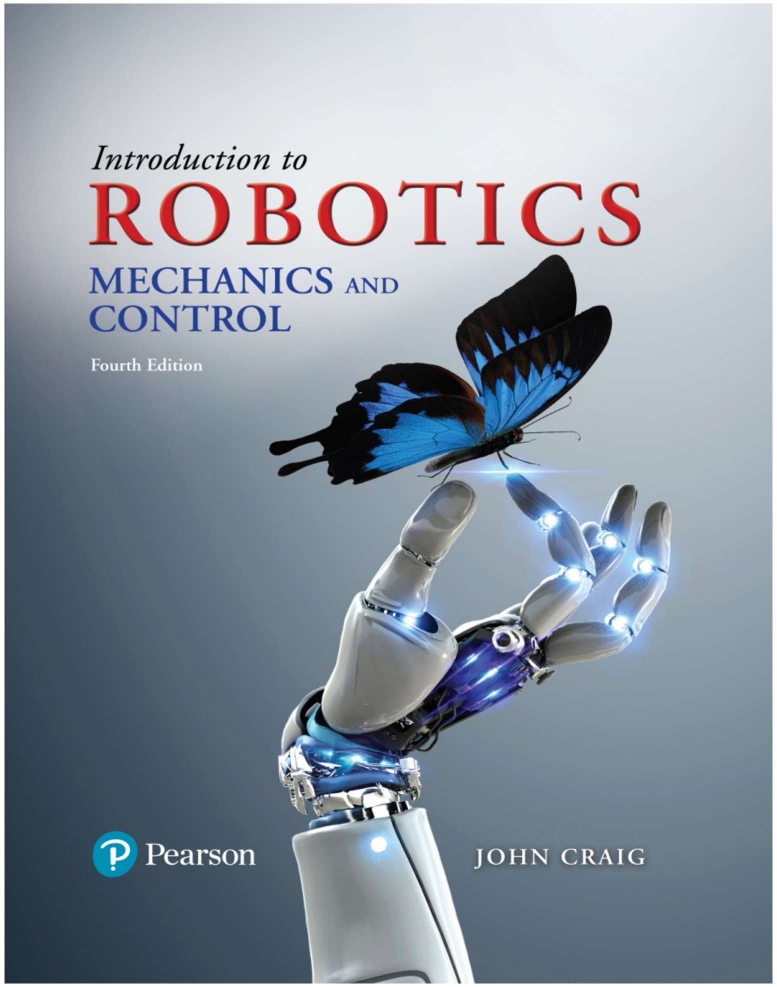
COMP20170: Introduction to Robotics

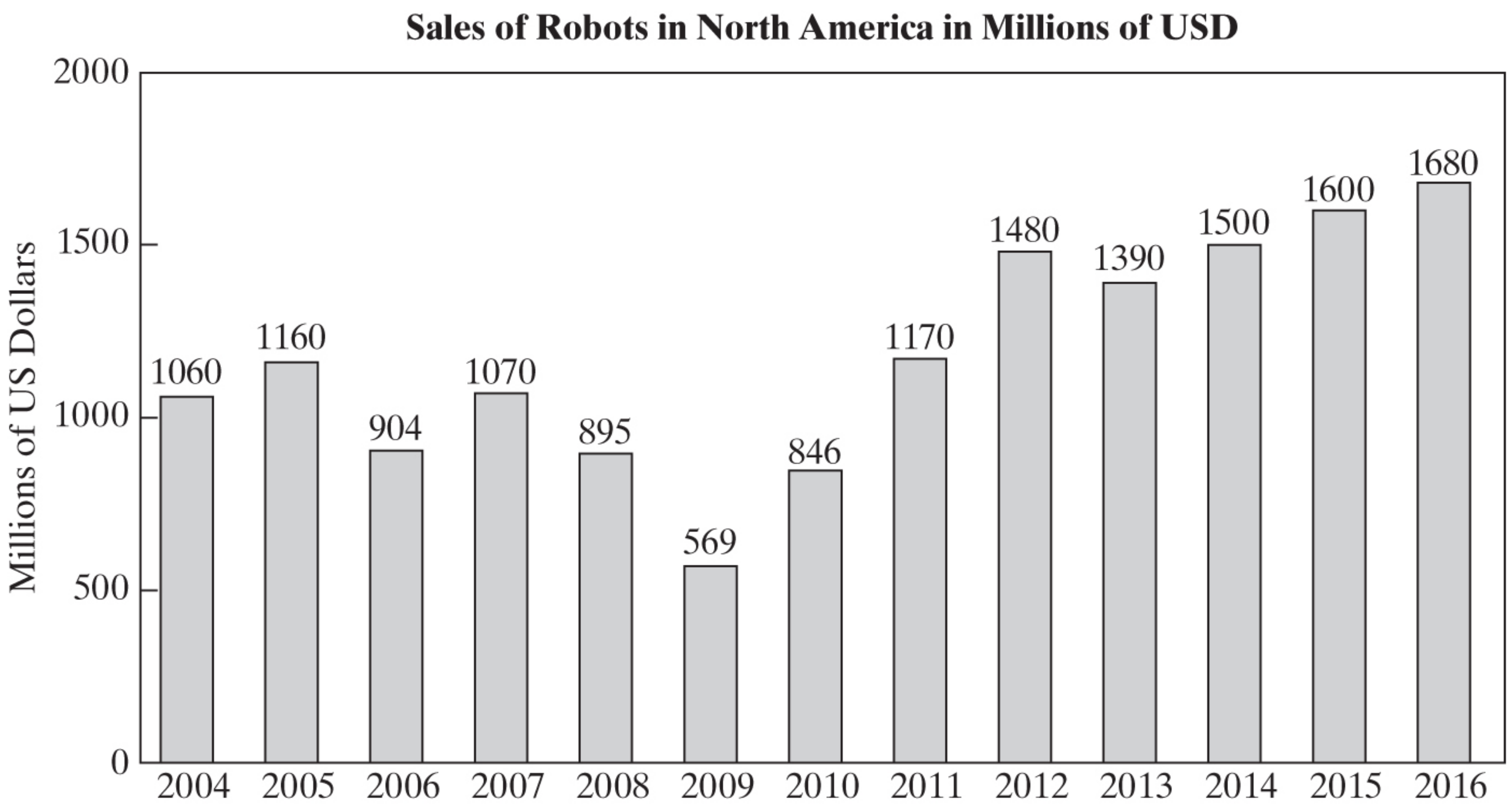
Mechanics and Control

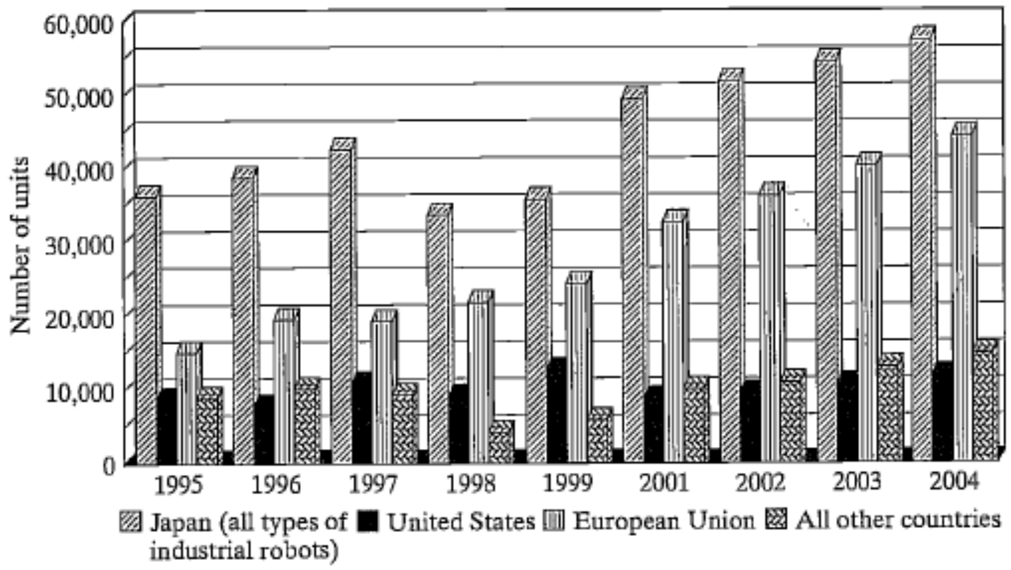
4th Edition



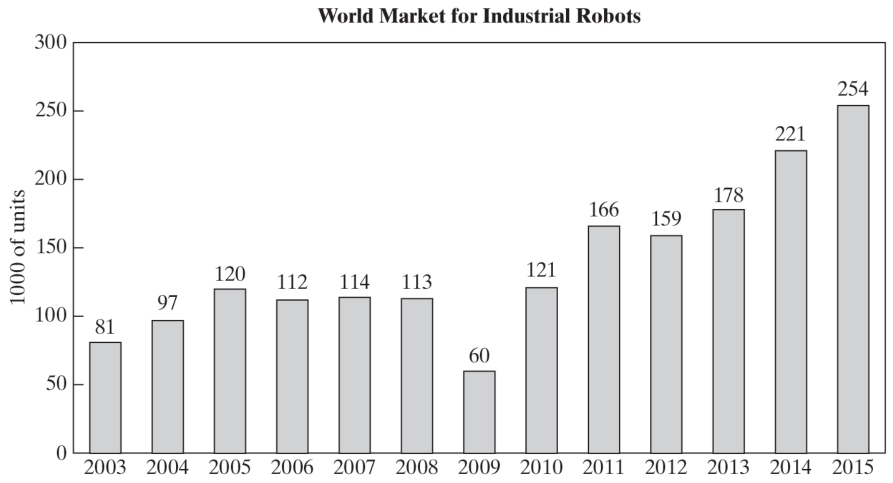
1. Introduction

Sales of industrial robots in North America in millions of U.S. dollars. *Source:* Robotic Industries Association.



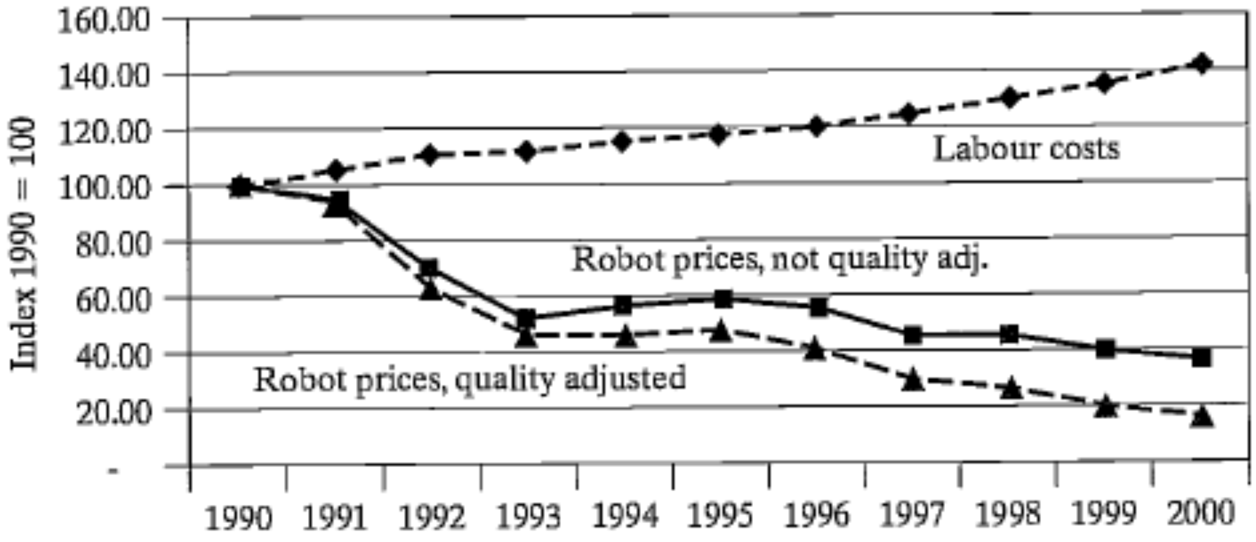


Yearly installations of multipurpose industrial robots for 1995—2000 and forecasts for 2001—2004

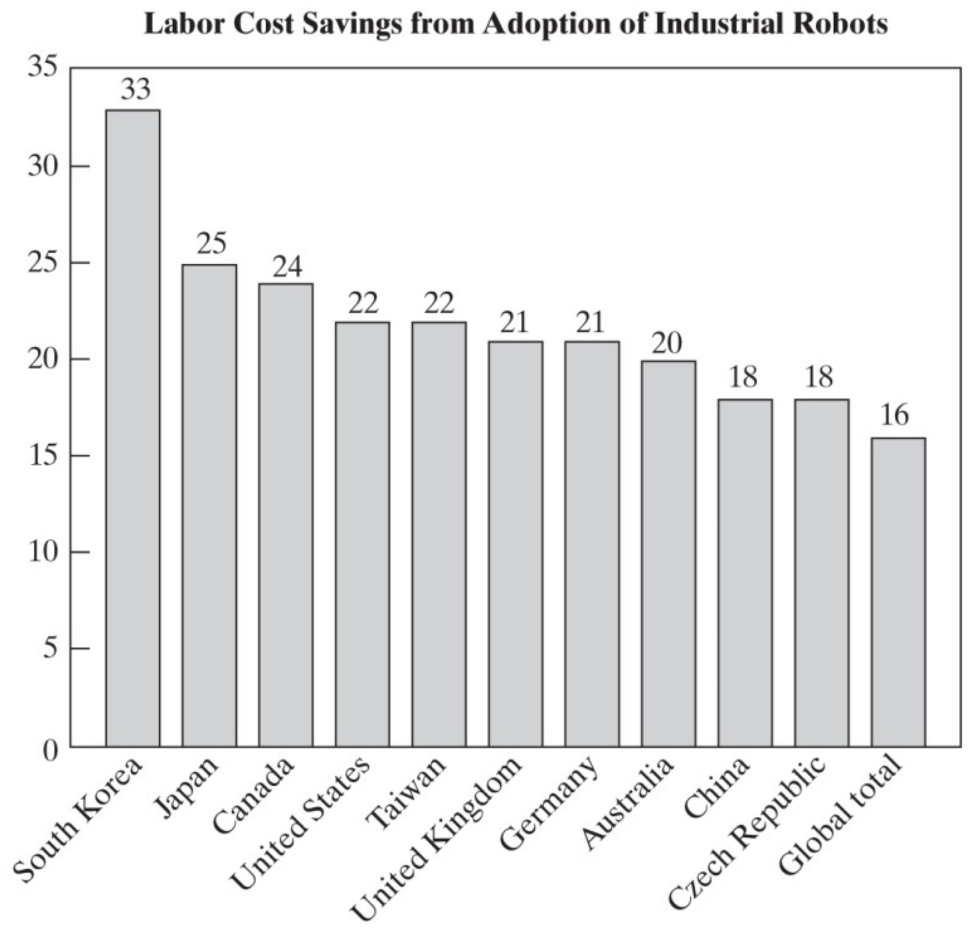


Yearly installations of multipurpose industrial robots. *Source:* World Robotics 2016.

Robot prices compared with human labor costs in the 1990s



Labor cost savings from adoption of industrial robots. Estimated as a percentage in 2025. *Source:* The Boston Consulting Group.



Study of the mechanics and control of manipulators.
I.e. A modern 7 degree-of-freedom robot.

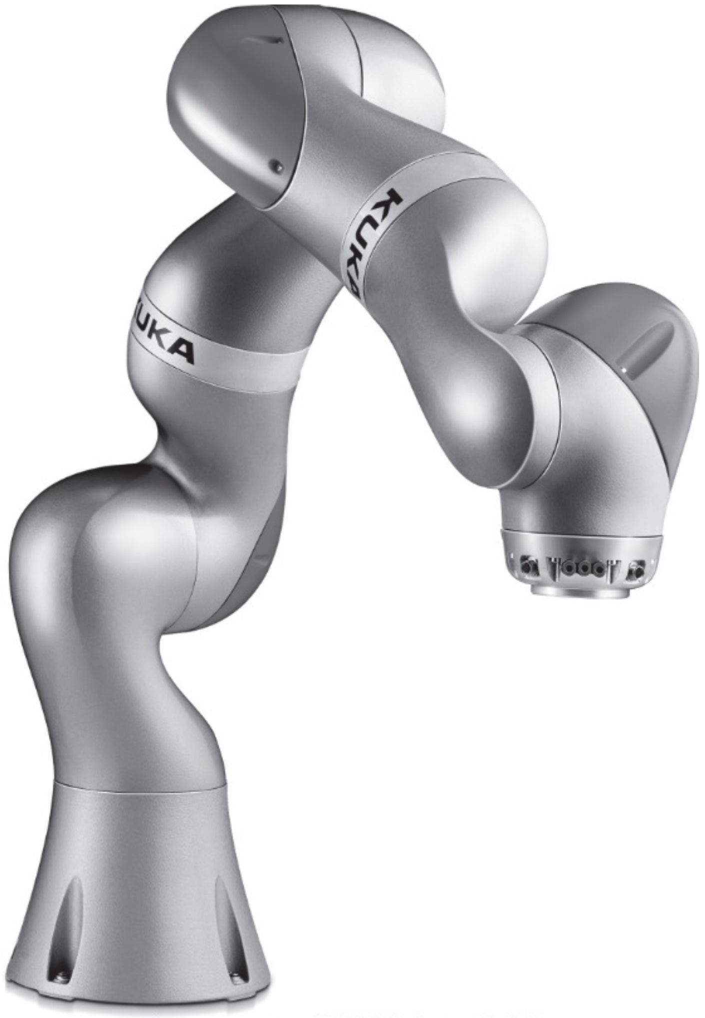


Image courtesy KUKA Roboter GmbH.

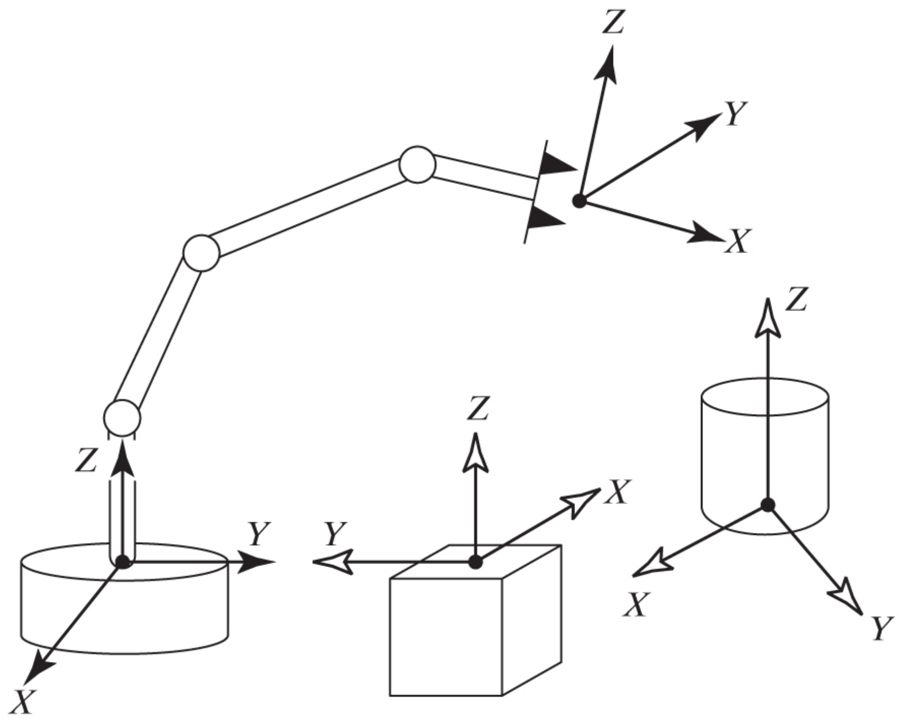
- Assembly by industrial robots, accounted for 10% of installations.
- We will focus on the mechanics and control of the most important form of the industrial robot, the mechanical manipulator.
- Machines which are for the most part limited to one class of task are considered fixed automation.
- By and large, the study of the mechanics and control of manipulators is not a new science, but merely a collection of topics taken from "classical" fields:
    - Mechanical engineering contributes methodologies for the study of machines in static and dynamic situations.
    - Mathematics supplies tools for describing spatial motions and other attributes of manipulators.
    - Control theory provides tools for **designing and evaluating algorithms to realize desired motions or force applications**.
    - Electrical-engineering techniques are brought to bear in the design of sensors and interfaces for industrial robots, and **computer science contributes a basis for programming these devices to perform a desired task**.

The mechanics and control of mechanical manipulators (Introduction).

- **Description of position and orientation** $\longleftarrow$
- Forward Kinematics of manipulators
- Inverse Kinematics of manipulators
- Velocities, static forces, singularities
- Dynamics
- Trajectory generation
- Manipulator design and sensors
- Linear position control
- Nonlinear position control
- Force control
- Programming robots
- Off-line programming and simulation

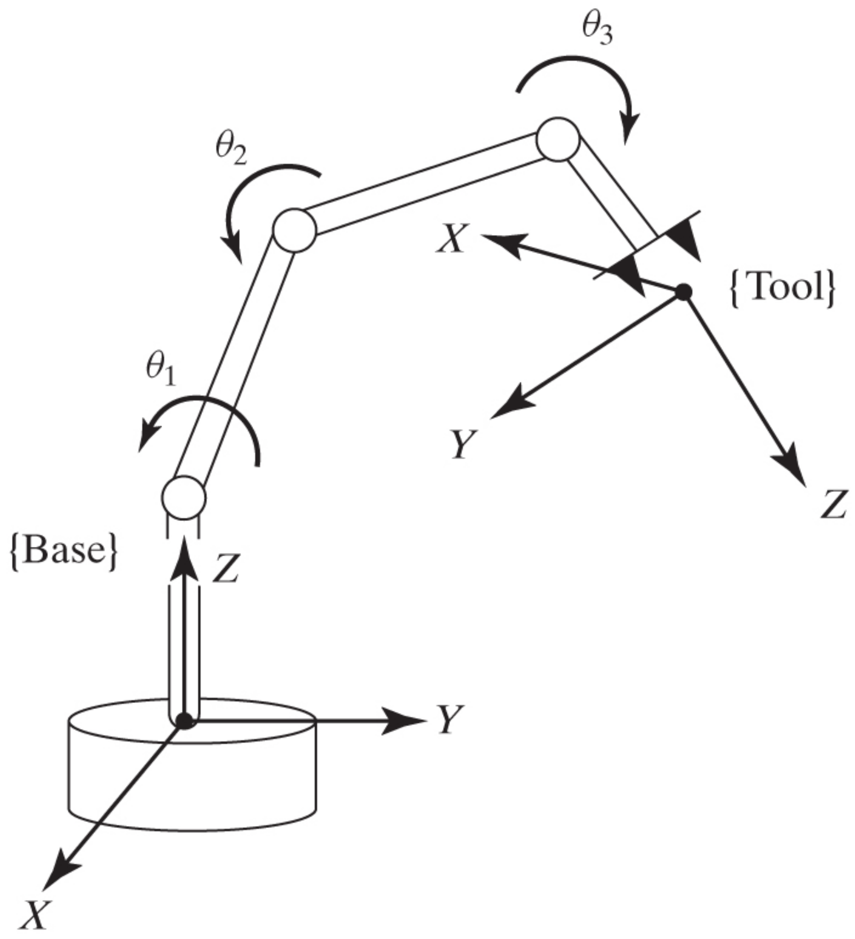
Description of position and orientation:

Coordinate systems or “frames” are attached to the manipulator and to objects in the environment.



Location of objects in three-dimensional space. These objects are the links of the manipulator, the parts
and tools with which it deals, and other objects in the manipulator's environment.
These objects are described by just two attributes:
position and orientation.
In order to describe the position and orientation of a body in space, **we will always attach a coordinate system, or frame, rigidly to the object**. We then proceed to describe the position and orientation of this frame with respect to some reference
coordinate system.
Any frame can serve as a reference system within which to express the position and orientation of a body, so we often think of transforming or changing the description of these attributes of a body from one frame to another.

Forward kinematics for manipulators



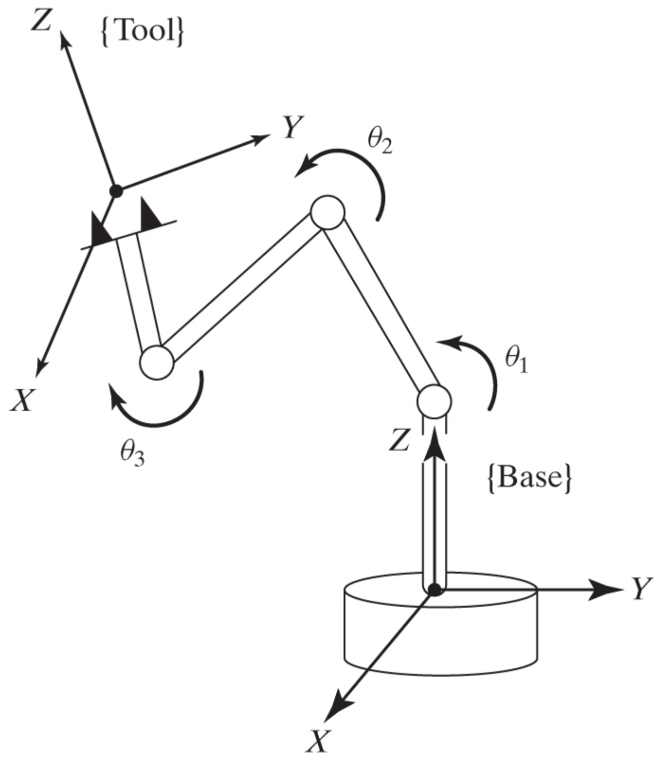
Kinematic equations describe the tool frame relative to the base frame as a function of the joint variables.

The number of degrees of freedom that a manipulator possesses is the number of independent position variables that would have to be specified in order to locate all parts of the mechanism. This is a general term used for any mechanism.

For example, a four-bar linkage has only one degree of freedom (even though there are three moving members). In the case of typical industrial robots, because a manipulator is usually an open kinematic chain, and because each joint position is usually defined with a single variable, the number of joints equals the number of degrees of freedom.

A very basic problem in the study of mechanical manipulation is called forward kinematics. This is the static geometrical problem of computing the position and orientation of the end-effector of the manipulator. Specifically, given a set of joint angles, the forward kinematic problem is to compute the position and orientation of the tool frame relative to the base frame.

Inverse kinematics for manipulators



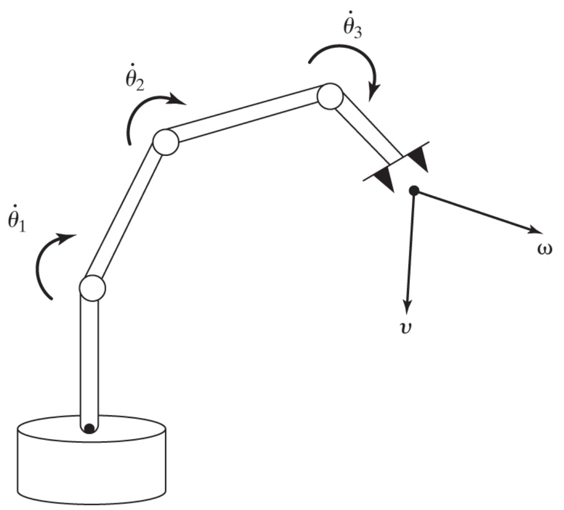
For a given position and orientation of the tool frame, values for the joint variables can be calculated via the inverse kinematics.

Given the position and orientation of the end-effector of the manipulator, calculate all possible sets of joint angles that could be used to attain this given position and orientation. This is a fundamental problem in the practical use of manipulators.
This is a rather complicated geometrical problem that is routinely solved thousands of times daily in human and other biological systems. In the case of an artificial system like a robot, we will need to create an algorithm in the control computer that can make this calculation. In some ways, solution of this problem is the most important element in a manipulator system.
We can think of this problem as a mapping of "locations" in 3-D Cartesian space to "locations" in the robot's internal joint space. This need naturally arises anytime a goal is specified in external 3-D space coordinates. Some early robots lacked this algorithm—they were simply moved (sometimes by hand) to desired locations, which were then recorded as a set of joint values (i.e., as a location in joint space) for later playback. Obviously, if the robot is used purely in the mode of recording and playback of joint locations and motions, no algorithm relating joint space to Cartesian space is needed. These days, however, it is rare to find an industrial robot that lacks this basic inverse kinematic algorithm.

The inverse kinematics problem is not as simple as the forward kinematics one. Because the kinematic equations are nonlinear, their solution is not always easy (or even possible) in a closed form. Also, questions about the existence of a solution and about multiple solutions arise. Study of these issues gives one an appreciation for what the human mind and nervous system are accomplishing when we, seemingly without conscious thought, move and manipulate objects with our arms and hands.

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.

Velocities, static forces, singularities



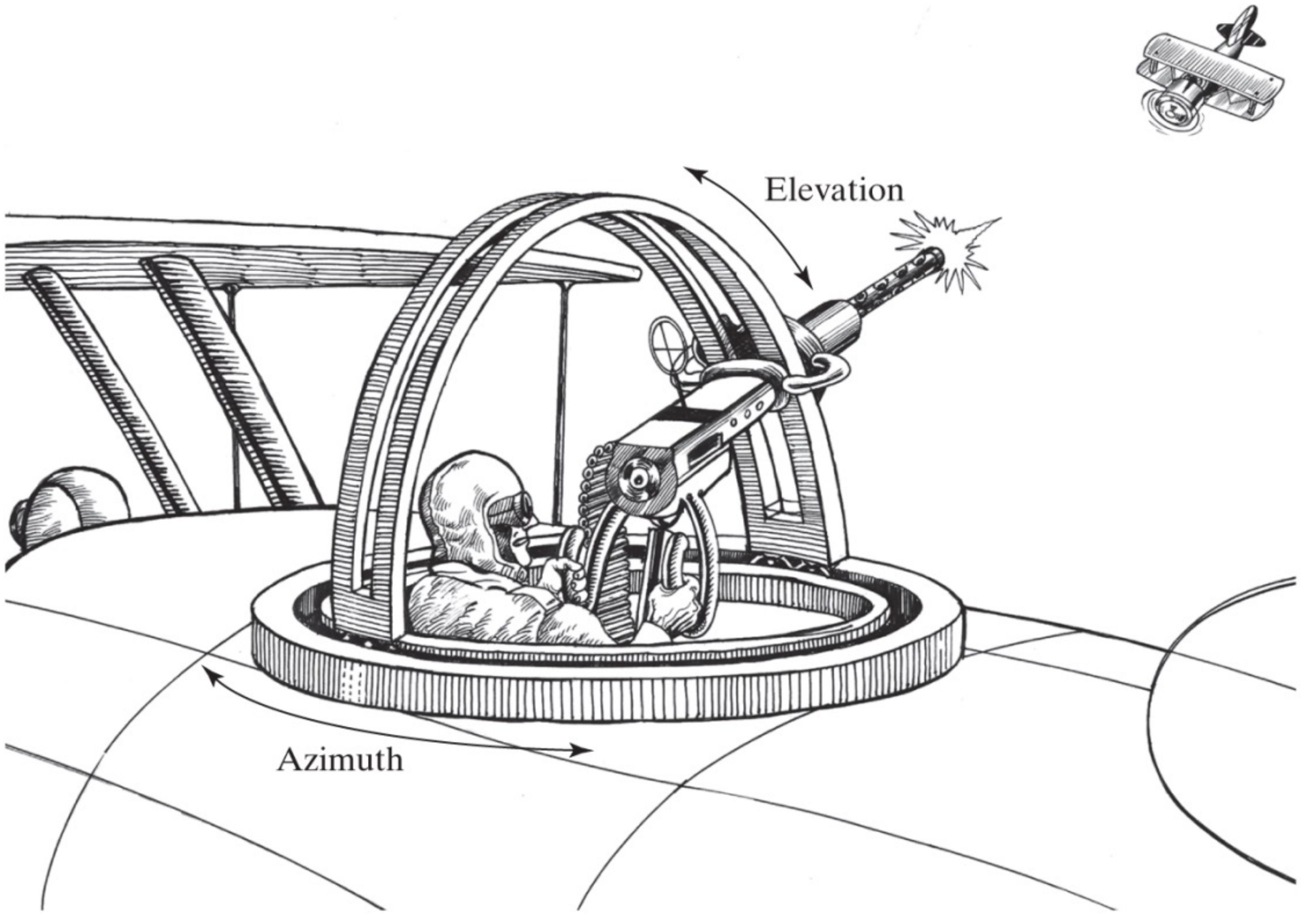
The geometrical relationship between joint rates and velocity of the end-effector can be described in a matrix called the Jacobian.

In addition to dealing with static positioning problems, we may wish to analyze manipulators in motion. Often, in performing velocity analysis of a mechanism, it is convenient to define a matrix quantity called the Jacobian of the manipulator.

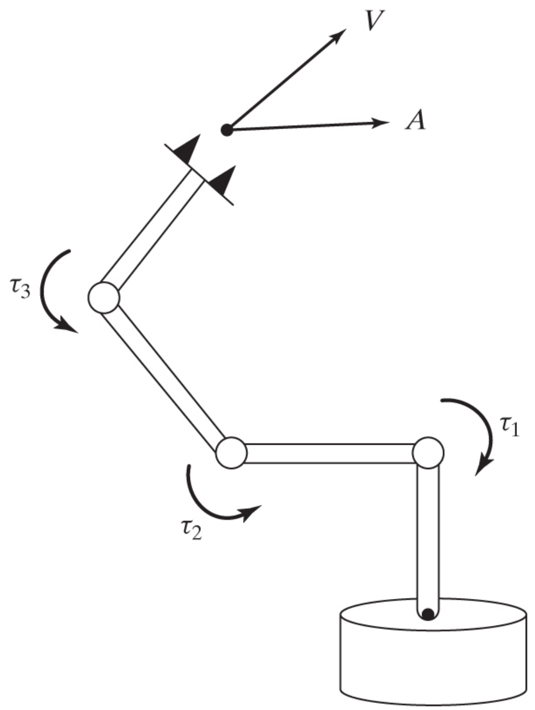
The Jacobian specifies a mapping from velocities in joint space to velocities in Cartesian space.

The nature of this mapping changes as the configuration of the manipulator varies. At certain points, called singularities, this mapping is not invertible. An understanding of the phenomenon is important to designers and users of manipulators.

Consider the World War I biplane with a pilot and a rear gunner. The rear-gunner mechanism is subject to the problem of singular positions.



Dynamics



The relationship between the torques applied by the actuators and the resulting motion of the manipulator is embodied in the dynamic equations of motion.

Dynamics is a huge field of study devoted to studying 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.

The exact form of the required functions of actuator torque depend on the spatial and temporal attributes of the path taken by the end-effector and on the mass properties of the links and payload, friction in the joints, and so on.

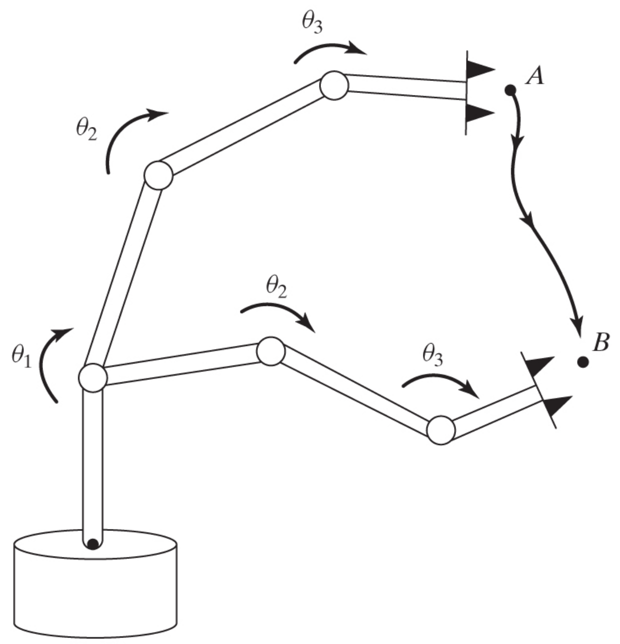
One method of controlling a manipulator to follow a desired path involves calculating these actuator torque functions by using the dynamic equations of motion of the manipulator.

Many of us have experienced lifting an object that is actually much lighter than we (e.g., getting a container of milk from the refrigerator which we thought was full, but was nearly empty). Such a misjudgment of payload can cause an unusual lifting motion. This kind of observation indicates that the human control system is more sophisticated than a purely kinematic scheme. Rather, our manipulation control system makes use of knowledge of mass and other dynamic effects. Likewise, algorithms that we construct to the motions of a robot manipulator should take dynamics into account.

A second use of the dynamic equations of motion is in simulation. By reformulating the dynamic equations so that acceleration is computed as a function of actuator torque, it is possible to simulate how a manipulator would move under application of a set of actuator torques.

As computing power becomes more and more cost effective, the use of simulations is growing in use and importance in many fields.

Trajectory generation



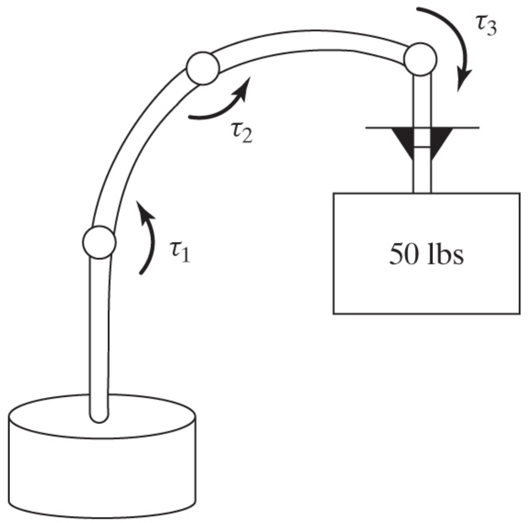
In order to move the end-effector through space from point *A* to point *B*, we must compute a trajectory for each joint to follow.

A common way of causing a manipulator to move from here to there in a smooth, controlled fashion is to cause 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 the appears coordinated. Exactly how to compute these motion functions is the problem of trajectory generation.

Often, a path is described not only by a desired destination but also by some intermediate locations, or via points, through which the manipulator must pass en route to the destination. In such instances the term spline is sometimes used to refer to a smooth function that passes through a set of via points.

In order to force the end-effector to follow a straight line (or other geometric shape) through space, the desired motion must be converted to an equivalent set of joint motions. This Cartesian trajectory generation will also be considered.

Manipulator design and sensors

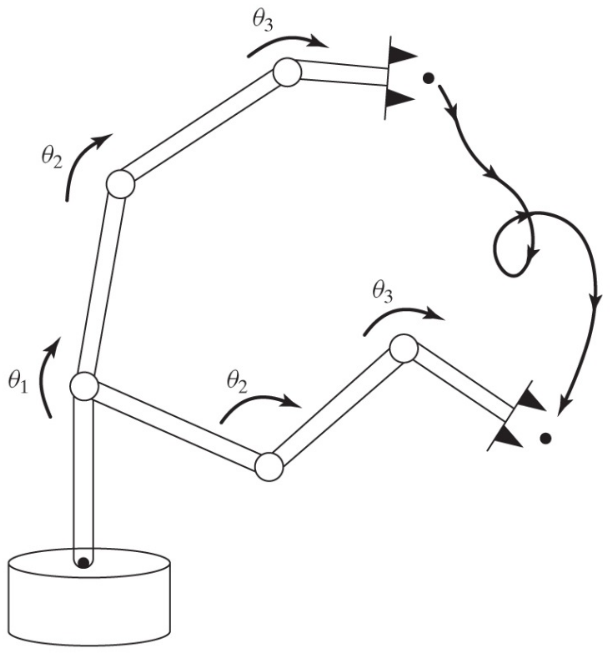


The design of a mechanical manipulator must address issues of actuator choice, location, transmission system, structural stiffness, sensor location, and more.

For example, a specialized robot designed solely to place electronic components on a flat circuit board does not need to have more than four joints. Three joints allow the position of the hand to attain any position in three-dimensional space, with a fourth joint added to allow the hand to rotate the grasped component about a vertical axis. In the case of a universal robot, it is interesting that fundamental properties of the physical world we live in dictate the "correct" minimum number of joints—that minimum number is six.

Integral to the design of the manipulator are issues involving the choice and location of actuators, transmission systems, and internal-position (and sometimes force) sensors.

Linear position control



In order to cause the manipulator to follow the desired trajectory, a position-control system must be implemented. Such a system uses feedback from joint sensors to keep the manipulator on course.

Some manipulators are equipped with stepper motors or other actuators that can execute a desired trajectory directly. However, the vast majority of manipulators are driven by actuators that supply a force or a torque to cause motion of the links.

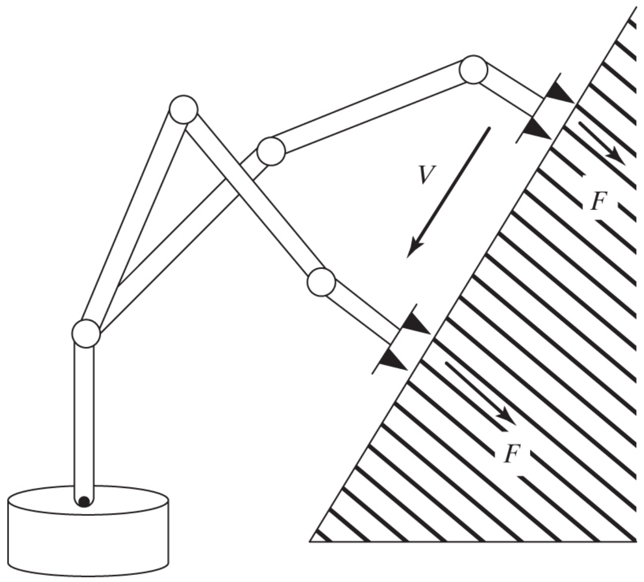
In this case, an algorithm is needed to compute torques that will cause the desired motion. The problem of dynamics is central to the design of such algorithms, but does not in itself constitute a solution. A primary concern of a position control system is to compensate automatically for errors in knowledge of the parameters of a system and to suppress disturbances that tend to perturb the system from the desired trajectory. To accomplish this, position and velocity sensors are monitored by the control algorithm, which computes torque commands for the actuators.

Nonlinear position control

Although control systems based on approximate linear models are popular in current industrial robots, it is important to consider the complete nonlinear dynamics of the manipulator when synthesizing control algorithms. Some industrial robots are now being introduced which make use of nonlinear control algorithms in their controllers. These nonlinear techniques of controlling a manipulator promise better performance than do simpler linear schemes.

This course will not cover non-linear position control

Force control



In order for a manipulator to slide across a surface while applying a constant force, a hybrid position–force control system must be used.

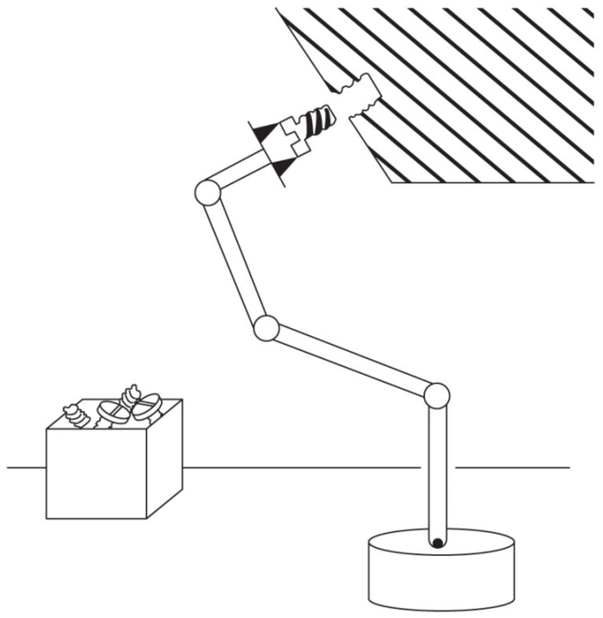
The ability of a manipulator to control forces of contact when it touches parts, tools, or work surfaces seems to be of great importance in applying manipulators to many real-world tasks.

Force control is complementary to position control, in that we usually think of only one or the other as applicable in a certain situation. When a manipulator is moving in free space, only position control makes sense, because there is no surface to react against.

When a manipulator is touching a rigid surface, however, position-control schemes can cause excessive forces to build up at the contact or cause contact to be lost with the surface when it was desired for some application.

Manipulators are rarely constrained by reaction surfaces in all directions simultaneously, so a mixed or hybrid control is required, with some directions controlled by a position-control law and remaining directions controlled by a force-control law.

Programming robots



Desired motions of the manipulator and end-effector, desired contact forces, and complex manipulation strategies can be described in a *robot programming language.*

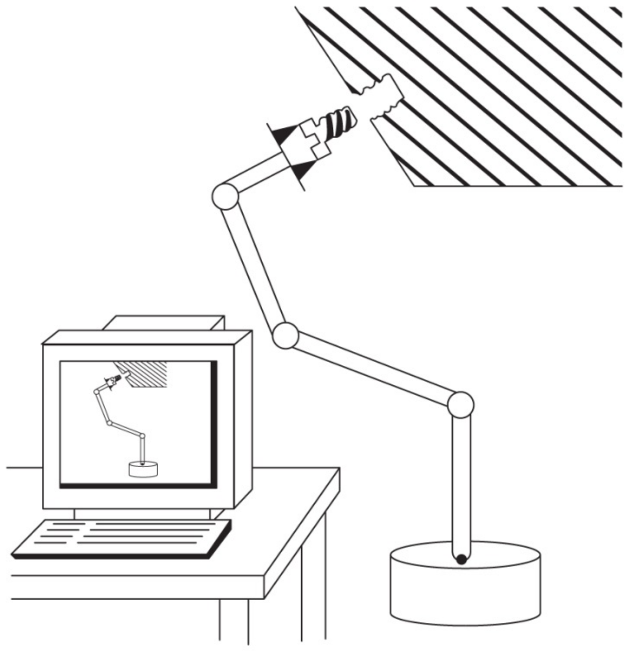
A robot programming language serves as the interface between the human user and the industrial robot. Central questions arise:

How are motions through space described easily by the programmer?

How are multiple manipulators programmed so that they can work in parallel?

How are sensor-based actions described in a language?

Off line programming and simulation



Off-line programming systems, generally providing a computer graphics interface, allow robots to be programmed without access to the robot itself during programming.

An off-line programming system is a robot programming environment that has been sufficiently extended, generally by means of computer graphics, that the development of robot programs can take place without access to the robot itself. A common argument raised in their favor is that an off-line programming system will not cause production equipment (i.e., the robot) to be tied up when it needs to be reprogrammed; hence, automated factories can stay in production mode a greater percentage of the time.