```
%                          问题三：掉头空间                              %
%   (运行前请clear工作区,避免出现错误!)

% 参数设置
num_benches = 30;  % 只看30个板凳
head_length = 3.41;
body_length = 2.20;
benches_width = 0.3;
hole_to_head_distance = 0.275;
T=20;
v_head = 1.0;
dt = 0.01;
% 设置板凳孔的初始位置和角度设置
cnt=1; %计数

p=0.55;
dp=-0.0001;            %螺距每次减少0.01
r0 = 4.5;           % 初始时半径 4.5

stop_signal1 = false ;
stop_signal = false ;

for k =p:dp:0.3
    if stop_signal
        break;  % 退出主循环
    end
    stop_signal1 = false;

    cnt=cnt+1;
    theta0 = 9*pi/k;
    theta_start=theta0+pi;  %初始的角度增加180度
    r_start=theta_start*k/(2*pi);  %初始时刻龙头位置
    % 初始化龙头的位置
    positions(1,: , cnt) = [r_start * cos(theta_start), ...
        r_start * sin(theta_start)];
    L = [head_length - 2 * hole_to_head_distance;...
        repmat(body_length - 2 * hole_to_head_distance, num_benches-1, 1)];

    % 初始化龙身和龙尾位置
    initial_theta = theta_start;
    initial_r = r_start;
    for i = 2:num_benches
        delta_theta(i) = L(i-1) / initial_r;
        initial_theta = initial_theta + delta_theta(i);
        initial_r = k / (2 * pi) * initial_theta;
        positions(i, 1, cnt+1) = initial_r * cos(initial_theta);
        positions(i, 2, cnt+1) = initial_r * sin(initial_theta);
    end
        current_theta=theta_start;
        current_r = r_start;
    for j = dt:dt:T
        %出现终止信号退出循环
        if stop_signal1
            break;
        end
        t = round(j / dt);
        % 计算龙头位置
        theta_head = current_theta - v_head * dt / current_r;
```

```
r_head = k / (2 * pi) * theta_head;
positions(1, :, cnt+1) = [r_head * cos(theta_head), r_head * sin(theta_head)];

% 更新龙头前把手极坐标的角度和半径
current_theta = theta_head;
current_r = r_head;
% 更新第一节龙身前把手的角度和半径
initial_theta = current_theta;
initial_r = current_r;

% 计算龙身和龙尾位置
for i = 2:num_benches
    delta_theta(i) = L(i-1) / initial_r;
    initial_theta = initial_theta + delta_theta(i);
    initial_r = k / (2 * pi) * initial_theta;
    positions(i, 1, cnt+1) = initial_r * cos(initial_theta);
    positions(i, 2, cnt+1) = initial_r * sin(initial_theta);
end

 for i = 1:(num_benches-1)
    % 计算方向向量
    dx = positions(i+1, 1, cnt+1) - positions(i, 1, cnt+1);
    dy = positions(i+1, 2, cnt+1) - positions(i, 2, cnt+1);
    length = sqrt(dx^2 + dy^2);
    ux = -dy / length;  % 垂直方向的x分量
    uy = dx / length;   % 垂直方向的y分量

    % 计算每段线段两侧的四个顶点
    x_left1 = positions(i, 1, cnt+1) + ux * benches_width / 2;
    y_left1 = positions(i, 2, cnt+1) + uy * benches_width / 2;
    x_right1 = positions(i, 1, cnt+1) - ux * benches_width / 2;
    y_right1 = positions(i, 2, cnt+1) - uy * benches_width / 2;

    x_left2 = positions(i+1, 1, cnt+1) + ux * benches_width / 2;
    y_left2 = positions(i+1, 2, cnt+1) + uy * benches_width / 2;
    x_right2 = positions(i+1, 1, cnt+1) - ux * benches_width / 2;
    y_right2 = positions(i+1, 2, cnt+1) - uy * benches_width / 2;

    % 计算龙头方向向量
    dx = positions(2, 1, cnt+1) - positions(1, 1, cnt+1);
    dy = positions(2, 2, cnt+1) - positions(1, 2, cnt+1);
    length = sqrt(dx^2 + dy^2);
    ux = -dy / length;  % 垂直方向的x分量
    uy = dx / length;   % 垂直方向的y分量
    % 定义延长的倍数
    scale_factor = (3.41/2.86-1)/2;  % 延长1.5倍长度
    % 计算延长后的新的两个端点
    new_x1 = positions(1, 1, cnt+1) - scale_factor * dx;  % 起点向外延长
    new_y1 = positions(1, 2, cnt+1) - scale_factor * dy;
    new_x2 = positions(2, 1, cnt+1) + scale_factor * dx;  % 终点向外延长
    new_y2 = positions(2, 2, cnt+1) + scale_factor * dy;
    % 计算延长后的四个顶点
    new_x_left1 = new_x1 + ux * benches_width / 2;
    new_y_left1 = new_y1 + uy * benches_width / 2;
    new_x_right1 = new_x1 - ux * benches_width / 2;
    new_y_right1 = new_y1 - uy * benches_width / 2;

    new_x_left2 = new_x2 + ux * benches_width / 2;
    new_y_left2 = new_y2 + uy * benches_width / 2;
    new_x_right2 = new_x2 - ux * benches_width / 2;
    new_y_right2 = new_y2 - uy * benches_width / 2;


    % %图像显示占用电脑资源影响运行速度，可以选择注释提高运行效率
    % % 绘制延长后的宽线段
    % patch([x_left1, x_left2, x_right2, x_right1], ...
    %     [y_left1, y_left2, y_right2, y_right1], 'b');
    % patch([new_x_left1, new_x_left2, new_x_right2,...
    %     new_x_right1], [new_y_left1, new_y_left2, ...
    %     new_y_right2, new_y_right1], 'r');


    %龙头的线段
    cnt1_head=[new_x_left1,new_y_left1];  %龙头的起点
    cnt2_head=[new_x_right1,new_y_right1]; %龙头的终点

    %龙身的矩形
    Q1_body=[x_left1,y_left1];
    Q2_body=[x_right1,y_right1];
    Q3_body=[x_left2,y_left2];
    Q4_body=[x_right2,y_right2];

    if check_intersection(cnt1_head, cnt2_head, Q1_body, Q2_body) || ...
       check_intersection(cnt1_head, cnt2_head, Q2_body, Q4_body) || ...
       check_intersection(cnt1_head, cnt2_head, Q4_body, Q3_body) || ...
       check_intersection(cnt1_head, cnt2_head, Q3_body, Q1_body)
        stop_signal1 = true;  % 设置标志
    end
 end
 if sqrt(positions(1, 1, cnt+1).^2+positions(1, 2, cnt+1).^2)<4.5
     stop_signal = false;
 else
     stop_signal = true;
 end

disp(['p = ' num2str(k) 'm)']);
disp(['t = ' num2str(j) 's)']);


% %图像显示占用电脑资源影响运行速度，可以选择注释提高运行效率
% pause(0.01);
% clf;
% hold on;
% axis equal;
% xlabel('X (米)');
% ylabel('Y (米)');
% xlim([-12, 12]);
% ylim([-12, 12]);
% grid on;
% title(['板凳把手位置示意图 (p = ', num2str(k), 'm)(t = ',...
%         num2str(j), 's)']);
% plot(positions(1:end, 1, cnt+1), positions(1:end, 2, cnt+1), ...
%     'go-', 'MarkerSize', 3, 'LineWidth', 1,'MarkerFaceColor', 'r');
% hold off;
```

```
    end

end
    disp(['龙头线段与龙身相交，停止模拟（p = ' num2str(k-dp) 'm）']);
```