Definitions of a robot:

• Robot Institute of America (1980): programmable, multifunction manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks.

• Russell and Norvig (2003): physical agent that performs tasks by manipulating the physical world.

Types of robots:

• Industrial robots – manipulators (robot arms).

• Automatic Guided Vehicles (AGV) – industrial mobile robots.

• Collaborative robots (industrial or not).

• Service robots.

• Autonomous vehicles.

• Search & rescue, de-mining, etc.

• Legged (walking) robots.

• Humanoids.

Properties of a robot:

• A robot is an autonomous system which exists in the physical world.

• A robot has a physical body in the physical world.

• A robot has sensors and it can perceive its environment.

• A robot has effectors and actuators – it can act in the environment.

• A robot has controller which allows it to be autonomous.

Sensors:

• Proprioceptive sensors – measure internal states, e.g., encoder, inclinometer, compass.

• Exteroceptive sensors – measure objects relative to the robot.

• Sensing allows the robot to know its state, which can be observable, partially observable, or unobservable.

• State can be discrete or continuous.

• State space consists of all possible states in which the system can be.

Effectors and Actuators:

• Effector – any device on a robot that has an effect on the environment

• Actuator – a mechanisms that allows the effector to execute an action or movement, e.g., motors.

• Effectors and actuators provide two main types of activities.

• Mobile (autonomous) robotics – their main activity is (loco)motion.

• Manipulators – their main activity is handling objects.

Degrees of freedom:

• Degree of Freedom (DOF) is the minimal required number of independent parameters to completely specify the motion of a mechanical system.

• In 3D space, a body has 6 DOF.

• Translational DOF – x, y, z.

• Rotational DOF – roll, pitch, yaw.

• Controllable DOF – the number of the DOF that are controllable, i.e., a robot has an actuator for such a DOF.

Mobile (autonomous) robots:

• Mobile robot – its basic functionality is locomotion.

• Navigation is the method of determining position, course and distance traveled

• environment model building,

• determination of the position with respect to this model,

• planning the motion to the goal using the map and the current position information

• In robotics, measurement data is obtained from sensors that have their point of view (we call it a coordinate system).

• We say that point p has a position with respect to the global coordinate system.

• RGB (red, green, blue) colors correspond to axes (X, Y, Z).

• At least each sensor has its own coordinate system.

• Why we need to know the relations between coordinate systems? To have a unified view of the world from multiple sensors. Without it, there are no gains from multi-sensory setups.

• Multi-sensory system calibration - the process of finding the relations between the coordinate systems of different sensors.

• In each walking robot/manipulator, we have coordinate systems at each joint.

The properties of the rotation matrix

1. orthonormal – each row/columns is a unit vector and the columns/rows are orthogonal.

2. its determinant is equal to +1 – the length of the vector is unchanged after transformation.

3. the inverse is the same as the transpose, R −1 = R T .

What if some condition is not met? It is NOT a rotation matrix. Something went wrong.

As an example, the determinant of -1 indicated a left-handed coordinate system.

∙ A point is described by a bound coordinate vector that represents its displacement from the origin of a reference coordinate system.

∙ Points and vectors are different things even though they are each described by a tuple of numbers. We can add vectors but not points. The difference between the two points is a vector.

∙ A set of points that represent a rigid object can be described by a single coordinate frame and these points are described by constant vectors relative to that coordinate frame.

∙ The position and orientation of an object’s coordinate frame is referred to as its pose.

∙ A relative pose (transformation) describes the pose of one coordinate frame with respect to another.

∙ A coordinate vector describing a point can be represented with respect to a different coordinate frame by applying the relative pose to the vector.

∙ We can perform algebraic manipulation of expressions written in terms of relative poses (transformations).

3D

∙ We use right-hand rule when in doubt about the direction of the axes

∙ Transforming the coordinate systems is 3D is similar to 2D

∙ The transformation matrix now has a size of 4 by 4

∙ The inverse of the rotation matrix is just its transpose

∙ a 3D rotation matrix has a size of 3 by 3

∙ The order of rotations matters

∙ There are some drawbacks to the rotation matrix representation (memory efficiency, using more parameters than necessary)

3D transformation matrix ATB:

∙ Can be understood as a pose (position and orientation) of a coordinate system B in the coordinate system A

∙ Can be understood as a 3D transformation matrix that moves points from coordinate system B to coordinate system A

∙ The transformation matrix (4 by 4) consists of a rotation matrix (3 by 3) and a translation vector (3 by 1)

∙ We use homogeneous coordinates for points when using transformations

∙ Similarly to the 2D, we have a simplified equation for the inverse of the transformation

Possible 3D rotation representations:

∙ Rotation matrix (SO(3))

∙ Direction Cosine Matrix (DCM)

∙ Euler angles

∙ Axis-angle representation

∙ Quaternion

Properties of quaternion - q = (qw, qx, qy, qz):

∙ Effective and known equations to multiple transformations

∙ Efficient memory representation

∙ May require normalization after several numerical operations

∙ Geometric interpretation

∙ Easy to convert to axis-angle representation

∙ Non-minimal representation (not perfect for optimization)

∙ There are many 3D rotation representations: a rotation matrix, Euler angles, axis-angle, and quaterion.

∙ Each representation has certain advantages and disadvantages

∙ Conversions between representations are very useful

NORMALIZING ROTATION MATRIX (the algo is on Telegram)

When to orthonormalize the rotation matrix? When determinant is not equal to 1. After tenths of processing steps. In the end, it can only improve the results.

QUATERNION NORMALIZATION

When to normalize quaternion? The normalization problem also occurs for the quaternion after tenths of operations when ∥q∥^2 is not equal to 1.

Quaternion normalization algorithm:

There is only one, simple way: q ′ = q / ∥q∥^2

DH

Most kinematics libraries do accept the DH parameters and for that reason, it’s a reasonable approach to begin with. DH is standardized and reduces ambiguity while modelling.

DH ALTERNATIVES:

∙ Screw Theory representations

∙ Hayati-Roberts

∙ other geometric modellings

------------------------------------------------------------------------------------------------------------------------------------------

Manipulator

Manipulator has n joints and n + 1 links. Joints can be prismatic or rotational. The state of the i-th joint is usually called qi .

Most of the modern manipulators consist mostly of joints that are rotational.

Link is a mechanical element. Joint is a connection between links.

A link between two joints can be specified by four parameters.

The goal of FK is to compute the pose of end-effector given the angles of each joint.

------------------------------------------------------------------------------------------------------------------------------------------

Spherical wrists:

∙ A key component of all modern arm-type robots

∙ 3 axes of rotation intersecting at a common point

∙ Any wanted rotation can be achieved in the wrist

IK with spherical wrists:

∙ We can solve independently for rotation and translation

∙ IK for translation (position of spherical wrist)

∙ IK for rotation (only three wrist rotations)

Numerical IK:

∙ Uses optimization to reach wanted configuration

∙ Might not reach desired position

∙ It is computationally expensive (compared to closed-form solution)

∙ Universal solution that works for all types of manipulators

∙ Robotic manipulation is currently rapidly evolving due to the collaborative robots and deep learning

∙ Robotic manipulator is said to have n degrees of freedom (n + 1 links and n joints)

∙ Joints can be rotational and prismatic

∙ Forward kinematics (FK) determines the end effector pose based on the state of joints

∙ FK can be determined based on a series of transformation matrices

∙ Denavit-Hartenberg (DH) notation is a way of making it easier

∙ Assigning axes in DH notation can be tricky

∙ Most of used manipulators already have known kinematics equations

∙ There are programming libraries for robot kinematics that can be used

Redundant manipulator:

A redundant manipulator is a robot with more than six joints.

∙ Six joints are sufficient in theory

∙ Joint limits and singularities in practice limit the working space

∙ Additional joints result in infinite number of IK solutions

∙ There is a need to introduce additional constraint (i.e. joint-coordinate vector elements have the smallest magnitude)

JOINT-SPACE VS TASK-SPACE (CARTESIAN) MOTION (screenshots describing these spaces’ definitions on Telegram)

Joint-space:

+ Faster execution (solve IK at waypoints only)

+ Actuator motion is smooth and easier to validate

- Intermediate points not guaranteed to respect joint limits or collisions

Task-space (Cartesian) motion:

+ Motion is predictable (interpolation in task space)

+ Better handling of obstacles and collisions

- Slower execution (solve IK every time step)

- Actuator motion not necessarily smooth and harder to validate

∙ The inverse kinematics (IK) problem is finding the value of the joints based on the end effector pose

∙ There might be 0, 1, 2, or infinitely many solutions

∙ We prefer analytical (closed-form) solutions over numerical solutions

∙ Spherical wrists can be used to decouple the translation and rotation problems

∙ We can plan motion in joint-space and in task-space (Cartesian)

∙ We have different ways of interpolating motion (i.e. trapezoidal velocity, polynomial)

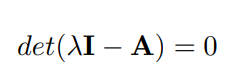
Model Predictive Control (MPC) are a multivariable control algorithm that uses:

• a dynamic model of the process

• a cost function J over the receding horizon

• an optimization algorithm minimizing the cost function J using the control input u

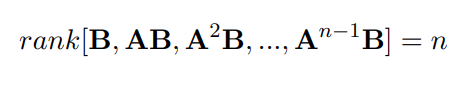
A continuous linear time-invariant system is exponentially stable if and only if the system has eigenvalues with strictly negative real parts. (i.e., in the left half of the complex plane).



(full screenshot on Telegram)

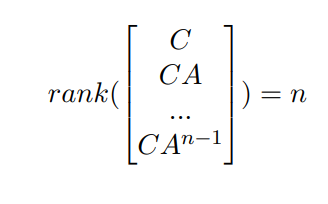
Lyapunov theorems – we conclude about stability system according to the linearization of the system in the defined point or the function defined on the phase space, which can be used to prove the stability of an equilibrium point.

The state controllability condition implies that it is possible – by admissible inputs – to steer the states from any initial value to any final value within some finite time window. A continuous time-invariant linear state-space model is controllable if and only if:



(full screenshot on Telegram)

Observability says if the internal states of a system can be inferred by knowledge of its external outputs. A system is observable if:



(full screenshot on Telegram)

State (Luenberger) observer

We have access to the input u(t) and the output from the system y(t). Is it possible to estimate the state x(t) by measuring u(t) and y(t)? – It is possible if the system is observable.

(full screenshot on Telegram)

The system is stable (estimation converges to the system state x(t)) when the error e decreases.

Kalman filter estimates the state of the system in the presence of Gaussian noise using a dynamic model of the system, current measurements, covariance of the process and observation noise. Important assumptions: the system is linear, Gaussian measurement noise.

Kalman filter – application:

• state estimation

• sensor fusion

• guidance, navigation, control of vehicles

• economy

• Extended Kalman Filter – for nonlinear systems

Perception is a key component of any autonomous robot as it is necessary to make decisions/plans and later execute them.

MULTI-SENSORY SYSTEMS

Why are so many sensors used? Sensors with different measurement characteristics guarantee safe working in many conditions.

Exposure - the amount of light per unit area reaching a frame of photographic film or the surface of an electronic image sensor.

Exposure triangle

Exposure is determined by:

1. ISO

2. Aperture

3. Shutter speed

ISO can be understood as artificial ’brightness’:

∙ greater ISO results in a brighter image (ISO200 is two times brighter than ISO100)

∙ greater ISO comes at a price of greater noise (grainy noise)

Aperture size - the size of the focal point that light travels through. The aperture also determines the depth of field (the distances where objects appear to be in sharp focus).

Shutter speed - the amount of time how long the aperture remains open.

What elements influence the final projection?

∙ 3D point location

∙ Pose (position + orientation) of the camera

∙ Intrinsic camera parameters (focal lengths, image center

∙ Distortions

Radial distortions:

∙ stem from imperfectness of lenses (focal lengths are not the same in each location)

∙ greater distortions can be observed on the borders of the image.

Tangential distortions:

∙ stem from an imperfect assembly of the camera (the lens is not parallel to the image plane)

∙ usually has less influence than radial distortions.

CAMERA CALIBRATION

The goal of the calibration is to find the parameters of the chosen model and the distortions that always occur due to real-life imperfections.

How to perform camera calibration? Capture 20-30 images of the camera observing a chessboard marker with a known size and then use a publicly available tool to find the camera (focal lengths and image center) and distortion parameters.

• Mobile robots are the most ’autonomous’ group of robots

• Some methods and techniques used for robot navigation draw from the AI algorithms and paradigms

• Some AI algorithms are applied directly in mobile robotics

• To use these robots as testbeds for development you should also know some specific methods and representations

• Mobile robots show a growing commercial potential

• Self-driving cars are founded on the vast experience in mobile robotics