MPC-based path following for mobile robots

Assignment 4, 4SC000, TU/e, 2022-2023

Suppose that we wish to provide a control law for a unicycle-type mobile robot to follow a periodic path. The model of the unicycle-type robot is

$$\begin{aligned} \dot{x}(t) &= V(t)\cos(\theta(t)) \\ \dot{y}(t) &= V(t)\sin(\theta(t)) \\ \dot{\theta}(t) &= \omega(t) \end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}$ and $y(t) \in \mathbb{R}$ are the coordinates of the planar position of the robot $p = (x, y)$ at time $t \in [0, \infty)$, and $\theta(t) \in \mathbb{R}$ is the yaw angle of the robot at time $t \in [0, \infty)$. Both $V(t)$ and $\omega(t)$ are control inputs. Since the path is periodic it is assumed to be described by the Fourier series

$$\bar{p}(r) = \gamma(r), \quad \gamma(r) = a + \sum_{k=1}^{N} b_k \cos\left(\frac{2\pi}{T}kr\right) + c_k \sin\left(\frac{2\pi}{T}kr\right)$$

with $a \in \mathbb{R}^2$, $b_k \in \mathbb{R}^2$, $c_k \in \mathbb{R}^2$, $k \in \{1, 2, \ldots, N\}$, for $r \geq 0$. To consider the path over a period we can consider $r \in [0, T)$. To characterize any periodic path, N should be ∞ but, in practice, a finite N, even small, can be used to describe paths of interest.

Consider a periodic trajectory along the path with constant speed $\bar{V}$ defined by $\gamma(\delta^{-1}(s_0 + \bar{V}t))$ for $t \geq 0$, and for some $s_0 = \delta(r_0)$, $r_0 \in [0, T)$ defining the start of the trajectory along the path, and where δ is a map that provides the length of the path up to $r \in [0, T)$. Formally, if we denote the arclength of the path by s,

$$s = \delta(r) = \int_0^r \|\gamma'(v)\| dv. \tag{2}$$

Then, from the description of $\bar{p}(r)$, one can find $\bar{x}(t)$, $\bar{y}(t)$, $\bar{\theta}(t)$, and $\bar{\omega}(t)$ describing such a trajectory such that

$$\begin{aligned} \dot{\bar{x}}(t) &= \bar{V}\cos(\bar{\theta}(t)) \\ \dot{\bar{y}}(t) &= \bar{V}\sin(\bar{\theta}(t)) \\ \dot{\bar{\theta}}(t) &= \bar{\omega}(t) \end{aligned} \tag{3}$$

and, for every $t \geq 0$, $(\bar{x}(t), \bar{y}(t)) = \bar{p}(r)$ for some $r \geq 0$. Such a procedure is described in Appendix 1.

Let us define the position error variables in the world frame

$$e_x(t) = x(t) - \bar{x}(t), \quad e_y(t) = y(t) - \bar{y}(t)$$

and $e(t) = \begin{bmatrix} e_x(t) & e_y(t) & e_\theta(t) \end{bmatrix}^\intercal$ and rotate these to the body axis of the vehicle

$$\begin{bmatrix} \bar{e}_x(t) \\ \bar{e}_y(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) \\ -\sin(\theta(t)) & \cos(\theta(t) \end{bmatrix} \begin{bmatrix} e_x(t) \\ e_y(t) \end{bmatrix}$$

Defining an angle error is harder; for instance $\theta - \bar{\theta}$ is ill-defined as the angles are defined module 2π. Defining them in a finite interval (e.g. from $-\pi$ until π) could solve this issue but would create other issues, namely instantaneous jumps not accounted in the simple dynamics of the angles. One way to solve his is to work with

$$\rho(t) = \begin{bmatrix}\cos(\theta(t)) & \sin(\theta(t))\end{bmatrix}^\intercal \qquad \bar{\rho}(t) = \begin{bmatrix}\cos(\bar{\theta}(t)) & \sin(\bar{\theta}(t))\end{bmatrix}^\intercal$$

whose dynamics are

$$\dot{\rho}(t) = \begin{bmatrix}0 & -\omega \\ \omega & 0\end{bmatrix}\rho(t), \qquad \dot{\bar{\rho}}(t) = \begin{bmatrix}0 & -\bar{\omega} \\ \bar{\omega} & 0\end{bmatrix}\bar{\rho}(t).$$

Then $e_\rho(t) = 1 - \rho(t)^\intercal\bar{\rho}(t)$ can be seen as an error variable.

Given these error variables, the proposed control law is

$$\begin{aligned} V(t) &= \bar{V}(1 - e_\rho(t)) - k_1\bar{e}_x(t) \\ \omega(t) &= \bar{\omega}(t) - k_2\bar{V}\bar{e}_y(t) - k_3 h(\rho(t), \bar{\rho}(t)) \end{aligned} \tag{4}$$

for positive, but otherwise arbitrary, control gains k_1, k_1, k_3 and, for $\Gamma = \begin{bmatrix}0 & -1 \\ 1 & 0\end{bmatrix}$,

$$h(\rho(t), \bar{\rho}) = \begin{cases} -\rho(t)^\intercal\Gamma^\intercal\bar{\rho}(t) \text{ if } \rho(t)^\intercal\bar{\rho}(t) \geq 0 \\ 1 \text{ if } \rho(t)^\intercal\bar{\rho}(t) < 0 \text{ and } \rho(t)^\intercal\Gamma^\intercal\bar{\rho}(t) < 0 \\ -1 \text{ if } \rho(t)^\intercal\bar{\rho}(t) < 0 \text{ and } \rho(t)^\intercal\Gamma^\intercal\bar{\rho}(t) > 0 \end{cases}$$

This control law ensures that

$$\lim_{t\to\infty} e_x(t) = 0, \lim_{t\to\infty} e_y(t) = 0, \lim_{t\to\infty} e_\rho(t) = 0$$

This can be established by using the Lyapunov arguments in Appendix 2.

Assignment 4.1 Program a matlab function

```
[x,y,rho] = unicycletrajtrack(a,b,c,T,Vbar,r0,xi0,K,Tmax,kappa)
```

which provides the outputs $x(t)$, $y(t)$, $\rho(t)$resulting from simulating the trajectory tracking control law evaluated at times t = [0:kappa:Tmax]. Namely, row vectors x=$\begin{bmatrix}x(0) & x(\kappa) & \dots & x(T_{\max} - \kappa) & x(T_{\max})\end{bmatrix}$, y=$\begin{bmatrix}y(0) & y(\kappa) & \dots & y(T_{\max} - \kappa) & y(T_{\max})\end{bmatrix}$ and a matrix rho=$\begin{bmatrix}\rho_1(0) & \rho_1(\kappa) & \dots & \rho_1(T_{\max} - \kappa) & \rho_1(T_{\max}) \\ \rho_2(0) & \rho_2(\kappa) & \dots & \rho_2(T_{\max} - \kappa) & \rho_2(T_{\max})\end{bmatrix}$. The input parameters are

(i) the parameters of the Fourier series description of the periodic path, namely a 2 dimensional column vector a, two $2 \times N$ matrices $b = \begin{bmatrix}b_1 & b_2 & \dots & b_N\end{bmatrix}$, $b = \begin{bmatrix}c_1 & c_2 & \dots & c_N\end{bmatrix}$ and a positive scalar T;

(ii) the additional parameters $\bar{V}$ and r_0 to define the trajectory;

(iii) a colum vector with initial condition xi0 = $\begin{bmatrix}x(0) & y(0) & \cos(\theta(0)) & \sin(\theta(0))\end{bmatrix}^\intercal$;

(iv) a column vector with the control law gains $K = \begin{bmatrix}k_1 & k_2 & k_3\end{bmatrix}^\intercal$;

(v) two parameters kappa, Tmax to specify times at which the output should be evaluted t = [0:kappa:Tmax].

Trajectory tracking is often too demanding for the actuation and can lead to large pose errors. The harsh constraint that a time-parameterized reference must be tracked, can be replaced by the constraint that the distance to the path is kept small (ideally zero) and a velocity reference along the path is followed. This is called path-following. One can see path-following as adding an extra degree of freedom on the positioning reference along the path; rather than $(\bar{x}(t), \bar{y}(t), \bar{\theta}(t))$ we can have $(\bar{x}(s), \bar{y}(s), \bar{\theta}(t))$, with $s = \zeta([x\, y, \theta])$ depending on the state (pose). This map ξ is often specified in an ad-hoc manner. A common approach is to pick the closest point along the path to (x, y).[1]

Suppose that rather than explicitly specifying s we compute it online according to the following policy. Consider a trajectory that starts at a given $r_0 \in [0, T)$ and is defined as $\gamma(\delta^{-1}(\bar{V}t + \delta(r_0)))$ for $t \geq 0$. Let

$$J(\xi(0), r_0) = \int_0^M e_x^2(t) + e_y^2(t) + e_\rho(t) dt, \qquad M = \infty$$

be the cost of the trajectory tracking policy in 4.1 for this trajectory for a given initial pose $\xi(0) = \begin{bmatrix} x(0) & y(0) & \theta(0) \end{bmatrix}^\intercal$ Then

$$r = \zeta(\xi) = \mathrm{argmin}_{\bar{r} \in \mathcal{T}} J(\xi(t), \bar{r})$$

where ideally $\mathcal{T} = [0, T)$, but in practice $\mathcal{T} = [0, T) \cap \{k\nu | k \in \mathcal{Z}\}$, where ν is the step of the grid of r values. Also in practice M must be a large constant. Computing a new s can be seen as recomputing the trajectory. Since we cannot implement such trajectory re-computation at every time t we will implement it at discrete steps $t = t_k = k\tau$. For $k = 0$ the trajectory corresponding to $r = \zeta(\xi(0))$ is computed and the control law in 4.1 is applied in the interval $[0, \tau)$. At $k = 1$ a new trajectory corresponding to $r = \zeta(\xi(\tau))$ is computed and the control law in 4.1 is applied in the interval $[\tau, 2\tau)$. The same procedure is applied for $k = 2$, $k = 3$, etc. Since the trajectory for the control law is recomputed at every time step $k\tau$ we can see this as an MPC strategy for trajectory optimization that results in a path following strategy. For the sake of simplicity assume that Tmax is a multiple of τ, i.e., Tmax $= j\tau$ for some integer j and that τ is a multiple of κ, i.e., $\tau = i\kappa$ for some integer i. Moreover, assume that $M =$ Tmax.

Assignment 4.2 Program a matlab function

```
[x,y,rho] =
unicyclepathfollowing(a,b,c,T,Vbar,xi0,K,Tmax,kappa,tau,nu)
```

that implement this path following strategy rather than the trajectory tracking policy and has the same input and output arguments as unicyclepathfollowing except for the additional parameters τ, ν and the absent parameter r_0 which is not computed and not given.

[1] For more information on path following, see live script LQpathfollowing.mlx, `https://drive.matlab.com/sharing/16908d81-b2af-4bbe-944f-05cb07aec688`

Appendix 1

Note that one can write a closed-form expression for $\gamma'(r)$

$$\gamma'(r) = \sum_{k=1}^{N} -b_k(\frac{2\pi}{T}k)\sin\left(\frac{2\pi}{T}kr\right) + c_k(\frac{2\pi}{T}k)\cos\left(\frac{2\pi}{T}kr\right)$$

Since velocity is constant we know that

$$s(t) = \beta(t), \quad \beta(t) = \bar{V}t$$

These two equations allow us to write r as a function of t. Such a function is denoted by

$$r = \alpha(t)$$

Two equivalent ways to compute this function r are as follows. First we can note that

$$\beta(t) = \delta(\alpha(t))$$

and since $\beta'(t) = \bar{V}$, we must have $\delta'(\alpha(t))\alpha'(t) = \bar{V}$ Since $\delta'(r) = \|\gamma'(r)\|$, α is the solution to the differential equation

$$\dot{\alpha}(t) = \frac{\bar{V}}{\|\gamma'(\alpha(t))\|}$$

with $\alpha(0) = 0$. Second, we can numerically compute $\delta^{-1}(s)$ and determine $\alpha(t)$ from

$$\alpha(t) = \delta^{-1}(t\bar{V}) \tag{5}$$

Differentiating this equation, we obtain the same differential equation as in the first method. In fact, using the equation for the derivative of the inverse

$$\frac{d}{dt}\delta^{-1}(t\bar{V}) = \frac{\bar{V}}{\|\gamma'(r)\||_{r=\delta^{-1}(t\bar{V})}}.$$

After obtaining $\alpha(t)$, we can obtain $\bar{x}(t)$ and $\bar{y}(t)$ from

$$\begin{bmatrix}\bar{x}(t)\\ \bar{y}(t)\end{bmatrix} = \gamma(\alpha(t)).$$

From (3) we can conclude that

$$\bar{\rho}(t) = \begin{bmatrix}\frac{\dot{\bar{x}}(t)}{\bar{V}} & \frac{\dot{\bar{y}}(t)}{\bar{V}}\end{bmatrix}^{\mathsf{T}}$$

and

$$\bar{\theta}(t) = \mathrm{atan2}(\dot{\bar{y}}(t), \dot{\bar{x}}(t)).$$

Thus we need to take the derivatives of $\bar{x}(t)$ and $\bar{y}(t)$. We obtain

$$\begin{bmatrix}\dot{\bar{x}}(t)\\ \dot{\bar{y}}(t)\end{bmatrix} = \gamma'(\alpha(t))\dot{\alpha}(t) = \gamma'(\alpha(t))\frac{\bar{V}}{\|\gamma'(\alpha(t))\|}$$

where we have used (5). To compute $\bar{\omega}$ we can either differentiate the expression for $\bar{\theta}(t)$ or notice that from (3) we must have

$$\ddot{\bar{x}}(t) = -\bar{V}\sin(\bar{\theta}(t))\bar{\omega}(t)$$
$$\ddot{\bar{y}}(t) = \bar{V}\cos(\bar{\theta}(t))\bar{\omega}(t)$$

From this expression we conclude that

$$\bar{\omega}(t) = \frac{-\sin(\bar{\theta}(t))\ddot{\bar{x}}(t) + \cos(\bar{\theta}(t))\ddot{\bar{y}}(t)}{\bar{V}} = \frac{\bar{\rho}(t)^\intercal \Gamma^\intercal \begin{bmatrix} \ddot{\bar{x}}(t) \\ \ddot{\bar{y}}(t) \end{bmatrix}}{\bar{V}}$$

Thus we need to compute the second derivatives of $\bar{x}(t)$ and $\bar{y}(t)$. Computing an extra derivative in the expressions for $\dot{\bar{x}}(t)$ and $\dot{\bar{y}}(t)$, we obtain

$$\begin{bmatrix} \ddot{\bar{x}}(t) \\ \ddot{\bar{y}}(t) \end{bmatrix} = \gamma''(\alpha(t)) \frac{\bar{V}^2}{\|\gamma'(\alpha(t))\|^2} - \gamma'(\alpha(t)) \frac{(\gamma'(\alpha(t)).\gamma''(\alpha(t)))\bar{V}^2}{\|\gamma'(\alpha(t))\|^4}$$

where $a.b$ denotes the inner product of vectors a and b. Note that one can write a closed-form expression for $\gamma''(r)$

$$\gamma''(r) = \sum_{k=1}^{N} -b_k(\frac{2\pi}{T}k)^2 \cos\left(\frac{2\pi}{T}kr\right) - c_k(\frac{2\pi}{T}k)^2 \sin\left(\frac{2\pi}{T}kr\right)$$

Appendix 2

We will first establish that the error

$$\bar{e}(t) = \begin{bmatrix} \bar{e}_x(t) & \bar{e}_y(t) \end{bmatrix}^\intercal$$

satisfies

$$\dot{\bar{e}}(t) = \begin{bmatrix} 0 & \omega(t) \\ -\omega(t) & 0 \end{bmatrix} \bar{e}(t) + \begin{bmatrix} V(t) - \bar{V}\rho(t)^\intercal \bar{\rho}(t) \\ \bar{V}\rho(t)^\intercal \Gamma \bar{\rho}(t) \end{bmatrix}$$

In fact,

$$\begin{aligned}
\frac{d}{dt}\begin{bmatrix} \bar{e}_x(t) \\ \bar{e}_y(t) \end{bmatrix} &= (\frac{d}{dt}\begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) \\ -\sin(\theta(t)) & \cos(\theta(t) \end{bmatrix})\begin{bmatrix} e_x(t) \\ e_y(t) \end{bmatrix} + \begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) \\ -\sin(\theta(t)) & \cos(\theta(t) \end{bmatrix}(\frac{d}{dt}\begin{bmatrix} e_x(t) \\ e_y(t) \end{bmatrix}) \\
&= \begin{bmatrix} 0 & \omega(t) \\ -\omega(t) & 0 \end{bmatrix} \underbrace{\begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) \\ -\sin(\theta(t)) & \cos(\theta(t) \end{bmatrix}\begin{bmatrix} e_x(t) \\ e_y(t) \end{bmatrix}}_{=\bar{e}(t)} \\
&+ \underbrace{\begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) \\ -\sin(\theta(t)) & \cos(\theta(t) \end{bmatrix}}_{\begin{bmatrix} \rho(t)^\intercal \\ \rho(t)^\intercal \Gamma^\intercal \end{bmatrix}} \underbrace{\begin{bmatrix} \cos(\theta(t))V(t) - \cos(\bar{\theta}(t))\bar{V}(t) \\ \sin(\theta(t))V(t) - \sin(\bar{\theta}(t))\bar{V}(t) \end{bmatrix}}_{\begin{bmatrix} \rho(t)V(t) & -\bar{\rho}(t)\bar{V} \end{bmatrix}} \\
&= \begin{bmatrix} 0 & \omega(t) \\ -\omega(t) & 0 \end{bmatrix} \bar{e}(t) + \begin{bmatrix} V(t) - \bar{V}\rho(t)^\intercal \bar{\rho}(t) \\ \bar{V}\rho(t)^\intercal \Gamma \bar{\rho}(t) \end{bmatrix}
\end{aligned}$$

where the fact that $\Gamma^\intercal = -\Gamma$ was used. Consider the following candidate Lyapunov function, for some positive constant ϵ,

$$V(\bar{e}(t)) = \bar{e}_x(t)^2 + \bar{e}_y(t)^2 + \epsilon e_\rho(t).$$

Then

$$\begin{aligned}
\frac{d}{dt}V(\bar{e}(t)) &= 2\bar{e}_x(t)(\frac{d}{dt}\bar{e}_x(t)) + 2\bar{e}_y(t)(\frac{d}{dt}\bar{e}_y(t)) - \epsilon(\dot{\rho}(t)^\intercal \bar{\rho}(t) + \rho(t)^\intercal \dot{\bar{\rho}}(t)) \\
&= 2\bar{e}_x(t)(\omega(t)\bar{e}_y(t) + V(t) - \bar{V}\rho(t)^\intercal \bar{\rho}(t)) \\
&\quad + 2\bar{e}_y(t)(-\omega(t)\bar{e}_x(t) + \bar{V}\rho(t)^\intercal \Gamma \bar{\rho}(t)) - \epsilon(\omega\rho(t)^\intercal \Gamma^\intercal \bar{\rho}(t) + \bar{\omega}\rho(t)^\intercal \Gamma \bar{\rho}(t)) \\
&= 2\bar{e}_x(t)(V(t) - \bar{V}\rho(t)^\intercal \bar{\rho}(t)) - 2\bar{e}_y(t)\bar{V}\rho(t)^\intercal \Gamma^\intercal \bar{\rho}(t) - \epsilon(\omega\rho(t)^\intercal \Gamma^\intercal \bar{\rho}(t) + \bar{\omega}\rho(t)^\intercal \Gamma \bar{\rho}(t)).
\end{aligned}$$

Plugging in the control law (4) and making $\epsilon = 2/k_2$ we obtain

$$\frac{d}{dt}V(\bar{e}(t)) = -2k_1\bar{e}_x^2(t) + 2\frac{k_3}{k_2}h(\rho(t),\bar{\rho}(t))\rho(t)^\intercal\Gamma^\intercal\bar{\rho}(t)$$

which is negative semi-definite. Namely it is zero in the manifold $\{\bar{e}|\bar{e}_y(t) = 0\}$. This manifold contains no equilibra. Thus, using LaSalle's invariance principle, we conclude that $\bar{e}(t) \to 0$ as $t \to \infty$ and this implies that $e(t) \to 0$ as $t \to \infty$.