

# ЯППИ – УНИВЕРСАЛЬНЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ МИССИЙ АНПА

А.С. Пугачев, А.И. Боровик

Предлагается универсальный язык для написания миссий автономных необитаемых подводных аппаратов (АНПА). Главными особенностями предлагаемого языка являются: наличие пользовательских команд для описания основных задач подводных исследований, использование команд на русском языке, читаемость создаваемых миссий и их понятность для персонала, участвующего в применении подводной робототехники, отсутствие строгой типизации и параметризации команд, возможность расширения языка за счет создания новых команд на базе уже имеющихся. Статья описывает основные параметры языка, синтаксис, семантику и базовые команды, а также языковые инструменты – модуль верификации, который в диалоговом режиме позволяет разрешить неопределенность, возникающую в силу нестройной параметризации, модуль составных команд, позволяющий описать сложные команды на основе базовых, модуль расширения языка, отвечающий за добавление новых команд, и транслятор. Транслятор языка позволяет транслировать создаваемые миссии в коды программ-заданий для АНПА различных типов. Созданный язык назван ЯППИ – Язык Программирования Подводных Исследований и на данном этапе исследований тестируется и отлаживается на аппаратах ИПМТ ДВО РАН с учетом накопленного в институте опыта применения АНПА.

**Ключевые слова:** АНПА, миссия, программа-задание, язык программирования миссий АНПА, ЯППИ.

## Введение

К настоящему времени автономные необитаемые подводные аппараты зарекомендовали себя как технологичный и надёжный инструмент для решения широкого спектра задач подводных исследований – поиск затонувших объектов, обследование трубопроводов и подводных телекоммуникационных кабелей, вскрытие донной обстановки и составление батиметрических карт дна, научные исследования подводного мира и так далее. Важной спецификой АНПА является работа этих устройств в автономном режиме, с минимальным вмешательством со стороны оператора во время выполнения задания. В этих условиях очень большую значимость приобретает миссия аппарата – последовательность действий робота, которая заранее описывается оператором, помещается на компьютер и выполняется роботом во время его нахождения под водой [1, 2]. Миссия, также называемая программой-миссией или планом-заданием, обычно составляется с использованием одного из двух инструментов – языка описания миссии или графического планшета для планирования тра-

ектории движения аппарата на карте местности [3]. Программа, написанная на языке описания миссий, в дальнейшем либо компилируется в бинарный код программы для компьютера системы управления робота, либо исполняется на этом компьютере пошагово специальным интерпретатором. Для того чтобы проверить корректность составления задания до его непосредственного исполнения роботом, служат специальные инструменты, отображающие ход выполнения задачи на двумерном планшете или в трехмерной среде при выполнении симуляции [3]. План, составленный на графическом планшете, обычно не нуждается в дополнительной проверке в ходе симуляции, – он сразу переводится в конечный код программы-миссии. Оба подхода имеют свои преимущества и недостатки. Использование языка программирования миссий позволяет описать последовательности действий простыми командами, однако требует от оператора знания основ программирования и некоторого языка программирования, не всегда высокоуровневого. Наличие планшета позволяет составлять миссию без особых знаний и навыков, однако зачастую требует слишком большого

количества действий для решения задач, подразумевающих некоторую цикличность.

В ИПМТ ДВО РАН за всю историю существования института было создано несколько различных средств составления миссий робота, как языковых, так и графических. К настоящему моменту существует несколько параллельно применяемых в разных проектах инструментариев, каждый со своей историей и накопленным опытом применения. Подобная ситуация имеет несколько негативных последствий – сложность переучивания операторов для работы с новым роботом, проблемы в обмене опытом, когда решенные в одной среде задачи повторно решаются в другой, невозможность применения инструментария оператора и готовых миссий из одного проекта в других и так далее. Данная проблема не специфична для ИПМТ – в целом в отечественной и мировой подводной робототехнике на данный момент не существует универсальных систем составления программ-заданий – в каждом робототехническом проекте применяется свое решение, и все знания, умения и разработанный инструментарий операторов часто неприменимы на роботах, отличных от тех, с которыми конкретный оператор имел дело [4].

В данной работе предлагается новый инструмент создания миссий АНПА, ядром которого является разработанный Язык Программирования Подводных Исследований – ЯППИ. В настоящий момент язык тестируется и отлаживается на основе накопленного ИПМТ ДВО РАН опыта применения аппаратов собственного производства и может не учитывать некоторую специфику аппаратов других производителей.

## 1. Основные характеристики разработанного языка

Применение предлагаемого языка и его инструментариев на практике должно решать следующие задачи:

- унификация миссий – возможность создавать на ЯППИ миссии для различных АНПА; код на ЯППИ должен средствами языка транслироваться в код «нативного» языка программирования миссий конкретного робота («родного» языка миссий робота, с точки зрения транслятора ЯППИ являющегося *целевым*);
- формализация накопленного опыта применения подводных роботов, возможность использования удачных решений, полученных в ходе работы с одним аппаратом, на других платформах;
- возможность создавать миссии для решения основных задач подводных исследований

(вскрытие донной обстановки, поиск объектов, обследование протяженных объектов, детальное обследование точечных подводных объектов и т.п.) с помощью простых и лаконичных команд;

- упрощение подготовки операторов, возможность лёгкого переучивания оператора с одной робототехнической программно-аппаратной платформы на другую;
- получение миссий, написанных на близком к естественному языку, код которых должен быть понятен помимо оператора всем остальным участникам процесса применения робота – от штурмана до начальника экспедиции;
- лояльность к ошибкам оператора – оператор может забыть указать (или намеренно опустить, если на этапе составления миссии нет полной информации об условиях ее исполнения) часть параметров какой-либо команды – в этом случае языковые средства должны уточнить необходимую информацию на этапе трансляции кода в «нативный» язык.

При создании ЯППИ были выделены следующие основные требования:

- использование кириллицы для команд, команды должны представлять собой понятные слова и словосочетания русского языка, разделитель «\_»;
- наличие базовых команд для решения основных задач АНПА – поисковых и обследовательских;
- использование всех возможностей целевых языков программирования миссий робота для реализации пользовательских команд ЯППИ; в случае отсутствия необходимого функционала в целевом языке – реализация данных функций за счет возможностей модулей ЯППИ;
- наличие возможности расширения языка за счет внедрения новых команд, описываемых как последовательность базовых;
- отсутствие строгой типизации и параметризации – оператор может указывать параметры команды в любой последовательности, а также пропускать (забывать) любые параметры.

В состав инструментария языка должны входить следующие модули:

- **лексический, синтаксический и семантический анализаторы** текста, осуществляющие разбор исходного кода миссии на лексемы и переводящие его во внутреннее представление для дальнейшей интерпретации;

- **модуль верификации**, позволяющий в ходе диалога с оператором уточнить параметры функций, пропущенные в ходе составления оригинального плана-задания; результатом работы данного модуля является файл без каких-либо неопределенностей, который сохраняется на диске отдельно от исходного;
- **модуль составных команд**, определяющий составные команды на основе команд из базового набора, что дает возможность описать некоторые часто встречающиеся миссии с большим количеством повторяющихся команд более лаконично;
- **модуль расширения языка**, позволяющий добавлять пользовательские команды на основе базовых, составных и других команд расширенного набора; язык не допускает описания пользователем (оператором) своих команд внутри тела программы-миссии, но позволяет добавлять новые команды посредством данного модуля;
- **транслятор языка**, позволяющий на базе кода на ЯППИ получить «нативный» код миссии конкретного робота; данный модуль должен состоять из отдельных библиотек, отвечающих за реализацию функций ЯППИ средствами конкретной аппаратной платформы: транслятор должен быть реализован так, чтобы по возможности использовать аналоги функций ЯППИ в «нативном» языке и, только в случае их отсутствия, реализовывать нужные функции средствами примитивов движе-

ния и управления оборудованием, входящих в состав базового языка ЯППИ.

Процесс получения финального, «нативного» кода миссии из кода, написанного на ЯППИ, может быть представлен в виде следующей схемы обработки файла миссии инструментарием ЯППИ (см. рис. 1):

1. Лексический, синтаксический и семантический анализаторы, на входе имеющие код миссии на ЯППИ, составленный пользователем, на выходе – внутреннее представление программы в памяти.

2. Модуль верификации, на выходе выдающий файл с кодом на ЯППИ без неопределенностей, который в дальнейшем может быть использован при доработке миссии или ее анализе.

3. Модуль расширения языка, на выходе выдающий файл, в котором все команды из расширенного набора заменены на их реализацию наборами составных команд.

4. Модуль составных команд, на выходе выдающий файл, в котором все команды из составного набора заменены на их реализацию наборами базовых команд либо, в зависимости от целевого языка миссий, готовую программу для трансляции в «нативный» язык аппарата.

5. Транслятор языка, на выходе выдающий файл миссии на «нативном» языке конкретного робота.

## 2. Синтаксис языка и команды

Разработанный язык программирования является императивным, ориентированным на выпол-

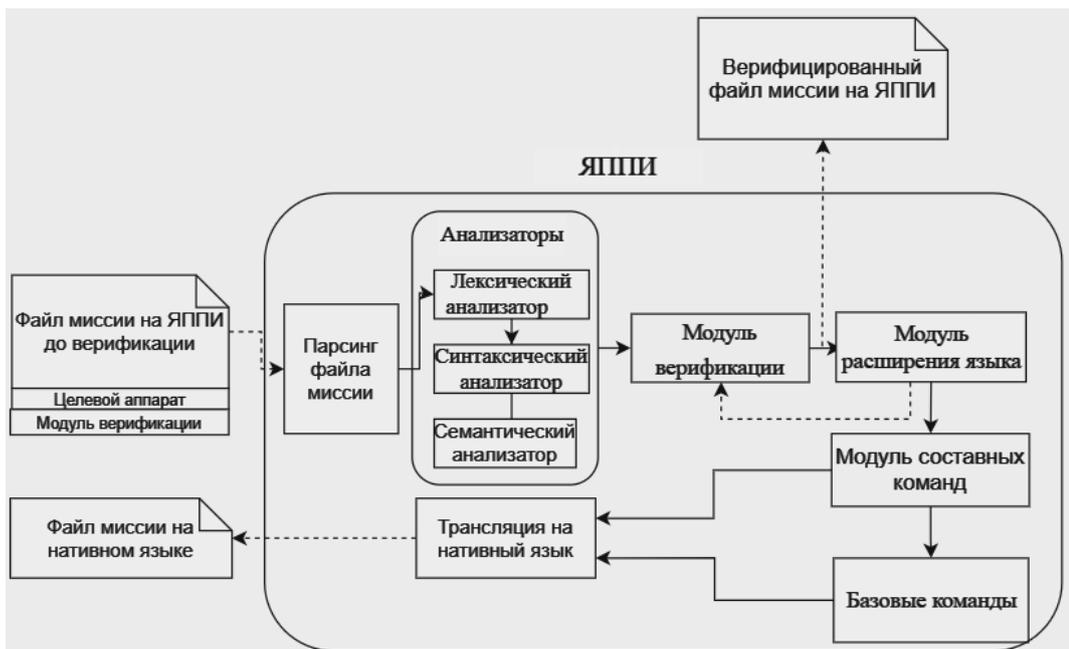


Рис. 1. Процесс трансляции кода миссии на ЯППИ

нение задач обзорно-поискового и поисково-обследовательских классов и содержит три основные конструкции – переменные, команды и события (описываемые ключевыми словами и специальными командами). Стандартная миссия на ЯППИ состоит из последовательного вызова следующих команд:

1. Команда «миссия», описывающая основные параметры выполнения программы.
2. Опциональная команда «район\_работ» с описанием района выполнения миссии.
3. Команды, последовательно описывающие задачи для АНПА с параметрами их выполнения.
4. Опционально, ключевые слова и команды, описывающие события, которые могут произойти в ходе выполнения миссии, и реакцию аппарата на них.

Пример простой миссии на ЯППИ, задающей обследование района прямоугольной формы с помощью гидролокатора бокового обзора (ГБО), приведен на рис. 2. Детальное описание синтаксиса и семантики языка приведено далее в этой главе.

### 2.1. Переменные

Как и в большинстве языков программирования, в ЯППИ переменные задаются с помощью идентификатора «=», перед которым указывается имя переменной, а после – значение. Допустимые типы переменных: целое число, число с плавающей точкой, координаты, массив, файл. Значения переменных (для размерных единиц) задаются в Международной системе единиц (СИ), координаты – в системе WGS-84 в градусах. Координаты широты и долготы отделяются друг от друга запятой и заключаются в квадратные скобки, при их объявлении применяется следующая нотация: <признак>градусы\*минуты'секунды". Признак может принимать значения: «СШ» или «ЮШ» для северной и южной широты, «ВД»

1. **высота** = 5
- 2.
3. **скорость** = 0.5
- 4.
5. **координаты1** = [131.901104\*ВД, 43.109489\*СШ]
- 6.
7. **координаты2** = [131\*54.066'ВД, 43\*6.569'СШ]
- 8.
9. **координаты3** = [131\*54'4"ВД, 43\*6'34.2"СШ]
- 10.
11. **массив\_координат** = [[131.901104\*ВД, 43.109489\*СШ],
12. [131.904816\*ВД, 43.109379\*СШ],
13. [131.908442\*ВД, 43.108659\*СШ],
14. [131.911296\*ВД, 43.107656\*СШ],
15. [131.915502\*ВД, 43.106936\*СШ]]
- 16.
17. **ид1** = файл(metadata/objects/id1)

Рис. 3. Пример описания переменных. Переменные координаты1, координаты2 и координаты3 равнозначны

или «ЗД» для восточной и западной долготы соответственно и может располагаться как перед основной строкой со значением координаты, так и после неё. Числовые значения градусов, минут и секунд могут быть как целыми, так и дробными. Массивы задаются последовательностью значений, разделённых запятыми и заключённых в квадратные скобки.

Для числовых переменных допускаются операции сложения («+»), вычитания («-»), умножения («\*») и деления («/») как на константы, так и на другие переменные. Массивы допускают динамическое добавление элементов с использованием оператора «+» и очистку с помощью выражения «имя\_массива = []». Подробнее синтаксис описания переменных представлен на рис. 3.

### 2.2. Базовые команды

В языке реализованы шесть базовых команд – «миссия», «район\_работ», «движение», «зависа-

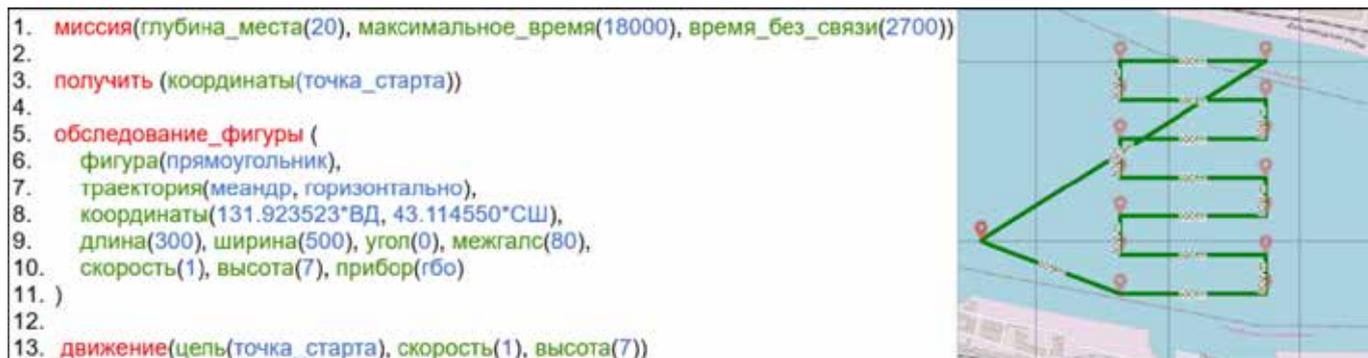


Рис. 2. Миссия обследования района прямоугольной формы с помощью ГБО, написанная на ЯППИ. Справа показана траектория движения АНПА в ходе выполнения миссии

ние», «устройство» и «получить», которые всегда непосредственно транслируются в нативный язык аппарата. Команды ЯППИ задаются посредством использования ключевых слов и круглых скобок, внутри которых записываются необходимые параметры. Открывающаяся круглая скобка «(» обозначает начало команды, закрывающаяся «)» определяет конец (см. рис. 4). Подобная структура задания позволяет идентифицировать начало и конец команды, в случае если пользователь переносит команду на несколько строк.

```

1. #пример
2.
3. движение(скорость(0.5), цель(131.901104°ВД, 43.109489°СШ), высота(4))
4.

```

Рис. 4. Пример описания команды

У каждой команды есть перечень обязательных и необязательных параметров. Для необязательных параметров модуль верификации не формирует запросы (но может формировать, если запустить его со специальным ключом). Внутри описания команды параметры указываются в произвольном порядке через запятую в формате «имя\_параметра(значение, опциональное\_уточнение)» или просто «имя\_параметра» (в случае, если параметр не подразумевает значения).

В начале программы-миссии указывается базовая команда «миссия» с одним обязательным параметром «глубина\_места», с помощью которого устанавливается предполагаемая глубина места в районе работ (для выбора оптимальных параметров заглупления аппарата и работы гидроакустической аппаратуры).

Команда «миссия» также имеет следующие необязательные параметры:

«максимальное\_время», указывающий максимальное возможное время работы аппарата в рамках миссии, при превышении которого должно выполняться автоматическое завершение работы;

«время\_без\_связи», задающий максимальное время работы АНПА без связи с оператором, при превышении которого выполнение миссии должно быть остановлено;

«балласт», при указании которого погружение аппарата выполняется с использованием балласта (если параметр «балласт» не указан, предполагается, что начальное заглупление выполняется под моторами).

При отсутствии команды «миссия» в тексте программы модуль верификации должен запросить у пользователя значения для всех ее обязательных па-

раметров (и для всех необязательных, если модуль запущен со специальных ключом).

Команда «район\_работ» определяет параметры акватории, в которой выполняются работы, и содержит следующие необязательные аргументы – «границы», «маяки\_дб» и «запретные\_зоны». Первый параметр задает массивом точек границы района работ, второй устанавливает массив координат маяков ГАНС с длинной базой, а третий определяет массив прямоугольных зон, находящихся внутри района работ, запрещенных для навигации АНПА (берега, мели, затопленные объекты и т.п.). Более детальное описание данной команды и ее параметров выходит за рамки этой статьи.

Команда «движение» предназначена для задания перемещения АНПА и содержит несколько обязательных аргументов:

аргумент «скорость», позволяющий задать скорость движения аппарата в среде (в м/с);

аргумент «цель» или «цели», задающий, в первом случае, целевую точку движения АНПА, во втором – массив точек, через которые АНПА должен последовательно пройти;

аргумент «высота», устанавливающий высоту движения аппарата относительно грунта (в метрах).

аргумент «глубина», задающий целевую глубину погружения относительно поверхности моря (в метрах);

Аргументы «высота» и «глубина» являются взаимоисключающими и принимают только положительные значения. В случае если были заданы оба параметра на этапе работы модуля верификации, программа сформулирует запрос на выбор одного из них.

Команда «зависание» позволяет аппарату удерживаться на заданном горизонте определенное количество времени и имеет четыре параметра:

«высота» или «глубина», которые, как и в команде «движение», определяют вертикальное положение аппарата относительно грунта или поверхности воды соответственно;

«время», определяющий временной интервал нахождения АНПА в состоянии зависания;

«координаты», задающий координаты удержания АНПА и являющийся опциональным (если данный параметр не указан, аппарат не пытается удерживаться в конкретной точке);

«условие» – условие, при котором происходит выход из цикла ожидания до истечения указанного времени; список возможных значений этого параметра включает в себя «уточнение\_координат» и «сигнал\_от\_устройства».

Команда «*устройство*» позволяет включить, выключить или настроить конкретный прибор из состава исследовательского оборудования аппарата. Команда имеет три аргумента: «*прибор*», задающий устройство («*фотокамера*», «*ГБО*», «*МЛЭ*», «*ЭМИ*», «*профилограф*» и т.д.), «*статус*», с возможными значениями «*вкл*» и «*выкл*», а также аргумент «*параметры*», принимающий значение двумерного массива вида «*[[имя\_параметра1, значение\_параметра1], [имя\_параметра2, значение\_параметра2], ...]*». Список возможных параметров работы зависит от конкретного оборудования и, например, для гидролокатора бокового обзора может включать параметр «*ширина\_полосы*», а для фотокамеры – «*кадров\_в\_секунду*».

Команда «*получить*» используется для получения значений из системы управления АНПА и объявления на их основе новых переменных. Параметрами команды являются имена атрибутов работы АНПА, значения которых запрашиваются у системы управления, а в качестве значений параметров указываются имена для создаваемых переменных. Например, команда «*получить(координаты(координаты1))*» создаст переменную «*координаты1*» и запишет в нее координаты текущего местоположения АНПА.

Список базовых команд языка и их параметров может изменяться и дорабатываться по мере внедрения ЯППИ в работу.

### 2.3. Составные команды

Составные команды основаны на базовых и выполняют наиболее часто встречающиеся задачи АНПА. В настоящий момент в ЯППИ реализованы 3 составные команды – «*обследование\_фигуры*», «*обследование\_линии*» и «*обследование\_точки*».

Команда «*обследование\_фигуры*» предназначена для обследования акватории определенной формы. Помимо атрибутов «*высота*» или «*глубина*» и «*скорость*», задающих параметры движения аппарата во время обследования, команда также имеет атрибут «*координаты*», принимающий значение точки, от которой строится фигура обследования, атрибут «*фигура*», который определяет геометрическую форму обследуемой территории, атрибут «*траектория*», задающий траекторию движения аппарата, и параметр «*прибор*», в котором указывается работающее исследовательское оборудование. Также могут задаваться опциональные атрибуты: «*межгалс*», который определяет частоту пересечения фигуры в метрах и «*угол*», позволяющий повернуть фигуру на указанное значение градусов. В качестве фигур обследования используются: «*квадрат*», «*прямоугольник*» и

«*произвольная\_фигура*», задаваемая массивом точек [5, 6]. Для задания длины прямоугольника и стороны квадрата используется параметр «*длина*», для задания ширины прямоугольника – параметр «*ширина*». Поддерживаемые ЯППИ на данный момент траектории обследования фигур: «*меандр*» и «*спираль*». Для траектории «*меандр*» может быть указано уточнение, определяющее способ построения траектории – «*горизонтально*» или «*вертикально*». Параметры галса подхода к фигуре (скорость и вертикальный канал) выбираются таким образом, чтобы осуществить безопасный и оперативный подход к начальной точке обследования.

Команда «*обследование\_линии*» позволяет проводить обследование акватории вдоль заданной ломанной линии. В качестве возможных траекторий движения можно использовать пересекающиеся или параллельные линии, что позволяет гибко адаптировать маршрут обследования в зависимости от задач миссии. Данная команда принимает в качестве параметров атрибуты движения («*высота*» или «*глубина*», «*скорость*»), а также параметры «*координаты*», «*траектория*», «*расстояние*», «*межгалс*» и «*прибор*». Параметр «*координаты*» задает линию обследования в виде массива координат, параметр «*траектория*» позволяет выбрать одну из двух возможных траекторий: «*ломаная*» или «*параллельная*». В первом случае параметр «*межгалс*» задает расстояние между центрами галсов «*змейки*», а параметр «*расстояние*» – длины галсов «*змейки*». Во втором случае задается движение АНПА параллельно целевой линии на определяемой параметром «*расстояние*» дистанции перпендикулярного смещения.

Команда «*обследование\_точки*» описывает обследование указанной точки акватории. На вход подаются параметры движения («*высота*» или «*глубина*», «*скорость*»), а также параметры «*координаты*», «*прибор*», «*траектория*» и «*проходов*». Параметр «*траектория*» задает фигуру обследования и может принимать значения «*звезда*» или «*спираль*», параметр «*проходов*» задает количество проходов аппаратом заданной точки.

### 2.4. Обработка событий

В языке предусмотрены методы для описания реакций АНПА на события. Данный механизм позволяет АНПА в реальном времени анализировать ход выполнения миссии и, на основе ряда условий, принимать решение о дальнейших действиях [7–9]. Использование этого механизма возможно только для тех аппаратов, языки миссий которых поддерживают работу с событиями.

Описание события начинается с ключевого слова «событие», которое открывает блок описания события. Внутри этого блока первой указывается команда «триггер», задающая событие, на которое АНПА должен отреагировать в процессе выполнения миссии. В качестве параметров команды «триггер» задаются:

событие срабатывания триггера (например, «объект\_на\_гбо»), значением которого будет переменная, уточняющая характер события (например, ссылка на файл, задающий объект поиска для алгоритма анализа акустических изображений);

параметры, имеющие вид *получить\_имя\_характеристики(имя\_переменной)*, позволяющие получить некоторые характеристики события и сохранить их в отдельной переменной (например, параметр *получить\_координаты(координаты\_объекта)*, позволяющий записать координаты найденного алгоритмом поиска объекта на ГБО-изображении в переменную *координаты\_объекта*).

После описания триггера в блоке описания события указывается последовательность команд, которая будет выполнена аппаратом при возникновении описываемого события. Завершается описание события ключевым словом «конец\_события».

По умолчанию областью действия события является вся миссия. Если область действия необходимо ограничить определенной последовательностью команд, тогда перед этой последовательностью нужно указать ключевое слово «задача», в скобках указав уникальный идентификатор задачи, а после – «конец\_задачи» с идентификатором закрываемого блока. ЯППИ позволяет выходить из задачи, описав в произвольном месте внутри блока ключевое слово

*выход\_из\_задачи*. В этом случае, система управления аппарата сразу перейдет к выполнению команды, следующей за ключевым словом *конец\_задачи*.

Пример описания миссии с событийным блоком на ЯППИ приведен на рис. 5. В данном примере описывается задача поиска объектов в прямоугольной области с помощью ГБО и их дообследование с помощью фотосистемы. Объекты описываются файлом

```

1. счетчик = 0
2. ид1 = файл(metadata/objects/id1)
3.
4. задача(1)
5. обследование_фигуры (
6.   фигура(прямоугольник),
7.   траектория(меандр, вертикально),
8.   координаты(131.923545*ВД, 43.114547*СШ),
9.   длина(500), ширина(300), угол(0), межгалс(80),
10.  скорость(1), высота(7), прибор(гбо)
11. )
12.
13. событие
14. триггер(объект_на_гбо(ид1),
15.  получение_координат(координаты_точки))
16.
17. обследование_точки(координаты(координаты_точки),
18.  траектория(звезда), высота(2), скорость(0.5),
19.  проходов(7), прибор(фотокамера))
20.
21. счетчик = счетчик + 1
22. конец_события
23.
24. событие
25. триггер(счетчик >= 10)
26. выход_из_задачи(1)
27. конец_события
28.
29. конец_задачи(1)

```

Рис. 5. Фрагмент миссии на ЯППИ с обработкой событий

```

а
1. обследование_гбо_прямоугольник(координаты($коорд), длина_района($длин),
2.   ширина_района($шир), ширина_полосы($полос=200), перекрытие($перек=30))
3. {
4.
5.   устройство(прибор(гбо), параметры([[ширина_полосы, $полос]])
6.
7.   обследование_фигуры(фигура(прямоугольник), траектория(меандр, вертикально),
8.     координаты($коорд), длина($длин), ширина($шир), угол(0),
9.     межгалс($полос - $перек), скорость(1), высота(7), прибор(гбо))
10. }
б
1.
2. обследование_гбо_прямоугольник(координаты(131.90885*ВД, 43.10523*СШ),
3.   длина_района(500), ширина_района(300), ширина_полосы(250), перекрытие(40))
4.

```

Рис. 6. Расширенная команда: а – описание команды в специальном файле, б – применение команды в миссии

специального вида, при дообследовании 10 объектов в области поисковая задача прерывается.

### 2.5. Расширенные команды

Помимо базовых и составных команд в языке предусмотрены команды расширенного функционала. Подобные команды реализованы на основе последовательности базовых и составных и могут объявляться пользователями языка самостоятельно, с использованием специального синтаксиса. При описании команды пользователь описывает также ее параметры и может задать для некоторых из них значения по умолчанию (в этом случае параметр становится необязательным). Например, команда «обследование\_гбо\_прямоугольник» с обязательными параметрами «координаты», «длина\_района», «ширина\_района» и необязательными параметрами «ширина\_полосы» и «перекрытие» выполняет задачу обследования акватории прямоугольной формы с помощью ГБО и содержит последовательный вызов базовой команды «устройство» и составной команды «обследование\_фигуры». Пример описания подобной команды, а также способ ее вызова в коде миссии приведены на рис. 6.

### 2.6. Комментарии

В языке предусмотрены однострочные комментарии для пояснения к исходному тексту программы. Идентификатором комментария служит символ «#», обозначающий, что данный текст не будет обрабатываться транслятором.

## 3. Модули языка

### 3.1. Анализаторы

Как и в большинстве языков программирования, в ЯППИ представлены лексический, синтаксический и семантический анализаторы. На этапе лексического анализа происходит построчное считывание файла миссий. Происходит форматирование данных, убираются пустые строки, комментарии, удаляются отступы. В результате работы лексического анализатора возвращается последовательность токенов.

На этапе синтаксического анализа программа распределяет токены по структурам данных в зависимости от того, к какому типу относится каждый токен. Токен может принадлежать переменной, команде, событию, либо его тип может быть не определен. В последнем случае анализатор добавляет токен к предыдущей конструкции, тип которой определен, поскольку на данном этапе считается, что неопреде-

ленный тип данных может являться незаконченной конструкцией одного из трех известных типов.

Семантический анализатор проводит проверку логики последовательности программы. Значения параметров заполняются переменными при необходимости. Происходит нормализация значений атрибутов к соответствующим типам данных.

### 3.2. Модуль верификации

В разработанном языке предусмотрен модуль проверки заданных параметров для каждой команды. После запуска программы, отработки лексического и синтаксического анализаторов на семантическом уровне происходит определение введенных параметров. В случае если для команды не был задан обязательный параметр, модуль запросит у пользователя ввод пропущенных данных.

Во время проверки переданных параметров данный модуль выполняет, в том числе, поиск неизвестных атрибутов. При их наличии программа уведомляет об этом пользователя и не учитывает их в работе.

Инструментарий ЯППИ на вход подается файл миссии на ЯППИ, название целевого аппарата (по нему будет выбран конкретный модуль транслятора для получения «нативного» кода миссии), а также параметры работы отдельных модулей языка, в частности модуля верификации. Для модуля верификации предусмотрен ключ, инструктирующий его запрашивать значения для всех параметров, даже необязательных.

### 3.3. Модули составных команд и расширения языка

Для реализации наиболее часто встречающихся задач был предусмотрен модуль, позволяющий реализовать составные команды на основе команд из базового набора для тех целевых языков описания миссий, которые не содержат данных команд. В случае, если в целевом языке есть реализация конкретной составной команды, модуль не будет выполнять ее преобразование в последовательность базовых – предполагается, что «нативная» реализация лучше учитывает специфику конкретного аппарата.

В ЯППИ предусмотрен модуль расширения функционала, который позволяет объявлять пользовательские команды на базе последовательности составных и базовых команд – *модуль расширения языка*. Данный модуль служит для выполнения двух задач: создания надстроек над встроенными расширенными командами, заполняя некоторые параметры автоматически, а также реализации более сложных алгоритмов на базе составных и базовых команд

языка ЯППИ. Файлы с описанием пользовательских команд помещаются в специальную директорию инструментария ЯППИ и сразу после этого становятся доступными для использования в миссиях АНПА. Библиотека расширенных функций будет пополняться по мере развития языка его основными пользователями. Подобный механизм формализует способ обмена опытом между операторами.

#### 4. Примеры миссий на ЯППИ

В качестве примеров использования ЯППИ можно рассмотреть несколько миссий, которые были разработаны на нем и экспортированы в целевой язык описания миссий, применяемый на АНПА ММТ-3500 производства ИПМТ ДВО РАН [10]. Особенностью данного языка является отсутствие реализации каких-либо команд из составного набора ЯППИ, поэтому модуль составных команд использовал внутренние реализации, основанные на базовых командах.

Первым примером является задача по обследованию водной акватории прямоугольной формы. Подобные задачи часто возникают, когда требуется обнаружить объект, местоположение которого неизвестно с высокой степенью точности. Для таких задач обычно выбирается конкретный участок акватории, форма которого может являться квадратом, прямоугольником или произвольной геометрической фигурой. Процесс обследования данного участка подразумевает прохождение всей выбранной области с целью её полного охвата [11, 12].

Одним из наиболее эффективных и распространенных методов планирования траектории при подобном обследовании является траектория в форме меандра. Форма меандра представляет собой чередующиеся параллельные линии, которые последовательно проходят через всю область. В результате такого подхода формируется характерная сетка проходов, которая позволяет минимизировать вероят-

ность пропуска исследуемого объекта. Код на ЯППИ и графическое отображение результирующей миссии ММТ-3500 во встроенном редакторе изображены на рис. 7.

Вторым примером может служить задача по движению АНПА вдоль заданной отдельными точками траектории. Одной из важных целей применения АНПА является проведение обследования протяженных объектов, таких как трубопроводы, электрические кабели и другие похожие конструкции, для выявления возможных повреждений или дефектов. Можно выделить два способа обследования – движение вдоль объекта либо движение «змейкой» с постоянным прохождением над объектом под определёнными углами. Первый вариант подходит для фиксации состояния объекта с использованием гидролокатора бокового обзора (в этом случае движение должно выполняться на небольшом отстоянии от объекта, чтобы на изображениях он находился вне зоны акустической тени) или многолучевого эхолота (в этом случае необходимо двигаться непосредственно над объектом). Второй вариант подходит для работ на небольших высотах прецизионным оборудованием (фотосистема, дефектоскопы, ЭМИ) – в этом случае многократное пересечение объекта позволяет, с одной стороны, обследовать его с разных углов, с другой – помогает избежать ухода с траектории из-за возможных навигационных ошибок. В данном примере на вход системе задаются координаты протяженного объекта, по которым впоследствии строится маршрут движения АНПА. Код на ЯППИ и графическое отображение миссии ММТ-3500 во встроенном редакторе для случая использования ломаной линии обследования представлены на рис. 8, а, для случая параллельной линии – на рисунке 8, б.

В качестве третьего примера рассмотрим задачу детального обследования подводного объекта с известными координатами. Обычно в таком случае обследование выполняется с использованием фотосистемы, но возможно применение и других точных

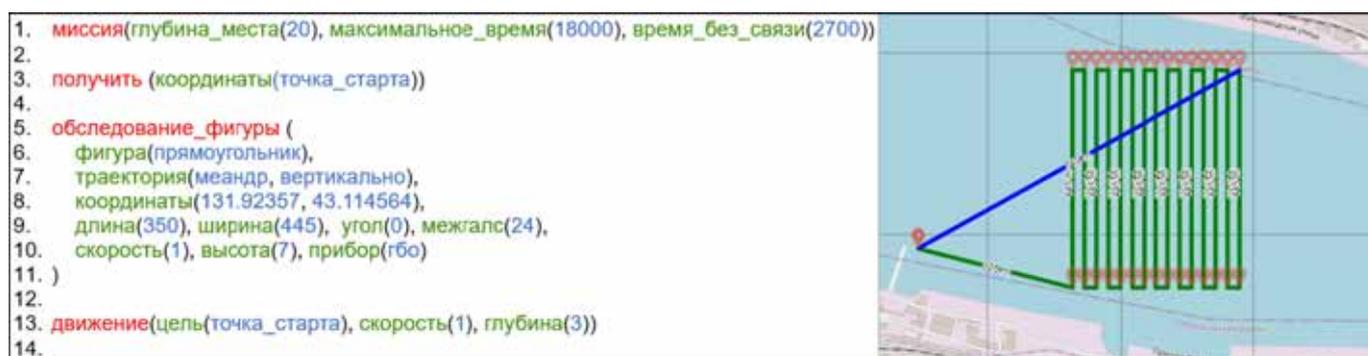


Рис. 7. Миссия обследования прямоугольной области

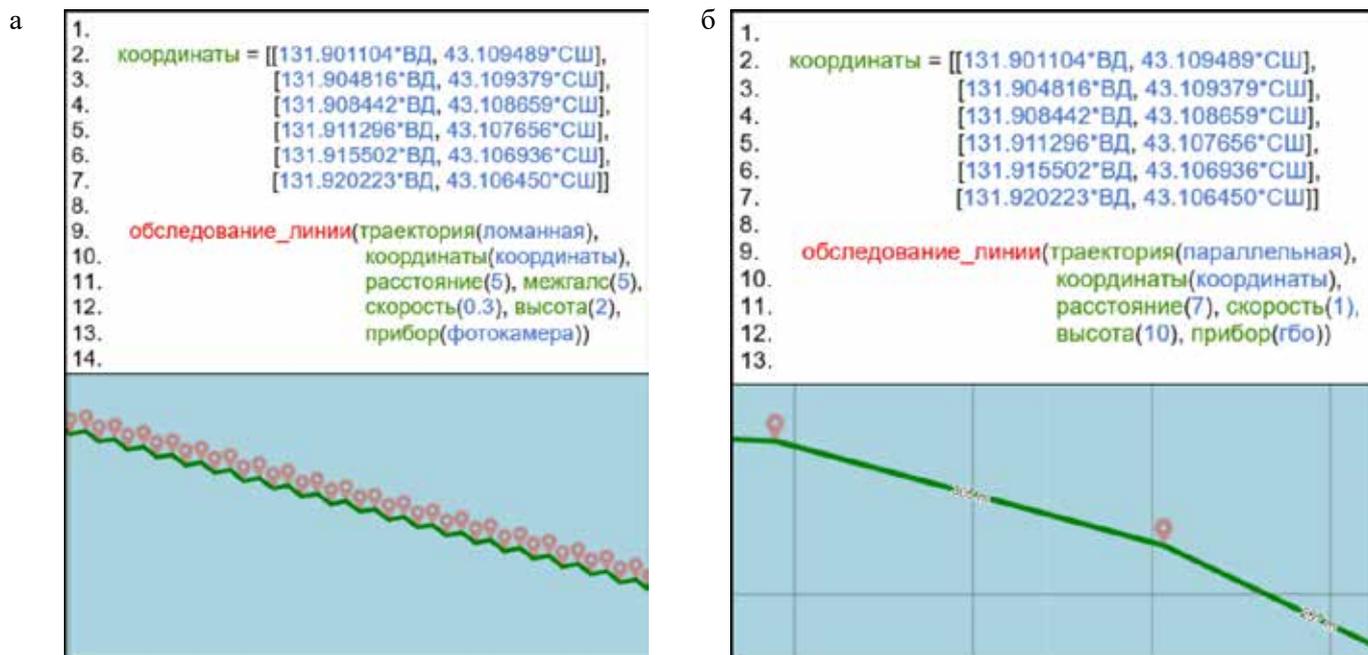


Рис. 8. Фрагменты миссии ЯППИ с обследованием линии: а – пример ломаной линии, б – пример параллельной линии

приборов, например, дефектоскопов или МЛЭ. В данной задаче траектория движения АНПА строится таким образом, чтобы аппарат пересек известные координаты несколько раз – для фиксирования объекта с разных сторон, а также для уверенного прохождения над объектом с учетом погрешности навигации. Код на ЯППИ и графическое отображение миссии ММТ-3500 во встроенном редакторе для случая обследования точки представлены на рис. 9.

```

1. обследование_точки(
2. координаты(131.901104*ВД, 43.109489*СШ),
3. траектория(звезда), расстояние(4.5)
4. высота(2), скорость(0.3),
5. проходов(7), прибор(фотокамера))
    
```

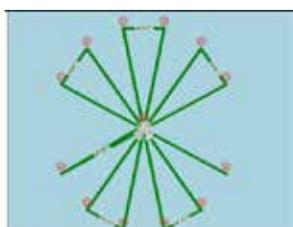


Рис. 9. Фрагмент миссии с обследованием точки

### Заключение

Полученный в ходе работы язык ЯППИ и инструментарий, позволяющий создавать миссии для системы управления аппарата ММТ-3500, в настоящий момент проходят отладку в ИПМТ ДВО РАН. В перспективе разработанный язык будет дорабатываться по мере использования, в частности, будет рассмотрена целесообразность добавления операторов ветвления, цикла и других конструкций, используемых в популярных высокоуровневых языках программирования. В ближайшее время будет реализована функция экспортирования и импортирования маршрутов АНПА с использованием картографического ПО – OpenCPN и SAS.Планета. Также для удобного написания программ-миссий на ЯППИ будет разработан специальный плагин для подсветки синтаксиса в редакторе кода Visual Studio Code. В дальнейшем планируется разработать трансляторы языка для других аппаратов, а также использовать ЯППИ в разрабатываемой интеллектуальной системе помощи в планировании миссий АНПА на базе нейросети.

### СПИСОК ИСТОЧНИКОВ

1. Агеев М.Д., Касаткин Б.А., Киселев Л.В., Молоков Ю.Г., Никифоров В.В., Рылов Н.И. Автоматические подводные аппараты. Л.: Судостроение, 1981. 224 с.
2. Инзарцев А.В., Киселев Л.В., Костенко В.В., Матвиенко Ю.В., Павин А.М., Щербатюк А.Ф. Подводные робототехнические комплексы: системы технологий применения. Владивосток: ИПМТ ДВО РАН, 2018. 368 с.
3. Инзарцев А.В. К вопросу о способах представления задания для обследовательского подводного робота / А. В. Инзарцев, А. В. Багницкий // Технические проблемы освоения Мирового океана. 2013. Т. 5. С. 376–381. EDN YSWCTR.

4. Боровик А.И. Компонентно-ориентированная программная платформа для автономных необитаемых подводных аппаратов: дис. ... канд. техн. наук: 05.13.11. Владивосток, 2019. 177 с.
5. Багницкий А.В., Инзарцев А.В. Автоматизация подготовки миссии для автономного необитаемого подводного аппарата в задачах обследования акваторий // Подводные исследования и робототехника. 2010. № 2(10). С. 17–24.
6. Инзарцев А.В., Багницкий А.В. Планирование и реализация траекторий движения автономного подводного робота при выполнении мониторинга в акваториях различных типов // Подводные исследования и робототехника. 2016. № 2(22). С. 25–35.
7. Павин А.М. Разработка алгоритмов поиска и обследования искусственных протяженных объектов с помощью автономного необитаемого подводного аппарата: автореф. дис. ... канд. техн. наук: 05.13.18. Владивосток, 2010. 21 с.
8. Инзарцев А.В. Представление заданий для автономного подводного робота на базе поведенческих методов / А.В. Инзарцев, А.В. Багницкий // XII Всеросс. совещ. по проблемам управления. М.: Институт проблем управления им. В.А. Трапезникова РАН, 2014. С. 3980–3990.
9. Arkin R. Behavior-Based Robotics. Cambridge, MA: MIT Press, 1998. 505 p.
10. Разработка автономного подводного робота, предназначенного для выполнения манипуляционных операций / А.Ю. Коноплин, Д.Н. Михайлов, В.В. Костенко [и др.] // Перспективные системы и задачи управления : Материалы XVIII Всеросс. науч.-практич. конф. и XIV молодежной школы-семинара. Таганрог: Лукоморье, 2023. С. 534–538. EDN FHMECK.
11. Inzartsev A., Pavin A. AUV Application for Inspection of Underwater Communications // Underwater Vehicles / ed. by Alexander V. Inzartsev. Vienna: In-Tech Publishers, 2009. P. 215–234. Open access: [http://www.intechopen.com/books/underwater\\_vehicles](http://www.intechopen.com/books/underwater_vehicles).
12. Мониторинг морского дна с применением технологий интеллектуальной обработки данных поисковых устройств на борту автономного необитаемого подводного аппарата / А.В. Инзарцев, Ю.В. Матвиенко, А.М. Павин, Н.И. Рылов // Подводные исследования и робототехника. 2015. № 2(20). С. 20–27. EDN VHXGBV.

## Справка об авторах

**БОРОВИК Алексей Игоревич**, к. т. н., в. н. с., зав. лабораторией робототехнических систем

Институт проблем морских технологий им. академика М.Д. Агеева Дальневосточного отделения Российской академии наук

Адрес: 690091, Владивосток, ул. Суханова, 5А

**Область научных интересов:** АНПА, ТНПА, системы управления роботами, робототехнические программные платформы, разработка программного обеспечения

**E-mail:** alexey@borovik.me

**Тел.:** 8(423)2-215-545, доб. 509

**ORCID:** 0000-0002-9696-2751

**ПУГАЧЕВ Андрей Сергеевич**, аспирант, инженер лаборатории робототехнических систем

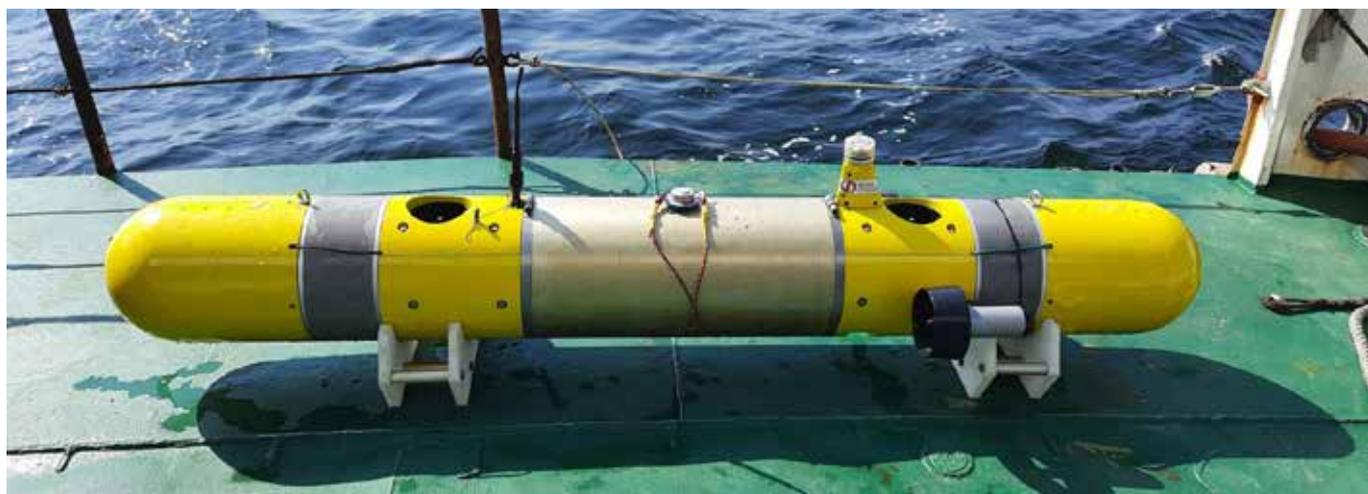
Институт проблем морских технологий им. академика М.Д. Агеева Дальневосточного отделения Российской академии наук

Адрес: 690091, Владивосток, ул. Суханова, 5А

**Область научных интересов:** вычислительные системы, компиляторы, нейронные сети, прикладное программное обеспечение

**E-mail:** pugachev@marine.febras.ru

**Тел.:** 8(423)2-215-545, доб. 512



### Для цитирования:

Пугачев А.С., Боровик А.И. ЯППИ – УНИВЕРСАЛЬНЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ МИССИЙ АНПА // Подводные исследования и робототехника. 2024. № 3 (49). С. 26–37. DOI: 10.37102/1992-4429\_2024\_49\_03\_03. EDN: GVSAMY.



# URPL – UNIVERSAL LANGUAGE FOR AUV MISSION PROGRAMMING

**A.S. Pugachev, A.I. Borovik**

This article proposes a universal language for developing autonomous underwater vehicle (AUV) missions. The main features of the proposed language are: the presence of high-level commands to describe the main tasks of underwater research, the use of commands in Russian, the readability of the created missions and their understandability for personnel involved in the use of underwater robotics, the lack of strict typing and parameterization of commands, the possibility of expanding the language by creating new commands based on the old ones. The article describes the main parameters of the language, syntax, semantics and basic commands, as well as language tools - a parameter refinement module, which interactively allows you to resolve uncertainty arising due to non-strict parameterization, low-level command module that implements more complex commands based on basic ones, a language extension module responsible for adding new commands, and a translator. The language translator allows you to translate created missions into task program codes for various AUVs. The created language is called URPL – Underwater Research Programming Language and at the current stage of research it is being tested and debugged on the AUVs of the IPMT FEB RAS, taking into account the experience accumulated at the institute.

**Keywords:** AUV, mission, task program, AUV mission programming language, URPL

## References

- Ageev M.D., Kasatkin B.A., Kiselev L.V., Molokov Ju.G., Nikiforov V.V., Rylov N.I. Avtomaticheskie podvodnye apparaty. L.: Sudostroenie, 1981. 224 s. (In Russ.).
- Inzarcev A.V., Kiselev L.V., Kostenko V.V., Matvienko Ju.V., Pavin A.M., Shherbatjuk A.F. Podvodnye robototekhnicheskie komplekсы: sistemy tehnologii primeneniya. Vladivostok: IPMT DVO RAN, 2018. 368 s. (In Russ.).
- Inzarcev A.V. K voprosu o sposobah predstavleniya zadaniya dlja obsledovatel'skogo podvodnogo robota. A.V. Inzarcev, A.V. Bagnickij. Tehnicheskie problemy osvoeniya Mirovogo okeana. 2013. T. 5. S. 376–381. EDN YSWCTR. (In Russ.).
- Borovik A.I. Komponentno-orientirovannaja programmnaja platforma dlja avtonomnyh neobitaemyh podvodnyh apparatov: dis. ... kand. tehn. nauk: 05.13.11. Vladivostok, 2019. 177 s. (In Russ.).
- Bagnickij A.V., Inzarcev A.V. Avtomatizacija podgotovki mis-sii dlja avtonomnogo neobitaemogo podvodnogo apparata v zadachah obsledovaniya akvatorij. Podvodnye issledovaniya i robototekhnika. 2010. № 2(10). S. 17–24. EDN ROTMCZ.
- Inzarcev A.V., Bagnickij A.V. Planirovanie i realizacija traektorij dvizheniya avtonomnogo podvodnogo robota pri vypolnenii monitoringa v akvatorijah razlichnyh tipov. Podvodnye issledovaniya i robototekhnika. 2016. № 2(22). S. 25–35.
- Pavin A.M. Razrabotka algoritmov poiska i obsledovaniya iskusstvennyh protjazhennyh ob#ektov s pomoshh'ju avtonomnogo neobitaemogo podvodnogo apparata: avtoref. dis. ... kand. tehn. nauk: 05.13.18. Vladivostok, 2010. 21 s.
- Inzarcev A.V. Predstavlenie zadaniy dlja avtonomnogo podvodnogo robota na baze povedencheskih metodov. A.V. Inzarcev, A.V. Bagnickij. XII Vseross. soveshh. po problemam upravleniya. M.: Institut problem upravleniya im. V.A. Trapeznikova RAN, 2014. S. 3980–3990.
- Arkin R. Behavior-Based Robotics. Cambridge, MA. MIT Press, May, 1998, 505 pp.
- Razrabotka avtonomnogo podvodnogo robota, prednaznachenogo dlja vypolneniya manipuljacionnyh operacij. A.Ju. Konoplin,

D.N. Mihajlov, V.V. Kostenko [i dr.]. Perspektivnye sistemy i zadachi upravleniya : Materialy XVIII Vseross. nauch.-praktich. konf. i XIV molo-dezhnoj shkoly-seminara. Taganrog: Lukomor'e, 2023. S. 534–538. EDN FHMECK.

11. Inzartsev A., Pavin A. AUV Application for Inspection of Underwater Communications. Underwater Vehicles, ed. by Alexander V. Inzartsev, In-Tech Publishers, Vienna, January, 2009, 582 pages, pp. 215–234, open access: [http://www.intechopen.com/books/underwater\\_vehicles](http://www.intechopen.com/books/underwater_vehicles).

12. Monitoring morskogo dna s primeneniem tehnologij intellektual'noj obrabotki dannyh poiskovyh ustrojstv na burtu avtonomnogo neobitaemogo podvodnogo apparata. A.V. Inzarcev, Ju.V. Matvienko, A.M. Pavin, N. I. Rylov. Podvodnye issledovaniya i robototekhnika. 2015. No. 2(20). S. 20–27. EDN VHXGBV.

## Information about the authors

**BOROVIK Alexey Igorevich**, Candidate of Technical Sciences, Leading Researcher, Head of the Laboratory of Robotic Systems Institute of Marine Technology Problems, Far Eastern Branch of Russian Academy of Science

**Address:** Russia, 690091, Vladivostok, Sukhanova str., 5A

**Research interests:** AUV, ROV, robot control systems, robotics software frameworks, software development

**E-mail:** alexey@borovik.me. **Phone:** +7(423)2-215-545, ext. 509

**ORCID:** 0000-0002-9696-2751

**PUGACHEV Andrey Sergeevich**, Postgraduate student, engineer at the laboratory of robotic systems Institute of Marine Technology Problems, Far Eastern Branch of Russian Academy of Science

**Research interests:** computing systems, compilers, neural networks, application software

**Address:** Russia, 690091, Vladivostok, Sukhanova str., 5A

**E-mail:** pugachev@marine.febras.ru. **Phone:** +7(423)2-215-545, ext. 512