Prof. Dr. Katja Mombaur,
Benjamin Reh,
Silvan Lindner

Robotics 1 (WS 2018/2019)

Exercise Sheet 5

Presentation during exercises in calendar week 49

Exercise 5.1 – Bouncing Ball

In this exercise we combine the integrator from sheet02 in a bouncing ball simulation with air friction. Furthermore we learn how to draw arrows in the meshup visualization.

1. **Free fall**
   Download the example, it contains the integrator and a simple meshup model of a ball. We first look at only the z component of the falling ball without friction.

   $$\ddot{z} = -g \tag{1}$$

   Perform an order reduction as explained in sheet02 and write the output to the file `animation.csv`. The Ball should fall downwards in an accelerated motion.

2. **Reflection**
   To let the ball bounce off the $z = 0$ plane, we regard it as a full elastic impact. Therefore we can simple reverse the sign of the velocity $\dot{z}$ when the ball reaches the floor.

   One characteristic of elastic impacts is the conservation of energy. Why is the energy conserved in this case of flipping the sign? How can we see in the simulation, that the energy is conserved?

3. **Air resistance**
   Objects moving through air receive a force acting against the direction of relativ motion.

   $$F = \frac{1}{2} \cdot A \cdot C_w \cdot \rho_{air} \cdot v_{rel}^2 \tag{2}$$

   with A being the covered area, C_w a constant relating to the form of the area and ρ the density of the air.

   Using Newton's law this can be written in our case as

   $$\ddot{z} = -\underbrace{\frac{1}{2} \cdot m^{-1} \cdot A \cdot C_w \cdot \rho_{air}}_{k} \cdot \dot{z} \cdot |\dot{z}| = -k \cdot \dot{z} \cdot |\dot{z}| \tag{3}$$

   using $k \geq 0$ as a constant factor. Be aware to conserve the sign for the direction of motion by replacing v^2 with $v \cdot |v| = v^2 \cdot \mathrm{sign}(v)$. Implement the air friction and observe the changed behaviour. Guess a value for k that makes the motion look "realistic".

4. **3D**
   Extend the model to be working in all three dimensions. Add four more virtual walls around the checkers board at ± 15 to keep the ball inside the space. Set the the initial values for $\dot{x}$ and $\dot{y} \neq 0$ to make the ball move in the two new dimenions.

5. **Wind**
   Adding wind is simple, as v_{rel} is the velocity between the air and the object. Add wind to the simulation.

6. **Arrows**
   Meshup offers the possibility to draw linear and rotational arrows, called "forces" and "torques" in the meshup nomenclature. This interface is a bit peculiar, so please be aware of the following:

   - The values for arrows are written in a separate CSV-file that <u>must</u> have the `.ff` ending. The format is
     `t, plx, ply, plz, mlx, mly, mlz, prx, pry, prz, mrx, mry, mrz` .
     `t` is the time, `plx` is the x-position of the linear arrow, `mly` is the y-magnitude (length) of the linear arrow, and `mrz` is the magnitute of the rotational arrow for z, and so forth...
   - To load them, start meshup with
     `meshup modelfile.lua animationfile.txt arrowfile.ff`
   - The position of the arrow is defined by its tip.
   - The magnitude of the arrows are scaled down internally. You will have to scale them up until you can see the arrows.
   - Although Meshup is in use within the group for several years, there are still bugs which will eventually be fixed.
     - When using arrows, disable shadows as it sometimes prevens arrows from being visible.
     - Several bugs already have been fixed since the beginning of this lecture. It is advisable to use the latest version of Meshup.

   Visualize the velocity vector as a linear vector attached to the ball. Since the arrow's reference point is defined by its tip, it is sufficient to visualize the arrow as a tail following the ball.

Exercise 5.2 – Rotation Matrices and Angular Velocity

Linear velocity can be computed straight forward by derivation of the position with respect to time. Obtaining the angular velocity from a rotation matrix, however, is not as intuitive, as a 3 dimensional vector is to be obtained from a 3×3 matrix. The angular velocity $\omega \in \mathbb{R}^3$ of a rotation matrix can be calculated as

$$[\omega]_\times = \dot{R}(t) R^T(t), \tag{4}$$

where the $[\,\cdot\,]_\times$ operator is defined as

$$[\omega]_\times := \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}.$$

- Consider the function
$$f(x, y) := [x]_\times \cdot y$$
for vectors $x, y \in \mathbb{R}^3$. What is the name of the operation defined by $f(\cdot, \cdot)$?

- Verify result (4) using Rodrigues' rotation formula.
Reminder: Rodrigues' Rotation Formula: The rotation matrix defined by angle-axis vector $v \in \mathbb{R}^3$, where $\|v\|_2$ describes the rotation angle and $v/\|v\|_2$ the rotation axis, is given by
$$\exp([v]_\times) = I + \frac{\sin(\|v\|_2)}{\|v\|_2} [v]_\times + \left(\frac{1 - \cos(\|v\|_2)}{\|v\|_2^2} \right) [v]_\times^2 \tag{5}$$
Hint: The formulation (5) defines a matrix exponential, so it is sufficient to apply the chain-rule to the left side of the equation when calculating the time derivative.

- Rotational interpolation: Construct a rotation, that rotates a point $p_0 \in \mathbb{R}^3$ around the rotational center $o \in \mathbb{R}^3$ onto $p1 \in \mathbb{R}^3$ (with $\|p_0 - o\|_2 = \|p_1 - o\|_2$) in a fixed time $[0, t_{end}]$ with given angular velocity $\omega(t) : [0, t_{end}] \to \mathbb{R}$. What properties must ω have? It is sufficient to give the interpolated rotation in terms of matrix exponentials (see left side of equation (5)).

Exercise 5.3 – Geometric Jacobian

You have learned about differential kinematics in the lecture. Given the geometric jacobian $J(\cdot)$, the end-effector linear velocity $\dot{p}_e \in \mathbb{R}^3$ and the angular velocity $\omega_e \in \mathbb{R}^3$ can be expressed as a function of joint configurations $q \in \mathbb{R}^n$ and joint velocities $\dot{q} \in \mathbb{R}^n$

$$\begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = \underbrace{\begin{bmatrix} J_{P_1}(q) & \cdots & J_{P_n} \\ J_{O_1}(q) & \cdots & J_{O_n} \end{bmatrix}}_{J(q)} \dot{q}, \tag{6}$$

where the Jacobian $J \in \mathbb{R}^{6\times n}$ can be split into velocity related parts $J_{P_i} = \partial p_e/\partial q_i \in \mathbb{R}^3$ and angular velocity components $J_{O_i} \in \mathbb{R}^3$. While the geometric Jacobian can explicitly be calculated from the forward kinematics formulation, a more structured approach can be applied after analyzing the contribution of different joint types to linear and angular velocity in the Jacobian. Using the DH-parametrization, this contribution can directly be computed.

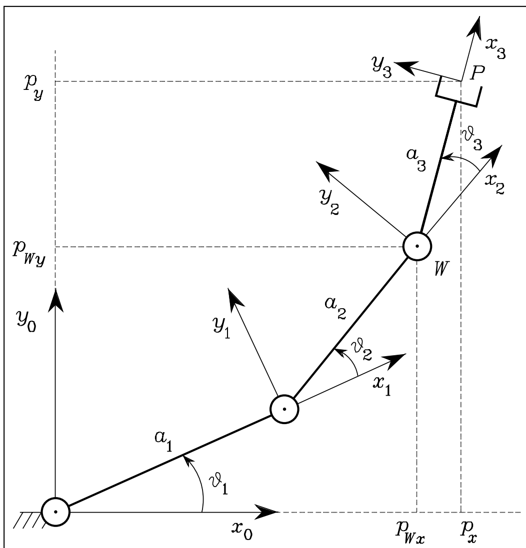
If joint i is revolute, the partial derivatives are given as

drehgelenk

$$\begin{aligned} J_{P_i} &= z_{i-1} \times (p_e - p_{i-1}) \\ J_{O_i} &= z_{i-1} \end{aligned} \tag{7}$$

where p_e is the end-effector position in world-frame, p_i the origin of the i-th DH-frame in world-coordinates and z_i the rotation axis of joint i represented in world frame.

- Give a formulation similar to (7) for prismatic joints.
- Use formulation (7) to calculate the geometric Jacobian of the three-link planar arm given in figure 1.
- As the three-link-arm is planar we are not interested in the linear velocity z-component and the angular velocity components x and y. Remove them from the Jacobian and use it to formulate a 2D-version of (6). Calculate the determinant of the resulting 3×3 Jacobian and identify all singularities.



The homogeneous transformation matrix between two consecutive DH-frames for the three-link planar arm (figure 1) is given as

$$T_i^{i-1}(\vartheta_i) = \begin{bmatrix} c_i & -s_i & 0 & a_i c_i \\ s_i & c_i & 0 & a_i s_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad i = 1, 2, 3$$

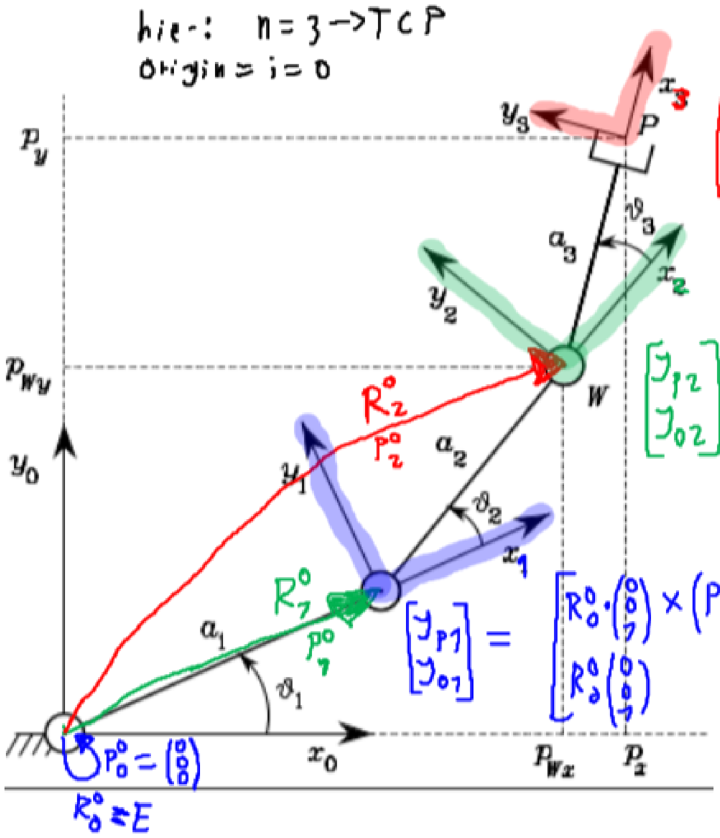
where

$$s_i := \sin(\vartheta_i)$$
$$c_i := \cos(\vartheta_i)$$

Figure 1: Three-link planar arm

hier: $n=3 \rightarrow$ TCP

origin $= i = 0$



$$\begin{bmatrix} J_{P3} \\ J_{O3} \end{bmatrix} = \begin{bmatrix} R_2^0 \cdot \begin{pmatrix}0\\0\\1\end{pmatrix} \times \left(p_3^0 - p_2^0\right) \\ R_2^0 \cdot \begin{pmatrix}0\\0\\1\end{pmatrix} \end{bmatrix}$$

$$\begin{bmatrix} J_{P2} \\ J_{O2} \end{bmatrix} = \begin{bmatrix} R_1^0 \cdot \begin{pmatrix}0\\0\\1\end{pmatrix} \times \left(p_3^0 - p_1^0\right) \\ R_1^0 \cdot \begin{pmatrix}0\\0\\1\end{pmatrix} \end{bmatrix}$$

$$\begin{bmatrix} J_{P1} \\ J_{O1} \end{bmatrix} = \begin{bmatrix} R_0^0 \cdot \begin{pmatrix}0\\0\\1\end{pmatrix} \times \left(p_3^0 - p_0^0\right) \\ R_0^0 \cdot \begin{pmatrix}0\\0\\1\end{pmatrix} \end{bmatrix}, \quad \text{mit } p_0^0 = \begin{pmatrix}0\\0\\0\end{pmatrix} \;\&\; R_0^0 = E = \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$$

$$J = \begin{bmatrix} J_{P1} & J_{P2} & J_{P3} \\ J_{O1} & J_{O2} & J_{O3} \end{bmatrix} \qquad p_3^0 = \begin{pmatrix} a_1c_1 + a_2c_2 + a_3c_3 \\ a_1s_1 + a_2s_2 + a_3s_3 \\ 0 \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

$$J_{P1} = \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix} \begin{pmatrix}0\\0\\1\end{pmatrix} \times p_3^0 = \begin{pmatrix}0\\0\\1\end{pmatrix} \times \begin{pmatrix}a\\b\\c\end{pmatrix} = \begin{pmatrix}-b\\a\\0\end{pmatrix} = \begin{pmatrix} -a_1s_1 - a_2s_2 - a_3s_3 \\ a_1c_1 + a_2c_2 + a_3c_3 \\ 0 \end{pmatrix}$$

$$\begin{matrix} 0 & a \\ 0 & b \end{matrix}$$

$$\boxed{\begin{pmatrix}0\\0\\1\end{pmatrix} \times \begin{pmatrix}x\\y\\z\end{pmatrix} = \begin{pmatrix}-y\\x\\0\end{pmatrix}}$$

$\hookrightarrow J_{O1} = \begin{pmatrix}0\\0\\1\end{pmatrix}$

$$J_{P2} = \begin{pmatrix}0\\0\\1\end{pmatrix} \times p_3^0 - \begin{pmatrix} a_1c_1 \\ a_1s_1 \\ 0 \end{pmatrix} = \begin{pmatrix}0\\0\\1\end{pmatrix} \times \begin{pmatrix} a_2c_2 + a_3c_3 \\ a_2s_2 + a_3s_3 \\ 0 \end{pmatrix} = \begin{pmatrix} -a_2s_2 - a_3s_3 \\ a_2c_2 + a_3c_3 \\ 0 \end{pmatrix}$$

$\hookrightarrow J_{O2} = \begin{pmatrix}0\\0\\1\end{pmatrix}$

$$\begin{bmatrix} c_i & -s_i & 0 & a_ic_i \\ s_i & c_i & 0 & a_is_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$J_{P_3} = ? \rightarrow R_2^0 = R_1 \cdot R_2 =$$

$$p_3^0 - p_2^0 = \begin{pmatrix} a_3c_3 \\ a_3s_3 \\ 0 \end{pmatrix}$$