Decentralised/plantwide control in industry: the BIG PICTURE

Antonio Sala

Lecture notes on control of complex systems
Universitat Politècnica de València, Spain

Video-presentations:
http://personales.upv.es/asala/YT/V/**pidbigpic0EN**.html (introduction, motivation)
http://personales.upv.es/asala/YT/V/**pidbigpic1EN**.html (process design)
http://personales.upv.es/asala/YT/V/**pidbigpic1BEN**.html (control structure)
http://personales.upv.es/asala/YT/V/**pidbigpic2EN**.html (controller tuning, implementation, data analysis)

Presentation

Motivation: There are soo many things told to you in soo many control courses... Courses concentrate on particular details but often omit the "big picture" on what you will see in industry, what THEY think your expertise should be.

Contents: almost everything you need to have in mind in "advanced control of complex industrial systems" to speak to technicians, management, engineers not specialized in control, without "Math jargon".

The “grand scheme of things”: master plan

“Process engineers”, “control engineers” may be assigned one of the following tasks (or must oversee several of them in “senior” roles):

1. Prior to control design/tuning: Control-oriented PROCESS design
2. Control Structure selection (information flow: who sees what, who does what)
3. Controller tuning (basically PIDs + predictive)
4. Sorting out implementation details (SCADA, network, event logging)
5. *Post-hoc* data analysis, quality control, continuous improvement

CV: controlled variable, **MV:** manipulated variable.

Alternative views of the "control problem"

- **Manufacturing:** Inventory control, scheduling, coordination of PLCs, robotics, people/machine collaboration, quality control
- **Aerospace:** More presence of "centralized" (state-space, matrices) formulae, Kalman filter (sensor fusion accelerometer+gps), LPV systems (linear parameter varying, linearization depends on wildly changing flight conditions from Mach 0.2 at 200 m altitude to Mach 4 at 15 Km altitude, say), mobile robotics, rotations, quaternions
- **Ph.D. student:** abstractions of control of interest to a theory-oriented audience (Mathematics: differential equations, Markov processes, stochastic processes); advanced physics or chemical kinetics for accurate modelling, adaptive & nonlinear control, . . .

1. Control-oriented PROCESS design is VERY important

- **Optimize operating point** to minimise "variable operation cost".
- Allow from som **"extra" MV power** to recover **quickly enough** in expected setpoint changes and disturbance scenarios departing from the nominal operating point.
- Select CV, MV so that we have a "fast" and "high-gain" MV→CV dynamics, and suitable disturbance rejection, i.e., low amplitude of "feedforward corrections" at frequencies of interest.
- In multivariable case, check that SVD principal maneuvres are sensible, check conditioning.

GET AS MANY GOOD SENSORS AS YOU CAN:

- **Control performance heavily depends on SENSOR technology!** Steering an autonomous car is (sort of) "**straightforward**"... IF you know its position and speed and that of obstacles, other cars, pedestrians, ...
- If it's cheap, go BUY it. A sensor never hurts.
  - The more sensors you have, the simpler your control/decision strategy will end up being: it's always better "measuring" than "simulating a not-so-accurate model" (unless your signal-to-noise ratio is rubbish, of course).
- **ideally "state feedback", "full-information control":** *if you accurately measure* **everything important**, *then proportional control does miracles.*

Theory says $u = -Kx$... and you need "controllability", we'll skip details.

1. Control-oriented PROCESS design is VERY important

If there is a **design flaw**

- bad plant conditioning,
- not enough actuator power for not-so-infrequent events,
- too few and/or noisy sensors,
- excessive disturbance effect,
- complex & slow MV→CV dynamics,
- lack of repeatability,
- heavy nonlinearity,
- excessive fault rate, . . .

probably, **no control strategy will solve my problems**.

However, if the process is **well designed**, then **simple, easy to understand,** strategies will perform **very well**.

2. Control structure selection

First, choose "**centralized**" vs. "**decentralized**"?.

Feeding "everything" in an oil refinery to a computer considering, say, a 310×190 transfer function matrix is **NOT** an option.

Thus, we will, in principle, stick to **decentralized** options.

In industry, usually, a "**centralized**" approach is pursued only if deficiencies arise or key primary controlled variables are at stake; some sub-systems are then controlled with centralized (i.e., multivariable predictive, LQR, Kalman, $\mathcal{H}_\infty$, ...) strategies.

*Multivariable predictive control loops for a handful of key subsystems is somehow frequent in, say, chemical industry. However, we'll not cover it in the scope of this material. Predictive (and other centralized options) will be our last tool when the things we discuss here cannot achieve the required performance and there are no further cheap sensing/actuating options left.

2. Control structure selection (decentralized)

Decentralized-based situations when building a control structure:

1. BASE setup:
   controlled vbles. = measured vbles; number of them = n. of manipulated vbles.
2. Missing sensors
3. Extra sensors
4. Extra manipulated variables
5. Safety loops
6. Refinements

2. Control structure, 2.1 Base setup

1. **BASE setup: controlled vbles. = measured vbles; number = n. of manipulated vbles.**
   - basic option: MULTILOOP, pairings with "*common-sense*" or RGA methodology
   - Diminish interaction by DECOUPLING, if needed.
   - **Ratio control** useful and easy to understand in many chemical processes (it's actually nonlinear control, but don't tell anyone), to decouple flow/level/temperature from concentration.
   - If plant is not diagonal-dominant or it is non-square, consider SVD decoupling... this relates to items 3 & 4 below, but SVD decoupling has the same number of "virtual" CV and MV.
2. Missing sensors
3. Extra sensors
4. Extra manipulated variables
5. Safety loops

2. Control structure, 2.2 Missing sensors

1. BASE setup: controlled vbles. = measured vbles; number = manipulated vbles.
2. **Missing sensors**
   - **Open-loop control:** of course, only "setpoint changes" can be scheduled in open loop; let's hope model is good and disturbances are not too large.
   - **Inferential/Indirect control.**
   - The "extreme" case is the "**state observer**" (idealized abstraction): from a "fraction" of sensors and the model, infer "all" of the state variables (we need an "observable" system). Inferential control requires estimating only the non-measured controlled variables, maybe with a simpler model considering only a sub-system of the whole plant to be controlled.
3. Extra sensors
4. Extra manipulated variables
5. Safety loops

2. Control structure, 2.3 Extra sensors

1. BASE setup: controlled vbles. = measured vbles; number = manipulated vbles.
2. Missing sensors
3. **Extra sensors**
   - **Sensor fusion:** reduce noise using several measurements related to the same state variable or controlled variable. Trivial approach: just "average" them; formal approach: Kalman filter.
   - **extra outputs** (affected by my MV): CASCADE [extraSens]... ideally with time-scale separation for independent tuning.
   - **disturbances** (not affected by my MV): FEEDFORWARD.
   - Info from **other subsystems/control loops**: DECOUPLING (close cousin to feedforward: knowing something that affects me but I don't have control on).
   - Sensing **my own MV**: improved ANTIWINDUP; actuator fault detection...
4. Extra manipulated variables
5. Safety loops

2. Control structure, 2.4 Extra MV

1. BASE setup: controlled vbles. = measured vbles; number = manipulated vbles.
2. Missing sensors
3. Extra sensors
4. **Extra manipulated variables** (MV)
   - Similar dynamics: **MV fusion** via **load-balancing** (maybe with on-line cost optimization), **split range**
   - Timescale separation: **cascade** [extraMV config.]
5. Safety loops

2. Control structure, 2.3+2.4 Extra sensors and MV

1. BASE setup: controlled vbles. = measured vbles; number = manipulated vbles.
2. Missing sensors
3.
4. **Both Extra sensors & extra manipulated variables**
   - Set **secondary references** to follow for secondary CVs, if they can be optimally computed from primary setpoints and (known) disturbances for economic benefit (e.g., gradual control).
   - **Double-cascade** (extra act. + extra sens), increase slave CV setpoint if main MV deviates too much from operating point.
   - **Many HIERARCHICAL cascade loops:** this is *how things really work*...The President gives orders to Education minister, then minister to General Director, then Local government, then Rector, then Department head... then I decide to write this material.
   - Extreme case (idealized abstraction): state-space representations, **every** "state variable" in my system, be it primary CV or not, must go to "**zero**" (operating point).
5. Safety loops

2. Control structure, 2.5 Safety issues

1. BASE setup: controlled vbles. = measured vbles; number = manipulated vbles.
2. Missing sensors
3. Extra sensors
4. Extra manipulated variables
5. **Safety issues in control loops**
   - **Extra fault-detection/monitoring sensors**: not related to CV or cascade control.
   - **Override control** (second control loop with "emergency" setpoint that, if reached, overrides the command of MV).
   - **Redundant sensors**: Three-sensor voting
   - **Redundant actuators** (two valves in series in a gas pipeline; backup generator, . . .).
   - **Emergency stop procedures** must remain "under control".

2. Control structure: possible refinements (I)

Everything done before considers "linearization" around the operating point. However, if plant is significantly nonlinear, consider **NONLINEARITY compensation** elements:

- **Static compensation of actuator/sensor nonlinearity**: deadzone compensation, inverse functions, calibration/lookup tables, etc.
- "feedback linearisation" (decoupling comes as a bonus)... but that usually requires "full" state feedback unless we have only invertible input or output nonlinearities.
  - "**Ratio control**" is a particular, easy to understand, case of said non-linear control in chemical processes.
  - "**Computed-torque**" acceleration control or "**Jacobian-based**" kinematic control in robotics are also well-studied cases.

2. Control structure: possible refinements (II)

- Controlled variable may be the maximum/minimum/mean of a set of sensors. [the latter related to SVD decoupling, possibly].
- Cross-limiting override (air-to-fuel ratio), multiple override,
- Measuring "parameters" (mass, length. . . with a non-additive effect but, say, multiplicative, etc. over my decisions; if it were "additive" it would be called *feedforward*).
- Image-based sensors or neural-networks: mostly in robotics, not in the scope of this material devoted to "industrial processes".

We don't know it all: secret recipes

"Secret" recipes specific to a particular technological sector exist; things are done "in a special way" in:

- Maximum power tracking in solar/wind generation
- Combustion efficiency & contaminant control in engines, gas turbines, ...
- Power sharing among multiple sources (wind turbines, power plants)
- Biomedical engineering: artificial pancreas, pacemakers, prosthetics, ...
- Autopilots, UAVs
- mobile robotics, self-driving cars...

KNOW your process or **learn from people that do**.

• Which things must be controlled for maximum performance/benefit? Why? • Which measurements are key for said performance? Why? • Which is the most cost-effective measurement/actuation technology? • Which are the most dangerous fail modes? • Which control structures do current state-of-the-art designs use?

3. Controller tuning

1. Don't underestimate ON-OFF control options
2. Why bothering with tuning?
3. Experimental, model-free, tuning
4. Model-based tuning
5. Advanced features

3. Controller tuning; 3.1 ON-OFF

1. **Don't underestimate ON-OFF control options**
   - If not-too stringent performance requirements (and abrupt power increments do not interfere other plant subsystems, and there are no fatigue/durability issues),
   - ON/OFF strategies are **very simple** (we only need an hysteresis error amplitude band to configure), no model needed;
   - they **perform well** in approximately **first-order** plants (typical: level, temperature, . . .).
2. Why bothering with tuning?
3. Experimental, model-free, tuning
4. Model-based tuning
5. Advanced features

3. Controller tuning; 3.2 Don't tune them all

1. Don't underestimate ON-OFF control
2. **Why bothering with tuning?**
   - A power plant, chemical site, etc. might have "hundreds" of loops... Do you think it's worth tuning each of them to great accuracy with identification, engineer time, etc.?
   - A level control loop will work well if target settling time is 5 minutes, irrespective of the slave servo-valve taking 5 seconds or 25 to provide the flow required by the master controller, irrespective of achieving it with 1% accuracy of with 5% accuracy...
   - Only a "handful" of top-level loops might need careful tuning.
   - If not-so-important controllers are provided with a specific process in mind, maybe "factory default" might suffice.
3. Experimental, model-free, tuning
4. Model-based tuning
5. Advanced features

3. Controller tuning; 3.3 Model-free.

1. Don't underestimate ON-OFF control
2. Why bothering with tuning?
3. **Experimental, model-free, tuning**
    - Use auto-tuning from the PID manufacturer when installing: press a button and forget (hopefully... algorithms are not that powerful and may fail).
    - By hand: [assuming STABLE plant, dominating 1st or 2nr order]
        - slow, sluggish response → increase P;
        - steady-state error remains (or it takes too long to be corrected) → increase I;
        - overshoot → decrease P, increase D... sometimes (low-frequency dominant pole) maybe decrease I.
        - MV too noisy → decrease D, increase noise filter in P or D actions (specially the latter one)... maybe decrease P.
    - Use some auto-tuning from literature, with "specially-crafted experiments" and computing PID gains from the results.
4. Model-based tuning
5. Advanced features

3. Controller tuning; 3.4 Model-based.

1. Don't underestimate ON-OFF control
2. Why bothering with tuning?
3. Experimental, model-free, tuning
4. **Model-based tuning.**
    - By tuning tables (using first order plus delay, integrator plus delay, etc. approximate models).
    - Option to use IMC structures (stable plants, easy tuning) if your PLC/SCADA implements it. *Not so good for disturbance rejection.
    - Root locus, optimal control, . . . predictive control (optimal least-squares control with saturation)

    Obtaining models is costly; a complex system may have MANY control loops (tens or hundreds of them)... if my control structure is very good (isolating independent subsystems, cascade, powerful/fast actuators, precise sensors, etc.) maybe my process is "easy to control" so I can spare a significant portion of "modelling" and "experimental identification" costs...
5. Advanced features

3. Controller tuning; 3.5 Advanced features.

1. Don't underestimate ON-OFF control
2. Why bothering with tuning?
3. Experimental, model-free, tuning
4. Model-based tuning
5. **Advanced features**
   - Consider the option of "two degree of freedom" if optimal setpoint and disturbance response are pursued.
   - Consider "detuning" sacrificing performance for robustness if significant plant variations (in delay or time constants) are expected in the controller lifetime.
   - "Tracking mode" integral antiwindup needed with decoupling/feedforward/feedback-linearization structures.
   - Gain scheduling (controller gains depending on operating point, pressure, . . .) if nonlinear/time-varying.

4. Implementation technology issues

- Hand-made C/C++ code isn't usually an option. What if you change job?
- Matlab, etc., licenses are expensive. Use tools available in your company.
- Integrate your PIDs in the PLC/SCADA framework; PIDs may either be **standalone** elements to communicate with sensors, actuators and other network elements or be **software-based** in PLCs or control computers.
- Setpoint changes may be frequent or may be just associated to start/stop procedures. Performance and safety of planned start/stop phases and emergency stop must be assessed.
- Actual controller implementation may need manual/auto/override bumpless transfer for isolated testing, safety-related events, start/stop procedures, etc.
- Supervise setpoints: disallow unsafe, out-of-limits values or too-fast changes.
- Log operational data, log operator commands and configuration changes to your control systems. Log alarms and override events, of course.

5. Continuous improvement: post-hoc data analytics

- Check, *a posteriori*, that performance/efficiency goals were achieved. Costly tests: suitable subsampling for quality control.
- Analyse logs to determine "abnormal events". They may be "bad" (faults: let's try that does not repeat) or "**good**" (we achieved surprisingly good results, let's try to repeat).
- Analyse the ' variability" (variance) of the logged historical records. Identify the "root causes" of variability to asses viability of resolving them. The guaranteed quality of my product (sale price) depends on the mean and variance of the CV (confidence int. 99%), and not on "settling time" or "poles" (not directly).
- Identify trends in demand, abnormal failure rates... These trends may be used in production planing and maintenance scheduling.
- Analysis can be manually carried out "by looking at spreadsheet tables and charts", or with specialised "data mining"/statistics software.

Conclusions

- Success of "**plant-wide control**" to **efficiently operate** a complex industrial plant is much more than "pole placement" or "root locus".
- A complex system working in good order requires a lot of steps, from process design, control tuning, to choice of communications/control software providers and suitable data analysis and quality control workflows.
- Many of them are "control-related"... I tried to summarise them here.
- The **building blocks** you learn in undergraduate courses are important (transfer function, Bode, stability, root locus, state-feedback, observer, ...), but they are just part of something larger... a Nobel price in hydrocarbon chemistry doesn't qualify you to lead a F1 racing team.
- **You need to have in mind the "BIG PICTURE" when working as a "senior" process control engineer / team leader.**

*Most of the ideas in these slides are **not so useful** to Ph.D. candidates, robotics, aerospace profiles.