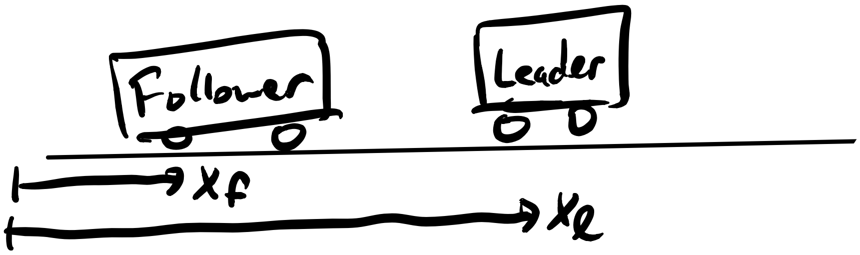
e.g. Convoy problem – Adaptive cruise control



- leader driven by a person
- follower driven by a computer which thru software can assign its speed

Objective: Write a program that decides on an appropriate speed in order to maintain a given inter-vehicle distance.

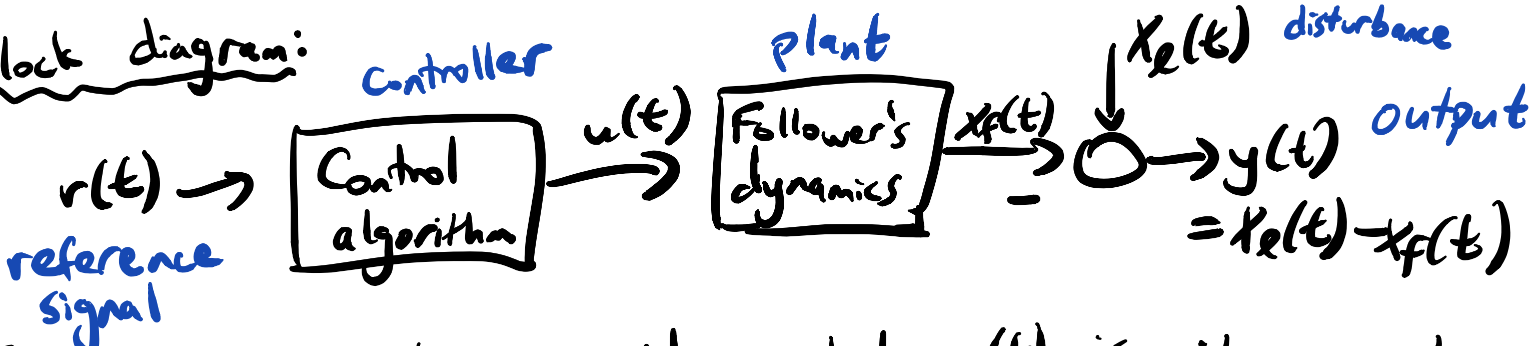
Assumptions: - cars only move in straight line on flat ground

- $\dfrac{dx_f(t)}{dt} = u(t)$, u is the follower's speed which can be assigned
- the leader's speed is unknown and beyond our control but they don't drive too wildly: $\dfrac{dx_\ell(t)}{dt} \approx$ constant

Option 1: Open-loop

- don't equip follower with any sensors – save \$
- program to decide speed only has access to a desired inter-vehicle distance $r(t)$

Block diagram:



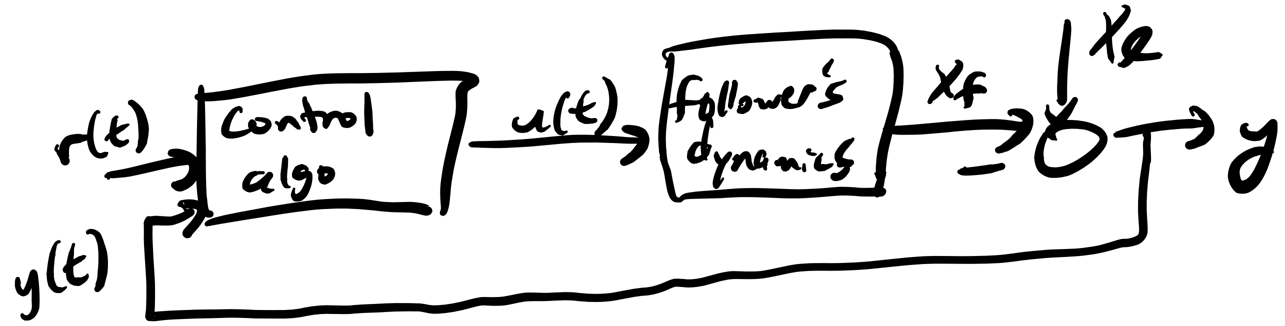
- Since program has no idea what $y(t)$ is, it cannot make a good decision.

$\Rightarrow$ Open-loop cannot work in this case.

- There are examples when it does work (e.g. toaster)

Option 2: Closed-loop

- equip follower w/ stereo camera or LIDAR (more $)
  so we can measure this intervehicle distance
  $y(t) = x_\ell(t) - x_f(t)$.
- our program now has access to both $r(t)$ and $y(t)$.



- simplest algorithm:

$$u(t) = \begin{cases} \bar{u} & \text{if } r(t) - y(t) < 0 \\ \underline{u} & \text{, else} \end{cases}$$

→ problems (too jerky, need to determine min/max constants, hard to determine good values since $\frac{dx_\ell}{dt}$ unknown)

- better choice: proportional error feedback

$$u(t) = -K_p(r(t) - y(t)), \quad K_p > 0 \text{ (gain)}$$

→ follower evolves according to

$$\frac{dx_f}{dt} = -K_p(r(t) - y(t)) \quad \text{(closed loop dynamics)}$$

- we'll learn that a better choice is proportional-integral error feedback

$$u(t) = -K_p\,(r(t) - y(t)) - K_i \int_0^t (r(\tau) - y(\tau))\,d\tau,$$

$K_p, K_i > 0$

▲

---

e.g. 1.4.1 (web server)

- Consider a web server that responds to GET and POST requests.
- Server maintains a buffer of pending requests so that requests arriving at a busy time aren't lost