6

Path Following Strategies for Nonlinear Systems

Path following is a control task that consists of following a prespecified path usually parameterized by its arc length. Usually it is assumed that a vehicle that follows the path moves forward, and the time information, comparing to tracking, is not a control demand, i.e., there are no temporal specifications.

Design of feedback controllers for the path following may be considered from two points of view. First, by separating the geometric and time information of a path, path following may be seen as a subproblem of trajectory tracking (Coelho and Nunes 2005). Second, looking at the controlled states, path following may be a part of stabilization (Canudas de Wit et al. 1996).

Path following control problems are addressed in a lot of works (see, e.g., Micaelli and Samson 1993, Samson and Ait-Abderrahim 1991 for the first results, Jiang and Nijmeijer 1996 and Canudas de Wit et al. 1993 for references). The path following problem may be addressed at the kinematic control level, as in Micaelli and Samson (1993), Samson and Ait-Abderrahim (1991), Seo and Lee (2006), or at the dynamic control level as in Bakker et al. (2010), Coelho and Nunes (2005), and Soetanto et al. (2003). Significant progress has been made in path following, in both the model formulation and controller design, from the time of publishing the pioneering works. In Soetanto et al. (2003), results obtained in Micaelli and Samson (1993) were extended to the dynamic level with parameter uncertainties, and a global convergence of following is achieved. In Coelho and Nunes (2005), a discrete state-space controller is applied. It guarantees overall system stability in the presence of external disturbances, modeling errors, and noise. In Seo and Lee (2006), bounded angular velocity errors are taken into account, and a sliding mode-type controller is designed to obtain a robust vehicle steering. The robustness applies to the input disturbance and uncertain dynamics. A classical presentation of the path following problem in the context of other control problems can be found in Canudas de Wit et al. (1996).

6.1 Path Following Strategies Based on Kinematic Control Models

Path following at the kinematic level is formulated as follows: Given a car-like vehicle of a specified kinematics, design a feedback controller that enables path following such that a distance to the path and the orientation error tend to zero.

Consider then a mobile robot whose kinematics is equivalent to that of a unicycle. The robot is equipped with two differential-drive wheels in the same axis and one castor wheel that prevents the robot platform from falling. Denote by (x,y) the position of the center of the rear axis and by φ the heading angle of the robot (see Figure 6.1).

The kinematic constraints on the robot motion have the form as presented in Example 3.2:

$$\begin{aligned} \dot{x} &= v\cos\ \ . \\ \dot{y} &= v\sin\ \ , \\ \dot{\ } &= \omega, \end{aligned} \tag{6.1}$$

where control inputs are the forward velocity $v = \dot{x}\cos\ \ + \dot{y}\sin\ \ $, and the robot angular velocity is ω. Also, it is usually assumed that motors of the robot are capable of delivering larger velocities than the ones desirable during normal operation.

Let us denote a control input (2 × 1) vector by $u = (v, \omega)$ and a velocity (2 × 1) vector of robot wheels by $u_r = (v_r, v_l)$. Then, the forward and angular

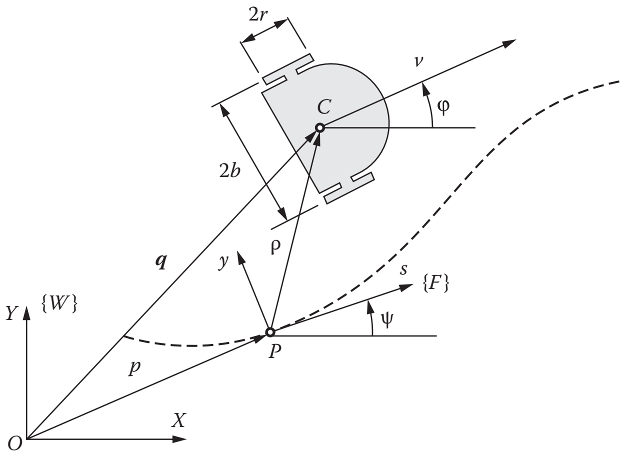

FIGURE 6.1
Mobile robot parameters.

velocities of the robot are related to the right and left wheel velocities as follows:

$$u = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 \\ 1/2b & -1/2b \end{bmatrix} \begin{bmatrix} v_r \\ v_l \end{bmatrix} \tag{6.2}$$

with $2b$ being the distance between the wheels.

The path following problem is parameterized as follows. The kinematic model of the robot is transformed to the Frenet frame {F} that may move along the path. The axes of the Frenet frame are the body axes of a virtual vehicle that should be followed by a real one. In what follows, we use the notation that is now standard in textbooks; for more details, see Micaelli and Samson (1993), Samson and Ait-Abderrahim (1991), and Soetanto et al. (2003).

The origin of the Frenet frame {F} enables evolving according to a conveniently defined time function. It yields an extra controller design parameter. According to Figure 6.1, P is an arbitrary point on the path to be followed, and C is the mass center of the robot. The frame {F} origin is located at P, and its axes are (s,y). Coordinates of C may be expressed by two vectors, either as $q = (X, Y, 0)$ or $\rho = (s_1, y_1, 0)$. The position vector for P can be denoted by p. The rotation matrix that describes transformation from the world coordinate frame {W} coordinates (X,Y) to the frame {F} coordinates (s,y) is $R = R(\psi)$, where ψ is the angle between the horizontal axis parallel to X and s. The following relations hold:

$$\begin{aligned} \dot{\psi} &= c_c(s)\dot{s}, \\ \dot{c}_c(s) &= g_c(s)\dot{s}, \end{aligned} \tag{6.3}$$

where $c_c(s)$ is the path curvature,

$$g_c(s) = \frac{dc_c(s)}{ds}.$$

The velocity of P in {W} expressed in {F} becomes

$$\left(\frac{dp}{dt}\right)_F = [\dot{s}, 0, 0]^T,$$

and the velocity of C in {W} is

$$\left(\frac{dq}{dt}\right)_W = \left(\frac{dp}{dt}\right)_W + R^{-1}\left(\frac{d\rho}{dt}\right)_F + R^{-1}(\dot{\psi} \times r). \tag{6.4}$$

Multiplying Equation (6.4) by R from the left yields

$$R\left(\frac{dq}{dt}\right)_W = \left(\frac{dp}{dt}\right)_F + \left(\frac{d\rho}{dt}\right)_F + \dot{\psi} \times r. \tag{6.5}$$

Using (6.3) and the relations

$$\left(\frac{dq}{dt}\right)_W = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ 0 \end{bmatrix}, \left(\frac{d\rho}{dt}\right)_F = \begin{bmatrix} \dot{s}_1 \\ \dot{y}_1 \\ 0 \end{bmatrix}, \ \dot{\psi} \times r = \begin{bmatrix} -c_c(s)\dot{s}y_1 \\ c_c(s)\dot{s}s_1 \\ 0 \end{bmatrix},$$

Equation (6.5) takes the form

$$R\begin{bmatrix} \dot{X} \\ \dot{Y} \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{s}(1-c_c(s)y_1)+\dot{s}_1 \\ \dot{y}_1 + c_c(s)\dot{s}s_1 \\ 0 \end{bmatrix}. \tag{6.6}$$

Solving Equation (6.6) for $\dot{s}_1, \dot{y}_1$ yields

$$\begin{aligned} \dot{s}_1 &= [\cos\psi \quad \sin\psi]\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} - \dot{s}(1-c_c(s)y_1), \\ \dot{y}_1 &= [-\sin\psi \quad \cos\psi]\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} - c_c(s)\dot{s}s_1. \end{aligned} \tag{6.7}$$

The relations in Equation (6.7) are different than the ones obtained in Micaelli and Samson (1993) due to the translation of the {F} origin. There, as well as in Seo and Lee (2006) and Canudas de Wit et al. (1996), $s_1 = 0$, because P is defined as the projection of C on the path. What follows is that Equations (6.7) have to be solved for $\dot{s}$, and the singularity shows up at $y_1 = 1/c_c$. It results in the requirement on an initial location of C, which must remain in a tube around the path. The tube radius must be less than $1/c_{c\max}$, where $c_{c\max}$ is the maximum path curvature. When $c_{c\max}$ is large, even in a small section of the path, this restriction constrains the robot initial position no matter where it starts its motion with regard to the path. When $s_1 \neq 0$, an extra degree of freedom is introduced to a controller design, but the singularity can be avoided. A new virtual target is assigned to the variable s_1.

The robot kinematics in {W} is described by Equations (6.1). Substituting (6.1) into (6.7) and denoting by $\theta_r = \quad - \psi$, we may obtain the kinematic model in {F} coordinates as

$$\begin{aligned} \dot{s}_1 &= -\dot{s}(1-c_c(s)y_1) + v\cos\theta_r, \\ \dot{y}_1 &= -c_c(s)\dot{s}s_1 + v\sin\theta_r, \\ \dot{\theta}_r &= \dot{\ } - c_c(s)\dot{s}. \end{aligned} \tag{6.8}$$

For the kinematic model (6.8), a controller may be designed. The control goal is to drive y_1, s_1 and θ_r to zero. Following Samson and Ait-Abderrahim (1991) and Soetanto et al. (2003), the control laws may be selected as

$$\begin{aligned} \dot{s} &= v\cos\theta_r + k_1 s_1, \\ \dot{\theta}_r &= \dot{\delta} - \gamma y_1 v \frac{\sin\theta_r - \sin\delta}{\theta_r - \delta} - k_2(\theta_r - \delta), \end{aligned} \tag{6.9}$$

where it is assumed that for some design parameter γ, the following hold:

- $\lim_{t\to\infty} v(t) \neq 0$,
- $\delta(0, v) = 0$,
- $y_1 v \sin\delta(y_1, v) \leq 0 \quad \forall y \quad \forall v$.

The function δ is significant in the transient maneuvers during the path approach phase (see Micaelli and Samson 1993 for details). To prove the convergence of the closed-loop Equations (6.8) and (6.9), the following Lyapunov function candidate can be selected as

$$V_1 = \frac{1}{2}\left(s_1^2 + y_1^2\right) + \frac{1}{2\gamma}[\theta_r - \delta(y_1, v)]^2. \tag{6.10}$$

It can be verified that for controls (6.9), $\dot{V}_1 \leq 0$.

In the kinematic control design, the robot velocity $v(t)$ follows a desired velocity function $v_d(t)$. The angular velocity $\dot{\theta}_r$ is assumed to be a control input.

6.2 Path Following Strategies Based on Dynamic Control Models

Let us now, using the backstepping approach, extend the kinematic controller (6.9) to the dynamic one. A dynamic model of the two-wheeled mobile robot, specified in {F} coordinate frame, is of the form

$$\begin{aligned} \dot{v} &= \frac{\tau_1 + \tau_2}{mr}, \\ \dot{\omega} &= \frac{b(\tau_1 - \tau_2)}{Ir} - c_c\ddot{s} - g_c\dot{s}^2, \end{aligned} \tag{6.11}$$

where m is the robot mass, I is its moment of inertia, r is the wheel radius, $2b$ is the distance between the wheels, and τ_1, τ_2 are control inputs.

The path following task at the dynamic level is formulated as follows: Given a desired velocity profile $v_d(t) > v_{\min} > 0$, design feedback control laws for τ_1 and τ_2 to drive y_1, θ_r and $v(t) - v_d(t)$ asymptotically to zero.

Notice the following differences between the kinematic and dynamic controller designs:

- In the dynamic design, a feedback controller has to be designed such that the tracking error $v(t) - v_d(t)$ approaches zero.
- The robot angular velocity is not a control input.

The virtual control law for the desired behavior of $\dot{\theta}_r$ in Equations (6.9) can be designed as follows (Krstic et al. 1995; Soetanto et al. 2003):

$$\dot{\theta} = \dot{\delta} - \gamma y_1 v \frac{\sin\theta_r - \sin\delta}{\theta_r - \delta} - k_2(\theta_r - \delta). \tag{6.12}$$

Let $e = \dot{\theta}_r - \dot{\theta}$ be the following error. Replacing $\dot{\theta}_r = e + \dot{\theta}$ in the derivative of the Lyapunov function candidate V_1 selected as in Equation (6.10), we may show that

$$\dot{V}_1 = -k_1 s_1^2 + y_1 v \sin\delta - \frac{(\theta_r - \delta)^2}{\gamma} + \frac{(\theta_r - \delta)}{\gamma} e. \tag{6.13}$$

Select now a new Lyapunov function candidate V_2 as

$$V_2 = V_1 + \frac{1}{2}\left[e^2 + (v - v_d)^2\right] \tag{6.14}$$

It can be verified that for

$$\dot{e} = -\frac{(\theta_r - \delta)}{\gamma} - \frac{k_3}{I} e, \qquad \dot{v} = \dot{v}_d - \frac{k_4}{m}(v - v_d), \tag{6.15}$$

the derivative of V_2 satisfies $\dot{V}_2 \le 0$.

Solving the robot dynamics (6.11) for τ_1 and τ_2, we obtain that

$$\begin{aligned} \tau_1 &= \frac{r}{2}\left[m f_2 - k_4(v - v_d) + \frac{I}{b} f_1 - \frac{k_3}{b} e \right], \\ \tau_2 &= \frac{r}{2}\left[m f_2 - k_4(v - v_d) - \frac{I}{b} f_1 + \frac{k_3}{b} e \right] \end{aligned} \tag{6.16}$$

with $f_1 = \ddot{\theta} - \frac{(\theta_r - \delta)}{\gamma} + c_c \ddot{s} + g_c \dot{s}^2$, $f_2 = \dot{v}_d$.

Details about the simulation study and the controller (6.16) performance with the additional adaptation for uncertainties of parameters m, I, r, b, can be found in Soetanto et al. (2003) and references therein.

Problems

1. Prove that the time derivative of the Lyapunov function candidate V_1 defined in Equation (6.10) is negative semi-definite for control inputs (6.9).
2. Derive the equations of motion for the mobile robot and transform them to the dynamic control model (6.11) in {F} frame.
3. Prove that the time derivative of the Lyapunov function candidate V_2 defined in Equation (6.14) is negative semi-definite for control inputs (6.16).

References

Bakker, T., K. Asselt, and J. Bontsema. 2010. A path following algorithm for mobile robots. *Auton. Robot.* 29:85–97.

Coelho, P., and U. Nunes. 2005. Path-following control of mobile robots in presence of uncertainties. *IEEE Trans. Robot*. 21(2):252–261.

Canudas de Wit, C., H. Khennouf, C. Samson, and O. Sordalen. 1993. Nonlinear control design for mobile robots. In *Recent trends in mobile robotics*, ed. Y.F. Zheng, World science series in robotics and automated systems 11:121–156.

Canudas-de-Wit, C., B. Siciliano, and G. Bastin, ed. 1996. *Theory of robot control*. Communications and Control Engineering Series. New York: Springer.

Jiang, Z., and H. Nijmeijer. 1999. A recursive technique for tracking control of nonholonomic systems in chained form. *IEEE Trans. Robot. Automat*. 44(2):265–279.

Krstic, M., I. Kanellakopoulos, and P. Kokotovic. 1995. *Nonlinear and adaptive control design*. New York: John Wiley & Sons.

Micaelli, A., and C. Samson. 1993. Trajectory tracking for unicycle type and two-steering wheels mobile robots. *Technical Report No. 2097*, *INRIA*, Sophia Antipolis, France.

Samson, C., and K. Ait-Abderrahim. 1991. Mobile robot control part 1: Feedback control of a nonholonomic mobile robot. *Technical Report No. 1281*, *INRIA*, Sophia Antipolis, France.

Seo, K., and J.S. Lee. 2006. Kinematic path-following control of a mobile robot under bounded angular velocity error. *Adv. Robot.* 20(1):1–23.

Soetanto, D., L. Lapierre, and A. Pascoal. 2003. Adaptive, non-singular path-following control of dynamic wheeled robots. In *Proc. 42nd IEEE Conf. Decision Contr.* 1765–1770. Maui, HI.