Autonomous and Mobile Robotics
Solution of Midterm Class Test, 2025/2026

Solution of Problem 1

(a) The extended configuration is $\boldsymbol{q} = (x, y, \theta, \phi_R, \phi_L) \in I\!R^2 \times (SO(2))^3$. The associated kinematic model is obtained by writing the unicycle equations, expressing v and ω in terms of ω_R and ω_L, and adding as state variables ϕ_R and ϕ_L. We obtain:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_R \\ \dot{\phi}_L \end{pmatrix} = \begin{pmatrix} a\cos\theta \cdot (\omega_R + \omega_L) \\ a\sin\theta \cdot (\omega_R + \omega_L) \\ b(\omega_R - \omega_L) \\ \omega_R \\ \omega_L \end{pmatrix} = \begin{pmatrix} a\cos\theta \\ a\sin\theta \\ b \\ 1 \\ 0 \end{pmatrix} \omega_R + \begin{pmatrix} a\cos\theta \\ a\sin\theta \\ -b \\ 0 \\ 1 \end{pmatrix} \omega_L = \boldsymbol{g}_1(\boldsymbol{q})\omega_R + \boldsymbol{g}_2(\boldsymbol{q})\omega_L,$$

where $a = r/2$ and $b = r/d$, being r the wheel radius and d the distance between the wheels.

(b) To answer, we study the controllability of the above kinematic model. We easily find

$$[\boldsymbol{g}_1, \boldsymbol{g}_2](\boldsymbol{q}) = \begin{pmatrix} -2ab\sin\theta \\ 2ab\cos\theta \\ 0 \\ 0 \\ 0 \end{pmatrix} = \boldsymbol{g}_3(\boldsymbol{q}) \qquad [\boldsymbol{g}_1, \boldsymbol{g}_3](\boldsymbol{q}) = \begin{pmatrix} -2ab^2\cos\theta \\ -2ab^2\sin\theta \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad [\boldsymbol{g}_2, \boldsymbol{g}_3](\boldsymbol{q}) = \begin{pmatrix} 2ab^2\cos\theta \\ 2ab^2\sin\theta \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Since the last two Lie brackets are linearly dependent on one another, we can conclude that the Lie Algebra rank condition is violated (to be precise, this is true provided that third-order brackets do not increase the rank, which is readily verified to be the case).

(c) Ignoring ϕ_L means dropping the last equation of the above kinematic model, i.e., deleting the last element of the input vector fields $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$. Call $\tilde{\boldsymbol{g}}_1$, $\tilde{\boldsymbol{g}}_2$ the amputated versions of $\boldsymbol{g}_1$, $\boldsymbol{g}_2$. One immediately finds

$$[\tilde{\boldsymbol{g}}_1, \tilde{\boldsymbol{g}}_2](\boldsymbol{q}) = \begin{pmatrix} -2ab\sin\theta \\ 2ab\cos\theta \\ 0 \\ 0 \end{pmatrix} = \tilde{\boldsymbol{g}}_3(\boldsymbol{q}) \qquad [\tilde{\boldsymbol{g}}_1, \tilde{\boldsymbol{g}}_3](\boldsymbol{q}) = \begin{pmatrix} -2ab^2\cos\theta \\ -2ab^2\sin\theta \\ 0 \\ 0 \end{pmatrix} \qquad [\tilde{\boldsymbol{g}}_2, \tilde{\boldsymbol{g}}_3](\boldsymbol{q}) = \begin{pmatrix} 2ab^2\cos\theta \\ 2ab^2\sin\theta \\ 0 \\ 0 \end{pmatrix},$$

i.e., the amputated versions of the previous brackets. Since[1]

$$\text{rank}\,(\boldsymbol{g}_1 \;\; \boldsymbol{g}_2 \;\; \boldsymbol{g}_3 \;\; \boldsymbol{g}_4) = \text{rank} \begin{pmatrix} a\cos\theta & a\cos\theta & -2ab\sin\theta & -2ab^2\cos\theta \\ a\sin\theta & a\sin\theta & 2ab\cos\theta & -2ab^2\sin\theta \\ b & -b & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} = 4,$$

we can conclude that the amputated kinematic model is controllable. It is therefore possible to achieve any desired value of x, y, θ and ϕ_R; one simple way to do this (not an optimal one) would be to bring x, y and θ to the desired value using a unicycle maneuver, and then move the ground contact point of the right wheel along a circle of appropriate radius to achieve the desired displacement of ϕ_R while x, y and θ go back to their desired value (see the solution of Midterm Test 2016/17, Problem 1). Clearly, one may similarly ignore ϕ_R and achieve a desired value for ϕ_L.

[1] This can be easily proven by swapping the first and the third columns, and then the second and the fourth columns (the matrix determinant does not change). The new matrix is block triangular, and its determinant is $4a^2b^4$.

Solution of Problem 2

(a) According to the problem statement, the flat outputs are

$$\begin{aligned}\eta_1 &= x + \frac{\cos\theta}{d}\\ \eta_2 &= z + \frac{\sin\theta}{d}\end{aligned}$$

and the reconstruction formula for θ is

$$\theta = \text{atan2}\,(\ddot{\eta}_2 + g, \ddot{\eta}_1) + k\pi = \text{atan}\frac{\ddot{\eta}_2 + g}{\ddot{\eta}_1}. \tag{1}$$

The first two reconstruction formulas are obviously

$$\begin{aligned}x &= \eta_1 - \frac{\cos\theta}{d} = \eta_1 - \frac{\cos(\text{atan}\frac{\ddot{\eta}_2+g}{\ddot{\eta}_1})}{d}\\ z &= \eta_2 - \frac{\sin\theta}{d} = \eta_2 - \frac{\sin(\text{atan}\frac{\ddot{\eta}_2+g}{\ddot{\eta}_1})}{d}.\end{aligned}$$

Together with the formula for θ, these allow reconstruction of the whole configuration.

As for the inputs, note that first two equations of the dynamic model can be written as

$$\begin{pmatrix}\ddot{x}\\ \ddot{z}\end{pmatrix} = \begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}\begin{pmatrix}f_1\\ f_2\end{pmatrix} - \begin{pmatrix}0\\ g\end{pmatrix} = \boldsymbol{T}(\theta)\begin{pmatrix}f_1\\ f_2\end{pmatrix} - \begin{pmatrix}0\\ g\end{pmatrix}, \tag{2}$$

with $\boldsymbol{T}(\theta)$ nonsingular, from which we obtain

$$\begin{pmatrix}f_1\\ f_2\end{pmatrix} = \boldsymbol{T}^{-1}(\theta)\begin{pmatrix}\ddot{x}\\ \ddot{z}+g\end{pmatrix} = \boldsymbol{T}^{-1}(\text{atan}\frac{\ddot{\eta}_2+g}{\ddot{\eta}_1})\begin{pmatrix}\ddot{x}\\ \ddot{z}+g\end{pmatrix}. \tag{3}$$

We need $\ddot{x}$ and $\ddot{z}$, which we obtain from the formulas for x and z:

$$\begin{aligned}\ddot{x} &= \ddot{\eta}_1 + \frac{1}{d}(\dot{\theta}\cos\theta + \ddot{\theta}\sin\theta)\\ \ddot{z} &= \ddot{\eta}_2 - \frac{1}{d}(-\dot{\theta}\sin\theta + \ddot{\theta}\cos\theta).\end{aligned}$$

Here, one uses again (1) to express θ and its derivatives as a function of η_1, η_2. Plugging the resulting expressions of $\ddot{x}$ and $\ddot{z}$ in (3) finally provides the input reconstruction formulas.

Trajectory planning from (x_s, z_s, θ_s) to (x_g, z_g, θ_g) may be performed by interpolating the flat outputs with the appropriate boundary conditions. These would include the terminal conditions (start and goal values of η_1, η_2) plus the additional boundary conditions needed to enforce the initial and final values of θ. The latter may be written using the reconstruction formula for θ:

$$\text{atan}\frac{\ddot{\eta}_2(0)+g}{\ddot{\eta}_1(0)} = \theta_s \qquad \text{atan}\frac{\ddot{\eta}_2(T)+g}{\ddot{\eta}_1(T)} = \theta_g,$$

where we assumed $t \in [0, T]$. To allow independent interpolation for η_1 and η_2, these may be separated as follows:

$$\begin{aligned}\ddot{\eta}_1(0) &= k_s\cos\theta_s & \ddot{\eta}_1(T) &= k_g\cos\theta_g\\ \ddot{\eta}_2(0)+g &= k_s\sin\theta_s & \ddot{\eta}_2(T)+g &= k_g\sin\theta_g,\end{aligned}$$

where k_s and k_g are arbitrary constants. The resulting algorithm is the following:

1. Compute the start and goal values of the flat outputs, respectively $(\eta_{1,s}, \eta_{2,s})$ and $(\eta_{1,g}, \eta_{2,g})$.
2. Choose η_1 and η_2 as two cubic polynomials in t, for $t \in [0, T]$, and impose the terminal conditions (their start and goal values) as well as the additional boundary conditions on their initial and final accelerations.
3. From $\eta_1(t)$ and $\eta_2(t)$, reconstruct the state variables and the associated inputs using the reconstruction formulas.

Note that we have not imposed the generalized velocities $\dot{x}$, $\dot{z}$, $\dot{\theta}$ at the start and at the goal (they are not specified in the problem statement); this means that they will be a byproduct of the interpolation. Often, one wants the initial and final generalized velocities to be zero (*rest-to-rest* motion): it is easy to realize that this translates to $\dot{\eta}_1$ and $\dot{\eta}_2$ being zero at the start and the goal. These additional conditions can be enforced by going from cubic to quintic polynomials.

(b) One way to solve the passigned control roblem is to perform exact input-output linearization via feedback. To this end, consider the input-output map, which is expressed by (2). Since the decoupling matrix $\boldsymbol{T}(\theta)$ is invertible, we can set

$$\begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \boldsymbol{T}^{-1}(\theta) \begin{pmatrix} u_1 \\ u_2 + g \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 + g \end{pmatrix}, \tag{4}$$

thus obtaining a second-order linear mapping between the outputs and the new inputs u_1, u_2:

$$\begin{aligned} \ddot{x} &= u_1 \\ \ddot{z} &= u_2. \end{aligned}$$

Global exponential tracking of the desired trajectory $x_d(t)$, $z_d(t)$ is then guaranteed by the following PD+feedforward control law

$$\begin{aligned} u_1 &= \ddot{x}_d + k_{p1}(x_d - x) + k_{d1}(\dot{x}_d - \dot{x}) \\ u_2 &= \ddot{z}_d + k_{p2}(z_d - z) + k_{d2}(\dot{z}_d - \dot{z}), \end{aligned}$$

as long as the control gains k_{p1}, k_{d1}, k_{p2}, k_{d2} are positive.

Note that to compute u_1, u_2 we need (in addition to the desired trajectory) the outputs x, z and their first-order derivatives. The expression of the original control inputs f_1, f_2 is then found by plugging u_1, u_2 into (4); the orientation θ is needed for this computation. Overall, computing the control law requires x, z, θ, $\dot{x}$ and $\dot{z}$.

(c) Although $\dot{\theta}$ is not needed by our controller, we design a filter for estimating the whole state for simplicity. It is however easy to modify the filter proposed below so as to avoid estimating $\dot{\theta}$.

With the state $\boldsymbol{x} = (x, z, \theta, \dot{x}, \dot{z}, \dot{\theta}) = (x_1, \ldots, x_6)$, the continuous-time model of the system is written as

$$\begin{aligned} \dot{x}_1 &= x_4 \\ \dot{x}_2 &= x_5 \\ \dot{x}_3 &= x_6 \\ \dot{x}_4 &= f_1 \cos x_3 - f_2 \sin x_3 \\ \dot{x}_5 &= -g + f_1 \sin x_3 + f_2 \cos x_3 \\ \dot{x}_6 &= -d\, f_2. \end{aligned} \tag{5}$$

Using Euler integration over the sampling interval T_s, a discrete-time version of this model is written as

$$\begin{aligned} x_{1,k+1} &= x_{1,k} + x_{4,k}\, T_s \\ x_{2,k+1} &= x_{2,k} + x_{5,k}\, T_s \\ x_{3,k+1} &= x_{3,k} + x_{6,k}\, T_s \\ x_{4,k+1} &= x_{4,k} + f_{1,k} \cos x_{3,k}\, T_s - f_{2,k} \sin x_{3,k}\, T_s \\ x_{5,k+1} &= x_{5,k} - g\, T_s + f_{1,k} \sin x_{3,k}\, T_s + f_{2,k} \cos x_{3,k}\, T_s \\ x_{6,k+1} &= x_{6,k} - d\, f_{2,k}\, T_s. \end{aligned}$$

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

As for the measurement model, we have a total of four measurements coming from the sensors at each sampling instant. The first is the height of the CoM over the ground

$$y_{1,k} = z_k - \sin x_k = x_{2,k} - \sin x_{1,k}$$

while the second is directly the robot orientation:

$$y_{2,k} = \theta_k = x_{3,k}.$$

The last two measurements are the CoM accelerations $\ddot{x}$, $\ddot{z}$. These are input-level quantities that cannot be written as function of the states; therefore, we cannot include them as such in the measurement model. We have two options.

1. The first is to use $\ddot{x}_k$, $\ddot{z}_k$ to reconstruct the actual inputs, as in (3). For doing this, we need θ, which we may replace with the current estimate:
$$\begin{pmatrix} f_{1,k} \\ f_{2,k} \end{pmatrix} = \boldsymbol{T}^{-1}(\hat{\theta}_k) \begin{pmatrix} \ddot{x}_k \\ \ddot{z}_k + g \end{pmatrix}.$$
These input values would then be used in the prediction stage.

2. The second is to integrate numerically the measurements $\ddot{x}_k$, $\ddot{z}_k$ to transform them into virtual measurements of $\dot{x}_{k+1}$, $\dot{z}_{k+1}$, which are states. To do this, however, we need the velocities $\dot{x}_k$, $\dot{z}_k$, which we may replace with their current estimates. This option would require using in the prediction stage the (nominal) force inputs $f_{1,k}$ and $f_{2,k}$ coming from the controller.

Let us choose the first option. The measurement model is therefore simply

$$\boldsymbol{y}_k = \begin{pmatrix} x_{2,k} - \sin x_{1,k} \\ x_{3,k} \end{pmatrix}.$$

This model is also assumed to be perturbed by a white gaussian noise with zero mean and known covariance.

The rest of the solution is straightforward: linearize the motion and measurement models (only the fourth and fifth equations of the former and the first equation of the latter are actually nonlinear) and then write the EKF equations.

Solution of Problem 3

(a) FALSE. Chow Theorem is a necessary and sufficient condition of controllability only for driftless systems. However, the robot of Problem 2 contains a drift even when $g = 0$; indeed, it is a second-order model that, once converted to state-space form, will display a drift on the first three equations, see (5).

(b) TRUE. If $g = 0$, the first two equations of the dynamic model are in the form

$$\begin{pmatrix} \ddot{x} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \qquad \text{so that} \qquad \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \ddot{x} \\ \ddot{z} \end{pmatrix}$$

In particular, the second equation is $f_2 = -\sin\theta\,\ddot{x} + \cos\theta\,\ddot{z}$. Plugging this in the third equation of the dynamic model we get $d\,\sin\theta\,\ddot{x} - d\,\cos\theta\,\ddot{z} - \ddot{\theta} = 0$, which is a second-order constraint in the form $\boldsymbol{a}^T(\boldsymbol{q})\ddot{\boldsymbol{q}} = 0$.

An alternative reasoning is the following. For $g = 0$, the system equations state that

$$\ddot{\boldsymbol{q}} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} f_1 + \begin{pmatrix} -\sin\theta \\ \cos\theta \\ -d \end{pmatrix} f_2 = \boldsymbol{h}_1(\boldsymbol{q}) f_1 + \boldsymbol{h}_2(\boldsymbol{q}) f_2$$

i.e., $\ddot{\boldsymbol{q}}$ belongs to $\text{span}\{\boldsymbol{h}_1(\boldsymbol{q}),\ \boldsymbol{h}_2(\boldsymbol{q})\}$. Therefore, $\ddot{\boldsymbol{q}}$ must be orthogonal to any vector (field) $\boldsymbol{a}(\boldsymbol{q})$ which is orthogonal to both $\boldsymbol{h}_1(\boldsymbol{q})$ and $\boldsymbol{h}_2(\boldsymbol{q})$, or $\boldsymbol{a}^T(\boldsymbol{q})\ddot{\boldsymbol{q}} = 0$, and the claim is true. One possible choice of $\boldsymbol{a}(\boldsymbol{q})$ is the vector product of $\boldsymbol{h}_1(\boldsymbol{q})$ and $\boldsymbol{h}_2(\boldsymbol{q})$:

$$\boldsymbol{a}(\boldsymbol{q}) = \boldsymbol{h}_1(\boldsymbol{q}) \times \boldsymbol{h}_2(\boldsymbol{q}) = \det \begin{pmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & -d \end{pmatrix} = \begin{pmatrix} -d\sin\theta \\ d\cos\theta \\ 1 \end{pmatrix}$$

which leads to the same constraint written before (with a change of sign).

(c) FALSE. Differential flatness guarantees that we can plan a trajectory which is *admissible*, i.e., is consistent with the robot model; but it does not ensure that it will be *feasible*, i.e., that the associated inputs will be inside the given bounds. However, if we were dealing with a robot described by a driftless kinematic model, then the claim would be TRUE, because for such a model any trajectory can be slowed down until the velocity bounds are satisfied (*uniform scaling*).

(d) TRUE. Think about trajectory tracking based on static input-output linearization, in which point B can be driven along arbitrary trajectories.

(e) FALSE. At steady state, the robot Cartesian coordinates will track the reference trajectory exactly. Since these coordinates are precisely the flat outputs, the orientation will evolve as dictated by the reconstruction formula, independently on the initial configuration. One may argue that the reconstruction formula actually provides two evolutions of the orientation, which differ by π (forward and backward motion); in this sense, it would be TRUE that the elicited motion depends on the initial configuration of the unicycle.