

**ВЕРИФИКАЦИЯ ФУНКЦИОНАЛЬНЫХ АЛГОРИТМОВ СИСТЕМЫ
УПРАВЛЕНИЯ СУДОВЫМ РУЛЕВЫМ ПРИВОДОМ***Шевченко В.А.**Одесская национальная морская академия*

В работе верифицированы алгоритмы функционирования системы управления рулевым приводом. Получен граф состояний рулевого привода, описаны состояния рулевого привода и условия перехода между ними.

Разработана логическая схема алгоритма функционирования системы управления рулевым приводом, которая может лечь в основу при разработке программного обеспечения таких систем.

Ключевые слова: судно, рулевой привод, алгоритм, граф, результатная функция.

Введение. Одним из определяющих факторов безопасного движения судна является надежная работа судового рулевого комплекса. Надежность рулевого комплекса зависит не только от безотказной работы его силовых узлов, но и устойчивой работы его системы управления, алгоритмическое обеспечение которой позволит в полной мере реализовывать задачи оператора или авторулевого, а также вовремя диагностировать, а в некоторых случаях и обеспечивать толерантное к неисправностям [1] управление.

Актуальность исследований. Надежность судового рулевого комплекса определяется надежностью рулевого привода и ее управляющего устройства – авторулевого или автопилота. В свою очередь, надежность авторулевого определяется надежностью аппаратной части и программного обеспечения. Современные технологии обеспечивают высокую надежность аппаратной части, выпускаемой в виде контроллеров, предназначенных для работы в различных условиях окружающей среды и устойчивых к влиянию вредных технологических процессов.

В то же время программное обеспечение современных систем управления рулевым приводом не раскрывается производителями и, поэтому проанализировать его в большинстве случаев не представляется возможным. Кроме того, базовые алгоритмы такого программного обеспечения не носят унифицированный характер, и зависят от предпочтений проектирующей и исполняющей организаций.

Поэтому верификация алгоритмов функционирования как на стадии разработки с целью их унификации, так и в процессе эксплуатации и наладки является важной и актуальной.

Постановка задачи. Задачей данной работы является верификация функциональных алгоритмов системы управления рулевым приводом с целью обеспечения надежной его работы.

Результаты исследований. В работе [2] приведены концепция и теоретические основы описания состояний объекта и условий переходов между ними. В основу такой концепции заложена теория графов.

Первый этап верификации выполним при помощи графа, рис.1, который позволяет достаточно наглядно и компактно отобразить совокупности состояний рулевого привода $LS(N_{LS})$: $LS(N_{LS}) = ULS(i), i \in N_{LS}$,

где $N_{LS} = \overline{0,10}$ – нумерация состояний локальных процессов рулевого привода (РП).

Для системы РП номера состояний имеют следующие значения: 0 – агрегат стоит в режиме «местный»; 1 – стоит в режиме «дистанционный»; 2 – стоит в режиме «автомат»; 3 – работает в режиме «автомат»; 4 – стоит в режиме «следающий»; 5 – работает в режиме «следающий»; 6 – привод работает и перо руля находится в диаметральной плоскости; 7 – перекидка влево; 8 – перекидка вправо; 9 – руль в расчетной позиции; 10 – остановлен нормально; 11 – заблокирован по аварии.

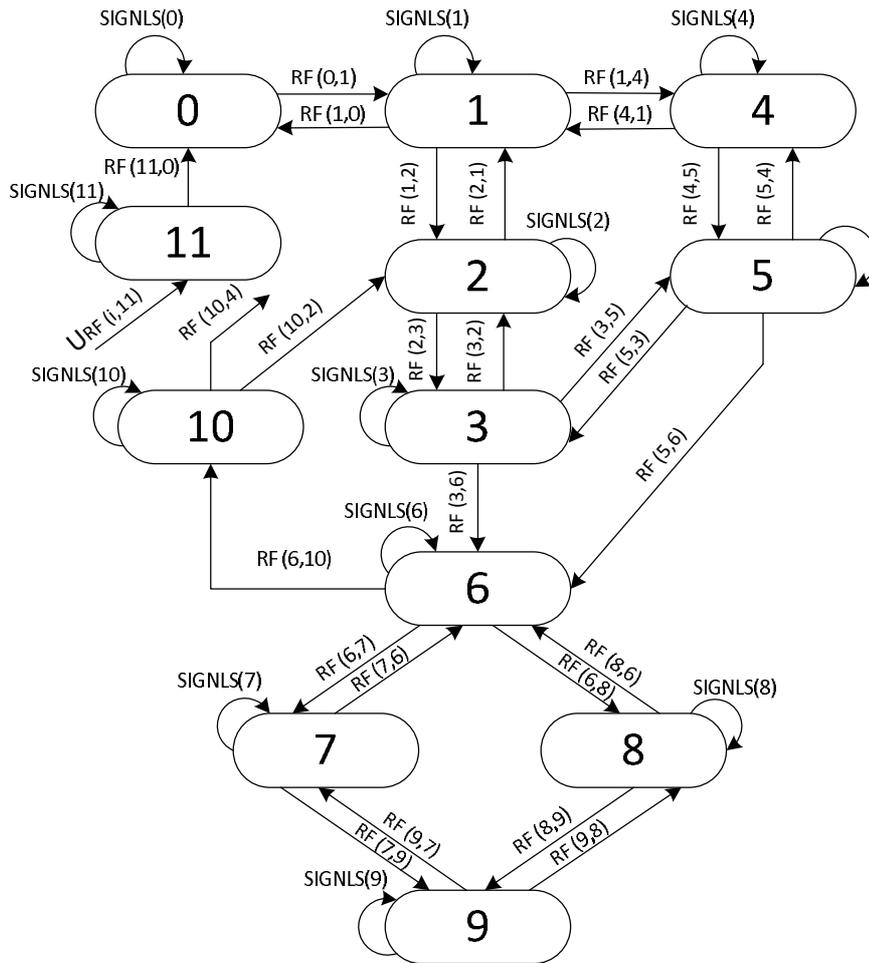


Рисунок 1 – Граф состояний системы рулевого привода

Каждая вершина графа $GR(i) \in GR(N_{GR})$, $i \in N_{GR}$ сопоставима с вполне определенным предикативным состоянием $LS(i) \in LS(N_{LS})$, где N_{GR} и N_{LS} – нумерации, взаимно связанные однозначным соответствием $|N_{GR}| = |N_{LS}| = 12$. Например, если рулевой привод работает, и перо руля находится в диаметральной плоскости, то только один предикат должен быть истинным $LS(6) = И$, а остальные – ложными. Под термином «предикативные состояния» будем понимать систему 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.

Для описания слов-признаков состояний РП будем использовать следующие сигналы и команды: **LCL** – сигнал, принимающий истинное значение, если переключатель выбора вида управления на щите управления рулевым приводом установлен в положение «местный», а не «дистанционный»; **RMT** – сигнал, принимающий истинное значение, если переключатель выбора вида управления на щите управления рулевым приводом установлен в положение «дистанционный»; **AUT** – сигнал, принимающий истинное значение, если переключатель выбора вида управления на пульте авторулевого установлен в положение «автоматический», а не «следящий»; **FU** – сигнал, принимающий истинное значение, если переключатель выбора вида управления на пульте авторулевого установлен в положение «следящий»; **MST** – команда, принимающая истинное значение, если была нажата кнопка пуска насоса на местном либо дистанционном пульте, а также в случае поступления команды на пуск от координатора; **MSP** – команда, принимающая

истинное значение, если была нажата кнопка остановки насоса на местном либо дистанционном пульте, а также в случае поступления команды на останов от координатора; **PWRK** – сигнал истинный, если насос в работе; **MEM** – сигнал истинный, если насос остановлен по аварии; **RPMD** – сигнал, принимающий истинное значение, если перо руля находится в диаметральной плоскости; **RMVP** – сигнал и команда являются истинными, если руль переключается влево либо поступила команда на переключку руля влево; **RMVS** – сигнал и команда являются истинными, если руль переключается вправо либо поступила команда на переключку руля вправо; **BRMV** – сигнал является истинным, когда расчетное время переключки истекло; **RDP** – сигнал, принимающий истинное значение, если перо руля находится в расчетной позиции; **RPP** – сигнал, принимающий истинное значение, если перо руля находится слева от диаметральной плоскости; **RPS** – сигнал, принимающий истинное значение, если перо руля находится справа от диаметральной плоскости; **RMVSP** – сигнал и команда принимают истинное значение, если переключка руля остановлена либо поступила команда на остановку переключки пера руля.

С учетом принятых обозначений, состояния агрегата описываются следующими словами-признаками:

$$\begin{aligned} \text{SIGNLS}(0) &= LCL \& \overline{PWRK} ; \text{SIGNLS}(1) = RMT \& \overline{PWRK} \& \overline{MST} ; \\ \text{SIGNLS}(2) &= RMT \& \overline{PWRK} \& \overline{MST} \& AUT ; \text{SIGNLS}(3) = RMT \& PWRK \& AUT ; \\ \text{SIGNLS}(4) &= RMT \& \overline{PWRK} \& \overline{MST} \& FU ; \text{SIGNLS}(5) = RMT \& PWRK \& FU ; \\ \text{SIGNLS}(6) &= PWRK \& (AUT \vee FU) \& RPMD ; \\ \text{SIGNLS}(7) &= PWRK \& (AUT \vee FU) \& RMVP \& \overline{BRMV} ; \\ \text{SIGNLS}(8) &= PWRK \& (AUT \vee FU) \& RMVS \& \overline{BRMV} ; \\ \text{SIGNLS}(9) &= PWRK \& (AUT \vee FU) \& RDP \& BRMV ; \\ \text{SIGNLS}(10) &= \overline{PWRK} \& MSP ; \text{SIGNLS}(11) = \overline{PWRK} \& MEM . \end{aligned}$$

Для окончательной формализации программы функционирования локальной подсистемы рулевого привода необходимо получить систему результатных функций, отображающих условие и процедуру перехода от одного состояния к другому. Детализируя исследуемый процесс и используя язык результатных функций, условия переходов представим в виде следующей системы:

$$\begin{aligned} RFLS(0 \rightarrow 1) &\cong LS(0) \& RMT \rightarrow WR : MRMT, LS(1) \\ RFLS(1 \rightarrow 0) &\cong LS(1) \& LCL \rightarrow WR : MLCL, LS(0); CL : MRMT \\ RFLS(1 \rightarrow 2) &\cong LS(1) \& AUT \rightarrow WR : MAUT, LS(2) \\ RFLS(2 \rightarrow 1) &\cong LS(2) \& RMT \& \overline{AUT} \rightarrow WR : MRMT, LS(1); CL : MAUT \\ RFLS(2 \rightarrow 3) &\cong LS(2) \& MST \& \overline{MEM} \rightarrow WR : PWRK, LS(3) \\ RFLS(3 \rightarrow 2) &\cong LS(3) \& MSP \rightarrow WR : LS(2); CL : PWRK \\ RFLS(1 \rightarrow 4) &\cong LS(1) \& FU \rightarrow WR : MFU, LS(4) \\ RFLS(4 \rightarrow 1) &\cong LS(4) \& \overline{FU} \& RMT \rightarrow WR : MRMT, LS(1); CL : MFU \\ RFLS(4 \rightarrow 5) &\cong LS(4) \& MST \& \overline{MEM} \rightarrow WR : PWRK, LS(5) \\ RFLS(5 \rightarrow 4) &\cong LS(5) \& MSP \rightarrow WR : LS(4); CL : PWRK \end{aligned}$$

$$RFLS(3 \rightarrow 5) \cong LS(3) \& \overline{AUT} \& FU \rightarrow WR : MFU, LS(5); CL : MAUT$$

$$RFLS(5 \rightarrow 3) \cong LS(5) \& \overline{FU} \& AUT \rightarrow WR : MAUT, LS(3); CL : MFU$$

$$RFLS(3 \rightarrow 6) \cong LS(3) \& (RPP \& MRMVS \vee RPS \& MRMVP \vee RPMD) \rightarrow \\ \rightarrow WR : RPMD, LS(6), RMVSP(t+1) \text{ if } RPMD(t+1) \& (RMVS(t) \vee RMVP(t))$$

$$RFLS(5 \rightarrow 6) \cong LS(5) \& (RPP \& MRMVS \vee RPS \& MRMVP \vee RPMD) \rightarrow \\ \rightarrow WR : RPMD, LS(6), RMVSP(t+1) \text{ if } RPMD(t+1) \& (RMVS(t) \vee RMVP(t))$$

$$RFLS(6 \rightarrow 7) \cong LS(6) \& MRMVP \& \overline{MRMVSP} \& \overline{MRMVS} \rightarrow WR : RMVP, LS(7), B(RMV); CL : MRPMD$$

$$RFLS(7 \rightarrow 6) \cong LS(7) \& MRMVSP \& RPMD \rightarrow WR : MRPMD, LS(6)$$

$$RFLS(6 \rightarrow 8) \cong LS(6) \& MRMVS \& \overline{MRMVSP} \& \overline{MRMVP} \rightarrow WR : MRMVS, LS(8), B(RMV); CL : MRPMD$$

$$RFLS(8 \rightarrow 6) \cong LS(8) \& MRMVSP \& RPMD \rightarrow WR : MRPMD, LS(6)$$

$$RFLS(7 \rightarrow 9) \cong LS(7) \& RDP \& BRMV \rightarrow WR : MRDP, LS(9), MRMVSP; CL : MRMVP$$

$$RFLS(9 \rightarrow 7) \cong LS(9) \& MRMVP \& \overline{MRMVSP} \& \overline{MRMVS} \rightarrow WR : MRMVP, LS(7), B(RMV); CL : MRDP$$

$$RFLS(8 \rightarrow 9) \cong LS(8) \& RDP \& BRMV \rightarrow WR : MRDP, LS(9), MRMVSP; CL : MRMVS$$

$$RFLS(9 \rightarrow 8) \cong LS(9) \& MRMVS \& \overline{MRMVSP} \& \overline{MRMVP} \rightarrow WR : MRMVS, LS(8), B(RMV); CL : MRDP$$

$$RFLS(6 \rightarrow 10) \cong LS(6) \& MSP \rightarrow WR : \overline{PWRK}, LS(10)$$

$$RFLS(10 \rightarrow 2) \cong LS(10) \& AUT \rightarrow WR : MAUT, LS(2)$$

$$RFLS(10 \rightarrow 4) \cong LS(10) \& FU \rightarrow WR : MFU, LS(4)$$

$$RFLS(11 \rightarrow 0) \cong LS(11) \& RST \& LCL \& \overline{MEM} \rightarrow WR : MLCL, LS(0); CL : MEM$$

Для доказательства правомерности такого подхода к описанию программы, пользуясь графом, рис. 1, и приведенной выше системой результатных функций, рассмотрим работу системы РП по переводу РП из некоторых общих состояний в конечное (руль находится в расчетной позиции).

Допустим, РП находится в состоянии STS(3), которое характеризуется словом-признаком $RMT \& PWRK \& AUT$, т.е. насос рулевой машины находится в работе и при этом выбраны режим работы «дистанционный» и вид управления «автоматический». Как видно из графа, существует только один путь перехода из STS(3) в STS(6), для прохождения которого формируются команды по перемещению пера в диаметральной плоскости, при этом, если руль находится слева от диаметральной плоскости RPP , то формируется команда на перекидку пера руля вправо $MRMVS$ пока руль не достигнет диаметральной плоскости $RPMD$, как только это произойдет, перекидка руля прекратится $RMVSP(t+1)$ и произойдет переход в состояние STS(6). Аналогичные действия произойдут, в случае, если перо руля находится справа от диаметральной плоскости. Если же РП находится в состоянии STS(3) и перо руля уже находится в диаметральной плоскости $RPMD$, то произойдет переход в состояние STS(6).

Из состояния STS(6) при поступлении команды на перекидку руля влево $MRMVP$ и отсутствии команд на перекидку вправо $MRMVS$ и прекращение перекидки $MRMVSP$ РП переходит в состояние STS(7). При этом в память заносится значение расчетного времени, отведенного на перекидку $B(RMV)$ и начинается его отсчет.

Из состояния STS(7) РП перейдет в состояние STS(9) как только перо руля достигнет расчетной позиции *RDP*, и время на переключку истечет *BRMV*.

Полученную таким образом систему результатных функций и слов-признаков состояний положим в основу алгоритма работы локальной подсистемы рулевого привода, который можно выразить следующей логической схемой алгоритма (ЛСА):

$$\begin{aligned}
 & S_{и}.INPUT.MLS(0,t-1) \overset{26}{\uparrow} \downarrow \overset{1}{SIGN} : LS(0).LS(0) \overset{3}{\uparrow} \downarrow \overset{2}{MLS(0,t)} \omega \overset{69}{\uparrow} \downarrow \overset{3}{SIGN} : LS(1). \\
 & LS(1) \overset{5}{\uparrow} \downarrow \overset{4}{MLS(1,t)} \omega \overset{69}{\uparrow} \downarrow \overset{5}{SIGN} : LS(2).LS(2) \overset{7}{\uparrow} \downarrow \overset{6}{MLS(2,t)} \omega \overset{69}{\uparrow} \downarrow \overset{7}{SIGN} : LS(3). \\
 & LS(3) \overset{9}{\uparrow} \downarrow \overset{8}{MLS(3,t)} \omega \overset{69}{\uparrow} \downarrow \overset{9}{SIGN} : LS(4).LS(4) \overset{11}{\uparrow} \downarrow \overset{10}{MLS(4,t)} \omega \overset{69}{\uparrow} \downarrow \overset{11}{SIGN} : LS(5). \\
 & LS(5) \overset{13}{\uparrow} \downarrow \overset{12}{MLS(5,t)} \omega \overset{69}{\uparrow} \downarrow \overset{13}{SIGN} : LS(6).LS(6) \overset{15}{\uparrow} \downarrow \overset{14}{MLS(6,t)} \omega \overset{69}{\uparrow} \downarrow \overset{15}{SIGN} : LS(7). \\
 & LS(7) \overset{17}{\uparrow} \downarrow \overset{16}{MLS(7,t)} \omega \overset{69}{\uparrow} \downarrow \overset{17}{SIGN} : LS(8).LS(8) \overset{19}{\uparrow} \downarrow \overset{18}{MLS(8,t)} \omega \overset{69}{\uparrow} \downarrow \overset{19}{SIGN} : LS(9). \\
 & LS(9) \overset{21}{\uparrow} \downarrow \overset{20}{MLS(9,t)} \omega \overset{69}{\uparrow} \downarrow \overset{21}{SIGN} : LS(10).LS(10) \overset{23}{\uparrow} \downarrow \overset{22}{MLS(10,t)} \omega \overset{69}{\uparrow} \downarrow \overset{23}{SIGN} : \\
 & LS(11).LS(11) \overset{25}{\uparrow} \downarrow \overset{24}{MLS(11,t)} \omega \overset{69}{\uparrow} \downarrow \overset{25}{WR} : NO - LS.JMP : EM \omega \overset{69}{\uparrow} \downarrow \overset{26}{MLS(1,t-1)} \overset{30}{\uparrow} \\
 & \overset{29}{SIGN} : LS(1).LS(1) \overset{27}{\uparrow} \omega \overset{4}{\downarrow} \overset{27}{SIGN} : LS(0).LS(0) \overset{28}{\uparrow} \omega \overset{2}{\downarrow} \overset{28}{SIGN} : LS(2).LS(2) \overset{29}{\uparrow} \omega \overset{6}{\downarrow} \\
 & \downarrow \overset{29}{SIGN} : LS(4).LS(4) \overset{25}{\uparrow} \omega \overset{10}{\downarrow} \overset{30}{MLS(2,t-1)} \overset{33}{\uparrow} \overset{31}{SIGN} : LS(2).LS(2) \overset{31}{\uparrow} \omega \overset{6}{\downarrow} \overset{31}{SIGN} : \\
 & LS(1).LS(1) \overset{32}{\uparrow} \omega \overset{4}{\downarrow} \overset{32}{SIGN} : LS(3).LS(3) \overset{1}{\uparrow} \omega \overset{4}{\downarrow} \overset{33}{MLS(3,t-1)} \overset{37}{\uparrow} \overset{37}{SIGN} : LS(3).LS(3) \\
 & \overset{34}{\uparrow} \omega \overset{8}{\downarrow} \overset{34}{SIGN} : LS(2).LS(2) \overset{35}{\uparrow} \omega \overset{6}{\downarrow} \overset{35}{SIGN} : LS(6).LS(6) \overset{36}{\uparrow} \omega \overset{14}{\downarrow} \overset{36}{SIGN} : LS(5). \\
 & LS(5) \overset{1}{\uparrow} \omega \overset{12}{\downarrow} \overset{37}{MLS(4,t-1)} \overset{40}{\uparrow} \overset{38}{SIGN} : LS(4).LS(4) \overset{38}{\uparrow} \omega \overset{10}{\downarrow} \overset{38}{SIGN} : LS(1).LS(1) \overset{39}{\uparrow} \\
 & \omega \overset{4}{\downarrow} \overset{39}{SIGN} : LS(5).LS(5) \overset{1}{\uparrow} \omega \overset{12}{\downarrow} \overset{40}{MLS(5,t-1)} \overset{45}{\uparrow} \overset{41}{SIGN} : LS(5).LS(5) \overset{41}{\uparrow} \omega \overset{12}{\downarrow} \overset{41}{SIGN} : \\
 & \overset{42}{SIGN} : LS(4).LS(4) \overset{42}{\uparrow} \omega \overset{10}{\downarrow} \overset{42}{SIGN} : LS(3).LS(3) \overset{43}{\uparrow} \omega \overset{8}{\downarrow} \overset{43}{SIGN} : LS(6).LS(6) \overset{44}{\uparrow} \omega \overset{14}{\downarrow} \\
 & \downarrow \overset{44}{SIGN} : LS(11).LS(11) \overset{1}{\uparrow} \omega \overset{24}{\downarrow} \overset{45}{MLS(6,t-1)} \overset{50}{\uparrow} \overset{46}{SIGN} : LS(6).LS(6) \overset{46}{\uparrow} \omega \overset{14}{\downarrow} \overset{46}{SIGN} : \\
 & LS(8).LS(8) \overset{47}{\uparrow} \omega \overset{18}{\downarrow} \overset{47}{SIGN} : LS(7).LS(7) \overset{48}{\uparrow} \omega \overset{16}{\downarrow} \overset{48}{SIGN} : LS(10).LS(10) \overset{49}{\uparrow} \omega \overset{22}{\downarrow} \overset{49}{SIGN} : \\
 & \overset{51}{SIGN} : LS(11).LS(11) \overset{1}{\uparrow} \omega \overset{24}{\downarrow} \overset{50}{MLS(7,t-1)} \overset{54}{\uparrow} \overset{51}{SIGN} : LS(7).LS(7) \overset{51}{\uparrow} \omega \overset{16}{\downarrow} \overset{51}{SIGN} : \\
 & LS(6).LS(6) \overset{52}{\uparrow} \omega \overset{14}{\downarrow} \overset{52}{SIGN} : LS(9).LS(9) \overset{53}{\uparrow} \omega \overset{20}{\downarrow} \overset{53}{SIGN} : LS(11).LS(11) \overset{1}{\uparrow} \omega \overset{24}{\downarrow} \overset{54}{SIGN} : \\
 & \overset{58}{MLS(8,t-1)} \overset{58}{\uparrow} \overset{55}{SIGN} : LS(8).LS(8) \overset{55}{\uparrow} \omega \overset{18}{\downarrow} \overset{55}{SIGN} : LS(6).LS(6) \overset{56}{\uparrow} \omega \overset{14}{\downarrow} \overset{56}{SIGN} : \\
 & LS(9).LS(9) \overset{57}{\uparrow} \omega \overset{20}{\downarrow} \overset{57}{SIGN} : LS(11).LS(11) \overset{1}{\uparrow} \omega \overset{24}{\downarrow} \overset{58}{MLS(9,t-1)} \overset{62}{\uparrow} \overset{62}{SIGN} : LS(9). \\
 & LS(9) \overset{59}{\uparrow} \omega \overset{20}{\downarrow} \overset{59}{SIGN} : LS(7).LS(7) \overset{60}{\uparrow} \omega \overset{16}{\downarrow} \overset{60}{SIGN} : LS(8).LS(8) \overset{61}{\uparrow} \omega \overset{18}{\downarrow} \overset{61}{SIGN} : \\
 & LS(11).LS(11) \overset{1}{\uparrow} \omega \overset{24}{\downarrow} \overset{62}{MLS(10,t-1)} \overset{66}{\uparrow} \overset{63}{SIGN} : LS(10).LS(10) \overset{63}{\uparrow} \omega \overset{22}{\downarrow} \overset{63}{SIGN} : LS(2).
 \end{aligned}$$

$$\begin{aligned}
 &LS(2) \overset{64}{\uparrow} \omega \overset{6}{\downarrow} \overset{64}{\downarrow} SIGN : LS(4).LS(4) \overset{65}{\uparrow} \omega \overset{10}{\downarrow} \overset{65}{\downarrow} SIGN : LS(11).LS(11) \overset{1}{\uparrow} \omega \overset{24}{\downarrow} \overset{66}{\downarrow} \\
 &MLS(11, t-1) \overset{68}{\uparrow} SIGN : LS(11).LS(11) \overset{67}{\uparrow} \omega \overset{24}{\downarrow} \overset{67}{\downarrow} SIGN : LS(0).LS(0) \overset{3}{\uparrow} \omega \overset{2}{\downarrow} \overset{68}{\downarrow} WR : \\
 &NO - MLS. \overset{1}{\uparrow} \omega \overset{69}{\downarrow} OUTPUT.PR : WAIT.S_k
 \end{aligned}$$

где S_n и S_k – соответственно операторы начала и конца алгоритма; $INPUT$ и $OUTPUT$ – соответственно операторы ввода и вывода данных; $MLS(i, t-1)$ – в предыдущем $(t-1)$ технологическом цикле агрегат находился в i -том состоянии; $SIGN:LS(i)$ – определение значения i -го состояния агрегата; $LS(i)$ – i -тое состояние агрегата истинно; $MLS(i, t)$ – регистрация i -го состояния в текущем (t) технологическом цикле; $WR:NO-LS$ – регистрация сообщения «состояние не определено»; $JMP:EM$ – безусловный переход к обработке неопределённых ситуаций; $WR:NO-MLS$ – регистрация сообщения «нет памяти состояния»; $PR:WAIT$ – ожидание начала следующего технологического цикла.

Выводы. Таким образом, были получены алгоритмы функционирования системы управления рулевым приводом, понятно описывающие состояния рулевого привода. Также получен граф состояний рулевого привода, детально описаны его вершины – состояния рулевого привода и условия переходов между ними.

Разработана логическая схема алгоритма, которая в компактной форме описывает процесс функционирования рулевого привода и может быть использована в дальнейшем в качестве базовой при разработке или исследованиях систем управления рулевыми приводами.

Дальнейшие исследования следует связать с глубоким анализом аварийных состояний, не описанных в графе, а также способами их определения, оценки и последующей компенсации.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Пипченко А. Д. Повышение надежности работы рулевых машин методами теории отказа-толерантного управления / А. Д. Пипченко, В. А. Шевченко // Материалы Международной научно-практической конференции. – Херсон : Издательство Херсонского государственного морского института, 2009. – С. 268-272.

2. Пипченко А. Н. Синтез функциональных алгоритмов управления в многоуровневых распределенных микропроцессорных системах управления судовой электроэнергетики / А. Н. Пипченко, В. В. Пономаренко // Тез. докладов 7-ой Всесоюзной научно-технической конференции «Проблемы комплексной автоматизации судовых технических средств». – Л. : НТО им. А.Н. Крылова, 1989. – С. 28-29.

Шевченко В.А. ВЕРИФІКАЦІЯ ФУНКЦІОНАЛЬНИХ АЛГОРИТМІВ СИСТЕМИ УПРАВЛІННЯ СУДНОВИМ РУЛЬОВИМ ПРИВОДОМ

В роботі верифіковані алгоритми функціонування системи керування стерновим приводом. Отриманий граф станів стернового приводу, описані стани стернового приводу та умови переходу між ними.

Розроблена логічна схема алгоритму функціонування системи керування стерновим приводом, що може лягти в основу при розробці програмного забезпечення таких систем.

Ключові слова: судно, стерновий привод, алгоритм, граф, результатна функція.

Shevchenko V.A. VERIFICATION OF THE FUNCTIONAL ALGORITHMS SHIP MANAGEMENT STEERING GEAR

This paper presents verified control algorithms of steering gear control system. Steering gear state graph, described states and transfer conditions between them are received.

Logic diagram of steering gear control system is developed. Such diagram could be used as a basic when developing steering gear control systems software.

Keywords: ship, steering gear, algorithm, graph, resultant function.