Task1：

```
clc
clear all
close all
%% ************************** Open image ***************************
fprintf('Initializing figures...\n')
h_fig = figure;
h_3d = gca;
axis equal
grid on
view(3);
xlabel('x [m]'); ylabel('y [m]'); zlabel('z [m]')
quadcolors = 1;
set(gcf,'Renderer','OpenGL')
%% ************************** Set initial conditions ***********************
fprintf('Setting initial conditions...\n')
starttime = 0;              % start time         [s]
tstep     = 0.01;           % time step          [s]
cstep     = 1/30;           % Time of image acquisition [s]
totaltime = 5;              % Exercise time    [s]
nstep     = cstep/tstep;
%%%%%%%%%%%%%%%%%% UAV parameters
uav.motor = 100*[0.1,0,-0.1,0,0,0;0,0.1,0,-0.1,0,0;0,0,0,0,0,0.04];    %Aircraft shape
uav.text_dist = 100*0.0333;                                % Mark position
uav.v = 40;                                                %          [m/s]        144 km/h
uav.height = 100;                                          %          [m]
uav.theta = pi/3;                                          %
uav.pos = [0 0 uav.height];                                % Starting position
uav.cameraT = 30;                                          % 30 frames per second
%% **************************Start simulation*****************************
for time = starttime:tstep:totaltime
    %%%%%%%%%%%%%%%%%%Simulation start
    if time == 0                % Initialize UAV
        uav = UavPlot(uav, h_3d);
        h_title = title(sprintf('iteration: %d, time: %4.2f', time/tstep, time));
    else                        % Update drone
        next_pos = uav.pos+[uav.v*tstep 0 0];
        uav = UpdateUavPlot(uav,next_pos);
        set(h_title, 'String', sprintf('iteration: %d, time: %4.2f', time/tstep, time))
    end
    %%%%%%%%%%%%%%%%%% End of simulation
end
%%%%%%%%%%%%%%%% Calculate captured image area
figure
```

```
Score= zeros(120,1);

for h = 1:1:120
    S = (2*h/tan(uav.theta)*sin(pi/4))^2;
    fprintf('The area of the shooting area is% 4.2f m^2\n',S);
    %%%%%%%%%%%%%% Calculate all captured pixels
    PX = 16000000;
    P = PX /S/1000;
    S = S /sqrt(S*S+P*P);
    P = P /sqrt(S*S+P*P);

    w1 = 0.5;
    w2 = 1 - w1;
    ScoreS(h) = w1 * S;
    ScoreP(h) = w2 * P;
    Score(h) = w1 * S + w2 * P;
end

plot(Score,'r','lineWidth',2);
grid on

xL = [0 10    33 80 100 120];
yL = sort([0 Score(10) Score(80)      Score(100) Score(120) 1]);
set(gca,'xtick',xL,'ytick', yL);xlabel('h[m]');ylabel('Score');

title('h - Score')
fprintf('The best theta is% 4.2f ° \n',max(find(Score == max(Score))));
hold on
plot([33 33],[min(Score) max(Score)],'b--')
hold off
%% compute theta
load 3d_compute_theta.mat
h_fig = figure;
h_3d = gca;
axis equal
grid on
view(3);
xlabel('x [m]'); ylabel('y [m]'); zlabel('z [m]')
quadcolors = 1;
set(gcf,'Renderer','OpenGL')
uav = thetaPlot(uav, h_3d);
%% h - S
figure
h = 0:0.001:200;
```

```
S = 2 * h.^2 ./3;

plot(h,S,'r','lineWidth',2);
grid on

xL = [0 30 60 90 100 120 150 180 200];
yL = [0 600 2400 5400 100*100*2/3 120*120*2/3 150*150*2/3 180*180*2/3 200*200*2/3];
set(gca,'xtick',xL,'ytick', yL);xlabel('h[m]');ylabel('S [m^2]');

title('h - S')
```

Task2：

```
%%
clc
clear all
close all

%% ***************************    Open image*****************************
fprintf('Initializing figures...\n')
h_fig = figure;
h_3d = gca;
axis equal
grid on
view(3);
xlabel('x [m]'); ylabel('y [m]'); zlabel('z [m]')
quadcolors = 1;
set(gcf,'Renderer','OpenGL')
%% ***************************** Set initial conditions    ************************
fprintf('Setting initial conditions...\n')
starttime = 0;              % start time         [s]
tstep         = 0.01;         % time step          [s]
cstep         = 1/30;         % Time of image acquisition [s]
totaltime = 5;              % Exercise time    [s]
nstep         = cstep/tstep;

%%%%%%%%%%%%%%%%%% UAV parameters
uav.motor = 100*[0.1,0,-0.1,0,0,0;0,0.1,0,-0.1,0,0;0,0,0,0,0,0.04];    %Aircraft shape
uav.text_dist = 100*0.0333;                                  %Mark position
uav.v = 40;                                                  %                  [m/s]         144 km/h
uav.height = 33;                                             %               [m]
uav.theta = pi/3;                                           %
uav.pos = [0 0 uav.height];                                 % Starting position
uav.cameraT = 30;                                            % 30 frames per second
```

```
%% **************************Start simulation*****************************
for time = starttime:tstep:totaltime
    %%%%%%%%%%%%%%%% start
    if time == 0                    % Initialize UAV
        uav = UavPlot(uav, h_3d);
        h_title = title(sprintf('iteration: %d, time: %4.2f', time/tstep, time));
    else                            % Update uav
        next_pos = uav.pos+[uav.v*tstep 0 0];
        uav = UpdateUavPlot(uav,next_pos);
        set(h_title, 'String', sprintf('iteration: %d, time: %4.2f', time/tstep, time))
    end
    %%%%%%%%%%%%%%%%end
end
%%   %%%%%%%%%%%%%%%%%%   Calculate captured image area
figure
Score= zeros(89,1);
index = 1
for theta = 1:1:89
    dtheta = pi*theta/180;
    S = (2*uav.height / tan(dtheta)*sin(pi/4))^2;
    SSSS(theta) = S;
    fprintf('The area of the shooting area is% 4.2f m^2\n',S);
    %%%%%%%%%%%% Calculate all captured pixels
    PX = 16000000;
    P = PX /S/1000;
    S = S /sqrt(S*S+P*P);
    P = P /sqrt(S*S+P*P);

    w1 = 0.5;
    w2 = 1 - w1;
    ScoreS(theta) = w1 * S;
    ScoreP(theta) = w2 * P;
    Score(theta) = w1 * S + w2 * P;
end
plot(Score,'r','lineWidth',2);
grid on
xL = [0 10 20 30 60 80];
yL = sort([0 Score(10) Score(20)      Score(30) Score(60) Score(80)]);
set(gca,'xtick',xL,'ytick', yL);xlabel('theta');ylabel('Score');
title('theta - Score')
max(find(Score == max(Score)))
hold on
plot([60 60],[min(Score) max(Score)],'b--')
hold off
```

```
%% theta - S
figure
plot(SSSS,'r','lineWidth',2);
grid on
xlabel('theta');ylabel('S[m^2]');
title('theta - S')
%%    for h
sScore = zeros(120,89);
for height = 1:1:120
    for theta = 1:1:89
        dtheta = pi*theta/180;
        S = (2*height / tan(dtheta)*sin(pi/4))^2;
        %%%%%%%%%%%%%    Calculate all captured pixels
        PX = 16000000;
        P = PX /S/1000;
        S = S /sqrt(S*S+P*P);
        P = P /sqrt(S*S+P*P);

        w1 = 0.5;
        w2 = 1 - w1;
        Score = w1 * S + w2 * P;
        sScore(height,theta) = Score;
    end
end

for i = 1:1:120
    index(i) = max(find(sScore(i,:)==max(sScore(i,:))));
end
imagesc(sScore)
xlabel('theta( ° )')
ylabel('h(m)')
colorbar
title('2D-Fractional field')
hold on
for i = 1:1:120
    plot(index(i),i,'+');
end

hold off

figure
mesh(sScore)
view(-43,16);
xlabel('theta( ° )');ylabel('h(m)');zlabel('Score');
```

```
title('3D-Fractional field')
hold on
for i = 1:1:120
    plot3(index(i),i,sScore(i,index(i)),'r.');
end

hold off
```

Task3

a.

```
%%   Discrete velocity

clc
clear all
close all

%% ************************** open figure ***************************
fprintf('Initializing figures...\n')
h_fig = figure;
h_3d = gca;
axis equal
grid on
view(3);
xlabel('x [m]'); ylabel('y [m]'); zlabel('z [m]')
quadcolors = 1;
set(gcf,'Renderer','OpenGL')
%% ************************** Setting initial conditions ***********************
fprintf('Setting initial conditions...\n')
starttime = 0;              %   s
tstep         = 0.1;          %   s
cstep         = 1/30;         %   s
totaltime = 5;              %   s
nstep         = cstep/tstep;

%%%%%%%%%%%%%%%%%%
uav.motor = 100*[0.1,0,-0.1,0,0,0;0,0.1,0,-0.1,0,0;0,0,0,0,0,0.04];
uav.text_dist = 100*0.0333;
uav.v = 40;                                         %       [m/s]       144 km/h
uav.height = 100;                                   %       [m]
uav.theta = pi/3;                                   %
uav.pos = [0 0 uav.height];                         %
uav.cameraT = 30;                                   %

target.v = 40;      %
```

```
target.pos = [0 0 0];
%% **************************start simulation*****************************
index = 1 ;
thetasave = zeros(501,1);
for time = starttime:tstep:totaltime
    %%%%%%%%%%%%%%%%start
    target.time = time;
    if time == 0                    %
        uav = UavPlot(uav, h_3d, target);
        h_title = title(sprintf('iteration: %d, time: %4.2f', time/tstep, time));
    else                                %
        target= getTargetV_1(target);
        target.pos = target.pos + [target.v*tstep 0 0];
        next_pos = uav.pos+[uav.v*tstep 0 0];
        uav = getRtheta(uav,target);
        uav = UpdateUavPlot(uav,next_pos,target);
        set(h_title, 'String', sprintf('iteration: %d, time: %4.2f ,theta: %4.2f', time/tstep, time,uav.theta))
    end
    %%%%%%%%%%%%%%%%end
     pause(0.1);

    thetasave(index) = uav.theta;
    index = index +1;
    %capture
%       if time == 1 || time == 2 || time == 3 || time == 4
%           pause;
%       end
end
%% plot the theta
% xxxxx = find(thetasave> 4*pi/9)
% thetasave(xxxxx) = 2 * 4*pi/9 - thetasave(xxxxx) ;

figure
load theta_1.mat
plot(thetasave,'r','lineWidth',2);
title(t - theta)
xL = [0 100 200 300 400 500];
xlim([0 500])
grid on
set(gca,'xtick',xL);xlabel('t[ms]');ylabel('theta');
hold on
plot([100 100],[0 thetasave(100)],'b--')
plot([200 200],[0 thetasave(200)],'b--')
plot([300 300],[0 thetasave(300)],'b--')
```

```
plot([400 400],[0 thetasave(400)],'b--')
sprintf('The theta between [%4.2f ° ,%4.2f ° ]', min(thetasave)*180/pi,max(thetasave)*180/pi);
%% %%%%%%%%%%%%%%%%%%%%%%%%the change of v
figure
vv = zeros(500,0);
xx = zeros(500,0);
vv(1:100) = 10;
vv(101:200) = 30;
vv(201:300) = 70;
vv(301:400) = 100;
vv(401:500) = 30;
subplot(1,2,1)

plot(vv(1:100),'r','lineWidth',2);
hold on
plot(101:200,vv(101:200),'r','lineWidth',2);
plot(201:300,vv(201:300),'r','lineWidth',2);
plot(301:400,vv(301:400),'r','lineWidth',2);
plot(401:500,vv(401:500),'r','lineWidth',2);

xL = [0 100 200 300 400 500];
yL = sort([0 10 30 70 100 120]);
set(gca,'xtick',xL,'ytick', yL);xlabel('t[ms]');ylabel('v[m/s]');
grid on
title(t - v)
plot([100 100],[10 30],'b--')
plot([200 200],[30 70],'b--')
plot([300 300],[70 100],'b--')
plot([400 400],[100 30],'b--')

subplot(1,2,2)
for i = 1:500
    if i<101
        xx(i) = 10 * i /500;
    elseif i<201
        xx(i) = xx(100) + 30*(i-100)/500;
    elseif i<301
        xx(i) = xx(200) + 70*(i-200)/500;
    elseif i<401
        xx(i) = xx(300) + 100*(i-300)/500;
    else
        xx(i) = xx(400) + 30*(i-400)/500;
    end
end
```

```
plot(xx,'r','lineWidth',2);
title(t - x)
xL = [0 100 200 300 400 500];
grid on
yL = sort([0 1 7.5 22.5 42.5]);
set(gca,'xtick',xL,'ytick', yL);xlabel('t[ms]');ylabel('x[m]');
```

Second part:

```
%%    Continuous speed
clc
clear all
close all

%% ************************** open figure ***************************
fprintf('Initializing figures...\n')
h_fig = figure;
h_3d = gca;
axis equal
grid on
view(3);
xlabel('x [m]'); ylabel('y [m]'); zlabel('z [m]')
quadcolors = 1;
set(gcf,'Renderer','OpenGL')
%% ************************** Setting initial conditions ***********************
fprintf('Setting initial conditions...\n')
starttime = 0;              %    s
tstep         = 0.1;          %    s
cstep         = 1/30;         %    s
totaltime = 5;              %    s
nstep         = cstep/tstep;

%%%%%%%%%%%%%%%%%%
uav.motor = 100*[0.1,0,-0.1,0,0,0;0,0.1,0,-0.1,0,0;0,0,0,0,0,0.04];    %
uav.text_dist = 100*0.0333;                                  %
uav.v = 40;                                                  %              [m/s]         144 km/h
uav.height = 100;                                            %              [m]
uav.theta = pi/3;                                            %
uav.pos = [0 0 uav.height];                                  %
uav.cameraT = 30;                                            %

target.v = 40;       %
target.pos = [0 0 0];
%% **************************start simulation*****************************
index = 1 ;
```

```
thetasave = zeros(501,1);
for time = starttime:tstep:totaltime
    %%%%%%%%%%%%%%%%%start
    target.time = time;
    if time == 0                    %
        uav = UavPlot(uav, h_3d, target);
        h_title = title(sprintf('iteration: %d, time: %4.2f', time/tstep, time));
    else                            %
        target= getTargetV_2(target);
        target.pos = target.pos + [target.v*tstep 0 0];
        next_pos = uav.pos+[uav.v*tstep 0 0];
        uav = getRtheta(uav,target);
        uav = UpdateUavPlot(uav,next_pos,target);
        set(h_title, 'String', sprintf('iteration: %d, time: %4.2f ,theta: %4.2f', time/tstep, time,uav.theta))
    end
    %%%%%%%%%%%%%%%%%end
    pause(0.1);
    thetasave(index) = uav.theta;
    index = index +1;
    %capture
%     if time == 1 || time == 2 || time == 3 || time == 4
%         pause;
%     end
end
%% the theta change figure
% xxxxx = find(thetasave> 4*pi/9)
% thetasave(xxxxx) = 2 * 4*pi/9 - thetasave(xxxxx) ;

figure
load theta_2.mat
plot(thetasave,'r','lineWidth',2);
title(t - theta)
xL = [0 100 200 300 400 500];
xlim([0 500])
grid on
set(gca,'xtick',xL);xlabel('t[ms]');ylabel('theta');
hold on
plot([100 100],[0 thetasave(100)],'b--')
plot([200 200],[0 thetasave(200)],'b--')
plot([300 300],[0 thetasave(300)],'b--')
plot([400 400],[0 thetasave(400)],'b--')
sprintf('The theta between [%4.2f°  ,%4.2f°  ]', min(thetasave)*180/pi,max(thetasave)*180/pi)
```

```
%% the change of v

figure
vv = zeros(500,0);
xx = zeros(500,0);

index = 1;
for i = 0 :0.01: 5
    if i<1
        vv(index) = i * 10;
        index = index + 1;
    elseif i<2
        vv(index) = 10 + (i-1) * 20;
        index = index + 1;
    elseif i<3
        vv(index) = 30 + (i-2) * 40;
        index = index + 1;
    elseif i<4
        vv(index) = 70 + (i-3) * 30;
        index = index + 1;
    else
        vv(index)= 100 - (i-4) * 70;
        index = index + 1;
    end
end
subplot(1,2,1)


plot(vv,'r','lineWidth',2);
hold on
xL = [0 100 200 300 400 500];
yL = sort([0 10 30 70 100 120]);
set(gca,'xtick',xL,'ytick', yL);xlabel('t[ms]');ylabel('v[m/s]');
grid on
title(t - v)
plot([100 100],[0 10],'b--')
plot([200 200],[0 30],'b--')
plot([300 300],[0 70],'b--')
plot([400 400],[0 100],'b--')


subplot(1,2,2)
for i = 1:500
    if i<101
```

```
        xx(i) = 5 * i * i /10000;
    elseif i<201
        xx(i) = xx(100)+    10*(i-100)/100 +10*(i-100)*(i-100)/10000;
    elseif i<301
        xx(i) = xx(200)+ 30*(i-200)/100 + 20*(i-200)*(i-200)/10000;
    elseif i<401
        xx(i) = xx(300)+ 70*(i-300)/100 + 15*(i-300)*(i-300)/10000;
    else
        xx(i) = xx(400)+ 100*(i-400)/100 - 35*(i-400)*(i-400)/10000;
    end
end
plot(xx,'r','lineWidth',2);
title(t - x)
xL = [0 100 200 300 400 500];
grid on
yL = sort([0 5 25 75 160 160+65]);
set(gca,'xtick',xL,'ytick', yL);xlabel('t[ms]');ylabel('x[m]');
hold on
plot([100 100],[0 5],'g--')
plot([200 200],[0 25],'g--')
plot([300 300],[0 75],'g--')
plot([400 400],[0 160],'g--')
plot([500 500],[0 160+65],'g--')
```

Task4：

```
%%
clc
clear all
close all

%% ***************************** open figure *****************************
fprintf('Initializing figures...\n')
h_fig = figure;
h_3d = gca;
axis equal
grid on
view(3);
xlabel('x [m]'); ylabel('y [m]'); zlabel('z [m]')
quadcolors = 1;
set(gcf,'Renderer','OpenGL')
%% ***************************** Setting initial conditions. ************************
fprintf('Setting initial conditions...\n')
starttime = 0;
tstep         = 0.1;
```

```
cstep       = 1/30;
totaltime = 5;
nstep       = cstep/tstep;

%%%%%%%%%%%%%%%%
uav.motor = 100*[0.1,0,-0.1,0,0,0;0,0.1,0,-0.1,0,0;0,0,0,0,0,0.04];
uav.text_dist = 100*0.0333;
uav.v = 40;
uav.height = 100;
uav.theta = pi/3;
uav.pos = [0 0 uav.height];
uav.cameraT = 30;

%% **************************start simulation*****************************

for time = starttime:tstep:totaltime
    %%%%%%%%%%%%%%%%start
    uav.time = time;
    if time == 0
        uav = UavPlot(uav, h_3d);
        h_title = title(sprintf('iteration: %d, time: %4.2f, height: %4.2f', time/tstep, time, uav.height));
    else
        next_pos = uav.pos+[uav.v*tstep 0 0];

        uav.height = getH(uav.time);
        next_pos(3) = uav.height;
        uav = UpdateUavPlot(uav,next_pos);
        set(h_title, 'String', sprintf('iteration: %d, time: %4.2f, height: %4.2f', time/tstep, time,uav.height))
    end
    %%%%%%%%%%%%%%%%end
    pause(0.1)

%       if time==1 || time==2 || time==3 ||time==4
%           pause
%       end
end

%%
load ans_1.mat
resxx = resx;
resyy = resy;
smoothxx = smoothx;
smoothyy = smoothy;
```

```
load ans_2.mat
resxx = [ resx resxx];
resyy = [ resy resyy];
smoothxx =[smoothx smoothxx ];
smoothyy =[smoothy smoothyy ];

load ans_3.mat
resxx = [resx resxx];
resyy = [resy resyy ];
smoothxx =[smoothx smoothxx ];
smoothyy =[smoothy smoothyy ];

figure
rectangle('Position',[20,0,10,72],'FaceColor','c')    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
rectangle('Position',[34,0,9,52],'FaceColor','c')    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
rectangle('Position',[55,0,1,5],'FaceColor','g')
rectangle('Position',[65,0,15,100],'FaceColor','g')
rectangle('Position',[100,0,1,5],'FaceColor','g')
rectangle('Position',[108,0,1,5],'FaceColor','g')
rectangle('Position',[125,0,10,90],'FaceColor','b')
rectangle('Position',[145,0,10,90],'FaceColor','b')
rectangle('Position',[165,0,10,90],'FaceColor','b')
hold on
plot(resxx,resyy,'r','lineWidth',2);
grid on
title('Optimal path of UAV')

figure
rectangle('Position',[20,0,10,72],'FaceColor','c')    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
rectangle('Position',[34,0,9,52],'FaceColor','c')    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
rectangle('Position',[55,0,1,5],'FaceColor','g')
rectangle('Position',[65,0,15,100],'FaceColor','g')
rectangle('Position',[100,0,1,5],'FaceColor','g')
rectangle('Position',[108,0,1,5],'FaceColor','g')
rectangle('Position',[125,0,10,90],'FaceColor','b')
rectangle('Position',[145,0,10,90],'FaceColor','b')
rectangle('Position',[165,0,10,90],'FaceColor','b')
hold on
plot(smoothxx,smoothyy,'r','lineWidth',2);
grid on
title('Optimal smoothing path of UAV')
```