Robotics Zusammenfassung

Robot word origin

robota → Czech for forced labor

introduced 1920 in a written play by Karel Čapek

robot → an artificial entity built specifically for implementing a practical task or function

What is a Robot

↳ some form of intelligence & embodied (some physicality)

History of Robots

↳ 250 BC → first automated water clock

280 - 220 BC → first humanoid → wine pouring maid

1 - 2nd Century AD → first building automation (door opener w/steam)

1206 → water powered drumming band

1495 → Da Vinci (mechanical knight, self propelled car)

Automation

↳ auto + mato ⇒ automated device that acts by itself

auto = itself, mato = thinking

clockwork to automatas

↳ clocks helped organice cities in middle age

automatas → drive to simulate lifes
  ↳ swiss watch makers became famous for automatas
    ↳ expensive
    ⇒ drawing automatas, playing music (dolls)

Perception
↳ is hard

Morevac's Paradox
↳ reasoning requires little computation, but sensorimotor and perception skills require enormous computational resources
  ↳ robots could play chess before they could walk

Robot Loop
Sense → Think → Act

↳ Robot state → current config

Robot state space → all possible states

observable State → fully visible states of robot

partially observable states → hidden states

↳ States can be descrete (on/off) or continuous (rotor speed ...)

Sensors

Proprioceptive → measures values internal to the robot

↳ gyroscope, accelerometer ...

Extroceptive → provide information about environment

↳ camera, heat detection, ultrasonic distance...

Passive vs Active Sensors

Passive

↳ measure environmental energy entering the sensor

↳ temperature, microphone, camera

Active

↳ emit energy into environment & measures the reaction

↳ RADAR, LIDAR

RADAR: emits radio waves and analyzing reflections

LIDAR: emits laser beams & measures time it takes to bounce back

Computer Vision

↳ visually perceive & interact with the environment

→ extract information from images

⇒ Robotic vision → translate extracted information to action

⇒ we see an image, a computer sees an applied computation

↳ ex. Matrix of the brightness of each pixel

Cameras

↳ directional Sensor → direction from where the light is hitting it

photometric Sensor → details about the light (brightness & color)

A digital image is a 2D array of pixels that is formed by focusing light onto a 2D array of sensing elements

Digital Camera

Light hits Pixel → charge generation → charge collection → signal amplification → charge to voltage conversion → Signal amplification → digital conversion

Images

Binary Images → pixels are one byte (black/white)

Grey scale Images → pixels represent intensity of light (0-255)

Color Images → three color layers stacked, each pixel represents light intensity of the color (0-255)

Bayer Filter

↳ color fille array in most digital cameras

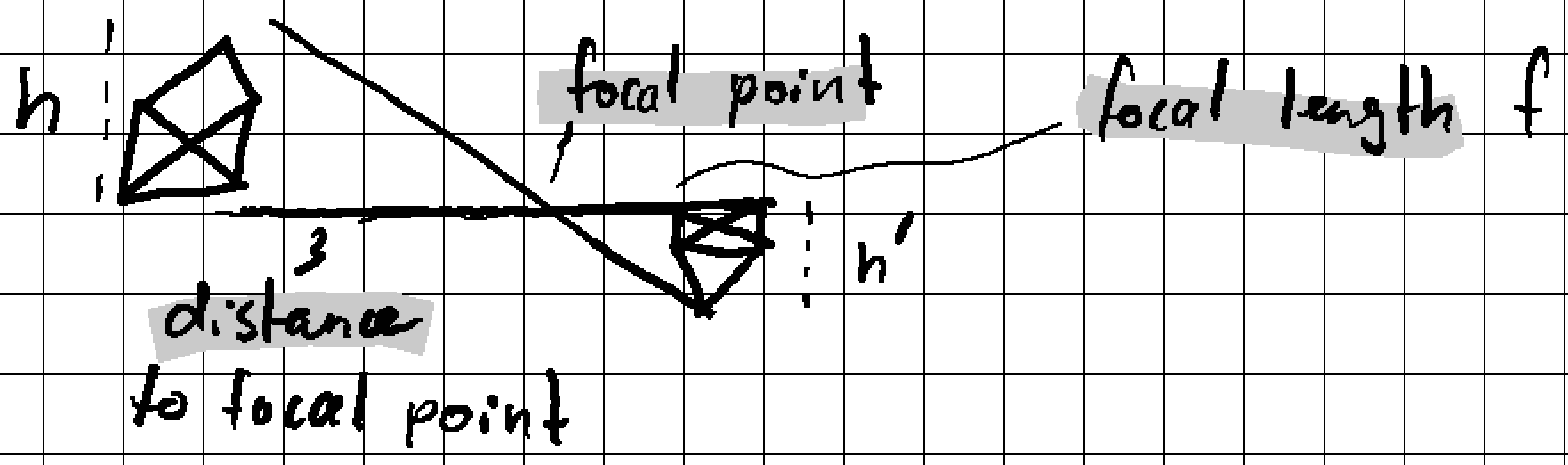
  ↳ mosaic pattern of RGB sensors (50% more green sensors)

→ each pixel is 1 color

Demosaicing → process pixel based on neighboring pixels for full colored image

Optics

↳ Image size is inversly proportional to the distance from the focal point



Stereo Vision → two cameras can be used to create depth

Image Processing

Binarization → change pixel intensity to 1/0

Thresholding → pixel is 1/0 based on a fixed μ

$\text{pixel} < \mu = 0 \;||\; \text{pixel} > \mu = 1$

Thresholding with Otsu Method

- ↳ try every possible threshold
  - ↳ for each value calculate how well the pixels are seperated foreground & background
  - ↳ choses threshold that minimizes the overlap (maximum contrast)

Find intensity I_t which maximizes the between class variance σ_B^2

$$\sigma_B^2 = W_b W_f (\mu_b - \mu_f)^2$$

W_b → Nr of Pixel in the Background / total pixels

W_f → Nr of Pixel in the Foreground / total pixels

μ_b → Mean intensity of Background

μ_f → Mean intensity of Foreground

⇒ works only well, if the histogram has a bimodial distribution

Connectivity

↳ Distinguish multiple objects in an image

Image segmentation

↳ partition segments based on similar pixel values

↳ eighter binary values or intensity value

0	0	0	0	0	0	0
1	1	0	0	1	0	0
1	1	0	0	1	0	0
0	1	0	0	0	0	0

Moments

↳ used to summarice various aspects of an image

(ex. radius, area, centroid ...)

Smoothing & Bluring

Low-& High-Pass Filtering

Low-pass → lets low frequncies through

↳ smooths and removes detail/noise

high-pass → lets high frequencies through

↳ retains contours, enheces edges

Frequncies → high → rapid change in intensity

Low → slow, gradual change in intensity

Convolution

↳ moving a mask over an image
  ↳ 1D / 2D

Average kernel (Box Blur)

↳ each pixel in the resulting image has a value equal to the average of its neighbours

Gaussion Function to Blur

↳ weighted blur, were the closest neighbouring pixel have more impact than pixels that are further away.

Median Filter

↳ preserves sharp transitions but removes small variations in brightness
  ↳ usefull to reduce noise
    ↳ look at neighbour pixel and apply Median to each pixel

Bilateral Filter

↳ Smooth an image, while preserving edges
  ↳ replaces each pixel value with a weighted average of nearby pixel values, taking into account the variation in intensity of the pixels to preserve edges

Edge Detection

↳ regions were a sudden change in intensity occurs

step edge → abrupt change

ramp edge → gradual transition

roof edge → increase to maximum, then decreases

First Derivate

↳ rate of change at any instant

Sobel Operator

↳ uses two small filters to calculate gradients in the horizontal and vertical directions

→ these gradients highlight edges

↳ strong changes → edges

small changes → flat areas

Second Derivate

↳ rate of change of the rate of change

  ↳ how quick does the steepness change

Laplacian Operator

↳ measures how much the intensity changes around a point in both directions

  ↳ when there is a sharp change, the second derivative becomes large

Canny Edge Detector

1. apply gausian filter to reduce noise
2. Find intensity gradiants
3. apply gradient magnitude thresholding
4. thin eges to one pixel
5. apply double threshold to determin potential edges
6. track edge by histeresis: suppress weak edges
   ↳ transform weak into strong pixels if at least one direct neighbour is a strong one.

Industrial Robots

→ rigid and accurate
  ↳ good for positioning tasks

Types

- **Gantry Robots** → linear robots for large & heavy lifting
- **Cylindrical Robots** → range of motion is a cylinder
- **Spherical Robot** → range of motion is a sphee (angle joint)
- **SCARA Robot** → asembly line robot arm rigid on Z axis
- **Parallel Robot** → parallelograms connected to a common base
  ↳ rigid, light weight, fast

Applications

- Machine tending transfer → transfer workpieces from/to machines
- Palletize
- Pick and Place
- ASeembly
- welding
- painting
- milling & drilling

Components of a robot

- Joints
- Sensors
- Transmission & gear boxes
- Actuators
- wrists
- end effectors

End Effectors

- Grippers → external → close to grab
  internal → open to grab

- Process Tool → process a work piece
  ↳ paint, weld, vacum cups ...

- Hands

Robot Kinematics

Degrees of Freedom

↳ the number of independent movements an object can make with respect to a coordinate system

⇒ a robotic arm needs **6 DOF** to specify the position and orientation of an end-effector

- → 3 DOF for positioning (x, y, z)
- → 3 DOF for orientation (yaw, pitch, roll)
  - ⇒ more then 6 are redundant as we need only 6. → Human arm is consideed kinematically redundant

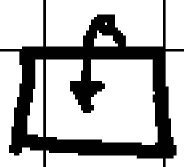
Spherial wrist

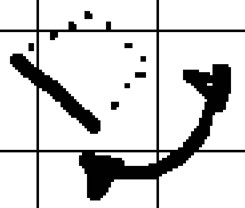
↳ decouple positioning from orientation

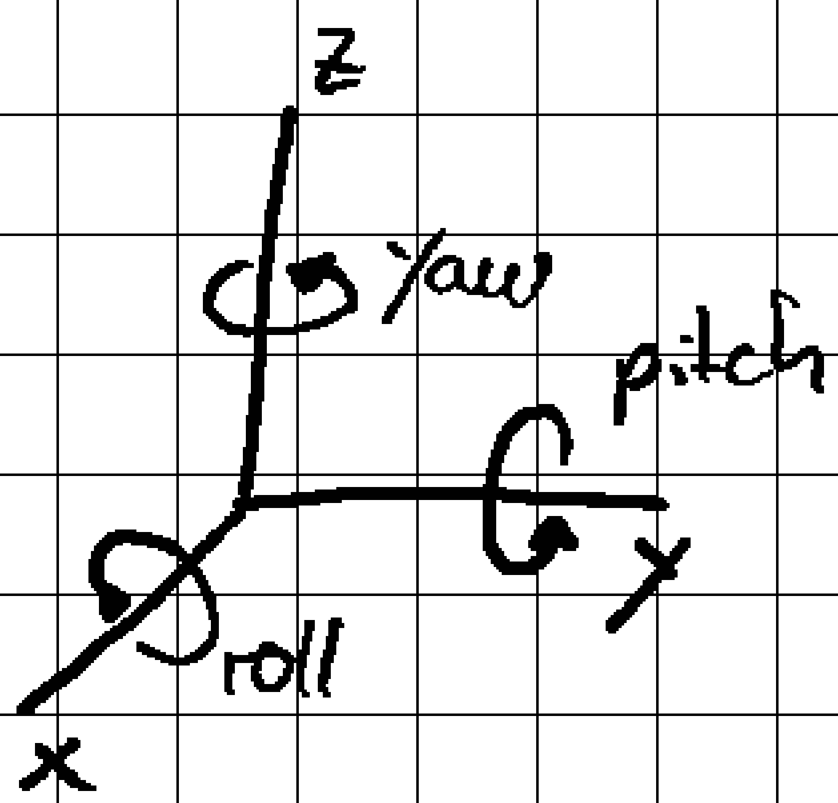
position with joints

orient with wrist

roll:  (twist)

pitch:  (tilt up / down)

yaw:  (turn left / right)



Robot Joints

- Prismatic → sliding motion (P)
- Revolute → rotation (R)

↳ with this we can describe a robot:

ex: 3 rotating joints → RRR-Robot

Forward kinematics

↳ given a robots joints angles, what is the position and orientaton of the end-effector?

Calculate Forward kinematics

→ compute transformation Matrices for each joint

→ multiply matrices to find end position

Homogenous Transformation Matrix

↳ put rotational Matrix & displacement Matrix together

Denavit - Hartenberg Convention

↳ standardized way to describe the geometry of a robots joints and links, using 4 parameters

Parameters:

- Link Length a → distance between joints along x-axis
- Link Twist α → angle between joint axes
- Joint offset d → distance along current joint axis to the links x-axis
- Joint angle θ → rotation around the current z-axis

How to:

- Assign coordinate Frame to each joint
- use the 4 parameters to describe the relative position and orientation of one joint's frame to the next
- combine transformations to compute the end-effectors position.

Inverse Kinematics

↳ given a position and orientation of the end-effector, what angles have the robot joints to be at?

Calculate angles

- look from a 45° angle
- identify helpful triangles
- compute angles of triangles
  - ↳ Phytagoras
  - ↳ Trigonometry
  - ↳ Law of Cosines

Robot Control

↳ motion control, force control, hybrid motion force control

Motion Control

↳ described in a task or cartesian space

↳ trajectory-following control is performed in the joint space

Task / Cartesian space

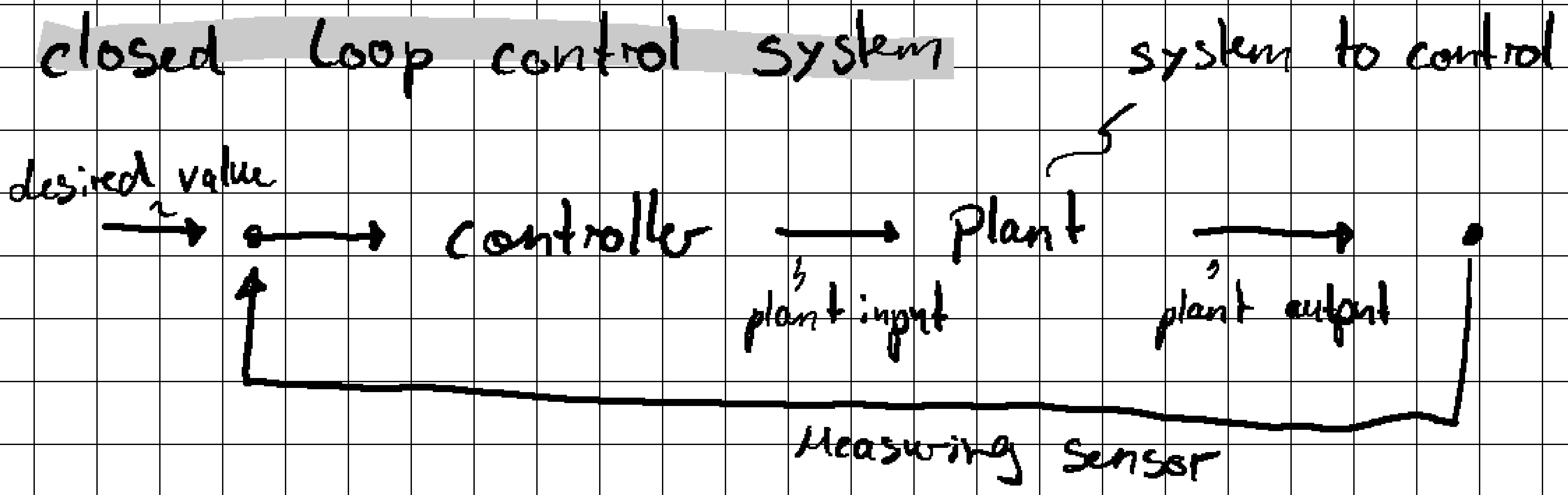
↳ defined by the position and orientation of the end-effector

Joint Space

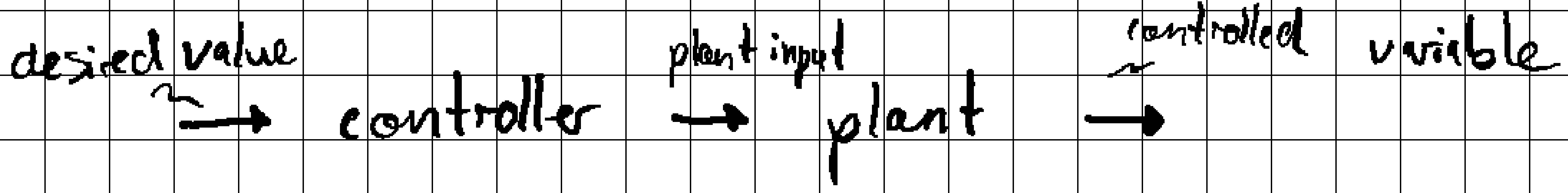
↳ defined by a vector whose components are the translational and angular displacements of each joint of a robotic link

Control System

closed Loop control system



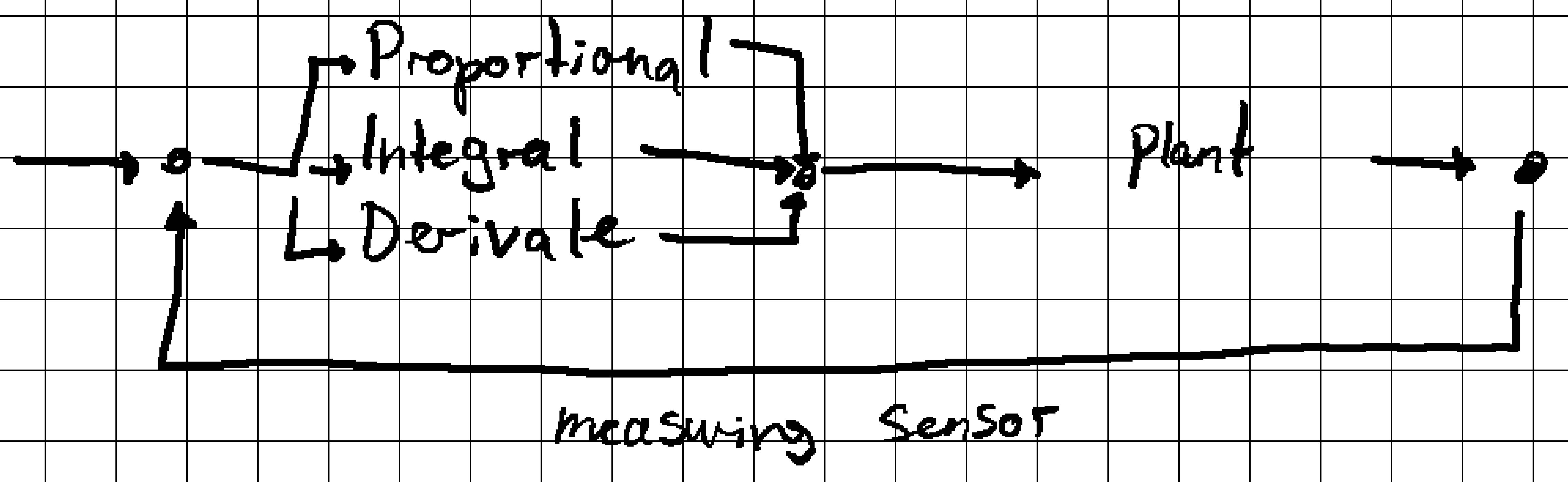
open Loop control system



⇒ open loop systems work best in a stable environment were the relationship between input and output stays consistent (ex: washing machine → runs x minutes)

⇒ closed loop changes behaviour based on measurements (ex: air conditioning → based on room temperature)

Proportional Integral Derivate Controller



⇒ proportional → reacts to current error
Integral → reacts to accumulation of past errors
Derivate → reacts to the rate of change of the error

Proportional Controller

↳ A correction is applied. The correction is proportional to the difference between the desired and measured value (ex: used to decide the moving speed)
↳ first fast, then slow

Integral Controller

↳ avoid stady-state errors
(ex: keep altitude of drone)

Proportional Derivate Controller

↳ if error is changing too quickly, the derivative will reduce the control output (ex: prevent drone over tilt)

Indoor Positioning System

↳ Indoor GPS

→ network of infrared transmitters and receivers

  ↳ milimeter accuracy

Transmitter → fixed Positions

Receivers → mounted on objects being tracked

Control System → synchronize beacons & compute object position in real time

Computer Vision Tracking

↳ calculate trajectory of an object over time by locating its position in every frame

→ reduces hardware cost, easy to install, scalable

→ not as accurate as other solutions, processing time can be long

Eye-in-Hand (ENH)

↳ two or more cameras directly mounted to the end-effector

  ↳ limited field of view → see more when robot moves

Eye to Hand (E2H)

↳ camera mounted with view of end-effector

  ↳ field of view maybe temorarily obstructed when arm is moveing

Interception and Grasping Tasks

Interception of moving target can be an interception of

- manoeuvoring objects → changes motion randomly & quickly
- non-manoeuvoring objects → motion change continuous path and constant speed

Assembly Tasks

↳ consistent quality, minimizing cost

⇒ most popular procedure → peg-in-hole

↳ 33% of all automated tasks

Sorting Tasks

↳ ex: waste sorting with arm & vision

Pick and Place Tasks

↳ challanges when picking small or reflective objects

Precision Agriculture

↳ 90% currenty require human involvement
  ↳ 50% of total cost is labor cost

Automation Tasks:

- Diseas & Insect Detection
- Phenotyping
- Harvesting
- Plant management
- Plant monitoring
- Plant seeding
- Weeding

End - Effectors:

- cutting blades (rotary)
- Sprayer
- cutting by pressing
- Sprayer with camera

Soft End - Effectors & Soft Sensors

↳ variation in size, shape etc require soft grips to not damage the product
  ↳ require sensors for surface roughness etc.
    ↳ need to be flexible to be mounted on e.g. Gripper

Soft End - Effectors

- pneumatic Fingers (air pressure)
- silicon gripper
- suction gripper

Capacitive Pressure Sensor

↳ to give feedback to e.g. gripper

chemical Sensor

↳ assessment of - gases
- PH level
- pesticide resedues

Use cases for environmental Sensors

- Volcanic research
- early fire Detection
- emission monitoring (methane, CO_2 ...)
- air quality

Internet of Things (IOT)

→ first connected device → 1982 – Coca Cola machine

⇒ sensors & actuators embedded in physical objects

RFID & NFC

↳ for auto Identification & wireless communication

RFID

↳ retransmission of radio waves
  ↳ origin in espionage → altering shape of resonator when exposed to an audio signal

Operating principle

↳ Tag needs energy to power microchip and transmit data back

Far Field & Near Field RFID

Industrial IOT (IIOT)

↳ smart sensors, actuators etc. to enhance manufactoring and industrial processes

⇒ focus on automation & efficiency

Internet of Robotic Things (IORT)

↳ enables robot to interact with other devices

IOT to WOT

↳ enables discovery & usage of things like Webpages

↳ Integration in foreign systems

↳ Integration in Web-based services (e.g. search engine)

⇒ control via Hypermedia structure

⇒ HTTP is not a good option to bring the WoT to constraint devices as it has a lot of overhead

The Constrained Application Protocol (CoAP)

↳ lightweight Protocol designed for IOT devices with limited power, memory and processing

- Lightweight
- RESTful → has methods like GET ... but simpler and optimized for low power
- Runs on UDP → reduces overhead, enables multicast
- Supports Resource Discovery → discover & interact with others
- Low Latency

WoT Thing Description

↳ standardized Description of a "Thing" with properties, actions, events

⇒ Link "Things" to other "Things" or Data

SPARQL

↳ Query Language to retrive and manipulate data stored in the Resource Description Framework (RDF), which is used to represent linked data on the web

Tools to Work with Robots

- ROS - Robot Opeating System
  - ↳ set of software Libraries
  - ↳ enables creation of robot digital twins
- Gazabo
  - ↳ Design & Test robots in virtual environments
- Webots
  - ↳ Open Source Simulator
  - ↳ Rapid prototyping environment

Common Sense for Robots

↳ ability to understand and interact with the world

- Context Awareness
- Reasoning
- Goal Oriented Behaviour
- Social understanding
- Learning from experience

⇛ Information that people take for granted

When did AI begin

1942 - story 'Runaround'

1950 - first general purpose computer becomes available to scientific community

1955 - combinatorial problems

1956 - Heuristics to solve specific problems (geometry, calculus...)

1962 - Heuristics to reduce search space

1969 - Slow down of neural network research as it was shown that we dont have enough computational power

1970... - AI becomes interesting for industries

1980 - Expert System flourish

1980 - neural network development for signal processing
1990 - Multi-Agent Approach were born
2000 - statistical processing given the availability of data on the Web
mid 2000 - "deep learning" gains popularity
2012 - unsupervised learning proves its possible to identify cats
2014 - Google buys DeepMind startup
2015 - Facebook releases DeepFace to automatically identify and tag users in pictures
2016 - Google Deep Mind defeats Master Go Player

Knowledge Graph

↳ structural representation of information that captures relationships between entities

Semantic Web

↳ make data machine-readable so machines can understand, interpret and process it

→ mapping keywords to results

Ontologies

↳ structured Frameworks that define and organize knowledge about a specific domain by describing concepts, their properties and relationships between them

Uniform Resource Identifiers (URIs)

↳ Identifies a resource e.g. Webpage

Resource Descriptive Framework (RDF)

↳ representation Language for URIs

RDF Schema (RDFS)

↳ extends the RDF to support the expression of structured vocabularies (e.g. hierarchies)

Web Ontology Language (OWL)

↳ extends the RDFS to provide more expessivity

SPARQL

↳ Query Language for RDF

Treade - Offs

→ cost of ontology development vs completeness

→ Complexity vs Explainability

→ Performance vs Expressivity

Deep vs Shallow ontologies

Deep → considerable effort to build
- in science (e.g. biology, medicine)
  - e.g classify animals into categories including detailed information like diet etc...

Shallow → unchanging terms to organize large amount of data
- e.g classify animals into categories like bird, fish... without more detail

Folksonomies

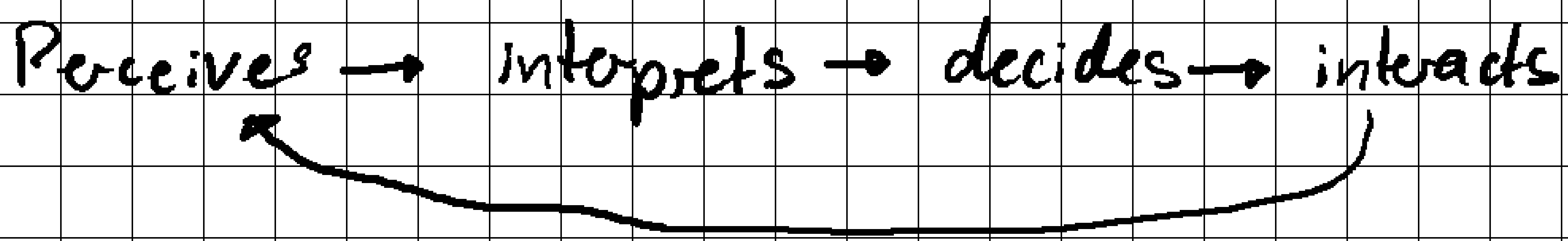
- end user tag items to make it easier to find

  e.g. Instagram

Know Rob

- Robot plans automatically adapt when the robots knowledge changes.

Robots that understand their Environment



Toward Grounded Commonsense Reasoning

↳ Robots must reason about what additional information they need to make appropriate decisions, and then actively perceive the environment to gather that information

Human Robot Interaction (HRI)

↳ how robots interact with people and how people respond to their presence

→ how can a robot approach safe and appropriate

→ what social rules does the robot need to follow

⇒ if a robot is human like, humans will interact with them using their existing experience in human to human interaction

Robot Design

↳ how to design, so that people want to interact with it ?

Outside-in approach

↳ start from the interaction you went the robot to be engaged in and determin its outside shape based on it

- → smaler size to not be intimidating
- → pet like interaction

Robot morphology and form

- → androids & humanoids resemble humans in appearence
  - ↳ increases the expectations regarding their capabilities
- → Zoomorphic robots
  - ↳ shaped like animals we are familiar with (cat, dog)
    - ↳ also animals that we dont interact with (dinosaur)
- → Robjects
  - ↳ robotic artifacts designed based on objects (e.g. moving trash cans)

⇒ Design to inspire people to see them as social agents

Affordances

↳ the way someone can use an object

→ Design in a way that make the possible interactions obvious

→ robot talks → we expect it to understand us

→ robot displays emotion → we expect it to read emotions

⇒ Don't Mix Metaphors

↳ animals that talk human like } people may find disturbing

↳ partially coverd in skin

Anthromorphizing

↳ humanizing an object

Pareidolia

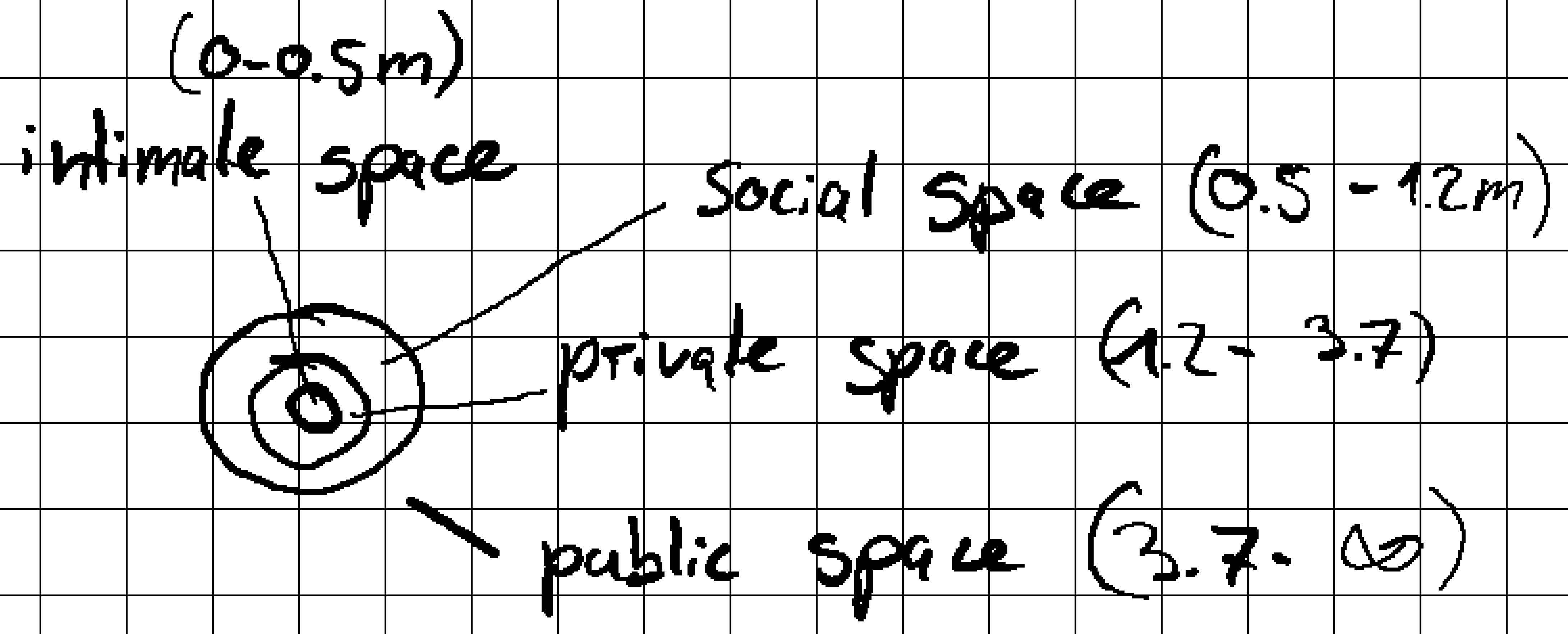
↳ the effect of seeing human like features in random patterns or mundane objects

Uncanny Valley

↳ region of negative emotional response towards robots that seem "almost" human

Proxemics

↳ Describe how people take up space in relation to others and how spatial positioning influences attitudes, behaviours and interpersonal interaction



⇒ robots should hold appropriate distance to increase acceptance

Simultaneos Lacation Mapping (SLAM)

↳ mapping environment and the people in it, including the direction that they are facing

Non Verbal Communication

↳ cues signal "true" attitude

- Gaze → signals interest, facilitate interaction
- Deicic Gesture → pointing to something → establishes attention
- Iconic Gesture → illustrate whats being said (in front etc.)
- Symbolic Gesture → e.g. waving to say hello
- Beat Gesture → go along with rhythm of speach
- Mimicry → replication of behaviour of another person
- Posture → signal emotional state

Robot Perception of Non Verbal Cues

↳ drive engagement by reading the above cues to keep people engaged

⇒ Touch → people feel more comfortable when initiating an interaction including touch

↳ robots have to be designed for it (safety)

Research Methods

Exploratory and Confirmation Research

aim to find general information about a domain that has not previously be examined in detail

test Hypothesis based on gathered information, trying to confirm something

Correlation vs Causation

relationship or pattern between two variables, but one doesn't necessarily cause the other

one variable directly influences or causes a change in another

Qualitative Methods

↳ understanding experiences, meanings, concepts

Quantitative Methods

↳ Measuring & analyzing nummerical data

Mixed Methods

↳ exploratory with experiments to confirm

Considerations

↳ In a study we need to consider demographic, location, context