

УДК 656.257:681.32

Усовершенствование методов испытаний микропроцессорной централизации на безопасность применения

В. Ф. Кустов, А. Ю. Каменев

Украинская государственная академия железнодорожного транспорта
(г. Харьков, Украина)

Кафедра «Автоматика и компьютерное телеуправление движением поездов»

kvf@satep.com.ua

Аннотация. Усовершенствован метод имитационных испытаний микропроцессорной централизации стрелок и сигналов путем использования имитационной модели нижнего уровня и реальных устройств верхнего и среднего уровней. Разработан метод комбинированных испытаний путем синтеза имитационного и физического моделирования работы устройств нижнего уровня. Предложены технические средства для их реализации. Рассмотрен вопрос практического применения приведенных результатов работы.

Ключевые слова: имитационные испытания; комбинированные испытания; метод обособленных групп; метод обособленных объектов; метод обособленных каналов; микропроцессорная централизация.

1 Введение

Одним из основных этапов доказательства безопасности железнодорожной автоматики и телемеханики (ЖАТ), в том числе микропроцессорной централизации стрелок и сигналов (МПЦ), являются испытания, проводимые на этапе разработки и внедрения. Среди них выделяют испытания имитационные (машинных моделей) и стендовые [1], [2]. Методы и средства их проведения, описанные в работах [1]–[4], обладают такими недостатками, как:

– неполный учет особенностей построения и функционирования много-

уровневых иерархических систем управления, к которым следует отнести МПЦ, в том числе разделение функций программного (ПО) и аппаратного обеспечения (АО);

– ограниченные возможности и сложность подтверждения адекватности имитационных (машинных) моделей системы в целом, реализованных специальным ПО на ЭВМ;

– сложность обеспечения необходимого тестового покрытия, для чего требуются значительные ресурсы при проведении стендовых испытаний опытной системы.

Для решения указанных проблем предлагается использование моделей нижнего уровня и реальных устройств верхнего и среднего уровней на этапе имитационных испытаний и имитационного и физического моделирования работы устройств нижнего уровня на этапе стендовых испытаний [5], [6]. В основу проводимых разработок положены условная порядковая классификация (УПК) моделей каждого объекта *obj*, задействованных в испытаниях (табл. 1), и взаимные отношения между объектами управления и контроля (ОУК), их связями и свойствами, установленные в работах [7], [8].

ТАБЛИЦА 1 Условная порядковая классификация моделей, задействованных в испытаниях

Порядок	Принцип абстракции	Изм. во времени	Назначение	Обозначение
0	Идентичная оригиналу	Динамическая	Динамические испытания	$model^0(obj), obj^{(0)}$
1	Физическая	Динамическая	Динамические испытания	$model^1(obj), obj^{(1)}$
2	Биологическая	Динамическая	Динамические испытания	$model^2(obj), obj^{(2)}$
3	Имитационная	Динамическая	Динамические испытания	$model^3(obj), obj^{(3)}$
4	Математическая	Динамическая	Конфигурация ПО	$model^4(obj), obj^{(4)}$

2 Метод имитационных испытаний микропроцессорной централизации

Существующий подход к проведению имитационных испытаний ЖАТ предполагает использование единой имитационной модели всех компонентов системы, реализованной на вычислительном устройстве, которая воспроизводит их функционирование (в том числе и при воздействии дестабилизирующих факторов) [1], [4]. Учитывая, что общее количество единиц оборудования верхнего и среднего уровней МПЦ значительно меньше нижнего (которое определяется количеством ОУК), предлагается исполь-

зовать специализированную имитационную модель (СИМ) нижнего уровня, воспроизводящую работу микропроцессорных объектных контроллеров (МПК) согласно протоколу взаимодействия. Устройства верхнего и среднего уровней должны обрабатываться при этом реальным программно-аппаратным комплексом. Формализованное описание предлагаемого подхода на примере МПЦ с трехканальной подсистемой обработки логических зависимостей (ПОЛЗ) на базе промышленных серверов, согласно приведенной УПК (табл. 1), представляется следующими отношениями [9]:

$$\begin{aligned}
 model^{0 \vee 2 \vee 3}(MPC) \subset & \left\{ \begin{aligned} & model^{2 \vee 3}(DSP) \times model^{0 \vee 3}(int_{0,1}) + \\ & + model^2(SHN) \times model^0(int_{0,1}), \\ & [model^0(ARM_{DSP-1} \cup ARM_{DSP-2}) \times model^{0 \vee 3}(int_{0,1}) + \\ & + model^0(ARM_{SHN}) \times model^0(int_{0,1})] \times model^0(int_{1,2}), \end{aligned} \right. \quad (1) \\
 \subset model^{2 \vee 3}(DF) \times & \left\{ \begin{aligned} & model^0(SER_1 \cup SER_2 \cup SER_3) \times \\ & \times model^0(int_{1,2}) \times model^0(FILE_{int_{2,3}}), \\ & model^3 \left(\bigcup_{l=1}^m MPK \times A \right) \times model^0(FILE_{int_{2,3}}), \end{aligned} \right.
 \end{aligned}$$

где DF , DSP , SHN , ARM , $FILE$, SER , int , MPK , A – обозначения соответственно дестабилизирующих факторов; дежурного по станции (ДСП); электромеханика СЦБ (ШН); автоматизированных рабочих мест (АРМ) персонала; программных файлов; серверов ПОЛЗ; интерфейсов взаимодействия между уровнями; множеств МПК и ОУК;

m – количество МПК на станции, МПЦ которой проходит испытания ($m = [MPK] = [A]$).

Индексами элементов в формуле (1) обозначена их принадлежность к другим элементам, причем индексы возле символа SER определяют номер сервера (канала резервирования) ПОЛЗ, а возле int – номера уровней, между которыми обеспечи-

вает взаимодействие соответствующий интерфейс (0 – персонал МПЦ, 1 – верхний, 2 – средний, 3 – нижний). Например, $int_{0,1}$ обозначает человеко-машинный интерфейс оперативного и технического персонала; $int_{1,2}$ – интерфейс между верхним и средним уровнем и т. д.

Согласно выражению (1), рассматриваемый метод предполагает создание средствами СИМ виртуальной интерфейсной линии между средним и нижним уровнем (рис. 1, а). Объектами испытаний в данном случае выступают ПО и АО АРМ ДСП № 1, 2 (основного и резервного), АРМ ШН, серверов № 1, 2, 3 ПОЛЗ и устройства интерфейса между ними, реализованные основной и резервной локальными вычислительными сетями (ЛВС-1, 2).

Процедура испытаний выполняется по схеме, изображенной на рис. 1, б. Исходной составляющей процесса является СИМ 1, синтезируемая по связи 3 на основании конфигурационного файла, учитывающего отношения между ОУК, их связями и свойствами для конкретной станции. Для автоматизации испытаний формируется программный комплекс тестирования (ПКТ) МПЦ 2, синтезируемый аналогично СИМ. Процедура выполняется на основании программы и методики (ПМИ), с использованием которой формируется 4 тестовый скрипт (сценарий), закладываемый 5 в ПКТ. Согласно ему ПКТ устанавливает начальные, текущие и конечные состояния ОУК (в том числе имитирует их отказы и сбои) 6 и воспроизводит работу ДСП 7.

Установленные состояния ОУК и команды ДСП подаются к ПОЛЗ, реакция которой контролируется средствами СИМ, АРМ ДСП, ШН и ПКТ по связям соответственно 8, 9, 10, 11. Также по связи 11 ПКТ устанавливает различные состояния серверов (в том числе дестабилизирует их работу) и фиксирует внутреннюю их реакцию на воздействия внешних факторов. Реакция на воздействия ПКТ и

СИМ автоматически регистрируется 12 в электронном протоколе. Дополнительно информация архивируется службой протоколирования МПЦ, которая может быть считана средствами АРМ ШН (куда она поступает по связи 10) или непосредственно (путем расшифровки протоколов). Сравнение протоколов с электронными отчетами ПКТ 20 дает возможность проверки функционирования службы протоколирования как неотъемлемой составляющей безопасности применения системы. При отсутствии автоматизации испытаний человек-испытатель непосредственно руководствуется положениями ПМИ 13, имитируя согласно им работу ДСП 14, изменяя состояния ОУК 15 и внося дестабилизацию в работу ПОЛЗ 16. С помощью интерфейсов оператора 15, обслуживающего персонала 10, настройки и тестирования 15, 16 исследуется реакция различных узлов на воздействия, которые заносятся в протокол испытаний 18. При частичной автоматизации ПКТ может использоваться для выполнения отдельных этапов ПМИ, для чего осуществляется его предварительная настройка до начала эксперимента, запуск и останов 17 в процессе испытаний.

При условии адекватности СИМ предлагаемый метод обеспечивает проверку безопасности функционирования устройств верхнего и среднего уровней, а также ЛВС между ними в полном объеме. Адекватность СИМ подтверждается методом тестирования на предмет соответствия кодовых комбинаций, которые она генерирует, реальному протоколу обмена. Ограничение предложенного метода – отсутствие проверки нижнего уровня, который предполагается идеально функционирующим. Согласно нормативному документу [2] проверка всех уровней и подсистем в комплексе возлагается на стендовые испытания.

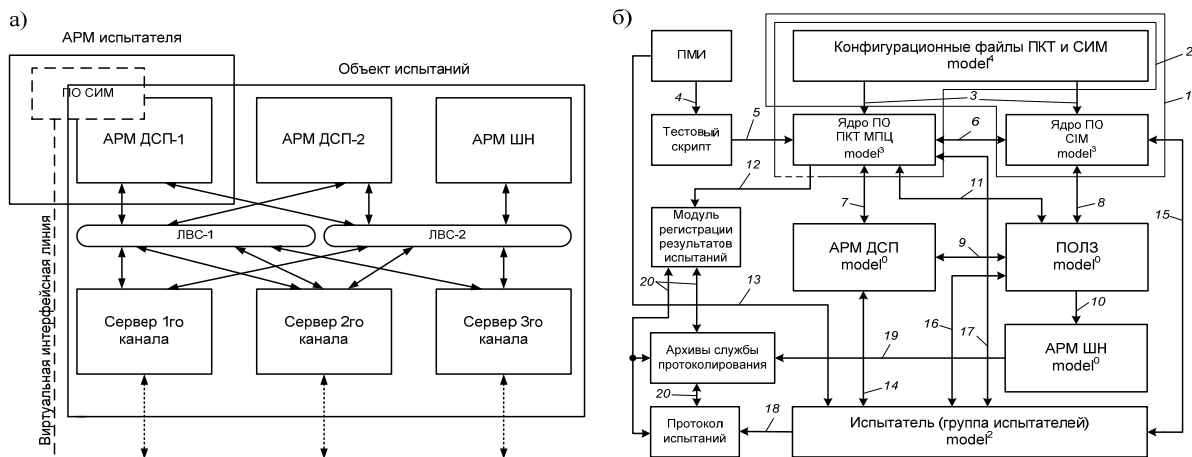


Рис. 1 Структурная схема испытательного комплекса специализированной имитационной модели (а) и взаимодействие составляющих процесса имитационных испытаний (б)

3 Методы стендовых испытаний микропроцессорной централизации

Стендовые испытания предполагают максимальную приближенность к эксплуатационным условиям, поэтому их результаты принято считать наиболее достоверными из тех, которые достигаются в лабораторных условиях [1]–[3]. Традиционные их методы (ТСИ) для систем МПЦ предполагают полное воспроизведение устройств верхнего и среднего уровней, но только частичное воспроизведение нижнего – путем подключения ограниченной выборки МПК к стенду. Учитывая взаимные зависимости между ОУК, управляемых МПК, при ТСИ обеспечивается и ограниченное тестовое покрытие технологических ситуаций. Для его увеличения требуется и увеличение выборки, что приводит к росту ресурсов для испытаний [5].

Для обеспечения необходимого тестового покрытия при минимальных ресурсах разработаны методы комбинирован-

ных испытаний (МКИ) как подвид стендовых путем синтеза имитационного и физического моделирования работы МПК и ОУК [6].

МКИ выполняются по процедуре, аналогичной изображенной на рис. 1, б, но при условии, что работа части МПК воспроизводится реальными устройствами, а ОУК – физическими моделями (макетами), другой части МПК и ОУК – программными модулями СИМ. При этом в составе стенда используются два комплекта драйверов взаимодействия с нижним уровнем: реальные (для первой части) и виртуальные (для второй). С учетом изоморфизма трех уровней МПЦ на множестве $Y = A \cup Z \cup U$ (где A , Z и U – подмножества соответственно ОУК, их связей и свойств) [8] для достижения данной цели множества ОУК A , их программные модули ПОЛЗ ML и МПК LL разбиваются на изоморфные классы толерантности (τ), состоящей в управлении единой группой ОУК и использовании общего типа драйвера:

$$\left\{ \begin{array}{l} A = \bigcup_{i=1}^n A_i, \bigcap_{i=1}^n A_i = \emptyset, \\ LL = \bigcup_{i=1}^n LL_i, \bigcap_{i=1}^n LL_i = \emptyset, \\ ML = \bigcup_{i=1}^n ML_i, \bigcap_{i=1}^n ML_i = \emptyset, \end{array} \right. \quad \tau \subset \Lambda \times \Lambda : \left\{ \begin{array}{l} \forall A_i \leftarrow \gamma_i(\alpha) \rightarrow \exists! LL_i \subset LL, \\ \forall LL_i \leftarrow \gamma_i(\alpha) \rightarrow \exists! ML_i \subset ML, \\ \forall ML_i \leftarrow \gamma_i(\alpha) \rightarrow \exists! A_i \subset A, \end{array} \right. \quad (2)$$

где $\Lambda = \{\alpha_j\} = \left\{ \left\{ a_j \in A, ll_j \in LL, ml_j \in ML \right\}_j \right\}$ – множества взаимно однозначных элементов;
 $\gamma_i(\alpha) = \overline{\gamma_i(\alpha)} = \langle \gamma_{i1}, \gamma_{i2}, \dots, \gamma_{im} \rangle$ – функция, реализуемая драйвером i -й группы ОУК.

Векторный характер функций $\gamma_i(\alpha) \in Y(\alpha)$ обусловлен многоканальностью МПК нижнего уровня, каждый канал которых управляется своей составляющей драйвера. В общем виде МКИ формализуются следующим выражением:

$$\left\{ \begin{array}{l} \exists(\Lambda_i \subset \Lambda) \rightarrow \left[\begin{array}{l} \exists \left([a_{\vartheta} \in A_i] \leftrightarrow [ll_{\vartheta\eta} \in \overline{ll}_{\vartheta} \in LL_i] \xrightarrow{\gamma_{i\eta}} [ml_{\vartheta} \in ML_i] \right): \\ (a_{\vartheta} \rightarrow \text{model}^1(a_{\vartheta})) \wedge (ll_{\vartheta\eta} \rightarrow \text{model}^0(ll_{\vartheta\eta})) \wedge (\gamma_{i\eta} \rightarrow \text{model}^0(\gamma_{i\eta})) \end{array} \right], \\ \exists(\Lambda_v \subset \Lambda) \rightarrow \left[\begin{array}{l} \exists \left([a_{\iota} \in A_v] \leftrightarrow [ll_{\iota\varsigma} \in \overline{ll}_{\iota} \in LL_v] \xrightarrow{\gamma_{i\varsigma}} [ml_{\iota} \in ML_v] \right): \\ (a_{\iota} \rightarrow \text{model}^1(a_{\iota})) \wedge (ll_{\iota\varsigma} \rightarrow \text{model}^0(ll_{\iota\varsigma})) \wedge (\gamma_{i\varsigma} \rightarrow \text{model}^0(\gamma_{i\varsigma})) \end{array} \right] \end{array} \right. \quad (3)$$

$i, v = \overline{1, n}, \vartheta = \overline{1, [\Lambda_i]}, \iota = \overline{1, [\Lambda_v]}, \eta, \varsigma = \overline{1, m}, \vartheta\eta \neq \iota\varsigma,$

где $\overline{ll}_{\vartheta}$ и \overline{ll}_{ι} – МПК, а $ll_{\vartheta\eta}$ и $ll_{\iota\varsigma}$ – их составляющие каналы управления и контроля;
 a_{ϑ} и a_{ι} – ОУК, подключенные к МПК;
 ml_{ϑ} и ml_{ι} – программные модули соответствующих ОУК в составе ПО ПОЛЗ;
 i и v – номера групп (классов толерантности), определенных формулой (2);
 ϑ и ι – номера МПК и ОУК соответственно в составе групп i и v ;

η и ς – номера каналов в составе МПК соответственно ϑ и ι ;
 n и m – количество соответственно групп МПК (ОУК) и их каналов.
 Техническая реализация МКИ выполняется на базе комбинированного испытательного комплекса (КИК МПЦ), общая структура которого, согласно (3), приведена на рис. 2 [10].

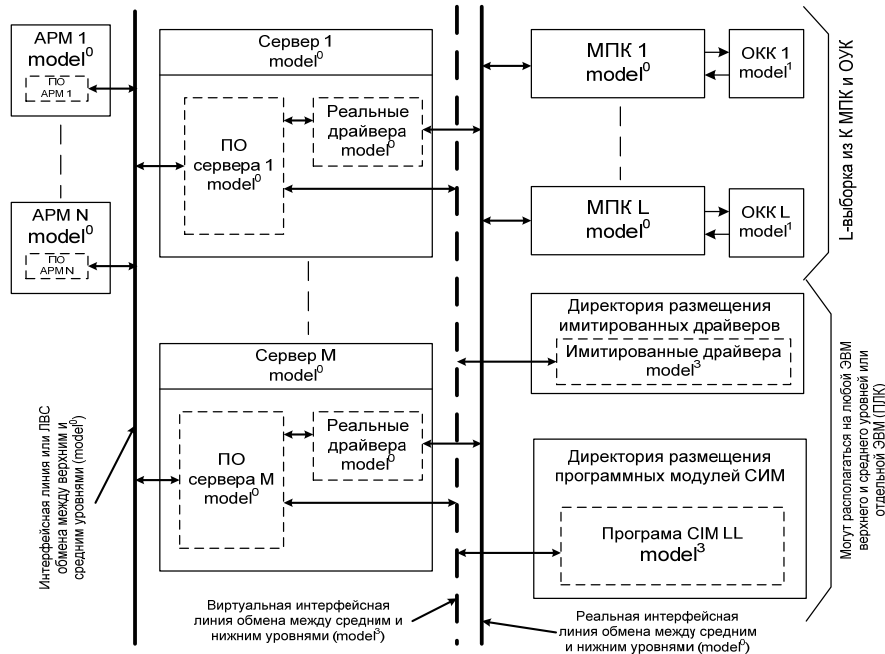


Рис. 2 Обобщенная структурная схема комбинированного испытательного комплекса

Прикладное ПО каждого сервера (канала) ПОЛЗ взаимодействует с реальными МПК и подключенными к ним макетами ОУК по реальной интерфейсной линии и с программными модулями СИМ – по виртуальной. Таким образом, выполняется программно-пространственное разграничение доступа к элементам нижнего уровня, воспроизводимым моделями различных порядков. Для этого определенной корректировки требует ПО ПОЛЗ, которое необходимо возвращать в исходное состояние после окончания испытаний. Поэтому предпочтительным для применения МКИ являются МПЦ с единым ядром ПО, адаптация которого под конкретную станцию и интерфейсные линии выполняется редактированием конфигурационных файлов [11]. Такая архитектура рекомендована документом Р-843 [12].

В зависимости от способа выполненного разграничения в пределах МКИ разработаны три метода испытаний, определенные в соответствии с выражением (3).

1. Метод обособленных групп (МОГ), который состоит в следующем:

– в пределах каждой группы $\Lambda_i \subset \Lambda$ порядок моделей каждого элемента $\alpha_g = \{a_g, ll_g, ml_g\}_g$ и интерфейсного драйвера $\gamma_i(\alpha_g) \in Y(\alpha_g)$ не изменяется: $\alpha_g^{(p1, p1, p3=\text{const})}$, $\gamma_i^{(p4=\text{const})}(\alpha_g)$, где $p1, p2, p3, p4$ – порядки моделей ОУК, МПК, программных модулей ПО ПОЛЗ и драйверов (или их составляющих);

– существует хотя бы одна группа $\Lambda_i \subset \Lambda$, для которой $p1 = 1, p2 = 0, p4 = 0$ и хотя бы одна группа $\Lambda_{k \neq i} \subset \Lambda$, для которой $p1 = 3, p2 = 3, p4 = 3$ ($p3 \equiv 0$ во всех случаях):

$$\begin{cases} \exists \Lambda_i \rightarrow \forall a_g \in A_i : (a_g \rightarrow \text{model}^1(a_g)) \wedge (ll_g \rightarrow \text{model}^0(ll_g)) \wedge (\gamma_i \rightarrow \text{model}^0(\gamma_i)), \\ \exists \Lambda_{k \neq i} \rightarrow \forall a_k \in A_k : (a_k \rightarrow \text{model}^3(a_k)) \wedge (ll_k \rightarrow \text{model}^3(ll_k)) \wedge (\gamma_k \rightarrow \text{model}^3(\gamma_k)), \end{cases} \quad (4)$$

$$i, k = \overline{1, n}, \vartheta = \overline{1, [\Lambda_i]}, \iota = \overline{1, [\Lambda_k]}.$$

Таким образом, сущность МОГ состоит в испытаниях при взаимодействии ПО ПОЛЗ с реальными МПК и макетами ОУК для отдельных их групп без деления на элементы, с одной стороны, и с модулями СИМ для других групп также без их деления – с другой.

Для технической реализации МОГ в ПО каждого сервера ПОЛЗ или его конфигурационных файлах для каждой группы Λ_i , согласно выражению (4), прописываются адреса доступа к разным точкам подключения интерфейсных драйверов – соответственно реальных $\gamma_i^{(0)}(\alpha_j)$ и имитированных $\gamma_{k \neq i}^{(3)}(\alpha_j)$. На примере взаимодействия групп стрелок и светофоров с реальными драйверами, а групп ввода-вывода и датчиков счета осей (ДСО) с имитируемыми для 1-го сервера ПОЛЗ системы типа МПЦ-С настройка КИК

МПЦ на МОГ выполняется корректировкой основного конфигурационного файла (табл. 2) [13].

Параметры каждой интерфейсной САН-линии для каждого канала определенной группы описываются соответствующей строчкой в табл. 2.

Путь доступа к каждому драйверу определяется параметром $Name = \text{"[путь_доступа]"} : \text{"[/dev/[название_файла]"} - \text{ для реальных драйверов и } \text{"[/net/DSP1/dev/[название_файла]"} - \text{ для имитируемых. Параметры временных («темповых») файлов драйверов идентичны с позиции взаимодействия ПО всех серверов ПОЛЗ } (\varepsilon \subset Y^{(0)}(\alpha) \times Y^{(3)}(\alpha), \text{ где } \varepsilon - \text{ отношение эквивалентности [9]), поэтому формат данных, поступление которых моделируется с нижнего уровня или на него, не зависит от порядка модели и соответствует протоколу обмена между сер-$

верами ПОЛЗ и МПК. Таким образом обеспечивается отмеченное на рис. 2 разграничение реальной и виртуальной линий.

Возврат к рабочей версии ПО после испытаний по МОГ требует корректировки путей к точкам подключения драйверов, которые в составе КИК МПЦ воспроизводились СИМ. На примере табл. 2 это достигается исключением фрагмента «/net/DSP1» в строках с 6-й по 9-ю.

Основным преимуществом МОГ следует считать простоту перехода от рабочей к тестовой версии ПО и наоборот. Главный недостаток проявляется в том, что при подключении к стенду ограниченной выборки МПК и ОУК не воспроизводится влияние на них работы других

элементов той же группы, с которыми подключенные ОУК связаны логически.

2. *Метод обособленных объектов (МОО)*, который состоит в наличии хотя бы одной группы Λ_i , в пределах которой существует хотя бы одна пара элементов $\alpha_g = \{a_g, ll_g, ml_g\}_g$, $\alpha_{i \neq g} = \{a_i, ll_i, ml_i\}_i$ и соответствующих им драйверов $\gamma_i(\alpha_g)$, $\gamma_i(\alpha_i)$, порядки моделей составляющих которых следующие: для элементов $\alpha_g \leftrightarrow \gamma_i(\alpha_g) - p1 = 1, p2 = 0, p3 = 0, p4 = 0$; для элементов $\alpha_i \leftrightarrow \gamma_i(\alpha_i) - p1 = 3, p2 = 3, p3 = 0, p4 = 3$:

$$\exists \Lambda_i \rightarrow \exists (a_g, a_i) \in A_i : \begin{cases} (a_g \rightarrow \text{model}^1(a_g)) \wedge (ll_g \rightarrow \text{model}^0(ll_g)) \wedge (\gamma_i \rightarrow \text{model}^0(\gamma_i)), \\ (a_i \rightarrow \text{model}^3(a_i)) \wedge (ll_i \rightarrow \text{model}^3(ll_i)) \wedge (\gamma_i \rightarrow \text{model}^3(\gamma_i)), \end{cases} \quad (5)$$

$$i = \overline{1, n}, \quad g, i = \overline{1, [\Lambda_i]}.$$

Таким образом, при использовании МОО из всех $m_1 + m_2 = [\Lambda_i]$ объектов отдельной группы Λ_i работа m_1 компонентов нижнего уровня и их драйверов воспроизводится реальными МПК с подключенными физическими макетами ОУК, а других m_2 – модулями СИМ.

В отличие от МОГ техническая реализация МОО, согласно формуле (5), требует не только разадресации доступа к точкам подключения интерфейсных драйверов, воспроизводимых разнопорядковыми моделями, но и формирования дополнительных интерфейсных линий, по которым должны обмениваться данными программные модули части объектов определенной группы (линия, обслуживаемая драйверами $\gamma_i^{(0)}(\alpha_j)$ для m_1 объектов и драйверами $\gamma_{k \neq i}^{(3)}(\alpha_j)$ для m_2 объектов). Фрагмент конфигурации первого сервера ПОЛЗ МПЦ-С при ее настройке на испытания по МОО на примере взаимодействия программных

модулей одной стрелки с реальными МПК, а других стрелок с модулями СИМ приведен в табл. 3.

В соответствии с табл. 3 в пределах группы стрелок организовывается две пары CAN-линий (по каждому из двух каналов стрелочных МПК) с разными путями доступа к драйверам (имитируемым и реальным) в параметре Name=" ". Каждой CAN-линии присваивается собственный номер в параметре Number=" " (соответственно 1, 2 для 1-го и 2-го каналов виртуальных линий и 10-го, 11-го – для реальных). Параметр Num_Controllers=" " определяет количество МПК (реальных или имитируемых), подключенных к соответствующим CAN-линиям. Среди $m_1 + m_2 = 18$ стрелок работа $m_2 = 1$ воспроизводится физическими моделями, подключенными к реальным МПК, а $m_2 = 17$ – модулями СИМ, что определено этим параметром в соответствующих описаниях по каждому каналу. На этом корректировка секций описания CAN-линий и драйверов в конфигурационном файле заканчивается.

ТАБЛИЦА 2 Фрагмент конфигурации сервера при использовании метода обособленных групп

Группа	Фрагмент конфигурационного файла		Примечание
Стрелки	1к	<DriverCAN Number="1" Type="ОБЪЕКТ_MK" Name="/dev/CAN1680SW1" Num_Controllers="18" sort="1" Time="250" ProtNameState="D1" ProtNameCtrl="C1"/>	Драйвер ($\gamma_{11}(\alpha)$) ⁽⁰⁾
	2к	<DriverCAN Number="2" Type="ОБЪЕКТ_MK" Name="/dev/CAN1680SW2" Num_Controllers="18" sort="1" Time="250" ProtNameState="D2" ProtNameCtrl="C2"/>	Драйвер ($\gamma_{12}(\alpha)$) ⁽⁰⁾
Светофоры	1к	<DriverCAN Number="3" Type="ОБЪЕКТ_MK" Name="/dev/CAN1680SM1" Num_Controllers="38" sort="1" Time="250" ProtNameState="D3" ProtNameCtrl="C3"/>	Драйвер ($\gamma_{21}(\alpha)$) ⁽⁰⁾
	2к	<DriverCAN Number="4" Type="ОБЪЕКТ_MK" Name="/dev/CAN1680SM2" Num_Controllers="38" sort="1" Time="250" ProtNameState="D4" ProtNameCtrl="C4"/>	Драйвер ($\gamma_{22}(\alpha)$) ⁽⁰⁾
Ввод-вывод	1к	<DriverCAN Number="7" Type="IO_MK" Name="/net/DSP1/dev/CAN1680IN1" Num_Controllers="6" sort="0" Time="250" ProtNameState="D5" ProtNameCtrl="C5"/>	Драйвер ($\gamma_{31}(\alpha)$) ⁽³⁾
	2к	<DriverCAN Number="8" Type="IO_MK" Name="/net/DSP1/dev/CAN1680IN2" Num_Controllers="6" sort="0" Time="250" ProtNameState="D6" ProtNameCtrl="C6"/>	Драйвер ($\gamma_{32}(\alpha)$) ⁽³⁾
ДСО	1к	<DriverCAN Number="5" Type="SKPU_MK" Name="/net/DSP1/dev/CAN1680AX1" Num_Controllers="43" sort="1" Time="400" ProtNameState="A1" ProtNameCtrl=""/>	Драйвер ($\gamma_{41}(\alpha)$) ⁽³⁾
	2к	<DriverCAN Number="6" Type="SKPU_MK" Name="/net/DSP1/dev/CAN1680AX2" Num_Controllers="43" sort="1" Time="400" ProtNameState="A2" ProtNameCtrl=""/>	Драйвер ($\gamma_{42}(\alpha)$) ⁽³⁾

Для разграничения доступа каждого объекта по разным CAN-линиям в секциях соответствующих объектов выполняются ссылки на номера этих линий путем изменения значений параметров $CAN=""$ по каждому каналу. В табл. 3 показано, что программный модуль стрелки 1 взаимодействует с реальным МПК по CAN-линиям № 10, 11, а других стрелок – с модулями СИМ по линиям № 1, 2. Если последовательно пересчитывать в табл. 3 стрелки сначала нечетной, а затем четной горловины, то последняя позиция в таблице, определяющая стрелку № 2Q, выбирается по правилам нумерации этих элементов на

станции по каждой горловине согласно положениям, приведенным в [14].

Аналогично выполняется разграничение доступа для объектов других групп. При этом обратный переход от тестовой к рабочей версии ПО состоит в последовательном выполнении следующих операций: удалении описания интерфейсных линий, приведенных в 4-й и 5-й строках табл. 3; установке значений параметров $Num_Controllers = "m_1 + m_2"$ во 2-й и 3-й строках табл. 3; установке значений параметров $CAN=""$, соответствующих номерам линий, определенных во 2-й и 3-й строках табл. 3 для 1-го и 2-го каналов в

секциях объектов, которые были настроены на способ моделирования, определенный ссылкой на драйверы в 4-й и 5-й строках табл. 3 (на примере этой таблицы данный параметр требует корректировки в 6-й и 7-й строках). Процедура проводится для всех групп, разделяемых на объекты с разными способами моделиро-

вания. Учитывая, что рабочая версия ПО отличается от тестовой в нескольких строках конфигурационного файла, размещенных в различных секциях, она требует контрольной проверки с использованием СИМ после комбинированных испытаний.

ТАБЛИЦА 3 Фрагмент конфигурации сервера при использовании метода обособленных объектов

Секции	Фрагмент конфигурационного файла	Примечание
CAN-линии	1к <DriverCAN Number="1" Type="ОБЪЕКТ_МК" Name="/net/DSP1/dev/CAN1680SW1" Num_Controllers="17" sort="1" Time="250" ProtNameState="D1" ProtNameCtrl="C1"/>	Драйвер $(\gamma_{11}(\alpha))^{(0)}$
	2к <DriverCAN Number="2" Type="ОБЪЕКТ_МК" Name="/net/DSP1/dev/CAN1680SW2" Num_Controllers="17" sort="1" Time="250" ProtNameState="D2" ProtNameCtrl="C2"/>	Драйвер $(\gamma_{12}(\alpha))^{(0)}$
	1к <DriverCAN Number="10" Type="ОБЪЕКТ_МК" Name="/dev/CAN1680SW1" Num_Controllers="1" sort="1" Time="250" ProtNameState="D1" ProtNameCtrl="C1"/>	Драйвер $(\gamma_{11}(\alpha))^{(3)}$
	2к <DriverCAN Number="11" Type="ОБЪЕКТ_МК" Name="/dev/CAN1680SW2" Num_Controllers="1" sort="1" Time="250" ProtNameState="D2" ProtNameCtrl="C2"/>	Драйвер $(\gamma_{12}(\alpha))^{(3)}$
Стр. 1	1к <Controller CSID="01" type="1" CAN="10"/>	CAN-линия № 10
	2к <Controller CSID="02" type="1" CAN="11"/>	CAN-линия № 11
Стр. 3	1к <Controller CSID="03" type="1" CAN="1"/>	CAN-линия № 1
	2к <Controller CSID="04" type="1" CAN="2"/>	CAN-линия № 2
Стр. ...	1к
	2к
Стр. 2Q*	1к <Controller CSID="03" type="1" CAN="1"/>	CAN-линия № 1
	2к <Controller CSID="04" type="1" CAN="2"/>	CAN-линия № 2

*Q – количество стрелок четной горловины станции.

Основное преимущество МОО состоит в возможности воспроизведения взаимных зависимостей между объектами одной группы, а недостаток – в относительной сложности перехода от рабочей к тестовой версии ПО и наоборот.

МОО и МОГ могут применяться в комбинации друг с другом, когда часть групп делится на объекты с разными способами моделирования, а другая – не делится.

3. Метод обособленных каналов (МОК), применимый только для систем, в составе нижнего уровня которых имеются многоканальные (резервированные) МПК. Требованиям к нему является наличие хотя бы одной группы Λ_i , в пределах которой есть хотя бы один элемент $\alpha_g = \{a_g, \vec{l}_g, ml_g\}_g$, где $\vec{l}_g = \langle ll_{g1}, ll_{g2}, \dots, ll_{gm} \rangle$, для которого имеют место следующие порядки моделей:

$$\begin{array}{ll}
\text{– для части элементов} & \text{– для другой части} \\
\left(a_g \leftarrow ll_{g\eta} \xrightarrow{\gamma_{g\eta}} ml_g \right) - p1 = 1, p2 = & \left(a_g \leftarrow ll_{g\mu} \xrightarrow{\gamma_{g\mu}} ml_g \right), \mu \neq \eta \\
= 0, p3 = 0, p4 = 0; & \left(a_g \leftarrow ll_{g\mu} \xrightarrow{\gamma_{g\mu}} ml_g \right), \mu \neq \eta - \\
& p1 = 3, p2=3, p3 = 0, p4 = 3:
\end{array}$$

$$\begin{aligned}
\exists \Lambda_i \rightarrow \exists a_g \in A_i : & \left\{ \begin{array}{l} (a_g \rightarrow \text{model}^1(a_g)) \wedge (\exists ll_{g\eta} \rightarrow \text{model}^0(ll_{g\eta})) \wedge (\gamma_i \rightarrow \text{model}^0(\gamma_i)), \\ (a_g \rightarrow \text{model}^3(a_g)) \wedge (\exists ll_{g\mu} \rightarrow \text{model}^3(ll_{g\mu})) \wedge (\gamma_i \rightarrow \text{model}^3(\gamma_i)), \end{array} \right. \quad (6) \\
[\{ll_{g\eta}^{(0)}\}] = \left[\sum_{\eta=1}^{m-\mu} ll_{g\eta} \right], & [\{ll_{g\mu}^{(3)}\}] = \left[\sum_{\mu=1}^{m-\eta} ll_{g\mu} \right], \quad \eta, \mu = \overline{1, m}, \quad \eta \neq \mu.
\end{aligned}$$

Характерной особенностью МОК является использование разных способов моделирования одного ОУК a_g в одном эксперименте. При этом средствами ПО ПОЛЗ он воспринимается единым и неделимым, благодаря чему обеспечивается возможность имитации расхождений его состояния по каналам средствами СИМ при условии, что оставшаяся часть каналов воспроизводится средствами реальных МПК. Учитывая, что СИМ позволяет воспроизвести значительно больше расхождений без конструктивного повреждения физического образца МПК или корректировки его программной прошивки, основным назначением МОК является исследование безопасности применения и надежности системы МПЦ при сбоях, отказах и повреждениях отдельных каналов соответствующих МПК.

Еще одной особенностью МОК является невозможность самостоятельного их использования, необходима обязательная комбинация с МОГ, МОО или с обоими одновременно.

При использовании МОК в комбинации с МОГ каждое из объединений множеств каналов МПК с номерами η и μ в пределах групп Λ_i

$$\begin{aligned}
\bigcup_{g=1}^r ll_{g\eta} \leftrightarrow \gamma_{i\eta}(\alpha), \bigcup_{g=1}^r ll_{g\mu} \leftrightarrow \\
\leftrightarrow \gamma_{i\mu}(\alpha), [\Lambda_i] = r,
\end{aligned}$$

которым соответствуют драйверы этих каналов, воспроизводится моделями с неизменными порядками составляющих своих объектов: или $p1 = 1, p2 = 0$ и $p4 = 0$ (для первого множества), или $p1 = 3, p2 = 3$ и $p4 = 3$ (для второго), т. е. часть каналов группы объектов в целом взаимодействует с реальными МПК, а другая часть – с модулями СИМ без деления групп и выделения из них объектов. При этом множество моделей ОУК распадается на два подмножества с разными значениями $p1$: $A_i^{(p1)} = A_i^{(1)} \cup A_i^{(3)}$.

Согласно формулам (3) и (6):

$$\begin{aligned}
\exists \Lambda_i \rightarrow \forall a_g \in A_i & \left\{ \begin{array}{l} (a_g \rightarrow \text{model}^1(a_g)) \wedge \left(\forall ll_{g\eta} \in \bigcup_{g=1}^r ll_{g\eta} \rightarrow \text{model}^0(ll_{g\eta}) \right) \wedge (\gamma_{i\eta} \rightarrow \text{model}^0(\gamma_{i\eta})), \\ \eta, \mu = \overline{1, m}, \\ \eta \neq \mu \\ g = \overline{1, r} = [\Lambda_i] \end{array} \right. : \left\{ \begin{array}{l} (a_g \rightarrow \text{model}^3(a_g)) \wedge \left(\forall ll_{g\mu} \in \bigcup_{g=1}^r ll_{g\mu} \rightarrow \text{model}^3(ll_{g\mu}) \right) \wedge (\gamma_{i\mu} \rightarrow \text{model}^3(\gamma_{i\mu})). \end{array} \right. \quad (7)
\end{aligned}$$

Конфигурация ПО ПОЛЗ при испытаниях по МОК в комбинации с МОГ с учетом формулы (7) и рис. 2 (на примере группы светофоров системы МПЦ-С) приведена в табл. 4.

В табл. 4 первый канал группы светофоров настроен на физическое моделирование, а второй – на имитационное. Аналогично выполняется настройка для других групп. Переход от рабочей к тестовой версии ПО и наоборот, как и при МОГ, осуществляется только корректировкой параметра *Name* = " " для групп с разными способами моделирования каналов.

Использование МОК в комбинации с МОО состоит в разных способах моделирования различных каналов МПК и их драйверов в пределах одного объекта. Из формулы (3) следует, что условием применения данной комбинации является наличие такой пары элементов $(\alpha_{\vartheta}, \alpha_{i \neq \vartheta}) \in \Lambda_i$ и соответствующих им драйверов $\gamma_i(\alpha_{\vartheta}), \gamma_i(\alpha_i)$, для которых упорядоченные множества порядков моделей каналов элементов $(\overline{ll}_{\vartheta}, \overline{ll}_{i \neq \vartheta})$ и драйверов, соответствующих одним и тем же индексам, не равны друг другу (при выполнении общего условия (6)):

$$\begin{aligned}
 \overline{ll}_{\vartheta}^{(p2_{\vartheta})} &= \overline{ll}_{\vartheta 1}^{(p2_{\vartheta 1})}, \overline{ll}_{\vartheta 2}^{(p2_{\vartheta 2})}, \dots, \overline{ll}_{\vartheta m}^{(p2_{\vartheta m})} \rightarrow \overline{p2}_{\vartheta} = \overline{p2}_{\vartheta 1}, \overline{p2}_{\vartheta 2}, \dots, \overline{p2}_{\vartheta m}, \\
 \overline{ll}_{\iota}^{(p2_{\iota})} &= \overline{ll}_{\iota 1}^{(p2_{\iota 1})}, \overline{ll}_{\iota 2}^{(p2_{\iota 2})}, \dots, \overline{ll}_{\iota m}^{(p2_{\iota m})} \rightarrow \overline{p2}_{\iota} = \overline{p2}_{\iota 1}, \overline{p2}_{\iota 2}, \dots, \overline{p2}_{\iota m}, \\
 \overline{\gamma}_{i \rightarrow \vartheta}^{(p4_{(i \rightarrow \vartheta)})} &= \overline{\gamma}_{(i \rightarrow \vartheta) 1}^{(p4_{(i \rightarrow \vartheta) 1})}, \overline{\gamma}_{(i \rightarrow \vartheta) 2}^{(p4_{(i \rightarrow \vartheta) 2})}, \dots, \overline{\gamma}_{(i \rightarrow \vartheta) m}^{(p4_{(i \rightarrow \vartheta) m})} \rightarrow \\
 &\rightarrow \overline{p4}_{(i \rightarrow \vartheta)} = \overline{p4}_{(i \rightarrow \vartheta) 1}, \overline{p4}_{(i \rightarrow \vartheta) 2}, \dots, \overline{p4}_{(i \rightarrow \vartheta) m}, \\
 \overline{\gamma}_{i \rightarrow \iota}^{(p4_{(i \rightarrow \iota)})} &= \overline{\gamma}_{(i \rightarrow \iota) 1}^{(p4_{(i \rightarrow \iota) 1})}, \overline{\gamma}_{(i \rightarrow \iota) 2}^{(p4_{(i \rightarrow \iota) 2})}, \dots, \overline{\gamma}_{(i \rightarrow \iota) m}^{(p4_{(i \rightarrow \iota) m})} \rightarrow \overline{p4}_{(i \rightarrow \iota)} = \overline{p4}_{(i \rightarrow \iota) 1}, \overline{p4}_{(i \rightarrow \iota) 2}, \dots, \overline{p4}_{(i \rightarrow \iota) m}, \\
 \overline{\gamma}_{i \rightarrow \vartheta}^{(p4_{(i \rightarrow \vartheta)})} &\leftarrow \overline{ll}_{\vartheta}^{(p2_{\vartheta})} \neq \overline{ll}_{\iota}^{(p2_{\iota})} \rightarrow \overline{\gamma}_{i \rightarrow \iota}^{(p4_{(i \rightarrow \iota)})}, p2_{\vartheta \eta} = p4_{\vartheta \eta} = 0 \vee 3, p2_{\iota \eta} = p4_{\iota \eta} = 0 \vee 3, \\
 &i = \overline{1, n}, \vartheta, \iota = \overline{1, m}, \vartheta \neq \iota, \eta = \overline{1, m}.
 \end{aligned}
 \tag{8}$$

ТАБЛИЦА 4 Фрагмент конфигурации при комбинации методов обособленных каналов и групп

Группа	Фрагмент конфигурационного файла	Примечание
Светофоры	1к <DriverCAN Number="3" Type="OBJECT_MK" Name="/dev/CAN1680SM1" Num_Controllers="38" sort="1" Time="250" ProtNameState="D3" ProtNameCtrl="C3"/>	Драйвер $(\gamma_{21}(\alpha))^{(0)}$
	2к <DriverCAN Number="4" Type="OBJECT_MK" Name="/net/DSP1/dev/CAN1680SM2" Num_Controllers="38" sort="1" Time="250" ProtNameState="D4" ProtNameCtrl="C4"/>	Драйвер $(\gamma_{22}(\alpha))^{(3)}$

При этом значения отдельных элементов $p2_{\eta\eta}$, $p4_{\eta\eta}$, $p2_{\eta\eta}$, $p4_{\eta\eta}$ могут совпадать, существенным является только порядок их расположения и соответствие условию (6).

Таким образом, в общем случае при комбинации МОК с МОО группы делятся по такому правилу:

– из всех $r_1 + r_2 + r_3 = r$ ее членов для r_1 объектов организовывается взаимодействие только с модулями СИМ посредством имитированных драйверов по всем каналам;

– для r_2 – только с реальными МПК с подключенными макетами ОУК по всем каналам;

– для r_3 – с модулями СИМ для части каналов и с реальными МПК и ОУК для другой части каналов в пределах каждого объекта.

При этом последние r_3 объектов могут быть разделены на $r_{31} + r_{32} + \dots + r_{3h} = r_3$ элементов с разными значениями векторов $p2_{\eta\eta}$ и $p4_{\eta\eta}$ ($p4_{\eta\eta}$ и $p4_{\eta\eta}$).

Общее выражение для комбинации МОК с МОО очень громоздко, что обусловлено значительным количеством сочетаний (C_r^2) на множестве каналов [9].

На примере МПЦ с двухканальными МПК с учетом формул (3), (6) и (8) оно имеет следующий вид:

$$\left. \begin{array}{l} \exists \Lambda_i \rightarrow \exists (a_{\eta}, a_i) \in A_i \\ \eta, \mu = \overline{1, m}, \eta \neq \mu \\ \vartheta = \overline{1, r} = [\overline{\Lambda_i}], i = \overline{1, n}, \end{array} \right\} \left\{ \begin{array}{l} \left[\left((a_{\eta} \rightarrow (a_{\eta})^{(1)}) \wedge (\exists ll_{\eta\eta} \rightarrow (ll_{\eta\eta})^0) \wedge (\gamma_{i\eta} \rightarrow (\gamma_{i\eta})^0) \right), \right. \\ \left. \left((a_{\eta} \rightarrow (a_{\eta})^3) \wedge (\exists ll_{\eta\mu} \rightarrow (ll_{\eta\mu})^3) \wedge (\gamma_{i\mu} \rightarrow (\gamma_{i\mu})^3) \right), \right. \\ \left[\left((a_i \rightarrow (a_i)^3) \wedge (\exists ll_{i\eta} \rightarrow (ll_{i\eta})^3) \wedge (\gamma_{i\eta} \rightarrow (\gamma_{i\eta})^3) \right), \right. \\ \left. \left((a_i \rightarrow (a_i)^1) \wedge (\exists ll_{i\mu} \rightarrow (ll_{i\mu})^0) \wedge (\gamma_{i\mu} \rightarrow (\gamma_{i\mu})^0) \right), \right. \\ \left((a_i \rightarrow (a_i)^1) \wedge (\exists ll_{i\eta} \rightarrow (ll_{i\eta})^0) \wedge (\gamma_{i\eta} \rightarrow (\gamma_{i\eta})^0) \right), \\ \left((a_i \rightarrow (a_i)^1) \wedge (\exists ll_{i\mu} \rightarrow (ll_{i\mu})^0) \wedge (\gamma_{i\mu} \rightarrow (\gamma_{i\mu})^0) \right), \\ \left. \left((a_i \rightarrow (a_i)^3) \wedge (\exists ll_{i\eta} \rightarrow (ll_{i\eta})^3) \wedge (\gamma_{i\eta} \rightarrow (\gamma_{i\eta})^3) \right), \right. \\ \left. \left((a_i \rightarrow (a_i)^3) \wedge (\exists ll_{i\mu} \rightarrow (ll_{i\mu})^3) \wedge (\gamma_{i\mu} \rightarrow (\gamma_{i\mu})^3) \right). \right. \end{array} \right. \quad (9)$$

Фигурная скобка в формуле (9) определяет конъюнкцию, а квадратная – дизъюнкцию строк, которые ими объединены. Техническая реализация МОК с МОО аналогична реализации МОО в отдельности. На примере первого сервера ПОЛЗ системы МПЦ-С, согласно формулам (8) и (9), соответствующий фрагмент конфигурации для светофоров приведен в табл. 5.

В табл. 5 предполагается, что из $r = r_1 + r_2 + r_3 = 38$ светофоров:

– $r_1 = 35$ моделируются имитационно (M7 – M2R);

– $r_2 = 1$ – физически по обоим каналам (M5);

– $r_3 = 2$ – имитационно и физически по разным каналам (M1, M3).

Среди $r_3 = r_{31} + r_{32}$ светофоров $r_{31} = 1$ моделируется имитационно по первому каналу и физически по второму (M1), а $r_{32} = 1$ – наоборот (M3).

Принято, что все светофоры – маневровые, поэтому литер «M2R» последнему светофору присвоен по тем же принципам, что и последней стрелке в табл. 3.

Таким образом, составление тестовой и рабочей версий ПО при МОК идентично МОГ или МОО в зависимости от того, в комбинации с каким из них выступает МОК. Этим определяются характерные

МОГ или МОО преимущества и недостатки для МОК. Возможна комбинация трех методов, при этом для множества Λ должны выполняться оба условия (7) и (9).

ТАБЛИЦА 5 Фрагмент конфигурации при комбинации методов обособленных каналов и объектов

Секции	Фрагмент конфигурационного файла		Примечание
Интерфейсные линии	1к	<DriverCAN Number="3" Type="OBJECT_MK" Name="/net/DSP1/dev/CAN1680SM1" Num_Controllers="36" sort="1" Time="250" ProtNameState="D3" ProtNameCtrl="C3"/>	Драйвер $(\gamma_{21}(\alpha))^{(0)}$
	2к	<DriverCAN Number="4" Type="OBJECT_MK" Name="/net/DSP1/dev/CAN1680SM2" Num_Controllers="36" sort="1" Time="250" ProtNameState="D3" ProtNameCtrl="C4"/>	Драйвер $(\gamma_{22}(\alpha))^{(0)}$
	1к	<DriverCAN Number="12" Type="OBJECT_MK" Name="/dev/CAN1680SM1" Num_Controllers="2" sort="1" Time="250" ProtNameState="D3" ProtNameCtrl="C3"/>	Драйвер $(\gamma_{21}(\alpha))^{(3)}$
	2к	<DriverCAN Number="13" Type="OBJECT_MK" Name="/dev/CAN1680SM2" Num_Controllers="2" sort="1" Time="250" ProtNameState="D4" ProtNameCtrl="C4"/>	Драйвер $(\gamma_{22}(\alpha))^{(3)}$
Св. М1	1к	<Controller CSID="AF" type="2" CAN="3"/>	CAN-линия № 3
	2к	<Controller CSID="B0" type="2" CAN="13"/>	CAN-линия № 13
Св. М3	1к	<Controller CSID="45" type="2" CAN="12"/>	CAN-линия № 12
	2к	<Controller CSID="46" type="2" CAN="4"/>	CAN-линия № 4
Св. М5	1к	<Controller CSID="47" type="2" CAN="12"/>	CAN-линия № 12
	2к	<Controller CSID="48" type="2" CAN="13"/>	CAN-линия № 13
Св. М7	1к	<Controller CSID="41" type="2" CAN="3"/>	CAN-линия № 3
	2к	<Controller CSID="42" type="2" CAN="4"/>	CAN-линия № 4
Св. ...	1к
	2к
Св. М2R*	1к	<Controller CSID="41" type="2" CAN="3"/>	CAN-линия № 3
	2к	<Controller CSID="42" type="2" CAN="4"/>	CAN-линия № 4

R* – количество светофоров четной горловины станции.

4 Использование методов имитационных и комбинированных испытаний на этапе пусконаладочных работ и в процессе эксплуатации

Дополнительным применением предлагаемых методов имитационных испытаний и МКИ может быть использование их на этапе пусконаладочных работ. С их помощью может быть выполнена отладка необходимых программно-аппаратных

компонентов всех уровней системы МПЦ без препятствий для осуществляемых поездных и маневровых перемещений, а также без необходимости выполнения масштабных операций на поле.

В частности, на базе МКИ может быть локализован отлаживаемый узел (схема управления отдельной стрелкой, светофором и т. д.) без использования других устройств, при условии, что необходимый набор состояний объектов, с которыми

данный узел связан логически, воспроизводится СИМ. На основе метода имитационных испытаний может быть выполнена отладка устройств верхнего и среднего уровней МПЦ, когда устройства нижнего не подключены либо имеются основания полагать некорректность их функционирования.

Учитывая, что испытания, проводимые в лабораторных условиях, не могут в полной мере спрогнозировать поведение системы, необходима ее техническая диагностика в процессе эксплуатации. Для периодического контроля ее работы возможно прогнозирование технического состояния МПК путем моделирования предотказных состояний, которое состоит в имитации опасных отказов в их отдельных каналах на базе МОК. С его использованием подтверждается своевременное предупреждение опасного отказа МПК в целом. В то же время на базе метода имитационных испытаний можно локализовать поиск повреждений в процессе эксплуатации (установлена причастность или непричастность устройств нижнего уровня).

5 Практическое применение разработанных методов

Предложенные методы активно применяются при разработке, сертификации и внедрении микропроцессорных систем МПЦ-С разработки ООО «НПП «САТЭП», которые эксплуатируются на ряде станций промышленного железнодорожного транспорта. Результаты эксплуа-

тационных испытаний на объектах внедрения, а также сбор данных в процессе эксплуатации подтверждают адекватность предлагаемых методов и закладываемых в них моделей. Внешний вид КИК МПЦ и окна испытательной модели (СИМ и комбинированной), примененных во время сертификации МПЦ-С (начало 2013 г.), приведены рис. 3, [5], [13].

Аппаратура нижнего и среднего уровней расположена в специализированных шкафах в том виде, в котором она находится в эксплуатации (рис. 4, а). Устройства верхнего уровня располагаются отдельно и взаимодействуют с ПОЛЗ по сети *Ethernet* (рис. 4, б).

При воздействиях на систему МПЦ-С устанавливались различные исходные состояния согласно ПМИ. В частном случае, во время подачи микросекундных импульсных помех (МИП), в цепи CAN-интерфейса одним из исходных состояний было открытое состояние светофора, который не должен перекрываться при воздействии МИП (рис. 4, в). При этом состояния других объектов, с которыми светофор логически связан, воспроизводились с использованием МКИ (МОГ+МОО+МОК).

Таким образом, были обеспечены все необходимые состояния объектов, с которыми система представителей МПК связана логическими зависимостями. Соответствующий фрагмент мнемосхемы станции приведен на рис. 5.

а) б)

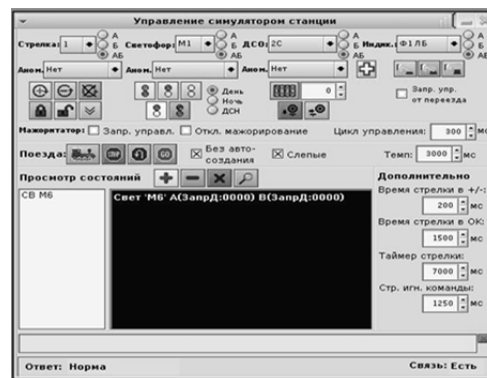


Рис. 3 Внешний вид испытательного комплекса и окна испытательной модели МПЦ-С

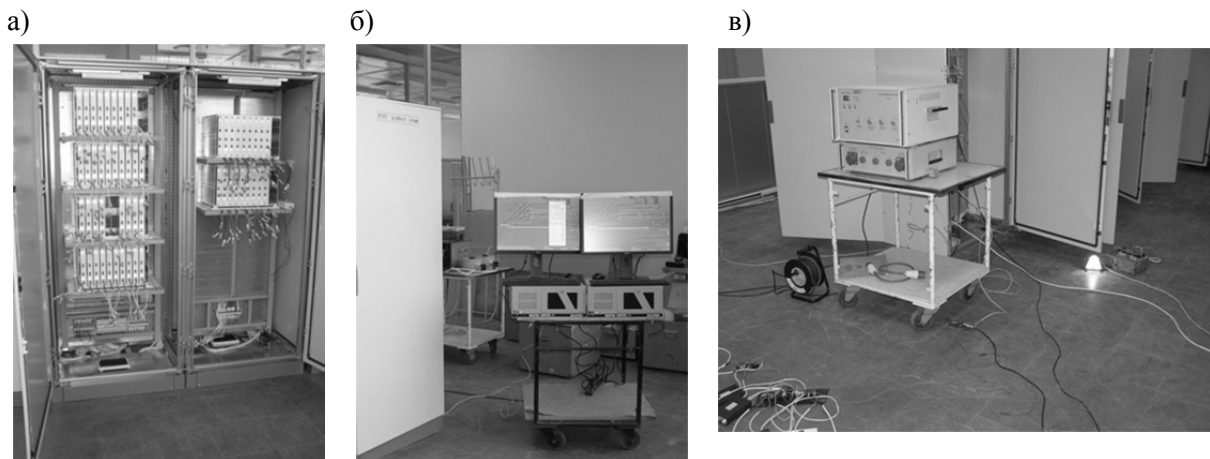


Рис. 4 Технические средства испытательного комплекса системы МПЦ-С:
а) шкаф контроллеров; б) верхний уровень; в) генератор помех и макет светофора

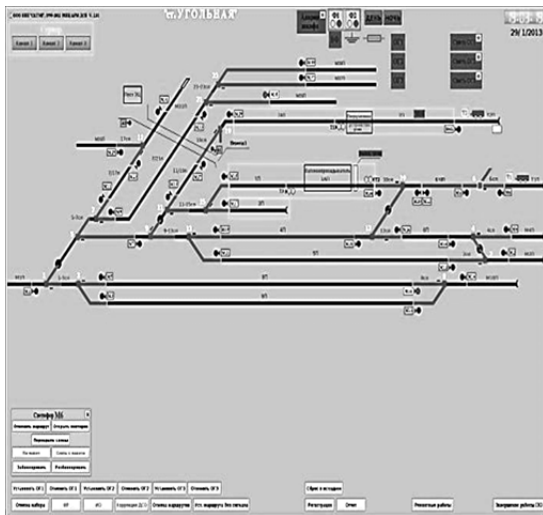


Рис. 5. Видеокادر контроля состояний системы МПЦ-С при испытаниях на безопасность и электромагнитную совместимость

Отдельным этапом проведены испытания с использованием МКИ при исходном защитном состоянии когда светофор М6 перекрыт, стрелка 6 потеряла контроль, участки 6/10П и 6СП ложно заняты. При этом во время испытаний подтверждалось отсутствие перехода соответствующих объектов в опасные состояния. Применение МКИ позволило в приведенном случае использовать одну систему представителей МПК для всех групп при полном наборе необходимых технологических ситуаций и экспериментальных воздействий.

6 Заключение

Предложенные методы испытаний позволяют повысить эффективность экспериментальной методологии доказательства безопасности систем МПЦ за счет увеличения тестового покрытия и сокращения материальных ресурсов. Наиболее результативным с точки зрения воспроизведения работы системы является применение комбинации методов обособленных групп, объектов и каналов в разных сочетаниях. На примере системы МПЦ-С установлено, что в этом случае достоверность испытаний превышает 98 % с экономией количества объектных контроллеров по сравнению с традиционными стендовыми испытаниями в 3,6 раза (при общем тестовом покрытии). Развитие возможностей использования данных разработок на этапе пусконаладочных работ и в процессе эксплуатации – перспективное направление дальнейших исследований в этой области.

Библиографический список

1. Сапожников В. В. Сертификация и доказательство безопасности систем железнодорожной автоматики / В. В. Сапожников, Вл. В. Сапожников, В. И. Талалаев. – М. : Транспорт, 1997. – 288 с. – ISBN 5-277-02000-4.
2. ОСТ 32.41–95. Безопасность железнодорожной автоматики и телемеханики. Методы доказательства безопасности систем и

устройств железнодорожной автоматики и телемеханики. – М., 1995 – 20 с.

3. *Ургансков Д. И.* Методы обеспечения и средства доказательства безопасности микропроцессорных систем железнодорожной автоматики и телемеханики: дис. ... канд. техн. наук: 05.22.08 / Д. И. Ургансков. – СПб. : Петербургский гос. ун-т путей сообщения, 2003. – 219 с.

4. *Кустов В. Ф.* Методика проведения испытаний на модели для доказательства безопасности технического средства системы управления и регулирования движением поездов / В. Ф. Кустов, К. С. Клименко // Испытания систем железнодорожной автоматики и телемеханики на безопасность и электромагнитную совместимость : труды междунар. семинара. – Гомель : БелГУТ, 2001. – С. 112–117. – ISBN 985-6550-64-5.

5. *Каменев А. Ю.* Особенности применения экспериментальных методов доказательства безопасности систем микропроцессорной централизации стрелок и сигналов / А. Ю. Каменев // Информационно-управляющие системы на железнодорожном транспорте. – 2011. – № 4. – С. 104–111. – ISSN 1681-4886.

6. *Каменев А. Ю.* Синтез методов испытаний на имитационных и физических моделях программируемых технических средств управления движением поездов / А. Ю. Каменев // Информационно-управляющие системы на железнодорожном транспорте : материалы докл. 24-й Международной научно-практической конференции, г. Алушта. – 2011. – № 5. – С. 128. – ISSN 1681-4886.

7. *Кустов В. Ф.* Формализация технических средств управления и контроля при лабораторных исследованиях : сб. науч. трудов

УкрГАЖТ / В. Ф. Кустов, А. Ю. Каменев. – 2012. – Вып. 134. – С. 156–162. – ISSN 1994-7852.

8. *Каменев А. Ю.* Математические модели для синтеза средств испытаний станционных систем автоматики : сб. науч. трудов ДонИЖТ / А. Ю. Каменев – 2012. – Вып. 31. – С. 73–84. – ISSN 1993-5579.

9. *Сигорский В. П.* Математический аппарат инженера / В. П. Сигорский. Изд. 2-е, стереотип. – Киев : Техника, 1977. – 768 с. – ISBN 100-0-00003-572-1.

10. *Патент 77047.* Украина G05B 23/00. Комбинированный испытательный комплекс микропроцессорной централизации стрелок и сигналов / А. Ю. Каменев, В. Ф. Кустов. – № 201208749; заявл. 16.07.2012; опубл. 25.01.2013, Бюл. № 2. – 6 с.

11. *Estublier J., Leblang D., Hoek A. and other.* Impact of Software Engineering Research on the Practice of Software Configuration Management / ACM Transactions on Software Engineering and Methodology. – 2005. – Vol. 14, No. 4. – Pp. 1–48.

12. Требования к программному обеспечению устройств железнодорожной автоматики и телемеханики / Памятка ОСЖД Р-843 от 05.11.2004. – Варшава, 2004. – 16 с.

13. *ООО «НПП «САТЭП».* Системы и устройства [Электронный ресурс]. – Режим доступа: <http://www.satep.com.ua>. – Загл. с экрана. – (Дата обращения: 10.03.2013).

14. *Кокурин И. М.* Эксплуатационные основы устройств железнодорожной автоматики и телемеханики / И. М. Кокурин, Л. Ф. Кондратенко. – М. : Транспорт, 1989. – 184 с. – ISBN 5-277-00945-0.

На кафедре «Автоматика и телемеханика на железных дорогах» защитили диссертации 15 соискателей из других стран (4 – из Китая, 4 – из Польши, 3 – из Болгарии, 3 – с Кубы и 1 – из Узбекистана).