Introduction to Controls

RRC Summer School 2025 Day 5

Sarthak Mishra
MS CSE - Under Dr. Spandan Roy
RRC Lab Ground floor

What do we mean by Control?

The word **"control"** generally means:

As a verb (to control):

- To **direct**, **manage**, or **influence** something.
  - *Example: She controls the budget for the department.*
- To **restrain** or **regulate** behavior or action.
  - *Example: He tried to control his anger.*

As a noun (a control):

- The **power** or **ability** to direct or influence something.
  - *Example: The manager has full control over the project.*
- A **device** or **mechanism** used to operate something.
  - *Example: Use the remote control to turn on the TV.*

Why Control?

We control things to **achieve predictable outcomes** and **solve problems**.

- **Engineering:** Control systems (like thermostats or autopilots) help maintain stability and efficiency.
- **Computing:** We program machines to follow rules so they behave reliably.
- **Medicine:** Control of dosage, procedures, or interventions improves health outcomes.

Why? Because **without control, systems drift into chaos or inefficiency.**

Controls for Robotics

Motion Control

1. A robotic arm uses servo motors and PID controllers to precisely reach a target position.
2. A warehouse robot follows a planned path to avoid obstacles while delivering goods.
3. A self-balancing robot (like a Segway) uses gyroscopes to maintain upright posture.

Levels of Control in Robots - High, Low

🧠 High-Level Control (Strategic or Supervisory Layer)

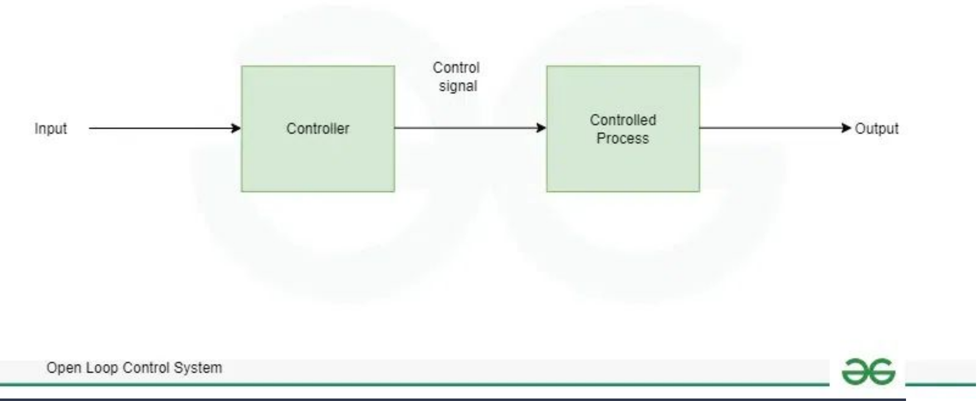
1. **Purpose:**
   - Handles **decision-making**, **planning**.
   - Focuses on **what** the robot should do.
2. **Examples of Functions:**
   - Path planning and trajectory generation
   - SLAM (Simultaneous Localization and Mapping)
   - Behavior selection and task scheduling
   - Object recognition and scene understanding
3. **Input/Output:**
   - Inputs: Global goals, environmental data, sensor maps
   - Outputs: Desired trajectories or high-level commands

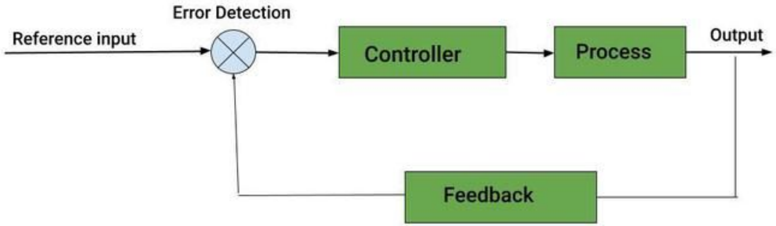
Levels of Control in Robots – High, Low

⚙️ Low-Level Control (Servo or Motor Control Layer)

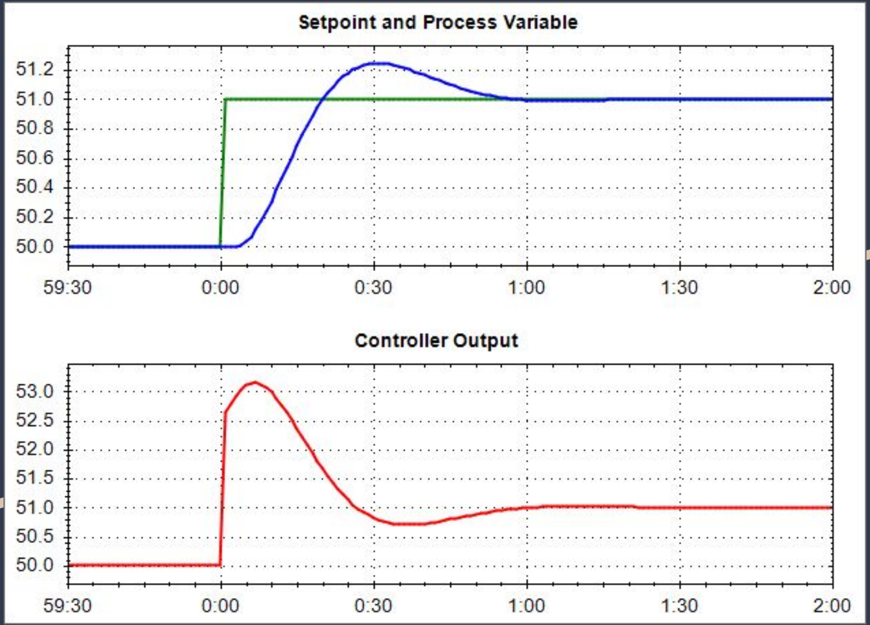
1. **Purpose:**
    - Manages **direct control of actuators** like motors and joints.
    - Focuses on **how** the robot executes the commands.
2. **Examples of Functions:**
    - PID control for motor speed/position
    - Joint torque or force control
    - Sensor feedback processing (e.g., encoder, IMU)
3. **Input/Output:**
    - Inputs: Commands from high-level controller (e.g., setpoint)
    - Outputs: PWM signals, motor currents, actuator positions

General Control System Structure





General Control System Components



1. Input Trajectory / Reference Signal

- **What we want the system to follow or achieve.**
- Examples:
  - A setpoint (constant target value).
  - A time-varying trajectory (e.g., a path for a robot to follow).

2. Controller

- **Purpose**: Computes control inputs to drive the system towards a desired state or trajectory.

3. Control Input

- **The actual input signal applied to the plant** (e.g., voltage to a motor).

4. Plant / System / Model

- **The physical process being controlled.**
- **Mathematical Model**:
  - Can be **linear** or **nonlinear**, **time-invariant** or **time-varying**.
  - Represented as: **State-space form**: x_dot = f(x,u)

5. Feedback

- **Essential for closed-loop control**.
- **Types**:
  - **State feedback**: Requires full knowledge of internal system states.
  - **Output feedback**: Uses only measured outputs.
- Helps correct deviations due to disturbances, noise, or model errors.

Where to apply Control?

In robotics, **control** is applied to influence a system's **state** toward a desired goal.

But what is a State?

Definition of System State

- A **state** is a set of measurable or observable quantities that describe the current condition of a dynamic system.
- It includes all information necessary to determine the future evolution of the system.

Properties of System States

- **Minimality**: The state should be the smallest possible set that fully describes the system.

- **Deterministic**: Knowing the current state and input, the next state is predictable (in deterministic systems).

- **Observable**: It should be possible to infer the internal state from outputs (observability).

- **Controllable**: The state should be alterable using control inputs (controllability).

- **Time-dependent**: States often evolve over time based on system dynamics.

Where to apply Control? - Examples

1. Drone (e.g., quadcopter)

Typical State Variables:

- Position: [x,y,z][x, y, z][x,y,z]
- Orientation (attitude): roll, pitch, yaw (or quaternion)
- Linear velocity: [vx,vy,vz][v_x, v_y, v_z][vx,vy,vz]
- Angular velocity: [ωx,ωy,ωz][ω_x, ω_y, ω_z][ωx,ωy,ωz]

Where to Apply Control:

- Attitude control (roll, pitch, yaw)
- Thrust control (altitude or vertical velocity)
- Position control in 3D space

Examples:

- Stabilizing hover → control attitude and vertical thrust
- Waypoint navigation → control x,y,zx, y, zx,y,z positions
- Yaw tracking → maintain a desired heading while flying

Where to apply Control? - Examples

2. Robotic Arm (e.g., 6-DOF manipulator)

Typical State Variables:

- Joint angles: $\theta_1, \theta_2, ..., \theta_n$
- Joint velocities: θ_dot_1,θ_dot_2,...,θ_dot_n
- End-effector position and orientation (via forward kinematics)

Where to Apply Control:

- Joint-space control (position or velocity control per joint)
- Cartesian-space control (control end-effector pose)
- Force/torque control at the end-effector

Examples:

- Pick-and-place → control joint positions or end-effector trajectory
- Applying force on a surface → control end-effector force/torque
- Painting/welding → follow a precise path in workspace

Where to apply Control? - Examples

3. 2-Wheel Differential Drive Robot

Typical State Variables:

- Position: [x,y]
- Orientation: θ
- Left and right wheel velocities: vL,vR

Where to Apply Control:

- Wheel velocities (differential control)
- Linear and angular velocity (v,ω)

Examples:

- Move forward/backward → control both wheels equally
- Rotate in place → control wheels in opposite directions
- Navigate to a target point → control both wheels for trajectory