



ФОРМАЛЬНОЕ ОПИСАНИЕ АСУ СЛОЖНЫМИ ТЕХНОЛОГИЧЕСКИМИ ОБЪЕКТАМИ И АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ АЛГОРИТМОВ УПРАВЛЕНИЯ

В.Е. Захарченко (ООО НВФ "Сенсоры. Модули. Системы")

Рассматриваются принципы формализованного описания АСУ сложными технологическими объектами, анализируются их недостатки, предлагается модель описания АСУТП. Приводится пример практического применения этой модели при формальном описании системы контроля и управления гидроагрегатом, внедренной на Жигулевской ГЭС. Исследуется возможность автоматизации тестирования алгоритмов управления. Описывается принцип построения автоматизированной системы тестирования алгоритмов на основе программно-технических средств фирмы Siemens.

Способы описания и моделирования сложных систем

В настоящее время известны несколько способов формализованного описания АСУ сложными технологическими объектами. Рассмотрим наиболее распространенные из них.

"Режимное" описание систем

Одной из черт сложных систем является наличие нескольких качественно различных, последовательно сменяющих друг друга во времени. Инженеры обычно называют их "режимами функционирования". Например, бассейн с двумя трубами обладает сложным поведением, поскольку реакция системы управления на переполнение качественно отличается от реакции при нормальном уровне и при опустошении. Такой тип сложного поведения можно реализовать, если описать всю совокупность допустимых, простых, в некотором смысле, частных поведений (возможно, это будет иерархическая структура), и указать правила переключения с одного режима управления на другой.

Каждое конкретное поведение можно отождествить со значением некоторой дискретной переменной, а мгновенные переключения текущего поведения — с дискретными событиями. Для передачи информации о дискретных событиях в другие блоки используют специальные переменные — сигналы. Набор дискретных состояний вместе с условиями переходов из одного состояния в другое образует обычное дискретное поведение. В моменты переходов могут происходить мгновенные скачкообразные изменения значений переменных. Поскольку в каждом из дискретных состояний элементарный блок ведет себя как некоторая непрерывная система, то поведение блока в целом является непрерывно-дискретным или гибридным (www.exponenta.ru/soft/others/mvs/ds_sim.asp). Граф переходов, узлам которого приписаны некоторые непрерывные отображения, а дугам — условия переходов и выполняемые действия, называется гибридным автоматом.

Графовое описание систем

Изучение отношений между элементами и подсистемами, определение роли и места каждой подсистемы в общем процессе функционирования системы составляют предмет структурного анализа. Так как схема со-

пряжения с любой сложной системой представляется как совокупность предикатов, определенных на множестве входов/выходов ее элементов, то для изучения структуры сложной системы используют аппарат математической логики и теории графов (ooad.asf.ru/standards/uml/umlsystem/index.asp). Система представляется в виде графа, который характеризуется множествами состояний и переходов из одного состояния в другое [3]. При таком подходе к описанию систем управления ТП отсутствует формализация управления.

Описание сложных динамических систем

Сложную динамическую систему можно представить как некоторую детерминированную совокупность элементов (подсистем), каждая из которых в процессе своего функционирования преследует некоторую цель $F_i(S_i)$. Цели и задачи сложной системы достигаются в результате выполнения определенной композиции, состоящей из множества целевых функций ее подсистем:

$F(S) = \Phi[F_1(S_1), F_2(S_2), \dots, F_n(S_n)]$, где S — сложная система, S_1, \dots, S_n — ее подсистемы, F_1, \dots, F_n — цели функционирования соответствующих подсистем, Φ — математическое описание закономерных связей между перечисленными целями. Предполагается, что подсистема S_i сложной системы, как и вся система S в целом, функционирует во времени, и в каждый момент времени t она находится в одном из возможных состояний $q_i(t)$. С течением времени подсистема и система в целом под воздействием внешних и внутренних факторов переходят из одного состояния в другое; в процессе функционирования системы она взаимодействует с внешней средой и другими системами, получая от них входной поток $X(t)$ и выдавая выходной поток $Y(t)$ событий, энергетических или материальных объектов. Существует несколько способов математической формализации таких процессов. К ним относятся Марковские процессы, сети Петри, семантические сети, конечные автоматы и алгоритмы (ustenko.fromru.com).

В таком описании каждое из состояний системы — "черный ящик", имеющий входные/выходные параметры, при этом система рассматривается как управляющая и управляемая. Остается определить $X(t)$ и $Y(t)$.

Описание сложных систем управления

Из указанных способов только в описании динамических систем присутствует управление, в остальных управление можно задать неявно. В области технологической автоматики управление определяет поведение объекта, и очень важно, чтобы управление было своевременным и достаточным. Математическое определение способов управления $Y(t)$ можно выразить с помощью алгоритмов [5].

Тогда любую систему управления O технологическими объектами можно представить как тетраду:

$$O = (P, S, A, C),$$

где P – множество параметров $P_1..P_N$ (аналоговых и дискретных), количественно определяющих соответствующие свойства объекта управления. Свойство объекта (системы объектов) – это некоторая числовая, геометрическая, физическая или любая другая, в том числе вообразаемая, его характеристика, стабильная при определенных стабильных условиях, по которой объекты и (или) конструкции могут отличаться друг от друга или от своего предшествующего во времени и/или пространстве состояния. Состоянием объекта называется совокупное конкретное значение всех его свойств [6]. Состояние системы управления определяется конкретными значениями параметров;

S – множество состояний $S_1..S_M$ объекта (системы) управления; A – множество алгоритмов управления $A_1..A_M$ системой, каждый из которых соответствует состоянию $S_i, i \in \overline{1, M}$; C – матрица переходов S_i в состояние $S_k, i, k \in \overline{1, M}$.

Поскольку алгоритм управления может иметь сложную разветвленную структуру, каждый элемент множества алгоритмов A_i , присущих для состояния S_i представляет собой набор реализаций алгоритмов $A_i = \{a_{i1}..a_{ir}\}$, где a_{ic} – вариант выполнения алгоритма $A_i, c \leq r, r > 0$ – общее число вариантов выполнения алгоритма A_i ; и $A_i = 0$, если в состоянии S_i нет алгоритмов.

Следует отметить, что инициализация того или иного алгоритма однозначно определяется значениями параметров системы, их можно описать системой условий. Множество состояний и переходов можно представить в виде графа [3, 4].

Особенности описания модели для систем управления РВ

В системе РВ правильность ее функционирования зависит не только от логической корректности вычислений, но и от времени, за которое эти вычисления производятся. Иными словами, система работает в РВ, если ее быстродействие адекватно скорости протекания физических процессов на объектах контроля или управления [7]. Практически все системы промышленной автоматизации являются системами РВ.

Тогда модель системы управления РВ $O(t)$ можно представить следующим образом:

$$O(t) = (P, S, A(t), C(t)),$$

где $A(t)$ и $C(t)$ приобретают ограничения по времени t .

¹ В зависимости от типа системы РВ (жесткого или мягкого) описанные ограничения могут незначительно изменяться.

Возможны следующие ограничения по времени¹: действие (последовательность действий) должно быть выполнено через фиксированное время от заданного события/действия или за время, не превышающее заданное время от определенного события/действия.

Система контроля и управления гидроагрегатом Жигулевской ГЭС

Описанная модель систем управления применялась при создании системы контроля и управления гидроагрегатом на Жигулевской ГЭС, реализованной на базе ПЛК SIMATIC S7 400-й серии фирмы Siemens. Система была внесена в Государственный реестр средств измерений как комплекс программно-технического контроля и управления гидроагрегатом "Гидроконт 400". В системе контролируются около 250 параметров, и выделено несколько состояний (рисунок).

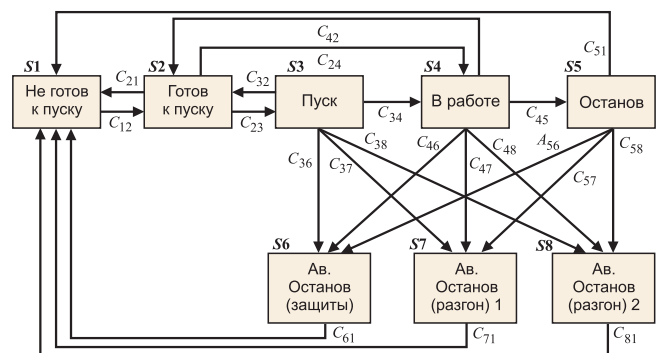
Для систем управления Жигулевской ГЭС разработан программный имитатор параметров системы. Объединив ЛСКУ и имитатор, можно автоматизировать процесс тестирования алгоритмов управления.

Автоматизация тестирования алгоритмов управления

Алгоритмы управления – каркас систем управления ТП. Именно в них задаются порядок и мера воздействий на объект управления в зависимости от состояния объекта. Будем считать, что алгоритмы написаны в полном соответствии с техническим заданием к проекту. Алгоритмы управления, не соответствующие техническому заданию, не рассматриваются.

Для проверки алгоритмов системы необходимо протестировать каждую ветвь каждого алгоритма минимум один раз. Для этого на вход системы нужно подать определенную комбинацию параметров. Если говорить об автоматизации проверки алгоритмов, то комбинация параметров, подаваемая на вход, должна определяться автоматически на основании описания алгоритмов.

Проверка выполнения ветки алгоритма заключается в следующем: система переводится в начальное состояние, инициализируется выполнение алгоритма, затем устанавливаются значения параметров системы в порядке и объеме, необходимом для выполнения конкретной ветви алгоритма. В процессе фиксирует-



Граф состояний и переходов локальной системы контроля и управления (ЛСКУ) гидроагрегатом

ся факт выдачи управляющих воздействий, контролируется время выполнения, длительность задержек, изменения состояния системы.

В системах управления технологическими объектами присутствует механизм информационных технологических сообщений, с помощью которых можно определить, в каком состоянии находится объект управления, выполняется ли в данный момент какой-либо алгоритм, стадия завершения алгоритма и т.п. Неотъемлемой частью технологического сообщения является метка времени.

Таким образом, при наличии достаточного числа технологических сообщений можно контролировать временные зависимости алгоритма управления.

Предпосылки создания автоматизированной системы тестирования алгоритмов в АСУТП

На текущий момент в АСУТП тестирование алгоритмов управления, реализованных в контроллере, осуществляется "вручную". Для сложных объектов создают специальные имитаторы: программные, аппаратные и программно-аппаратные. Основными их недостатками, как правило, являются оторванность от реальности и высокая стоимость.

В процессе тестирования алгоритмов управления обычно принимают участие сотрудники, имитирующие изменения состояния объекта управления и наблюдающие за реакцией системы управления. Имитация одного-двух параметров обычно не вызывает особых трудностей, но что делать, если нужно имитировать несколько десятков параметров, или среди имитируемых параметров есть такие, которые нужно изменить за очень малое для человека или жестко фиксированное время?

Также автоматизация тестирования будет полезна в случаях, когда несколько систем имеют однотипную алгоритмическую структуру или алгоритмы системы управления должны периодически проверяться (например, раз в год). Периодическая проверка алгоритмов интересна в смысле комплексного тестирования. В этом случае можно говорить о тестировании экземпляров алгоритмов.

В результате тестирования вручную создается отчет. Если однотипных объектов несколько, и все они имеют совпадающие алгоритмы, то отчеты по тестированию этих объектов необязательно будут идентичными в силу человеческого фактора.

Недостатки "ручного" тестирования алгоритмов управления систем:

1. невозможность имитации большого числа сигналов одновременно;
2. невозможность подачи серии сигналов с малым интервалом времени (например, 500 мс) или с фиксированным интервалом;
3. отчет также должен создаваться "вручную", то есть затрачивается время и усилия персонала;
4. отчет может не иметь постоянной формы.

Таким образом, имеет смысл автоматизировать следующие функции: имитацию значений всех управ-

ляющих параметров системы; имитацию любого состояния системы из всего множества состояний S ; задание комбинации изменений с заданным интервалом времени; фиксацию времени появления соответствующих сообщений; формирование отчета. Если все перечисленные функции будут выполняться автоматически, то оператор должен руководить последовательностью прохождения тестов и следить за выдачей управляющих сигналов. Теоретически можно автоматизировать и эти области, но тогда встанет задача проверки самой системы тестирования алгоритмов.

Если АСУТП можно описать некоторой моделью, то создание автоматизированной системы проверки алгоритмов будет значительно облегчено.

Инженерное решение задачи автоматизации тестирования алгоритмов на базе программно-аппаратных средств фирмы Siemens.

Под имитацией значений параметров системы, в данном случае, понимается возможность программной подмены реального значения параметра системы на заданное оператором. Для автоматизации тестирования система управления должна поддерживать такую возможность. Для имитации можно использовать:

- стандартные средства SCADA-системы;
- специальное пользовательское приложение,

способное читать и записывать в ячейки памяти контроллера данные, заменяющие значения датчика на пользовательские. Например, для контроллеров Siemens 300/400 серии можно использовать функции специальной библиотеки PRODAVE (этот способ применен на Жигулевской ГЭС) или различные коммуникационные возможности контроллеров (например, по сети Industrial Ethernet посредством протоколов TCP, OPC и др.).

Используя представленную модель описания АСУТП, рассмотрим четыре шага, необходимые для создания автоматизированной системы (подсистемы) проверки алгоритмов управления.

1. Создание БД или файла с описанием состояний системы S , где для каждого состояния определяются допустимые значения всех параметров управления.

2. Описание в БД или в файлах всех реализаций алгоритмов $A(t)$, которые необходимо проверить, в виде цепочки событий, то есть в виде последовательности изменений параметров.

3. Описание матрицы переходов $C(t)$, чтобы проверить все возможные способы запуска алгоритма. Правильность выполнения алгоритма можно определить по изменению состояния системы и технологическим сообщениям с меткой времени. Каждой ветви алгоритма будет соответствовать комбинация сообщений и изменений состояний системы. Для реализации автоматического поиска необходимого сообщения полезно знать устройство системы сообщений. Рассмотрим два способа устройства системы сообщений:

- использование циклического буфера контроллера, рассчитанного на ограниченное число сообщений;

- 200...1000 ед. Читать данные из буфера можно, подключившись к контроллеру по сети MPI\PROFIBUS\ Industrial Ethernet с удаленного ПК посредством пользовательского приложения;

- использование средства Alarm Logging SCADA-системы SIMATIC WinCC версии 6.0. Система сообщений WinCC хранит данные в таблицах SQL Server 2000 (порядка нескольких миллионов сообщений). Доступ к этим данным осуществляется помощью открытых интерфейсов WinCC Open Development Kit или OLE DB.

4. Выявление соответствий между описанием последовательности действий алгоритмов и комбинации сообщений и состояний системы. Тогда станет возможным автоматический поиск сообщений и автоматическое определение изменения состояния системы.

Формирование отчета реализуется стандартными средствами языков программирования.

Оператор, используя графический интерфейс системы тестирования, указывает исходное состояние системы управления, после чего, выбрав соответствующую ветвь алгоритма, контролирует ее выполнение. Система тестирования имитирует необходимые значения параметров, по окончании алгоритма считает соответствующие сообщения с метками времени и сгенерирует отчет.

*Захарченко Виталий Евгеньевич – инженер ООО НВФ "Сенсоры. Модули. Системы".
Контактный телефон (846)269-15-20. E-mail: zevs@sms-samara.ru*

Заключение

Таким образом, в работе представлен способ формального описания АСУТП. Показана возможность применения разработанной модели на примере системы контроля и управления гидроагрегатом на Жигулевской ГЭС. Описан типовой механизм тестирования алгоритмов управления на базе предложенной модели АСУТП.

Список литературы

1. *D.Harel, Statecharts: A Visual Formalism for Complex Systems. Science of Computer Programming. North-Holland. Vol.8. No.3. 1987.*
2. *Booch G., Jacobson I. and Rumbaugh J. The Unified Modeling Language for Object-Oriented Development. Documentation Set Version 1.1. September. 1997.*
3. *Бусленко Н.П., Калашиников В.В., Коваленко И.Н. Лекции по теории сложных систем. М. 1973.*
4. *Жуков Д.Ю. Методы описания и анализа распределенных систем. 2000.*
5. *Robin Milner. Communication and Concurrency. Prentice-Hall. 1989.*
6. *Крылов С.М. Формально-технологические модели в общей теории систем // Известия Самарского научного центра РАН. Том 5. 2003. №1.*
7. *Сорокин С.А. Системы реального времени // Современные технологии автоматизации. 1997. №2.*