Autonomous and Mobile Robotics
Solution of Midterm Class Test, 2021/2022

Solution of Problem 1

(a) The geometric constraint on the robot motion is expressed as

$$x^2 + y^2 + z^2 = r^2.$$

By differentiating w.r.t. time, one obtains the corresponding kinematic constraint, which is Pfaffian:

$$x\,\dot{x} + y\,\dot{y} + z\,\dot{z} = 0 \qquad \text{or} \qquad \boldsymbol{a}^T\dot{\boldsymbol{q}} = 0 \quad \text{with} \quad \boldsymbol{a}^T = (x \;\; y \;\; z)^T$$

(b) From a global viewpoint, the configuration $\boldsymbol{q}$ of the robot must belong to a 2-dimensional subset of $I\!R^3$, i.e., the sphere (a manifold). From a local viewpoint, its generalized velocity $\dot{\boldsymbol{q}}$ is limited to a 2-dimensional subspace of $I\!R^3$, i.e., the null space of $\boldsymbol{a}^T(\boldsymbol{q})$, which is the tangent plane at $\boldsymbol{q}$. Therefore, both global and local mobility are restricted.

(c) A basis for $\mathcal{N}(\boldsymbol{a}^T(\boldsymbol{q}))$ consists of the following vector fields (other choices are possible)

$$\boldsymbol{g}_1(\boldsymbol{q}) = \begin{pmatrix} y \\ -x \\ 0 \end{pmatrix} \qquad \boldsymbol{g}_2(\boldsymbol{q}) = \begin{pmatrix} z \\ 0 \\ -x \end{pmatrix}$$

and thus the kinematic model is

$$\dot{\boldsymbol{q}} = \boldsymbol{g}_1(\boldsymbol{q})u_1 + \boldsymbol{g}_2(\boldsymbol{q})u_2 \qquad \text{or} \qquad \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} y \\ -x \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} z \\ 0 \\ -x \end{pmatrix} u_2$$

One easily finds

$$\boldsymbol{g}_3(\boldsymbol{q}) = [\boldsymbol{g}_1, \boldsymbol{g}_2]\,(\boldsymbol{q}) = \begin{pmatrix} 0 \\ z \\ -y \end{pmatrix}$$

and being

$$\text{rank}\,(\boldsymbol{g}_1(\boldsymbol{q}) \;\; \boldsymbol{g}_2(\boldsymbol{q}) \;\; \boldsymbol{g}_3(\boldsymbol{q})) = \text{rank} \begin{pmatrix} y & z & 0 \\ -x & 0 & z \\ 0 & -x & -y \end{pmatrix} = 2$$

the robot is not controllable, as expected.

Solution of Problem 2

The $(2,3)$ chained form is

$$
\begin{aligned}
\dot{z}_1 &= v_1 \\
\dot{z}_2 &= v_2 \\
z_3 &= z_2\, v_1
\end{aligned}
$$

The first two equations are simple integrators. Therefore, it is trivial to design v_1 and v_2 so as to drive z_1 and z_2 to their destination in finite time. After this, one may use the sinusoidal controls suggested by the problem: choosing appropriately their parameters, also z_3 will go to its destination, while z_1 and z_2 will go back to where they started from, i.e., their destination.

(a) A possible algorithm is this:

Phase 1 Set $v_1 = \mathrm{sign}(z_{1,f} - z_{1,i})$ and $v_2 = \mathrm{sign}(z_{2,f} - z_{2,i})$ for $t \in [0, T_1]$, with $\mathrm{sign}(0) = 0$ and $T_1 = \max(|z_{1,f} - z_{1,i}|, |z_{2,f} - z_{2,i}|)$. At the end of this phase, z_1 and z_2 will be at $z_{1,f}$ and $z_{2,f}$, respectively, while z_3 will be at a certain value[1] $z_3(T_1)$.

Phase 2 Set $v_1 = a \sin \omega(t - T_1)$ and $v_2 = b \cos \omega(t - T_1)$ for $t \in [T_1, T_1 + 2\pi/\omega]$, with the constraint $a\, b\, \pi/\omega^2 = z_{3,f} - z_3(T_1)$. At the end of this phase, z_1 and z_2 will return to $z_{1,f}$ and $z_{2,f}$, respectively, while z_3 will be at $z_{3,f}$.

(b) Phase 1 contains no parameters. In Phase 2 we have three parameters (a, b and ω) which must be chosen so as to satisfy one constraint ($ab\pi/\omega^2 = z_{3,f} - z_3(T_1)$). Parameters a and b will be the amplitudes of the Phase 2 inputs, and therefore they should be chosen keeping into account existing bounds on the available inputs. These parameters also affect the trajectory: since z_1 and z_2 during Phase 2 trace an ellipse[2], a and b will determine the shape of this ellipse. In particular, $a > b$ ($a < b$) will give an ellipse elongated along z_1 (z_2). As for ω, it will determine the duration of Phase 2 and therefore the total time needed to go from $\boldsymbol{z}_i$ to $\boldsymbol{z}_f$.

(c) Being $\boldsymbol{z}_i = (0,0,0)$ and $\boldsymbol{z}_f = (1,1,1)$, the duration of Phase 1 is $T_1 = \max(1,1) = 1$. Assuming that ω is chosen arbitrarily, the dotal duration of the motion will be $1 + 2\pi/\omega$. The trajectory on the z_1, z_2 plane will be a line from $(0,0)$ to $(1,1)$ in Phase 1, followed by an ellipse in Phase 2.

[1] It is easy to compute $z_3(T_1)$ in closed form by integrating the chained form equations under the Phase 1 inputs.

[2] This intuitive fact can be proven by integrating the first two equations of the chained form under the Phase 2 inputs and verifying that the solutions z_1 and z_2 are the parametric equations of an ellipse.

Solution of Problem 3

Let (x_i, y_i, θ_i) be the configuration vector of each unicycle $(i = 1, 2)$. The coordinates of each barycenter are

$$x_{ci} = x_i + d\,\cos\theta_i \tag{1}$$
$$y_{ci} = y_i + d\,\sin\theta_i \tag{2}$$

while the barycenter of the team (masses are identical) is located at

$$x_b = \frac{x_{c1} + x_{c2}}{2} \tag{3}$$
$$y_b = \frac{y_{c1} + y_{c2}}{2} \tag{4}$$

(a) Input-output linearization is the most convenient approach. The output is the position of the team barycenter, while the inputs are the driving and steering velocities of each robot. Using the kinematic model of the two unicycles, one easily obtains

$$\dot{x}_b = \frac{v_1\cos\theta_1 - d\,\omega_1\sin\theta_1 + v_2\cos\theta_2 - d\,\omega_2\sin\theta_2}{2}$$
$$\dot{y}_b = \frac{v_1\sin\theta_1 + d\,\omega_1\cos\theta_1 + v_2\sin\theta_2 + d\,\omega_2\cos\theta_2}{2}$$

that is

$$\begin{pmatrix} \dot{x}_b \\ \dot{y}_b \end{pmatrix} = \frac{1}{2}\begin{pmatrix} \cos\theta_1 & -d\sin\theta_1 & \cos\theta_2 & -d\sin\theta_2 \\ \sin\theta_1 & d\cos\theta_1 & \sin\theta_2 & d\cos\theta_2 \end{pmatrix}\begin{pmatrix} v_1 \\ \omega_1 \\ v_2 \\ \omega_2 \end{pmatrix} = \frac{1}{2}\boldsymbol{T}(\theta_1, \theta_2)\,\boldsymbol{u} \tag{5}$$

where $\boldsymbol{T}$ is the $2{\times}4$ decoupling matrix and $\boldsymbol{u}$ is the 4-dimensional vector collecting all velocity inputs. Matrix $\boldsymbol{T}$ has clearly rank 2 if $d \neq 0$. Under this assumption, we may use the linearizing feedback[3]

$$\boldsymbol{u} = 2\,\boldsymbol{T}^{\#}(\theta_1, \theta_2)\,\boldsymbol{a} \tag{6}$$

where $\boldsymbol{T}^{\#} = \boldsymbol{T}^T(\boldsymbol{T}\boldsymbol{T}^T)^{-1}$ is the $4{\times}2$ pseudoinverse of $\boldsymbol{T}$ and the 2-dimensional vector $\boldsymbol{a} = (a_1, a_2)$ represents the new inputs. Using (6) in (5) we get a linear (simple integrators) input-output map:

$$\begin{pmatrix} \dot{x}_b \\ \dot{y}_b \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

Therefore, we can achieve global exponential tracking by simply setting

$$a_1 = \dot{x}_b^* + k_1(x_b^* - x_b) \tag{7}$$
$$a_2 = \dot{y}_b^* + k_2(y_b^* - y_b) \tag{8}$$

with $k_1, k_2 > 0$. The actual velocity inputs for the unicycle are obtained plugging (7–8) in (6).

[3]Note that (6) provides the least-squares linearizing feedback, but other choices are possible. For example, we one may add a null-space term to optimize (locally) a certain cost function. This null-space term will have no effect on the output tracking.

(b) The variables needed to implement the control law (6–7–8) are θ_1, θ_2 for the decoupling matrix and x_b, y_b for the output error; in turn, the computation of x_b, y_b via (3–4) and (1–2) requires the knowledge of x_1, y_1, θ_1 and x_2, y_2, θ_2. Therefore, our controller is centralized: it assumes knowledge of the states of all robots and generated control inputs for all robots.

(c) As just noted, to implement the proposed controller we need an estimate of the configuration of both unicycles. To this end, we can design a localization module based on the EKF.

A discrete-time motion model is readily written as

$$
\begin{aligned}
x_{1,k+1} &= x_{1,k} + T_s\, v_{1,k} \cos\theta_{1,k} \\
y_{1,k+1} &= y_{1,k} + T_s\, v_{1,k} \sin\theta_{1,k} \\
\theta_{1,k+1} &= \theta_{1,k} + T_s\, \omega_{1,k} \\
x_{2,k+1} &= x_{2,k} + T_s\, v_{2,k} \cos\theta_{2,k} \\
y_{2,k+1} &= y_{2,k} + T_s\, v_{2,k} \sin\theta_{2,k} \\
\theta_{2,k+1} &= \theta_{2,k} + T_s\, \omega_{2,k}
\end{aligned}
$$

where T_s is the sampling interval. This motion model is assumed to be perturbed by a white gaussian noise with zero mean and known covariance.

For the i-th robot ($i = 1, 2$), the measurement $\Delta\phi_i$ of the wheel encoder is used to reconstruct the actual value of $v_{i,k} = r\,\Delta\phi_i / T_s$, where r is the wheel radius. Since we have no measurement of the wheel orientation, the commanded value of $\omega_{i,k}$ will be used (i.e., the value computed via the control law).

The measurements are the distance between the robots and the distance of each robot from the landmark. This leads to the following measurement model:

$$
\boldsymbol{h}_k = \begin{pmatrix} \sqrt{(x_{2,k} - x_{1,k})^2 + (y_{2,k} - y_{1,k})^2} \\ \sqrt{(x_L - x_{1,k})^2 + (y_L - y_{1,k})^2} \\ \sqrt{(x_L - x_{2,k})^2 + (y_L - y_{2,k})^2} \end{pmatrix}
$$

where x_L, y_L are the known coordinates of the landmark. Also this model is assumed to be perturbed by a white gaussian noise with zero mean and known covariance.

Note that, although each robot provides its own measurement of the inter-robot distance, we are going to use only one of the two (if we duplicated the first component of $\boldsymbol{h}_k$, its Jacobian matrix $\boldsymbol{H}_k$ would not be full rank). Another possibility is to use the average of the distance measurements.

The rest of the solution is straightforward: linearize the motion and measurement models and then write the EKF equations. In the block scheme, the wheel encoders will be used in the prediction stage of the filter, while the range finders will be used in the correction stage.