Exercise 1 – Kinematics of a robotic leg

Overview

In this exercise, we are going to analyze the kinematics of a single robot leg with a point foot. Starting with a set of generalized coordinates, we will elaborate relative rotations, translations, and homogeneous transformations. Subsequently, using foot point Jacobians, we will describe contact constraints and perform a trajectory tracking task using inverse kinematics.

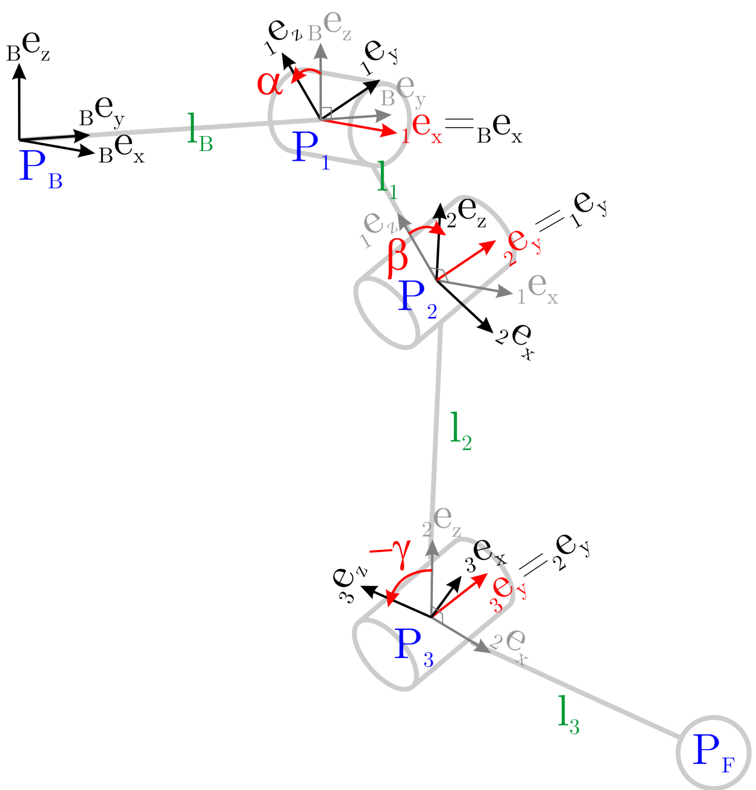

Figure 1: Kinematics of a single robot leg.

Figure 1 illustrates a single leg that is attached to a body fixed frame B. This leg has three degrees of freedom consisting of relative rotations about α (alpha) around ${}_B\mathbf{e}_x$, about β (beta) around ${}_1\mathbf{e}_y$, and about γ (gamma) around ${}_2\mathbf{e}_y$. Hence, the generalized coordinates are given by $\mathbf{q} := [\alpha\ \beta\ \gamma]^T$.

This exercise has an associated Python script (`Ex1.py`) which provides a baseline skeleton of the code. Most questions can be solved by hand or through other methods, but may be easier via the scripts. Questions where Python is necessary are indicated (Python). Before running the code, please read the associated README for installation instructions.

Q1 Relative Rotation Matrix

Given the kinematic description of a single leg with three degrees of freedom ($\mathbf{q} := [\alpha\ \beta\ \gamma]^T$) we determine the relative rotation matrices $\mathbf{R}_{AC}$ rotating a vector $\mathbf{R}$ from an arbitrary coordinate system C to A:

$$ {}_A\mathbf{r} = \mathbf{R}_{AC}\ {}_C\mathbf{r}. \tag{1}$$

As a function of the generalized coordinates α, β, γ, what are the three relative rotation matrices $\mathbf{R}_{B1}$, $\mathbf{R}_{12}$, $\mathbf{R}_{23}$?

Answer:

$$\mathbf{R}_{B1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$\mathbf{R}_{12} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$\mathbf{R}_{23} = \begin{bmatrix} \cos\gamma & 0 & \sin\gamma \\ 0 & 1 & 0 \\ -\sin\gamma & 0 & \cos\gamma \end{bmatrix}$$

Q2 Homogeneous Transformation

Given the relative rotation matrices from the previous problem and choosing unitary link lengths ($l_B = l_1 = l_2 = l_3 = 1$), we determine the homogeneous transformation $\mathbf{H}$ that transforms the footpoint position $\mathbf{r}_F$ represented in coordinate frame 3 to coordinate frame B:

$$\begin{bmatrix} {}_B\mathbf{r}_{BF} \\ 1 \end{bmatrix} = \mathbf{H}_{B3} \begin{bmatrix} {}_3\mathbf{r}_{3F} \\ 1 \end{bmatrix}. \tag{2}$$

As a function of the generalized coordinates α, β, γ, what are:

1. the relative position vectors ${}_1\mathbf{r}_{12}$, ${}_2\mathbf{r}_{23}$, ${}_3\mathbf{r}_{3F}$, and ${}_B\mathbf{r}_{B1}$?
2. the homogeneous transformation matrices $\mathbf{H}_{B1}$, $\mathbf{H}_{12}$, and $\mathbf{H}_{23}$?
3. the foot point position vector in B, ${}_B\mathbf{r}_{BF}$?

Answer:

1. Since these are all in their local frames, ${}_1\mathbf{r}_{12} = {}_2\mathbf{r}_{23} = {}_3\mathbf{r}_{3F} = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T$, and ${}_B\mathbf{r}_{B1} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$.
2. Homogeneous transformations can be written as composed from the relative rotation matrices and the relative position vectors in those frames:

$$\mathbf{H}_{B1} = \left[\begin{array}{ccc|c} & \mathbf{R}_{B1} & & {}_B\mathbf{r}_{B1} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 1 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{H}_{12} = \left[\begin{array}{ccc|c} & \mathbf{R}_{12} & & {}_1\mathbf{r}_{12} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

$$\mathbf{H}_{23} = \left[\begin{array}{ccc|c} & \mathbf{R}_{23} & & {}_2\mathbf{r}_{23} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

3. To calculate the foot point position vector, first create the cumulative transformation matrix $\mathbf{H}_{B3} = \mathbf{H}_{B1}\mathbf{H}_{12}\mathbf{H}_{23}$. Then, using Eq. 2, solve for ${}_B\mathbf{r}_{BF}$:

$$
\begin{aligned}
{}_B\mathbf{r}_{BF} &= \begin{bmatrix} -\sin\beta - \cos\beta\sin\gamma - \cos\gamma\sin\beta \\ \sin\alpha + \cos\beta\sin\alpha + \cos\beta\cos\gamma\sin\alpha - \sin\alpha\sin\beta\sin\gamma + 1 \\ \cos\alpha\sin\beta\sin\gamma - \cos\alpha\cos\beta - \cos\alpha\cos\beta\cos\gamma - \cos\alpha \end{bmatrix} \\
&= \begin{bmatrix} -\sin(\beta+\gamma) - \sin(\beta) \\ \sin(\alpha)(\cos(\beta+\gamma) + \cos(\beta) + 1) + 1 \\ -\cos(\alpha)(\cos(\beta+\gamma) + \cos(\beta) + 1) \end{bmatrix}
\end{aligned}
$$

Q3 Jacobians and Differential Kinematics

Given the end effector position ${}_B\mathbf{r}_{BF}$ as a function of generalized coordinates $\mathbf{q}$:

1. determine the corresponding Jacobian $\mathbf{J}_{BF} = \frac{\partial_B\mathbf{r}_{BF}}{\partial\mathbf{q}}$.

2. determine the generalized coordinate velocity $\dot{\mathbf{q}}_t$ from a current configuration $\mathbf{q}_t = \begin{bmatrix}0 & 60^\circ & -120^\circ\end{bmatrix}^T$ and a given target foot velocity ${}_B\dot{\mathbf{r}}_{BF} = \begin{bmatrix}0 & 0 & -1\text{m/s}\end{bmatrix}^T$. (NOTE: This will require a matrix inversion, we recommend using a programming tool to help with this.)

Answer:

1. The Jacobian can be found by taking the partial derivative of ${}_B\mathbf{r}_{BF}$ with respect to the generalized coordinates $\mathbf{q}$. Recall that $\frac{\partial\sin(x+y)}{\partial x} = \cos(x+y)$ (make the substitution $z = x+y$, then $\partial z = \partial x$). Thus:

$$
\mathbf{J}_{BF} = \begin{bmatrix} 0 & -\cos(\beta+\gamma) - \cos(\beta) & -\cos(\beta+\gamma) \\ \cos(\alpha)(\cos(\beta+\gamma) + \cos(\beta) + 1) & -\sin(\alpha)(\sin(\beta+\gamma) + \sin(\beta)) & -\sin(\beta+\gamma)\sin(\alpha) \\ \sin(\alpha)(\cos(\beta+\gamma) + \cos(\beta) + 1) & \cos(\alpha)(\sin(\beta+\gamma) + \sin(\beta)) & \sin(\beta+\gamma)\cos(\alpha) \end{bmatrix}
$$

2. To solve this, we use the equation $\dot{\mathbf{r}} = \mathbf{J}\dot{\mathbf{q}}$. Thus, $\dot{\mathbf{q}} = \mathbf{J}^{-1}\dot{\mathbf{r}}$. Substituting the specified target velocity ${}_B\dot{\mathbf{r}}_{BF}$ and current configuration $\mathbf{q}_t$ into the Jacobian found in the previous step yields:

$$
\begin{aligned}
\dot{\mathbf{q}}_t &= \mathbf{J}^{-1}_{BF|q=q_t}\,{}_B\dot{\mathbf{r}}_{BF} \\
&= \begin{bmatrix} 0 & -1 & -\frac{1}{2} \\ 2 & 0 & 0 \\ 0 & 0 & -\frac{\sqrt{3}}{2} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ -\frac{\sqrt{3}}{3} \\ \frac{2\sqrt{3}}{3} \end{bmatrix}
\end{aligned}
$$

Q4 Numerical Inverse Kinematics

In this exercise, we will solve an inverse kinematics problem of finding the generalized coordinates $\mathbf{q}_{goal}$ for a given target pose ${}_B\mathbf{r}_{BF,goal}$.

1. (Python) First, create a simple program to solve this problem numerically. We have an existing functional form of ${}_B\mathbf{r}_{BF} = {}_B\mathbf{r}_{BF}(\mathbf{q})$. Since the functional form of ${}_B\mathbf{r}_{BF}$ is often hard to analytically invert, we will implement a numerical approach. See the Q4 block in the associated `Ex1.py` template code to get started. HINT: A simple gradient descent is sufficient for this task. Initialise the estimate of the generalized coordinates $\mathbf{q}$ to the current state $\mathbf{q}_t$. Then, calculate the corresponding ${}_B\mathbf{r}_{BF}(\mathbf{q})$. In each step, calculate a new estimate for $\mathbf{q}$ using $\mathbf{q}' = \mathbf{q} + \mathbf{J}^{-1}(\mathbf{r}_{goal} - \mathbf{r})$, where the inversion can be approximated using the Moore-Penrose pseudo-invserse (`np.linalg.pinv(.)` in Python).

2. With unitary segment length, and ${}_B\mathbf{r}_{BF,goal} = \begin{bmatrix}0.2 & 0.5 & -2\end{bmatrix}^T$, how many different solutions $\mathbf{q}_{goal}$ exist?

3. The following goal position ${}_B\mathbf{r}_{BF,goal} = \begin{bmatrix}-1.5 & 1 & -2.5\end{bmatrix}^T$ is not within the reaching range of our leg. What are the generalized coordinates that bring the leg as close as possible to the target position?

Answer:

1. (See Ex1_answer.py)
2. 2 solutions - First, notice that the goal is within reach. Then, notice that the rotations around β and γ are in the same plane (since ${}_2\mathbf{e}_y$ and ${}_3\mathbf{e}_y$ are always aligned), so these two coordinates are coupled. The two solution configurations arise from the resulting redundancy (e.g. when plotting l_2 and l_3 as a rhomboid where P_2 and P_F are fixed, P_3 can be on either side of the line $\overline{P_2P_F}$). Only in the redundant fully extended or folded cases ($\gamma \in \{0, \pi\}$) where P_3 is on $\overline{P_2P_F}$ does this redundancy dissapear.
3. $\mathbf{q} = \begin{bmatrix}0 & 45° & 0\end{bmatrix}$. This can be constructed geometrically. Since the target point is in the $y = 1$ plane α must be 0. By extending toward a point that is out of reach one would stretch the leg as much as possible (i.e. $\gamma = 0$) and then compute β such that $|P_f - P_{goal}|$ is minimal. In our case, since P_{goal} relative to joint P_2 in the y-plane is $\begin{bmatrix}1.5 & 0 & 1.5\end{bmatrix}$, the angle needs to be 45°.

Q5 Trajectory Following: Inverse Differential Kinematics

Extending from the previous parts, we will now build a working controller based on inverse differential kinematics to control a footpoint trajectory. Given a desired footpoint goal trajectory $\mathbf{r}_F(t)$ and a starting configuration $\mathbf{q}_0$, we have to determine the joint velocities $\dot{\mathbf{q}} = \begin{bmatrix}\dot{\alpha} & \dot{\beta} & \dot{\gamma}\end{bmatrix}^T$. The target trajectory is a basic circle in the xz-plane parameterized by time, $\mathbf{r}_F(t) = \mathbf{c} + \rho\begin{bmatrix}\sin(2\pi f t) & 0 & \cos 2\pi f t)\end{bmatrix}^T$, for a given center location $\mathbf{c}$, radius ρ and frequency f. The target velocity is simply $\dot{\mathbf{r}}_F = \frac{d\mathbf{r}_F}{dt}$.

1. (Python) Implement a simple proportional controller to determine the desired Cartesian foot point velocity (i.e. $\dot{\mathbf{r}}(t) = \dot{\mathbf{r}}_F(t) + K_p(\mathbf{r}_F(t) - \mathbf{r})$) and apply inverse differential kinematics to determine the joint velocities. The code will then visualize the system respose using the `Ex1_extras.TrajectoryPlotter` function. Try different values for the controller gain and different parameters for the trajectory to see the effect on the system response.
2. Given a desired foot point velocity $\dot{\mathbf{r}}_F$ and generalized coordinates $\mathbf{q}$, which of the following statements are correct?
   (a) If the leg is in a non-singular configuration, there is a unique solution for $\dot{\mathbf{q}}$
   (b) In theory, $\dot{\mathbf{r}}_F$ can always be achieved
   (c) $\gamma = 0$ makes the foot point Jacobian rank deficient

Answer:

1. (See Ex1_answer.py)
2. (a) True
   (b) False
   (c) True