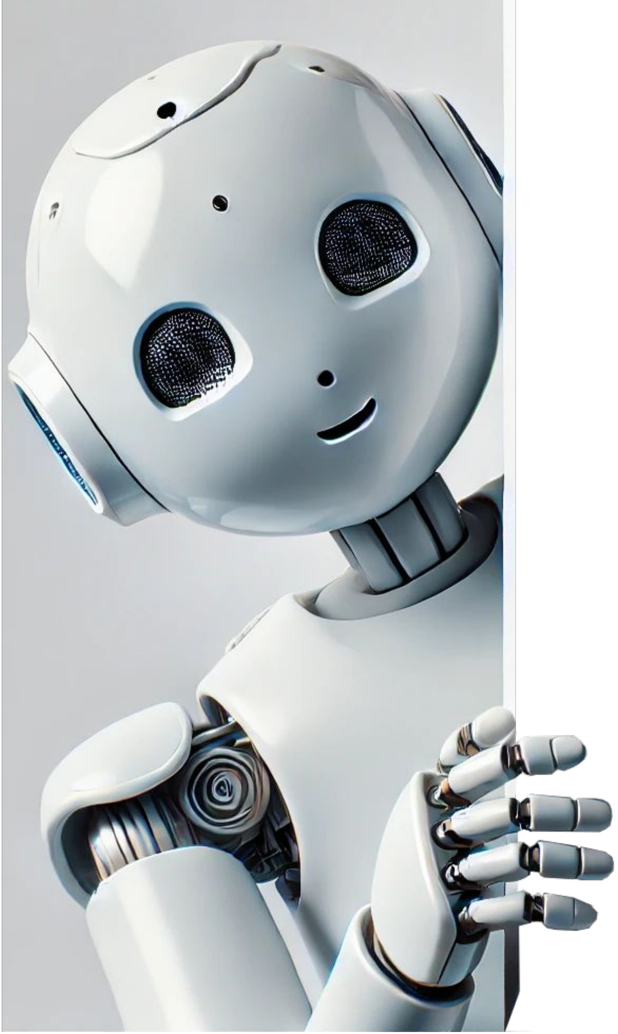


# Informal Introduction to robot control

Kirill Krinkin

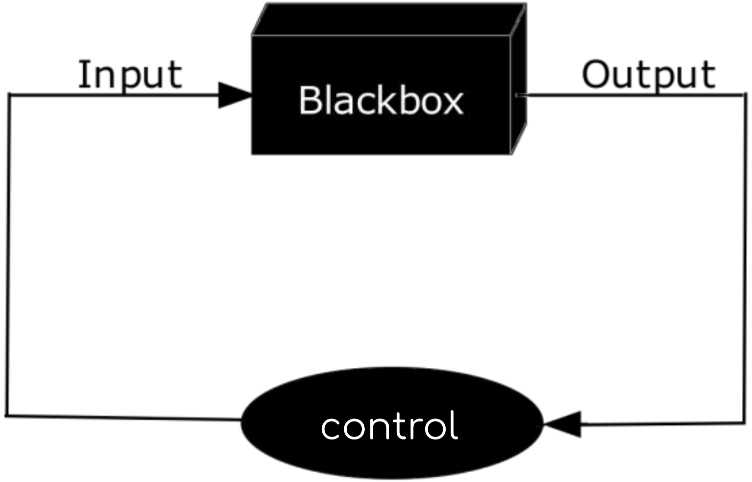
Serbia, Tršić, 30 June 2025

# Today

- Main concepts
- Actors in control system
- Sensors and actuators
- Open-Loop control
- Closed-Loop control
- Controller and control model
- PID Controller overview

# Robot control and control theory

Robot is doing a backflip




Control theory standpoint

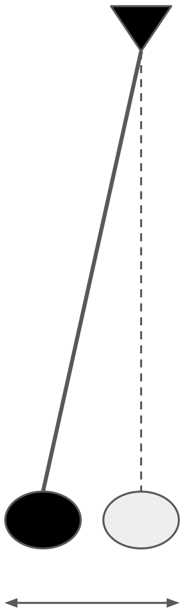
https://www.thedrive.com/tech/16179/say-hello-to-atlas-the-backflipping-robot-ninja-from-boston-dynamics

# Dynamic System

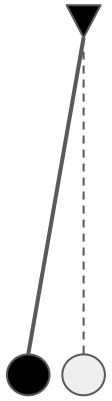
- Definition: A system whose state changes over time according to a predefined **rule**.

  Note:
  In modeling, a dynamic system is a mathematical abstraction of a real-world (physical or otherwise) system called the plant; the model captures the rules governing its behavior.

- Key components
  - **State**: a vector of variables **x(t)** describing the system at time **t.**
  - **Evolution Rule**: a function **f** that specifies how the state evolves over time.
  - **Initial State**: the value $\mathbf{x(t_0)}$ at the initial time $\mathbf{t_0}$



# Dynamic System Notation



$t_1$ $t_0$

$\mathrm{x}(\mathrm{t}_1)$ $\mathrm{x}(\mathrm{t}_0)$

System states in $t_1$ and $t_0$

- t – moment of time (variable)
- time interval:

$$(t_1 - t_0) \rightarrow \Delta t \rightarrow dt$$

- state change

$$((\mathrm{x}(\mathrm{t}_1) - x(t_0)) \rightarrow dx(t)$$

if change interval very small

$$\frac{dx(t)}{dt} = f\big(x(t), t\big)$$

Speed of change

Evolution rule

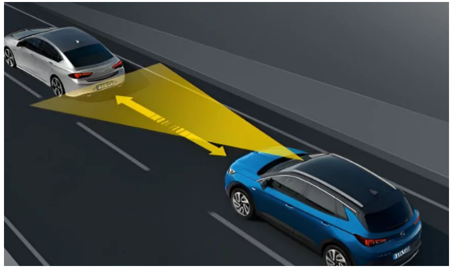
# Uncertainty

- Definition: uncertainty refers to the lack of precise knowledge about the system's true state or behavior due to noise, modeling errors, or external disturbances.

- Uncertainty Sources:
    - Sensor noise: measurements (e.g., position, velocity)
    - Modeling errors: simplifications or unknown dynamics in the plant's model
    - External disturbances: unpredictable influences from the environment
        - wind gusts
        - surface irregularities

# Idea of control



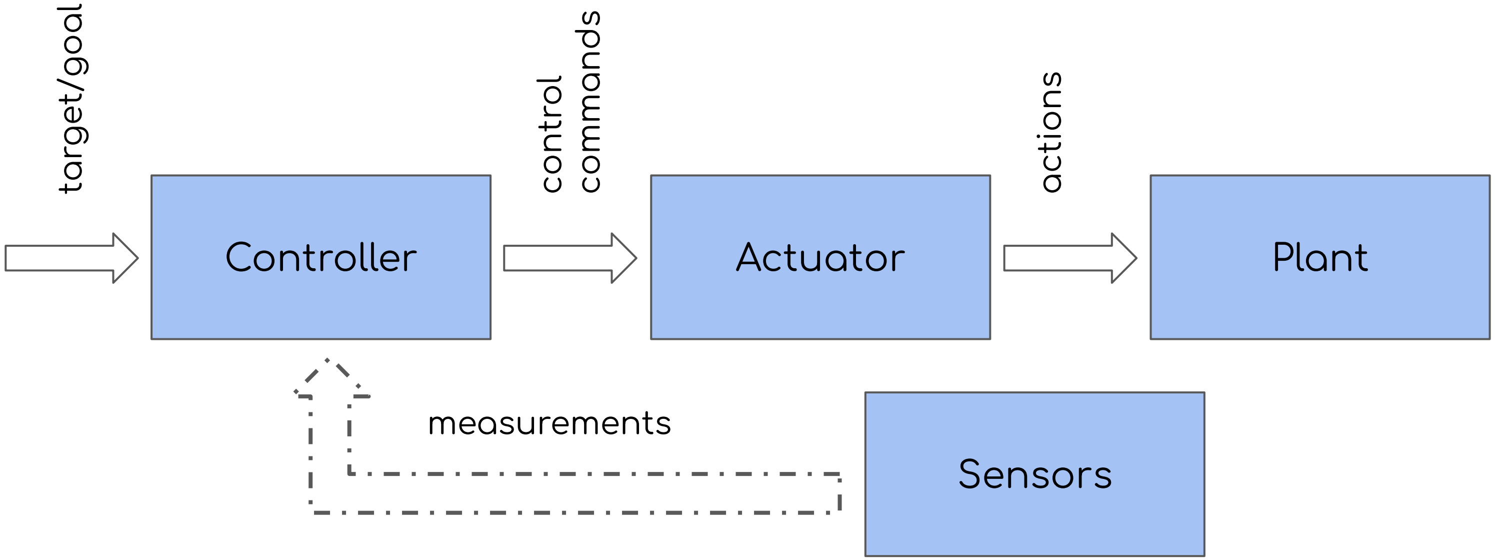
# Control

- control is the process of manipulating system inputs to guide its behavior toward a desired goal or reference signal.

- Key Aspects:
  - Feedback: measuring the system's output and adjusting inputs to correct deviations.
  - Feedforward: applying known control actions in advance to counteract predictable disturbances.
  - Performance Metrics: stability, accuracy (tracking error), response speed, and robustness.

- Example: A robot arm controller uses encoder readings (feedback) to adjust motor torques, maintaining the arm at a desired position despite payload changes.

# Main actors in control system



# The Plant (Object of Control) / Actuators

The plant is the real-world system or process being controlled. Its behavior is described by the dynamic system model.

- Characteristics:
  - Dynamics (intrinsic evolution of the plant's state)
  - Inputs: actions from actuators that influence the state.
  - Outputs: measurable variables used for feedback


Servo-motors

Actuators are devices that convert control signals into physical actions, applying forces or motions to the plant.

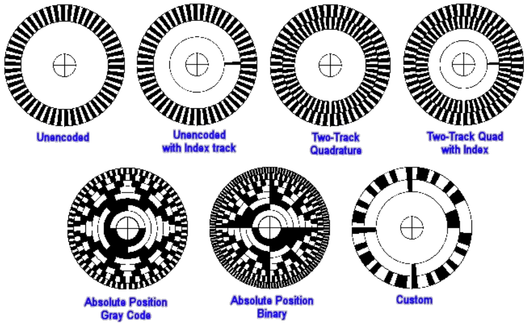
Motor

mass–spring–damper system

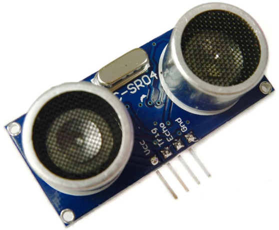
# Sensors

Devices that measure physical quantities and provide data for feedback in control systems.

- Types:
  - Proprioceptive: for internal state e.g.: encoders, IMU, accelerators
  - Exteroceptive: for external environment (cameras, LIDAR, range finders.
- Characteristics:
  - Noise & Precision
  - Sampling Rate
  - Range & Resolution

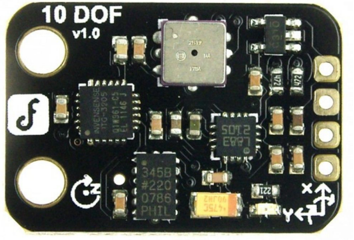



encoders



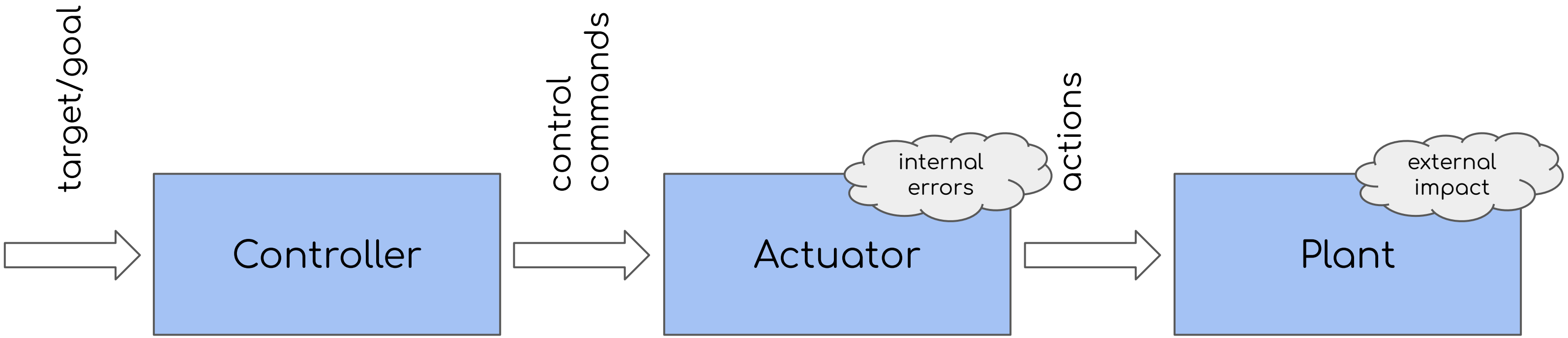
Ultrasonic range


gyroscope




IMU

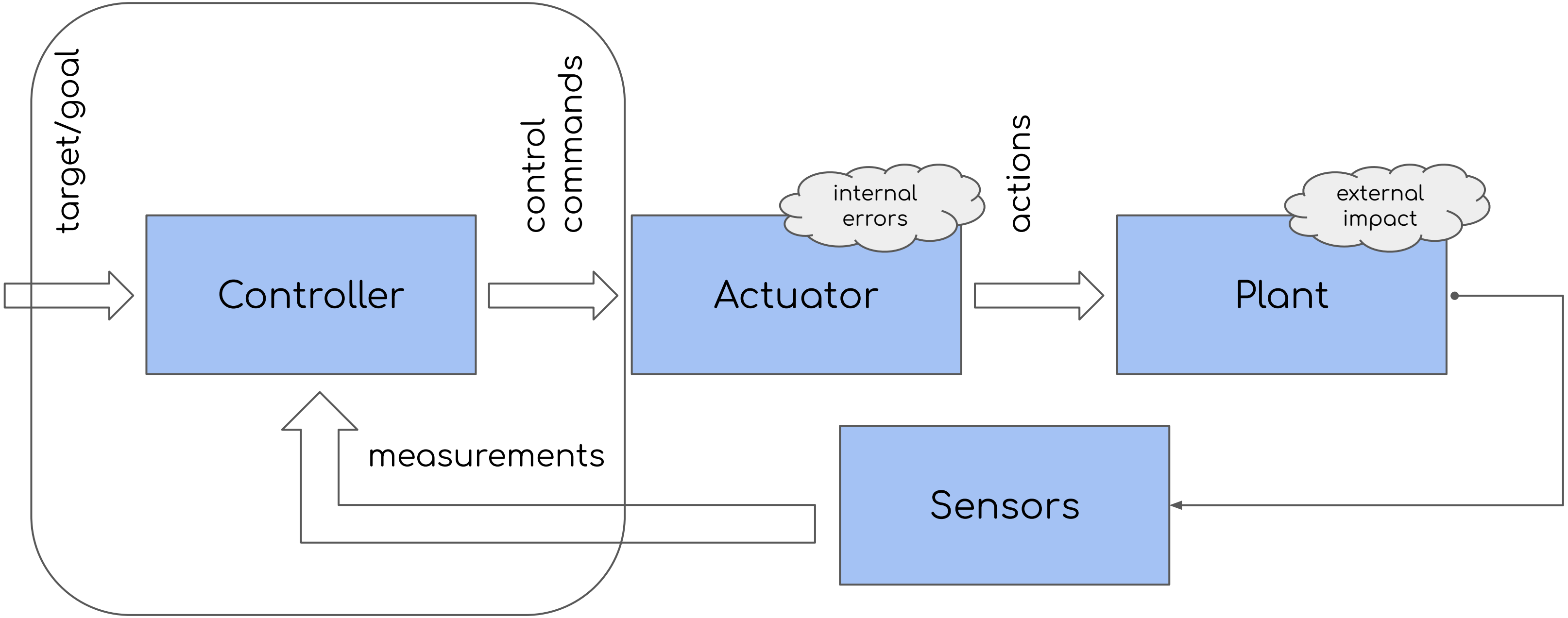
# Open-Loop Control




- No Feedback
- Predictive Inputs: relies on accurate models or predetermined schedules.
- Simplicity: easier to implement but sensitive to disturbances and modeling errors.

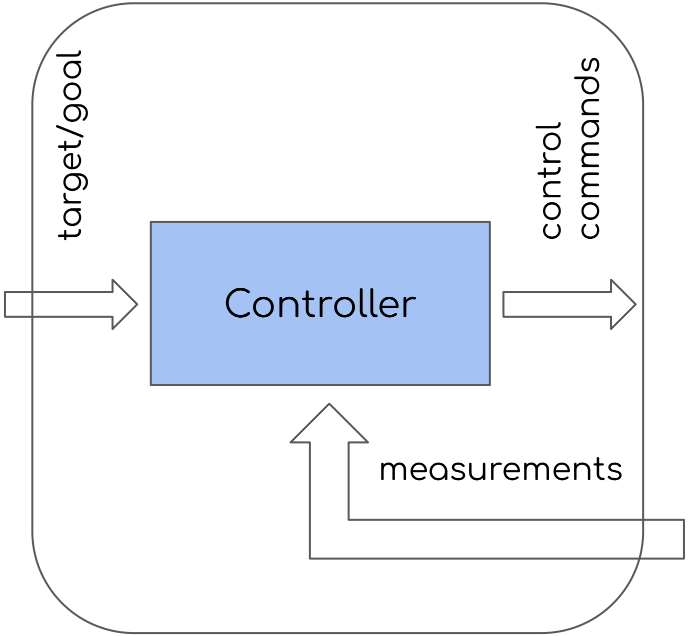
- Advantages:
  - Low complexity and cost.
  - Fast response since no sensor processing or feedback calculation.
- Disadvantages:
  - Cannot correct for unexpected disturbances or model inaccuracies.
  - Performance degrades if the plant or environment changes.

# Closed-Loop Control




Feedback loop

# Closed-Loop Characteristics




- continuous measurement of outputs and correction of inputs.
- difference e(t) = r(t) - y(t) guides the controller's action (error).
- can compensate for disturbances and model inaccuracies

- Disadvantages:
  - Increased complexity and cost due to sensors and computation.
  - Potential for instability if not properly designed

# Control Model

- Definition: a mathematical representation of how control inputs influence a dynamic system to achieve desired behavior.

- Key Components:
  - Control Input u(t): external signals applied to the system.
  - Controller g: an algorithm or rule that computes u(t) based on the system's state and the reference.
  - Reference Signal r(t): the desired state or trajectory.
  - Plant: the dynamic system being controlled.

$$\frac{dx(t)}{dt} = f\left(x(t), u(t), t\right)$$

$$u(t) = g\big(x(t), r(t), t\big)$$

y(t) for observation

# Controller

A controller is an algorithm or a device that computes control inputs based on the error between a desired reference and the measured output of the plant.

- Key Elements:
  - Error signal: $e(t) = r(t) - y(t)$ indicates deviation from the reference.
  - Control law: a rule $u(t) = g(e(t), e'(t), \ldots)$ that determines the input.
  - Parameters (gains): tuning constants
- Objectives:
  - Minimize tracking error
  - Ensure system stability
  - Achieve desired transient response (rise time, overshoot, settling time).

# PID Controller

JetBrains | Neapolis University Pafos

Proportional-Integral-Derivative (PID) controller minimizes deviation from the reference computes the control input as a weighted sum of:

- current error
- cumulative error
- speed of error change

**Control rule:**

$$\mathrm{u(t)} = \mathrm{K}_p\mathcal{P} + K_p\mathcal{I} + K_p\mathcal{D}$$

PID

Parameters:

- Kp – proportional gain
- Ki – integral gain
- Kd – derivative gain

# PID Controller components

P (proportional): eliminates present error

$$u(t) = K_p\mathcal{P} + K_p\mathcal{I} + K_p\mathcal{D}$$

I (integral): eliminates past error

D (derivative): eliminates future error

# Proportional Component

$$u_p(t) = K_p\, e(t)$$

- produces an output proportional to the current error
- Provides immediate corrective effort
- Larger value increases responsiveness but may cause overshoot

# Integral Component

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

- sums the error over time to eliminate steady-state offset
- accumulates past errors to drive long-term correction
- helps to remove residual bias but can introduce lag and overshoot if too large

Notation:

$$x_1 + x_2 + \ldots + x_n \rightarrow \sum_{i=1}^{n} x_i$$

or if x continuous

$$\int_1^n x dx$$

# Derivative Component

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

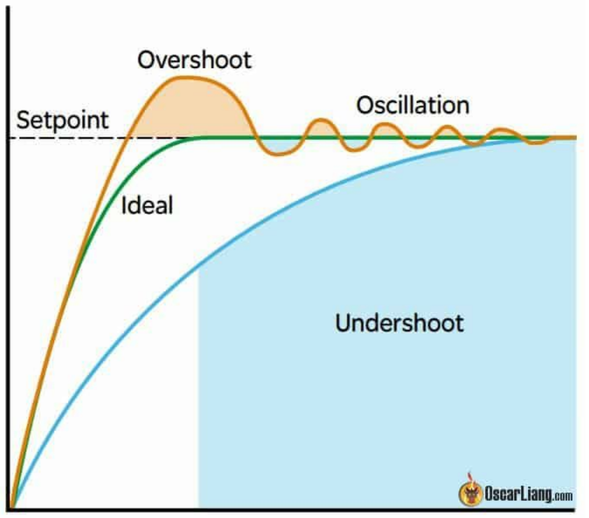
- derivative component predicts future error by evaluating its rate of change
- Reacts to the speed of error change, damping oscillations
- Improves stability and reduces overshoot but is sensitive to measurement noise

# PID controller – a Full Form

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

# Trade-offs

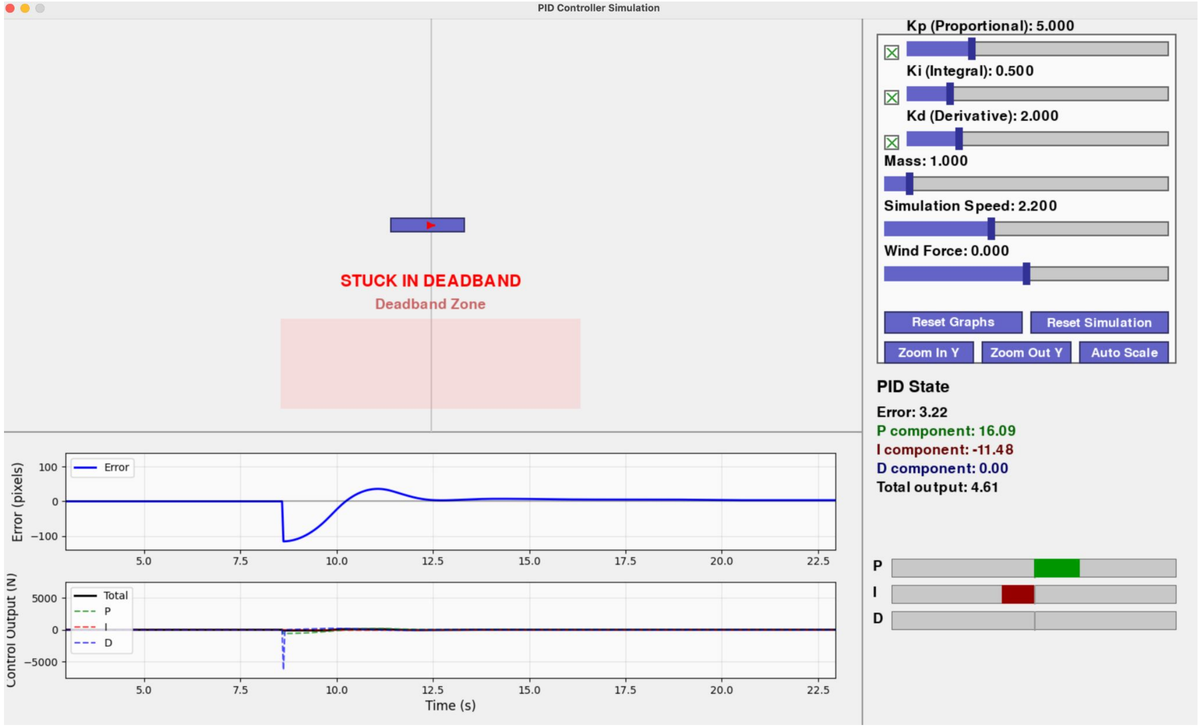
- Balancing Gains:
  - High Kp: faster response, but increased overshoot and risk of instability.
  - High Ki: eliminates steady-state error, but slows transient response and can cause excessive overshoot.
  - High Kd: improves damping and reduces overshoot, but amplifies measurement noise and may introduce jitter.
- Trade-off Summary:
  - Speed vs Stability: raising Kp speeds up reaction but can destabilize the system.
  - Accuracy vs Overshoot: increasing Ki improves long-term accuracy but can induce oscillations.
  - Damping vs Noise Sensitivity: larger Kd smooths response but magnifies sensor noise.




https://oscarliang.com/pid/

# Demo

https://github.com/krinkin/pid-simulation



# Questions?

# Bachelor's Degree in Computer Science and Artificial Intelligence with JetBrains

JET BRAINS

Neapolis University Pafos

Landing

Materials

Chat

# Contacts

Kirill Krinkin
kirill@krinkin.com

# References

- PROBABILISTIC ROBOTICS
- Introduction to Autonomous Mobile Robots
- SLAM Course by Cyrill Stachniss
- Stepic: Robot Operating System (a bit outdated)
- Multiple View Geometry in Computer Vision, Second Edition
- … many **more**

# Real world is real


Ken Goldberg
Roboticist,
Professor UC Berkeley