Лекция 4. Алгоритм Сугено. Элементы ТАУ

4.1. Нечеткие системы Сугено

4.1.1. Примеры

Пример 4.1. Рассмотрим четкую логическую систему, определяющую сумму подоходного налога по прогрессивной шкале. Входная переменная x – доход, тыс. руб./месяц, выходная y – налог, тыс. руб/месяц.

1. ЕСЛИ доход – менее 40 тыс. руб., ТО налог – 10%, $y = 0.1\,x$.
2. ЕСЛИ доход – от 40 до 100 тыс. руб., ТО налог – 13%, $y = 0.13\,x$.
3. ЕСЛИ доход – от 100 до 500 тыс. руб., ТО налог – 15%, $y = 0.15\,x$.
4. ЕСЛИ доход – более 500 тыс. руб., ТО налог – 20%, $y = 0.2\,x$.

Таким образом, сумма налога определяется линейной формулой, но коэффициент в формуле определяется системой правил.

Пример 4.2. На первой лекции приводился пример расчета коэффициента гидравлического сопротивления λ в зависимости от числа Рейнольдса Re и относительной шероховатости поверхности трубы ε.

1. ЕСЛИ $Re \le 2300$, ТО $\lambda = \dfrac{64}{Re}$.
2. ЕСЛИ $2300 < Re \le 10000$, ТО $\lambda = \dfrac{64}{Re}(1-\gamma) + \dfrac{0.3164}{\sqrt[4]{Re}}\gamma$, где $\gamma = 1 - e^{-0.002\,(Re-2320)}$.
3. ЕСЛИ $Re > 10000$, ТО $\lambda = 0.11\left(\varepsilon + \dfrac{68}{Re}\right)^{0.25}$.

В этом примере выходная переменная λ нелинейно зависит от входной переменной Re.

4.1.2. Правила в системах Сугено

Нечеткие системы М.Сугено (Michio Sugeno – японский математик) являются обобщением логических систем такого типа, т.е. они совмещают свойства нечеткой логики и четкой математики. Алгоритм Сугено следует применять, если есть несложная, но адекватная математическая модель – функциональная зависимость между выходными и входными переменными. Причем коэффициенты этой модели различны для разных ситуаций, а выбор между ситуациями осуществляется с помощью нечетких правил.

Фаззификация проводится только для входных переменных, а для выходных переменных задаются четкие термы, состоящие из названий и формул, соответствующих этим названиям.

Правила в алгоритме Сугено устроены следующим образом. Пусть, как и ранее, $\vec{x} = [x_1, x_2, \dots]$ – числовые (четкие) входные переменные, $\vec{X} = [X_1, X_2, \dots]$ – соответствующие им лингвистические (нечеткие) переменные. На выходе задаются только четкие переменные $\vec{y} = [y_1, y_2, \dots]$. Каждой нечеткой входной переменной X_r соответствуют термы $T_r^1, T_r^2, \ldots$ с функциями принадлежности $\mu_r^k(x_i)$.

Условия в правилах такие же, как и в алгоритме Мамдани, т.е. это нечеткие утверждения, состоящее из утверждений вида $X_r = T_r^k$, связанных между собой логическими операциями И, ИЛИ и НЕ, например, НЕ $X_1 = T_1^1$ И $X_2 = T_2^3$. Степень истинности условия i-того правила, как и ранее, обозначим L_i.

Следствие i-го правила – значения выходных переменных, $y_j = f_{ij}(\vec{x})$, вычисленные через значения входных.

В Fuzzy Logic Toolbox предусмотрены только два типа зависимостей для выходных переменных: постоянная (constant), $f_{ij}(\vec{x}) = c_{ij}$, и линейная (linear), $f_{ij}(\vec{x}) = \vec{a}_{ij}{}'\vec{x} + c_{ij}$ (напомним, что векторы умножаются по правилу "строка на столбец", а знак $'$ обозначает транспонирование вектора).

Таким образом, правило может иметь, например, такой вид

$$\text{ЕСЛИ } X_1 = T_1^1 \text{ ИЛИ НЕ } X_2 = T_2^3, \text{ ТО } y_1 = a_1\, x_1 + a_2\, x_2 + a_3,\ y_2 = c_3.$$

Как и в системах Мамдани, можно вместо буквенных обозначений переменных использовать слова естественного языка. Для этого нужно дать словесные значения не только термам входных переменных, но и функциям $f_{ij}(\vec{x})$, т.е. создать *четкие термы* для выходных переменных. В этом случае вместо знака равенства следует использовать связку *is* для англоязычных названий или знак "–" (тире) для русскоязычных.

4.1.3. Алгоритм Сугено

Алгоритм Сугено для каждой выходной переменной y состоит из следующих этапов:

- для i-го правила вычисляются степень истинности условия L_i и значение выходной переменной y_i в соответствии с правилом;

- вычисляется результирующее значение выходной переменной – как среднее по всем правилам,

$$\overline{y} = \frac{\sum\limits_{i=1}^{n} L_i y_i}{\sum\limits_{i=1}^{n} L_i},$$

  где n – количество правил.

Хотя термы выходных переменных являются четкими, правила, выбирающие между ними, – нечеткие, поэтому второй этап алгоритма Сугено тоже называется дефаззификацией, как и в алгоритме Мамдани.

Пример 4.3. Вспомним еще раз пример системы управления смесителем в душе. Выходная переменная "горячий-кран" может иметь такие четкие термы постоянного типа:

$$\begin{array}{lclclcl} \text{“сильно-закрыть”} & = & -40\%, & \text{“немного-закрыть”} & = & -20\%, & \text{“не-трогать”} = 0\%, \\ \text{“немного-открыть”} & = & 20\%, & \text{“сильно-открыть”} & = & 40\%. & \end{array}$$

Для четких термов линейного типа нужно задать линейные формулы, причем можно обойтись меньшим количеством термов, поскольку эти термы более гибкие. Например для переменной "горячий-кран" достаточно двух термов – "сильно-повернуть" и "немного-повернуть",

$$\begin{array}{lclcll} \Delta q_2 & = & -2\,(T-40) - 0.4\,(q-50) & = & -2\,T - 0.4\,q + 100 & \text{– для “сильно-повернуть”;} \\ \Delta q_2 & = & -(T-40) - 0.2\,(q-50) & = & -T - 0.2\,q + 50 & \text{– для “немного-повернуть”.} \end{array}$$

Термы для переменной "холодный кран" могут иметь те же названия, но *формулы будут другие*.

Соответственно, можно обойтись и меньшим количеством правил (см. рис. 4.2), если использовать не только операцию И, но и ИЛИ.

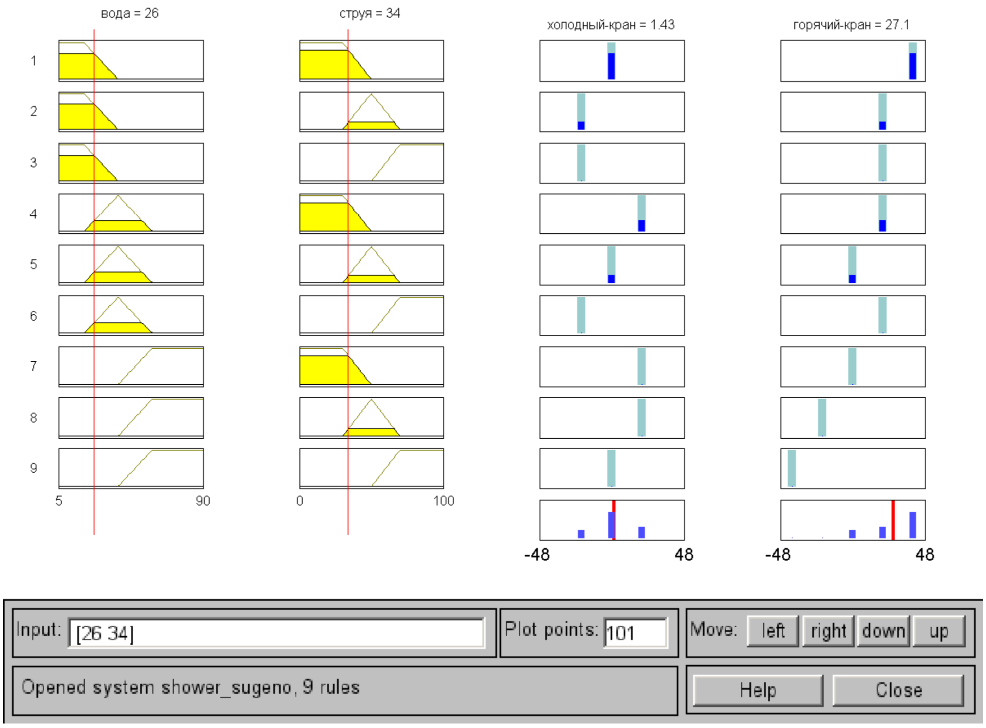

Рис. 4.1. Алгоритм Сугено для выходных переменных постоянного типа

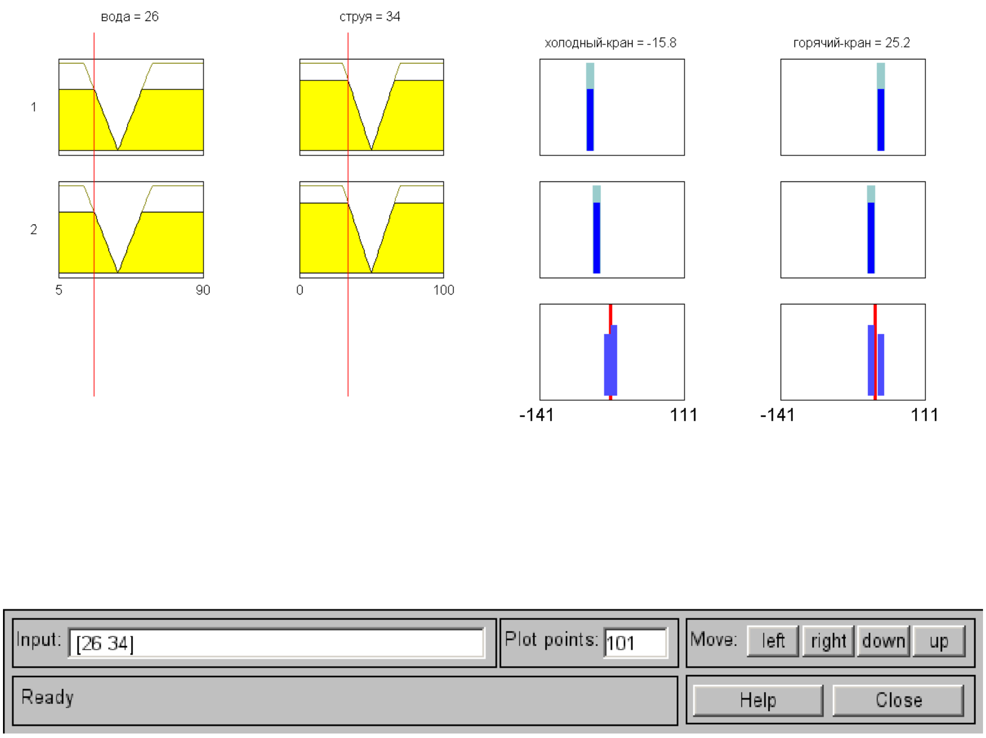

Рис. 4.2. Алгоритм Сугено для выходных переменных линейного типа

4.2. Элементы теории автоматического управления

4.2.1. Основные термины

Контур регулирования состоит из *объекта* регулирования и *регулятора.* Объект регулирования определяется *регулируемой величиной* y, *управляющим параметром (воздействием)* u и *возмущением* e, которые зависят от времени t. Для объекта u и e – это вход, y – выход, а для регулятора u – это выход, а y и e – вход. Все эти параметры, вообще говоря, векторные, т.е. могут состоять из нескольких компонент.

В примере со смесителем воды объектом был сам смеситель, регулируемые величины y_1 и y_2 – это температура и расход воды, управляющие параметры u_1 и u_2 – это изменения степеней открытия кранов, возмущения e_1 и e_2 – температура воды в холодной и горячей трубах.

Цель регулирования – добиться того, чтобы регулируемая величина как можно меньше отклонялась от y^* – требуемого значения регулируемой величины, которое называется *уставкой.*

Применяются два основных принципа регулирования:

- *по возмущению* – управляющий параметр u определяется через возмущение e;
- *по отклонению* – управляющий параметр u определяется через отклонение $\varepsilon = y - y^*$, которое также называется *сигналом рассогласования* (в технике) или *невязкой* (в математике).

В первом случае регулятор пытается устранить *причину* рассогласования, а во втором – борется с его *следствием.*

Для регулирования по возмущению мы должны достаточно точно измерить все возмущения и иметь хорошую математическую модель объекта – характеристику.

Статическая характеристика – зависимость регулируемой величины от управляющего параметра в установившемся режиме, $y = y(u)$ при $t \to \infty$.

Динамическая характеристика – то же, но в неустановившемся режиме, $y = y(u, t)$.

Модель должна быть достаточно точной и простой для расчета контроллером в реальном времени. В жизни такое бывает редко. Поэтому чаще используется регулирование по отклонению, т.е. с использованием *обратной связи*, когда на вход объекта подается управляющий параметр u, вычисленный через его выход y. Различают *отрицательную* (рост y приводит к такому измерению u, что y убывает и возвращается к исходному значению) и *положительную* (рост y приводит такому изменению u, что y возрастает и уходит все дальше от исходного значения) обратную связь. Понятно, что для целей регулирования необходима отрицательная обратная связь.

4.2.2. Критерии качества регулирования

При изменении уставки регулятор сразу изменяет управляющее воздействие, но в силу инерционности объекта регулируемая величина будет меняться с некоторым запаздыванием. В зависимости от настроек регулятора она будет либо плавно приближаться к новой уставке, либо колебаться вокруг нее с постепенным затуханием. Этот процесс называется *переходным.* Переходный процесс считается законченным к моменту времени t_0, если сигнал рассогласования перестает ощутимо меняться к этому времени, т.е. $|\varepsilon(t + \Delta t) - \varepsilon(t)| < \delta, \quad t > t_0$, где δ – погрешность вычислений.

Если после окончания переходного процесса регулируемая величина не будет равна уставке, то разность между ними $\varepsilon_s = \lim\limits_{t\to\infty}(y^* - y(t))$ называется *статической ошибкой.*

В реальности установившийся режим не существует, т.к. меняющиеся во времени возмущения влияют на регулируемую величину, а регулятор пытается это влияние нейтрализовать. Поэтому переходный процесс можно считать условно законченным, если сигнал рассогласования колеблется в некотором допустимом коридоре, $|\varepsilon(t)| < \Delta, \quad t > t_0$.

Реакция регулятора на возмущения и на изменение уставки может привести к резким скачкам регулируемой величины. Это крайне нежелательно, причем чем больше амплитуда скачков, тем хуже. Это явление называются *перерегулированием.*

На основе этой терминологии можно сформулировать основные цели и критерии качества регулирования.

Цель регулирования считается достигнутой, если, несмотря на возмущения, переходный процесс условно заканчивается к некоторому моменту времени t_0, а затем сигнал рассогласования плавно (без перерегулирования) колеблется в допустимом коридоре.

Регулирование тем качественнее, чем

- меньше амплитуда перерегулирования в течение переходного периода;
- меньше длительность переходного периода t_0;
- меньше ширина коридора колебаний сигнала рассогласования Δ в условно-установившемся режиме.

4.2.3. ПИД-регуляторы

Пропорционально-интегрально-дифференциальный (ПИД) регулятор – самый распространенный вид автоматических регуляторов. Сокращение "ПИД" связано не с конкретной технической реализацией, а с математической формулой, которая запрограммирована в регуляторе,

$$u(t) = K_p\,\varepsilon(t) + K_i \int\limits_0^t \varepsilon(x)\,dx + K_d \frac{d\varepsilon}{dt}(t),$$

где K_p, K_i, K_d – коэффициенты усиления.

Пропорциональная составляющая вырабатывает выходной сигнал, противодействующий отклонению регулируемой величины от уставки. Он тем больше, чем больше это отклонение. Если входной сигнал равен заданному значению, то выходной равен нулю. Однако при использовании только пропорционального регулятора значение регулируемой величины всегда будет ненулевая статическая ошибка.

Чем больше коэффициент усиления пропорциональной составляющей, тем меньше статическая ошибка, однако при слишком большом коэффициенте усиления возникают автоколебания.

Интегральная составляющая позволяет регулятору со временем учесть статическую ошибку. Если объект не испытывает внешних возмущений, то через некоторое время регулируемая величина стабилизируется на заданном значении, сигнал пропорциональной составляющей будет равен нулю, а выходной сигнал будет полностью обеспечивать интегральная составляющая.

Дифференциальная составляющая пропорциональна темпу изменения отклонения регулируемой величины и предназначена для противодействия отклонениям от уставки, которые прогнозируются в будущем. Отклонения могут быть вызваны внешними возмущениями или запаздыванием воздействия регулятора на объект.

В цифровых регуляторах управляющее воздействие формируется не непрерывно, а дискретно, через определенные промежутки времени Δt. Поэтому интегрирование и дифференцирование проводится численно. Переходя от непрерывного времени t к дискретному n, т.е. номерам моментов времени, получим

$$u_n = K_p\,\varepsilon_n + K_i \Delta t \sum_{k=1}^{n} \varepsilon_k + K_d \frac{\varepsilon_n - \varepsilon_{n-1}}{\Delta t}.$$

Здесь для интеграла была использована формула правых прямоугольников.

Для программной реализации удобнее перейти к рекуррентной формуле. Вычтем предыдущее управляющее воздействие из текущего, а затем перенесем его в правую часть равенства,

$$u_n = u_{n-1} + K_p\,(\varepsilon_n - \varepsilon_{n-1}) + K_i \Delta t\, \varepsilon_n + K_d \frac{\varepsilon_n - 2\,\varepsilon_{n-1} + \varepsilon_{n-2}}{\Delta t}.$$

В начальный момент времени еще нет никакого управления, поэтому $u_0 = 0$.

Существуют различные методы настройки ПИД-регуляторов, большинство из которых основано на анализе передаточной функции. Однако на практике часто подбирают параметры “на глазок”. Для этого вначале задают нулевые значения для всех коэффициентов и постепенно изменяют K_p, проверяя, что получилось по графику переходного процесса. Как только появляется перерегулирование, K_p фиксируют и переходят к настройке K_i.

С увеличением K_i статическая ошибка стремится к нулю и длительность переходного процесса вновь уменьшается. Как только появляется перерегулирование, K_i фиксируют и переходят к настройке K_d. Иногда достаточно пропорциональной и интегральной составляющей регулятора, а дифференциальная ничего дополнительного не дает. В этом случае оставляют $K_d = 0$.