Supplemental Material For Spong - Robot Dynamics and Control

Eirik Kvalheim

March 15, 2019

Problem 1: Forward Kinematics

The first problem encountered is to describe both the position of the tool, and the locations A and B (and most likely the entire surface S) with respect to a common coordinate system. The reason why we want to do this to express mathematically where the robot is, with regards to the surface. In Chapter 2 we will give some background on representations of coordinate systems and how these coordinate systems relate to each other.

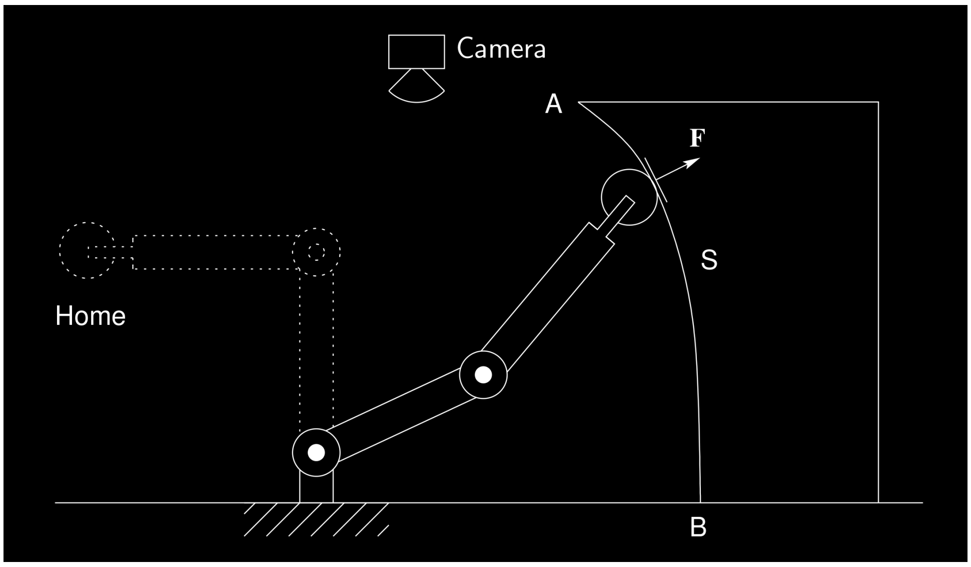

Figure 1: Two-link planar robot example

Typically, the manipulator will be able to sense its own position in some manner using internal sensors, often position encoders, but sometimes resolvers, especially for revolute joints. In the case of Figure 2, the sensors for measuring degrees of rotation are located in the center of joints 1 and 2, so that they can measure directly the joint angles θ_1 and θ_2. Therefore, we also need to express the positions A and B in terms of these joint angles. This leads to the **Forward Kinematics problem**, which is the cornerstone of understanding robotics. Rephrased, we can say that **Forward Kinematics** is a mathematical problem where we want to determine the position and orientation of the end-effector or tool in terms of the robots joint variables and link lenghts. We will study this topic thoroughly later on in Chapter 3.

It is customary to establish a fixed coordinate system, called the **world** or **base** frame to which all objects including the manipulator are referenced. In real world applications of robots, these frames rarely coincide, but for simplification purposes they can sometimes be the same. The **base** frame should always be located at the robots base, this convention is however not always followed. In our case we follow the convention and establish the base coordinate frame denoted $o_0x_0y_0$ at the base of the robot. Furthermore, we will establish the coordinate system $o_1x_1y_1$ at the center of joint 2, and the coordinate system $o_2x_2y_2$ at the tip of the robot where it's end-effector is located. This is shown below in Figure 2.

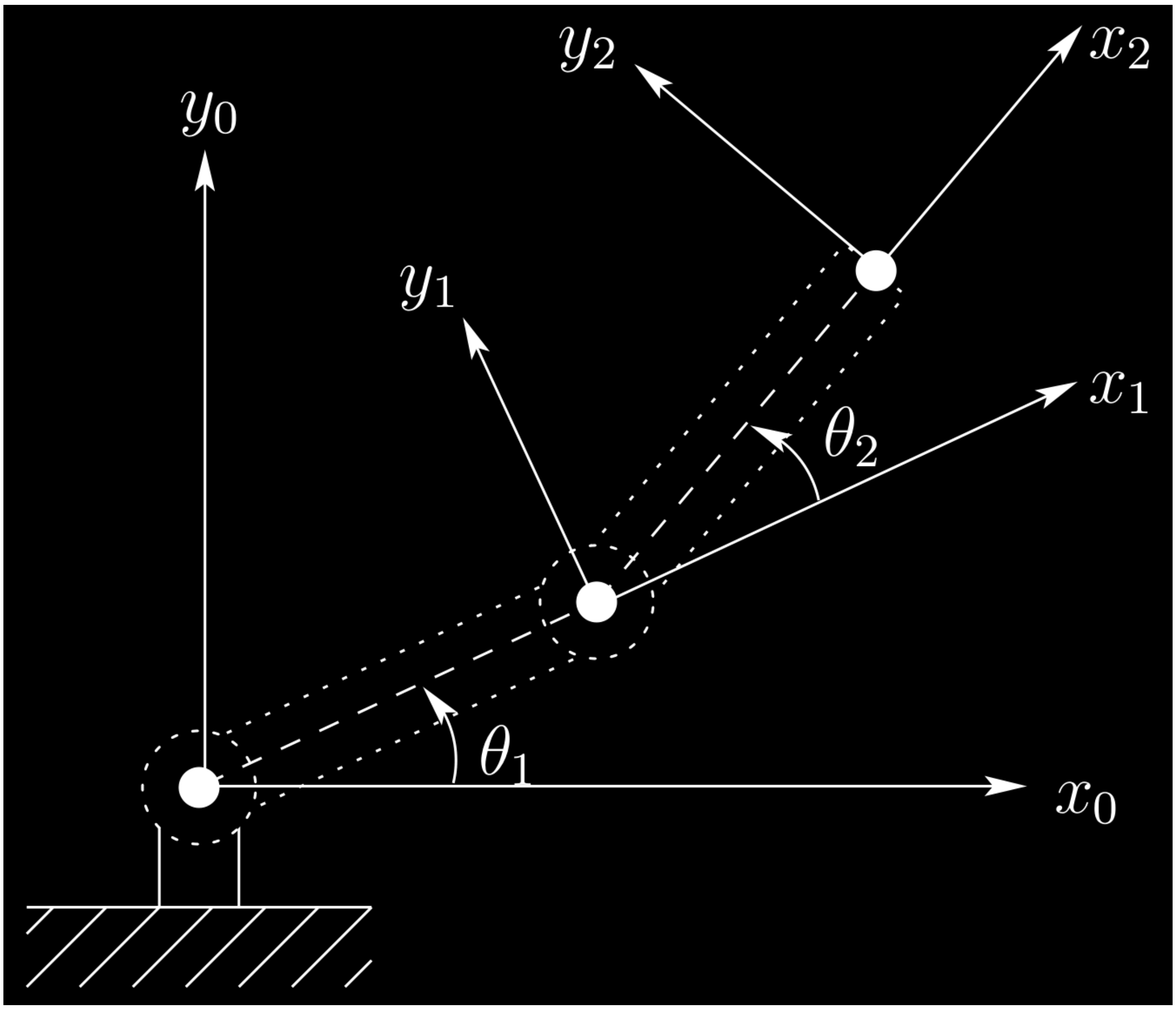

Figure 2: Coordinate frames for two-link planar robot

The somewhat mystical notation $o_0x_0y_0$ is usually written short as o_0. By adding x_0y_0 we intend to specify that the coordinate system o_0 is spanned by the vectors x_0 and y_0.
Why do we need that, why two vectors, should we not specify three, or maybe four, just to be sure? Well, the robot in Figure 2 lives in a two-dimensional space, thus it can not move in a third direction, meaning we only need two vectors to mathematically describe the robots action in the world. If you are not familiar with vectors, just imagine these vectors as arrows of infinite length making up the whole coordinate system.

Let now the tool of the robot be located at the origin of o_2, and let (x, y) be the coordinates of the tool, expressed in the **base** coordinate frame. This means that x represent a distance between the origin of o_0 and some random place on the x_0 vector, and y represent the position along the y_0 vector. We see now that since we have specified the x_0 and y_0 vectors, we know in which direction x and y are going. Thus the specification of x_0 and y_0 becomes significant. Moreover, with some basic trigonometry, we can express the coordinates (x, y) by just drawing some lines:

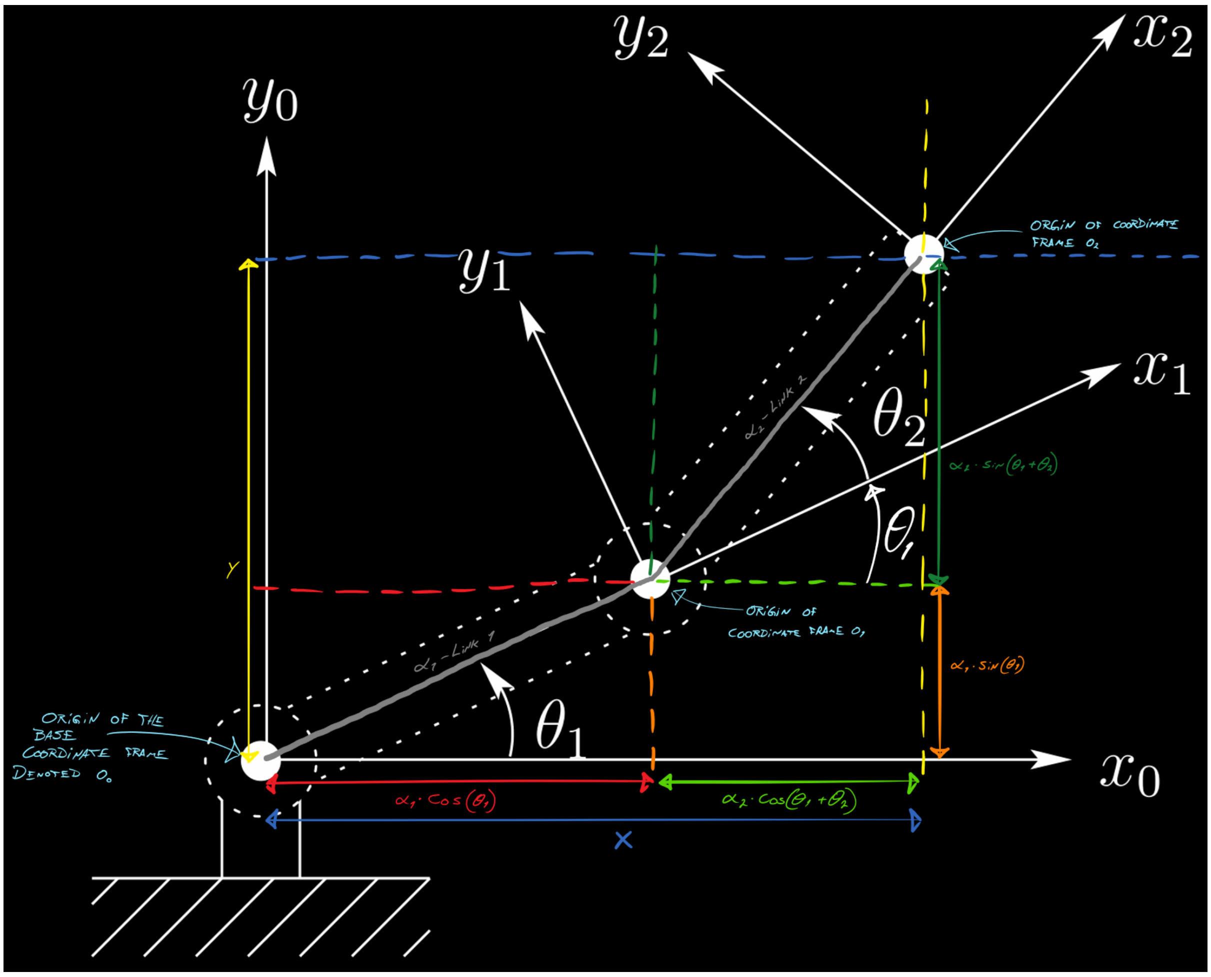

Figure 3: Coordinate frames for two-link planar robot

Hence we see that when α_1 and α_2 are the lengths of the two links, respectively, we get

$$x = \alpha_1 \cos\theta_1 + \alpha_2 \cos(\theta_1 + \theta_2) \tag{0.1}$$

$$y = \alpha_1 \sin\theta_1 + \alpha_2 \sin(\theta_1 + \theta_2) \tag{0.2}$$

If you don't know what $\cos(x)$ and $\sin(x)$ are, don't despair. Just think of them as functions that takes in a number (which could be any number on the real number line) and outputs a value which lies in the continuous interval of real numbers between -1 and 1.
I.e.
$\cos(\frac{\pi}{2}) = 0$, $cos(\pi) = -1$, $cos(1048576\pi) = 1$, $sin(\frac{\pi}{2}) = 1$, $sin(\frac{3\pi}{2}) = -1$, $sin(1048576\pi) = 0$.

We have now found the position of the end-effector. Thus, it remains to express how the coordinate system o_2 which represent the orientation of the tool, has rotated with respect to o_0. We do this by arranging the representation in a rather different way called a matrix. A matrix is a collection of numbers arranged into a fixed number of rows and columns, which is convenient for us, since the tool coordinate frame o_2 has two vectors x_2 and y_2 rotated with respect to two other vectors x_0 and y_0 (which composes o_0). Thus we will need four numbers to represent our rotation.

Let now

- γ_1 represent the rotation of x_2 with respect to the x axis of the base frame, namely x_0
- γ_2 represent the rotation of x_2 with respect to the y axis of the base frame, namely y_0
- γ_3 represent the rotation of y_2 with respect to the x axis of the base frame, namely x_0
- γ_4 represent the rotation of y_2 with respect to the y axis of the base frame, namely y_0

Then

$$\begin{bmatrix} \gamma_1 & \gamma_3 \\ \gamma_2 & \gamma_4 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix} \tag{0.3}$$

And we now have our orientation representation. To summarize, the equations (0.1), (0.2) and (0.3) represent the robots position and orientation in the two-dimensional world it is operating within, and hence these are the **Forward Kinematics equations**. For a six degree-of-freedom robot these equations are quite complex and cannot be written down as easily as for the two-link manipulator. The general procedure that we discuss in Chapter 3 establishes coordinate frames at each joint and allows one to transform systematically among these frames using matrix transformations. The procedure that we use is referred to as the **Denavit-Hartenberg convention**.

Problem 2: Inverse Kinematics

Now, given the joint angles θ_1 and θ_2 we can determine the end-effector coordinates x and y. In order to command the robot to move to location B we need the inverse; that is, we need the joint variables θ_1 and θ_2 in terms of the x and y coordinates of B. This is the problem of **Inverse Kinematics**. In other words, if we know x and y in the forward kinematic equations (0.1-0.3), we wish to use this to make an equation for each joint angle, such that if we input x and y in our equations, the correct joint values will be found.
A solution may not be easy to find nor is there a unique solution of the **Inverse Kinematics** in general. We can see, for example, in the case of a two-link planar mechanism that there may be no solution, if the given (x, y) coordinates are out of reach of the manipulator. If the given (x, y) coordinates are within the manipulator's reach there may be two solutions as shown in Figure 4, the so-called **elbow up** and **elbow down** configurations, or there may be exactly one solution if the manipulator must be fully extended to reach the point. There may even be an infinite number of solutions in some cases.

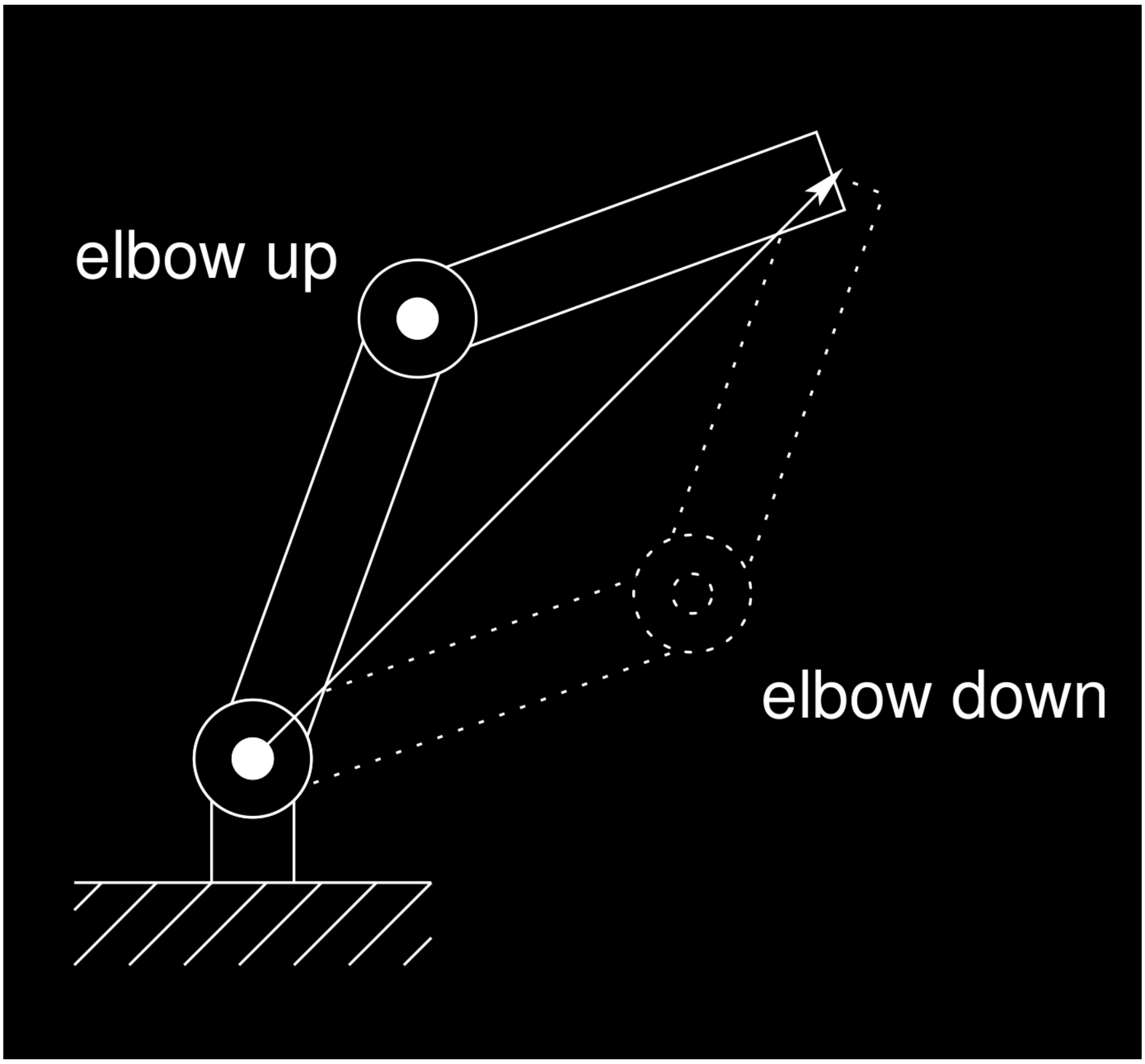

Figure 4: Multiple inverse kinematic solutions.

Consider now the diagram of Figure 5.

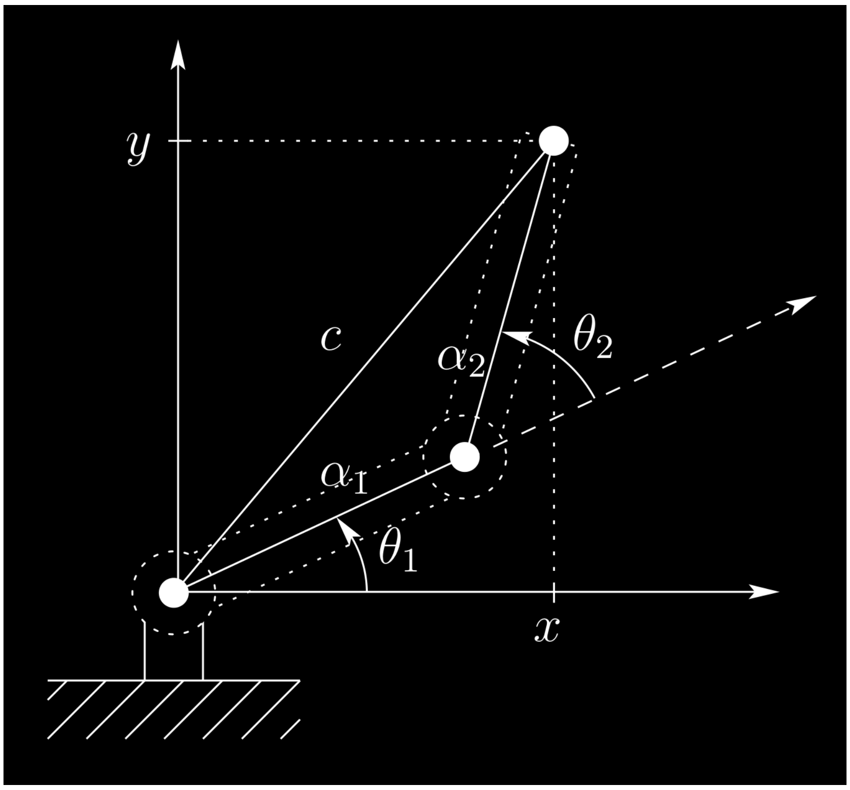

Figure 5: Solving for the joint angles of a two-link planar arm.

By using the **Law of Cosines** we can derive (simply by putting the different lengths above into the definition) an equation containing θ_2:

$$cos(\theta_2) = \frac{x^2 + y^2 - \alpha_1^2 - \alpha_2^2}{2\alpha_1\alpha_2} = D \tag{0.4}$$

Furthermore, notice that if we use some of our well-known *Pythagorean Identities*, θ_2 can be found by:

$$\theta_2 = \tan^{-1}\left(\frac{\pm\sqrt{1 - D^2}}{D}\right) \tag{0.5}$$

There are several ways of solving the **Inverse Kinematics** equations, but the advantage of this approach is that both the **elbow-up** and **elbow-down** solutions are recovered by choosing the positive and negative signs in (0.5), respectively.

Moreover, it can be shown that θ_1 is given as

$$\theta_1 = \tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left(\frac{\alpha_2 \sin\theta_2}{\alpha_1 + \alpha_2 \cos\theta_2}\right) \tag{0.6}$$

Notice that the angle θ_1, depends on θ_2. This makes sense physically since we would expect to require a different value for θ_1, depending on which solution is chosen for θ_2. The derivation and specific details related to the equations stated above are covered thoroughly later on in the book.

Problem 3: Velocity Kinematics

In order to follow a contour at constant velocity, or at any prescribed velocity, we must know the relationship between the velocity of the tool and the joint velocities. In our case we can differentiate Equations (0.1) and (0.2) to obtain equations describing the velocity in the x and y direction. These equations will enable us to derive the **Jacobian** matrix of the manipulator, which is a fundamental object to determine for any manipulator.
In Chapter 5 we present a systematic procedure for deriving the Jacobian for any manipulator in the so-called cross-product form.

We can look at properties of the Jacobian to determine in which case the manipulator is said to be in a singular configuration, such as shown in Figure 1.28 for $\theta_2 = 0$:

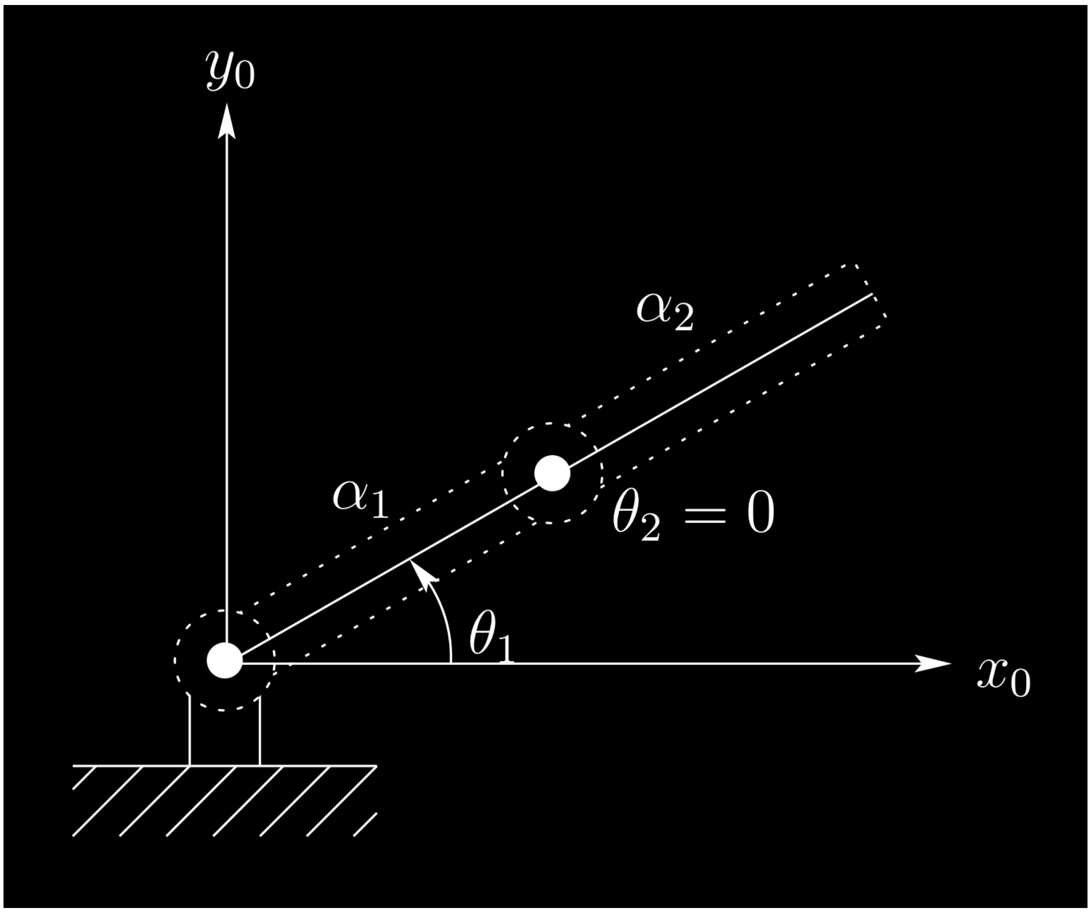

Figure 6: A singular configuration.

The determination of such singular configurations is important for several reasons. At singular configurations there are infinitesimal motions that are unachievable; that is, the manipulator end-effector cannot move in certain directions. In the above cases the end effector cannot move in the direction parallel to x_2, from a singular configuration. Singular configurations are also related to the non-uniqueness of solutions of the inverse kinematics. For example, for a given end-effector position, there are in general two possible solutions to the inverse kinematics. Note that the singular configuration separates these two solutions in the sense that the manipulator cannot go from one configuration to the other without passing through the singularity. For many applications it is important to plan manipulator motions in such a way that singular configurations are avoided.