Trajectory-tracking Sliding Mode Control for Two-Wheeled Mobile Robot

Sizhu Cheng, Chen Liu, Chengbao Wu and Chenyao Song

ABSTRACT

According to the mechanical structure and operation principle of differential-drive two-wheeled robot, and aiming at the nonlinear systems with uncertainties variable structure, a trajectory tracking model based on sliding mode control (SMC) was designed by utilizing state vector to establish the model of system and controller. The simulating results show that robot can track line, circle and S shape trajectories well, which gave reasonable dynamic responses, adjustment performance, as well as perfect disturbance rejection. The system can eliminate errors according to the deviation from sliding surface by switching the structure of controller and is robust to external disturbance.

INTRODUCTION

Before designing a controller, the system model should be defined firstly. In reality, most systems are nonlinear, and the model is imprecise, with actual uncertainty[1][2]. From a control point of view, modeling inaccuracies can be classified into two major kinds: 1. Structured (or parametric) uncertainties, and 2. Unstructured uncertainties (or unmodelled dynamics)[3]. For any practical design, the system model must be explicitly defined[4][5]. For the class of systems to which it applies, sliding controller design provides a systematic approach to the problem of maintaining stability and consistent performance in the face of modeling imprecision[6]. It has been widely used in manipulator, ground and submarine operation vehicles, automotive transmission systems and engines, high-capacity motors and power control systems[7][8].

Sizhu Cheng, Chen Liu, Chengbao Wu, Chenyao Song, School of Aircraft Maintenance Engineering, Guangzhou Civil Aviation College, Guangzhou, China 510470
1. KINETIC MODEL OF MOBILE ROBOT

Position coordinate of robot was defined by state parameters \([x(t), y(t), \theta(t)]^T\), with \((x, y)\) gives the gravity centre and \(\theta\) gives the attitude angle. Angular velocity of robot’s left and right wheel were indicated by \(\omega_L\) and \(\omega_R\). Instead of controlling the speeds of left wheel and right wheel directly, it is possible to control the linear velocity \(v(t)\) and the angular velocity \(\omega(t)\) of the robot, as show in figure 1. The kinetic equation is used to describe the relationship between speed \((v(t), \omega(t))\) and state parameters \([x(t), y(t), \theta(t)]^T\), and the kinetic model of the wheel is obtained as followed:

\[
\begin{align*}
\dot{x}(t) &= v(t) \cdot \cos \theta(t) \\
\dot{y}(t) &= v(t) \cdot \sin \theta(t) \\
\dot{\theta}(t) &= \omega(t)
\end{align*}
\]  

(1)

DESIGN OF TRAJECTORY TRACKING CONTROLLER

Control Requirements

The trajectory of the robot is controlled by cascade control, in which the outer loop adopts sliding mode control, and the inner loop uses PID control. Figure 2 is the general control scheme of SMC loop.

Now considered the single-input system as followed:

\[
\begin{bmatrix}
x \\
y \\
\theta
\end{bmatrix}
= f(x, y, \theta) + b(x, y, \theta) \cdot u
\]  

(2)

\(u\) indicated the output control signal of the controller \([v(t), \omega(t)]^T\) in figure 2, which including torque signal of wheel. Function \(f(x, y, \theta)\) and \(b(x, y, \theta)\) are imprecise, and there are many sources of uncertainties[9][10]. In figure 2, \([x_d, y_d, \theta_d]^T\) indicates the desired position coordinate of the trajectory would be tracked. The control design that should be taken into consideration in reality, is when model \(f(x, y, \theta)\) and \(b(x, y, \theta)\) containing uncertainties, actual state \([x, y, \theta]^T\) can still track specific time-varying desired state \([x_d, y_d, \theta_d]^T\). This assumption can be detailed implemented by designing a robust controller including SMC and PID control, and the controller should output control signal vector \([v_c, \omega_c]^T\) before disturbances, as shown in figure 2. In a second order system, position or velocity cannot change abruptly, so that any desired trajectory feasible from time \(t = 0\) necessarily starts with the same position and velocity as those of the plant[11].
Design Sliding Mode Controller

![Figure 1. Two-wheeled differential-drive robot with state parameters \((x, y, \theta)^T\).](image)

Figure 2. General control scheme of SMC loop.

In figure 2, the state vector of robot \([x_r, y_r, \theta_r]^T\) was defined by real position vector at the next moment, and error vector was defined as \([x_e, y_e, \theta_e]^T = [x_d, y_d, \theta_d]^T - [x_r, y_r, \theta_r]^T\). \((v_r(t), \omega_r(t))\) indicates the real line velocity and angular velocity at the next moment of the robot. Therefore, the error vector of trajectory tracking should be:

\[
\begin{bmatrix}
    x_e \\
    y_e \\
    \theta_e
\end{bmatrix} =
\begin{bmatrix}
    \cos \theta_d & \sin \theta_d & 0 \\
    -\sin \theta_d & \cos \theta_d & 0 \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x_r - x_d \\
    y_r - y_d \\
    \theta_r - \theta_d
\end{bmatrix}
\] (3)

Differentiate the upper forms and substituted into function (3):

\[
\begin{align*}
    \dot{x}_e &= -v_{d} + v_r \cdot \cos(\theta_r) + y_e \cdot \omega_d \\
    \dot{y}_e &= v_r \cdot \sin(\theta_r) - x_e \cdot \omega_d \\
    \dot{\theta}_e(t) &= \omega_r - \omega_d
\end{align*}
\] (4)

\([v_d \quad \omega_d]^T\) is the desired velocity control signal that input to sliding mode controller. The following sliding surfaces are proposed:

\[
s_i = \dot{x}_e + k_1 x_e
\] (5)
\[ s_2 = \dot{y}_e + k_2 y_e + k_0 \text{sgn}(y_e) |\theta_e| \]  

(6)

\[ V = \frac{1}{2} s^T \cdot s \]

is the Lyapunov function candidate[12], and

\[ \dot{V} = s_1 \cdot \dot{s}_1 + s_2 \cdot \dot{s}_2 \]

\[ \dot{V} = s_1 \cdot (-Q_1 \cdot s_1 \cdot \text{sgn}(s_1)) + s_2 \cdot (-Q_2 \cdot s_2 \cdot \text{sgn}(s_2)) \]

\[ \dot{V} = -s^T \cdot Q \cdot s - P_1 \cdot |s_1| - P_2 \cdot |s_2| \]  

(7)

As \( \dot{V} \) is negative semi-definite function, it is sufficient to choose \( Q_i \) and \( P_i \) such that \( Q_i P_i \geq 0 \)[13]. The above formulas are simultaneously and mathematically transformed, the final control signal output by the controller should be:

\[ \dot{v}_e = -Q_1 s - P_1 \text{sgn}(s) - k_1 \dot{y}_e - y_e \dot{\omega}_e - \dot{\omega}_e \omega_e + v_r \theta \sin(\theta_r) + \dot{v}_e \]  

(8)

\[ \omega_e = \omega_e + \frac{-Q_2 s - P_2 \text{sgn}(s) - k_2 \dot{y}_e - v_e \sin(\theta_r) + \dot{x}_r \omega_r + \dot{x}_r \omega_r}{v_r \cos(\theta_r) + k_0 \text{sgn}(y_e) \text{sgn}(\theta_r)} \]  

(9)

**SIMULATIONS**

SMC loop translated variables \( v_e \) and \( \omega_e \) to PID loop, and PID loop feedback \( \omega_r, v_r, x_r, y_r, \theta_r \) signals to SMC loop.

**Straight Line Trajectory**

The positions of the virtual and actual robot are shown in figure 3. For the desired signal, only the X direction of the trajectory increases, which gives a straight line trajectory, and the desired distance was 1.7m. The real output is deflected slightly with the largest error less than \( 20 \times 10^{-3} \), and the result was reasonably. The error reached the maximum at 1.5m, and then decreased.

![Figure 3. Position of the virtual and actual robot in Cartesian position(m).](image-url)
Circle Trajectory

The positions of the virtual and actual robot are shown in figure 4. The input desired trajectory is a circle with a radius of 1m, and the robot actually runs three cycles. From figure 4b), it can be found that at the beginning the error is zero while keep increasing. The largest error of circle trajectory was about 0.12m, and the error was steady, which was steady-state error. The reason of the steady-state error can be seen from the desired and real linear and the angular velocities with error in figure 5. In figure 5b), the angular velocity of the robot had obvious oscillation, while there was no steady-state error. However, in figure 5a), the linear velocity had steady-state error all the time and larger than the desired velocity of input control signal, which led to the radius of real trajectory larger than the desired one. Although steady-state error and oscillations exists, the linear and angular velocities are stable, the system is still robust to resist the disturbance in actual operation.

![Figure 4. Position of the virtual and actual robot (m).](image)

![Figure 5. Desired and real linear/angular velocities with error.](image)
S trajectory

The positions of the virtual and actual robot are shown in figure 6. The desired trajectory began from (0,0), and the curvature of S shape was 2m-1. In actual operation, the robot crossed the first arc with the curvature was 1.67m-1, the second curves was 1.54m-1, the third curves was 3.33m-1, the fourth curves was 2.5m-1, and the fifth curves was 2.22m-1. In figure 6, the green error line shows that the maximum position error was about 0.2m, and the maximum curvature error was 1.33m-1. Substantially, the output trajectory was in parallel with the input one, which was the steady-state error. The actual curvature radius of the first half section was less than the set value, and the actual curvature radius of the second half segment was greater than the set value. In addition, the actual running track was always on the right side of the setting trajectory. This phenomenon indicated that there was no oscillation of the system when the trajectory was self-correcting. Besides, the error reached the maximum in the middle of the route. The error from the first arc to the middle section was an increasing process, and the error between the middle and the end was a decreasing process, which meant the system had the ability to correct itself.

CONCLUSIONS

According to the simulation results, the conclusions are summaries as below:

1. Robot can track the desired input signal well under the robust control with sliding mode controller, and the simulation of the straight line trajectory was the best one;

2. The real trajectories were stable, while the output linear and angular velocities had slight oscillations. Oscillation was caused by the coupling of variables, the noise of the hardware and the interference of the external environment, which cannot be completely eliminated.

3. Sliding mode control overcomes the uncertainties of the rigid mechanical modeling and the friction modeling of the nonlinear system with the trajectory of the robot as the controlled object, and shows good robustness to the external disturbance and unmodeled dynamics.
ACKNOWLEDGEMENT

This project was financially supported by the National Natural Science Foundation of China (No. 51575117 and 61179051).

REFERENCES