

第4章 跟踪轨迹仿真

4.1 几类跟踪系统中常用曲线的模拟

本节给出几种常用的跟踪目标轨迹数据,包括目标做匀速直线运动、目标做蛇形机动、圆周运动、先做匀速直线运动再作圆周运动,以及在二维平面内做任意方向机动的模拟轨迹数据。本书不仅给出目标做各种运动的测量数据,还给出相应的程序,供读者使用。有的程序调用了函数,有的程序没有,并没有统一的格式,给出这些程序的另一个目的在于让读者熟悉这些 MATLAB 程序,方便读者编制自己的测量数据模拟程序。

a. 匀速直线运动模拟

下面函数 `funtrackingline` 能够模拟目标做匀速直线运动数据,其中输入变量 `a` 表示匀速直线运动的斜率, `t` 是一个向量,该向量通过从运动的开始时间到终止时间以采样周期为间隔对时间序列进行采样得到, `R` 是测量噪声,输出变量包括两个,一个是 `yreal`,它表示目标的真实运动,不含有传感器的测量噪声,另一个是 `ym`,加入了传感器的测量噪声,是实际系统的测量数据。

```
function [yreal,ym]=funtrackingline(a,t,R)
```

```
yreal=a*t;
```

```
ym=yreal+randn(size(yreal))*sqrt(R);
```

例 4.1 调用函数 `funtrackingline`,在程序中先设置输入参数,比如斜率 `a` 为 3, `t` 是从 0 到 10、采样间隔 0.1 的采样时刻向量,测量噪声的方差 `R` 为 30,然后把结果画出来,如图 4.1 所示。图中横坐标是 `t`,纵坐标是测量数据。图中画出两条曲线,一条是目标运动的实际数据,一条是带有传感器测量噪声的测量数据。具体程序如下:

```
clc
```

```
clear
```

```
a=3;t=0:0.1:10;R=30;
```

```
[yreal,ym]=funtrackingline(a,t,R);
```

```
plot(t,yreal,t,ym)
```

```
xlabel('t')
```

```
ylabel('测量数据')
```

```
legend('目标运动的实际数据','传感器测量数据')
```

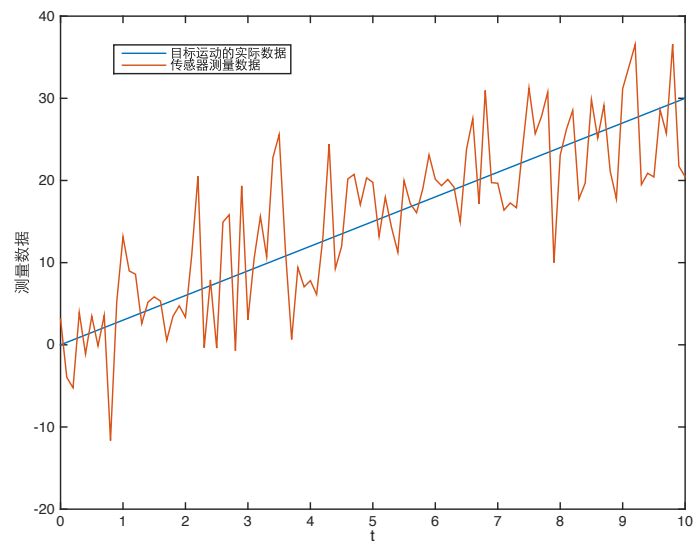


图 4.1 匀速直线运动模拟数据

例4.2 蛇型机动的正弦曲线模拟

给出函数 `funtrackingsnake`，里面包括四个输入变量，其中 a 是正弦曲线的振幅， ω 指正弦曲线的角加速度，和前面一样， t 表示从开始到结束由每个采样点组成的采样时刻向量， R 是测量噪声的方差。同样，输出变量有两个，一个是不带测量噪声的 y_{real} ，一个是带有测量噪声的 y_m 。

```
function [yreal,ym]=funtrackingsnake(a,omig,t,R)
yreal=a*sin(omig*t);
ym=yreal+randn(size(yreal))*sqrt(R);
```

调用函数 `funtrackingsnake`，设置相应的参数再进行适当标注，程序如下：

```
a=2;omig=pi/4;R=3; t=0:0.1:10;
[yreal,ym]=funtrackingsnake(a,omig,t,R);
plot(t,yreal,t,ym)
xlabel('t')
ylabel('测量数据')
legend('目标运动的实际数据','传感器测量数据')
```

程序结果如图 4.2 所示。

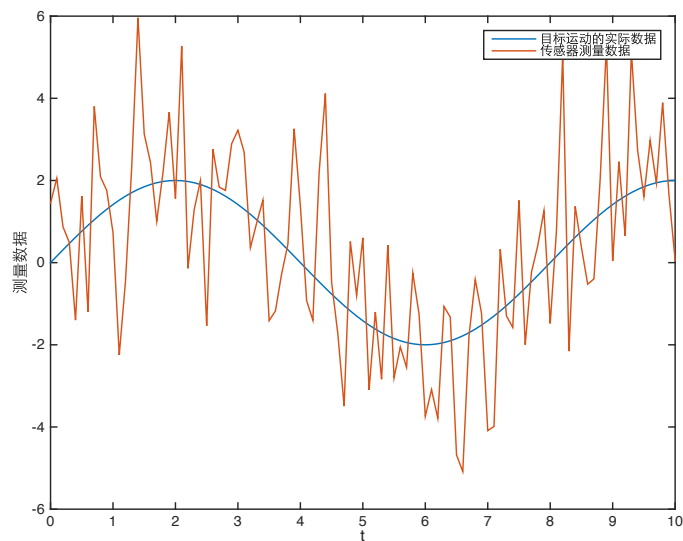


图 4.2 蛇型机动的正弦曲线模拟数据

例 4.3 圆周运动模拟

下面这段程序没有使用函数，直接给出了产生圆周运动的程序，其中各个量的说明已经在程序中%后面做了标注。程序中直接设置了几个参量，包括起点（2000,0），线速度300，半径2000，圆心（0,0），读者可以根据自己的实际需要来重新定义各个变量的具体数值。ww是角度，程序中用正弦、余弦分别计算横纵坐标的数值，利用 `plot(x(1,:), x(2,:), '*')` 画出二维图，第二张图分别给出横纵坐标的量，把它们画在一张图上。

程序具体如下：

```
clc
clear
x0=[2000,0];
v=300;%线速度
T=1;
r=2000;%半径
w=v/r;
op=[0 0];%圆心
x=[];xt=[];www=[];ww=0;
for t=1:T:50
    ww=ww+w;
    x2=[cos(ww)*r+op(1);sin(ww)*r+op(2)];
    x=[x x2];
    xt=[xt t];
    www=[www ww];
end
plot(x(1,:), x(2,:), '*')
xlabel('横轴模拟数据')
ylabel('纵轴模拟数据')
figure
```

```
plot(xt,x,'*')
```

程序结果如图 4.3 所示。

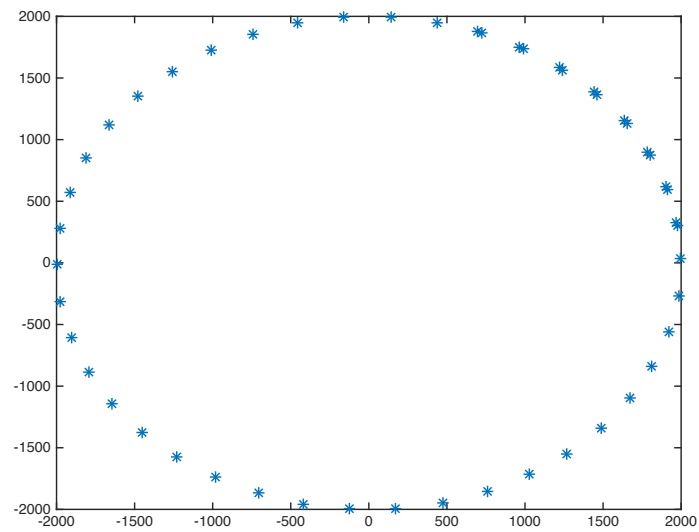


图 4.3 目标在二维平面内圆周运动模拟数据

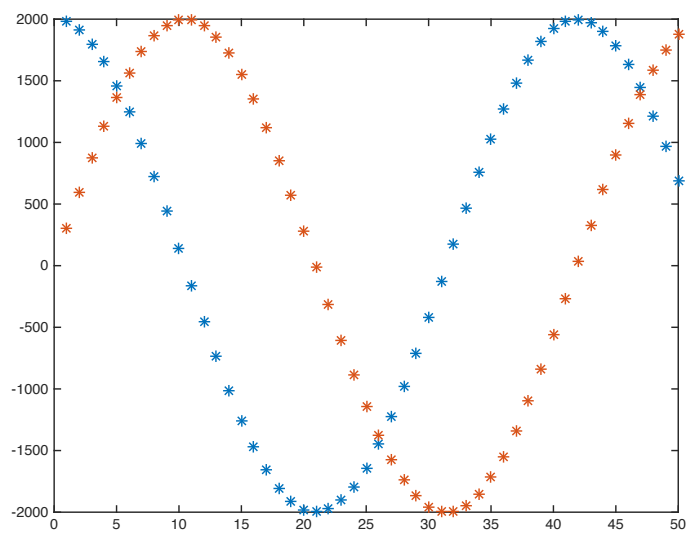


图 4.4 目标做圆周运动时的横、纵轴数据

例 4.4 匀速直线运动+圆周运动目标的真实轨迹模拟

```
function [xt,y]=funtrackinglinecircle(T,Tt,Tz,R)
```

%T:采样周期

%Tt:直线运行时间

%Tz: 圆周运行时间

%R: 测量噪声方差

%本程序模拟匀速直线运动+圆周运动目标的真实轨迹

%功能描述:

```

% 针对某机动目标运动的特点，假设某运动目标经历了两个航行阶段：
% 初始匀速直线运动、匀速圆周转弯机动运动。将目标建立在二维坐标系中，初始位置为
XXX，目标线速度为XXX；
% 以XX角速度向斜上方运动，目标运行Tt秒后，作向心 $\omega$  ms/s2的匀加速圆周运动，速度的
大小v保持不变。
x0=[0 -2000]';%初始位置
v=200;%速度
vj=pi/4;%与水平方向所成角度

x=[];xt=[];
for t=1:T:Tt
    x2=[sin(vj)*v+x1(1);cos(vj)*v+x1(2)];
    x=[x x2];
    xt=[xt t];
    x1=x2;
end

% % % 圆周运动
x0=x2;
v=200;%线速度
r=2000;%半径
w=v/r;
op=[x0(1)-r x0(2)];%圆心
x1=x0;www=[];ww=0;
for tt=t:Tz+t
    ww=ww+w
    x2=[cos(ww)*r+op(1);sin(ww)*r+op(2)];
    x=[x x2];
    xt=[xt tt];
    x1=x2;
    www=[www ww];
end
y=x+randn(size(x))*sqrt(R);

```

例4.4 中funtrackinglinecircle函数的输入量有四个：采样周期T，直线运动时间Tt，圆周运动时间Tz和测量噪声的方差R。从函数中可看出，有些变量是已经设置好的，比如初始位置（0，-2000），还有线速度以及作直线运动时与水平方向产生的角度。程序先进行一段直线运动，运行到最后的点x2。结束for循环之后的下一个起点就是x2，设置了线速度以及运动的半径，给出圆心，又用for循环里的正、余弦进行圆周模拟运动。最后，对给出的目标真实运动轨迹x加上测量噪声，噪声方差是R。

函数有两个输出：一个是xt，一个是y。xt是运动的时间，y是加上测量噪声之后的传感器测量的输出。下面的程序调用上面的函数，程序调用的时候要设置四个输入量，如采样周期为1秒，直线运动时间为20秒，然后做圆周运动50秒，测量噪声方差为1000，最终的结果如图4.5和图4.6所示。图4.5模拟的是目标在平面上运动的情形，可以看出目标从起点开始先做直线运动，然后向左上方转弯，走了一个圈。因为数据带有一定的测量噪声，所

以图形看起来不太规整。图4.6给出的测量数据是目标的真实轨迹和测量噪声的叠加，进行目标跟踪仿真研究的时候使用图4.6的数据。

```
T=1;Tt=20;Tz=50;R=1000;
```

```
[xt,y]=funtrackinglinecircle(T,Tt,Tz,R);
```

```
plot(y(1,:),y(2,:))
```

```
xlabel(' 横轴模拟数据')
```

```
ylabel(' 纵轴模拟数据')
```

```
figure
```

```
subplot(2,1,1),plot(xt,y(1,:))
```

```
xlabel(' t ');ylabel(' 横轴模拟数据')
```

```
subplot(2,1,2),plot(xt,y(2,:))
```

```
xlabel(' t ');ylabel(' 纵轴模拟数据')
```

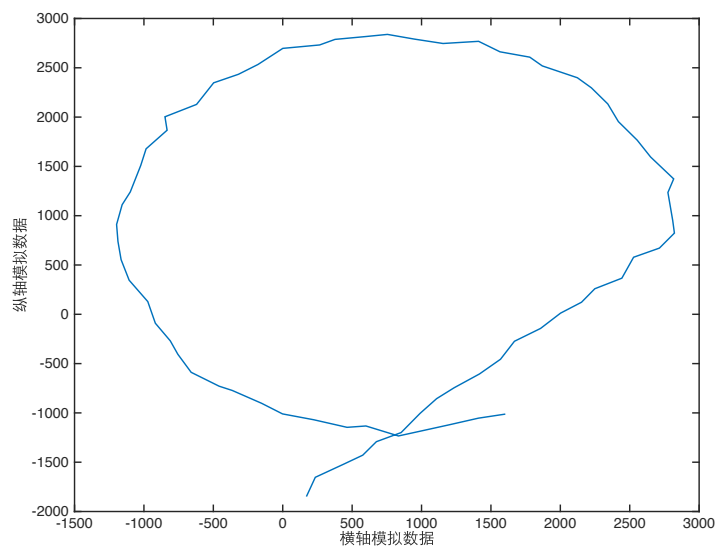


图 4.5 目标先做直线运动、再做圆周运动时的二维平面运动

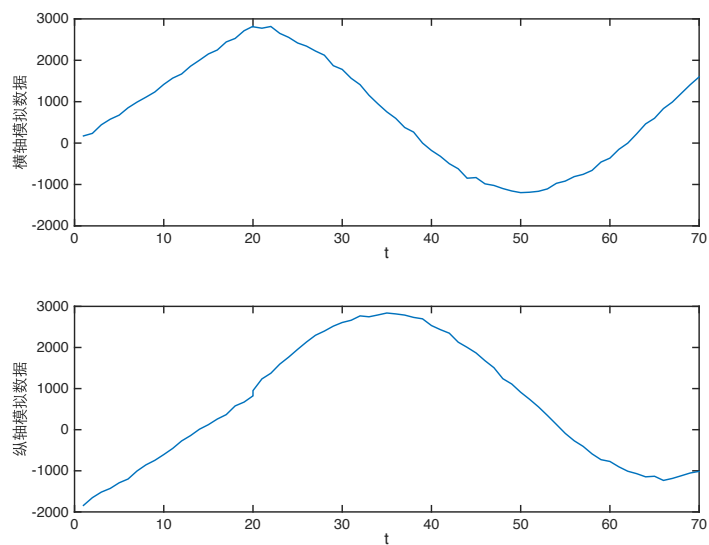


图 4.6 目标先做直线运动、再做圆周运动时的横、纵轴数据

4.2 GPS 跟踪系统的机动目标轨迹模拟

首先介绍一下 GPS 系统的数据采集特征，即传感器测量特征。GPS 系统可以测量二维空间，并且该测量与横、纵坐标不相关。一般情况下，横、纵坐标都含有噪声，方差已知，因此模拟的测量数据需要包含横纵坐标，是 2 维数据。

例4.5 下面给出在2维平面内目标做任意曲线运动的测量数据模拟程序。这段程序使用了函数axis，这个函数可以设置坐标轴，即横、纵坐标的起点和终点，axis([0 100 0 100])表示横、纵坐标的起点和终点都是从0到100，前两个数值表示横坐标的起点和终点，后两个数值表示纵坐标的起点和终点。hold on函数的功能是保持目前图形窗口的状态，即接下来的图会画在目前的图形窗口上。

程序中还使用了两个disp函数，功能是给出说明，disp()里面的任何字符都会显示在窗口上。这里显示的内容是本程序的使用说明：使用鼠标左键点出每一个点，点击右键表示点击该点后结束数据产生步骤。为什么这样提示呢？因为这段程序可以使用坐标在横、纵坐标轴从0~100的区间的任意一点来画点，并记住每一个点的前后顺序，还能把这些点连成一条曲线。那么怎么实现记录每一个点呢？用ginput这个函数，用for循环来记录所有用左键点击的点。ginput函数输入1表示点左键，输出是点击点的横、纵坐标值。每点击一个点，在程序中都会用红色圆圈画出来，不过注意在点击鼠标左键点时不要太快，否则会丢点。接下来把所有点用函数spline连成一条光滑曲线，使用spline函数需要设置插入的点数，程序中在每两个点之间插入10个点，这样能形成一条光滑曲线，来模拟在平面上机动目标走过的曲线。

最后使用subplot，把横、纵坐标放在一张图上，其中实线是机动目标走过的真实的轨迹，不带测量噪声。红色点模拟的是带有测量噪声的传感器测量输出。

%使用说明：使用鼠标左键可以产生轨迹的各个点，鼠标右键为结束点。

```
axis([0 100 0 100])
hold on
% 程序变量初始化
xy = [];
n = 0;
% 使用循环，得到鼠标点击左键是的坐标位置
disp('Left mouse button picks points.')          %%在主窗口上提示操作方法
disp('Right mouse button picks last point.')
but = 1;
while but == 1
    [xi,yi,but] = ginput(1);          %%该函数可以得到鼠标左键的坐标点
    plot(xi,yi,'ro')
    n = n+1;
    xy(:,n) = [xi;yi];
end
% 利用插值函数获得光滑曲线，模拟目标运动曲线和测量数据。
t = 1:n;
ts = 1: 0.1: n;
```

```

xys = spline(t,xy,ts);
plot(xys(1,:),xys(2:),'b-'); %%用蓝色直线画出目标运动曲线
Rx=10;Ry=10;    %%设置横轴、纵轴的测量噪声方差Rx、Ry。
plot(xys(1,:)+randn(size(xys(1,:)))*sqrt(Rx),xys(2,:)+randn(size(xys(1,:)))*sqrt(Ry),'k. ');    %%，在2维坐标下用黑色点画出每一个测量数据
xlabel('x轴');ylabel('y轴')
hold off
%在另一张图上分别画出横、纵轴的目标运动数据和测量数据。
figure
subplot(2,1,1),plot(xys(1,:))
hold on
subplot(2,1,1),plot(xys(1,:)+randn(size(xys(1,:)))*sqrt(Rx),'r.')
ylabel('横轴模拟数据');

subplot(2,1,2),plot(xys(2,:))
hold on
subplot(2,1,2),plot(xys(2,:)+randn(size(xys(1,:)))*sqrt(Rx),'r.')
ylabel('纵轴模拟数据')
hold off
%存储数据
save mytarget1 xys ts

```

程序运行结果如图4.7、4.8所示。

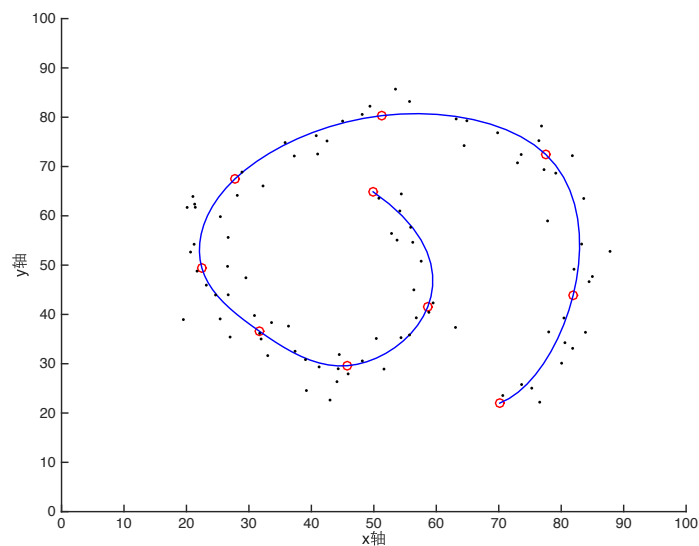


图 4.7 GPS系统的机动目标轨迹模拟

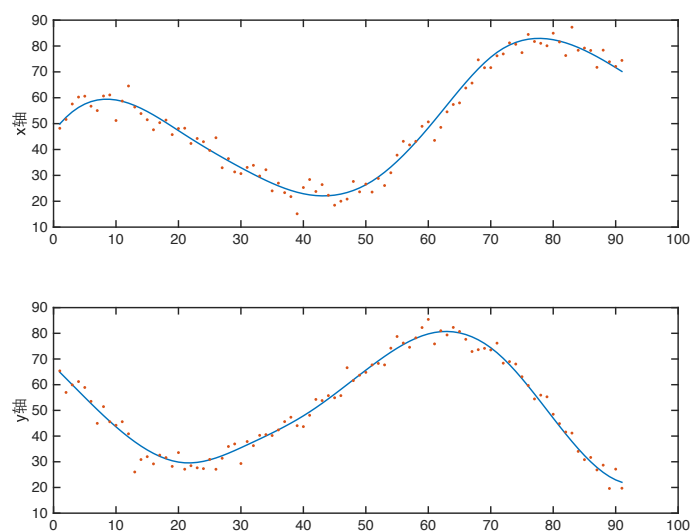


图 4.8 GPS 系统的机动目标轨迹横、纵坐标轴数据模拟

4.3 小结

本章给出了系统仿真需要的一些数据，并给出两种典型系统，GPS 系统的测量模型及数据。在跟踪系统仿真研究中，这些数据是非常有用的。这些数据可以作为系统的实际测量值和目标的真实轨迹参考数据，来定量评估估计方法的性能。除了给出这些数据之外，本章更重要的内容是给出产生这些数据的 MATLAB 程序，熟悉这些程序的读者可以根据自己的需要设计参数、甚至修改程序产生不同的轨迹数据，这些都为目标跟踪仿真研究者提供了大大的便利。