基于模型预测控制的无人驾驶车辆轨迹跟踪问题研究

1. 研究背景

无人驾驶车辆运动规划与控制需要通过对车辆运动学或者动力学系统的控制来实现。建立合理的车辆系统模型不仅是设计模型预测控制器的前提，也是实现车辆道路跟踪功能的基础。因此，在建立模型预测控制器时，必须根据无人驾驶车辆的具体行驶工况，通过选取合适的控制变量，建立能够准确描述无人驾驶车辆运动关系约束的运动学模型。

车辆在地面运动的动力学过程是非常复杂的，为了尽量描述车辆运动，需要建立复杂的微分方程组，并用多个状态变量来描述其运动。用于模型预测控制的模型只要能够表现出车辆运动学与动力学约束，就可以使模型预测控制器实现预定控制目的。本文通过建立能够尽量准确反映车辆运动特性，并且有利于模型预测控制器设计的简化车辆运动学模型。在此基础上，对建立的模型进行验证。

2. 车辆运动学建模

车辆转向运动模型如图 2.1 所示。在惯性坐标系 OXY 下，（X_r，Y_r）和（X_f，Y_f）分别为车辆后轴和前轴轴心的坐标，φ为车体的航向角，δ_f为前轮偏角，v_r为车辆后轴中心速度，v_f为车辆前轴中心速度，l 为轴距。（下标 f 代表前轮，r 代表后轮。）

图 2.2 所示是车辆转向过程示意图，R 为后轮转向半径，P 为车辆的瞬时转动中心，M 为车辆后轴轴心，N 为前轴轴心。假设转向过程中车辆质心侧偏角保持不变，即车辆瞬时转向半径与道路曲率半径相同。

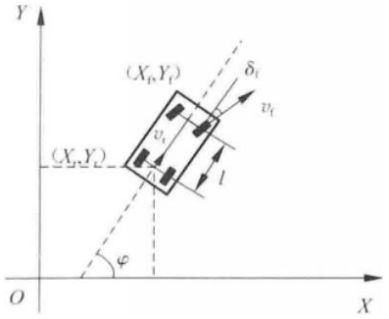

图 2.1 车辆运动模型

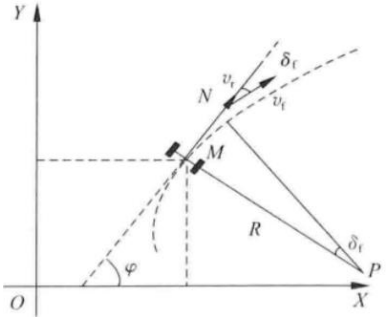

图 2.2 转向示意图

在后轴行驶轴心（X_r，Y_r）处，速度为：

$$v_r = \dot{X}_r \cos\varphi + \dot{Y}_r \sin\varphi \tag{2.1}$$

前、后轴的运动学约束为：

$$\begin{cases} \dot{X}_f \sin(\varphi + \delta_f) - \dot{Y}_f \cos(\varphi + \delta_f) = 0 \\ \dot{X}_r \sin(\varphi) - \dot{Y}_r \cos(\varphi) = 0 \end{cases} \tag{2.2}$$

由式（2.1）和式（2.2）。联合可得：

$$\begin{cases} \dot{X}_r = v_r \cos\varphi \\ \dot{Y}_r = v_r \sin\varphi \end{cases} \tag{2.3}$$

根据前后轮的几何关系可得：

$$\begin{cases} X_f = X_r + l cos\varphi \\ Y_f = Y_r + l sin\varphi \end{cases} \tag{2.4}$$

将式（2.3）和（2.4）代入（2.2），解得横摆角速度为：

$$\omega = \frac{v_r}{l} \tan\delta_f \tag{2.5}$$

式中，ω为车辆横摆角速度；同时，由ω和车速v_r可得到转向半径 R 和前轮偏角δ_f：

$$\begin{cases} R = \frac{v_r}{\omega} \\ \delta_f = \arctan(\frac{l}{R}) \end{cases} \tag{2.6}$$

由式（2.3）和式（2.5）可得到车辆运动学模型为：

$$\begin{bmatrix} \dot{X}_r \\ \dot{Y}_r \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} cos\varphi \\ sin\varphi \\ \tan(\delta_f / l) \end{bmatrix} v_r \tag{2.7}$$

该模型可被进一步表示为更为一般的形式：

$$\dot{\varepsilon}_{kin} = f_{kin}(\varepsilon_{kin}, u_{kin}) \tag{2.8}$$

其中，状态量$\varepsilon_{kin} = [X_r \quad Y_r \quad \varphi]^T$，控制量$u_{kin} = [v_r \quad \delta_f]^T$。在无人驾驶车辆的路径跟踪控制过程中，往往希望以$[v_r \quad \omega]$作为控制量，将式（2.5）代入式（2.7）中，该车辆运动学模型可以被转换为如下形式：

$$\begin{bmatrix} \dot{X}_r \\ \dot{Y}_r \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} cos\varphi \\ sin\varphi \\ 0 \end{bmatrix} v_r + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \tag{2.9}$$

上述的车辆运动学模型是非线性模型，表示为$\dot{\varepsilon}_{kin} = f_{kin}(\varepsilon_{kin}, u_{kin})$，我们需要对方程进行线性近似，得到线性时变的运动学模型。对上式的右边项在任意点

(ε_r, u_r)进行泰勒级数展开，只保留一次项，忽略高次项，得到：

$$\dot{\varepsilon} = f(\varepsilon_r, u_r) + \frac{\partial f}{\partial \varepsilon}(\varepsilon - \varepsilon_r) + \frac{\partial f}{\partial u}(u - u_r) \tag{2.10}$$

也可以写为：

$$\dot{\varepsilon} = f(\varepsilon_r, u_r) + J_f(\varepsilon)(\varepsilon - \varepsilon_r) + J_f(u)(u - u_r) \tag{2.11}$$

式中，$J_f(\varepsilon)$为 f 相对于ε的雅克比矩阵，$J_f(u)$为 f 相对于 u 的雅克比矩阵。

将式（2.11）与式（2.8）相减，得到：

$$\dot{\tilde{\varepsilon}} = A(t)\tilde{\varepsilon} + B(t)\tilde{u} \tag{2.12}$$

式中，$\tilde{\varepsilon} = \varepsilon - \varepsilon_r$，$\tilde{u} = u - u_r$，$A(t) = J_f(\varepsilon)$，$\mathrm{B(t)} = J_f(u)$。

由此，我们得到了新的状态方程，并且是线性时变模型。该方程是连续的，不能直接用于模型预测控制器的设计，需要对其进行离散化处理。我们采用近似离散化，即：

$$\begin{cases} A_{k,t} = I + TA(t) \\ B_{k,t} = TB(t) \end{cases} \tag{2.13}$$

结合式（2.12）和式（2.13），得到：

$$\tilde{\varepsilon}(k+1) = A_{k,t}\tilde{\varepsilon}(k) + B_{k,t}\tilde{u}(k) \tag{2.14}$$

至此，我们得到了非线性系统在任意一个参考点处线性化后的系统。该系统是设计线性模型预测控制算法的基础。我们将车辆的非线性运动学模型使用上述线性化方法得到一个离散线性时变状态方程：

$$\tilde{\varepsilon}_{kin}(k+1) = A_{kin}(k)\tilde{\varepsilon}_{kin}(k) + B_{kin}(k)\tilde{u}_{kin}(k) \tag{2.15}$$

其中，各矩阵和状态变量如下所示：

$$\tilde{\varepsilon}_{kin} = \begin{bmatrix} x - x_r \\ y - y_r \\ \varphi - \varphi_r \end{bmatrix}$$

$$A_{kin}(k) = \begin{bmatrix} 1 & 0 & -v_r sin\varphi_r T \\ 0 & 1 & v_r cos\varphi_r T \\ 0 & 0 & 1 \end{bmatrix}$$

$$B_{kin}(k) = \begin{bmatrix} cos\varphi_r T & 0 \\ sin\varphi_r T & 0 \\ \dfrac{\tan\delta_{f,r} T}{l} & \dfrac{v_r T}{l cos^2(\delta_{f,r})} \end{bmatrix}$$

上式中，T 为采样时间，k 为采样时刻，已证明，该非线性运动学模型完全

可控，因此，只要控制输入 u 不为 0 时，线性化之后的模型也是可控的。

3. 模型预测控制器设计

3.1 预测方程

首先，考虑以下的离散线性化模型：

$$x(k+1) = A_{k,t}x(k) + B_{k,t}u(k) \tag{3.1}$$

设定

$$\xi(k|k) = \begin{bmatrix} x(k|k) \\ u(k-1|k) \end{bmatrix} \tag{3.2}$$

可以得到一个新的状态空间表达式：

$$\begin{aligned} \xi(k+1|k) &= \tilde{A}_k\xi(k|k) + \tilde{B}_k\Delta u(k|k) \\ \eta(k|k) &= \tilde{C}_k\xi(k|k) \end{aligned} \tag{3.3}$$

式中各矩阵的定义如下：

$$\widetilde{\mathrm{A}}_k = \begin{bmatrix} \tilde{A}_k & \tilde{B}_k \\ 0_{m\times n} & I_m \end{bmatrix}$$

$$\tilde{B}_k = \begin{bmatrix} B_k \\ I_m \end{bmatrix}$$

$$\tilde{C}_k = [C_k \quad 0]$$

如果系统的预测时域为N_p，控制时域为N_c，那么，预测时域内的状态量和系统输出量可用以下算式计算：

$$\xi(k+N_p|k) = \tilde{A}_t^{N_P}\xi(k|k) + \tilde{A}_k^{N_P-1}\tilde{B}_k\Delta u(k|k) + \cdots + \tilde{A}_k^{N_P-N_c-1}\tilde{B}_k\Delta u(k+N_c|k) \tag{3.4}$$

$$\eta(k+N_p|k) = \tilde{C}_k\tilde{A}_k^{N_p}\xi(k|k) + \tilde{C}_k\tilde{A}_k^{N_p-1}\tilde{B}_k\Delta u(k|k) + \cdots + \tilde{C}_k\tilde{A}_k^{N_P-N_c-1}\tilde{B}_k\Delta u(k+N_c|k) \tag{3.5}$$

为了使整个关系更加明确，将系统未来时刻的输出以矩阵的形式表达：

$$\mathrm{Y}(k) = \psi_k\xi(k|k) + \Theta_k\Delta u(k) \tag{3.6}$$

式中：

$$\mathrm{Y}(k) = \begin{bmatrix} \eta(k+1|k) \\ \eta(k+2|k) \\ \cdots \\ \eta(k+N_c|k) \\ \cdots \\ \eta(k+N_p|k) \end{bmatrix}$$

$$\psi_k = \begin{bmatrix} \widetilde{C_k}\widetilde{A_k} \\ \widetilde{C_k}\widetilde{A_k}^2 \\ \cdots \\ \widetilde{C_k}\widetilde{A_k}^{N_c} \\ \cdots \\ \widetilde{C_k}\widetilde{A_k}^{N_p} \end{bmatrix}$$

$$\Delta \mathrm{U}(k) = \begin{bmatrix} \Delta U(k|k) \\ \Delta U(k+1|k) \\ \cdots \\ \Delta U(k+N_c|k) \end{bmatrix}$$

$$\Theta_k = \begin{bmatrix} \widetilde{C_k}\tilde{B}_k & 0 & 0 & 0 \\ \widetilde{C_k}\tilde{A}_k\tilde{B}_k & \widetilde{C_k}\tilde{B}_k & 0 & 0 \\ \cdots & \cdots & 0 & \cdots \\ \widetilde{C_k}\tilde{A}_k^{N_c-1}\tilde{B}_k & \widetilde{C_k}\tilde{A}_k^{N_c-2}\tilde{B}_k & \ddots & \widetilde{C_k}\tilde{B}_k \\ \widetilde{C_k}\tilde{A}_k^{N_c}\tilde{B}_k & \widetilde{C_k}\tilde{A}_k^{N_c-1}\tilde{B}_k & \cdots & \widetilde{C_k}\tilde{A}_k\tilde{B}_k \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{C_k}\tilde{A}_k^{N_p-1}\tilde{B}_k & \widetilde{C_k}\tilde{A}_k^{N_c-1}\tilde{B}_k & \cdots & \widetilde{C_k}\tilde{A}_k^{N_p-N_c-1}\tilde{B}_k \end{bmatrix}$$

通过式（3.6）可知，在预测时域内的状态量和输出量都可以通过系统当前的状态量$\xi(k|k)$和控制时域内的控制增量$\Delta \mathrm{U}(k)$计算得到。这也是模型预测控制算法中“预测”功能的实现。

3.2 优化求解

实际上，系统的控制增量是未知的，只有通过设定合适的优化目标，并对其进行求解，才能得到控制时域内的控制序列。我们采取的目标函数如下：

$$\phi(\mathrm{k}) = \sum_{j=1}^{N} \check{X}^T(k+j|k) Q \check{X}(k+j|k) + \tilde{u}^T(k+j-1|k) R \tilde{u}(k+j-1|k) \quad （3.7）$$

其中，Q 和 R 分别是状态量和控制量的权重矩阵，整个表达式的功能是使系统能够尽快且平稳的跟踪上期望轨迹。我们对上式进一步优化，将对控制增量的约束引入到目标函数，避免控制量跳变对系统性能的影响：

$$\mathrm{J}(\xi(\mathrm{k}), \mathrm{u}(\mathrm{k}-1), \Delta \mathrm{u}(\mathrm{k})) = \sum_{i=1}^{N_P} \left\| \eta(k+i|k) - \eta_{ref}(k+i|k) \right\|_Q^2 + \sum_{i=1}^{N_c-1} \| \Delta \mathrm{u}(k+i|k) \|_R^2 \quad （3.8）$$

其中，第一项反映了系统对参考轨迹的跟随能力，第二项反映了对控制量平稳变化的要求。同时，在实际控制系统中，往往需要满足系统状态量以及控制量的一些约束条件，一般如下：

控制量约束：

$$u_{min}(k+i) \le u(k+i) \le u_{max}(k+i), i = 0,1,\cdots N_c - 1 \quad （3.9）$$

控制增量约束：

$$\Delta u_{min}(k+i) \le \Delta u(k+i) \le \Delta u_{max}(k+i), i = 0,1,\cdots N_c - 1 \quad （3.10）$$

对于以上形式的优化目标，可以通过适当的处理将其转换为二次规划问题。

二次规划是典型的数学优化问题。将式（3.8）写成矩阵的形式：

$$\Phi(\mathrm{k}) = \frac{1}{2}\Delta U^T(k)H(k)\Delta U(k) + f^T(k)\Delta U(k) + d(k) \qquad (3.11)$$

上式中，

$$H(k) = \Theta_k^T Q \Theta_k + R$$

$$f^T(k) = 2\mathrm{E}(\mathrm{k})^T Q \Theta_k$$

$$d(k) = \mathrm{E}(\mathrm{k})^T Q \mathrm{E}(\mathrm{k})$$

$$\mathrm{E}(\mathrm{k}) = \psi_k \xi(k|k) - Y(k)$$

3.3 反馈机制

在每个控制周期内完成对式（3.11）的求解后，得到了控制时域内的一系列控制输入增量：

$$\Delta U_t^* = \left[\Delta u_t^*, \Delta u_{t+1}^*, \cdots, \Delta u_{t+N_c-1}^*\right]^T \qquad (3.12)$$

根据模型预测控制的基本原理，将该控制序列中第一个元素作为实际的控制输入增量作用于系统，即：

$$\mathrm{u}(\mathrm{t}) = \mathrm{u}(\mathrm{t}-1) + \Delta u_t^* \qquad (3.13)$$

系统执行这一控制量直到下一时刻。在新的时刻，系统根据状态信息重新预测下一段时域的输出，通过优化过程得到一个新的控制增量序列。如此循环往复，直到系统完成控制过程。

4. 仿真实验

无人驾驶车辆在一个给定位置出发，通过离散轨迹点或者连续轨迹函数的指引，最终跟踪上期望轨迹。轨迹跟踪仿真基本设定为：车辆从坐标原点出发，以期望纵向速度 v=1m/s 跟踪一条直线 y=2。采样时间为 50ms，仿真时间设定为 20s，在 MATLAB 环境下对无人驾驶车辆的直线轨迹跟踪过程进行仿真。

利用第三部分设计的控制器，在 MATLAB 中编写相关程序，得到仿真结果如图 4.1 所示，仿真实验可大致分为四个部分：参考轨迹生成、仿真系统变量初始化、系统矩阵变量定义和控制器设计。

首先，生成参考轨迹，本文选取直线 y=2 作为参考轨迹进行跟踪，该部分的程序如下：

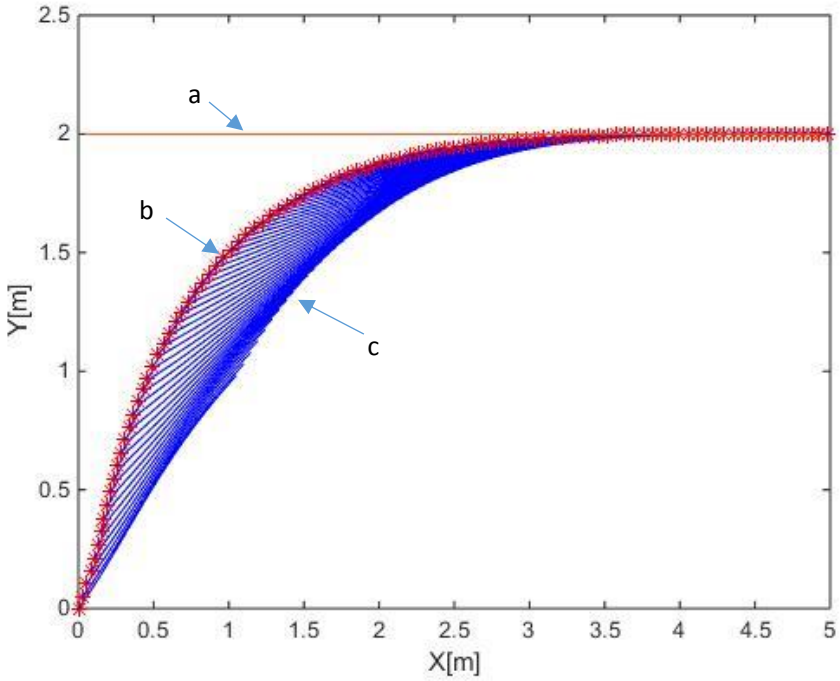

图 4.1 轨迹跟踪结果

```
%第一部分
% 参考轨迹生成
N=100;    %参考轨迹点数量
T=0.05;   %采样周期
Xout=zeros(N,3);      %N 行 3 列矩阵
Tout=zeros(N,1);      %N 行 1 列矩阵
for k=1:1:N
    Xout(k,1)=k*T;
    Xout(k,2)=2;
    Xout(k,3)=0;
    Tout(k,1)=(k-1)*T;
end
```

第二部分是系统变量初始化，包括状态量和控制量的个数及其初始值设定，初始状态等。相关程序如下：

```
% 第二部分
%仿真系统基本情况介绍
```

```
Nx=3;       %状态量个数
Nu=2;       %控制量个数
[Nr,Nc]=size(Xout);   %返回 Xout 的行数和列数
Tsim=20;     %仿真时间
X0=[0 0 pi/3];   %车辆初始状态
L=1;                %车辆轴距
vd1=1;              %参考系统的纵向速度
vd2=0;              %参考系统的前轮偏角
```

第三部分是根据控制系统的维度信息，提前定义好相关矩阵并赋值。相关程序如下：

```
%第 3 部分
%矩阵定义
x_real=zeros(Nr,Nc);
x_piao=zeros(Nr,Nc);
u_real=zeros(Nr,Nu);
u_piao=zeros(Nr,Nu);
x_real(1,:)=X0;
x_piao(1,:)=x_real(1,:)-Xout(1,:);
X_PIAO=zeros(Nr,Nx*Tsim);
XXX=zeros(Nr,Nx*Tsim);     %用于保存每个时刻预测的所有状态值
q=[1 0 0;0 1 0;0 0 0.5];
Q_cell=cell(Tsim,Tsim);
for i=1:1:Tsim
    for j=1:1:Tsim
        if i==j
            Q_cell{i,j}=q;
        else
            Q_cell{i,j}=zeros(Nx,Nx);
        end
    end
```

```
end
Q=cell2mat(Q_cell);
R=0.1*eye(Nu*Tsim,Nu*Tsim);
```

第四部分是模型预测控制的主体，在每一控制周期的开始，获取系统当前状态量，更新状态空间方程。根据目标函数式（3.11），求解一个标准二次规划问题，最后将序列求解的第一个解施加到系统上。相关程序如下：

```
%第四部分
%模型预测控制
for i=1:1:Nr
    t_d=Xout(i,3);
    a=[1 0 -vd1*sin(t_d)*T;
       0 1 vd1*cos(t_d)*T;
       0 0 1];
    b=[cos(t_d)*T 0;
       sin(t_d)*T 0;
       vd2*T/L vd1*T/(cos(vd2)^2)];
    A_cell=cell(Tsim,1);
    B_cell=cell(Tsim,Tsim);
    for j=1:1:Tsim
        A_cell{j,1}=a^j;
        for k=1:1:Tsim
            if k<=j
                B_cell{j,k}=(a^(j-k))*b;
            else
                B_cell{j,k}=zeros(Nx,Nu);
            end
        end
    end
    A=cell2mat(A_cell);
    B=cell2mat(B_cell);
```

```
H=2*(B'*Q*B+R);
f=2*B'*Q*A*x_piao(i,:)';
A_cons=[];
b_cons=[];
lb=[-0.2;-0.64];
ub=[0.2;0.64];
[X,fval(i,1),exitflag(i,1),output(i,1)]=quadprog(H,f,A_cons,b_cons,[],[],lb,ub);
X_PIAO(i,:)=(A*x_piao(i,:)'+B*X)';
if i+j<Nr
    for j=1:1:Tsim
        XXX(i,1+3*(j-1))=X_PIAO(i,1+3*(j-1))+Xout(i+j,1);
        XXX(i,2+3*(j-1))=X_PIAO(i,2+3*(j-1))+Xout(i+j,2);
        XXX(i,3+3*(j-1))=X_PIAO(i,3+3*(j-1))+Xout(i+j,3);
    end
else
    for j=1:1:Tsim
        XXX(i,1+3*(j-1))=X_PIAO(i,1+3*(j-1))+Xout(Nr,1);
        XXX(i,2+3*(j-1))=X_PIAO(i,2+3*(j-1))+Xout(Nr,2);
        XXX(i,3+3*(j-1))=X_PIAO(i,3+3*(j-1))+Xout(Nr,3);
    end
end
u_piao(i,1)=X(1,1);
u_piao(i,2)=X(2,1);
Tvec=[0:0.05:4];
X00=x_real(i,:);
vd11=vd1+u_piao(i,1);
vd22=vd2+u_piao(i,2);
XOUT=dsolve('Dx-vd11*cos(z)=0','Dy-vd11*sin(z)=0','Dz-
vd22=0','x(0)=X00(1)','y(0)=X00(2)','z(0)=X00(3)');
```

```
        t=T;
        x_real(i+1,1)=eval(XOUT.x);
        x_real(i+1,2)=eval(XOUT.y);
        x_real(i+1,3)=eval(XOUT.z);
        if i<Nr
            x_piao(i+1,:)=x_real(i+1,:)-Xout(i+1,:);
        end
        u_real(i,1)=vd1+u_piao(i,1);
        u_real(i,2)=vd2+u_piao(i,2);

        figure(1);
        plot(Xout(1:Nr,1),Xout(1:Nr,2));
        hold on;
        plot(x_real(i,1),x_real(i,2),'r*');
        xlabel('X[m]');
        ylabel('Y[m]');
        hold on;
        for k=1:1:Tsim
            X(i,k+1)=XXX(i,1+3*(k-1));
            Y(i,k+1)=XXX(i,2+3*(k-1));
        end
        X(i,1)=x_real(i,1);
        Y(i,1)=x_real(i,2);
        plot(X(i,:),Y(i,:),'b')
        hold on;
    end
```

得到仿真结果如图 4.1 所示，图中曲线 a 为期望轨迹，星号曲线 b 为车辆位置，曲线 c 为预测时域内车辆位置。图中可以看出，车辆在模型预测控制器的作用下快速跟踪上了期望轨迹，最后能够沿着期望轨迹稳定行驶，达到了预期效果；

图中车辆实际行驶的曲线 b 与预测模型输出的位置曲线 c 并不重合，主要是因为采用了线性误差模型作为预测模型，而车辆位置的解采用的是非线性运动学模型。因为预测模型往往不是对被控系统的精确建模，而是对系统的合理简化。

仿真过程中，系统的状态量随时间变化的曲线如图 4.2 所示，控制量随时间的变化曲线如图 4.3 所示。

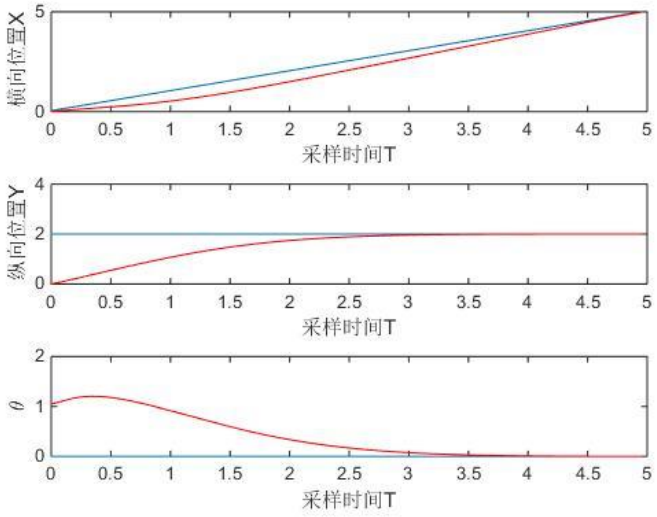

图 4.2 系统状态量随时间变化曲线

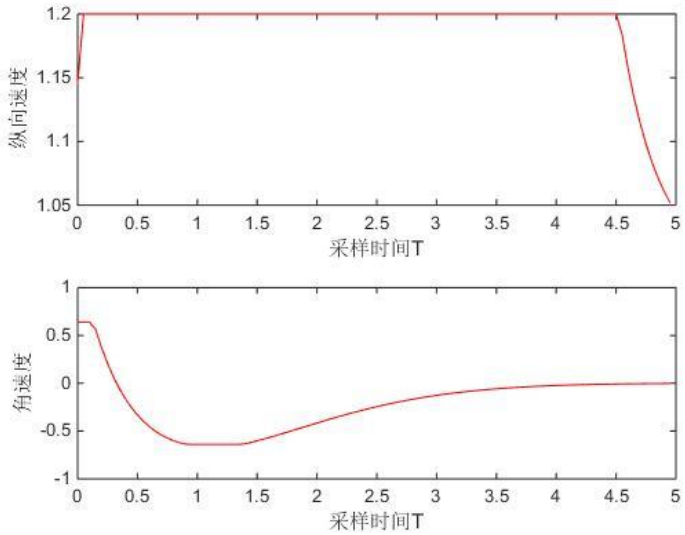

图 4.3 控制量随时间变化曲线