EMARO/ROBOTICS – DIPLOMA EXAM QUESTIONS

1. Definition of reference frames for manipulators

Reference frames are sets of rules or agreements that observers use to measure the position and orientation of a rigid body [3]. When dealing with manipulators, reference frames are important to describe their position and orientation accurately. There are different types of reference frames for manipulators, including [10]:

- Global reference frame: This is a universal coordinate frame, as defined by the x, y, and z axes. The joints of the robot move simultaneously in a coordinated manner to create motions along the three major axes. The world reference frame is used to define the motions of the robot relative to the other objects, define other parts and machines with which the robot communicates, and define motion trajectories [10].
- Local reference frame: A local reference frame is a fixed reference frame on the rigid body that allows its position to be represented in an abbreviated form. In other words, the position of a rigid body refers to the position held by the origin of a reference frame fixed to the rigid body [3]. The frames can be denoted by the symbol {}; for example, {i-1} denotes a frame whose origin is in joint i-1 [7].

In general, a rigid body in three-dimensional space has **six degrees of freedom**: three rotational and three translational.

A conventional way to describe the position and orientation of a rigid body is to attach a frame to it. After defining a reference coordinate system, the position and orientation of the rigid body are fully described by the position of the frame's origin and the orientation of its axes, relative to the reference frame.

2. Description of orientation in robotics

A **rotation matrix** describes the relative orientation of two such frames. The columns of this 3 × 3 matrix consist of the unit vectors along the axes of one frame, relative to the other, reference frame.
Thus, the relative orientation of a frame {b} with respect to a reference frame {a} is given by the rotation matrix R

For more read

https://en.wikibooks.org/wiki/Robotics_Kinematics_and_Dynamics/Description_of_Position_and_Orientation

https://www.coppeliarobotics.com/helpFiles/en/positionOrientationTransformation.htm

1. Euler Angles (yaw, roll, and pitch)
2. Rotation matrix
3. Quaternions

3. Denavit-Hartenberg notation

Denavit-Hartenberg (DH) notation is a widely used method for describing the kinematics of robotic manipulators. It provides a systematic way to assign coordinate frames to each joint of the robot, which allows for the calculation of the position and orientation of the end effector.

The DH notation uses four parameters to describe the relationship between adjacent coordinate frames:

1. The link length, denoted by "a", which is the distance between the z-axes of adjacent frames along the x-axis of the current frame.
2. The link twist, denoted by "α", which is the angle between the z-axes of adjacent frames around the x-axis of the current frame.
3. The link offset, denoted by "d", which is the distance between the x-axes of adjacent frames along the z-axis of the current frame.
4. The joint angle, denoted by "θ", which is the angle between the x-axes of adjacent frames around the z-axis of the current frame.

Using these parameters, the transformation matrix between adjacent frames can be calculated. The transformation matrix is a 4×4 homogeneous matrix that describes the translation and rotation between the frames.

The DH notation provides a convenient way to describe the kinematics of robotic manipulators, and is widely used in robotics research and industry.

4. Methods of direct and inverse kinematic problem solution for serial manipulators

For serial manipulators, there are two well-known problems in kinematics: the direct and inverse problems [Source 1].

The direct kinematics problem involves computing the position and orientation of the end effector of a manipulator given the joint variables [Source 2].

In contrast, the inverse kinematics problem involves determining the joint variables required to achieve a desired position and orientation of the end effector [Source 2].

There are several methods to solve the direct and inverse kinematic problems for serial manipulators, including [Sources 3, 5, 7, 10, and 11]:

- Analytical solutions: These are closed-form expressions for the joint positions as functions of the desired end effector position and orientation [Source 10].
- Geometrical solutions: These methods use geometric relationships to determine the joint variables required to achieve a desired end effector position and orientation [Source 3].
- Numerical solutions: These are iterative methods that compute the joint variables required to achieve a desired end effector position and orientation. Examples of numerical methods include the Jacobian method, the inverse Jacobian method, and the Newton-Raphson method [Sources 3, 5, and 12].

In conclusion, for serial manipulators, there are two well-known problems in kinematics: the direct and inverse problems. There are several methods to solve these problems, including analytical, geometrical, and numerical solutions. Examples of numerical solutions include the Jacobian method, the inverse Jacobian method, and the Newton-Raphson method [Sources 1-3, 5, 7, 10, 11, and 12].

5. Jacobians in robotics

https://www.rosroboticslearning.com/jacobian?utm_content=cmp-true

6. Methods of description of manipulator dynamics

Lagrangian, Kane, FEA

Newton's second law for a rigid body i:
change of linear momentum = sum of external forces acting on body i

Euler's equation for a rigid body i:
change of angular momentum = sum of external torques acting on the body

7. Direct and inverse dynamic problems in robotics

Direct dynamics in robotics involves computing the motion of the robot manipulator given the joint torques [Sources 1, 3, and 6]. Direct dynamics can be used to simulate the motion of the robot manipulator under the influence of external forces and torques, or to design control strategies for the manipulator [Sources 1 and 3]. The dynamic equations of motion can be derived using the Lagrange equations of motion or the Newton-Euler equations, and numerical methods can be used to compute the motion of the robot manipulator given the joint torques [Sources 1, 3, and 6].

Direct dynamics is mainly used for simulation purposes [Sources 1 and 6]. For example, direct dynamics can be used to simulate the motion of a robot manipulator under the influence of gravity and external forces, which can help in designing and testing control strategies for the manipulator [Sources 1 and 3]. Direct dynamics can also be used to study the effect of forces upon the movements of the robot manipulator [Source 6].

In conclusion, direct dynamics in robotics involves computing the motion of the robot manipulator given the joint torques. Direct dynamics can be used to simulate the motion of the robot manipulator or design control strategies. The dynamic equations of motion can be derived using the Lagrange equations of motion or the Newton-Euler equations, and numerical methods can be used to compute the motion of the robot manipulator given the joint torques. Direct dynamics is mainly used for simulation purposes [Sources 1, 3, and 6].

Inverse dynamics in robotics involves computing the joint torques required to achieve a desired motion of the robot manipulator [Sources 0, 7, and 11]. Inverse dynamics can be used in robot control and trajectory planning, as it can be used to convert desired positions, velocities, and accelerations into the joint generalized forces required to achieve those accelerations [Sources 0 and 11]. Inverse dynamics can also be used to check or ensure that a proposed trajectory can be executed without exceeding the actuators' limits [Sources 5-7].

Inverse dynamics is also used to compute the forces when external forces are involved, such as when a humanoid robot is picking something up or carrying something. Inverse dynamics can help compute the internal forces for a humanoid picking something up, or carrying something. Inverse dynamics can even help articulated bodies to react to external forces, such as a getting hit by a projectile. Additionally, external ground forces need to be kept in mind when an articulated body is walking on the floor. Even leaning on a wall causes normal forces. Inverse Dynamics can take into account all of these external forces [Source 7].

In conclusion, inverse dynamics in robotics involves computing the joint torques required to achieve a desired motion of the robot manipulator. Inverse dynamics can be used in robot control and trajectory planning, as well as to compute the forces when external forces are involved. Inverse dynamics is a fundamental concept in robotics and is used in many applications [Sources 0, 7, and 11].

---

8. Joint space schemes for trajectory generation

In the field of robotics, both task space and joint space are used to describe the configuration and motion of robotic systems. They represent different perspectives or coordinate systems in which the robot's movements can be analyzed and controlled.

Joint Space:
Joint space refers to the configuration space of a robot, which is defined by the joint variables or joint angles of the robot's individual joints. In joint space, the robot's configuration is described by a set of joint angles or joint positions, usually denoted as q = [q1, q2, ..., qn], where n is the number of joints in the robot.

In joint space, the robot's motion is described by the changes in joint angles or joint velocities. Controlling the robot in joint space involves specifying the desired joint positions, velocities, or torques. It is often used in tasks such as trajectory planning, inverse kinematics, and joint-level control.

The advantage of working in joint space is that it provides a direct representation of the robot's internal structure and allows for precise control over each joint. However, it does not explicitly capture the robot's interaction with the external environment or the desired end-effector motion.

9. Cartesian paths description and programming for serial manipulators

Task Space:
Task space, also known as Cartesian space or operational space, refers to the coordinate system defined by the position and orientation of the robot's end-effector in the external world. It describes the robot's motion in terms of its position and orientation in three-dimensional space.

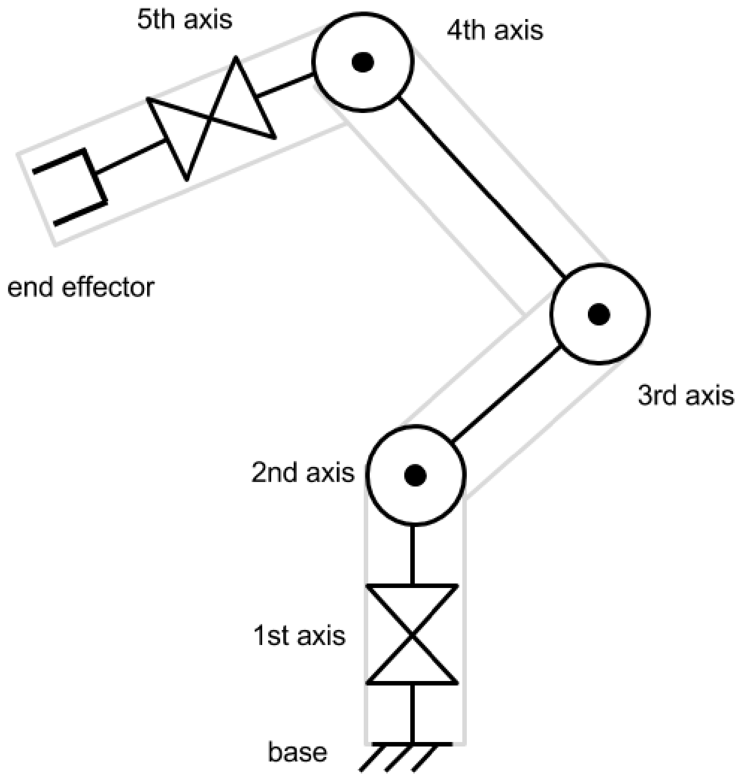
In task space, the robot's configuration is represented by the pose or position and orientation of the end-effector, often denoted as X = [x, y, z, roll, pitch, yaw]. The position components (x, y, z) describe the end-effector's location, while the orientation components (roll, pitch, yaw) describe its rotational pose.

Working in task space allows for specifying the desired position and orientation of the end-effector directly, making it more intuitive for many applications. It is particularly useful in tasks such as motion planning, trajectory generation, and operational-level control.

The advantage of task space is that it provides a higher-level representation of the robot's behavior in relation to the external world. It allows for specifying goals and constraints in a more intuitive manner, regardless of the robot's internal joint structure. However, it may require additional computations and transformations to map the desired task space motions to joint space commands.

In summary, joint space and task space provide different perspectives for describing and controlling the motion of robotic systems. Joint space focuses on the robot's internal joint variables, while task space describes the robot's position and orientation in the external world. Both representations are important in robotics and are used in various contexts depending on the specific requirements of the task at hand.

Cartesian space is most likely the one you're most familiar with if you've ever taken an algebra class. It's space broken down into X, Y, and Z coordinates in a simple grid. The point (3, 1, 2) is a point in cartesian space.



However, this kind of information is not always particularly helpful in robotics. For example, how would you tell a robotic arm, like the one in the diagram above, to position its end effector in a certain location? Depending on where it is, there might be a number of ways for it to position itself there... or it might be entirely impossible.

So instead, you tell each of its joints exactly what position to be in. By giving the robot the angles each joint should be, the robot can then move to the desired location. This is joint space.

The trick, of course, is calculating what these angles should be. But if you know the lengths of each segment, this isn't terribly complicated.

https://www.quora.com/What-is-the-difference-between-a-Cartesian-space-and-joint-space-in-robotics

---

Cartesian paths describe a continuous path in space that a robot end-effector can follow. These paths are defined in Cartesian space, which is the space of position and orientation in three-dimensional (3D) space. Cartesian path planning is the process of generating a path in Cartesian space that a robot manipulator can follow to execute a task. The path is defined by a set of waypoints in 3D space that describe the desired position and orientation of the end-effector [Sources 3 and 10].

Programming Cartesian paths for serial manipulators involves generating a smooth trajectory in Cartesian space and then converting it to a joint-space trajectory that the robot can follow. One way to generate Cartesian paths is to use continuous genetic algorithms (CGAs). CGAs can be used to solve the path generation problem for robot manipulators by obtaining smooth geometric paths in the joint space of the manipulator [Sources 3, 6, 8, and 10].

The inverse kinematics problem is then formulated as an optimization problem based on the concept of minimizing the accumulative path deviation. The CGAs use smooth curves to represent the required geometric paths in the joint space throughout the evolution process. This approach can be applied to any general serial manipulator with positional degrees of freedom that might not have any derived closed-form solution for its inverse kinematics. The generality and efficiency of the proposed algorithm are demonstrated through simulations that include 2R and 3R planar manipulators, PUMA manipulator, and a general 6R serial manipulator [Source 6].

In conclusion, Cartesian paths describe a continuous path in space that a robot end-effector can follow, and Cartesian path planning is the process of generating a path in Cartesian space that a robot manipulator can follow to execute a task. Programming Cartesian paths for serial manipulators involves generating a smooth trajectory in Cartesian space and then converting it to a joint-space trajectory that the robot can follow. One way to generate Cartesian paths is to use continuous genetic algorithms (CGAs).

10. Position control of manipulators

Position control of manipulators refers to the ability to precisely control the position of robotic manipulator arms. It involves determining and controlling the joint angles or Cartesian coordinates of the robot's end effector (tool or gripper) in order to achieve desired positions in space.

There are several approaches to position control of manipulators:

1. Joint-Space Control: In joint-space control, the control inputs are directly applied to the robot's joint actuators. The joint angles are controlled to achieve the desired end effector position. This approach requires accurate knowledge of the robot's kinematic model and joint feedback.
2. Cartesian-Space Control: In Cartesian-space control, the control inputs are applied to the end effector directly in Cartesian coordinates (e.g., x, y, z). The control system computes the required joint angles to achieve the desired end effector position. This approach is often used when the desired position in space is specified rather than individual joint angles.
3. Hybrid Control: Hybrid control combines joint-space and Cartesian-space control. It involves a combination of controlling the end effector position and individual joint angles simultaneously. This approach is useful in scenarios where certain joints need to be controlled precisely while allowing other joints to move more freely.
4. Inverse Kinematics: Inverse kinematics is the mathematical process of determining the joint angles required to achieve a desired end effector position. It involves solving the inverse relationship between the robot's joint angles and the corresponding Cartesian coordinates of the end effector.
5. Feedback Control: Feedback control techniques, such as PID (Proportional-Integral-Derivative) control, are commonly used to improve the accuracy and stability of position control. Feedback control involves measuring the difference between the desired position and the actual position of the end effector and applying control actions based on the error.

The choice of position control method depends on the specific requirements of the robotic application, the complexity of the manipulator's kinematics, and the desired performance characteristics. Advanced control techniques, such as adaptive control and nonlinear control, can also be employed for more complex and precise position control tasks.

11. Position/force control

Position/force control is a concept in robotics and automation that involves controlling the position and force exerted by a robot or a robotic system. It is commonly used in applications where precise control over both position and force is required, such as assembly tasks, material handling, and contact-sensitive operations.

In position control, the primary objective is to control the robot's end-effector or tool to reach and team maintain a specific position or trajectory in space. The position control system typically takes input commands, such as desired positions or trajectories, and uses sensors and algorithms to calculate the necessary control signals to achieve the desired positions. The control signals are usually applied to the robot's joint motors or actuators to drive the robot's movement.

Force control, on the other hand, focuses on controlling the force exerted by the robot's end-effector or tool during interaction with the environment. This control method is used when tasks require applying a specific force or maintaining a desired force level. Force control systems utilize force sensors or tactile sensors to measure the forces at the end-effector and generate control signals to adjust the applied forces.

Position/force control combines these two control strategies to achieve a more advanced level of control. It allows the robot to perform tasks that involve both precise positioning and force-sensitive interactions. In position/force control, the control system simultaneously regulates the position and force of the robot's end-effector based on the task requirements and environmental conditions.

To implement position/force control, a robotic system typically requires a combination of hardware and software components. These may include position sensors (such as encoders or vision systems) to measure the robot's position, force sensors or tactile sensors to measure forces, and control algorithms that process the sensor data and generate appropriate control signals. The control signals are then sent to the robot's actuators to adjust the robot's motion and applied forces.

Overall, position/force control enables robots to perform tasks with increased precision, adaptability, and sensitivity to the environment. It finds applications in various industries, including manufacturing, healthcare, and research, where fine-grained control over both position and force is crucial for successful task execution.

12. Definition of static stability in walking machines

Legged robots are inspired by biological systems (animals, insects). Compared with wheeled locomotion, legged locomotion is more complex and more difficult to control. Legged
locomotion needs to be stable. Two types of stability are discussed in the literature:

• Static stability: a statically stable robot can stand without falling over. The key is to have enough legs to provide support. Humans are not statically stable, standing up appears to be effortless, but we are actively using active control to achieve balance. For us, balancing is largely unconscious and must be learned.

The condition for static stability is very simple: the robot’s center of gravity must fall under the robot polygon of support. For a two legged robot, the polygon of support is a straight line as shown in figure 1. Figure 2 shows a comparison between static and dynamic walking.

Static walking Dynamic walking

-

13. Comment the main functions of walking robots control systems (also: draw the general scheme)

Are used where a robot should move in a difficult tarrain

The main functions of walking robots control systems are:

1. Locomotion control: This module controls the movement of the robot's legs to achieve stable and efficient walking. The control system needs to ensure that each leg is placed in the right position and at the right time to maintain balance and forward motion. It also needs to adjust the stride length and step frequency to adapt to different terrains and walking speeds.
2. Sensing and perception: This module uses sensors to detect the robot's environment and provide feedback to the control system. The sensors may include cameras, LIDAR, sonar, and other environmental sensors. The perception algorithms process the sensor data to extract meaningful information about the environment, such as the position of obstacles, terrain features, and other relevant information.
3. Planning and decision-making: This module plans the robot's trajectory and decides on the actions needed to achieve the desired goal. It takes into account the current state of the robot, the environment, and any task-specific requirements or constraints. The planning and decision-making module needs to consider factors such as the robot's stability, energy efficiency, walking speed, and obstacle avoidance.
4. Actuation and control: This module generates the control signals that dictate the robot's actuators' actions, such as motors, joints, and manipulators. It translates the high-level plans and decisions into specific control commands to achieve the desired robot motions. The actuation and control module need to adjust the torque, speed, and position of the actuators to maintain balance and stability while walking.

The general scheme of an autonomous robot control system involves multiple layers or modules that work together to enable the robot to perceive its environment, make decisions, and execute actions. The perception layer collects data from sensors and processes it to extract meaningful information about the environment. The planning and decision-making layer plans the robot's trajectory and decides on the actions needed to achieve the desired goal. The control layer generates the control signals that dictate the robot's actuators' actions, such as motors, joints, and manipulators. Feedback control mechanisms continuously monitor the robot's state, compare it to the desired state, and adjust the control signals to maintain stability and performance. The overall goal of the control system is to enable the robot to navigate autonomously and achieve its goals in a safe and efficient manner.

14. Give the sensors types which are used in mobile robots (walking machines, wheeled robots)

Sensors can be classified depending on:

1. what they measure, and
2. how they measure it.

What is measured:

▼ Exteroceptive (EC) sensors

▼ Proprioceptive (PC) sensors

used to measure the external world – information from the robot environment, e.g., distances to objects, object images, light intensity, sound amplitude.

used to measure the internal state of a robot, e.g., motor speed, motor current, heading of the robot, battery voltage.

Mobile robots depend heavily on exteroceptive sensors.

Measurement principle (how it is measured):

- ▼ Active (A) sensors

  emit the required energy and measure the environmental reaction, examples
  are wheel encoders, ultrasonic sensors, and laser range finders. they have better performance, but active sensing has some influence on the environment.

- ▼ Passive (P) sensors

  sensors measure ambient energy entering the sensor (energy coming from
  the environment): temperature probes, CCD and CMOS cameras, microphones, etc.

15. General structure of autonomous robot control system

The general structure of an autonomous robot control system typically involves multiple layers or modules that work together to enable the robot to perceive its environment, make decisions, and execute actions. Here is a simplified representation of the general structure:

1. Perception Layer:
   - Sensors: The robot is equipped with various sensors such as cameras, LIDAR, radar, depth sensors, and other environmental sensors. These sensors collect data about the robot's surroundings, including objects, obstacles, landmarks, and other relevant information.
   - Perception Algorithms: The perception algorithms process the sensor data to extract meaningful information about the environment. This may involve tasks such as object detection, recognition, localization, mapping, and scene understanding. The perception algorithms provide an understanding of the robot's surroundings to higher-level modules.
2. Planning and Decision-Making Layer:
   - Localization: The robot estimates its own position and orientation in the environment using techniques like Simultaneous Localization and Mapping (SLAM) or other localization methods.
   - Mapping: The robot builds and maintains a representation of the environment, which may include a 2D or 3D map. This map helps the robot navigate and plan its actions.
   - Path Planning: Based on the perception and mapping information, the robot plans a trajectory or path to reach its goal while avoiding obstacles. Path planning algorithms generate feasible paths that the robot can follow.
   - Task Planning: This module determines the high-level goals and plans the sequence of actions needed to achieve them. It takes into account the robot's current state, the environment, and any task-specific requirements or constraints.
   - Decision-Making: The decision-making component considers different factors such as the robot's objectives, task requirements, environmental conditions, and robot's capabilities to make informed decisions about the robot's actions.
3. Control Layer:

- Motion Control: The motion control module generates low-level control signals that dictate the robot's actuators' actions, such as motors, joints, and manipulators. It translates the high-level plans and decisions into specific control commands to achieve the desired robot motions.
- Feedback Control: Feedback control mechanisms continuously monitor the robot's state, compare it to the desired state, and make necessary adjustments to maintain stability, accuracy, and safety. This may involve sensor fusion, closed-loop control, and PID control techniques.

4. Execution Layer:
   - Actuators: The actuators are the physical components of the robot that perform the desired actions. These can include motors, grippers, arms, or any other mechanisms that allow the robot to interact with its environment.
   - Execution Monitoring: This module monitors the execution of planned actions and ensures that they are performed correctly and safely. It may include error detection, fault tolerance, and recovery mechanisms to handle unexpected events or failures.
5. Learning and Adaptation:
   - Learning Algorithms: The control system may incorporate learning algorithms to improve the robot's performance over time. This could involve machine learning techniques, reinforcement learning, or adaptive control to adapt to changing environments, optimize actions, or improve decision-making capabilities.

It's important to note that the specific architecture and components of an autonomous robot control system can vary depending on the robot's application, complexity, and the desired level of autonomy. The described structure provides a general framework for understanding the key components involved in controlling an autonomous robot.

16. What is robotics?

Robotics is a multidisciplinary field that encompasses the design, construction, programming, and use of robots. A robot is a machine or an autonomous system that can perform tasks autonomously or semi-autonomously, often mimicking human actions or performing tasks that are difficult or dangerous for humans to accomplish.

Robotics combines various areas of knowledge, including mechanical engineering, electrical engineering, computer science, and artificial intelligence. Robots can be physical machines that interact with their environment, or they can be software-based agents that perform tasks in virtual or digital spaces.

The field of robotics aims to develop intelligent machines that can sense their surroundings, make decisions based on that information, and then manipulate objects or perform actions to achieve specific goals. This involves creating robots with sensors to perceive the environment, actuators to carry out physical actions, and control systems to process information and generate appropriate responses.

Applications of robotics span a wide range of industries and domains, including manufacturing, healthcare, agriculture, space exploration, transportation, entertainment, and more. Industrial robots, for example, are used in factories for tasks such as assembly, welding, and packaging, while robotic surgical systems assist surgeons in performing delicate operations. Autonomous drones and self-driving cars are other examples of robotics technology being applied in real-world scenarios.

Overall, robotics is a rapidly evolving field that continues to advance and find new applications, with the goal of creating machines that can assist and augment human capabilities, automate tasks, and improve efficiency and safety in various domains.

17. Please give the robots classification concering diverse criteria (those which you are familiar with)?

Robots can be classified in various ways based on different criteria. Here are some common classifications:

1. Based on Application:
    - Industrial Robots: These robots are designed for industrial automation applications, such as manufacturing, assembly, and material handling tasks.
    - Service Robots: Service robots are intended to assist humans in various settings, such as healthcare, hospitality, household chores, and personal assistance.
    - Medical Robots: These robots are used in medical settings for tasks such as surgery, diagnostics, rehabilitation, and patient care.
    - Agricultural Robots: Agricultural robots are used in farming and agriculture for tasks like harvesting, planting, crop monitoring, and livestock management.
    - Exploration Robots: These robots are deployed in environments that are difficult or dangerous for humans, such as space exploration, deep-sea exploration, or hazardous environments.
2. Based on Mobility:
    - Stationary Robots: Stationary robots remain fixed in a particular location and perform tasks within their designated workspace.
    - Mobile Robots: Mobile robots have the ability to move and navigate autonomously or with minimal human intervention. They can operate in various environments and may include wheeled robots, legged robots, drones, or autonomous vehicles.
3. Based on Physical Configuration:
    - Manipulator Robots: These robots consist of one or more arms with multiple joints, often with an end-effector (such as a gripper or tool) for performing tasks.
    - Humanoid Robots: Humanoid robots are designed to resemble and mimic human form and movements, with capabilities for walking, grasping objects, and interacting with the environment.
    - Wheeled Robots: These robots move on wheels or tracks and are commonly used for navigation on flat surfaces.
    - Legged Robots: Legged robots use legs for mobility, enabling them to traverse challenging terrain or navigate uneven surfaces.
    - Aerial Robots: Aerial robots, or drones, are robots that fly or hover in the air, often equipped with cameras or other sensors for aerial surveillance, inspection, or delivery tasks.
4. Based on Autonomy Level:
    - Teleoperated Robots: These robots are controlled remotely by a human operator.
    - Semi-Autonomous Robots: These robots can perform tasks autonomously to some extent but still require human intervention or supervision in certain aspects.
    - Fully Autonomous Robots: Fully autonomous robots can operate and make decisions independently without human intervention.
5. Based on Complexity:
    - Simple Robots: These robots have basic functionalities and are often designed for educational purposes or simple tasks.
    - Complex Robots: Complex robots are advanced systems that incorporate sophisticated hardware, sensors, and software for performing intricate tasks or interacting with complex environments.

It's important to note that these classifications are not mutually exclusive, and a robot can belong to multiple categories depending on its characteristics and capabilities.

18. Definition of ZMP?

The Zero Moment Point (ZMP) is a concept used in robotics and biomechanics to determine the stability of a walking or running humanoid robot or a bipedal creature. It is defined as the point on the ground where the net external horizontal forces and moments acting on the system are balanced.

When a humanoid robot or a bipedal creature is in motion, it experiences various forces and moments that can cause it to lose balance and fall. The ZMP is the point on the ground where the resultant moment is zero, meaning that the forces acting on the robot or creature are in equilibrium. By controlling the ZMP, the stability and balance of the robot or creature can be maintained.

To achieve stable walking or running, the ZMP should always be kept within the support polygon, which is the area defined by the contact points of the feet with the ground. If the ZMP moves outside the support polygon, it indicates an imbalance, and the robot or creature needs to take corrective measures to regain stability, such as adjusting the position of its feet or altering its walking pattern.

The ZMP concept is widely used in the field of robotics to develop control algorithms and strategies for maintaining balance and stability in humanoid robots. By continuously calculating and controlling the ZMP, robots can walk, run, or perform other dynamic motions while avoiding falls and maintaining stability.

https://slideplayer.com/slide/8522198/

19. Give general characteristics of control methods used in walking robots?

There are several control methods used in walking robots, each with its own set of characteristics. Here are some general characteristics of the most common control methods:

1. Open-loop control: This method involves pre-programmed sequences of movements that are executed without feedback from sensors. It is simple and efficient, but lacks adaptability to changing environments.
2. Closed-loop control: This method uses feedback from sensors to adjust the robot's movements in real-time. It is more adaptable to changing environments, but can be more complex and computationally intensive.
3. Central pattern generators (CPGs): CPGs are neural networks that generate rhythmic patterns of movement. They are often used in legged robots to generate walking or running gaits. CPGs are robust and adaptable, but can be difficult to tune.
4. Reinforcement learning: This method involves training the robot to learn from its own experiences through trial and error. It can be highly adaptable and efficient, but requires significant computational resources and can be difficult to implement.
5. Hybrid control: This method combines multiple control methods to take advantage of their strengths and mitigate their weaknesses. For example, a robot may use open-loop control for basic movements and closed-loop control for more complex tasks.

Overall, the choice of control method depends on the specific requirements of the robot and its environment.

20. Dynamic stability of bipeds

The dynamic stability of bipeds refers to the ability of a bipedal system (such as a human or a humanoid robot) to maintain balance and prevent falling during walking or running motions. Achieving dynamic stability involves controlling the body's motion and adjusting the center of mass (CoM) position in response to external disturbances and changes in the environment.

Here are some key factors and concepts related to the dynamic stability of bipeds:

1. Zero-Moment Point (ZMP):
    - The Zero-Moment Point is a concept used to analyze and control the dynamic stability of bipedal locomotion. It refers to a point on the ground where the total moment acting on the biped is zero. It provides a reference point for maintaining balance and preventing falling.
    - The ZMP is calculated based on the ground reaction forces and the center of mass (CoM) position of the biped. By controlling the ZMP, stability can be maintained by adjusting the CoM position relative to the support polygon formed by the feet.
2. Capture Point:
    - The Capture Point is another concept used in the analysis of dynamic stability in bipeds. It represents a point on the ground that, if the biped moves to it, it will be instantaneously at rest.
    - The Capture Point takes into account the dynamics of the bipedal system and predicts where the CoM should be placed to achieve stable walking or running. It is often used for real-time control of bipedal locomotion.
3. Feedback Control:
    - Feedback control techniques are commonly used to achieve dynamic stability in bipeds. Sensors such as force/torque sensors, joint encoders, and inertial measurement units (IMUs) provide information about the state of the biped, which is then used to generate control signals.
    - Feedback control systems continuously monitor and adjust the joint angles, torques, and CoM position to maintain stability. Proportional-Integral-Derivative (PID) controllers or more advanced control algorithms can be used to achieve stable and robust walking or running.
4. Gait Planning and Control:
    - Gait planning and control algorithms play a crucial role in maintaining dynamic stability. These algorithms determine the trajectory of the CoM, joint angles, and torques based on the desired walking speed, terrain, and environmental conditions.
    - Adaptive gait control strategies, such as adjusting step length, step frequency, and step width, can be employed to respond to perturbations and ensure stability during locomotion.
5. Mechanical Design:
    - The mechanical design of the bipedal system also influences its dynamic stability. Factors such as the distribution of mass, joint stiffness, and damping properties can impact stability and the ability to recover from disturbances.
    - In humanoid robotics, advancements in mechanical design, such as the use of compliant joints, passive dynamics, or active balance mechanisms, can enhance the dynamic stability of bipedal robots.

Achieving dynamic stability in bipeds is a complex and ongoing research area, involving a combination of control strategies, feedback systems, gait planning, and mechanical design considerations. Researchers and engineers continuously explore new approaches to improve the stability and robustness of bipedal locomotion, enabling human-like walking or running in humanoid robots and contributing to advancements in prosthetics and rehabilitation technologies.

21. What will be the future trend in development of robotics? Why?

Autonomous Mobile Robots

Intelligent Robots

Cobots

Robotics Cybersecurity

Drones

Internet of Robotic Things

Humanoid Robots(health care)

Automated Guided Vehicles

Assistive Robots

Cobot use will continue to rise

Autonomous mobile robots are arriving

Electronics assembly automation will grow

More intelligent and adaptive robotics

22. Position control versus torque control. What method is commonly used in industrial manipulators.

In industrial manipulators, both position control and torque control methods are commonly used, but the choice depends on the specific requirements and application of the manipulator. Let's explore the characteristics and typical applications of each control method:

1. Position Control:
   - Method: Position control aims to accurately control the position and trajectory of the manipulator's end effector. It involves specifying the desired position or path, and the control system adjusts the manipulator's joints to achieve the desired position.
   - Characteristics: Position control provides precise positioning of the end effector, allowing for tasks such as pick-and-place operations, assembly, and precise positioning of objects. It is widely used in applications where accuracy, repeatability, and path following are crucial.
   - Control Techniques: Position control can be achieved using various techniques, such as PID (Proportional-Integral-Derivative) control, model-based control, or advanced control algorithms like adaptive control or model predictive control (MPC).
2. Torque Control:
   - Method: Torque control focuses on controlling the torque or force exerted by the manipulator's joints or end effector. It enables precise force regulation and impedance control, allowing the manipulator to interact with its environment or perform tasks requiring specific force profiles.
   - Characteristics: Torque control is beneficial for tasks like robotic material handling, force-based assembly, contact tasks (e.g., polishing, grinding), and applications where the manipulator needs to interact with external forces or comply with external constraints.
   - Control Techniques: Torque control can be achieved through techniques such as force/torque sensing, force feedback control, or impedance control. These techniques enable the manipulator to maintain a desired force level or adapt its behavior based on interaction forces.

In practice, industrial manipulators often employ a combination of position and torque control methods. For example, in a robotic arm, position control is commonly used for reaching and positioning the end effector accurately, while torque control is employed for tasks that involve force sensing or interaction with the environment.

The control method used depends on the specific application requirements. Some manipulators may primarily rely on position control, while others may heavily emphasize torque control. Advanced control strategies, such as hybrid position/force control, can also be employed to achieve a balance between precise positioning and force regulation.

It's important to note that industrial manipulators can have complex control systems that integrate position and torque control methods, along with other control techniques, to optimize performance and achieve the desired tasks in industrial settings.

23. What is a servo motor? What is the difference between typical DC motors with encoders and servo motors? Which one will provide more accurate control?

What is a Servo Motor?

A type of electric motor that functions as a rotary actuator and allows for a precise control of angular velocity, position and acceleration is known as **servo motor**.

A servo motor consists of a sensor for position feedback on their rotor to control its position and speed at all times. Thus, the servo motors are generally characterized by their closed-loop control system design.

There are two types of servo motors: **AC servo motors** and **DC servo motors**. AC servo motors are used in large applications, while DC servo motors are used in small electronic devices like robotic joints, camera autofocus systems, antenna positioning systems, CNC machines, etc.

What is a DC Motor?

As its name suggests, a **DC motor** is the type of electric motor that converts the electrical energy in form of direct current (DC) into mechanical energy (rotation of shaft).

The DC motors have two types of constructions namely **brushed DC motor** and **brushless DC motor**. But, when we use the keyword DC motor, it simply means that we are talking about brushed DC motor.

A DC motor consists of a stator part which forms the magnetic field system and a rotor part that acts as the armature of the motor. A commutator is also mounted on the shaft of the rotor. The commutator segments are connected to the ends of the armature winding. The carbon brushes are used to make the electrical contact with the commutator and the armature. Therefore, in order to start the DC motor, a direct current is allowed to pass into the armature winding through the commutator and brushes.

The electric current flowing through the armature winding produces a magnetic field which interacts with the magnetic field of stator to produce a torque on the rotor. Due to this torque, the rotor spins to produce output mechanical energy at the shaft.

The DC motors are commonly used in electronic devices, cranes, toys, elevators and lifts, power tools, and in many other appliances.

Difference between Servo Motor and DC Motor

The following table highlights all the significant differences between a servo motor and a DC motor

Basis of Difference	Servo Motor	DC Motor
Definition	A servo motor is a rotatory actuator that allows for a precise control of	A DC motor is a type of electrical machine that uses electrical

	angular position, velocity and acceleration.	energy to convert it into mechanical energy for driving a mechanical load.
Number of wires	There are three wires in a servo motor namely power, control and ground.	DC motors have only two wires, namely, power and ground.
System components	The assembly of a servo motor involves four major parts: a suitable motor, gearing set, position sensor (encoder) and a control circuit.	In the case of a DC motor, there is only an individual DC machine with no extra components.
Speed	The speed of servo motors is high, typically from 1000 RPM to 6000 RPM.	DC motors have moderate speed, it depends on the type of the motor.
Control mechanism	Servo motors involve complex control mechanism. However, they are highly controllable.	The control mechanism of DC motors is simple. We just need to reverse the leads to change the direction of rotation and change the voltage to change the speed.
Efficiency	The efficiency of servo motors is high.	The efficiency of DC motors is average and comparatively less than that of servo motors.
Suitability	Servo motors are most suitable for intermittent applications.	DC motors are suitable for continuous applications.
Reliability	Servo motors are highly reliable.	The reliability of DC motors is moderate.
Cost	Servo motors are relatively expensive.	DC motors are less costly than servo motors.
Applications	Servo motors are extensively used in industrial automation, robotic joins, camera autofocus systems, and many other positioning systems.	DC motors are used to drive mechanical loads. These are widely used in cranes, elevators, lifts, toys, kitchen appliances, etc.

Conclusion

Conclusion

The most important difference that you should note here is that a servo motor is used in positioning systems, while a DC motor is used to drive the usual mechanical loads.

24. What are the DD motors? (DD - Direct Drive) What is the difference between DD and DC motors based control? When DD are used?

Here are some passages from relevant books that answer your questions:

Passage	Book Name	Application in Real Life	Key Learnings
"DD (Direct Drive) motors are electric motors that directly drive a load without any mechanical transmission elements such as gears or belts. In DD motors, the stator and rotor are combined to form a single unit, which	"Electric Motors and Drives: Fundamentals, Types and Applications" by Austin Hughes	Used in high-precision applications where accuracy and reliability are critical factors, such as robotics or medical equipment. They can also be used in industrial automation systems and machine tools where precise motion control is required.	DD motors offer several advantages over conventional motor designs including higher efficiency, lower maintenance requirements, quieter operation due to their reduced number of

eliminates the need for gearboxes or other mechanical components."			moving parts. However they require more complex control systems than DC motors.
"DC motor speed control techniques include varying the voltage applied to the armature windings (field-controlled), varying the field current through the stator winding (armature-controlled), or using pulse-width modulation (PWM) techniques."	"Electric Motors & Control Techniques" by Irving M. Gottlieb	Compared to DC motor control techniques , Direct drive requires relatively complex feedback loops with position sensors/encoders/accelerometers etc., making it more expensive but providing benefits like smoother operation and precise speed regulation.	When designing a system with direct-drive technology one should consider its complexity of design compared to traditional methods when considering cost-effectiveness of implementation .

In summary, DD (Direct Drive) motors directly drive loads without needing additional mechanical components like gearing or belts. The main difference between DD and DC motor lies in their speed control technique. While DC uses simple voltage variation method ,DD requires relatively complex feedback loops with position sensors/encoders/accelerometers etc. DD motors are generally used in high-precision applications where accuracy and reliability are critical factors, such as robotics or medical equipment.

25. How is defined static stability in walking machines

Static stability in walking machines refers to the ability of a robot to remain upright when at rest or under acceleration and deceleration Source 6. A statically stable robot is well balanced and does not fall over when standing, meaning that its center of gravity is within its ground contact base Source 4. The minimum number of ground contact points required for a statically stable robot is three Source 4.

For walking machines with at least four legs, several static and dynamic stability criteria have been defined. However, there is no stability margin that accurately predicts robot stability when inertial and manipulation effects are significant Source 1 Source 2. In such cases, every momentum-based stability margin fails, and the use of an unsuitable stability criterion yields unavoidable errors in the control of walking robots Source 2.

A new stability margin has been proposed that accurately measures robot stability, considering dynamic effects arising during motion Source 2. This stability margin is proven to be the only exact stability margin when robot dynamics and manipulation forces exist, and numerical comparison has been conducted to support the margin's suitability Source 2.

In summary, static stability in walking machines is the ability to maintain balance and remain upright when at rest or under acceleration and deceleration. For walking machines with at least four legs, it is crucial to consider both static and dynamic stability criteria, and an appropriate stability margin should be used to accurately predict robot stability under various conditions.

26. EMG signals: measurement, analysis, utilisation.

EMG (electromyography) signals are electrical signals produced by the contraction of muscles. They can be measured, analyzed, and utilized in various ways for different purposes. Here's a breakdown of each aspect:

1. Measurement:
   EMG signals are typically measured using surface electrodes or needle electrodes. Surface electrodes are placed on the skin above the muscle of interest, while needle electrodes are inserted directly into the muscle tissue. These electrodes detect the electrical activity generated by muscle contractions.

2. Analysis:
   EMG signals can be analyzed to extract valuable information about muscle activity and function. Some common analysis techniques include:

   - Amplitude analysis: Examining the magnitude or intensity of the EMG signal to determine the level of muscle activation.
   - Frequency analysis: Analyzing the spectral content of the EMG signal to assess muscle fatigue or identify specific muscle activation patterns.
   - Time-domain analysis: Analyzing the temporal characteristics of the EMG signal, such as onset time, duration, and muscle activation patterns.
   - Co-contraction analysis: Assessing the coordination and activation patterns of multiple muscles working together.

   Advanced signal processing techniques, such as wavelet analysis, Fourier transform, or machine learning algorithms, can also be employed to extract additional features or patterns from the EMG signals.

3. Utilization:
   EMG signals find applications in various fields, including:

   - Clinical diagnosis: EMG signals can help diagnose neuromuscular disorders, such as muscle dystrophy, nerve damage, or myopathy. Abnormal EMG patterns can provide insights into the underlying condition.
   - Rehabilitation and physiotherapy: EMG signals can be used to monitor muscle activity during rehabilitation exercises and assess muscle strength and coordination improvements over time.
   - Prosthetics and orthotics: EMG signals can be used as control inputs for prosthetic limbs or orthotic devices. By measuring muscle activity, users can perform specific gestures or movements, which are then translated into control commands for the devices.
   - Human-computer interaction: EMG signals can enable gesture recognition and control in human-computer interaction systems, such as virtual reality applications or gaming interfaces.
   - Sports and performance analysis: EMG signals can provide insights into muscle activation patterns, fatigue levels, and movement efficiency in athletes, helping optimize training programs and enhance performance.

It's important to note that accurate interpretation and analysis of EMG signals require expertise in signal processing and knowledge of the specific application context.

Electromyography (EMG) is a technique for evaluating and recording the electrical activity produced by skeletal muscles. EMG is performed using an instrument called an electromyograph to produce a record called an electromyogram Source 1. The EMG signals can be analyzed to detect abnormalities, activation level, or recruitment order, or to analyze the biomechanics of human or animal movement Source 1.

EMG measures the electrical activity of the muscle during rest, slight contraction, and forceful contraction. Muscle tissue does not normally produce electrical signals during rest. When an electrode is inserted, a brief period of activity can be seen on the oscilloscope, but after that, no signal should be present Source 2.

To acquire EMG signals, one or more small needles (also called electrodes) are inserted through the skin into the muscle. The electrical activity picked up by the electrodes is then displayed on an oscilloscope (a monitor that displays electrical activity in the form of waves) Source 2. There are different configurations for acquiring EMG signals, such as multipolar configurations, which use more than two detecting surfaces to acquire the EMG signal with the help of a reference electrode Source 4.

EMG signals are essentially made up of superimposed motor unit action potentials (MUAPs) from several motor units. For a thorough analysis, the measured EMG signals can be decomposed into their constituent MUAPs Source 1. The signals can be processed through techniques like rectification, which translates the raw EMG signal to a signal with a single polarity, usually positive Source 1.

EMG has various applications, including the diagnosis of neurological and neuromuscular problems, biofeedback, ergonomic assessment, and research in biomechanics, motor control, neuromuscular physiology, movement disorders, postural control, and physical therapy Source 1. In addition, EMG can be used to sense isometric muscular activity where no movement is produced, enabling the definition of a class of subtle motionless gestures to control interfaces without being noticed and without disrupting the surrounding environment Source 1.

27. Discuss the bone remodeling phenomenon

Bone remodeling is a dynamic process that occurs throughout our lives, involving the continuous removal of old bone tissue and the formation of new bone tissue.

osteoblasts - build new bone tissue

osteoclasts - break down bone

This allows bone to grow , heal, and adapt to changing condition

28. Biomechanical injury criteria (in biomechanics of impact)

In biomechanics, injury criteria are used to understand and predict the likelihood and severity of injuries based on the mechanical forces and loads applied to the human body. These criteria help researchers, engineers, and professionals in fields such as sports, automotive safety, and ergonomics design safer products and environments.

Here are some commonly used injury criteria in biomechanics:

Certainly! Here's an overview of injury criteria for head injuries, spinal injuries, thoracic injuries, and abdominal injuries:

1. Head Injury Criteria:
   a. Head Injury Criterion (HIC): The HIC is a measure used to assess the potential for head injury, particularly traumatic brain injury (TBI), in automotive crashes or other impact scenarios. It considers the acceleration and duration of head impact to estimate the likelihood of injury.
   b. Concussion Grading Systems: Various grading systems, such as the Sports Concussion Assessment Tool (SCAT) and the Glasgow Coma Scale (GCS), are used to evaluate and classify the severity of concussions and head injuries based on symptoms, cognitive function, and neurological examination.
2. Spinal Injuries:
   a. Thoracic and Lumbar Spine Injury Criteria: Injury criteria for the thoracic and lumbar spine typically consider metrics like compression force, shear force, bending moments, and specific thresholds for injury to spinal structures like vertebrae, discs, and ligaments.
   b. Spinal Cord Injury Criteria: The International Standards for Neurological Classification of Spinal Cord Injury (ISNCSCI) is commonly used to classify and evaluate spinal cord injuries based on neurological function and level of injury.

3. Thoracic Injuries:
   a. Chest Injury Criteria: Several criteria are used to evaluate thoracic injuries, such as the Chest Injury Criterion (CIC), which assesses the likelihood of injury to the chest region based on metrics like chest acceleration and deflection.
   b. Rib Fracture Criteria: Specific criteria exist for assessing the likelihood of rib fractures, considering factors like peak rib deflection, energy absorbed, and the number of fractured ribs.
4. Abdominal Injuries:
   a. Abdominal Injury Scale (AIS): The AIS is a scoring system that assigns values to different types of abdominal injuries, considering the severity and anatomical location of the injury. This scale helps evaluate and classify abdominal trauma based on factors like organ damage, bleeding, and associated injuries.

It's important to note that these injury criteria are often used in conjunction with medical imaging, clinical assessment, and other diagnostic tools to provide a comprehensive evaluation of injuries. Additionally, injury criteria may differ based on specific guidelines, research findings, and the context in which they are applied.

29. Present and discuss an algorithm of kinematic analysis program (capable to deal with an arbitrary multibody system).

Newton-Raphson method

1. Read the initial configuration approximation, q, and set variables.
2. Set t = t0.
3. Calculate the initial approximation q using a series expansion.
4. Calculate the constraint Φ(q, t) using a specific equation from example 5.5.
5. Calculate the Jacobian matrix Φq(q, t) using another equation from example 5.5.
6. Solve the set of linear equations ΦqΔq = -Φ.
7. Update q by substituting q = q + Δq.
8. If the measure of constraint violation (e.g., Φ^TΦ) is greater than an acceptable threshold, go back to step 4 and repeat the process.
9. Calculate Φt(q, t) using a particular equation from example 5.5.
10. Calculate the Jacobian matrix Φq(q, t) using the updated coordinates q from step 7.
11. Solve the set of linear equations Φqq̇ = -Φt.
12. Calculate the vector Γ(q̇, q, t) using specific equations from example 5.5.
13. Solve the set of linear equations ΦqΔq̇ = -Γ.
14. If t is less than tK, update t by adding Δt and return to step 3.

This process continues until a satisfactory solution is obtained or until a certain condition is met. The Newton-Raphson method is widely used for solving systems of equations and finding roots of nonlinear functions.

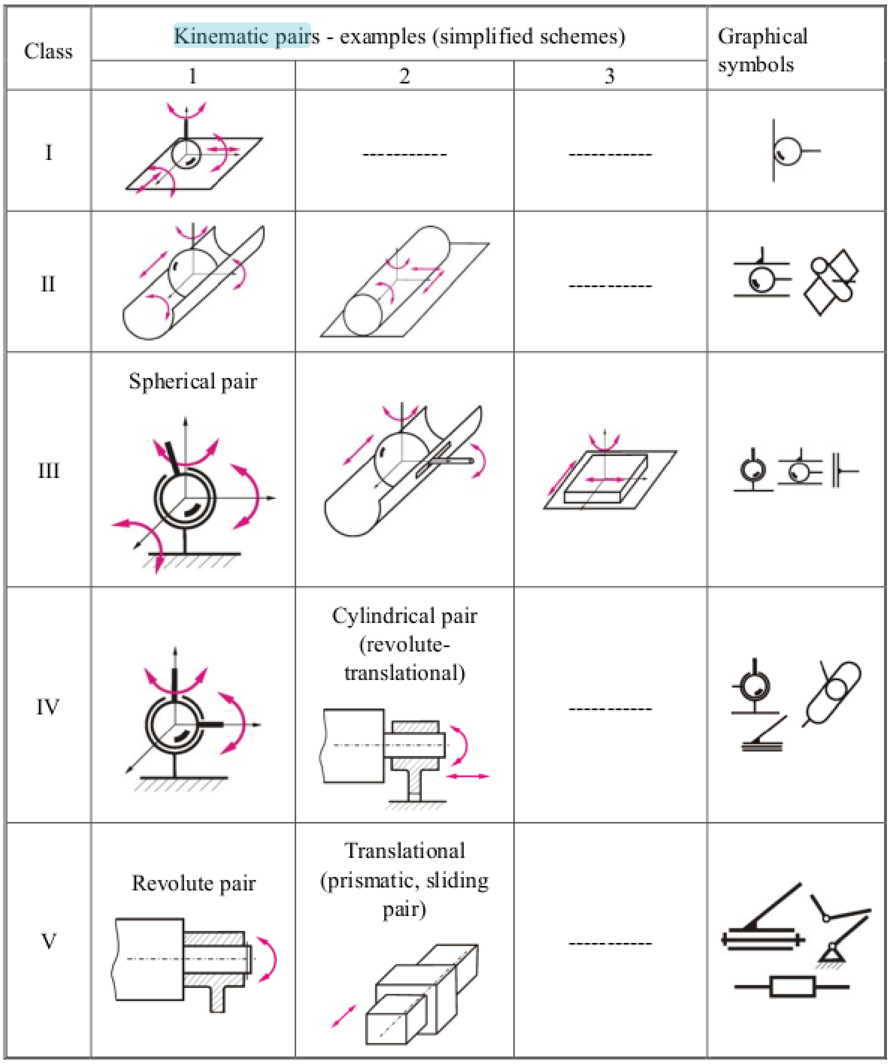
30. What are the kinematic constraints equations and how are they related with kinematic pairs?

The constraints which are imposed by all kinematic pairs of mechanism are called kinematic constraints, and can be defined by the set of holonomic constraint equations in the general form:

$\Phi K = \Phi K (q) = 0$.

Kinematic pair (joint) is a connection between two bodies. Kinematic pair provides some physical constraints on the relative motion between the two bodies. The kind of the relative motion permitted by a pair is governed by the form of the contact surfaces between the bodies.

Kinematic pairs are classified into 5 classes, from I to V depending on the number of degrees of freedom which are constrained in the relative motion (it is assumed that one of the elements is fixed and the other moves relative to it).



Class	Kinematic pairs - examples (simplified schemes) 1	2	3	Graphical symbols
I		-----------	-----------	
II			-----------	
III	Spherical pair			
IV		Cylindrical pair (revolute-translational)	-----------	
V	Revolute pair	Translational (prismatic, sliding pair)	-----------	

31. Write and discuss equations the motion of a rigid body in 2D and 3D space.

Planar

32. Logical planner in agent-based systems (situation calculus in first-order logic, plan operators, comparison of STRIPS and ADL, plan search procedure, partial-ordered plan) (Art.Intel.)

In agent-based systems, a logical planner is responsible for generating plans or sequences of actions that enable an agent to achieve its goals within a given environment. The situation calculus, expressed in first-order logic, is a common formalism used for representing and reasoning about actions and their effects in such planners. Let's explore some key concepts related to logical planning in agent-based systems:

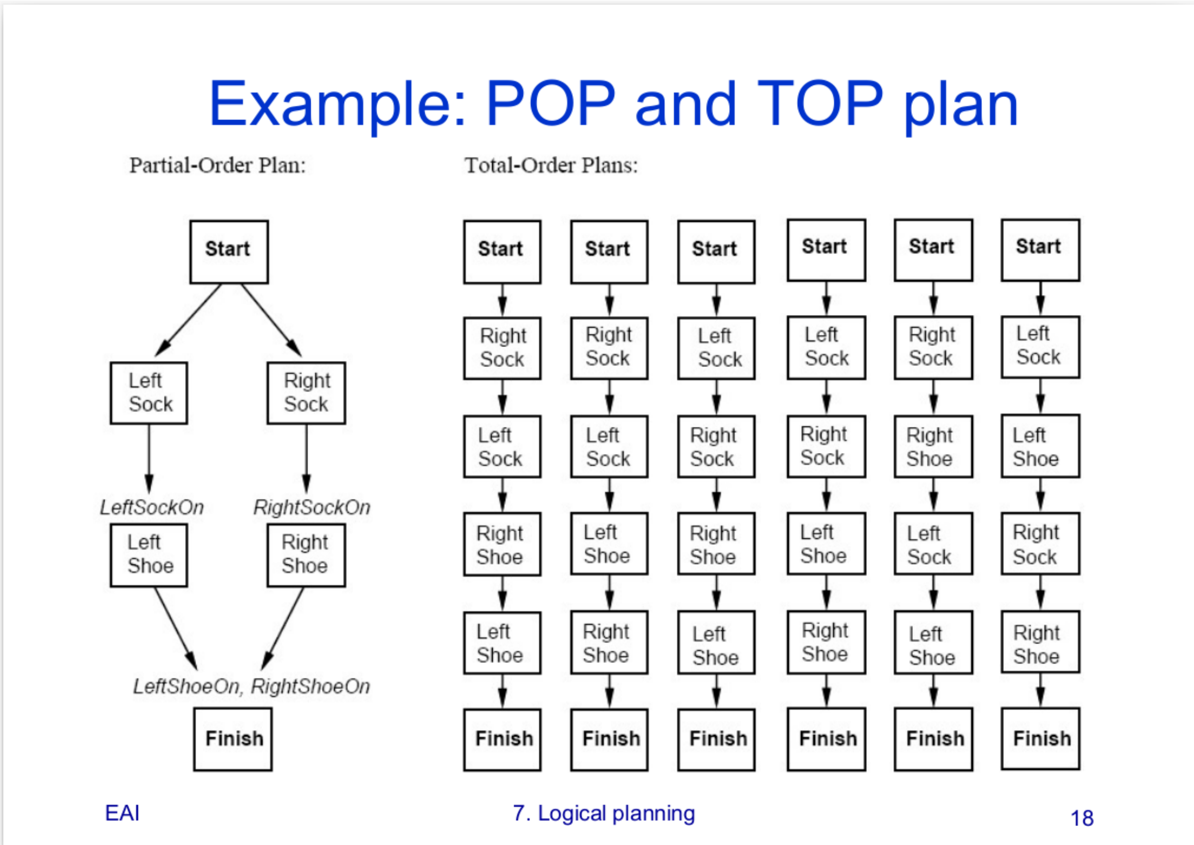
1. Situation Calculus in First-Order Logic:
   The situation calculus provides a formal framework for representing actions and their effects. It uses first-order logic to describe the state of the world and how it changes as a result of actions. The framework typically includes axioms that define the initial state, action preconditions, action effects, and the progression of time.
2. Plan Operators:
   Plan operators are used to describe actions and their effects in a planning domain. Each operator consists of a name, a set of preconditions, and a set of effects. Preconditions specify the conditions that must hold true for an action to be applicable, while effects describe the changes in the state that occur when the action is executed.
3. Comparison of STRIPS and ADL:
   STRIPS (Stanford Research Institute Problem Solver) and ADL (Action Description Language) are two widely used planning formalisms.

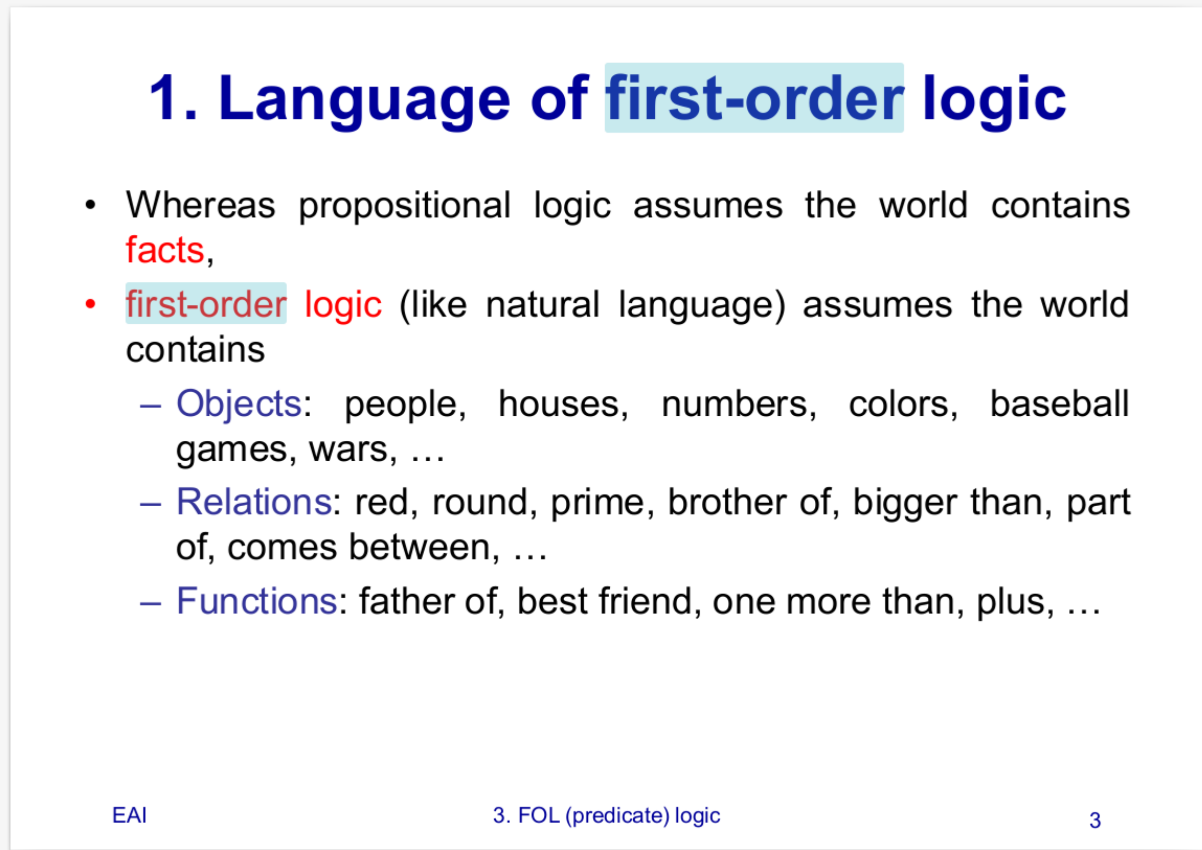
- STRIPS: STRIPS is a simple and influential planning formalism. It allows only deterministic actions and represents the preconditions and effects of actions using logical literals. It does not support the representation of negation, disjunction, or quantifiers.
- ADL: ADL is an extension of STRIPS that provides additional expressive power. It supports non-deterministic actions, negative preconditions and effects, disjunctions, and existential quantification. ADL allows for more flexible and complex planning domains.

STRIPS	ADL
Only positive literals in conditions, e.g. $At(\mathrm{Home}) \wedge Has(\mathrm{Milk})$	Positive and negative literals in conditions: e.g. $\neg At(\mathrm{home}) \wedge \neg Has(\mathrm{Milk})$
Closed world assumption: unmentioned literals are false	Open world assumption: unmentioned literals are unknown.
Effect $(P \wedge \neg Q)$ means: *add P and delete Q*	Effect $(P \wedge \neg Q)$ means: *add P and $\neg Q$, and delete $\neg P$ and Q*
Only ground literals (predicate form) in goals. E.g. $At(\mathrm{P1,WAW}) \wedge At(\mathrm{P2,WAW})$	Quantified variables in goals. E.g. $\exists x\ At(\mathrm{P1},x) \wedge At(\mathrm{P2},x)$
Goals are conjunctions. E.g. $At(\mathrm{Home}) \wedge Has(\mathrm{Milk})$	Goals allow conjunction and disjunction. E.g. $At(\mathrm{home}) \wedge (Has(\mathrm{Milk}) \vee Has(\mathrm{Banana}))$
No support for equality. No types.	Equality predicate (e.g. $\mathrm{x=y}$) is built in. Variables can have types, e.g.$(x\mathrm{: Plane})$

2. Plan Search Procedure:
   The plan search procedure involves finding a sequence of actions that transforms the initial state into a state that satisfies the agent's goals. This process typically involves a combination of search algorithms and heuristics. Common search algorithms include breadth-first search, depth-first search, A* search, and more advanced techniques like partial-order planning.

3. Partial-Ordered Plan:
   A partial-ordered plan is a representation of a plan that allows for the parallel execution of actions whenever their preconditions are met simultaneously. It represents the causal dependencies between actions and captures the temporal constraints between them. A partial-ordered plan is often represented as a directed acyclic graph (DAG), where actions are nodes, and edges represent the causal relationships between them.





33. Stochastic inference in Dynamic Bayesian Networks (purpose and definition of Bayesian networks, Markov processes and first-order assumptions, DBN and its inference tasks, HMM, Kalman Filter and Particle Filter) (Art.Intel.)

Conditional independence

Definition.

Variables X, Y are conditionally independent given variable Z iff:

$$\boxed{P(X, Y \mid Z) = P(X|Z)\, P(Y|Z)}$$

Conclusions

1. Independence of X and Y **does not imply** their conditional independence.
2. Conditional independence $(X, Y \mid Z)$ **does not imply** the independence of X and Y.

EAI 9. Bayesian nets 5

3. Bayesian net

- A simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions
- Syntax:
  - a set of nodes, one per variable
  - a directed, acyclic graph (link ≈ "directly influences")
  - Incoming links of given node represent a conditional distribution for this node given its parents:

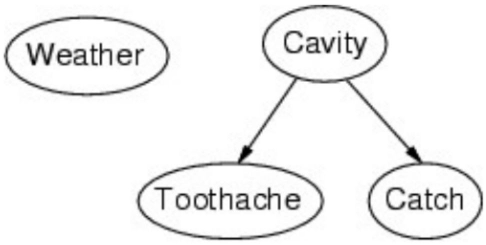
$$\mathbf{P}\,(X_i \mid \text{Parents}\,(X_i))$$

- In the simplest case, conditional distribution is represented as a conditional probability table (CPT), giving the distribution over X_i for each combination of parent values.

EAI 9. Bayesian nets 9

Example

- Topology of network encodes **conditional independence** assertions:



- *Weather* is independent of the other variables
- *Toothache* and *Catch* are conditionally independent given *Cavity*

EAI 9. Bayesian nets 10

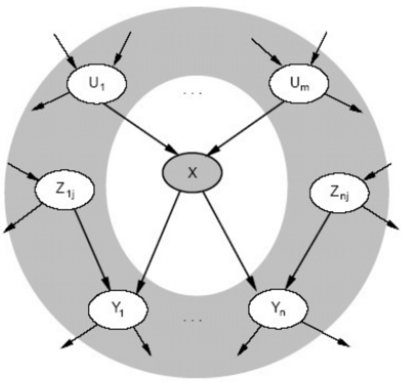
Markov blanket

A stronger formulation of local semantics: a node is conditionally independent of all the nodes in the network outside of its **Markov blanket**.

Markov_blanket(X) =

parents(X) + *children*(X) + *parents*(*children*(X)) - X.

E.g., Markov blanket for X
is denoted by grey area,
i.e., **if** B $\notin$ **Markov_blanket**(X)
then for every Z:
p(X, A | Z) = p(X |Z) p(B | Z)



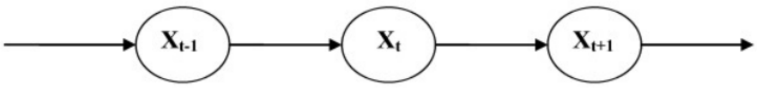
EAI 9. Bayesian nets 16

Markov process (Markov chain)

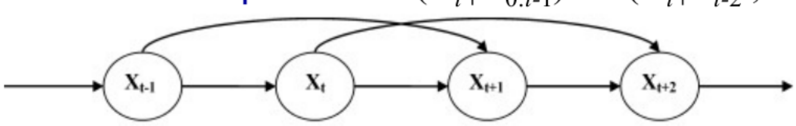
It answers the question: how to construct a Bayes net from state and evidence variables in time - what are the parents?

Markov assumption: X_t depends on a bounded subset of $X_{0:t-1}$

First-order Markov process: $P(X_t | X_{0:t-1}) = P(X_t | X_{t-1})$



Second-order Markov process: $P(X_t | X_{0:t-1}) = P(X_t | X_{t-2}, X_{t-1})$



Sensor Markov assumption: $P(E_t | X_{0:t}, E_{0:t-1}) = P(E_t | X_t)$

States are parents of **evidence** variables.

EAI 11. Dynamic Bayesian nets 4

DBN, HMM, Kalman Filter, and Particle Filter are all important concepts in the field of artificial intelligence and probabilistic modeling. They are used for various inference tasks in different domains. Let's explore each of them briefly:

1. Dynamic Bayesian Network (DBN):
   A Dynamic Bayesian Network is a probabilistic graphical model that represents the dependencies between random variables over time. It extends the concept of a Bayesian network to model temporal relationships. DBNs are commonly used for sequential data modeling and prediction tasks. Inference tasks in DBNs include filtering (estimating the current state given all the observations up to that time), prediction (estimating the future state given all previous observations), and smoothing (estimating past states given all the observations).
2. Hidden Markov Model (HMM):
   A Hidden Markov Model is a statistical model that represents a system with unobserved states that can only be indirectly observed through a sequence of observable outputs. HMMs are widely used for problems involving temporal data, such as speech recognition, natural language processing, and bioinformatics. Inference tasks in HMMs include decoding (finding the most likely sequence of hidden states given the observations), learning (estimating the model parameters from the observed data), and evaluation (computing the likelihood of a sequence of observations given the model).
3. Kalman Filter:
   The Kalman Filter is an optimal recursive algorithm for estimating the state of a dynamic system from a series of noisy observations. It is widely used in various fields, including control systems, robotics, and navigation. The Kalman Filter performs a recursive Bayesian estimation by combining predictions from a system model with measurements from sensors. It provides an estimate of the current state and updates it as new measurements become available. The Kalman Filter is particularly efficient for linear Gaussian systems.
4. Particle Filter:
   The Particle Filter, also known as Sequential Monte Carlo (SMC) method, is a non-parametric filtering algorithm used to estimate the state of a system based on a sequence of observations. Unlike the Kalman Filter, the Particle Filter is capable of handling nonlinear and non-Gaussian systems. It represents the state estimate using a set of particles, where each particle carries a weight that represents its likelihood. Inference tasks in Particle Filters include prediction, filtering, and smoothing. The Particle Filter is commonly used in applications such as simultaneous localization and mapping (SLAM) in robotics and target tracking.

These inference tasks and filtering algorithms play crucial roles in probabilistic modeling and decision-making in artificial intelligence and are widely used in various domains to handle uncertainties and make informed predictions and estimations based on observed data.

34. Camera calibration (coordinate systems and transformations, intrinsic and extrinsic camera parameters, parameter estimation procedure, non-linear distortions) (Computer Vision)

Camera calibration is a fundamental task in computer vision that involves estimating the intrinsic and extrinsic parameters of a camera to accurately relate the 3D world coordinates to the corresponding 2D image coordinates. The calibration process enables accurate measurements and enables various computer vision tasks such as object tracking, 3D reconstruction, and augmented reality.

Let's explore the key concepts and steps involved in camera calibration:

1. Coordinate Systems and Transformations:
    - World Coordinate System (WCS): It represents the 3D world coordinates of a scene.
    - Camera Coordinate System (CCS): It represents the 3D coordinates with respect to the camera's optical center.
    - Image Coordinate System (ICS): It represents the 2D pixel coordinates in the image plane.
    - Transformations: The transformations involve converting points between these coordinate systems. The transformation matrices include rotation, translation, and projection matrices.
2. Intrinsic Camera Parameters:
    - Focal Length (fx, fy): The focal length determines the camera's zooming capability.
    - Principal Point (cx, cy): It represents the principal point or optical center of the camera.
    - Skew Coefficient (s): It accounts for any non-perpendicularity between the image axes.
    - Distortion Coefficients: These coefficients correct for lens distortion effects such as radial and tangential distortions.
3. Extrinsic Camera Parameters:
    - Rotation (R): It describes the camera's orientation in the world coordinate system.
    - Translation (t): It represents the camera's position in the world coordinate system.
4. Parameter Estimation Procedure:
    - Image Acquisition: Capture a set of calibration images with known calibration patterns (e.g., checkerboard) from different viewpoints.
    - Feature Extraction: Detect and extract the corners or keypoints from the calibration pattern in the images.
    - Correspondence Estimation: Establish correspondences between the detected features in the 2D image and their known 3D coordinates in the calibration pattern.
    - Parameter Estimation: Solve the calibration equations to estimate the camera's intrinsic and extrinsic parameters. This can be achieved using algorithms like Direct Linear Transform (DLT) or Zhang's method.
    - Refinement: Refine the initial parameter estimates to minimize the reprojection error, typically using optimization techniques like Levenberg-Marquardt.
5. Non-linear Distortions:
    - Radial Distortion: Radial distortion causes straight lines to appear curved. It is typically modeled using one or more radial distortion coefficients.
    - Tangential Distortion: Tangential distortion occurs due to lens imperfections, causing the image to appear skewed. It is modeled using tangential distortion coefficients.

Camera calibration is an essential step in computer vision applications to ensure accurate and reliable results. It enables precise measurements and corrects for lens distortions, allowing algorithms to work effectively in real-world scenarios.

35. Stereo-vision (principle, normalization (registration) of the stereo-pair, epipolar constraint, solving the correspondence problem for depth-map estimation) (Computer Vision)

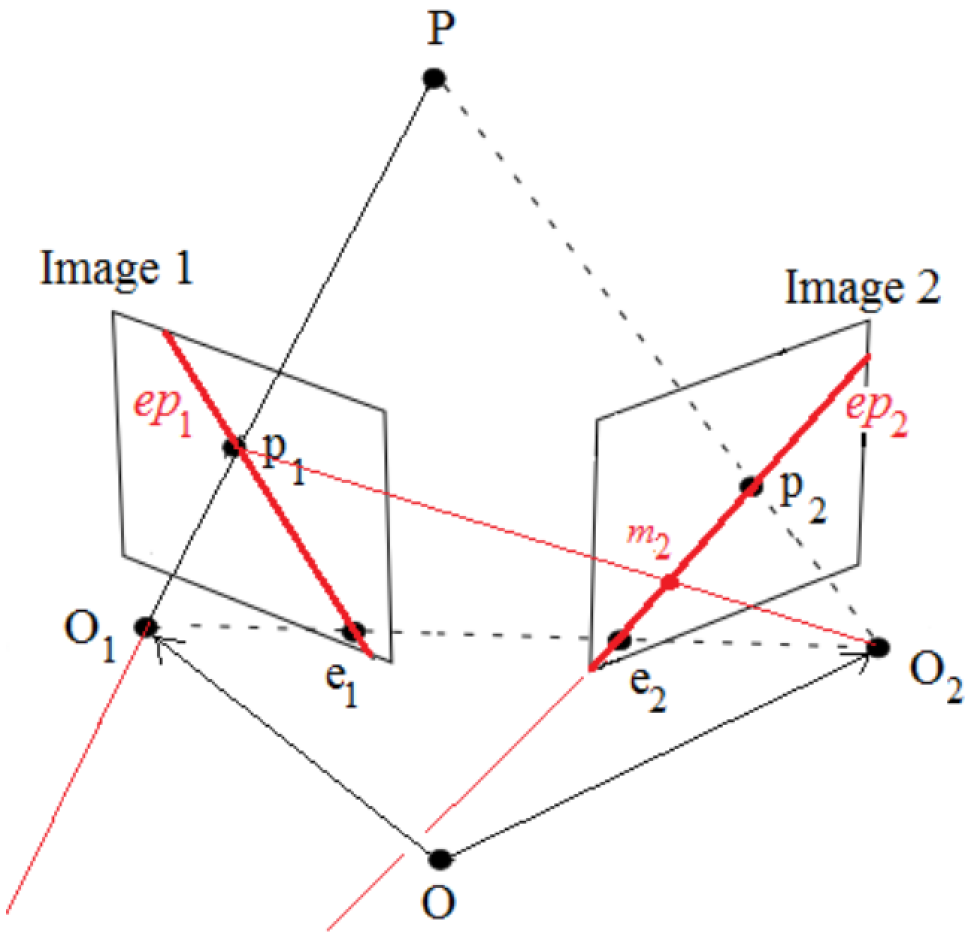
Stereo vision is a technique used in computer vision to extract depth information from a pair of stereo images. It leverages the principle of triangulation, which states that the relative positions of corresponding points in the two images can be used to estimate the depth or distance of those points from the camera.

To perform stereo vision, several steps are involved, including normalization or registration of the stereo pair, applying the epipolar constraint, and solving the correspondence problem to estimate a depth map.

1. Normalization (Registration) of the Stereo Pair:
   In stereo vision, it is important to ensure that the two images are correctly aligned or registered. This involves compensating for any differences in camera position, orientation, or lens distortion. The goal is to bring corresponding points in both images onto the same pixel coordinates. Common techniques for registration include camera calibration, where intrinsic and extrinsic parameters of the cameras are estimated, and rectification, which transforms the images so that the epipolar lines become horizontal and correspondences lie on the same scanline.

   Normalization, or rectification, of a pair of stereo images is a preprocessing step used to align and rectify the images in such a way that corresponding features become horizontally aligned. This process simplifies the task of matching features between the two images, making subsequent stereo correspondence algorithms more effective.

2. Epipolar Constraint:
   The epipolar constraint is based on the geometric relationship between two cameras capturing a stereo pair. It states that the corresponding points in the left and right images must lie on the same epipolar line. An epipolar line is the intersection of the image plane with the plane defined by the camera centers and a 3D point. By restricting the search for correspondences along the epipolar lines, the matching problem becomes one-dimensional instead of two-dimensional, simplifying the task of finding corresponding points.



Epipolar lines ep_1, ep_2

Key points in the diagram:

- **O_1 and O_2**: These are the optical centers (or camera centers) of two cameras (or two viewpoints).
- **P**: A point in the real-world scene being captured by both cameras.
- **p_1 and p_2**: These are the projections of point P onto the two image planes of the cameras. p_1 is on Image 1 and p_2 is on Image 2.
- **e_1 and e_2 (epipoles)**: These are the points of intersection of the line joining the camera centers (O_1O_2) with the image planes. An epipole is essentially the projection of one camera's optical center onto the other camera's image plane. All epipolar lines pass through the epipole.
- **ep_1 and ep_2 (epipolar lines)**: These are the lines on the image planes that represent the projection of the ray connecting the optical center of one camera to a point in the other camera's image plane. Each point in one image has its corresponding epipolar line in the other image.

Epipolar Line:

- For every point in one image, there is a corresponding line (epipolar line) in the second image where the matching point can be found. This significantly reduces the complexity of the search from 2D to 1D.
- In the context of the diagram, if you pick point p_1 in Image 1, the corresponding point p_2 in Image 2 must lie on the epipolar line ep_2.

Epipole:

- The epipole is a unique point in each image. As mentioned above, it's the projection of one camera's optical center onto the image plane of the other camera. All epipolar lines in an image intersect at the epipole.

$P_2^T \hat{T} R P_1 = 0$

3. Solving the Correspondence Problem for Depth-Map Estimation:
   The correspondence problem refers to the challenge of determining which points in one image correspond to the points in the other image. In stereo vision, this problem is typically solved by matching image features or pixels between the left and right images. Common techniques for solving the correspondence problem include block matching, where local image regions are compared, and feature-based methods, where distinctive features (e.g., corners, edges) are matched.

Once correspondences are established, the depth map can be estimated by triangulating the corresponding image points using the camera baseline (the distance between the two cameras) and the disparity (horizontal pixel offset) between the matched points. The disparity represents the apparent shift of a point between the left and right images and is inversely proportional to the depth.

By repeating this process for multiple points in the stereo pair, a dense depth map can be generated, providing a 3D representation of the scene.

Stereo vision has numerous applications, including depth estimation for robotic navigation, 3D reconstruction, augmented reality, and autonomous vehicles.

36. Explain the DFT and FFT transforms (purpose and definition of DFT, inverse DFT, interlace decomposition and butterfly computation in FFT, aliasing problem, computational complexity of DFT and FFT) (Signal Processing)

Given the time domain signal, the process of calculating the frequency domain is called decomposition, analysis, or simply the **DFT**.

Knowing the frequency domain, calculation of the time domain is called synthesis or the **inverse DFT**.

DFT basis functions- a set of sine and cosine waves with unity amplitude:

$c_k(t) = \cos(2\pi k t / N)$

$s_k(t) = \sin(2\pi k t / N)$

where the parameters, k, and N, determine the frequency of the wave.
In an N point DFT, $k \in [0,1,\ldots, N/2]$.

Calculating the DFT

- A correlation based DFT if the DFT has less than 32 points;
- A FFT (Fast Fourier Transform) is preferred otherwise.

1. Applications of the DFT
   a. The DFT can calculate a signal's frequency spectrum.
   b. The DFT can find a system's frequency response from the system's impulse response, and vice versa.

This allows systems to be analyzed in the frequency domain.

c. The DFT can be used as an intermediate step in signal processing. The classic example of this is FFT convolution, a fast algorithm for convolving signals.

2. Spectral Anlysis of signals

Many things oscillate in our universe. For example,

- speech is a result of vibration of the human vocal cords;
- ship's propellers generate periodic displacement of the water, and so on.

   1. The shape of the time domain signal is not important in these signals.
   2. The key information is in the frequency, phase and amplitude of the component sinusoids.
   3. The DFT is used to extract this information.

3. Convolution via the Frequency Domain

Given an input signal and impulse response, we need to find the resulting output signal and do not want to do convolution:

1. Transform the two signals into the frequency domain,
2. Multiply them, and then
3. Transform the result back into the time doman

An i**nterlaced decomposition** is used each time a signal is broken in two, that is, the signal is separated into its even and odd numbered samples. The best way to understand this is by inspecting Fig. 12-2 until you grasp the pattern.

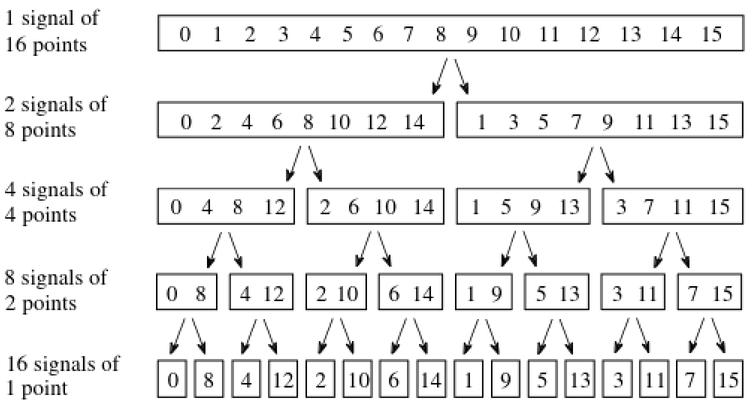


FIGURE 12-2
The FFT decomposition. An N point signal is decomposed into N signals each containing a single point. Each stage uses an *interlace decomposition*, separating the even and odd numbered samples.

The phenomenon of sinusoids changing frequency during sampling is aliasing.

Discrete periodic signals may have a problem of aliasing.

The DFT formula requires O(M2) operations, whereas the Fast Fourier Transform (FFT) will be of complexity O(M * log2(M)).

37. Explain the digital filter types: FIR and IIR filter (linear time-invariant systems, impulse response, convolution, recursive equation, transmittance function – form, poles and zeros, relation to recursive filter parameters) (Signal Processing)

7A. Digital filters

1.2 Implementation of a digital filter

Filter kernel

The most straightforward way to implement a digital filter is by *convolving* the **input signal** with the **digital filter's** ***impulse response*** (when the *impulse response* is used in this way, it is given the name **filter kernel**).

Filters carried out by convolution are called **Finite Impulse Response** or **FIR** filters.

Recursive filter

Another way to make digital filters is called **recursion**.

- Recursive filters are using previously calculated values from the ***output***, besides points from the ***input***.
- Recursive filters are defined by a set of **recursion coefficients**.

W.Kasprzak: *Signal Processing*. 7A - 7

Impulse response of recursive filters

The impulse responses of recursive filters are composed of **sinusoids** that **exponentially decay** in amplitude.

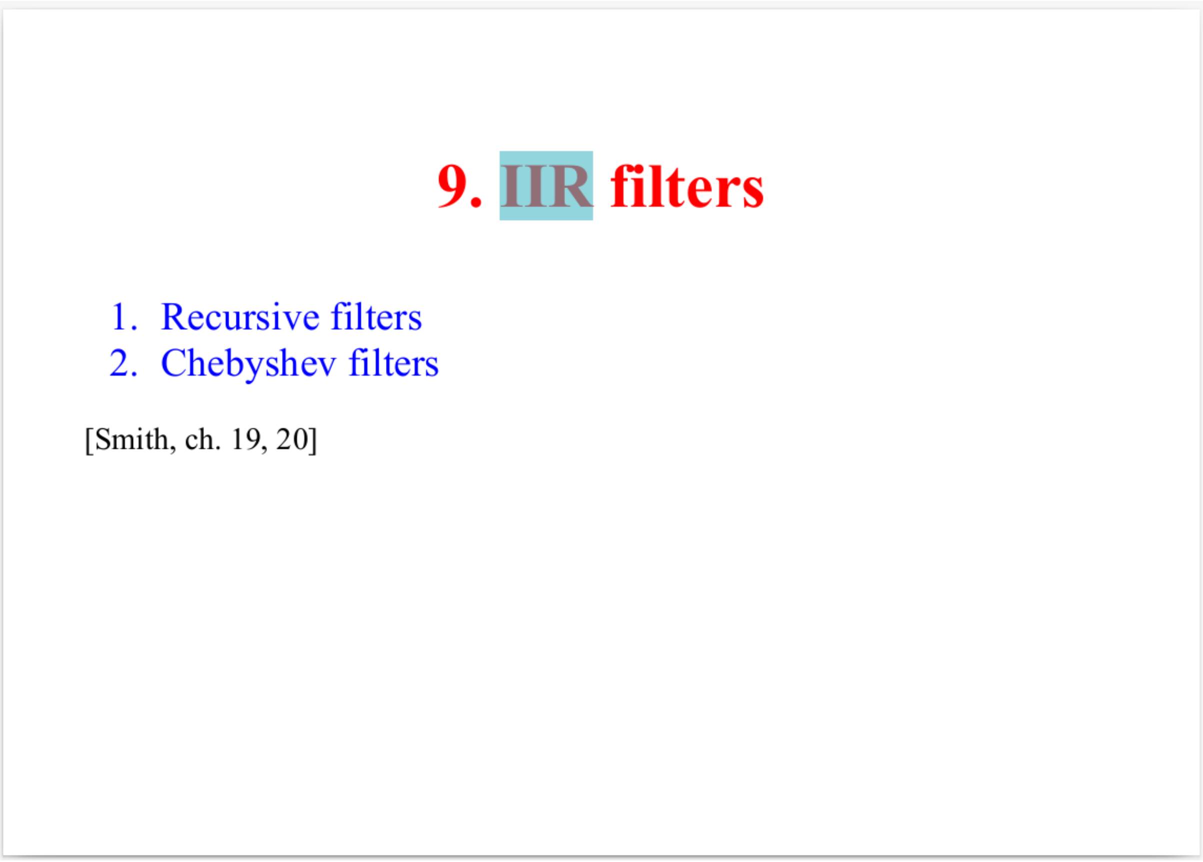
- This makes their impulse responses ***infinitely long***.
- After the amplitude drops below the round-off noise of the system, the remaining samples can be ignored.

Recursive filters are also called **Infinite Impulse Response** or **IIR** filters.

7B. FIR filter

1. Moving average filter
2. Windowed-sinc filter

[Smith, ch. 15, 16]



Transfer Function is the ratio of Laplace Transform of output to the Laplace Transform of input, when all the initial conditions are assumed to zero.

https://www.youtube.com/watch?v=AvaZ_E-nFTk

Poles and zeros

https://www.youtube.com/watch?v=Em5TuH4TVr4

https://www.youtube.com/watch?v=p1Jz0pZoKtY

https://www.youtube.com/watch?v=9yNQBWKRSs4&t=184s