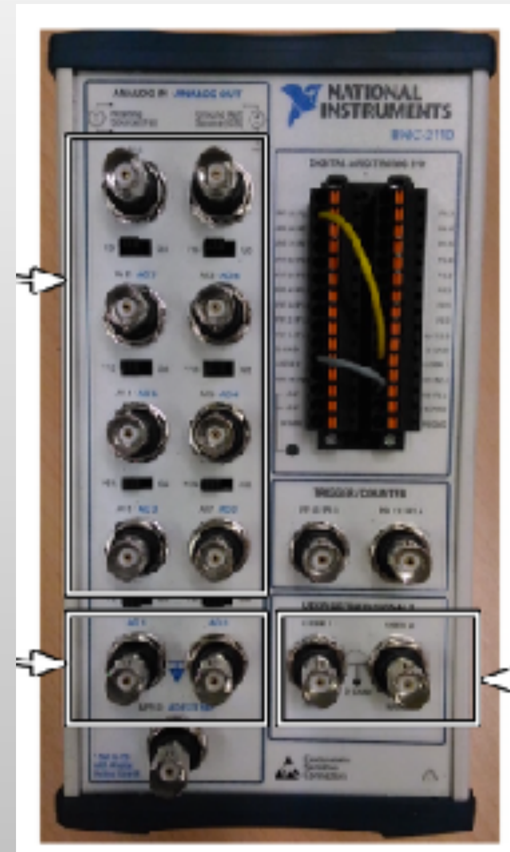
   1.2. **Which quantities and signals are relevant (input-output-etc)? On which order of magnitude? *Units*??**

   1.3. **Which kind of model is more natural/works better? (Deterministic, Statistical, Stochastic,...)**

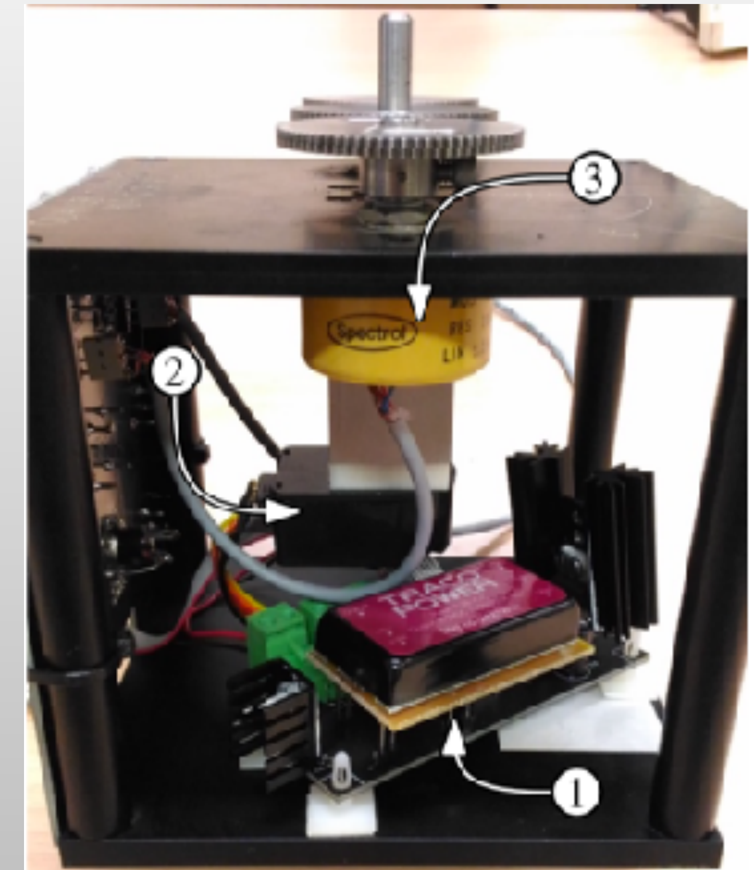
   1.4. **From application and control requirements, guess the level of detail needed and orient yourself toward a design approach;**

Our System (LAB 0,1,2,3; 4 will be different)

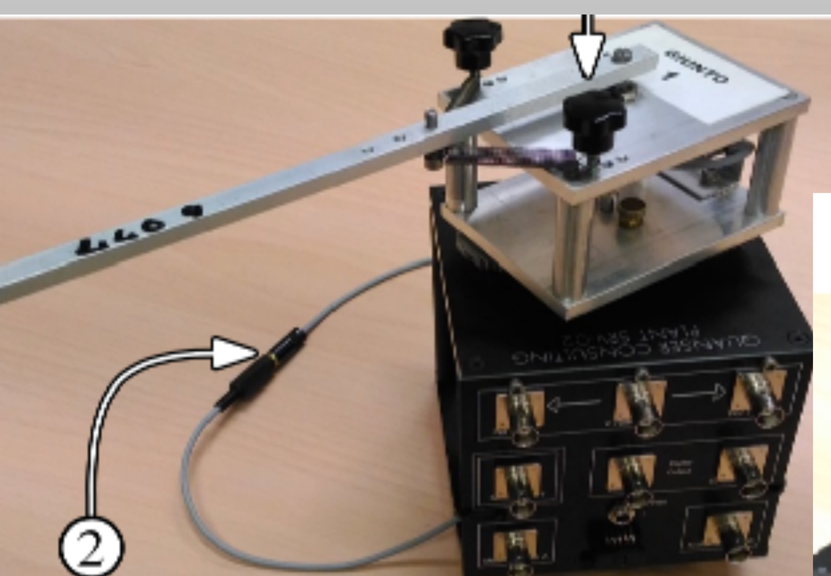
Computer running
MATLAB (realtime)



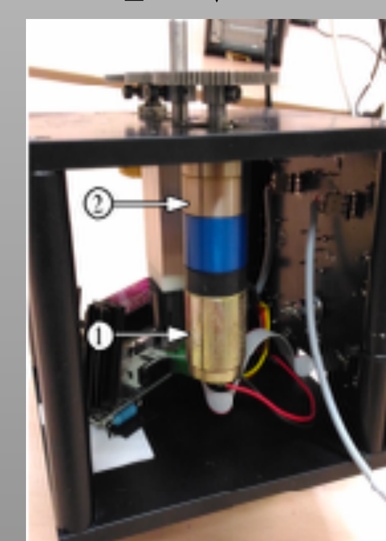
I/O interface



Power drive and
DC engine



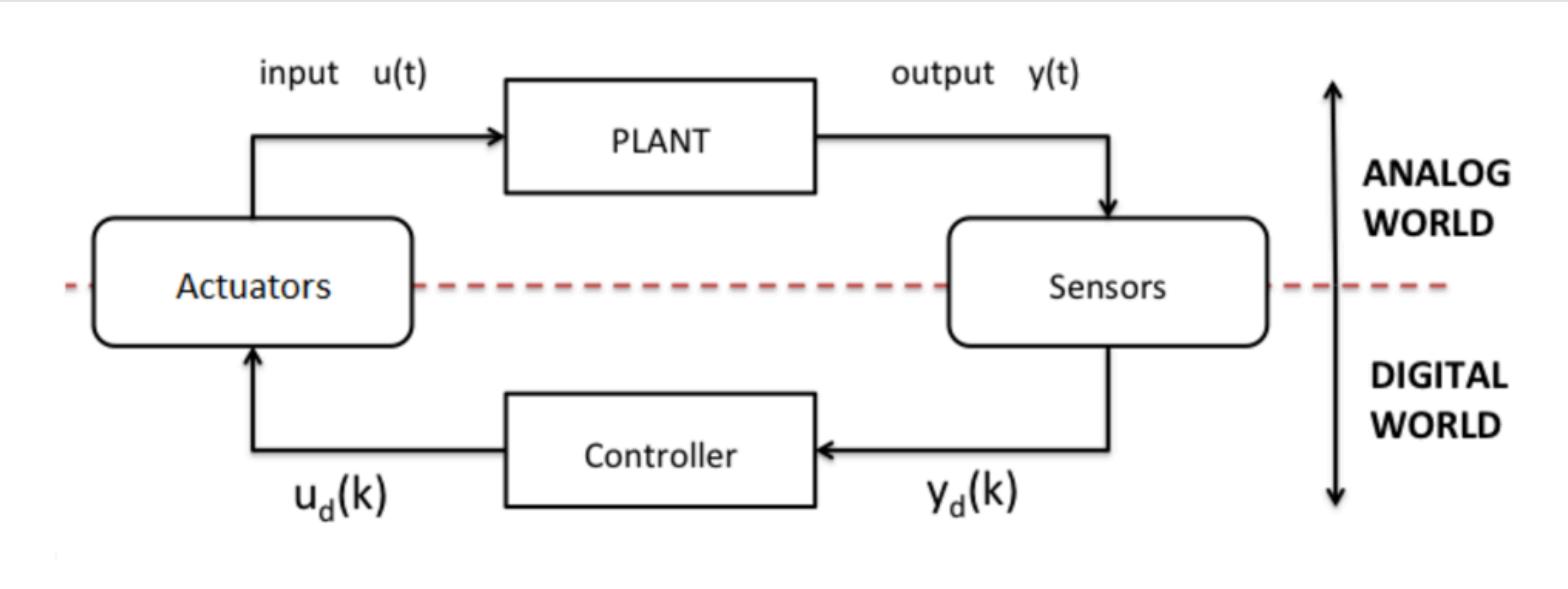
Loads:
Disk or Flexible Joint



Gear box

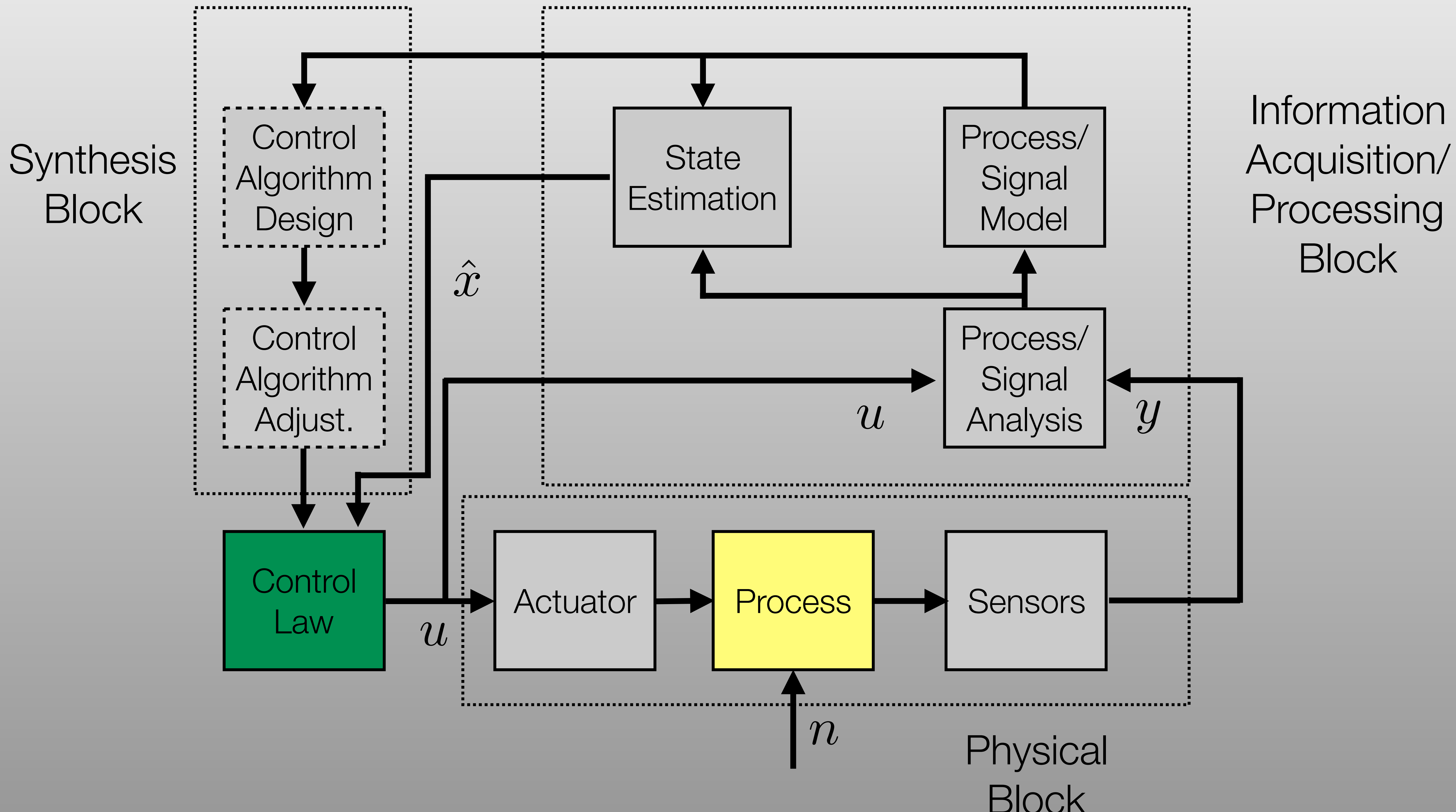
Analog vs Digital Domains (in closed loop)

- The system includes both analog and physical components, as well as their interfaces:



- Particular attention needs to be payed to AD DA converters.
  We will see a good part of the contents of “*digital control systems*” in practice.

Digital Control System Architecture



Design of Control Systems

2. **Building up a model:**

2.1. **From basic physical laws/first principles;**

We will derive the model(s) from physics.

Key issue: complexity vs accuracy;

simple approximations often work.

2.2. **From data:** Fix a class of model and find the one that fits your data “the best”, according to some criterion;

- **Parametric class of models;**

We will refine the model estimating parameters (friction) from data;

- **Non-Parametric class of models;**

2.3. **Inclusion of Uncertainty, Noise, Non-Ideal characteristics, etc ...**

Design of Control Systems

3. **Validation of the model:**

   3.1. **Test the model on data (different from the one used for identification);**

   Indirectly: we will use the estimated parameters to improve controller's performance. If it works, good estimate/model.

   3.2. **If it doesn't work: Re-tune the parameters, Refine the model, Different or larger class or type, less approximations and ideal-behavior hypothesis..**

Design of Control Systems

4. **Control Problem and Its Solution**

4.1. **Acquire (and translate if needed) Controlled Performance Requirements;**

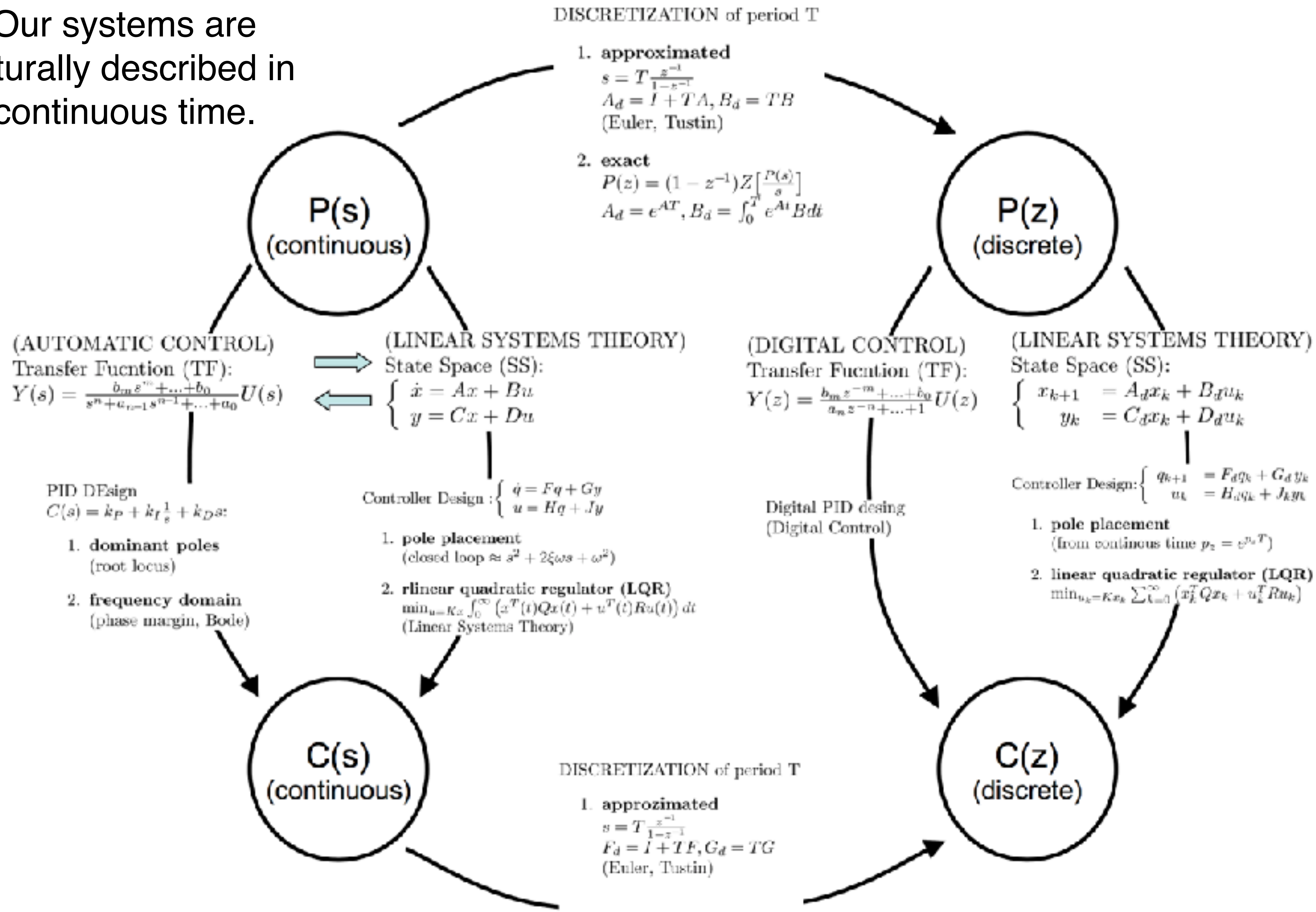
4.2. **Choose:**
-appropriate representation for time on the controller side(Cont. vs disc.; Sampling time);
- architecture (open-loop, feedback, feedforward, precomp;…)**;**

4.3. **Pick a design approach and try to solve the control problem;**
(Standard controller; pole allocation; eigenvalue allocation, LQR)

4.4. **If it doesn't work:**
Change design approach, or change structure;
Change type of model or refine existing one;

Some of the approaches we will use…

Our systems are naturally described in continuous time.



Design of Control Systems

5. **Simulation:**

   5.1. **Test the controlled system with numerical simulation;**
   Include most relevant non-ideal features and do some worse case analysis.
   (*virtual prototyping*... especially if the system is expensive!)

   5.2. **If it doesn't work: Back to 4. or 2.;**

6. **Implementation and Experimental Verification;**

   6.1.**Choose implementation type (also analogic vs digital);**

   6.2. **Choose Actuators and Sensors if not already part of the model;**

   6.3. **If it doesn't work: Tune the controller, or back to 6.1-2. or 4. or 2.**