Control

Madhu N. Belur

Control & Computing group,
Electrical Engineering Dept, IIT Bombay

15th Oct 2013

Outline

- Control: block diagram approach
- open-loop, feedback
- Causality, linearity
- Transfer function models: frequency domain approach
- State space models
- Poles, zeros
- Stability
- Feedback controller design: P, PD, PID

open-loop systems: causality

- Consider the system G
- Input u affects output y
- Input and output are functions of time t
- Output $y(t_0)$ depends on values of $u(t)$ only for $t \leqslant t_0$:

open-loop systems: causality

- Consider the system G
- Input u affects output y
- Input and output are functions of time t
- Output $y(t_0)$ depends on values of $u(t)$ only for $t \leqslant t_0$: causality
- Physical systems are causal: non-anticipating
- Output now cannot depend on future input

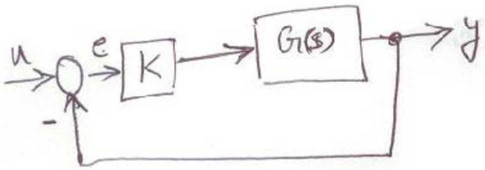
- Scaling of input u results in output scaled by same amount
- Input trajectory u suppose gives output y, then $2u$ gives $2y$
- If u_1 and u_2 give outputs y_1 and y_2 then

Linear systems

- Scaling of input u results in output scaled by same amount
- Input trajectory u suppose gives output y, then $2u$ gives $2y$
- If u_1 and u_2 give outputs y_1 and y_2 then $u_1 + u_2$ gives output $y_1 + y_2$
- Systems in nature are linear at least for small deviations
- (Loosely speaking) 'nonlinearities are like dominant second-order effects'
- When first order effect is zero, then nonlinearity cannot be ignored
- An intuition gathered from (simplified) linear model helps analyzing nonlinear systems

Why control?

- We want tracking: output should 'track' a given profile, give input suitably
- Regulation: temperature regulation, market/exchange rate regulation
- Policy $\equiv$ control
- Automatic control: input is given as some law based on output value
- For temperature control: heat/cold fluid input depends on thermostat reading



- Concern of 'over-correction': can cause instability
- Time-delays in the system: input's influence on output visible only after some time
- Physical systems are governed by differential equations

Differential equation, transfer function

Consider function $u(t)$ and $y(t)$, and suppose

$$u = 7y \qquad \text{and} \qquad u + \frac{d}{dt}u = 7y - \frac{d}{dt}y$$

- In one case, $y = \frac{1}{7}u$. In other case?

Differential equation, transfer function

Consider function $u(t)$ and $y(t)$, and suppose

$$u = 7y \qquad \text{and} \qquad u + \frac{d}{dt}u = 7y - \frac{d}{dt}y$$

- In one case, $y = \frac{1}{7}u$. In other case? 'dynamic' relation
- In fact, suppose input u is zero. Then $\frac{d}{dt}y = 7y$: unstable
- If initial value of $y(0)$ is 3, then $y(t) = 3e^{7t}$
- We take 'Laplace transform' of above differential equation and

Differential equation, transfer function

Consider function $u(t)$ and $y(t)$, and suppose

$$u = 7y \qquad \text{and} \qquad u + \frac{d}{dt}u = 7y - \frac{d}{dt}y$$

- In one case, $y = \frac{1}{7}u$. In other case? 'dynamic' relation
- In fact, suppose input u is zero. Then $\frac{d}{dt}y = 7y$: unstable
- If initial value of $y(0)$ is 3, then $y(t) = 3e^{7t}$
- We take 'Laplace transform' of above differential equation and
- $u(t) \to U(s)$ and $y(t) \to Y(s)$ gives, $\frac{d}{dt} \to s$
- $\frac{Y(s)}{U(s)} = \frac{s+1}{7-s}$

For the system with input u and output y, the 'transfer function' is $\frac{s+1}{7-s}$.

Transfer function

Only for linear, time-invariant differential systems:

- Time-invariant: the input and output variables can depend on time, but

Transfer function

Only for linear, time-invariant differential systems:

- Time-invariant: the input and output variables can depend on time, but
  the laws relating input and output (and their derivatives) do not depend on time explicitly
- Same experiment is repeated/carried-out tomorrow, same results
- 'Differential': systems governed by differential equations: $G(s)$
- Transfer function $G(z)$: discrete time systems: difference equations

Today: transfer function $G(s)$: continuous time LTI systems
(Convention: $G(s)$: continuous time, $s \to \frac{d}{dt}$
$G(z)$: discrete time, $z \to$ 'forward shift map': non-causal

Why automatic control?

Automatic-control ≡ feedback control

- With open-loop: even if system is 'unstable', can control output y by suitably choosing input u provided
  - System transfer function is known accurately, and
  - 'Initial condition' (meaning y and some number of derivatives) are known precisely
- If not known precisely, then cannot guarantee good control of output y
- In practice, parameters are not known precisely: we want 'robustness'

Robust design ≡ We design for 'nominal' values,
but same design works for 'nearby' other values also

Open-loop control

Predecide what input $u^\star$ will achieve required output $y^\star$.

- No need to 'sense' actual output y (no sensors required)
- Not reasonable in practice. Traffic light timings require re-adjustment
- Not possible if system is unstable
- Often sub-optimal

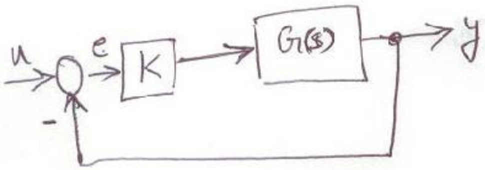
Stability

Various (related) notions:

- Output y goes to zero, when input u goes to zero
- Output y remains bounded, when input u is bounded
- Output y is bounded, when input is 'identically' zero (for any initial condition)
- Output y goes to zero, when input is identically zero (for any initial condition)

Stability conclusion can change depending on input/output classification of variables

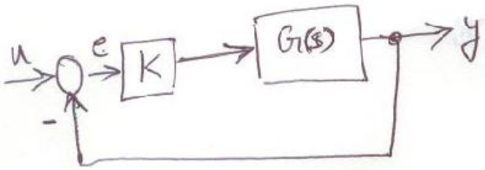
Feedback control



Examples:

- Automatic temperature regulation (in spite of disturbances: external heat)
- Servo-motor control: remove off-set
- Traffic light control, based on actual vehicle flow data (rather than preset-timings)
- More generally, queue management: internet routers
- Market-regulation (inflation, exchange rate), using bank's 'cash-reserve-ratio', interest-rate
- Satellite control, robotics

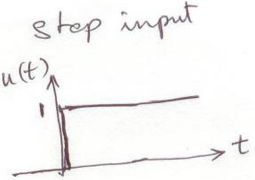
Why feedback control?

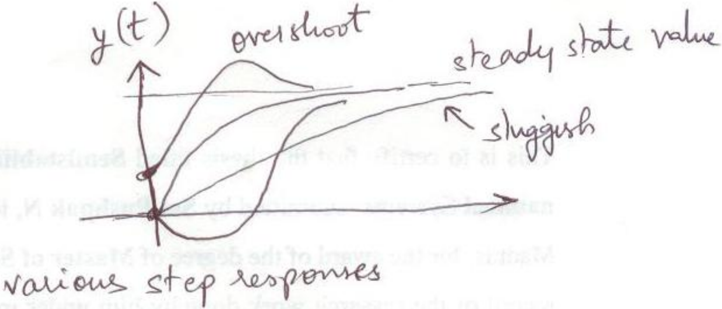


Feedback control is required for

- To achieve stability: closed-loop-stability
- To make y track u even better
- Have less steady state error ($\lim_{t\to\infty} y(t) - u(t)$)
- Faster transients: output y 'tracks' u fast
- Optimization: least energy spent in input u, least 'total' deviation of output y

Step input and step response





Desired step-response

- Stability: output settles to some value
- Ideally, output value = input value (at steady state, at least)
- Output responds 'quickly' to input (not 'sluggishly')

Basic principle:

- Compare output value with desired value =: error $e(t)$
- Feed error back: large-error : more corrective signal $c(t)$
- Make corrective signal 'proportional' to error: P-controller $c(t) = ke(t)$: design k
- $c = k_P e + k_D \frac{d}{dt} e$: PD-controller, good for quickening
- $c = k_P e + k_D \frac{d}{dt} e + k_I \int e$: PID-controller ($k_I$ for making steady state error = 0
- All this: at this point: thumb-rules: can cause instability
- Top priority: stability of the closed loop

High gain feedback

- To make output-error zero ‘quickly’: use high gain
- High-gain can result in instability: especially for delayed systems
- Designing k_P, k_D and k_I values requires intuition and transfer function knowledge

Transfer function $G(s)$

SISO $\equiv$ Single input, single output
$G(s) = \frac{n(s)}{d(s)} = \frac{s+1}{7-s}$

- Roots of denominator $d(s) \equiv$ poles of the system/transfer function
- Roots of numerator $n(s) \equiv$ zeros of the system
- DC gain: put $s = 0$, and evaluate G
- Poles, zeros and DC gain give transfer function (SISO)

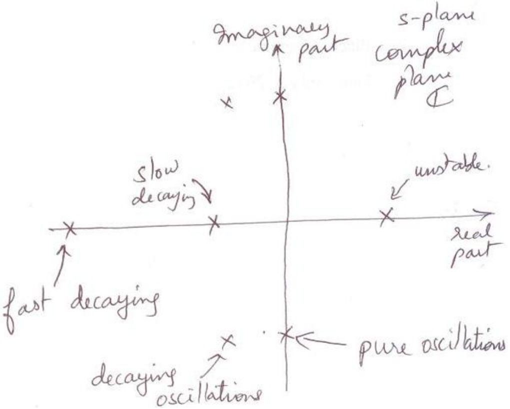
Stability

- For stability, poles should be on the left-half-complex plane
- All roots of the denominator should have real part negative
- Real-part zero: ‘marginal’ stability
- Real part of some pole: instability

Transfer function

- Usually ‘proper’ transfer function: numerator degree $\leqslant$ denominator degree
- More poles than zeros
- ‘Smoothening’ : ‘relative degree’ : den-degree $-$ num-degree
- If input is discontinuous, output gets ‘smoothened’ if relative degree is high
- $G(s) = \frac{1}{2}$: an integrator: output is the integral of the input
- $G(s) = s$: differentiator: not non-causal (for continuous time): tachometer
  (Misconception is that improper transfer functions are non-causal)

Complex plane



Transfer function

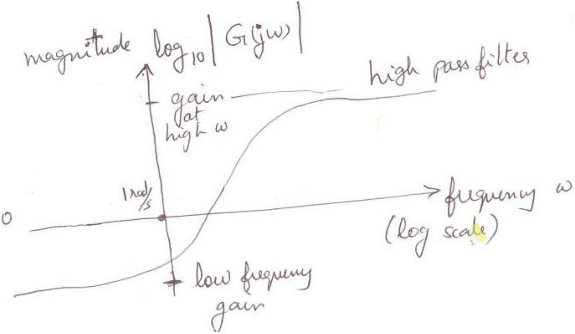
$G(s)$ indicates 'amount of pure scaling' for exponential inputs

- If input $u(t) = e^{-2t}$, then output $y(t) = G(-2)e^{-2t}$
- If input $u(t) = \sin 3t$, then output $y(t) = |G(3j)| \sin(3t + \angle(G(3j)))$
- Output has phase-lag (or lead) for sinusoidal inputs: lag $= \angle G(3j)$
- Amplification of sinusoidal input $= |G(3j)|$

'Frequency domain analysis'

Bode plot

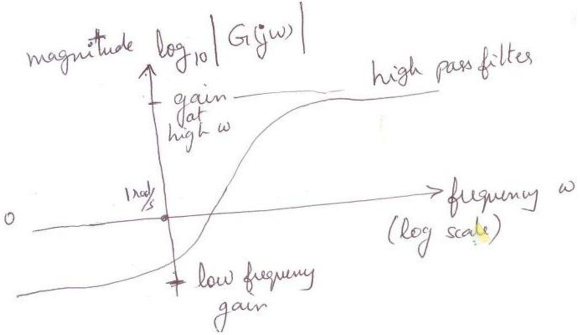
Magnitude plot: plot $|G(j\omega)|$ versus frequency ω: both in log-scales



Similarly phase-plot

Bode plot

Magnitude plot: plot $|G(j\omega)|$ versus frequency ω: both in log-scales



Similarly phase-plot
Frequency domain analysis

Instability

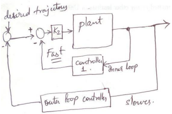
- If open-loop is unstable, then closed loop can be made stable using feedback
- Sometimes using just P-controller: constant gain feedback
- Sometimes, derivatives and integrals of output: dynamic controller

Further control objectives: optimal control: time-optimality, fuel-optimality, etc.

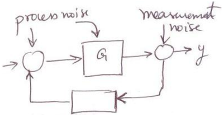
Inner and outer loop

In many applications: a fast inner-loop and a slow outer-loop is designed separately

For example: optimal trajectory tracking: satellite launch vehicle trajectory to space



Trade-offs



Controller implementation

- Controllers are of-late implemented digitally. (Earlier analog controllers)
- Controllers are designed using more sophisticated packages: Scilab, Matlab
- Plants (systems to be controlled) often MIMO: intuition less helpful
- Packages come with their limitations (properness, etc.)
- Real time adaptability
- Computational intensity: distributed control

Trade-offs: theoretical and practical

- Noise added at separate points in the loop: cannot achieve disturbance attenuation due to both noises
- Output regulation versus input energy usage
- Time-optimality versus input energy usage
- Accurate system parameter knowledge versus robustness
- Controller being robust and controller having to be implemented accurately