



(12) 发明专利

(10) 授权公告号 CN 108279692 B

(45) 授权公告日 2020.12.22

(21) 申请号 201810043819.4

CN 105607646 A, 2016.05.25

(22) 申请日 2018.01.17

US 2017255832 A1, 2017.09.07

(65) 同一申请的已公布的文献号

CN 106557809 A, 2017.04.05

申请公布号 CN 108279692 A

CN 107883961 A, 2018.04.06

US 9760806 B1, 2017.09.12

(43) 申请公布日 2018.07.13

Masaya INOUE 等. Robot Path Training

and Planning Using LSTM Network.

《Proceedings of the 2017 JSME Conference on Robotics and Mechatronics》. 2017, 1-4.

ZHANG Xuelian 等. UUV Dynamic Path

Planning and Trap Escape Strategies in

Unknown Environment. 《Proceedings of the

35th Chinese Control Conference》. 2016,

温志文 等. 基于改进蚁群算法的UUV三维路径规划方法. 《鱼雷技术》. 2016, 第24卷(第2期),

Ayesha Khan 等. Using Recurrent Neural

Networks (RNNs) as Planners for Bio-

Inspired Robotic Motion. 《2017 IEEE

Conference on Control Technology and

Applications (CCTA)》. 2017,

王宏健 等. 基于高斯混合容积卡尔曼滤波

的UUV自主导航定位算法. 《仪器仪表学报》

. 2015, 第36卷(第2期),

审查员 李江平

(73) 专利权人 哈尔滨工程大学

地址 150001 黑龙江省哈尔滨市南岗区南

通大街145号哈尔滨工程大学科技处

知识产权办公室

(72) 发明人 王宏健 林常见 么洪飞 肖瑶

张宏瀚 张雪莲

(51) Int. Cl.

G05D 1/06 (2006.01)

G06N 3/00 (2006.01)

G06N 3/04 (2006.01)

G06N 3/08 (2006.01)

(56) 对比文件

CN 105607646 A, 2016.05.25

CN 107092254 A, 2017.08.25

CN 101833338 A, 2010.09.15

CN 105843234 A, 2016.08.10

CN 106020213 A, 2016.10.12

CN 105549600 A, 2016.05.04

CN 101408772 A, 2009.04.15

权利要求书4页 说明书10页 附图4页

(54) 发明名称

一种基于LSTM-RNN的UUV动态规划方法

(57) 摘要

本发明公开了一种基于LSTM-RNN的UUV动态规划方法,属于无人潜航器领域,包括如下步骤:

步骤(1):选择几何模型构建障碍环境模型;步骤

(2):利用蚁群算法建立用于获取数据集的UUV动

态规划器;步骤(3):设计用于动态规划的LSTM-

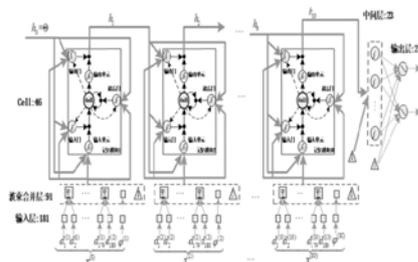
RNN网络模型;步骤(4):获取数据集;步骤(5):利

用数据集中训练集的数据训练LSTM-RNN网络,得

到基于LSTM-RNN网络的动态规划器;步骤(6):将

声纳探测信息及目标点信息输入至基于LSTM-

RNN网络的运动规划器,获得UUV下一时刻的航向及航速。本方法具有强大的学习能力,同时还具有非常强的泛化能力,这使得所实现的动态规划器适用于复杂的环境。同时满足实时性的要求,并且所规划的路径符合UUV的运动特性。



CN 108279692 B

1. 一种基于LSTM-RNN的UUV动态规划方法,其特征在于,包含如下步骤:

步骤(1):选择几何模型构建障碍环境模型;

步骤(2):利用蚁群算法建立用于获取数据集的UUV动态规划器;

步骤(3):设计用于动态规划的LSTM-RNN网络模型;

步骤(4):获取数据集;

步骤(5):利用数据集中训练集的数据训练LSTM-RNN网络,得到基于LSTM-RNN网络的动态规划器;

步骤(6):将声纳探测信息及目标点信息输入至基于LSTM-RNN网络的运动规划器,获得UUV下一时刻的航向及航速;

所述的步骤(1)具体为:对于椭圆形或圆形障碍,存储椭圆形或圆形障碍的两个对角点坐标,利用这两点坐标,计算出椭圆圆心和长短半径,获得椭圆形或圆形障碍的全部信息;对于多边形障碍,存储多边形顶点的坐标,由任意一个顶点开始,按顺时针或者逆时针顺序将多边形顶点依次存储于一个顺序链表中,通过顺序链表获取多边形障碍的全部信息;

所述的步骤(2)具体为:蚁群算法信息素更新规则为:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k,$$

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{当蚂蚁}k\text{经过边}e_{ij} \\ 0 & \text{其他} \end{cases};$$

上式中, i, j 是状态点序号, m 是蚁群中蚂蚁总数, ρ 是挥发系数, Q 是常数, L_k 是蚂蚁 k 寻食付出的代价,即路径长度;

每一代蚂蚁中排名前四分之一的蚂蚁走过的路径的信息素强化规则为:

$$\tau_{ij}(t+1) = \lambda \times \tau_{ij}(t)$$

$$\lambda = 1 + 0.5 \times \frac{D}{L_k},$$

上式中, D 是起点到终点的欧式距离;

信息素浓度的限制处理:

$$\tau_{ij}(t+1) = \begin{cases} \tau_{\min} & \text{如果} \tau_{ij}(t+1) < \tau_{\min} \\ \tau_{\max} & \text{如果} \tau_{ij}(t+1) > \tau_{\max} \\ \tau_{ij}(t+1) & \text{其他} \end{cases};$$

上式中, τ_{\min} 是自行设置的信息素浓度上限, τ_{\max} 是自行设置的信息素浓度下限;

蚁群算法的状态转移规则如下:

$$p_{ij}^k = \frac{(\tau_{ij})^\alpha (\eta_j)^\beta}{\sum_{r \in allow_i} (\tau_{ir})^\alpha (\eta_r)^\beta};$$

p_{ij}^k 为蚂蚁 k 处于 i 点处时下一步转移到 i 的可视点 j 的概率, $allow_i$ 为蚂蚁 k 允许做状态转

移的*i*点的可视点集合, η 是启发式函数,取为*i*点到目标点距离的倒数, α 为信息素的重要程度, β 为启发函数的重要程度;

所述的步骤(3)中的LSTM-RNN网络模型:包括输入层,波束合并层,隐藏层,中间层以及输出层五个部分;

输入层:接收样本输入,接收样本有181维,前180维为模拟感知器探测的局部环境信息,最后1维为UUV当前位置与目标点在局部坐标下形成的夹角;

波束合并层:波束合并层是对输入层邻近波束求和后取平均;合并领域的大小为2,且不重叠,即该层每个单元对输入层每相邻两个单元的输出值求和取平均后输出;波束合并层共91维,其中前90维为合并后的局部环境特征,最后一维特征不做处理,直接输出;波束合并层的输出和一个偏置项一起作为记忆模块的输入,采用全连接形式;

隐藏层:包括10个记忆模块;记忆模块包括输入单元、输出单元、输入门、输出门、遗忘门、Cell;输入单元、输出单元、输入门、输出门、遗忘门、Cell的维度都为46;采用的序列长度为10,按序列展开后有10个记忆模块,上一时刻的模块输出反馈到下一时刻的模块中;假定第一个记忆模块虚拟的前驱模块输出恒为0向量;输入门、输出门、遗忘门的激活函数是sigmoid函数,输入单元、输出单元的激活函数是tanh函数;

中间层:第10个记忆模块的输出添加上偏置后以全连接方式输入到中间层,维度为23,激活函数是tanh函数;

输出层:中间层加上一个偏置项后以全连接方式输入到输出层,输出层维度为2,激活函数为线性函数,分别输出UUV的航向和航速;

所述的步骤(4)具体为:在全局坐标系中生成若干组随机地图,采用教师系统进行运动规划,水平覆盖角为 360° 、包含180个波束的二维仿真声呐在每个路径节点采集环境信息,所采集的信息为训练样本,训练样本构成训练集,将教师系统输出的航向及航速作为训练标签;

二维仿真声呐在每个路径节点采集的环境信息为:

$$d_i = [d_i^1, d_i^2, \dots, d_i^{180}],$$

上式中, d_i^j 为当前时刻第*i*个波束探测到的距离信息;

当第*i*个波束未探测到障碍时, $d_i^j = 200m$;

将*t*时刻仿真声呐探测到的距离信息做如下处理:

$$D_i = [200 - d_i^1, 200 - d_i^2, \dots, 200 - d_i^{180}];$$

将*t*时刻UUV所在位置与目标点形成的角度表示为 φ_i ,则*t*时刻LSTM-RNN网络的输入向量为:

$$x_t = [D_i, \varphi_i];$$

*t*时刻LSTM-RNN网络的标签:

$$l_t = [\theta_t, v_t],$$

上式中, θ_t 表示期望的UUV的航向, v_t 表示期望的UUV的航速;

所述的步骤(5)具体为:

步骤(5.1):初始化LSTM-RNN网络各层权重,令*i*=0;

步骤(5.2):将训练集中的输入数据 $[x_{t-9}, \dots, x_t]$ 馈入网络,经过LSTM-RNN网络的前向传播过程得到对应输出 y_t ;

步骤(5.3):计算网络输出 y_t 与对应标签 l_t 的误差;

步骤(5.4):判断是否训练集中数据均完成计算,若是,执行步骤(5.5),否则 $t=t+1$,并返回步骤(5.2);

步骤(5.5):利用标准梯度下降误差反向传播算法更新LSTM-RNN网络权重;

步骤(5.6): $i=i+1$,并判断 $i \text{ MOD } 500$ 是否为零,若是,执行(5.7),否则返回步骤(5.2);

步骤(5.7):在测试集中随机选取50个序列,使用当前网络进行测试,计算平均误差;

步骤(5.8):若测试集中均方误差连续10次没有减小,则停止训练,否则返回步骤(5.2)。

2.根据权利要求1所述的一种基于LSTM-RNN的UUV动态规划方法,其特征在于:所述的LSTM-RNN网络的前向传播过程为:

输入门:

$$i_t^{(in)} = x_t W_{X-I} + h_{t-1} W_{H-I} + s_{t-1} W_{S-I},$$

$$i_t^{(out)} = f(i_t^{(in)});$$

遗忘门:

$$f_t^{(in)} = x_t W_{X-F} + h_{t-1} W_{H-F} + s_{t-1} W_{S-F},$$

$$f_t^{(out)} = f(f_t^{(in)});$$

记忆模块中的输入单元:

$$g_t^{(in)} = x_t W_{X-G} + h_{t-1} W_{H-G},$$

$$g_t^{(out)} = g(g_t^{(in)});$$

Cell更新:

$$s_t = f_t^{(out)} \bullet s_{t-1} + i_t^{(out)} \bullet g_t^{(out)};$$

输出门:

$$o_t^{(in)} = x_t W_{X-O} + h_{t-1} W_{H-O} + s_t W_{S-O},$$

$$o_t^{(out)} = f(o_t^{(in)});$$

记忆模块输出:

$$h_t = o_t^{(out)} \bullet h(s_t);$$

输出层:

$$y_t^{(in)} = h_t W_{H-Y},$$

$$y_t^{(out)} = y(y_t^{(in)});$$

计算损失:

$$L_t = \frac{1}{2}(y_t^d - y_t^{(out)})^2 ;$$

上述式子中,X表示网络输入,Y表示网络输出,G表示记忆模块的输入单元,S表示记忆模块的Cell,H表示整个记忆模块;分别使用I、O、F表示输入门、输出门、遗忘门;W表示权值矩阵, W_{i-j} 表示i单元与j单元的权值;符号“ \cdot ”表示向量或矩阵之间对应元素相乘;函数f为门控单元激活函数;函数g、h、y分别是输入单元、输出单元、输出层的激活函数;当一个完整的序列完成一次前向传播后,可以得到损失函数:

$$L = \sum_{t=1}^T L_t = \sum_{t=1}^T \frac{1}{2}(y_t^d - y_t^{(out)})^2 ;$$

上式中,T为序列长度。

3. 根据权利要求2所述的一种基于LSTM-RNN的UUV动态规划方法,其特征在于:所述的标准梯度下降误差反向传播算法:

输出层局部梯度:

$$\delta_t^{(Y)} = \frac{\partial L}{\partial y_t^{(in)}} = \frac{\partial L_t}{\partial y_t^{(in)}} = -(y_t^d - y_t^{(out)}) \cdot y_t'(y_t^{(in)}) ;$$

记忆模块局部梯度:

$$\begin{aligned} \delta_t^{(H)} &= \frac{\partial L}{\partial h_t} = \frac{\partial L_t}{\partial y_t^{(in)}} \frac{\partial y_t^{(in)}}{\partial h_t} + \frac{\partial L_{t+1}}{\partial i_{t+1}^{(in)}} \frac{\partial i_{t+1}^{(in)}}{\partial h_t} + \frac{\partial L_{t+1}}{\partial f_{t+1}^{(in)}} \frac{\partial f_{t+1}^{(in)}}{\partial h_t} + \frac{\partial L_{t+1}}{\partial o_{t+1}^{(in)}} \frac{\partial o_{t+1}^{(in)}}{\partial h_t} + \frac{\partial L_{t+1}}{\partial g_{t+1}^{(in)}} \frac{\partial g_{t+1}^{(in)}}{\partial h_t} \\ &= \delta_t^{(Y)} W_{H-Y}^T + \delta_{t+1}^{(I)} W_{H-I}^T + \delta_{t+1}^{(F)} W_{H-F}^T + \delta_{t+1}^{(O)} W_{H-O}^T + \delta_{t+1}^{(G)} W_{H-G}^T ; \end{aligned}$$

输出门局部梯度:

$$\delta_t^{(O)} = \frac{\partial L}{\partial o_t^{(in)}} = \frac{\partial L_t}{\partial o_t^{(in)}} = \frac{\partial L_t}{\partial h_t} \frac{\partial h_t}{\partial o_t^{(in)}} = \delta_t^{(H)} \cdot h_t(s_t) \cdot f'(o_t^{(in)}) ;$$

Cell状态局部梯度:

$$\begin{aligned} \delta_t^{(S)} &= \frac{\partial L}{\partial s_t} = \frac{\partial L_t}{\partial h_t} \frac{\partial h_t}{\partial s_t} + \frac{\partial L}{\partial s_{t+1}} \frac{\partial s_{t+1}}{\partial s_t} + \frac{\partial L_{t+1}}{\partial i_{t+1}^{(in)}} \frac{\partial i_{t+1}^{(in)}}{\partial s_t} + \frac{\partial L_{t+1}}{\partial f_{t+1}^{(in)}} \frac{\partial f_{t+1}^{(in)}}{\partial s_t} + \frac{\partial L_t}{\partial o_t^{(in)}} \frac{\partial o_t^{(in)}}{\partial s_t} \\ &= \delta_t^{(H)} \cdot o_t^{(out)} \cdot h'(s_t) + \delta_{t+1}^{(S)} \cdot f_{t+1}^{(out)} + \delta_{t+1}^{(I)} W_{S-I}^T + \delta_{t+1}^{(F)} W_{S-F}^T + \delta_t^{(O)} W_{S-O}^T ; \end{aligned}$$

遗忘门局部梯度:

$$\delta_t^{(F)} = \frac{\partial L}{\partial f_t^{(in)}} = \frac{\partial L_t}{\partial s_t} \cdot s_{t-1} \cdot f'(f_t^{(in)}) = \delta_t^{(S)} \cdot s_{t-1} \cdot f'(f_t^{(in)}) ;$$

输入门局部梯度:

$$\delta_t^{(I)} = \frac{\partial L}{\partial i_t^{(in)}} = \frac{\partial L_t}{\partial s_t} \frac{\partial s_t}{\partial i_t^{(in)}} = \delta_t^{(S)} \cdot g_t^{(out)} \cdot f'(i_t^{(in)}) ;$$

记忆模块输入单元局部梯度:

$$\delta_t^{(G)} = \frac{\partial L}{\partial g_t^{(in)}} = \frac{\partial L_t}{\partial s_t} \frac{\partial s_t}{\partial g_t^{(in)}} = \delta_t^{(S)} \cdot i_t^{(out)} \cdot g'(g_t^{(in)}) .$$

一种基于LSTM-RNN的UUV动态规划方法

技术领域

[0001] 本发明属于无人潜航器领域,尤其涉及一种基于LSTM-RNN的UUV动态规划方法。

背景技术

[0002] UUV在未知环境下的动态路径规划能力是体现其智能水平的重要指标之一。传统动态规划方法往往遭受着环境模型精度与规划实时性矛盾的困扰,且在具有大量随机运动障碍的复杂环境中,还需设计辅助策略才能实现理想的规避,这些辅助策略的设计是相当复杂的,而且需要大量的计算时间。因此,探寻一种简单、廉价、高效、易于实现的动态路径规划方法具有重大的理论和实践价值。

[0003] 随着陆地资源消耗日趋枯竭,各国开始大量投入金钱、时间和人力开发水下资源,水下无人潜航器被重点所关注,而各种各样的无人潜航器在达到复杂作业及安全航行时,均要按照相应的航行路径进行操作,才能完成一次水下潜航。为了实施最佳航行计划、保持潜航器能在预定的航向上运行,要保证潜航器的操纵性,潜航器具备很好的操纵性时,能稳定维持行驶航向、深度及航速,又能快速对航向、深度及航速进行改变,正确的执行种种操作。水下无人潜航器路径的控制,也影响着无人潜航器的工作性能,需要对水下潜航器的路径进行有效的控制。

[0004] 路径规划,即找到一条从起点到终点的无碰撞安全路径,并且使特定的规划指标如路径最短、风险最小、任务完成量最多等达到最优。路径规划算法在国内外已有相当多的研究,比较常用的有A*算法、Dijkstra算法、蚁群和遗传等仿生智能算法。然而基于上述传统算法的实时路径规划系统,存在环境模型的精度以及路径的优化程度与规划的实时性相互矛盾的问题。深度学习是目前最具潜力的人工智能算法,其强大的非线性拟合能力可以为UUV提供一套简单、高效、易于实现的运动规划系统。这使得UUV具有了一定的自主学习的能力,从而减少了其对感知设备的依赖,增强了其在避碰过程中的灵活度。运动规划系统的规划能力随着UUV在运动的过程中的不断学习而逐步提高。

[0005] 中国专利CN107368076A公开了一种智能环境下机器人运动路径深度学习控制规划方法,该方法使用深度学习DBN网络解决运载机器人全局路径规划问题。所提深度学习DBN网络包括一层输入层、三层隐藏层以及一层输出层,其中可视层与第一隐藏层构成第一层限制玻尔兹曼机RBM,第一隐藏层与第二隐藏层构成第二层限制玻尔兹曼机RBM,第二隐藏层与第三隐藏层构成BP网络层。该专利采用逐层训练的方法依次对两层限制玻尔兹曼机RBM和BP网络层进行训练。与全局路径规划不同,动态规划器需要根据过去一段时间内的环境信息实时输出规划结果。相较于限制玻尔兹曼机RBM,LSTM-RNN具有处理较长时间序列的能力,且更适用于复杂的规划问题。

发明内容

[0006] 本发明的目的在于公开灵活度高,能够适应复杂环境的一种基于LSTM-RNN的UUV动态规划方法。

[0007] 本发明的目的是这样实现的：

[0008] 一种基于LSTM-RNN的UUV动态规划方法，包含如下步骤：

[0009] 步骤(1)：选择几何模型构建障碍环境模型：

[0010] 对于椭圆形或圆形障碍，存储椭圆形或圆形障碍的两个对角点坐标，利用这两点坐标，计算出椭圆圆心和长短半径，获得椭圆形或圆形障碍的全部信息；对于多边形障碍，存储多边形顶点的坐标，由任意一个顶点开始，按顺时针或者逆时针顺序将多边形顶点依次存储于一个顺序链表中，通过顺序链表获取多边形障碍的全部信息。

[0011] 步骤(2)：利用蚁群算法建立用于获取数据集的UUV动态规划器：

[0012] 蚁群算法信息素更新规则为：

$$[0013] \quad \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k,$$

$$[0014] \quad \Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{当蚂蚁}k\text{经过边}e_{ij} \\ 0 & \text{其他} \end{cases};$$

[0015] 上式中， i, j 是状态点序号， m 是蚁群中蚂蚁总数， ρ 是挥发系数， Q 是常数， L_k 是蚂蚁 k 寻食付出的代价，即路径长度；

[0016] 每一代蚂蚁中排名前四分之一的蚂蚁走过的路径的信息素强化规则为：

$$[0017] \quad \tau_{ij}(t+1) = \lambda \times \tau_{ij}(t)$$

$$[0018] \quad \lambda = 1 + 0.5 \times \frac{D}{L_k},$$

[0019] 上式中， D 是起点到终点的欧式距离；

[0020] 信息素浓度的限制处理：

$$[0021] \quad \tau_{ij}(t+1) = \begin{cases} \tau_{\min} & \text{如果} \tau_{ij}(t+1) < \tau_{\min} \\ \tau_{\max} & \text{如果} \tau_{ij}(t+1) > \tau_{\max} \\ \tau_{ij}(t+1) & \text{其他} \end{cases};$$

[0022] 上式中， τ_{\min} 是自行设置的信息素浓度上限， τ_{\max} 是自行设置的信息素浓度下限。

[0023] 蚁群算法的状态转移规则如下：

$$[0024] \quad p_{ij}^k = \frac{(\tau_{ij})^\alpha (\eta_j)^\beta}{\sum_{r \in \text{allow}_i} (\tau_{ir})^\alpha (\eta_r)^\beta};$$

[0025] p_{ij}^k 为蚂蚁 k 处于 i 点处时下一步转移到 i 的可视点 j 的概率， allow_i 为蚂蚁 k 允许做状态转移的 i 点的可视点集合， η 是启发式函数，取为 i 点到目标点距离的倒数， α 为信息素的重要程度， β 为启发函数的重要程度；

[0026] 步骤(3)：设计用于动态规划的LSTM-RNN网络模型：

[0027] LSTM-RNN网络模型：包括输入层，波束合并层，隐藏层，中间层以及输出层五个部分；

[0028] 输入层:接收样本输入,接收样本有181维,前180维为模拟感知器探测的局部环境信息,最后1维为UUV当前位置与目标点在局部坐标下形成的夹角;

[0029] 波束合并层:波束合并层是对输入层邻近波束求和后取平均;合并领域的大小为2,且不重叠,即该层每个单元对输入层每相邻两个单元的输出值求和取平均后输出;波束合并层共91维,其中前90维为合并后的局部环境特征,最后一维特征不做处理,直接输出;波束合并层的输出和一个偏置项一起作为记忆模块的输入,采用全连接形式;

[0030] 隐藏层:包括10个记忆模块;记忆模块包括输入单元、输出单元、输入门、输出门、遗忘门、Cell;输入单元、输出单元、输入门、输出门、遗忘门、Cell的维度都为46;采用的序列长度为10,按序列展开后有10个记忆模块,上一时刻的模块输出反馈到下一时刻的模块中;假定第一个记忆模块虚拟的前驱模块输出恒为0向量;输入门、输出门、遗忘门的激活函数是sigmoid函数,输入单元、输出单元的激活函数是tanh函数;

[0031] 中间层:第10个记忆模块的输出添加上偏置后以全连接方式输入到中间层,维度为23,激活函数是tanh函数;

[0032] 输出层:中间层加上一个偏置项后以全连接方式输入到输出层,输出层维度为2,激活函数为线性函数,分别输出UUV的航向和航速。

[0033] 步骤(4):获取数据集:

[0034] 在全局坐标系中生成若干组随机地图,采用教师系统进行运动规划,水平覆盖角为 360° 、包含180个波束的二维仿真声呐在每个路径节点采集环境信息,所采集的信息为训练样本,训练样本构成训练集,将教师系统输出的航向及航速作为训练标签;

[0035] 二维仿真声呐在每个路径节点采集的环境信息为:

$$[0036] \quad d_i = [d_i^1, d_i^2, \dots, d_i^{180}],$$

[0037] 上式中, d_i^i 为当前时刻第i个波束探测到的距离信息;

[0038] 当第i个波束未探测到障碍时, $d_i^i = 200m$;

[0039] 将t时刻仿真声纳探测到的距离信息做如下处理:

$$[0040] \quad D_t = [200 - d_t^1, 200 - d_t^2, \dots, 200 - d_t^{180}];$$

[0041] 将t时刻UUV所在位置与目标点形成的角度表示为 φ_t , 则t时刻LSTM-RNN网络的输入向量为:

$$[0042] \quad x_t = [D_t, \varphi_t];$$

[0043] t时刻LSTM-RNN网络的标签:

$$[0044] \quad l_t = [\theta_t, v_t],$$

[0045] 上式中, θ_t 表示期望的UUV的航向, v_t 表示期望的UUV的航速

[0046] 步骤(5):利用数据集中训练集的数据训练LSTM-RNN网络,得到基于LSTM-RNN网络的动态规划器:

[0047] 具体的训练步骤为:

[0048] 步骤(5.1):初始化LSTM-RNN网络各层权重,令 $i = 0$;

[0049] 步骤(5.2):将训练集中的输入数据 $[x_{t-9}, \dots, x_t]$ 馈入网络,经过LSTM-RNN网络的前向传播过程得到对应输出 y_t :

[0050] 步骤(5.3):计算网络输出 y_t 与对应标签 l_t 的误差;

[0051] 步骤(5.4):判断是否训练集中数据均完成计算,若是,执行五,否则 $t=t+1$,并返回步骤(5.2);

[0052] 步骤(5.5):利用标准梯度下降误差反向传播算法更新LSTM-RNN网络权重;

[0053] 步骤(5.6): $i=i+1$,并判断 $i \text{ MOD } 500$ 是否为零,若是,执行七,否则返回步骤(5.2);

[0054] 步骤(5.7):在测试集中随机选取50个序列,使用当前网络进行测试,计算平均误差;

[0055] 步骤(5.8):若测试集中均方误差连续10次没有减小,则停止训练,否则返回步骤(5.2)。

[0056] 步骤(5.2)中的LSTM-RNN网络的前向传播过程为:

[0057] 输入门:

$$[0058] \quad i_t^{(in)} = x_t W_{X-I} + h_{t-1} W_{H-I} + s_{t-1} W_{S-I},$$

$$[0059] \quad i_t^{(out)} = f(i_t^{(in)});$$

[0060] 遗忘门:

$$[0061] \quad f_t^{(in)} = x_t W_{X-F} + h_{t-1} W_{H-F} + s_{t-1} W_{S-F},$$

$$[0062] \quad f_t^{(out)} = f(f_t^{(in)});$$

[0063] 记忆模块中的输入单元:

$$[0064] \quad g_t^{(in)} = x_t W_{X-G} + h_{t-1} W_{H-G},$$

$$[0065] \quad g_t^{(out)} = g(g_t^{(in)});$$

[0066] Cell更新:

$$[0067] \quad s_t = f_t^{(out)} \bullet s_{t-1} + i_t^{(out)} \bullet g_t^{(out)};$$

[0068] 输出门:

$$[0069] \quad o_t^{(in)} = x_t W_{X-O} + h_{t-1} W_{H-O} + s_t W_{S-O},$$

$$[0070] \quad o_t^{(out)} = f(o_t^{(in)});$$

[0071] 记忆模块输出:

$$[0072] \quad h_t = o_t^{(out)} \bullet h(s_t);$$

[0073] 输出层:

$$[0074] \quad y_t^{(in)} = h_t W_{H-Y},$$

$$[0075] \quad y_t^{(out)} = y(y_t^{(in)});$$

[0076] 计算损失:

$$[0077] \quad L_t = \frac{1}{2} (y_t^d - y_t^{(out)})^2.$$

[0078] 上述式子中,X表示网络输入,Y表示网络输出,G表示记忆模块的输入单元,S表示记忆模块的Cell,H表示整个记忆模块;分别使用I、O、F表示输入门、输出门、遗忘门.W表示权值矩阵, W_{i-j} 表示i单元与j单元的权值;符号“ \bullet ”表示向量或矩阵之间对应元素相乘;函

数f为门控单元激活函数；函数g、h、y分别是输入单元、输出单元、输出层的激活函数；当一个完整的序列完成一次前向传播后，可以得到损失函数：

$$[0079] \quad L = \sum_{t=1}^T L_t = \sum_{t=1}^T \frac{1}{2} (y_t^d - y_t^{(out)})^2 ;$$

[0080] 上式中，T为序列长度。

[0081] 步骤(5.5)中的标准梯度下降误差反向传播算法为：

[0082] 输出层局部梯度：

$$[0083] \quad \delta_t^{(y)} = \frac{\partial L}{\partial y_t^{(in)}} = \frac{\partial L_t}{\partial y_t^{(in)}} = -(y_t^d - y_t^{(out)}) \cdot y'(y_t^{(in)}) ;$$

[0084] 记忆模块局部梯度：

$$[0085] \quad \delta_t^{(H)} = \frac{\partial L}{\partial h_t} = \frac{\partial L_t}{\partial y_t^{(in)}} \frac{\partial y_t^{(in)}}{\partial h_t} + \frac{\partial L_{t+1}}{\partial i_{t+1}^{(in)}} \frac{\partial i_{t+1}^{(in)}}{\partial h_t} + \frac{\partial L_{t+1}}{\partial f_{t+1}^{(in)}} \frac{\partial f_{t+1}^{(in)}}{\partial h_t} + \frac{\partial L_{t+1}}{\partial o_{t+1}^{(in)}} \frac{\partial o_{t+1}^{(in)}}{\partial h_t} + \frac{\partial L_{t+1}}{\partial g_{t+1}^{(in)}} \frac{\partial g_{t+1}^{(in)}}{\partial h_t} \\ = \delta_t^{(y)} W_{H-Y}^T + \delta_{t+1}^{(I)} W_{H-I}^T + \delta_{t+1}^{(F)} W_{H-F}^T + \delta_{t+1}^{(O)} W_{H-O}^T + \delta_{t+1}^{(G)} W_{H-G}^T ;$$

[0086] 输出门局部梯度：

$$[0087] \quad \delta_t^{(O)} = \frac{\partial L}{\partial o_t^{(in)}} = \frac{\partial L_t}{\partial o_t^{(in)}} = \frac{\partial L_t}{\partial h_t} \frac{\partial h_t}{\partial o_t^{(in)}} = \delta_t^{(H)} \cdot h(s_t) \cdot f'(o_t^{(in)}) ;$$

[0088] Cell状态局部梯度：

$$[0089] \quad \delta_t^{(S)} = \frac{\partial L}{\partial s_t} = \frac{\partial L_t}{\partial h_t} \frac{\partial h_t}{\partial s_t} + \frac{\partial L}{\partial s_{t+1}} \frac{\partial s_{t+1}}{\partial s_t} + \frac{\partial L_{t+1}}{\partial i_{t+1}^{(in)}} \frac{\partial i_{t+1}^{(in)}}{\partial s_t} + \frac{\partial L_{t+1}}{\partial f_{t+1}^{(in)}} \frac{\partial f_{t+1}^{(in)}}{\partial s_t} + \frac{\partial L_t}{\partial o_t^{(in)}} \frac{\partial o_t^{(in)}}{\partial s_t} \\ = \delta_t^{(H)} \cdot o_t^{(out)} \cdot h'(s_t) + \delta_{t+1}^{(S)} \cdot f_t^{(out)} + \delta_{t+1}^{(I)} W_{S-I}^T + \delta_{t+1}^{(F)} W_{S-F}^T + \delta_t^{(O)} W_{S-O}^T ;$$

[0090] 遗忘门局部梯度：

$$[0091] \quad \delta_t^{(F)} = \frac{\partial L}{\partial f_t^{(in)}} = \frac{\partial L_t}{\partial s_t} \cdot s_{t-1} \cdot f'(f_t^{(in)}) = \delta_t^{(S)} \cdot s_{t-1} \cdot f'(f_t^{(in)}) ;$$

[0092] 输入门局部梯度：

$$[0093] \quad \delta_t^{(I)} = \frac{\partial L}{\partial i_t^{(in)}} = \frac{\partial L_t}{\partial s_t} \frac{\partial s_t}{\partial i_t^{(in)}} = \delta_t^{(S)} \cdot g_t^{(out)} \cdot f'(i_t^{(in)}) ;$$

[0094] 记忆模块输入单元局部梯度：

$$[0095] \quad \delta_t^{(G)} = \frac{\partial L}{\partial g_t^{(in)}} = \frac{\partial L_t}{\partial s_t} \frac{\partial s_t}{\partial g_t^{(in)}} = \delta_t^{(S)} \cdot i_t^{(out)} \cdot g'(g_t^{(in)}) .$$

[0096] 步骤(6)：将声纳探测信息及目标点信息输入至基于LSTM-RNN网络的运动规划器，获得UUV下一时刻的航向及航速。

[0097] 本发明的有益效果为：

[0098] 本发明利用了LSTM-RNN在长时间序列上强大的非线性拟合能力实现了一个端到端的简单高效的动态规划器。设计的LSTM-RNN网络，不仅有强大的学习能力，同时还具有非常强的泛化能力，这使得所实现的动态规划器适用于复杂的环境。同时，该动态规划器满足实时性的要求，并且所规划的路径符合UUV的运动特性。

附图说明

- [0099] 图1是一种基于LSTM-RNN的UUV动态规划方法的LSTM-RNN网络结构图。
 [0100] 图2是一种基于LSTM-RNN的UUV动态规划方法的感知设备仿真示意图。
 [0101] 图3是一种基于LSTM-RNN的UUV动态规划方法的角度示意图。
 [0102] 图4是一种基于LSTM-RNN的UUV动态规划方法在静态环境中的仿真效果图。
 [0103] 图5是一种基于LSTM-RNN的UUV动态规划方法在动态环境中不同时刻的仿真效果图。

具体实施方式

[0104] 下面结合附图来进一步描述本发明：

[0105] 一种基于LSTM-RNN的UUV动态规划方法，包含如下步骤：

[0106] 步骤(1)：选择几何模型构建障碍环境模型：

[0107] 对于二维的几何模型，遵循“使用最少的数据量，描述最完整的信息”的原则，将障碍物按形状分为椭圆形或圆形障碍，多边形障碍两种类型。

[0108] 对于椭圆形或圆形障碍，存储椭圆形或圆形障碍的两个对角点坐标，利用这两点坐标，计算出椭圆圆心和长短半径，获得椭圆形或圆形障碍的全部信息。

[0109] 对于多边形障碍，存储多边形顶点的坐标，由任意一个顶点开始，按顺时针或者逆时针顺序将多边形顶点依次存储于一个顺序链表中，这样就可以通过顺序链表还原多边形障碍的信息。

[0110] 步骤(2)：利用蚁群算法建立用于获取数据集的UUV动态规划器：

[0111] 蚁群算法的信息素更新规则如下：

$$[0112] \quad \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k,$$

$$[0113] \quad \Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{当蚂蚁}k\text{经过边}e_{ij} \\ 0 & \text{其他} \end{cases};$$

[0114] 上式中， i, j 是状态点序号， m 是蚁群中蚂蚁总数， ρ 是挥发系数， Q 是常数， L_k 是蚂蚁 k 寻食付出的代价，即路径长度。

[0115] 为了加快收敛速度，对每一代蚂蚁中排名前四分之一的蚂蚁走过的路径做信息素强化：

$$[0116] \quad \tau_{ij}(t+1) = \lambda \times \tau_{ij}(t)$$

$$[0117] \quad \lambda = 1 + 0.5 \times \frac{D}{L_k},$$

[0118] 上式中， D 是起点到终点的欧式距离。

[0119] 为了使蚁群算法在搜索后期仍具有一定的探索能力，在信息素更新以及信息素强化完成后，再做信息素浓度的限制处理：

$$[0120] \quad \tau_{ij}(t+1) = \begin{cases} \tau_{\min} & \text{如果 } \tau_{ij}(t+1) < \tau_{\min} \\ \tau_{\max} & \text{如果 } \tau_{ij}(t+1) > \tau_{\max} \\ \tau_{ij}(t+1) & \text{其他} \end{cases} ;$$

[0121] 上式中, τ_{\min} 是自行设置的信息素浓度上限, τ_{\max} 是自行设置的信息素浓度下限。

[0122] 蚁群算法的状态转移规则如下:

$$[0123] \quad p_{ij}^k = \frac{(\tau_{ij})^\alpha (\eta_j)^\beta}{\sum_{r \in allow_i} (\tau_{ir})^\alpha (\eta_r)^\beta} ;$$

[0124] p_{ij}^k 为蚂蚁k处于i点处时下一步转移到i的可见点j的概率, $allow_i$ 为蚂蚁k允许做状态转移的i点的可见点集合, η 是启发式函数, 取为i点到目标点距离的倒数。 α 为信息素的重要程度。 β 为启发函数的重要程度

[0125] 步骤(3): 设计用于动态规划的LSTM-RNN网络模型:

[0126] 如图1, LSTM-RNN网络模型包括输入层, 波束合并层, 隐藏层, 中间层以及输出层五个部分。

[0127] 输入层: 接收样本输入, 接收样本有181维, 如图2, 前180维为模拟感知器探测的局部环境信息, 如图3, 最后1维为UUV当前位置与目标点在局部坐标下形成的夹角;

[0128] 波束合并层: 波束合并层是对输入层邻近波束求和后取平均, 利用相邻波束间的高度相关性, 降低输入维度;

[0129] 合并领域的大小为2, 且不重叠, 即该层每个单元对输入层每相邻两个单元的输出值求和取平均后输出;

[0130] 波束合并层共91维, 其中前90维为合并后的局部环境特征, 最后一维特征不做处理, 直接输出;

[0131] 波束合并层的输出和一个偏置项一起作为记忆模块的输入, 采用全连接形式。

[0132] 隐藏层: 记忆模块中Cell的维度取为46, 内部的输入单元、输出单元、输入门、输出门、遗忘门的维度都为46; 采用的序列长度为10, 按序列展开后有10个记忆模块, 上一时刻的模块输出反馈到下一时刻的模块中; 由于第一个模块没有实质的前驱模块, 假定其虚拟的前驱模块输出恒为0向量; 门控单元采用的激活函数都是sigmoid函数, 输入、输出单元的激活函数是tanh函数;

[0133] 中间层: 第十个记忆模块输出添加上偏置后以全连接方式输入到中间层, 维度为23, 激活函数是tanh函数;

[0134] 输出层: 中间层加上一个偏置后以全连接方式输入到输出层, 输出层维度为2, 激活函数为线性函数, 分别输出UUV的航向和航速。

[0135] 步骤(4): 获取数据集:

[0136] 如图4, 在全局坐标系中生成若干组随机地图, 采用教师系统进行运动规划, 水平覆盖角为 360° 、包含180个波束的二维仿真声呐在每个路径节点采集环境信息, 所采集的信息为训练样本, 将教师系统输出的航向及航速作为训练标签。

[0137] 二维仿真声呐在每个路径节点采集环境信息为:

[0138] $d_i = [d_i^1, d_i^2, \dots, d_i^{180}]$,

[0139] 其中 d_i^i 为当前时刻第 i 个波束探测到的距离信息, 当第 i 个波束未探测到障碍时, $d_i^i = 200m$;

[0140] 为了简化输入信息, 将 t 时刻声纳探测到的距离信息做如下处理:

[0141] $D_t = [200 - d_t^1, 200 - d_t^2, \dots, 200 - d_t^{180}]$;

[0142] 将 t 时刻UUV所在位置与目标点形成的角度表示为 φ_t , 则 t 时刻LSTM-RNN网络的输入向量

[0143] $x_t = [D_t, \varphi_t]$;

[0144] t 时刻LSTM-RNN网络的标签

[0145] $l_t = [\theta_t, v_t]$,

[0146] 上式中, θ_t 表示期望的UUV的航向, v_t 表示期望的UUV的航速。

[0147] 步骤(5): 利用训练集中数据训练LSTM-RNN网络, 得到基于LSTM-RNN网络的动态规划器:

[0148] 将训练数据中的二维仿真声呐在每个路径节点采集的环境信息作为输入数据, 将训练集中相对应的训练标签作为输出数据, 对LSTM-RNN网络进行训练, 从而获得基于LSTM-RNN网络的动态规划器;

[0149] 损失函数为平方损失函数;

[0150] 在训练开始前对数据集中数据做如下预处理: 数据集的归一化方法为减去列最小值后除以列最大值和列最小值的差。

[0151] 在训练过程中输入数据和标签对应关系为:

[0152] 若时刻 $t < 10$, 则 $[x_1, \dots, x_t]$ 对应标签 l_t , 若 $t \geq 10$, 则 $[x_{t-9}, \dots, x_t]$ 对应标签 l_t 。

[0153] LSTM-RNN网络的训练过程可表述为:

[0154] 步骤(5.1): 初始化LSTM-RNN网络各层权重, 令 $i = 0$;

[0155] 步骤(5.2): 将训练集中的输入数据 $[x_{t-9}, \dots, x_t]$ 馈入网络, 经过LSTM-RNN网络的前向传播

[0156] 过程得到对应输出 y_t ;

[0157] 步骤(5.3): 计算网络输出 y_t 与对应标签 l_t 的误差;

[0158] 步骤(5.4): 判断是否训练集中数据均完成计算, 若是, 执行五, 否则 $t = t + 1$, 并返回步骤(5.2);

[0159] 步骤(5.5): 利用标准梯度下降误差反向传播算法更新LSTM-RNN网络权重;

[0160] 步骤(5.6): $i = i + 1$, 并判断 $i \text{ MOD } 500$ 是否为零, 若是, 执行七, 否则返回步骤(5.2);

[0161] 步骤(5.7): 在测试集中随机选取50个序列, 使用当前网络进行测试, 计算平均误差;

[0162] 步骤(5.8): 若测试集中均方误差连续10次没有减小, 则停止训练, 否则返回步骤(5.2)。

[0163] LSTM-RNN网络的前向传播过程为:

[0164] 输入门:

$$[0165] \quad i_t^{(in)} = x_t W_{X-I} + h_{t-1} W_{H-I} + s_{t-1} W_{S-I},$$

$$[0166] \quad i_t^{(out)} = f(i_t^{(in)});$$

[0167] 遗忘门:

$$[0168] \quad f_t^{(in)} = x_t W_{X-F} + h_{t-1} W_{H-F} + s_{t-1} W_{S-F},$$

$$[0169] \quad f_t^{(out)} = f(f_t^{(in)});$$

[0170] 记忆模块中的输入单元:

$$[0171] \quad g_t^{(in)} = x_t W_{X-G} + h_{t-1} W_{H-G},$$

$$[0172] \quad g_t^{(out)} = g(g_t^{(in)});$$

[0173] Cell更新:

$$[0174] \quad s_t = f_t^{(out)} \bullet s_{t-1} + i_t^{(out)} \bullet g_t^{(out)};$$

[0175] 输出门:

$$[0176] \quad o_t^{(in)} = x_t W_{X-O} + h_{t-1} W_{H-O} + s_t W_{S-O},$$

$$[0177] \quad o_t^{(out)} = f(o_t^{(in)});$$

[0178] 记忆模块输出:

$$[0179] \quad h_t = o_t^{(out)} \bullet h(s_t);$$

[0180] 输出层:

$$[0181] \quad y_t^{(in)} = h_t W_{H-Y},$$

$$[0182] \quad y_t^{(out)} = y(y_t^{(in)});$$

[0183] 计算损失:

$$[0184] \quad L_t = \frac{1}{2} (y_t^d - y_t^{(out)})^2.$$

[0185] 上述式子中,X表示网络输入,Y表示网络输出,G表示记忆模块的输入单元,S表示记忆模块的Cell,H表示整个记忆模块;分别使用I、O、F表示输入门、输出门、遗忘门。W表示权值矩阵, W_{i-j} 表示i单元与j单元的权值。符号“ \bullet ”表示向量(或矩阵)之间对应元素相乘。函数f为门控单元激活函数,一般取sigmoid函数;函数g、h、y分别是输入单元、输出单元、输出层的激活函数,根据具体任务选择适合类型的激活函数。当一个完整的序列完成一次前向传播后,可以得到损失函数:

$$[0186] \quad L = \sum_{t=1}^T L_t = \sum_{t=1}^T \frac{1}{2} (y_t^d - y_t^{(out)})^2;$$

[0187] 上式中,T为序列长度。

[0188] 标准梯度下降误差反向传播算法为:

[0189] 输出层局部梯度:

$$[0190] \quad \delta_t^{(y)} = \frac{\partial L}{\partial y_t^{(in)}} = \frac{\partial L_t}{\partial y_t^{(in)}} = -(y_t^d - y_t^{(out)}) \bullet y_t'(y_t^{(in)});$$

[0191] 记忆模块局部梯度:

$$\begin{aligned}
[0192] \quad \delta_t^{(H)} &= \frac{\partial L}{\partial h_t} = \frac{\partial L_t}{\partial y_t^{(in)}} \frac{\partial y_t^{(in)}}{\partial h_t} + \frac{\partial L_{t+1}}{\partial i_{t+1}^{(in)}} \frac{\partial i_{t+1}^{(in)}}{\partial h_t} + \frac{\partial L_{t+1}}{\partial f_{t+1}^{(in)}} \frac{\partial f_{t+1}^{(in)}}{\partial h_t} + \frac{\partial L_{t+1}}{\partial o_{t+1}^{(in)}} \frac{\partial o_{t+1}^{(in)}}{\partial h_t} + \frac{\partial L_{t+1}}{\partial g_{t+1}^{(in)}} \frac{\partial g_{t+1}^{(in)}}{\partial h_t} \\
&= \delta_t^{(Y)} W_{H-Y}^T + \delta_{t+1}^{(I)} W_{H-I}^T + \delta_{t+1}^{(F)} W_{H-F}^T + \delta_{t+1}^{(O)} W_{H-O}^T + \delta_{t+1}^{(G)} W_{H-G}^T ;
\end{aligned}$$

[0193] 输出门局部梯度:

$$[0194] \quad \delta_t^{(O)} = \frac{\partial L}{\partial o_t^{(in)}} = \frac{\partial L_t}{\partial o_t^{(in)}} = \frac{\partial L_t}{\partial h_t} \frac{\partial h_t}{\partial o_t^{(in)}} = \delta_t^{(H)} \bullet h(s_t) \bullet f'(o_t^{(in)}) ;$$

[0195] Cell状态局部梯度:

$$\begin{aligned}
[0196] \quad \delta_t^{(S)} &= \frac{\partial L}{\partial s_t} = \frac{\partial L_t}{\partial h_t} \frac{\partial h_t}{\partial s_t} + \frac{\partial L}{\partial s_{t+1}} \frac{\partial s_{t+1}}{\partial s_t} + \frac{\partial L_{t+1}}{\partial i_{t+1}^{(in)}} \frac{\partial i_{t+1}^{(in)}}{\partial s_t} + \frac{\partial L_{t+1}}{\partial f_{t+1}^{(in)}} \frac{\partial f_{t+1}^{(in)}}{\partial s_t} + \frac{\partial L_t}{\partial o_t^{(in)}} \frac{\partial o_t^{(in)}}{\partial s_t} \\
&= \delta_t^{(H)} \bullet o_t^{(out)} \bullet h'(s_t) + \delta_{t+1}^{(S)} \bullet f_{t+1}^{(out)} + \delta_{t+1}^{(I)} W_{S-I}^T + \delta_{t+1}^{(F)} W_{S-F}^T + \delta_t^{(O)} W_{S-O}^T ;
\end{aligned}$$

[0197] 遗忘门局部梯度:

$$[0198] \quad \delta_t^{(F)} = \frac{\partial L}{\partial f_t^{(in)}} = \frac{\partial L_t}{\partial s_t} \bullet s_{t-1} \bullet f'(f_t^{(in)}) = \delta_t^{(S)} \bullet s_{t-1} \bullet f'(f_t^{(in)}) ;$$

[0199] 输入门局部梯度:

$$[0200] \quad \delta_t^{(I)} = \frac{\partial L}{\partial i_t^{(in)}} = \frac{\partial L_t}{\partial s_t} \frac{\partial s_t}{\partial i_t^{(in)}} = \delta_t^{(S)} \bullet g_t^{(out)} \bullet f'(i_t^{(in)}) ;$$

[0201] 记忆模块输入单元局部梯度:

$$[0202] \quad \delta_t^{(G)} = \frac{\partial L}{\partial g_t^{(in)}} = \frac{\partial L_t}{\partial s_t} \frac{\partial s_t}{\partial g_t^{(in)}} = \delta_t^{(S)} \bullet i_t^{(out)} \bullet g'(g_t^{(in)}) .$$

[0203] 步骤(6):将声纳探测信息及目标点信息输入至基于LSTM-RNN网络的运动规划器,获得UUV下一时刻的航向及航速。

[0204] 在具体的工作过程中,本发明的使用过程是这样的:

[0205] 第一步:将UUV布放在起始位置;

[0206] 第二步:仿真声纳开始获取UUV当前位置的环境信息 x_t ;

[0207] 第三步:将仿真声纳获取的环境信息 $[x_{t-9}, \dots, x_t]$ 输出到LSTM-RNN,获得下一时刻UUV的航向及航速;

[0208] 第四步:UUV执行运动指令,到达下一路径点;

[0209] 第五步:判断UUV是否到达目标点,若是,动态规划器停止工作,否则返回第二步。

[0210] 如图5,本发明在动态环境中不同时刻的仿真效果图。

[0211] 本发明提供了一种基于LSTM-RNN的UUV动态规划方法,利用了LSTM-RNN在长时间序列上强大的非线性拟合能力实现了一个端到端的简单高效的动态规划器。设计的LSTM-RNN网络,不仅有强大的学习能力,同时还具有非常强的泛化能力,这使得所实现的动态规划器适用于复杂的环境。同时,该动态规划器满足实时性的要求,并且所规划的路径符合UUV的运动特性。

[0212] 这里必须指出的是,本发明所公开的技术手段不仅限于上述实施方式所公开的技术手段,还包括由以上技术特征任意组合所组成的技术方案。

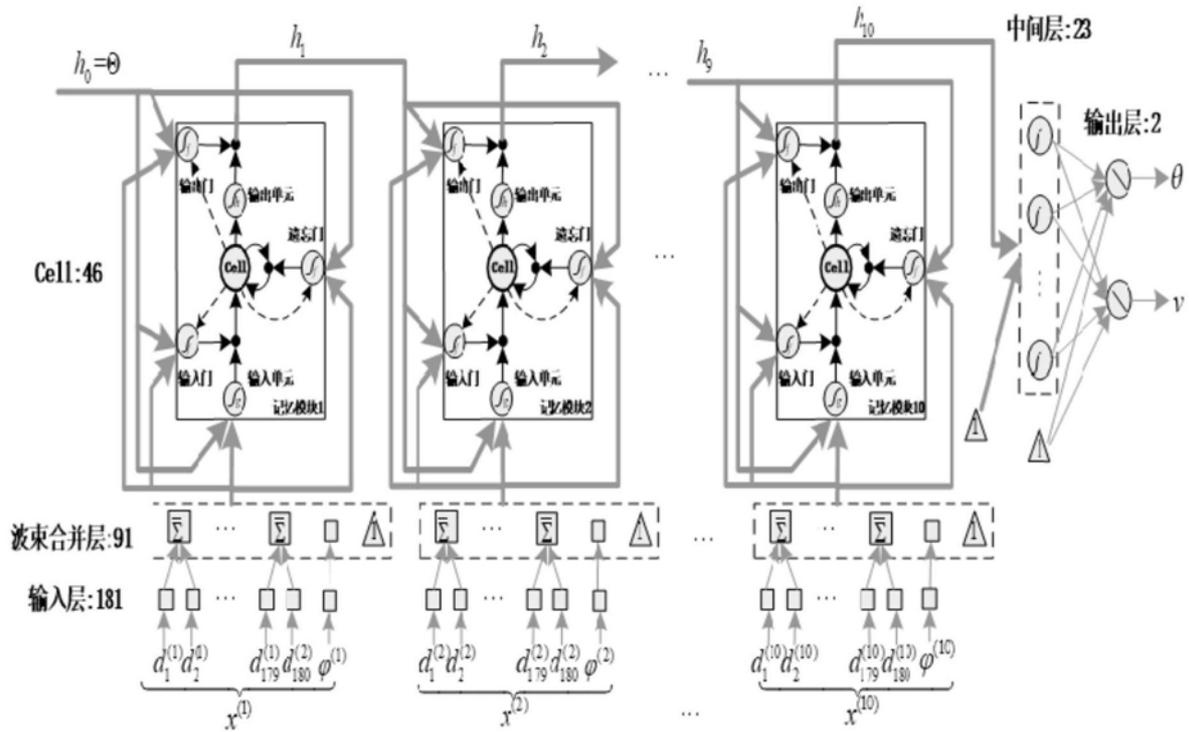


图1

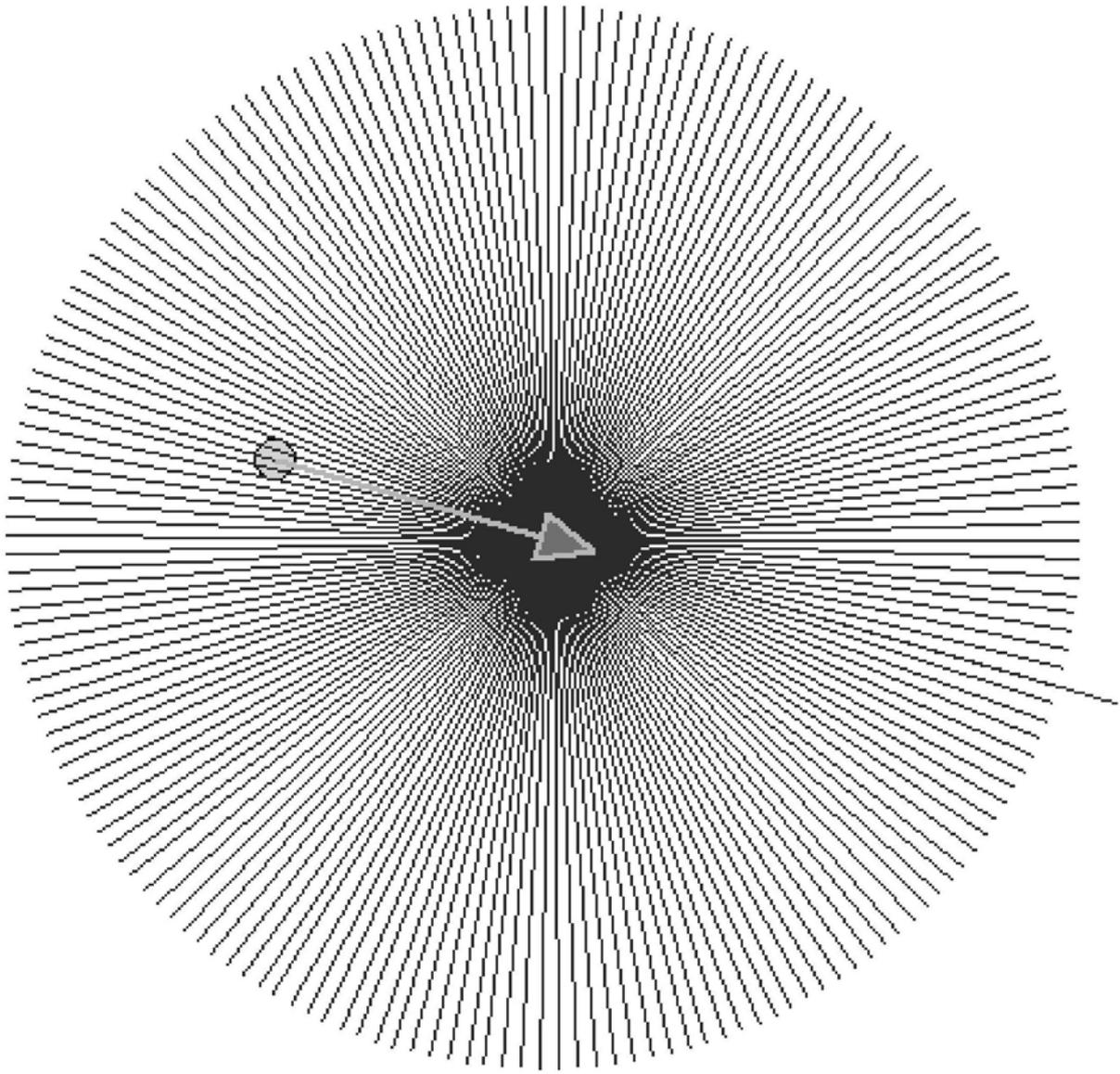


图2

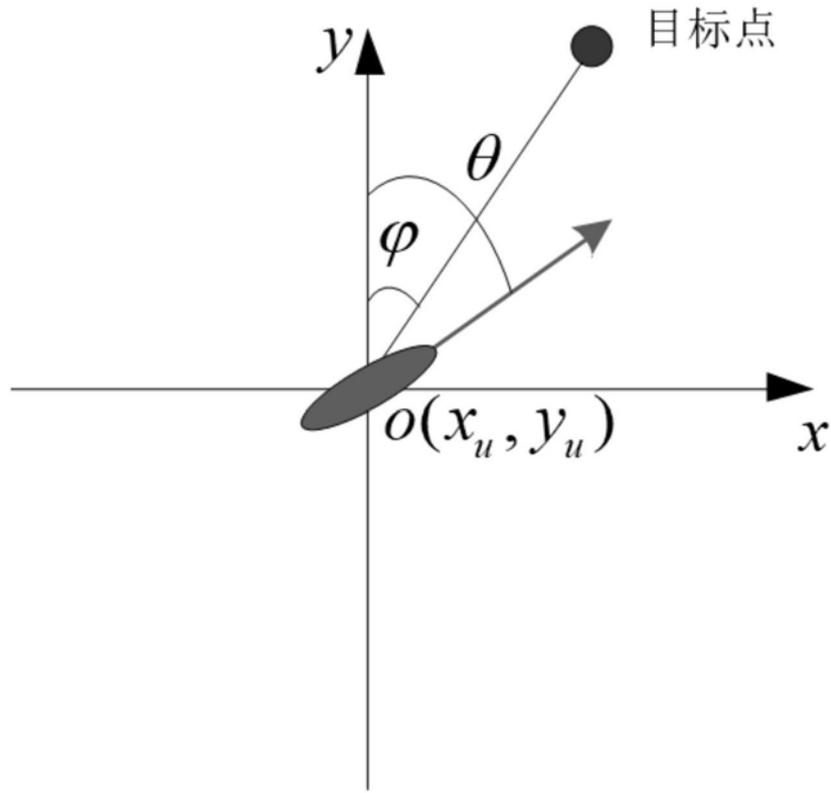


图3



图4

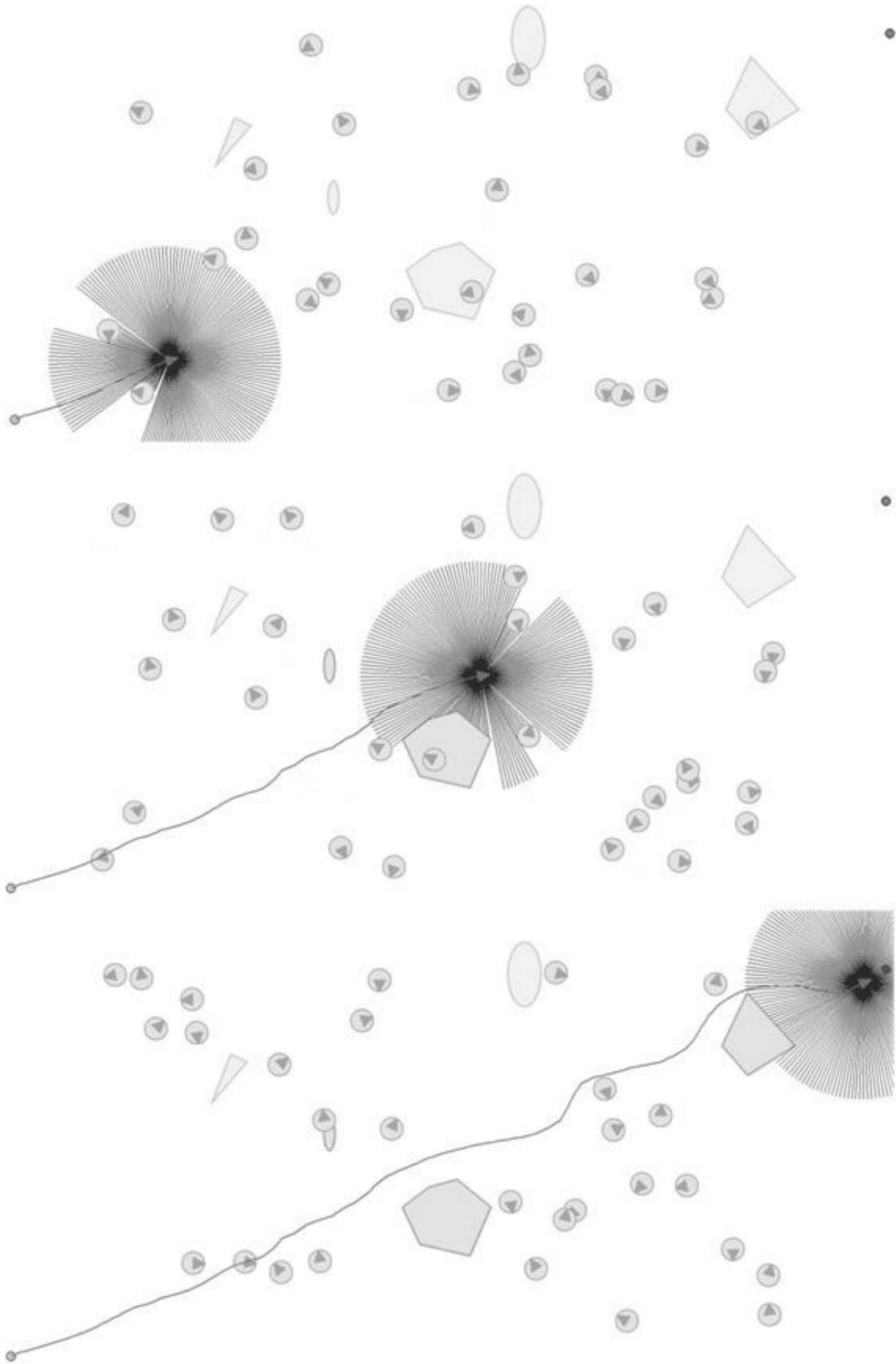


图5