Control strategies

We need to be in control of things

TECHNICAL UNIVERSITY OF CLUJ-NAPOCA ROMANIA

November 11, 2018

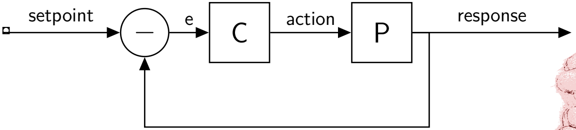
Agenda

- Control theory background
- Actuator dynamics
- Independent joint control
- Computed torque control
- Force control

Control theory

Feedback loops

The robot is a process, and if we want to accomplish some tasks, we need to be able to control its various aspects.

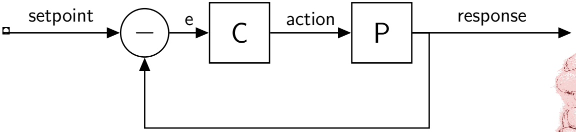


A simple process with feedback

Control theory

Feedback loops

The robot is a process, and if we want to accomplish some tasks, we need to be able to control its various aspects.



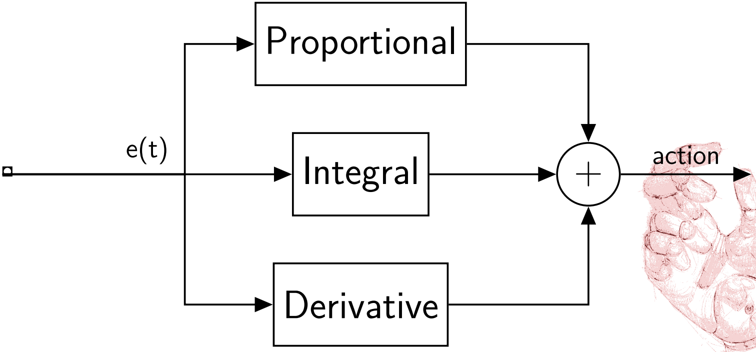
A simple process with feedback

What is the setpoint? What is the response? What is the action?

Control theory

PID controller

One of the most basic, robust and used controller of all times!

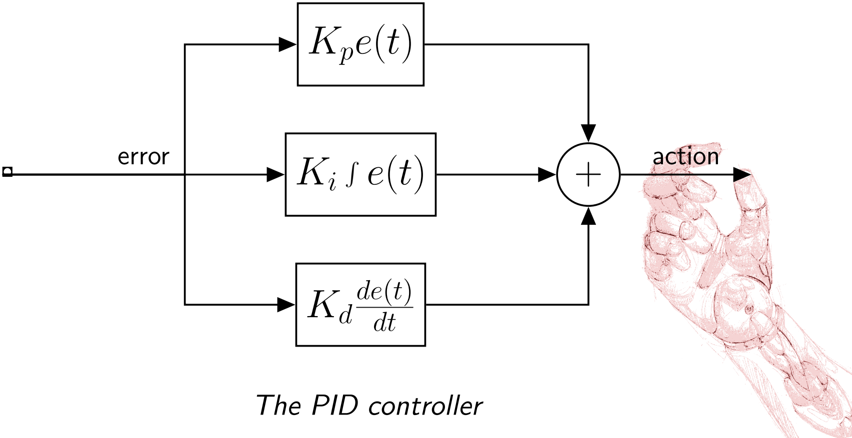


The PID controller

Control theory

PID controller

One of the most basic, robust and used controller of all times!



The PID controller

Control theory

PID controller

How could we use such a controller for an articulated robot?

Control theory

PID controller

How could we use such a controller for an articulated robot?

- We could define the joint coordinates as input and end-effector pose as our output.

Control theory

PID controller

How could we use such a controller for an articulated robot?

- We could define the joint coordinates as input and end-effector pose as our output.
- We could solve the inverse kinematics/velocities and control each joint individually.

Control theory
PID controller

How could we use such a controller for an articulated robot?

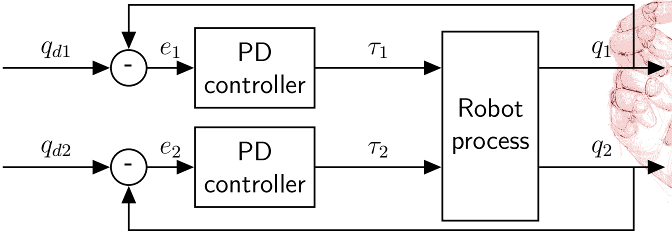
- We could define the joint coordinates as input and end-effector pose as our output.
- We could solve the inverse kinematics/velocities and control each joint individually.

What are the pros/cons of each solution?

Robotic controllers

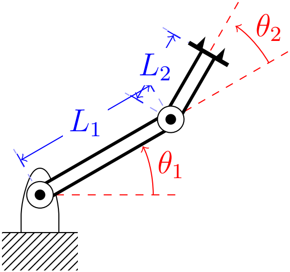
Independent joint control

With this control strategy, we control each joint individually. If we are controlling e.g. position, then we need to solve the inverse kinematics to define the joint coordinates. These are then used as our setpoints.



Robotic controllers

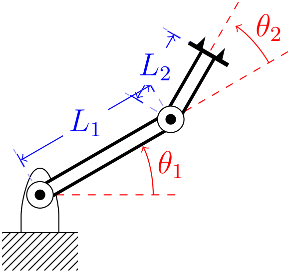
Independent joint control



The independent joint control is considering that each joint moves independently and can therefore be controlled separately. Under which cases is this assumption correct?

Robotic controllers

Independent joint control



The independent joint control is considering that each joint moves independently and can therefore be controlled separately. Under which cases is this assumption correct?

The effects of inertia need to be 'small'!

Robotic controllers

Independent joint control

The independent joint control can take us rather far as long as:

- The motions performed are slow.
- If this is the case, then each controller can deal with the other joints motion as disturbances.
- We tune each controller diligently.

Independent joint control

Tunning the parameters

To tune the PID parameters analytically, we need to write a transfer function for each joint coordinate. To do that, we need the equation of motion for each joint.

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau$$

Independent joint control

Tunning the parameters

To tune the PID parameters analytically, we need to write a transfer function for each joint coordinate. To do that, we need the equation of motion for each joint.

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau$$

Can we 'decouple' the joint coordinates?

Independent joint control

Tunning the parameters

To tune the PID parameters analytically, we need to write a transfer function for each joint coordinate. To do that, we need the equation of motion for each joint.

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau$$

Can we 'decouple' the joint coordinates?
Why not?

Independent joint control

Tunning the parameters

If we want to decouple the joint coordinates, we need for each joint to consider the effects from the motion of the other joints as disturbances.

$$d_{ii}\ddot{q}_i + c_{ii}\dot{q} = \tau - w$$

Independent joint control

Tunning the parameters

If we want to decouple the joint coordinates, we need for each joint to consider the effects from the motion of the other joints as disturbances.

$$d_{ii}\ddot{q}_i + c_{ii}\dot{q} = \tau - w$$

Where the term w contains all the off diagonal elements from matrices D and C, and the gravity terms G.

Independent joint control

Tunning the parameters

If we consider a PD controller, then the input signal becomes:

$$\tau_i = K_{Di}\dot{e}_i + K_{Pi}e_i$$

And the equation of motion becomes:

$$d_{ii}\ddot{q}_i + (c_{ii} + K_{Di})\dot{q}_i + K_{Pi}q_i = K_{Pi}q_{di} - w$$

If we consider:

$$e_i = q_{di} - q_i, \dot{e}_i = \dot{q}_{di} - \dot{q}_i, \dot{q}_{di} = 0$$

Independent joint control

Tunning the parameters

$$d_{ii}\ddot{q}_i + (c_{ii} + K_{Di})\dot{q}_i + K_{Pi}q_i = K_{Pi}q_{di} - w$$

This represents a second-order system, which you should know how to calculate PD parameters for a stable fast response.

Independent joint control

Tunning the parameters

$$d_{ii}\ddot{q}_i + (c_{ii} + K_{Di})\dot{q}_i + K_{Pi}q_i = K_{Pi}q_{di} - w$$

This represents a second-order system, which you should know how to calculate PD parameters for a stable fast response.

What happens when our assumptions are not met?

Control theory

System linearization

Starting from the dynamic model of the robot:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = u$$

We are seeking an input function that can convert this model into a linear closed loop system.

Control theory

System linearization

Starting from the dynamic model of the robot:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = u$$

We are seeking an input function that can convert this model into a linear closed loop system.

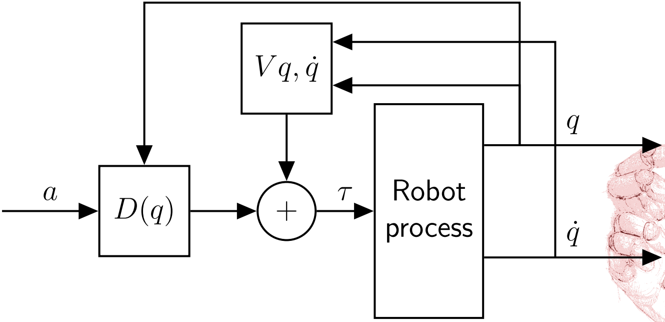
What about this one:

$$u = D(q)a + C(q,\dot{q})\dot{q} + G(q)$$

$$\ddot{q} = a$$

Control theory

System linearization



Control theory

System linearization

We can therefore 'linearize' our system, and then control it using state-space feedback for a linear system!

Control theory

System linearization

We can therefore 'linearize' our system, and then control it using state-space feedback for a linear system! What should our input

for the new system be? (i.e. a)

Control theory

Computed torque control

An obvious input would be:

$$a = -K_0 q - K_1 \dot{q} + r$$

And the closed loop form of our system becomes:

$$\ddot{q} + K_1 \dot{q} + K_0 q = r$$

Where r is our reference.

Control theory

Computed torque control

An obvious input would be:

$$a = -K_0 q - K_1 \dot{q} + r$$

And the closed loop form of our system becomes:

$$\ddot{q} + K_1 \dot{q} + K_0 q = r$$

Where r is our reference. Choosing an r to follow the desired trajectories of $q, \dot{q}, \ddot{q}$ like this:

$$r(t) = \ddot{q}^d(t) + K_0 q^d(t) + K_1 \dot{q}^d(t)$$

We end up with zero tracking error.

Control theory

Computed torque control

How do we calculate the desired $q, \dot{q}, \ddot{q}$?

Questions?