

Autonomous and Mobile Robotics

Solution of Midterm Class Test, 2016/2017

Solution of Problem 1

1. The augmented configuration vector is $\mathbf{q} = (x \ y \ \theta \ \phi)^T$, with the usual meaning for x, y, θ . The configuration space is $\mathbb{R}^2 \times SO(2) \times SO(2)$ and has dimension $n = 4$.
2. Since the wheel rotation ϕ is a generalized coordinate, the *driving* angular velocity $\omega_\phi = \dot{\phi}$ can be directly taken as the first velocity input in place of v . The latter will then be obtained as $v = R\omega_\phi$, where R is the wheel radius. As usual, the second velocity input will be the *steering* angular velocity $\omega_\theta = \dot{\theta}$. This leads to the following kinematic model:

$$\begin{aligned} \dot{x} &= R\omega_\phi \cos \theta \\ \dot{y} &= R\omega_\phi \sin \theta \\ \dot{\theta} &= \omega_\theta \\ \dot{\phi} &= \omega_\phi \end{aligned} \quad \text{i.e.,} \quad \dot{\mathbf{q}} = \begin{pmatrix} R \cos \theta \\ R \sin \theta \\ 0 \\ 1 \end{pmatrix} \omega_\phi + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \omega_\theta = \mathbf{g}_1(\mathbf{q}) \omega_\phi + \mathbf{g}_2(\mathbf{q}) \omega_\theta$$

The same model could have been obtained by the classical approach, i.e., writing down the kinematic constraints and solving for the generalized velocities $\dot{\mathbf{q}}$. In this case, there are two such constraints: one is pure rolling, and the other is the relationship $v = R\omega_\phi$, which can be indeed rewritten as a kinematic constraint as follows

$$\sqrt{\dot{x}^2 + \dot{y}^2} - R\dot{\phi} = 0 \quad (*)$$

However, this constraint is not linear in $\dot{\mathbf{q}}$; therefore, the usual procedure (find a basis for the null space of the constraint matrix) cannot be applied. One should first solve the pure rolling constraint (which is linear) and then use (*) in the partial solution. This is equivalent to the direct augmentation procedure illustrated above.

As an alternative, one may note that $v = v(\cos^2 \theta + \sin^2 \theta) = \dot{x} \cos \theta + \dot{y} \sin \theta$, so that $v = R\omega_\phi$ can be rewritten as

$$\dot{x} \cos \theta + \dot{y} \sin \theta - R\dot{\phi} = 0 \quad (**)$$

which is linear in $\dot{\mathbf{q}}$. Correspondingly, the constraint matrix accounting for pure rolling and (**) becomes

$$\begin{pmatrix} \sin \theta & -\cos \theta & 0 & 0 \\ \cos \theta & \sin \theta & 0 & -R \end{pmatrix}$$

whose null space is spanned by the above vector fields \mathbf{g}_1 and \mathbf{g}_2 .

3. To study controllability, we use the accessibility rank condition. One easily obtains

$$[\mathbf{g}_1, \mathbf{g}_2] = \begin{pmatrix} R \sin \theta \\ -R \cos \theta \\ 0 \\ 0 \end{pmatrix} \triangleq \mathbf{g}_3 \quad [\mathbf{g}_1, \mathbf{g}_3] = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad [\mathbf{g}_2, \mathbf{g}_3] = \begin{pmatrix} R \cos \theta \\ R \sin \theta \\ 0 \\ 0 \end{pmatrix} \triangleq \mathbf{g}_4$$

Controllability is then proven by noting that

$$\text{rank}(\mathbf{g}_1 \mathbf{g}_2 \mathbf{g}_3 \mathbf{g}_4) = \text{rank}(\mathbf{g}_3 \mathbf{g}_4 \mathbf{g}_1 \mathbf{g}_2) = \text{rank} \begin{pmatrix} R \sin \theta & R \cos \theta & R \cos \theta & 0 \\ -R \cos \theta & R \sin \theta & R \sin \theta & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = 4$$

because the determinant of the last matrix is $-R^2$ (the determinant of a block triangular matrix is the product of the determinants of the block on the diagonal).

4. To move the robot from $\mathbf{q}_s = (x_s, y_s, \theta_s, \phi_s)$ to $\mathbf{q}_g = (x_g, y_g, \theta_g, \phi_g)$, first take care of x, y, θ as follows: (1) rotate the wheel around the vertical axis until sagittal axis goes through (x_g, y_g) ; (2) drive the contact point in a straight line from (x_s, y_s) to (x_g, y_g) ; and (3) rotate the wheel around the vertical axis so as to achieve the desired orientation θ_g . The final step is to change the wheel angle from ϕ_3 (its value at the end of step 3) to ϕ_g . This may be obtained by (4) moving the contact point along a circle, so that x, y, θ will go back to x_g, y_g, θ_g . In particular, the wheel angle variation along the full circle must match $\Delta_\phi = \phi_g - \phi_3$; denoting by r the circle radius, this requires $2\pi r = R \Delta_\phi$, from which we get $r = R \Delta_\phi / 2\pi$.

Solution of Problem 2

The (2,3) chained form is

$$\begin{aligned}z_1' &= \tilde{v}_1 \\z_2' &= \tilde{v}_2 \\z_3' &= z_2 \tilde{v}_1\end{aligned}$$

Note the use of the geometric version of the kinematic model (derivatives w.r.t. s and geometric inputs) because we are assigned a path planning problem. The problem may be solved by using either flat outputs or parameterized inputs; for a (2,3) chained form, the resulting path will be the same (see “Robotics: Modelling, Planning and Control”, Problem 11.12). Let us choose the first route because it provides the required configuration space path without integration.

The flat outputs are z_1, z_3 . Using the third and the first equation, the remaining state variable can be reconstructed as $z_2 = z_3'/z_1'$. We must therefore choose $z_1(s)$ and $z_3(s)$, $s \in [s_i, s_f]$, so as to satisfy their endpoint conditions

$$z_1(s_i) = z_3(s_i) = 0 \quad z_1(s_f) = z_3(s_f) = 1$$

as well as the boundary conditions on z_2

$$\frac{z_3'(s_i)}{z_1'(s_i)} = 0 \quad \frac{z_3'(s_f)}{z_1'(s_f)} = 1$$

Several choices are possible. For example, letting $s \in [0, 1]$, we can use a 1st-order polynomial for z_1 and a 3rd-order polynomial for z_3

$$\begin{aligned}z_1(s) &= a_1 s + b_1 \\z_3(s) &= a_2 s^3 + b_2 s^2 + c_2 s + d_2\end{aligned}$$

whose derivatives w.r.t. to s are

$$\begin{aligned}z_1'(s) &= a_1 \\z_3'(s) &= 3 a_2 s^2 + 2 b_2 s + c_2\end{aligned}$$

By imposing the previous conditions, one finds

$$\begin{aligned}z_1(s) &= s \\z_3(s) &= -s^3 + 2s^2\end{aligned}$$

and correspondingly

$$z_2(s) = \frac{z_3'(s)}{z_1'(s)} = -3s^2 + 4s$$

Solution of Problem 3

1. The output variable is x . According to the unicycle equations, we have directly

$$\dot{x} = v \cos \theta$$

This simple, scalar input-output map can be linearized by using the input transformation $v = u/\cos \theta$, where u is the new input. This leads to

$$\dot{x} = u$$

and therefore it is sufficient to set $u = k_x(x_d - x)$ (i.e., $v = k_x(x_d - x)/\cos \theta$) to drive exponentially the output to x_d , provided that $k_x > 0$.

So far, the steering velocity ω is free. We can use it to keep the robot parallel to the corridor, so that it will not collide with the lateral walls. Since we have

$$\dot{\theta} = \omega$$

the control law $\omega = -k_\theta \theta$, with $k_\theta > 0$, will drive the robot orientation θ exponentially to zero. Assuming¹ that $\theta(0) < \pi/2$, this will also guarantee that $\theta < \pi/2$ at all times, so that the input transformation for v is never affected by the potential singularity.

2. To implement the previous controller, we need real-time estimates of x and θ , but not of y . Also, the two sensor measurements (θ and d) do not depend on y . Therefore, we can restrict our attention to first and third equations of the unicycle, and build a two-dimensional filter.

Using, e.g., Euler integration, the noise-free discrete-time model (process dynamics) is easily obtained as

$$\begin{aligned}x_{k+1} &= x_k + v_k T_s \cos \theta_k \\ \theta_{k+1} &= \theta_k + \omega_k T_s\end{aligned}$$

where T_s is the sampling interval. As usual, the discrete-time velocity inputs v_k and ω_k can be reconstructed from wheel encoder readings (see, e.g., the formulas in the AMR slides on ‘Odometric Localization’).

The noise-free output equation (measurement model) is

$$h_k = \begin{pmatrix} \theta_k \\ \ell - x_k \end{pmatrix}$$

where we have used the fact that $d = \ell - x$ thanks to the placement of the world frame.

The rest of the solution is straightforward: linearize the process dynamics (note that the measurement model is already linear) and then derive the EKF equations.

¹If necessary, this condition can always be achieved by a preliminary rotation on the spot.