

УДК 518.001.57:621.039.5+681.14

Принципы построения имитационной модели автоматизированной системы контроля и управления ядерной энергетической установкой

ГАВРИЛОВ П. А., ТРЕХОВ В. Е.

Имитационное моделирование, получившее в последнее время широкое распространение, является универсальным и мощным средством исследования сложных систем. Для некоторых систем, натурные эксперименты с которыми невозможны или очень дороги, имитационное моделирование — единственно приемлемое средство анализа и синтеза, поскольку математические модели оказываются громоздкими и необозримыми. К таким системам относятся автоматизированные системы контроля и управления (АСКУ) ядерными энергетическими установками (ЯЭУ), в особенности установками с реакторами большой мощности.

Структура и особенности АСКУ, обусловленные сочетанием сложности, многофункциональности и жестких требований к обеспечению ядерной безопасности, рассмотрены в работах [1—3]. Разработка АСКУ связана с необходимостью применения специальных машинных методов анализа и синтеза управления, поскольку при проектировании возрастает роль взаимосвязи алгоритмов как системы автономных, асинхронно взаимодействующих процессов. В полном объеме начинают проявляться характерные особенности и трудности проектирования программного обеспечения, предназначенного для работы в реальном масштабе времени [4], особенности синтеза мультипроцессорных и многомашинных управляющих вычислительных систем (УВС) агрегатного типа [5].

В работах некоторых авторов, например [6], показано, что для исследования таких систем наиболее удобны именно методы имитационного моделирования. Идеология имитационного моделирования, восходящая к теории агрегативных систем [7], как раз и позволяет выделить в таком многообразии алгоритмов общие черты, получить машинно реализуемое представление о процессах управления, удобное для исследования.

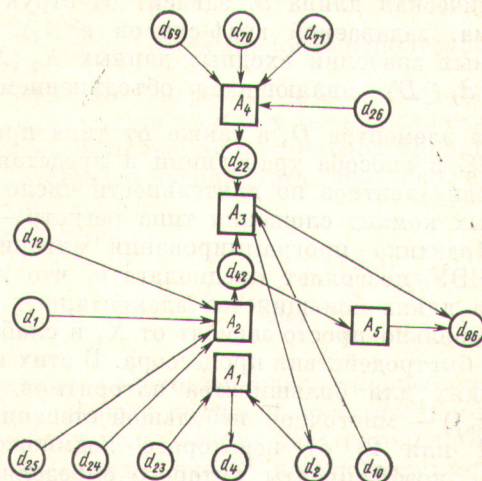
Рассмотрим основные черты имитационной модели АСКУ ЯЭУ и ее особенности. Совокупность алгоритмов разбивается на конечное множество элементарных процессов, для каждого определены входные и выходные массивы информации, закон преобразования входных массивов в выходные, а также регламент выполнения, затраты времени

и памяти, потребности в устройствах. С момента начала процесса и до его окончания считается, что обмена информацией с другими процессами не происходит, а входные и выходные массивы читаются и записываются мгновенно в начале и в конце процесса.

Самое общее и наглядное представление о системе алгоритмов в этом случае дает ориентированный граф (рис. 1), где квадраты обозначают элементарные процессы, кружки — входные и выходные массивы информации, стрелки указывают направление чтения и записи массивов по отношению к элементарным процессам. Структурные принципы имитационной модели схематически показаны на рис. 2.

Обозначим множество элементарных процессов $A = \{A_i\}$, $i = 1, 2, \dots, \text{card}(A)$, а множество массивов информации $D = \{d_i\}$, $i = 1, 2, \dots, \dots, \text{card}(D)$. Здесь и далее $\text{card}(X)$ — это число элементов некоторого конечного множества X . Без ограничения общности будем считать, что каждому алгоритму соответствует единственный и отдельный элементарный процесс.

Для элементов A_i , $i = 1, 2, \dots, \text{card}(A)$, введем такие параметры: R_i — регламент выполнения;



Р и с. 1. Представление алгоритмов в имитационной модели

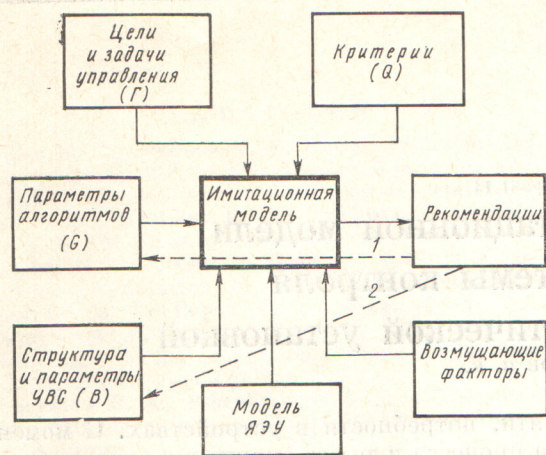


Рис. 2. Принципы построения имитационной модели АСКУ ЯЭУ

a_i — динамическая длина; S_i — статическая длина; Π_i — «изображающая программа»; P_i — приоритет.

Регламент R_i представляет собой закон генерации заявок на выполнение алгоритма A_i и задается на основе данных исследования динамики ЯЭУ и поставленных целей управления. Для АСКУ ЯЭУ характерны следующие регламенты алгоритмов [2]: циклические с периодами, определяемыми динамикой объекта или внешними требованиями; «врезающийся», при котором заявка обуславливается событиями на объекте или в УВС.

Циклические регламенты определяются периодом повторения алгоритмов T_i , возможно, зависящим от времени; для описания «врезающихся» регламентов необходимо ввести множество событий $E = \{E_i\}$, $i = 1, 2, \dots, \text{card}(E)$, и для некоторых $A_j \in A$ подмножество начальных E_j^H и конечных событий E_j^K .

Динамическая длина a_i зависит от структуры алгоритма, задаваемой граф-схемой $g(A_i)$, и от конкретных значений входных данных X_i ($X_i = \bigcup_j d_j$, $d_j \in D$), являющихся объединением нескольких элементов D , а также от типа процессора УВС и способа трансляции и представляет собой эквивалентное по длительности число элементарных команд сложения типа регистр — память. Практика программирования алгоритмов АСКУ ЯЭУ позволяет предполагать, что динамическая длина большинства элементарных процессов довольно просто зависит от X_i и слабо зависит от быстродействия процессора. В этих предположениях для большинства алгоритмов $a_i = M_n(\{x_k\})$ — многочлен небольшой степени (порядка 1 или 2) от некоторого подмножества $\{x_k\} \in X_i$, коэффициенты которого определяются средствами графоанализа. Заметим, что имитационная модель позволяет учитывать не только

нелинейные зависимости a_i от перечисленных выше параметров, но и зависимости, вообще не описываемые никакими аналитическими формулами.

Статическая длина S_i для программных алгоритмов представляет собой число ячеек оперативной памяти, занимаемых программой (вместе с ячейками для множества внутренних данных), а для схемных алгоритмов полагается равной нулю.

Изображающая программа является процедурой (выполнение которой практически происходит мгновенно), преобразующей входные массивы выходных по алгоритмам с заданной степенью соответствующим реальным алгоритмам из A . Кроме параметров из D , Π_i может преобразовывать любые из параметров, описывающих алгоритмы как объект моделирования. В частности, Π_i может вычислять a_i по любому заданному закону. Выполнение Π_i осуществляется в начале и в конце элементарного процесса.

Приоритет P_i является слегка расширенным аналогом приоритета в реальных УВС и может быть абсолютным, относительным или нейтральным. Для некоторых исследований целесообразно также рассматривать беспriorитетные системы обслуживания заявок либо задавать матрицу приоритетов в самом общем виде.

Элементы d_i множества D имеют следующие параметры: l_i и π_i — длина и тип массива. Тип массива отражает направление чтения или записи по отношению к элементам АСКУ, устройств УВС и т. д., а также определяет размерность (машинные слова, байты, разряды и т. д.). Физически π_i чаще всего однозначно определяет устройствами УВС — источниками или приемниками информации; l_i выражается числом элементов размерности, соответствующей π_i . Граф, описывающий систему алгоритмов (G на рис. 2), может быть задан указанием для каждого $A_i \in A$ списков входных данных из $D = Z_i^{\text{вх}}$ и выходных данных из $D = Z_i^{\text{вых}}$.

Материальная часть УВС описывается графом (см. рис. 2) на множестве устройств $U = \{u_i\}$, $i = 1, 2, \dots, \text{card}(U)$. При этом возможны два способа описания УВС: когда структура известна полностью или известна частично. Во втором случае считается, что известны параметры серий выпускаемых устройств вычислительной техники и структурные требования к типовым вычислительным комплексам. Для первого случая B соответствует с той или иной степенью детализации электрической схеме УВС, а элементы u_i множества U имеют такие параметры: u_i — производительность; $c_{i,j}$, $j = 0, 1, 2, \dots, \text{card}(U)$ — емкостные параметры; $e_{i,j}$, $i = 1, 2, 3$ — параметры надежности стоимости и «предпочтения». Во втором случае добавляется еще один параметр, значение которого на первом этапе исследования неизвестно (n_i — количество устройств типа i).

Структурные требования к вычислительным комплексам приводят к необходимости учета так называемых «условий насыщения», задаваемых неравенствами вида

$$\sum_j \alpha_j c_{ij} j n_j \geq \beta_i, \quad (1)$$

где $\alpha_j = 1$ или 0 ; β_i — целое положительное число.

Производительность устройств УВС задается числом элементарных действий устройств в секунду. Например, для процессора такой единицей служит число операций сложения с фиксированной точкой типа регистр — память в секунду. Единицы производительности и динамической длины должны быть согласованы. Когда быстродействие постоянно во времени,

$$\tau_{ij} = a_i / v_j \quad (2)$$

дает продолжительность выполнения алгоритма A_i на процессоре u_j .

Для емкостей устройств введено матричное представление, поскольку это дает удобное единообразие. Например, характеристикой устройства оперативной памяти является емкость как число ячеек памяти, а также емкость как число подключаемых процессоров и устройств прямого доступа.

Параметры e_{ij} дают возможность получать и исследовать различные оптимизационные критерии и задаются либо путем анализа технических данных, либо с помощью экспертных оценок.

Для задания связи между целями и задачами управления на множестве целей $Y = \{y_i\}$, $i = 1, 2, \dots, \text{card}(Y)$, строится граф Γ , вершинами которого являются цели, подцели и логические связки типа И, ИЛИ, а дуги указывают взаимобусловленность целей, причем первичными являются подцели, заключающиеся в выполнении алгоритмов из A в соответствии с их регламентом. Граф Γ может иметь различный вид в зависимости от технологических режимов ЯЭУ и состояния готовности или исправности устройств АСКУ и материальной части УВС. Вершины графа Γ размечены параметрами достижимости δ_i , $i = 1, 2, \dots, \text{card}(Y)$, которые равны единице, если цель достижима, и нулю в противном случае.

Специальное множество элементарных процессов задает простую модель поведения ЯЭУ, существенную при использовании динамики АСКУ, и возмущающие факторы (сбои и отказы оборудования УВС и АСКУ, случайные ошибки программирования и генерации математического обеспечения, ошибки оператора и т. д.).

В модели используются разнообразные критерии качества функционирования АСКУ (множество Q), среди которых находятся и общеизвестные критерии вычислительных систем реального времени — загрузка УВС и отдельных устройств, надежность, штрафные критерии теории очередей.

При проектировании АСКУ ЯЭУ на различных этапах возникают характерные задачи алгоритмического синтеза, синтеза УВС, синтеза дисциплин обслуживания заявок и т. д. Между этими задачами, к сожалению, отсутствует естественная иерархия, и в процессе проектирования возникают различные обратные связи. Одним из преимуществ имитационного моделирования является машинное представление всех или наиболее существенных «деталей» АСКУ и УВС, позволяющее легко менять варианты.

Когда известна структура УВС, необходимо решать задачи рационального распределения алгоритмов по процессорам УВС. При этом УВС современных АСКУ является существенно многомашинной и мультипроцессорной системой. Проектировщику приходится учитывать противоречивые требования равномерной загрузки устройств при заданных резервах производительности и емкости, а также необходимость обеспечения надежности выполнения функциональных групп алгоритмов. При этом простое назначение подмножества алгоритмов на различные процессоры по какому-либо одному критерию, например надежности, может привести к неоправданно большому потокам информации между машинами, к недогрузке одних и чрезмерной перегрузке других устройств УВС. Необходим комплексный подход с учетом всего множества алгоритмов и нескольких критериев оптимизации, автоматизации сложных вычислений ряда параметров; некоторые параметры в принципе невозможно выразить в виде аналитических формул. После того как некоторое приемлемое решение относительно распределения алгоритмов по процессорам найдено, необходимо решить задачу синтеза дисциплины диспетчеризации алгоритмов. Эта задача сводится к назначению алгоритмам приоритетов.

Пусть t_i^W — общее время наработки алгоритма A_i за время моделирования $t_{\max}$. Значение $\frac{1}{t_{\max}} \sum_{i=1}^{\text{card}(A)} t_i^W$ является аналогом используемой в теории очередей «загрузки» УВС. Подчеркнем, что в условиях произвольных (непериодических) регламентов выполнения алгоритмов никаким способом, кроме имитации, это значение вычислить невозможно.

Выражение

$$n_{\text{пр}} = \left\lceil \alpha_{\text{пр}} \frac{1}{t_{\max}} \sum_{i=1}^{\text{card}(A)} t_i^W \right\rceil \quad (3)$$

определяет необходимое число процессоров. Здесь $\alpha_{\text{пр}}$ — заданный резерв производительности; $\lceil x \rceil$ — ближайшее целое сверху действительного числа x . Имеется в виду, что t_i^W получено суммированием значений $\tau_i = a_i / v_{\text{пр}}$, где $v_{\text{пр}}$ — быстродействие процессора.

Пусть $f_{\pi}^{\text{зн}}$ и $f_{\pi}^{\text{чт}}$ — средние, но, вообще говоря, нестационарные потоки записи и чтения информации по устройствам типа π . Требования к производительности устройств могут быть учтены выражениями

$$n_{\pi} = \max \left\{ \left\lceil \frac{f_{\pi}^{\text{зн}}}{v_{\pi}} \right\rceil, \left\lceil \frac{f_{\pi}^{\text{чт}}}{v_{\pi}} \right\rceil \right\}, \quad (4)$$

где n_{π} и v_{π} — число и производительность устройств типа π .

Кроме выражений (3) и (4), основанных на критериях производительности, необходимо также учитывать емкостные требования. Например, число устройств оперативной памяти определяется выражением

$$n_{\text{у.оп}} = \left\lceil \alpha_c \frac{S_{\text{max}}}{C_{0, \text{у.оп}}} \right\rceil, \quad (5)$$

где S_{max} — максимально необходимый объем оперативной памяти; α_c — заданный коэффициент запаса; $C_{0, \text{у.оп.}}$ — емкость оперативной памяти.

Особенностью проектирования АСКУ является следующее обстоятельство. При решении перечисленных задач разбиения алгоритмов по процессорам и синтеза дисциплины диспетчеризации может оказаться, что условия (1) не удовлетворяются ни при каких дисциплинах обслуживания заявок и для части алгоритмов нарушается условие соответствия регламенту. Тем не менее для разработчика АСКУ это еще не означает, что отсутствует решение задачи синтеза. Некоторые алгоритмы, в особенности алгоритмы циклического действия, период которых определяется внешними требованиями (алгоритмы расчета технико-экономических показателей), могут допустить пересмотр их регламента в сторону уменьшения загрузки. Это наиболее очевидно для периодических алгоритмов, загрузка УВС которыми выражается отношением τ_i/T_i , где период входит в знаменатель. Возникающая при этом обратная связь изображена на рис. 2 (пунктир 1).

Современный уровень вычислительной техники, пригодной для использования в АСКУ ЯЭУ, позволяет вообще отказаться от традиционного

ниями управления. Сильно выраженная модульность, агрегатность строения как материальной части УВС, так и ее общего и специального математического обеспечения указывают на необходимость постановки и решения задачи синтеза УВС на стадии проектирования алгоритмов управления. При этом реальные возможности вычислительной техники накладывают на набор алгоритмов специфическую обратную связь (см. рис. 2, пункт 2).

Имитационная модель реализована в виде комплекса программ на языке АЛГОЛ-60 (транслятор Академии наук ГДР) для ЭВМ БЭСМ-6, в который входят: имитатор дискретных асинхронно-взаимодействующих процессов, моделирующий динамику поведения АСКУ; программы графоанализа алгоритмов, написанных на языках МНЕМОКОД и ФОРТРАН II для СМ.ЭВМ, и экспресс-анализа исходных характеристик множества алгоритмов; программа поиска адекватной структуры УВС по критериям производительности и емкости; средства накопления данных, получаемых в имитационных экспериментах, организации архива на магнитной ленте и диалога с пользователем; программа вычисления характеристик качества АСКУ.

В заключение приведем краткое описание расчета на имитационной модели, в котором с помощью этой модели решается задача синтеза УВС, когда известны характеристики алгоритмов и массивов данных и тип устройств УВС. Требуется определить состав устройств, распределение задач по вычислительным комплексам и вычислить характеристики прохождения задач в системе. Оператор ведет диалог с БЭСМ-6 через устройство Видеотон-340. Протокол диалога печатается на АЦПУ.

По запросу ЭВМ вводятся исходные данные из заготовленного ранее на диске файла. При этом оператор может придать любым алгоритмам статус диск-резидентных или ОЗУ-резидентных задач. Оператор может получить по требованию таблицы алгоритмов, данных или устройств. В данном случае таблица алгоритмов выглядит так:

ШИФР	ИМЯ	ПРИОРИТЕТ	ПЕРИОД	АЛГОРИТМЫ			
				ДЛИТЕЛЬНОСТЬ	ОБЪЕМ	ВХОДЫ	ВЫХОДЫ
111	КНФР2П	1	120.0	0.091	235	1, 2, 3, 4, 5	2, 3, 5
...	...	...	...	...	...	...	...
117	РМТК12	7	900.0	850.0	600	8, 7, 9, 18, 19, 34, 35, 28, 29	36, 37, 38, 39

подхода, когда сначала на объекте устанавливаются вычислительные средства, приблизительно удовлетворяющие информационным нуждам оператора, а затем делается попытка улучшить и расширить систему в соответствии с растущими требова-

Оператор может заменить, исправить, дополнить любые сведения об алгоритмах, данных и устройствах. Алгоритмы могут выполняться с самым разнообразным регламентом. Для задания сложных регламентов и любых других осо-

бенностей алгоритмов служит аппарат изображающих программ. Привязка изображающих программ к конкретным алгоритмам проводится оператором в режиме диалога.

В ответ на запросы модели оператор вводит некоторую дополнительную информацию, например количество разделов оперативной памяти в вычислительном комплексе, максимально допустимую загрузку комплекса, максимально допустимый объем занятой оперативной памяти в разделе.

Модель выдает сообщение:

СТРУКТУРА УВС ЗАДАНА?

После ответа оператора НЕТ модель рассчитывает вариант структуры УВС и предлагает его оператору в виде сообщений следующего формата:

КОЛИЧЕСТВО ВК: 3

ВК № 1

РАЗДЕЛ № 1

АЛГОРИТМЫ

111 112 113

ДАННЫЕ:

2 3 5 8 16 24 25

ЗАГРУЗКА: 0.00

ПАМЯТЬ: 20685

РАЗДЕЛ № 2

АЛГОРИТМЫ:

114 115 116

ДАННЫЕ:

9 17 20 32 33

ЗАГРУЗКА: 0.05

ПАМЯТЬ: 16028

Затем печатается информация о требуемых ресурсах устройства связи с объектом относительно каждого вычислительного комплекса (ВК).

Оператор может изменить предлагаемую структуру, т. е. перераспределить и данные по ВК, добавить новые ВК или уменьшить их число. Далее оператор в ответ на запросы модели устанавливает режимы имитационного эксперимента (время моделирования, коэффициенты вариации, время выполнения алгоритмов, режимы трассировки процессов и т. д.). После этого происходит имитация выполнения алгоритмов. По окончании имитационного эксперимента выдается:

а) информация о потоках данных:

ВК № 1:

ТИП

ДАННЫХ

МАКСИМАЛЬ-

НЫЙ ПОТОК

НА ВХОДЕ В

АЛГОРИТМЫ

МАКСИМАЛЬ-

НЫЙ ПОТОК

НА ВЫХОДЕ

ИЗ АЛГОРИТ-

МОВ

СРЕДНИЙ ПО-

ТОК НА ВХОДЕ

В АЛГОРИТМЫ

СРЕДНИЙ ПО-

ТОК НА ВЫХО-

ДЕ ИЗ АЛГОРИТ

МОВ

1

424.07

0.00

3.88

0.00

2

24768.71

10598.98

280.39

116.42

...

б) информация о характеристиках потока заявок:

АЛГОРИТМ

СРЕДНЕЕ

ВРЕМЯ

ОЖИДА-

НИЯ

СРЕДНЕЕ

ВРЕМЯ

ВЫПОЛНЕ-

НИЯ

МАКСИМАЛЬ-

НОЕ ВРЕМЯ

ОЖИДАНИЯ

МАКСИМАЛЬ-

НОЕ ВРЕМЯ

ВЫПОЛНЕ-

НИЯ

ОТНОШЕНИЕ

ДЛИТЕЛЬНО-

СТИ ОБСЛУ-

ЖИВАНИЯ К

ПЕРИОДУ

111

0.0

0.095

0.0

0.1

0.00083

112

0.0

0.133

0.0

0.14

0.00116

...

117

13.217

888.934

69.901

934.793

1.11643

...

в) информация об общих затраченных ресурсах производительности, памяти, простых процессах.

Структура УВС может быть по желанию оператора пересчитана по результатам имитации. После этого оператор может вернуться к началу работы с моделью и просчитать все заново. Для 22 алгоритмов и 54 различных массивов данных время имитации на один вариант (модельное время 20000.0) составляет 35 с. Объем памяти комплекса программ составляет 31364₈ слов БЭСМ-6 (без учета вспомогательных программ графоанализа алгоритмов).

СПИСОК ЛИТЕРАТУРЫ

1. Емельянов И. Я., Гаврилов П. А. — В кн.: Опыт эксплуатации АЭС и пути дальнейшего развития атомной энергетики. Т. II. Обнинск, изд. ФЭИ, 1974, с. 288.
2. Емельянов И. Я., Гаврилов П. А., Селиверстов Б. Н. Управление и безопасность ядерных энергетических реакторов. М., Атомиздат, 1975.
3. Čigarev V., Trechov V. E. — Kernenergie, 1975, Bd 18, H. 10, S. 311 bis 313.
4. Никитин А. И. — В кн.: I Всесоюз. симп. по математическому обеспечению вычислительных систем, работающих в реальном масштабе времени (тезисы докладов). Киев, изд. Ин-та кибернетики АН УССР, 1972, с. 3.
5. Липаев В. В., Колин К. К., Серебровский Л. А. Математическое обеспечение управляющих ЦВМ. М., Советское радио, 1972.
6. Parish C. — Comput. J., 1975, v. 18, N 2, p. 117.
7. Бусленко Н. П. Моделирование сложных систем. М., Наука, 1968.

Поступила в Редакцию 24.03.80