Optimization, Robotics & Biomechanics (ORB)
Institute of Computer Engineering (ziti)
Heidelberg University

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

Robotics 1 (WS 2018/2019)

Exercise Sheet 1

Presentation during exercises in calendar week 45

Exercise 1.1 – Rolling sphere

A solid sphere (ball) rolls down a slope that is inclined towards the vertical by $\alpha = 60^\circ = \frac{\pi}{3}$. The experiment starts with the ball at rest, e.g. there is no initial velocity.

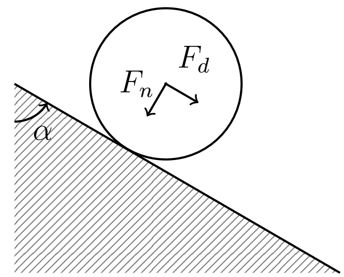

Figure 1: Sphere on slope

1. *Free fall*: First we assume the ball to be a point mass that falls freely under the influence of the gravitational acceleration $g = 9.81\,\frac{\text{m}}{\text{s}^2} = 9.81\,\frac{\text{N}}{\text{kg}}$ without the slope or any friction.
   (a) Calculate the end velocity after falling 10 m using the Conservation of Energy.
   (b) How long does the mass fall?
2. *Frictionless slinding*: The ball is still assumed to be a point mass but slides without friction down the slope.
   (a) Determine the force F_d that drives the mass down the slope using trigonometry. Resolve the gravitational force F_g into a normal force F_n that is orthogonal to the slope and a downhill force F_d that acts parallel to the slope.
   (b) Calculate the end velocity after sliding down on the slope for 10 m.
   (c) How long does the mass slide?
3. *Rolling sphere:* The ball is solid with a constant density rolling down the slope. There is no friction and the ball does not slide.
   (a) Deduce the rolling condition: Find the relation between the rotation angle φ and the translation along the slope s. Note that the sphere performs a full rotation while it rolls the length of a full circumference. Determine the time derivative of the rolling condition to get the relation between ω and v.
   (b) Calculate the end velocity for the sphere after rolling down on the slope for 10 m. Use the Conservation of Energy.

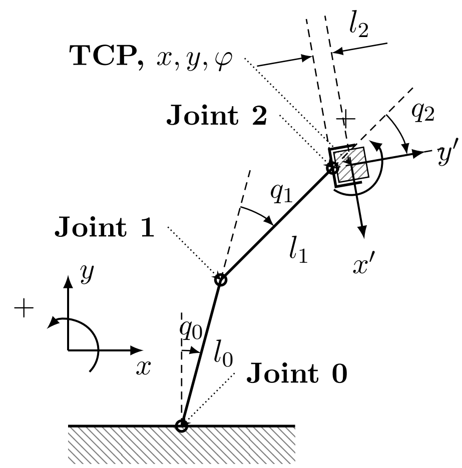

Figure 2: Scheme of extended model

Exercise 1.2 – Create your own lua-model

In the following session, we will create a three limb model as shown in Figure 2. The model should consist of 3 segments with 3 planar revolute degrees of freedom and a free-moving cube (linear motion about x, y and rotation about φ). The basis is fixed to the ground.

- Write a lua-model for the three limb manipulator with segment lengths $l_0 = 1.0$, $l_1 = 1.0$ and $l_2 = 0.15$.
- Write the `COLUMNS` section for an animation file, such that they are ordered in the following way: `| time | x | y |` φ `| q0 | q1 | q2 |`
- Test your model using the provided `"animation.csv"`.

Hint 1: Revisit the lua-model and animation file from exercise 0.2 on the previous sheet.

Hint 2: You can use `DATA_FROM:animation.csv` instead of `DATA:` inside the animation file to import data from an additional file.

Exercise 1.3 – Inverse Kinematics

Imagine the following case: the simple planar scheme (Figure 2) models the kinematics of a real planar industrial manipulator and the cube is a tool block for a specific production task. Now, the block should move along a given trajectory $TCP(t) = [x(t), y(t), \varphi(t)]^T$ to avoid collisions during a manufacturing process. This is a path planning and optimization challenge devoted to your cooperation partner. The question is now, what are the trajectories of $q(t) = [q_0(t), q_1(t), q_2(t)]^T$ dependent on the final motion of the block?

- Express the trajectories $q(t)$ analytically in terms of the cubes trajectory $TCP(t)$. **Hint:** Calculate the position of joint 2 depending on $TCP(t)$ first. Then calculate the joint angles using trigonometry.
- **(optional)** Are there constraints for $x(t)$, $y(t)$ and $\varphi(t)$?
- Use the provided `C++` skeleton `"inverse_kinematic/main.cpp"` and implement your analytic formulation such that `q = invserseKinematic(TCP)`. Run the code and check the console output to see if your solution is correct.
- **(optional)** The `C++` code exports a new `animation.csv`. Visualize the trajectory using the lua-model you created in the previous exercise.

Mechanics Cheat Sheet

Position	s	$1\,\mathrm{m}$	Angle	φ	$1\,\mathrm{rad}$
Mass	m	$1\,\mathrm{kg}$	Inertia	I	$1\,\mathrm{kg}\cdot\mathrm{m}^2$
Velocity	$v = \dot{s} = \frac{d}{dt}s$		Angular Velocity	$\omega = \dot{\varphi} = \frac{d}{dt}\varphi$	
Acceleration	$a = \ddot{s} = \frac{d}{dt^2}s$		Angular Acceleration	$\dot{\omega} = \ddot{\varphi} = \frac{d}{dt^2}\varphi$	
Eq. of Motion	$s = \frac{1}{2}at^2 + v_0t + s_0$		Eq. of Motion	$\varphi = \frac{1}{2}\dot{\omega}t^2 + \omega_0t + \varphi_0$	
Impulse	$p = mv$		Angular Momentum	$H = I\omega$	
Kinetic Energy	$E_{\mathrm{kin,trl}} = \frac{1}{2}mv^2$		Kinetic Energy	$E_{\mathrm{kin,rot}} = \frac{1}{2}I\omega^2$	
Potential Energy	$E_{\mathrm{pot}} = mgh$				