Trajectory Tracking Control Algorithm for a Wheeled Mobile Robot

Aishashwini, Abhinav Aggarwal, Vardhaman Jain, Gaurav Kumar, Shresth Grover

WMRs (wheeled mobile robots) are a type of underactuated mechanical device that may be controlled. They're becoming more used in a variety of industrial applications that require high velocity and low energy usage. The tracking control to a desired trajectory is one of the WMR control challenges that frequently attracts researchers' attention. Adaptive backstepping, block backstepping, and adaptive sliding mode control have all been proposed as nonlinear control approaches for the research and application of wheeled mobile robots in recent years. A recently developed trajectory tracking control system for a wheeled mobile robot is presented in this study, which includes hierarchical sliding mode and backstepping control algorithms. The closed-loop stability and zero tracking error are guaranteed by the backstepping hierarchical sliding mode tracking control. The method's usefulness and aptitude for practical applications are demonstrated by numerical simulation results.

I. Introduction

Several control strategies have been used for the stabilisation, regulation, and control of linear and nonlinear dynamical systems in recent decades. It is simple to find a control Lyapunov function for stability and optimization problems in linear autonomous systems. Finding a proper control Lyapunov function, on the other hand, is a difficult task for nonlinear control systems. The backstepping control technique is a recursive design approach that links the selection of a control Lyapunov function with the design of a feedback controller, ensuring tight feedback systems' global asymptotic stability. In the control literature, the active backstepping control method is a useful tool for overcoming the limits of the feedback linearization approach. The block backstepping control technique is a more widely applicable backstepping control approach in the control literature. The adaptive backstepping control technique is a variant of the backstepping control method that use estimations for unknown system characteristics. For control systems with uncertainties, the robust backstepping control approach is an effective backstepping strategy. SMC (sliding mode control) is a nonlinear control method in control systems that modifies the dynamics of a nonlinear system by applying a discontinuous control signal (or, more precisely, a set-valued control signal) that causes the system to "slide" along a cross-section of the system's normal behaviour. The state-feedback control law is not a time-dependent function. Instead, depending on where it is in the state space, it can move from one continuous structure to another. As a result, sliding mode control is a way of variable structure control. Trajectories are continually moving toward a neighbouring zone with a different control structure, hence the eventual trajectory will not reside wholly inside one control structure. Instead, it will glide along the control structures' limits. The sliding mode is the motion of the system as it slides along these limits, and the sliding (hyper)surface is the geometrical locus containing the boundaries. In the context of modern control theory, any variable structure system, like a system under SMC, may be viewed as a special case of a hybrid dynamical system as the system both flows through a continuous state space but also moves through different discrete control modes. Any variable structure system, such as one under SMC, may be considered as a particular instance of a hybrid dynamical system in the framework of current control theory since it flows through a continuous state space while also moving through multiple discrete control modes.

II. System Dynamics

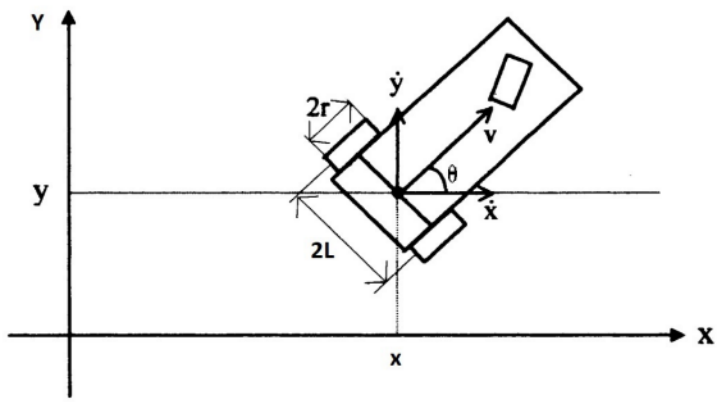

Fig. 1 Wheeled Mobile Robot

Let the position and orientation of the the robot be defined by $q = [x, y, \theta]^T$. Then using Euler-Lagrange equation, the dynamics can be written as:

$$M(q)\ddot{q} = C(q,\dot{q})\dot{q} = E(q)T - A^T(q)\lambda \tag{1}$$

where

$$M(q) = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J \end{bmatrix}, C(q,\dot{q}) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, T = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} \tag{2}$$

$$E(q) = \frac{1}{r}\begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ L & -L \end{bmatrix} \text{ and } A^T(q) = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} \tag{3}$$

Using Lagrangian multiplier,

$$\lambda = -m\dot{\theta}(\dot{x}\cos\theta + \dot{y}\sin\theta) \tag{4}$$

along with nonholonomic constraints gives the dynamic model of the system:

$$\ddot{\theta} = b_1 u_1 \tag{5}$$
$$\ddot{x} = \frac{\lambda}{m}\sin\theta + b_2 u_2 \cos\theta \tag{6}$$
$$\ddot{y} = -\frac{\lambda}{m}\cos\theta + b_2 u_2 \sin\theta \tag{7}$$

where $b_1 = \frac{L}{rJ}$, $b_2 = \frac{1}{rm}$ are constants and $u_1 = T_1 - T_2$ and $u_2 = T_1 + T_2$ are the control inputs.

III. Control Algorithm

The dynamical equation is separated into two subsystems, with the first depending only on control input u_1 and the second subsystem depends on control input u_2. We use backstepping controller for subsystem (1) and hierarchical sliding mode control.

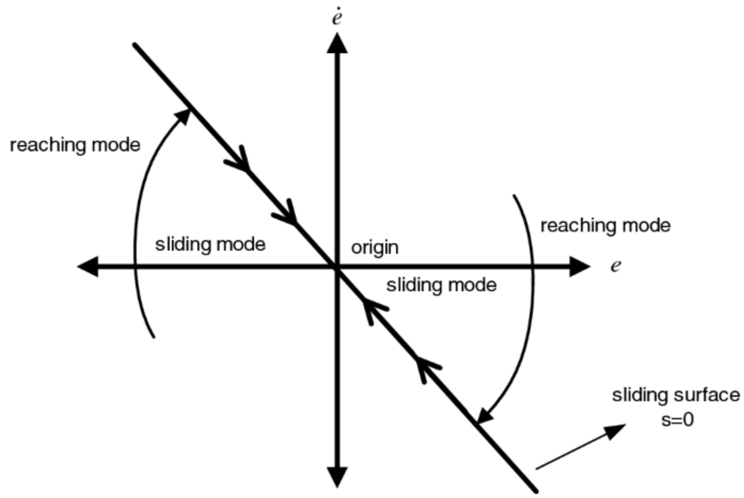

Fig. 2 Sliding mode Control

A. Back-stepping Control

Let $q_r = [x_r, y_r, \theta_r]^T$ be the reference posture for the robot (position and orientation). Let the tracking error be,

$$E = [e_\theta, e_x, e_y]^T = [\theta - \theta_r, x - x_r, y - y_r]^T \tag{8}$$

Define new state variables:

$$z_1 = e_\theta \tag{9}$$

$$z_2 = \dot{\theta} + \alpha_1 \tag{10}$$

where α_1 is a virtual control input to the subsystem. Let a Lyapunov function candidate:

$$V_{11} = \frac{1}{2} z_1^2 \tag{11}$$

Thus,

$$\dot{V}_{11} = z_1 \dot{z}_1 = z_1 (z_2 - \alpha_1 - \dot{\theta}_r) \tag{12}$$

Let $\alpha_1 = a_1 z_1 - \dot{\theta}_r$, where $a_1 > 0$ then

$$\dot{V}_{11} = -a_1 z_1^2 + z_1 z_2 \tag{13}$$

Next define the control Lyapunov function,

$$V_1 = V_{11} + \frac{1}{2}z_2^2 \tag{14}$$

Taking derivative

$$\dot{V}_1 = \dot{V}_{11} + z_2\dot{z}_2 = -a_1z_1^2 + z_1z_2 + z_2\dot{z}_2 \tag{15}$$

$$= -a_1z_1^2 + z_2(z_1 + b_1u_1 + \dot{\alpha}_1) \tag{16}$$

Using the the control law:

$$u_1 = -\frac{1}{b_1}(z_1 + a_2z_2 + \dot{\alpha}_1) \tag{17}$$

where $a_2 > 0$, we get

$$\dot{V}_1 = -a_1z_1^2 - a_2z_2^2 < 0, \forall z_1, z_2 \neq 0 \tag{18}$$

Hence the control law makes θ asymptotically converge to θ_r.

B. Hierarchical Sliding Mode Control

The second subsystem is given by:

$$\ddot{x} = \frac{\lambda}{m}\sin\theta + b_2u_2\cos\theta \tag{19}$$

$$\ddot{y} = -\frac{\lambda}{m}\cos\theta + b_2u_2\sin\theta \tag{20}$$

Define new state variables,

$$X = [x_1, x_2, x_3, x_4]^T = [x, \dot{x}, y, \dot{y}]^T \tag{21}$$

Now the state-space representation of the subsystem can be written as

$$\dot{x}_1 = x_2 \tag{22}$$

$$\dot{x}_2 = f_1(X) + g_1(X)u_2 \tag{23}$$

$$\dot{x}_3 = x_4 \tag{24}$$

$$\dot{x}_4 = f_2(X) + g_2(X)u_2 \tag{25}$$

where $f_1(X) = \frac{\lambda}{m}\sin\theta$, $g_1(X) = b_2\cos\theta$, $f_2(X) = -\frac{\lambda}{m}\cos\theta$, $g_2(X) = b_2\sin\theta$. These equation can be transformed into equivalent tracking error state space representation:

$$\dot{e}_1 = e_2 \tag{26}$$

$$\dot{e}_2 = f_1(X) + g_1(X)u_2 - \ddot{x}_r \tag{27}$$

$$\dot{e}_3 = e_4 \tag{28}$$

$$\dot{e}_4 = f_2(X) + g_2(X)u_2 - \ddot{y}_r \tag{29}$$

where $e_1 = e_x$ and $e_3 = e_y$. Let us now design control. Consider the first subsystem:

$$\dot{e}_1 = e_2 \tag{30}$$
$$\dot{e}_2 = f_1(X) + g_1(X)u_{21} - \ddot{x}_r \tag{31}$$
$$\tag{32}$$

where u_{21} acts as a virtual control input to the subsystem. Define the first sliding surface as

$$S_1 = c_1 e_1 + e_2 \tag{33}$$

with $c_1 > 0$. Taking time derivative,

$$\dot{S}_1 = c_1 \dot{e}_1 + \dot{e}_2 \tag{34}$$
$$= c_1 e_2 + f_1 + g_1 u_{21} \tag{35}$$
$$= c_1 e_2 + f_1 + g_1(u_{eq21} + u_{sw21}) \tag{36}$$

where u_{eq21} and u_{sw21} are the equivalent and switching control components of the virtual control input u_{21}, respectively. Let us select these control input as

$$u_{eq21} = \frac{-c_1 \dot{e}_1 - f_1}{g_1} \tag{37}$$
$$u_{sw21} = \frac{-k_1 S_1 + \eta sgn(S_1) - \ddot{x}_r}{g_1} \tag{38}$$

where $k_1 > 0$ and $\eta > 0$ are the design parameters and $sgn(S_1)$ is the signum function of S_1. Substituting these above,

$$\dot{S}_1 = -k_1 S_1 - \eta_1 sgn(S_1) \tag{39}$$

Similarly choose the second sliding surface as

$$S_2 = \lambda_1 S_1 + \beta_1 s_2 \tag{40}$$

where $s_2 = c_2 e_3 + e_4$ with $c_2 > 0$. Next taking time derivative,

$$\dot{S}_2 = \lambda_1(c_1 e_2 + f_1 + g_1 u_2 - \ddot{x}_r) + \beta_1(c_2 e_4 + f_2 + g_2 u_2 - \ddot{y}_r) \tag{41}$$
$$= \lambda_1(c_1 e_2 + f_1 + g_1(u_{21} + u_{eq22} + u_{sw22}) - \ddot{x}_r) + \beta_1(c_2 e_4 + f_2 + g_2(u_{21} + u_{eq22} + u_{sw22}) - \ddot{y}_r) \tag{42}$$
$$= -k_2 S_2 - \eta_2 sgn(S_2) \tag{43}$$

where we have chosen ,

$$u_{eq22} = \frac{-c_2 e_4 + f_2}{g_2} \tag{44}$$
$$u_{sw22} = -u_{sw21} - \frac{\lambda_1 g_1 u_{eq22} + \beta_1 g_2 u_{eq21}}{\lambda_1 g_1 + \beta_1 g_2} - \frac{k_2 S_2 + \eta_2 sgn(S_2) - \lambda_1 \ddot{x}_r - \beta_1 \ddot{y}_r}{\lambda_1 g_1 + \beta_1 g_2} \tag{45}$$

with $k_2 > 0$ and $\eta_2 > 0$ are design parameters. Finally the control law is given by,

$$u_2 = -\frac{\lambda_1 f_1(X) + \beta_1 f_2(X) + \lambda_1 c_1 e_2 + \beta_1 c_2 e_4}{\lambda_1 g_1(X) + \beta_1 g_2(X)} - \frac{\eta_2 sat(S_2) + k_2 S_2 - \lambda_1 \ddot{x}_r - \beta_1 \ddot{y}_r}{\lambda_1 g_1(X) + \beta_1 g_2(X)} \tag{46}$$

where signum function is replaced by saturation function defined by

$$s = \begin{cases} sgn(s), |s| > 1 \\ s, |s| \leq 1 \end{cases} \tag{47}$$

C. Stability Proof

Theorem 1. *The dynamic model is stabilized by the selecting the control law as*

$$T = [T_1, T_2]^T = \left[\frac{u_1 + u_2}{2}, \frac{u_1 - u_2}{2}\right]^T \tag{48}$$

Proof. Let V be a positive definite function

$$V = \frac{z_1^2}{2} + \frac{z_2^2}{2} + \frac{S2^2}{2} \tag{49}$$

Taking time derivative,

$$\dot{V} = -a_1 z_1^2 - a_2 z_2^2 - S_2 \dot{S}_2 \tag{50}$$

$$= -a_1 z_1^2 - a_2 z_2^2 - k_2 S_2^2 - \eta_2 S_2 sgn(S_2) \tag{51}$$

which is negative definite. Hence the system is asymptotic stable in Lyapunov sense. □

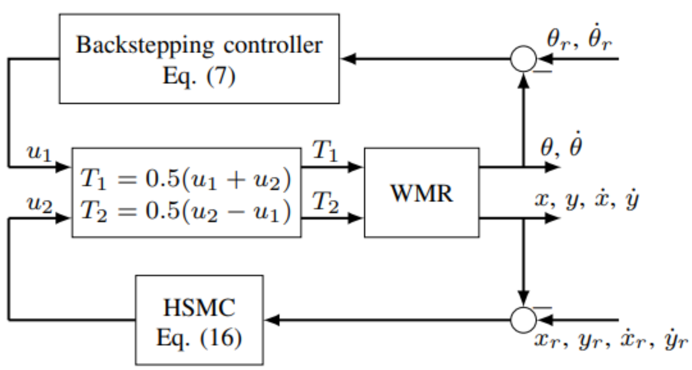

Fig. 3 Control Block Diagram

IV. Simulation

We simulate the system model using the derived control in MATLAB. The simulation parameters are: $m = 1.038$ kg, $J = 0.818$ kg/m2, $r = 0.025$ m and $L = 0.075$ m. The design parameters are: The initial conditions is $q_0 = [1, 0, \pi/2]^T$ and the WMR is to track a circular trajectory. It is shown from the figures that the WMR follows the desired sinusoidal and circle trajectories with small settling time. The x- and y-axis

Table 1 Simulation Parameters and Initial Conditions

Parameter	Value
b_1	2
a_2	4
c_1	75
c_2	75
k_2	5
a_1	120
η_2	5
β_1	200

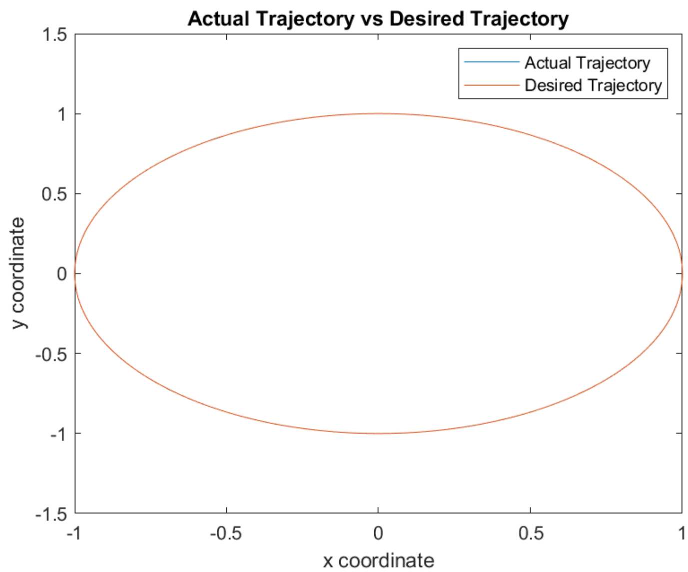

Fig. 4

V. Conclusion

A synthesis technique for a trajectory tracking control algorithm for wheeled mobile robots was proposed in this study. The suggested tracking control approach, which combines a backstepping controller with a hierarchical sliding controller, has demonstrated its usefulness in practise. The design approach is straightforward, but the controller implementation requires an accurate model of the WMR and measurable state variables. The backstepping hierarchical sliding control system is being extended to output feedback in a current study. When the precise WMR model is known, a synthesis technique is to use an observer, and when the WMR model has unknown parameters, to use an adaptive control strategy.

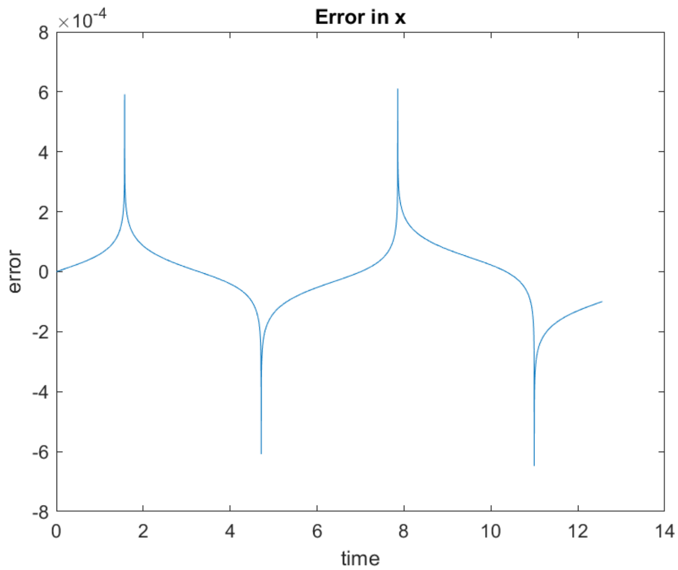

Fig. 5

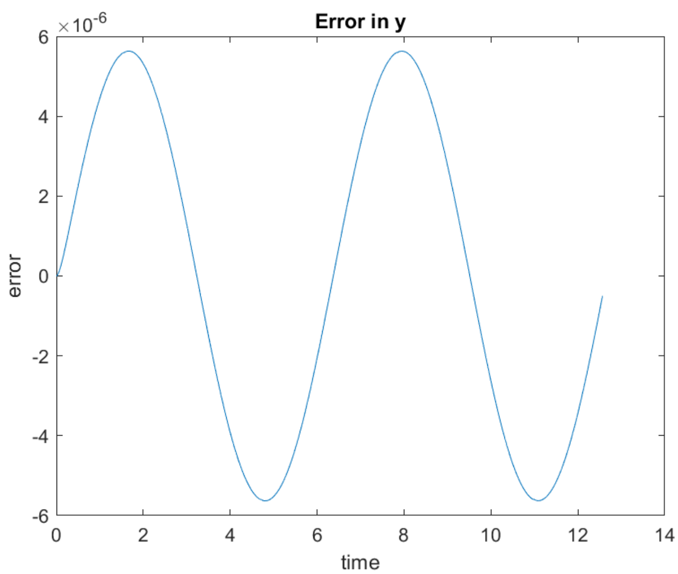

Fig. 6

References

[1] P. T. H. Sen, N. Q. Minh, D. T. T. Anh, D. T. T. Anh and P. X. Minh, "A New Tracking Control Algorithm for a Wheeled Mobile Robot Based on Backstepping and Hierarchical Sliding Mode Techniques," 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), 2019, pp. 25-28, doi: 10.1109/ICA-SYMP.2019.8646288.

[2] S. Rudra, R. K. Barai, and M. Maitra, "Design and implementation of a block backstepping based tracking control for nonholonomic wheeled mobile robot," Int. J. Robust and Nonlinear Control, vol. 26, pp. 30183035, 2016.

[3] Le Xuan Hai, Nguyen Van Thai, Vu Thi Thuy Nga, Hoang Thi Tu Uyen, Nguyen Thanh Long, Thai Huu Nguyen, Phan Xuan Minh, "High order sliding mode control with anti-sway based compensation on artificial neural network by PSO algorithm for overhead crane," Vietnam J. Science and Technology, vol. 55, no. 3, pp. 347356, 2017.

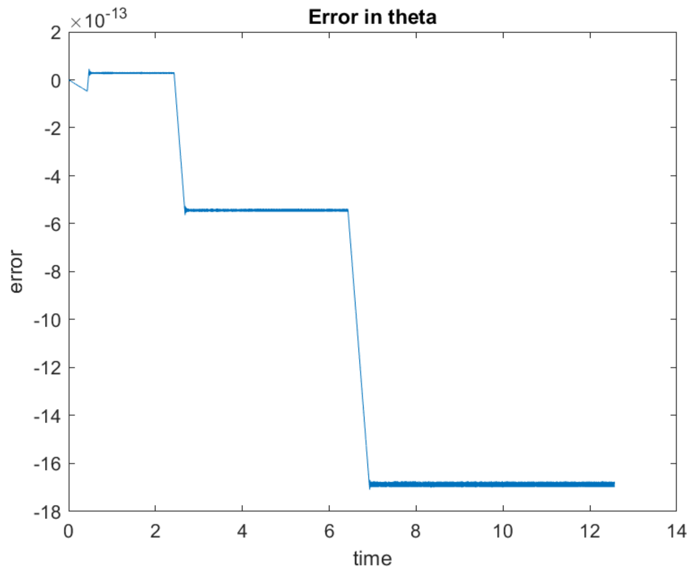

Fig. 7

[4] D. Quian, J. Yi, and D. Zhao, “Hierarchical sliding mode control for a class of SIMO under-actuated systems,” Control and Cybernetics, vol. 37, no. 1, pp. 159175, 2008.

[5] K. Shojaei, M. M. Shahri, A. T. Tarakameh, and B. Tabibian, “Adaptive trajectory tracking control of a differential drive wheeled mobile robot,” Robotica, vol. 29, pp. 391402, 2011.

```
% For cicular Trajectory


% Firstly Backstepping control for u1
omega =  1;
m = 1.038;
T = 2*pi/omega;
dt =0.0001;
t = 0:dt:2*T;
% For reference trajectory
theta_r = (pi/2 + omega*t);
a1 = 120;
a2= 4;
R=1;
b2 = 1/(0.025*m);
theta_r_dot = omega;
theta_r_ddot = 0;
f = theta_r_ddot +(a1+a2)*theta_r_dot +(1+a1*a2)*(theta_r);
% for actual trajectory
theta = zeros(1,length(t));
theta_dot = zeros(1,length(t));
theta(1)= pi/2;
theta_dot(1)= omega;
for i =1:length(t)-1

    theta(i +1) = theta(i) + theta_dot(i)*dt;
```

```
    theta_dot(i+1) = (f(i) -(a1+a2)*theta_dot(i) - (1+a1*a2)*theta(i))*dt + theta_dot(i);

end
% Now heirarchical sliding mode control for u2 to conrol x and y

x_r = R*cos(omega*t);
y_r = R*sin(omega*t);
x_dot_r = -R*sin(omega*t)*omega;
y_dot_r = R*cos(omega*t)*omega;
x_ddot_r = -R*cos(omega*t)*omega^2;
y_ddot_r = -R*sin(omega*t)*omega^2;
x=zeros(1,length(t));
y=zeros(1,length(t));
x_dot=zeros(1,length(t));
y_dot=zeros(1,length(t));
x(1)=1;
y_dot(1)=omega*R;
for i=1:length(t)-1
    x(i+1) =  x(i) + x_dot(i)*dt;
    y(i+1) =  y(i) + y_dot(i)*dt;
    lambda = -m*theta_dot(i)*(x_dot(i)*cos(theta(i))+y_dot(i)*sin(theta(i)));
    e_1 = x(i)-x_r(i);
    e_2 = x_dot(i)-x_dot_r(i);
    e_3 = y(i)-y_r(i);
    e_4 = y_dot(i) - y_dot_r(i);
    u = SlidingModeControl(0.01,lambda,theta(i),e_4,e_1,e_2,e_3,x_ddot_r(i),y_ddot_r(i));
    x_dot(i+1) = (lambda/m*sin(theta(i))+b2*u*cos(theta(i)))*dt+x_dot(i);
    y_dot(i+1) = (-1*lambda/m*cos(theta(i))+b2*u*sin(theta(i)))*dt+y_dot(i);
end
plot(x,y,'DisplayName','Actual Trajectory')
hold on
plot(x_r,y_r,'DisplayName','Desired Trajectory')
legend
xlabel('x coordinate')
ylabel('y coordinate')
title("Actual Trajectory vs Desired Trajectory")
hold off

plot (t,(x-x_r))
xlabel('time')
ylabel('error')
title("Error in x")

plot (t,(y-y_r))
xlabel('time')
ylabel('error')
title("Error in y")

plot (t,(theta-theta_r))
xlabel('time')
ylabel('error')
title("Error in theta")
```

```
function[u2] = SlidingModeControl(lambda1,lambda,theta,e4,e1,e2,e3,x__r,y__r)
m=1.038;
beta1= 200;
b2 = 1/(0.025*m);
c1= 75;
c2= 75;
eta2=5;
k2 =5;
S1 = c1*e1+e2;
S2 = lambda1*S1+beta1*(c2*e3+e4);
u2 = -(lambda1*(lambda/m)*sin(theta)+beta1*(-lambda/m)*cos(theta)+
lambda1*c1*e2+beta1*c2*e4)/(lambda1*b2*cos(theta)+beta1*b2*sin(theta))-
(eta2*sat(S2)+k2*S2-lambda1*x__r-beta1*y__r)/(lambda1*b2*cos(theta)+beta1*b2*sin(theta));
end

function [y] = sat(x)
s_mod = abs(x);
if (s_mod>1)
    y = sign(x);
else
    y = x;
end

end
```