**2024美赛B题参考完整解题过程**

**题目**

Maritime Cruises Mini-Submarines (MCMS) 是一家总部位于希腊的公司，生产能够将人类运送到海洋最深处的潜水器。潜水器被移至该位置并脱离主船进行部署。 MCMS 现在希望利用他们的潜水器带领游客冒险探索爱奥尼亚海海底沉船残骸。

然而，在此之前，他们需要通过制定安全程序来赢得监管机构的批准，以**应对与主船失去通信以及可能出现机械缺陷（包括潜水器失去推进力）的情况。**特别是，他们希望您开发一个模型来预测潜水器随时间变化的位置。与典型的陆地或海面搜救不同，有缺陷的潜水器可能会发现自己位于海底或水下某个中性浮力点。它的位置可能进一步受到洋流、海洋不同密度和/或海底地理的影响。你的任务是：

**• 定位-** 开发一个模型来预测潜水器随时间变化的位置。

o 这些预测有哪些不确定性？ o 潜水器可以定期向主船发送哪些信息，以在事故发生前减少这些不确定性？潜水器需要什么样的设备才能做到这一点？

**准备 -** 如果有的话，您建议公司在主船上配备哪些额外的搜索设备，以便在必要时部署？您可以考虑不同类型的设备，但还必须考虑与该设备的可用性、维护、准备情况和使用相关的成本。如有必要，救援船可能需要携带哪些额外设备来提供协助？

**搜索 -** 开发一个模型，该模型将使用来自您的位置模型的信息来推荐设备的初始部署点和搜索模式，以便最大限度地缩短定位丢失潜水器的时间。确定找到潜水器的概率作为时间和累积搜索结果的函数。

**推断——**如何扩展您的模型以考虑加勒比海等其他旅游目的地？您的模型将如何改变以考虑在同一区域内移动的多个潜水器？

**思路完整解题过程**

**一、潜水器定位问题**

**有缺陷的潜水器位于海底或水下某个中性浮力点的意义：**

这句话意味着，当潜水器出现机械故障时，它可能无法上浮到水面或下沉到海底，而是漂浮在水中的某个深度，这个深度处潜水器不会因为浮力差异继续上浮或下沉，达到了一种中性浮力状态。在这种状态下，潜水器既不受重力拉扯下沉，也不会因为浮力过大而上浮，这使得搜救行动更加复杂，因为潜水器可能位于任何深度。

**不确定性：**

环境因素，洋流影响，数据错误（概率低），失去部分数据，仪器测量误差（可以忽略不计）

**考虑三维空间中点的运动状态描述：（可以定期发送的信息）**

1. 位置：用一个三维坐标系中的点来表示，比如使用笛卡尔坐标系的 (*x*,*y*,*z*)。
2. 速度：速度是位置随时间的变化率，也是一个向量，表示方向和大小。在三维空间中，速度可以表示为vx，vy，vz。
3. 加速度：加速度是速度随时间的变化率，同样是一个向量

对于旋转运动，可以添加：

1. 角速度：角速度是物体旋转速率的量度，表示为向量，方向遵循右手规则。
2. 角加速度：角加速度是角速度随时间的变化率，表示为向量。

**初步模型：最基础的三维空间获取潜水器位置**

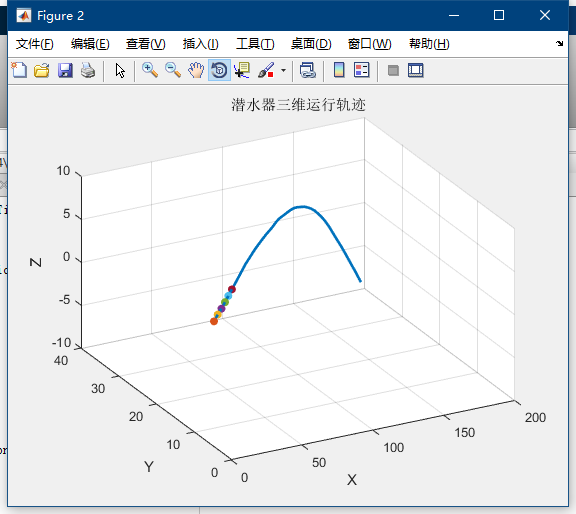
|  |
| --- |
| function newPosition = predictPosition(currentPosition, velocity, acceleration, angularVelocity, angularAcceleration, deltaTime)  % 初始化位置和速度  x = currentPosition(1);  y = currentPosition(2);  z = currentPosition(3);  vx = velocity(1);  vy = velocity(2);  vz = velocity(3);  ax = acceleration(1);  ay = acceleration(2);  az = acceleration(3);  % 更新速度和位置  vx\_new = vx + ax \* deltaTime;  vy\_new = vy + ay \* deltaTime;  vz\_new = vz + az \* deltaTime;  x\_new = x + vx \* deltaTime + 0.5 \* ax \* deltaTime^2;  y\_new = y + vy \* deltaTime + 0.5 \* ay \* deltaTime^2;  z\_new = z + vz \* deltaTime + 0.5 \* az \* deltaTime^2;  newPosition = [x\_new, y\_new, z\_new];  end |

题目无数据，对于数据干扰和扰动需要自行定义。

这里采用随机漫步模型构造潜水器的运行状态：

|  |
| --- |
| % 初始化随机数生成器，确保每次运行结果都不同  rng('shuffle');  % 模拟参数 - 引入随机性  initialPosition = rand(1, 3) \* 10; % 初始位置，随机值范围[0, 10]  velocity = rand(1, 3) \* 5; % 初始速度，随机值范围[0, 5]  acceleration = rand(1, 3) \* 0.2 - 0.1; % 加速度，随机值范围[-0.1, 0.1]  angularVelocity = rand \* 2 \* pi; % 角速度，随机值范围[0, 2π]  angularAcceleration = rand \* 0.2 - 0.1; % 角加速度，随机值范围[-0.1, 0.1]  deltaTime = 1; % 时间步长，每秒更新一次  simulationDuration = 40; % 模拟总时长（秒），保持不变  % 初始化位置存储  positions = zeros(simulationDuration + 1, 3);  positions(1, :) = initialPosition;  % 模拟运动  for t = 1:simulationDuration  % 更新位置  currentPosition = positions(t, :);    % 引入随机漫步模型来模拟海流的随机影响  % 随机分量，模拟海流对潜水器运动的影响  randomComponent = rand(1, 3) \* 0.2 - 0.1; % 随机值范围[-0.1, 0.1]    % 更新速度，加入随机分量模拟海流影响  velocity = velocity + acceleration \* deltaTime + randomComponent;    % 根据更新后的速度计算新位置  newPosition = predictPosition(currentPosition, velocity, acceleration, angularVelocity, angularAcceleration, deltaTime);    % 存储新位置  positions(t + 1, :) = newPosition;  end  % 绘制三维轨迹  figure;  plot3(positions(:, 1), positions(:, 2), positions(:, 3), 'LineWidth', 2);  grid on;  xlabel('X');  ylabel('Y');  zlabel('Z');  title('潜水器三维运行轨迹');  % 逐秒展示位置  hold on;  for t = 1:size(positions, 1)  scatter3(positions(t, 1), positions(t, 2), positions(t, 3), 'filled');  pause(1); % 暂停一秒，以模拟实时更新  end |

输出结果：



到这里，我们构造出了一个真实的运动运行轨迹。接下来基于贝叶斯推理，继续构造我们的预测模型。

2、概率模型确定：贝叶斯推理

贝叶斯推理用于根据实时数据（如深度、声纳数据）更新潜水艇位置的概率分布。根据每秒接受到的数据，考虑扰动后，估计潜水器的位置概率。

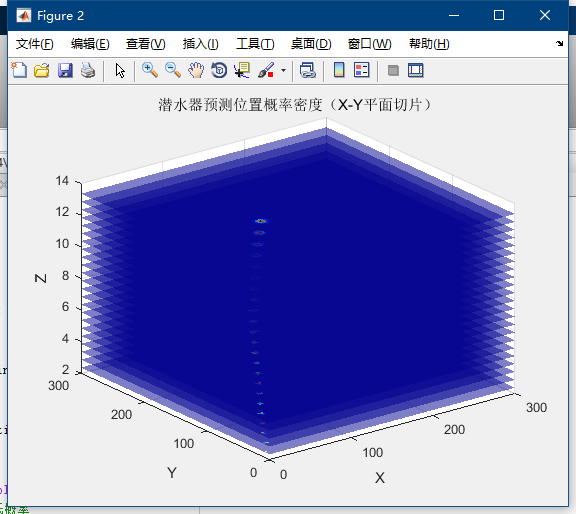
P(A∣B) 是在已知 B 发生的条件下，A 发生的条件概率（后验概率）。

P(B∣A) 是在已知 A 发生的条件下，B 发生的条件概率。

P(A) 是 A 发生的先验概率，即在未获得 B 的信息前，A 发生的概率。

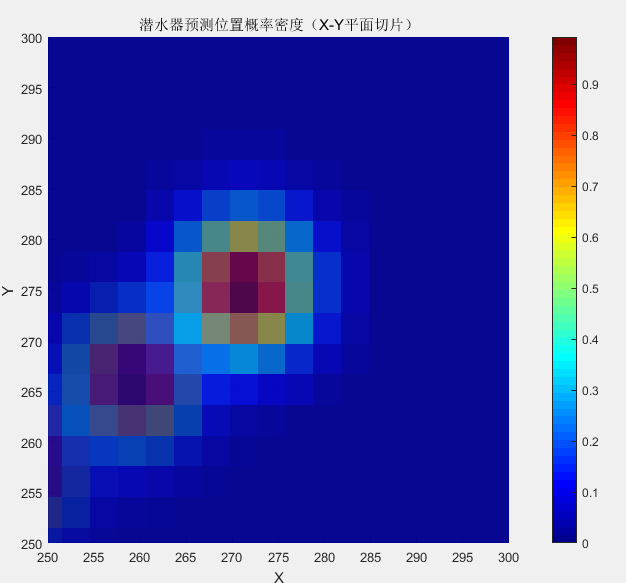
P(B) 是 B 发生的总概率，可以视为标准化常量，确保概率的总和为1。

|  |
| --- |
| % 假设基于上述潜水器运动模拟的代码已经执行，得到了positions数组  % 初始化随机数生成器  rng('shuffle');  % 计算XYZ轴的显示范围  maxDisplacement = max(abs(velocity) + abs(acceleration) \* simulationDuration) \* simulationDuration;  xRange = [min(positions(:, 1)) - maxDisplacement, max(positions(:, 1)) + maxDisplacement];  yRange = [min(positions(:, 2)) - maxDisplacement, max(positions(:, 2)) + maxDisplacement];  zRange = [min(positions(:, 3)) - maxDisplacement, max(positions(:, 3)) + maxDisplacement];  % 模拟参数  simulationDuration = 40; % 模拟总时长（秒）  % 绘制三维轨迹和概率密度  figure;  hold on;  grid on;  xlabel('X');  ylabel('Y');  zlabel('Z');  title('潜水器预测位置概率密度（X-Y平面切片）');  view(3); % 设置三维视图  % 选择Z轴（深度）的切片高度为初始Z位置  zSlice = initialPosition(3);  % 遍历每个时间点，绘制潜水器位置的概率密度  for t = 1:size(positions, 1)  std = initialStd + (t-1) \* stdIncreaseRate; % 更新标准差    % 在X-Y平面生成网格  [X, Y] = meshgrid(linspace(0, 300, 100), linspace(0, 300, 100));    % 计算当前时间点潜水器位置的概率密度分布  P = exp(-((X-positions(t, 1)).^2 + (Y-positions(t, 2)).^2) / (2\*std^2));    % 绘制概率密度  surf(X, Y, zSlice\*ones(size(X)), P, 'EdgeColor', 'none');  colormap('jet'); % 绿色表示高概率，红色表示低概率  alpha(0.5); % 设置透明度以便于观察轨迹    % 更新Z轴切片高度，使其随时间轻微上升，以分离不同时间点的概率分布  zSlice = zSlice + 0.5;    pause(0.1); % 暂停一秒，模拟实时更新  end  colorbar; % 显示颜色条 |



直接运行代码结果如图所示。

这样可以动态构造出来位置的大致估计范围。通过调整x和y轴获得更清晰的预测。



**相关设备和对应指标：**

具体的海域和环境条件有很大的变化。

|  |  |  |  |
| --- | --- | --- | --- |
| **设备** | **指标** | **单位** | **期待正常值范围** |
| **声学多普勒流速剖面仪（ADCP）** | 水流速度 | 米/秒（m/s） | 0 - 2 m/s[1](https://chat.openai.com/c/887f46ca-6a67-41b0-b938-e392ef13298e" \l "user-content-fn-1%5E" \t "_new) |
| **电导率、温度、深度传感器（CTD）** | 水温 | 摄氏度（°C） | -2 - 35 °C[2](https://chat.openai.com/c/887f46ca-6a67-41b0-b938-e392ef13298e" \l "user-content-fn-2%5E" \t "_new) |
| 盐度 | 无量纲（实际盐度单位） | 33 - 37 PSU[2](https://chat.openai.com/c/887f46ca-6a67-41b0-b938-e392ef13298e" \l "user-content-fn-2%5E" \t "_new) |
| 深度 | 米（m） | 变化大，依据海域 |
| **漂流浮标** | 海面流速 | 米/秒（m/s） | 0 - 2 m/s[3](https://chat.openai.com/c/887f46ca-6a67-41b0-b938-e392ef13298e" \l "user-content-fn-3%5E" \t "_new) |
| 海面温度 | 摄氏度（°C） | -2 - 35 °C[3](https://chat.openai.com/c/887f46ca-6a67-41b0-b938-e392ef13298e" \l "user-content-fn-3%5E" \t "_new) |
| **遥感卫星传感器** | 海面温度 | 摄氏度（°C） | -2 - 35 °C[4](https://chat.openai.com/c/887f46ca-6a67-41b0-b938-e392ef13298e" \l "user-content-fn-4%5E" \t "_new) |
| 海面高度 | 米（m） | 变化大，依据海平面变化 |
| 海面盐度 | 无量纲（实际盐度单位） | 33 - 37 PSU[4](https://chat.openai.com/c/887f46ca-6a67-41b0-b938-e392ef13298e" \l "user-content-fn-4%5E" \t "_new) |
| 海面风速 | 米/秒（m/s） | 0 - 20 m/s[4](https://chat.openai.com/c/887f46ca-6a67-41b0-b938-e392ef13298e" \l "user-content-fn-4%5E" \t "_new) |
| 海浪高度 | 米（m） | 0 - 15 m[4](https://chat.openai.com/c/887f46ca-6a67-41b0-b938-e392ef13298e" \l "user-content-fn-4%5E" \t "_new) |

**二、主船配备设备和维护**

本题定位——主观问题，开放式问题

主成分分析、层次分析法，成本效益分析，多目标优化

本题目根据实际情况来做：时间宽裕按照数学问题，时间不足按照简答题。

**数学问题：成本效益分析模型**

CER=E/C

其中 *CER* 表示成本效益比率， *E* 表示效益值， *C* 表示成本值。



其中，*R* 表示搜救范围，*A* 表示精确度，*D* 表示可靠性。*w*1,*w*2,*w*3 是这些因素的权重，根据实际情况调整。

假设我们有三种搜救设备：声纳、潜水器和无人机。下面的MATLAB代码将计算每种设备的成本效益比率，并确定最佳设备组合。

|  |
| --- |
| % 初始化设备数据  devices = {'声纳', '潜水器', '无人机'};  costs = [50000, 150000, 30000]; % 各设备的总成本（购买、维护、操作）  ranges = [500, 300, 1000]; % 搜救范围  accuracies = [80, 90, 85]; % 精确度  reliabilities = [95, 90, 80]; % 可靠性  % 权重设定  w\_range = 0.4;  w\_accuracy = 0.3;  w\_reliability = 0.3;  % 成本效益比率计算  CERs = zeros(size(devices));  for i = 1:length(devices)  E = w\_range \* ranges(i) + w\_accuracy \* accuracies(i) + w\_reliability \* reliabilities(i);  C = costs(i);  CERs(i) = E / C;  end  % 输出结果  for i = 1:length(devices)  fprintf('%s 的成本效益比率为：%f\n', devices{i}, CERs(i));  end  [~, bestDeviceIndex] = max(CERs);  fprintf('推荐选择的设备是：%s\n', devices{bestDeviceIndex}); |

**输出：**

|  |
| --- |
| **声纳 的成本效益比率为：0.005050**  **潜水器 的成本效益比率为：0.001160**  **无人机 的成本效益比率为：0.014983**  **推荐选择的设备是：无人机** |

**简答题思路：列表格和矩阵选择层次分析方法**

步骤：定义问题和层次结构-》构造判断矩阵-》一致性检验-》结果分析

1. 定义一个判断矩阵（例如，基于某些准则对几个选项进行评价）。
2. 计算判断矩阵的最大特征值及其对应的特征向量（即优先权向量）。
3. 进行一致性检验，确保判断的一致性可接受。

|  |
| --- |
| % 定义判断矩阵  A = [1 2 5; 1/2 1 2; 1/5 1/2 1];  % 计算特征值和特征向量  [V,D] = eig(A);  % 提取最大特征值及其对应的特征向量  [maxEigVal, maxEigValIndex] = max(max(real(D)));  priorityVector = real(V(:, maxEigValIndex)) / sum(real(V(:, maxEigValIndex)));  % 输出优先权向量  disp('优先权向量:');  disp(priorityVector);  % 计算一致性指标CI  n = size(A,1);  CI = (maxEigVal - n) / (n - 1);  % 查找相应的平均随机一致性指数（RI）  % 这里给出常见的RI值，适用于不同大小的判断矩阵  RI = [0 0 0.58 0.9 1.12 1.24 1.32 1.41 1.45];  % 计算一致性比率CR  CR = CI / RI(n);  % 输出一致性检验结果  disp('一致性比率CR:');  disp(CR);  if CR < 0.1  disp('一致性可接受。');  else  disp('一致性不可接受，请重新评估判断矩阵。');  end |

运行结果：



当然也可以直接根据相关信息，查找数据和资料列表对比。

公司在主船上配备以下额外的搜索设备：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **设备名称** | **类型** | **可用性** | **维护要求** | **准备情况** | **成本** | **优点** | **缺点** |
| **多波束声纳** | 定位与映射设备 | 高 | 中等 | 快速 | 高 | 高精度地图绘制，大范围搜索 | 成本高，需要专业操作 |
| **侧扫声纳** | 搜索设备 | 高 | 中等 | 快速 | 中 | 广泛用于海底搜索，能发现沉船残骸 | 分辨率受水深影响 |
| **自主水下无人机（AUV）** | 探索设备 | 中 | 高 | 中等 | 高 | 深水探索能力强，可携带多种仪器 | 维护成本高，操作复杂 |
| **水下机器人（ROV）** | 操作设备 | 高 | 高 | 中等 | 高 | 可进行精细操作，如打捞、拍照 | 受控于船上操作，有线连接限制 |
| **救生筏** | 救援设备 | 高 | 低 | 快速 | 低 | 快速部署，用于海面救援 | 仅限于水面使用，不能深水救援 |
| **通信浮标** | 通信设备 | 高 | 中等 | 快速 | 中 | 可以提供潜水器和主船之间的通信中继点 | 受天气和海面条件影响 |

需要有足够的引文和资料支撑。

产品需要比较精确。

完成相关分类（最前边再加一列）

可以考虑的产品：

多波束声纳：用于海底地形测绘，能够快速提供大范围内的海底地图，有助于确定潜水器可能的位置。

侧扫声纳：特别适用于搜索大面积海底，寻找潜水器或其残骸。侧扫声纳可以提供高分辨率的海底图像。

自主水下无人机（AUV）：配备声纳和摄像头的AUV能够进行深水搜索，可编程以执行特定搜索模式，覆盖大范围区域。

遥控水下机器人（ROV）：在确定潜水器大致位置后，使用ROV进行精细搜索。ROV可以携带高清摄像头和操控臂，用于实地确认和必要的救援操作。

浮标和定位标签：用于标记已搜索区域和潜水器可能的位置，以及作为潜水器和救援船之间的通信中继点。

水下通信系统：提高潜水器与主船之间的通信效率，包括声通信和可能的水下数据传输系统。

紧急定位信标（EPIRB）：一旦潜水器遇险，可以手动或自动触发，发送求救信号和位置信息到卫星，由卫星转发给救援中心。

三、**开发一个模型，该模型将使用来自您的位置模型的信息来推荐设备的初始部署点和搜索模式，以便最大限度地缩短定位丢失潜水器的时间。确定找到潜水器的概率作为时间和累积搜索结果的函数。**

搜索方法：关键点搜索，面搜索

初始点：考虑选择概率最大的，最深的，考虑环境对初始点的影响（响应时间等）

搜索模式:需要结合实际情况考虑（洋流影响，潜水器在水中的运动轨迹）

潜水器常用的搜索方法及其数学建模：

**圆形搜索（Circle Search）：**这是一种相对简单直接的方法，通常围绕一个固定的参考点进行一系列圆形搜索，逐步增加圆的周长。

**杰克斯塔（Jackstay）搜索：**这是一种使用潜水员或遥控水下航行器（ROV）沿着搜索线——杰克斯塔进行的搜索程序。杰克斯塔是两个固定点或锚点之间的线。搜索时，潜水员从一个锚点开始，沿线搜索直到达到第二个锚点。然后移动第二个锚点以移动线，并继续搜索

网格搜索（Grid Search）、弧形搜索（Arc Search）挡风玻璃雨刷搜索（Windshield-Wiper Search）等等。根据实际的潜水器运动的轨迹建立相关的动力学方程。

**缺少时间的简单做法:**找出来关键点，按照旅行商问题的思路做优化，设计相关的启发式算法求得搜索的路径。（可能会忽略时间的影响）

|  |
| --- |
| function tsp\_ga  % 遗传算法解决三维旅行商问题（TSP）  % 参数设置  cityNum = 15; % 城市数量  popSize = 50; % 种群大小  maxGen = 100; % 最大迭代次数  crossRate = 0.8; % 交叉率  mutationRate = 0.1; % 变异率  elitismNum = 1; % 精英保留数量  % 随机生成城市坐标  cities = rand(cityNum, 3) \* 100;  % 生成初始种群  pop = zeros(popSize, cityNum);  for i = 1:popSize  pop(i,:) = randperm(cityNum);  end  % 适应度评估  fitness = zeros(1, popSize);  for i = 1:popSize  fitness(i) = calcFitness(pop(i,:), cities);  end  % 进化开始  for gen = 1:maxGen  % 选择操作  selectedPop = select(pop, fitness, popSize - elitismNum);    % 交叉操作  crossedPop = crossover(selectedPop, crossRate);    % 变异操作  mutatedPop = mutate(crossedPop, mutationRate);    % 精英策略  [maxFitness, elitismIndex] = max(fitness);  elitismIndividual = pop(elitismIndex, :);    % 更新种群  pop = [mutatedPop; elitismIndividual];    % 评估新的种群  for i = 1:popSize  fitness(i) = calcFitness(pop(i,:), cities);  end    % 打印信息  [maxFitness, index] = max(fitness);  disp(['Generation: ', num2str(gen), ' Best Fitness: ', num2str(maxFitness)]);    % 可视化  plot3(cities(:,1), cities(:,2), cities(:,3), 'ro');  hold on;  bestRoute = pop(index,:);  for i = 1:length(bestRoute)-1  plot3([cities(bestRoute(i),1), cities(bestRoute(i+1),1)],...  [cities(bestRoute(i),2), cities(bestRoute(i+1),2)],...  [cities(bestRoute(i),3), cities(bestRoute(i+1),3)], 'b-');  end  plot3([cities(bestRoute(end),1), cities(bestRoute(1),1)],...  [cities(bestRoute(end),2), cities(bestRoute(1),2)],...  [cities(bestRoute(end),3), cities(bestRoute(1),3)], 'b-');  hold off;  drawnow;  end  % 计算路径长度  function totalDist = calcFitness(route, cities)  totalDist = 0;  for i = 1:length(route)-1  totalDist = totalDist + norm(cities(route(i),:) - cities(route(i+1),:));  end  totalDist = totalDist + norm(cities(route(end),:) - cities(route(1),:));  end  % 选择操作  function selectedPop = select(pop, fitness, selectNum)  selectedPop = zeros(selectNum, size(pop,2));  fitProb = fitness / sum(fitness);  fitCumsum = cumsum(fitProb);  for i = 1:selectNum  r = rand;  for j = 1:length(fitCumsum)  if r <= fitCumsum(j)  selectedPop(i,:) = pop(j,:);  break;  end  end  end  end  % 交叉操作  function crossedPop = crossover(pop, crossRate)  crossedPop = pop;  pairNum = size(pop,1) / 2;  for i = 1:pairNum  if rand < crossRate  crossPoint = randi([1, size(pop,2)-1]);  crossedPop(2\*i-1:2\*i,:) = cross(pop(2\*i-1:2\*i,:), crossPoint);  end  end  end  function children = cross(parents, point)  children = parents;  children(1,point+1:end) = parents(2,point+1:end);  children(2,point+1:end) = parents(1,point+1:end);  end  % 变异操作  function mutatedPop = mutate(pop, mutationRate)  mutatedPop = pop;  for i = 1:size(pop,1)  if rand < mutationRate  mutationPoints = randperm(size(pop,2),2);  temp = mutatedPop(i,mutationPoints(1));  mutatedPop(i,mutationPoints(1)) = mutatedPop(i,mutationPoints(2));  mutatedPop(i,mutationPoints(2)) = temp;  end  end  end  end |

**四、如何扩展您的模型以考虑加勒比海等其他旅游目的地？您的模型将如何改变以考虑在同一区域内移动的多个潜水器？**

两个问题：地点迁移和多智能体的应用。

地点迁移：主要区别在于每个地点的扰动可能会不相同。

**整体思路：**潜水器的运动模型+**环境模型**=潜水器的实际运动模型，环境问题即为本问题需要解决的。

可以考虑洋流的连续动态系统模型：



* *v*(*t*)是洋流在时间*t*的速度。
* *dtdv*是洋流速度的变化率。
* *α*是摩擦系数，表示摩擦力对洋流速度的减缓作用。
* *β*是外部驱动力对洋流速度影响的系数。
* *F*是外部驱动力，可以由风力、温差等因素综合决定，这里假设为常数以简化模型。

|  |
| --- |
| % 参数设置  alpha = 0.1; % 摩擦系数  beta = 2; % 外部驱动力影响系数  F = 1; % 外部驱动力，假设为常数  % 初始条件  v0 = 0; % 初始时刻洋流速度  % 时间设置  tspan = [0 50]; % 模拟时间：从0到50  % 定义洋流速度变化的微分方程  dvdt = @(t, v) -alpha \* v + beta \* F;  % 使用ode45求解微分方程  [t, v] = ode45(dvdt, tspan, v0);  % 绘制结果  plot(t, v, 'LineWidth', 2);  xlabel('时间 t');  ylabel('洋流速度 v(t)');  title('洋流速度随时间的变化');  grid on; |

**考虑相关的洋流后的方案：**

|  |
| --- |
| **% 初始化随机数生成器，确保每次运行结果都不同**  **rng('shuffle');**  **% 模拟参数 - 引入随机性和环境因素**  **initialPosition = rand(1, 3) \* 10; % 初始位置，随机值范围[0, 10]**  **velocity = rand(1, 3) \* 5; % 初始速度，随机值范围[0, 5]**  **acceleration = rand(1, 3) \* 0.2 - 0.1; % 加速度，随机值范围[-0.1, 0.1]**  **angularVelocity = rand \* 2 \* pi; % 角速度，随机值范围[0, 2π]**  **angularAcceleration = rand \* 0.2 - 0.1; % 角加速度，随机值范围[-0.1, 0.1]**  **deltaTime = 1; % 时间步长，每秒更新一次**  **simulationDuration = 40; % 模拟总时长（秒），保持不变**  **k = 20; % k秒后潜水艇失去推进力**  **% 初始化位置存储**  **positions = zeros(simulationDuration + 1, 3);**  **positions(1, :) = initialPosition;**  **% 环境影响因素初始化（简化模型）**  **envEffect = [0.1, 0.1, -0.02]; % 假设环境因素导致的速度变化**  **envAngularEffect = 0.05; % 假设环境因素导致的角速度变化**  **% 模拟运动**  **for t = 1:simulationDuration**  **% 更新位置**  **currentPosition = positions(t, :);**    **% 引入随机漫步模型来模拟海流的随机影响**  **randomComponent = rand(1, 3) \* 0.6 - 0.3;**    **% 在k秒后失去推进力**  **if t > k**  **acceleration = [0, 0, 0]; % 加速度设置为零**  **end**    **% 更新速度和角速度，加入随机分量和环境影响**  **velocity = velocity + acceleration \* deltaTime + randomComponent + envEffect;**  **angularVelocity = angularVelocity + angularAcceleration \* deltaTime + envAngularEffect;**    **% 根据更新后的速度和角速度计算新位置**  **newPosition = predictPosition(currentPosition, velocity, acceleration, angularVelocity, angularAcceleration, deltaTime);**    **% 存储新位置**  **positions(t + 1, :) = newPosition;**  **end**  **% 绘制三维轨迹**  **figure;**  **plot3(positions(:, 1), positions(:, 2), positions(:, 3), 'LineWidth', 2);**  **grid on;**  **xlabel('X');**  **ylabel('Y');**  **zlabel('Z');**  **title('潜水器三维运行轨迹');**  **hold on;**  **% 逐秒展示位置**  **for t = 1:size(positions, 1)**  **scatter3(positions(t, 1), positions(t, 2), positions(t, 3), 'filled');**  **pause(0.1); % 暂停0.1秒，以模拟实时更新**  **end** |

**相关参考文献：**

1. AirMed&Rescue.. "Innovation for all: latest advancements in SAR equipment." Retrieved from https://www.airmedandrescue.com
2. Zeng, Q., Luo, Y., & Zhang, L. (2021). "Search and rescue system-of-systems influence degree evaluation of aviation equipment based on simulation." Scientific Reports, 11, 12345. https://doi.org/10.1038/s41598-021-91870-2
3. Christensen, B. L., & Sciberras, A. (2019). "A COST-BENEFIT ANALYSIS OF NAVY STATION SEARCH AND RESCUE (SAR) LOCATED IN KEY WEST, FL, AND WHIDBEY ISLAND, WA." Retrieved from https://archive.org/details/acostbenefitanal1094562751
4. Defense Advancement. (2023). "Combat Search & Rescue (CSAR) | Military SAR Equipment & Technology." Retrieved from https://www.defenseadvancement.com