

АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ НЕЧЕТКИХ ПРАВИЛ ДЛЯ УПРАВЛЕНИЯ МОБИЛЬНЫМ РОБОТОМ С ГУСЕНИЧНЫМ ШАССИ НА ОСНОВЕ ЧИСЛОВЫХ ДАННЫХ

© 2017 Е.А. Пташко, В.И. Ухоботов

*Челябинский государственный университет
(454001 Челябинск, ул. Братъев Каширинных, д. 129)*

E-mail: eugene.ptashko@gmail.com, ukh@csu.ru

Поступила в редакцию: 08.06.2017

В данной работе рассматривается актуальная на данный момент проблема генерации набора нечетких правил для системы нечеткого вывода Мамдани на основе числовых данных, получаемых в процессе обучения управляемой системы. Предлагаемый в статье подход к решению данной проблемы базируется на алгоритмах четкой и нечеткой кластеризации, таких как алгоритм горной кластеризации и алгоритм Густафсона–Кесселя. Он позволяет значительно упростить процесс формирования набора нечетких правил и минимизировать участие человека в этом процессе, позволяя автоматически подбирать количество правил, а также определять все необходимые параметры каждого из них. Для реализации предложенного подхода были написаны две компьютерные программы. Первая из них собирает числовые данные при управлении человеком гусеничной тележкой. На основе собранных данных эта программа строит базу нечетких правил управления гусеничным шасси. Эта база нечетких правил и ее компьютерная реализация в дальнейшем используется во второй программе для автоматизированного управления мобильным роботом на гусеничном шасси на плоскости при помощи изменения силы тяги каждой из гусениц в зависимости от положения цели, к которой робот должен приблизиться на заданное расстояние.

Ключевые слова: система нечеткого вывода, нечеткая кластеризация, компьютерная реализация.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Пташко А.Е., Ухоботов В.И. Автоматическая генерация базы нечетких правил на основе числовых данных для управления мобильным роботом с гусеничным шасси // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 3. С. 60–72. DOI: 10.14529/cmse170304.

Введение

Мобильные роботы на гусеничном шасси применяются в условиях плохой проходимости. Их уравнения движения существенно отличаются от уравнений движения роботов на колесном шасси. Существуют разные методы, с помощью которых осуществляется синтез управления мобильными роботами на гусеничном шасси. Например, используются принципы и методы синергетической теории управления [1–4]. В настоящее время большое применение приобрели системы управления, для работы которых используется теория нечетких множеств [5]. Такая популярность объясняется тем, что они позволяют строить управление для задач, где по разным причинам затруднено построение точных математических моделей и алгоритмов работы с ними. Несомненно, нечеткие системы управления [6–11] по оптимальности своей работы уступают точным математическим алгоритмам, однако, зачастую построение таких алгоритмов не всегда возможно, либо их вычислительная сложность слишком велика, что делает целесообразным применение нечеткой логики.

Однако есть один очень важный момент, который затрудняет использование систем управления на основе нечеткого вывода — наборы таких правил необходимо формировать вручную. Это очень трудоемкий процесс, и порой он требует от одного или нескольких экспертов подбора огромного числа числовых параметров. Поэтому актуальной является задача минимизации участия человека, а в идеале, полной автоматизации построения базы нечетких правил и подбора необходимых параметров. Следует отметить, что существуют работы, в которых предлагаются решения такой задачи [12].

Целью работы является исследование и реализация подхода, при котором человек будет принимать участие в составлении базы правил опосредованно. Реализация подхода выполнена на примере задачи построения системы автоматизированного управления гусеничной тележкой.

Статья организована следующим образом. В первом разделе описывается задача, на примере которой был реализован предложенный подход. Во втором разделе приводится описание непосредственно самого подхода, обосновывается причина его использования. Третий раздел посвящен аспектам компьютерной реализации построения набора нечетких правил и их дальнейшего использования. В заключении приводятся основные результаты работы.

1. Описание задачи

Имеется гусеничная тележка (рис. 1) и некоторый объект, уравнение движения которого неизвестно (цель). Задача состоит в том, чтобы построить автоматизированное управление гусеничной тележкой на плоскости при помощи системы нечеткого вывода Мамдани [8], которое бы позволяло ему догнать цель, то есть приблизиться к ней на некоторое заданное расстояние. В данной задаче направление движения цели задается человеком, а управление тележкой осуществляется за счет изменения мощности левой и правой гусеницы. Координаты (x, y) гусеничной тележки на плоскости и угол θ поворота его оси относительно оси x изменяются согласно следующей системе уравнений [1, 2, 4]:

$$\left\{ \begin{array}{l} \dot{x} = V \cos \theta \\ \dot{y} = V \sin \theta \\ \dot{\theta} = \omega \\ m \dot{V} = u_1 - F_{\text{тр}} \\ I_z \dot{\omega} = -\frac{B u_2}{2} - M_r \\ u_1 = f^r + f^l \\ u_2 = f^r - f^l, \end{array} \right. \quad (1)$$

где V — линейная скорость, ω — угловая скорость, B — расстояние между гусеницами, m — масса тележки, I_z — момент инерции относительно центра тележки, f^r, f^l — силы тяги правой и левой гусеницы соответственно, M_r — момент сопротивления повороту. Сила трения $F_{\text{тр}}$ в (1) вычисляется по формуле:

$$F_{\text{тр}} = k g m \quad (2)$$

где k — коэффициент трения поверхности, по которой движется тележка, g — ускорение свободного падения. Момент сопротивления повороту вычисляется следующим образом:

$$M_r = \frac{\mu m g l}{4} \quad (3)$$

где μ — коэффициент сопротивления повороту тележки, который зависит от характеристик поверхности, по которой движется тележка, а также от конструкции ее гусеницы, l — длина гусеницы.

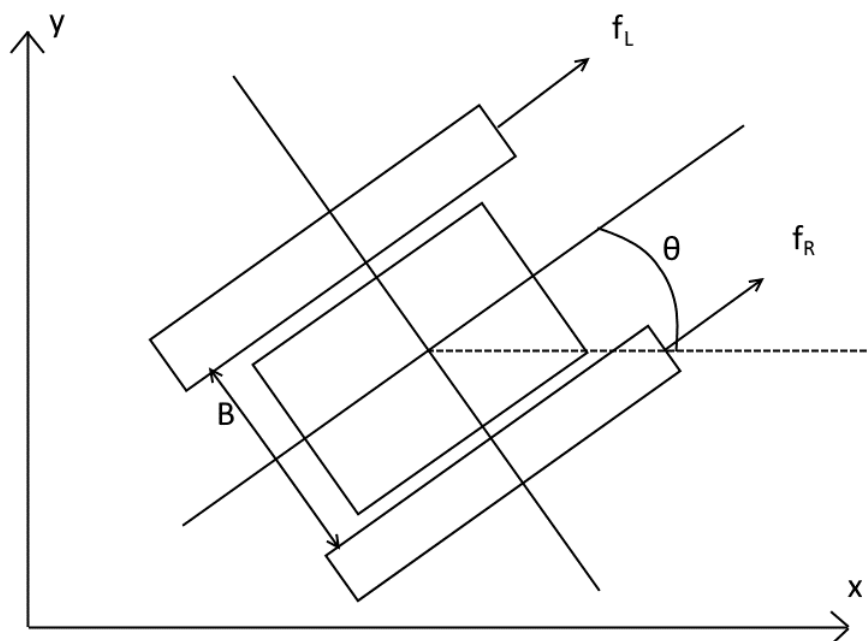


Рис. 1. Графическое представление положения гусеничной тележки на плоскости

Следует отметить, что в работе [13] приведены более полные уравнения движения, для рассматриваемых в статье управляемых систем.

2. Построение набора нечетких правил

Как уже было сказано выше в статье, набор нечетких правил строится на основе набора числовых данных. Получаются эти данные следующим путем: один человек управляет направлением цели, второй в это время, управляя силой тяги каждой из гусениц тележки, пытается сделать так, чтобы тележка приблизилась на заданное расстояние. В равные промежутки времени берутся координаты центра тележки, угол его поворота относительно цели и расстояние до цели. Таким образом, обучающая выборка будет выглядеть следующим образом:

$$\left\{ \left\{ f_1^r, f_1^l, a_1, d_1 \right\}, \dots, \left\{ f_n^r, f_n^l, a_n, d_n \right\} \right\}, \quad (4)$$

где f_n^r, f_n^l — силы тяги правой и левой гусеницы, a_i — угол поворота тележки относительно цели, d_i — расстояние от центра тележки до цели, n — размер обучающей выборки. Чтобы минимизировать фактор человеческой ошибки, в выборку попадают только те моменты времени, в которые расстояние до цели было меньше, либо равно расстоянию до цели в предыдущий момент времени.

Построенные в трехмерном пространстве наборы точек $\{f_1^r, a_1, d_1\}, \dots, \{f_n^r, a_n, d_n\}$ и $\{f_1^l, a_1, d_1\}, \dots, \{f_n^l, a_n, d_n\}$ будут выглядеть примерно так, как показано на рис. 2, а именно, будет наблюдаться концентрация обучающих примеров в некоторых областях пространства. Исходя из данного наблюдения, можно сделать вывод о том, что между

входными (a_i, d_i) и выходными (f_n^r, f_n^l) аргументами существует некоторая зависимость. Если говорить в терминах нечетких множеств, то это означает, что существует некоторое правило вывода, связывающее терм-множества аргументов, подаваемых на вход, с терм-множествами, получаемыми в результате. Это означает, что задача построения набора нечетких правил сводится к разбиению обучающей выборки на области с наиболее высокой концентрацией обучающих примеров (на кластеры).

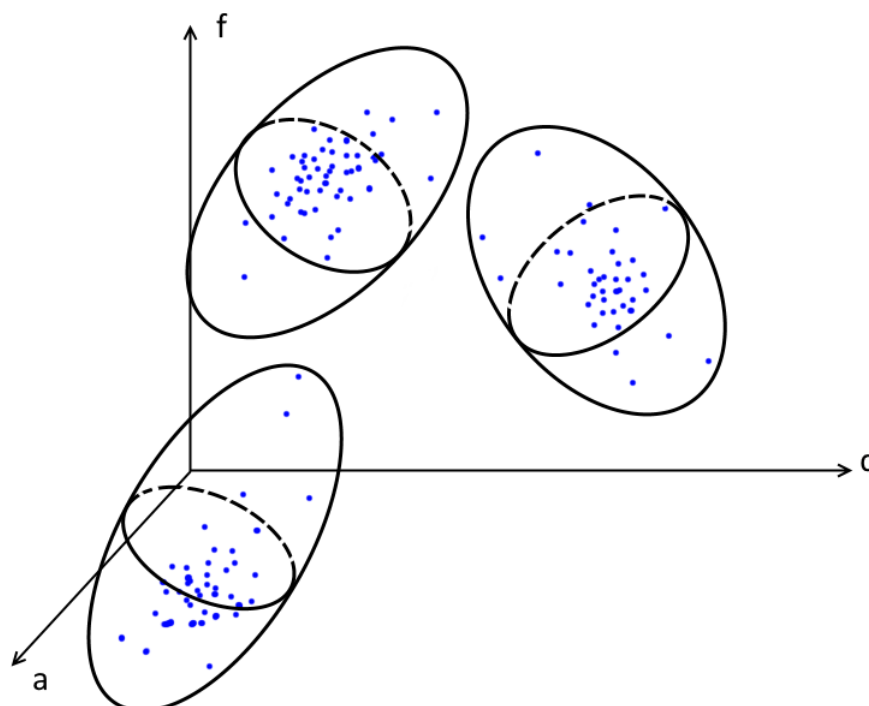


Рис. 2. Схематичное изображение обучающей выборки в пространстве

Существуют различные алгоритмы разбиения пространства на кластеры [14]. Все они делятся на алгоритмы четкой и нечеткой кластеризации. При четкой кластеризации результатом работы алгоритма является набор непересекающихся кластеров, при этом каждый элемент пространства может находиться только в одном кластере. При нечеткой кластеризации получающиеся кластеры могут пересекаться друг с другом, а каждый элемент пространства находится в каждом кластере с некоторой степенью принадлежности к нему. Очевидно, что для разбиения обучающей выборки, подходит именно нечеткая кластеризация, так как только она позволит выделить кластеры в виде нечетких множеств.

Существует несколько алгоритмов нечеткой кластеризации. Самые популярные из них — это алгоритм нечетких s -средних и алгоритм Густафсона—Кесселя [15]. Для решения задачи в статье используется второй алгоритм, так как он позволяет выделять кластеры в форме гиперэллипсов, в то время, как с помощью первого алгоритма получаются кластеры в виде гиперсфер. Кластеры в форме гиперэллипсов наиболее точно отображают связь между компонентами входа и выхода. Однако алгоритм Густафсона—Кесселя также имеет свои недостатки. Первый недостаток заключается в том, что для его работы необходимо знать количество кластеров. Второй недостаток связан с тем, что этот алгоритм очень ресурсоемкий. Для того, чтобы минимизировать проблемы с производительностью, необходимо предварительно найти приблизительные значения центров кластеров. Такие

значения ищутся в статье при помощи алгоритма горной кластеризации [16], так как он не зависит от числа кластеров и способен сам их определять.

Управление строится независимо для левой и правой гусеницы, поэтому дальнейшее описание работы алгоритмов пойдет на примере правой гусеницы. Для левой гусеницы управление строится аналогичным образом.

2.1. Горная кластеризация

Алгоритм горной кластеризации [16] состоит из трех шагов. На первом шаге определяются точки, которые потенциально могут быть центрами кластеров. В задаче о центрах кластеров изначально за потенциальные центры принимаются все объекты обучающей выборки. Обозначим их как $z_i = \{f_i^r, a_i, d_i\}$, $i = \overline{1, n}$.

Второй шаг заключается в расчете так называемого потенциала каждой из точек, который показывает возможность того, что в ее окрестности может быть сформирован кластер. Точки, в окрестностях которых наиболее плотно расположены другие объекты выборки, обладают наиболее высоким потенциалом, который рассчитывается по формуле

$$P(z_i) = \sum_{k=1}^n \exp(-\alpha \cdot D(z_i, x_k)), \quad (5)$$

$$D(z_i, x_k) = \sqrt{\|z_i - x_k\|^2}.$$

В этой формуле $P(z_i)$ — потенциал точки z_i , α — положительная константа, которая характеризует масштаб расстояния между объектами и принимается равной единице, деленной на среднее расстояние между объектами.

На третьем шаге в качестве первого центра кластера назначается точка с наибольшим потенциалом, которую назовем вершиной. В дальнейшем для оставшихся потенциальных центров кластеров потенциалы пересчитываются, без учета влияния уже найденной вершины по формуле

$$P_2(z_i) = P_1(z_i) - P_1(v_1) \exp(-\beta \cdot D(z_i, v_1)). \quad (6)$$

Здесь v_1 — вершина, найденная на первом этапе алгоритма, $P_1(z_i)$ — потенциал точки, рассчитанный на втором шаге, $P_2(z_i)$ — новый потенциал точки.

Дальнейший процесс пересчета потенциалов и выделения центров кластеров продолжается итерационно до тех пор, пока значение находимых вершин превышает некоторый заданный порог. При решении задачи в качестве порога было выбрано значение потенциала первой вершины $P_1(v_1)$, деленное пополам.

Полученные при помощи алгоритма горной кластеризации центры кластеров, как было сказано выше, используются как приближительные начальные значения центров кластеров в алгоритме Густафсона—Кесселя.

2.2. Алгоритм Густафсона—Кесселя

Алгоритм Густафсона—Кесселя [15] является расширенным вариантом стандартного алгоритма нечетких с-средних с использованием адаптивной нормы для каждого кластера, что позволяет оптимизировать с его помощью не только координаты центров кластеров, но норм-порождающие матрицы всех кластеров. Это позволяет выделять кластеры произвольной формы.

Целью алгоритма является такое размещение центров кластеров, которое является решением следующей задачи:

$$\sum_i \sum_j u_{ij}^m d^2(x_j, c_i) \rightarrow \min, \quad 0 \leq u_{ij} \leq 1, \quad \sum_{i=1}^K u_{ij} = 1 \text{ для всех } j = \overline{1, n} \quad (7)$$

Здесь K — число кластеров, c_i — центр i -го кластера, m — степень «размытости» кластера (обычно принимается равной 2), u_{ij} — степень принадлежности j -го вектора i -му кластеру, а расстояние между вектором x_j и c_i определяется как

$$d(x_j, c_i) = \|x_j - c_i\| = \sqrt{(x_j - c_i)^T A_i (x_j - c_i)}.$$

Норм-порождающей в данном случае является симметричная положительно определенная матрица A_i , которая на начальном этапе заполняется случайными значениями из интервала $[0, 1]$, а начальные значения c_i , как уже упоминалось ранее, вычисляются при помощи алгоритма горной кластеризации.

С учетом применения метода неопределенных множителей Лагранжа [15] решение задачи (7) сводится к следующему алгоритму.

На первом шаге формируется матрица коэффициентов принадлежности векторов к кластерам

$$u_{ij} = \frac{1}{\sum_{k=1}^K \left(\frac{d^2(x_j, c_i)}{d^2(x_j, c_k)} \right)^{\frac{1}{m-1}}}. \quad (8)$$

Далее определяются новые координаты центров кластеров по формуле

$$c_i = \frac{\sum_{j=1}^N x_j u_{ij}^m}{\sum_{j=1}^N u_{ij}^m}. \quad (9)$$

Затем для каждого центра кластера формируется матрица ковариаций

$$s_i = \sum_{j=1}^N u_{ij}^m (x_j - c_i) (x_j - c_i)^T. \quad (10)$$

С ее помощью находятся новые масштабирующие матрицы

$$A_i = \sqrt[N]{\det(S_i)} \cdot S_i^{-1}. \quad (11)$$

Если изменения центров кластеров и матриц ковариации не превышают заранее заданного порога, то работа алгоритма считается завершенной. В противном случае их значения пересчитываются, начиная с первого шага.

2.3. Формирование набора нечетких правил

Разбив обучающую выборку на нечеткие кластеры и найдя центры таких кластеров, строится набор нечетких правил. Для каждого кластера все его точки проецируются на каждую ось, соответствующую входным аргументам, и значениям, получаемым на выходе (рис. 3). Для каждой такой проекции строится треугольное нечеткое число (l, c, r) [10]. Его ядром c является точка с координатой соответствующего центра кластера на данной оси. Числом l будет наименьшая координата вектора из обучающей выборки на

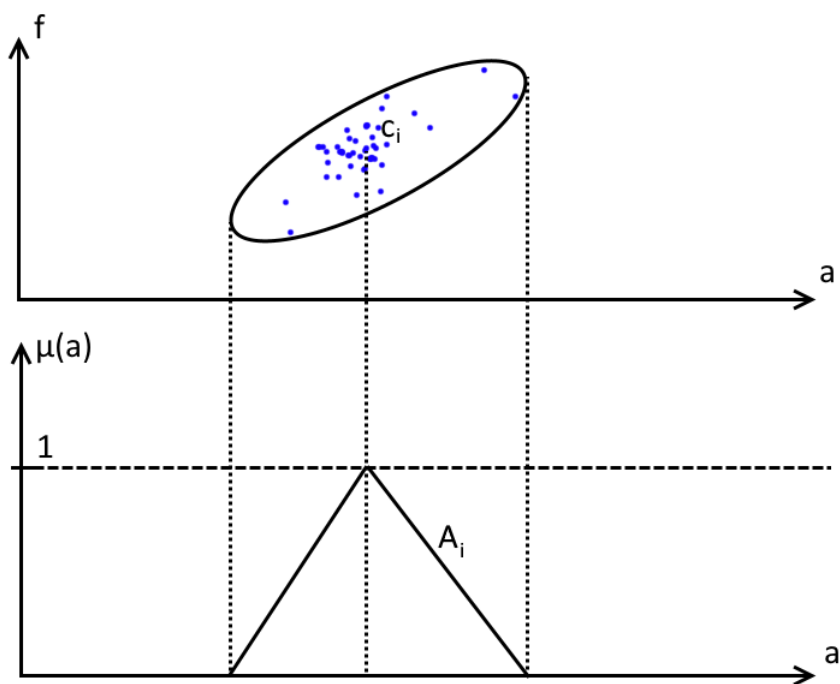


Рис. 3. Проекция кластера на ось и формирование функции принадлежности

рассматриваемой оси, попавшего в кластер. Аналогично, числом r будет наибольшая координата вектора из кластера по оси.

Построив таким образом функции принадлежности для каждого кластера, можно сформировать правила. Входным параметрам a_i, d_i и выходу f_i^r соответствуют терм-множества A_i, B_i, C_i с треугольными функциями принадлежности, найденными после проекции кластеров на соответствующие оси. Правило будет выглядеть следующим образом:

$$\text{если } a_i \text{ есть } A_i, d_i \text{ есть } B_i, \text{ то } f_i^r \text{ есть } C_i. \quad (12)$$

Возможны случаи, когда после проекции кластеров на какую-либо из осей, параметры получаемых функций принадлежности отличаются незначительно. В таком случае получаемые терм-множества объединяются. В качестве ядра берется среднее значение всех таких точек в объединяемых функциях, а в качестве чисел l и r — минимальное и максимальное значение из левых и правых границ соответственно. К примеру, терм-множество A_1 с параметрами функции принадлежности $(0; 5; 10)$ и терм-множество A_2 $(0.05; 4.96; 10.1)$ будут объединены в новое множество A_3 $(0; 4.98; 10.1)$.

3. Программная реализация

Решение задачи управления гусеничной тележкой было реализовано в виде двух программ, написанных на языке C# в среде разработки Microsoft Visual Studio 2015. Графический интерфейс обеих программ одинаков и показан на рис. 4.

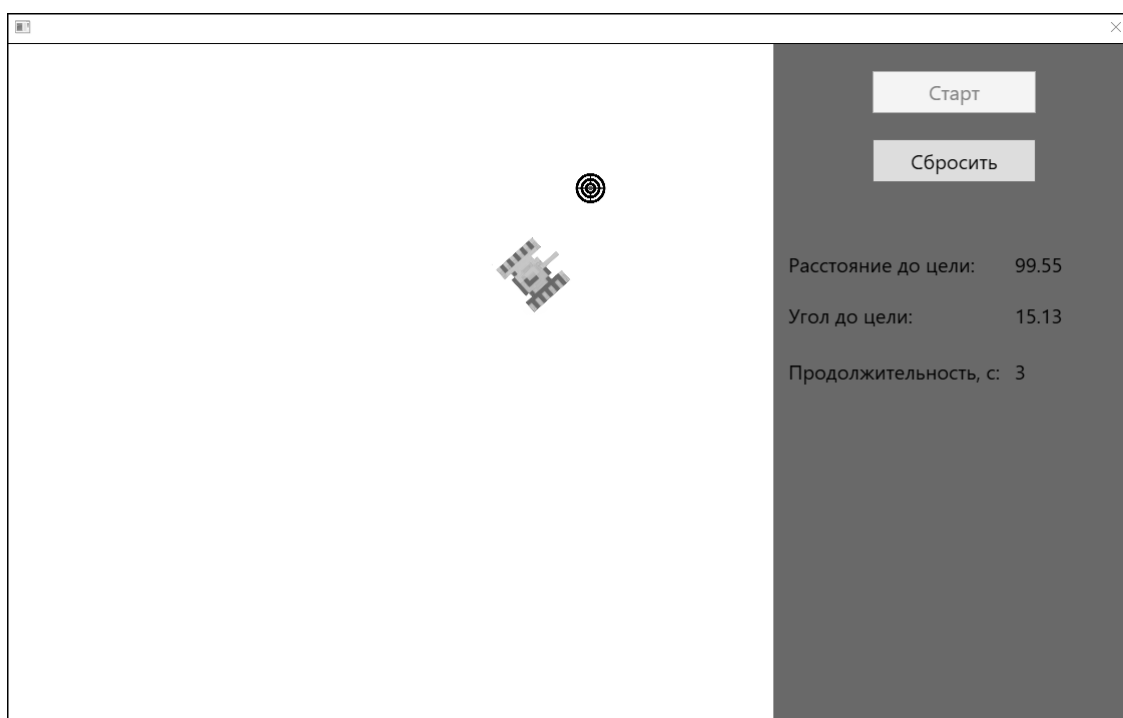


Рис. 4. Графический интерфейс реализованных программ

Первая программа предназначена для формирования обучающей выборки и формирования набора базы нечетких правил. Сбор данных осуществляется следующим образом: первый игрок при помощи компьютерной мыши задает направление цели, второй игрок в этот момент при помощи мини джойстиков на контроллере Microsoft Xbox 360 управляет силой тяги каждой из гусениц тележки таким образом, чтобы максимально приблизиться к цели. Через равные промежутки времени программа записывает в массив такие параметры, как сила тяги левой и правой гусеницы, расстояние от центра тележки до цели, угол поворота тележки относительно цели. По истечении заданного времени программа формирует на основе собранного массива данных базу нечетких правил и записывает ее в особом формате в txt файл. Пример построенных терм-множеств для нечеткой переменной angle изображен на рис. 5. Angle, distance, leftTrackPower, rightTrackPower — названия нечетких переменных (рис. 6); very_small, big и др. — получившиеся терм-множества с параметрами соответствующих им функций принадлежности; if (angle is very_small) and (distance is big) then (rightTrackPower is small) — получившиеся правила нечеткого вывода (рис. 7).

Вторая программа принимает на вход сгенерированный txt файл, который в дальнейшем использует для автоматизированного управления гусеничной тележкой. После нажатия на кнопку "Старт" целью все так же управляет человек, задавая направление движения посредством компьютерной мыши, а силой тяги гусениц управляет программа при помощи полученной системы нечеткого вывода.

Тестирование второй программы показало, что траектория движения тележки, управляемой при помощи сформированных нечетких правил не оптимальна, но вполне приемлема. Объясняется это тем, что правила строятся на основе данных, полученных при управлении тележки человеком, которое также далеко от идеального.

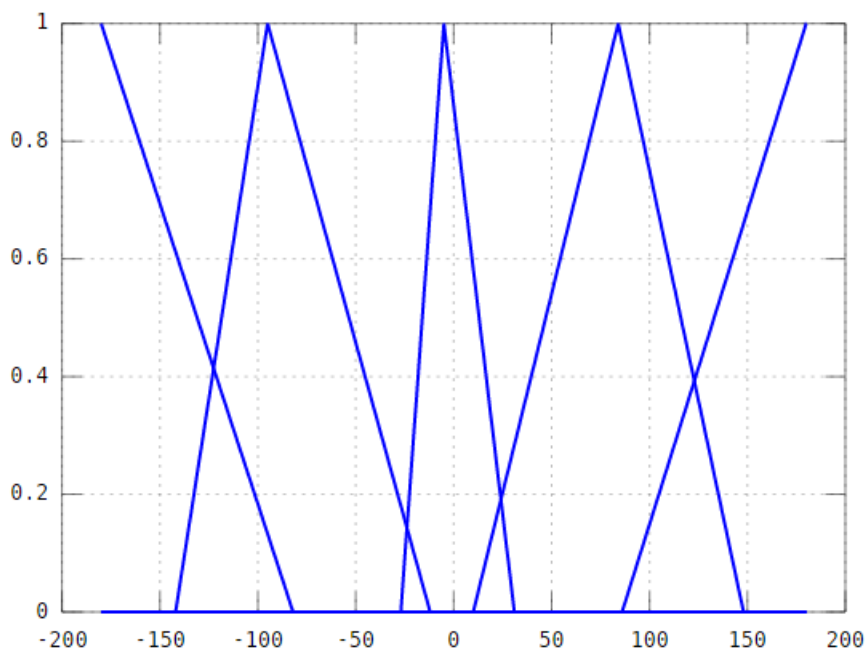


Рис. 5. Графическое изображение термов переменной angle

```

angle, -180, 180
very_small
-180, -180, -82.3621637063856
small
-142.637836293614, -95.6378362936144, -12.3621637063856
mean
-27.6378362936144, -5.63783629361439, 31.6378362936144
big
10.3621637063856, 84.3621637063856, 148.637836293614
very_big
86.3621637063856, 180, 180
    
```

Рис. 6. Пример сформированных параметров для терма angle

```

if (angle is very_small) and (distance is big) then (rightTrackPower is small)
if (angle is mean) and (distance is small) then (rightTrackPower is very_big)
if (angle is very_small) and (distance is small) then (rightTrackPower is small)
if (angle is small) and (distance is small) then (rightTrackPower is small)
if (angle is big) and (distance is small) then (rightTrackPower is very_big)
if (angle is very_big) and (distance is big) then (rightTrackPower is very_big)
if (angle is mean) and (distance is big) then (rightTrackPower is very_big)
if (angle is very_big) and (distance is small) then (rightTrackPower is very_big)
if (angle is big) and (distance is big) then (rightTrackPower is very_big)
    
```

Рис. 7. Пример сформированных нечетких правил для переменной rightTrackPower

Следует отметить, что функциональность, которая отвечает за формирование базы правил и построение управления на ее основе были реализованы в отдельной библиотеке, что означает возможность ее применения для решения схожих задач.

Заключение

В работе исследован подход для формирования набора нечетких правил по набору числовых данных.

Работа данного подхода была продемонстрирована на примере решения автоматизированного управления гусеничной тележкой. Для этого было реализовано две программы, одна из которых позволяет человеку получить базу нечетких правил без непосредственного ручного набора этих правил и подбора необходимых параметров.

Исследованный подход позволяет минимизировать участие человека при формировании системы управления, основанной на нечетком выводе, так как позволяет автоматически определять необходимое количество нечетких правил и подбирать соответствующие им параметры. Управление, осуществляемое при помощи формируемых нечетких правил, не оптимально, однако, вполне может быть использовано, так как достаточно точно соответствует управлению со стороны человека.

В дальнейшем планируется изучение проблемы построения набора правил с функциями принадлежности разного вида (трапециевидного, сигмоидального и др.), а также разработка подходов, которые позволили бы оптимизировать параметры уже построенного набора правил.

Работа выполнена при поддержке гранта Фонда перспективных научных исследований ФГБОУ ВО «Челябинский государственный университет» (2017 г.).

Литература

1. Вонг Дж. Теория наземных транспортных средств. М.: Машиностроение, 1982. 284 с.
2. Забавников Н.А. Основы теории транспортных гусеничных машин. М.: Машиностроение, 1975. 448 с.
3. Колесников А.А. Синергетическая теория управления. М.: Энергоатомиздат, 1994. 344 с.
4. Скляр А.А., Скляр С.А. Синергетическое управление мобильным роботом с гусеничным шасси // Известия ЮФУ. Технические науки. 2011. № 6. С. 118–125
5. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976. 161 с.
6. Castro J.L. Fuzzy Logic Controllers are Universal Approximators // IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics. Vol. 25. 1995. P. 629–635. DOI: 10.1109/21.370193
7. Kosko B. Fuzzy Systems as Universal Approximators // IEEE Transactions on Computers. 1994. Vol. 43, No. 11. P. 1329–1333. DOI: 10.1109/12.324566
8. Mamdani E.H. Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis // IEEE Transactions on Computing. 1977. Vol. 26. P. 1182–1191. DOI: 10.1109/TC.1977.1674779
9. Wang L.X. Fuzzy Systems are Universal Approximators // Proceedings of IEEE International Conference on Fuzzy Systems (San Diego, CA, USA, March, 8 – March, 12, 1992). IEEE Xplore Press, 1992. P. 1163–1169. DOI: 10.1109/FUZZY.1992.258721
10. Ухоботов В.И. Избранные главы теории нечетких множеств. Челябинск: Изд-во Челяб. гос. ун-та, 2011. 245 с.

11. Ухоботов В.И., Величко В.С. Моделирование заданного движения пятизвенного механизма // *Вестник ЮУрГУ. Серия: Вычислительная математика и информатика*. 2013. Т. 2. № 3. С. 104–110.
12. Cordon O., F. Herrera, P. Villar. Generating the Knowledge Base of a Fuzzy Rule-Based System by the Genetic Learning of the Data Base // *IEEE Transactions on fuzzy systems*, Vol. 9, No. 4, 2001. P. 667–674. DOI: 10.1109/91.940977
13. Сергеев Л.В. Теория танка. М: Военная ордена Ленина Краснознаменная академия бронетанковых войск имени маршала Советского Союза Р.Я. Малиновского, 1973. 496 с.
14. Hartigan J. Clustering Algorithms. Wiley, New York, NY, 1975. 351 p.
15. Gustafson D.E., Kessel W.C. Fuzzy Clustering With a Fuzzy Covariance Matrix // *Proceedings IEEE Conference on Decision and Control Including the 17th Symposium on Adaptive Processes*. (San Diego, CA, USA, January, 10 – January, 12, 1978). IEEE Xplore Press, 1978. P. 761–766. DOI: 10.1109/CDC.1978.268028
16. Chiu S., Fuzzy Model Identification Based on Cluster Estimation // *Journal of Intelligent & Fuzzy Systems*. 1994. Vol. 2, No. 3. P. 267–278. DOI: 10.3233/IFS-1994-2306

Пташко Евгений Анатольевич, аспирант, кафедра теории управления и оптимизации, Челябинский государственный университет (Челябинск, Российская Федерация)

Ухоботов Виктор Иванович, д.ф.-м.н., профессор, зав. каф. теории управления и оптимизации, Челябинский государственный университет (Челябинск, Российская Федерация)

DOI: 10.14529/cmse170304

AUTOMATIC GENERATION OF FUZZY RULES FOR CONTROL OF A MOBILE ROBOT WITH TRACK CHASIS BASED ON NUMERICAL DATA

© 2017 E.A. Ptashko, V.I. Ukhobotov

Chelyabinsk State University (st. Bratiev Kashirinikh 129, Chelyabinsk, Russia 454001)

E-mail: eugene.ptashko@gmail.com, ukh@csu.ru

Поступила в редакцию: 08.06.2017

In this paper, we consider the problem of generating a set of fuzzy rules for the Mamdani fuzzy inference system based on numerical data obtained in the learning process of a managed system. The approach proposed in the article to solve this problem is based on algorithms for clear and fuzzy clustering, such as the mining clustering algorithm and the Gustafson – Kessel algorithm. It allows you to significantly simplify the process of forming a set of fuzzy rules and minimize the participation of a person in this process, allowing you to automatically select the number of rules, as well as determine all the necessary parameters for each of them. To implement the proposed approach, two computer programs were written. The first of them collects numeric data when a person manages a robot. Based on the collected data, this program builds a base of fuzzy rules for controlling mobile robot on a tracked chassis. This base of fuzzy rules and its computer implementation is further used in the second program for automated control of a mobile robot in the plane by varying the tractive force of each of the tracks depending on the position of the target to which the robot should approximate a given distance.

Keywords: fuzzy inference system, fuzzy clustering, computer implementation.

FOR CITATION

Ptashko E.A., Ukhobotov V.I. Automatic Generation of Fuzzy Rules for Control of a Mobile Robot with Track Chasis Based on Numerical Data. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2017. vol. 6, no. 3. pp. 60–72. (in Russian) DOI: 10.14529/cmse170304.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Wong J.Y. Theory of Ground Vehicles, 3rd Edition. Wiley-Interscience, 2001. 284 p.
2. Zabavnikov N.A. *Osnovy teorii transportnykh gusenichnykh mashin* [The Fundamentals of the Theory of Transport Caterpillar Vehicles]. M.: Mechanical Engineering, 1975. 448 p.
3. Kolesnikov A.A. *Sinergeticheskaya teoriya upravleniya* [Synergetic Theory of Management]. M.: Energoatomizdat, 1994. 344 p.
4. Scliarov A.A., Scliarov S.A. Synergetic Control of Tracked Mobile Robot. *Izvestiya YuFU. Tekhnicheskie nauki* [SFU News. Technical Science]. 2011. no. 6. pp. 118–125. (in Russian)
5. Zadeh L.A. The Concept of a Linguistic Variable and its Application to Approximate Reasoning. American Elsevier Publishing Company, New York, 1973. 161 p.
6. Castro J.L. Fuzzy Logic Controllers are Universal Approximators. *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics*. 1995. vol. 25. pp. 629–635. DOI: 10.1109/21.370193
7. Kosko B. Fuzzy Systems as Universal Approximators. *IEEE Transactions on Computers*. 1994. vol. 43, no. 11. pp. 1329–1333. DOI: 10.1109/12.324566
8. Mamdani E.H. Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis. *IEEE Transactions on Computing*. 1977. vol. 26. pp. 1182–1191. DOI: 10.1109/TC.1977.1674779
9. Wang L.X. Fuzzy Systems are Universal Approximators. *Proceedings of IEEE International Conference on Fuzzy Systems (San Diego, CA, USA, March, 8 – March, 12, 1992)*. IEEE Xplore Press, 1992. pp. 1163–1169. DOI: 10.1109/FUZZY.1992.258721
10. Ukhobotov V.I. *Izbrannye glavy teorii nechetkikh mnozhestv*. [Selected Chapters of the Theory of Fuzzy Sets]. Chelyabinsk, Publishing of the Chelyabinsk State University, 2011. 245 p.
11. Ukhobotov V.I. Simulation of a Given Motion of a Five-Link Mechanism. *Vestnik Yuzho-Uralskogo gosudarstvennogo universiteta. Vychislitel'naya matematika i informatika* [Bulletin of South Ural State University. Series: Computational Mathematics and Informatics]. 2013. vol. 2, no. 3. pp. 104–110. (in Russian)
12. Cordon O., F. Herrera, P. Villar. Generating the Knowledge Base of a Fuzzy Rule-Based System by the Genetic Learning of the Data Base. *IEEE Transactions on Fuzzy Systems*. 2001. vol. 9, no. 4. pp. 667–674. DOI: 10.1109/91.940977
13. Sergeev L.V. *Teoriya tanka*. [Tank Theory]. Military Order of Lenin Red Banner Academy of Armored Forces Named After Marshal of the Soviet Union R.Ya. Malinovsky, 1973. 496 p.

14. Hartigan J. Clustering Algorithms. Wiley, New York, NY, 1975. 351 p.
15. Gustafson D.E., Kessel W.C. Fuzzy Clustering With a Fuzzy Covariance Matrix. Proceedings IEEE Conference on Decision and Control Including the 17th Symposium on Adaptive Processes. (San Diego, CA, USA, January, 10 – January, 12, 1978). IEEE Xplore Press, 1978. pp. 761–766. DOI: 10.1109/CDC.1978.268028
16. Chiu S., Fuzzy Model Identification Based on Cluster Estimation. Journal of Intelligent & Fuzzy Systems. 1994. vol. 2, no. 3. pp. 267–278. DOI: 10.3233/IFS-1994-2306