

《系统建模与仿真》 项目报告

题目： 四轮小车运动学建模与仿真

成员：	姓名	学号
	<u>韩骐骏</u>	<u>12212635</u>
	<u>林志芑</u>	<u>12110933</u>

系统设计与智能制造学院

2024 年 6 月

目录

1 项目内容与要求	1
1.1 背景	1
1.2 内容	2
1.3 要求	3
1.3.1 任务一	3
1.3.2 任务二	3
1.4 期望目标	4
2 任务实现	4
2.1 逆运动学理论推导	4
2.1.1 整车的运动分解	4
2.1.2 轮子角速度的表示	5
2.1.3 逆运动学结果表示	5
2.2 顺运动学推导	6
2.3 建模与仿真	6
2.3.1 确定仿真模块的框架	6
2.3.2 编写模块内部函数	7
2.3.3 仿真模块与轨迹绘制	8
3 分析与总结	10
3.1 任务一	10
3.1.1 逆运动学公式	10
3.1.2 顺运动学公式	11
3.2 任务二	11
4 参考文献	11
5 附件	12
5.1 myconsole.m 代码	12
5.2 Simulink 仿真模块	14

5.3 其余代码及运行结果	14
---------------------	----

项目名称：四轮小车的运动学建模与仿真

1 项目内容与要求

1.1 背景

麦克纳姆轮是一种能够实现全方向移动的全向轮，如图 1-1 所示，麦克纳姆轮的基本结构由轮毂和围绕轮毂的辊子组成，辊子的轴线与轮毂轴线呈一定角度，通常为 45° ，这些辊子可以把轮子的转动向纵向和横向两个方向分解，使机器人或车辆在水平面上实现全向移动和旋转而不需要改变轮子自身的方向。

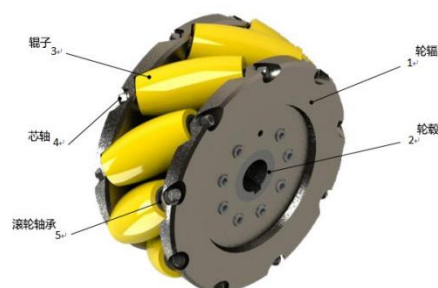


图 1-1

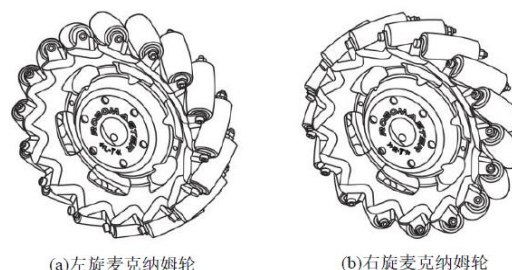


图 1-2

如图 1-2 所示，麦克纳姆轮有左旋和右旋两种形式，二者呈手性对称关系，须成套使用。

本项目中，麦克纳姆轮小车使用 O 型布局，这是一种能实现底盘全向运动及旋转的布局方式，是最常用的一种麦轮布局方式，O 型指辊子与地面接触所形成的形状，而俯视底盘看到的辊子倾斜方向与接触地面的小辊子偏置方向相反。图 3 所示即本项目中所要分析的麦克纳姆轮底盘及其布局示意图。



图 1-3

1.2 内容

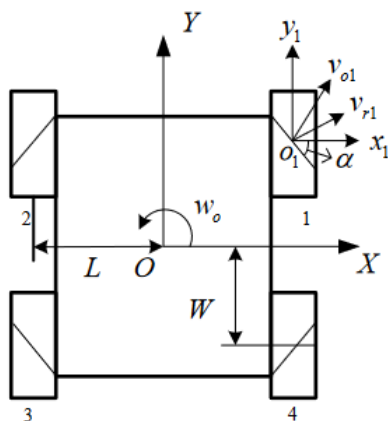


图 1-4

建立如图 1-4 所示的麦克纳姆轮小车运动学模型，建立以小车中心为原点的直角坐标系 $O - XY$ ，以右上方的麦克纳姆轮组 1 为例进行分析，建立以轮心 o_1 为原点的直角坐标系 $o_1 - x_1y_1$ ，图中麦轮简化模型的斜线代表麦轮直线、曲线或自转等方式运行时与地面接触的小辊子的偏置方向。当电机驱动麦克纳姆轮旋转时，车轮的运动有两种，一是以普通方式向着垂直于驱动轴的方向运动，二是与地面接触的辊子在摩擦力的作用下绕其自身轴线旋转。

根据图 1-4 建立的运动学模型推导出麦克纳姆轮小车的正逆运动学公式（逆运动学即根据底盘速度 V_x 、 V_y 和角速度 ω 推导出四个轮子的角速度 $\omega_i, i = 1, 2, 3, 4$ ，正运动学解算即根据四轮角速度 $\omega_i, i = 1, 2, 3, 4$ 解算出底盘速度 V_x 、 V_y 和角速度 ω ）

1.3 要求

已知 变量	含义
ω	$O - XY$ 坐标系下小车 yaw 轴方向的角速度
v_{oi}	$o_1 - x_1y_1$ 坐标系下轮组 i 的轴心运动速度
v_{ri}	$o_1 - x_1y_1$ 坐标系下轮组 i 接触地面的辊子的运动速度
α	接触地面的辊子的轴线与麦克纳姆轮轴线的夹角
L	小车底盘左右轮轮心距的一半
W	小车底盘前后轮轮心距的一半
R	轮子半径

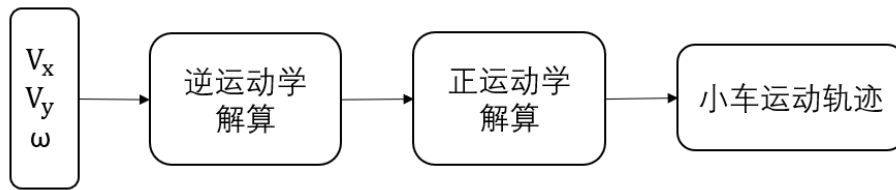
1.3.1 任务一

- (1) 在 $o_1 - x_1y_1$ 坐标系下推导出轮组 1 的速度 v_1 (3 分)
- (2) 在 $O - XY$ 坐标系下推导出轮组 1 的速度 v_1 (3 分)
- (3) 联立 (1) (2) 中两式推导出轮组 1 的角速度 ω_1 与底盘速度 V_x 、 V_y 和角速度 ω 的关系 (2 分)
- (4) 根据前面的步骤推导出完整的麦克纳姆轮正逆运动学公式 (4 分)

1.3.2 任务二

根据任务一得到的运动学公式，在 Simulink 和 MATLAB 中搭建联合仿真，给定小车速度 $V_x = \sin(t)$ ， $V_y = \cos(t)$ ， $\omega = 0$ ，取 $\alpha = 45^\circ$ 、 $L = 0.16\text{m}$ 、 $W = 0.12\text{m}$ 、 $R = 0.06\text{m}$ ，搭建如图 5 所示的仿真，在 MATLAB 中画出小车的运动轨迹。

在报告中展示 Simulink 模型的仿真界面截图及绘制的小车平面运动轨迹图。



1.4 期望目标

推导出麦克纳姆轮小车的正逆运动学公式并在 Simulink 中搭建模型，并用搭建的模型代入题目给定的参数，在 MATLAB 中画出小车的运动轨迹。

2 任务实现

2.1 逆运动学理论推导

2.1.1 整车的运动分解

对于任意的一个轮子，整车的速度 \vec{V} ，角速度 $\vec{\omega}$ 及其对应的旋转半径 \vec{r} 和轮子轴心的运动速度 \vec{v} ，存在关系：

$$\vec{v} = \vec{V} + \vec{v}_r = \vec{V} + \vec{\omega} \times \vec{r}$$

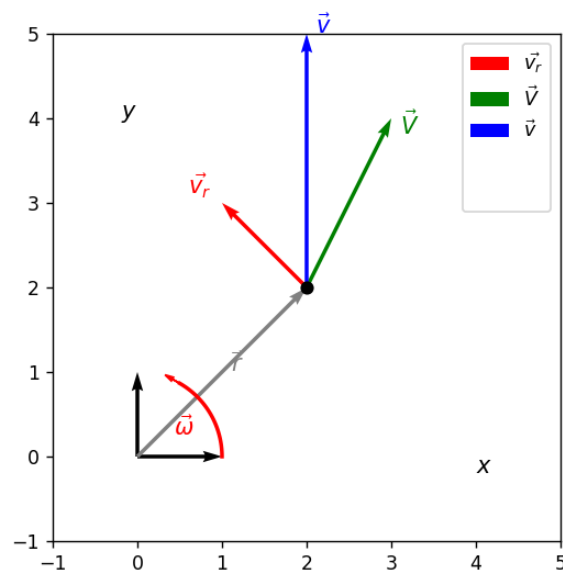


图 2-1

据此分别考虑小车四个轮子的 x 和 y 方向：

$$V_{01x} = V_x - \omega W, \quad V_{01y} = V_y + \omega L \quad (1-1)$$

$$V_{02x} = V_x + \omega W, \quad V_{02y} = V_y - \omega L \quad (1-2)$$

$$V_{03x} = V_x + \omega W, \quad V_{03y} = V_y - \omega L \quad (1-3)$$

$$V_{04x} = V_x - \omega W, \quad V_{04y} = V_y - \omega L \quad (1-4)$$

2.1.2 轮子角速度的表示

$$\omega_1 R = \frac{-V_{01x} \cos(\alpha) + V_{01y} \sin(\alpha)}{\sin(\alpha)} = -V_{01x} \cot(\alpha) + V_{01y} \quad (2-1)$$

$$\omega_2 R = \frac{-V_{02x} \cos(\alpha) + V_{02y} \sin(\alpha)}{\sin(\alpha)} = -V_{02x} \cot(\alpha) + V_{02y} \quad (2-2)$$

$$\omega_3 R = \frac{-V_{03x} \cos(\alpha) + V_{03y} \sin(\alpha)}{\sin(\alpha)} = -V_{03x} \cot(\alpha) + V_{03y} \quad (2-3)$$

$$\omega_4 R = \frac{-V_{04x} \cos(\alpha) + V_{04y} \sin(\alpha)}{\sin(\alpha)} = -V_{04x} \cot(\alpha) + V_{04y} \quad (2-4)$$

2.1.3 逆运动学结果表示

由 2.1.1, 代入, 用 V_x, V_y, L, W, α 表示四个轮子的角速度

$$\omega_1 = \frac{1}{R} [-\cot(\alpha)V_x + V_y + (L + W\cot(\alpha))\omega] \quad (3-1)$$

$$\omega_2 = \frac{1}{R} [\cot(\alpha)V_x + V_y - (L + W\cot(\alpha))\omega] \quad (3-2)$$

$$\omega_3 = \frac{1}{R} [-\cot(\alpha)V_x + V_y - (L + W\cot(\alpha))\omega] \quad (3-3)$$

$$\omega_4 = \frac{1}{R} [\cot(\alpha)V_x + V_y + (L + W\cot(\alpha))\omega] \quad (3-4)$$

也可以写成紧凑的矩阵形式:

$$\begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{pmatrix} = \frac{1}{R} \begin{pmatrix} -\cot(\alpha) & 1 & (L + W\cot(\alpha)) \\ \cot(\alpha) & 1 & -(L + W\cot(\alpha)) \\ -\cot(\alpha) & 1 & -(L + W\cot(\alpha)) \\ \cot(\alpha) & 1 & (L + W\cot(\alpha)) \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ \omega \end{pmatrix} \quad (4)$$

2.2 顺运动学推导

由算式(3-1)到(3-4)，用 $(\omega_1, \omega_2, \omega_3, \omega_4, \alpha, L, W, R)$ 表示 (V_x, V_y, ω)

$$V_x = \frac{1}{4} R \tan(\alpha) (-\omega_1 + \omega_2 - \omega_3 + \omega_4) \quad (5)$$

$$V_y = \frac{1}{4} R (\omega_1 + \omega_2 + \omega_3 + \omega_4) \quad (6)$$

$$\omega = \frac{R}{4(L + W\cot(\alpha))} (\omega_1 - \omega_2 - \omega_3 + \omega_4) \quad (7)$$

2.3 建模与仿真

2.3.1 确定仿真模块的框架

本项目使用 MATLAB2024a 构建，[源码](#)可在 GitHub 中访问（其中‘master’分支为符合课题任务要求的分支）。仿真器开始时间为 0.0，停止时间为 10.0s，使用自动步长，自动求解类型的定步长求解器进行仿真。

逆运动学的输入变量为 $(V_x, V_y, \omega, \alpha, L, W, R)$ ，输出变量为 $(\omega_1, \omega_2, \omega_3, \omega_4)$ 。

顺运动学的输入变量为 $(\omega_1, \omega_2, \omega_3, \omega_4, \alpha, L, W, R)$ ，输出变量为 (V_x, V_y, ω) 。

据此，得到 Simulink 的函数框架。

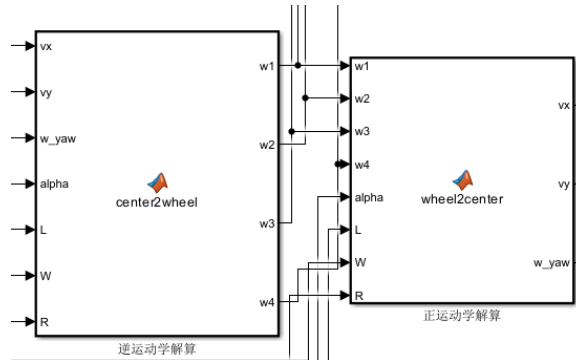


图 2-2

2.3.2 编写模块内部函数

逆运动学模块

由(3-1)到(3-4)，利用中间变量 $offset$ ，编写函数 $center2wheel$ 。

```
project ▸ 逆运动学解算
1  function [w1,w2,w3,w4] = center2wheel(vx,vy,w_yaw,alpha,L,W,R)
2  %args:
3  % vx: 整车x方向速度分量
4  % vy: 整车y方向速度分量
5  % w_yaw: 整车绕yaw轴角速度，逆时针为正
6  % alpha:接触地面的辊子的轴线与麦克纳姆轮轴线的夹角
7  % L: 小车底盘左右轮轮心距的一半
8  % W: 小车底盘前后轮轮心距的一半
9  % R: 轮子半径
10
11  %return:
12  %w1:右上
13  %w2:左上
14  %w3:左下
15  %w4:右下
16  offset = w_yaw * (L + W * cot(alpha));
17
18  w1 = (-vx * cot(alpha) + vy + offset)/R;
19  w2 = (vx * cot(alpha) + vy - offset)/R;
20  w3 = (-vx * cot(alpha) + vy - offset)/R;
21  w4 = (vx * cot(alpha) + vy + offset)/R;
22
23  end
24
```

图 2-3

顺运动学模块

① 由(5)(6)(7)，编写函数 $wheel2center$ 。

```
project ▸ 正运动学解算
1  function [vx,vy,w_yaw] = wheel2center(w1,w2,w3,w4,alpha,L,W,R)
2  arg = 4*cot(alpha);
3
4  %底盘输出速度
5  vx = -(R*(w1 - w2 + w3 - w4))/arg;
6  vy = (R*(w1 + w2 + w3 + w4))/4;
7  w_yaw = (R*(w1 - w2 - w3 + w4))/(4*L + W*arg);
8
9  end
10
```

图 2-4

② 或由(4)，将矩阵算式简写，

$$\omega = \frac{1}{R}AX \quad (8)$$

$$A^T V = A^T A X \quad (9)$$

$$X = (A^T A)^{-1} A^T V \quad (10)$$

因此可以直接利用矩阵进行计算。此时模块内部函数如下：

```

1  function [vx,vy,w_yaw] = wheel2center(w1,w2,w3,w4,alpha,L,W,R)
2  v = [w1,w2,w3,w4] * R;
3  offset = L + W * cot(alpha);
4  arg = cot(alpha);
5  %逆运动解算矩阵center to wheels
6  c2w = [-arg,1,offset;arg,1,-offset;-arg,1,-offset;arg,1,offset];
7
8  %底盘输出速度
9  out = (c2w.'*c2w)\c2w.*v.';
10
11  vx = out(1);
12  vy = out(2);
13  w_yaw = out(3);
14
15  end
16

```

图 2-5

由于矩阵求逆的时间复杂度在 $O(n^2)$ 以上¹，故在实际模型中，我们采用第一种方式求解运动学参数。

2.3.3 仿真模块与轨迹绘制

理论轨迹（ground truth）推导

由于 $V_x = \sin(t)$ ， $V_y = \cos(t)$ ， $\omega = 0$ ， t 从 0 到 10s，可知小车运动轨迹的形状应为圆心位于(1,0)，半径为1，圆心角为10的完整圆弧，且小车出发时车头朝 y 轴正方向，出发后沿顺时针方向运行。

Simulink 仿真

编写轨迹绘制的代码（图 2-6）。

将 $V_x = \sin(t)$ ， $V_y = \cos(t)$ ， $\omega = 0$ ， $\alpha = 45^\circ$ 、 $L = 0.16\text{m}$ 、 $W = 0.12\text{m}$ 、

$R = 0.06\text{m}$ 输入模型参数。

```
myconsole.m  +
1      result = sim("project.slx");
2
3      figure;
4      %绘制小车运行轨迹
5      plot(result.x.Data, result.y.Data);
6      xlabel('Horizontal Displacement');
7      ylabel('Vertical Displacement');
8      title('Trajectory of the Movement');
9      axis equal; % 使x和y轴比例相同，更适合观察轨迹
10     grid on;
11
```

图 2-6

小车的运动轨迹如图 2-7 所示(出发点为(0, 0))。可以发现，小车仿真轨迹与理论轨迹相同。

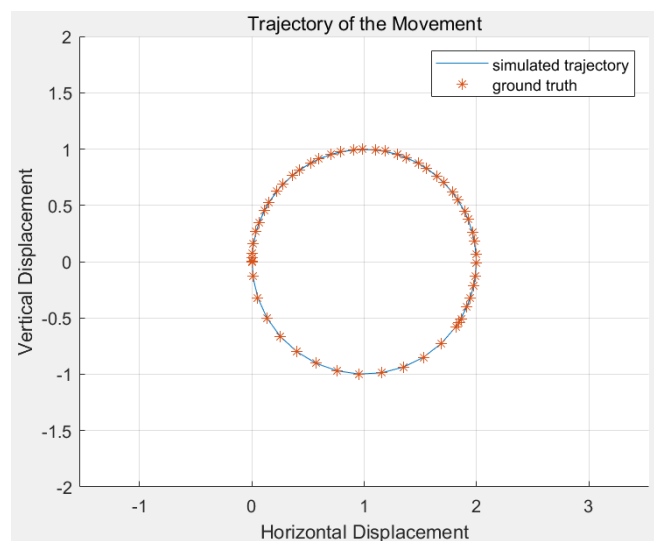


图 2-7

然而，仅通过图 2-7 并不能说明仿真是完全正确的，因为有可能理论轨迹与仿真轨迹的方向（顺时针或逆时针）不同。为此，我们实时记录了小车 y 方向的速度，并把结果保存在时间序列中。y 方向速度的变化情况如图 2-8 所示。

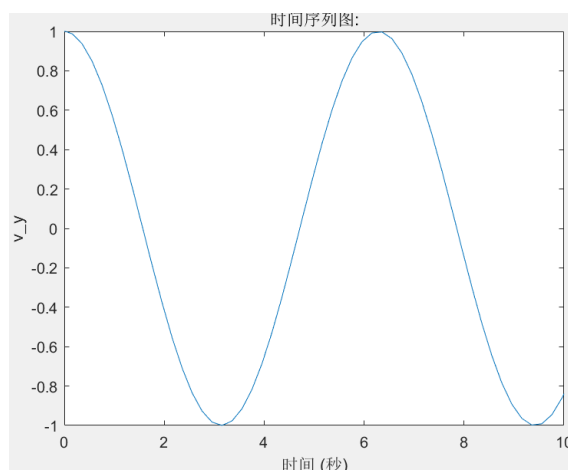


图 2-8

可以看到，出发时小车 y 方向速度为正，故小车运动方向为顺时针方向，与预期方向一致。

另外，仅通过图 2-7 也不能判断小车最终停下的位置是否正确。为此，我们在控制台中监控了小车的终点的 x 和 y 坐标，结果发现，仿真器中小车的停止坐标与理论值相同（输出结果见图 2-9）。

```
>> myconsole
      "simulation end point:"      "1.839"      "-0.544"

      "ground truth: "      "1.839"      "-0.544"

      match!
```

图 2-9

3 分析与总结

3.1 任务一

3.1.1 逆运动学公式

$$\begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{pmatrix} = \frac{1}{R} \begin{pmatrix} -\cot(\alpha) & 1 & (L + W\cot(\alpha)) \\ \cot(\alpha) & 1 & -(L + W\cot(\alpha)) \\ -\cot(\alpha) & 1 & -(L + W\cot(\alpha)) \\ \cot(\alpha) & 1 & (L + W\cot(\alpha)) \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ \omega \end{pmatrix}$$

3.1.2 顺运动学公式

$$V_x = \frac{1}{4} R \tan(\alpha) (-\omega_1 + \omega_2 - \omega_3 + \omega_4)$$

$$V_y = \frac{1}{4} R (\omega_1 + \omega_2 + \omega_3 + \omega_4)$$

$$\omega = \frac{R}{4L + 4W \cot(\alpha)} (\omega_1 - \omega_2 - \omega_3 + \omega_4)$$

3.2 任务二

在给定的参数下，小车的运动轨迹如图。



4 参考文献

[1] Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2001) [1990]. "28.4: Inverting matrices". Introduction to Algorithms (2nd ed.). MIT Press and McGraw-Hill. pp. 755–760. ISBN 0-262-03293-7.

5 附件

5.1 myconsole.m 代码



myconsole.m



project.slx

```
result = sim("project.slx");

figure;
%x 方向速度输入
subplot(2,1,1);
plot(result.vx_input.Data);
xlabel('time/s');
ylabel('input vx');
title('Input Velocities')
%y 方向速度输入
subplot(2,1,2);
plot(result.vy_input.Data);
xlabel('time/s');
ylabel('input vy');

figure;
%绘制小车理论运行轨迹
subplot(2,2,1);
plot(result.x_desired.Data,result.y_desired.Data);
xlabel('x');
ylabel('y');
title('Ground Truth');
axis equal; % 使 x 和 y 轴比例相同，更适合观察轨迹
grid on;
%绘制小车仿真轨迹
subplot(2,2,2);
plot(result.x.Data, result.y.Data);
xlabel('x');
ylabel('y');
title('Generated Trajectory');
axis equal; % 使 x 和 y 轴比例相同，更适合观察轨迹
grid on;
%绘制小车 y 方向速度变化趋势
subplot(2,2,3);
plot(result.vy)
```

```

title('Vertical Velocity');
ylabel('v_y');
xlabel('time/s');
%四个轮子的角速度
subplot(2,2,4);
hold on
plot(result.w1, '*')
plot(result.w2, '*')
plot(result.w3)
plot(result.w4)
title('Angular Speed of Each Wheel');
legend('w1', 'w2', 'w3', 'w4');
xlabel('time/s');
ylabel('w');
hold off

%计算停止位置的坐标
x_end = round(result.x.Data(end),3);
y_end = round(result.y.Data(end),3);
arg_end = round(atan(y_end/(x_end-1)) * 180 / pi,3);
disp(["simulation end point:", x_end, y_end])

arg_truth = round(atan(sin(10)/(-cos(10)))* 180 / pi,3);
x_truth = round(1-cos(10),3);
y_truth = round(sin(10),3);
disp(["ground truth: ", x_truth, y_truth])

if x_end==x_truth && y_end==y_truth
    disp("match!")
else
    disp("unmatch!")
end

```


5.2 Simulink 仿真模块

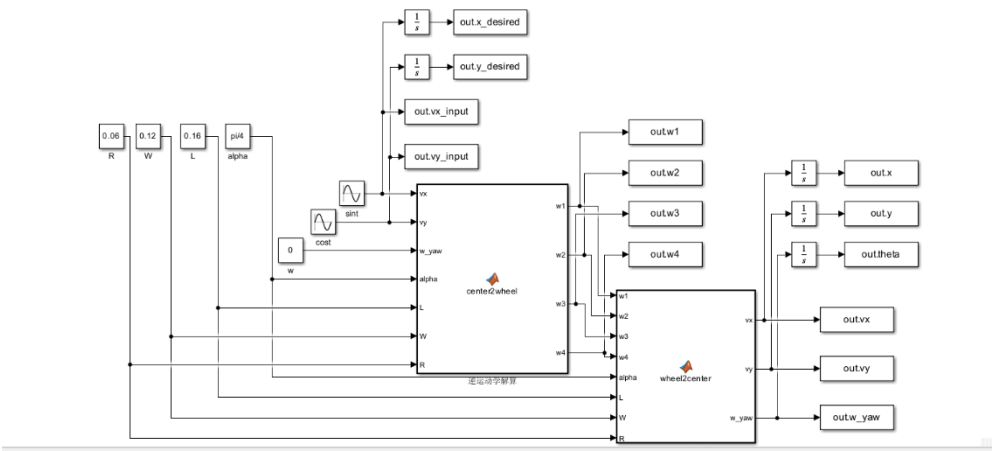


图 5-1

5.3 其余代码及运行结果

图 5-1 为仿真时模型输入的 x 轴方向和 y 轴方向的速度。

图 5-2 分别展示了小车实际运动轨迹，小车理论运动轨迹，小车运动时 y 轴方向的速度以及四个轮子的角速度。

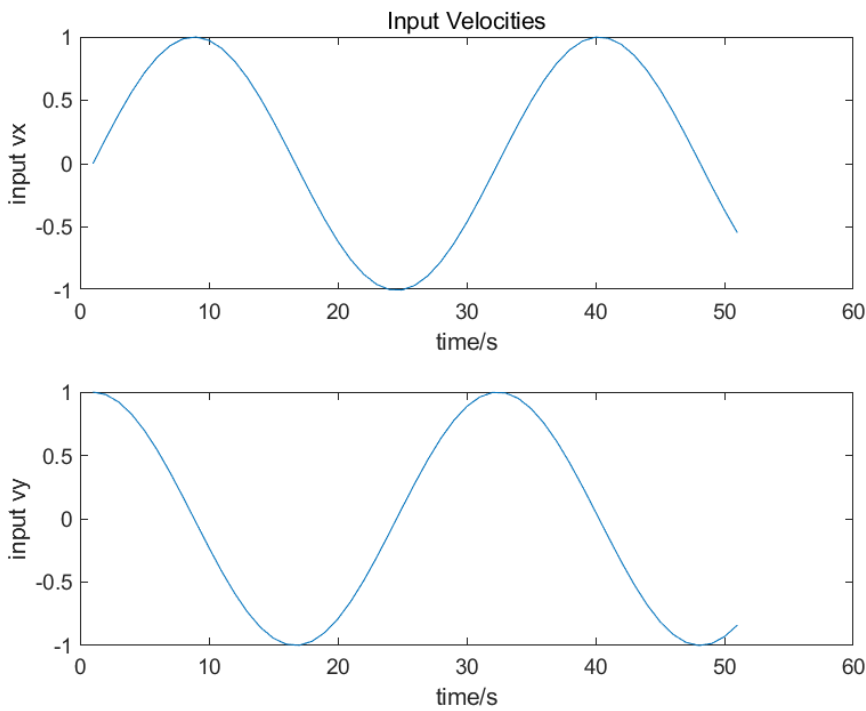


图 5-2

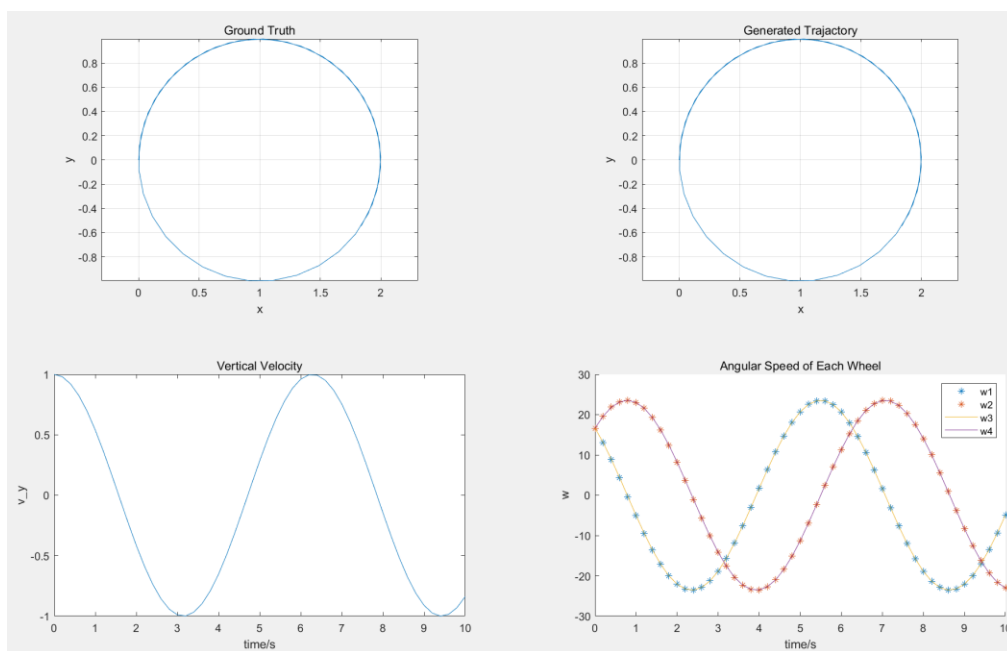


图 5-3