

Для обработки тупиковой ситуации:

**Шаг 1:** Если робот двигался по кратчайшему маршруту и наткнулся на препятствие, то необходимо удалить интервалы указателей, которые образуют заблокированную связь.

**Шаг 2:** Связать выбранный интервал последнего указателя с тупиком.

**Шаг 3:** Вернуть робота к предыдущему (дотупиковому) указателю.

Приведенный выше упрощенный алгоритм стратегического планировщика не содержит всех нюансов, необходимых для успешного движения в нестационарной среде, однако он позволяет глобально оценить, как это все происходит.

#### 4. Экспериментальные результаты

Тестирование системы стратегического планирования проводилось на базе программного моделирующего комплекса для управления автономным мобильным роботом. Топологическая карта на базе нейронной сети встречного распространения имеет следующие параметры: количество перекрестков (указателей)  $n=100$ ,  $\varepsilon$ -окрестность притяжения указателя равна 2 метрам – т.е. примерно равна максимальной дистанции восприятия объектов сенсорными устройствами. В процессе тестирования системы был подобран необходимый набор правил эффективного движения в сложной операционной среде. В ходе ее тестирования в различных операционных средах было обнаружено, что один и тот же эвристический алгоритм не всегда оптимален во всех операционных средах и требует некоторых косметических изменений при наличии определенных требований к поставленной задаче.

#### 5. Заключение

В данной статье была рассмотрена система глобального планирования маршрута движения для мобильного робота. Она состоит из трех важных компонентов: топологической карты, нейронной сети планирования кратчайшего маршрута и эвристического алгоритма планирования поведением. Структура данной системы планирования мотивируется тем, что наиболее оптимальный вариант получается при расщеплении сложной задачи планирования движением мобильного робота на отдельные подзадачи и решение с их помощью соответствующих подсистем навигационной системы. Только взаимодействие различных концепций и моделей позволяет создать компактную систему, оперирующую в реальном масштабе времени.

УДК 681.3

*Димаков В.М.*

## ЛОКАЛИЗАЦИЯ АВТОНОМНОГО МОБИЛЬНОГО РОБОТА В ДИНАМИЧЕСКОЙ СРЕДЕ: НЕЙРОСЕТЕВОЕ РЕШЕНИЕ ПРОБЛЕМЫ

### 1. Введение

Под проблемой локализации обычно понимается привязка собственной (подвижной) координатной системы робота к стационарной системе координат среды, в которой оперирует робот. Как правило, система локализации является частью системы картографии и планирования, которая определяет структуру и характер знаний о среде. При этом каждая система должна оцениваться с точки зрения соотношения эффективности и стоимости, что определяет целесообразность ее использования. Здесь под эффективностью будем понимать способность системы успешно решать поставленную задачу в течение заданного промежутка времени. Поскольку навигационные системы для транспортных средств являются систе-

благодарности

Автор выражает благодарность рецензентам за полезные замечания, оказанные в процессе написания данной статьи.

#### СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Thrun S. and Bücken A. Learning Maps for Indoor Mobile Robot Navigation// Carnegie-Mellon University, School of Computer Science/ Technical Report CMU-CS-96-121 – 1996.
2. Thrun S. Probabilistic Algorithms in Robotics// Carnegie-Mellon University, School of Computer Science/ Technical Report CMU-CS-00-126 – 2000.
3. Thrun S. Particle Filters in Robotics// In Proceedings of Uncertainty in AI (UAI) – 2002.
4. Sathiya Keerthi S., Ravindran B.A Tutorial Survey of Reinforcement Learning// Dept. of Computer Science and Automation, Indian Institute of Science – Bangalore, 1994.
5. Smart W.D. and Kaelbling L.P. Effective Reinforcement Learning for Mobile Robots// Proceedings of the IEEE International Conference on Robotics and Automation – 2002.
6. Thrun S. Efficient Exploration in Reinforcement Learning// Carnegie-Mellon University, School of Computer Science/ Technical Report CMU-CS-92-102 – 1992.
7. Kaelbling L.P., Cassandra A.R. and Kurien J.A. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation// In proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems – 1996.
8. Lefevre P., Prüß A. and Zimmer U.R. ALICE – Topological Exploration, Cartography and Adaptive Navigation on a Simple Mobile Robot// TSRPC '94, Leeuwenhorst/ The Netherlands, June 24-26 – 1994.
9. Лебедев Д.В., Штайль Й.Я. Нейросетевая модель для планирования путей автономного робота с учетом динамических изменений в среде// Материалы 13-й научно-технической конференции «Экстремальная робототехника», 16-18 апреля 2002/ ЦНИИ РТК – СПб, 2002.
10. K.Schiling, V.Golovko and V.Dimakov. Neural system for mobile robot autonomous navigation// Proc. of Workshop on Design Methodologies for Signal Processing, Zakopane, Poland, August 29-30, 1996/ Gliwice:Institute of Electronics, Silesian Technical University - 1996 - p.124-130.
11. V.Dimakov. The Neural Network Approach for the Shortest Path Planning Problem// In Proc. of the 2<sup>nd</sup> ICSC Symposium on Neural Computation, Berlin, Germany/ Eds.: H.Bothe&R.Rojas, International Computer Science Conventions, Canada/Switzerland – 2000 – pp.70-75.

мами реального времени, то на требования к подобным системам накладываются также жесткие временные ограничения. Под критерием стоимость рассматривают вычислительные и трудовые затраты на разработку данной системы. В данном случае, они могут рассматриваться условно, поскольку подобные системы являются, в основном, продуктом научных исследований.

В последнее время мобильные роботы стали оборудоваться датчиками системы глобального позиционирования GPS [1,2]. Их разрешение может достигать 2см [1], что ставит под вопрос саму постановку задачи локализации и вообще целесообразность разработки подобных систем. К сожалению, подобные датчики доступны не везде и используются пре-

имущественно в военной области. Вследствие этого, актуальность проблемы сохраняется.

При постановке задачи для системы локализации учитываются четыре немаловажных фактора: неточность сенсорных устройств, неточность одометрической системы транспортной платформы, динамический характер среды и случайное изменение позиции робота вследствие воздействия внешних факторов (например, перенос робота человеком в другое положение [3]).

Первый фактор решается с использованием различного рода датчиков, которые компенсируют недостатки друг друга. Чаще всего используются ультразвуковые датчики вследствие их дешевизны для определения расстояний до препятствий, лазерные и инфракрасные сканеры для определения препятствий в двухмерном диапазоне, CCD-камеры, и даже радар для определения препятствий в 3-х мерном пространстве [1]. Использование каждого из этих датчиков ограничено различного рода природными факторами (солнечное излучение, дождь и т.п.), типом отражающих поверхностей, наличием фантомов при отражении звуковых волн [4]. Все это свидетельствует о нечеткости воспринимаемой роботом среды и, вследствие этого, ее неопределенности.

Фактор неточности одометрической системы подвижной платформы тоже влияет на решение задачи локализации существенным образом. На практике, каждое движущее колесо автономного мобильного робота оборудовано оптическими датчиками. С их помощью отслеживается перемещение робота и, тем самым, контролируется его система координат [5]. Однако в процессе локализации происходит накопление одометрической ошибки благодаря множеству внешних и внутренних факторов: проскальзыванию колес, шероховатости пола, ошибки квантования оптических датчиков, установленных на колесах. Здесь различают систематические и несистематические ошибки. Систематические ошибки обычно устраняются в процессе калибровки, тогда как несистематические являются следствием недостатков механической основы робота.

Существенное влияние на систему локализации оказывает также динамический характер среды. Под динамичностью здесь понимается наличие движущихся объектов (люди, другие роботы), открытые/закрытые двери помещений, перемещаемые предметы (например, ящики на складе [6]). Причем надежностью системы локализации является толерантность к подобному виду изменений среды. Сложность здесь заключается в том, чтобы система смогла правильно распознать положение робота при наличии случайных помех и, в то же время, не спутать с другой похожей ситуацией.

Воздействие случайных факторов на агент (робот) может также усложнить проблему локализации в пространстве. Это особенно актуально в случае малых роботов, которые могут быть телепортированы в любую позицию без уведомления их навигационной системы об этом. Примером здесь могут послужить роботы Aibo (Sony), играющие в футбол и которые могут быть перенесены судьей в необходимую точку футбольного поля [3].

Все вышеперечисленное формирует требования к системе локализации автономного транспортного средства и определяет способ построения навигационной системы в целом.

Суть всех методов локализации основано на том, чтобы в ходе исследования роботом незнакомой территории необходимо выделить из всего объема информации такие отличительные признаки (маяки), которые являлись бы уникальными для идентификации, в будущем, положения робота в пространстве. Существует фактически два подхода в решении задачи о локализации: *вероятностный* и *топологический*.

Вероятностный метод локализации основан на использовании метода Монте Карло (MCL-метод) [3,7]. Данный метод включает в себя элементы байесовской условной вероятности,

калмановского фильтра и марковские цепей. В данном случае, первоначально строится карта размещений (occupancy grid), которая является матрицей ячеек, принимающих значения в диапазоне [0,1], в зависимости от того, занята ли ячейка препятствием или нет. Дискретизация такой карты размещений, естественно, зависит от размера операционной среды, так как требует значительных вычислительных затрат для ее организации и поддержки.

Согласно методу MCL [7], чтобы найти положение робота  $\mathbf{x}_k$  в пространстве, необходимо рекурсивно вычислить вероятность  $p(\mathbf{x}_k | \mathbf{Z}^k)$  нахождения его в этой позиции при наличии соответствующих данных от сенсорных устройств  $\mathbf{Z}^k$ . Сам метод состоит из двух фаз: *фазы предсказания* и *фазы изменения*. В первой фазе строится модель движения для предсказания текущего положения робота в форме предсказательной вероятностной функции (PDF)  $p(\mathbf{x}_k | \mathbf{Z}^{k-1})$ . Предполагается, что текущее положение робота  $\mathbf{x}_k$  зависит только от предыдущего состояния  $\mathbf{x}_{k-1}$  (цепь Маркова) и соответствующего управляющего воздействия  $\mathbf{u}_{k-1}$ . Таким образом, *модель движения* описывается как условная вероятность  $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ . При этом предсказательная функция описывается следующим образом

$$p(\mathbf{x}_k | \mathbf{Z}^{k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{Z}^{k-1}) d\mathbf{x}_{k-1}. \quad (1)$$

Вторая фаза называется *фазой изменения*, которая включает информацию от сенсорных датчиков для получения последующей вероятностной функции  $p(\mathbf{x}_k | \mathbf{Z}^k)$ . Предполагается, что данные от сенсорных устройств на текущем шаге  $\mathbf{z}_k$  условно независимы от предыдущих измерений  $\mathbf{Z}^{k-1}$  для состояния  $\mathbf{x}_k$  и задаются в виде вероятности  $p(\mathbf{z}_k | \mathbf{x}_k)$ , что выражает вероятность того, что робот в положении  $\mathbf{x}_k$  наблюдает состояние окружающей среды  $\mathbf{z}_k$ . Таким образом, последующая вероятностная функция получается из теоремы Байеса

$$p(\mathbf{x}_k | \mathbf{Z}^k) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{Z}^{k-1})}{p(\mathbf{z}_k | \mathbf{Z}^{k-1})} \quad (2)$$

После фазы изменения, процесс повторяется рекурсивно. К достоинствам MCL-метода можно отнести эффективность глобальной локализации в условиях неопределенности среды, но требует значительный объем памяти и времени, а также программная модель сложно реализуема. В настоящее время данный подход является частью, так называемого, метода *синхронной локализации и картографии* (Simultaneous Localization and Mapping (SLAM)) [3,8], используемого для навигации автономного мобильного робота в нечеткой среде.

Диаметрально противоположным методом локализации является *топологический метод*. Он основан на технологии кластеризации знаний, предложенной Кохоненом (самоорганизующиеся карты)[9] и Фритцке (растущие структуры ячеек) [10]. Центральной идеей *качественных топологических моделей среды* (Qualitative Topological Word Model, QT-Model) [11] является реконструкция совместно с топологическими соседями «ситуации»-положения робота, в котором он уже побывал исходя из представленной сенсорной информации. Под «ситуацией» здесь понимается обнаруженные роботом границы препятствий в метрической системе координат.

Центральным аспектом в формировании топологической карты является процесс адаптации. В процессе исследования роботом незнакомой местности происходит адаптация выбранной ячейки топологической карты с наименьшим расстоянием до препятствий для новой ситуации. Одновременно с

адаптацией выбранной ячейки топологической карты выполняется также адаптация ее соседей, для которых выполняется условие смежности с ней. При этом адаптация осуществляется только для ячеек, из геометрически ограниченной, некоторым радиусом, области поиска. Скорость адаптации регулируется соответствующими коэффициентами, которые также влияют на степень корреляции топологической карты с окружающим миром исходя из следующего принципа: чем быстрее адаптируется топологическая карта к окружающей среде, тем меньше корреляция между ними.

К достоинствам топологических карт можно отнести эффективность глобальной локализации, относительно минимальные требования к объему используемой памяти при локализации положения робота в пространстве, толерантность к ошибкам сенсорных устройств. В качестве основного недостатка можно отметить значительные временные затраты на создание QT-модели среды.

Рассматриваемый в данной статье метод локализации представляет собой вариант топологической карты, который не требует создания глобальной модели среды, а только описание ее отдельных компонентов. Выбор этих компонентов и является, собственно, целью задачи локализации – т.е. найти такие отличительные признаки среды (маяки), которые позволили бы однозначно определить местонахождение робота. Как правило, общий объем полезной информации для задачи локализации невелик и варьируется характером исследуемой среды. В качестве цели исследования будем рассматривать модель среды, в качестве которой может быть промышленное или офисное здание. При этом в такой модели среды содержится множество помещений, коридоров и дверей. Из всего многообразия информации о среде можно выделить следующие ее структурные компоненты: *тупики* (содержит одно направление движения), *коридоры* (подразумевает 2 направления движения) и *перекрестки* (подразумевает более чем 2 направления движения). Наиболее информативным структурным компонентом является здесь перекресток, который определим в качестве «маяка» для нашей системы локализации. Это является следствием эвристического предположения о том, что робот, в любом случае, из тупиков и вдоль коридоров движется единственно возможным маршрутом до первого перекрестка, где происходит уже дальнейший выбор маршрута движения для достижения цели. Поэтому, только в данном случае имеет смысл активизировать процесс локализации. Описываемая далее в статье система реализована в виде нейронной сети встречного распространения (НСВР), имеющей в своей основе нейронную сеть Кохонена.

Остальная часть статьи организована следующим образом: во втором разделе содержится структура и описание НСВР для решения задачи локализации; третий раздел приводит метод кодирования и реконструкции положения робота в среде; четвертый раздел содержит экспериментальные результаты, полученные в процессе компьютерного моделирования навигационной системы совместно с системой локализации, а также ее анализ и оценка; и в заключении делаются основные выводы по данной работе.

## 2. Структура нейронной сети для решения задачи локализации.

Перед описанием нейронной сети сначала определим, что понимается под *перекрестком*, описанным в предыдущем разделе. Примером такой структурной компоненты среды может послужить ситуация, показанная на рис. 1.

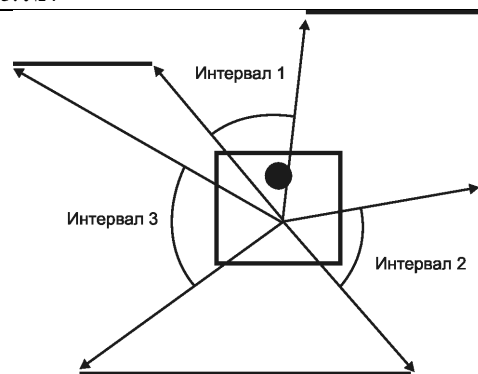


Рис. 1. Пример перекрестка с тремя интервалами.

Под *перекрестком*  $CR=\{I,O\}$  здесь мы будем понимать совокупность интервалов  $I$  и препятствий  $O$  операционной среды в некоторой окрестности робота, для которой мощность множества  $I$ , т.е. количество интервалов, принадлежащих перекрестку, должно превышать 2.

Изображенный на рисунке перекресток содержит три интервала, под которыми будем понимать возможные варианты движения робота. Идентификация интервалов осуществляется реактивной навигационной системой робота в процессе движения к целевой точке. Далее, для задачи локализации все пространство вокруг робота разбивается на  $S=120$  равномерных секторов, ширина каждого из которых равна  $3^\circ$ . Таким образом, все описываемое пространство вокруг робота составляет  $360^\circ$ . Кодирование информации для каждого сектора осуществляется в бинарном виде: 1, если сектор принадлежит интервалу, 0 – в обратном случае. Тем самым получается бинарный вектор, который является входной информацией для НСВР. В качестве выходной информации для данной сети используются декартовы координаты  $(x,y)$ , определяющие местоположение робота для данной ситуации. Общая структура НСВР представлена на рис. 2.

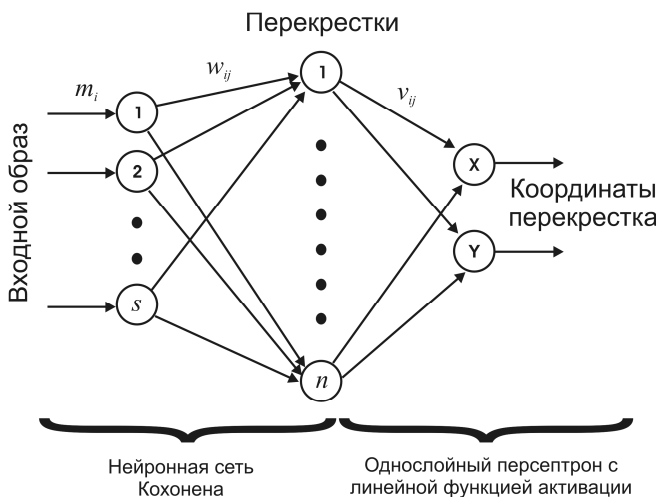


Рис. 2. Структура НСВР для решения задачи локализации.

Совершенно очевидно, что в данном случае осуществляется, так называемая, «привязка» перекрестков к координатной системе среды.

Обучение данной нейронной сети осуществляется в два этапа: при подаче на вход сети бинарного образа перекрестка осуществляется выбор нейрона-победителя промежуточного слоя, исходя из принципа «победитель берет все», а затем происходит обучение однослойной нейронной сети с линей-

ной функцией активации представленной паре декартовых координат.

Для нейронной сети Кохонена, формально это выглядит следующим образом:

$$z_k = \begin{cases} 1, & \text{если } q_k = \min_j(q_j) \\ 0, & \text{иначе} \end{cases} \quad (3)$$

Здесь формула (3) определяет условие выбора нейрона-победителя  $z_k$  промежуточного слоя нейронной сети, если он обладает минимальным евклидовым расстоянием  $q_k$  между входным вектором  $M = (m_i)$  и весовым вектором матрицы  $W=(w_{ij})$ , которое вычисляется следующим образом

$$d_j = \sum_i |m_i - w_{ij}|, \quad (4)$$

а затем нормализуется по следующему правилу

$$q_j = \left( \frac{d_j}{f_j + 1} \right) \cdot (\alpha_j + \beta_j - \alpha_j \cdot \beta_j) + 10^6 (\chi_j \cdot \delta_j). \quad (5)$$

Здесь  $f_j$  – частота выбора  $j$ -того нейрона сети Кохонена, которая подразумевает собой количество побед, одержанных  $j$ -м нейроном в процессе конкуренции. В процессе инициализации сети Кохонена значение частоты приравнивается нулю и с каждой победой оно увеличивается на единицу. Чем больше частота нейрона, тем больше вероятность выбора его в будущем при выполнении определенных условий.

Выражение в скобках в формуле (5) с параметрами  $\alpha_j$  и  $\beta_j$  определяет условие, которое «разрешает»  $j$ -тому нейрону принять участие в конкурентной борьбе. Соответственно, выражение с параметрами  $\chi_j$  и  $\delta_j$  блокирует его участие в конкуренции.

Здесь параметр  $\alpha_j$  определяет попадание входного вектора  $M$  в  $\varepsilon$ -окрестность ( $\varepsilon = s - p$ )  $j$ -того нейрона и описывается следующей формулой

$$\alpha_j = \begin{cases} \text{sign} \left( \frac{(s-p)}{d_j} - 1 \right), & \text{если } d_j > 0 \\ 1, & \text{иначе} \end{cases} \quad (6)$$

где  $p$  ( $0 \leq p < s$ ) – это наименьшая допустимая корреляция между входным вектором  $M$  длины  $s$  и весовым вектором  $j$ -того нейрона. Параметр  $\beta_j$  «разрешает»  $j$ -тому нейрону сети, который ни разу не был победителем, принять участие в конкуренции и определяется следующим образом

$$\beta_j = \text{sign}(0.5 - f_j). \quad (7)$$

Параметр  $\chi_j$  является противоположностью параметра  $\alpha_j$  и «запрещает» нейрону принимать участие в конкуренции, т.к. входной вектор не попадает в его  $\varepsilon$ -окрестность

$$\chi_j = \text{sign} \left( \frac{d_j}{(s-p)} - 1 \right) \quad (8)$$

Параметр  $\delta_j$  является противоположностью параметра  $\beta_j$  и, соответственно, «запрещает» нейрону участвовать в конкуренции по причине того, что он хотя бы однажды уже был победителем

$$\delta_j = \text{sign}(f_j - 0.5) \quad (9)$$

Здесь функция  $\text{sign}(x)$  определяется следующим выражением:

$$\text{sign}(x) = \begin{cases} 1, & \text{если } x \geq 0 \\ 0, & \text{иначе} \end{cases} \quad (10)$$

После того, как  $k$ -тый нейрон сети Кохонена в ходе конкуренции стал победителем, для него выполняется изменение весового вектора и частоты по следующему упрощенному правилу:

$$\begin{aligned} W_k &= M, \\ f_k &= f_k + 1. \end{aligned} \quad (11)$$

Здесь также необходимо добавить, что инициализация весовой матрицы сети Кохонена осуществляется случайным образом в диапазоне от 0 до 1. При этом частота использования каждого нейрона приравнивается к нулю.

Выходной слой сети встречного распространения является однослойной нейронной сетью с линейной функцией активации. Здесь выходное значение ( $x, y$ ) определяется исходя из следующих уравнений:

$$x = \sum_j v_{j1} \cdot z_j, \quad (12)$$

$$y = \sum_j v_{j2} \cdot z_j, \quad (13)$$

где  $V = (v_{ij})$  – весовая матрица выходного слоя. Ее обучение осуществляется после того, как на выходе нейронной сети Кохонена было получено унарное решение, и которое происходит по следующим правилам:

$$v_{j1} = v_{j1} + (x - v_{j1}) \cdot z_j, \quad (14)$$

$$v_{j2} = v_{j2} + (y - v_{j2}) \cdot z_j. \quad (15)$$

Инициализация весовой матрицы  $V$  выходного слоя осуществляется произвольным образом.

### 3. Кодирование и реконструкция положения робота в среде

Прежде чем решать задачу локализации с помощью нейронной сети встречного распространения, необходимо подготовить для нее соответствующую информацию. Это справедливо как для входной информации (бинарного образа перекрестка) и так и для выходной (его местоположение). В идеальном случае желаемая информация для кодирования получается только в том случае, если робот находится в центре этого перекрестка. Но в реальной ситуации, как правило, перекресток обнаруживается раньше, чем робот достигнет его центра (рис.3). При этом вполне естественно, что перекресток может быть обнаружен роботом с разных его сторон. Поэтому необходим здесь механизм, который бы однозначно определял местоположение перекрестка и его бинарный образ.

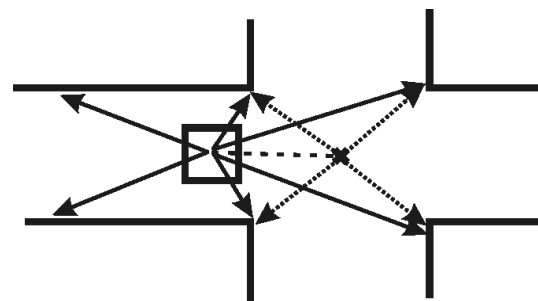


Рис. 3. Пример положения робота при локализации перекрестка.

Исходя из поставленной проблемы, задача локализации решается в пять этапов:

1. Вычисление координаты центра перекрестка. Оно осуществляется путем нахождения геометрического центра масс координат обнаруженных интервалов и текущего

положения робота. При этом интервал «позади» робота, в расчет не принимается.

2. «Перенос» положения робота в вычисленную позицию центра перекрестка с отображением всех интервалов относительно новой системы координат. На рис. 3 такие интервалы изображены пунктирными линиями.
3. Кодирование бинарного образа перекрестка для НСВР с помощью преобразованных интервалов по описанному в разделе 2 правилу.
4. Сохранение или восстановление сохраненных координат перекрестка с помощью нейронной сети встречного пространства
5. Если на НСВР производился опрос сохраненных координат перекрестка, то произвести соответствующие изменения координат текущего положения робота.

#### 4. Экспериментальные результаты

Тестирование системы локализации производилось на базе программной модели навигационной системы для управления автономным мобильным роботом. Параметры НСВР использовались следующие: количество перекрестков  $n = 100$ , параметр корреляции  $p = 115$  (96%) при длине бинарного входного образа  $s = 120$ . Поскольку количество обрабатываемой сенсорной информации для системы локализации сокращено до необходимого минимального уровня, то процесс организации и модификации топологической карты не требует больших вычислительных затрат. В процессе тестирования полученной системы локализации, в условиях сложного лабиринта, был получен фактически стопроцентный результат. К полученным результатам, однако, необходимо сделать несколько комментариев:

1. В процессе моделирования системы рассматривались только идеальные сенсорные устройства – т.е. расстояние до препятствий «определялось» со 100% точностью.
2. Среда моделирования являлась одноагентной – т.е. в ней не присутствовали другие движущиеся объекты.
3. При тестировании системы локализации в реальной среде необходимо выбирать такой параметр корреляции  $p$ , чтобы система была толерантна к случайным помехам и подвижным препятствиям, и, в то же время, не происходило ложное определение координат перекрестка.

#### Заключение

В данной статье была рассмотрена система локализации положения робота в пространстве на базе нейронной сети встречного распространения. Выбор данного подхода был обусловлен эвристическим представлением о характере местности, которое позволяет из всего объема информации выделить для системы локализации только те отличительные признаки среды, по которым робот может быстро и эффективно определить свое местоположение. Реализация системы была построена на технологии топологических карт Кохонена, которые успешно себя зарекомендовали в задачах локализации

УДК 681.3.005.23

*Прожерин И.Г.*

## ПРИБЛИЖЕННЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА, ИСПОЛЬЗУЯ РЕШЕНИЕ ЗАДАЧИ О НАЗНАЧЕНИИ

### 1. ВВЕДЕНИЕ

Задача коммивояжера занимает центральное место среди труднорешаемых задач комбинаторной (дискретной) оптимизации. Все существенные идеи решения таких задач или были первоначально предложены для решения задачи коммивояжера, или, как правило, прошли проверку на этой задаче [1].

и планирования маршрута движения для автономного транспортного средства.

#### Благодарности

Автор благодарит рецензентов за полезные замечания, оказанные в процессе обсуждения и написания данной статьи.

#### СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Heath-Pastore T., Everett H.R. and Bonner K. Mobile Robots for Outdoor Security Applications// American Nuclear Society 8th International Topical Meeting on Robotics and Remote Systems (ANS'99), 25-29 April, 1999 – Pittsburgh, PA, 1999.
2. Sukkarieh S., Nebot E.M. and Durrant-Whyte H.F. Achieving integrity in an INS/GPS navigation loop for autonomous land vehicle applications// IEEE International Conference on Robotics and Automation, 16-20 May 1998, Leuven, Belgium – 1998.
3. Thrun S. Particle Filters in Robotics// In Proceedings of Uncertainty in AI (UAI) – 2002.
4. Chong K.S. and Kleeman L. Mobile robot map building for an advanced sonar array and accurate odometry// International Journal Robotics Research – Jan 1999 – Vol 18, No. 1 – pp.20-36.
5. Chong K.S. and Kleeman L. Accurate odometry and error modelling for a mobile robot// IEEE International Conference on Robotics and Automation, Albuquerque, USA – April 1997 – pp.2783-2788.
6. Laird R.T., Everett H.R., Gilbreath G.A., Inderieden R.S. and Grant K.J. Early User Appraisal of the MDARS Interior Robot// American Nuclear Society 8th International Topical Meeting on Robotics and Remote Systems (ANS'99), 25-29 April, 1999 – Pittsburgh, PA, 1999.
7. Dellaert K.J., Fox D., Burgard W. and Thrun S. Monte Carlo Localization for Mobile Robots// IEEE International Conference on Robotics and Automation (ICRA99) – 1999.
8. Williams S.B., Dissanayake G. and Durrant-Whyte H.F. An Efficient Approach to the Simultaneous Localisation and Mapping Problem// International Conference on Robotics and Automation, May 2002 – Washington, DC. – vol.1 – pp.406-411.
9. Kohonen T. Statistical Pattern Recognition Revisited// Advanced Neural Computers/Eckmiller R. (Editor), Elsevier Science Publishers B.V. (North-Holland) – 1990.
10. Fritzke B. Growing Cell Structures – A Self-organizing Network for Unsupervised and supervised Learning// International Computer Science Institute/ Technical Report 93-026 – Berkeley, California.
11. Zimmer U.R. Self-Localization in Dynamic Environments // IEEE/SOFT International Workshop BIES '95, May 30-31, 1995 – Tokyo, Japan, 1995.

Задача коммивояжера является значимой в области дискретной оптимизации. Пока приемлемое точное решение по критерию времени и памяти возможно только для числа пунктов около 20. После многих неудачных попыток поиска глобального оптимума для большого числа точек становится понятным, что точное решение связано со структурой рас-

*Прожерин Игорь Геннадиевич, ассистент каф. «Информатики и прикладной математики» Брестского государственного технического университета.*

*Беларусь, БГТУ, 224017, г. Брест, ул. Московская, 267.*