

ВЕРИФИКАЦИЯ ФУНКЦИОНАЛЬНЫХ АЛГОРИТМОВ СИСТЕМЫ УПРАВЛЕНИЯ МНОГОАГРЕГАТНОЙ СУДОВОЙ ЭЛЕКТРИЧЕСКОЙ СТАНЦИЕЙ

На современных специализированных и пассажирских судах с целью обеспечения надежности электрического снабжения и по критериям экономичности используются многоагрегатные электрические станции. Так, например, на пассажирских судах с электродвижением электрическая станция состоит не менее, чем из шести генераторных агрегатов (ГА). Задачи управления многоагрегатными электрическими станциями имеют актуальность [1], а надежность работы систем управления станциями определяется надежностью аппаратной части и программного обеспечения [2]. Современные технологии обеспечивают высокую надежность аппаратной части, выпускаемой в виде контроллеров. Кроме того, аппаратная часть является "видимой" и доступной, как для разработчиков, так и для эксплуатационников, в то время как программная остается "невидимой" и, в большинстве случаев, недоступной. В этой связи разработку программного обеспечения для каждого отдельного объекта, ввиду отсутствия программного аналога, приходится начинать при "нулевых условиях", с поиском эффективных методов верификации задач и синтеза функциональных алгоритмов.

Задача данной работы – разработка способа верификации функциональных алгоритмов системы управления многоагрегатной судовой электрической станцией с целью обеспечения бесперебойного электроснабжения и повышения экономичности ее работы.

На первом этапе верификации использован метод графа-автомата [3], который позволяет достаточно наглядно и компактно отобразить совокупность состояний объекта и переходов между ними. Однако сам по себе граф автомата содержит слабо развитую синтаксическую структуру, что затрудняет проверку семантической корректности (полноты и непротиворечивости) формализуемого задания. Указанный недостаток устраняется применением систем булевых функций (функторов) для описания совокупности признаков состояний объекта и условий перехода.

Граф содержит совокупность пронумерованных вершин $S(N_{STS})$

$$= \bigcup_{i=1}^n S(i), \quad i \in N_{STS}, \quad |N_S| = n, \quad \text{и незапрещенных переходов между ними}$$

из множества $JMP(N_S \times N_S) = \bigcup_{i,j}^{mn}$, $i \in N_{STS}, j \in N_{STS}$.

Простой функтор в общем случае описывается системой:

$$\begin{aligned}
 v_a(t+1) &:= f_a(x_1(t+1), \dots, x_m(t+1), v_1(t), \dots, v_g(t), y_1(t), \dots, y_p(t)); \\
 \dots & \dots \dots \dots \dots \dots \dots \dots \dots \dots \\
 v_i(t+1) &:= f_i(x_1(t+1), \dots, x_m(t+1), v_g(t), y_1(t), \dots, y_p(t)); \\
 y_i(t+1) &:= f_j(x_1(t+1), \dots, x_m(t+1), \dots, v_g(t), y_1(t), \dots, y_p(t)); \\
 \dots & \dots \dots \dots \dots \dots \dots \dots \dots \dots \\
 y_r(t+1) &:= f_r(x_1(t+1), \dots, x_m(t+1), \dots, v_g(t), y_1(t), \dots, y_p(t)),
 \end{aligned} \tag{1}$$

где $(v_a, \dots, v_i; v_1, \dots, v_g) \subseteq V$, $(y_j, \dots, y_r; y_1, \dots, y_p) \subseteq Z$, а функции f_a, \dots, f_r являются булевыми, переменные $x_1, \dots, x_m; v_1, \dots, v_g; y_1, \dots, y_p$ входными, а $v_a, \dots, v_i; y_j, \dots, y_r$ выходными функторными переменными.

Сложный функтор в общем случае описывается совокупностью выражений:

$$\begin{aligned}
 v_a(t+1), \dots, v_i(t+1), y_i(t+1), \dots, y_r(t+1) &:= F_A(x_1(t+1), \dots, x_m(t+1), \\
 v_1(t), \dots, v_g(t), y_i(t), \dots, y_p(t); \\
 \dots & \dots \dots \dots \dots \dots \dots \dots \dots \dots
 \end{aligned} \tag{2}$$

$$\begin{aligned}
 (v_a(t+1), \dots, v_i(t+1), y_i(t+1), \dots, y_r(t+1) &:= F_T(x_1(t+1), \dots, x_m(t+1), \\
 v_1(t), \dots, v_g(t), y_i(t), \dots, y_p(t),
 \end{aligned}$$

а сложный предикат совокупностью:

$$\begin{aligned}
 F_A(x_1(t+1), \dots, x_m(t+1), v_1(t), \dots, v_g(t), y_i(t), \dots, y_p(t)); \\
 \dots & \dots \dots \dots \dots \dots \dots \dots \dots \dots
 \end{aligned} \tag{3}$$

$$F_T(x_1(t+1), \dots, x_m(t+1), v_1(t), \dots, v_g(t), y_i(t), \dots, y_p(t)),$$

в которых каждая F_A, \dots, F_T является либо булевой функцией, либо ей соответствует некоторая аппаратная схема с памятью с входными и выходными переменными $x_{\overline{1,m}}, v_{\overline{1,g}}, y_{\overline{1,p}}; v_a, \dots, v_j; y_{\overline{j,a}}$ (последние в случае предиката отсутствуют).

Выражения, входящие в совокупность (1), (2), (3), позволяют перейти к составлению непротиворечивой системы эталонных признаков $SIGN(P(N_{STS}))$ состояний объекта и переходов $SIGN(P(N_{STS} \times N_{STS}))$ путем отбора нужных и реальных кодовых комбинаций значений переменных X и V с учетом такого требования как минимум длины сло-

ва признака при условиях достаточности и непротиворечивости. Непротиворечивость проверяется попарным сравнением слов признаков, а противоречия устраняются доопределением из имеющегося числа переменных и предикатов, либо, если не удастся, то введением новых, дополнительных переменных.

Задача управления составом ГА связана с такими критериями как экономичность, надежность бесперебойного электроснабжения в нормальных и аварийных ситуациях. При рассмотрении критерия экономичности, так или иначе, возникает задача выбора рациональной очередности включения/отключения ГА для данного режима работы и технического состояния агрегатов.

Будем считать, что в управляющей системе предусмотрен задатчик, с помощью которого можно задать любую последовательность (очередность) $SQ(F) \in SQ(N_{SQ})$, взятую из упорядоченного, пронумерованного множества $SQ(N_{SQ})$ всех возможных последовательностей. Тогда, если в станции установлены m генераторов $|GA| = m$, то множество очередностей составит $|SQ(N_{SQ})| = m!$, т.е. с ростом m число очередностей быстро возрастает. Поэтому в современных станциях не стремятся к получению всей размерности $m!$, поскольку это является избыточным с точки зрения затрат ресурсов производительности вычислительной техники и усложнением управляющей программы. В связи с этим при отборе элементов подмножества $SQ(F) \in SQ(N_{SQ})$ руководствуются тем, чтобы была обеспечена возможность постановки каждого ГА в любую очередь. Для этого вполне достаточно располагать последовательностями объемом $|SQ(F)| = m$:

$$\left| \begin{array}{l} SQ(1) \\ SQ(2) \\ SQ(3) \\ \dots \\ \dots \\ SQ(m-1) \\ SQ(m) \end{array} \right| \cong \left| \begin{array}{cccccccc} 1 & 2 & 3 & \dots & \dots & \dots & & m \\ 2 & 3 & 4 & \dots & \dots & m & & 1 \\ 3 & 4 & 5 & \dots & m & 1 & & 2 \\ \dots & \dots & \dots & \dots & \dots & \dots & & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & & \dots \\ m-1 & m & 1 & 2 & \dots & \dots & & m-2 \\ m & 1 & 2 & \dots & \dots & m-2 & & m-1 \end{array} \right| \quad (4)$$

Синтез алгоритма задания очередностей (4) вызывает меньше трудностей, чем синтез полноразмерного алгоритма, поскольку можно обойтись выбором текущей $SQ(i)$ последовательности среди m последовательностей.

Для перехода к описанию при помощи графа-автомата необходимо определить количество состояний объекта и технологического процесса. Количество состояний станции в первую очередь определяется количеством ГА в ней.

В качестве примера верификации приведем формализацию алгоритма функционирования трехагрегатной электрической станции, принцип которой может быть использован при формализации алгоритмов станции с любым количеством ГА.

Для системы трехагрегатной станции номера состояний имеют следующие значения: 1 – работает под нагрузкой один ГА(k); 2 – к ГА(k) синхронизируется ГА(l); 3 – ГА(k) и ГА(l) работают в параллели; 4 – ГА(k) и ГА(l) распределяют нагрузку; 5 – ГА(l) сбрасывает нагрузку на ГА(k); 6 – к ГА(k) и ГА(l) синхронизируется ГА(m); 7 – ГА(k), ГА(l) и ГА(m) работают в параллели; 8 – ГА(k), ГА(l) и ГА(m) распределяют нагрузку; 9 – ГА(m) сбрасывает нагрузку на ГА(k) и ГА(l); 10 – СЭС обесточена.

Каждая вершина графа $GR(i) \in GR(N_{GR})$, $i \in N_{GR}$ сопоставима с вполне определенным предикативным состоянием $LC(i) \in LC(N_{LC})$, где N_{GR} и N_{LC} – нумерации, взаимно связанные однозначным соответствием $|N_{GR}| = |N_{LC}| = 10$. Например, если в параллели работают два ГА, нагрузка не перераспределяется и не происходит ее сброса, то только один предикат должен быть истинным $LC(3) = И$, а остальные – ложными. Под термином "предикативные состояния" понимается система предикатов, лишенных парадоксов в том смысле, что каждое состояние должно описываться отличным от другого, полным и непротиворечивым набором слов-признаков, а установлению состояния STS должен существовать вполне определенный алгоритм. В таком случае граф состояний трехагрегатной станции примет вид (рис. 1).

Способ описания состояния раскроем на примере описания 7-го состояния и нескольких его переходов:

$$SIGNLC(7) \cong N(k) \& N(l) \& N(m) \& \overline{MSY} \& \overline{MEM} \& \overline{MSP(i)} \& \overline{MCHL} \quad (5)$$

где $SIGNLC(7)$ – признак седьмого состояния; $N(k)$, $N(l)$ и $N(m)$ – к шинам подключены три ГА; \overline{MSY} – синхронизация не происходит; \overline{MEM} – аварийные состояния отсутствуют; $\overline{MSP(i)}$ – команд на останов ГА не поступало; \overline{MCHL} – распределения нагрузок не происходит.

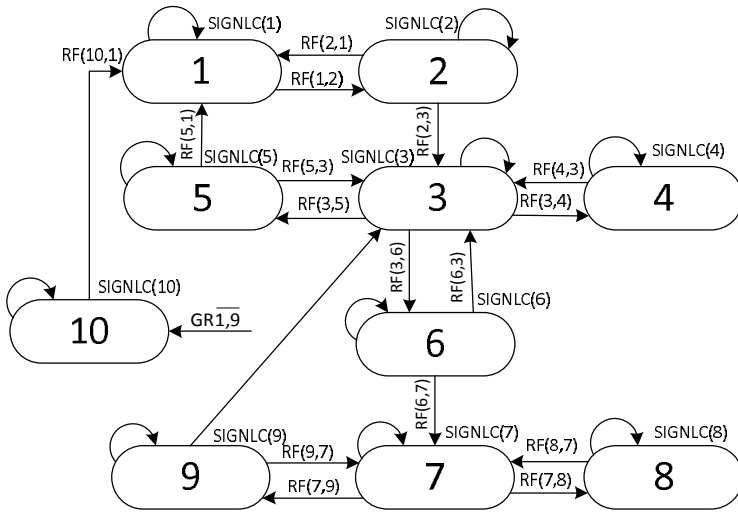


Рис. 1. Граф состояний трехагрегатной станции

Переход из состояния 7 в состояние 9 может быть описан следующим образом:

$$\begin{aligned} \text{RFLC}(7 \rightarrow 9) \cong \text{LC}(7) \& \text{ MUNL}(m) \& \overline{\text{MEM}} \rightarrow (\text{WRLC}(9), \\ \text{WRUNL}(m), \text{WRB}(\text{UNL}), \text{WRTHD}(\text{UNL})), \end{aligned} \quad (6)$$

где RF – результатная функция; приставки M (message) и WR (wright) при командах означают поступившее для исполнения сообщение с командой и запись состояния, команды или данных в память; MUNL – команда разгрузки ГА; WRB – запись в память времени, отведенного для разгрузки; WRTHD – запись в память порогового значения нагрузки ГА, при котором произойдет отключение ГА.

Этот этап верификации дает возможность перейти к более детальной форме верификации на базе алгоритмического языка логических схем алгоритма (ЛСА). С учетом (5), (6), ЛСА модуля SBLC(7) примет вид:

$$\begin{aligned}
S_{\text{H}}\{\text{SBINP.SBCNT.SBLD.SBOVL}\}\text{SIGNLC}(7) \uparrow^1 \\
\uparrow^1 |LD(m) - LD(k)| > \varepsilon \vee |LD(l) - LD(k)| > \varepsilon \vee |LD(m) - LD(l)| > \varepsilon \uparrow^2 \omega \uparrow^3 \\
\downarrow^2 \text{MSP}m(t) \uparrow^4 \omega \uparrow^5 \downarrow^4 \text{MSP}m(t+1) \uparrow^{\kappa} \omega \uparrow^5 \downarrow^3 \text{WRLC}(8).\text{JMSBLSH} \downarrow^5 \\
\text{WRLC}(9).\text{JMUNL}(m) \downarrow^1 \text{WRNOLC.DOP} \downarrow^{\kappa} S_{\kappa}
\end{aligned} \tag{7}$$

где SBINP, SBCNT, SBLD, SBOVL – подпрограммы ввода данных процесса, счета количества подключенных к шинам ГА, измерения нагрузки, выполнения операций разгрузки при перегрузках соответственно; LD(m, k, l) – нагрузка каждого из трех работающих ГА; ε – допустимое отклонение нагрузки у работающих в параллели ГА; JBSLSH – переход на подпрограмму распределения нагрузки генераторов; WRNOLC – запись в память неопределенного состояния; DOP – запрос на принятие решения оператором.

Выводы. В работе предложен способ верификации функциональных алгоритмов системы управления судовой электрической станцией. Верификация состоит из трех этапов: построение графа состояний; описание при помощи переключательных функций состояний системы и условий переходов между ними; синтез логических схем алгоритмов работы системы управления станцией.

Сочетание графа-автомата и ЛСА является функционально полным набором верификации, достаточным для разработки программного обеспечения с учетом спецификации аппаратной части.

Рассмотренные функциональные алгоритмы системы управления являются своеобразным фундаментом, на котором целесообразно строить алгоритмы оптимизации, толерантного к неисправностям и адаптивного управления, а также алгоритмы работы локальных подсистем станций. В этой связи дальнейшие исследования целесообразно посвятить разработке алгоритмов управления электрической станцией по таким критериям, как наработка часов, удельный расход топлива и техническое состояние ГА.

СПИСОК ЛИТЕРАТУРЫ

1. Головки С. В. Моделирование автоматической системы управления судовой электростанции с учетом состояния оборудования // Вестник Астраханского государственного технического университета: научн. журнал. – 2014. – Вып. 3. – С. 58 – 63.

2. Пипченко А. Н. Проблемы оптимизации судовых электростанций / А.Н. Пипченко, А.А. Толстов // Научн. журнал. – Вып. 6. – Л.: Судостроение, 1985. – С. 77 – 84.

3. Пипченко А. Н. О некоторых методологических аспектах построения рациональных структур микропроцессорных систем в судовой энергетике / А.Н. Пипченко, В. В. Пономаренко // Электрооборудование и автоматизация установок и систем: сб. науч. тр. – Николаев: НКИ, 1988. – С. 47 – 58.