

Fuzzy Logic Toolbox 2.2.11 MatLab

Modeling Inverse Kinematics in a Robotic Arm

http://www.mathworks.com/products/fuzzylogic/demos.html?file=/products/demos/shipping/fuzzy/invkine_codepad.html

Факультет Компьютерных Наук и Технологий
Донецкий Национальный Технический Университет

Перевод выполнен для портала магистров ДонНТУ
<http://masters.donntu.edu.ua/>

перевод :
ст.гр. КС09м
Соломка О.С.

Fuzzy Logic Toolbox 2.2.11 MatLab

Моделирование Инверсной Кинематики для Робота-манипулятора

Что такое инверсная кинематика?

Кинематика это наука о движении. В двусочленном манипуляторе с учетом углов в суставах, кинематические уравнения позволяют определить положение его окончания. Инверсная (обратная) кинематика ссылается на обратный процесс. При известном положении кончика роботизированной руки, необходимо высчитать углы суставов для обеспечения желаемой локации. Как правило, существует более одного решения, а иногда задача может быть достаточно трудноразрешимой .

Это типичная задача в области робототехники, которую необходимо разрешить для управления роботом-манипулятором и выполнения поставленных задач. В 2-мерном пространстве ввода[3], с двусочленным манипулятором, задача сводится к нахождению двух углов, которые участвуют в желаемой координации. Первый угол - угол между первой консолью и базой крепления. Второй угол между первой и второй консолью (рис. 1).

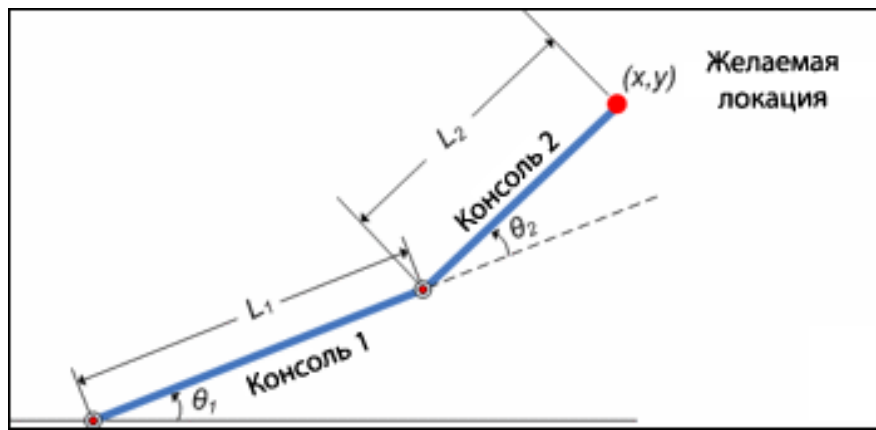


Рисунок 1: Иллюстрация показывает двусочленённый манипулятор с двумя углами, θ_1 и θ_2

Почему используется нечеткая логика?

Для простых структур, как двусочленённый манипулятор, можно математически вывести размер углов в местах соединения консолей с учетом нужного положения кончика робастного манипулятора. Вместе с тем для более сложных структур (например, n -сочлененных робастных манипуляторов, работающих в 3-мерном пространстве ввода) выведение математического решения для инверсной кинематики может оказаться достаточно сложной задачей.

При использовании нечеткой логики, мы можем построить систему нечеткого предположения, которую может описать инверсная кинематика, если известна прямая кинематика поставленной задачи, что позволит обойти необходимость в разработке аналитических решений. Кроме того, нечеткое решение является простым для понимания и не требует специальных фоновых знаний, чтобы понять и оценить его.

В текущей статье, в общих чертах для осуществления такого решения описаны и разработаны подробные шаги прикладного метода.

Обзор решений с использованием нечеткой логики.

Поскольку прямая кинематика формулируется для двусочлененного манипулятора, для известных x и y - координат кончика манипулятора, выводится весь диапазон углов поворота двух консолей. Координаты и углы

сохраняются для использования в качестве обучающих данных для обучения ANFIS[1] сети (адаптивная нейро-нечеткая система предположения).

Во время обучения, сеть ANFIS учится составлять карту координат (x , y) по углам (θ_1 , θ_2). Обученная сеть ANFIS затем используется в качестве части более широкой системы контроля для управления роботным манипулятором. Зная нужное положение манипулятора, система управления использует обученную сеть ANFIS для расчета углового положения суставов и воздействует на суставы манипулятора так, чтобы переместить его в нужную локацию.

Что такое ANFIS?

ANFIS (АНСП) означает Адаптивная Нейро-нечеткая Система Предположения. Это гибрид нейро-нечеткого метода, который преподносит возможные пути к обучению нейронных сетей для систем нечеткого предположения. Данный алгоритм раскрывает класс функций[2] типа Sugeno систем ANFIS, и использует обучающие данные ввода-вывода.

При этом положении, данные ввода-вывода относятся к набору данных "координаты-углы". Координаты являются входными данными для ANFIS а углы являются производными выходными. Данный алгоритм "обучает" ANFIS составлять карту сопоставления "координат к углам" с помощью процесса, называемого обучением. В конце обучения, обученная сеть ANFIS обучается картам ввода-вывода и готова к развертыванию в более объемную систему контроля.

Генерация данных

Пусть θ_1 - угол между первой консолью и базой крепления. Тогда θ_2 - угол между второй и первой консолью (см. рис. 1). Пусть длина первой консоли будет l_1 , второй l_2 .

Будем считать, что первое сочленение имеет ограниченную свободу для вращения и может вращаться между 0 и 90 градусов. Так же, предположим, что второе сочленение имеет ограниченную свободу для вращения и может вращаться между 0 и 180 градусов (это предположение снимает необходимость для обработки некоторых особых случаев, которые

будут путать изложение). Таким образом, $0 \leq \theta_1 \leq \pi/2$ и $0 \leq \theta_2 \leq \pi$.

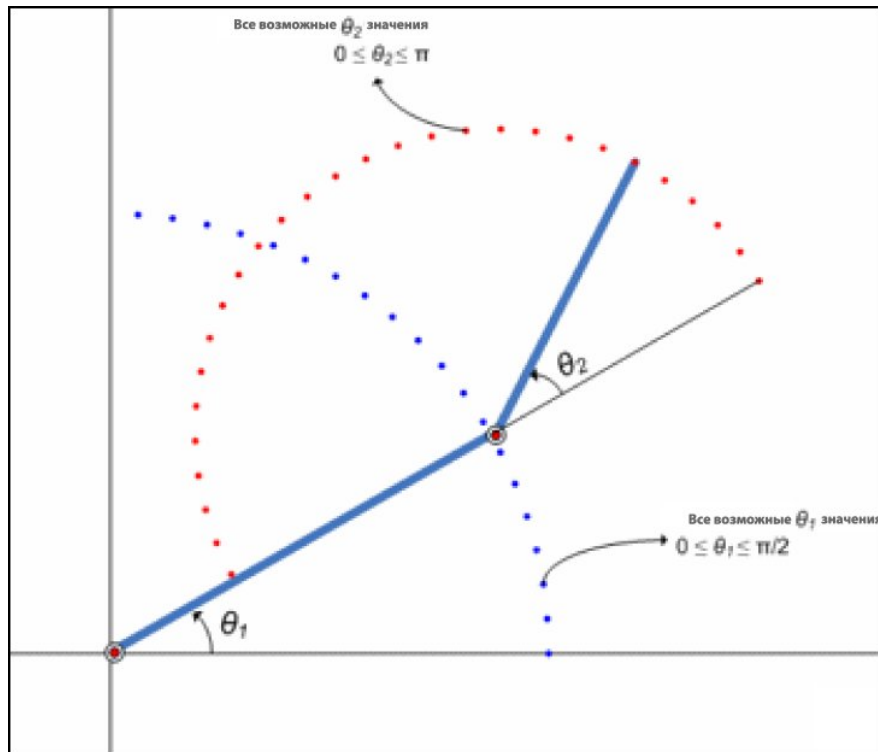


Рисунок 2. Иллюстрация с указанием всех возможных значений θ_1 и θ_2 .

Теперь для каждой комбинации значений θ_1 и θ_2 , координаты x и y выводятся с использованием формулировки прямой кинематики.

Следующий фрагмент кода показывает, как создаются данные для всех сочетаний значения θ_1 и θ_2 и сохраняются в матрицы для использования в качестве обучающих данных. Причина для сохранения данных в две матрицы объясняется в следующем разделе.

```
l1 = 10;           % длина первой консоли
l2 = 7;           % длина второй консоли

theta1 = 0:0.1:pi/2; % все возможные значения theta1
theta2 = 0:0.1:pi;   % все возможные значения theta2

[THETA1, THETA2] = meshgrid(theta1, theta2); % генерация сетки значений theta1 и theta2

X = l1 * cos(THETA1) + l2 * cos(THETA1 + THETA2); % вычисление x координат
Y = l1 * sin(THETA1) + l2 * sin(THETA1 + THETA2); % вычисление y координат

data1 = [X(:) Y(:) THETA1(:)]; % создание x-y-theta1 набора данных
data2 = [X(:) Y(:) THETA2(:)]; % создание x-y-theta2 набора данных
```

Следующий график отображает все точки XY данных, полученные при проведении цикла при различных комбинаций θ_1 и θ_2 и выведения x и y координат для каждого. Иллюстрация может быть получена с помощью кода-фрагмента, представленного ниже. Для облегчения понимания на рисунке 3 представлен график.

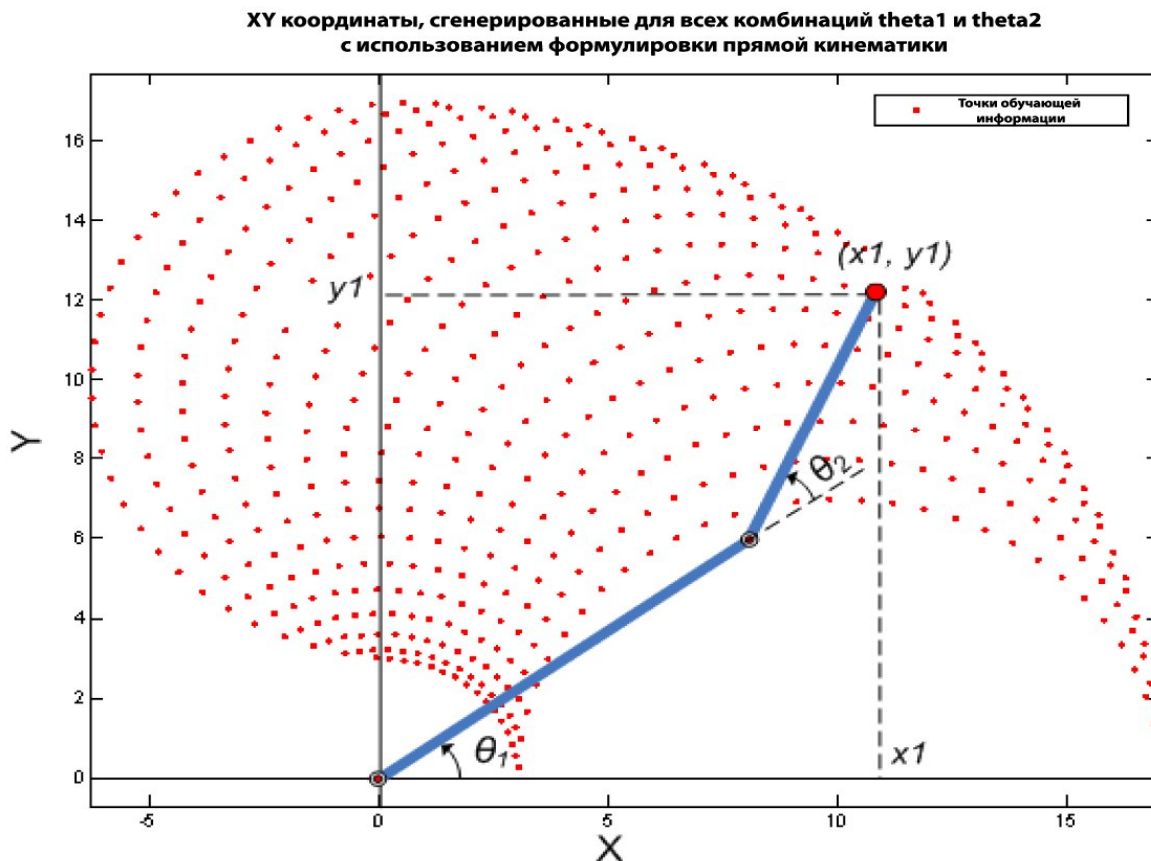


Рисунок 3: Создание XY координат для всех комбинаций θ_1 и θ_2 с использованием формулировки прямой кинематики

Создание сетей ANFIS

Один из подходов к решению этой проблемы, является создание 2-х ANFIS сетей, одна служит для предсказания θ_1 а вторая для предсказания θ_2 .

Для того чтобы ANFIS сети имели возможность предсказывать размер углов, они должны быть обучены примерами ввода-вывода данных. Для этого первая сеть ANFIS обучается X и Y координатами в качестве входных данных и соответствующих θ_1 значений на выходе. Матрица $Data1$

содержит x - y - θ_1 наборы данных, необходимых для подготовки первой сети ANFIS и используется в качестве набора данных для обучения первой сети ANFIS.

Точно так же, вторая сеть ANFIS обучается X и Y координатами в качестве входных данных и соответствующих θ_2 значений на выходе. Матрица $data_2$ содержит x - y - θ_2 набор данных, необходимых для подготовки второй ANFIS сети и используется в качестве набора данных для обучения второй ANFIS сети.

`anfis` - функция, которая используется для подготовки ANFIS сети. Есть несколько вариантов синтаксиса данной функции. При вызове со следующим синтаксисом, `anfis` автоматически создает FIS[4], Sugeno типа и обучает её с помощью обучающих данных, переданных функции.

Для запуска следующего кода нужно всего несколько минут :

```
fprintf('-->%s\n','Начало обучения первой ANFIS сети. Это может занять некоторое время в зависимости от вашей
операционной системы')
anfis1 = anfis(data1, 7, 150, [0,0,0,0]);           % обучение первой ANFIS сети
fprintf('-->%s\n','Начало обучения второй ANFIS сети. Это может занять некоторое время в зависимости от вашей
операционной системы')
anfis2 = anfis(data2, 6, 150, [0,0,0,0]);           % обучение второй ANFIS сети

-->Начало обучения первой ANFIS сети. Это может занять некоторое время в зависимости от вашей операционной
системы
-->Начало обучения второй ANFIS сети. Это может занять некоторое время в зависимости от вашей операционной
системы
```

Первый параметр `anfis` является обучающими данными, второй параметр - количество функций принадлежности[2], используемые для характеристики каждого входа и выхода, третий параметр ряд обучающих эпох[5] и последний параметр опция для отображения прогресса при выполнении процесса обучения. Значения для количества эпох и количества функций принадлежности, были выведены после многократных экспериментов с различными значениями.

Fuzzy Logic Toolbox 2.2.11 поставляется с GUI, что позволяет с легкостью создавать ANFIS сети и экспериментировать с ними.

`anfis1` и `anfis2` представляют собой две подготовленных ANFIS сети, которые будут развернуты в крупную систему управления.

После завершения обучения, две ANFIS сети обучены аппроксимировать углы (θ_1 , θ_2) как функцию от координат (x , y).

Одно из преимуществ использования нечеткого подхода является то, что сети ANFIS теперь аппроксимируют углы для координат, которые являются аналогичными, но не точно такими же, как во время обучения. Например, обученная сеть ANFIS способна теперь аппроксимировать углы для координат, которые лежат между двумя точками, включенными в обучающий набор данных. Это позволит контроллеру плавно перемещать манипулятор во входном пространстве[3].

Теперь у нас есть две обученных сети ANFIS, которые готовы быть развернутыми в более крупную систему, которая будет использовать эти сети для управления роботизированным манипулятором.

Валидация ANFIS Сети

Обучивши сети, следующим важным шагом является проверка сетей для определения, насколько хорошо ANFIS сети будут работать в крупной системе управления.

Так как эта демо задача связана с двусочлененным манипулятором, для которого может быть получена формулировка инверсной кинематики, можно сопоставить ответы, которые ANFIS сети производят с ответами из полученной формулы.

Давайте предположим, что сети ANFIS могут иметь малую погрешность в рабочем диапазоне $0 < x < 2$ и $8 < y < 10$.

```
x = 0:0.1:2;      % x координаты для валидации
y = 8:0.1:10;    % y координаты для валидации
```

Значения theta1 и theta2 выводятся математически от координат x и y с использованием формулировки обратной(инверсной) кинематики.

```
[X, Y] = meshgrid(x,y);
c2 = (X.^2 + Y.^2 - l1^2 - l2^2)/(2*l1*l2);
s2 = sqrt(1 - c2.^2);
THETA2D = atan2(s2, c2);      % формулируем theta2

k1 = l1 + l2.*c2;
k2 = l2*s2;
THETA1D = atan2(Y, X) - atan2(k2, k1);      % формулируем theta1
```

THETA1D и THETA2D - это переменные, которые содержат значения theta1 и theta2 выведенные с помощью формулировки обратной кинематики.

Значения theta1 и theta2 предсказанные с помощью обученных сетей ANFIS, получены с помощью команды evalfis которая определяет FIS для данных входных значений.

Здесь evalfis используется, чтобы узнать выходные значения FIS для тех же значений x-y использованных ранее в формулировке обратной кинематики.

```
XY = [X(:) Y(:)];
THETA1P = evalfis(XY, anfis1);      % theta1 предсказанное anfis1
THETA2P = evalfis(XY, anfis2);      % theta2 предсказанное anfis2
```

Теперь мы можем видеть, насколько близко выходные данные FIS сопоставимы с выведенными значениями (рис. 4).

```

theta1diff = THETA1D(:) - THETA1P;
theta2diff = THETA2D(:) - THETA2P;

subplot(2,1,1);
plot(theta1diff);
ylabel('THETA1D - THETA1P')
title('Выведенное theta1 - Предсказанное theta1')

subplot(2,1,2);
plot(theta2diff);
ylabel('THETA2D - THETA2P')
title('Выведенное theta2 - Предсказанное theta2')

```

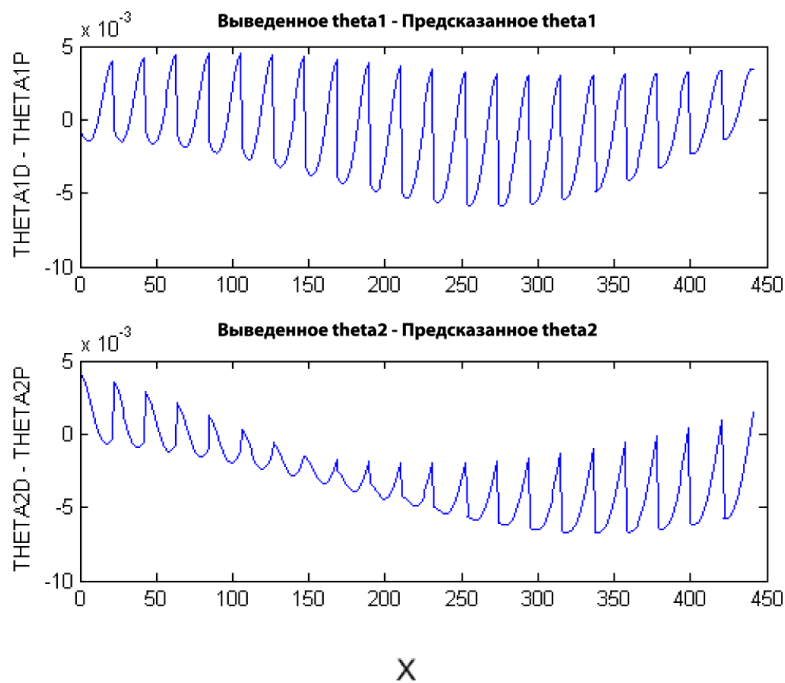


Рисунок 4. Ошибка предсказания

Ошибки находятся в $1e-3$ диапазоне, который удовлетворяет допустимую погрешность для развертывания в крупную систему управления. Однако это может быть не приемлемым для другого применения, в этом случае параметры функции `anfis` могут быть оптимально изменены до приемлемого значения. Кроме того, другие методы, такие как выбор ввода или альтернативного способа моделирования могут быть использованы для изучения данной задачи.

Создание Решения Вокруг Обученных ANFIS Сетей

Сейчас при решении конкретной задачи, такой как поднятие роботом объекта на конвейере, большая система управления будет использовать обученные сети ANFIS в качестве эталона или таблицы поиска, чтобы определить, какие углы должны быть у манипулятора, для желаемой локации его кончика. Зная желаемое значение углов в суставах и их текущее значение, система может применять нужное воздействие для их регулировки.

Команда `invkin` запускает графический интерфейс, который показывает, как работают две обученных ANFIS сети, когда есть запрос трассировки эллипса (рис. 5).

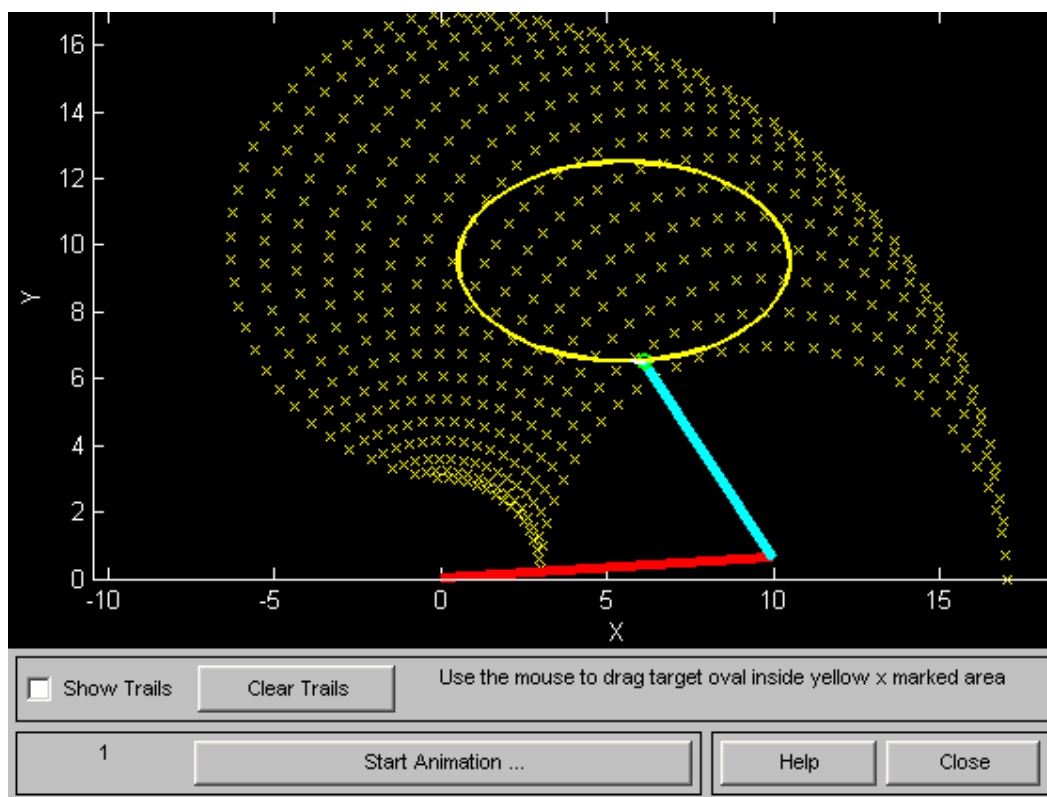


Рисунок 5: Демо-GUI для моделирования обратной кинематики.

Две ANFIS сети, используемые в демо были предварительно обучены и развернуты в более крупную систему, которая управляет кончиком двусочленного манипулятора для трассировки эллипса во входном пространстве.

Эллипс для трассировки можно перемещать. Переместите эллипс в другую локацию и наблюдайте, как система реагирует перемещением кончика робота-манипулятора с текущей локации до ближайшей точки на новой локации эллипса. Также отметим, что система реагирует на перемещения плавно до тех пор, пока трассировочный эллипс будет

находится в пределах "x" отмеченных точек, которые представляют собой данные сетки использовавшейся для обучения сетей. После того как эллипс выходит за пределы диапазона данных, с которыми она прошла обучение ANFIS сеть реагирует непредсказуемо. Это подчеркивает важность наличия обоснованных и репрезентативных данных для обучения. Все данные должны быть получены на основе ожидаемого диапазона работы, чтобы избежать такой непредсказуемости и нестабильности эмиссий.

Заключение

Эта демонстрация проиллюстрировала использование ANFIS для решения задач обратной кинематики. Нечеткая логики находит множество других применений в других областях техники, таких как нелинейный контроль, автоматическое управление, обработка сигналов, системы идентификации, распознавания образов, прогнозирования временных рядов, анализа данных, финансовые приложения и т.д.

Словарь ссылок

[1] ANFIS - адаптивные нейро-нечеткие системы предсказания. Метод для автоматической подстройки систем предсказания типа Sugeno на основе обучающих данных.

[2] Класс функций (функции принадлежности) - функция, которая определяет степень, в которой входной принадлежит множеству или имеет отношение к концепции.

[3] Входное пространство (пространство ввода) - это термин, используемый для определения диапазона всех возможных значений.

[4] FIS - система нечеткого предсказания. Общее название для систем, которые используют нечеткие обоснования к карте входного пространства для выходного пространства.

[5] Эпоха – одна эпоха обучения представляет собой одно целостное представление о всех образцах/точек данных/строк из набора данных для подготовки FIS. Представляются каждые входные примеры и вычисляемые FIS выводы, которые сравниваются с желаемыми выводами, для определения отклонений. После чего параметры функций принадлежности настраиваются для уменьшения ошибки между желаемым результатом и фактическим выходом FIS.