

УДК 681.518.5

ПОДХОД К РАЗРАБОТКЕ ПРОГРАММ КОНТРОЛЯ С ИСПОЛЬЗОВАНИЕМ МОДЕЛЕЙ СОСТОЯНИЯ СЛОЖНЫХ ТЕХНИЧЕСКИХ СИСТЕМ

Семенов И.П., Третьяков В.В. (Москва)

Введение

Сложные технические системы (СТС) представляют собой сложные по внутренним связям и большие по количеству элементов системы, трудно поддающиеся прямым способам моделирования [1]. В состав СТС интегрированы различные электронные, электромеханические, механические и прочие приборы автоматики (ПА), например, электромеханические или твердотельные реле, силовые ключи, датчики давления, временные устройства и пр. Приборы объединены между собой большим количеством функциональных связей, а режимы и алгоритмы их работы зависят от работы и состояний других ПА.

Разработка СТС предусматривает создание систем контроля, предназначенных для контроля технического состояния СТС путем измерения параметров выходных сигналов и проверки алгоритмов работы различных встроенных элементов. Контроль технического состояния СТС проводится с использованием контрольно-измерительной аппаратуры (КИА) и осуществляется под управлением программ контроля, написанных на соответствующем языке программирования. Функционально системы контроля могут быть построены по-разному, например, с элементами самодиагностики [2]. Далее рассматриваются системы контроля, внешние по отношению к проверяемому объекту, как обладающие наиболее широкими возможностями диагностирования. Функциональные схемы таких систем контроля приведены на рис. 1. Программы контроля состоят из определенного набора операций A , регистрирующих несоответствия технического состояния СТС требуемому путем:

- выдачи на СТС внешних воздействий X ;
- измерения выходных сигналов Y в процессе работы СТС;
- оценки соответствия результатов измерений требованиям Y_M к СТС.

В процессе разработки программ контроля и систем контроля для проверок СТС используются индивидуальные решения и методы, зависящие от опыта и квалификации разработчиков, что может приводить к сложностям при сопровождении программ контроля: большим временным затратам на доработку, установление причины неисправности, в обеспечении проверок СТС и обучении молодых специалистов.

Различия в программах контроля в большинстве своем обусловлены способами задания требуемого состояния проверяемой системы на всех этапах алгоритма контроля. Как правило, используются решения, требующие жестко фиксировать состояние системы в программном коде. Следует отметить, что одно и то же решение может быть реализовано несколькими способами. Зачастую это приводит к увеличению времени разработки и сопровождения, ошибкам, дублированию или разрастанию кода.

В работе предлагается подход к разработке программ контроля с использованием моделей состояния СТС с целью:

- снижения количества ошибок;
- увеличения скорости разработки;
- повышения диагностических качеств систем контроля;
- упрощения сопровождения программ контроля.

Особенности программ контроля с использованием модели состояний

Подход к разработке с использованием модели состояний предполагает программную реализацию имитационной модели M СТС в программе контроля. В отличие от подхода с жестким фиксированием состояний СТС в данном подходе состояние СТС рассчитывается в процессе выполнения программы, а не задается заранее в виде массивов Y_M требований к выходным сигналам. Функциональная схема системы контроля, построенной с использованием подхода с использованием модели состояний, представлена на рисунке 1Б.

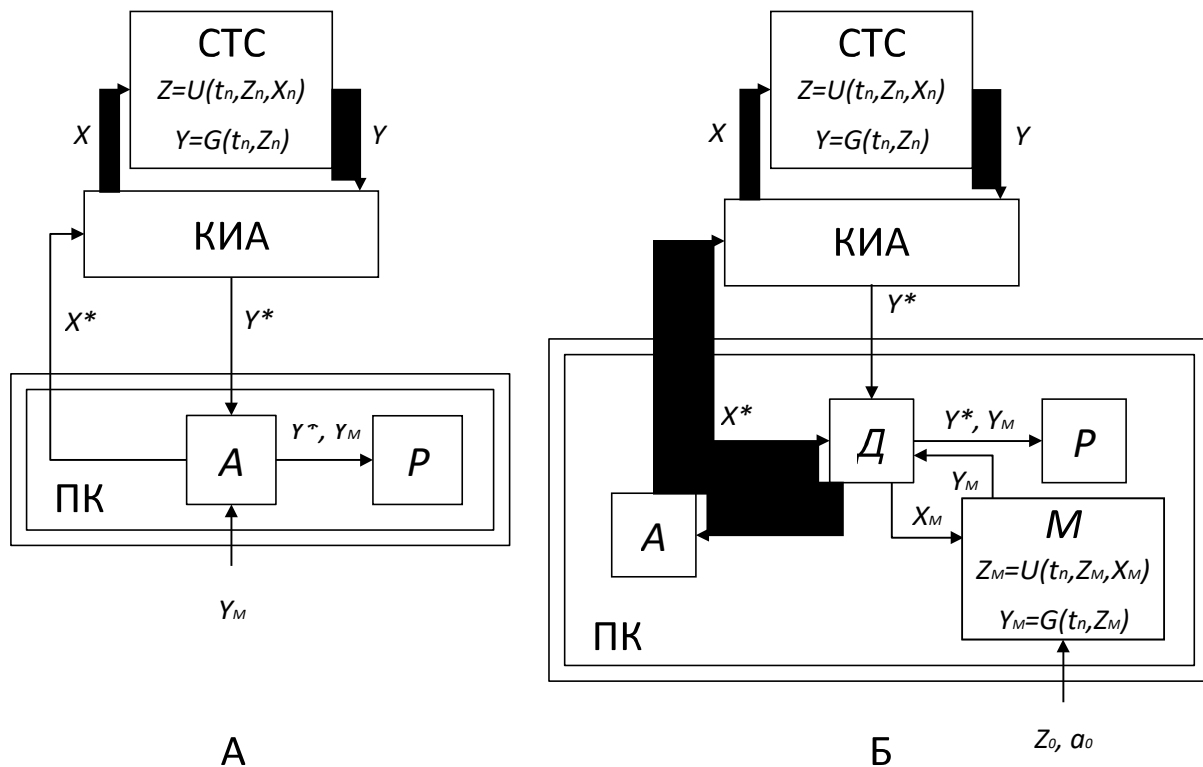


Рис. 1. Упрощенные функциональные схемы систем контроля СТС

А – подход с жестко фиксированными состояниями СТС

Б – подход с использованием модели состояний СТС

Разнообразие физических принципов, на которых основана работа элементов СТС и сложность построения структуры СТС обуславливает использование агрегативной математической схемы (А - схемы) для их моделирования [1,3]. Таким образом, модель M СТС требуется для симуляции Y_M откликов Y^* на инициируемые программой контроля воздействия X^* и в общем виде представляет собой программную реализацию агрегативной математической схемы ($Z_M=U(t_n, Z_M, X_M)$, $Y_M=G(t_n, Z_M)$), состоящей из: агрегативных моделей составных частей СТС – приборов автоматики и интерфейсных агрегатов, моделирующих функциональную связь ПА в СТС, их влияние друг на друга. Сложность модели состояний, используемой в программе, определяется не только сложностью объекта контроля, но и возможностью, а также необходимостью наблюдения внутренних процессов, происходящих в элементах СТС, для оценки ее технического состояния. Стоит отметить, что, преимущественно, элементы СТС могут быть описаны с помощью более простой математической схемы – в виде конечных автоматов. Таким образом, для упрощения программ контроля целесообразно моделировать лишь те процессы, которые позволяют выполнять оценку состояния СТС или приводят к его смене. Это утверждение позволяет исключить из моделей составных частей СТС сложные расчеты

переходных процессов, которые заменяются событиями, происходящими в дискретные моменты времени. Для реализации предложенного подхода наилучшим образом подходит технология дискретно-событийного моделирования [4,6].

Реализация программ контроля с использованием модели состояний

Применительно к программам контроля подход с использованием модели состояний СТС основывается на принципе работы диспетчера D , показанного на рисунке 1 и реализованного посредством поведенческого шаблона проектирования «Посредник» (Mediator) [5, 6]. Использование диспетчера позволяет инкапсулировать взаимодействие между несколькими модулями программы, ослабить связи между ними, сделать структуру программы контроля более гибкой и масштабируемой.

Программная реализация модели содержит определенные структуры данных, как правило:

- массив состояний элементов (приборов), так как количество приборов в различных системах и в вариантах исполнения одной и той же СТС может быть различным;

- массив состояний выходных сигналов (электрических цепей, контактов и пр.), так как количество выходных сигналов, которые возможно контролировать, различается от системы и варианта исполнения;

- массив цепей, описывающий адреса доступных для проверки выходных сигналов, к которым подключаются различные измерители КИА или на которые выдаются внешние воздействия, так как для каждой системы разрабатываются индивидуальные комплекты принадлежностей и схемы подключения изделий к КИА сильно различаются.

Значения элементов массивов состояний приборов и массивов состояний контактов автоматически изменяются в процессе работы программы контроля в результате вызова метода пересчета состояния контактов приборов, зависящего от откликов СТС на внутренние или внешние воздействия.

Программная реализация диспетчера D подразумевает использование следующих методов:

1. Метод инициализации начального состояния. Метод объявления глобальных констант a_0 и массивов Z_0 начального (исходного) состояния СТС.

2. Метод выдачи внешних воздействий A . Данный метод выдает на СТО внешние воздействия X^* с различными качественными и количественными характеристиками в соответствии с набором тестовых воздействий СТС. Внешние воздействия могут содержать: импульсы напряжения различной длительности и амплитуды, кодовые команды по различным цифровым интерфейсам, температурные и механические воздействия, создание избыточного давления или разности давлений, временные задержки и т.д.

3. Метод пересчета состояния контактов всех приборов, задействованных в СТС. Метод в зависимости от пройденных режимов работы i , включения или отключения различных приборов, выданных или полученных команд, изменяет элементы массивов состояний приборов и контактов. То есть, данный метод описывает поведение всей системы и преобразует вектор выходных сигналов системы в новый вектор Y_M выходных сигналов СТС.

4. Метод оценки реального состояния системы и сравнения его с состоянием, вычисленным с помощью имитационной модели СТС. Метод подключает специальные измерительные элементы к адресам СТС, объявленным в методе с описанием массивов. Метод путем опроса всех соединений из массива цепей формирует реальное состояние системы Y^* и сравнивает его с состоянием Y_M , ранее полученным при вычислении в

имитационной модели. При несоответствии происходит обращение к методу вывода диагностических сообщений P и завершение исполнения программы контроля.

5. Метод вывода диагностических сообщений P . Метод выводит на интерфейс взаимодействия с оператором диагностическую информацию о выявленном несоответствии в СТС (например, печатает адреса цепей, объявленные в массивах, измеренное состояние системы, состояние, рассчитанное в имитационной модели и пр.).

Принцип работы программ контроля с использованием модели состояний

На рисунке 2 изображен принцип работы программ контроля с использованием модели состояний, формализованный в результате обобщения опыта разработки программ контроля СТС.

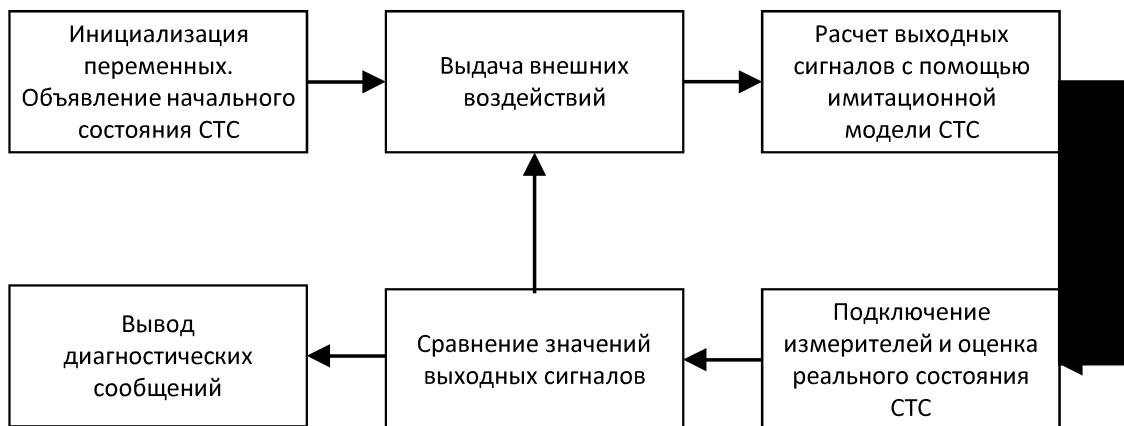


Рис. 2. Принцип работы программ контроля с использованием модели состояний

Изначально происходит вызов метода с объявлением переменных и описанием массивов. Объявляется начальное (исходное) состояние Z_0 СТС. Происходит инициализация имитационной модели СТС, моделей составных частей СТС и моделей, описывающих функциональную связь ПА в СТС. Например, для простого фрагмента СТС, приведенного на рисунке 3, будут объявлены следующие массивы:

- массив состояний элементов $Z_0 = (1,1,1)$, указывающий включенное состояние трех реле (приняты обозначения 1 – реле включено, 0 – реле отключено);
- массив состояний выходных сигналов $Y_{M0} = (1)$;
- массив цепей $a = (X - Y)$.

Далее диспетчер D в соответствии с алгоритмом проверки (набором тестовых воздействий СТС) вызывает метод выдачи внешних воздействий A . По завершении работы метод передает в диспетчер информацию о своем выполнении и о параметрах выданных воздействий X^* . Для фрагмента СТС, приведенного на рисунке 3, могут быть выданы следующие воздействия:

- отключение одного из трех реле K_2, K_3, K_4 ;
- отключение пары реле совместно K_2 и K_3, K_2 и K_4, K_3 и K_4 ;
- отключение всех трех реле совместно.

Диспетчер вызывает метод пересчета состояния контактов всех приборов, задействованных в СТС, передавая ему информацию о выданных внешних воздействиях X_M . Основываясь на полученной информации и предыдущем состоянии, метод посредством описанной имитационной модели СТС рассчитывает новый вектор Y_M выходных сигналов СТС и отправляет его в диспетчер. Для фрагмента СТС,

приведенного на рисунке 3, модель будет представлять собой тривиальное логическое выражение:

$$Y_{UM} = (K2.2 \wedge K4.2 \wedge K2.5) \vee (K3.2 \wedge K4.2 \wedge K3.5) \vee (K3.2 \wedge K2.5) \vee (K2.2 \wedge K3.5) = \\ = (K2 \wedge K3) \vee (K2 \wedge K4) \vee (K3 \wedge K4).$$

Диспетчер, получив вектор Y_M выходных сигналов системы, рассчитанный с помощью имитационной модели СТС, вызывает метод оценки реального состояния системы. Данный метод подключает к контактам СТС различные измерители КИА и обрабатывает информацию с них, формируя тем самым вектор Y^* измеренных выходных сигналов системы, который он направляет в диспетчер по завершению своей работы. Так к входу и выходу мажоритарного переключателя X и Y соответственно будут, например, параллельно подключены измерительный элемент и источник напряжения амплитудой 5 В через резистивную нагрузку номиналом 500 Ом. Измерительный элемент производит пороговую оценку фрагмента по сопротивлению. При измеренном значении $R_n < 50 \text{ Ом}$ – состояние $Y^* = 1$, при измеренном значении $R_n > 100 \text{ Ом}$ – состояние $Y^* = 0$.

Диспетчер, получив вектор Y^* измеренных выходных сигналов и вектор Y_M выходных сигналов, рассчитанный с помощью имитационной модели СТС, сравнивает их между собой. При отрицательном результате сравнения диспетчер вызывает метод P выдачи диагностических сообщений, выводит на печать адреса соединений, в которых зарегистрировано несоответствие, и завершает работу программы. При положительном результате сравнения диспетчер вызывает метод A выдачи новых внешних воздействий, и алгоритм работы программы повторяется, пока не выдадутся все внешние воздействия в соответствии с набором тестовых воздействий СТС.

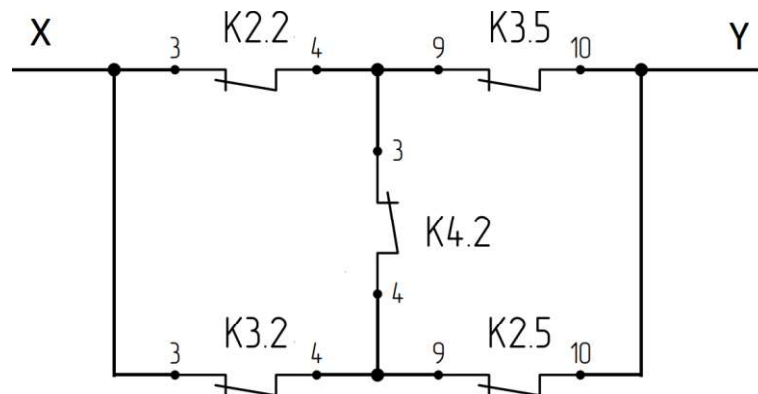


Рис. 3. Фрагмент СТС. Мажоритарный переключатель

Достоинства программ контроля с использованием модели состояний

С применением подхода к разработке программ контроля с использованием модели состояний и определенных шаблонов можно значительно упростить и сократить сроки разработки программ контроля новых СТС, так как в подавляющем большинстве случаев исчезает необходимость разрабатывать индивидуальные программные методы и решения.

С применением решений, требующих жестко фиксировать состояние системы в программном коде, для разработки новых программ контроля разработчику необходимо пересчитывать все проверяемые состояния СТС и переписывать значительную часть программного кода. Например, для рассмотренного фрагмента схемы требуется реализовать в программе контроля выдачу внешних воздействий и контроль состояния цепи $X - Y$ в соответствии с таблицей 1. При увеличении

количества цепей и элементов СТС сложность программной реализации резко возрастает.

При применении подхода к разработке программ контроля с использованием модели состояний необходимо внести незначительные корректировки в шаблонные методы, реализованные в других проектах, при этом большая часть программного кода остается нетронутой за счет реализации диспетчера *Д*. В случае отсутствия необходимых методов и моделей разработчику программ контроля требуется выполнить реализацию метода или имитационной модели с помощью специального языка программирования и дополнить проект.

Для разработки новой программы контроля необходимо:

- внести корректировки в метод инициализации начального состояния (изменить количество проверяемых приборов, изменить количество проверяемых внешних сигналов, изменить адреса и количество цепей в массиве цепей);
- изменить имитационную модель СТС и описать новую последовательность операций, происходящих в новой системе и зависящих от определенных внешних и внутренних воздействий;
- внести корректировки в метод вывода диагностических сообщений.

Таблица 1

Описание метода выдачи внешних воздействий с расчетом состояния СТС

№ шага	Псевдокод для выдачи внешнего воздействия	Состояние элементов СТС после воздействия			Состояние цепи X – Y	Псевдокод для контроля состояния СТС
		K2	K3	K4		
0	Отсутствует (СТС в исходном состоянии)	1	1	1	1	ОценкаЦепи(XY,1);
1	Отключить(K4);	1	1	0	1	ОценкаЦепи(XY,1);
2	Отключить(K3);	1	0	0	0	ОценкаЦепи(XY,0);
3	Включить(K4);	1	0	1	1	ОценкаЦепи(XY,1);
4	Отключить(K2);	0	0	1	0	ОценкаЦепи(XY,0);
5	Включить(K3);	0	1	1	1	ОценкаЦепи(XY,1);
6	Отключить(K4);	0	1	0	0	ОценкаЦепи(XY,0);
7	Включить(K2);	1	1	0	1	ОценкаЦепи(XY,1);
8	Отключить(K2,K3);	0	0	0	0	ОценкаЦепи(XY,0);
9	Включить(K2,K3,K4);	1	1	1	1	ОценкаЦепи(XY,1);

Таким образом, структура программного кода остается неизменной, меняется только размерность массивов, используемых для оценки состояния системы. При этом вызов метода оценки состояния системы не требует передачи в него аргументов. Такое построение программ контроля позволяет исключить лишние структуры и условия, что приводит к лучшей читаемости и масштабируемости программного кода, что в следствие уменьшает порог вхождения разработчиков и сокращает временные затраты на доработку или установление причины неисправности СТС.

Также значительно уменьшается вероятность возникновения ошибок при разработке новых программ контроля, так как необходимо вносить изменения только в небольшую часть кода, которая соответствует описанию работы системы.

Сделанные выводы нашли практическое подтверждение на примере проекта по разработке программы контроля СТС, выполненного авторами в условиях недостаточно определенных технических требований. Изначально применен подход с жестким фиксированием состояний СТС, содержащий несколько сотен фрагментов, аналогичных приведенному на рисунке 3. На анализ СТС и разработку программы

было затрачено три недели рабочего времени. После уточнения технических требований к СТС на корректировку программы затрачено еще две недели из-за большого количества ошибок в расчете требуемого состояния СТС. После очередной корректировки требований к СТС программа контроля была переработана с использованием модели состояний за две недели. С применением данного подхода к разработке последующие изменения СТС не приводили к большим временным затратам, и программы контроля дорабатывались в течение одного-двух дней при сравнимом объеме изменений СТС. Приведенные временные оценки являются субъективными, однако пример демонстрирует общий эффект – подход с использованием модели состояний позволяет увеличить скорость разработки программ контроля СТС на порядок.

Кроме того, предложенный подход предполагает контроль состояния всех приборов системы в определенные промежутки времени, что в сравнении с используемыми на сегодняшний момент решениями позволяет улучшить диагностические качества программ контроля, так как после определенных воздействий будет оцениваться состояние всей системы в целом, а не ее отдельных частей. Поскольку системы контроля осуществляют проверку технического состояния СТС и, по сути, реализуют критерий соответствия СТС заданным требованиям, то программы контроля проходят полный цикл тестирования и испытаний на заведомо исправном СТС. Тем самым обеспечивается высокая степень достоверности работы модели. Несоответствия в работе модели и СТС, возникающие в дальнейшем, в подавляющем большинстве случаев рассматриваются как отказы СТС и требуют ее ремонта.

Заключение

Внедрение подхода к разработке программ контроля с использованием имитационных моделей СТС позволяет:

- достичь лучшей масштабируемости программного кода;
- достичь лучшей читаемости программного кода;
- снизить количество ошибок при разработке;
- увеличить скорость разработки;
- повысить диагностические качества программ контроля;
- упростить сопровождение программ контроля.

Литература

1. Советов Б.Я., Яковлев С.А. Моделирование систем: учеб. для вузов. 3-е изд., перераб. и доп. М.: Высш. шк., 2001. 343 с.
2. Мальцев Г.Н., Назаров А.В., Якимов В.Л. Имитационное моделирование процесса диагностирования сложной технической системы с высоким уровнем автономности функционирования // Моделирование систем и процессов. 2016. № 4. С. 34–43.
3. Бусленко Н.П., Калашников В.В., Коваленко И.Н. Лекции по теории сложных систем. М.: Советское радио, 1973. 440 с.
4. Родионов А.С. Разработка систем дискретного имитационного моделирования информационных сетей: автореф. дис. ... докт. техн. наук. Новосибирск, 2002.
5. Гамма Э., Хелм Р., Джонсон Р., Влассидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Питер, 2001. 368 с.
6. Бхаргава А. Грокаем алгоритмы: иллюстрированное пособие для программистов и любопытствующих. СПб.: Питер, 2017. 288 с.