

Stochastic Tracking: Theory

Previous work in fish tracking by an autonomous underwater vehicle (AUV) has been limited to tracking a single individual. However, to monitor entire populations by AUV tracking, it is not feasible to pair a large number of AUVs with the members of a representative sample of the population. Instead, a smaller number of AUVs may track the population by statistical means under a stochastic control.

Stochastic tracking consists of three steps. First, the problem is simplified by discretizing the space of possible fish locations into a finite graph, so that the fish's location is represented by a node and the fish's movement is represented by an edge. Second, data in the robot's memory of past fish locations is used to generate a *transition model*, which predicts future distributions of fish population density over the graph. Finally, a Markov process is constructed to control the AUVs' positions on the graph, minimizing the difference between the population densities of the robots and fish at each node and time.

1 Discretizing the Space

We will represent the fish's location by Cartesian coordinates in the plane of the surface of the water, with the origin set to some latitude / longitude coordinate O . Latitude / longitude coordinates of the fish measured during the track are mapped to this plane. Let $r[t]$ denote the x -coordinate (distance east of O) and y -coordinate (distance north of O) of the fish at time $t > 0$.

Since the fish's location cannot be determined at regular intervals, the times for which $r[t]$ is defined also fail to occur at regular intervals. Such an uneven sampling is inconvenient for modeling the fish's location. Interpolation is used to generate an approximation $R[t]$ of the fish's location, defined for any $t = kT_{\text{sample}}$, where k is a nonnegative integer. For convenience, we may use the notation $R[k] = R[kT_{\text{sample}}]$ to index the fish's location by integer k .

This work simplifies the tracking problem by discretizing the plane of all possible fish coordinates into N discrete locations, known as *nodes*. These nodes may be chosen by any process, such as dividing the region into a grid or clustering historical data to calculate N centroids. While the nodes used in this work were stationary, this is not a requirement. The set of nodes will be denoted by V , and the physical location of node i at time t will be represented by the function $V_i(t)$.

The path of a robot is confined to certain *edges* between nodes. An edge is an ordered pair (i, j) , with $i, j \in V$, and for each edge a physical path $e(i, j)$ exists that the robot is to follow when moving from node i to node j . There need not exist an edge between any pair of nodes, as in some cases, it may not make sense for the robot to move between such nodes, particularly if finding an obstacle-free path between the nodes would be difficult. If E is the set of all edges, then the directed graph (V, E) is the *location graph*.

2 Transition Model

Since many species of fish show periodic patterns of movement, analysis of previous fish locations may predict future distributions of the fish population over the location graph. These predictions take the form of a *transition model*, which consists of a time-inhomogeneous transition matrix $T[k]$

and a population density vector $\mathbf{p}[k]$. If $N = |V|$, then $T[k]$ is an $N \times N$ left stochastic matrix in which the j, i -th entry indicates the probability of the fish at node i transitioning to node j at time k . The population density $\mathbf{p}[k]$ is a stochastic column vector in which the i -th entry is proportional to some measure of the density of the fish population at node i and time k . The transition model can either be constructed by fitting a transition matrix to previous fish data and calculating a corresponding population density vector, or a population density vector can be fit to previous data and a corresponding transition matrix be extrapolated.

2.1 Transition Matrix Approach

A transition model may be constructed by generating a transition matrix from previous data. Once the fish's location $R[k]$ is discretized to the N nodes of the graph, we may represent the fish's movement between each time step as $d_i[k]$, defined recursively as follows:

$$d_i[k] = \begin{cases} R[k+1] & R[k] = i \\ d_i[k-1] & R[k] \neq i \text{ and } k > 0 \\ i & R[k] \neq i \text{ and } k = 0 \end{cases}$$

Informally, $d_i[k]$ represents the node that the fish will occupy at time step $k+1$ if it's location at time step k is node i . Of course, we only observe the fish's transition from one source node per time step. If $R[k] = i$ and $R[k+1] = j$, then we may easily choose $d_i[k] = j$; however, we are left to guess where the fish *would* have gone from some other node $x \neq i$ to choose the value of $d_x[k]$. The above definition makes the guess that, if $R[n+1] = y$ for the largest $n < k$ at which $R[n] = x$, then $d_x[k] = y$. An alternative definition of $d_i[k]$ may make the guess that $d_i[k] = R[k+1]$ for any i , though such a definition may overemphasize the current location of the fish, at the expense of losing information about the fish's path.

From $d_i[k]$, the transition matrix is calculated as follows:

$$T[k]_{i,j} = \sum_{m=0}^L f(m|k, \sigma/T_{\text{sample}}) \langle d_j[m] = i \rangle,$$

where L is the largest value such that $d_i[L]$ is defined, $f(x|\mu, \sigma)$ is a discrete normal distribution (defined for integers 0 to $L-1$ with a total sum of 1), and $\langle P \rangle$ is 1 if the proposition P is true and 0 otherwise. The population density vector can then be predicted as

$$\mathbf{p}[k] = \left(\prod_{m=0}^{k-1} T[m] \right) \mathbf{p}[0],$$

where $\mathbf{p}[0]$ can either be an estimate of the population density at $k=0$ or the vector $\mathbf{1}/N$.

2.2 Population Density Approach

Alternatively, a transition model may be constructed by generating a population density vector from previous data. Let $\vec{R}[k] = (x_k, y_k)$ be a vector denoting the fish's location at time step k , and let $\vec{V}_i = (x_i, y_i)$ denote the location of node i . From these quantities we may define an *attenuation vector* $\mathbf{a}[k]$ as follows:

$$\mathbf{a}_i[k] = e^{-\|\vec{R}[k] - \vec{V}_i[k]\|^2 / (2\sigma_{\text{atten}}^2)}$$

where σ_{atten} is a free parameter. Informally, $\mathbf{a}[k]$ describes the weight of a particular node in the fish's location. If the fish's location at some k precisely equals the location of node i , then $\mathbf{a}_i[k] = 1$; but as the fish moves away from i , then $\mathbf{a}_i[k] \rightarrow 0$. The parameter σ_{atten} sets the "attenuation" of the a node's assigned weight as the fish moves away from it.

We then define the population density vector as a weighted sum of the attenuation vector:

$$\mathbf{p}[k] = \sum_m^L f(m|k, \sigma/T_{\text{sample}}) \mathbf{a}[m].$$

Ideally, we would like to construct a transition matrix $T[k]$ such that $\mathbf{p}[k+1] = T[k]\mathbf{p}[k]$. In some cases, however, such a construction may not be possible. Instead, we find a $T[k]$ that minimizes $\|\mathbf{p}[k+1] - T[k]\mathbf{p}[k]\|^2$, subject to the constraint that $T[k]$ is a stochastic matrix.

3 Stochastic Control

The final step is to construct a Markov process that will minimize the difference in robot population density and fish population density at each time. Again, there are two approaches: tracking the fish transition matrix, or attempting to match population density directly. In the former case, the robot simply uses the transition matrix generated by the transition model. In the latter case, a feedback control is used to generate a transition matrix at each time step that will match the robot population density to fish population density.

The entries of the robot transition matrix often exhibit periodic behavior. In these cases, performing a Fourier transform of the transition probabilities, filtering out some number of the largest Fourier coefficients, and reconstructing the transition probabilities from these coefficients can fit a sum of sinusoids to entries of the transition matrix.

3.1 Feedback Control for Density Tracking

In tracking population density, the objective is to minimize the difference between fish population density $\mathbf{p}[k]$ and robot population density $\boldsymbol{\rho}[k]$. A feedback control may be used to this end. After defining an error vector $\mathbf{e}[k] = \mathbf{p}[k] - \boldsymbol{\rho}[k]$, This control generates a transition matrix $T[k]$ at each time k as follows:

$$T_{j,i}[k] = \begin{cases} K\mathbf{e}_j[k]/(\boldsymbol{\rho}_j[k] + \epsilon) & i \neq j \\ 1 + K'\mathbf{e}_j[k]/(\boldsymbol{\rho}_j[k] + \epsilon) & i = j \end{cases}$$

The columns are subsequently normalized so that $T[k]$ is a stochastic matrix.

Insert proof of $\boldsymbol{\rho}$ converging on \mathbf{p} here...

3.2 Edge Removal

In some cases, it may be necessary to ensure that the robot has zero probability of traversing a particular edge, if such a path would be physically difficult to navigate. In these cases, linear programming may be recursively used to calculate a transition matrix $T[k]$ so that the error

$$\left\| \left(\prod_{m=0}^{k-1} T[m] \right) T[k] \boldsymbol{\rho}[0] - \mathbf{p}[k] \right\|^2,$$

(where $\boldsymbol{\rho}[0]$ is the initial population density of the robots) is minimized subject to the constraint that any entries of T corresponding to a disallowed transition are set to zero.