We will focus on the mechanics and control of mechanical manipulators. This study will comprise of a collection of topics taken from various fields:

1. Mechanical Engineering -> methodologies for the study of machines in static and dynamic situations
2. Mathematics -> tools for describing spatial motions and other attributes of manipulators
3. Control theory -> tools for designing and evaluating algorithms to realize desired motions or force applications
4. Electrical Engineering -> design of sensors and interfaces for industrial robots
5. Computer Science -> programming robots to perform a desired task

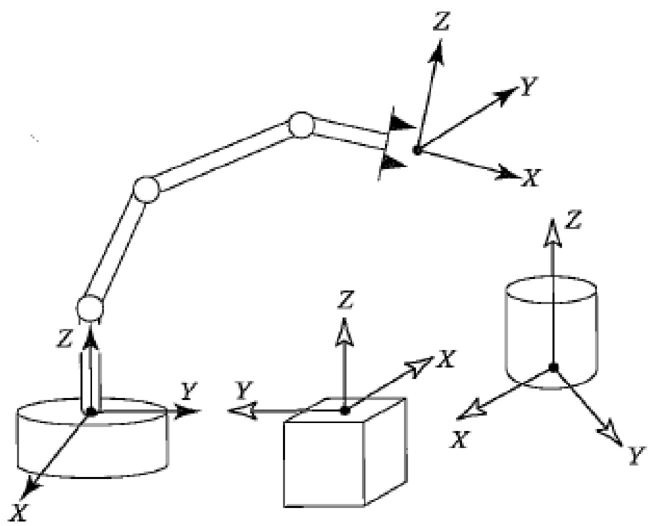
Before we jump into the lecture proper, a brief introduction of some terminologies and concepts is presented:

In the study of robotics, a very important concern is the location of objects in 3-D space. Generally, this is described using two attributes:

1. Position
2. Orientation

In order to describe the position and orientation of a body in space, a **COORDINATE SYSTEM** is generally attached rigidly to the body. Then the position and orientation of this coordinate system with respect to some reference coordinate system is used to describe the position and orientation of the body.

Note: The coordinate system is expressed using the Cartesian space, where every location in the 3-D space is expressed using an ordered triplet of axes(x, y, z). The Cartesian space is also known as the task space or operational space.



In addition to the location of objects in 3-D space, another important concern in robotics is the motion of objects. **KINEMATICS** is the science of motion without regard to the forces which cause it. Here, the emphasis is on the study of position and higher derivatives of position, such as velocity and acceleration.

The basic structure that we will study is made up of links which are connected by joints that allow motion. Generally, these joints have sensors built in to facilitate measurement and joint actuators to power each joint. The type of position information of interest depends on the type of joint used between the links. In the case of rotary or revolute joints (figure 1a), the position information is the joint angle. For sliding or prismatic joints (figure 1b), the position information is the joint offset which is basically the translation displacement.

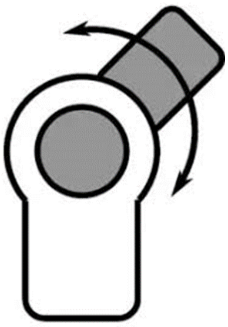


Figure 1a: Revolute Joint

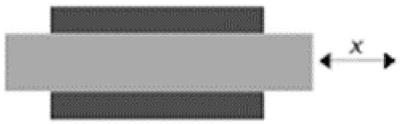
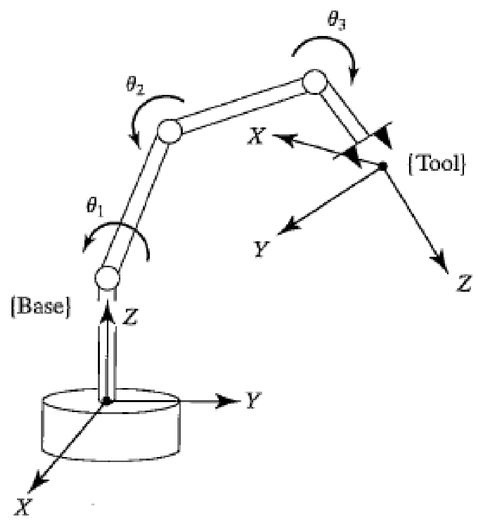


Figure 1b: Sliding Joint

Typically there exist one or more position information for any robotic structure. The number of independent position variables that would have to be specified in order to locate all parts of the structure is known as the **DEGREE OF FREEDOM** of the structure. For the standard manipulator that we will study, the number of joints equals the number of degrees of freedom. The standard manipulator consists of a series of links and joint structure which is mounted on a base. At the free end of the manipulator is the **END-EFFECTOR (TOOL)**, which depending on the intended application of the robot could be a variety of devices such as a gripper or an electromagnet. An important objective in the study of manipulators is to determine the location of the end-effector at different times. This is generally specified by giving a description of the **TOOL FRAME** relative to the **BASE FRAME**. This objective is accomplished through:

1. Forward Kinematics -> given a set of joint angles, compute the location (position and orientation) of the tool frame relative to the base frame. This is also described as changing the representation from a joint space description into a Cartesian space description.
2. Inverse Kinematics -> given the location of the end-effector (tool frame relative to base frame) of the manipulator, calculate all possible sets of joint angles that could be used to attain this given position and orientation.



In order for a robot to routinely implement either kinematics, we will need to create an algorithm, which the control computer can execute to do the calculations. The existence or nonexistence of a kinematic solution defines the **WORKSPACE** of a given manipulator. The lack of a solution means that the manipulator cannot attain the desired position and orientation because it lies outside of the manipulator's workspace. In other words, the workspace of the manipulator is the set of all locations of the end-effector at every given time.

In addition to dealing with static positioning problems, robotics also has to consider manipulators in motion. To do this, a matrix quantity called the **JACOBIAN** of the manipulator is used. The Jacobian specifies a mapping from velocities in joint space to velocities in Cartesian space. The nature of this mapping changes as the configuration of the manipulator varies. At certain points, called singularities, this mapping is not invertible.

Other important considerations are the **DYNAMICS** and **TRAJECTORY GENERATION** of robots. Dynamics is concerned with the study of the forces required to cause motion. In order to accelerate a manipulator from rest, glide at a constant end-effector velocity, and finally decelerate to a stop, a complex set of torque functions must be applied by the joint actuators. **TRAJECTORY GENERATION** is concerned with the study of ways of causing a manipulator to move from one location to another in a smooth controlled fashion. This is usually achieved by causing each joint to move as specified by a smooth function of time. Commonly, each joint starts and ends its motion at the same time, so that they appear coordinated. In order to force the end-effector to follow a pattern through space, the desired motion must be converted to an equivalent set of joint motions.

In the day to day operation of robots, errors in knowledge of the parameters of a system and disturbances that tend to perturb the system from the desired trajectory do occur. In order to mitigate these effects a **CONTROL SYSTEM** is used to automatically compensate for these errors and disturbances. Our study on control systems for manipulators will focus on linear control to develop control algorithm for the manipulator.

The ability of a manipulator to control forces of contact when it touches objects is of great importance in applying manipulators to many real-world tasks. **FORCE CONTROL** is complementary to position control. When a manipulator is moving in free space, only position control makes sense, because there is no surface to react against. However, when a manipulator is touching a rigid surface, position-control schemes can cause excessive forces to build up at the contact.

Finally, we will **PROGRAM** the manipulator. However, to do this we must first understand how motions through space, sensor-based actions, and so on are described for programming.

<u>NOTATION</u>

1. Variables written in uppercase represent vectors or matrices. Lowercase variables are scalars. For example P is a matrix or a vector while p is a scalar.
2. Leading subscripts and superscripts identify which coordinate system a quantity is written in. For example, ${}^{A}P$ represents a position vector written in coordinate system {A}.
3. Trailing superscripts are used to indicate the inverse (P^{-1})and transpose of a matrix(P^{T}).
4. Shorthand may be used for Trigonometric functions $\cos\theta_1 \equiv c\theta_1 \equiv c_1$
5. Vectors are column vectors. Hence, row vectors will have the transpose indicated explicitly. For example, If D is specified as a vector, then D is assumed to be a column vector while D^T is a row vector.