Adaptive Systems

Ezequiel Di Paolo
COGS

Control Engineering

Control

- The next few lectures will take a look at the field of autonomous robotics.
- Natural and artificial nervous systems are often thought of as *control systems* or *controllers*.
- We will start by taking a quick look at Control Engineering because this field has many obvious parallels, influences, applications and useful concepts with respect to autonomous robotics.

Control

- Control engineering grew out of applied mechanics and its development in the mid 20th century was tightly bound up with the Cybernetics movement. (Process regulation, target-seeking devices, etc.)
- Control engineering is the application of mathematical techniques to the design of algorithms and devices to control processes or pieces of machinery. It often uses some kind of *model* of the thing to be controlled.

Linear systems

- If a system can be modelled by a set of linear differential equations there are well understood techniques for getting exact analytical solutions, and so designing controllers so that the output of the system is the required one.

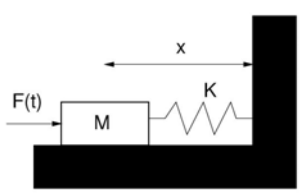
$$\sum_{i,k} a_{ki} D^k x_i(t) = \sum c_j x_j(t) + f(t)$$

$$D^k = \frac{d^k}{dt^k}$$

Example

- Forced spring

$$M\frac{d^2x}{dt^2} - Kx = F(t)$$



Real systems …

… are non-linear in almost every practical application.

- Example: a pendulum

$$L\frac{d^2\theta}{dt^2} + g\sin\theta = 0$$

- Much of control engineering theory and practice is concerned with linear systems. Control of non-linear systems is a very complex area.

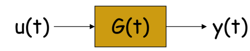
Linearization

- The most common trick: Approximate the nonlinear system by a linear system in the region of interest. Use Taylor series expansion and throw away all non-linear terms.
- For the pendulum, if $\theta \approx 0$ then $\sin(\theta) \approx \theta$ and so:

$$L\frac{d^2\theta}{dt^2} + g\theta \approx 0$$

Block diagrams

- Block diagrams are often used for system analysis and design of controllers as they facilitate the study of input/output relationships: *transfer functions*.

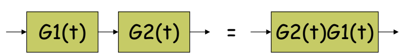


- u(t) = input
- y(t) = output
- G(t) = transfer function or operator:

  y(t) = G(t)[u(t)]

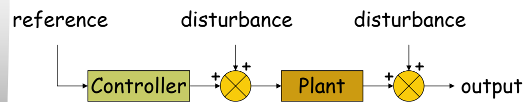
Block diagrams

- They have their own algebra;



- The transfer operator could be anything (a function, a derivative, an integration of the input signal, etc.) It is mathematically easier to use complex variables applying the *Laplace transform* which allows a simplified treatment. This is also known as "working in the domain of frequencies".

Feedforward control



Also known as *open-loop control*. A controller sends an input signal to the plant. It does not compensate for disturbances that occur after the control stage.

Feedforward control

Example: A simple toaster is an open-loop control system. Input: heat, controlled variable: colour of toasted bread. Others: stepper motors.

In feedforward control there is an implicit model (however simple) of the plant (e.g., some knowledge of the correlation between heating time and colour of bread). Open-loop is by no means a synonym for crude.

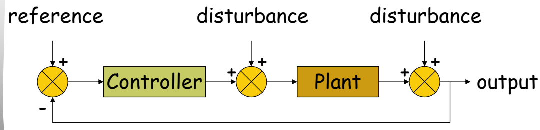
Feedback

- Feedback can be inherent in the system because of physical factors, or may be introduced deliberately to stabilize the dynamics

  Negative feedback tends to lead to stability
  Positive feedback tends to lead to instability.

- Simplest feedback controller: on/off control. For certain values of errors it gives a maximum input signal to the plant, and for other values, it gives a minimum signal (often no signal at all). Example: the thermostat.

Feedback control



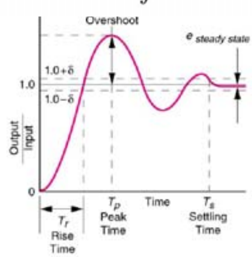
Closed-loop control. The output is sensed and compared with the reference. The resulting error signal is fed back to the controller.

PID: 99% of real world control

- Proportional control: input is proportional to error. For high gains or large errors, it will tend to overshoot and oscillate. There is always a steady-state error that cannot be corrected.
- Integral control: can get rid of steady-state error by integrating it over time
- Derivative control: can reduce settling time by giving a better dynamical response. Reduces overshoot by damping.

PID control

$$u = K_p e + K_i \int e dt + K_d \frac{de}{dt}$$



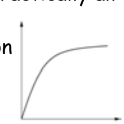
Non-linear control

- Linear controllers are generally valid only over small operational ranges.
- Hard non-linearities cannot even be approximated by linear systems.
- Model uncertainties can sometime be tolerated by the addition of non-linear terms.
- Non-linear systems often have multiple equilibrium points, plus periodic, or chaotic attractors. Small disturbances (even noise) can induce radically different behaviours.

Typical non-linearities

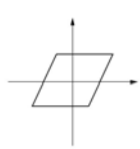
- On top of intrinsic system non-linearities some typical non-linearities can be found in practically all systems.

Saturation



Deadzone

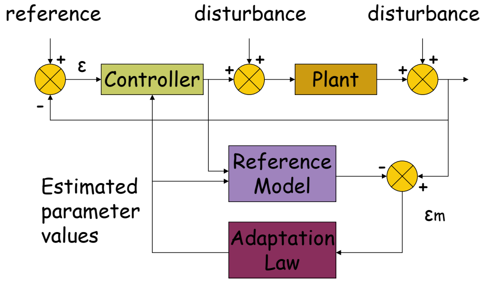
Backlash, hysteresis



Adaptive control

- A feedforward **reference model** of the plant is dynamically refined as the system runs. May be able to cope with various levels of uncertainty and adapt to different working regimes in non-linear systems.
- The model's output is used for comparison with actual output. The error is used to estimate modification in parameters following and **adaptation law**; new parameters are fed back both to the controller and the model.

Adaptive control



Neural network control

- Different kinds of neural networks can be used as feedforward reference models, as estimators, as predictors, or (less often) as controllers (both open- and closed-loop).
- "Linear mindset" persists, and there are very little applications using neural networks as non-linear controllers.
- A few examples using evolutionary methods, e.g., control of a simulated 2-D inverted pole (Gomez & Miikkulainen, 1998)

Fuzzy control

- Fuzzy controllers: Rule based controllers. Do not need to handle complex computations. Follows scheme: Fuzzification, Inference, Defuzzification. Can be more robust.
- Degrees of memberships to fuzzy classes determine the strength of the rules that apply.

Fuzzy control

- Eg. Inverted pendulum (rule set)

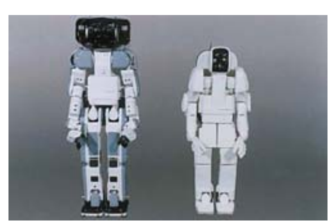
		Velocity						
		NL	NM	NS	Z	PS	PM	PL
	NL	PL	PL	PM	PM	PS	PS	Z
A	NM	PL	PM	PM	PS	PS	Z	NS
n	NS	PM	PM	PS	PS	Z	NS	NS
g	Z	PM	PS	PS	Z	NS	NS	NM
l	PS	PS	PS	Z	NS	NS	NM	NM
e	PM	PS	Z	NS	NS	NM	NM	NL
	PL	Z	NS	NS	NM	NM	NL	NL

Robotics

- A traditional method to control robot motion is to solve *inverse dynamics* of the robot, that is, to calculate the required torque to generate the desired trajectories
- Very computationally intensive as the number of degrees of freedom increases (sometimes even impossible). Also very brittle as many parameters (such as load, friction, etc.) vary with time.
- But that's often how it's done!

Robotics: controlled trajectories

- Amazing feats of engineering: Eg. Honda Robot

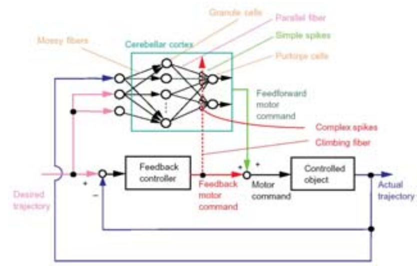


Questions…

- How relevant is control engineering for understanding natural adaptive behaviour? Depends on how far you take the metaphor of the brain as the controller of the body.

Usual view

- Sensory inputs and internal drives provide setpoints for the brain which controls the body. (Kawato on forward models in the cerebellum)



Less common view

- PCT (perceptual control theory, W.T. Powers). *Behaviour controls perception* by minimizing the error between some desired perceptual setpoint and actual perception. Setpoints are the outcome of other controlled processes that cascade back into the brain.

Questions…

- Isn't there a contradiction between thinking of an autonomous systems as controlled? Is it not autonomy the equivalent of self-law? Tim Smithers proposes autonomy means intrinsic control. Controller and plant cannot be separated, they are a single system. (why speak of control at all then?)
- What about mutual feedback between nervous system, rest of the organism, and environment? Can that be described using the language of control engineering?