PID controller

\[ \omega(t) = K_P e(t) + K_d \dot{e}(t) + K_i \int_{\tau=0}^{\tau=t} e(\tau) d\tau \]

Perhaps the most widely used controller in industry and robotics.
Perhaps the easiest to code.

You will also see it as:

\[ \omega(t) = K_P \dot{e}(t) + T_d \ddot{e}(t) + \frac{1}{T_i} \int_{\tau=0}^{\tau=t} e(\tau) d\tau \]

Tips: how to tune the PID

- Ziegler-Nichols heuristic:
  - First, use only the proportional term. Set the other gains to zero.
  - When you see consistent oscillations, record the proportional gain \( K_u \) and the oscillation period \( T_u \)

<table>
<thead>
<tr>
<th>Ziegler-Nichols method</th>
<th>( K_P )</th>
<th>( T_i )</th>
<th>( T_d )</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>( 0.5K_u )</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PI</td>
<td>( 0.45K_u )</td>
<td>( T_u/1.2 )</td>
<td>-</td>
</tr>
<tr>
<td>PD</td>
<td>( 0.8K_u )</td>
<td>-</td>
<td>( T_u/8 )</td>
</tr>
<tr>
<td>classic PID(( I ))</td>
<td>( 0.6K_u )</td>
<td>( T_u/2 )</td>
<td>( T_u/8 )</td>
</tr>
</tbody>
</table>

Physical models of how systems move

Main question: what is the next state given the current state and controls?
Today

• Idealized physical models of robotic vehicles

Why idealized?

• “All models are wrong, but some are useful” – George Box (statistician)

• Model: a function that describes a physical phenomenon or a system, i.e. how a set of input variables cause a set of output variables.

• Models are useful if they can predict reality up to some degree.

• Mismatch between model prediction and reality = error / noise

Noise

• Anything that we do not bother modelling with our model

• Example 1: “assume frictionless surface”

• Example 2: Taylor series expansion (only first few terms are dominant)

• With models, can be thought of as approximation error.

Idealized physical models of robotic vehicles

• Omnidirectional motion
• Dubins car
• Differential drive steering
• Ackerman steering
• Unicycle
• Cartpole
• Quadcopter
The state of an omnidirectional robot

State := Configuration := \( \mathbf{x} := \) vector of physical quantities of interest about the system

\[
\mathbf{x} = \begin{bmatrix} Gp_x, Gp_y, G\theta \end{bmatrix}
\]

Position of the robot's frame of reference \( \mathbf{C} \) with respect to a fixed frame of reference \( \mathbf{G} \), expressed in coordinates of frame \( \mathbf{G} \). Angle is the orientation of frame \( \mathbf{C} \) with respect to frame \( \mathbf{G} \).

Control of an omnidirectional robot

Control := \( \mathbf{u} := \) a vector of input commands that can modify the state of the system

\[
\mathbf{u} = \begin{bmatrix} Cv_x, Cv_y, C\omega_z \end{bmatrix}
\]

Linear and angular velocity of the robot’s frame of reference \( \mathbf{C} \) with respect to a fixed frame of reference \( \mathbf{G} \), expressed in coordinates of frame \( \mathbf{C} \).
**Dynamics** of an omnidirectional robot

Dynamical System := Dynamics := a function that describes the time evolution of the state in response to a control signal.

\[
\frac{dx}{dt} = \dot{x} = f(x, u)
\]

Continuous case:

\[
\begin{align*}
\dot{p}_x &= v_x \\
\dot{p}_y &= v_y \\
\dot{\theta} &= \omega_z
\end{align*}
\]

Note: reference frames have been removed for readability.

---

**Inertial frames of reference**

- **G**, the global frame of reference is fixed, i.e. with zero velocity in our previous example.
- But, in general it can move as long as it has zero acceleration. Such a frame is called an “inertial” frame of reference.
- Newton’s laws hold for inertial reference frames only. For reference frames with non-constant velocity we need the theory of General Relativity.
- So, make sure that your global frame of reference is inertial, preferably fixed.

---

**The state of a simple car**

State = [Position and orientation]

Position of the car’s frame of reference $C$ with respect to a fixed frame of reference $G$, expressed in frame $G$.

The angle is the orientation of frame $C$ with respect to $G$.

\[
x = [Gp_x, G\dot{p}_y, G\theta]
\]

**The controls of a simple car**

Controls = [Forward speed and angular velocity]

Linear velocity and angular velocity of the car’s frame of reference $C$ with respect to a fixed frame of reference $G$, expressed in coordinates of $C$.

\[
u = [Cv_x, C\omega_z]
\]
The dynamical system of a simple car

\[ \begin{align*}
\dot{p}_x &= v_x \cos(\theta) \\
\dot{p}_y &= v_x \sin(\theta) \\
\dot{\theta} &= \omega_z
\end{align*} \]

Note: reference frames have been removed for readability.

Kinematics vs Dynamics

- Kinematics considers models of locomotion independently of external forces and control.
- For example, it describes how the speed of a car affects the state without considering what the required control commands required to generate those speeds are.
- Dynamics considers models of locomotion as functions of their control inputs and state.

Special case of simple car: the "Dubins vehicle"

- Can only go forward
- Constant speed
- You only control the angular velocity

Special case of simple car: Dubins car

- Can only go forward
- Constant speed
- You only control the angular velocity
Dubins car: motion primitives

- The path of the car can be decomposed to L(eft), R(ight), S(traight) segments.

Dubins car: Dubins boat

- Why do we care about a car that can only go forward?
- Because we can also model idealized airplanes and boats
- Dubins boat = Dubins car

Dubins car: Instantaneous Center of Rotation

IC = Instantaneous Center of Rotation
The center of the circle circumscribed by the turning path.
Undefined for straight path segments.

Pitch angle $\phi$ and forward velocity determine descent rate
Yaw angle $\theta$ and forward velocity determine turning rate

Dubins airplane in 3D

- $\dot{\phi} = v_x \cos(\theta) \sin(\phi)$
- $\dot{\theta} = v_x \cos(\theta) \sin(\phi)$
- $\dot{\phi} = v_x \cos(\phi)$
- $\dot{\phi} = \omega_y$

$\theta$ is yaw
$\phi$ is pitch
Holonomic constraints

• Equality constraints on the state of the system, but not on the higher-order derivatives:
  \[ f(\mathbf{x}, t) = 0 \]

• For example, if you want to constrain the state to lie on a circle:
  \[ ||\mathbf{x}||^2 - 1 = 0 \]

• Another example: train tracks are a holonomic constraint.

Non-holonomic constraints

• Equality constraints that involve the derivatives of the state (e.g. velocity) in a way that it cannot be integrated out into holonomic constraints, i.e.
  \[ f(\mathbf{x}, \dot{\mathbf{x}}, t) = 0 \]
  but not
  \[ f(\mathbf{x}, t) = 0 \]

The Dubins car is non-holonomic

• Dubins car is constrained to move straight towards the direction it is currently heading. It cannot move sideways. It needs to “parallel park” to move laterally.

• In a small time interval \( dt \) the vehicle is going to move by \( \delta \mathbf{p}_x \) and \( \delta \mathbf{p}_y \) in the global frame of reference. Then from the dynamical system:

  \[
  \begin{align*}
  \delta \mathbf{p}_x \sin(\theta) &= v_x \cos(\theta) \sin(\theta) \\
  \delta \mathbf{p}_y \cos(\theta) &= v_x \sin(\theta) \cos(\theta)
  \end{align*}
  \]

  \[
  \begin{align*}
  \delta \mathbf{p}_x \sin(\theta) &= \delta \mathbf{p}_x \cos(\theta) \\
  v_x \sin(\theta) &= v_y \cos(\theta)
  \end{align*}
  \]

  Car is constrained to move along the line of current heading, i.e. non-holonomic.

The state of a unicycle

\[ \mathbf{x} = \left[ ^G p_x, ^G p_y, ^G \theta \right] \]

State = [Position, Orientation]
Position of the unicycle’s frame of reference \( \mathbf{U} \) with respect to a fixed frame of reference \( \mathbf{G} \), expressed in coordinates of frame \( \mathbf{G} \). Angle is the orientation of frame \( \mathbf{U} \) with respect to frame \( \mathbf{G} \).

Q: Would you put the radius of the unicycle to be part of the state?
The state of a unicycle

\[ \mathbf{x} = [G\, p_x, \, G\, p_y, \, G\, \theta] \]

State = [Position, Orientation]
Position of the unicycle's frame of reference U with respect to a fixed frame of reference G, expressed in coordinates of frame G. Angle is the orientation of frame U with respect to frame G.

Q: Would you put the radius of the unicycle to be part of the state?
A: Most likely not, because it is a constant quantity that we can measure beforehand. But, if we couldn't measure it, we need to make it part of the state in order to estimate it.

Controls of a unicycle

\[ \mathbf{u} = [U\, \omega_z, \, U\, \omega_y] \]

Controls = [Yaw rate, and pedaling rate]
Yaw and pedaling rates describe the angular velocities of the respective axes of the unicycle's frame of reference U with respect to a fixed frame of reference G, expressed in coordinates of U.

Dynamics of a unicycle

\[
\begin{align*}
\dot{p}_x &= r \omega_z \cos(\theta), \\
\dot{p}_y &= r \omega_z \sin(\theta), \\
\dot{\theta} &= \omega_z
\end{align*}
\]

\( r \) = the radius of the wheel
\( r \omega_y \) is the forward velocity of the unicycle

The state of a differential drive vehicle

\[ \mathbf{x} = [G\, p_x, \, G\, p_y, \, G\, \theta] \]

State = [Position, Orientation]
Position of the vehicle's frame of reference D with respect to a fixed frame of reference G, expressed in coordinates of frame G. Angle is the orientation of frame D with respect to frame G.
Controls of a differential drive vehicle

\[ \mathbf{u} = [D \omega_l, D \omega_r] \]

Controls = [Left wheel and right wheel turning rates]
Wheel turning rates determine the linear velocities of the respective wheels of the vehicle’s frame of reference D with respect to a fixed frame of reference G, expressed in coordinates of D.

- \( v_l = (W - H/2) \omega_l \)
- \( v_r = (W + H/2) \omega_r \)
- \( v_x = (v_l + v_r)/2 \)

Dynamics of a differential drive vehicle

Special cases:
- moving straight
- in-place rotation
- rotation about the left wheel

The state of a double-link inverted pendulum (a.k.a. Acrobot)

\[ \mathbf{x} = [\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2] \]

State = [angle of joint 1, joint 2, joint velocities]
Angle of joint 2 is expressed with respect to joint 1. Angle of joint 1 is expressed compared to down vector.
Controls of a double-link inverted pendulum (a.k.a. Acrobot)

\[ u = \begin{bmatrix} \tau_1 \end{bmatrix} \]

Controls = [torque applied to joint 1]

Dynamics of a double-link inverted pendulum (a.k.a. Acrobot)

Provided here just for reference and completeness. You are not expected to know this.

The state of a single-link cartpole

\[ x = \begin{bmatrix} Gp_x, G\dot{p}_x, G\theta, G\dot{\theta} \end{bmatrix} \]

State = [Position and velocity of cart, orientation and angular velocity of pole]
Controls of a single-link cartpole

Controls = [Horizontal force applied to cart]

Balancing a triple-link pendulum on a cart

Extreme Balancing

The state of a double integrator

State = [Position along x-axis]
Controls of a double integrator

\[ \mathbf{u} = [G_u, u_x] \]

This corresponds to applying force to a brick of mass 1 to move on frictionless ice. Where is the brick going to end up? Similar to curling.

Dynamics of a double integrator

\[ \dot{\mathbf{x}} = \mathbf{F} \]

The state of a quadrotor

\[ \mathbf{x} = [G\phi, G\theta, G\psi, G\dot{\phi}, G\dot{\theta}, G\dot{\psi}] \]

State = [Roll, pitch, yaw, and roll rate, pitch rate, roll rate]

Angles are with respect to the global frame.

Controls of a quadrotor

\[ \mathbf{u} = [T_1, T_2, T_3, T_4] \]

Controls = [Thrusts of four motors]

OR

\[ \mathbf{u} = [M_1, M_2, M_3, M_4] \]

Controls = [Torques of four motors]

Notice how adjacent motors spin in opposite ways. Why?
What if all four motors spin the same direction?

Controllability

- A system is controllable if there exist control sequences that can bring the system from any state to any other state, in finite time.
- For example, even though cars are subject to non-holonomic constraints (can’t move sideways directly), they are controllable. They can reach sideways states by parallel parking.

Dynamics of a quadrotor

Passive Dynamics

- Dynamics of systems that operate without drawing (a lot of) energy from a power supply.
- Interesting because biological locomotion systems are more efficient than current robotic systems.
Passive Dynamics

- Dynamics of systems that operate without drawing (a lot of) energy from a power supply.
- Usually propelled by their own weight.
- Interesting because biological locomotion systems are more efficient than current robotic systems.