- start by showing a short video of Sofie balancing some pen

- (turning around from a chair)
- amazing, Sofie! you are an automatic control girl!
- hello, I am Steffi, and today we are continuing our learning how to make a robot balance on its wheels

Making robots balance
Part 2

What are we doing today?

- show how making the pen stand is similar to making the robot stand
- make a demo of a nice pre-made controller that we know it works – but we won't say how it works at the beginning
- we will discover how it works little by little, and eventually discover the simplest, but probably the most important strategy: the P controller!

Our purposes

Our purposes

- have fun

Our purposes

- have fun
- understand the world a bit better

Our purposes

- have fun
- understand the world a bit better
- see that math is useful

What is going to happen in today's part

What is going to happen in today's part

- connect the pen-balancing and the robot-balancing problems

What is going to happen in today's part

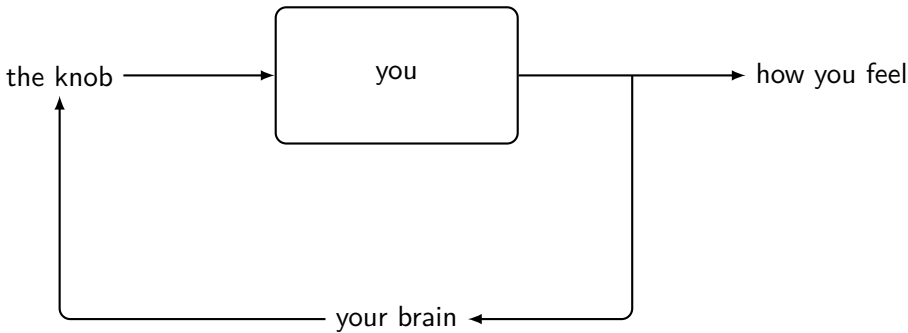
- connect the pen-balancing and the robot-balancing problems
- discover together what is the "P controller"

Connecting things

- the robot: this is something that connects to the pen-balancing problem before
- also the robot tips
- but then the wheels act like the hands!
- so the intuition is to make the wheels spin as one would move the hands
- but just a moment: when we were balancing the pen we were first of all seeing what was happening, and then also thinking at what to do!
- where are the eyes here? and where is the brain?

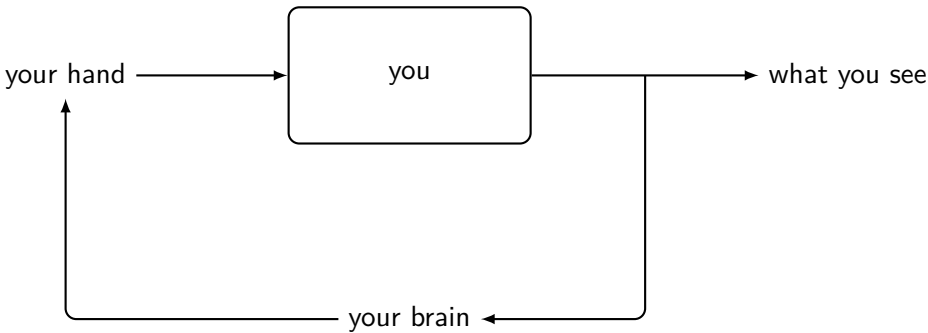
Connecting the robot with the pen - a small recap, first

The case of the shower



Connecting the robot with the pen - a small recap, first

The case of the balancing pen



Connecting the robot with the pen - what is what

you | *the robot*

Connecting the robot with the pen - what is what

you

your hands

the robot

the wheels (and the motors)

Connecting the robot with the pen - what is what

you

your hands

your eyes

the robot

the wheels (and the motors)

sensors that measure “how tilted”

Connecting the robot with the pen - what is what

you

your hands

your eyes

your brain

the robot

the wheels (and the motors)

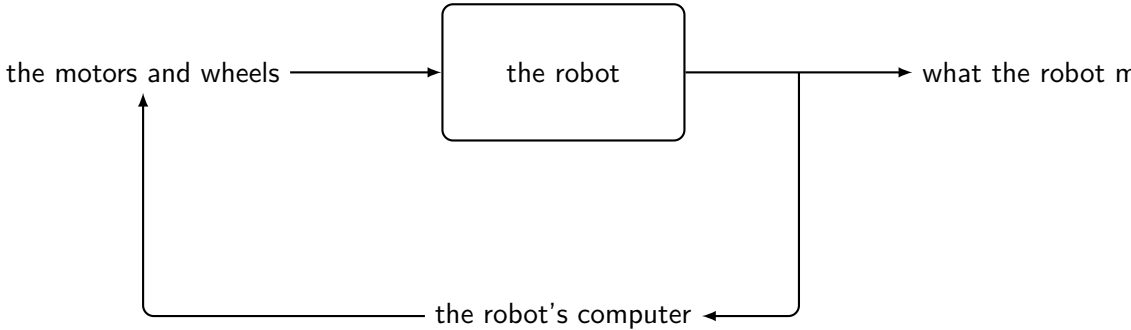
sensors that measure “how tilted”

a small embedded computer

What is what

- let's see what is what in the robot
- the wheels are these ones
- the motors are these ones
- the sensors are these ones - there are similar ones in the smartphones
- the embedded computer is this one
- all the rest is stuff that is needed to bring the power here and there
- if you find this fascinating then consider becoming a electronics engineer!

The robot as a block scheme



What is what

- but now I have been talking too much!
- let's do as following: now I give you a pre-made brain, and we see if this brain work
- I don't tell you how I have created it, I just want you to try it so to see if things work and get something moving

Tryout pause!

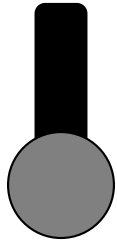
- follow the instructions in the “tryout A” manual
- does the robot stand on its own?

post-experience discussion

- hope things worked!
- so now let's discover the most important part of the brain we just inserted in the computer, the famous "P controller"
- we will arrive at it little by little, listing and testing some heuristics control strategies, and eventually arriving to the P controller

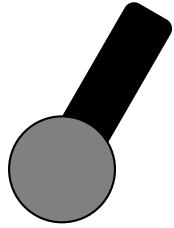
A first heuristic

A first heuristic



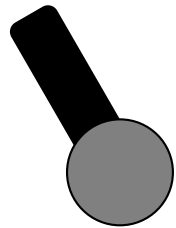
$\Longrightarrow$ do nothing

A first heuristic



$\Longrightarrow$ spin the wheels as fast as possible clockwise

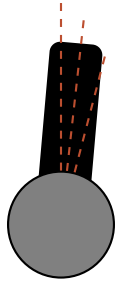
A first heuristic



$\Longrightarrow$ spin the wheels as fast as possible counterclockwise

- this approach is problematic
- this controller is “too nervous”: as soon as we are not perfectly aligned then it reacts as much as it can
- let me show what this means in practice with the problem of making the pen stand
- (example)
- see? this 'brain', this controller, does not have too much sense
- let's make another heuristic

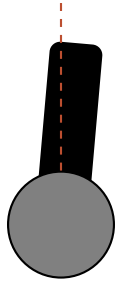
A second heuristic



$\Longrightarrow$ depending on the zone, spin the wheels more or less fast

- also this approach is a bit problematic
- when the system is traversing the transition zones the behavior is 'bumpy': sudden increase, sudden decrease
- did you ever went in a car with somebody that drives turning the wheel like in this way? It is kind of a similar thing, you see what I mean? And it is not so comfortable, true?
- I would say i prefer somebody that drives smoothly, that moves the wheel in a fluid way, without abrupt changes
- and this is the intuition that leads us to the P controller

The P (i.e., proportional) controller



$\Longrightarrow$ speed of the wheels $= P \cdot$ angular error

- but how do we design P?
- to understand we should ask ourselves: what is the effect of having different Ps?
- if I put P = a tiny tiny number, what is going to happen?
- if I put P = a huge number, what is going to happen?
- let's have a chat all together, and see what we think about this

Discussion pause!

- if I put P = a tiny tiny number, what is going to happen?
- if I put P = a huge number, what is going to happen?

- amazing, I am proud of your chatting!
- but I will leave you with a cliffhanger; we are going to discuss the effects of choosing P in the next part
- so, be ready to move to the next part of the experience! See you in 'part 3', where we add two ingredients to this controller, and then make our own version of the robot brain!