CHAPTER 9

Control of robotic systems

9.1. Path tracking control

In this section we use the following model for a robot:

$$\begin{aligned} \dot{x} &= v\cos\phi \\ \dot{y} &= v\sin\phi \\ \dot{\phi} &= \omega \end{aligned} \tag{9.1}$$

9.1.1. Curvature driving. Suppose now the reference path (given by some path planning algorithm) to track is

$$\begin{aligned} x_d(t) &= p(t) \\ y_d(t) &= q(t),\ 0 \le t \le T. \end{aligned} \tag{9.2}$$

We assume that $\dot{p}(t)^2 + \dot{q}(t)^2 \neq 0$ for any $t \in [0, T]$.

In order to track the trajectory, we need

$$\begin{aligned} \dot{x_d} &= v_d\cos(\theta_d) \\ \dot{y_d} &= v_d\sin(\theta_d). \end{aligned} \tag{9.3}$$

Solving the equations we have

$$v_d(t) = \sqrt{\dot{p}(t)^2 + \dot{q}(t)^2}\ \text{(forward)},\ \theta_d(t) = atan(\dot{p}(t), \dot{q}(t)).$$

This gives

$$\omega_d(t) = \dot{\theta}_d(t) = \frac{\ddot{q}(t)\dot{p}(t) - \ddot{p}(t)\dot{q}(t)}{v_d(t)^2}.$$

By this way we obtain an open-loop control $v_d(t),\ \omega_d(t)$ for the trajectory tracking. In fact this is the unique control that gives exact tracking.

However, the above controller is not robust at all with respect to disturbances and measurement errors. Thus we present a feedback controller that does the tracking only approximately, but robustly.

9.1.2. Virtual vehicle approach. In the *virtual vehicle* approach, the motion of the reference point (the virtual vehicle) on the planned trajectory is governed by a differential equation containing error feedback. It can be viewed as a combination of the conventional trajectory tracking, where the reference trajectory is parameterized in time, and a dynamic path following approach, where the criterion is to stay close to the geometric path, but not necessarily close to an a priori specified point at a given time. The main idea

behind our approach can be seen in Figure 1, and the reason for calling the reference point, together with the associated differential equation, a virtual vehicle is that the reference point is moving on the path that we want the platform to follow. At the same time it has its own dynamics for describing the motion, and one of the advantages with our approach is that it is quite robust with respect to measurement errors and external disturbances. When one uses directional sensors, it is important that the relative orientation of the robot is known. This naturally could be achieved if we could specify at what relative orientation the look-ahead distance should be kept. It seems natural that we choose a point (x_L, y_L) on the robot's axis of orientation, a distance d_ρ from the center (figure 1). Furthermore, by explicitly setting d_ρ to $L > 0$ and fixing the position to the axis of orientation we now control a point that can be driven arbitrarily close to $(p(s), q(s))$ without causing ϕ_d to be undefined. This particular choice of control point also contributes to the control of the orientation towards the reference point, since controlling (x_L, y_L) towards $(p(s), q(s))$ means simultaneously controlling ϕ towards ϕ_d.

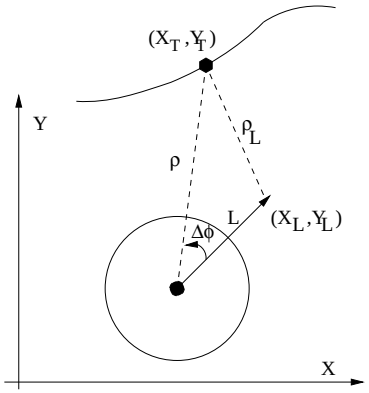

FIGURE 1. Virtual vehicle.

Now define the new point (x_L, y_L) on the axis of orientation a distance L from the center of the robot, (x, y).

$$\begin{aligned} x_L &= x + L\cos\phi \\ y_L &= y + L\sin\phi \end{aligned} \tag{9.4}$$

Derivation and the unicycle model gives

$$\begin{bmatrix} \dot{x}_L \\ \dot{y}_L \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\phi & -L\sin\phi \\ \sin\phi & L\cos\phi \end{bmatrix}}_{A} \begin{bmatrix} v \\ \omega \end{bmatrix} := \begin{bmatrix} v_L \\ \omega_L \end{bmatrix}.$$

It is well known that by this way one can feedback linearize the dynamics since the matrix A is always nonsingular. Thus,

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\frac{1}{L}\sin\phi & \frac{1}{L}\cos\phi \end{bmatrix} \begin{bmatrix} v_L \\ \omega_L \end{bmatrix}. \tag{9.5}$$

If the controls v_L and ω_L are chosen as

$$\begin{aligned} v_L &= -k(x_L - x_r(s)) + \dot{x}_r \\ \omega_L &= -k(y_L - y_r(s)) + \dot{y}_r, \end{aligned} \tag{9.6}$$

including both a proportional and a derivative part, x_L and y_L are driven towards x_r and y_r. In this case ρ is controlled towards L instead of zero and therefore ϕ_d does not approach a singularity. Using (9.6) in (9.5) we get

$$\begin{aligned} v &= -k(L - \rho\cos\Delta\phi) + \sqrt{p'^2 + q'^2}\,\cos(\theta_r - \phi)\dot{s} \\ \omega &= \frac{k\rho}{L}\,\sin\Delta\phi + \frac{1}{L}\sqrt{p'^2 + q'^2}\,\sin(\theta_r - \phi)\dot{s} \end{aligned}$$

where θ_r is the orientation of the trajectory in the point $(p(s), q(s))$. We let

$$\dot{s} = \frac{v_0}{\sqrt{p'^2 + q'^2}}\, e^{-\alpha\rho_L} \tag{9.7}$$

and end up with

$$\begin{cases} v &= k(\rho\cos\Delta\phi - L) + v_0 e^{-\alpha\rho_L}\cos(\theta_r - \phi) \\ \omega &= \frac{k\rho}{L}\,\sin\Delta\phi + \frac{v_0}{L}\sin(\theta_r - \phi). \end{cases} \tag{9.8}$$

Remark: If the reference path is only given as a collection of dense way points $\{(x_d(s), y_d(s)) : s = 1, 2, \ldots, N\}$, then (9.7) can be modified as

$$s_{k+1} = s_k + \sigma(\rho_k), \tag{9.9}$$

where

$$\sigma = \begin{cases} 0 & \text{if } \rho_k > \epsilon_\rho \\ 1 & \text{if } \rho_k \leq \epsilon_\rho \end{cases}$$

A new control objective here is to align the orientation of the robot and the orientation of the path using $\theta_r - \phi$. Trying hard to align the robot to the orientation of the path when the robot is far away is not very appropriate, and is suppressed by the exponential term. When the robot comes closer and ρ_L approaches zero the situation changes.

$$\Delta\phi \to 0 \quad \Leftrightarrow \rho_L \to 0 \quad \Rightarrow \quad \rho \to L$$

$$\Rightarrow \begin{cases} v &\to v_0\cos(\theta_r - \phi) \\ \omega &\to \frac{v_0}{L}\sin(\theta_r - \phi) \end{cases}$$

With a perfect alignment, $\Delta\phi = 0$, $\theta_r - \phi = 0$ and $\rho = L$, thus it is clear that v_0 represents the desirable tracking speed.

9.2. Consensus problem

We begin by considering a flock of N birds. Each bird flies with the same speed but with possibly different directions. Namely

$$v_i = (v\cos\theta_i, v\sin\theta_i)^T,$$

where θ_i is the heading of bird i, and for the sake of simplicity, we assume the birds fly on a plane.

Now suppose for each bird, it changes its heading by the following model:

$$\dot{\theta}_i = u_i,$$

or

$$\theta_i(t+1) = \theta_i(t) + u_i(t) := \omega_i(t).$$

An interesting question is how each bird should update its heading so eventually we have

$$\theta_1(t) = \cdots = \theta_N(t).$$

It turns out

$$\omega_i(t) = \frac{1}{N}\sum_{j=1}^{N}\theta_j(t)$$

will do the trick. One thing we should be careful here is to calculate the average heading properly. We can rewrite the control as

$$\omega_i(t) = \theta_i(t) + \frac{1}{N}\sum_{j\neq i}(\theta_j(t) - \theta_i(t)),$$

or,

$$u_i(t) = \frac{1}{N}\sum_{j\neq i}(\theta_j(t) - \theta_i(t)). \tag{9.10}$$

If we use a local coordinate $\theta_i \in (-\pi, \pi]$, then $\omega_i(t)$ can be calculated properly.

A problem with the controller (9.10) is that each bird needs to know the relative headings of all the other birds. In the following we show that this is not necessary.

Now we consider a system of N agents:

$$\dot{x}_i = u_i,\ i = 1, \cdots, N \tag{9.11}$$

where x_i can be viewed as heading, position or other quantities.

We define the consensus problem as follows:

Consensus problem:

Find $u_i(t)$ such that as $t \to \infty$ we have

$$x_1(t) = x_2(t) = \cdots = x_N(t),$$

here we assume that agent i can only detect relative errors $x_j - x_i$ of its neighbors, namely $j \in N_i$.

Similar to the flocking problem, we consider a controller of the following type:

$$u_i(t) = \sum_{j\in N_i} a_{ij}(x_j - x_i), \tag{9.12}$$

where a_{ij} are positive weights. If we let $x = (x_1, \cdots, x_N)^T$, then

$$\dot{x} = -Lx, \tag{9.13}$$

where

$$L = D - A = diag(\sum_{j\neq 1} a_{1j}, \cdots, \sum_{j\neq N} a_{Nj}) - [a_{ij}].$$

Now define

$$V(x) = x^T L x = \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i} a_{ij}(x_j - x_i)^2.$$

PROPOSITION 9.1. *The consensus problem is solved, namely as $t \to \infty$, $x_1(t) = \cdots = x_N(t)$ in (9.13), if and only if*

$$V(x) = 0 \iff x_1 = x_2 = \cdots = x_N. \tag{9.14}$$

In fact, in this case

$$\lim_{t\to\infty} x_i(t) = \frac{1}{N}\sum_{i=1}^{N} x_i(0).$$

9.2.1. Connection to graph theory. We take graph as a collection of nodes (vertices) and edges that connect the nodes, denoted by $G = (V, E)$. If we consider each agent i as a node in a graph and each positive weight a_{ij}

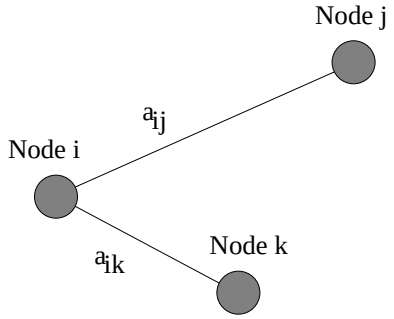

FIGURE 2

defines an edge between Node i and Node j, then the matrix A given above uniquely defines a graph. A is called the *adjacency matrix* of the graph, D the *degree matrix* and L the *graph Laplacian*.

We say a graph is *connected* if any two nodes are connected by edges.

PROPOSITION 9.2. *The consensus problem is solved if and only if the associated graph is connected.*