Chapter 1

Robot Mechanisms

1.1 Kinematic Forms

In this class, we study two basic robot forms that accomplish two very different kinds of jobs. Mobile robots typically have wheels, which they use to locomote on the ground. Manipulator arms are usually bolted to a fixed surface and have a serial chain linkage which can change shape, as well as a hand or *end-effector* for manipulating objects. Note that a manipulator arm can also be mounted on a mobile robot, giving rise to the problem of mobile manipulation.

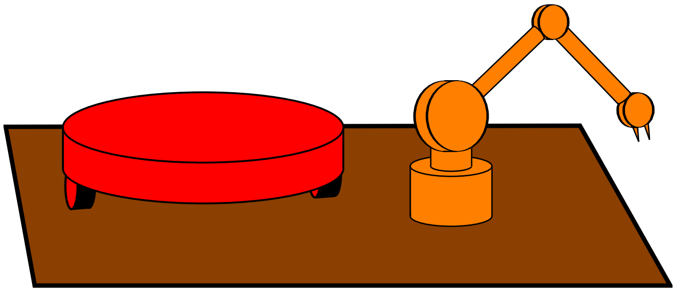

Figure 1.1: Mobile robot (red, with two wheels) and manipulator arm (orange, fixed to the table).

1.1.1 Mobile robots

Mobile robots come with a variety of drive mechanisms. Wheels may be drive wheels or free spinning. They may have a fixed axle, a steered axle, or a free axle. A free-spinning, free-axle wheel is a *caster*. Since some drive mechanisms need only two wheels, a caster is added to maintain stability on a flat surface.

The major drive mechanisms are differential drive, bicycle steering, and Ackerman steering. Differential drive is the simplest, and it's what's in a Roomba vacuum cleaner. It involves two wheels that share a fixed axis and can be driven at different rates to achieve turning. Bicycle steering involves one drive wheel with a fixed axle and one free-spinning wheel with a steered axle. Ackerman steering is what most cars use. It is similar to a pair of bicycles – two parallel fixed wheels in back and two steered wheels in the front. We will return to these drive mechanisms in the Kinematics unit.

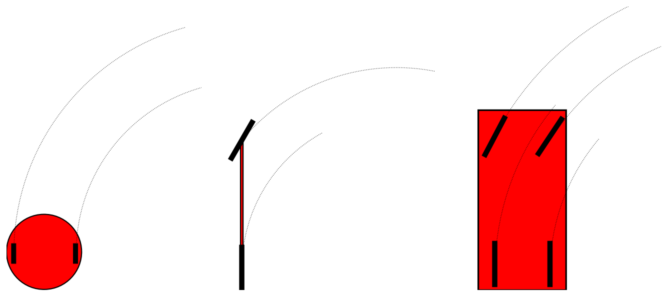

Figure 1.2: Differential drive, bicycle steering, and Ackerman steering. The tracks produced by the wheels are shown.

1.1.2 Manipulator arms

The most common mechanism of manipulator arm comprises a series of rigid links connected by joints. Joints impose constraints on the motion of neighboring links. Although joints can be complex, all possible joints can be constructed as composites of two basic forms. *Revolute joints* rotate one link about an axis with respect to another link, whereas *prismatic joints* slide one link along an axis with respect to another link.

The purpose of joints is to constrain or prevent motion. Each of these joint models constrains motion except in one direction (whether angular or linear). This direction of allowed motion is often referred to as a *degree of freedom*. Thus, when someone talks about the number of degrees of freedom that a mechanism has, they are referring to the number of joints in it, loosely speaking.

A manipulator arm typically comprises an alternating sequence of rigid links and actuated, sensed joints.

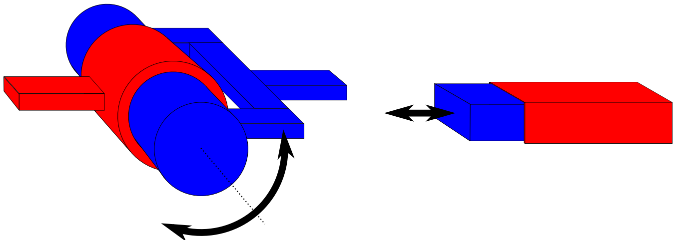

Figure 1.3: Revolute and prismatic joints. Both possess one degree of freedom.

1.2 Actuators and Sensors

One definition of a robot is a machine that can sense, think, and act. The thinking is handled by computation, but the other two capabilities require hardware. *Actuators* cause the robot to move deliberately in the world in a predictable way, and they are most often electric motors. Motors exert a torque (force) in proportion to the voltage applied to them. Many other types of actuators exist, including hydraulic, pneumatic, and chemical actuation.

Sensors detect targeted physical properties in the world, and can be configured to target the robot itself (*proprioceptive sensors*) or the outside word (*exterioceptive sensors*). Examples of proprioception often used in robots are:

1. *encoders* – measure absolute or relative position of a revolute joint by counting ticks.
2. *torque sensor* – detect the amount of force a motor is exerting on a joint.

These sensors are used to build a *controller* to drive a motor to set its joint to the value desired by the robot's programming. Several examples of exterioception are:

1. *imaging sensor* – a camera or similar device to build a pixel grid depicting a scene in front of the sensor. Imaging can be done in the visual spectrum, infrared, etc. RGB+D cameras like the Kinect use structured light, projected in the infrared spectrum, to reconstruct a depth component.
2. *range finder* – a depth sensor, often employing a sweeping beam to detect depth in a swept volume. Examples include sonar and laser range-finders (also called LIDAR).

1.3 Error

Both sensors and actuators are subject to error. It is therefore a fact of life that nothing a robot does will ever be perfect or certain. This fact gives rise to two important problems in robotics.

Estimation is the act of fusing together all the available evidence from the complete history of sensed data and actions taken to determine a best guess about what is true in the world. Estimation can be used to build maps, localize objects of interest, or localize the robot in the world.

Planning is the act of reasoning about uncertain world state (an estimate) as well as uncertain future actions to maximize the likelihood of successfully accomplishing a task. Planning can be used to move objects around with respect to other objects (e.g. doing the dishes, handling nuclear materials) or to move the robot itself with respect to the world (e.g. self-driving cars).

We require a set of tools to reason about error and uncertainty, which will be the focus of the final section of the course. A system produces readings that include both signal and noise (error). We cannot know how much the signal and noise contribute to an individual reading. However, we frequently employ assumptions that enable us to make useful statements about the noise characteristics. Often, these methods work by collecting a large quantity of samples and computing statistics on them.

Accuracy and Precision. Two important concepts used to measure error are:

1. *accuracy* – a measure of how close a robot can come to its desired position, or how close a sensor reading is to the *ground truth* or the genuine value that it should ideally return.
2. *repeatability* or *precision* – a measure of how close a robot can come to its previous attempts at the same motion or the same measurement.

The terms *accurate* and *precise* are often used qualitatively to measure motors, sensors, or any hardware or software system that generates a real-valued result. However, if we are measuring a sequence of output samples from a system, we can quantitatively estimate whether the system is accurate by the following test:

$$\left|\frac{\mu - t_a}{t_a}\right| \leq e_a,$$

where μ is the mean of the samples, t_a is the known target value, and e is a threshold error, e.g. 5%. In this case, we would say that a system is accurate if its average error from the target is less than 5%. If the system is perfectly accurate, then $\mu = t_a$, yielding an error of zero.

We can also estimate whether the system is precise by the following test:

$$\frac{\sigma}{\mu} \leq e_p,$$

where σ is the standard deviation of the samples, μ is the mean of the samples, and e_p is a threshold error, e.g. 5%. That is, a precise system has a standard deviation that is less than 5% of its mean.

A system can be both accurate and precise, accurate but not precise, precise but not accurate, or neither accurate nor precise. For example, suppose that a mobile robot is using a range finder to measure the distance to an obstacle known to be 10m away. Over several measurements, a range finder that is both accurate and precise will return distances that are both close to the ground truth on average (accurate) and clustered together (precise), such as the series of readings 9.91m, 10.07m, and 10.02m. A range finder that is accurate but not precise might return the series of readings 9.49m, 10.51m, and 9.95m — the mean of these measurements is close to the ground truth, but the variance is large. On the other hand, a range finder that is precise but not accurate might return the readings 10.85m, 10.79m, and 10.98m — these measurements have little variance, but their average is quite far from the ground truth. Finally, if the range finder returns measurements that are all over the place and do not average out to be close to the ground truth, such as 9.6m, 11.4m, and 13.01m, then we consider the range finder to be neither accurate nor precise.

Depending on the cause of an error, it might tend to decrease accuracy, precision, or both. When we talk about a decrease in accuracy or precision, we mean that $|(\mu - t_a)/t_a|$ and σ/μ get larger.

Uncertainty. Another tool used to analyze error is uncertainty, which is a measure or estimate of an unknown error. There are three common assumptions that we use to model potentially uncertain sensors or actuators:

1. *determinism* – the motors and sensors are perfectly accurate and repeatable.
2. *nondeterminism* – the motors and sensors are not repeatable. Repeated attempts to read a sensor or actuate a motor will yield different results unpredictably.
3. *stochasticity* – the motors and sensors are not repeatable, but their readings or actions are drawn from a probability distribution that is related to the commanded action or the ground truth of the sensor target. Thanks to the *law of large numbers*, many stochastic sensor readings could be used to estimate the ground truth.

Uncertainty in sensing and actuation can be thought of as nature's actions in response to the robot's commands. Sometimes nature cooperates to a larger or smaller extent. In the worst case (nondeterminism), no amount of sensor readings is sufficient to reconstruct the ground truth of the target.