На правах рукописи

Лях Татьяна Викторовна

ДИНАМИЧЕСКАЯ ВЕРИФИКАЦИЯ ПРОЦЕСС-ОРИЕНТИРОВАННЫХ ПРОГРАММ УПРАВЛЕНИЯ КИБЕРФИЗИЧЕСКИМИ СИСТЕМАМИ

Специальность 05.13.18 – Математическое моделирование, численные методы и комплексы программ

А В Т О Р Е Ф Е Р А Т

диссертации на соискание ученой степени
кандидата технических наук

Новосибирск – 2020

Работа выполнена в Федеральном государственном бюджетном учреждении науки «Институт автоматики и электрометрии» Сибирского отделения Российской академии наук

Научный руководитель: доктор технических наук Зюбин В. Е.

Официальные оппоненты: *Окольнишников Виктор Васильевич,* д-р техн. наук, Федеральное государственное бюджетное научное учреждение «Федеральный исследовательский центр информационных и вычислительных технологий» (ФИЦ ИВТ), вед. науч. сотр. (г. Новосибирск).

Бессонов Алексей Владимирович, к.т.н, ООО БФТ (Бюджетные и Финансовые Технологии), главный программист (г. Москва)

Ведущая организация: Федеральное государственное бюджетное образовательное учреждение высшего образования «Сибирский государственный университет телекоммуникаций и информатики».

Защита состоится “28” декабря 2020 г. В 11 часов

на заседании диссертационного совета К 003.005.02 в Институте автоматики и электрометрии СО РАН по адресу:

630090, г. Новосибирск, пр. ак. Коптюга, 1

С диссертацией можно ознакомиться в библиотеке Института автоматики и электрометрии СО РАН.

Автореферат разослан “____”___2020 г.

Ученый секретарь диссертационного совета

д. ф.-м.н. Ильичев Л.В.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность работы и степень ее разработанности

На сегодняшний день киберфизические системы (КФС) с повышенными требованиями к надежности получили широкое распространение как в области промышленности, так и в области пользовательских устройств (ПЛК, промышленный интернет вещей, встраиваемые системы, системы управления и т.д.). Алгоритмы управления (АУ) КФС характеризуются неопределенной продолжительностью работы, взаимодействием с окружающей средой, зависимостью реакций от событий окружающей среды, необходимостью согласовывать реакцию алгоритма с динамическими характеристиками внешней среды и логическим параллелизмом (наличие независимых и слабо связанных потоков управления в алгоритме). Эта специфика приводит к появлению специализированных языковых средств и технологий программирования (языки МЭК 61131-3, MATHLAB (Simulink), G (NI LabVIEW), язык Дракон, модельно-ориентированное проектирование), в частности, развиваемых в ИАиЭ в виде процесс-ориентированного программирования (языки Reflex и IndustrialC). Процесс-ориентированное программирование базируется на концепции гиперпроцесса – множества взаимодействующих процессов с выполнимыми состояниями.

Программное обеспечение (ПО) современных КФС достигло такой сложности, что его верификация является отдельной областью исследования. Для верификации ПО развиваются методы динамической и так называемой формальной верификации. Применение формальных методов к верификации ПО КФС затруднено языковой гетерогенностью, большим объемом ручных операций, а также возрастанием сложности создаваемых моделей.

Наиболее распространенный подход – динамическая верификация ПО КФС. При этом преобладает ручная проверка на этапе пуско-наладки. Специалист, проводящий ручную проверку, по сути выступает в качестве имитатора объекта управления: сначала задает значения на входах управляющего ПО, а затем контролирует значения выходных сигналов. Этот способ в силу трудоемкости и сложности реализации позволяет провести верификацию только простейших свойств ПО.

Свойство открытости означает, что верификация свойств ПО КФС требует наличия объекта управления. С другой стороны, исследование ПО на реальном объекте управления может привести к критической ситуации (поломке оборудования, аварии и т. п.).

Современная тенденция для контроля качества управляющих программ предлагает интеграцию методов динамической верификации в итерационный процесс разработки ПО КФС (разработка, управляемая моделями – MDD,

модельно-ориентированное проектирование – MBD, разработка через тестирование – TDD). Активно развиваются подходы к верификации ПО КФС, в которых различные типы тестирования (MBT, системное, модульное, регрессионное, интеграционное, пассивное, Back-to-Back) сочетается с использованием физических или программных моделей объекта управления (Е. Брингман, Р. Изерман, В. В. Кулямин). Другим перспективным подходом к динамической верификации ПО КФС является метод мониторинга поведения системы в процессе ее штатной работы (runtime verification) (Э. Барточчи). Однако он не позволяет исследовать поведение ПО при различных условиях.

На сегодняшний день слабо изучены модели и методы динамической верификации процесс-ориентированных программ управления КФС. Отсутствие контроля качества создаваемого процесс-ориентированного ПО серьезно увеличивает риски проекта при разработке киберфизических систем. Таким образом, исследование и разработка моделей и методов динамической верификации процесс-ориентированного ПО КФС, объединяющих методы тестирования, мониторинга и моделирования, является актуальной проблемой.

Объект исследования – программное обеспечение КФС.

Предмет исследования – модели и методы динамической верификации процесс-ориентированных программ управления КФС.

Цель работы – разработка методов и моделей динамической верификации процесс-ориентированных программ управления КФС.

В соответствии с поставленной целью в работе решаются **следующие основные задачи**:

1. Исследовать существующие подходы к динамической верификации программ управления КФС, сформулировать требования к разрабатываемым моделям, вычислительным методам и комплексам программ динамической верификации процесс-ориентированных программ управления КФС.
2. Предложить формальную модель динамической верификации процесс-ориентированных программ управления КФС.
3. Определить численный метод количественной оценки меры соответствия программ управления КФС требованиям.
4. Разработать программный комплекс для динамической верификации процесс-ориентированных программ управления КФС на основании предложенных методов и моделей.
5. Практически апробировать разработанный подход динамической верификации программ управления КФС в области промышленной автоматизации на реальных технологических объектах.

Основная гипотеза. Программы управления КФС, специфицируемые процесс-ориентированными средствами, образуют отдельный класс программ,

динамическая верификация которых на программном имитаторе окружающей среды позволит повысить качество разрабатываемых систем.

Научная новизна. В диссертационной работе предложен, разработаны и исследованы модели и методы динамической верификации процесс-ориентированных программ управления КФС, которые использует средства процесс-ориентированного программирования и концепцию виртуальных объектов.

Принципиальный вклад в развитие технологии процесс-ориентированного программирования вносят следующие научные результаты, полученные автором:

1. Предложена четырехкомпонентная формальная модель системы динамической верификации процесс-ориентированных программ управления КФС, состоящая из верифицируемого алгоритма управления, модели объекта управления, диспетчера тестовых сценариев и блока верификации. Отличительная особенность модели в том, что компоненты специфицируется как гиперпроцессы, имеющие выделенные каналы сообщений для синхронизации и совместного функционирования по верификации управляющих программ. При этом модель предполагает, что конечный итог верификации управляющей программы представлен вектором результатов исполнения тестовых сценариев.

2. Разработан численный метод определения вектора результатов исполнения тестовых сценариев, специфицированный для процесс-ориентированных ПО. Отличительная особенность метода заключается в последовательной активации гиперпроцессов “диспетчер”, “объект управления”, “алгоритм управления” и “верификатор”. При этом гиперпроцесс “верификатор” на основании состояний своих процессов вычисляет результат исполнения тестового сценария как вектор вещественных чисел в диапазоне [0,1].

3. Разработана архитектура программного комплекса динамической верификации на базе среды LabVIEW, который состоит из компонентов “объект управления” и “алгоритм управления”, специфицированных на языке Reflex, и ядра, предоставляющего пользовательский интерфейс и функционирующего по алгоритму, задаваемому кодом “верификатора” и “диспетчера”.

Основные положения, выносимые на защиту:

1. Формальная модель динамической верификации процесс-ориентированных программ управления КФС, которая включает четыре взаимодействующих расширенных гиперпроцесса, специфицирующих верифицируемый алгоритм управления, виртуальный объект управления, диспетчер управления тестовыми сценариями и блок верификации, позволяет оценить соответствие проверяемой процесс-ориентированной программы наложенным на нее требованиям и учитывает специфику управляющих процесс-ориентированных программ КФС.

2. Численный метод определения вектора результатов выполнения тестовых сценариев, в котором блок «диспетчер» управляет порядком прохождения тестов, позволяет с помощью блока «верификатор» оценить меру соответствия проверяемой процесс-ориентированной управляющей программы КФС требованиям в виде вектора результатов тестовых сценариев.

3. Архитектура программного комплекса динамической верификации на базе среды LabVIEW, которая состоит из компонентов “объект управления” и “алгоритм управления”, специфицированных на языке Reflex, и ядра, позволяет бесшовно интегрировать модули на языке Reflex и проводить динамическую верификацию управляющих программ на языке Reflex.

Теоретическая и практическая значимость результатов исследования. Разработанные метод и модель динамической верификации процесс-ориентированных программ управления КФС упрощают процесс разработки, верификации и сопровождения управляющего ПО на протяжении жизненного цикла системы управления КФС. Применительно к области промышленной автоматизации, получаемые преимущества дают возможность снизить сроки разработки, сократить сроки пуско-наладочных работ и повысить качество создаваемого программного обеспечения. Это имеет особое значение при автоматизации технологических процессов и научных исследований, а также при отработке и внедрении новых наукоемких технологий.

Решение было практически апробировано на задаче автоматизации Большого солнечного вакуумного телескопа (пос. Листвянка, Иркутская обл.). В проекте был верифицирован алгоритм управления подсистемой вакуумирования. Также решение было практически апробировано при разработке виртуальных лабораторных стендов, используемых для обучения студентов технических специальностей (ФИТ НГУ) разработке программ управления КФС. Разработанные механизмы бесшовной интеграции алгоритмических блоков, описанных на языке Reflex, в среду LabVIEW были апробированы на задаче автоматизации углоизмерительной машины НОНИУС.

Документальные подтверждения эффективности полученных результатов при практическом использовании приведены в Приложении к диссертации.

Методология и методы исследования. Задачи, поставленные в работе, решались с использованием процесс-ориентированного подхода к разработке программ управления КФС. Подходы к верификации программ управления КФС были выявлены на основе анализа доступных научных публикаций русских и зарубежных авторов. Эмпирическим путем была доказана эффективность разработанного подхода в задачах автоматизации технологических процессов и физико-технических исследований.

Личный вклад автора. Выносимые на защиту результаты получены при непосредственном участии соискателя. В опубликованных работах участие автора заключалось в написании текстов работ, самостоятельном проведении исследовательских работ и непосредственном участии в разработке концептуального подхода к верификации программ управления КФС. Также автор самостоятельно разработала архитектуру комплексов автоматизированной и автоматической верификации, реализовала ПО комплексов верификации и виртуальных лабораторных стендов и подобрала тестовые задачи для комплекса виртуальных лабораторных стендов.

Внедрение полученных результатов. Полученные результаты были использованы в работах по созданию автоматизированных цифровых комплексов:

1) разработка и тестирование управляющего ПО вакуумной подсистемой БСВТ;
2) разработка и тестирование управляющего ПО углоизмерительной машины НОНИУС.

Апробация работы. Результаты диссертации докладывались на международных и всероссийских конференциях, в том числе:

- Индустриальные информационные системы – 2013 (г. Новосибирск, Россия, 24-28 сентября 2013 г.);
- Всероссийская научно-техническая конференция «Современные проблемы радиоэлектроники» (г. Красноярск, Россия, 6-8 мая 2014 г.);
- Девятая международная Ершовская конференция PSI-2014 (г. Санкт-Петербург, Россия, 24-27 июня 2014 г.);
- 17th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (Эрлагол, Алтай, Россия, 30 июня – 4 июля, 2016 г.);
- V международная конференция «Математическое и компьютерное моделирование» (г. Омск, Россия, 1 декабря 2017 г.);
- International Siberian Conference on Control and Communications (SIBCON-2019) (г. Томск, Россия, 18-20 апреля 2019 г.);
- International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON 2019) (г. Новосибирск, Россия, 21-22 октября 2019 г.).

Публикации. По теме работы опубликовано 15 работ, из которых 5 статей в рецензируемых журналах из Перечня ВАК и 5 публикаций в трудах и материалах международных конференций. 4 работы опубликованы в рецензируемых научных журналах, индексируемых в международной базе Scopus и 2 работы опубликованы в Web of Science. Получено 2 свидетельства о государственной регистрации программ для ЭВМ.

Объем и структура работы. Диссертация состоит из введения, четырех глав, заключения и приложения. Объем работы – 133 страницы основного текста, содержит 17 рисунков и 3 таблицы. Список литературы включает в себя 106 наименований.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении приводится обоснование актуальности выбранной темы, формулируется цель и задачи работы, представлены основные положения, выносимые на защиту, научная и практическая значимость работы и приведена информация о внедрении полученных результатов.

В первой главе обсуждается специфика алгоритмов управления КФС. Приводится сравнение программных языковых средств, использующихся при описании управляющих программ КФС. Проводится критический анализ существующих подходов к верификации программного обеспечения КФС. Приводится критический анализ математических моделей, используемых для описания КФС. Сформулированы требования к разрабатываемым моделям, вычислительным методам и комплексам программ динамической верификации процесс-ориентированных программ управления КФС.

Алгоритмы управления КФС обладают рядом специфических свойств.

1) открытость – алгоритм управления взаимодействует с окружающей средой;
2) событийность – алгоритм реагирует на события окружающей среды и воздействует на происходящие в окружении физические процессы;
3) продолжительность функционирования управляющего алгоритма не определена;
4) синхронизм – реакции управляющего алгоритм должны синхронизоваться с событиями окружающей среды;
5) логический параллелизм – структура управляющего алгоритма отражает параллелизм процессов, происходящих в окружающей среде.

Реализация алгоритмов управления средствами языков общего назначения чревато чрезмерным усложнением программной архитектуры при росте сложности алгоритма. Поэтому в области КФС развиваются специализированные языковые средства для разработки управляющего ПО. Например, в области промышленной автоматизации используются языки МЭК 61131-3, MATLAB (Simulink), G (NI LabVIEW), Reflex. Исследователи альтернативных лингвистических средств для описания управляющих программ КФС предлагают и практически обосновывают эффективность предметно-ориентированных языков на основе модели конечного

автомата. К таким языкам относятся процесс-ориентированные языки Reflex и Industrial C, основанные на математической модели гиперпроцесса.

При использовании процесс-ориентированных языков при разработке управляющих программ КФС отдельную задачу представляет доказательство корректности разрабатываемого ПО. Верификация является одним из основных методов обеспечения корректности ПО. При разработке процесс-ориентированного ПО методы, использующиеся для верификации программного обеспечения общего назначения, слабо применимы. Поэтому наиболее распространенный подход – ручная отладка на этапе пуско-наладки: проверяющий контролирует реакцию алгоритма на различные ситуации, постепенно усложняя тесты. Подход очень трудоемкий. Он приводит к серьезным психологическим нагрузкам на разработчиков, не гарантирует полноту верификации, затрудняет контроль качества верификации и, в конечном итоге, усложняет разработку управляющего ПО.

Поэтому разработка моделей и методов верификации процесс-ориентированных алгоритмов управления интересна не только с теоретической, но и с практической стороны. Методы, разрабатываемые в области формальной верификации КФС, на текущем этапе сопряжены с серьезными ограничениями на сложность верифицируемых алгоритмов, которые делают эти методы трудно применимыми для практических задач.

Современная тенденция заключается в интеграции методов динамической верификации управляющих программ КФС в итерационный процесс разработки ПО с использованием программных имитаторов окружающей среды. Одним из ключевых преимуществ при использовании программных имитаторов является возможность регулировать физические параметры окружающей среды, а также имитировать ситуации, которые было бы невозможно или опасно проверять на реальном объекте (к примеру, аварии). Синхронизация программ управления КФС с физическими событиями окружающей среды может привести к увеличению сроков разработки при верификации на реальном объекте. При динамической верификации использование программного имитатора позволяет минимизировать эти задержки.

Программы управления КФС реагируют как на события физической составляющей КФС, так и на воздействия от оператора. Поэтому при динамической верификации свойств ПО КФС требуется моделировать:

1) поведение физической составляющей КФС;
2) поведение алгоритма управления КФС;
3) поведение оператора КФС;
4) планирование сценариев работы на объекте управления;

5) мониторинг реакций алгоритма управления в контексте текущего сценария.

В Институте автоматики и электрометрии была разработана общая схема верификации управляющих программ КФС (рис. 1), которая состоит из следующих шагов:

1) верифицируемый алгоритм управления (его часть) реализуется программно и оформляется в обособленный алгоритмический блок;

2) модель физической внешней среды КФС (ее части) реализуется программно и оформляется в обособленный алгоритмический блок, называемый виртуальный объект управления (ВОУ);

3) проводится верификация через создание тестовых ситуаций (сценариев) и контроль реакции алгоритма управления;

4) код алгоритма управления и (или) виртуального объекта управления корректируется по результатам верификации.

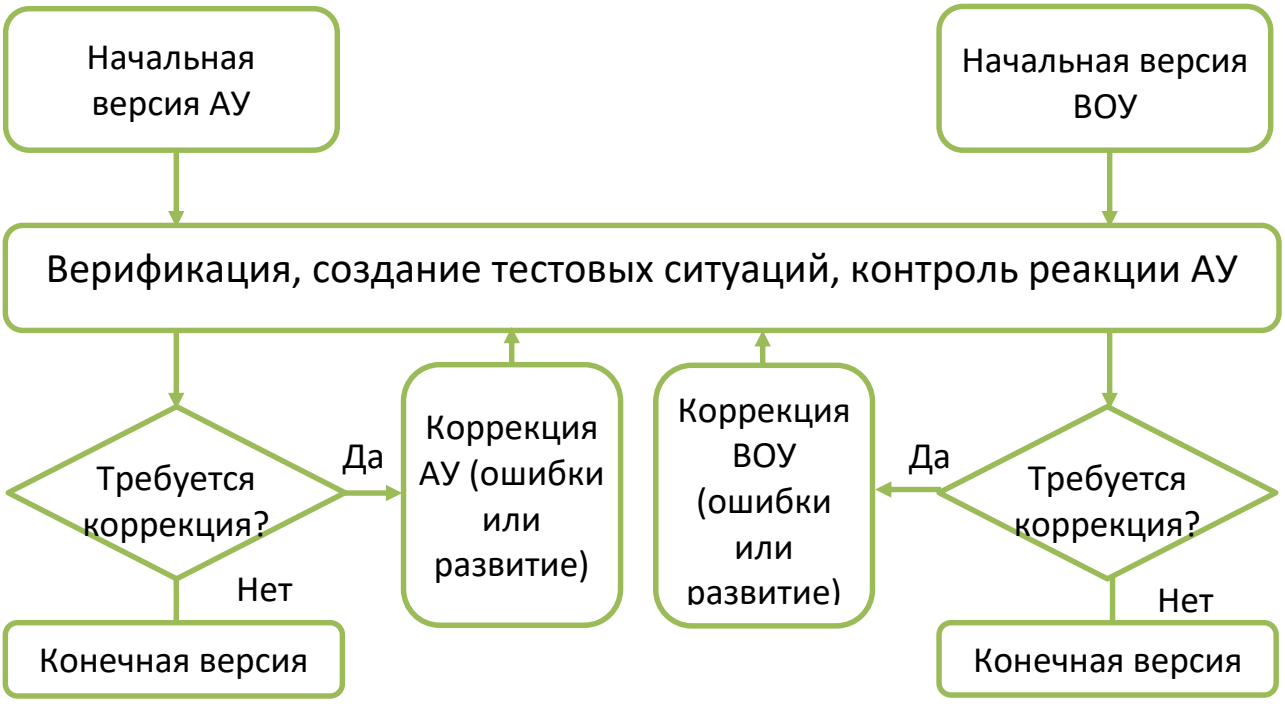

Рис. 1. Общая схема динамической верификации алгоритмов управления технологическим объектом: АУ – алгоритм управления, ВОУ – виртуальный объект управления.

Схема позволяет использовать итерационный подход к разработке промышленных алгоритмов управления КФС.

Исследования показывают, что автоматизация динамической верификации управляющего ПО КФС приводит к снижению затрат на проверку корректности ПО. Исследователи также отмечают, что языковая гетерогенность увеличивает нагрузку на разработчиков и затрудняет внедрение новых методов верификации.

Были сформулированы требования к разрабатываемому подходу верификации управляющих программ КФС:

1. Верификация программ управления КФС должна происходить динамически на программном имитаторе объекта управления.
2. Процесс верификации должен быть интегрирован в итерационную разработку ПО КФС.
3. Проверка корректности разрабатываемой программы управления КФС должна происходить автоматически.
4. Модель должна включать проверку корректности реакций алгоритма управления (АУ), имитацию поведения объекта управления и управление прохождением тестовых сценариев;
5. Метод должен обеспечивать возможность изменений масштаба времени;
6. При моделировании тестовых ситуаций должна предоставляться возможность изменения параметров внешней среды КФС и взаимодействия ПО с оператором;
7. Описание поведения АУ, имитатора объекта управления, а также мониторинга поведения АУ и управления тестовыми сценариями должно быть унифицированным.

Во второй главе описывается четырёхкомпонентная формальная модель динамической верификации процесс-ориентированных программ управления КФС с использованием программной модели объекта управления (ВОУ). Также описывается численный метод определения вектора результатов исполнения тестовых сценариев.

Для моделирования проверки корректности реакций алгоритма управления, имитации поведения объекта управления и управления прохождением тестовых сценариев использовалась математическая модель гиперпроцеса, которая была расширена типом данных «Очередь сообщений». Такой подход также позволит обеспечить унификацию описания АУ, ВОУ, мониторинга поведения и описание тестовых сценариев.

Четырехкомпонентная формальная модель динамической верификации программ управления КФС – это кортеж:

DV = (Controller, Plant, Dispatcher, Verifier, AR, TS, Qm, rez(ts))

DV включает четыре расширенных гиперпроцесса (рис. 2):

Controller (1) - гиперопроцесс, моделирующий алгоритм управления. Входные данные Controller – управляющие команды от оператора КФС (5), а также сигналы от окружающей среды (сигналы датчиков); выходные данные – управляющие воздействия на физическую компоненту КФС и диагностические сообщения для оператора (6).

Plant (2) – гиперопроцесс, моделирующий виртуальный объект управления (ВОУ). Входные данные Plant – управляющие воздействия Controller, а также конфигурационные сообщения, которые задают режимы работы объекта и его физические параметры (7). Выходные данные – значения сигналов датчиков;

Dispatcher (3) – гиперопроцесс, моделирующий управление сценариями тестирования. Выходные данные Dispatcher - управляющие сообщения, имитирующие сообщения от оператора КФС для Controller (5) и настроечные сообщения Plant (7). Dispatcher запускает и останавливает выполнение тестовых сценариев;

Verifier (4) – гиперопроцесс, обеспечивающий проверку реакций верифицируемого ПО в соответствии с требованиями. Входные данные – управляющие воздействия от Controller на Plant, значения сигналов датчиков, передаваемые от Plant в Controller, и диагностические сообщения для оператора от Controller (6). Выходные данные – отчет о результатах верификации (8).

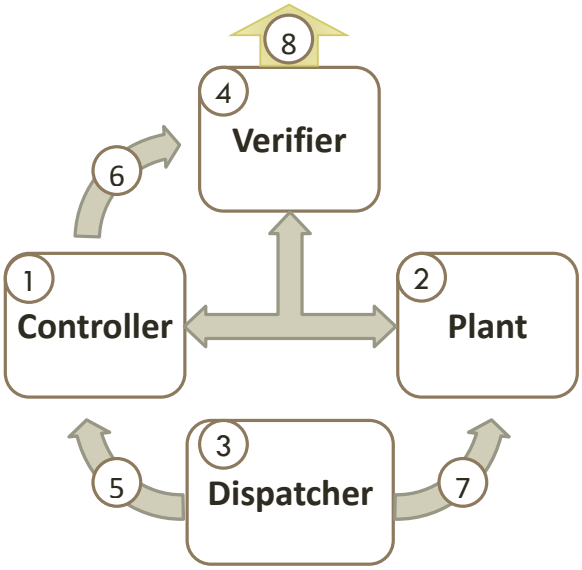

Рис. 2. Общая схема четырёхкомпонентной формальной модели динамической верификации программ управления КФС.

$\boldsymbol{AR \subseteq P_V}$ – множество требований к верифицируемому ПО, где $\boldsymbol{P_V}$ - все множество процессов Verifier**.**

A – множество весов требований, где каждый элемент $\boldsymbol{\alpha_i}$ этого множества $\boldsymbol{\alpha_i \in [0, 1]}$.

TS – множество тестовых сценариев.

Qm – вектор результатов исполнения тестовых сценариев.

res(ts) **–** функция, вычисляющая компоненту вектора Qm для тестового сценария ts.

Тестовый сценарий – это упорядоченная последовательность внешних команд для объекта управления и управляющего ПО, воздействующих на ПО

(например, команда от оператора), а также управляющих режимами работы объекта управления.

Каждый тестовый сценарий $\boldsymbol{ts} \subseteq \boldsymbol{TS}$ – это кортеж

$$\boldsymbol{ts} = \left(\boldsymbol{M_{D\to C}}, \boldsymbol{M_{D\to P}}, \boldsymbol{ar'_{TS}}\right),$$ где:

$\boldsymbol{M_{D\to C}}$ – кортеж сообщений, который получает Controller от Dispatcher при проверке конкретного сценария.

$\boldsymbol{M_{D\to P}}$ – кортеж сообщений, который получает Plant от Dispatcher при проверке конкретного сценария

$\boldsymbol{ar'_{TS}} \subseteq \boldsymbol{AR}$ – эти процессы моделируют проверяемые на текущем шаге воздействия на управляющее ПО и ВОУ. По запуску текущей тестовой ситуации эти процессы переходят из пассивной функции-состояния в начальную активную функцию.

Критерием успешности прохождения тестового сценария выступает функция ***res(ts),*** вычисляемая по завершению проверки каждого тестового сценария.

Вектор результатов исполнения тестовых сценариев определяется как:

$\boldsymbol{Qm = (res(ts_1), \ldots, res(ts_m))}$

Численный метод определения вектора результатов исполнения тестовых сценариев Qm представлен на рис. 3-5 и заключается в последовательной активации гиперопроцессов Dispatcher, Plant, Controller и Verifier (рис. 3). Dispatcher управляет порядком прохождения тестов через механизм обмена сообщениями, а блоки Plant, Controller и Verifier последовательно активируются, взаимодействуя через выделенную область данных.

Dispatcher (рис. 4). управляет запуском, остановкой и переключением тестовых сценариев, задавая или набор строго определенных сценариев работы, или реализуя систему генерации сценариев работы, или же совмещая обе эти роли. Когда Dispatcher исполняет тестовый сценарий ts, он последовательно отправляет сообщения $M_{D\to C}$ и $M_{D\to P}$, соответствующие этому сценарию, тем самым имитируя присутствие оператора и настраивая Plant. Передаются значения физических параметров для Plant и команды на переключение режимов его работы – например, если в тестовом сценарии исследуются реакции ПО при штатной работе или режиме неисправности.

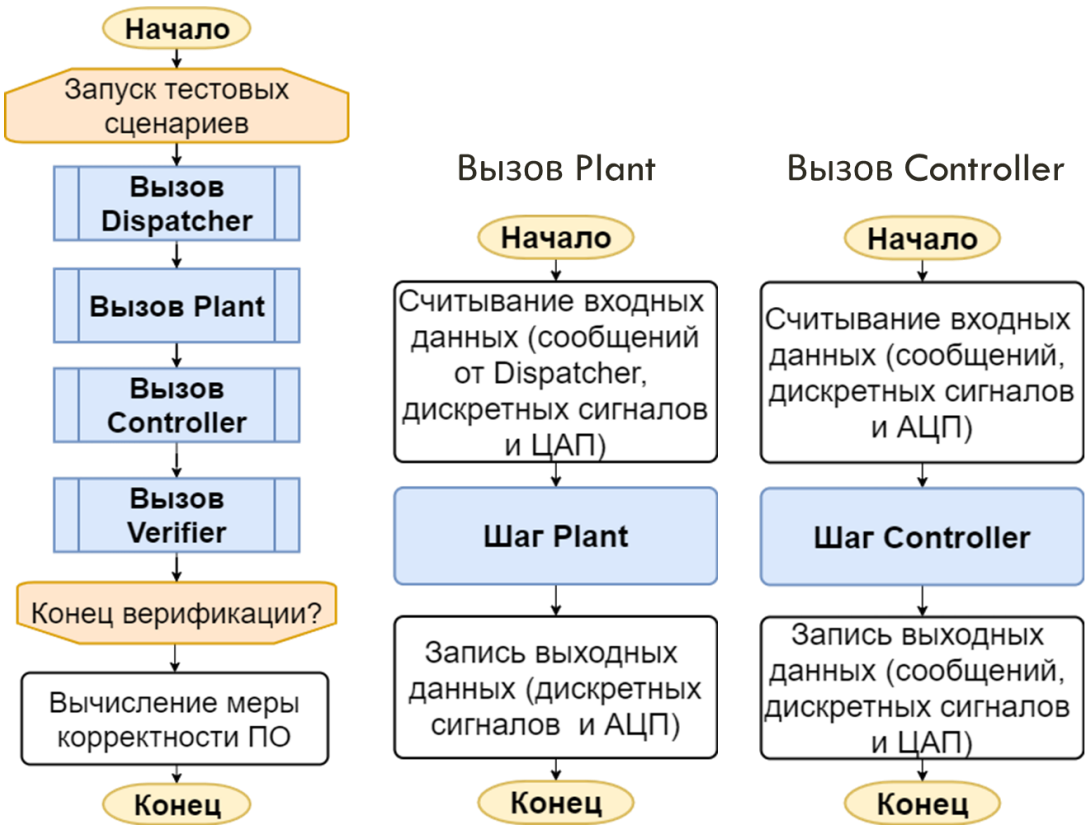

Рис. 3. Численный метод определения вектора результатов исполнения тестовых сценариев.

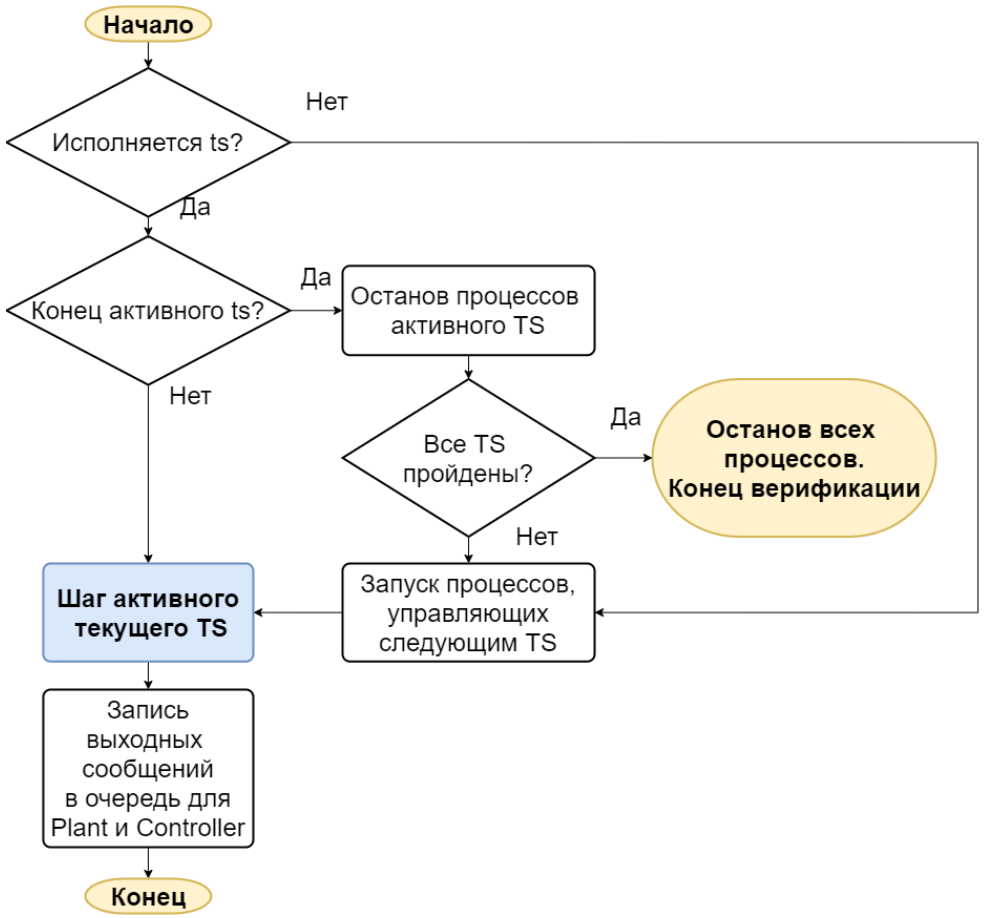

Рис. 4. Численный метод определения бинарного вектора результатов исполнения тестовых сценариев: шаг Dispatcher.

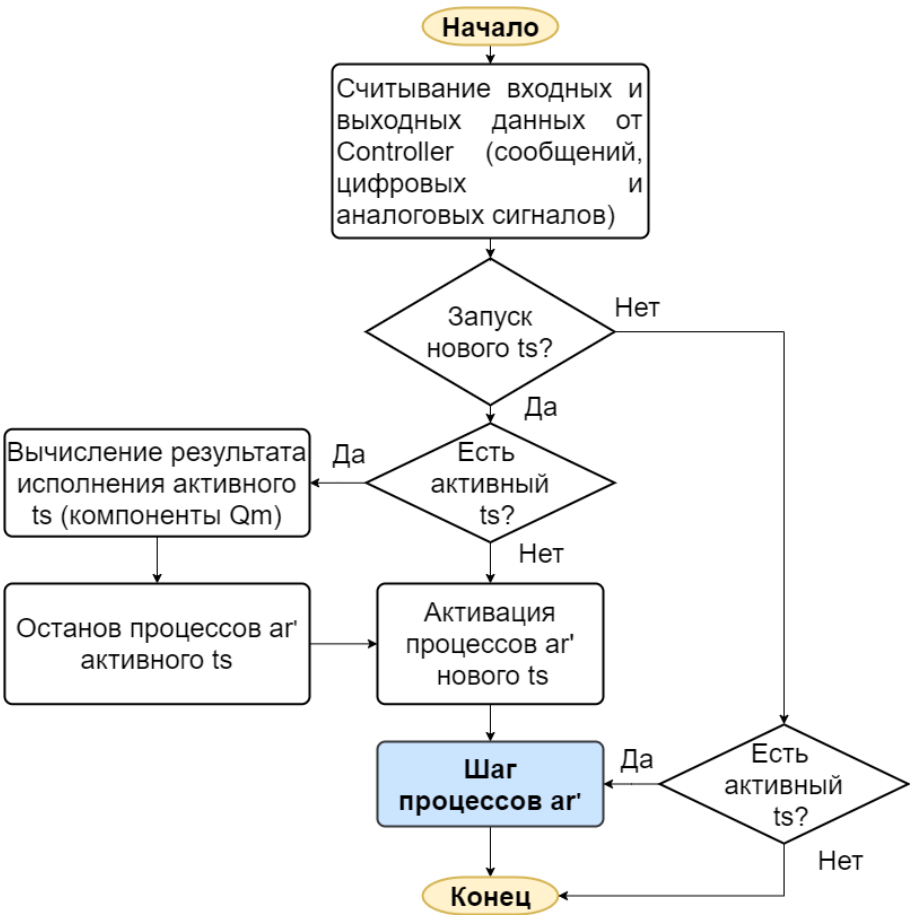

Рис. 5. Численный метод определения бинарного вектора результатов исполнения тестовых сценариев: шаг Verifier.

В процессе работы Controller генерирует последовательность управляющих воздействий на Plant, а также упорядоченную последовательность диагностических выходных сообщений для оператора КФС. Поведение управляющего ПО на текущем тестовом сценарии считается корректным, если Controller генерирует корректную последовательность воздействий и диагностических сообщений. В противном случае, алгоритм управления считается некорректным. Verifier (рис. 5) по срезу данных между Controller и Plant на каждом цикле активации DV определяет корректность реакций Controller. По завершению тестового сценария вычисляется результат исполнения тестового сценария:

$$res(ts) = \frac{\sum_i a_i * (curr\_state(p_i) != s^{ERROR})}{\sum_i a_i}, \text{ где}$$

$\boldsymbol{p_n} \in \boldsymbol{AR}$ $\forall$ n $\in$ [1, R], R – мощность множества $\boldsymbol{ar'_{TC}}$

curr_state(proc) – функция, которая возвращает текущее состояние от процесса proc;

$\boldsymbol{s^{ERROR}}$ – выделенная пассивная функция-состояние «останов по ошибке».

$\boldsymbol{\alpha_i} \in$ [0, 1] – вес каждого требования.

В случае соблюдения всех требований тестового сценария rez(ts) равен 1. Вектор Qm = (res (ts_1) , … , res(ts_m)) формируется на основании результатов исполнения тестовых сценариев.

В третьей главе описана архитектура программных комплексов автоматизированной и автоматической верификации программ управления КФС на языке Reflex в виде LabVIEW-приложения. Приведены результаты апробации предлагаемого подхода динамической верификации процесс-ориентированных программ управления КФС.

Предложенный метод динамической верификации процесс-ориентированных программ управления КФС был реализован в двух вариантах: в комплексе автоматизированной верификации и в комплексе автоматической верификации управляющих программ на языке Reflex.

В комплексе автоматизированной верификации программ управления КФС (рис.6) управление сценариями работы и контроль реакции алгоритма производится оператором через графический интерфейс (1). Графический интерфейс предоставляет оператору возможность отправлять штатные команды алгоритму управления (6) через очередь сообщений (2) и контролировать сообщения от управляющего ПО через очередь сообщений (4), управлять поведением Plant (9) через очередь сообщений (3) и контролировать сообщения от Plant через очередь сообщений (5). Также на GUI отображаются состояния входных (7) и выходных (8) дискретных сигналов алгоритма управления. Для имитации входных аналоговых сигналов от датчиков/АЦП, организован дополнительный канал связи между Plant и Controller (10), а для имитации выходных аналоговых сигналов (ЦАП) – канал связи (11).

Недостатки подхода:

1) визуальный контроль за реакцией ПО и сложность анализа отображаемой информации приводит к ошибкам верификации (вероятность не заметить ошибки в работе программы);

2) на каждой итерации должна быть заново проверена реакция ПО в соответствии со списком тестовых ситуаций, что означает большое количество рутинных действий и также приводит к ошибкам (пропуску тестовых ситуаций).

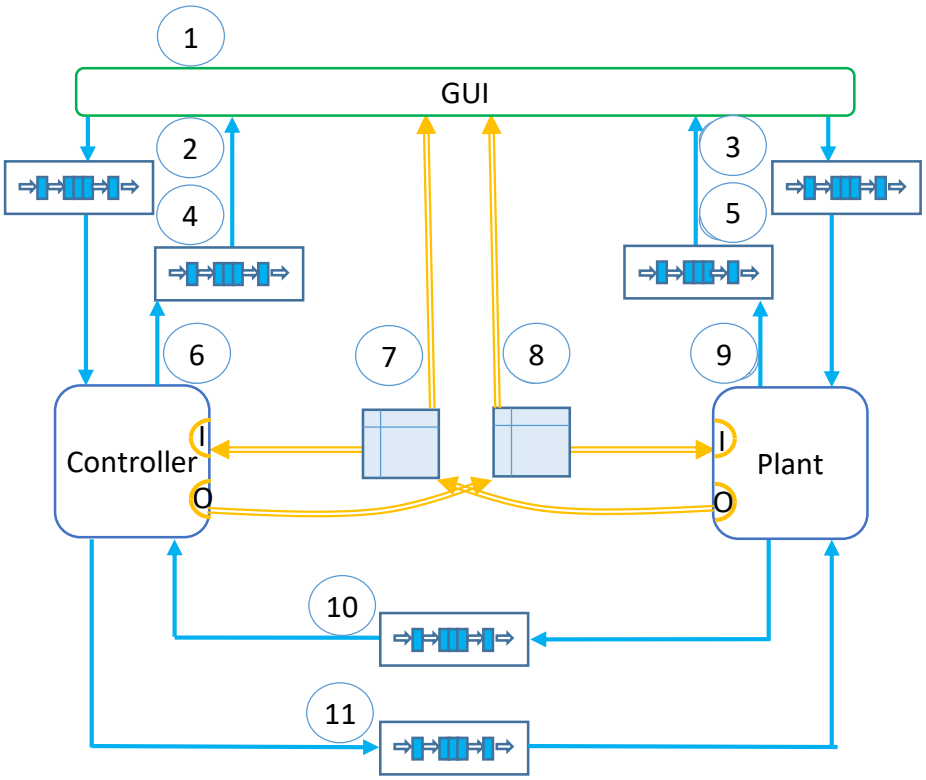

Рис. 6. Архитектура комплекса автоматизированной динамической верификации управляющих программ КФС на языке Reflex.

Для устранения недостатков автоматизированной верификации был разработан комплекс автоматической верификации управляющих ПО (рис. 7). В отличие от ручной верификации управление сценариями работы производится алгоритмическим блоком управления сценариями (Dispatcher) (9), а контроль реакции управляющей программы – алгоритмическим блоком верификации (Verifier) (2). Dispatcher через штатную очередь сообщений (8) посылает команды для Controller (4), имитируя действия оператора, а также управляет поведением Plant (7) через очередь сообщений (10). При этом Plant эмулирует как штатную работу объекта управления, так и отказы его элементов. Параллельно Dispatcher передает информацию о запускаемых сценариях в Verifier через очередь сообщений (3). В свою очередь Verifier определяет корректность Controller на основании информации о запускаемых сценариях, сообщений от Controller, передаваемых через буфер сообщений (1) и состояния буферов входных/выходных сигналов Controller (5, 6).

Автоматическая верификации построена на взаимодействии управляющей программы, виртуального объекта управления (компьютерной модели), блока управления сценариями работы и блока верификации. Управление сценариями работы, имитация действий оператора системы управления и контроль реакции алгоритма производятся автоматически. При этом эмулируется как штатная работа объекта управления, так и отказы его элементов. Автоматическая схема верификации исключает ошибки оператора, проводящего верификацию.

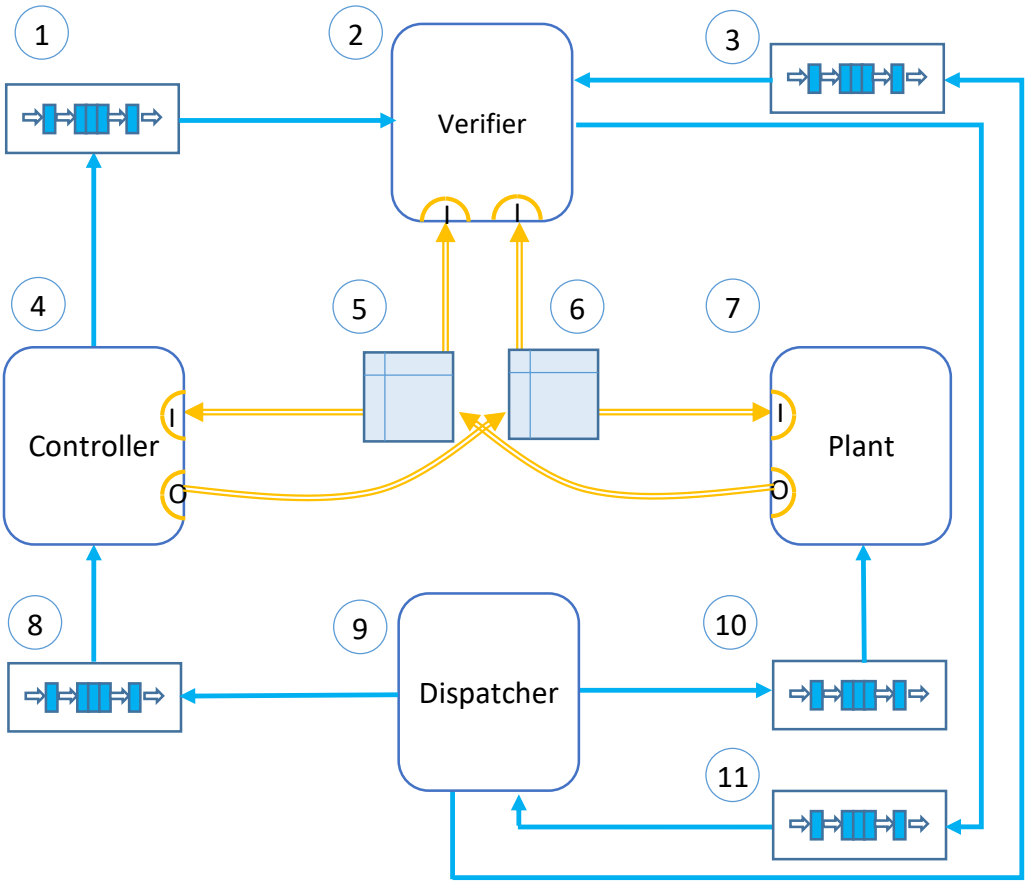

Рис. 7. Схема взаимодействия модулей комплекса автоматической динамической верификации.

Апробация предлагаемого подхода динамической верификации процесс-ориентированных алгоритмов управления КФС была проведена на следующих задачах:

1. Тестовая задача автоматизации электросушилки для рук.
2. Тестовая задача автоматизации системы управления уровнем воды в баке.
3. Программа управления подсистемой вакуумирования Большого солнечного вакуумного телескопа БСВТ.

В ЗАКЛЮЧЕНИИ подводятся итоги диссертационной работы, формулируются ее основные результаты.

ОСНОВНЫЕ ВЫВОДЫ И РЕЗУЛЬТАТЫ

В диссертации получены следующие основные научные результаты:

1. Предложена четырехкомпонентная формальная модель динамической верификации процесс-ориентированных программ управления КФС на имитаторе объекта управления.

2. Предложен численный метод динамической верификации процесс-ориентированных программ управления КФС.

3. Разработаны программные решения, использованные при проектировании комплексов автоматизированной и автоматической динамической верификации программ на языке Reflex.

4. Подход апробирован на задаче верификации вакуумной подсистемой Большого солнечного телескопа (БСВТ). Исследованы существующие подходы к верификации процесс-ориентированных программ управления КФС.

Разработанные методы динамической верификации могут использоваться при создании процесс-ориентированных программ управления КФС, а также для подготовки специалистов в области разработки КФС.

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ В ИЗДАНИЯХ, РЕКОМЕНДОВАННЫХ ВАК

1. Ковадло П. Г., Лубков А. А., Бевзов А. Н., Будников К. И., Власов С. В., Зотов А. А., Колобов Д. Ю., Курочкин А. В., Котов В. Н., Лылов С. А., Лях Т. В., Максимов А. С., Перебейнос С. В., Петухов А. Д., Пещеров В. С., Попов Ю. А., Русских И. В., Томин В. Е. Система автоматизации Большого солнечного вакуумного телескопа // Автометрия. 2016. Т. 52, вып. 2. С. 97–106.

2. Лях Т. В., Зюбин В.Е., Сизов. М. М. Опыт применения языка Reflex при автоматизации Большого солнечного вакуумного телескопа // «Промышленные АСУ и контроллеры». 2016. №7. С. 37-43.

3. Розов А. С., Лях Т. В., Краснов Д. В., Санжиев Е. С. Практическая апробация языка IndustrialC на примере автоматизации установки термовакуумного напыления // Вестн. НГУ. Серия: Информационные технологии. 2017. Т. 15, № 3. С. 90–99.

4. Лях Т. В., Зюбин В. Е., Гаранина Н. О. Автоматизированная верификация алгоритмов управления сложными технологическими объектами на программных имитаторах // Вестник НГУ Серия: Информационные технологии. - 2018. - Том 16, Выпуск № 4. - С. 85-94.

5. Лях Т. В., Зюбин В. Е., Гаранина Н. О. Автоматическая верификация алгоритмов управления в кибер-физических системах на программных имитаторах // Автометрия. 2019. Т. 55, вып. 2. С. 103-113.

ПРОЧИЕ ПУБЛИКАЦИИ ПО ТЕМЕ РАБОТЫ

6. Лях Т. В. Реализация концепции виртуальных объектов управления в среде LabVIEW на основе механизма DLL. // Сборник тезисов докладов всероссийской конференции "Индустриальные информационные системы" ИИС-2013 (г. Новосибирск, Россия, 24-28 апреля, 2013). С. 38.

7. Лях Т.В, Зюбин В.Е. Использование методов процесс-ориентированного программирования для задачи управления системой большого вакуумного

солнечного телескопа // Материалы Всероссийской научно-технической конференции «Современные проблемы радиоэлектроники» (г. Красноярск, Россия, 6-8 мая, 2014). С. 294-298.

8. Лях Т. В., Зюбин В.Е. Применение концепции виртуальных объектов управления для решения задач промышленной автоматизации // Материалы Девятой международной Ершовской конференции PSI-2014 (г. Санкт-Петербург, Россия, 24-27 июня, 2014). С. 57-64.

9. Лях Т. В., Зюбин В.Е. Использование языка Рефлекс в системах управления на базе Qt // Материалы XV Всероссийской конференции молодых ученых по математическому моделированию и информационным технологиям (г. Тюмень, Россия, 29-31 октября, 2014). С. 69-70.

10. Liakh T., Zyubin V. The Reflex Language Usage to Automate the Large Solar Vacuum Telescope // 17th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM). (Erlagol, Altai Republic, Russia, June 30 2016-July 4, 2016). pp. 137-139.

11. Краснов Д. В., Нефедов Д. В., Санжиев Е. С., Лях Т. В., Розов А. С. Практическая апробация процесс-ориентированной технологии программирования на открытых микроконтроллерных платформах // Вестник ВСГУТУ. 2017. Т. 66, вып.3. С.85-92.

12. Лях Т. В., Зюбин В. Е. Модельно-ориентированный метод автоматической верификации управляющих алгоритмов // Материалы XVII Всероссийской конференции молодых учёных по математическому моделированию и информационным технологиям. (Россия, г. Новосибирск, 30 октября – 3 ноября, 2016). С. 94-95.

13. Лях Т. В., Зюбин В. Е. Автоматическая верификация алгоритмов управления сложными технологическими объектами на программных имитаторах // Материалы V-й международной конференции «Математическое и компьютерное моделирование» - 2017 (г. Омск, Россия, 1 декабря, 2017). С. 128-130.

14. Liakh T., Rozov A., Zyubin V. LabVIEW-based automatic verification of process-oriented software // In 2019 International Siberian Conference on Control and Communications, SIBCON 2019 - Proceedings. Institute of Electrical and Electronics Engineers Inc. 2019. (Tomsk, Russian Federation, 18-20 April, 2019). pp. 1-4.

15. Liakh T., Anureev I., Garanina N., Rozov A., Zyubin V. Four-Component Model for Dynamic Verification of Process-Oriented Control Software for Cyber-Physical Systems. // 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON 2019), (Novosibirsk, Russian Federation, 21-22 October, 2019), pp. 0466-0471.