The third part of this work is devoted to simulated robot devices, learning models of their kinematic structure, and using these models for simple directional control tasks such as reaching for objects. Learning is realized by algorithms that mimic brain function at least to some degree. Therefore the framework developed herein could explain how the brain learns motor control. Of course, there is no proof because a concrete implementation in one or the other programming language is far from being comparable to brain imaging results that merely highlight activity in certain regions for certain tasks. Nonetheless, this work tries to make a connection from neuron level (neurobiology) to the functional, cognitive level (psychology).

Chapter 7, the first chapter of this part, introduces robot devices and the background knowledge required for simple kinematic control. Kinematics describe the motion of bodies, that is, position, velocity, and acceleration. Dynamics, on the other hand, are concerned with forces acting on the bodies, e.g. friction, momentum of inertia, and external forces such as gravity. Dynamics are not considered in this work for several reasons: First, current robots often come with sophisticated low-level dynamic controllers and swiftly realize the kinematic commands of a user. Second, dynamics is where animals and robots are very different because muscles are not comparable to the servo motors of typical robots. Third, dynamics are better expressed as a set of differential equations than as a function.

Robots are usually an assembly of limbs, joints, and some sort of actuators – servo motors typically, but pneumatic elements are possible as well. Physical bodies are connected by either rotational (or translational) joints. The physical construction of limbs and joints is a so called kinematic chain as for example illustrated in Figure 7.1. The number of joints defines the Degrees of Freedom (DoF) of a kinematic chain. For example, the kinematic chain in Figure 7.1 has two DoF and the human arm has seven DoF.

A very basic property of a kinematic chain is that its state is uniquely defined by the state of the joints, where state means position and its derivatives: the first derivative of position is velocity, the second one is acceleration. Simply put, location and orientation of the end effector or any other part of the kinematic
Figure 7.1: A kinematic chain composed of two limbs (and an imaginary base) with two rotational joints. The \((x, y)\) position of the gripper is defined by joint angles \(\theta_1\) and \(\theta_2\).

chain can be computed from the joint angles. The so called forward kinematics are covered in Section 7.1.

A certain task does not always require all joints, or more specifically, can sometimes be solved with multiple joint configurations. While this may seem like a problem, it is actually more of an opportunity. The so called kinematic redundancy allows to reach the same task location with different joint configurations – which helps to fulfill secondary constraints such as obstacle avoidance, reducing energy consumption, or maintenance of comfortable joint states. On the contrary, it is sometimes impossible to produce a particular task space trajectory due to so called singularities at certain joint configurations. Both kinematic redundancy and singularities are the topic of Section 7.2.

In order to realize a given task, a controller need not know about the forward kinematics but instead requires the inverse formulation: What joint state is required for a desired task state? Section 7.3 discusses the inverse kinematics and how to smartly invert the forward mapping while not only avoiding singularities but also incorporating other constraints.

Finally, given a full task and a robot device, the controller must plan a trajectory from current to target location. However, sophisticated planning is not the topic of this work and a very simple directional scheme is employed in Section 7.4 by simply moving in a straight line to the target. To summarize briefly, the present chapter covers the basics of kinematic robot control, but learning a kinematic model is discussed in the follow-up chapter.

### 7.1 Task Space and Forward Kinematics

Usually the task of a robot cannot be described in terms of joint angles. Instead, a task will often be described by Cartesian coordinates of the end effector, eventually together with an orientation. However, any other task space formu-