Lecture 5, Sep 21, 2023

Introduction to Control Theory

Stability

- Consider the first-order linear time-invariant system $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$
  - We diagonalize the system so that $\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P} = \boldsymbol{\Lambda}$
  - Then we express $\boldsymbol{x}$ as a linear combination of eigenvectors: $\boldsymbol{x}(t) = \sum_{\alpha=1}^{n} \eta_\alpha(t)\boldsymbol{p}_\alpha$
    - η are the coordinates
  - Substituting it back into the equation of motion, we get $\dot{\eta}_\alpha = \lambda_\alpha \eta$ for $\alpha = 1, \dots, n$
  - Therefore we can solve it as $\eta_\alpha(t) = \eta_\alpha(0)e^{\lambda_\alpha t}$
- For this system, we know $\boldsymbol{x} = \boldsymbol{0}$ is a solution; when talking about stability, we consider the long-term behaviour of the differential equation and see if it goes back to 0
  - The $\eta_\alpha(t)$ are disturbances to the system, so we want them to be eliminated eventually
- If $\mathrm{Re}(\lambda_\alpha) < 0$ then as $t \to \infty$, we have all $\eta_\alpha \to 0 \implies \boldsymbol{x} \to \boldsymbol{0}$

Definition

A linear system $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$ is *stable* if $\mathrm{Re}(\lambda_\alpha) \leq 0$ for all α; it is *asymptotically stable* if $\mathrm{Re}(\lambda_\alpha) < 0$. This works even for nondiagonalizable matrices by considering their Jordan forms.

For nonlinear systems $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$, we can consider local stability in the neighbourhood of a solution by linearizing the system using the Jacobian,

$$\boldsymbol{A} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^T} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

PID Control

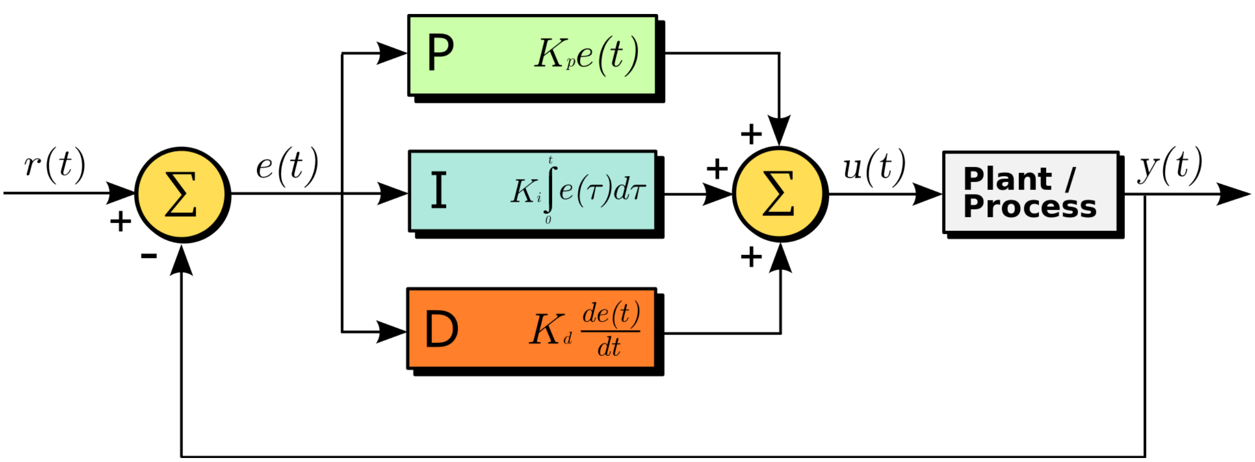

Figure 1: Diagram of PID control.

- PID control can be used to address two types of problems: the regulator problem (eliminating disturbances to the system) and the servo or tracking problem (tracking the output to a trajectory)

- In control theory the thing being controlled is referred to as the *plant*, a combination of actuators and processes
- Consider a simple single-variable, first-order linear system $\dot{x} + \sigma x = u, x(0) = x_0$ where x is the state variable and u is the control variable; we want to track the system to x_d
  - The eigenvalue for this system is $-\sigma$ (see this by $\dot{x} = -\sigma x$), so if $\sigma < 0$, this system is unstable
  - Define the error $e = x - x_d$ and let $u = -k_p e(t)$, so $\dot{x} + (\sigma + k_p)x = x_d$
  - For this system, $x_h = e^{-(\sigma+k_p)t}, x_p = \dfrac{k_p x_d}{\sigma + k_p}$ so the solution is $\dfrac{k_p x_d}{\sigma + k_p} + \left(x_0 - \dfrac{k_p x_d}{\sigma + k_p}\right) e^{-(\sigma+k_p)t}$
    * Therefore even if $\sigma < 0$, as long as we choose a sufficiently high k_p, the system can be stable
    * However if we let $t \to \infty$ we have $x = \dfrac{k_p x_d}{\sigma + k_p} \neq x_d$, so we have a steady-state error
  - Let's add an integral term: $u = -k_p e - k_i \int e(\tau)\,\mathrm{d}\tau$
    * Substituting in u and differentiating, we have $\ddot{x} + (\sigma + k_p)x + k_i x = k_i x_d$
    * The general solution is $x_h = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t}$
    * The particular solution is just $x = x_d$
    * The complete solution is $x(t) = x_d + c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t}$
      - $\lambda = \dfrac{-(\sigma + k_p) \pm \sqrt{(\sigma + k_p)^2 - k_i}}{2}$
      - If $\mathrm{Re}(\lambda_i) < 0$, then as $t \to \infty$, $x(t) \to x_d$ and we have no steady-state error
      - Now λ might have an imaginary component, so our system may have oscillations; it could be underdamped, overdamped or critically damped depending on the gains
    * This system is stable if $k_i > 0, k_p + \sigma > 0$
- More generally, our state equation can be $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$
  - Our feedback is $\boldsymbol{u} = -\boldsymbol{F}\boldsymbol{x}$ where $\boldsymbol{F}$ is the gain matrix
  - This gives $\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{F})\boldsymbol{x}$
  - We can now make this system stable by finding an $\boldsymbol{F}$ that modifies the eigenvalues of $\boldsymbol{A}$
    * Whether we can always find such an $\boldsymbol{F}$ is related to the controllability of the system
- Even more generally, for nonlinear systems $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$, we can choose to linearize locally as before, or we can try heuristic feedback, with either linear or nonlinear control
- Now consider a second-order plant $\ddot{x} + \sigma\dot{x} + \eta x = u$
  - We will also add a derivative term: $u(t) = -k_p e(t) - k_d \dot{e}(t) - k_i \int e(\tau)\,\mathrm{d}\tau$
  - Substituting this and differentiating, we will get a third order differential equation
  - This gives us a new set of stability requirements

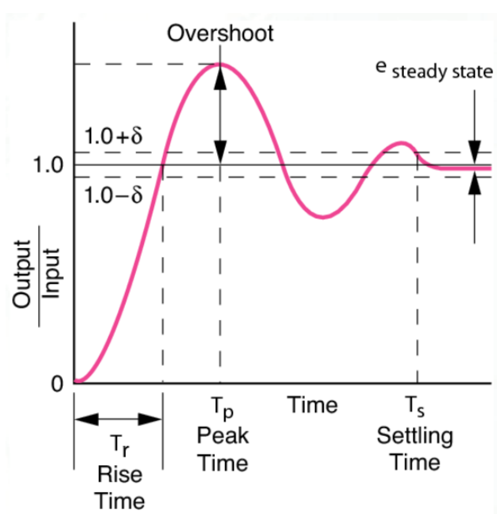

Figure 2: Example response of a PID controller.

- Response characteristics:
  - Rise time: the amount of time for the output to approach the input
    * There is no set convention on this; often it's defined as the time from 0 to 100% of the desired output, sometimes it's 10% to 90%
  - Overshoot: the amount over the desired output that the maximum value is
  - Settling time: time to reach and stay within a certain band δ of the desired output
  - Steady-state error: remaining error as $t \to \infty$
- The gains change the characteristics of the response; depending on the system, different characteristics may be desired

Gain	Rise Time	Overshoot	Settling Time	Steady Error	Stability
k_p	Decrease	Increase	Little effect	Decrease	No effect
k_i	Decrease	Increase	Increase	Eliminate	Degrade
k_d	Little effect	Decrease	Decrease	No effect	Improve

Figure 3: Effect of increasing different gains on a PID controller.

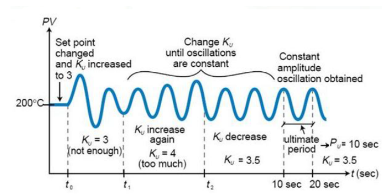

Figure 4: Ziegler-Nichols tuning sequence.

- The Ziegler-Nichols method is one among many methods to tune PID gains:
  1. Suppress the integral and derivative terms completely
  2. Create a small disturbance by suddenly changing the setpoint
  3. Increase k_p until the system is oscillating with constant amplitude
  4. Record the gain value as k_u, the oscillation period T_u, and refer to the table to set k_p, k_i, k_d

Type of Control	k_p	k_i	k_d
P	$0.50k_u$	---	---
PI	$0.45k_u$	$1.2k_u/T_u$	---
PD	$0.80k_u$	---	$0.125k_uT_u$
Classic PID	$0.60k_u$	$2.0k_u/T_u$	$0.125k_uT_u$

Figure 5: Ziegler-Nichols table of gains.

- PID control is prone to common problems:
  - Noise in the derivative: derivatives are typically numerically calculated and can be quite noisy
    * This can be mediated by attaching a low-pass filter on the signal to remove high-frequency components
  - Integral windup: error can build up in the integral term, making it overwhelm the other control terms
    * This can be mediated by removing the i term after the desired value is reached, capping the error integral, or reinitializing the i term
  - Deadband: the region where the control input does not affect the actuator (e.g. due to friction)
    * This can be mediated by commanding a minimum control input when in the deadband so the control is not useless

Applications in Robotics

- Consider robot with a bicycle model; we want to drive it to a desired goal point (x_d, y_d)
  - Proportional control: $v = -k_{p,v}\sqrt{(x-x_d)^2+(y-y_d)^2}, \theta_d = \tan\frac{y-y_d}{x-x_d}, \gamma = -k_{p,\gamma}(\theta-\theta_d)$
- What if we wanted to follow a line $ax+by+c=0$?
  - We can measure the crosstrack error by $\delta = \frac{ax+by+c}{\sqrt{a^2+b^2}}$ (normal distance to line)
  - Then $\gamma_\delta = -k_{p,\delta}\delta$ makes us steer the robot towards the line
  - But now we want to keep the robot on the line, so let $\theta_d = \tan^{-1}\left(-\frac{a}{b}\right)$ and $\gamma_\theta = -k_{p,\theta}(\theta-\theta_d)$ steers us towards the line
  - These two terms are combined, and a fixed speed is added for this simple proportional control
- What if we wanted to follow a path?
  - Let $e = \sqrt{(x-x_d)^2+(y-y_d)^2} - d$, and then apply PI control on the velocity using this error
    * In effect this follows a set point at a distance d ahead all the time
    * This is because without the $-d$, e will always be positive and so we will get integral wind-up, where the integral term overwhelms the control
  - The steering can be controlled using the same way as when moving to a goal point
- Consider a robot with a unicycle model; we want to move it to a pose (x_d, y_d, θ_d)
  - We will transform our variables (x, y, θ) to (ρ, α, β), where ρ is the distance to the setpoint, α is the angle from the line that connects directly to the target
    * $\rho = \sqrt{\Delta_x^2 + \Delta_y^2}$
    * $\alpha = \tan^{-1}\frac{\Delta_y}{\Delta x} - \theta$
    * $\beta = -\theta - \alpha$
  - We want to regulate $(\rho, \alpha, \beta) = (0, 0, 0)$
    * Apply proportional control on v with ρ, and ω with α and β