Chapter 3

Modeling

The robots dynamics is derived in this chapter. The analysis is text book standard and no new ideas or methods are presented. This chapter may be used as a guide to modeling a high degree of freedom revolute robot manipulator. The robot in question is seen in figure 3.1. Readers looking solely for obstacle avoidance theory may skip this chapter in its entirety. We need to perform the modeling in order to implement a simulator on which to test our controller, which is our motivation for including this chapter. We will first derive the forward kinematics and then the inverse kinematics. We will then present the robot model in full as well as derive som usefull functions needed for our controller. An overview of the methods we will use can be found in [10].

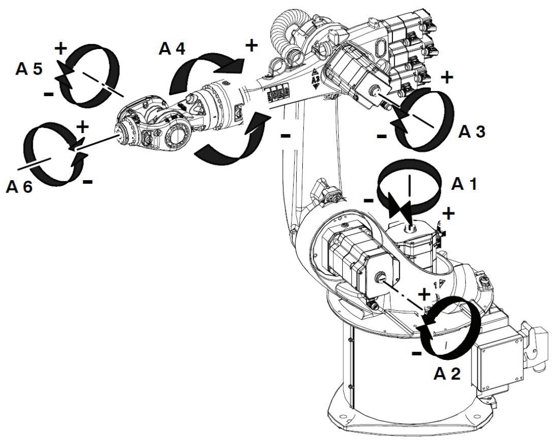

Figure 3.1: The KUKA KR16 robot with the rotational joints labeled $A_1 - A_6$ courtesy of KUKA.

3.1 Forward kinematics

The first step is to find the forward kinematics, i.e. the geometric relationship between the configuration of the robot and the position and orientation of the end effector. We derive this relationship using the DH-convention. The Denavit-Hartenberg (DH) convention is an algorithm used to derive the kinematics of a chain of rigid bodies using a small number of parameters, 4 instead of 6. This is achieved by cleverly choosing the coordinate axis for each joint. The joint diagram and appropriate coordinate axes is seen in figure 3.2.

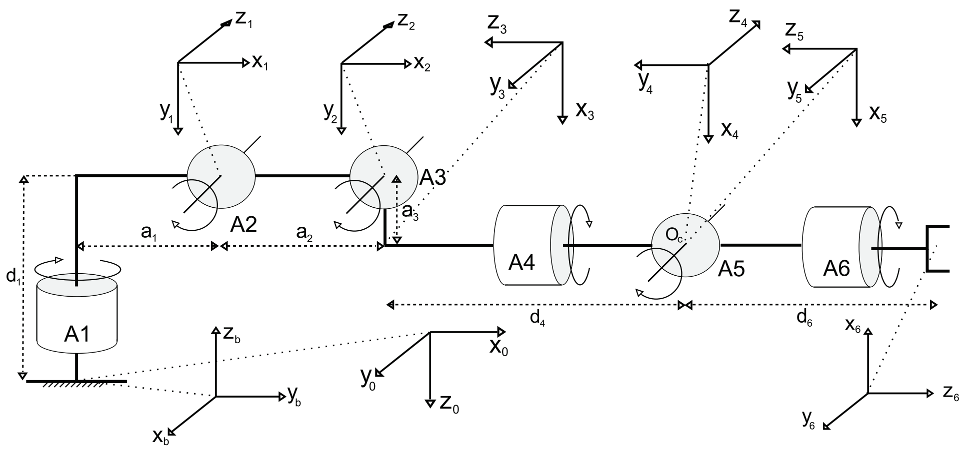

Figure 3.2: The joints of the robot labelled $A_1 - A_6$ with coordinate systems following the DH-convention. The arrows indicate the positive rotation direction. The lengths of the links are labelled a_i, d_i. The dotted lines indicate the placement of the coordinate axes.

We observe that the first three joints make up an elbow manipulator, and the last three are composed as a spherical wrist as defined in [10] p.87. We assign the right handed coordinate axis for each joint such that the z_i-vector is parallel with the rotation axis and the x_i-vector is perpendicular to z_i and z_{i-1}. The z vectors are placed such that the positive rotation angle is as it is defined for the robot from the manufacturer. The origin of each coordinate system is placed at the common normal between z_i and z_{i-1} and is arbitrary when they are parallel. The DH-parameters are given by the transformations from coordinate system i to $i+1$ via the rotational and translational transformations given in (3.1)

	q	d	a	α
0	$\frac{\pi}{2}$	0	0	π
1	q_1	$-d_1$	a_1	$\frac{\pi}{2}$
2	q_2	0	a_2	0
3	$q_3 + \frac{\pi}{2}$	0	a_3	$-\frac{\pi}{2}$
4	q_4	$-d_4$	0	$\frac{\pi}{2}$
5	q_5	0	0	$-\frac{\pi}{2}$
6	$q_6 + \pi$	$-d_6$	0	π

Table 3.1: The DH-parameters for the KUKA-16K robot where $q_i = 0\ \forall i$ results in the pose of figure 3.2. q_i are the joint variables.

$$T_i^{i+1} = \mathrm{Rot}_{z,q_i}\mathrm{Trans}_{z,d_i}\mathrm{Trans}_{x,a_i}\mathrm{Rot}_{x,\alpha_i} \tag{3.1}$$

Where Rot_{z,q_i} denotes a rotation q_i about z and Trans_{x,a_i} means a translation a_i along x. The parameters are given in table 3.1.

We get the homogeneous transformation from the base coordinate system to the end effector by multiplying the transformations in order:

$$T = T_b^6 = T_b^0 T_0^1 \cdots T_5^6 \tag{3.2}$$

We denote the base frame, or the world frame as b. The columns of forward kinematics homogeneous transformation is given by $T = [T_1, T_2, T_3, T_4]$ where $s_i = \sin(q_i)$,$c_i = \cos(q_i)$ and $c/s_{ij} = cos/sin(q_i + q_j)$.

$$T_1 = \begin{bmatrix} -c_6(c_5(c_1 s_4 - s_{23} c_4 s_1) - c_{23} s_1 s_5) - s_6(c_1 c_4 + s_{23} s_1 s_4) \\ s_6(c_4 s_1 - s_{23} c_1 s_4) + c_6(c_5(s_1 s_4 + s_{23} c_1 c_4) + c_{23} c_1 s_5) \\ -c_6(s_{23} s_5 - c_{23} c_4 c_5) - c_{23} s_4 s_6 \\ 0 \end{bmatrix} \tag{3.3}$$

$$T_2 = \begin{bmatrix} c_6(c_1 c_4 + s_{23} s_1 s_4) - s_6(c_5(c_1 s_4 - s_{23} c_4 s_1) - c_{23} s_1 s_5) \\ s_6(c_5(s_1 s_4 + s_{23} c_1 c_4) + c_{23} c_1 s_5) - c_6(c_4 s_1 - s_{23} c_1 s_4) \\ (s_{23} s_5 - c_{23} c_4 c_5) \\ 0 \end{bmatrix} \tag{3.4}$$

$$T_3 = \begin{bmatrix} s_5(c_1 s_4 - s_{23} c_4 s_1) + c_{23} c_5 s_1 \\ c_{23} c_1 c_5 - s_5(s_1 s_4 + s_{23} c_1 c_4) \\ -s_{23} c_5 - c_{23} c_4 s_5 \\ 0 \end{bmatrix} \tag{3.5}$$

$$T_4 = \begin{bmatrix} s_1(a_1 + d_4c_{23} - a_3s_{23} + a_2c_2) + d_6(s_5(c_1s_4 - s_{23}c_4s_1) + c_{23}c_5s_1) \\ c_1(a_1 + d_4c_{23} - a_3s_{23} + a_2c_2) - d_6(s_5(s_1s_4 + s_{23}c_1c_4) - c_{23}c_1c_5) \\ d_1 - a_3c_{23} - d_4s_{23} - a_2s_2 - d_6(s_{23}c_5 + c_{23}c_4s_5) \\ 1 \end{bmatrix} \tag{3.6}$$

3.2 Inverse kinematics

We derive the inverse kinematics of the robot in this section. This is a necessity if one chooses to employ joint space control as one needs to map the desired end effector trajectory to a trajectory given in the joint space. It may also be used when performing task space control to provide information about desirable poses. The inverse kinematics is the mapping from the end effector to the joint space:

$$q = T^{-1}(o, R) \tag{3.7}$$

Where $o = [x, y, z]^T$ is the position of the end effector given in the base frame and R is a rotation matrix describing the orientation of the end effector. The problem of determining the inverse kinematics is in general a nontrivial problem of solving 12 highly nonlinear equations with 6 unknowns. The problem is however not that hard given the specific robot configuration from figure 3.2 and forward kinematics in accordance with the DH-convention, [10] p.96. This stems from the fact that the three last joints make up a spherical wrist where the coordinate axes (4,5) have the same origin called the wrist center, o_c. We use this to split the inverse kinematics problem in two, a procedure that is called Kinematic decoupling.

The location of the wrist center with respect to the base frame is uniquely given by:

$$o_c = o - d_6R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.8}$$

This allows us to first find $q_{1,2,3}$ using o_c, and then finding the last 3 using the orientation of the end effector.

3.2.1 The first three joint angles

We assume that the robot is in a nonsingular configuration. $o_c^T = [x_c, y_c, z_c]^T$ is the wrist center given in the base frame, using the translations from the base to o_c we see that:

$$o_c = \begin{bmatrix} s_1(a_1 + a_2c_2 + c_{23}d_4 - a_3s_{23}) \\ c_1(a_1 + a_2c_2 + c_23d_4 - a_3s_{23}) \\ d_1 - a_3c_{23} - a_2s_2 - d_4s_{23} \end{bmatrix} \tag{3.9}$$

We find q_1 easily since since the location of o_c in the x_by_b-plane is a function of q_1, i.e.:

$$\frac{x_c}{y_c} = \tan q_1 \tag{3.10}$$

This gives us the two possible solutions for q_1:

$$\{q_{11}, q_{12}\} = \{\text{Atan2}(x_c, y_c), \text{Atan2}(x_c, y_c) + \pi\}\,; \tag{3.11}$$

Where Atan2 $\in [0, 2\pi)$ is the four quadrant arctangent function. There are two possible values for q_1, these refer to the right arm or left arm configurations.

To find the second two angles we first rotate the robot into the y_bz_b-plane since we know q_1:

$$o_{2_{\text{rot}}} = \begin{bmatrix} 0 \\ a_1 \\ d_1 \end{bmatrix} \quad o_{c_{\text{rot}}} = Rot_{z_b,q_1}o_c = \begin{bmatrix} x_c\cos(q_1) - y_c\sin(q_1) \\ y_c\cos(q_1) + x_c\sin(q_1) \\ z_c \end{bmatrix} = \begin{bmatrix} 0 \\ o_{c1} \\ o_{c2} \end{bmatrix} \tag{3.12}$$

We find q_3 by observing that the distance between o_2 and o_c is a function of q_3:

$$d = \|o_c - o_2\|^4 = a_2^2 + a_3^2 + d_4^2 - 2a_2a_3\sin(q_3) + 2a_2d_4\cos(q_3) \tag{3.13}$$

The fourth power of the distance is used since the result is slightly simpler. This gives us the two solutions for q_3:

$$\{q_{31}, q_{32}\} = \left\{ -2\text{atan}\left(\frac{a_3 + \sqrt{a_3^2 + d_4^2 - D^2}}{d_4 + D}\right), -2\text{atan}\left(\frac{a_3 - \sqrt{a_3^2 + d_4^2 - D^2}}{d_4 + D}\right) \right\} \tag{3.14}$$

Where $D = (d - a_2^2 - a_3^2 - d_4^2)/(2a_2)$. If the plus sign is chosen, then we get the q_3 angle corresponding to $q_{1,1}$. If $\Delta_i = a_3^2 + d_4^2 - D_i^2 < 0$ then we have no solutions for the given q_{1i}.

The solution for q_2 is an exercise in trigonometry. The equations one needs to solve is the wrist center position as a function of q_2 and is given by:

$$\begin{bmatrix} o_{c1} \\ o_{c2} \end{bmatrix} = \begin{bmatrix} a_1 + d_4 c_{23} - a_3 s_{23} + a_2 c_2 \\ d_1 - a_3 c_{23} - d_4 s_{23} - a_2 s_2 \end{bmatrix} \tag{3.15}$$

It's solution is:

$$q_{2k} = \text{Atan2}(o_{c1} - a_1, o_{c2} - d_1) - \text{Atan2}\left[a_2 + \delta\cos(q_{3k}^{'}), -\delta\sin(q_{3k}^{'})\right] \tag{3.16}$$

Where $\delta = \sqrt{a_3^2 + d_4^2}$, $q_{3k}^{'} = q_{3k} + \text{atan}\left(\frac{a_3}{d_4}\right)$ and $k \in \{1, 2\}$ refers to the different solutions possible for q_3.

The solution set now looks like:

$$Q_1 = \Big\{\{q_{11}, q_{21}, q_{31}\}, \{q_{11}, q_{22}, q_{32}\}, \{q_{12}, q_{21}, q_{31}\}, \{q_{12}, q_{22}, q_{32}\}\Big\} \tag{3.17}$$

$$Q_2 = \Big\{\{q_{11}, q_{21}, q_{31}\}, \{q_{11}, q_{22}, q_{32}\}\Big\} \tag{3.18}$$

$$Q_3 = \Big\{\{q_{12}, q_{21}, q_{31}\}, \{q_{12}, q_{22}, q_{32}\}\Big\} \tag{3.19}$$

$$\{q_1, q_2, q_3\} \in \begin{cases} Q_1 & \text{if} \quad \Delta_{1,2} > 0 \\ Q_2 & \text{if} \quad \Delta_1 > 0, \Delta_2 < 0 \\ Q_3 & \text{if} \quad \Delta_1 < 0, \Delta_2 > 0 \\ \emptyset & \text{if} \quad \Delta_{1,2} < 0 \end{cases}$$

Where D_i corresponds to q_{1i} and the angles within a bracket depends upon each other in the following way:

$$\{q_{11}, q_{21}, q_{31}\} \triangleq \{q_{11}, q_{21}(q_{11}, q_{31}), q_{31}(q_{11})\} \tag{3.20}$$

We see that there are either zero,two or four solution for the first three angles. They correspond to the permutations of the elbow-up/down, arm right/left robot configurations as defined in [10] p.104.

3.2.2 The last three joint angles

The last three joint angles are the Euler parameter for a given rotation matrix. We may construct the rotation matrix from the wrist center to the end effector as a function of $q_{4,5,6}$ since we now know the first three joint angles. This is R_6^3:

$$R_6^3 = R_3^0 R = r_{ij} \tag{3.21}$$

We know from the forward kinematics that:

$$R_6^3 = \begin{bmatrix} s_4 s_6 - c_4 c_5 c_6 & -c_6 s_4 - c_4 c_5 s_6 & c_4 s_5 \\ -c_4 s_6 - c_5 c_6 s_4 & c_4 c_6 - c_5 s_4 s_6 & s_4 s_5 \\ -c_6 s_5 & -s_5 s_6 & -c_5 \end{bmatrix} \tag{3.22}$$

This gives us the solutions for the last angles:

$$\{q_{41}, q_{42}\} = \{\text{Atan2}(r_{23}, r_{13}), \text{Atan2}(r_{23}, r_{13}) + \pi\} \tag{3.23}$$
$$\{q_{51}, q_{52}\} = \{\text{acos}(r_{33}), -\text{acos}(r_{33})\} \tag{3.24}$$
$$\{q_{61}, q_{62}\} = \{\text{Atan2}(-r_{32}, -r_{31}), \text{Atan2}(-r_{32}, -r_{31}) + \pi\} \tag{3.25}$$

With the solution set:

$$\{q_4, q_5, q_6\} \in \Big\{\{q_{41}, q_{51}, q_{61}\}, \{q_{42}, q_{52}, q_{62}\}\Big\} \tag{3.26}$$

There are two solutions to this problem given a triplet $q_{1,2,3}$ and a nonsingular configuration. So we either have zero, four or eight solutions for the nonsingular inverse kinematics.

3.2.3 Singularities of the inverse kinematics

A singular configuration of the robot is the case where there are an infinite number of solutions to the inverse kinematics problem. Since we can calculate the first three angles independently of the last three, we know that the singularities for the first three joints are the same ones as for an elbow manipulator. This is when the wrist center is on the z_b axis such that $o_c^T = [0, 0, o_{cz}]$. q_1 hence becomes a free variable. The solution we use here is to pick the last known value for q_1 and use this to calculate the other angles as before.

The spherical joint is in a singular configuration without self intersection when $q_5 = 0$. The matrix R_6^3 now has the form:

$$R_6^3 = \begin{bmatrix} -c_{46} & -s_{46} & 0 \\ -s_{46} & c_{46} & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{3.27}$$

As before we have a case where either q_4 or q_6 is a free variable. The solution is to assume that q_4 is the same as the last known value, and then get q_6 from the expression:

$$q_4 = \text{Atan2}(-r_{12}, -r_{11}); \tag{3.28}$$

3.3 Velocity kinematics

We consider the velocity kinematics which is the next step in the modeling procedure. The velocity kinematics describe the linear and angular velocity of points on the robot as a function of the joint velocities. The velocity is described by the Jacobian of the forward kinematics, and we specifically need the velocity of the mass centers of the links.

3.3.1 Positions of the mass centers

We obtain the position of the mass centers using the forward kinematics. We assume that the mass density in the joints are sufficiently uniform such that the mass center will be located somewhere along the line between two joints. We call these lengths r_i, and we readily compute the mass centers given in the base coordinate system:

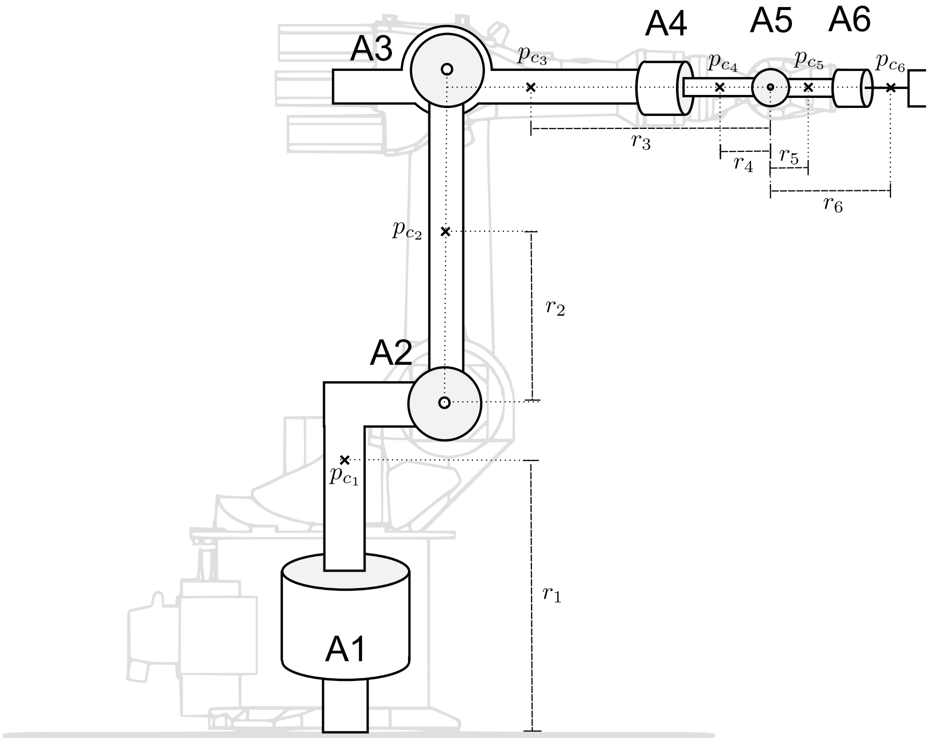

Figure 3.3: The KUKA robot with the centers of gravity labeled p_{c_i} for each link.

$$p^b_{c_1} = \begin{bmatrix} 0 \\ 0 \\ r_1 \end{bmatrix} \qquad p^b_{c_2} = o_1 + R^b_2 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} r_2$$

$$p^b_{c_3} = o_5 + R^b_4 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} r_3 \qquad p^b_{c_4} = o_5 + R^b_4 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} r_4$$

$$p^b_{c_5} = o_5 + R^b_5 \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} r_5 \qquad p^b_{c_6} = o_5 + R^b_5 \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} r_6$$

where

$$R^b_k = R^b_0 R^0_1 \cdots R^{k-1}_k, \quad T^b_k = \begin{bmatrix} R^b_k & o^b_k \\ 0 & 1 \end{bmatrix} \tag{3.29}$$

We note that the mass centers p_3, p_4 and p_5, p_6 are located on the same line.

3.3.2 The Jacobian

The Jacobian J is the mapping from the joint velocities to the linear and angular velocities of any given point on the robot. The Jacobian derived in this section is also referred to as the geometric Jacobian in order to separate it from the analytic Jacobian which we will not use. The relationship is the following:

$$v_i^b = J_{v_i}\dot{q}, \quad \omega_i^b = J_{\omega_i}\dot{q} \tag{3.30}$$

Where $\omega, v \in \mathbb{R}^{3\times 1}$ are respectively the linear and angular velocities and $J_{\omega_i}, J_{v_i} \in \mathbb{R}^{3\times 6}$. We get the Jacobian for the linear velocities of the mass centers either by differentiation of the mass centers with respect to time.

$$v_i^b = \dot{p}_{c_i}^b = J_{v_i}\dot{q} \quad \Rightarrow \quad J_{v_i} = \frac{\partial p_{c_i}^b}{\partial q} \tag{3.31}$$

Where $q = [q_1, \ldots, q_6]^T$. Alternatively we may use the handy formula from [10] p.133 which holds true for a *fixed* point p on the robot with revolute joints:

$$J_p(q) = \begin{bmatrix} z_0 \times (p - o_0) & z_0 \times (p - o_1) & \ldots z_0 \times (p - o_5) \end{bmatrix} \tag{3.32}$$

Where the row i of J_p is the zero vector if p is not a function of q_i. From [10] p.133 we have the following Jacobian for the angular velocities given revolute joints using the DH-convention:

$$J_{\omega_i} = \begin{bmatrix} z_0^b u(i-1) & z_1^b u(i-2) & z_2^b u(i-3) & \cdots & z_6^b u(i-6) \end{bmatrix} \tag{3.33}$$

Where u is the Heaviside function, $u(x) = 1\,\forall\, x \geq 0, u(x) = 0\,\forall\, x < 0$ and z_i^b is the z vector of the coordinate system i expressed in the base frame. We have for instance that:

$$J_{\omega_3} = \begin{bmatrix} z_0^b & z_1^b & z_2^b & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} R_0^b \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & R_1^b \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & R_2^b \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & 0 & 0 & 0 \end{bmatrix} \tag{3.34}$$

The Jacobian for the first two mass centers are as follows:

$$J_1 = \begin{bmatrix} J_{v_1} \\ J_{\omega_1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.35}$$

$$J_2 = \begin{bmatrix} J_{v_2} \\ J_{\omega_2} \end{bmatrix} = \begin{bmatrix} c_1(a_1 + r_2 c_2) & -r_2 s_1 s_2 & 0 & 0 & 0 & 0 \\ -s_1(a_1 + r_2 c_2) & -r_2 c_1 s_2 & 0 & 0 & 0 & 0 \\ 0 & -r_2 c_2 & 0 & 0 & 0 & 0 \\ 0 & -c_1 & 0 & 0 & 0 & 0 \\ 0 & s_1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.36}$$

We see that the linear velocity of the first mass center is zero and that $\omega_1 = -\dot{q}_1$ as is apparent by inspecting figure 3.3.

3.4 The singularities of the Jacobian

The Jacobian does not in general have full rank. The angles where J looses rank are called singularities and are of interest as a task space synchronization controller uses the inverse of the Jacobian. These singularities are the same as for the inverse kinematics as the nonsingularity of the Jacobian is a necessary and sufficient condition for the forward kinematics to be a diffeomorphism. These joint angle values have a physical interpretation as the robot looses one or more degrees of freedom at a singularity. This happens for instance when the robot is stretched out fully in one direction, and the end effector is on the boundary of the reachable workspace.

We present the determinant of the jacobian for inspection as we have allready considered the singularities in view of the inverse kinematics.

$$\det\{J\} = a_2 s_5 \left[a_3 c_3 + d_4 s_3\right] \left[a_1 + d_4 c_{23} - a_3 s_{23} + a_2 c_2\right] \tag{3.37}$$

3.5 The joint space dynamical model

We are now ready to state the dynamical robot model given in the joint space. From [10] p.254 we have that the mass matrix M in equation (1.6) is given by:

$$M(q) = \sum_{i=1}^{6} \left\{ m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i^b(q) I_i R_i^b(q)^T J_{\omega_i}(q) \right\} \tag{3.38}$$

Where m_i are the mass of link i and I_i is the inertia matrix for link i about the mass center expressed in the body attached frame. We see from figure 3.2 that he body attached frames to the mass centers $[p_{c_1}, p_{c_2}, p_{c_3}, p_{c_4}, p_{c_5}, p_{c_6}]$ are in order $[R_1^b, R_2^b, R_3^b, R_4^b, R_5^b, R_6^b]$. We assume that the inertia matrices for the last three joints are diagonal since the last three joint are small compared to the first three, i.e:

$$I_k = \begin{bmatrix} I_{k_{xx}} & I_{k_{xy}} & I_{k_{xz}} \\ I_{k_{xy}} & I_{k_{yy}} & I_{k_{yz}} \\ I_{k_{xz}} & I_{k_{yz}} & I_{k_{zz}} \end{bmatrix} \forall\, k \in \{1,2,3\}, \quad I_k = \begin{bmatrix} I_{k_{xx}} & 0 & 0 \\ 0 & I_{k_{yy}} & 0 \\ 0 & 0 & I_{k_{zz}} \end{bmatrix} \forall\, k \in \{4,5,6\} \tag{3.39}$$

The full mass matrix is rather large, and we would need about 3 pages to write it out explicitly assuming one page has 50 lines.

The Coriolis matrix C is given by:

$$C(q,\dot{q}) = \begin{bmatrix} \dot{q}^T C_1(q) \\ \vdots \\ \dot{q}^T C_6(q) \end{bmatrix} \tag{3.40}$$

Where the elements of C_k are given by:

$$C_{k_{(i,j)}} = \frac{1}{2} \left\{ \frac{\partial M_{(k,j)}}{\partial q_i} + \frac{\partial M_{(k,i)}}{\partial q_j} - \frac{\partial M_{(k,i)}}{\partial q_k} \right\} \tag{3.41}$$

The Coriolis matrix is even larger than M and we would need about 5 pages to write it down. We note that both M and C are independent of q_1. This is justified by observing that the robots kinetic energy is independent of its base angle.

The gravity vector $G(q)$ is defined as:

$$G(q) = \frac{\partial P}{\partial q} \tag{3.42}$$

Where P is the potential energy of the robot given in the world frame as:

$$P_i = m_i g p^b_{c_1 z} \tag{3.43}$$

Where $p^b_{c_1 z}$ is the height of the centers of gravity for each link.

Some of the shortest elements of the system matrices are given by:

$$\begin{aligned}
M_{(5,6)} &= 0 \\
M_{(4,6)} &= I_{6_{zz}} \cos(q_5) \\
M_{(3,6)} &= M_{(2,6)} = I_{6_{zz}} \sin(q_4)\sin(q_5) \\
M_{(4,5)} &= -\cos(q_6)\sin(q_5)\sin(q_6)(I_{6_{xx}} - I_{6_{yy}}) \\
C_{(5,5)} &= \frac{1}{2}\dot{q}_6 \sin(2q_6)(I_{6_{xx}} - I_{6_{yy}}) \\
C_{(6,6)} &= 0
\end{aligned}$$

3.6 Task space mapping

In order to express our dynamical robot model in the task space, we need to do a little more work. We will need to find a representation of the orientation of the end effector using a $\mathbb{R}^{3\times 1}$ vector. We also need the derivative of the Jacobian with respect to time. We will not explicitly derive the task space dynamics as is common using the *Analytic Jacobian* [10] p.140. We will later use the roll pitch yaw angle error in place of the actual roll pitch yaw angles for the end effector when performing this mapping. We will see later why this is advantageous. Writing down an analytic Jacobian for the orientation error dynamics is possible, but one needs a matrix with a translational mapping to achieve this. This will be the final step of our lengthy modeling exercise.

3.6.1 A minimal representation $SO(3)$

We need to use a minimal representation of $SO(3)$ in order to find a robot model that is useful for task space control. The orientation information is given as a rotation matrix, which is subject to three normality constraints and three orthogonality constraints. So expressing an orientation using a rotational matrix is highly redundant as only three degrees of freedom are left after accounting for the constraints. Representing $SO(3)$ by three parameters may be done in a number of ways, but sadly there are no nonsingular minimal representations of $SO(3)$. That is, there exsists no mappings from $R \in SO(3) \mapsto \Phi \in \mathbb{R}^{3\times 1}$ such that the mapping and its inverse are defined for all $R \in SO(3)$ for both position and velocity. A nonsingular representation is however possible using quaternions, but a quaternion is not a minimal representation and using quaternions will result in nonlinear closed loop system. Commonly used minimal representations are axis/angle and Euler angles. We will use Euler angles, and specifically roll-pitch-yaw angles to parametrize orientation. We choose this parametrization since it is intuitive, easily implementable, and it produces a linear orientation error system. Used in a certain way it is also possible to place the singularity 90° away from the desired orientation.

The roll pitch yaw angles are defined as the ZYX Euler parameters ϕ, θ, ψ such that:

$$R = R_z(\phi)R_y(\theta)R_x(\psi) \tag{3.44}$$

Where $R_z(\phi)$ denotes a rotational about the z-axis with ϕ radians. Given a rotational matrix R, it is possible to extract the roll pitch yaw angles with:

$$\begin{aligned} \psi &= \text{Atan2}(R_{32}, R_{33}) \\ \phi &= \text{Atan2}(R_{21}, R_{11}) \\ \theta &= \text{Atan2}(-R_{31}, \cos(\phi)R_{11} + \sin(\phi)R_{21}) \end{aligned} \tag{3.45}$$

Where Atan2 is the two argument arcatan function. It is possible to find a linear relationship between the velocity of the Euler parameters and the angular velocity of a rotational matrix by differentiating the Euler parameters with respect to time. If we denote $\Phi = [\phi, \theta, \psi]^T$ as the vector consisting of the roll pitch yaw angles, and the angular velocity ω given by the relationship between the joint velocity and the geometric Jacobian $\omega = J_\omega \dot{q}$ then we have:

$$\omega = B\dot{\Phi} = \begin{bmatrix} 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 1 & 0 & -\sin(\theta) \end{bmatrix} \dot{\Phi} \tag{3.46}$$

3.6.2 The time derivative of the Jacobian

We will need the time derivative of the Jacobian in order to achieve a mapping from the joint space to the task space. This motivates us to find a closed form expression of the time derivative of the Jacobian as it is needed for our controller.

We recall that J consists of two matrices J_{v_p} and J_{ω_p} respectively mapping the joint velocities to the linear velocity of a fixed point p on the robot and its angular velocity. We assume that p is some point on the robot, the i'th column of J_{v_p} is either given by $z_i(q) \times (p(q) - o_i(q))$ or it is the zero vector. The i'th column of J_ω is given by $z_i(q)$ or zero.

The time derivative of J may be taken column wise, so we may without loss of generality consider a column of J_ω:

$$\frac{d}{dt} z_i = \omega_i \times z_i = J_{\omega_i}\dot{q} \times z_i \tag{3.47}$$

The result is straightforward to derive, and the derivation identities used may be found in [5] p.243. The derivative of a column of J_{v_p} is given by:

$$\Big[z_i(q) \times (p(q) - o_i(q))\Big] = \frac{d}{dt} z_i(q) \times (p(q) - o_i(q)) + z_i(q) \times \frac{d}{dt}(p(q) - o_i(q)) \tag{3.48}$$

Where we have used the fact that the cross product obeys the product rule. Substituting in $\dot{z}_i$, $\dot{q} = J_{v_q}\dot{q}$ and $\dot{o}_i = J_{v_{o_i}}\dot{q}$ gives:

$$\frac{d}{dt}\Big[z_i(q) \times (p(q) - o_i(q))\Big] = [(J_{\omega_i}\dot{q}) \times z_i] \times (p - o_i) + z_i \times [(J_{v_p} - J_{o_i})\dot{q}] \tag{3.49}$$

Where the k'th column of J_{v_p} and J_{o_i} are given by:

$$\mathrm{Col}_k\{J_{v_p}\} = z_k \times (p - o_k) \tag{3.50}$$
$$\mathrm{Col}_k\{J_{o_i}\} = z_k \times (o_i - o_k) \tag{3.51}$$

And are zero for $k > i$. All these expressions are given as explicit functions of q and $\dot{q}$ which make them suitable for on-line implementation.