EECE 5550
Mobile Robotics

Lecture 18: Introduction to Feedback Control

Derya Aksaray

Assistant Professor

Department of Electrical and Computer Engineering

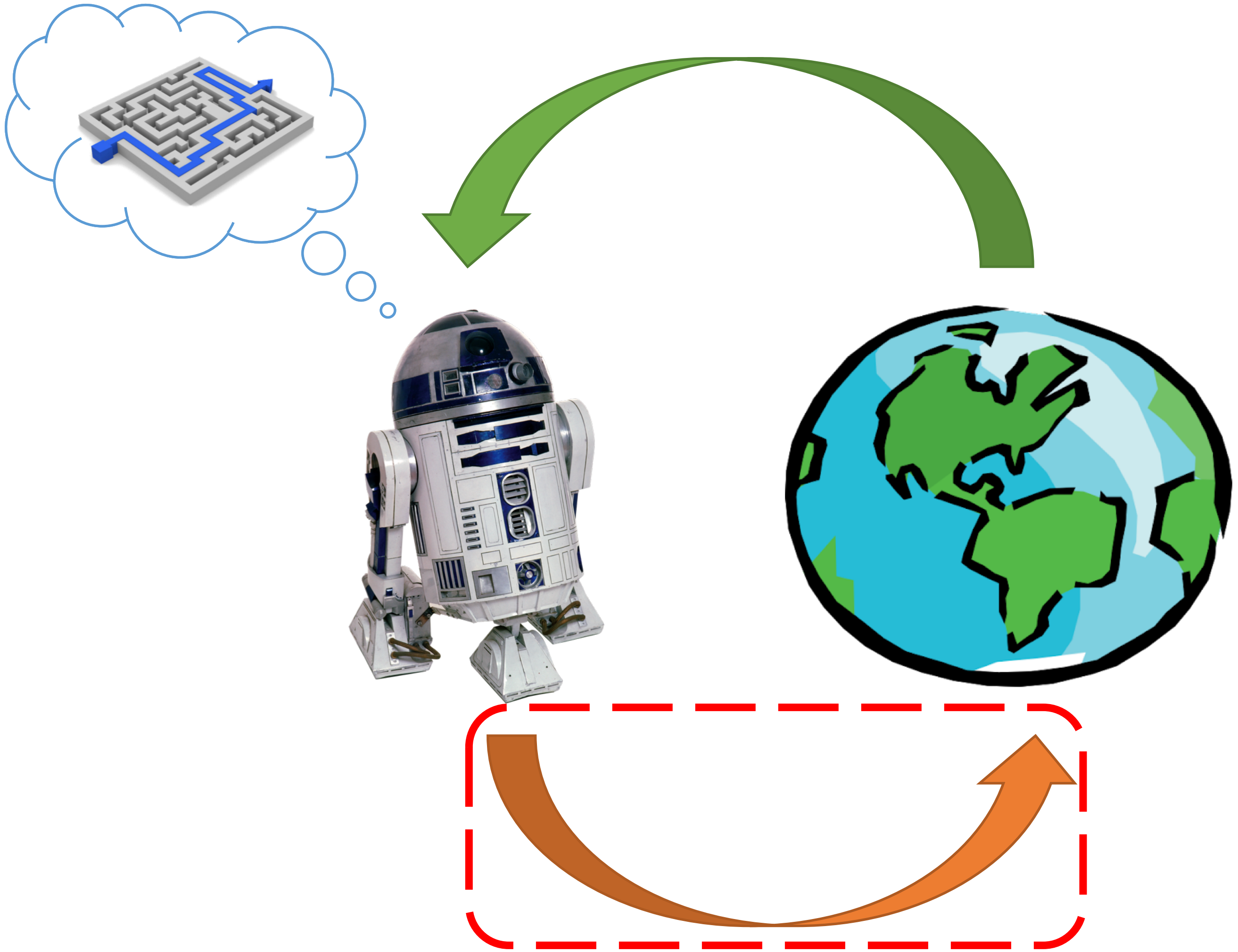
Northeastern University

Recap: The Central Dogma of Robotics

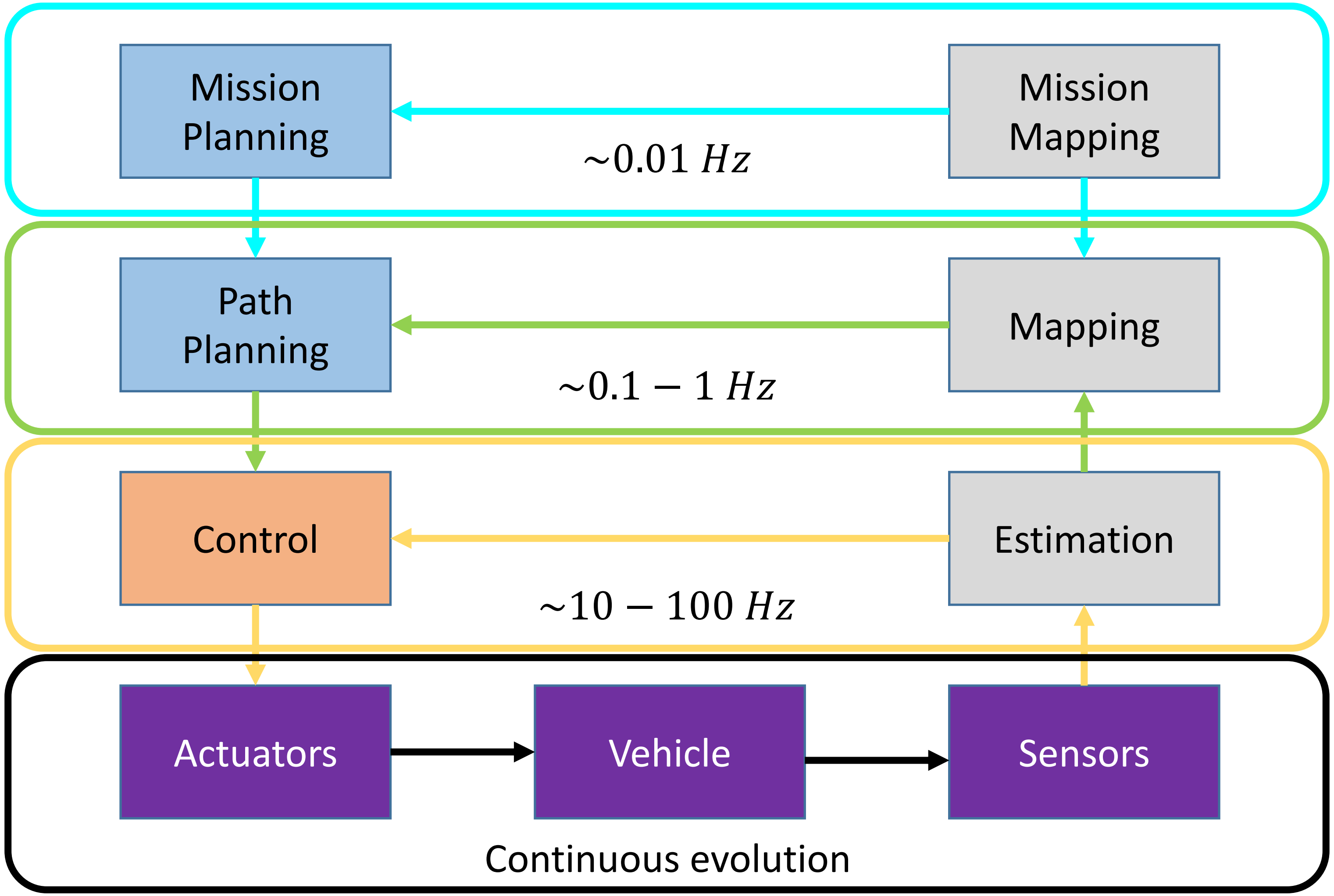
Sense → Think → Act

- Sense: Process sensor data to construct a model of the world
- Think: Construct a plan to move from the current state to the goal state
- Act: Control actuators to execute plan

Today

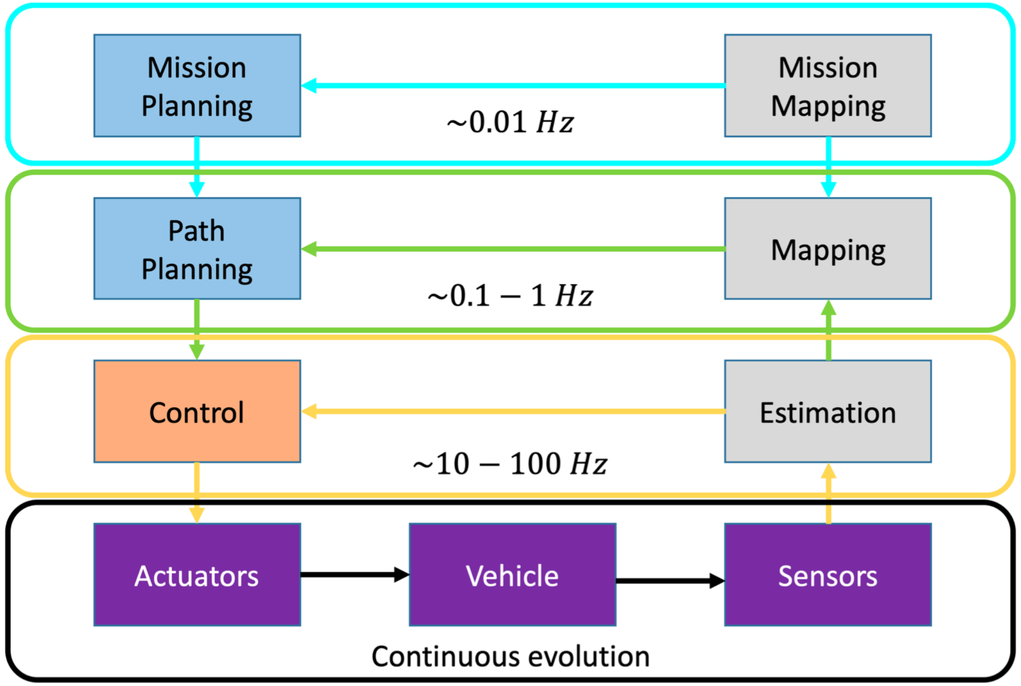


Time scale separation

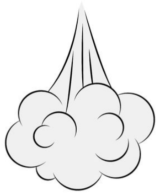


Time scale separation

- Separating planning and control
  - Simplified planning
  - Real-time
  - Use of old information
  - Abstractions should be consistent!



Control Theory



Wind gust

State: The current representation of the system

- Wheeled robot: x-y coordinate and heading angle

Dynamics: Description of how the state changes

Reference: What we want the system to do

- Regulation (maintaining a desired state)
- Path following (tracking a state trajectory)
- Trajectory tracking (tracking a state trajectory with explicit timing)

Hovering

Output: Measurement of (some aspects of the) system

- We may not always measure the state

Input: Control signal

- Mapping the reference to proper control inputs

Control Theory = how to pick the input signal?

- Stability: Lyapunov stability, BIBO stability, etc.
- Robustness
- Optimality

Open Loop Control

- Control inputs
  - Rotor power (adjusting RPM)
  - Duration of rotor operation (setting time)

- Max power for 5 sec.

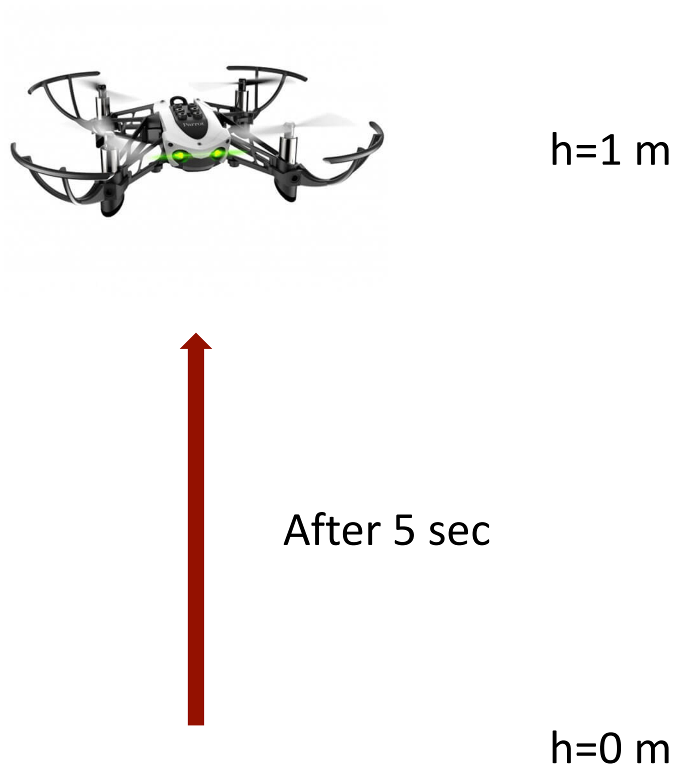
h=1 m

h=0 m

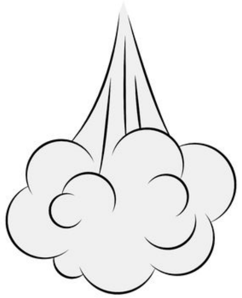
Open Loop Control

- Control inputs
  - Rotor power (adjusting RPM)
  - Duration of rotor operation (setting time)

- Max power for 5 sec.



Open Loop Control



- Control inputs
  - Rotor power (adjusting RPM)
  - Duration of rotor operation (setting time)

h=1 m

- Max power for 5 sec.

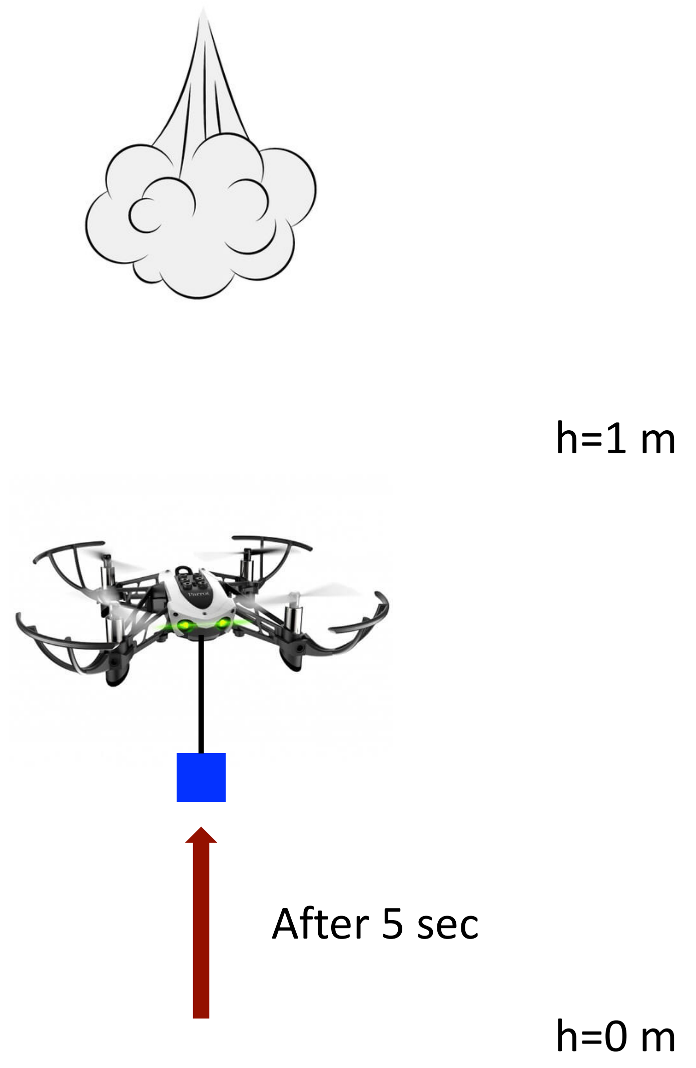
Some disturbance
(e.g., extra payload, gust)

h=0 m

Open Loop Control

- Control inputs
  - Rotor power (adjusting RPM)
  - Duration of rotor operation (setting time)

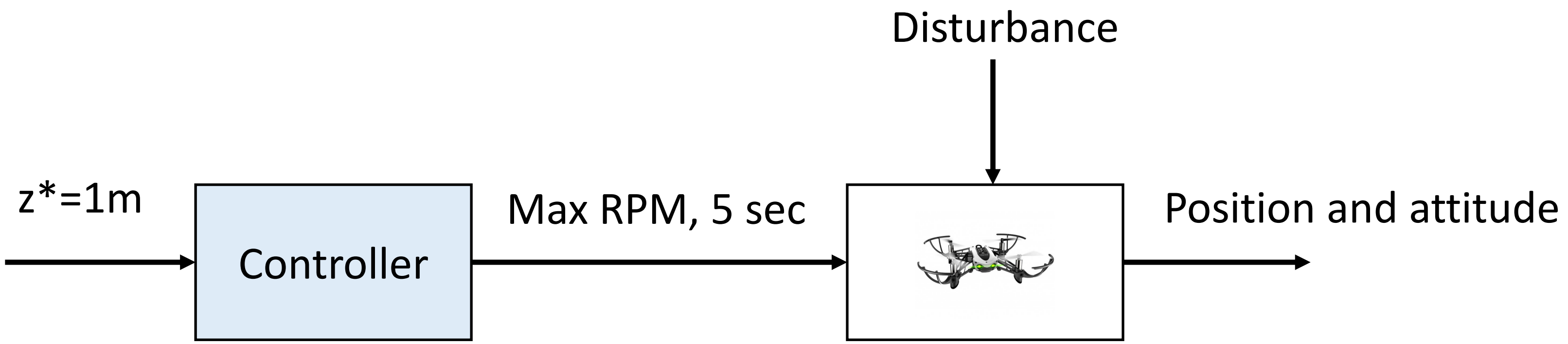
- Max power for 5 sec.



Open Loop Control

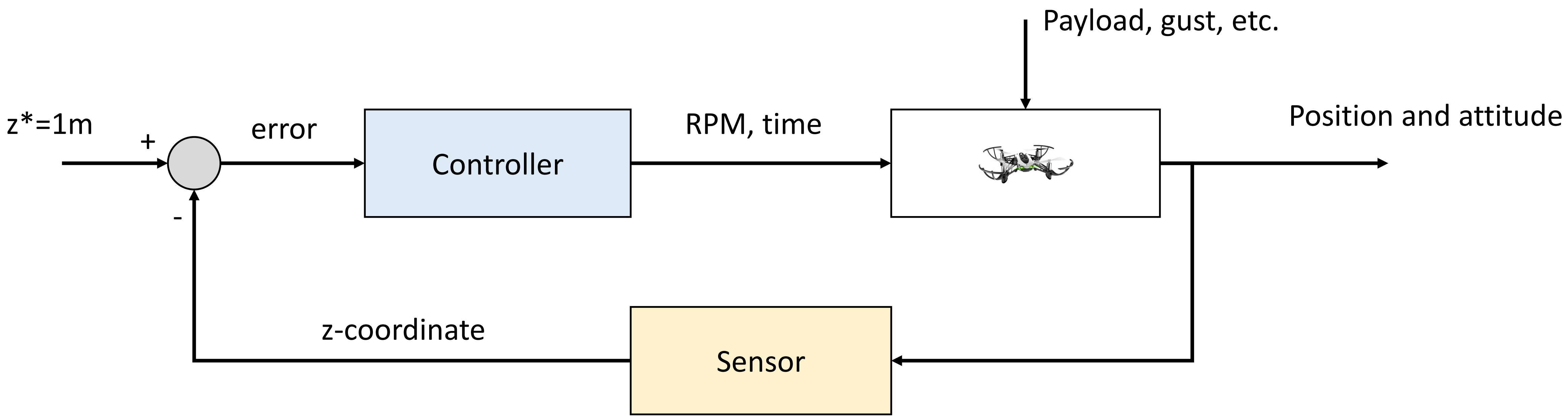
- Control inputs
  - Rotor power (adjusting RPM)
  - Duration of rotor operation (setting time)

- Max power for 5 sec.

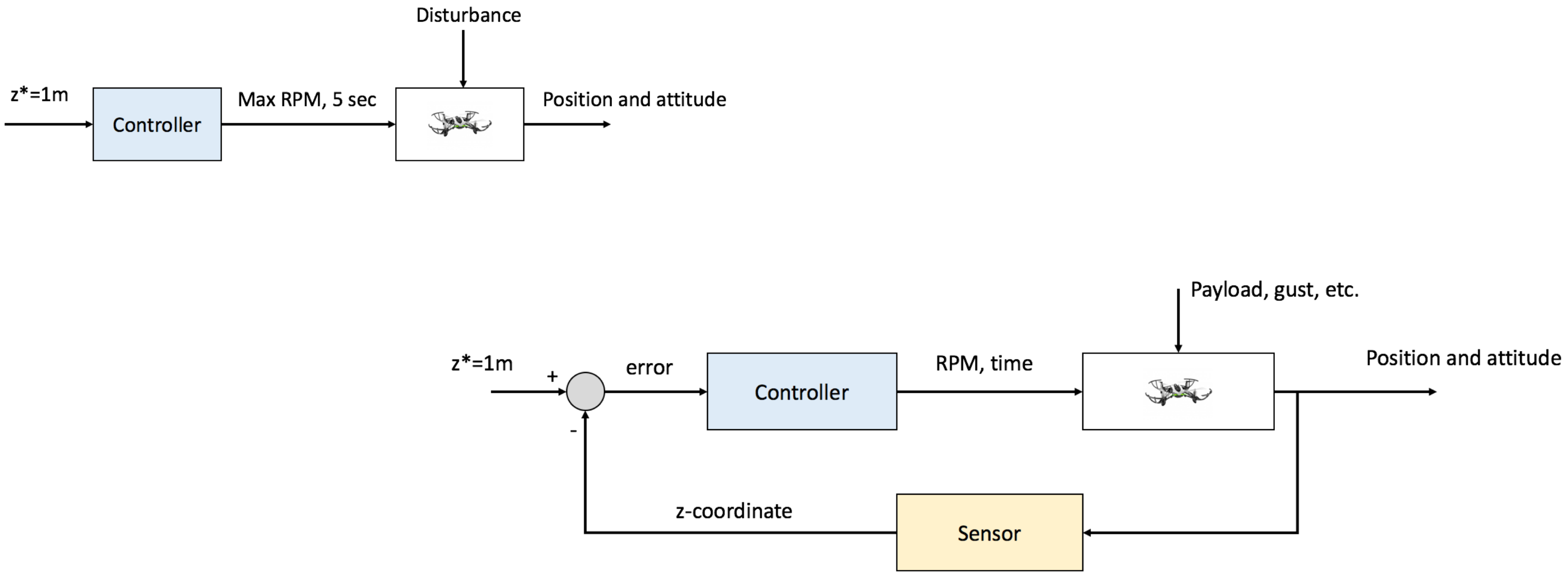


Closed Loop Control

- Use feedback to make the system response less sensitive to external disturbances and internal variations of the system parameters.

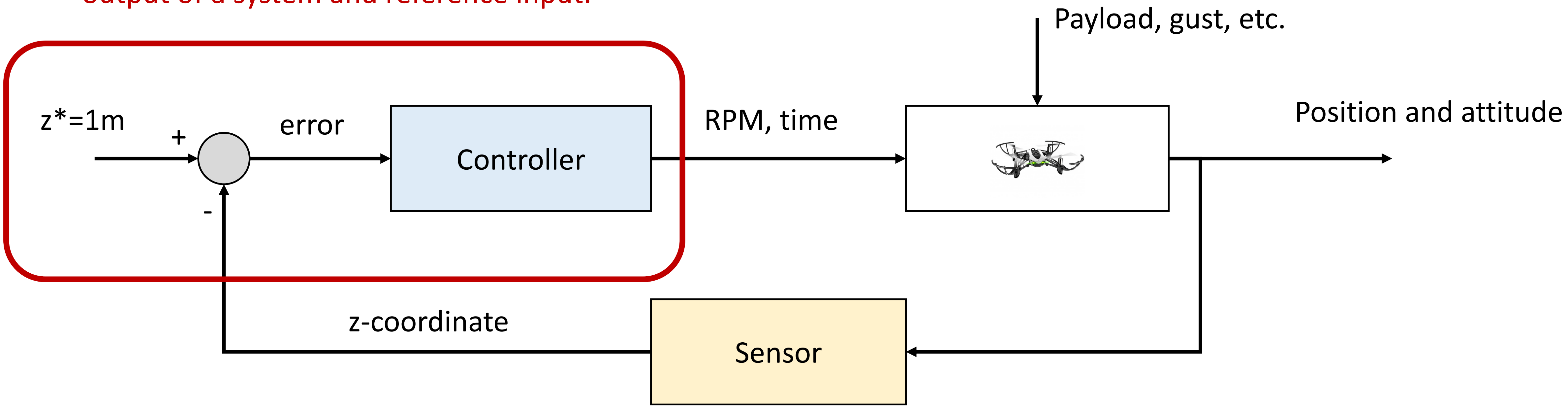


Open vs Closed Loop



Closed Loop Control = Feedback Control

Goal: Reduce the difference between the output of a system and reference input.



Open-loop Control

No state feedback!

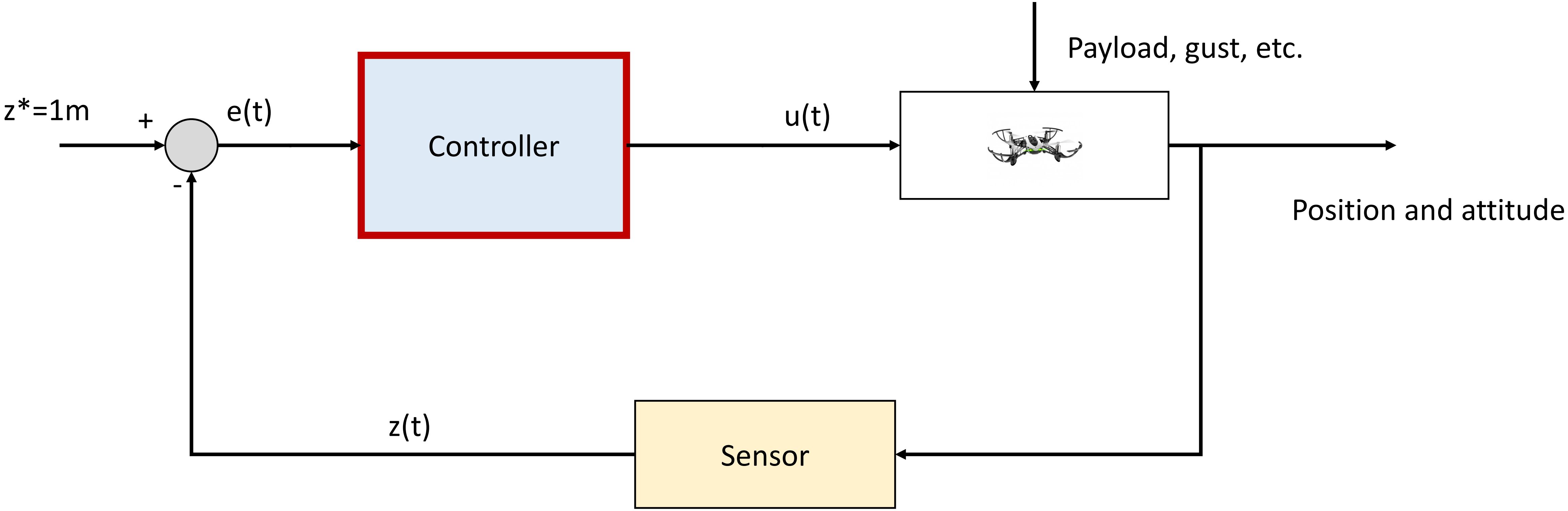
Feedback Control Examples

System	Control Inputs	Output	Sensors
Jet aircraft	Elevator, rudder, etc.	Altitude, heading	Altimeter, GPS

Feedback Control Examples

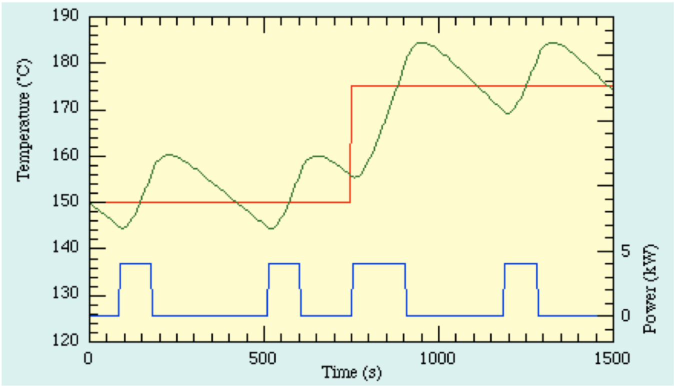
System	Control Inputs	Output	Sensors
Car (ACC)	Throttle	Velocity	Speedometer

How to design controller?



Bang-Bang Control

Example: Thermostats



- On-off controller
- u_{min} and u_{max}
  - $u_{min} = 0$

$$u(t) = \begin{cases} u_{min} & if\ e(t) < 0 \\ u_{max} & if\ e(t) > 0 \end{cases}$$

where $e(t) = z^* - z(t)$

What is desired controller performance?

Advantages

Simplest control architecture

Cost-effective

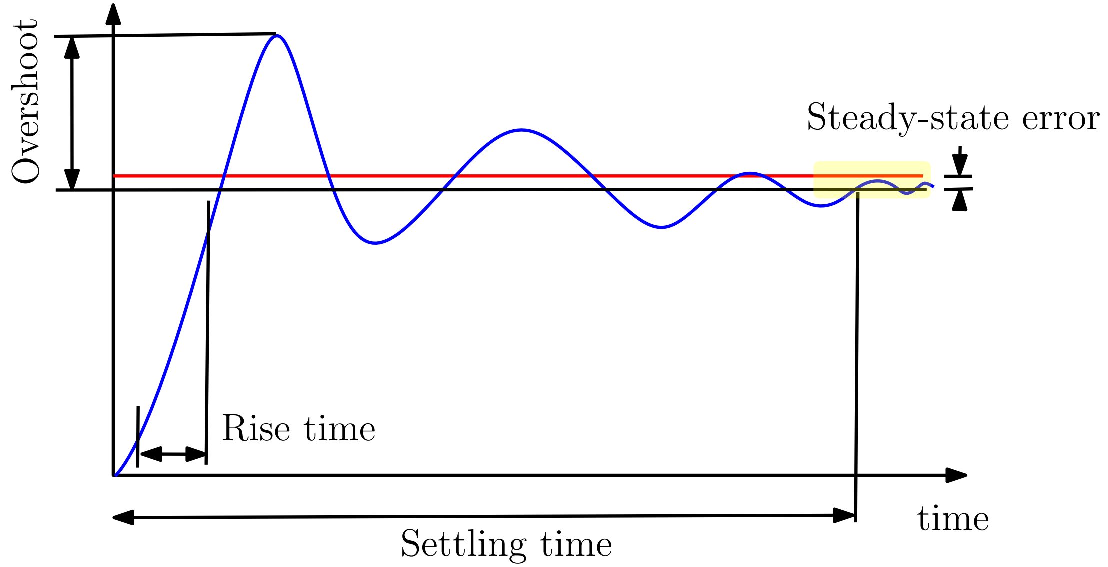
Disadvantages

Persistent oscillations

No tight performance

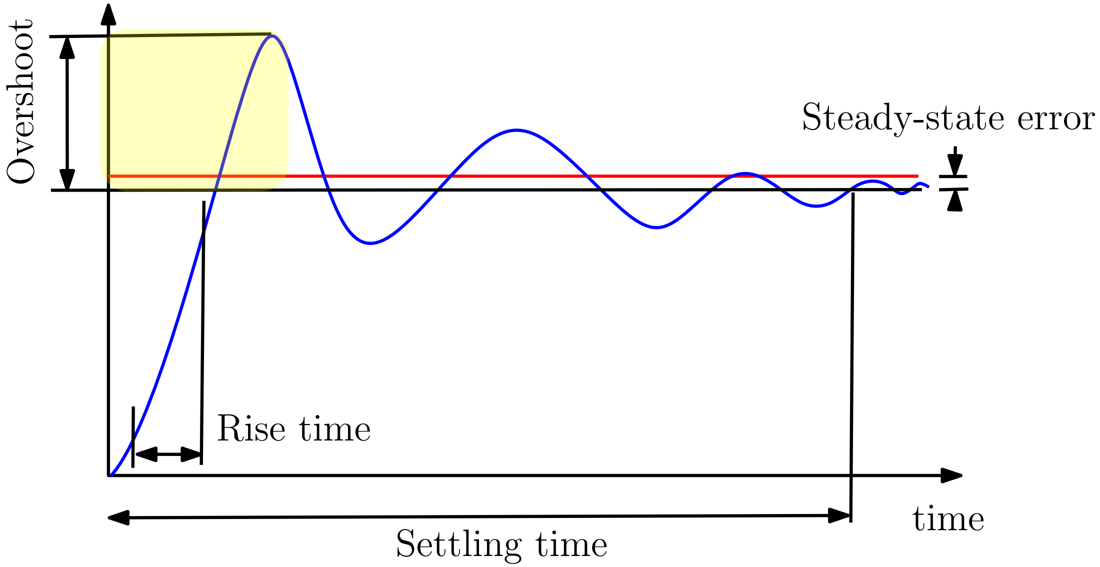
Controller Performance

- Steady-state error
  - Zero if possible– accurate tracking



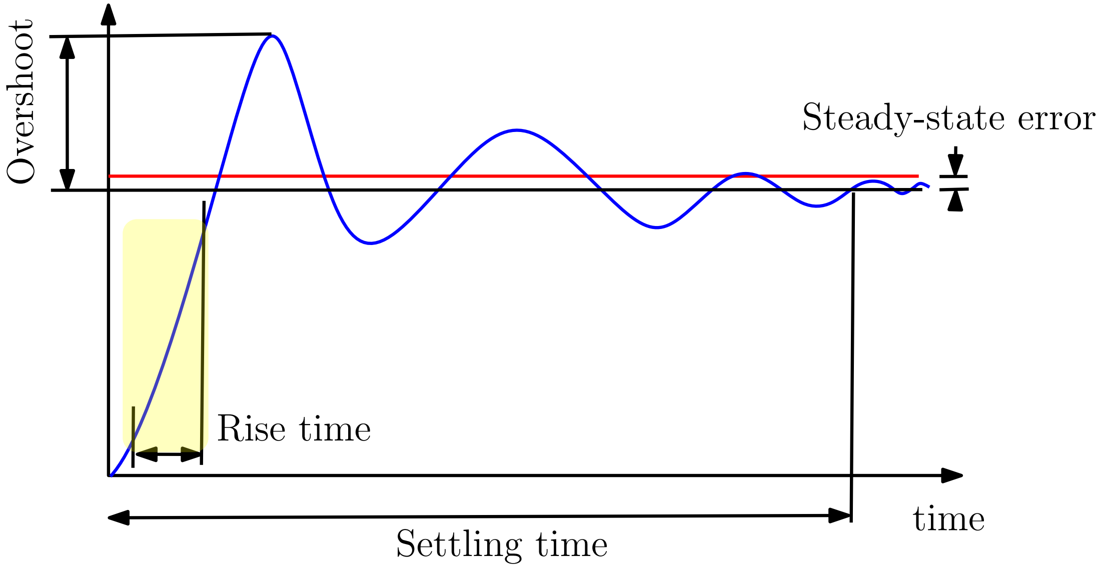
Controller Performance

- Steady-state error
  - Zero if possible– accurate tracking
- Overshoot
  - Minimal – Not huge deviations



Controller Performance

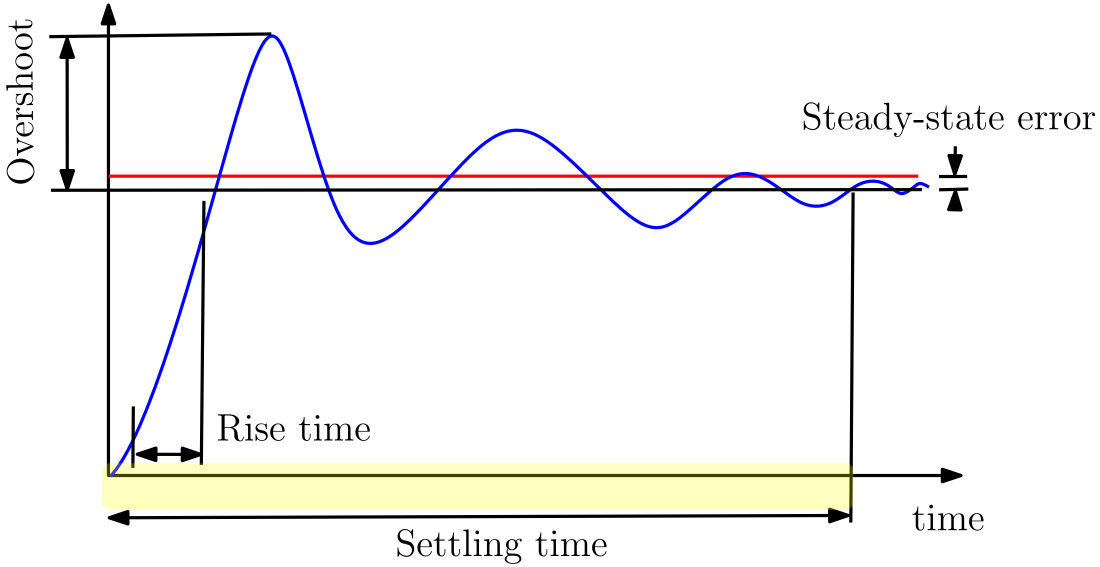
- Steady-state error
  - Zero if possible– accurate tracking
- Overshoot
  - Minimal – Not huge deviations
- Rise time
  - Small – Quick response



the time for the output to go from
10% to 90% of the steady state value

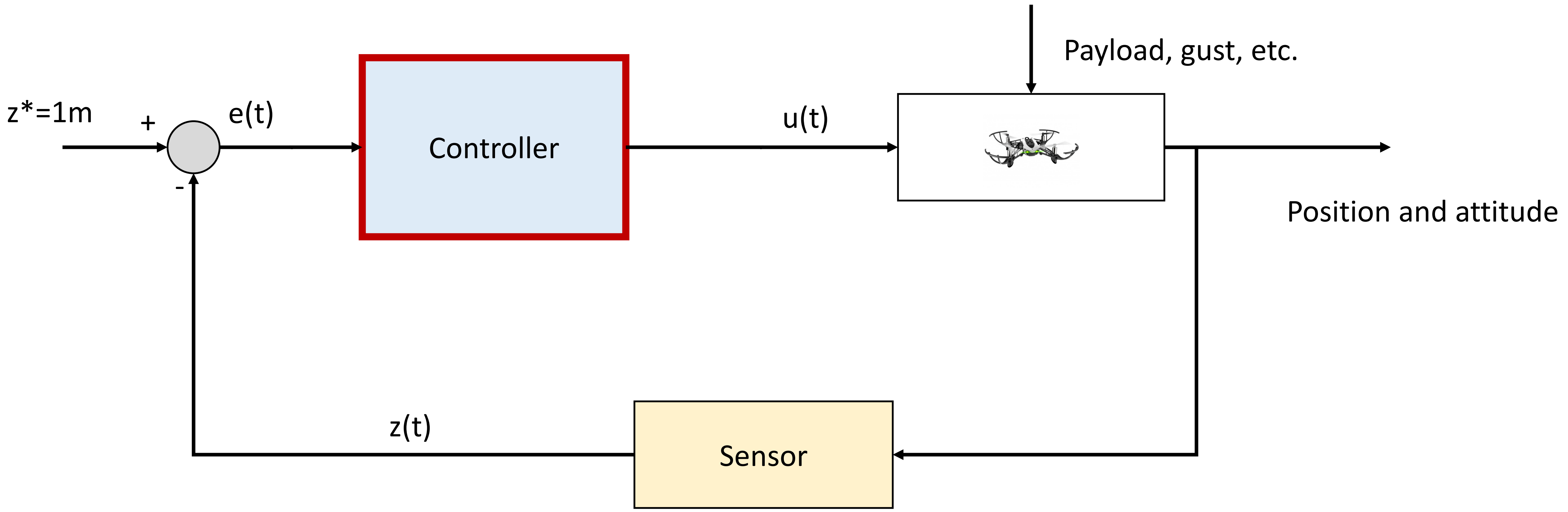
Controller Performance

- Steady-state error
  - Zero if possible– accurate tracking
- Overshoot
  - Minimal – Not huge deviations
- Rise time
  - Small – Quick response
- Settling time
  - Minimal – Quick response, less oscillations

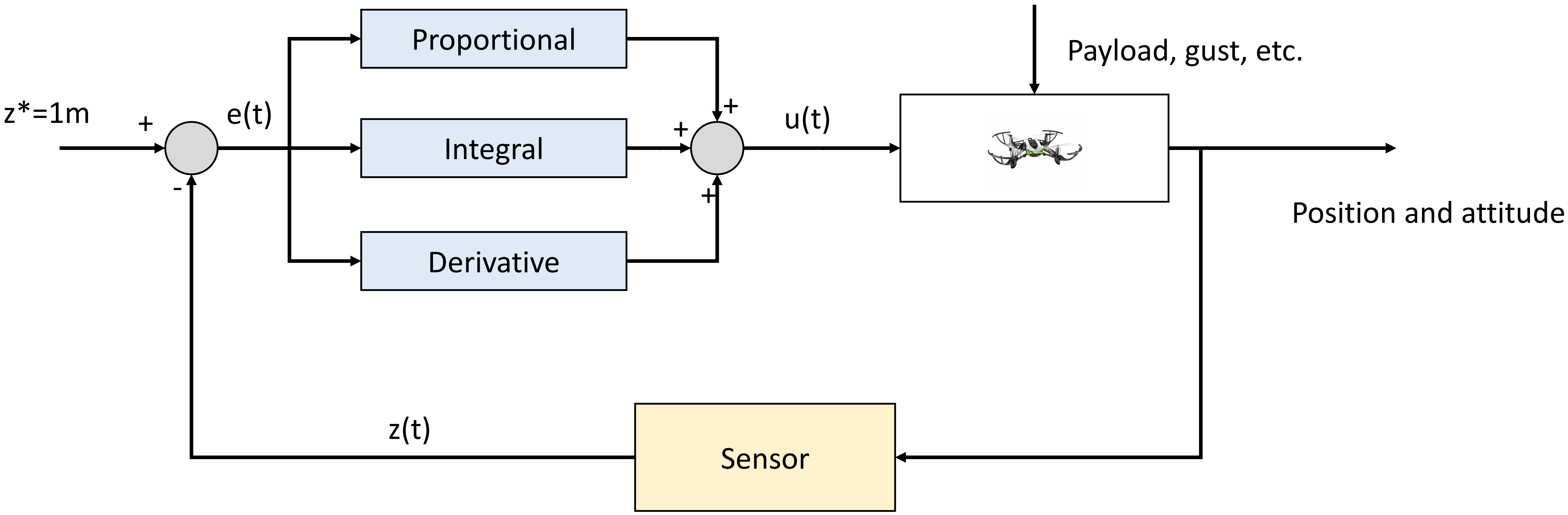


the time for the output to converge within $\pm5\%$ of the steady-state value

How to design controller?

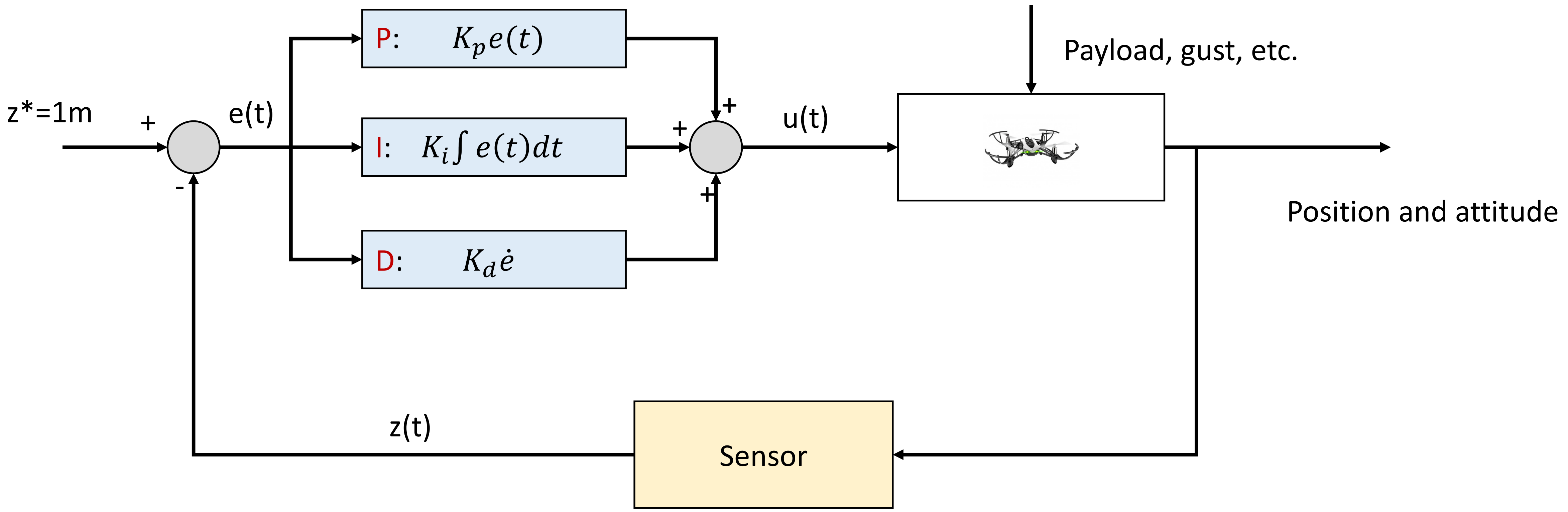


PID Control



z*=1m
+
-
e(t)
Proportional
Integral
Derivative
+
+
+
u(t)
Payload, gust, etc.
Position and attitude
z(t)
Sensor

PID Control



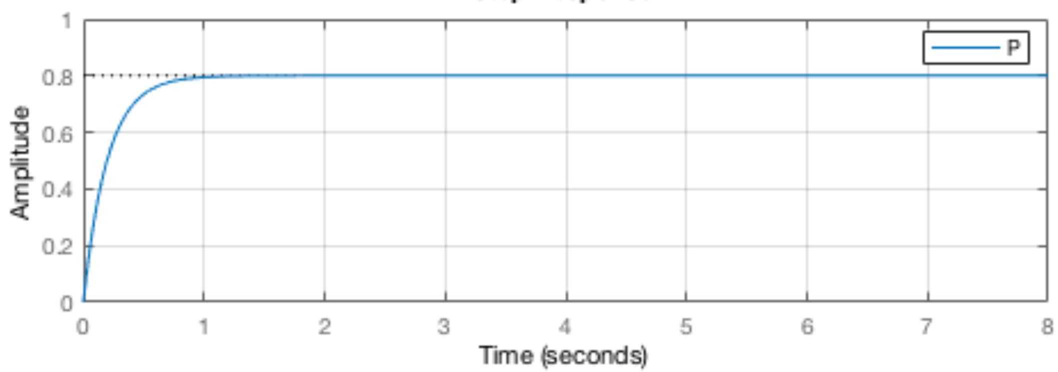
z*=1m
+
-
e(t)
P: $K_p e(t)$
I: $K_i \int e(t)dt$
D: $K_d \dot{e}$
+
+
+
u(t)
Payload, gust, etc.
Position and attitude
z(t)
Sensor

Proportional Control

First order system
$\dot{x} + ax = bu$

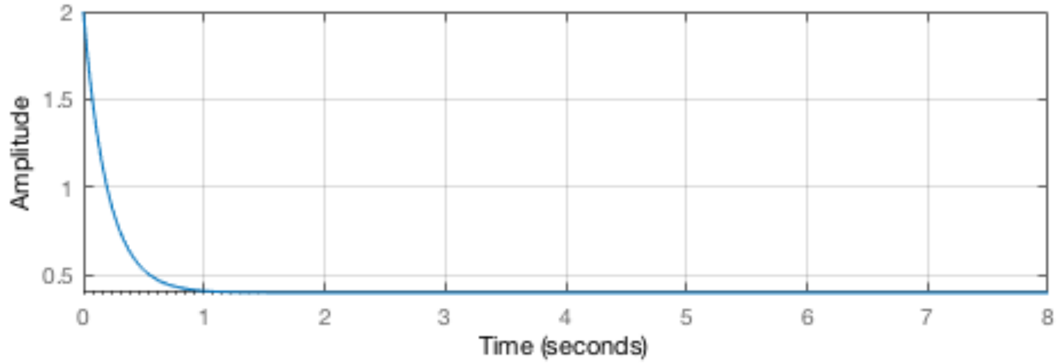
$$u(t) = K_p e(t) = K_p(z^* - z(t))$$

Proportional gain



- Provides small control when the error is small
  - Avoids excessive control effort
- Disadvantages
  - Steady-state error

Control effort



Integral Control

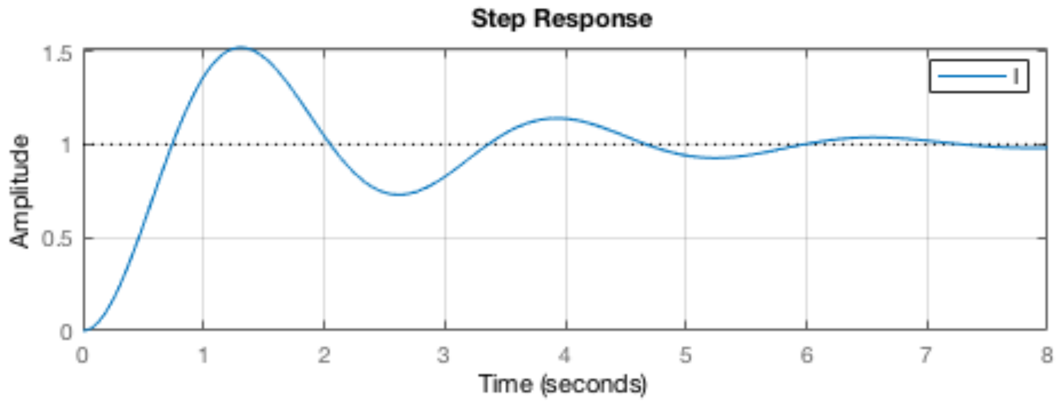
$$u(t) = K_i \int_0^t e(\tau)d\tau$$

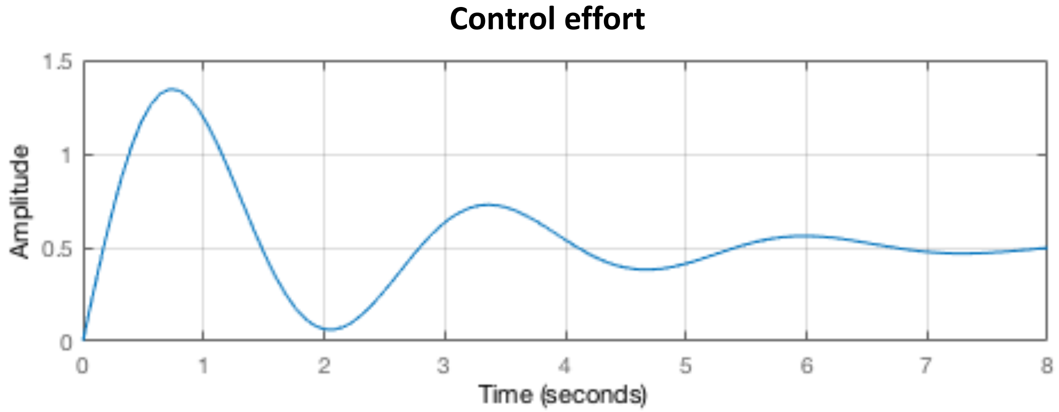
Integral gain

- Feedback based on the past values of the error signal
  - Can remove steady-state error

- Disadvantages:
  - Higher max deviation (overshoot)
  - Longer response time (long oscillations)

First order system
$$\dot{x} + ax = bu$$





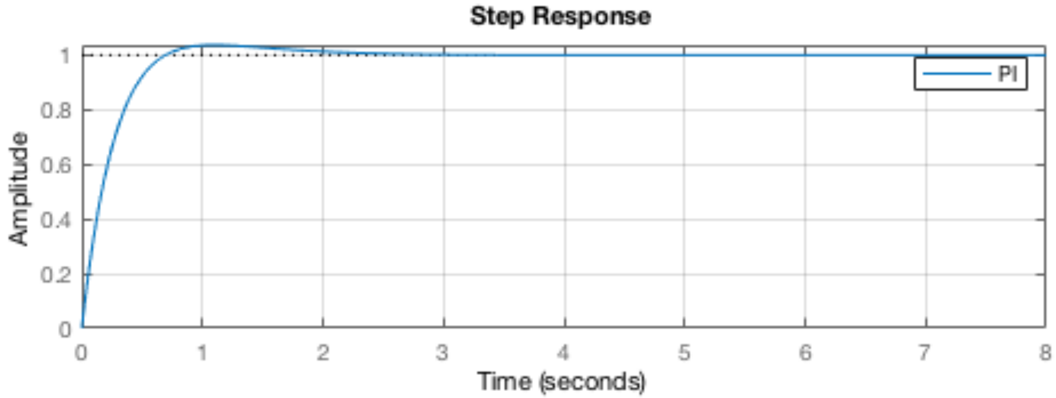
PI Control

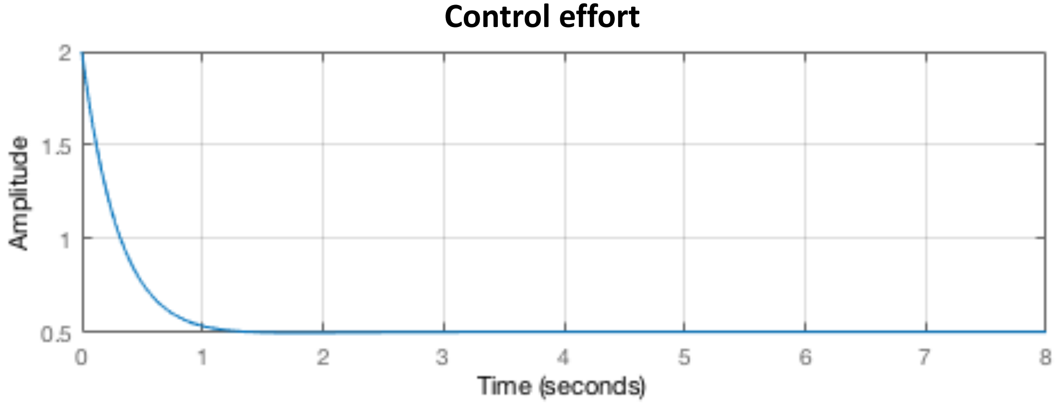
First order system

$$\dot{x} + ax = bu$$

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau$$

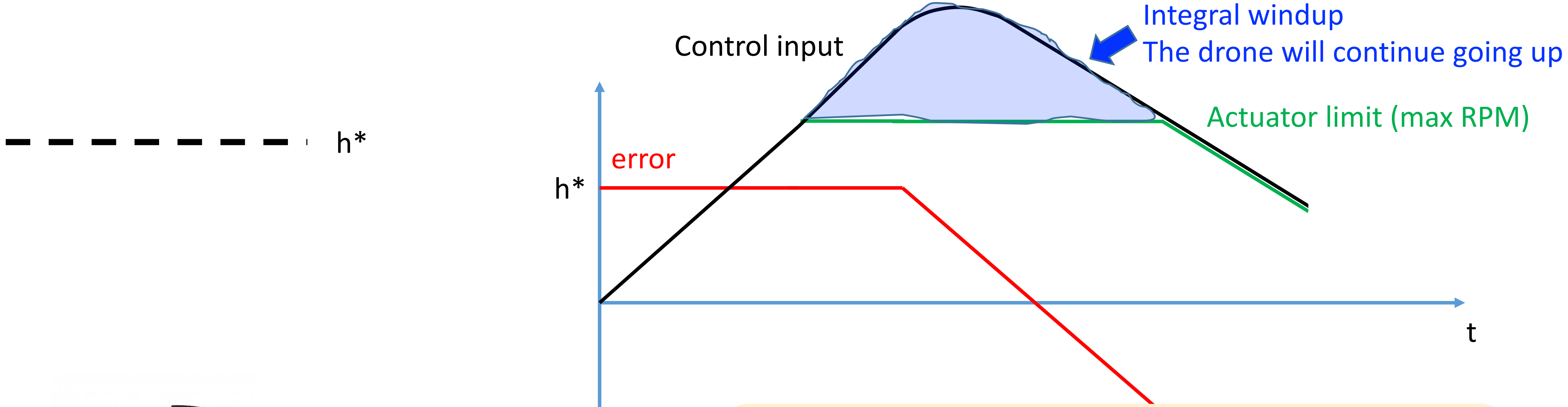
- Feedback based on the
  - error magnitude
  - past values of the error signal





Potential problem – integrator windup

- What happens if the control input is greater than the actuator limits?



You are holding the drone
You are releasing the drone

Some practical fixes:

- Limit integral error values
- Stop integral error while the commands are in the non feasible region
- Reduce gain of integral error term

Derivative Control

$$u(t) = K_d \frac{de(t)}{dt}$$

Derivative gain

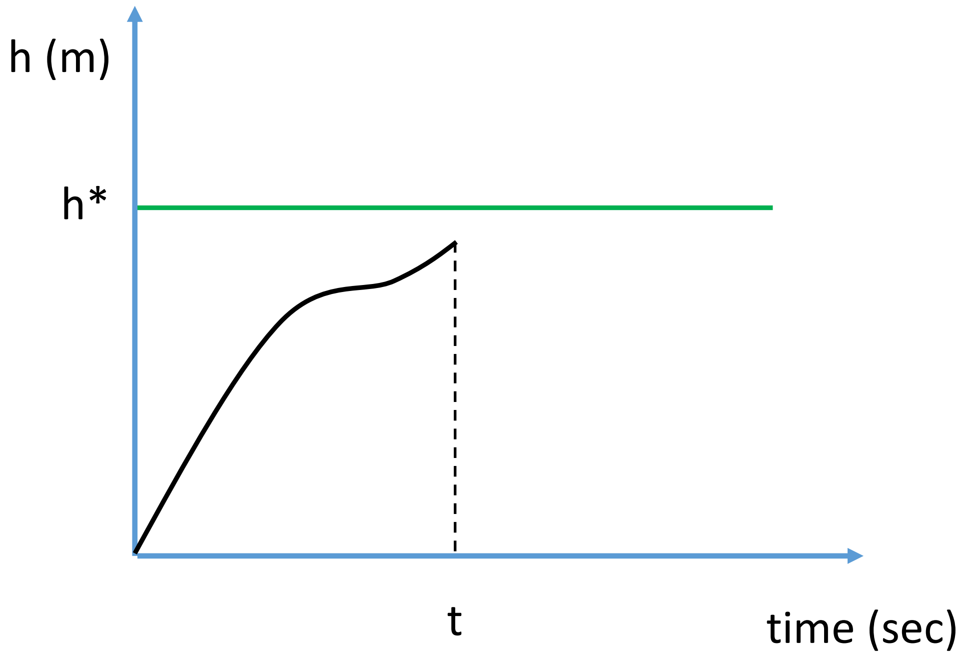
$$e(t) = z^* - z(t)$$

$$\frac{de(t)}{dt} = -\frac{dz(t)}{dt}$$

- Feedback based on the predicted future values of the error signal
- Anticipatory control
  - Anticipates incorrect trend of the error and counteract it
  - Faster response than P and I controllers
- Disadvantages:
  - Huge error + no change in error → No derivative control

Comprehensive Questions

- What is PID control?



$$e = h^* - h$$
$$\dot{e} = -\dot{h}$$

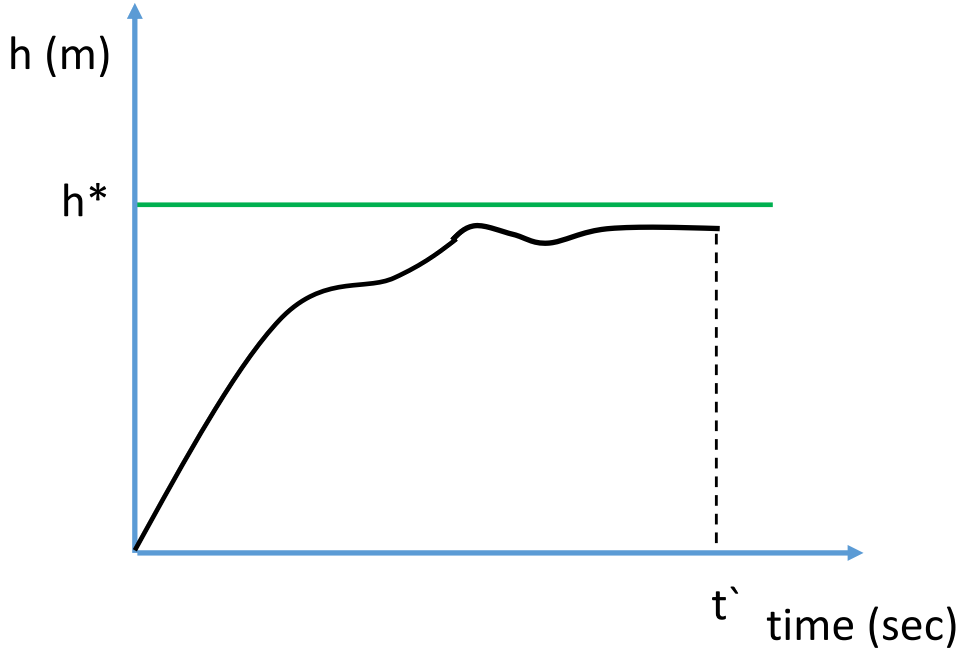
$$\begin{aligned} u_{PD} &= K_p e + K_d \dot{e} \\ &= K_p(h^* - h) - K_d \dot{h} \end{aligned}$$

$$\begin{aligned} u_P &= K_p e \\ &= K_p(h^* - h) \end{aligned}$$

$$u_P \; ? \; u_{PD}$$

Comprehensive Questions

- What is PID control?



$$e = h^* - h$$
$$\dot{e} = -\dot{h}$$

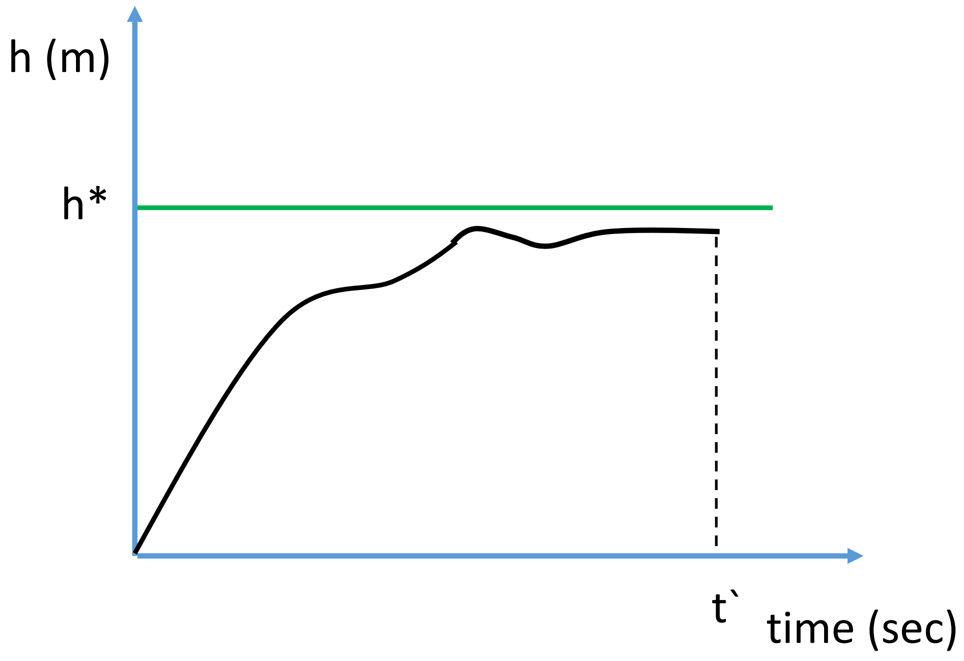
$$u_{PD} = K_p e + K_d \dot{e}$$
$$= K_p(h^* - h) - K_d \dot{h}$$

$$u_P = K_p e$$
$$= K_p(h^* - h)$$

$$u_P \; ? \; u_{PD}$$

Comprehensive Questions

- What is PID control?



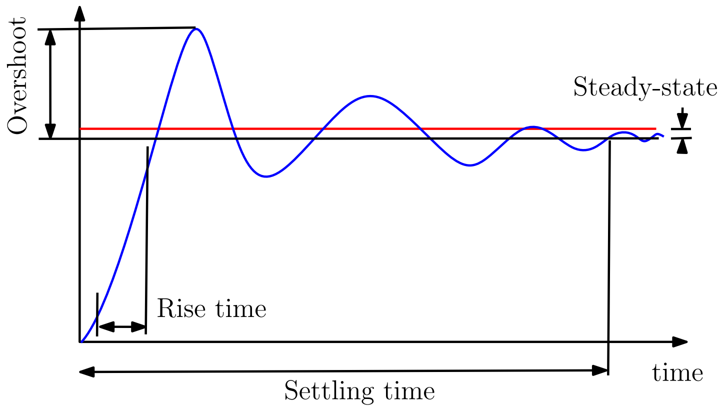
$$e = h^* - h$$
$$\dot{e} = -\dot{h}$$

$$\begin{aligned} u_{PD} &= K_p e + K_d \dot{e} \\ &= K_p(h^* - h) - K_d \dot{h} \end{aligned}$$

$$\begin{aligned} u_{PID} &= K_p e + K_D \dot{e} + K_I \int_0^t e dt \\ &= K_p(h^* - h) - K_D \dot{h} + K_I \int_0^t (h^* - h) dt \end{aligned}$$

Characteristics of P, I, D Controllers

	Benefits
K_p	Provides feedback based on the error magnitude
K_i	Eliminates the steady-state error
K_d	Reduces oscillations



Characteristics of P, I, D Controllers

	Benefits
K_p	Provides feedback based on the error magnitude
K_i	Eliminates the steady-state error
K_d	Reduces oscillations

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}$$

How do you select the controller type and the parameters?

Choice of Controller Type

- If small steady-state error is acceptable
  - P controller
  - In addition less oscillations, then PD control
- If zero steady-state error
  - PI controller
  - In addition less oscillations, then PID controller
- Order of the system matters too
  - First order: In general P or PI
  - Second order: In general PD or PID

Tuning the Parameters

a) Build a simple model

b) Use analysis or experimental data to determine unknown parameters in the model

c) Use this model to select the PID gains so that the closed-loop has certain desired characteristics

(for linear systems such as $\dot{x} = Ax + Bu$, we can show relationship between the gains and the desired system behavior)

(for nonlinear systems, selection of PID gains is often trial and error)

d) Do analysis and run simulations (check certain safety factors, margins for stability)

e) Implement and experimentally test

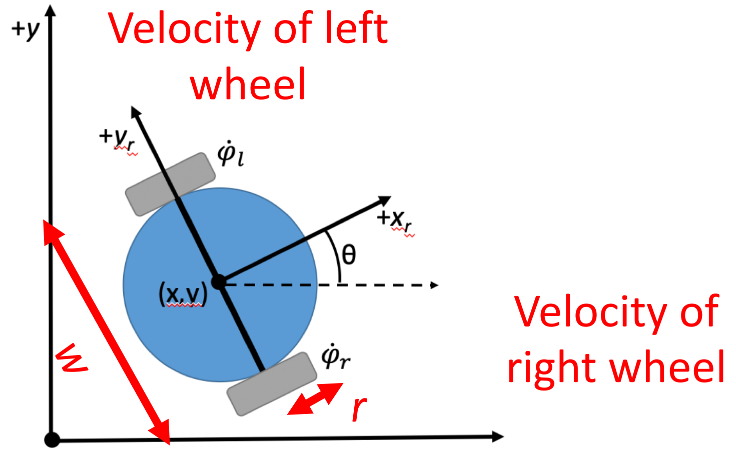
f) Iterate if needed

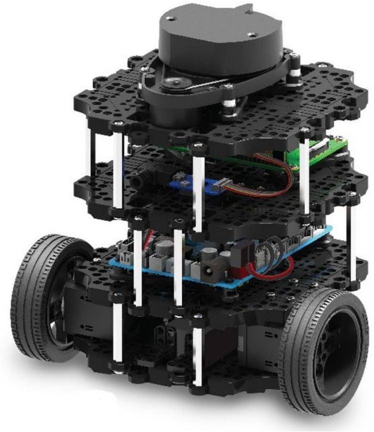
Controlling differential drive robots

- Recall the kinematic model of a differential drive robot

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{r}{2}(\dot{\varphi}_r + \dot{\varphi}_l) \\ 0 \\ \frac{r}{w}(\dot{\varphi}_r - \dot{\varphi}_l) \end{pmatrix}$$

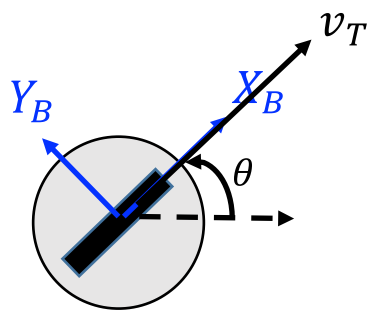
$$\dot{x} = \frac{r}{2}(\dot{\varphi}_r + \dot{\varphi}_l)\cos(\theta)$$
$$\dot{y} = \frac{r}{2}(\dot{\varphi}_r + \dot{\varphi}_l)\sin(\theta)$$
$$\dot{\theta} = \frac{r}{w}(\dot{\varphi}_r - \dot{\varphi}_l)$$





It is not very natural to control left and right wheel velocities!!!

Controlling differential drive robots



Recall the unicycle model:

- Inputs: velocity v_T, and angular velocity v_R
- Dynamics:

$$\left.\begin{aligned} \dot{x} &= v_T \cos(\theta) \\ \dot{y} &= v_T \sin(\theta) \\ \dot{\theta} &= v_R \end{aligned}\right\}$$

Design control for this model

$$\left.\begin{aligned} \dot{x} &= \tfrac{r}{2}(\dot{\varphi}_r + \dot{\varphi}_l)\cos(\theta) \\ \dot{y} &= \tfrac{r}{2}(\dot{\varphi}_r + \dot{\varphi}_l)\sin(\theta) \\ \dot{\theta} &= \tfrac{r}{w}(\dot{\varphi}_r - \dot{\varphi}_l) \end{aligned}\right\}$$

Implement this model

$$v_T = \frac{r}{2}(\dot{\varphi}_r + \dot{\varphi}_l) \quad \rightarrow \quad \frac{2v_T}{r} = \dot{\varphi}_r + \dot{\varphi}_l$$

$$v_R = \frac{r}{w}(\dot{\varphi}_r - \dot{\varphi}_l) \quad \rightarrow \quad \frac{v_R w}{r} = \dot{\varphi}_r - \dot{\varphi}_l$$

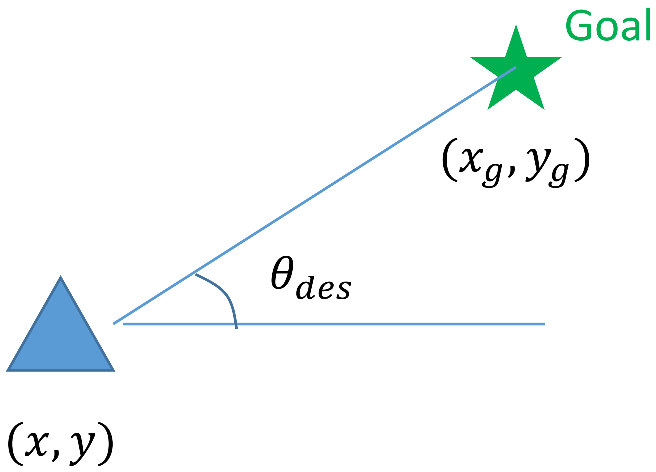
$$\dot{\varphi}_r = \frac{2v_T + v_R w}{2r} \qquad \dot{\varphi}_l = \frac{2v_T - v_R w}{2r}$$

Controlling differential drive robots

How to drive a robot to a goal location?

Let's consider constant velocity v_0, then we will control angular velocity v_R

$$e = \theta_{des} - \theta \qquad v_R = PID(e)$$



$$\theta_{des} = \arctan\left(\frac{y_g - y}{x_g - x}\right)$$

Discretization

$$x^+ = x + v_0 \cos(\theta)\Delta t$$
$$y^+ = y + v_0 \sin(\theta)\Delta t$$
$$\theta^+ = \theta + v_R \Delta t$$

```
Repeat every Δt sec. {
```
$$e = \theta_{des} - \theta$$
$$e_{dot} = (e - e_{prev})/\Delta t$$
$$e_{int} = e_{int} + e\Delta t$$
$$u = K_p e + K_i e_{int} + K_d e_{dot}$$
$$e_{prev} = e$$
```
Send_control(u)}
```

A more formal approach

- Numerical integration and observing system response for a given set of (K_p, K_i, K_d) is practical. However, tuning the gains may require many trials.
- Alternatively, we can solve ordinary differential equations (ODEs) and design the gains with respect to desired system behaviors.

Example:

$$m\ddot{h} = 4k_T\omega^2 - mg$$

I.C.

$$h(0) = h_0$$

$$\dot{h}(0) = \dot{h}_0$$

2nd order ODE

Linear/Nonlinear ODE

describing the effect of an input u on an output y

$$a_n y^{[n]}(t) + a_{n-1} y^{[n-1]}(t) + \cdots + a_1 \dot{y}(t) + a_0 y(t) = b_m u^m(t) + \cdots + b_1 \dot{u}(t) + b_0 u(t)$$

- $y^k = \frac{d^k y}{dt^k}$
- $a_n, \dots, a_0, b_m, \dots, b_0$ are constants that are selected to model the dynamical system
- An n^{th} order differential equation requires n initial conditions (ICs):

$$y(0) = y_0, \qquad \dot{y}(0) = \dot{y}_0 \quad , \dots , \qquad y^{[n-1]}(0) = y_0^{[n-1]}$$

Linear ODE: only involves linear combinations of $y, u, \dot{y}$, etc.

Nonlinear ODEs: may contain terms like $u^2(t), \sin\big(y(t)\big)$, etc.

$$y^{[n]}(t) = f\big(y(t), \dot{y}(t), \dots, y^{[n-1]}(t), u(t), \dot{u}(t), \dots, u^m(t)\big)$$

- Higher fidelity models
- Nonlinear control design and analysis (not the scope of this course)
- Linearizing the systems and using linear control design and analysis (some parts will be covered in this course)

Principle of superposition

- An important fact linear ODEs satisfy

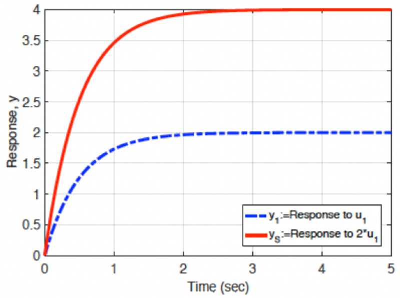
Suppose that

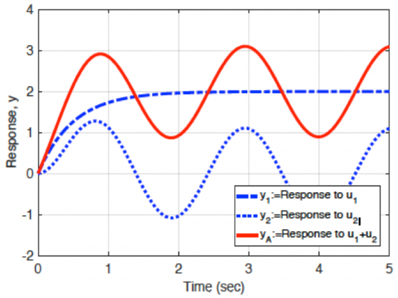
- $y_1(t)$ is the solution to linear ODE with input $u_1(t)$ and zero IC
- $y_2(t)$ is the solution to linear ODE with input $u_2(t)$ and zero IC

Then the linear ODE satisfies the following two properties:

- **Scaling:** For any constant $c \in \mathbb{R}$, the solution of the linear ODE with input $u_S(t) = cu_1(t)$ and zero IC is given by $y_S(t) = cy_1(t)$.
- **Additivity:** The solution to the linear ODE with input $u_A(t) = u_1(t) + u_2(t)$ and zero IC is given by $y_A(t) = y_1(t) + y_2(t)$

- Nonlinear ODEs do not satisfy these properties, as a result they are more challenging to analyze!





$$\dot{y}(t) + 2y(t) = 4u(t)$$

Left: $y_1(t)$ and $y_S(t)$ to inputs $u_1(t) = 1$ and $u_S(t) = 2u_1(t)$ with zero IC

Right: $y_1(t), y_2(t)$, and $y_A(t)$ to inputs $u_1(t) = 1, u_2(t) = \sin(3t)$, and $u_A(t) = u_1(t) + u_2(t)$ with zero IC.

Solving ODEs

Homogeneity: A linear ODE is homogeneous if all non-zero terms contain x or its derivatives.

Homogeneous	Non-homogeneous
$\dot{x} + ax = 0$	$\dot{x} + ax = 2$
$\ddot{x} + a\dot{x} + bx = 0$	$\ddot{x} + a\dot{x} + bx = \sin(t)$

Non-homogeneity arises from a forcing function (input) in dynamical systems.

- force/torque (mechanical), voltage/current source (electrical), etc.

$$a_n y^{[n]}(t) + a_{n-1} y^{[n-1]}(t) + \cdots + a_1 \dot{y}(t) + a_0 y(t) = b_m u^m(t) + \cdots + b_1 \dot{u}(t) + b_0 u(t)$$

IC: $y(0) = y_0, \quad \dot{y}(0) = \dot{y}_0 \quad , \dots , \quad y^{[n-1]}(0) = y_0^{[n-1]}$

Homogenous solution

Homogeneous solution is the free response of a linear system with input u and output y by setting the input zero. i.e., $u(t) = 0$, for all $t \geq 0$.

$$a_n y^{[n]}(t) + a_{n-1} y^{[n-1]}(t) + \cdots + a_1 \dot{y}(t) + a_0 y(t) = 0$$

IC: $y(0) = y_0,\ \ \dot{y}(0) = \dot{y}_0 \quad , \ldots , \ y^{[n-1]}(0) = y_0^{[n-1]}$

1. Solve for the n roots $\{s_1, \ldots, s_n\} \subset \mathbb{C}$ of the characteristic equation.
2. Form the general solution as $y(t) = \sum_{i=1}^{n} c_i e^{s_i t}$.
3. Use the n initial conditions to solve for n unknown coefficients $\{c_1, \ldots, c_n\} \subset \mathbb{C}$

Note: if the system has repeated roots, the general solution at step 2 needs to be altered.

Example: $\ddot{x} + a\dot{x} + bx = 0 \ \rightarrow \ \ s^2 + as + b = 0$

- **Distinct real roots $\boldsymbol{s_1 \neq s_2}$:** $x(\mathrm{t}) = c_1 e^{s_1 t} + c_2 e^{s_2 t}$
- **Repeated real roots $\boldsymbol{s_1 = \ s_2 = r}$:** $x(\mathrm{t}) = c_1 e^{rt} + c_2 t e^{rt}$
- **Complex roots $\boldsymbol{s_{1,2} = \alpha \pm \beta i}$:** $x(\mathrm{t}) = c_1 e^{(\alpha+\beta i)t} + c_2 e^{(\alpha-\beta i)t} = c_1 e^{\alpha t}\cos(\beta t) + c_2 e^{\alpha t}\sin(\beta t)$

Stability

Definition: A linear system is stable if the free response returns to zero ($y(t) \to 0$ as $t \to \infty$) for any initial condition. The system is called unstable if it is not stable.

Fact: A linear system is stable if and only if all roots of the characteristic equation have strictly negative real part, i.e. $Re\{s_i\} < 0$ for all i where Re denotes the real part of the root.

Example: $\ddot{x} + a\dot{x} + bx = 0 \;\rightarrow\; s^2 + as + b = 0$

Distinct real roots $\boldsymbol{s_1 \neq s_2}$: $x(\mathrm{t}) = c_1 e^{s_1 t} + c_2 e^{s_2 t}$
Repeated real roots $\boldsymbol{s_1 = s_2 = r}$: $x(\mathrm{t}) = c_1 e^{rt} + c_2 t e^{rt}$
Complex roots $\boldsymbol{s_{1,2} = \alpha \pm \beta i}$: $x(\mathrm{t}) = c_1 e^{(\alpha+\beta i)t} + c_2 e^{(\alpha-\beta i)t} = c_1 e^{\alpha t}\cos(\beta t) + c_2 e^{\alpha t}\sin(\beta t)$

More details on stability analysis will be discussed in the following lectures.

Solution to nonhomogeneous ODE

$$a_n y^{[n]}(t) + a_{n-1} y^{[n-1]}(t) + \cdots + a_1 \dot{y}(t) + a_0 y(t) = b_m u^m(t) + \cdots + b_1 \dot{u}(t) + b_0 u(t)$$

IC: $y(0) = y_0,\ \ \dot{y}(0) = \dot{y}_0 \quad , \dots ,\ y^{[n-1]}(0) = y_0^{[n-1]}$

1. Solve for the n roots $\{s_1, \dots, s_n\} \subset \mathbb{C}$ of the characteristic equation.
2. Find any particular solution $y_P(t)$
3. Form the general solution as $y(t) = y_P(t) + \sum_{i=1}^{n} c_i e^{s_i t}$.
4. Use the n initial conditions to solve for n unknown coefficients $\{c_1, \dots, c_n\} \subset \mathbb{C}$

Note: if the system has repeated roots, the general solution at step 3 needs to be altered.

Example: $\ddot{y} + a\dot{y} + by = f(t)$

- How to find the particular solution, $y_P(\mathrm{t})$?
- $\boldsymbol{f(t)}$ **is** $\boldsymbol{e^{kt}}$**:** use γe^{kt}
- $\boldsymbol{f(t)}$ **is a polynomial:** use a generic polynomial of same degree
- $\boldsymbol{f(t)}$ **is** $\boldsymbol{\sin(\beta t)}$ **or** $\boldsymbol{\cos(\beta t)}$**:** use $\gamma_1 \sin(\beta t) + \gamma_2 \cos(\beta t)$
- When $f(t)$ is the sum of familiar functions, solve each separately and add.

You can use off-the-shelf solvers to solve ODEs!