

УДК 004.021+004.891

Поиск пути в неизвестном окружении с использованием нейро-нечеткой модели управления

*Гребенюк А. А., студент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Программное обеспечение ЭВМ и информационные технологии»*

*Научный руководитель: Rogozin O.V., к.т.н, доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
bauman@bmstu.ru*

Введение

Задача поиска пути в неизвестном окружении является наиболее популярной в сфере любительской и профессиональной робототехники. В настоящее время большое число рутинных задач становятся автоматизированными, что в некоторых случаях требует разработки автономной системы управления и интеграции алгоритма поиска пути или обхода препятствий в неё. Примерами могут служить, как военные или гражданские автоматические беспилотные летательные аппараты (напр. квадрокоптеры), так и роботы для домашнего использования (напр. роботы-пылесосы). Также задача поиска алгоритма обхода препятствий актуальна для различных конкурсов и олимпиад.

Целью данной работы является разработка алгоритма поиска пути в неизвестном окружении с использованием нейро-нечеткой модели управления.

Ключевыми задачами являются:

- Выбор структуры нейро-нечеткой модели управления и алгоритма обучения нейронной сети
- Составление обучающей выборки, обучение и тестирование на модели окружения и объекта
- Анализ полученных результатов и сравнение по точности и скорости работы данного алгоритма с алгоритмом, использующим нечеткую модель управления.

1. Нечеткая модель управления

При использовании нечеткого подхода к решению поставленной задачи, система управления состоит из набора правил с заданными функциями принадлежности термов. В данной работе рассматривалась совокупность правил, позволяющих произвести

логический нечеткий вывод по алгоритму Мамдани, вида:

$$\begin{array}{l} \text{if} \\ (I_1 \text{ is } T_{\alpha_1}) \wedge \dots \wedge (I_1 \text{ is } T_{\alpha_2}) \wedge \dots \wedge (I_k \text{ is } T_{\beta_1}) \wedge \dots \wedge (I_k \text{ is } T_{\beta_2}) \\ \text{then} \\ (O_1 \text{ is } T_{\xi_1}) \wedge \dots \wedge (O_1 \text{ is } T_{\xi_2}) \wedge \dots \wedge (O_p \text{ is } T_{\theta_1}) \wedge \dots \wedge (O_p \text{ is } T_{\theta_2}), \\ \alpha_i \in \{\tau_\alpha\}, \beta_i \in \{\tau_\beta\}, \xi_i \in \{\tau_\xi\}, \theta \in \{\tau_\theta\}, k \in [1, N_I], p \in [1, N_O], \end{array}$$

где $\{\tau_i\}$ – множество номеров термов i логической переменной, N_I, N_O – число входных и выходных переменных соответственно, T_q – термы входных (I_m) и выходных (O_n) переменных.

Основным недостатком такого подхода является необходимость самостоятельного задания вида функций принадлежности и базы знаний.

1.1 Алгоритм нечеткого вывода

Общий алгоритм нечеткого вывода состоит из следующих основных частей:

- Фаззификация (приведение к нечеткости)
- Агрегация подусловий
- Активизация подзаключений
- Аккумуляция заключений
- Дефаззификация (приведение к четкости)

Т. к. в данной работе использовался нечеткий вывод по алгоритму Мамдани, приведем подробное описание его основных шагов.

На вход поступает вектор чётких значений лингвистических переменных: $A = \{a_1, \dots, a_n\}$. Для удобства описания, пусть выводится значение только одной переменной a_{out} .

- Фаззификация:

$B = \{b_i | b_i = \mu_t(a_i), i \in (1, n)\}$, μ_t – функция принадлежности соответствующего терма, b_i – нечеткое значение лингвистической переменной.

- Агрегация подусловий:

$C = \{c_k | c_k = \min(b_k), k \in (1, m_j), j \in (1, n_r)\}$, m_j – число подусловий в j правиле, n_r – общее число правил

- Активизация подзаключений:

$$\mu_k^*(x) = \min_k(c_k, \mu_k(x)), k \in (1, m_j)$$

- Аккумуляция заключений:

$$\mu_{final}(x) = \max_k(\mu_k^*(x)), k \in (1, m_j)$$

- Дефаззификация(метод центриды):

$$a_{out} = \frac{\int_{x_{min}}^{x_{max}} \mu_{final}(x) \cdot x dx}{\int_{x_{min}}^{x_{max}} \mu_{final}(x) dx}$$

Средняя вычислительная сложность алгоритма Мамдани: $O(n_{out} \cdot n_r \cdot m_{j_{avg}})$, n_{out} – число выходных переменных, $m_{j_{avg}}$ – среднее число подусловий в antecedенте правила.

Исходя из оценки сложности видно, что наибольшим образом скорость вывода по алгоритму Мамдани зависит от числа правил.

2. Нейро-нечеткая модель управления

При использовании нейро-нечеткой системы управления, её структура представляется в виде многослойной нейронной сети, каждый из слоев которой отвечает за определенный шаг алгоритма нечеткого логического вывода. Основное отличие нейро-нечеткой системы управления от нечеткой состоит в том, что при использовании первой нет необходимости в задании и изменении параметров функций принадлежности, т. к. эти данные нейронная сеть извлекает естественным образом в процессе обучения. Также существует возможность извлечения базы знаний в целом, что подразумевает ещё большую гибкость реализации, однако усложняет процесс обучения. Именно этот подход был выбран для реализации поставленной задачи.

Для извлечения базы знаний из обучающей выборки используется самоорганизующаяся карта Кохонена, структура которой представлена на рис. 1.

G.N. Marichal et al. / Fuzzy Sets and Systems 124 (2001) 171–179

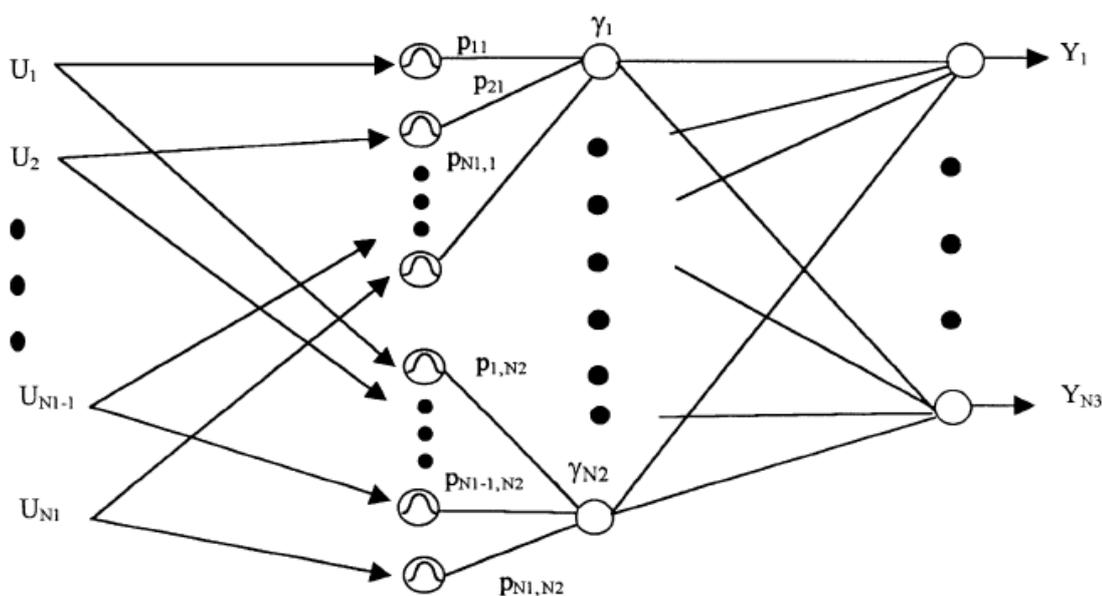


Рис. 1. Структура нейро-нечеткой сети

Описание структуры:

- N_i, N_r, N_o — число входных и выходных переменных, и число правил соответственно, где $i \in [1, N_i]; j \in [1, N_r]; k \in [1, N_o]$

- Фаззификация: $p_{ij} = \frac{1}{2\sigma\sqrt{2\pi}} \cdot e^{-\frac{(u_i - m_{ij})^2}{2\sigma_{ij}^2}}$

- Скрытый слой: $\gamma_j = \min_i(p_{ij})$

- Дефаззификация: $y_k = \frac{\sum_{j=1}^{N_r} sv_{jk} \cdot \gamma_j}{\sum_{j=1}^{N_r} \gamma_j}$

$m_{ij}, \sigma_{ij}, sv_{jk}$ - математическое ожидание, среднеквадратичное отклонение функции принадлежности и оценка выходного значения k нейрона относительно j нейрона в скрытом слое соответственно.

- Параметры $m_{ij}, \sigma_{ij}, sv_{jk}$ являются варьируемыми на завершающем этапе обучения карты Кохонена.

2.1 Алгоритм обучения

- Инициализация весов карты:

Может быть произведена случайным образом, посредством выбора случайного наблюдения или иным способом.

- Цикл:

1. Выбор очередного наблюдения $X(t)$

2. Нахождение ближайшего вектора весов в некоторой метрике (BMU/Winner)

$$\|X(t) - W_{j_{winner}}\| \leq \|X(t) - W_i\|, \forall i \in [1, N]$$

3. Изменение найденного и соседних векторов с целью приближения к наблюдению

$$W_j(t+1) = W_j(t) + \alpha(t) \cdot e^{-\frac{\|j - j_{winner}\|^2}{2\sigma(t)^2}} \cdot (X(t) - W_j(t)),$$

где $0 < \alpha(t) < 1$ - скорость обучения (монотонно убывает), $\sigma(t)$ - функция, уменьшающая число соседей с каждой итерацией

К сожалению, полученных таким образом значения весов недостаточно точно аппроксимирует базу знаний. Более того, результат применения данного алгоритма обучения

сильно зависит от выбранных параметров $\alpha(t)$ и $\sigma(t)$.

Для уточнения базы знаний, применим следующий алгоритм:

- Получение начального приближения базы знаний

- Оптимизация числа правил:

$$\|W_j\| := \frac{(\|W_j\| + \|W_i\|)}{2} \forall i \leq j, j \in [1, N]$$

- Подстройка параметров:

Обучение методом нормализованного градиентного спуска, выбранный из-за высокой скорости работы.

2.2 Алгоритм обучения методом градиентного спуска (NLMS)

На завершающем этапе обучения карты Кохонена необходимо произвести подстройку всех варьируемых параметров (параметров функций принадлежности). Для этого используется алгоритм нормализованных наименьших квадратов (NLMS). Нормирование необходимо для сходимости этого метода обучения.

Алгоритм:

- Вычисление ошибки карты: $E = \frac{1}{2} \sum_1^{N_{output}} (y_k - \bar{y}_k)^2$
- Подстройка параметров: $m(t + 1) = m(t) - \frac{\mu \cdot \partial E}{\partial m(t)}$, μ – скорость обучения.

3. Модель объекта

Объект не обладает качественной информацией об окружающей среде, т. е. не может определить форму препятствия, находящегося на его пути, для работы алгоритма поиска пути необходимо оснастить его датчиками.

Очевидно, что большое количество датчиков сильно затруднит логическое описание работы алгоритма, в то время как слишком малое приведет к невозможности принятия решения. Таким образом выберем минимально необходимое число датчиков — 4.

Опишем модель объекта:

- Объект негабаритный
- Может двигаться только вперед с постоянной скоростью
- Имеет дальнометры слева, справа и по центру.
- Имеет компас, т. е. может определять угол по отношению к направлению движения на цель
- Может изменять угол движения произвольным образом

Такая модель достаточна для описания подвижного объекта в неизвестном двумерном окружении и не избыточна. Она имеет ряд серьезных ограничений, которые, однако не являются недостатками: наличие «мертвых зон» у датчиков на углах объекта

компенсируется его не габаритностью, а отсутствие возможности двигаться назад — возможностью поворота на угол более 90°.

4. Алгоритм поиска пути

Исходя из выбора структуры нейро-нечеткой сети и алгоритма её обучения, запишем схему алгоритма поиска пути в неизвестном окружении, представленную на рис. 2.

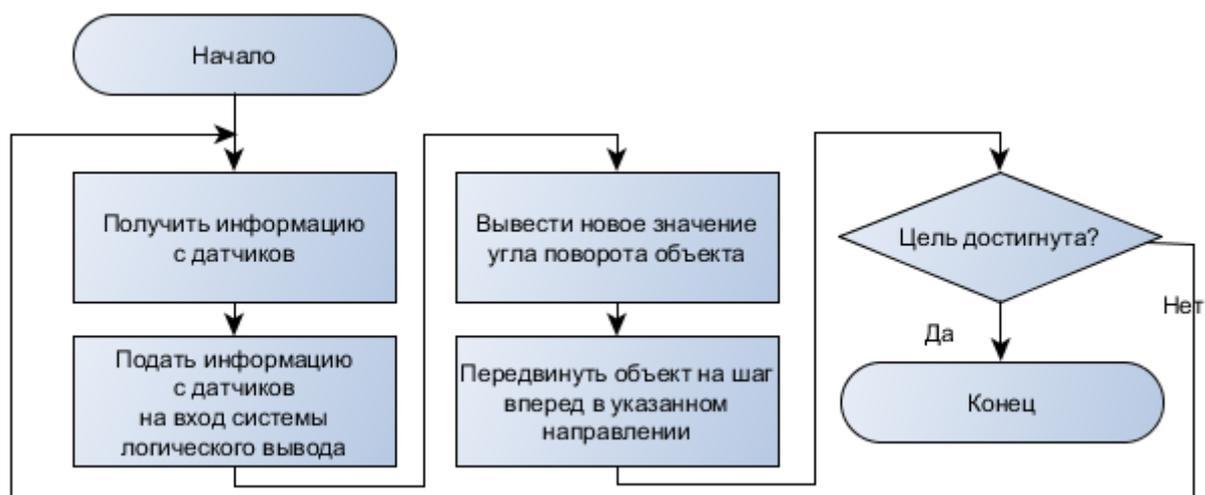


Рис. 2. Схема алгоритма поиска пути в неизвестном окружении

5. Результаты работы ПО

В результате тестирования разработанного алгоритма на модели окружения были получены траектории движения объекта с использованием нечеткой модели управления (см. рис. 3) и нейро-нечеткой (см. рис. 4).

Данные по исследованию на скорость представлены в таблице 1.

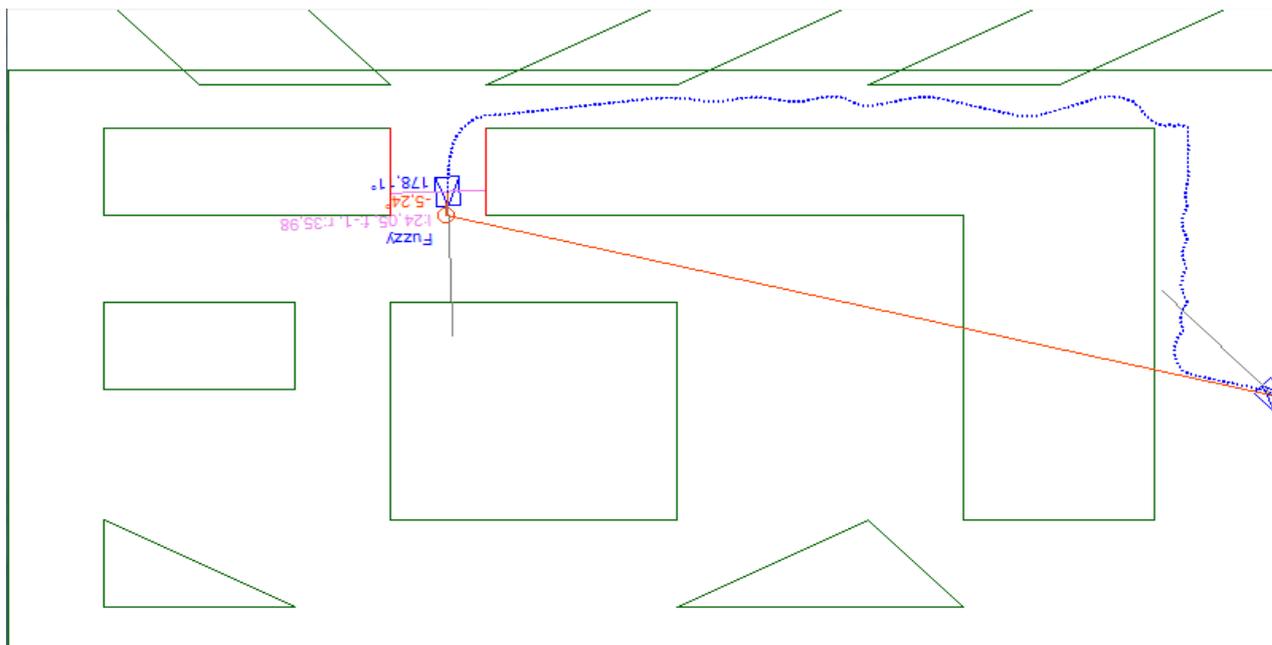


Рис. 3. Траектория, полученная с применением нечеткой модели управления

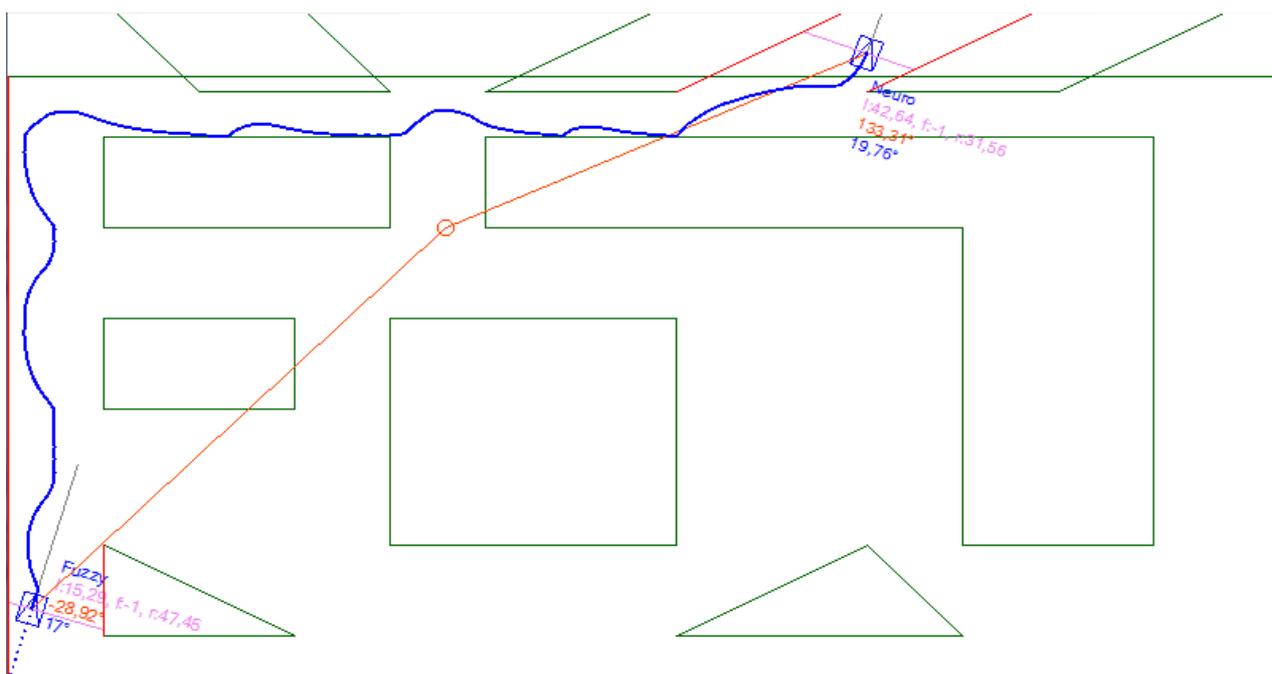


Рис. 4. Траектория, полученная с применением нейро-нечеткой модели управления

Время выполнения одного шага алгоритма поиска пути(мс)	
Нейро-нечеткая модель	Нечеткая модель
1.5	600

Заключение

В ходе работы был разработан алгоритм поиска пути в неизвестном окружении с использованием нейро-нечеткой модели управления. Для проведения сравнительного анализа также была разработана модель нечеткой системы управления на основе нечеткого вывода Мамдани. Сравнительный анализ поведения двух моделей показывает, что нечеткая модель выигрывает в точности работы, однако существенно проигрывает в скорости. Использование самоорганизующейся карты Кохонена в основе нейро-нечеткой модели управления позволяет избежать необходимости задания базы знаний и функций принадлежности термов вручную, однако вносит погрешность в точность. Также, нейросеть Кохонена является сложной в обучении, т. к. является сильно зависимой от начального приближения. Ошибки в обучении видны на рис.4, где происходит столкновение с препятствием.

Дальнейшая работа в этом направлении ведется в повышении точности нейро-нечеткой модели управления с использованием других нейронных сетей в основе модели. Также, следующим шагом является расширение постановки задачи до задачи одновременного построения карты местности и навигации(SLAM), с использованием разработанного алгоритма обхода препятствий.

Список литературы

1. Marichal G.N. et al. Obstacle avoidance for a mobile robot: A neuro-fuzzy approach. // Fuzzy Sets and Systems. 2001. № 124.2. P. 171 –179. DOI: [10.1016/S0165-0114\(00\)00095-6](https://doi.org/10.1016/S0165-0114(00)00095-6).
2. Nedjah N., de Macedo Mourelle L. Fuzzy Systems Engineering: Theory and Practice. Rio De Janeiro: Springer. 2005. 181 p.
3. Spenko M., Iagnemma K., Dubowsky S. High Speed Hazard Avoidance for Mobile Robots in Rough Terrain. // Proceedings of the SPIE Conference on Unmanned Ground Vehicle Technology VI. 2004. P. 127-139.
4. Atomicus. Алгоритм Мамдани в системах нечеткого вывода. Режим доступа: <http://habrahabr.ru/post/113020/> (дата обращения 11.06.2014).