CHAPTER 3



第 3 章　水下目标跟踪中的节点调度理论

本章探讨了水下目标跟踪中节点调度相关的技术体系。在水下目标跟踪过程中，参与目标跟踪的传感器节点数越多，传感器节点的采样周期越短，传感器节点能够采集到的目标的量测信息就会越多，目标的跟踪精度也会越高。然而，参与目标跟踪的节点数过多或节点发送量测信息次数过于频繁都会导致 UWSN 消耗大量的能量。因此，如何有效地解决“高精度-低能耗”的矛盾是水下目标跟踪技术领域的重要课题之一。节点调度技术就是根据特定的准则引入传感器节点管理技术，在目标跟踪精度满足特定需求的情况下，最大限度地降低整个系统的能耗。在此背景下，本章从“节点深度”“能量分配”以及“动态分簇”3 个角度入手，详细讨论了这些因素对水下目标跟踪性能的影响，在此基础上研究相应的调度优化问题设计，并结合水下目标跟踪算法，实现了目标跟踪精度与能耗之间的有效权衡。

3.1　基于深度调节的节点调度方法

水下节点的部署是 UWSN 系统组网的基础，良好的节点部署方案能够增强网络的信息感知能力，降低数据通信成本，为开展 UWSN 应用提供了有利的支撑。由于水下节点的成本相对较高，目前很难实现和陆地传感器网络相似的密集部署方式，在实际环境中更常见的是在监测水域部署少量节点，组建稀疏的 UWSN 系统，这就可能降低个别区域节点覆盖率和探测能力。在 UWSN 目标跟踪问题中，当目标在多个节点的公共区域或者距离节点较近的范围内运动时，节点获得的量测数据比较完备，可以提供充足的信息以支撑目标跟踪任务。而当目标运动到节点覆盖率较低的区域时，由于网络的探测能力有限，系统可能无法获得足够的有效信息，无法实现满足精度的跟踪结果。为了解决节点稀疏部署导致的网络覆盖、探测能力不足的问题，近年来节点的深度调节机制逐渐成为了 UWSN 部署等领域的研究热点。节点通过自身搭载的深度调节装置，可以结合 UWSN 具体的工作需求，沿深度方向垂直移动，自适应地改变网络通信拓扑和节点之间的位置结构，能够在部署少数节点的情况下，提高网络的覆盖、通信能力，保障 UWSN 稳定、灵活地工作。目前，节点的深度调节机制主要应用于 UWSN 节点部署、路由协议等研究领域[84-86]，针对 UWSN 目标跟踪的研究应用比较匮乏。由于 UWSN 节点之间的位置结构对于目标跟踪的精度有着重要影响，良好的节点位置结构不仅可以扩大网络的覆盖范围，还可以从多个方位获取更加完备的目标量测数据，为融合中心进行目标状态估计提供充足的信息支撑。而节点深度调节机制可以帮助 UWNS 构成多样、灵活的节点位置结构，提高网络的监测范围和探测能力，这对于目标跟踪问题有着重要意义。因此，如何合理利用节点深度调节机制，通过节点深度的动态调节，实现节点位置结构的自组织调整，克服 UWSN 节

点稀疏部署导致的量测信息不足的弊端，进而提高目标跟踪的精度是本节需要解决的问题。

针对这一问题，本节提出了基于深度调节的 UWSN 节点调度方法。通过节点在深度方向的垂直移动，自适应地调整节点之间的位置结构，从而获得更加完备的量测信息，提高目标跟踪的精度。为了衡量节点深度调节对目标跟踪精度的影响，本节研究了节点深度变化与费舍尔信息矩阵之间的定量关系，并以此作为目标跟踪精度的判据，结合节点深度调节的移动能耗建立了目标函数，并通过优化问题的求解获得节点的深度调节距离，自组织地改变了节点的位置结构，解决了 UWSN 节点部署稀疏导致的量测信息不足等问题，实现了目标跟踪精度的提高。

3.1.1 跟踪模型

水下运动目标的状态模型描述为

$$\boldsymbol{x}_{k+1}=\boldsymbol{F}_k\boldsymbol{x}_k+\boldsymbol{w}_k \tag{3-1}$$

其中，$\boldsymbol{x}_k=[x_k,\dot{x}_k,y_k,\dot{y}_k,z_k,\dot{z}_k]^{\mathrm{T}}\in\mathbb{R}^6$ 表示目标在 k 时刻的运动状态；(x_k,y_k,z_k) 表示 k 时刻目标在空间坐标系中的位置；$(\dot{x}_k,\dot{y}_k,\dot{z}_k)$ 则表示目标在相应方向的速度；$\boldsymbol{F}_k\in\mathbb{R}^{6\times6}$ 表示运动目标在 k 时刻的状态转移矩阵；$\boldsymbol{w}_k$ 表示服从正态分布 $N(\boldsymbol{0}_6,\boldsymbol{Q}_k)$ 的过程噪声。

搭载了水听器的探测节点部署在监测水域的不同位置，以获取运动目标信息，其中节点 s 的位置表示为 $\boldsymbol{p}^s=[x^s,y^s,z^s]^{\mathrm{T}}$。节点 s 在 $k+1$ 时刻的量测模型可以表示为

$$z_{k+1}^s=h_{k+1}^s(\boldsymbol{x}_{k+1},\boldsymbol{p}^s)+v_{k+1}^s \tag{3-2}$$

其中，$h_{k+1}^s:\mathbb{R}^6\times\mathbb{R}^3\mapsto\mathbb{R}$ 表示节点 s 的量测方程；v_{k+1}^s 表示服从正态分布 $N(0,R_{k+1}^s)$ 的量测噪声。假定各节点量测噪声独立同分布且不随时间变化，并设节点获取量测为关于目标的纯距离量测，此时

$$h_{k+1}^s(\boldsymbol{x}_{k+1},\boldsymbol{p}^s)=\sqrt{(x_{k+1}-x^s)^2+(y_{k+1}-y^s)^2+(z_{k+1}-z^s)^2} \tag{3-3}$$

假设每个时刻有 N 个节点参与目标跟踪任务，在集中式融合中节点将各自的量测数据发送至融合中心，融合中心在获取多传感器的量测 $\boldsymbol{Z}_{k+1}=[z_{k+1}^1,z_{k+1}^2,\cdots,z_{k+1}^N]^{\mathrm{T}}$ 后，可以利用 EKF 等算法对目标状态进行估计，相应的多传感器量测联合似然函数为

$$p(\boldsymbol{Z}_{k+1}\mid\boldsymbol{x}_{k+1})=\prod_{s=1}^{N}p(z_{k+1}^s\mid\boldsymbol{x}_{k+1}) \tag{3-4}$$

其中，单个节点 s 的量测似然 $p(z_{k+1}^s\mid\boldsymbol{x}_{k+1})$ 为

$$p(z_{k+1}^s\mid\boldsymbol{x}_{k+1})=\frac{1}{\sqrt{2\pi R_{k+1}^s}}\exp\left\{-\frac{1}{2R_{k+1}^s}[z_{k+1}^s-h_{k+1}^s(\boldsymbol{x}_{k+1},\boldsymbol{p}^s)]^2\right\} \tag{3-5}$$

节点的深度调节能力保证了节点在深度方向的位置调整，节点可以移动到监测水域的不同深度，实现特定区域的网络连通和探测。目前已有很多关于实现节点深度调节的工程研究，如在文献[87]中，节点通过活塞式的气泵可以吸入或排出水，通过改变浮力实现节点的上下移动；在文献[88]中，通过缆绳锚在不同深度的节点，可以利用卷扬机调整缆绳的长度，实现节点的深度移动；这些工程成果为关于节点深度调节机制的理论研究和应用拓展提供了基础和思路。图 3.1 给出了 UWSN 节点深度调节示意图，节点通过缆绳锚在水域不同的位置，在探测目标过程中，可以根据需求调节自身深度。假定节点在每个采样间隔内可以移动的范围为 $[-d_{\max},d_{\max}]$，其中，正值表示向下移动，深度增加；负值表示向上移动，深度减小。因此，节点的深度变化为

$$Z_{k+1}^s=Z_k^s+d_{k+1}^s \tag{3-6}$$

其中，Z_k^s 表示 k 时刻的节点深度，d_{k+1}^s 表示节点移动的距离。

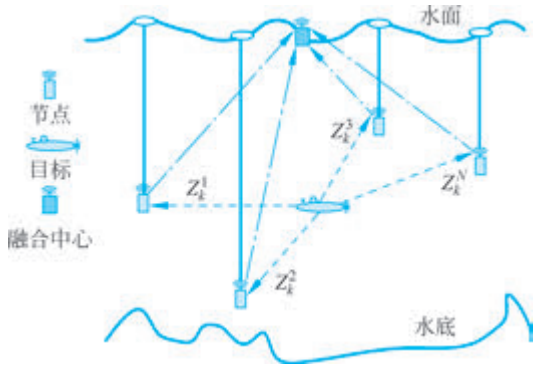


图 3.1 UWSN 节点深度调节示意图

3.1.2 节点深度对跟踪性能的影响

节点之间的位置结构对目标跟踪精度有着重要影响，良好的节点位置结构可以获得更加完备的目标量测信息，进而实现目标跟踪精度的提高。与静态节点相比，节点的深度调节可以动态地改变节点位置，实现灵活多样的节点位置结构，进而获得更加完备的量测数据。如图 3.2 和图 3.3 所示，当几个节点位于同一平面时，根据量测数据可能估计出错误位置的虚假目标，而当节点构成相对立体的结构时，可以确定唯一目标。因此，为了设计目标跟踪过程中合理的节点深度调节方案，有必要对节点深度和目标跟踪精度之间的关系进行研究。由于节点在深度调节过程中，与目标之间在水平面上的距离并不受影响，只会改变节点与目标之间的相对高度。因此，为了更准确地衡量目标跟踪精度，分别从目标深度已知和目标深度未知两种情况考虑，研究节点深度变化的影响。

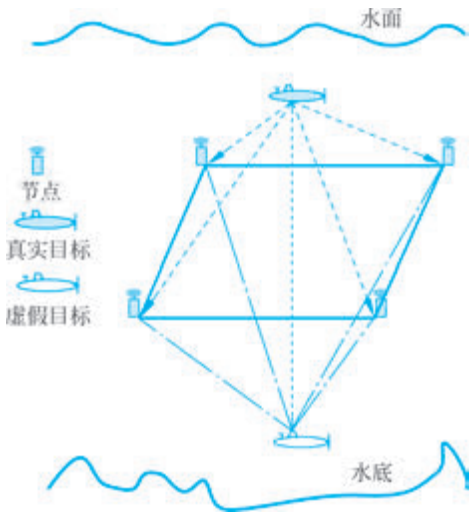


图 3.2 多个节点共面时可能出现虚假目标

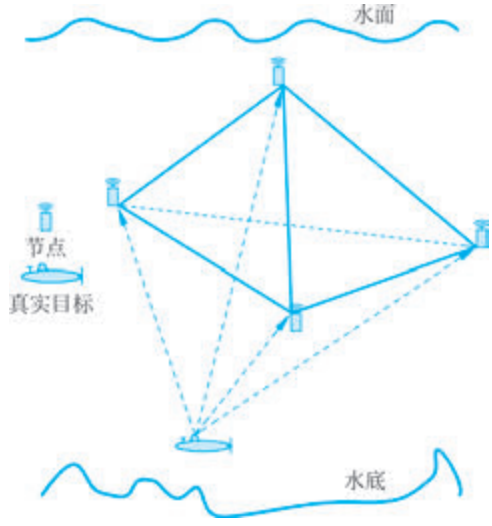


图 3.3 多个节点立体构成结构时只有唯一目标

假设目标的深度 Z_k 为已知信息，则目标可类似为二维平面运动的模型，仅需对水平面上的目标状态进行估计。相应地，式(3-3)可改写为

$$h_{k+1}^s(\boldsymbol{x}_{k+1},\boldsymbol{p}^s)=\sqrt{(x_{k+1}-x^s)^2+(y_{k+1}-y^s)^2+(h-Z_{k+1}^s)^2} \tag{3-7}$$

其中，h 为已知的目标深度。

为了评估目标跟踪的实时精度，可以计算量测数据的 FIM。FIM 的逆定义了目标状态估计的均方

误差下界，它是目标跟踪问题中用于衡量跟踪精度的经典判据，更大的 FIM 意味着量测数据可以提供更多的目标信息[89]。因此，为了通过节点深度调节，实现目标跟踪精度的提升，需要研究节点深度位置与 FIM 之间的定量关系，进而通过最大化 FIM 设计节点深度调节方案，构成最佳的节点位置结构，提高量测数据的信息量，减小目标跟踪的误差。令 $\boldsymbol{J}_{k+1}$ 表示 FIM，其逆给出了目标状态估计均方误差的下界。

$$E\left[(\hat{\boldsymbol{x}}_{k+1}-\boldsymbol{x}_{k+1})(\hat{\boldsymbol{x}}_{k+1}-\boldsymbol{x}_{k+1})^{\mathrm{T}}\right]\geqslant\boldsymbol{J}_{k+1}^{-1} \tag{3-8}$$

由于仅需要估计目标水平面上的位置(x_{k+1},y_{k+1})，因此可以暂时将目标状态简化为 $\boldsymbol{x}_{k+1}=[x_{k+1},y_{k+1}]^{\mathrm{T}}$，此时 $\boldsymbol{J}_{k+1}$ 可写为一个 2×2 的矩阵

$$\boldsymbol{J}_{k+1}(i,j)=E_{p(\boldsymbol{Z}_{k+1},\boldsymbol{x}_{k+1})}\left[-\frac{\partial^2\log p(\boldsymbol{Z}_{k+1},\boldsymbol{x}_{k+1})}{\partial\boldsymbol{x}_{k+1}(i)\partial\boldsymbol{x}_{k+1}(j)}\right],\quad i,j=1,2 \tag{3-9}$$

其中，$\boldsymbol{J}_{k+1}(i,j)$表示矩阵 $\boldsymbol{J}_{k+1}$ 的(i,j)元素；$E_{p(\boldsymbol{Z}_{k+1},\boldsymbol{x}_{k+1})}[\cdot]$表示关于联合分布 $p(\boldsymbol{Z}_{k+1},\boldsymbol{x}_{k+1})$的期望；$\boldsymbol{x}_{k+1}(i)$表示目标状态 $\boldsymbol{x}_{k+1}$ 的第 i 个元素。

根据式(3-2)与式(3-7)，在给定目标状态 $\boldsymbol{x}_{k+1}$ 下，各节点的量测 z_{k+1}^{s} 服从均值非零的正态分布。因此，基于文献[90]中的计算方法，可以进一步化简费舍尔信息。

$$\boldsymbol{J}_{k+1}(i,j)=\left[\frac{\partial\boldsymbol{h}_{k+1}(\boldsymbol{x}_{k+1})}{\partial\boldsymbol{x}_{k+1}(i)}\right]^{\mathrm{T}}\boldsymbol{R}_{k+1}^{-1}\left[\frac{\partial\boldsymbol{h}_{k+1}(\boldsymbol{x}_{k+1})}{\partial\boldsymbol{x}_{k+1}(j)}\right] \tag{3-10}$$

其中，$\boldsymbol{h}_{k+1}(\boldsymbol{x}_{k+1})=[h_{k+1}^{1}(\boldsymbol{x}_{k+1},\boldsymbol{p}^{1}),h_{k+1}^{2}(\boldsymbol{x}_{k+1},\boldsymbol{p}^{2}),\cdots,h_{k+1}^{N}(\boldsymbol{x}_{k+1},\boldsymbol{p}^{N})]^{\mathrm{T}}$；$\boldsymbol{R}_{k+1}=\mathrm{diag}\{R_{k+1}^{1},R_{k+1}^{2},\cdots,R_{k+1}^{N}\}$。根据式(3-7)计算$\partial h_{k+1}^{s}(\boldsymbol{x}_{k+1},\boldsymbol{p}^{s})/\partial\boldsymbol{x}_{k+1}(1)$，可得

$$\frac{\partial h_{k+1}^{s}(\boldsymbol{x}_{k+1},\boldsymbol{p}^{s})}{\partial\boldsymbol{x}_{k+1}(1)}=\frac{\boldsymbol{x}_{k+1}-x^{s}}{\sqrt{(x_{k+1}-x^{s})^2+(y_{k+1}-y^{s})^2+(h-Z_{k+1}^{s})^2}} \tag{3-11}$$

由此可以得到 $\boldsymbol{J}_{k+1}(1,1)$为

$$\boldsymbol{J}_{k+1}(1,1)=\sum_{s=1}^{N}\frac{(x_{k+1}-x^{s})^2}{\left[(x_{k+1}-x^{s})^2+(y_{k+1}-y^{s})^2+(h-Z_{k+1}^{s})^2\right]R_{k+1}^{s}} \tag{3-12}$$

同理，可以得到

$$\begin{cases}\boldsymbol{J}_{k+1}(1,2)=\boldsymbol{J}_{k+1}(2,1)=\displaystyle\sum_{s=1}^{N}\frac{(x_{k+1}-x^{s})(y_{k+1}-y^{s})}{\left[(x_{k+1}-x^{s})^2+(y_{k+1}-y^{s})^2+(h-Z_{k+1}^{s})^2\right]R_{k+1}^{s}}\\ \boldsymbol{J}_{k+1}(2,2)=\displaystyle\sum_{s=1}^{N}\frac{(y_{k+1}-y^{s})^2}{\left[(x_{k+1}-x^{s})^2+(y_{k+1}-y^{s})^2+(h-Z_{k+1}^{s})^2\right]R_{k+1}^{s}}\end{cases} \tag{3-13}$$

结合式(3-12)及式(3-13)，可以得到 $\boldsymbol{J}_{k+1}$ 为

$$\boldsymbol{J}_{k+1}=\sum_{s=1}^{N}\frac{\begin{bmatrix}(x_{k+1}-x^{s})^2 & (x_{k+1}-x^{s})(y_{k+1}-y^{s})\\ (x_{k+1}-x^{s})(y_{k+1}-y^{s}) & (y_{k+1}-y^{s})^2\end{bmatrix}}{\left[(x_{k+1}-x^{s})^2+(y_{k+1}-y^{s})^2+(h-Z_{k+1}^{s})^2\right]R_{k+1}^{s}} \tag{3-14}$$

由式(3-14)可以看出，节点深度 Z_{k+1}^{s} 的变化会使距离量测 h_{k+1}^{s} 改变，进而对 $\boldsymbol{J}_{k+1}$ 产生影响。为了调节节点深度来提高目标跟踪精度，可以将 $\boldsymbol{J}_{k+1}$ 作为性能判据建立目标函数，通过最大化 $\boldsymbol{J}_{k+1}$ 来求解节点的最优深度位置。根据节点深度调节的模型(3-6)，假定各节点的深度移动距离为 $\boldsymbol{D}_{k+1}=[d_{k+1}^{1},d_{k+1}^{2},\cdots,d_{k+1}^{N}]^{\mathrm{T}}$，则可以根据下式求解最优深度调节距离

$$\boldsymbol{D}_{k+1}^{*}=\arg\max\det(\boldsymbol{J}_{k+1})$$

$$\text{s.t.}\ -d_{\max} \leqslant d_{k+1}^{s} \leqslant d_{\max},\quad s \in \{1,2,\cdots,N\} \tag{3-15}$$

其中，$\det(\boldsymbol{J}_{k+1})$表示 FIM 的行列式。根据式(3-14)，可以计算 $\det(\boldsymbol{J}_{k+1})$为

$$\begin{aligned}\det(\boldsymbol{J}_{k+1}) = &\left[\sum_{s=1}^{N}\frac{(y_{k+1}-y^{s})^{2}}{[(x_{k+1}-x^{s})^{2}+(y_{k+1}-y^{s})^{2}+(h-Z_{k+1}^{s})^{2}]R_{k+1}^{s}}\right]\times\\ &\left[\sum_{s=1}^{N}\frac{(x_{k+1}-x^{s})^{2}}{[(x_{k+1}-x^{s})^{2}+(y_{k+1}-y^{s})^{2}+(h-Z_{k+1}^{s})^{2}]R_{k+1}^{s}}\right]-\\ &\left[\sum_{s=1}^{N}\frac{(x_{k+1}-x^{s})(y_{k+1}-y^{s})}{[(x_{k+1}-x^{s})^{2}+(y_{k+1}-y^{s})^{2}+(h-Z_{k+1}^{s})^{2}]R_{k+1}^{s}}\right]^{2}\end{aligned} \tag{3-16}$$

令

$$\alpha_{k+1}^{s} = \frac{1}{[(x_{k+1}-x^{s})^{2}+(y_{k+1}-y^{s})^{2}+(h-Z_{k+1}^{s})^{2}]R_{k+1}^{s}} \tag{3-17}$$

则式(3-16)可以继续化简为

$$\begin{aligned}\det(\boldsymbol{J}_{k+1}) = &\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_{k+1}^{i}\alpha_{k+1}^{j}[(x^{i}-x_{k+1})^{2}(y^{j}-y_{k+1})^{2}]-\\ &\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_{k+1}^{i}\alpha_{k+1}^{j}[(x^{i}-x_{k+1})(y^{i}-y_{k+1})(y^{j}-y_{k+1})(x^{j}-x_{k+1})]\\ = &\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_{k+1}^{i}\alpha_{k+1}^{j}[(x^{i}-x_{k+1})(y^{j}-y_{k+1})-(x^{j}-x_{k+1})(y^{i}-y_{k+1})]^{2}\end{aligned} \tag{3-18}$$

在式(3-18)中，系数 α_{k+1}^{i} 包含了节点深度位置对费舍尔信息的值产生影响，$[x^{i}-x_{k+1})(y^{j}-y_{k+1})-(x^{j}-x_{k+1})(y^{i}-y_{k+1})]^{2}$ 则体现了节点之间水平位置结构的影响，这说明了良好的节点部署对于目标跟踪很重要。因为深度调节过程中节点的水平位置保持不变，且 α_{k+1}^{i} 和$[(x^{i}-x_{k+1})(y^{j}-y_{k+1})-(x^{j}-x_{k+1})(y^{i}-y_{k+1})]^{2}$ 都是非负值，所以 $\det(\boldsymbol{J}_{k+1})$的大小随 α_{k+1}^{i} 增加单调递增，而 α_{k+1}^{i} 的值随节点与目标的距离增加单调递减。因此，为了使 $\det(\boldsymbol{J}_{k+1})$的值最大，需要尽量减小每一个节点与目标之间的高度差，使两者的距离最小。综上，可以得到最优的节点深度调节距离。

$$\begin{aligned}&d_{k+1}^{s} = \arg\min\left|Z_{k}^{s}+d_{k+1}^{s}-h\right|\\ &\text{s.t.}\ -d_{\max} \leqslant d_{k+1}^{s} \leqslant d_{\max}\end{aligned} \tag{3-19}$$

因此，对于目标深度已知的情况，当节点位于目标上方时，节点应向下移动，当节点位于目标下方时，节点应向上移动，使节点与目标之间的高度差最小，这样可以获取更好的量测数据，提高目标跟踪的精度。

对于目标深度未知的情况，需要对目标 x、y、z 这 3 个方向的位置进行估计，相应的费舍尔信息是一个 3×3 的矩阵。

$$\boldsymbol{J}_{k+1} = \sum_{s=1}^{N}\alpha_{k+1}^{s}\begin{bmatrix}(x^{s}-x_{k+1})^{2} & (x^{s}-x_{k+1})(y^{s}-y_{k+1}) & (x^{s}-x_{k+1})(z^{s}-z_{k+1})\\ (y^{s}-y_{k+1})(x^{s}-x_{k+1}) & (y^{s}-y_{k+1})^{2} & (y^{s}-y_{k+1})(z^{s}-z_{k+1})\\ (z^{s}-z_{k+1})(x^{s}-x_{k+1}) & (z^{s}-z_{k+1})(y^{s}-y_{k+1}) & (z^{s}-z_{k+1})^{2}\end{bmatrix} \tag{3-20}$$

同样，需要最大化 $\det(\boldsymbol{J}_{k+1})$来计算最优深度。然而，目标深度未知时，三阶费舍尔信息矩阵的行列式计算比较复杂，很难获得像式(3-18)一样直观的结果。因此，为了以较快的速度计算并保证求解质量，可以采用启发式的智能优化方法进行搜索计算。具体的优化算法与完整的跟踪方案将在 3.1.3 节给出。

3.1.3 基于深度调节的水下目标跟踪算法

和声搜索算法(Harmony Search,HS)是2001年在文献[91]中提出的一种元启发式算法,它通过模拟音乐家在演奏过程中寻找最优和声的方式进行搜索。和声搜索算法的计算过程相对简单灵活,可以高效地求解复杂优化问题。因此,本节利用和声搜索算法求解节点的深度移动距离。所有节点的深度移动距离构成的向量 $\boldsymbol{D}_{k+1}=[d_{k+1}^1,d_{k+1}^2,\cdots,d_{k+1}^N]^{\mathrm{T}}$ 被定义为一个和声,它是优化问题(3-15)的一个可行解。和声搜索算法通过初始化和声、改进和声、更新和声库的过程逐渐搜索问题的最优解。下面介绍其具体过程。

1. 初始化

首先需要设置必要的参数,主要包括:和声库大小 H_{m},它决定和声库里的和声个数;和声记忆保留概率 p_{cr},它决定生成新和声的来源;记忆扰动概率 p_{ar},它决定和声的调整方式;微调带宽 b_{m},它用来调整和声的数值。

设置好参数后,在节点的深度移动范围 $[-d_{\max},d_{\max}]$ 上,随机初始化 H_{m} 个和声,组成和声库 $\boldsymbol{H}_{\mathrm{m}}$,其中每行表示一个和声。

$$\boldsymbol{H}_{\mathrm{m}}=\begin{bmatrix} d_1^1 & d_1^2 & \cdots & d_1^{N-1} & d_1^N \\ d_2^1 & d_2^2 & \cdots & d_2^{N-1} & d_2^N \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ d_{H_{\mathrm{m}}}^1 & d_{H_{\mathrm{m}}}^2 & \cdots & d_{H_{\mathrm{m}}}^{N-1} & d_{H_{\mathrm{m}}}^N \end{bmatrix} \tag{3-21}$$

2. 改进和声

为了得到问题的最优解,这一步需要对和声库内的和声进行修正改进。首先需要生成一个新的和声,新和声的每一个元素对应了一个节点的移动距离 d_n^s,有两种生成方式:一种是从现有和声库中随机选择一个已有元素,另一种是在可行域内随机生成一个新元素,具体生成方式由和声记忆保留概率 p_{cr} 决定。

$$d_n^s \leftarrow \begin{cases} d_n^s \in \{d_1^s,d_2^s,\cdots,d_{H_{\mathrm{m}}}^s\}, & \text{以概率 } p_{\mathrm{cr}} \\ d_n^s \in [-d_{\max},d_{\max}], & \text{以概率 } 1-p_{\mathrm{cr}} \end{cases} \tag{3-22}$$

在得到新的和声 $\boldsymbol{D}_n=[d_n^1,d_n^2,\cdots,d_n^N]$ 后,需要根据记忆扰动概率 p_{ar} 决定每个元素是否进一步微调。对于需要微调的元素,利用微调带宽 b_{m} 和一个区间 $[0,1]$ 上的随机数 α 进行调整。

$$d_n^s=d_n^s\pm\alpha\cdot b_{\mathrm{m}} \tag{3-23}$$

3. 更新和声库

在对新和声进行微调之后,需要将其与和声库中最差的和声进行比较,通过计算目标函数 $\det(\boldsymbol{J}_{k+1})$ 的值,将其中较差的一个淘汰掉,保留较好的和声,对和声库进行更新。

4. 判断终止搜索

和声搜索算法通过迭代搜索,不断地淘汰和声库中最差的和声,加入新的和声。当满足算法的终止条件时,如搜索次数到达上限,或者目标函数值达到要求,即可终止搜索,将和声库中最好的一个和声作为优化问题的解。

利用和声搜索算法,可以高效求得优化问题(3-15)的解(即所有节点深度移动距离的方案)。尽管

和声搜索算法只能获得次优解，但是对跟踪精度的影响较小，而且可以快速获得结果。需要说明的是，求解深度调节距离是为了获得更好的量测数据估计目标状态 $\hat{\boldsymbol{x}}_{k+1|k+1}$，因此在计算式(3-20)的费舍尔信息矩阵时，需要根据上一时刻估计结果 $\hat{\boldsymbol{x}}_{k|k}$ 预测当前的目标状态 $\hat{\boldsymbol{x}}_{k+1|k}$ 并代入计算。

至此，在目标跟踪过程中，针对目标深度已知和未知两种情况，通过最大化 FIM，分别给出了基于深度调节的节点调度方案。若目标深度已知，根据式(3-19)，只需动态调整节点深度，使节点与目标间的高度差最小，即可实现最大化 FIM；若目标深度位置未知，由于三阶矩阵的计算比较复杂，为了快速计算并保证求解质量，这里利用和声搜索算法求解深度移动距离，同样可以最大化 FIM。通过这种深度调节方法，使节点移动到不同的深度，可以克服稀疏网络覆盖能力差的弊端，使节点之间构成良好的位置结构，获取更加完备的量测数据，进而融合中心可以使用 EKF 等算法对目标进行状态估计，实现目标跟踪精度的提高。

虽然基于深度调节的节点调度方案可以帮助节点获得更加完备的量测数据，但不可避免的是，节点的深度移动会导致能量消耗，而 UWSN 的能量资源较为有限。因此，需要在提高目标跟踪精度的同时考虑节点移动的能耗，实现精度与能耗的平衡。假定节点移动的能耗与移动距离为线性关系，可以通过改变移动距离来调整能耗，因此设定综合考虑目标跟踪精度和节点移动能耗的目标函数。

$$f(\boldsymbol{D}_{k+1})=\frac{\det(\boldsymbol{J}_{k+1}^{0})}{\det(\boldsymbol{J}_{k+1}(\boldsymbol{D}_{k+1}))}+\beta\frac{\|\boldsymbol{D}_{k+1}\|_1}{Nd_{\max}} \tag{3-24}$$

式(3-24)中的第一项用于衡量目标跟踪精度，表示所有节点保持静止下的费舍尔信息，它与 $\det(\boldsymbol{J}_{k+1}(\boldsymbol{D}_{k+1}))$ 之间的比值越小，意味着通过深度调节增加的信息量越多；第二项用于衡量深度调节的能耗，$\|\boldsymbol{D}_{k+1}\|_1$ 表示移动向量 $\boldsymbol{D}_{k+1}$ 的 1 范数，可计算所有节点绝对移动距离的和，$\|\boldsymbol{D}_{k+1}\|_1/N$ 则对应所有节点的平均绝对移动距离，$d_{\max}$ 表示节点可以移动的最大范数，两者之间的比值越小，意味着节点移动距离越小，整体能耗也越少。此外，也可以使用移动向量的无穷范围 $\|\boldsymbol{D}_{k+1}\|_\infty$ 计算所有节点中最大的移动距离，通过衡量最大移动距离和移动范围的比值 $\|\boldsymbol{D}_{k+1}\|_\infty/d_{\max}$ 控制移动能耗。β 是用于调整跟踪精度和能量消耗之间权重的系数。在实时目标跟踪精度较高时，需要增大 β，更多地考虑能量消耗；而在误差较大时需要主要考虑跟踪精度，应当减小 β。因此，可以设定动态变化的 β 为

$$\beta=\Psi_0/\Psi_k \tag{3-25}$$

其中，Ψ_k 为上一时刻目标估计协方差的迹，Ψ_0 则是预先设定的参考误差。当跟踪误差超出阈值时，$\beta<1$，目标函数中跟踪精度部分权重增大；反之，跟踪误差较小时，节点移动能耗的权重较大。因此，最优的节点深度调节可以通过下式得到。

$$\boldsymbol{D}_{k+1}=\arg\min f(\boldsymbol{D}_{k+1}) \tag{3-26}$$

综上所述，可以利用和声搜索算法求解优化问题(3-26)，使节点移动到合适的深度，获得更加完备的量测数据，融合中心进而利用扩展卡尔曼滤波等算法进行目标状态估计。图 3.4 给出了基于节点深度调节的目标跟踪算法流程图。首先，融合中心利用上一时刻的估计结果对目标状态进行预测，然后计算各个节点的最优调节深度并调度至该位置，具体移动距离可以根据目标状况，利用目标函数式(3-15)或式(3-26)计算得到。

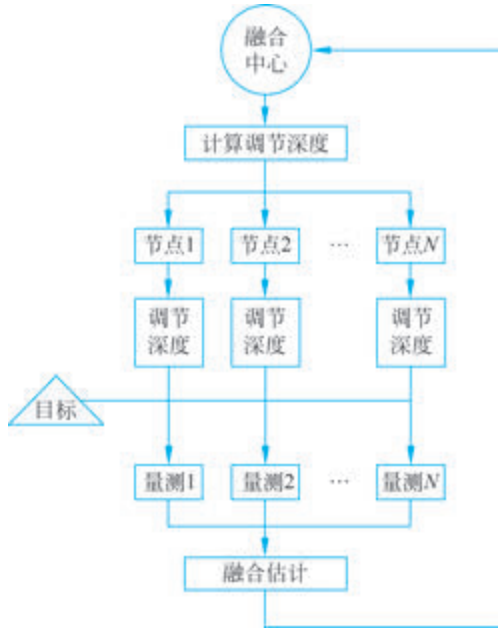


图 3.4 基于节点深度调节的目标跟踪算法流程图

节点在移动到相应位置后，获取新的目标量测数据并发送至融合中心。最后，融合中心融合所有节点的量测数据，采用扩展卡尔曼滤波算法估计目标的状态，并进入下一时刻的循环。

3.1.4 仿真结果与分析

为了验证和分析本节提出的算法，本节进行了计算机仿真实验，模拟了基于节点深度调节的UWSN目标跟踪过程，分别针对目标深度已知和目标深度未知两种情况进行了实验，对本节提出的基于节点深度调节、考虑能耗的深度调节和节点固定不动的方案进行了对比，并给出了仿真结果和性能分析。

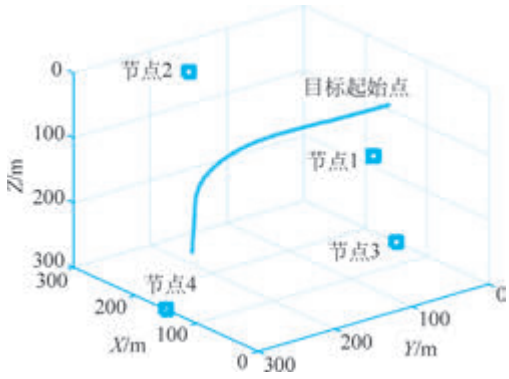


图 3.5 仿真场景

本节采用如图 3.5 所示的仿真场景，水下目标在一个三维空间中运动，UWSN 由 4 个固定节点组成，其中节点 1 和节点 2 部署在深度为 50m 处，节点 3 和节点 4 部署在深度为 300m 处，4 个节点构成四面体的结构，以保证多方位获取完备的目标量测，监测区域的整体范围大致为 300m×300m×300m，所有 4 个节点均可探测到监测区域的运动目标，且它们具有相同的探测范围和量测误差，量测误差均服从分布 $N(0,10)$，融合中心位于水面中心位置，通过获取 4 个节点的数据进行工作。目标的初始状态为 $\boldsymbol{x}_0=[100,3,50,1,50,2]^{\mathrm{T}}$，在时间区间 $[1,40]$，目标做匀速(Constant Velocity，CV)运动；在时间区间 $[41,80]$，目标做协调转弯(Coordinated Turn，CT)运动；在时间区间 $[81,100]$，目标做 CV 运动。具体的运动模型分别为

$$\boldsymbol{x}_{k+1}=\boldsymbol{F}_k^{\mathrm{CV}}\boldsymbol{x}_k+\boldsymbol{w}_k \tag{3-27}$$

$$\boldsymbol{x}_{k+1}=\boldsymbol{F}_k^{\mathrm{CT}}\boldsymbol{x}_k+\boldsymbol{w}_k \tag{3-28}$$

其中，状态转移矩阵 $\boldsymbol{F}_k^{\mathrm{CV}}$、$\boldsymbol{F}_k^{\mathrm{CT}}$ 以及过程噪声 $\boldsymbol{w}_k$ 的协方差矩阵 $\boldsymbol{Q}_k$ 分别为

$$\boldsymbol{F}_k^{\mathrm{CV}}=\begin{bmatrix}1&T&0&0&0&0\\0&1&0&0&0&0\\0&0&1&T&0&0\\0&0&0&1&0&0\\0&0&0&0&1&T\\0&0&0&0&0&1\end{bmatrix} \tag{3-29}$$

$$\boldsymbol{F}_k^{\mathrm{CT}}=\begin{bmatrix}1&\dfrac{\sin(\omega T)}{\omega}&0&-\dfrac{1-\cos(\omega T)}{\omega}&0&0\\0&\dfrac{\sin(\omega T)}{\omega}&0&-\sin(\omega T)&0&0\\0&\dfrac{1-\cos(\omega T)}{\omega}&1&\dfrac{\sin(\omega T)}{\omega}&0&0\\0&\sin(\omega T)&0&\cos(\omega T)&0&0\\0&0&0&0&1&T\\0&1&0&0&0&1\end{bmatrix} \tag{3-30}$$

$$\boldsymbol{Q}_k = q^2 \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 \\ 0 & 0 & \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \\ 0 & 0 & 0 & 0 & \frac{T^2}{2} & T \end{bmatrix} \tag{3-31}$$

其中，ω 是固定的角速度；T 是采样间隔；q^2 表示过程噪声的强度。此外，对于水下目标通常只考虑水平面的 CT 运动。

实验结果通过 100 次蒙特卡洛(MC=100)仿真平均得到，为了衡量目标跟踪的精度，通过计算均方根误差作为跟踪性能的判据：

$$\varepsilon_k = \sqrt{\frac{1}{\mathrm{MC}} \sum_{m=1}^{\mathrm{MC}} \left[(x_k^{i,1} - \hat{x}_k^{i,1})^2 + (x_k^{i,3} - \hat{x}_k^{i,3})^2 + (x_k^{i,5} - \hat{x}_k^{i,5})^2 \right]} \tag{3-32}$$

其中，$(x_k^{i,1}, x_k^{i,3}, x_k^{i,5})$和$(\hat{x}_k^{i,1}, \hat{x}_k^{i,3}, \hat{x}_k^{i,5})$分别表示目标在 k 时刻的真实位置和估计位置。

首先，针对目标深度已知的情况进行实验，为了评估算法的整体效果，图 3.6 给出了基于节点深度调节、考虑能耗的深度调节和节点固定不动 3 种方案下的目标跟踪结果。从图 3.6 可以看出，通过节点深度的自组织调节，节点之间的位置结构随目标运动不断改变，系统可实时获得更加完备的量测信息，从而有效提高目标跟踪精度；对于节点固定不动的方案，由于在三维环境中 4 个节点构成四面体结构，该方案也可以获得稳定的跟踪结果，但是与节点深度调节的方案相比，固定的节点位置结构缺乏对于动态目标的调整能力，因此跟踪精度最差；而考虑能耗的节点深度调节方案则介于两者之间，一方面通过节点深度的调节获取更完备的量测信息，同时考虑到节点深度移动会导致能量消耗，在精度和能耗之间取得了平衡，可以根据任务需求更灵活地调整。图 3.7 给出了两种节点深度调节方案下，4 个节点的平均移动距离，可以看出，考虑能耗的深度调节方案中节点的平均移动距离要小很多，而且整体过程比较

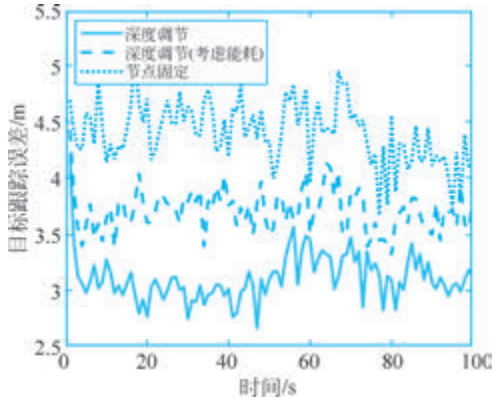


图 3.6 目标深度已知：3 种不同方案的目标跟踪误差对比

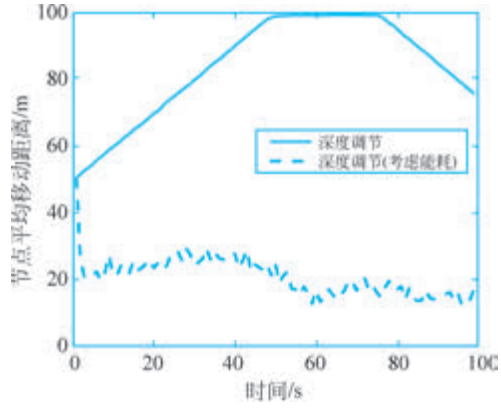


图 3.7 目标深度已知：两种深度调节方案的节点平均移动距离对比

稳定；而在只考虑精度的方案中，节点的平均移动距离呈线性变化，这是因为式(3-19)中给出的调节方案使各节点向与目标高度差减小的方向移动最大距离，且目标在深度方向做 CV 运动，使得节点移动距离呈线性变化。表 3.1 给出了 3 种方案的平均跟踪误差和节点平均移动距离。通过节点深度调节，跟踪精度可提高近 30%，但代价是节点的移动消耗；而考虑能耗的调节方案虽然只提高了 16%的跟踪精度，但是却减小了 75%的移动消耗，提供了性价比更高的选择。

表 3.1　目标深度已知：3 种不同方案的平均跟踪误差和节点平均移动距离对比

方　　案	平均跟踪误差/m	节点平均移动距离/m
深度调节	3.109	84.06
深度调节(考虑能耗)	3.695	20.82
节点固定	4.404	0

图 3.8 和图 3.9 分别给出了两种深度调节方案中，4 个节点各自深度调节距离的变化，其中正值表示向下移动，深度增加；负值表示向上移动，深度减小。在图 3.8 中，由于前期目标的深度在节点 1 和节点 2 的可移动范围内，因此这两个节点只需向下移动到和目标同一深度，而节点 3 和节点 4 由于无法移动至与目标等深位置，因此只能向上移动最大距离，尽量减小高度差；而跟踪后期随着目标上升，逐渐靠近节点 3 和节点 4，两者的移动距离不断减小，而节点 1 和节点 2 则需要向下移动最大距离。在图 3.9 中，由于考虑了节点移动的能耗，因此 4 个节点的移动距离不再呈线性变化，而且与不考虑能耗的深度调节相比，这种方案的节点移动距离要小多。前期由于目标与节点 1 和节点 3 在水平方向的位置较近，因此主要通过调节这两个节点的深度提高量测质量，节点 2 和节点 4 移动较小以节约能耗；随着目标运动，与节点 2 之间的距离变小，节点 2 也逐渐增加移动距离，当两者之间方位合适时，便再次减小移动以节约能耗；直到目标转弯逐渐远离节点 2 后，再次增加移动距离并保持在比较稳定的位置，同时节点 4 由于目标靠近而逐渐减小移动距离。总体而言，由于需要考虑跟踪精度和移动能耗两个因素，节点的深度调节变化与图 3.8 相比更加复杂，但通过 4 个节点的相互协作，仍然可以获取较好的量测信息，提高目标跟踪的精度。

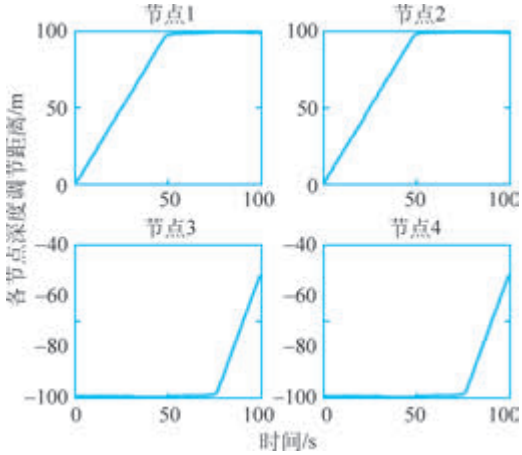


图 3.8　目标深度已知时 4 个节点的深度调节距离：不考虑能耗的深度调节方案

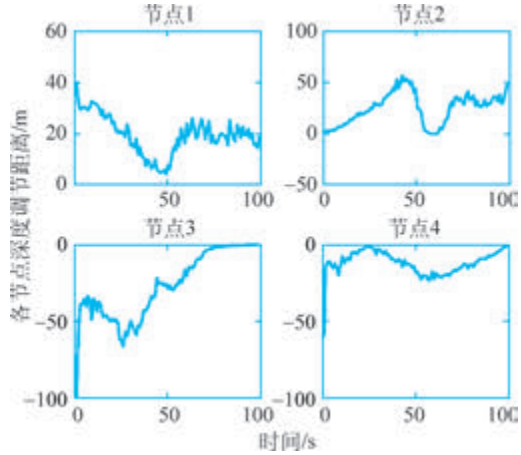


图 3.9　目标深度已知时 4 个节点的深度调节距离：考虑能耗的深度调节方案

随后，针对目标深度未知的情况，进行了 3 种方案的对比分析。图 3.10 给出了 3 种方案下的目标跟踪结果，可以看出，对于目标深度未知的情况，节点深度调节仍然可以达到提高目标跟踪精度的效果。与目标深度已知的结果类似，考虑能耗的深度调节实现了介于另外两种方案之间的跟踪精度。图 3.11

展现了两种深度调节方案中，所有节点的平均移动距离，显然，考虑能耗的情况下可以有效地减小移动距离，而且各个时刻的移动距离也比较稳定；相对而言，只考虑跟踪精度的方案专注于不断地调整节点位置结构，获取更完备的目标信息，因此节点移动距离也相对较多。此外，由于衡量精度的式(3-20)计算复杂，因此节点移动距离的变化也相对复杂，无法像目标深度已知情况一样呈线性变化。表 3.2 给出了 3 种方案的平均跟踪误差和节点平均移动距离，两种节点深度调节方案分别提高了 25%和 15%左右的跟踪精度，而考虑能耗的调节方案同时减小了三分之二的移动距离。对于跟踪精度需求较低的任务，可选择使用考虑能耗的深度调节方案，甚至减小深度调节频率，仅在跟踪误差过大时进行调节，并通过预测后续一段时间目标的整体状态调整节点位置结构，以保障后续区间的信息获取。

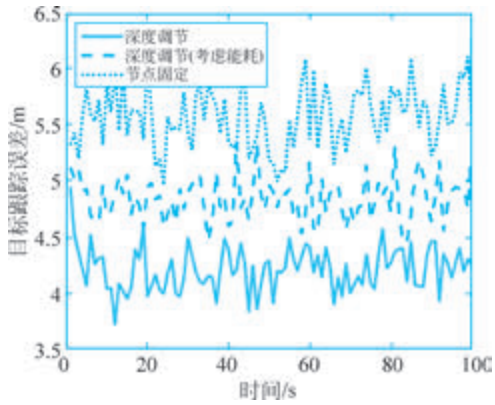


图 3.10 目标深度未知：3 种不同方案的目标跟踪误差对比

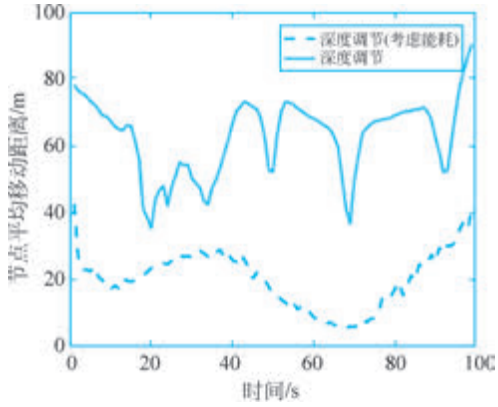


图 3.11 目标深度未知：两种深度调节方案的节点平均移动距离对比

表 3.2 目标深度未知：3 种不同方案的平均跟踪误差和节点平均移动距离对比

方　案	平均跟踪误差/m	节点平均移动距离/m
深度调节	4.217	62.81
深度调节(考虑能耗)	4.744	20.37
节点固定	5.574	0

图 3.12 和图 3.13 分别给出了两种深度调节方案中，4 个节点各自深度调节距离的变化。相比图 3.8，由于目标深度未知，图 3.12 中的节点移动距离不再线性变化，4 个节点有着不同的深度调节过程，这样可以更灵活地调整节点位置结构；图 3.13 中各节点的移动距离要比图 3.12 中小很多，但是两幅图中各节点的调节距离呈现了相似的变化趋势，这是由于能耗因素减小了图 3.13 中各节点的移动距离。此外，各节点的调节距离变化整体平稳，深度变化过快的情况不算很多，这有利于实际跟踪任务中深度调节方案的实施。

在考虑能耗的深度调节中，为了自适应地平衡跟踪精度和移动能耗，式(3-25)设置动态变化的权重系数 β，在跟踪误差较大时减小 β，主要考虑跟踪精度，在跟踪误差较小时增大 β，以增加移动能耗的影响。为了验证自适应调整的有效性，对不同参考误差 Ψ_0 下的跟踪结果进行了比较，Ψ_0 在区间 [10,100] 上间隔 10 取值，图 3.14 和图 3.15 分别给出了目标深度已知和未知两种情况，不同 Ψ_0 下基于考虑能耗的深度调节方案的平均跟踪误差和节点平均移动距离。由图 3.14 可以看出，对目标深度已知和未知两种情况，Ψ_0 从 10 增加到 100，平均误差虽然一直增加，但变化幅度不大；而图 3.15 中两种情况的平均移动距离，则减小了 70%左右，尤其是当 Ψ_0 为 10～40 时，移动距离下降非常显著，这说明通过引入权重系数 β，设置不同的参考误差 Ψ_0，可以有效地降低移动距离，并保持相对稳定的跟踪精度。此外，在实际应用中，还可以利用参考误差 Ψ_0 动态地改变深度调节的频率，当跟踪误差较小时，保

持固定的节点位置结构以降低能耗；而当跟踪误差超出误差阈值后，通过深度调节以提高目标跟踪精度。

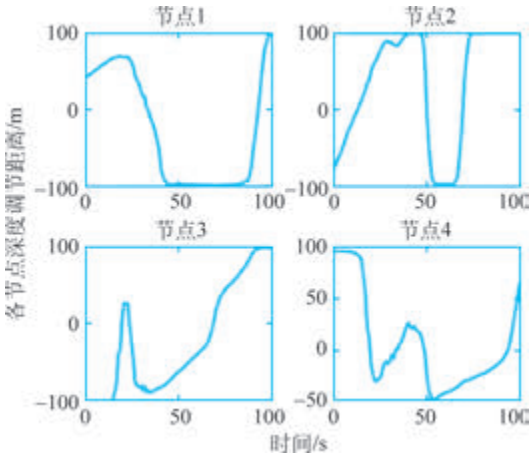


图 3.12 深度未知时 4 个节点的深度调节距离：不考虑能耗的深度调节方案

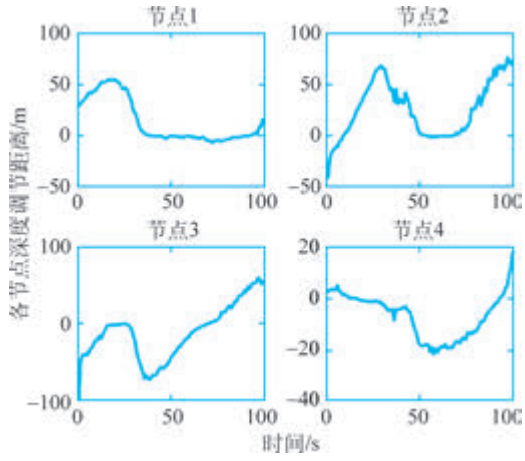


图 3.13 目标深度未知时 4 个节点的深度调节距离：考虑能耗的深度调节方案

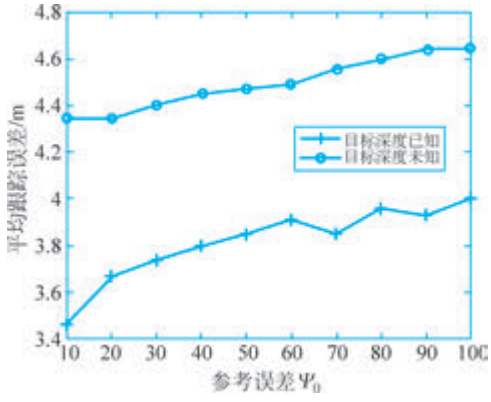


图 3.14 不同 Ψ_0 时的平均跟踪误差：目标深度已知和目标深度未知

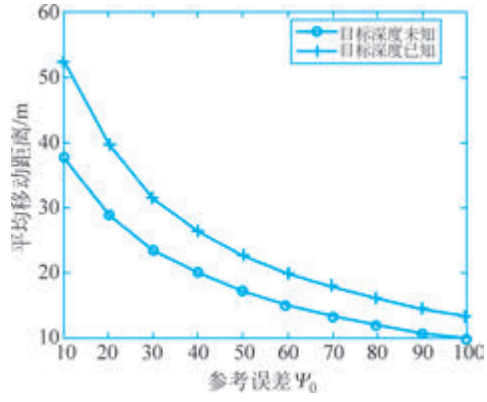


图 3.15 不同 Ψ_0 时的平均移动距离：目标深度已知和目标深度未知

3.2 基于能量分配的节点调度方法

目标跟踪是对目标状态进行连续估计的过程，节点处理、传输数据等工作都会消耗节点的电池能量，考虑到水下复杂环境带来的能量消耗，以及 UWSN 节点的供电电池难以更换等问题，通常需要采用压缩通信数据量或者减少数据传输次数等调度方式来降低节点能耗，然而，这些调度方式都会造成量测信息的损失，影响目标跟踪的精度。此外，搭载于 UWSN 节点的能量收集装置可以通过压电设备将水流运动能量转化为电能，为节点长期工作提供能量储备[92]。因此，需要综合节点能量变化和目标跟踪精度，合理分配节点用于数据处理的能量。考虑到节点能量是随时间不断变化的状态，每个时刻节点的能量消耗都会对后续时刻的能量状态产生影响；而各个节点与目标之间的距离、方位也随时间不断改变，即各个时刻节点的量测价值必定不同；因此，可以根据节点量测价值的变化，分配不同的能量用于数据处理。如果在压缩量测数据位数时不考虑量测价值的变化因素，而是在各个时刻分配相等的能量

处理数据，势必会导致较低的能量利用效率。更好的模式是当量测价值高时，节点可以消耗更多能量用于处理数据；而当量测价值低时，尽量减小节点的能量消耗，尽可能地提高有限能量资源的利用效率。因此，结合各个节点能量状态在时间维度上的充电和消耗过程，如何设计合理的基于能量分配的节点调度方法，在有限的资源条件下提高能量的利用效率和目标跟踪的精度，实现长期高效的目标跟踪是本节需要解决的主要问题。

针对这一问题，本节提出了基于长时能量分配的水下目标跟踪方法。通过对一段时间区间内（长时）各时刻可消耗能量的动态分配，实现全局目标跟踪性能的提高。由于节点个体能量状态在时间维度上前后关联的特点，本节结合水下节点能量收集过程，建立了节点能量状态的变化模型，并利用 FIM 对各节点在不同时刻的量测数据进行价值预测，给出了衡量有限时间区间内全局跟踪精度的目标函数。在节点能量变化模型的约束下将问题转化为马尔可夫决策过程（Markov Decision Process，MDP），实现了各个时刻节点可消耗能量的自组织分配，并依据能量预算对量测数据进行量化、传输处理，在有限的资源条件下实现了全局量测信息的最大化，以及能量利用效率和目标跟踪精度的提高。

3.2.1 跟踪模型

水下运动目标的状态模型通常描述为

$$\boldsymbol{x}_{k+1}=\boldsymbol{F}_k\boldsymbol{x}_k+\boldsymbol{w}_k \tag{3-33}$$

其中，$\boldsymbol{x}_k=[x_k,\dot{x}_k,y_k,\dot{y}_k,z_k,\dot{z}_k]^{\mathrm{T}}\in\mathbb{R}^6$ 表示目标在 k 时刻的运动状态；(x_k,y_k,z_k) 表示 k 时刻目标在空间坐标系中的位置；$(\dot{x}_k,\dot{y}_k,\dot{z}_k)$ 则表示目标在相应方向的速度；$\boldsymbol{F}_k\in\mathbb{R}^{6\times6}$ 表示运动目标在 k 时刻的状态转移矩阵；$\boldsymbol{w}_k$ 表示服从正态分布 $N(\boldsymbol{0}_6,\boldsymbol{Q}_k)$ 的过程噪声。

搭载了水听器的节点 s 在 $k+1$ 时刻的量测模型可以表示为

$$z_{k+1}^s=h_{k+1}^s(\boldsymbol{x}_{k+1})+v_{k+1}^s \tag{3-34}$$

其中，$h_{k+1}^s:\mathbb{R}^6\mapsto\mathbb{R}$ 表示节点 s 的量测方程；v_{k+1}^s 表示服从正态分布 $N(0,R_{k+1}^s)$ 的量测噪声。假定各节点量测噪声独立同分布且不随时间变化，并设节点获取量测为关于目标的纯距离量测，此时

$$h_{k+1}^s(\boldsymbol{x}_{k+1})=\sqrt{(x_{k+1}-x^s)^2+(y_{k+1}-y^s)^2+(z_{k+1}-z^s)^2} \tag{3-35}$$

其中，(x^s,y^s,z^s) 表示节点 s 的位置。

各个探测节点在跟踪过程中获取各采样时刻的目标距离量测，并将数据发送至 UWSN 的融合中心进行融合估计，以获得更为准确的目标状态估计。考虑到 UWSN 节点之间的通信带宽十分有限，以及各节点的能量有限，通常采用对量测数据进行量化处理的方式，以减小 UWSN 的通信数据量和能量消耗[93]。这种量化处理的方式不仅可以减小各个节点传输数据所消耗的能量，还可以减小融合中心同时接收的数据量，降低网络的带宽负担。各个节点通过对传感器获得的原始数据 z_{k+1}^s 进行量化处理，可以将其压缩为只有几个比特位的量化数据 $z_{k+1,q}^s$，并进一步将其发送至融合中心进行融合估计

$$z_{k+1,q}^s=\begin{cases}0, & \gamma_0<z_{k+1}^s<\gamma_1\\ 1, & \gamma_1<z_{k+1}^s<\gamma_2\\ \vdots & \vdots\\ L-1, & \gamma_{L-1}<z_{k+1}^s<\gamma_L\end{cases} \tag{3-36}$$

其中，原始量测依照预设的量化阈值集合 $\gamma=\{\gamma_0,\gamma_1,\cdots,\gamma_L\}$ 被压缩为 $L=2^m$ 个不同的量化数值，m 表

示量化数据的比特位数，本书假定 UWSN 各个节点共用同一个确定的量化阈值集合。由于量化处理的方式会损失量测数据信息，为了尽可能地保证目标跟踪精度，很多学者针对量化阈值的计算展开了研究，具体方法本书不再介绍，有兴趣的读者可参考文献[93]。因此，联合式(3-34)、式(3-35)及式(3-36)，在给定目标状态 $\boldsymbol{x}_{k+1}$ 的条件下，量化量测数值等于 l 的概率为

$$p(z_{k+1,q}^{s}=l \mid \boldsymbol{x}_{k+1})=Q\left(\frac{\gamma_l-h_{k+1}^{s}}{\sqrt{R_{k+1}^{s}}}\right)-Q\left(\frac{\gamma_{l+1}-h_{k+1}^{s}}{\sqrt{R_{k+1}^{s}}}\right) \tag{3-37}$$

其中，

$$Q(x)=\int_{x}^{+\infty}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{t^2}{2}\right)\mathrm{d}t \tag{3-38}$$

假设每个时刻有 N 个节点参与目标跟踪任务，节点将各自的量测数据量化处理后发送至融合中心。融合中心在获取量化量测 $\boldsymbol{Z}_{k+1,q}=[z_{k+1,q}^{1},z_{k+1,q}^{2},\cdots,z_{k+1,q}^{N}]^{\mathrm{T}}$ 后，可以利用滤波算法对目标状态进行融合估计。目标状态对应的多传感器联合量测似然函数为

$$p(\boldsymbol{Z}_{k+1,q} \mid \boldsymbol{x}_{k+1})=\prod_{s=1}^{N}p(z_{k+1,q}^{s} \mid \boldsymbol{x}_{k+1}) \tag{3-39}$$

其中，节点 s 的量测似然 $p(z_{k+1,q}^{s}|\boldsymbol{x}_{k+1})$ 可以通过式(3-37)计算获得。

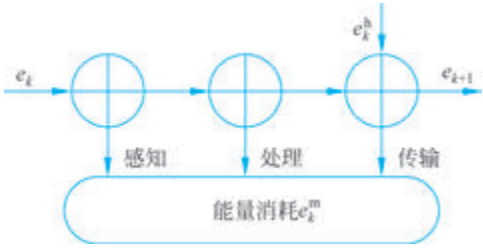


图 3.16　节点能量状态变化

每一个节点通过自身搭载的能量收集装置，可以将水流运动能量转化成电能为节点电池持续充电，文献[92]中的实验显示能量收集装置的最大功率密度可达 $350\mu\mathrm{W/cm}^3$。当节点处于休眠状态时，能量收集装置持续为电池充电至满电状态；当节点处于工作状态时，电池为节点供能的同时可持续充电，用于后续时刻工作。节点能量的收集和消耗过程如图 3.16 所示，e_k 表示电池在 k 时刻的能量状态，即该时刻节点可以利用的最大能量，经过感知、处理和传输等工作行为，节点消耗电能 e_k^{m}，通过水流转化吸收电能 e_k^{h}，进而得到下一时刻可供给节点利用的能量。因此，节点能量状态的变化模型如下：

$$e_{k+1}=\min\{e_k+e_k^{\mathrm{h}}-e_k^{\mathrm{m}},e_{\max}\} \tag{3-40}$$

其中，根据实验数据可以将能量收集值 e_k^{h} 近似构建为一类静态马尔可夫过程[94]，一个采样时刻内收集的能量可以分为 H 个数值，表示为 $e_k^{\mathrm{h}}\in\{\alpha_1,\alpha_2,\cdots,\alpha_H\}$；$e_{\max}$ 表示电池的最大容量，当可利用能量达到最大容量后，能量收集装置便处于休眠状态。根据预备知识中介绍的节点能耗模型，在通信距离和功率固定的条件下，节点的能量消耗主要由传输数据量的大小决定[95]。此外，深度调节也是节点能量消耗的主要因素，考虑到实际应用中可以每隔一段时间进行一次深度调节，而每个时刻都需要传输数据，因此本节主要考虑数据压缩处理对节点能量的影响。为了降低节点能量消耗，需要减少数据量的大小，而数据量减小则意味着量测信息的损失，会造成目标跟踪误差增大。建立节点能耗 e_k^{m} 与传输量测位数 m_k 的线性关系为

$$e_k^{\mathrm{m}}=\lambda m_k \tag{3-41}$$

其中，λ 表示传输一个单位数据时节点所消耗的能量。

3.2.2　能量分配对跟踪性能的影响

在目标跟踪过程中，各个节点在不同时刻获取的量测数据价值不同。考虑到水下节点能量资源

有限，应将尽可能多的能量用于处理高价值量测，充分保留其数据信息；而低价值量测可压缩为较少位后发送至融合中心，以减小节点能量消耗。由于节点能量状态在时间维度上前后关联，每一时刻的量测处理都会对后续时刻的能量状态产生影响，常见的短时能量分配仅考虑下一时刻的跟踪精度和任务需求，可能过早透支能量；而长时能量分配综合考虑一段时间区间内的整体跟踪需求，可以更合理地分配能量，将更多的资源用于处理高价值的量测数据，进而提高能量资源的利用效率和目标跟踪精度。因此，针对 UWSN 节点能量资源有限的问题，有必要研究长时能量分配对全局目标跟踪性能的影响。

FIM 是目标跟踪问题中用于衡量跟踪精度，实现网络自组织调度的经典判据。在一个时间区间$[1,K]$上，可以利用累加 FIM 作为全局跟踪精度的判据

$$\boldsymbol{J}^K=\sum_{k=1}^{K}q_k\boldsymbol{J}_k \tag{3-42}$$

其中，$\boldsymbol{J}_k$ 表示 k 时刻的费舍尔信息矩阵；$\boldsymbol{J}^K$ 表示时间区间$[1,K]$上的累加和；q_k 则是相应时刻的权重系数。

根据 FIM 的定义式，用 $\Delta_{\boldsymbol{x}}^{\boldsymbol{x}}\triangleq\nabla_{\boldsymbol{x}}\nabla_{\boldsymbol{x}}^{\mathrm{T}}$ 表示 $\boldsymbol{x}$ 的二阶微分算子，k 时刻的费舍尔信息矩阵可以改写为

$$\boldsymbol{J}_k=E_{p(\boldsymbol{Z}_{k,q},\boldsymbol{x}_k)}\left[-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k}\log p(\boldsymbol{Z}_{k,q},\boldsymbol{x}_k)\right] \tag{3-43}$$

由于 $p(\boldsymbol{Z}_{k,q},\boldsymbol{x}_k)=p(\boldsymbol{Z}_{k,q}\mid\boldsymbol{x}_k)p(\boldsymbol{x}_k)$，因此 $\boldsymbol{J}_k$ 可以被进一步拆分为两项

$$\begin{aligned}\boldsymbol{J}_k&=E_{p(\boldsymbol{x}_k)}\left[-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k}\log p(\boldsymbol{x}_k)\right]+E_{p(\boldsymbol{Z}_{k,q}\mid\boldsymbol{x}_k)p(\boldsymbol{x}_k)}\left[-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k}\log p(\boldsymbol{Z}_{k,q}\mid\boldsymbol{x}_k)\right]\\&=\boldsymbol{J}_k^{\mathrm{p}}+\boldsymbol{J}_k^{\mathrm{d}}\end{aligned} \tag{3-44}$$

其中，$\boldsymbol{J}_k^{\mathrm{p}}=E_{p(\boldsymbol{x}_k)}\left[-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k}\log p(\boldsymbol{x}_k)\right]$表示目标状态先验信息对费舍尔信息的贡献，这部分与量测数据没有关系；$\boldsymbol{J}_k^{\mathrm{d}}=E_{p(\boldsymbol{Z}_{k,q}\mid\boldsymbol{x}_k)p(\boldsymbol{x}_k)}\left[-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k}\log p(\boldsymbol{Z}_{k,q}\mid\boldsymbol{x}_k)\right]$表示量测数据 $\boldsymbol{Z}_{k,q}$ 对费舍尔信息的影响，当 N 个传感器节点协同进行跟踪时，将式(3-39)代入 $\boldsymbol{J}_k^{\mathrm{d}}$，可将其改写为

$$\begin{aligned}\boldsymbol{J}_k^{\mathrm{d}}&=\int_{\mathbb{R}^6}E_{p(\boldsymbol{Z}_{k,q}\mid\boldsymbol{x}_k)}\left[-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k}\log\prod_{s=1}^{N}p(z_{k+1,q}^{s}\mid\boldsymbol{x}_{k+1})\right]p(\boldsymbol{x}_k)\mathrm{d}\boldsymbol{x}_k\\&=\sum_{s=1}^{N}\int_{\mathbb{R}^6}\left[-\sum_{l=0}^{L-1}\Delta_{\boldsymbol{x}}^{\boldsymbol{x}}\left[\log p(z_{k+1,q}^{s}=l\mid\boldsymbol{x}_{k+1})\right]\cdot p(z_{k+1,q}^{s}=l\mid\boldsymbol{x}_{k+1})\right]p(\boldsymbol{x}_k)\mathrm{d}\boldsymbol{x}_k\end{aligned} \tag{3-45}$$

显然，$\boldsymbol{J}_k^{\mathrm{d}}$ 可以拆分为各个节点提供的 FIM 之和。因此对于单个节点 s，其对 FIM 的贡献为

$$\boldsymbol{J}_{s,k}^{\mathrm{d}}=\int_{\mathbb{R}^6}\boldsymbol{J}_{s,k}^{\mathrm{t}}(\boldsymbol{x}_k)p(\boldsymbol{x}_k)\mathrm{d}\boldsymbol{x}_k \tag{3-46}$$

其中，$\boldsymbol{J}_{s,k}^{\mathrm{t}}(\boldsymbol{x}_k)$为给定状态 $\boldsymbol{x}_k$ 时节点 s 的 FIM，且

$$\boldsymbol{J}_{s,k}^{\mathrm{t}}(\boldsymbol{x}_k)=-\sum_{l=0}^{L-1}\Delta_{\boldsymbol{x}}^{\boldsymbol{x}}\left[\log p(z_{k+1,q}^{s}=l\mid\boldsymbol{x}_{k+1})\right]\cdot p(z_{k+1,q}^{s}=l\mid\boldsymbol{x}_{k+1}) \tag{3-47}$$

根据式(3-40)，节点各时刻分配的能量决定了量测数据的量化位数和量化阈值集合，进而对数据的信息量和目标跟踪精度产生影响。因此，可以利用费舍尔信息与能量分配之间的关系表征这种影响。对于节点 s，当 k 时刻分配能量为 e_k^{m} 时，将量测似然函数(3-37)代入式(3-47)可推导得到 FIM[89]：

$$\boldsymbol{J}_{s,k}^{\mathrm{t}}(\boldsymbol{x}_k,e_k^{\mathrm{m}})=\frac{1}{2\pi R_k^s h_k^s(\boldsymbol{x}_k)^2}\varphi_k^s(\boldsymbol{x}_k,e_k^{\mathrm{m}})\times$$

$$\begin{bmatrix} (x_k-x^s)^2 & 0 & (x_k-x^s)(y_k-y^s) & 0 & (x_k-x^s)(z_k-z^s) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ (y_k-y^s)(x_k-x^s) & 0 & (y_k-y^s)^2 & 0 & (y_k-y^s)(z_k-z^s) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ (z_k-z^s)(x_k-x^s) & 0 & (z_k-z^s)(y_k-y^s) & 0 & (z_k-z^s)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3-48}$$

其中，

$$\varphi_k^s(\boldsymbol{x}_k, e_k^{\mathrm{m}}) = \sum_{l=0}^{L(e_k^{\mathrm{m}})-1} \frac{\left[\exp\left(-\frac{(\gamma_l(e_k^{\mathrm{m}})-h_k^s(\boldsymbol{x}_k))^2}{2R_k^s}\right)-\exp\left(-\frac{(\gamma_{l+1}(e_k^{\mathrm{m}})-h_k^s(\boldsymbol{x}_k))^2}{2R_k^s}\right)\right]^2}{p(z_{k+1,q}^s = l \mid \boldsymbol{x}_{k+1})} \tag{3-49}$$

其中，$L(e_k^{\mathrm{m}})=2^{e_k^{\mathrm{m}}/\lambda}$ 为实际分配能量 e_k^{m} 决定的量化量测等级；$\gamma_l(e_k^{\mathrm{m}})$ 和 $\gamma_{l+1}(e_k^{\mathrm{m}})$ 分别为量化量测 l 对应的量化阈值，同样根据分配能量查表决定。

综上，可以利用式(3-46)～式(3-49)计算 $\boldsymbol{J}_{s,k}^{\mathrm{d}}$。然而，由于难以获得式(3-46)的解析解，这里需要采用 MC 采样的方法近似计算，根据目标状态的先验分布进行采样，并通过目标状态的采样粒子计算相应的数值解

$$\boldsymbol{J}_{s,k}^{\mathrm{d}} \approx \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{J}_{s,k}^{\mathrm{t}}(\boldsymbol{x}_k^i) \tag{3-50}$$

其中，$\boldsymbol{x}_k^i$ 表示目标状态 $\boldsymbol{x}_k$ 的第 i 个粒子；M 表示采样粒子的个数；$\boldsymbol{J}_{s,k}^{\mathrm{t}}(\boldsymbol{x}_k^i)$ 则表示通过粒子 $\boldsymbol{x}_k^i$ 计算得到的节点 s 的 FIM。

通过式(3-50)计算能量消耗 e_k^{m} 与 $\boldsymbol{J}_{s,k}^{\mathrm{d}}$ 之间的定量关系，可以用于设计能量分配的方案。为了在跟踪过程区间 $[1,K]$ 上合理地分配能量，提高全局目标跟踪精度，可以将各时刻的 $\boldsymbol{J}_{s,k}^{\mathrm{d}}$ 代入式(3-42)建立目标函数。考虑到计算 $\boldsymbol{J}^K$ 的行列式比较复杂，且无法单独衡量每个时刻的 FIM，可以选择矩阵的迹作为目标函数值。文献[96]中证明了量化量测下最大化 FIM 的迹，可以实现和最大化行列式相当的结果，而且累加矩阵的迹可以拆分为各个矩阵迹的和，便于单独衡量各个时刻量测的价值。因此，本节通过最大化 FIM 的迹建立目标函数

$$\max_{\boldsymbol{e}^{\mathrm{m}}} \mathrm{tr}\left(\sum_{k=1}^{K} q_k \sum_{s=1}^{N}\boldsymbol{J}_{s,k}^{\mathrm{d}}(\boldsymbol{x}_k, e_k^{\mathrm{m}})\right) \tag{3-51}$$

其中，$\boldsymbol{e}^{\mathrm{m}}=[e_1^{\mathrm{m}}, e_2^{\mathrm{m}}, \cdots, e_K^{\mathrm{m}}]$ 表示节点各时刻的能量分配。

在能量分配过程中，需要综合考虑各个时刻的量测价值和全局跟踪精度，因此无法通过各时刻目标状态的一步预测来采样粒子，需要根据初始时刻目标的先验信息进行长时的预测采样

$$\boldsymbol{J}_{s,k}^{\mathrm{d}}(\boldsymbol{x}_k, e_k^{\mathrm{m}}) = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{J}_{s,k}^{\mathrm{t}}(\boldsymbol{x}_{k|0}^i, e_k^{\mathrm{m}}), \quad k=1,2,\cdots,K \tag{3-52}$$

其中，$\boldsymbol{x}_{k|0}^i$ 是利用先验信息获得的 k 时刻目标状态的预测粒子，可以通过目标的状态模型(3-33)迭代计算

$$\boldsymbol{x}_{k+1|0}^i = \boldsymbol{F}_k\boldsymbol{x}_{k|0}^i \tag{3-53}$$

通过式(3-53)获得目标状态的长时预测粒子后，基于式(3-51)给出的全局目标跟踪精度的优化函数，可以在给定能量分配方案 $\boldsymbol{e}^{\mathrm{m}}$ 下，对各时刻的 FIM 和全局目标跟踪精度进行预测。为了便于表述，算法 3.1 给出了计算各时刻 FIM 的伪代码。

算法 3.1 基于长时预测计算费舍尔信息矩阵

初始化	给定初始分布 $p(\boldsymbol{x}_0)$，根据 $p(\boldsymbol{x}_0)$ 采样得到初始粒子 $\{\boldsymbol{x}_0^i\}_{i=1}^M$
输入	$\{\boldsymbol{x}_{k-1\mid 0}^i\}_{i=1}^M$ 对于时刻 $k=1,2,3,\cdots,K$，循环进行以下步骤
步骤 1.	利用下式生成长时预测粒子 $\{\boldsymbol{x}_{k\mid 0}^i\}_{i=1}^M$ $\boldsymbol{x}_{k\mid 0}^i=\boldsymbol{F}_k\boldsymbol{x}_{k-1\mid 0}^i$
步骤 2.	利用长时预测粒子 $\{\boldsymbol{x}_{k\mid 0}^i\}_{i=1}^M$ 和给定的能量分配值 e_k^{m}，通过下式计算 k 时刻节点 s 的费舍尔信息矩阵 $\boldsymbol{J}_{s,k}^{\mathrm{d}}(\boldsymbol{x}_k,e_k^{\mathrm{m}})=\sum_{i=1}^{M}\boldsymbol{J}_{s,k}^{\mathrm{t}}(\boldsymbol{x}_{k\mid 0}^i,e_k^{\mathrm{m}})/M$
步骤 3.	利用费舍尔信息矩阵 $\{\boldsymbol{J}_{s,k}^{\mathrm{d}}(\boldsymbol{x}_k,e_k^{\mathrm{m}})\}_{s=1}^N$ 和给定的权重 q_k，通过下式计算 k 时刻的费舍尔信息矩阵 $\boldsymbol{J}_k^{\mathrm{d}}=\sum_{s=1}^{N}\boldsymbol{J}_{s,k}^{\mathrm{d}}(\boldsymbol{x}_k,e_k^{\mathrm{m}})$
输出	$\boldsymbol{J}_k^{\mathrm{d}}$

由于每个时刻能量消耗和收集的动态过程都会对后续时刻的能量状态产生影响，因此，在求解各个时刻能量分配 $\boldsymbol{e}^{\mathrm{m}}$ 的最优方案时，需要考虑能量状态在时间维度上前后关联的约束关系。根据式(3-40)的能量状态变化模型，节点各个时刻的能量消耗会对后续的能量状态产生影响，如果在某个时刻节点消耗了比较多的能量，则后续时刻节点可利用的能量就更加有限，可能导致跟踪误差的发散。因此，在对能量资源进行分配时，需要综合考虑时间区间内的能量约束

$$e_1^{\mathrm{m}}\leqslant e_1,e_2^{\mathrm{m}}\leqslant e_2,\cdots,e_K^{\mathrm{m}}\leqslant e_K \tag{3-54}$$

其中，$\{e_1,e_2,\cdots,e_K\}$ 表示各时刻节点可利用的最大能量，它们之间的关系可以利用式(3-40)迭代计算。根据能耗与量化量测位数的线性关系式(3-41)，能量分配的约束关系可以改写为

$$\begin{cases} e_1^{\mathrm{m}}\leqslant e_1 \\ e_2^{\mathrm{m}}\leqslant e_2=\min\{e_{\max},e_1+e_1^{\mathrm{h}}-e_1^{\mathrm{m}}\} \\ \vdots \\ e_K^{\mathrm{m}}\leqslant e_K=\min\{e_{\max},e_{K-1}+e_{K-1}^{\mathrm{h}}-e_{K-1}^{\mathrm{m}}\} \\ e_k^{\mathrm{m}}\in\{\lambda,2\lambda,\cdots,\bar{B}\lambda\},\quad k=1,2,\cdots,K \end{cases} \tag{3-55}$$

由于量化数据的位数是整数，将各时刻分配的能耗值设定为单位数据能耗 λ 的整数倍。此外，考虑到网络通信带宽的压力，设置 $\bar{B}$ 表示节点每次可传输数据的最大位数。至此，为了提高时间区间$[1,K]$上的全局目标跟踪精度，提高能量资源的利用效率，可以通过如下的优化问题计算长时能量分配方案：

$$\max_{\boldsymbol{e}^{\mathrm{m}}}\ \mathrm{tr}\Big(\sum_{k=1}^{K}q_k\sum_{s=1}^{N}\boldsymbol{J}_{s,k}^{\mathrm{d}}(\boldsymbol{x}_k,e_k^{\mathrm{m}})\Big) \tag{3-56}$$

在跟踪任务初始阶段，通过求解优化问题(3-56)获得长时能量分配方案，节点可合理分配各个时刻用于量测处理、传输的能量。通过最大化一段时间内的累加费舍尔信息，可将更多能量用于处理高价值量测，提高能量利用效率和目标跟踪精度。3.2.3 节将给出优化问题(3-56)的求解思路和跟踪方案。

3.2.3 基于长时能量分配的水下目标跟踪算法

由于节点电池的能量状态在时间维度上前后关联，各时刻的能量约束 e_k 也受到能耗 e_k^{m} 和能量收集 e_k^{h} 的影响，因此直接求解各时刻能量分配的最优解比较困难，考虑到能量收集过程是一类马尔可夫过程，该问题可以通过建立马尔可夫决策过程(Markov Decision Process，MDP)，利用动态规划(Dynamic Programming，DP)的方法求解[97,98]。

MDP 是基于马尔可夫理论的最优决策过程，它由一个五元组构成(C,A,P_a,R_a,β)。其中，C 是过程状态 c 的有限集合，A 表示各个状态下的动作集合，$P_a(c,c'):=p(c_{k+1}=c'|c_k=c,a_k=a)$表示 k 时刻，经过动作 a 使得状态 c 转移到 c'的概率；而 $R_a(c)$表示了这一动作带来的即时奖励；β 则是用于衡量即时奖励与未来奖励权重的衰减因子。MDP 的目标是通过一系列的最优动作决策 a_k，使得整个过程中的累积奖励最大化。典型的累积奖励形式如下：

$$\max\sum_{k=0}^{+\infty}\beta^k R_{a_k}(c_k) \tag{3-57}$$

在 MDP 中，为了衡量每个状态的好坏，决定每个状态下的最优动作，对每一个过程状态 c 定义了值函数 $f(c)$，用来评估该状态下可以产生的总体奖励，一部分是该状态下进行动作 a 的即时奖励，另一部分是经过 a 转移到的所有可能状态 c' 对应的总体奖励。因此，根据文献[97]中的贝尔曼公式(Bellman Equation)，每个过程状态的最优值函数为

$$f_*(c)=\max_{a\in A}\left[R_a(c)+\beta\sum_{c'\in C}P_a(c,c')f_*(c')\right] \tag{3-58}$$

为了将优化问题(3-56)构造成上述的 MDP，首先需要建立过程状态，令 $\boldsymbol{c}_k=(e_{k-1}^{\mathrm{h}},e_k)$表示 k 时刻的状态，由上一时刻的收集能量和当前时刻的可利用能量状态组成，每个时刻需要分配的能量 e_k^{m} 就是需要做出的决策动作，量测数据的价值可以作为奖励，即给定能量分配下的 $\boldsymbol{J}_k^{\mathrm{d}}(e_k^{\mathrm{m}})$就是即时奖励，而能量分配的目标就是要实现整个过程累积 FIM 的最大化。为了便于计算，这里设置各个时刻的权重系数 $q_k=1$，使各时刻具有相同的权重，保证跟踪任务的稳定性。在实际任务中，根据跟踪需求的不同，可以对不同阶段灵活地设置权重系数，在此不再详述。

基于能量变化模型(3-40)，可以计算 $\boldsymbol{c}_k$ 在进行动作 e_k^{m} 时的状态转移概率

$$p(\boldsymbol{c}_{k+1}\mid\boldsymbol{c}_k,e_k^{\mathrm{m}})=p(e_k^{\mathrm{h}}\mid e_{k-1}^{\mathrm{h}})p(e_{k+1}\mid e_{k-1}^{\mathrm{h}},e_k,e_k^{\mathrm{m}})=p(e_k^{\mathrm{h}}\mid e_{k-1}^{\mathrm{h}}) \tag{3-59}$$

可以看出，在能量消耗确定的情况下，状态 $\boldsymbol{c}_k$ 的转移概率和能量收集过程相同。通过式(3-58)给出的状态值函数迭代关系和式(3-59)的转移概率，状态 $\boldsymbol{c}_k$ 的最优值函数可以逐步迭代计算

$$\begin{aligned}f_k(\boldsymbol{c}_k)&=\max_{e_k^{\mathrm{m}}}\left[\mathrm{tr}(\boldsymbol{J}_k^{\mathrm{d}}(e_k^{\mathrm{m}}))+\sum_{\boldsymbol{c}_{k+1}}f_{k+1}(\boldsymbol{c}_{k+1})p(\boldsymbol{c}_{k+1}\mid\boldsymbol{c}_k,e_k^{\mathrm{m}})\right]\\&=\max_{e_k^{\mathrm{m}}}\left[\mathrm{tr}(\boldsymbol{J}_k^{\mathrm{d}}(e_k^{\mathrm{m}}))+\sum_{i=1}^{H}f_{k+1}(\alpha_i,e_{k+1})p(\alpha_i\mid e_{k-1}^{\mathrm{h}})\right]\end{aligned} \tag{3-60}$$

其中，$f_k(\boldsymbol{c}_k)$表示状态 $\boldsymbol{c}_k$ 的值函数，它是时间区间 $[k,K]$内的累加 FIM。

式(3-60)中第一项表示了分配 e_k^{m} 后，期望从 k 时刻获得的即时 FIM，第二项则是期望在 $[k+1,K]$区间上获得的未来奖励。通过这一递归关系逐步计算，可以获得 $[1,K]$区间上的总体奖励，即代表了全局跟踪精度的累加 FIM。而对于最后时刻 K，由于不必考虑未来奖励，$f_K(\boldsymbol{c}_K)$的大小仅由即时奖励 $\boldsymbol{J}_K^{\mathrm{d}}(e_K^{\mathrm{m}})$决定，为了最大化奖励，应该用尽所剩能量处理量测数据。因此，K 时刻的值函数为

$$f_K(\boldsymbol{c}_K)=\max_{e_K^{\mathrm{m}}}\mathrm{tr}(\boldsymbol{J}_K^{\mathrm{d}}(e_K^{\mathrm{m}}))=\mathrm{tr}(\boldsymbol{J}_K^{\mathrm{d}}(e_K)) \tag{3-61}$$

在初始状态 $\boldsymbol{c}_1=(e_0^{\mathrm{h}},e_1)$ 确定的情况下，DP 是求解这一类 MDP 的常用算法，可以利用贝尔曼公式将多时刻的决策优化转化为一系列单一时刻的优化问题。根据贝尔曼公式的最优准则[98]，基于动态规划的求解方案可以保证动作决策的最优性。由于能量收集和消耗的过程，每个时刻的状态 $\boldsymbol{c}_k$ 有着多种数值可能，对于所有可能出现的状态，都需要计算出相应的最优能量分配 $e_k^{\mathrm{m}}(\boldsymbol{c}_k)$。考虑到最后时刻 K 没有后续状态，只需用尽所有能量，因此可以采用逆序求解的方式从 $K\sim1$ 时刻逐步求解。首先，对于所有可能的状态 $(e_{K-1}^{\mathrm{h}},e_K)$，利用式(3-61)计算相应的最优值函数 $f_K(\boldsymbol{c}_K)$，并将结果记录下来。随后，从 $K-1\sim1$ 时刻，利用式(3-60)可以计算各时刻所有可能状态的最优值函数，并将相应的最优能量分配 $e_k^{\mathrm{m}}(\boldsymbol{c}_k)$ 保存。至此，已经计算出所有状态对应的最优动作，当初始状态 $\boldsymbol{c}_1=(e_0^{\mathrm{h}},e_1)$ 确定时，可以根据计算结果决定各个时刻需要的最优动作。在获得各时刻的最优能量分配方案后，各节点依照分配方案对量测数据进行处理和传输，融合中心进一步融合各节点的量化数据进行目标状态估计。由于量化量测模型的高度非线性，以及在计算费舍尔信息时需要采用蒙特卡洛采样的方法，因此，本节采用粒子滤波算法实现目标跟踪，整体的跟踪过程如算法 3.2 所示。

算法 3.2　基于长时能量分配的水下目标跟踪算法

初始化	给定初始分布 $p(\boldsymbol{x}_0)$，根据 $p(\boldsymbol{x}_0)$ 采样得到 M 个粒子 $\{\boldsymbol{x}_0^i\}_{i=1}^M$，权重统一设置为 $w_0^i=1/M$
输入	$\{\boldsymbol{x}_{k-1}^i\}_{i=1}^M,\{\boldsymbol{x}_{k-1|0}^i\}_{i=1}^M,\{w_{k-1}^i\}_{i=1}^M,\boldsymbol{z}_k$ 对于时刻 $k=1,2,\cdots$，循环进行以下步骤
步骤 1.	利用式(3-53)生成长时预测粒子 $\{\boldsymbol{x}_{k|0}^i\}_{i=1}^M$
步骤 2.	利用式(3-61)计算并记录 K 时刻所有可能状态的最优值函数 $f_K(\boldsymbol{c}_K)$
步骤 3.	利用式(3-60)计算并记录各时刻所有可能状态 $(e_{k-1}^{\mathrm{h}},e_k)$ 的最优值函数和相应动作
步骤 4.	根据计算结果确定各时刻的最优能量分配 e_k^{m}，利用式(3-36)和式(3-41)确定量化等级 L 和阈值集合 γ
步骤 5.	基于目标估计状态生成下一时刻的预测粒子 $\boldsymbol{x}_k^i\sim p(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1}^i)$
步骤 6.	计算权重 $w_k^i=w_{k-1}^i p(\boldsymbol{z}_k\mid\boldsymbol{x}_k^i)$ 并归一化得到 $\{\tilde{w}_k^i\}_{i=1}^M$
步骤 7.	得到 k 时刻估计为 $\hat{\boldsymbol{x}}_k=\sum_{i=1}^M\tilde{w}_k^i\boldsymbol{x}_k^i$
输出	$\{\boldsymbol{x}_k^i\}_{i=1}^M,\{\boldsymbol{x}_{k|0}^i\}_{i=1}^M,\{w_k^i\}_{i=1}^M,\hat{\boldsymbol{x}}_k$

3.2.4　仿真结果与分析

为验证和分析本节提出的算法，本节进行计算机仿真实验，模拟基于长时能量分配的 UWSN 目标跟踪过程，并与其他能量分配方案的跟踪结果对比，给出仿真结果和性能分析。

由于本节主要研究各节点能量分配对目标跟踪的影响，为了便于对比各个节点能量分配过程的区别，本节设定了一个二维平面内均匀部署的 UWSN，目标一直在监测区域内做 CT 运动，不考虑深度方向的运动，以保证各节点与目标轨迹之间的对称性，且这样的场景不会对目标跟踪的结果产生特殊影响。图 3.17 给出了具体的仿真场景，假定 4 个固定的节点均匀部署在二维平面水域，监测区域的整体范围大致为 300m×300m，所有 4 个节点均可探测到监测区域的运动目标，且它们具有相同的探测范围和量测误差，量测误差均服从分布 $N(0,10)$，融合中心位于水面中心位置，通过获取 4 个节点的数据进行工作，各节点量化量测的最大位数设定为 $\bar{B}=5$，而相应的量化阈值集合 γ 可以参考文献[93]的方法

确定。运动目标的初始状态为 $\boldsymbol{x}_0=[150,10,60,0]^{\mathrm{T}}$，在二维平面内做 CT 运动，时间长度为 40s。

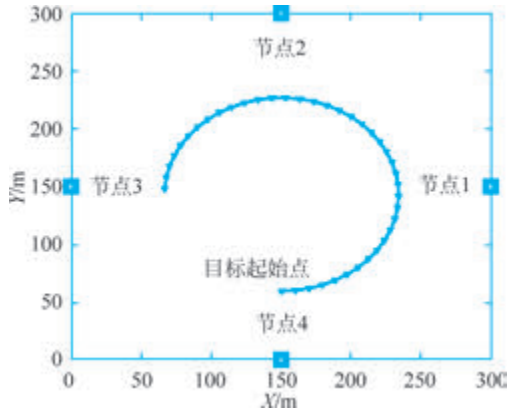


图 3.17　仿真场景

具体的运动模型为

$$\boldsymbol{x}_{k+1}=\boldsymbol{F}_k\boldsymbol{x}_k+\boldsymbol{w}_k \tag{3-62}$$

其中，状态转移矩阵 $\boldsymbol{F}_k$ 和过程噪声 $\boldsymbol{w}_k$ 的协方差矩阵 $\boldsymbol{Q}_k$ 分别为

$$\boldsymbol{F}_k=\begin{bmatrix}1 & \dfrac{\sin(\omega T)}{\omega} & 0 & -\dfrac{1-\cos(\omega T)}{\omega}\\ 0 & \cos(\omega T) & 0 & -\sin(\omega T)\\ 0 & \dfrac{1-\cos(\omega T)}{\omega} & 1 & \dfrac{\sin(\omega T)}{\omega}\\ 0 & \sin(\omega T) & 0 & \cos(\omega T)\end{bmatrix} \tag{3-63}$$

$$\boldsymbol{Q}_k=q^2\begin{bmatrix}\dfrac{2[\omega T-\sin(\omega T)]}{\omega^3} & \dfrac{1-\cos(\omega T)}{\omega^2} & 0 & \dfrac{\omega T-\sin(\omega T)}{\omega^2}\\ \dfrac{1-\cos(\omega T)}{\omega^2} & T & -\dfrac{\omega T-\sin(\omega T)}{\omega^2} & 0\\ 0 & -\dfrac{\omega T-\sin(\omega T)}{\omega^2} & \dfrac{2[\omega T-\sin(\omega T)]}{\omega^3} & \dfrac{1-\cos(\omega T)}{\omega^2}\\ \dfrac{\omega T-\sin(\omega T)}{\omega^2} & 0 & \dfrac{1-\cos(\omega T)}{\omega^2} & T\end{bmatrix} \tag{3-64}$$

其中，ω 是固定的角速度，T 是采样间隔，q^2 表示过程噪声的强度。

实验结果通过 100 次蒙特卡洛(MC=100)仿真平均得到，为了衡量目标跟踪的精度，本节通过计算均方根误差作为跟踪性能的判据。

$$\varepsilon_k=\sqrt{\frac{1}{\mathrm{MC}}\sum_{m=1}^{\mathrm{MC}}\left[(x_k^{i,1}-\hat{x}_k^{i,1})^2+(x_k^{i,3}-\hat{x}_k^{i,3})^2\right]} \tag{3-65}$$

其中，$(x_k^{i,1},x_k^{i,3})$和$(\hat{x}_k^{i,1},\hat{x}_k^{i,3})$分别表示目标在 k 时刻的真实位置和估计位置。

由于各节点的能量消耗与量化量测的位数线性相关，为便于表述，以一位数据的能耗 λ 作为能量单位，能量收集设定为包含两个状态$\{\alpha_1=\lambda,\alpha_2=3\lambda\}$的马尔可夫过程，相应的转移概率为

$$\boldsymbol{P}_\alpha=\begin{bmatrix}p_{\alpha_{11}} & p_{\alpha_{12}}\\ p_{\alpha_{21}} & p_{\alpha_{22}}\end{bmatrix} \tag{3-66}$$

其中，$p_{\alpha_{ij}}$ 表述能量收集状态 $e_k^{\mathrm{h}}=\alpha_i$ 转移到 $e_{k+1}^{\mathrm{h}}=\alpha_j$ 的概率。由此可以计算相应的稳定状态概率

$$\begin{cases} p_{\alpha_1}=\dfrac{p_{\alpha_{21}}}{p_{\alpha_{21}}+p_{\alpha_{12}}} \\ p_{\alpha_2}=\dfrac{p_{\alpha_{12}}}{p_{\alpha_{21}}+p_{\alpha_{12}}} \end{cases} \tag{3-67}$$

在仿真中设定 $p_{\alpha_1}=p_{\alpha_2}=0.5$，则能量收集平均值为 $\bar{\alpha}=p_{\alpha_1}\alpha_1+p_{\alpha_2}\alpha_2=2\lambda$。

本节算法综合考虑时间区间 [1,40] 上的全局跟踪性能，并给出了相应的跟踪结果(命名为：长时分配)。此外，还给出了其他能量分配方案下的跟踪结果作为对比。短时分配方案与本节方案相反，节点在各个时刻都会消耗尽可能多的能量进行量测，不考虑对后续时刻的影响；在均匀分配方案中，节点通过对初始能量和收集能量的预估，均匀地分配至各个时刻。图 3.18 给出了 3 种方案的跟踪误差，图 3.19 则展现了 3 种方案中所有节点能量分配的平均结果。相比于短时分配方案，本节提出的长时分配由于实现了全局最优的能量分配，可以显著提升目标跟踪的精度。可以看出，在跟踪的初期阶段(如时间区间 [1,10])，短时分配的方案可以实现最好的跟踪性能，这是因为前期电池有足够的储能，使得各节点的量测位数比较多，提供了充足的数据信息用于估计目标状态；而由于前期过度消耗能量，跟踪后期节点只能依赖上一时刻的收集能量，较低的能量收集效率使得节点不得不压缩数据量，跟踪结果也随之变差。而长时分配的方案综合考虑全局性能，在时间区间内合理地分配能量，尽管跟踪初期节省能量导致跟踪误差偏大，但是通过能量的合理分配，后期可以保证稳定良好的跟踪误差，实现了最好的跟踪精度。均匀分配的方案也是基于全局考虑的思路，每个节点对各时刻均匀分配所有能量，但是各个节点的方位不同，它们的量测信息量也有着不同的变化，都采用均匀分配的方案势必会导致能量利用率降低，因此得到了介于两者之间的跟踪误差。综合比较，在消耗等量资源的条件下，长时分配的方案实现了最好的跟踪结果，提高了能量的利用效率。

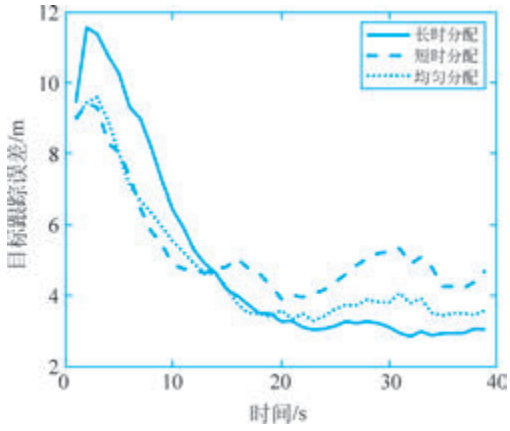


图 3.18 3 种分配方案下的目标跟踪误差

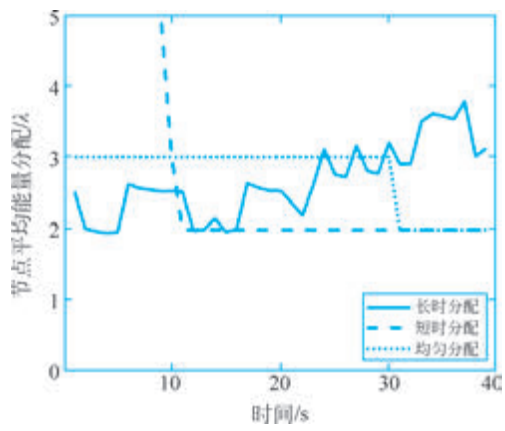


图 3.19 3 种分配方案下节点的平均能量分配

图 3.20 展现了目标跟踪过程中，4 个工作节点各自的能量分配的变化过程。可以看出，通过能量的自组织分配，各个节点的能量分配有着不同的变化过程，这是因为各个节点的量测价值存在差异，随目标运动的变化过程也不同。参考目标的运动场景图 3.2，初始阶段节点 1 和节点 4 与目标距离较近，量测价值更高，因此跟踪前期它们分配更多能量处理数据；随着目标逐渐靠近节点 2 和节点 3，它们可提供更多有效信息，其能量分配逐渐增多，节点 1 和节点 4 的能量分配则相应减少；在跟踪过程的最后

阶段，由于目标再次靠近节点 4，其能量分配再次增加，以提供更多量测信息。因此，和均匀分配方案相比，尽管两者的总体能量分配平均结果相似，但是长时分配方案通过各节点量测价值的评估，自组织地实现各节点不同的分配变化，使得能量资源得到有效利用，实现了最好的跟踪性能。

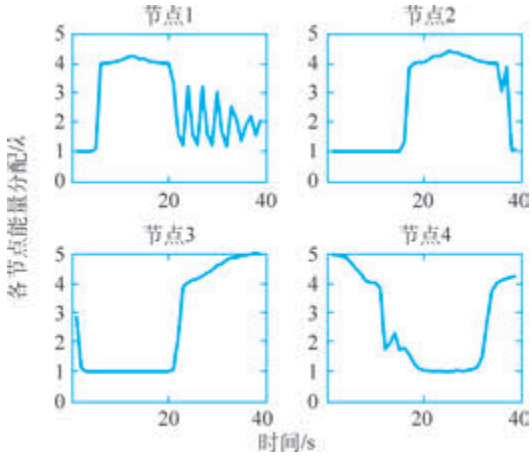


图 3.20　4 个节点各自的能量分配：长时分配方案

为了进一步研究能量分配对目标跟踪性能的影响，接下来的实验比较了不同参数下的实验结果。首先研究能量收集过程对目标跟踪的影响，在节点初始能量预算 $e_1=30\lambda$ 固定的情况下，图 3.21～图 3.24 给出了不同能量收集效率（$\alpha=\lambda,2\lambda,3\lambda,4\lambda$）下 3 种方案的实验结果。图 3.21 和图 3.22 展现了基于长时能量分配的目标跟踪误差和节点平均能量分配，图 3.23 和图 3.24 则展现了均匀分配和短时分配方案的跟踪误差。显然，无论能量收集效率的高低，长时分配方案总能保持稳定、良好的跟踪性能；对于短时分配方案，在能量收集效率足够高时，它也可以实现足够好的跟踪精度，但是当能量收集效率下降时，跟踪性能受到很大影响；而均匀分配的结果始终介于两者之间。表 3.3 记录了 3 种方案的平均跟踪误差，随着能量收集效率的提高，由于节点有了充足的能量处理各个时刻的量测数据，可以缓解信息的损失，所以另外两种方案的精度逐渐接近长时分配。然而，在实际的水下环境中节点的能量收集效率通常较低，节点可利用能量十分有限，提高能量的利用效率非常重要，长时分配的方案可以在较低吸收效率的条件下保持良好的跟踪精度，更适用于水下环境。

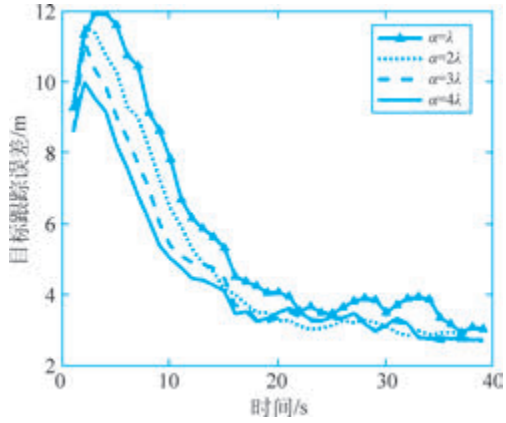


图 3.21　不同能量收集效率下的目标跟踪误差：长时分配方案

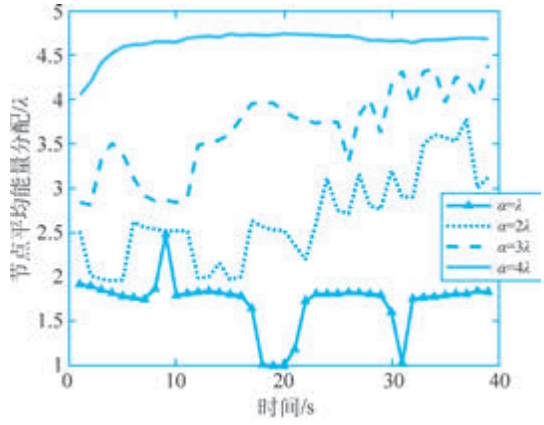


图 3.22　不同能量收集效率下的节点平均能量分配：长时分配方案

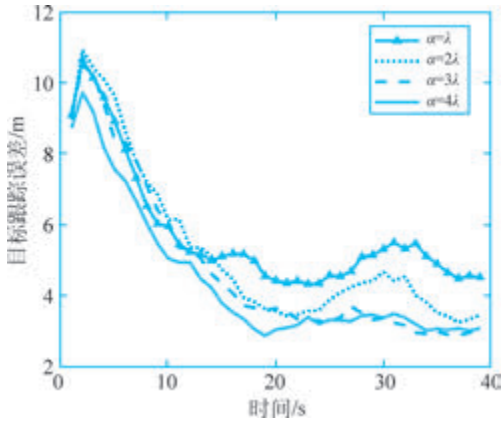


图 3.23 不同能量收集效率下的目标跟踪误差：均匀分配方案

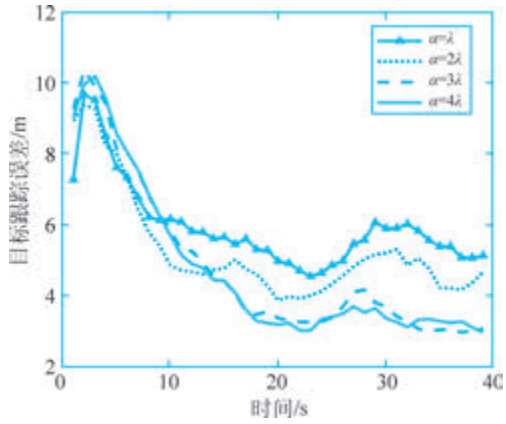


图 3.24 不同能量收集效率下的目标跟踪误差：短时分配方案

表 3.3 不同能量收集效率下 3 种方案的平均跟踪误差

方　案	$\alpha=\lambda$	$\alpha=2\lambda$	$\alpha=3\lambda$	$\alpha=4\lambda$
长时分配	4.230m	3.878m	3.612m	3.579m
均匀分配	5.467m	4.786m	4.037m	3.847m
短时分配	6.340m	5.511m	4.189m	4.014m

随后，在固定能量收集效率（$\alpha=2\lambda$）的条件下，研究了初始能量预算的变化对跟踪结果的影响。图 3.25 和图 3.26 展现了初始能量预算（$e_1=30\lambda,50\lambda,70\lambda,90\lambda$）下长时分配方案的跟踪误差和能量分配，图 3.27 和图 3.28 则展现了另外两种方案的跟踪误差。从图 3.27 和图 3.28 中以及表 3.4 的平均误差可以看出，即使可利用能量较少，长时分配的方案仍然能够保持稳定的跟踪精度，而另外两种方案受能量预算影响较大，只有在能量预算充足时才能实现较好的跟踪效果。考虑到水下节点能量十分有限，通常需要在保证一定跟踪精度的同时减小能量的消耗，基于长时能量分配的方案可以在较小的能量预算下实现良好的跟踪精度，这对于 UWSN 水下目标跟踪有着重要意义。

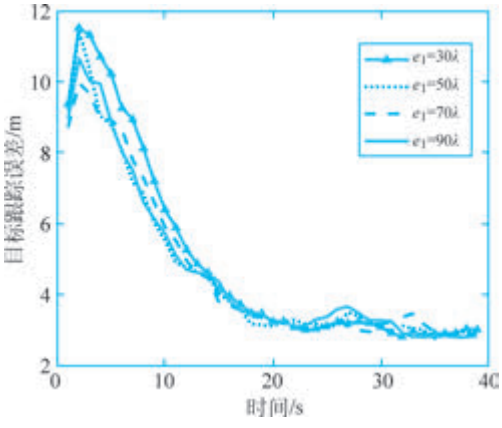


图 3.25 不同能量预算下的目标跟踪误差：长时分配方案

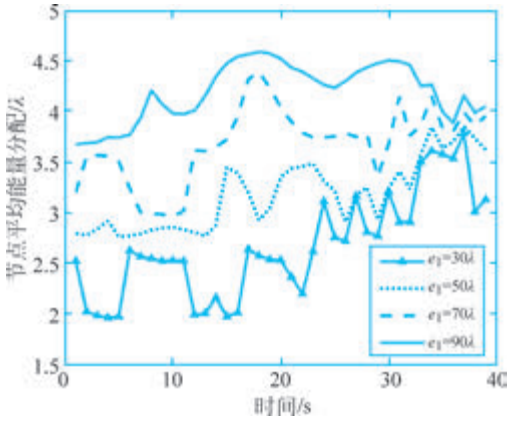


图 3.26 不同能量预算下的节点平均能量分配：长时分配方案

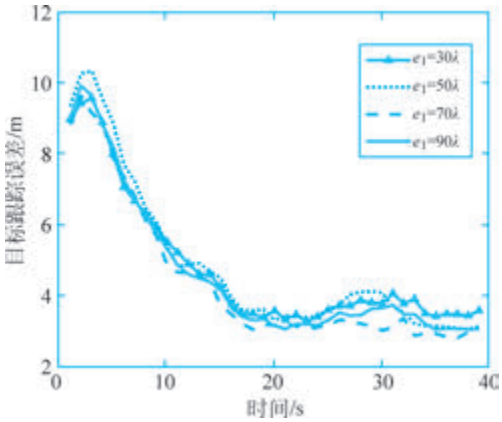


图 3.27 不同能量预算下的目标跟踪误差：均匀分配方案

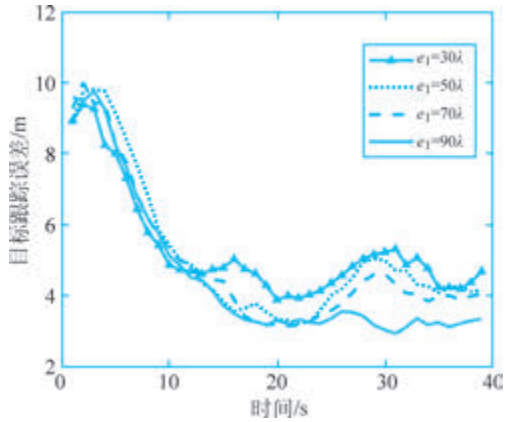


图 3.28 不同能量预算下的目标跟踪误差：短时分配方案

表 3.4 不同能量预算下 3 种方案的平均跟踪误差

方　　案	$e_1=30\lambda$	$e_1=50\lambda$	$e_1=70\lambda$	$e_1=90\lambda$
长时分配	3.878m	3.807m	3.792m	3.673m
均匀分配	4.786m	4.023m	3.810m	3.701m
短时分配	5.511m	4.915m	4.189m	3.668m

基于长时能量分配的方案利用动态规划算法求解问题，这种方法的计算复杂度为 $O(\bar{B}\cdot K\cdot H)$，当最大量化位数 $\bar{B}$ 和能量收集状态数量 H 固定时，复杂度随跟踪时间的长度而变化。当跟踪时间 K 很长时，可能会造成较大的计算负担。为了解决这一问题，本节对算法进行改进，进一步给出一种次优的解决方案。状态值函数式(3-60)由两部分组成：一部分是当前时刻的即时奖励，另一部分是后续过程的未来奖励。当 K 较大时，主要的计算负担来自递归计算未来奖励，因此可以将总时间划分为多个子区间，对多个子区间同时计算能量分配，以并行计算的方式提高计算效率，并根据子区间的时间长短均匀地将初始能量预算分配给子区间。对于每个子区间，只需计算相应时长 P 内的能量分配。当子区间 P 比较小时，每个子区间的计算量减小为 $O(\bar{B}\cdot P\cdot H)$，并行计算可以提高整体运行效率。尽管次优分配方案实现了各子区间内能量的最优分配，然而由于各子区间内的量测价值不同，相比全局长时分配，这种方法会使得跟踪精度的下降。在实验中，本节简单地将整个过程均分为长度为 P 的子区间，图 3.29 和表 3.5 给出了不同子区间长度下的平均跟踪误差。可以看出，这种次优方案的跟踪误差随着 P 减小而增大，逐渐逼近均匀分配方案。显然，当子区间长度 $P=1$ 时，由于初始能量的均分，且各时刻只考虑自己的即时奖励，次优方案等价于均匀分配方案。

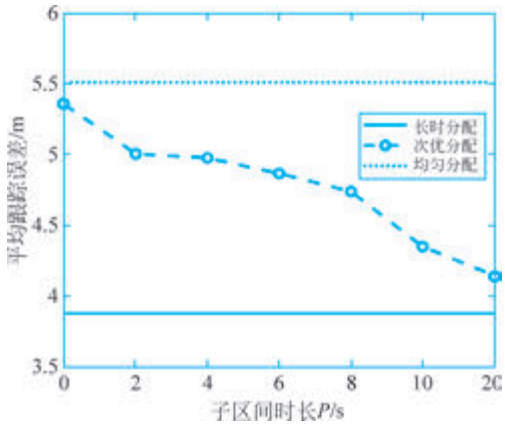


图 3.29 不同时长 P 下的平均跟踪误差：次优分配方案

表 3.5 不同时长 P 下的平均跟踪误差：次优分配方案

子区间时长/s	$P=1$	$P=2$	$P=4$	$P=6$	$P=8$	$P=10$	$P=20$
平均跟踪误差/m	5.358	5.006	4.977	4.867	4.740	4.361	4.146

3.3 基于动态分簇的节点调度方法

将多个测量平台灵活部署在待监测海域，可形成自组织的分布式 UWSN。该网络不仅能实时、连续、全天候地从多个角度获取多维度、互补性的目标探测信息，使目标探测不再受时间和空间约束，而且能通过各个平台之间的信息共享与协同工作，结合信息融合技术和目标跟踪技术，对水下目标的运动参数进行连续估计。目前各界学者已经提出了多种具有不同计算复杂度和跟踪精度的水下目标跟踪方法，目前主要面临两方面的挑战。一是目标的跟踪精度。基于到达时间差的方法要求各测量平台间始终精确同步[99]，这限制了该方法的应用；基于节点接收声源强度的方法容易受传播信道和噪声的影响，导致定位精度较差；方位作为目标辐射信号的最可靠参数之一，被广泛应用于水下目标的跟踪中[100]，基于纯方位的方法对节点定向能力要求较高，而且要求平台要至少进行一次机动，但这些条件在 UWSN 中都很容易满足，因而有必要深入研究如何利用已经获得的目标方位信息来进一步提高目标跟踪精度。二是网络的能耗问题。由于 UWSN 中的节点均使用电池提供能量，在特殊的水下环境中对能量耗尽的电池进行更换或维护是非常困难的，而且水下节点发送声波耗能较高，UWSN 的使用寿命往往取决于节点的能耗速度。因此在纯方位信息下，如何在保证跟踪精度收敛的同时，有效降低网络的能耗、延长网络使用寿命是本节需要解决的主要问题。

针对这一问题，本节提出了一种基于动态分簇的水下目标跟踪算法。该算法根据目标的融合状态预测值和融合预测误差协方差动态选择参与水下目标跟踪过程的簇头节点和簇成员节点，选择过程不仅考虑了节点纯方位信息的效用，而且考虑了节点的剩余能量和消耗能量，同时将节点的选择问题归类为数学中的背包问题，并利用 DP 的方法来选择最优节点组合实现目标函数的最大化。进一步地，提出一种基于动态簇的水下目标被动跟踪算法，在目标跟踪的过程中采用分量按标量加权的线性最小方差融合准则来极小化融合误差协方差的迹，求得最优融合状态估计，提高目标动态参数估计精度。

3.3.1 跟踪模型

本节采用的 UWSN 模型如图 3.30 所示，主要由普通节点、超级节点（融合中心）、漂浮节点、海岸基站构成。普通节点只能与邻近的普通节点和超级节点进行通信，主要负责水下目标声源的探测和估计；超级节点与普通节点、超级节点和漂浮节点之间均能直接进行通信，主要负责收集普通节点获得的目标信息，经必要的融合处理后，通过单跳或者多跳的方式传送到漂浮节点，超级节点是连通网络和收集、处理数据的关键节点；漂浮节点接收超级节点发送的信息，进一步融合处理后，以电磁波通信的方式将信

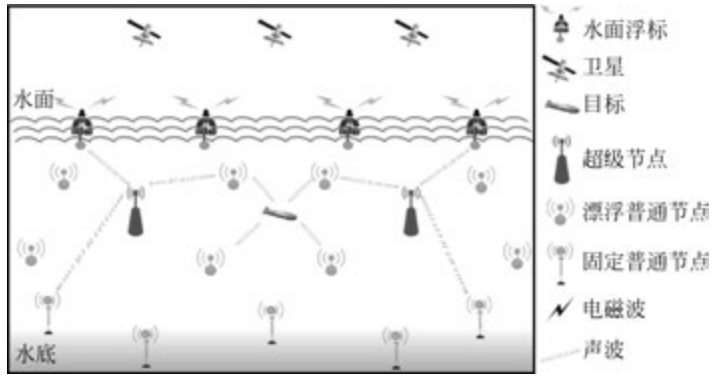


图 3.30 UWSN 模型

息转发给海岸基站。最后经过海岸基站的进一步融合处理，得到准确的水下目标运动参数信息。这里，假设所有的节点都是同步的，并可通过 GPS 或者节点的自定位算法确定自身的位置，目前关于节点自定位的研究已经有很多[101]，在这里不再详细进行说明。

假设在笛卡儿坐标系下，k 时刻目标在三维 UWSN 中的状态向量表示为

$$\boldsymbol{x}_k=[x_k,\dot{x}_k,\ddot{x}_k,y_k,\dot{y}_k,\ddot{y}_k,z_k,\dot{z}_k,\ddot{z}_k]^{\mathrm{T}} \tag{3-68}$$

其中，元素 $x_k,\dot{x}_k,\ddot{x}_k,y_k,\dot{y}_k,\ddot{y}_k$ 以及 $z_k,\dot{z}_k,\ddot{z}_k$ 分别表示目标在 x、y 和 z 方向的位置、速度和加速度。因此，目标的运动模型可以表示为

$$\boldsymbol{x}_k=\boldsymbol{F}_{k-1}\boldsymbol{x}_{k-1}+\boldsymbol{w}_{k-1} \tag{3-69}$$

其中，$\boldsymbol{F}_{k-1}\in\mathbb{R}^{9\times 9}$ 表示运动目标在 $k-1$ 时刻的状态转移矩阵，$\boldsymbol{w}_{k-1}$ 表示服从正态分布 $N(\boldsymbol{0}_9,\boldsymbol{Q}_{k-1})$ 的过程噪声。

在本节中，假设目标在三维监测区域内以某种方式进行机动运动，当目标进行 CV 运动时，目标的状态转移矩阵表示为

$$\boldsymbol{F}_{k-1}=\begin{bmatrix}\boldsymbol{F}_1 & 0 & 0\\ 0 & \boldsymbol{F}_1 & 0\\ 0 & 0 & \boldsymbol{F}_1\end{bmatrix},\quad \boldsymbol{F}_1=\begin{bmatrix}1 & T & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix} \tag{3-70}$$

当目标进行匀加速(Constant Acceleration，CA)运动时，目标的状态转移矩阵表示为

$$\boldsymbol{F}_{k-1}=\begin{bmatrix}\boldsymbol{F}_2 & 0 & 0\\ 0 & \boldsymbol{F}_2 & 0\\ 0 & 0 & \boldsymbol{F}_2\end{bmatrix},\quad \boldsymbol{F}_2=\begin{bmatrix}1 & T & T^2/2\\ 0 & 1 & T\\ 0 & 0 & 1\end{bmatrix} \tag{3-71}$$

当目标在 xOy 平面进行转弯速率为 ωrad/s 的 CT 运动，而在 z 轴方向进行 CV 运动时，目标的状态转移矩阵可以表示为

$$\boldsymbol{F}_{k-1}=\begin{bmatrix}\boldsymbol{F}_3 & 0\\ 0 & \boldsymbol{F}\end{bmatrix} \tag{3-72}$$

其中，

$$\boldsymbol{F}_3=\begin{bmatrix}1 & \dfrac{\sin(\omega T)}{\omega} & 0 & 0 & \dfrac{\cos(\omega T)-1}{\omega} & 0\\ 0 & \cos(\omega T) & 0 & 0 & -\sin(\omega T) & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & \dfrac{1-\cos(\omega T)}{\omega} & 0 & 1 & \dfrac{\sin(\omega T)}{\omega} & 0\\ 0 & \sin(\omega T) & 0 & 0 & \cos(\omega T) & 0\\ 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} \tag{3-73}$$

协方差矩阵 $\boldsymbol{Q}_{k-1}$ 表示为

$$\boldsymbol{Q}_{k-1}=q^2\begin{bmatrix}\boldsymbol{Q}_2 & 0 & 0\\ 0 & \boldsymbol{Q}_2 & 0\\ 0 & 0 & \boldsymbol{Q}_2\end{bmatrix},\quad \boldsymbol{Q}_2=\begin{bmatrix}T^4/2 & T^3/2 & T^2/2\\ T^3/2 & T^2/2 & T\\ T^2/2 & T & 1\end{bmatrix} \tag{3-74}$$

其中，q 表示过程噪声的强度；T 表示系统的采样间隔。

假设在 k 时刻共选择 $N_k(N_k\geqslant 2)$ 个节点参与目标的监测与跟踪任务，坐标分别表示为 $\boldsymbol{P}_{i,k}=(x_{i,k},y_{i,k},z_{i,k})$，其中，$i=1,2,\cdots,N_k$。每个节点都可以测量目标相对于自身的方向角 θ 和俯仰角 ϕ，

三维角度测量的几何表示如图 3.31 所示。第 i 个节点在 k 时刻的测量方程表示为

$$\boldsymbol{z}_{i,k}=\boldsymbol{h}(\boldsymbol{x}_{i,k})+\boldsymbol{v}_{i,k}=\begin{bmatrix}\widetilde{\theta}_{i,k}\\ \widetilde{\phi}_{i,k}\end{bmatrix}=\begin{bmatrix}\theta_{i,k}\\ \phi_{i,k}\end{bmatrix}+\begin{bmatrix}v_{i,k}^{\theta}\\ v_{i,k}^{\phi}\end{bmatrix} \tag{3-75}$$

其中，$\boldsymbol{v}_{i,k}=\begin{bmatrix}v_{i,k}^{\theta} & v_{i,k}^{\phi}\end{bmatrix}^{\mathrm{T}}$ 为零均值高斯白噪声，$v_{i,k}^{\theta}$ 和 $v_{i,k}^{\phi}$ 相互独立；协方差矩阵为 $\boldsymbol{R}_{i,k}=\mathrm{diag}\{(\delta_{i,k}^{\theta})^2,(\delta_{i,k}^{\phi})^2\}$。本节假设所有节点具有相同的测量噪声统计特性，即 $(\delta_{i,k}^{\theta})^2=(\delta_{i,k}^{\phi})^2=\delta_k^2$。$\boldsymbol{z}_{i,k}=\begin{bmatrix}\widetilde{\theta}_{i,k} & \widetilde{\phi}_{i,k}\end{bmatrix}^{\mathrm{T}}$ 表示方向角和俯仰角的测量值，$\begin{bmatrix}\theta_{i,k} & \phi_{i,k}\end{bmatrix}^{\mathrm{T}}$ 为方向角和俯仰角的准确值，分别表示为

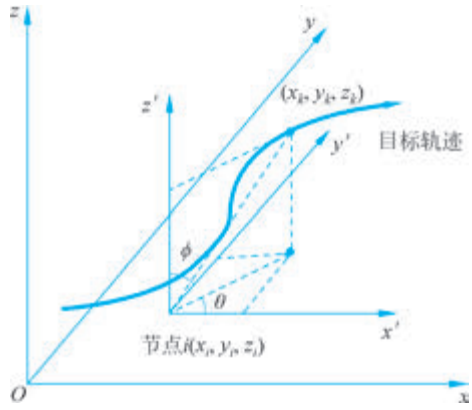


图 3.31　三维角度测量的几何表示

$$\theta_{i,k}=\arctan\frac{y_k-y_{i,k}}{x_k-x_{i,k}} \tag{3-76}$$

$$\phi_{i,k}=\arctan\frac{\sqrt{(y_k-y_{i,k})^2+(x_k-x_{i,k})^2}}{z_k-z_{i,k}} \tag{3-77}$$

其中，(x_k,y_k,z_k) 表示目标在 k 时刻的坐标。

3.3.2　基于自适应节点选择的动态分簇方法

在三维 UWSN 的监测区域中，共部署有 N 个传感器节点，假设所有的传感器节点都具有相同的感知范围 R_S 和通信范围 R_C，第 i 传感器节点的位置用 $\boldsymbol{P}_i=(x_i,y_i,z_i)$ 来表示，其中 $i=1,2,\cdots,N$。在 k 时刻，算法利用自适应选择的 N_k 个簇成员节点和簇头节点 CH_k 组成一个动态簇共同参与目标的跟踪过程。跟踪策略的基本思想如图 3.32 所示，其相应的基本流程如图 3.33 所示。

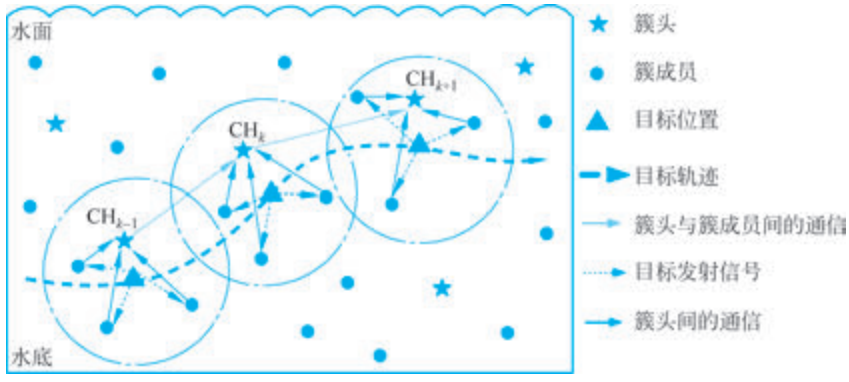


图 3.32　跟踪策略的基本思想

首先，选择的 N_k 个簇成员节点将根据上一时刻簇头节点 CH_{k-1} 的融合状态预测值 $\hat{\boldsymbol{x}}_{k|k-1}$ 以及融合预测误差协方差矩阵 $\hat{\boldsymbol{x}}_{k|k-1}$，分别使用 UKF 算法获得目标在 k 时刻的本地状态估计 $\hat{\boldsymbol{x}}_{i,k|k}$ 以及相应的误差协方差矩阵 $\boldsymbol{P}_{i,k|k}$，其中，$i=1,2,\cdots,N_k$，然后每一个簇成员节点将 $\hat{\boldsymbol{x}}_{i,k|k}$ 和 $\boldsymbol{P}_{i,k|k}$ 发送至 k 时刻的簇头节点 CH_k；其次，簇头节点 CH_k 作为融合中心，主要负责收集、融合各个簇成员节点的对目

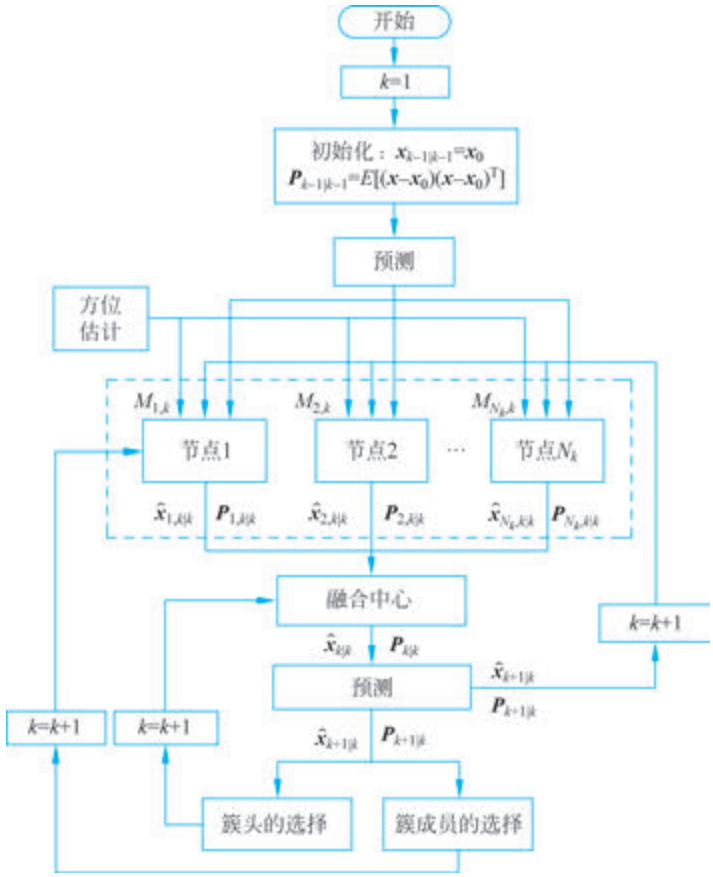


图 3.33 目标跟踪算法流程图

标的本地状态估计 $\hat{\boldsymbol{x}}_{i,k|k}$，并根据 $\hat{\boldsymbol{x}}_{i,k|k}$ 通过分布式融合算法得到 k 时刻目标状态的融合估计 $\hat{\boldsymbol{x}}_{k|k}$ 和融合后的协方差矩阵 $\boldsymbol{P}_{k|k}$，并将融合结果以单跳或多跳的方式传递给水面基站，水面基站通过电磁波通信的方式将信息发送到岸基控制中心，用于下一步的决策与控制；再次，簇头节点 CH_k 根据 k 时刻对目标的融合状态估计 $\hat{\boldsymbol{x}}_{k|k}$ 以及相应的融合误差协方差 $\boldsymbol{P}_{k|k}$，对 $k+1$ 时刻的目标状态进行预测得到目标的融合状态预测值 $\hat{\boldsymbol{x}}_{k+1|k}$ 以及融合预测误差协方差 $\boldsymbol{P}_{k+1|k}$；最后，簇头节点 CH_k 根据目标的融合状态预测值 $\hat{\boldsymbol{x}}_{k+1|k}$ 以及融合预测误差协方差矩阵 $\boldsymbol{P}_{k+1|k}$ 从被激活的簇成员节点候选集合 S_{k+1} 和簇头节点候选集合 C_{k+1} 中分别选择在 $k+1$ 时刻参与目标跟踪过程的簇成员节点和簇头节点，并将相应的融合状态预测值 $\hat{\boldsymbol{x}}_{k+1|k}$ 以及融合预测误差协方差矩阵 $\boldsymbol{P}_{k+1|k}$ 发送给选择的在 $k+1$ 时刻参与目标跟踪过程的簇头节点和簇成员节点。

假设在三维 UWSN 的监测区域中部署的所有传感器节点都具有相同的感知范围 R_{S} 和通信范围 R_{C}。为了降低网络能耗，每一个节点都可以根据实际情况在休眠（状态为 1）和活跃（状态为 0）两种状态之间相互转换。监测区域中每一个节点都配置有声敏传感器，当目标进入监测区域时，目标发出声波信号的声波强度可以被节点中的声敏传感器检测到。当节点检测到的目标发出声信号的声强超过一定的阈值，且节点的剩余能量超过一定的阈值时，该节点检测到目标，节点被激活，其他节点仍然保持在睡眠状态。

声源强度的衰减与距离成反比，k 时刻第 i 个传感器节点检测到的目标信号强度表示为

$$I_i = \frac{g_i I}{\| \boldsymbol{P}_i - \boldsymbol{T}_k \|^{\alpha}} \tag{3-78}$$

其中，g_i 表示第 i 个传感器的增益系数；I 表示目标辐射的声音强度；$\boldsymbol{T}_k$ 表示目标在 k 时刻的位置；$\| \boldsymbol{P}_i - \boldsymbol{T}_k \|$ 表示第 i 个传感器与目标之间的距离；α 表示强度衰减系数。

定义所有被激活的普通节点为簇成员候选集合 S_k，被激活的超级节点定义为簇头候选集合 C_k，S_k 和 C_k 分别需要满足以下条件

$$\begin{cases} C_k = \{ i \mid I_i \geqslant I_{\text{th}}/\varepsilon, \quad E_i \geqslant E_{\text{CH_th}} \} \\ S_k = \{ j \mid I_j \geqslant I_{\text{th}}/\varepsilon, \quad E_j \geqslant E_{\text{MEM_th}} \} \end{cases} \tag{3-79}$$

其中，$\varepsilon \in [0,1]$ 是一个自定义的参数，由于噪声的影响，被激活的传感器节点接收的声强度 $I_j \geqslant I_{\text{th}}/\varepsilon$，$\varepsilon$ 越大，表示候选集合 C_k 和 S_k 中候选节点的数目就越多；$I_{\text{th}}(\text{W/m}^2)$ 对应最大感知范围 R_{S} 处计算的声强；$E_{\text{MEM_th}}$ 和 $E_{\text{CH_th}}$ 为候选簇成员节点和候选簇头节点对应的能量硬门限，即节点的当前剩余能量值低于硬门限时，节点不能加入候选集合 C_k 和 S_k，k 时刻所有激活的超级节点和普通节点将有可能在 $k+1$ 时刻动态成簇，参与目标跟踪的过程。

簇头的选择在水下目标跟踪中起到至关重要的作用。由于簇头在进行数据收集、目标状态融合估计、节点选择、数据传输以及簇头切换时将会比普通的簇成员节点消耗更多的能量。因此，本节在簇头的选择时主要考虑能量消耗、节点剩余能量以及候选簇头节点与预测目标位置之间的关系。

假设在 k 时刻，簇头节点 CH_k 根据目标状态的融合估计 $\hat{\boldsymbol{x}}_{k|k}$ 预测目标在 $k+1$ 时刻的状态 $\hat{\boldsymbol{x}}_{k+1|k}$，并根据预测信息完成 $k+1$ 时刻簇头节点 CH_{k+1} 的选择，本节参考文献[102]提出的簇头节点的选择算法，根据如下规则完成簇头节点的选择：

若 $\text{CH}_k \in C_{k+1}$，则有

$$\text{CH}_{k+1} = \text{CH}_k \tag{3-80}$$

否则

$$\text{CH}_{k+1} = \left\{ i \left| \begin{array}{l} \text{dis}(i, \text{CH}_k) < R_{\text{C}} \\ \text{dis}(i, j \in S_{k+1}) < R_{\text{C}} \\ \arg\max(F_i) \end{array} \right. \right\}, \quad i \in C_{k+1} \tag{3-81}$$

其中，C_{k+1} 和 S_{k+1} 分别表示 $k+1$ 时刻簇头和簇成员的候选集合；$\text{dis}(i, \text{CH}_k)$ 是当前簇头 CH_k 与集合 C_{k+1} 中的第 i 个候选簇头节点之间的距离；同理，$\text{dis}(i, j \in S_{k+1})$ 表示第 i 个候选簇头节点与集合 S_{k+1} 中第 j 个候选簇成员节点之间的距离。

$$F_i = \alpha \frac{\langle \boldsymbol{r}_i, \boldsymbol{r}_{k+1} \rangle}{\| \boldsymbol{r}_i \| \cdot \| \boldsymbol{r}_{k+1} \|} + \beta \frac{\text{dis}(i, \text{CH}_k)^2}{R_{\text{C}}^2} + \gamma \frac{E_{i,k+1}^{\text{CH,init}} - E_{i,k+1}^{\text{CH}}}{E_{i,k+1}^{\text{CH,init}}} \tag{3-82}$$

其中，$\alpha + \beta + \gamma = 1$，用于根据实际场景调节 3 个分量的比率，图 3.34 说明了式(3-82)所表示的几何意义。$\langle \boldsymbol{r}_i, \boldsymbol{r}_{k+1} \rangle / (\| \boldsymbol{r}_i \| \cdot \| \boldsymbol{r}_{k+1} \|)$ 表示由当前簇头 CH_k 到目标预测位置 $\hat{\boldsymbol{x}}_{k+1|k}$ 的向量 $\boldsymbol{r}_{k+1}$ 与当前簇头 CH_k 到候选簇头 CH_{k+1} 的向量 $\boldsymbol{r}_i$ 之间的角度 ε 的余弦值，值越大，则表示第 i 个候选簇头 CH_i 的方向越靠近目标移动的方向；$\text{dis}(i, \text{CH}_k)^2 / R_{\text{C}}^2$ 表示当前簇头 CH_k 与候选簇头 CH_{k+1} 的距离越远，簇头交换的次数就越少，消耗能量越少；$(E_{i,k+1}^{\text{CH,init}} - E_{i,k+1}^{\text{CH}}) / E_{i,k+1}^{\text{CH,init}}$ 表示选择剩余能量较多的候选簇头节点作为下一时刻的簇头。其中，$E_{i,k+1}^{\text{CH,init}}$ 和 $E_{i,k+1}^{\text{CH}}$ 分别表

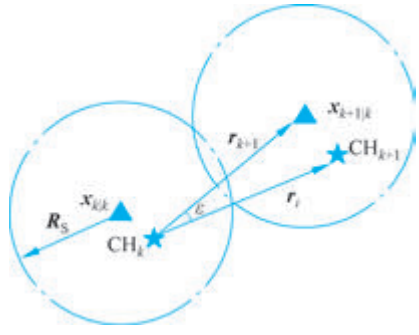


图 3.34 簇头选择的几何意义

示第 i 个候选簇头节点在时刻 $k+1$ 的初始能量和消耗能量。

水下节点的能量消耗主要是由传感器采集信息的能耗、CPU 控制和计算的能耗、水声通信的能耗等几部分构成[103]。因此，在 k 时刻第 i 个传感器节点的总能耗可以表示为

$$E_{i,k}=E_{i,k}^{\mathrm{com}}+E_{i,k}^{\mathrm{sens}}+E_{i,k}^{\mathrm{proc}} \tag{3-83}$$

但是，水下信道的特点决定了水下节点的绝大部分能量将用于水声通信，其他的能耗相对于水声通信的能耗相比均可以忽略不计。这里假设处于休眠状态的传感器节点的能耗为 0，节点状态转换的能耗忽略不计，即激活节点的能量消耗主要是决定于节点的通信能耗[104]。根据文献[95]中所提出的能耗模型，一个包含 N_{k+1} 个簇成员节点的簇头在动态簇的建立、数据传输以及簇头切换阶段需要消耗的能量分别表示为

$$E_{i,k+1}^{\mathrm{CH_build}}=E_{\mathrm{tx}}^{\mathrm{bro}}+E_{\mathrm{rx}}^{\mathrm{rep}}=l_{\mathrm{bro}}\left[e_{\mathrm{tx}}+P_{\mathrm{r}}A(R_{\mathrm{C}})\right]+N_{k+1}l_{\mathrm{rep}}e_{\mathrm{rx}} \tag{3-84}$$

$$E_{i,k+1}^{\mathrm{CH_trans}}=E_{\mathrm{tx}}^{\mathrm{fusion}}+E_{\mathrm{rx}}^{\mathrm{local}}=\sum_{i=1}^{N_{k+1}}\left[l_{\mathrm{fusion}}(e_{\mathrm{tx}}+P_{\mathrm{r}}A(d_i))+l_{\mathrm{local}}e_{\mathrm{rx}}\right] \tag{3-85}$$

$$E_{i,k+1}^{\mathrm{CH_swi}}=E_{\mathrm{tx}}^{\mathrm{swi}}+E_{\mathrm{rx}}^{\mathrm{swi}}=l_{\mathrm{swi}}\left[e_{\mathrm{tx}}+P_{\mathrm{r}}A(d_{\mathrm{hi}})+e_{\mathrm{rx}}\right] \tag{3-86}$$

其中，$E_{\mathrm{tx}}^{\mathrm{bro}}$ 表示簇头向各个簇成员发布数据包长度为 l_{bro} 的广播信息消耗的能量；$E_{\mathrm{rx}}^{\mathrm{rep}}$ 表示簇头接收 N_{k+1} 个簇成员回复的数据包长度为 l_{rep} 的信息消耗的能量；$E_{\mathrm{tx}}^{\mathrm{fusion}}$ 表示簇头向 N_{k+1} 个簇成员发布数据包长度为 l_{fusion} 的目标融合状态信息消耗的能量；$E_{\mathrm{rx}}^{\mathrm{local}}$ 表示簇头接收 N_{k+1} 个簇成员发送的数据包长度为 l_{local} 的本地目标状态估计信息消耗的能量；d_i 表示第 i 个候选簇头和目标预测位置之间的距离；$E_{\mathrm{tx}}^{\mathrm{swi}}$ 和 $E_{\mathrm{rx}}^{\mathrm{swi}}$ 分别表示簇头转换时发送和接收数据包长度为 l_{swi} 的信息消耗的能量；d_{hi} 表示第 i 个候选簇头和簇头CH_k 之间的距离；e_{tx} 和 e_{rx} 分别是发送和接收模块运行时需要的能量；P_{r} 为接收端所需的最低信号接收功率；$A(d)$为抵抗水声信道的能量衰减需要的增幅系数

$$A(d)=(d/1000)^{\tilde{\omega}}\bar{\alpha}^{d/1000} \tag{3-87}$$

其中，$\tilde{\omega}$ 是能量扩散因子，由声波的扩散方式决定(圆柱体扩散取 1，球体扩散取 2，实际情况取 1.5)；α 为吸收系数

$$\alpha=10^{\bar{a}/10} \tag{3-88}$$

$$\bar{a}=\frac{0.11f^2}{1+f^2}+\frac{44f^2}{4100+f^2}+2.75\times10^4f^2+0.003 \tag{3-89}$$

因此，簇头节点CH_{k+1} 在 $k+1$ 时刻消耗的总能量表示为

$$E_{i,k+1}^{\mathrm{CH}}=E_{i,k+1}^{\mathrm{CH_build}}+E_{i,k+1}^{\mathrm{CH_trans}}+E_{i,k+1}^{\mathrm{CH_swi}} \tag{3-90}$$

竞选簇头成功的超级节点将其簇首标志位 CH_Flag 置 1，并向周围的节点广播其成为簇头的消息。其他候选的簇头在接收到信号时，将不再参与簇头的竞选，将其簇首标志位 CH_Flag 置 0。候选簇成员在接收到消息后，记录簇头节点的信息。

通常，目标跟踪的最终目标是在实际资源有限的条件下，尽可能地提高对目标的跟踪精度。毫无疑问，当能够检测到目标的候选节点集合 S_{k+1} 中的所有节点都参与目标跟踪过程时，将获得最高的目标跟踪精度。但是，这会造成系统能量上的巨大浪费，并不适用于能量资源受限的 UWSN。因此，如何在牺牲较小的跟踪精度的前提下尽量降低网络能耗，进而延长整个网络的生命周期是本节研究的主要目的。为解决这个问题，在本节中，在每一时刻从所有激活的簇成员候选集合 S_{k+1} 中选择一个子集 S_{k+1}^{*} 来参与到目标的跟踪过程，该方法通过减少节点的数目来大幅度降低网络能耗，进而实现能量消耗和跟踪精度之间的有效权衡。

簇成员节点的选择主要考虑 3 个要素：

(1) 候选节点能提供的信息量；

(2) 候选节点进行网络通信的能量消耗;

(3) 候选节点的剩余能量。

鉴于以上 3 个要素,本节将通过效用函数和成本函数来构建最终的目标函数,完成簇成员节点的选择。效用函数被定义为后续测量所减少的目标不确定度,用来评价信息的有效性。成本函数通过最小化能量消耗和最大化剩余能量来定义。

与前几节类似,本节也采用 FIM 来设计效用函数,并作为簇成员节点选择的信息度量。该信息度量可直接利用簇头CH_k对目标在$k+1$时刻的预测目标状态$\hat{\boldsymbol{x}}_{k+1|k}$以及候选簇成员集合$S_{k+1}$中各个节点的位置得到。因此,在进行簇成员节点选择时,簇头CH_k不需要预先获得$k+1$时刻各个簇成员节点的实际观测值,即可以完成簇成员节点的选择。

由 2.2.7 节可知,随机参数的估计误差协方差矩阵的最小可达下界由 CRLB 给定,即满足

$$\boldsymbol{R}:=E\left[(\hat{\boldsymbol{x}}-\boldsymbol{x})(\hat{\boldsymbol{x}}-\boldsymbol{x})^{\mathrm{T}}\right]\geqslant\mathrm{CRLB}^{-1}=\boldsymbol{J}_{\mathrm{f}} \tag{3-91}$$

其中,$\boldsymbol{J}=\boldsymbol{J}^{\mathrm{d}}+\boldsymbol{J}^{\mathrm{p}}$ 表示 FIM;$\boldsymbol{J}^{\mathrm{d}}$ 表示量测 FIM;$\boldsymbol{J}^{\mathrm{p}}$ 表示先验 FIM。在高斯白噪声的条件下,根据 UKF 算法中的估计误差协方差更新公式,在$k+1$时刻$\boldsymbol{J}_{k+1}$和$\boldsymbol{J}_{k+1}^{\mathrm{p}}$可表示为

$$\begin{cases}\boldsymbol{J}_{k+1}=\boldsymbol{P}_{k+1|k+1}^{-1}\\ \boldsymbol{J}_{k+1}^{\mathrm{p}}=\boldsymbol{P}_{k+1|k}^{-1}\end{cases} \tag{3-92}$$

本节将利用 MLE 来估计$\boldsymbol{J}_{k+1}^{\mathrm{d}}$。根据目标的量测模型,在给定目标预测位置$(x,y,z)$的前提下,第$i$个节点在$k+1$时刻的测量值$\boldsymbol{z}_{i,k+1}$的条件概率密度函数表示为

$$P(\boldsymbol{z}_{i,k+1}\mid x,y,z)=\frac{1}{2\pi\delta_{i,k+1}^{\theta}\delta_{i,k+1}^{\phi}}\exp\left\{-\left(\frac{(\tilde{\theta}_{i,k+1}-\theta_{i,k+1})^2}{2(\delta_{i,k+1}^{\theta})^2}+\frac{(\tilde{\phi}_{i,k+1}-\phi_{i,k+1})^2}{2(\delta_{i,k+1}^{\phi})^2}\right)\right\} \tag{3-93}$$

由于本节假设所有节点具有相同的测量噪声统计特性,即$(\delta_{i,k+1}^{\theta})^2=(\delta_{i,k+1}^{\phi})^2=\delta_{i,k+1}^2=\delta_{k+1}^2$。因此,式(3-93)可简化为

$$P(\boldsymbol{z}_{i,k+1}\mid x,y,z)=\frac{1}{2\pi\delta_{k+1}^2}\exp\left\{-\frac{1}{2\delta_{k+1}^2}\left[(\tilde{\theta}_{i,k+1}-\theta_{i,k+1})^2+(\tilde{\phi}_{i,k+1}-\phi_{i,k+1})^2\right]\right\} \tag{3-94}$$

假设在$k+1$时刻从候选集合S_{k+1}中选择N_{k+1}个簇成员节点参与目标的跟踪过程,则测量值集合$\boldsymbol{z}_{k+1}$的联合条件概率密度函数$p(\boldsymbol{z}_{k+1}\mid x,y,z)$表示为

$$\begin{aligned}p(\boldsymbol{z}_{k+1}\mid x,y,z)&=\prod_{i=1}^{N_{k+1}}p(\boldsymbol{z}_{i,k+1}\mid x,y,z)\\&=\frac{1}{(2\pi\delta_{k+1}^2)^{N_{k+1}}}\exp\left\{-\frac{1}{2\delta_{k+1}^2}\sum_{i=1}^{N_{k+1}}\left[(\tilde{\theta}_{i,k+1}-\theta_{i,k+1})^2+(\tilde{\phi}_{i,k+1}-\phi_{i,k+1})^2\right]\right\}\end{aligned} \tag{3-95}$$

基于 MLE 的目标定位就是求目标的位置估计值$(\hat{x}_{k+1},\hat{y}_{k+1},\hat{z}_{k+1})$,使联合概率密度函数$p(\boldsymbol{z}_{k+1}\mid x,y,z)$最大,即

$$(\hat{x}_{k+1},\hat{y}_{k+1},\hat{z}_{k+1})=\arg\max_{(x,y,z)}p(\boldsymbol{z}_{k+1}\mid x,y,z) \tag{3-96}$$

对式(3-95)两边取对数,得其对数似然函数为

$$\begin{aligned}\ln p(\boldsymbol{z}_{k+1}\mid x,y,z)&=\sum_{i=1}^{N_{k+1}}\ln p(\boldsymbol{z}_{i,k+1}\mid x,y,z)\\&=-N_{k+1}\left[\ln(2\pi)-2\ln(\delta_{k+1}^2)\right]-\frac{1}{2\delta_{k+1}^2}\sum_{i=1}^{N_{k+1}}\left[(\bar{\theta}_{i,k+1}-\theta_{i,k+1})^2+(\bar{\phi}_{i,k+1}-\phi_{i,k+1})^2\right]\end{aligned} \tag{3-97}$$

则 MLE 的 CRLB 可以用如下的 FIM 的逆来表示

$$\bar{\boldsymbol{J}}_{k+1}=-E\begin{bmatrix}\dfrac{\partial^2\ln p(\boldsymbol{z}_{k+1}\mid x,y,z)}{\partial x^2} & \dfrac{\partial^2\ln p(\boldsymbol{z}_{k+1}\mid x,y,z)}{\partial x\partial y} & \dfrac{\partial^2\ln p(\boldsymbol{z}_{k+1}\mid x,y,z)}{\partial x\partial z}\\ \dfrac{\partial^2\ln p(\boldsymbol{z}_{k+1}\mid x,y,z)}{\partial y\partial x} & \dfrac{\partial^2\ln p(\boldsymbol{z}_{k+1}\mid x,y,z)}{\partial y^2} & \dfrac{\partial^2\ln p(\boldsymbol{z}_{k+1}\mid x,y,z)}{\partial y\partial z}\\ \dfrac{\partial^2\ln p(\boldsymbol{z}_{k+1}\mid x,y,z)}{\partial z\partial x} & \dfrac{\partial^2\ln p(\boldsymbol{z}_{k+1}\mid x,y,z)}{\partial z\partial y} & \dfrac{\partial^2\ln p(\boldsymbol{z}_{k+1}\mid x,y,z)}{\partial z^2}\end{bmatrix} \tag{3-98}$$

将式(3-98)进一步简化表示为

$$\boldsymbol{J}_{k+1}=\sum_{i=1}^{N_{k+1}}\frac{1}{2\delta_{k+1}^2}\begin{bmatrix}J_{11} & J_{12} & J_{13}\\ J_{21} & J_{22} & J_{23}\\ J_{31} & J_{32} & J_{33}\end{bmatrix} \tag{3-99}$$

其中，

$$J_{11}=\frac{(y-y_i)^2}{[(x-x_i)^2+(y-y_i)^2]^2}+\frac{(x-x_i)^2(z-z_i)^2}{[(x-x_i)^2+(y-y_i)^2+(z-z_i)^2]^2[(x-x_i)^2+(y-y_i)^2]}$$

$$J_{22}=\frac{(x-x_i)^2}{[(x-x_i)^2+(y-y_i)^2]^2}+\frac{(y-y_i)^2(z-z_i)^2}{[(x-x_i)^2+(y-y_i)^2+(z-z_i)^2]^2[(x-x_i)^2+(y-y_i)^2]}$$

$$J_{33}=\frac{(x-x_i)^2+(y-y_i)^2}{(z-z_i)^2[(x-x_i)^2+(y-y_i)^2+(z-z_i)^2]}$$

$$J_{12}=J_{21}=\frac{(y_i-y)(x-x_i)}{[(x-x_i)^2+(y-y_i)^2]^2}+\frac{(x-x_i)(y-y_i)(z-z_i)^2}{[(x-x_i)^2+(y-y_i)^2+(z-z_i)^2]^2[(x-x_i)^2+(y-y_i)^2]}$$

$$J_{13}=J_{31}=\frac{(x_i-x)(z-z_i)}{[(x-x_i)^2+(y-y_i)^2+(z-z_i)^2]^2}$$

$$J_{23}=J_{32}=\frac{(y_i-y)(z-z_i)}{[(x-x_i)^2+(y-y_i)^2+(z-z_i)^2]^2}$$

而 δ_{k+1}^2 的 CRLB 的计算可以参考文献[105]。

可以看出，$\bar{\boldsymbol{J}}_{k+1}$ 的值仅与目标的预测位置和集合 S_{k+1} 中的候选簇成员节点的位置有关。因此，不需要预先获得集合 $\bar{\boldsymbol{J}}_{k+1}$ 中每个簇成员节点的实际测量值便可完成 $\bar{\boldsymbol{J}}_{k+1}$ 的计算，实现簇成员节点的选择。因此，$\boldsymbol{J}_{k+1}$ 可以重写为

$$\boldsymbol{J}_{k+1}=\boldsymbol{J}_{k+1}^{\mathrm{p}}+\begin{bmatrix}\bar{\boldsymbol{J}}_{k+1} & \boldsymbol{0}_{6\times6}\\ \boldsymbol{0}_{6\times6} & \boldsymbol{0}_{6\times6}\end{bmatrix} \tag{3-100}$$

基于 MLE 的目标位置估计的方差与 $\boldsymbol{J}_{k+1}$ 的关系满足

$$\mathrm{Var}[\hat{x}_{k+1}]+\mathrm{Var}[\hat{y}_{k+1}]+\mathrm{Var}[\hat{z}_{k+1}]\geqslant\frac{\mathrm{tr}(\tilde{\boldsymbol{J}}_{k+1})}{\det(\tilde{\boldsymbol{J}}_{k+1})} \tag{3-101}$$

其中，$\tilde{\boldsymbol{J}}_{k+1}=[\boldsymbol{I}_3\quad\boldsymbol{0}_{3\times6}]\langle\boldsymbol{I}_9,\boldsymbol{J}_{k+1}\rangle[\boldsymbol{I}_3\quad\boldsymbol{0}_{3\times6}]^{\mathrm{T}}$。称 $\dfrac{\mathrm{tr}(\tilde{\boldsymbol{J}}_{k+1})}{\det(\tilde{\boldsymbol{J}}_{k+1})}$ 为精度几何因子(Geometric Dilution of

Precision，GDOP)指标。

在本节中，由于假设所有传感器节点具有相同的测量噪声统计特性，利用 $\widetilde{\boldsymbol{J}}_{k+1}$ 构造簇成员节点选择的效用函数

$$\Psi_{\text{utility}}(S_{k+1}^{*})=\det(\widetilde{\boldsymbol{J}}_{k+1}^{*}),\quad (S_{k+1}^{*}\subseteq S_{k+1},|S_{k+1}^{*}|=N_{k+1}) \tag{3-102}$$

其中，S_{k+1}^{*} 为拥有 N_{k+1} 簇成员构成的候选集合 S_{k+1} 的子集；$\widetilde{\boldsymbol{J}}_{k+1}^{*}$ 表示根据集合 S_{k+1}^{*} 中节点的位置和目标预测位置计算得到的 FIM。

与簇头节点选择的能量消耗模型类似，在 k 时刻，第 j 个簇成员节点在动态簇的建立和数据传输阶段需要消耗的能量表示为

$$\begin{cases}E_{j,k+1}^{\text{MEM_build}}=E_{\text{rx}}^{\text{bro}}+E_{\text{tx}}^{\text{rep}}=l_{\text{bro}}e_{\text{rx}}+l_{\text{rep}}[e_{\text{tx}}+P_{\text{r}}A(d_j)]\\ E_{j,k+1}^{\text{MEM_trans}}=E_{\text{rx}}^{\text{local}}+E_{\text{tx}}^{\text{fusion}}=l_{\text{local}}[e_{\text{tx}}+P_{\text{r}}A(d_j)]+l_{\text{fusion}}e_{\text{rx}}\end{cases} \tag{3-103}$$

其中，d_j 表示簇头CH_{k+1} 和第 j 个簇成员之间的距离。$E_{\text{rx}}^{\text{bro}}$ 表示簇成员接收长度为 l_{bro} 的簇头发布的广播信息消耗的能量；$E_{\text{tx}}^{\text{rep}}$ 表示簇成员发送长度为 l_{rep} 的回复信息消耗的能量；$E_{\text{rx}}^{\text{local}}$ 表示簇成员向簇头发送长度为 l_{local} 的本地目标状态估计消耗的能量；$E_{\text{tx}}^{\text{fusion}}$ 表示簇成员接收簇头发送的长度为 l_{fusion} 的融合目标状态信息消耗的能量。因此，第 j 个簇成员消耗的总能量表示为

$$E_{j,k+1}^{\text{MEM}}=E_{j,k+1}^{\text{MEM_build}}+E_{j,k+1}^{\text{MEM_cost}} \tag{3-104}$$

因此，在 $k+1$ 时刻，簇成员的成本函数可以表示为

$$\Psi_{\text{cost}}(S_{k+1}^{*})=\sum_{j=1}^{N_{k+1}}\frac{E_{j,k+1}^{\text{MEM,init}}-E_{j,k+1}^{\text{MEM}}}{E_{j,k+1}^{\text{MEM,init}}},\quad (S_{k+1}^{*}\subseteq S_{k+1},|S_{k+1}^{*}|=N_{k+1}) \tag{3-105}$$

其中，$E_{j,k+1}^{\text{MEM,init}}$ 是第 j 个候选簇成员在 $k+1$ 时刻拥有的初始能量。

跟踪精度和网络能耗是目标跟踪过程中关注的两个主要的性能指标，将成本函数和效用函数结合作为选择簇成员节点最终的目标函数。

$$\Psi(S_{k+1}^{*})=\lambda\Psi_{\text{utility}}(S_{k+1}^{*})+(1-\lambda)\Psi_{\text{cost}}(S_{k+1}^{*}),(S_{k+1}^{*}\subseteq S_{k+1},|S_{k+1}^{*}|=N_{k+1}) \tag{3-106}$$

其中，第一部分的效用函数 $\Psi_{\text{utility}}(S_{k+1}^{*})$ 决定了系统的目标跟踪精度，主要取决于目标预测位置与候选簇成员节点之间的相对距离以及簇成员节点的拓扑结构；而第二部分的成本函数 $\Psi_{\text{cost}}(S_{k+1}^{*})$ 则决定了系统的能量消耗，主要取决于簇头CH_{k+1} 与候选簇成员节点之间的相对距离。图 3.35 清楚地描述了上面两部分的关系。这两者之间是相互矛盾的，要想降低系统的能耗就要以牺牲一定的跟踪精度为代价。λ 表示用于调整两部分比例的联合因子。在跟踪过程中，应该根据实际场景选择一个合适的 λ 来调整两部分的相对比例。当需要获得较高的跟踪精度时，应该选择一个较大的 λ，反之则选择较小的 λ。

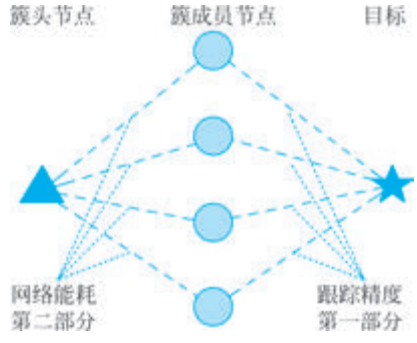


图 3.35 簇成员节点选择的几何意义

簇成员节点的选择对于系统的跟踪精度和网络能耗都具有决定性的作用。簇成员的选择就是从候选集 S_{k+1} 中选择由 N_{k+1} 个节点组成的子集 S_{k+1}^{*} 来使目标函数 $\Psi(S_{k+1}^{*})$ 最大化。因此，簇成员节点的自适应选择问题可以表述如下：

$$S_{k+1}^{*}=\arg\max\Psi(S_{k+1}^{*})$$

$$\text{s.t. } S_{k+1}^{*} \subseteq S_{k+1}, |S_{k+1}^{*}| = N_{k+1} \tag{3-107}$$

3.3.3 节将给出优化问题(3-107)的求解思路和跟踪方案。

3.3.3 基于动态分簇的水下目标跟踪算法

本节提出一种基于背包问题的簇成员节点的自适应选择策略，该方法将节点的选择问题归类为数学中的背包问题[106]，并利用动态规划的方法来选择一个最优的节点组合实现目标函数的最大化。

假设候选簇成员集合 S_{k+1} 中的节点表示物品，节点的数目 N_S 表示物品的种类，参与跟踪的簇成员节点的数目 N_{k+1} 表示背包的容量，N_{k+1} 的值可以在$[N_{\min}, N_S]$内变化，这里设置初始选择节点的最低数目 $N_{\min}=2$。假设每种物品的重量 $w_i=1(1\leqslant i\leqslant N_S)$，则对应目标函数的取值表示所选物品的总价值。通过设定跟踪误差阈值 Ψ_0，根据融合预测协方差 $\boldsymbol{P}_{k+1|k}$ 的迹可以自适应地调整背包问题中的背包容量 N_{k+1}。

(1) 当 $\mathrm{tr}(\boldsymbol{P}_{k+1|k})\leqslant\Psi_0$ 时，表示跟踪效果已经达到了预设的要求，可以在保证跟踪效果满足要求的前提下，适当减少选择的簇成员节点的数目；

(2) 当 $\mathrm{tr}(\boldsymbol{P}_{k+1|k})>\Psi_0$ 时，表示跟踪误差已经超过了预设的要求，要适当增加选择的簇成员节点的数目，以改善目标跟踪的效果。

算法首先构造一个二维数组 V 为$(N_S+1)\times(N_{k+1}+1)$的，数组元素 $V(i,j)$表示将前面 i 个物体装入载重量为 j 的背包所取得的最优价值，其中，$0\leqslant i\leqslant N_S$，$0\leqslant j\leqslant N_{k+1}$。每一个元素都对应一个二进制序列 $\boldsymbol{N}(:,i,j)=[n_1,n_2,\cdots,n_{N_S}]^{\mathrm{T}}$ 和一个集合 $P_{i,j}$，$P_{i,j}$ 初始为空集，用于保存决策序列 $\boldsymbol{N}(:,i,j)$中选择的簇成员节点。决策序列 $\boldsymbol{N}(:,i,j)$表示与 $V(i,j)$相对应的 N_S 个候选簇成员节点的状态，并满足

$$\boldsymbol{N}(m,i,j)=\begin{cases}1, & \text{选择第 } m \text{ 个节点}\\ 0, & \text{其他}\end{cases} \tag{3-108}$$

按照如下方式初始化数组 V 和二进制序列

$$\begin{cases} V(i,0)=V(0,j)=0,(0\leqslant i\leqslant N_S,0\leqslant j\leqslant N_{k+1})\\ \boldsymbol{N}(:,i,0)=\boldsymbol{N}(:,i,0)=[0,0,\cdots,0]_{N_S\times 1}^{\mathrm{T}}\end{cases} \tag{3-109}$$

即将 0 个物体装入载重量为 j 的背包中，或者将 i 个物体装入载重量为 0 的背包中，得到的价值均为 0。

综上，簇成员节点的选择问题可以归结为寻找一个集合 $P_{i,j}$ 使目标函数 $V(i,j)$达到最大的问题，即

$$\begin{cases} P_{i,j}=\arg\max V(i,j),\\ \text{s.t. } V(i,j)=\Psi(P_{i,j}),P_{i,j}\subseteq S_{k+1},|P_{i,j}|=j\end{cases} \tag{3-110}$$

算法 3.3 总结了自适应节点选择算法。

算法 3.3　自适应节点选择算法	
初始化	给定背包容量 N_{k+1}
输入	$\boldsymbol{P}_{k+1\|k}$，记 $\Phi=\mathrm{tr}(\boldsymbol{P}_{k+1\|k})$ 对于时刻 $k=1,2,\cdots$，循环进行以下步骤
步骤 1.	若 $\Phi>\Phi_0$，则增加选择节点的数目 $N_{k+1}=\min\{N_k+1,N_S\}$

续表

	若 $\Phi \leqslant \Phi_0$，$N_k > 2$，则减少选择节点的数目 $N_{k+1} = \min\{N_k - 1, N_S\}$ 若 $\Phi \leqslant \Phi_0$，$N_k \leqslant 2$，则保持最低节点的数目 $N_{k+1} = \min\{N_{\min}, N_S\}$
步骤 2.	求解最优节点组合。 对于 $i=1,2,\cdots,N_S$，$j=1,2,\cdots,N_{k+1}$，循环进行以下步骤。 令 $\Psi_1 = V(i-1,j)$，$\Psi_2 = \Psi(P_{i-1,j-w_i} \cup S_{k+1}(i))$ 若 $j < w_i$ 或 $\Psi_1 \geqslant \Psi_2$，则不选择第 i 个候选簇成员节点，且 $V(i,j) = \Psi_1$，$\boldsymbol{N}(:,i,j) = \boldsymbol{N}(:,i-1,j)$，$P_{i,j} = P_{i-1,j}$ 否则选择第 i 个候选簇成员节点，且 $V(i,j) = \Psi_2$，$P_{i,j} = P_{i-1,j-w_i} \cup S_{k+1}(i)$ $\boldsymbol{N}(:,i,j) = \boldsymbol{N}(:,i-1,j-w_i) + \boldsymbol{l}$ 其中，$\boldsymbol{l} = [0,\cdots,0,1,0,\cdots,0]$，$\boldsymbol{l}(i) = 1$。 当 $(i,j) = (N_S, N_{k+1})$ 时，对应的 $P_{i,j}$ 为最优节点组合。
输出	$P_{i,j}$

成功竞选为簇成员的普通节点将其成员标志位 MEM_Flag 置 1，然后向簇头节点回复一个信息（包含自身位置、剩余能量等），告诉簇头其是簇内成员。综上，在每一个时刻，选择的所有簇成员和簇头将动态组成一个跟踪簇，共同参与目标的跟踪过程。动态成簇的基本流程如图 3.36 所示。

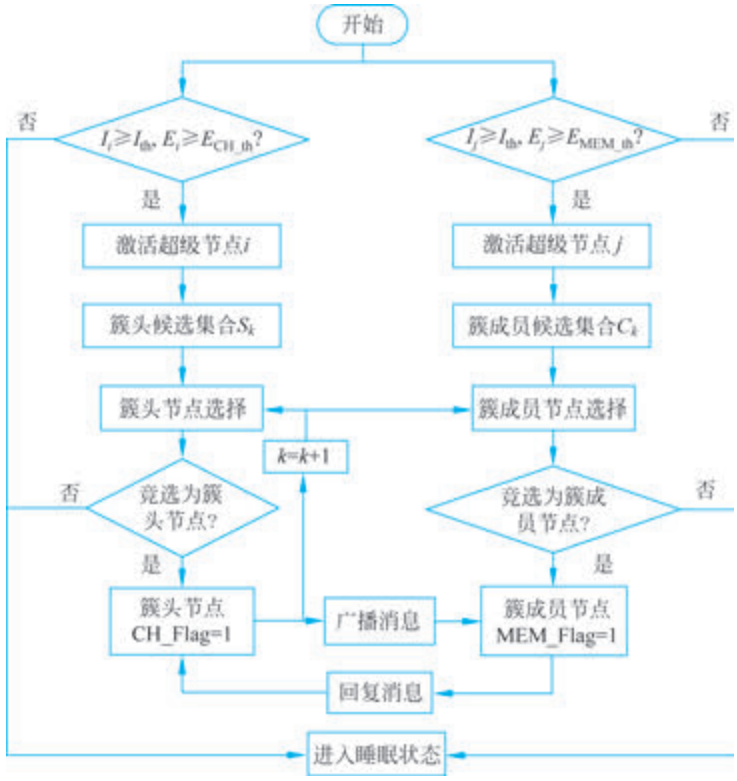


图 3.36 动态成簇流程图

融合结构主要分为集中式融合结构和分布式融合结构。在集中式融合结构中，节点将获取的原始量测数据完整上传到融合中心，融合中心再根据各个节点的原始量测信息估计目标状态。分布式融合

结构又分为完全分布式融合与部分分布式融合。在部分分布式融合结构中，每一个节点会根据本地测量信息经滤波后得到目标状态的本地估计，然后再将目标状态的本地估计上传到融合中心，最后，融合中心再融合各个节点的本地估计，最终得到目标的状态估计。完全分布式融合则认为每个节点均为融合中心，通过节点邻居间的数据交换得到一系列本地化的全局估计。为降低在数据传输过程中的能量消耗，本节选择部分分布式融合结构。

假设在 k 时刻，从集合 S_k 中选择 N_k 个簇成员节点参与目标的跟踪过程。第 i 个簇成员节点利用 UKF 滤波算法对目标状态的本地估计结果 $\hat{\boldsymbol{x}}_{i,k|k}$ 表示为

$$\hat{\boldsymbol{x}}_{i,k|k}=[\hat{\boldsymbol{x}}_{i,k|k}^{(1)},\hat{\boldsymbol{x}}_{i,k|k}^{(2)},\cdots,\hat{\boldsymbol{x}}_{i,k|k}^{(n)}]^{\mathrm{T}} \tag{3-111}$$

其中，$i=1,2,\cdots,N_k$；$\hat{\boldsymbol{x}}_{i,k|k}^{(j)}$ 表示 $\hat{\boldsymbol{x}}_{i,k|k}$ 的第 j 个元素。$\hat{\boldsymbol{x}}_{i,k|k}$ 是无偏的，且满足 $E[\hat{\boldsymbol{x}}_{i,k|k}]=E[\boldsymbol{x}_k]$，估计误差 $\hat{\boldsymbol{x}}_{i,k|k}=\boldsymbol{x}_k-\hat{\boldsymbol{x}}_{i,k|k}$，不同的节点之间相互独立，估计误差互不相关。因此，其对应的协方差矩阵 $\boldsymbol{P}_{i,k|k}$ 为对角矩阵，可以表示为

$$\boldsymbol{P}_{i,k|k}=\mathrm{diag}\{\boldsymbol{P}_{i,k|k}^{(1)},\boldsymbol{P}_{i,k|k}^{(2)},\cdots,\boldsymbol{P}_{i,k|k}^{(n)}\} \tag{3-112}$$

基于 N_k 个簇成员节点对目标状态的估计 $\hat{\boldsymbol{x}}_{i,k|k}$，簇头 CH_k 采用分量按标量加权的线性最小方差融合准则求得目标的最优融合估计值 $\hat{\boldsymbol{x}}_{k|k}=\left[\hat{\boldsymbol{x}}_{k|k}^{(1)},\hat{\boldsymbol{x}}_{k|k}^{(2)},\cdots,\hat{\boldsymbol{x}}_{k|k}^{(n)}\right]^{\mathrm{T}}$，并求得 $N_k\times n$ 的权矩阵。

$$\boldsymbol{A}=\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N_k,1} & a_{N_k,2} & \cdots & a_{N_k,n} \end{bmatrix} \tag{3-113}$$

使得 $\hat{\boldsymbol{x}}_{k|k}$ 的第 j 个分量的融合估计

$$\hat{\boldsymbol{x}}_{k|k}^{(j)}=a_{1,j}\hat{\boldsymbol{x}}_{1,k|k}^{(j)}+a_{2,j}\hat{\boldsymbol{x}}_{2,k|k}^{(j)}+\cdots+a_{N_k,j}\hat{\boldsymbol{x}}_{N_k,k|k}^{(j)} \tag{3-114}$$

是无偏的，即满足 $E[\hat{\boldsymbol{x}}_{k|k}^{(j)}]=E[\boldsymbol{x}_k^{(j)}]$，且极小化估计误差方差阵的迹 $\mathrm{tr}(\boldsymbol{P}_{k|k}^{(j)})$。进而将融合问题转变为求 n 个按标量加权的线性最小方差融合估计的问题。

对式(3-114)两边取数学期望有

$$\begin{aligned} E[\hat{\boldsymbol{x}}_{k|k}^{(j)}]&=E[a_{1,j}\hat{\boldsymbol{x}}_{1,k|k}^{(j)}+a_{2,j}\hat{\boldsymbol{x}}_{2,k|k}^{(j)}+\cdots+a_{N_k,j}\hat{\boldsymbol{x}}_{N_k,k|k}^{(j)}] \\ &=a_{1,j}E[\hat{\boldsymbol{x}}_{1,k|k}^{(j)}]+a_{2,j}E[\hat{\boldsymbol{x}}_{2,k|k}^{(j)}]+\cdots+a_{N_k,j}E[\hat{\boldsymbol{x}}_{N_k,k|k}^{(j)}] \\ &=(a_{1,j}+a_{2,j}+\cdots+a_{N_k,j})E[\hat{\boldsymbol{x}}_k^{(j)}] \end{aligned} \tag{3-115}$$

因此

$$a_{1,j}+a_{2,j}+\cdots+a_{N_k,j}=1 \tag{3-116}$$

进一步由式(3-114)可得融合估计误差为

$$\hat{\boldsymbol{x}}_k^{(j)}=\boldsymbol{x}_k^{(j)}-\hat{\boldsymbol{x}}_{k|k}^{(j)}=\sum_{i=1}^{N_k}a_{i,j}[\boldsymbol{x}_k^{(j)}-\hat{\boldsymbol{x}}_{i,k|k}^{(j)}]=\sum_{i=1}^{N_k}a_{i,j}\hat{\boldsymbol{x}}_{i,k}^{(j)} \tag{3-117}$$

因此，融合估计的误差协方差矩阵为

$$\begin{aligned} \boldsymbol{P}_{k|k}^{(j)}&=E[(\boldsymbol{x}_k^{(j)}-\hat{\boldsymbol{x}}_{k|k}^{(j)})(\boldsymbol{x}_k^{(j)}-\hat{\boldsymbol{x}}_{k|k}^{(j)})^{\mathrm{T}}] \\ &=E[\tilde{\boldsymbol{x}}_k^{(j)}(\tilde{\boldsymbol{x}}_k^{(j)})^{\mathrm{T}}]=\sum_{i=1}^{N_k}a_{i,j}\tilde{\boldsymbol{P}}_{i,k|k}^{(j)}a_{i,j} \end{aligned} \tag{3-118}$$

其中，$\tilde{\boldsymbol{P}}_{k|k}^{(j)}=\mathrm{diag}\{\boldsymbol{P}_{1,k|k}^{(j)},\boldsymbol{P}_{2,k|k}^{(j)},\cdots,\boldsymbol{P}_{N_k,k|k}^{(j)}\}$。因此，性能指标 $J=\mathrm{tr}(\boldsymbol{P}_{k|k}^{(j)})$ 可转化为

$$J=\mathrm{tr}(\boldsymbol{P}_{k|k}^{(j)})=\boldsymbol{a}_j^{\mathrm{T}}\tilde{\boldsymbol{P}}_{k|k}^{(j)}\boldsymbol{a}_j \tag{3-119}$$

其中，$\boldsymbol{a}_j=[a_{1,j},a_{2,j},\cdots,a_{N_k,j}]^{\mathrm{T}}$，则约束条件(3-116)可重写为

$$\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{e}=1,\quad \boldsymbol{e}=[1,1,\cdots,1]^{\mathrm{T}} \tag{3-120}$$

问题归结为在约束条件(3-116)下求权重系数 $\boldsymbol{a}_j(j=1,2,\cdots,n)$来极小化性能指标(3-119)。为解决这个问题，利用拉格朗日乘子法，引入辅助函数

$$F=J+\lambda(\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{e}-1) \tag{3-121}$$

令$\partial F/\partial \boldsymbol{a}_j\big|_{\boldsymbol{a}_j=\boldsymbol{a}_j^*}$，则有

$$\widetilde{\boldsymbol{P}}_{k|k}^{(j)}\boldsymbol{a}_j^*+\frac{1}{2}\lambda\boldsymbol{e}=\boldsymbol{0} \tag{3-122}$$

令 $\eta=\dfrac{1}{2}\lambda$，由式(3-120)和式(3-122)可得矩阵方程

$$\begin{bmatrix}\widetilde{\boldsymbol{P}}_{k|k}^{(j)} & \boldsymbol{e}\\ \boldsymbol{e}^{\mathrm{T}} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{a}_j^*\\ \eta\end{bmatrix}=\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{1}\end{bmatrix} \tag{3-123}$$

根据分块矩阵求逆公式，则有

$$\begin{bmatrix}\boldsymbol{a}_j^*\\ \eta\end{bmatrix}=\begin{bmatrix}\widetilde{\boldsymbol{P}}_{k|k}^{(j)} & \boldsymbol{e}\\ \boldsymbol{e}^{\mathrm{T}} & \boldsymbol{0}\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{1}\end{bmatrix}=\begin{bmatrix}\dfrac{(\widetilde{\boldsymbol{P}}_{k|k}^{(j)})^{-1}\boldsymbol{e}}{\boldsymbol{e}^{\mathrm{T}}(\widetilde{\boldsymbol{P}}_{k|k}^{(j)})^{-1}\boldsymbol{e}}\\ -\dfrac{1}{\boldsymbol{e}^{\mathrm{T}}(\widetilde{\boldsymbol{P}}_{k|k}^{(j)})^{-1}\boldsymbol{e}}\end{bmatrix} \tag{3-124}$$

因此

$$\boldsymbol{a}_j^*=\frac{(\widetilde{\boldsymbol{P}}_{k|k}^{(j)})^{-1}\boldsymbol{e}}{\boldsymbol{e}^{\mathrm{T}}(\widetilde{\boldsymbol{P}}_{k|k}^{(j)})^{-1}\boldsymbol{e}} \tag{3-125}$$

根据式(3-118)，可求得第 j 个元素的最优融合误差方差阵为

$$\boldsymbol{P}_{k|k}^{(j)}=\sum_{i=1}^{N_k}a_{i,j}^*\widetilde{\boldsymbol{P}}_{i,k|k}^{(j)}\boldsymbol{a}_{i,j}^* \tag{3-126}$$

根据式(3-126)，令 $a_{i,j}^*=1$，其中，$i=1,2,\cdots,N_k$，有 $\boldsymbol{P}_{k|k}^{(j)}=\widetilde{\boldsymbol{P}}_{k|k}^{(j)}$。由式(3-116)可知 $a_{i,j}^*\leqslant 1$，故 $\boldsymbol{P}_{k|k}^{(j)}\leqslant\widetilde{\boldsymbol{P}}_{k|k}^{(j)}$。记 $\widetilde{\boldsymbol{P}}_{k|k}=\mathrm{diag}\{\widetilde{\boldsymbol{P}}_{k|k}^{(1)},\widetilde{\boldsymbol{P}}_{k|k}^{(2)},\cdots,\widetilde{\boldsymbol{P}}_{k|k}^{(n)}\}$，$\boldsymbol{P}_{k|k}=\mathrm{diag}\{\boldsymbol{P}_{k|k}^{(1)},\boldsymbol{P}_{k|k}^{(2)},\cdots,\boldsymbol{P}_{k|k}^{(n)}\}$，可得 $\boldsymbol{P}_{k|k}\leqslant\widetilde{\boldsymbol{P}}_{k|k}$。

综上，经过分量按标量加权的线性最小方差信息融合算法求得目标的最优融合估计值的误差方差阵小于局部估计值的误差方差阵，因此，融合算法能有效提高目标状态的估计精度。将簇头节点的选择策略、簇成员节点的选择策略以及带反馈的分布式融合算法相结合便是本节提出的纯方位信息下基于动态分簇的水下目标跟踪算法。算法 3.4 给出了基于动态分簇的水下目标跟踪算法流程。

算法 3.4　基于动态分簇的水下目标跟踪算法	
初始化	给定 $\hat{\boldsymbol{x}}_{0\mid-1}$，$\boldsymbol{P}_{0\mid-1}$，$\mathrm{CH}_0$，$\boldsymbol{S}_0^*$
输入	$\hat{\boldsymbol{x}}_{k\mid k-1}$，$\boldsymbol{P}_{k\mid k-1}$，$\mathrm{CH}_k$，$\boldsymbol{S}_k^*$，$\boldsymbol{z}_k$ 对于时刻 $k=1,2,3,\cdots$，循环进行以下步骤
步骤 1.	本地状态估计。对于 $i=1,2,\cdots,N_k$，循环进行以下步骤 第 i 个簇成员节点获得量测信息 采用 UKF 滤波算法计算目标本地状态估计 $\hat{\boldsymbol{x}}_{i,k\mid k}$ 和协方差矩阵 $\boldsymbol{P}_{i,k\mid k}$

续表

	将目标本地状态估计 $\hat{\boldsymbol{x}}_{i,k\mid k}$ 和协方差矩阵 $\boldsymbol{P}_{i,k\mid k}$ 传输给簇头节点CH_k
步骤 2.	簇头节点CH_k 执行融合算法。对于 $j=1,2,\cdots,n$，循环进行以下步骤
	计算目标状态融合估计 $\hat{\boldsymbol{x}}_{k\mid k}^{(j)}$ 及其协方差矩阵 $\boldsymbol{P}_{k\mid k}^{(j)}$
步骤 3.	簇头节点CH_k 通过下式计算目标状态估计的一步预测 $\hat{\boldsymbol{x}}_{k+1\mid k}$ 及其协方差矩阵 $\boldsymbol{P}_{k+1\mid k}$
	$\hat{\boldsymbol{x}}_{k+1\mid k}=\boldsymbol{F}_k\hat{\boldsymbol{x}}_{k\mid k}$，$\boldsymbol{P}_{k+1\mid k}=\boldsymbol{F}_k\boldsymbol{P}_{k\mid k}\boldsymbol{F}_k^{\mathrm{T}}+\boldsymbol{Q}_k$
步骤 4.	更新簇头节点。根据簇头节点选择策略从簇头候选集合 C_{k+1} 中选择簇头节点CH_{k+1}，置簇首标志位 CH_Flag=1，并向周围节点广播其成为簇头节点的信息
步骤 5.	更新簇成员。确定需要选择的簇成员节点的个数 N_{k+1}，根据簇成员节点自适应选择算法求解最优的簇成员节点集合 S_{k+1}^{*}，簇成员节点标志位 MEM_Flag=1，并向簇头节点发送回复信息
输出	$\hat{\boldsymbol{x}}_{k+1\mid k}$，$\boldsymbol{P}_{k+1\mid k}$，$\mathrm{CH}_{k+1}$，$S_{k+1}^{*}$

3.3.4 仿真结果与分析

下面将对本节提出的基于纯方位的水下目标运动参数估计算法性能进行仿真验证和评估。本书假设 UWSN 所有的节点具有相同的感知范围和通信范围，UWSN 的主要参数设置如表 3.6 所示。

表 3.6 UWSN 的主要参数设置

参　数	取　值
检测区域	1000m×1000m×1000m
普通节点间的距离	200m
超级节点间的距离	300m
节点的感知距离 R_{S}	300m
节点的最大通信距离 R_{C}	500m
系统采样周期	1s
跟踪持续时间	100s

在仿真中，假设目标的真实初始状态为 $\boldsymbol{x}_0=[200,10,0,300,3,0,200,2,0]^{\mathrm{T}}$，目标状态的初始估计为 $\hat{\boldsymbol{x}}_0=[210,10,0,290,3,0,210,2,0]^{\mathrm{T}}$，相应的误差协方差为 $\hat{\boldsymbol{P}}_{0\mid 0}=\mathrm{diag}\{100,0,0,100,0,0,100,0,0\}$，所有的节点具有相同的测量噪声协方差表示为 $\boldsymbol{R}=\mathrm{diag}\{0.01,0.01\}$。为了更符合水下场景，目标在三维监测区域里移动并发生一定的机动，在 1～30s，目标做 CV 运动；在 31～60s，目标做转弯速率为 0.052rad/s 的 CT 运动；在 61～100s，目标做加速度为 $\boldsymbol{a}=[0.5,-0.2,0.1]^{\mathrm{T}}$ 的 CA 运动。初始簇头节点CH_0 的坐标为(500,500,500)，初始簇成员节点集合 S_0^{*} 的坐标为(250,250,250)和(750,750,750)。超级节点的初始能量在 10～100J 随机产生，普通节点的初始能量在 1～10J 随机产生。关于能耗模型、数据包以及过程噪声的其他参数设置如表 3.7 所示。本节的仿真结果均通过 500 次蒙特卡洛实验取平均值获得，跟踪误差均采用 RMSE 来进行评估。

表 3.7 能耗模型、数据包以及过程噪声的相关参数设置

参　数	取　值	参　数	取　值
n	512b	P_{r}	$50\mu\mathrm{J/b}\cdot(\mathrm{km})^2$
e_{tx}	2nJ/b	$\widetilde{\omega}$	2
e_{rx}	2nJ/b	α	20

续表

参　数	取　值	参　数	取　值
l_{bro}	64b	l_{local}	112b
l_{rep}	64b	l_{swi}	368b
l_{fusion}	112b	q^2	5

为了评估本节提出的水下目标跟踪算法整体的跟踪性能，首先，在预设跟踪误差阈值 $\Phi_0=5$ 和联合因子 $\lambda=0.6$ 的条件下对算法的跟踪性能进行仿真验证和分析，目标的真实轨迹和跟踪轨迹如图 3.37 所示。从图 3.37 可以看出，本节提出的跟踪算法能够成功实现对水下目标的精确跟踪。

为了验证本节所提出的融合算法的有效性，图 3.38 比较了基于单节点的 EKF 算法、基于单节点的 UKF 算法、基于矩阵加权的线性最小方差融合 UKF(MWMVarUKF)算法以及本节提出的基于分量按标量加权的线性最小方差融合 UKF(CWMVarUKF)算法的跟踪误差随时间的变化情况。从图 3.38 中的结果可以看出，4 种跟踪算法的跟踪误差都是收敛的，均能够实现对目标的跟踪。基于单节点的 EKF 算法的收敛速度最慢，跟踪误差最大，跟踪误差能够收敛至 2～4m；基于单节点的 UKF 算法的收敛速度次之，跟踪误差能够收敛至 2～4m；基于矩阵加权的线性最小方差融合 UKF 算法的收敛速度和跟踪精度都优于基于单节点的跟踪算法，其跟踪误差能够收敛至 2m 以内；基于分量按标量加权的最小方差融合 UKF 算法的收敛速度最快，跟踪误差能够收敛至 1m 以内，能够更快地实现对目标的精确跟踪。为了更加清晰地比较不同算法的跟踪性能，表 3.8 列出了几种目标跟踪算法详细的平均跟踪误差，从表 3.8 可以看出，本节提出的目标跟踪算法具有最高的跟踪精度，进一步验证了本节提出的融合算法的有效性。

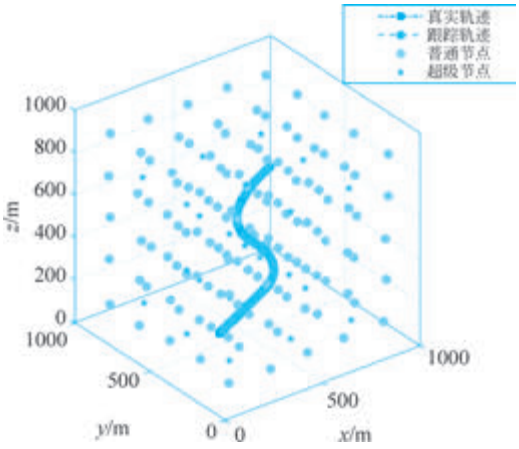


图 3.37　本节算法目标跟踪的效果

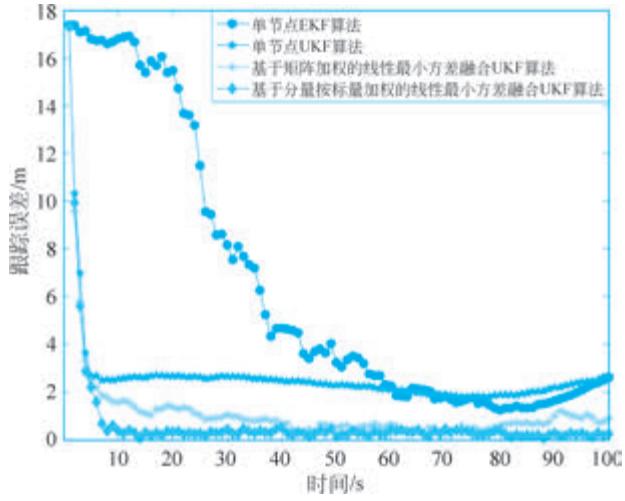


图 3.38　比较不同跟踪算法的跟踪误差

表 3.8　不同跟踪算法的跟踪误差

跟 踪 算 法	跟踪误差/m	跟 踪 算 法	跟踪误差/m
EKF	6.42	MWMVarUKF	1.15
UKF	2.61	CWMVarUKF	0.67

联合因子 λ 的值决定了系统将获得更高的跟踪精度还是消耗更少的能量。为了进一步揭示 λ 的值对本节所提跟踪算法的跟踪精度和能量消耗的影响，使 λ 的取值从 0～1 改变，每次取值间隔为 0.2，图 3.39 比较了在不同的 λ 值下本节所提跟踪算法的平均跟踪误差。从图中可以看出，随着 λ 值的增大，算法的跟踪精度大大减小。

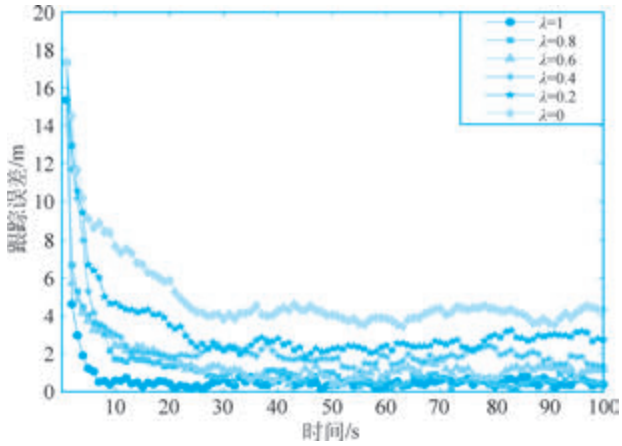


图 3.39 不同 λ 下算法的跟踪误差

为了进一步的定量分析和比较，表 3.9 详细比较了在不同的 λ 值下本节所提算法的跟踪误差和能量消耗。从表 3.9 可以看出，算法的跟踪误差随着 λ 值的减小而快速减小。但是，能量消耗的变化趋势与跟踪误差的变化趋势完全相反。当 λ 的值较大时，系统的能耗会迅速降低。当 $\lambda=0.4$ 和 $\lambda=0.6$ 时，算法可以在跟踪精度和能量消耗之间做出很好的权衡。由于跟踪的根本目的是实现对目标更准确的跟踪。因此，$\lambda=0.6$ 最适合本节所提跟踪算法，在该 λ 取值下，系统可以在损失较少跟踪精度的基础上节省更多能量。

表 3.9 不同 λ 下算法的跟踪误差和能量消耗

λ	跟踪误差/m	跟踪误差的变化速率	能量消耗/J	能量消耗的变化速率
0	4.99	0	1.34	−76.07%
0.2	3.33	−33.27%	1.78	−68.21%
0.4	2.35	−52.91%	2.22	−60.36%
0.6	1.56	−68.74%	3.05	−45.54%
0.8	1.24	−75.15%	4.16	−31.07%
1	1.10	−77.96%	5.60	0

预设跟踪误差阈值 Φ_0 在本节所提出的水下目标跟踪算法中起着至关重要的作用。通过调整 Φ_0 的值可以改变选择的簇成员节点的数目 N_{k+1}，进而自适应地调整所提跟踪算法的跟踪精度和能量消耗。

为了更明确地分析 Φ_0 的值对本节所提跟踪算法的跟踪性能和能量消耗的影响，将阈值 Φ_0 的取值从 3 逐步增加至 8，每次增量为 1。图 3.40 和图 3.41 分别展示了在不同的 Φ_0 下预测协方差矩阵的迹和所选簇成员节点数量 N_{k+1} 的变化情况。可以看出，随着 Φ_0 的增加，预测协方差矩阵的迹逐渐增大，而选择的簇成员节点的数目 N_{k+1} 逐渐减小。但是，当 Φ_0 大于 7 时，预测协方差矩阵的迹将逐渐收敛至一个固定的最大值，这个最大边界是由所需簇成员节点的数量 N_{k+1} 取值最小为 2 决定的，从图 3.41 也可以看出，随着 Φ_0 的增加，N_{k+1} 将逐渐收敛至 2。类似地，随着 Φ_0 的减小，预测协方差矩阵的迹也会收敛至一个最小值，这个最小边界值是由候选簇成员集合中有限的节点数目 N_S 决定的，实际上，如果非常 Φ_0 小，那么 N_{k+1} 将恰好等于候选簇成员节点的数目 N_S。在选择最优节点组合的前提下，当 $\Phi_0=4$，$\Phi_0=5$ 和 $\Phi_0=6$ 时，若预测协方差矩阵的迹大于预设跟踪误差阈值 Φ_0，则本节所提算法可以快速降低预测协方差矩阵的迹至 Φ_0，并在保证跟踪精度收敛的前提下，通过自适应地调整 N_{k+1} 使预测协方差矩阵的迹维持在预设跟踪误差阈值 Φ_0 附近，以最大限度地降低系统能耗。

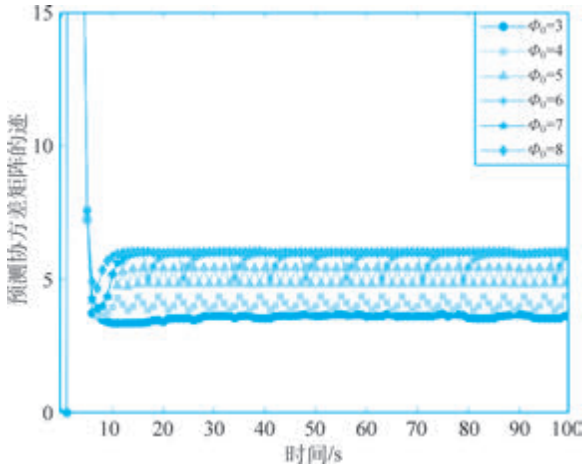


图 3.40 不同 Φ_0 下的预测协方差矩阵的迹

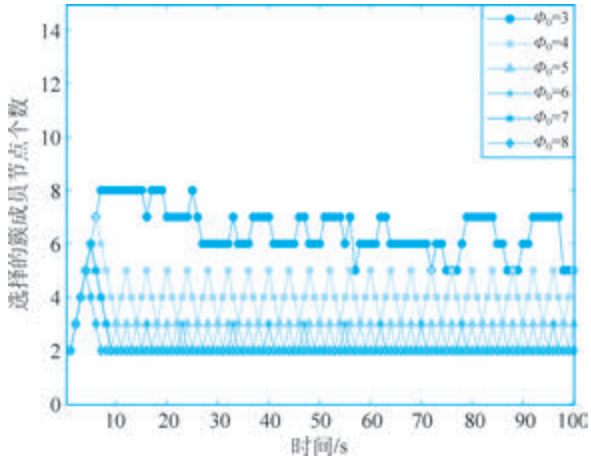


图 3.41 不同 Φ_0 下选择的簇成员节点数

为便于定量说明预设跟踪误差阈值 Φ_0 对本节所提算法的跟踪精度和网络能耗的影响，图 3.42 和图 3.43 分别显示了在不同的 Φ_0 下算法的平均跟踪误差和网络能耗的变化情况。比较图 3.42 和图 3.43 可以看出，Φ_0 的变化对网络能耗的影响与对跟踪精度的影响相比更加显著。此外，当 $3 \leqslant \Phi_0 \leqslant 6$ 时，Φ_0 的变化对本节所提算法的跟踪精度和网络能耗具有较大的影响，而当 $\Phi_0 < 3$ 或 $\Phi_0 > 6$ 时，Φ_0 的变化对跟踪精度和网络能耗的影响非常小。例如，当 Φ_0 从 4 增加到 5 时，平均跟踪误差降低了 33.10%，网络能耗降低了 44.01%。而当 Φ_0 从 2 增加到 3 时，跟踪误差仅增加了 3.28%，网络能耗也只降低了 4.09%。前者所引起的跟踪精度和网络能耗的变化是后者的 10 倍以上，这说明算法选择的簇成员节点的数目 N_{k+1} 已经达到候选集合中的最大数目 N_S 或者最低所需数目 2。以上分析表明，$\Phi_0 = 4$ 和 $\Phi_0 = 5$ 最适合本节所提的跟踪算法。为实现系统的跟踪精度和网络之间的有效平衡，本节的后续的所有仿真都以 $\Phi_0 = 5$ 为例。

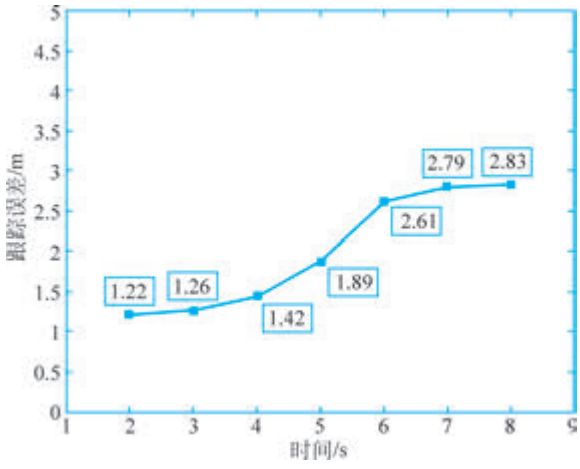


图 3.42 不同 Φ_0 下的跟踪误差

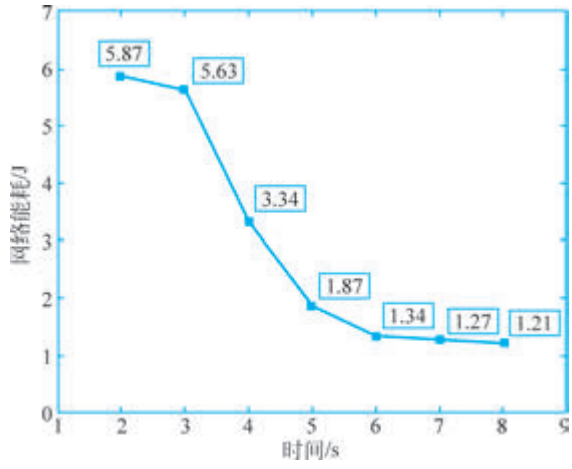


图 3.43 不同 Φ_0 下的网络能耗

最优节点组合的选择在提高跟踪精度或降低网络能耗两方面起着至关重要的作用，为验证本节提出的自适应最优节点组合选择算法的有效性，在预设跟踪误差阈值 $\Phi_0 = 5$ 和联合因子 $\lambda = 0.6$ 的条件下，图 3.44 和图 3.45 比较了 4 种节点选择策略的平均跟踪误差和网络能耗随时间的变化情况。最优节点组合就是从簇成员候选集合 $\boldsymbol{S}_{k+1}$ 中选择使目标函数最大的节点组合；最近节点组合就是从簇成

员候选集合中选择距离目标预测位置最近的节点组合。随机节点组合就是从簇成员候选集合中随机产生一个节点组合。为进行更加直观的分析和比较，表 3.10 列出了不同节点组合算法的平均跟踪误差和网络能耗。仿真结果表明，选择所有节点组合在目标跟踪精度和收敛速度方面都优于其他 3 种方法，但该方法的网络能耗也是最高的，显然这种方法不适用于能量有限的 UWSN 场景下的目标跟踪任务。随机节点组合在进行节点选择时没有考虑节点能够提供的信息效用以及节点的能量消耗，所以随机节点组合在跟踪精度和网络能耗方面都比最优节点组合要差。虽然最近节点组合的网络能耗最低，但由于没有考虑目标和节点之间的位置关系对跟踪精度的影响，因此具有较高的跟踪误差。最优节点组合在目标跟踪精度和网络能耗方面都具有较好的性能，与选择所有节点组合相比，最优节点组合节省了 71.40%的网络能耗，而跟踪误差仅仅增加了 26.61%。

彩色图片

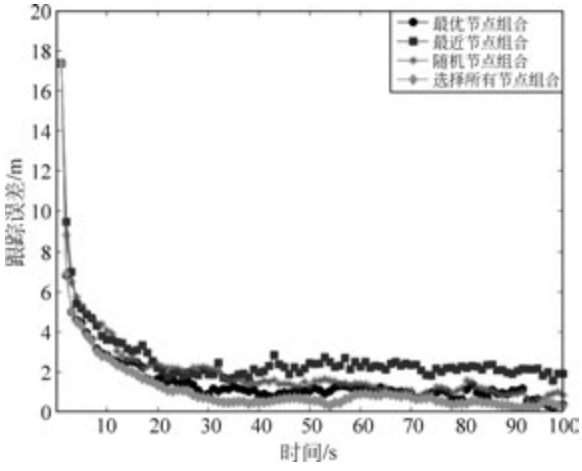


图 3.44 不同节点组合的跟踪误差

彩色图片

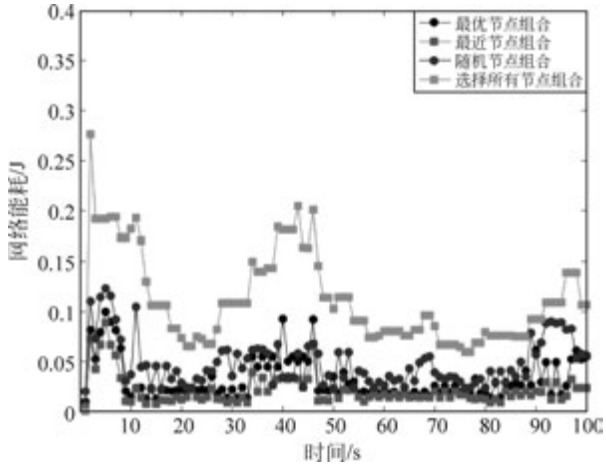


图 3.45 不同节点组合的网络能耗

表 3.10 不同节点选择算法的跟踪性能

节点选择算法	跟踪误差/m	网络能耗/J
最优节点组合	1.58	1.57
最近节点组合	2.68	1.24
随机节点组合	2.02	2.40
选择所有节点组合	1.23	5.49

此外，最优节点组合的跟踪精度比最近节点组合提高了 41.04%，但网络能耗仅增加了 26.61%。综上所述，自适应最优节点选择算法可以实现跟踪精度和网络能耗之间的有效平衡。也就是说，它可以在确保跟踪精度在可接受的范围内的同时，在最大限度地减少网络能耗，有效避免网络中某些节点因能量消耗过多而造成网络空洞，有效地延长 UWSN 的生命周期。

3.4 小结

本章深入探讨了水下目标跟踪中的节点调度技术，围绕目标跟踪精度与网络能耗平衡的核心问题，系统性地提出了多种优化方法，展示了节点调度在提升跟踪效率和延长网络寿命方面的关键作用。

本章以深度调节、能量分配和动态分簇为核心，从不同角度解决 UWSN 节点调度面临的技术挑战。首先，针对水下传感器稀疏部署导致的量测信息不足问题，本章提出了一种基于深度调节的节点调度方法。通过节点在深度方向上的自适应移动，动态优化节点位置结构，改善网络的覆盖范围和量测数据的

完整性。结合 FIM 作为跟踪精度的判据，并引入和声搜索算法进行优化，实验结果表明，该方法在稀疏网络条件下有效提升了目标跟踪精度，并在动态环境中展现出较强的鲁棒性与适应性。与传统静态节点布局相比，深度调节机制为动态目标跟踪提供了一种灵活高效的解决方案。然后，为解决水下传感器节点能量有限的问题，本章构建了节点能量收集与消耗的动态模型，并提出了一种长时能量分配策略。通过 MDP，实现了节点在时间维度上的能量优化分配，以最大化一段时间内的全局跟踪精度。实验结果表明，长时能量分配方案在同等能量预算条件下，相较于传统短时分配或均匀分配策略，能够显著提升能量利用效率并延长系统寿命。此外，该方法通过动态规划实现了节点能耗与量测精度之间的平衡，为复杂环境下的资源调度提供了理论支持。最后，针对节点能耗和协作效率之间的平衡问题，本章提出了一种动态分簇算法，通过动态选择簇头节点和簇成员节点，以实现跟踪精度与能量消耗的优化。该方法结合节点剩余能量与目标状态估计精度，并将节点选择建模为背包问题，利用动态规划方法求解最优节点组合。研究发现，动态分簇算法能够有效减少资源消耗，增强节点间的协作效率，同时显著提高目标跟踪的鲁棒性和整体性能。

总体而言，本章通过深入分析节点深度调节、能量分配和动态分簇等调度策略，解决了水下目标跟踪中存在的高能耗与低精度问题。各调度方法通过优化节点资源配置，动态调整节点位置和能量消耗，在保证精度的同时延长了系统的工作时间，为水下目标跟踪技术的应用提供了有力支持，并为未来的系统优化提供了新的思路。展望未来，随着传感器技术、人工智能和大数据分析的进一步发展，节点调度将在多目标协作、自适应调度与复杂环境适应等方向取得更多突破，为水下目标跟踪系统提供更智能、更高效的解决方案。