

СИСТЕМЫ И ПРОЦЕССЫ УПРАВЛЕНИЯ



УДК681.518.24

МОДЕЛИ СИСТЕМЫ ЭМУЛЯЦИИ СЛОЖНОГО ТЕХНОЛОГИЧЕСКОГО ОБЪЕКТА УПРАВЛЕНИЯ

НАЙДА В.В., ОКСАНИЧ А.П., ШЕВЧЕНКО И.В., КОНОХ И.С.

Разрабатывается теоретико-множественная модель структуры системы эмуляции поведения сложного технологического объекта при его взаимодействии с системой управления. Предлагается нечеткая вычислительная модель для решения задачи эмуляции поведения объекта управления.

Введение

Широкое распространение сложных автоматизированных технологических комплексов, включающих в себя технологические объекты управления и автоматизированные системы управления, а также необходимость сокращения сроков разработки и внедрения приводят к необходимости использования имитационного моделирования (эмуляции) в системах отладки АСУТП. Имитационные комплексы обеспечивают отработку взаимодействия проектируемых систем управления с объектом управления и окружающей средой. Несмотря на то, что математическое моделирование значительно уменьшает стоимость исследовательских и отладочных работ, обладает хорошей повторяемостью и безопасностью, оно не позволяет полностью исключить натурные испытания и исследования качества работы систем управления при взаимодействии с реальным объектом.

Для сложных объектов, таких как корабельные системы, системы управления летательных аппаратов, электрических станций, обогатительных установок, обязательным этапом является отладка и настройка программно-аппаратных компонент на испытательных комплексах.

Известны программно-аппаратные имитационные комплексы [1], которые позволяют для информационно-управляющих систем боевых кораблей формировать внешние сигналы на основе специальных аппаратно-программных средств. Дополнительными преимуществами являются возможность тренинга обслуживающего персонала и исследование реакции системы управления в нештатных ситуациях.

В источниках [1-4] показаны характерные приложения для отладочных комплексов с имитацией электрических сигналов. Для подобных комплексов актуаль-

ным является совершенствование специальных методов идентификации объектов управления и эмуляции их поведения, которые характеризуются высокой степенью автоматизации и точностью воспроизведения осведомительных сигналов с сохранением метрологических особенностей.

Идентификация многомерных объектов в кибернетике и других областях предусматривает построение с помощью средств математики соответствующих им операторов связи между функциями входных и выходных сигналов на основе доступной информации [5]. При этом процесс создания математической модели того или иного объекта состоит из двух основных этапов:

- 1) структурная идентификация – установление переменных величин и элементов структуры модели;
- 2) параметрическая идентификация – этап, во время которого на основе наблюдения входных и выходных сигналов объекта определяются значения параметров и устанавливается окончательная структура модели путем устранения лишних элементов.

В основе современной теории идентификации лежит моделирование исследуемых объектов при помощи уравнений (дифференциальных, разностных и т. п.). При этом сложность того или иного объекта самым непосредственным образом влияет на качество построения его модели. Если для описания некоторых объектов применяется информация, которая не может быть выражена количественно – так называемая семантическая, т. е. смысловая, качественная информация, то классическая теория оказывается плохо приспособленной для таких случаев. Основные причины малой эффективности или же вообще непригодности традиционных методов моделирования к подобным ситуациям состоят в следующем:

- не все входные и выходные параметры объекта могут описываться количественно;
- между рядом входных и выходных параметров невозможно установить количественные зависимости;
- существующие способы моделирования объектов приводят к таким громоздким конструкциям, что их практическое применение оказывается невозможным;
- объект эволюционирует во времени, его структура и функции изменяются.

Преодоление таких сложностей с высокой эффективностью возможно при использовании технологий искусственного интеллекта. Известна работа [6], которая посвящена исследованиям в области идентификации нелинейных многомерных объектов нечеткими базами знаний. Этот метод идентификации ранее был научно обоснован в предыдущих работах. Для решения задачи идентификации предлагается комплексное использование нечетких баз знаний, нейронных сетей и генетических алгоритмов, объединенных под термином интеллектуальные технологии или мягкие вычисления (Soft Computing).

При моделировании объектов, которые характеризуются указанными особенностями, возникает проблема построения так называемых логико-лингвистических моделей, т.е. моделей, в которых средства обработки информации основаны на логике, а экспериментальные данные представляются в лингвистической форме [6]. Такие модели должны основываться на системах знаний об исследуемом объекте, которые представляют собой концентрацию опыта специалистов (экспертов) в данной области. Для систем знаний, в свою очередь, должен быть выбран язык представления знаний, при помощи которого в модели можно было бы адекватно воссоздавать сведения о структуре объекта, и который по возможности должен быть близким к естественному языку (в простейшем варианте – быть ее определенным подмножеством) и характеризоваться достаточной формальностью и логичностью в целях построения компактной, строгой и четкой системы знаний.

Одним из таких современных формальных аппаратов для обработки экспертной естественно-языковой информации является теория нечетких множеств. В соответствии с этой теорией модель объекта представляет собой так называемую нечеткую базу знаний в виде совокупности логических высказываний. Адекватность таких моделей к данным эксперимента определяется качеством функций принадлежности [7], при помощи которых лингвистические оценки превращаются в количественную форму. Но поскольку функции принадлежности определяются экспертными методами, адекватность нечетких моделей целиком зависит от квалификации экспертов. Иначе говоря, проблема адекватности известных нечетких моделей остается открытой. А когда привлечение экспертов для построения модели оказывается невозможным по причине их отсутствия, в таком случае возникает проблема извлечения лингвистических знаний об объекте из экспериментальных данных.

Таким образом, актуальность поднятой проблемы обусловлена тем, что для моделирования многих объектов есть смысл применять логико-лингвистические модели, дающие возможность преодолеть трудности моделирования классическими методами. Однако во многих случаях повышенная сложность того или иного объекта или его новизна (и как следствие – недостаточное освоение) делают невозможным привлечение квалифицированных экспертов для построения таких моделей. Это порождает проблему отыскания закономерностей, которые бы легли в основу системы лингвистических знаний об объекте моделирования, из имеющихся статистических (экспериментальных) данных, характеризующих исследуемый объект. Такие модели обычно являются «грубыми», поэтому не менее важен вопрос о поиске эффективных методов их тонкой настройки.

Целью исследования является разработка моделей системы эмуляции поведения сложного технологического объекта при его взаимодействии с системой управления, позволяющих разработать информационную технологию эмуляции и испытаний без разработки и использования сложных систем дифференциальных уравнений и систем имитационного моделирования.

1. Формальная постановка задачи эмуляции объекта управления и тестирования системы управления

Для записи формальной постановки задачи эмуляции объекта управления (ОУ) и тестирования СУ введем следующие обозначения:

$T = \{t_i, i = \overline{1, L}\}$ – множество дискретных моментов времени в сеансе тестирования; $U = \{u_i(t) | u_i(t) \in D^{U_i}, i = \overline{1, I}\}$, $t \in T$ – множество управляющих воздействий на объект управления (ОУ);

$D^U = \{D^{U_i}, i = \overline{1, I}\}$ – область допустимых значений управляющих воздействий;

$P = \{p_j(t) | p_j(t) \in D^{P_j}, j = \overline{1, J}\}$, $t \in T$ – множество экспериментальных последовательностей входных сигналов для ОУ; $D^P = \{D^{P_j}, j = \overline{1, J}\}$ – области допустимых значений входных сигналов для ОУ;

$S = \{S_i; S_i = f(S_{i-1}, U, P), U \in D^U, P \in D^P, t \in T\}$ –

множество состояний ОУ; $SI = \{S_i; S_i = \varphi(S_i), t \in T\}$ – множество параметров, характеризующих состояние ОУ;

$Y = \{y_m(t), m = \overline{1, M}\}$, $t \in T$, $y_m(t) = \eta_m(t, S_t)$, $t \in T$ – множество выходов ОУ; $Z = \{z_n(t), n = \overline{1, N}\}$, $t \in T$ – множество задаваемых параметров, определяющих режим сеанса тестирования;

$C = \{c_k(t) | c_k(t) \in D^{C^k}, k = \overline{1, K}\}$, $t \in T$ – множество желаемых последовательностей экспериментальных сигналов на выходе ОУ;

$D^C = \{D^{C^k}, k = \overline{1, K}\}$ – область допустимых значений желаемых экспериментальных сигналов на выходе ОУ;

$Q = \{q_g, g = \overline{1, G}\}$ – множество целей тестирования СУ.

Тогда взаимодействие системы управления и объекта управления в процессе тестирования и эмуляции можно представить рядом отображений:

$\eta: U \times P \times S \times T \rightarrow Y$ – отображение множества управляющих воздействий U , выбранных из множества экспериментальных последовательностей входных сигналов P , множества состояний ОУ S на всём интервале времени T , в котором осуществляется тестирование, во множество выходов ОУ Y ;

$\mu: U \times P \times T \rightarrow S$ – отображение множества управляющих воздействий U , выбранных из множества экспериментальных последовательностей входных сигналов P на всём интервале времени T , на множество состояний S ;

$\psi: S \times T \rightarrow SI$ – отображение множества состояний ОУ во множество параметров состояния ОУ на интервале времени T ;

$\varphi: Q \times Y \times SI \rightarrow C$ – отображение множества целей тестирования Q , множества выходов ОУ Y и множества параметров состояний ОУ SI на множество желаемых последовательностей экспериментальных сигналов на выходе ОУ.

Выбор критерия качества процесса тестирования. В процессе тестирования система тестирования и эмуляции должна фиксировать последовательности входных сигналов ОУ, последовательности выходных сигналов ОУ и оценивать степень соответствия последовательности выходных сигналов Y желаемой последовательности C .

Считая, что успешность процесса тестирования определяется степенью совпадения выходных последовательностей Y и желаемых выходных последовательностей C в различных сценариях тестирования, запишем критерий успешности процесса тестирования в виде:

$$Q_T = \sum_{k=1}^K \sum_{i=1}^N \sum_{t=0}^T (y(u_t, s_t)_{kit} - c_{kit})^2 \rightarrow \min, \quad (1)$$

где k – номер сценария тестирования; K – количество сценариев; i – номер отрезка последовательности сигналов, на котором фиксируются значения входа и выхода; N – число отрезков последовательностей.

В систему ограничений в данной задаче входят:

а) ограничения на параметры состояния и управления:

$$\begin{aligned} s(t) &\in D^S, t \in T; \\ u_i(t) &\in D^{U_i}, i = \overline{1, I}, t \in T; \\ c_k(t) &\in D^{C_k}, k = \overline{1, K}, t \in T; \end{aligned}$$

б) показатели качества функционирования ОУ, зависящие от специфики и назначения ОУ.

Ограничения могут быть жесткими и нежесткими, в последнем случае они могут быть записаны в вероятностном виде, нечетком виде.

Таким образом, сформулирована задача эмуляции сложного технологического объекта управления и тестирования системы управления.

2. Структура системы эмуляции и тестирования

Структуру системы эмуляции и тестирования (СЭТ) представим набором:

$$STS = \langle IA(\eta, \mu, \psi, \varphi), F, DB, R, MM, OMI, ICS \rangle, \quad (2)$$

где IA – интеллектуальный агент, реализующий отображения η, μ, ψ, φ ; $DB(T, P, Y, C, D)$ – база данных, содержащая отсчеты времени, входные и выходные последовательности, сигналы и ограничения; F – множество функций СЭТ; $RH F \times Q$ – отображение множества функций СЭТ на множество целей тестирования комплекса СУ–ОУ; MM – мастер-монитор, осуществляющий диспетчерские функции в ходе процесса тестирования; OMI – интерфейс с объектом управления; ICS – интерфейс с системой управления.

3. Краткая характеристика комплекса функциональных задач СЭТ

Комплекс функциональных задач СЭТ предназначен для решения задачи оптимизации параметров системы управления сложного технологического объекта в процессе отладки и испытаний.

В комплекс входят следующие задачи:

- полуавтоматическая настройка входных и выходных сигналов тестирующих процедур;
- генерация и фиксация тестовых воздействий на ОУ и фиксация откликов ОУ;
- структурирование и анализ результатов тестирования;
- формирование отчетов и рекомендаций для оператора.

Комплекс задач решается в процессе испытаний СУ в режиме Hardware-in-the-loop. Продолжительность решения задач обусловлена продолжительностью процесса тестирования и испытаний. Периодичность решения задач определяется периодичностью проектных и пуско-наладочных работ на технологических объектах. Автоматизированное решение задач может быть прекращено по желанию оператора.

В процессе решения данного комплекса задач СЭТ связана информационными потоками с системой автоматического управления ОУ и оператором, проводящим тестирование.

Входной информационный поток комплекса задач содержит:

1. Исходные данные о входных и выходных сигналах ОУ и данные о связях «вход-выход» ОУ.
2. Сигналы СУ, подаваемые на ОУ, и сигналы ОУ – реакции на тестовые воздействия.

Выходной информационный поток комплекса задач содержит структурированные массивы значений выходных сигналов ОУ и результаты их анализа.

4. Концепция функционирования СЭТ

Предлагаемый метод решения задачи эмуляции поведения объекта управления базируется на идее накопления экспериментальных временных рядов входов и выходов ОУ и расчете на их основе точек новой последовательности, которая соответствует текущему входному управляющему сигналу. При этом целесообразно использовать предварительную классификацию последовательностей путем указания для каждой из них атрибутов, содержащих информацию о форме и величине управляющего воздействия, начальном состоянии объекта, производных выходной величины. Для объектов выше второго порядка при поиске ближайших точек в выбранных последовательностях необходимо учитывать их предыдущие состояния в моменты времени, определяемые постоянными времени передаточных функций. Информация о всех последовательностях хранится в БД, содержащей следующую информацию:

- тип управляющего воздействия;
- амплитуда входного воздействия;
- дополнительная характеристика управляющего воздействия (например, скорость нарастания линейного сигнала);

- начальное состояние (начальный выход) объекта;
- значение дискреты времени;
- путь и имя файла, непосредственно содержащего экспериментальные отчеты.

Последовательности, которые возможно использовать для расчета выхода объекта управления, выбираются на основании определения евклидова расстояния по набору переменных. Сюда входят тип и амплитуда управляющего воздействия, начальный выход объекта, время с момента изменения управляющего воздействия. Исходя из этого, целесообразно исходные непрерывные экспериментальные последовательности разбить на более короткие фрагменты для сокращения времени поиска.

После выбора нескольких последовательностей, которые удовлетворяют условию поиска, необходимо определить отдельные точки в них и рассчитать интерполированные значения.

Для универсальности метода необходимо на основе указанной информации рассчитать относительные отклонения сигналов и подать их на входы интеллектуального агента, роль которого играет нечеткая вычислительная модель.

5. Нечеткая вычислительная модель

Нечеткая модель имеет четыре входные переменные из множества X :

- $x_1 = \langle \text{type} \rangle$ – тип управляющего воздействия (ступенчатое или линейное);
- $x_2 = \langle \text{impact} \rangle$ – относительная амплитуда управляющего воздействия;
- $x_3 = \langle \text{sprev} \rangle$ – прошлое относительное значение выхода объекта;
- $x_4 = \langle \text{scurrent} \rangle$ – относительное значение выхода объекта в момент времени, которое непосредственно предшествует рассчитываемой точке.

Единственная выходная переменная out определяет коэффициент, на который следует умножить текущее значение экспериментального сигнала, чтобы получить искомое значение.

Для исследуемых примеров были выбраны диапазоны лингвистических переменных и количество термов (рис. 1). База правил (таблица) содержит 23 правила, которые определяют в лингвистических терминах выход нечеткого контроллера.

Нечеткая модель на первом этапе вычисляет дискретное значение переменной out_j на основании значений степени истинности термов переменных из множества X ($type$, $impact$, $sprev$, $scurrent$) в соответствии с базой знаний (таблица):

$$\bigcup_{p=1}^P \bigcap_{i=1}^n (x_i = A_i^k) \rightarrow out_j, \quad (3)$$

где A_i^k – лингвистическая оценка (терм) параметра x_i ; k – номер терма в соответствующем терм-множестве; p – номер конъюнктивной цепочки в нечетком правиле; P_j – количество конъюнктивных цепочек в правиле с номером j ; out_j – логический вывод по текущей ситуации в правиле j .

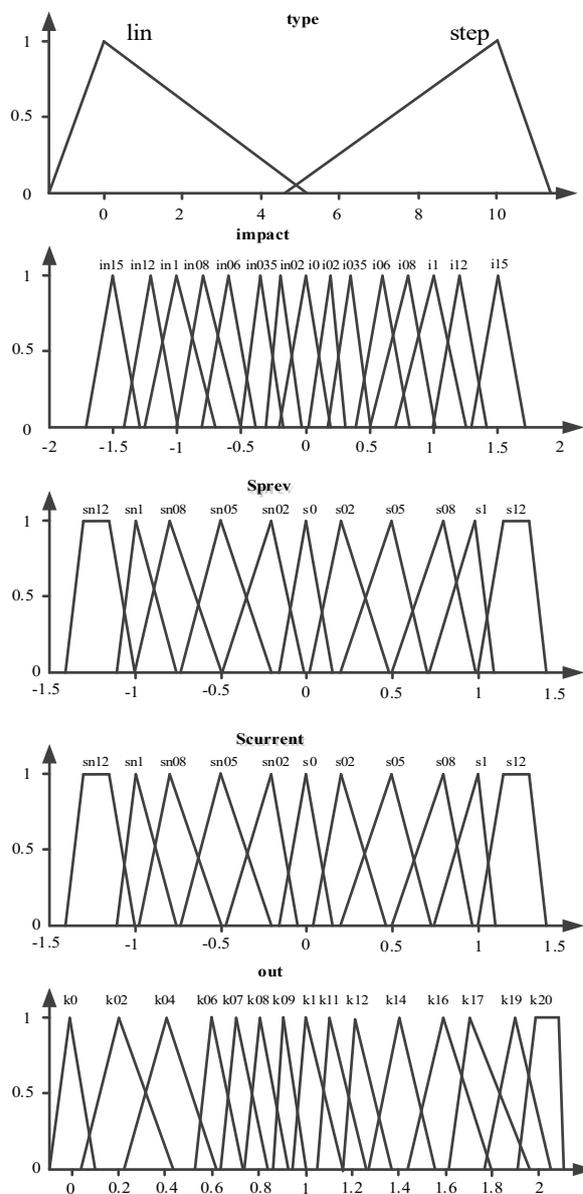


Рис. 1. Функции принадлежности входных и выходных переменных для расчета выхода объекта по экспериментальной последовательности

Преобразование агрегированных нечетких значений out_j производится центроидным методом.

Поскольку расчетная временная последовательность может быть получена путем интерполяции на основании двух имеющихся (ближайших по параметрам) последовательностей, в СЭТ используются два нечетких контроллера, реализующих описанные выше вычислительные функции.

Результаты вычислений нечетких контроллеров усредняются на основании значений расчетных коэффициентов по следующей формуле:

$$S = 0.5 \frac{(1 - |1 - k1|)}{(1 - |1 - k2|)} u1(t) + (1 - 0.5 \frac{(1 - |1 - k1|)}{(1 - |1 - k2|)}) u2(t), \quad (4)$$

где $u1(t)$, $u2(t)$ – текущие значения реальных экспериментальных сигналов; $k1$, $k2$ – расчетные коэффициенты, определенные нечеткими контроллерами для каждой последовательности в отдельности. Формула (4) описывает логику потоковых вычислений для формирования итогового результата.

№ п.п.	Экспертные правила
1	If (type is step) and (impact is i0) and (Sprev is s0) and (Scurrent is s0) then (out is k1)
2	If (type is step) and (impact is i02) and (Sprev is s0) and (Scurrent is s02) then (out is k09)
3	If (type is step) and (impact is i02) and (Sprev is sn02) and (Scurrent is s02) then (out is k08)
4	If (type is step) and (impact is i02) and (Sprev is s02) and (Scurrent is s05) then (out is k07)
5	If (type is step) and (impact is in08) and (Sprev is s0) and (Scurrent is sn08) then (out is k17)
6	If (type is step) and (impact is in1) and (Sprev is s0) and (Scurrent is sn1) then (out is k20)
7	If (type is step) and (impact is in1) and (Sprev is sn08) and (Scurrent is sn1) then (out is k20)
8	If (type is step) and (impact is i0) and (Sprev is sn1) and (Scurrent is sn05) then (out is k14)
9	If (type is step) and (impact is i0) and (Sprev is sn05) and (Scurrent is s0) then (out is k11)
10	If (type is step) and (impact is i0) and (Sprev is sn02) and (Scurrent is s0) then (out is k1)
11	If (type is step) and (impact is in1) and (Scurrent is sn1) then (out is k20)
12	If (type is step) and (impact is in08) then (out is k17)
13	If (type is step) and (impact is i0) and (Sprev is sn1) and (Scurrent is sn1) then (out is k17)
14	If (type is step) and (impact is i0) and (Sprev is sn1) and (Scurrent is sn08) then (out is k16)
15	If (type is step) and (impact is i0) and (Sprev is sn08) and (Scurrent is sn02) then (out is k12)
16	If (type is step) and (impact is i0) and (Sprev is sn08) and (Scurrent is s0) then (out is k1)
17	If (type is step) and (impact is i0) and (Sprev is sn05) and (Scurrent is s0) then (out is k1)
18	If (type is step) and (impact is i02) and (Sprev is s0) and (Scurrent is s0) then (out is k08)
19	If (type is step) and (impact is i02) and (Sprev is s0) and (Scurrent is s02) then (out is k08)
20	If (type is step) and (impact is i0) and (Sprev is s02) and (Scurrent is s02) then (out is k09)
21	If (type is step) and (impact is i0) and (Sprev is sn02) and (Scurrent is sn02) then (out is k11)
22	If (type is step) and (impact is i0) and (Sprev is sn02) and (Scurrent is s02) then (out is k08)
23	If (type is step) and (impact is i0) and (Sprev is s02) and (Scurrent is sn02) then (out is k12)
24	If (type is step) and (impact is in035) and (Scurrent is sn02) then (out is k12)
25	If (type is step) and (impact is in035) and (Scurrent is sn05) then (out is k14)
26	If (type is step) and (impact is i035) and (Scurrent is s02) then (out is k06)
27	If (type is step) and (impact is i0) and (Scurrent is s05) then (out is k06)
28	If (type is step) and (impact is i0) and (Scurrent is s02) then (out is k07)

Таким образом, выражения (3) и (4) в совокупности с базой знаний (таблица) представляют собой нечеткую вычислительную модель для решения задачи интерполяции выходного сигнала объекта управления в процессе эмуляции и тестирования.

Экспериментальная проверка нечеткой модели выполнялась путем модельных экспериментов с сигналами, формируемыми динамическими звеньями первого и второго порядков:

$$W_1(s) = \frac{3}{0,2s + 1}, \quad W_2(s) = \frac{3}{0,1s^2 + 0,2s + 1},$$

$$W_3(s) = \frac{0,4s + 3}{0,05s^2 + 0,1s + 1}.$$

Для экспериментальной проверки были выбраны двухступенчатые управляющие воздействия, при которых искомая последовательность имела такие воздействия, что ее значения в одно время были близки к первой, а в другое – ко второй последовательности, полученных от одного и того же звена. Таким образом, схема проведения модельных экспериментов включает модули задания управляющих воздействий и две одинаковые передаточные функции, моделирующие экспериментальные данные. Их выход моделирует найденные точки, по которым можно определить следующее значение выхода объекта. Начальные условия – нулевые. Два нечетких контроллера, каждый из которых на основании входной информации оценивает близость искомой последовательности к экспериментальной, и формируют расчетный коэффициент. Такая вычислительная процедура обеспечивает больший вес для того результата, чей коэффициент ближе к единице, что дает преимущество более близкой последовательности и позволяет лучше учесть нелинейные свойства объекта управления, если они есть. Рассчитанное и усредненное значение управляемой переменной используется для поиска новых точек в экспериментальных последовательностях и расчетах новых значений входных лингвистических переменных.

Результаты модельных экспериментов показаны на рис. 2. Так как в наличии имеется эталонный результат, то настройку нечеткой экспертной системы можно производить по технологии гибридных нейро-нечетких систем или путем последовательного пересмотра точек переходного процесса и формулирования экспертных правил для равномерно разбитых на термы лингвистических переменных. Выполнив такой процесс для объекта первого порядка, были сформулированы первые 17 экспертных правил. Аналогичные исследования для колебательного звена второго порядка позволили добавить еще 11 правил без изменения уже найденных.

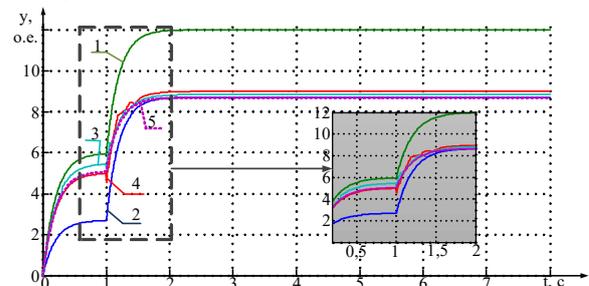


Рис. 2. Результаты модельных экспериментов по эмуляции выходного сигнала динамического звена первого порядка: 1, 2 – входные последовательности; 3, 4 – выходы нечетких контроллеров; 5 – усредненный итоговый результат

Выводы

1. Разработана формальная постановка задачи эмуляции объекта управления и тестирования системы управления для сложного технологического объекта.
2. Разработана теоретико-множественная модель структуры системы эмуляции поведения сложного технологического объекта при его взаимодействии с системой управления. Она отличается наличием интеллектуального агента, реализующего функции отображения множества управляющих воздействий, множества состояний объекта управления во множество выходов объекта, отображение множества функций системы эмуляции на множество целей тестирования, интерфейс с объектом управления и интерфейс с системой управления. Это дает возможность построения эффективной системы эмуляции и тестирования без использования имитационных моделей на основе уравнений динамики сложного объекта управления.
3. Предложена нечеткая вычислительная модель для решения задачи эмуляции поведения объекта управления, отличающаяся тем, что для вычисления реального выходного сигнала используется информация о текущем входном сигнале и нечеткая интерполяция выходного сигнала на основе имеющихся временных последовательностей входных и выходных сигналов объекта управления, что позволяет гибко моделировать поведение объекта управления без разработки и использования систем дифференциальных уравнений, а также повысить эффективность процесса испытаний и отладки системы управления.

Литература: 1. Ушаков В.С. Использование унифицированных имитационных комплексов при создании информационно-управляющих систем в судостроении // Современные технологии автоматизации. 2013. № 3. С. 102–108. 2. Ковалев С.Э. Схемная эмуляция в основе системы графического программирования / Промышленные измерения, контроль, автоматизация, диагностика (ПиКАД)". Киев. 4/2006. Режим доступа: <http://www.picad.com.ua/0406/0406.htm> 3. Ковалев С.Э. Схемная эмуляция в основе системы графического программирования (часть 2): Журнал "Промышленные измерения, контроль, автоматизация, диагностика (ПиКАД)". Киев – 1/2007. Режим доступа: <http://www.picad.com.ua/0107/0107.htm> 4. Волков Г.Я. Энергосбережение и имитационное моделирование. ОДО «Энергоэвент», Минск / Режим доступа: <http://www.energoevent.com/myfiles/Articles/EnergySaving.pdf>. 5. Зарубин В.С. Математическое моделирование в технике: Учеб. для вузов / Под ред. В.С. Зарубина, А.П. Крищенко. М.: МГТУ им. Н.Э. Баумана, 2003. 496 с. (Сер. Математика в техническом университете; Вып. XXI, заключительный). 6. Митюшкин Ю. И., Мокін Б. І., Ротштейн О. П. Soft Computing: ідентифікація закономірностей нечіткими базами знань. Монографія. Вінниця: УНІВЕРСУМ. Вінниця, 2002. 145 с. 7. Девятков В.В. Системы искусственного интеллекта: Учеб. пособие для вузов. М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. 352 с. 8. Конох І.С. Комп'ютеризований лабораторний комплекс для дослідження інтелектуальних цифрових систем керування електроприводом / [І. С. Конох, В. В. Найда, І. С. Гула] // Вісник КДПУ ім. М. Остроградського. Кременчук: КДПУ ім. М. Остроградського, 2008. Вип. 6 (53). Ч. 2. С. 17–21.

Transliterated bibliography:

1. Ushakov V.S. Ispolzovanie unifitsirovannyih imitatsionnyih kompleksov pri sozdanii informatsionno-upravlyayuschih sistem v sudostroenii // Sovremennyye tehnologii avtomatizatsii. 2013. # 3. S. 102–108.
2. Kovalev S.E. Shemnaya emulyatsiya v osnove sistemy graficheskogo programirovaniya / Promyishlennyye izmereniya, kontrol, avtomatizatsiya, diagnostika (PiKAD)". Kiev. 4/2006. Rezhim dostupa: <http://www.picad.com.ua/0406/0406.htm>
3. Kovalev S.E. Shemnaya emulyatsiya v osnove sistemy graficheskogo programirovaniya (chast 2): Zhurnal "Promyishlennyye izmereniya, kontrol, avtomatizatsiya, diagnostika (PiKAD)". Kiev – 1/2007. Rezhim dostupa: <http://www.picad.com.ua/0107/0107.htm>
4. Volov G.Ya. Energoberezhenie i imitatsionnoe modelirovanie. ODO «Energoevent», Minsk / Rezhim dostupa: <http://www.energoevent.com/myfiles/Articles/EnergySaving.pdf>.
5. Zarubin V.C. Matematicheskoe modelirovanie v tehnike: Ucheb. dlya vuzov / Pod red. V.C. Zarubina, A.P. Krischenko. M.: MGTU im. N.E. Baumana, 2003. 496 s. (Ser. Matematika v tehnichestom universitete; Vyip. XXI, zaklyuchitelnyiy).
6. Mityushkin Yu. I., Mokin B. I., Rotshteyn O. P. Soft Computing: identyfikatsiya zakonmirnostey nechitkymy bazamy znan'. Monohrafiya. Vinnytsya: UNIVERSUM. Vinnytsya, 2002. 145 s.
7. Devyatkov V.V. Sistemy iskusstvennogo intellekta: Ucheb. posobie dlya vuzov. M.: Izd-vo MGTU im. N.E. Baumana, 2001. 352 s.
8. Konokh I.S. Komp'yuteryzovanyy laboratornyy kompleks dlya doslidzhennya intelektual'nykh tsyfrovyykh system keruvannya elektropryvodom / [I. S. Konokh, V. V. Nayda, I. S. Hula] // Visnyk KDPU im. M. Ostrohrads'koho. Kremenchuk: KDPU im. M. Ostrohrads'koho, 2008. Vyp. 6(53). Ch. 2. S. 17–21.

Поступила в редколлегия 12.05.2016

Рецензент: д-р техн. наук, проф. Невлюдов И.Ш.

Найда Виталий Владимирович, аспирант кафедры информационно-управляющих систем Кременчугского национального университета им. М. Остроградского. Научные интересы: интеллектуальные информационные технологии процессов тестирования и наладки сложных систем управления. Адрес: Украина, 39600, Кременчуг, ул. Первомайская, 20, E-mail: vitalik.najda@yandex.ru

Оксанич Анатолий Петрович, д-р техн. наук, профессор, заведующий кафедрой информационно-управляющих систем Кременчугского национального университета им. М. Остроградского. Научные интересы: автоматизация сложных технологических процессов, оборудование и технологии производства материалов электронной техники. Адрес: Украина, 39600, Кременчуг, ул. Первомайская, 20, тел.: (05366) 30157. E-mail: oksanich@kdu.edu.ua

Шевченко Игорь Васильевич, д-р техн. наук, доцент, профессор кафедры информационно-управляющих систем Кременчугского национального университета им. М. Остроградского. Научные интересы: интеллектуальные информационные технологии контроля и управления в сложных технологических процессах. Адрес: Украина, 39600, Кременчуг, ул. Первомайская, 20, E-mail: ius.shevchenko@gmail.com

Конох Игорь Сергеевич, канд. техн. наук, доцент, доцент кафедры информационно-управляющих систем Кременчугского национального университета им. М. Остроградского. Научные интересы: промышленная автоматизация, интеллектуальные системы управления технологическими процессами. Адрес: Украина, 39600, Кременчуг, ул. Первомайская, 20, E-mail: kis_sau@mail.ru