Control of Industrial and Mobile Robots

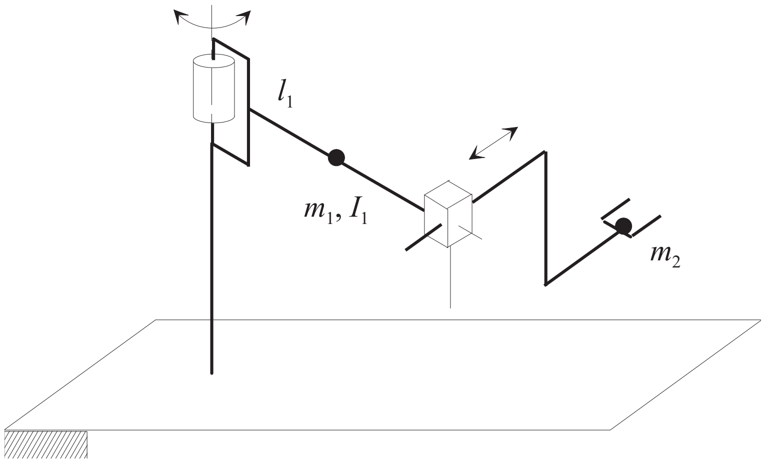
Prof. Rocco, Bascetta

January 15, 2020

SOLUTION

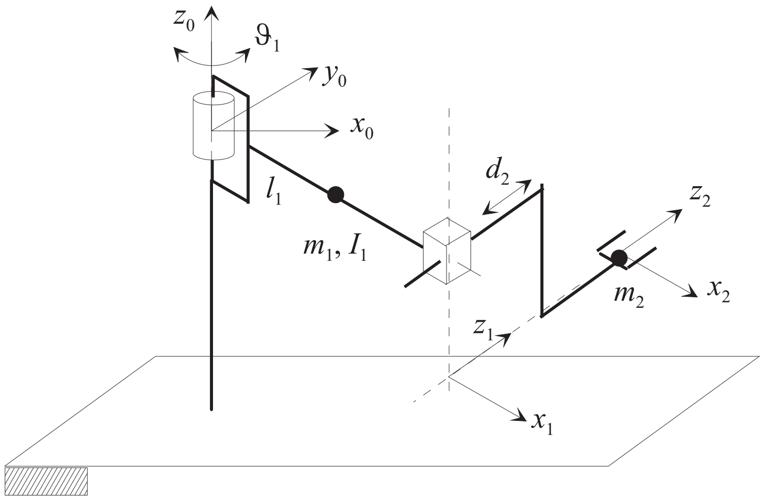
EXERCISE 1

1. Consider the manipulator sketched in the picture, where the mass of the second link is assumed to be concentrated at the end-effector:



Find the expression of the inertia matrix $\mathbf{B}(\mathbf{q})$ of the manipulator.

Denavit-Hartenberg frames can be defined as sketched in this picture:



Computations of the Jacobians:

Link 1

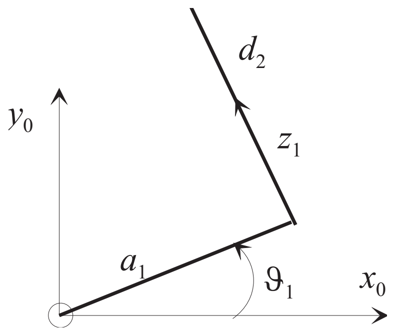
$$\mathbf{J}_P^{(l_1)} = \begin{bmatrix} \mathbf{j}_{P_1}^{(l_1)} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{z}_0 \times (\mathbf{p}_{l_1} - \mathbf{p}_0) & \mathbf{0} \end{bmatrix} = \begin{bmatrix} -l_1 s_1 & 0 \\ l_1 c_1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{J}_O^{(l_1)} = \begin{bmatrix} \mathbf{j}_{O_1}^{(l_1)} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{z}_0 & \mathbf{0} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$$

Link 2

$$\mathbf{J}_P^{(l_2)} = \begin{bmatrix} \mathbf{j}_{P_1}^{(l_2)} & \mathbf{j}_{P_2}^{(l_2)} \end{bmatrix} = \begin{bmatrix} \mathbf{z}_0 \times (\mathbf{p}_{l_2} - \mathbf{p}_0) & \mathbf{z}_1 \end{bmatrix} = \begin{bmatrix} -a_1 s_1 - d_2 c_1 & -s_1 \\ a_1 c_1 - d_2 s_1 & c_1 \\ 1 & 0 \end{bmatrix}$$

For the above computations, we can make reference to the following picture:



and to the following auxiliary vectors:

$$\mathbf{p}_{l_1} = \begin{bmatrix} l_1 c_1 \\ l_1 s_1 \\ 0 \end{bmatrix}, \mathbf{p}_{l_2} = \begin{bmatrix} a_1 c_1 - d_2 s_1 \\ a_1 s_1 + d_2 c_1 \\ \star \end{bmatrix}, \mathbf{p}_1 = \begin{bmatrix} a_1 c_1 \\ a_1 s_1 \\ 0 \end{bmatrix}, \mathbf{z}_1 = \begin{bmatrix} -s_1 \\ c_1 \\ 0 \end{bmatrix}$$

The inertia matrix can be computed now:

$$\begin{aligned} \mathbf{B}(\mathbf{q}) &= m_1 \mathbf{J}_P^{(l_1)^T} \mathbf{J}_P^{(l_1)} + I_1 \mathbf{J}_O^{(l_1)^T} \mathbf{J}_O^{(l_1)} + m_2 \mathbf{J}_P^{(l_2)^T} \mathbf{J}_P^{(l_2)} + \\ &= m_1 \begin{bmatrix} l_1^2 & 0 \\ 0 & 0 \end{bmatrix} + I_1 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + m_2 \begin{bmatrix} a_1^2 + d_2^2 & a_1 \\ a_1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} b_{11} & b_{12} \\ b_{12} & b_{22} \end{bmatrix} \end{aligned}$$

where:

$$\begin{aligned} b_{11} &= m_1 l_1^2 + I_1 + m_2 \left(a_1^2 + d_2^2 \right) \\ b_{12} &= m_2 a_1 \\ b_{22} &= m_2 \end{aligned}$$

2. Compute the matrix $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ of the Coriolis and centrifugal terms[1] for this manipulator.

   The only derivative in the Christoffel symbols which is different from zero is:

$$\frac{\partial b_{11}}{\partial q_2} = 2m_2 d_2$$

   therefore

$$\begin{array}{cc} c_{111} = 0 & c_{211} = -\frac{1}{2}\frac{\partial b_{11}}{\partial q_2} = -m_2 d_2 \\ c_{112} = c_{121} = \frac{1}{2}\frac{\partial b_{11}}{\partial q_2} = m_2 d_2 & c_{212} = c_{221} = 0 \\ c_{112} = 0 & c_{222} = 0 \end{array}$$

   The matrix of the Coriolis and centrifugal terms is thus:

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

   where:

$$\begin{aligned} c_{11} &= c_{111}\dot{q}_1 + c_{112}\dot{q}_2 = m_2 d_2 \dot{d}_2 \\ c_{12} &= c_{121}\dot{q}_1 + c_{122}\dot{q}_2 = m_2 d_2 \dot{\vartheta}_1 \\ c_{21} &= c_{211}\dot{q}_1 + c_{212}\dot{q}_2 = -m_2 d_2 \dot{\vartheta}_1 \\ c_{22} &= c_{221}\dot{q}_1 + c_{222}\dot{q}_2 = 0 \end{aligned}$$

3. Check that matrix $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{B}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is skew symmetric.

   We have that:

$$\begin{aligned} \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{B}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) &= \begin{bmatrix} 2m_2 d_2 \dot{d}_2 & 0 \\ 0 & 0 \end{bmatrix} - 2\begin{bmatrix} m_2 d_2 \dot{d}_2 & m_2 d_2 \dot{\vartheta}_1 \\ -m_2 d_2 \dot{\vartheta}_1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -2m_2 d_2 \dot{\vartheta}_1 \\ 2m_2 d_2 \dot{\vartheta}_1 & 0 \end{bmatrix} \end{aligned}$$

   which is a skew-symmetric matrix.

4. Cite one case in robotics where the property that matrix $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}})$ is skew symmetric is used.

   This property is used for example in the proof of stability of the PD + gravity compensation controller.

EXERCISE 2

[1] The general expression of the Christoffel symbols is $c_{ijk} = \frac{1}{2}\left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)$

1. Consider the generation of a position trajectory in the Cartesian space. Select as an initial point $\mathbf{p}_i = \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}$ and as a final point $\mathbf{p}_f = \begin{bmatrix} 0 \\ 1 \\ 5 \end{bmatrix}$. Write the expression of a segment connecting the initial and the final points, parameterized with the natural coordinate.

   The general expression of the segment is:

   $$\mathbf{p}(s) = \mathbf{p}_i + \frac{s}{\|\mathbf{p}_f - \mathbf{p}_i\|}(\mathbf{p}_f - \mathbf{p}_i)$$

   Since:

   $$\mathbf{p}_f - \mathbf{p}_i = \begin{bmatrix} -3 \\ 0 \\ 4 \end{bmatrix}, \|\mathbf{p}_f - \mathbf{p}_i\| = 5$$

   we have:

   $$\mathbf{p}(s) = \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} + \frac{s}{5}\begin{bmatrix} -3 \\ 0 \\ 4 \end{bmatrix}$$

2. Assume a travel time $T = 2s$. Design a trajectory, which covers the path determined in the previous step, using a cubic dependence on time.

   We need to find the coefficients of the polynomial:

   $$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

   with the boundary conditions:

   $$\begin{matrix} s(0) = 0 & s(2) = 5 \\ \dot{s}(0) = 0 & \dot{s}(2) = 0 \end{matrix}$$

   From the conditions at $t = 0$, we easily obtain $a_0 = a_1 = 0$. From the conditions at $t = 2$ we obtain the linear system:

   $$\begin{aligned} 4a_2 + 8a_3 &= 5 \\ 4a_2 + 12a_3 &= 0 \end{aligned}$$

   and then:

   $$\begin{aligned} a_2 &= 15/4 \\ a_3 &= -5/4 \end{aligned}$$

   Then the expression of the cubic polynomial is:

   $$s(t) = \frac{15}{4}t^2 - \frac{5}{4}t^3$$

3. Compute the maximum linear velocity of the end effector along the trajectory designed in the previous step. Check whether this maximum value exceeds a maximum admissible velocity of 2 m/s. In case, explain (without going through the computations) how you would modify the trajectory generation.

   The maximum linear velocity corresponds to the maximum value of the derivative of function $s(t)$:

   $$\|\dot{\mathbf{p}}\|_{\max} = \dot{s}_{\max} = \dot{s}(1) = \frac{30}{4} - \frac{15}{4} = 3.75$$

   Since the obtained maximum velocity is higher than the maximum admissible value, the time law has to be scaled, by suitably extending the positioning time.

4. Suppose that the manipulator is kinematically redundant for the task of end effector positioning. Write the expression of the inverse kinematics based on the *weighted* pseudo-inverse matrix and explain what is the optimization problem solved with this approach. Also explain what might be the reason to use the weighted pseudo-inverse instead of the standard pseudo-inverse matrix.

   The expression of the inverse kinematics is:

   $$\dot{\mathbf{q}} = \mathbf{J}_W^{\sharp}\dot{\mathbf{p}}$$

   where $\mathbf{J}_W^{\sharp}$ is the weighted pseudo-inverse matrix of the Jacobian, defined as:

   $$\mathbf{J}_W^{\sharp} = \mathbf{W}^{-1}\mathbf{J}^T\left(\mathbf{J}\mathbf{W}^{-1}\mathbf{J}\right)^{-1}$$

   This solution solves an optimization problem, where the cost function to be minimized is:

   $$g(\dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^T\mathbf{W}\dot{\mathbf{q}}$$

   subject to the constraint:

   $$\dot{\mathbf{p}} - \mathbf{J}\dot{\mathbf{q}} = 0$$

   If $\mathbf{W}$ is diagonal, it can be used to relatively weigh the joint velocities (a large W_i corresponds to a small $\dot{q}_i$).

EXERCISE 3

1. Write and explain the pseudocode of the algorithm to construct the probabilistic roadmap used by PRM.

   The PRM algorithm to construct the roadmap follows:

$V \leftarrow \emptyset$;
$E \leftarrow \emptyset$;
for $i = 1, \ldots, N$ **do**
  $\mathbf{q}_{rand} \leftarrow SampleFree_i$;
  $U \leftarrow Near\left(G, \mathbf{q}_{rand}, r\right)$;
  $V \leftarrow V \cup \{\mathbf{q}_{rand}\}$;
  foreach $\mathbf{u} \in U$ *in order of increasing* $\|\mathbf{u} - \mathbf{q}_{rand}\|$ **do**
    if $\mathbf{q}_{rand}$ *and* $\mathbf{u}$ *are not in the same connected component of* G **then**
      if $CollisionFree\left(\mathbf{q}_{rand}, \mathbf{u}\right)$ **then**
        $E \leftarrow E \cup \{(\mathbf{q}_{rand}, \mathbf{u})\}$;
      end
    end
  end
end
return $G = (V, E)$

N vertex are sampled from the free space, then for each vertex $\mathbf{q}_{rand}$ the set of near nodes, i.e., the set of nodes in a ball of radius r centred in $\mathbf{q}_{rand}$, is computed and all the collision free connections between $\mathbf{q}_{rand}$ and the nodes in the near node set that are not in the same connected component are generated.

2. Explain how the previous algorithm has to be modified in order to obtain sPRM and PRM* algorithms.

The only difference between sPRM and PRM is that sPRM connects all the nodes in the near node set, without checking if they are in the same connected component.
The sPRM algorithm to construct the roadmap follows:

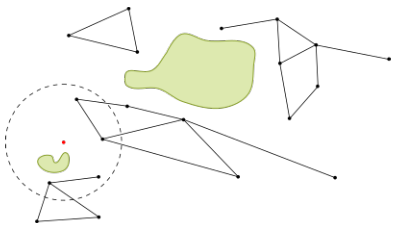
$V \leftarrow \{\mathbf{q}_{init}\} \cup \{SampleFree_i, i = 1, \ldots, N\}$;
$E \leftarrow \emptyset$;
foreach $\mathbf{v} \in V$ **do**
  $U \leftarrow Near\left(G, \mathbf{v}, r\right) \setminus \{\mathbf{v}\}$;
  foreach $\mathbf{u} \in U$ **do**
    if $CollisionFree\left(\mathbf{v}, \mathbf{u}\right)$ **then**
      $E \leftarrow E \cup \{(\mathbf{v}, \mathbf{u})\}$;
    end
  end
end
return $G = (V, E)$

The only difference between PRM and PRM* is in the way the near set is computed. In PRM* the radius of the near neighbourhood is related to the number of sampled nodes, i.e.,

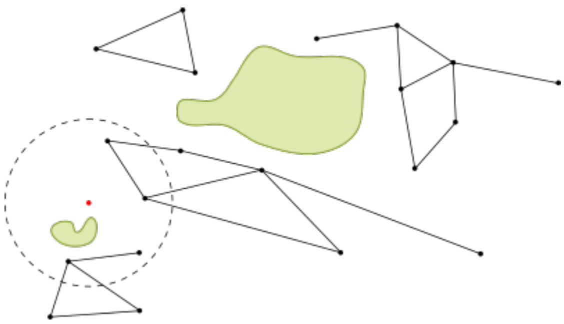
$$U \leftarrow Near\left(G, \mathbf{q}_{rand}, \gamma_{PRM}\left(\log\left(N\right)/N\right)^{1/d}\right) \setminus \{\mathbf{q}_{rand}\}$$

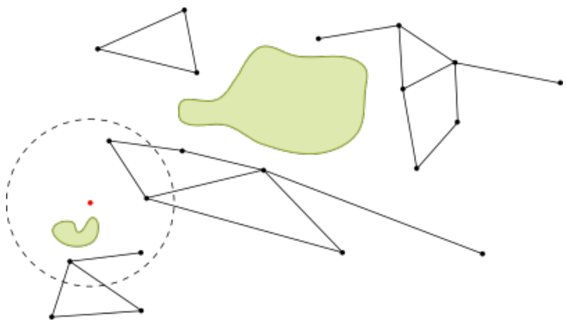
where d is the dimension of the configuration space, and γ_{PRM} is a suitable constant.

3. After the execution of some iterations of PRM algorithm we have the situation depicted below

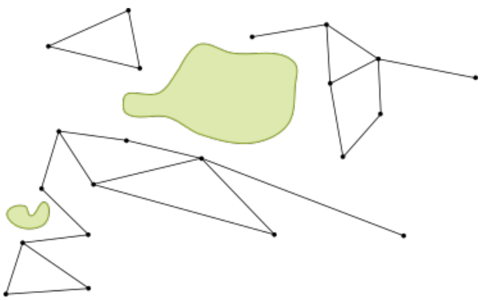


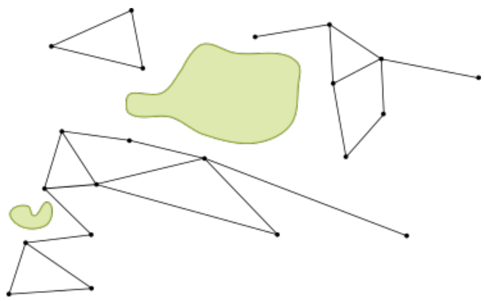
where the black dots are the nodes sampled in previous iterations, the red dot is the node sampled in the actual iteration, the green blobs are obstacles and the black dashed circle is the neighbourhood considered to defined the Near set.
Draw in the left picture the result of an iteration of PRM and in the right picture the result of an iteration of sPRM.





The result of an iteration of PRM (left picture) and of sPRM (right picture) is shown in the pictures below





4. What is the main difference between PRM (or sPRM or PRM*) and RRT (or RRT*)?

The main difference between PRM/sPRM/PRM* and RRT/RRT* is that the first are multiple query algorithms, while the second are single query algorithms.
In fact, PRM/sPRM/PRM* consist of two phases, the roadmap construction and the query phase, and on the same roadmap multiple queries, from different starting to different goal positions, can be performed.

With RRT/RRT*, instead, each execution of the algorithm is related to a specific starting/goal query.

EXERCISE 4

1. What is the *canonical simplified model for nonholonomic mobile robots*? Why is it important in the context of designing a controller for a nonholonomic robot?

   The canonical simplified model is a way to unify in a single model the unicycle, differential drive and bicycle kinematic models, in order to apply the same controller design procedure to the different kinematic models. The canonical simplified model has the following expression

$$\dot{x} = v\cos\theta$$
$$\dot{y} = v\sin\theta$$
$$\dot{\theta} = \omega$$

2. Show how the unicycle, differential drive, and rear-wheel drive bicycle kinematic models can be made equivalent to the canonical model.

   The canonical model is the model of a unicycle robot.
   The differential drive model can be reduced to the unicycle model applying the following transformation

$$\omega_R = \frac{v + \omega d/2}{r} \qquad \omega_L = \frac{v - \omega d/2}{r}$$

   where ω_R and ω_L are the right and left wheel velocities, r is the wheel radius, and d is the distance between left and right wheel contact point.
   For the rear-wheel drive bicycle we need to assume that the steering rate limit is so high that the steering angle can be changed instantaneously, then the bicycle model can be reduced to the unicycle model applying the following transformation

$$v = v \qquad \phi = \arctan\left(\frac{\omega \ell}{v}\right)$$

   where ϕ is the steering angle and ℓ the length of the bicycle.

3. Consider a bicycle kinematic model without reverse. Show how the actuation constraints $0 \le v \le v_M$ and $-\phi_M \le \phi \le \phi_M$ can be rewritten in terms of the canonical model input variables.

   Considering the transformation for the bicycle

$$v = v \qquad \phi = \arctan\left(\frac{\omega \ell}{v}\right)$$

   and assuming that $\arctan(x) \approx x$, we have

$$0 \le v \le v_M \qquad -\phi_M \le \frac{\omega \ell}{v} \le \phi_M$$

The second constraint can be rewritten as two separate constraints as follows

$$\omega \leq \frac{\phi_M}{\ell} v \qquad \omega \geq -\frac{\phi_M}{\ell} v$$

4. Considering a reference point P on the chassis of the robot, derive a control law for the canonical model that allows to reduce it to two independent double integrators.

Assuming (x, y, θ) is the pose of the robot and x_r, y_r are the coordinates of point P in the robot reference frame, the absolute coordinates of this point are given by

$$\begin{bmatrix} x_P \\ y_P \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_r \\ y_r \end{bmatrix}$$

Differentiating this relation we obtain

$$\begin{bmatrix} \dot{x}_P \\ \dot{y}_P \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} + \dot{\theta} \begin{bmatrix} -\sin\theta & -\cos\theta \\ \cos\theta & -\sin\theta \end{bmatrix} \begin{bmatrix} x_r \\ y_r \end{bmatrix}$$

Substituting then the canonical model equations and solving for v and ω

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{1}{x_r} \begin{bmatrix} x_r \cos\theta - y_r \sin\theta & x_r \sin\theta + y_r \cos\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \dot{x}_P \\ \dot{y}_P \end{bmatrix}$$