

УДК 004.896, 004.5  
DOI: 10.15827/0236-235X.136.543-554

Дата подачи статьи: 10.06.21  
2021. Т. 34. № 4. С. 543–554

## **Знаковая система управления мобильной платформой**

*М.А. Ровбо*<sup>1</sup>, инженер-исследователь, *rovboma@gmail.com*

*П.С. Сорокоумов*<sup>1</sup>, инженер-исследователь, *petr.sorokoumov@gmail.com*

<sup>1</sup> *Национальный исследовательский центр «Курчатовский институт», г. Москва, 123182, Россия*

В работе рассматривается объединение системы управления роботом на основе знаковой модели мира и человеко-машинных интерфейсов, в частности, голосового, в единую систему, обрабатывающую как команды пользователя, так и автономное поведение робота.

Разработанная система позволяет управлять роботом с помощью голосового интерфейса, обрабатывать длительные поведенческие процедуры, устранять неоднозначности команды за счет разрешения пространственных отношений и учета направления взгляда оператора. Это достигается путем интеграции всей необходимой для принятия решения и выбора действия роботом информации в модель мира, применения логического вывода для ее пополнения и использования эвристик для устранения неоднозначностей в речевой команде, когда известных фактов для этого не хватает. Система принятия решения на основе данной модели мира также обеспечивает автономную реакцию робота на особые ситуации.

Описывается архитектура разработанной системы и демонстрируется применимость данного подхода на примере имитационной модели мобильного робота в Gazebo. Моделирование показало возможность управления мобильной платформой с помощью разработанной системы, причем для оператора не прямое управление роботом оказалось более комфортным. Однако следует отметить рост вычислительной нагрузки в случае обработки большого числа объектов в мире, который частично компенсируется алгоритмами, позволяющими дополнять модель лишь той информацией, которая необходима для обработки текущей команды при логическом выводе, а также невозможность обработки противоречивой информации в текущей реализации.

**Ключевые слова:** семиотическая модель, система управления, робототехника, человеко-машинный интерфейс, голосовое управление.

Развитие робототехники и технологий искусственного интеллекта, а также доступность современных высокопроизводительных вычислительных элементов для бортовых систем дают возможность создавать многофункциональных роботов, способных выполнять различные задачи в сложной и неподготовленной среде (вне специально оборудованных производственных помещений), а также находить более широкое применение в быту. Это приводит к необходимости разработки соответствующих человеко-машинных интерфейсов, которые, с одной стороны, позволяют удобным способом отдавать ряд различных по своей природе команд (движение, взаимодействие с объектами), а с другой, являются по возможности наиболее простыми и понятными для использования неподготовленным оператором.

Высокоуровневые интерфейсы управления роботами бывают разными в зависимости от возможностей и назначения робота (указание навигационной цели в виде точки на карте, выбор целевого объекта на изображении, отдача команды голосом и др.). Объеди-

няет их необходимость передачи от человека роботу высокоуровневой информации о действиях или окружении робота. Распространенной формой организации такой информации являются семантические представления [1]. Считается, что, когда такая база знаний выступает в качестве элемента принятия решений в системе управления роботом, систему можно отнести к когнитивным архитектурам.

Семантический подход можно использовать в моделях мира роботов с различными целями: в человеко-машинных интерфейсах [2], в экспертных системах [3], для навигации [4]. В данной работе используется семантическая сеть (графовая структура для представления знаний [5]) как основа модели мира робота, которая помогает связывать объекты с их представлением в семантическом разборе команды оператора и обеспечивает логический вывод, позволяющий роботу реагировать на ситуации предсказуемым образом.

При создании интерфейсов управления для роботов на основе семантических технологий

современные исследователи применяют разные подходы. Некоторые для решения узкоспециализированных задач используют распознавание определенных, заранее фиксированных шаблонов голосовых команд, а семантическая информация в представлении робота выражается в виде семантической карты, сопоставляющей речевые маркеры (например, номер комнаты) и связываемые с ними структуры данных (например, координаты комнаты), как это сделано в работе [6]. Существуют разработки более универсальных онтологий, позволяющих использовать представления многих различных процессов, действий, объектов и отношений в рамках одной модели (например, представленных в [7]). Подход к построению интерфейса, близкий по идеям к данному исследованию, предлагается авторами [8]. Ими разработаны система управления роботом и голосовой интерфейс, позволяющие роботу воспринимать различные команды на естественном языке (с ограничениями, накладываемыми предметной областью) за счет применения подхода на основе концептуальных зависимостей [9] и базы знаний с логическим выводом для обработки команды. В отличие от нее авторы статьи разрабатывали систему управления, скорее, из соображений создания единого механизма как для реализации внутренних поведенческих процедур робота, так и для связи их с интерфейсом оператора, особое внимание уделив поддержке мультимодального интерфейса, который может содержать и речевую информацию, и другую, получаемую с иных датчиков или устройств, например, направление взгляда, что определило выбор иных моделей и программных средств, лежащих в основе предлагаемой системы.

Разработанная система управления мобильной платформой объединяет знаковое представление мира робота на основе модели данных RDF (Resource Description Framework) [10] и голосовой интерфейс в единую систему, обрабатывающую как команды пользователя, так и автономное поведение робота. В статье описывается архитектура системы и демонстрируется применимость данного подхода на примере имитационной модели мобильного робота в среде Gazebo.

### Архитектура системы управления роботом

Знаковое представление модели мира робота основано на определении знака, введен-

ном Г.С. Осиповым [11]. Согласно этому подходу, знак – это набор из четырех компонентов: имени  $n$ , характеризующего связь знака с культурным контекстом; образа  $p$ , характеризующего набор признаков знака; значения  $m$ , характеризующего наблюдавшиеся связи знака с другими знаками; личностного смысла  $a$ , характеризующего опыт использования означаемой сущности субъектом. Можно формализовать образ  $p$  как множество выражений, каждое из которых представляет собой предикат – истинное высказывание об означаемой сущности; значение  $m$  и личностный смысл  $a$  можно представить в виде наборов правил вывода. Правило состоит из условия применимости – списка предикатов, истинных для применения правила, и результата применения – списка предикатов, которые станут истинными либо ложными после применения правила. Такое представление правил позволяет формализовать их с помощью языка описания задач планирования STRIPS, поэтому далее они называются STRIPS-правилами.

Разработанная семиотическая система управления роботом состоит из базы знаний, содержащей набор знаков, и внешних интерфейсных модулей, транслирующих команды внешних систем в образы знаков и вычисляющих требуемую реакцию робота по образам знаков. Так как знаковое представление данных не зависит от интерфейса, по которому они поступают в систему, рассмотрим в качестве примера исполнение роботом голосовых команд на естественном языке. Общая структура значимых для этого процесса частей системы показана на рисунке 1.

Сначала выполняется распознавание текста пользовательских команд с помощью стандартных средств перевода голоса в текст. Затем этот текст поступает на другой внешний сервис – лингвистический анализатор, задача которого – распознать характерные для русской речи компоненты предложения и представить содержимое текста в особом формате. При этом команды могут содержать лишнюю информацию или быть некорректными, так что, непосредственно передавать их базе знаний не следует; кроме того, их формат изначально несовместим с используемым знаковой моделью.

Помимо самой речевой информации, распознаватель команд робота также может принимать вместе с ней дополнительные данные, например, направление взгляда пользователя,

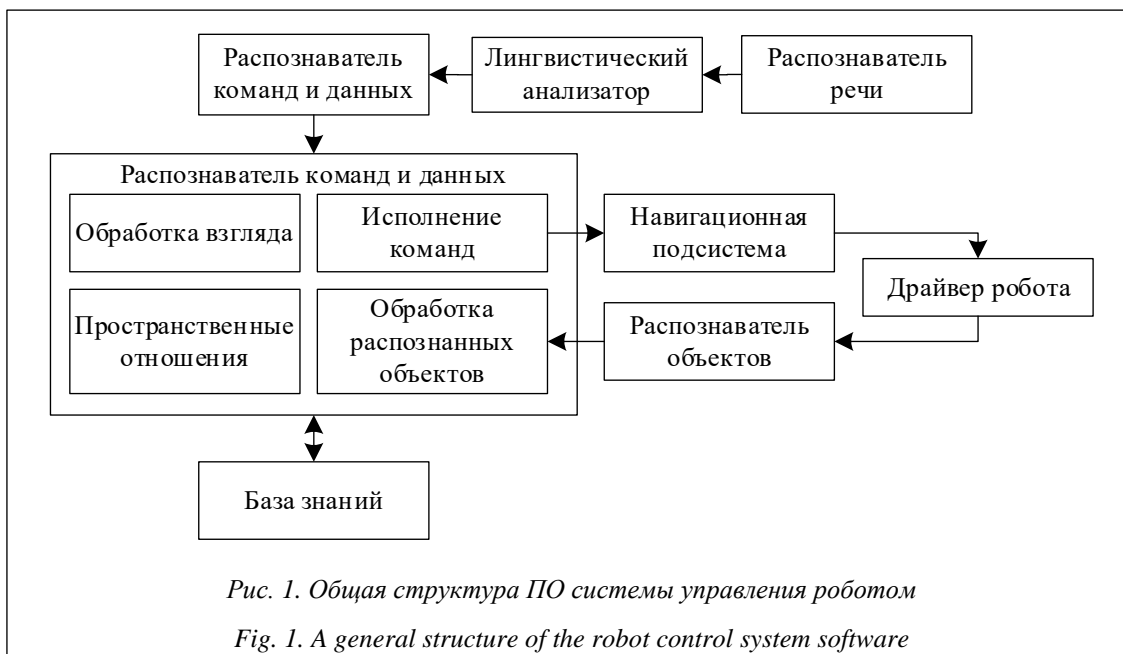


Рис. 1. Общая структура ПО системы управления роботом

Fig. 1. A general structure of the robot control system software

позволяющие уточнить контекст сообщения, определив объект интереса оператора. Сопоставление направления взгляда и известных роботу объектов в мире осуществляется отдельным модулем, который по сопоставлению данных о направлении взгляда и положению известных объектов способен сделать вывод о том, на что падает взгляд человека в текущий момент.

Для управления роботом голосовыми командами необходимо согласовать интерфейсы лингвистического анализатора (то есть программы, создающей формальное представление текста, изначально передаваемого на естественном языке) и системы управления роботом. Это преобразование выполняется в два этапа: перевод представленных в специализированном формате результатов разбора в единую для данной семиотической системы модель данных RDF и выделение информации для формирования команд. В модели RDF данные представляются в виде набора троек:  $\langle Subject, Predicate, Object \rangle$ .

Результатом работы распознавателя команд и данных является набор RDF-троек, которые можно интерпретировать как предикаты базы знаний модели мира робота. Например, для команды «поезжай к дому возле леса» может быть получен следующий разбор (рис. 2):

$\langle KI:Action \rangle \langle ki:type \rangle \langle ki:action \rangle .$   
 $\langle KI:Action \rangle \langle ki:actor \rangle \langle ki:me \rangle .$   
 $\langle KI:Action \rangle \langle ki:action\_type \rangle$   
 $\langle ki:move\_to \rangle .$

$\langle KI:Goal \rangle \langle ki:patient \rangle \langle KI:Goal \rangle .$   
 $\langle KI:Goal \rangle \langle ki:type \rangle \langle ki:house \rangle .$   
 $\langle KI:Goal \rangle \langle ki:near \rangle \langle KI:X \rangle .$   
 $\langle KI:X \rangle \langle ki:type \rangle \langle ki:forest \rangle .$

Интерпретировать эту команду можно так: «Действие *Action* заключается в перемещении меня (*me*), то есть робота, к объекту *Goal*, являющемуся домом и расположенному около объекта *X*, являющегося лесом». Здесь и далее названия переменных начинаются с заглавных букв, названия констант – со строчных. Данное разделение необходимо для работы правил вывода семиотической системы. В дальнейшем результат поступает в базу знаний для интерпретации.

Сама по себе семиотическая модель позволяет достаточно легко изменять поведение

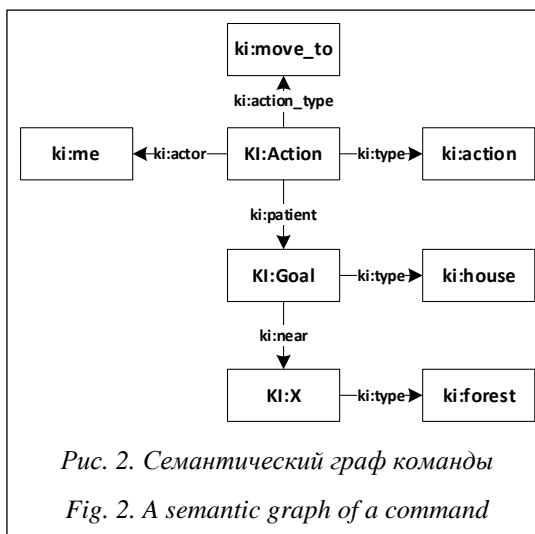
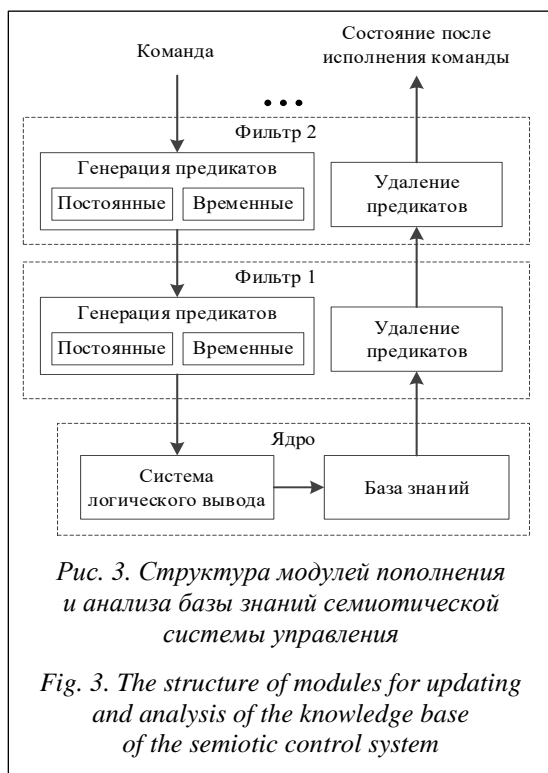


Рис. 2. Семантический граф команды

Fig. 2. A semantic graph of a command

системы и расширять ее возможности, однако без явного упорядочения этого процесса может начаться нежелательное увеличение сложности полученного ПО и, как следствие, усложнение сопровождения и модификации. Для предотвращения этого была предложена структура интерфейса ввода и порядка обработки информации для базы знаний и модулей, непосредственно работающих с ней, изображенная на рисунке 3.



Система разделена на ядро, выполняющее логический вывод и хранящее знания, и модули-фильтры. При загрузке системы каждый фильтр обязан сообщить:

- какие отношения между компонентами системы он планирует добавить в нее;
- какие из этих отношений не должны сохраняться в базе знаний надолго во избежание ее засорения промежуточными результатами большого количества команд;
- какие отношения могут быть удалены данным фильтром.

Фильтры выстраиваются в строгую последовательность (стек) в процессе инициализации. При выполнении команды алгоритм ее обработки выглядит следующим образом.

1. Принять команду (последовательность RDF-троек) от внешнего интерфейса.
2. Для каждой обрабатываемой процедуры (фильтра) в порядке, обратном порядку

их добавления, добавить к команде предикаты, при этом разделяя их на постоянные (будут сохранены в базе знаний) и временные (будут удалены из базы).

3. Выполнить логический вывод в ядре, сохранив постоянные предикаты в базе знаний.

4. Для каждой процедуры в порядке добавления убрать из базы предикаты в соответствии с логикой работы фильтра.

5. Убрать из базы временные предикаты этого фильтра.

6. Вернуть внешнему интерфейсу итоговое состояние базы знаний и результаты сопоставления переменных, если оно проводилось.

При такой организации системы, помимо ядра, были выделены следующие фильтры:

- исполнитель команд `instruction_processor`, отправляющий роботу считанные из базы знаний команды;
- обработчик пространственных отношений `space_rel_generator`, создающий нужные для выполнения команд отношения («около», «к западу от», «слева от» и т.п.);
- обработчик взгляда пользователя `gaze_relations_generator`, создающий для объектов признаки наличия взгляда на них;
- обработчик объектов, распознанных сенсорной системой робота, `recognition_processor`, вычисляющий их глобальные координаты по положению относительно сенсоров робота.

Тестовая команда «поезжай к дому возле леса» вызовет генерацию временных предикатов `near` («возле») для всех объектов, тип одного из которых – `house`, другого – `forest`; обрабатывать ее будет модуль `space_rel_generator`. Независимо от успеха этой генерации происходит попытка добавления данных из RDF в семиотическую систему. При этом для всех переменных, кроме объектов «действие» (`action`), среди известных базе знаний объектов должны найтись соответствия.

Если ни одной пары таких объектов не найдется (то есть нет данных о домах, стоящих возле леса), выполнение команды завершится ошибкой – «цель не найдена». Если вариантов таких домов больше одного, выбор будет осуществляться по эвристическим правилам; используемое правило выбирается из настроек системы и не меняется в процессе работы. В одном из вариантов, например, будет выбрана ближайшая цель, в другом – та, которая стоит перед роботом. Обработка эв-

ристик на текущий момент реализована не в виде фильтра, а как отдельная структура ядра, потому что используемые эвристики тесно связаны с процессом вывода. В дальнейшем можно вывести их из ядра в отдельный фильтр.

Пусть нужная пара объектов найдена, например, *house3* и *forest4*. Тогда после добавления данных в семиотическую систему в ней появится новый знак для индивидуального действия следующим образом:

$type(action1, action), actor(action1, me), action\_type(action1, move\_to), patient(action1, house3).$

Это значит, что роботу стало известно о необходимости выполнить действие такого рода. Приступить к его выполнению он может, применив STRIPS-правило «приступай к действию *move\_to*, если других действий нет» из множества *m* своего знака *me*, то есть

$move\_to$   
 $? type(X, action), actor(X, R), action\_type(X, move\_to), patient(X, Z), coord(Z, Pos), !completed(R, X), task(R, no)$   
 $+ task(R, X)$   
 $- task(R, no)$

Так как после добавления действия *action1* можно сопоставить переменным значения  $X = action1, R = me$  и  $Z = house3$ , при известных координатах дома *Pos* робот приступает к движению, то есть добавляет в свой образ выражение  $task(R = me, X = action1)$  и убирает из него выражение  $task(me, no)$ .

При обратном ходе команды модуль *instruction\_processor* заметит, что робот принял задачу, и отдаст навигационной подсистеме соответствующую команду перемещения в точку. При этом, так как конкретное место может оказаться для робота непроходимым, автоматически выполняется поиск ближайшей к месту назначения достижимой для робота точки (рис. 4).

Окончание перемещения фиксируется модулем *instruction\_processor* путем добавления к образу знака *action1* выражения  $finished(action1, me)$ . После этого STRIPS-правило

$move\_to\_end$   
 $? task(R, X), action\_type(X, move\_to), patient(X, Z), !completed(R, X), finished(X, R)$   
 $+ completed(R, X), task(R, no)$   
 $- task(R, X)$

приведет к тому, что робот пометит данную задачу как выполненную и будет готов к следующей команде, так как в его образ будет добавлено выражение  $task(me, no)$ .

Последовательность выполнения операций (то есть команды наподобие «поезжай к дереву, затем двигайся к дому возле камня») поддерживается без добавления особого обрабатывающего модуля путем введения специального варианта действия *delayed\_action*. В процессе формирования команд по данным лингвистического сервера для действий, выполняющихся после других действий, указывается тип *delayed\_action* вместо *action*. Кроме того, к такому действию добавляется дополнительный признак *after*, указывающий, какое именно действие является предшественником текущего. Например, команда из двух действий «поезжай к лесу, потом к человеку возле дома» может выглядеть следующим образом:

$\langle KI:A1 \rangle \langle ki:type \rangle \langle ki:action \rangle .$   
 $\langle KI:A1 \rangle \langle ki:actor \rangle \langle ki:me \rangle .$   
 $\langle KI:A1 \rangle \langle ki:action\_type \rangle \langle ki:move\_to \rangle .$   
 $\langle KI:A1 \rangle \langle ki:patient \rangle \langle KI:Goal1 \rangle .$   
 $\langle KI:Goal1 \rangle \langle ki:type \rangle \langle ki:forest \rangle .$   
 $\langle KI:A2 \rangle \langle ki:after \rangle \langle KI:A1 \rangle .$   
 $\langle KI:A2 \rangle \langle ki:type \rangle \langle ki:delayed\_action \rangle .$   
 $\langle KI:A2 \rangle \langle ki:actor \rangle \langle ki:me \rangle .$   
 $\langle KI:A2 \rangle \langle ki:action\_type \rangle \langle ki:move\_to \rangle .$   
 $\langle KI:A2 \rangle \langle ki:patient \rangle \langle KI:Goal2 \rangle .$   
 $\langle KI:Goal2 \rangle \langle ki:type \rangle \langle ki:human \rangle .$   
 $\langle KI:Goal2 \rangle \langle ki:near \rangle \langle KI:H \rangle .$   
 $\langle KI:H \rangle \langle ki:type \rangle \langle ki:house \rangle .$

В этом примере действия *A1* и *A2* связаны отношением *after*, задающим порядок их исполнения. На данный момент поддерживается только такое упорядочение, при котором

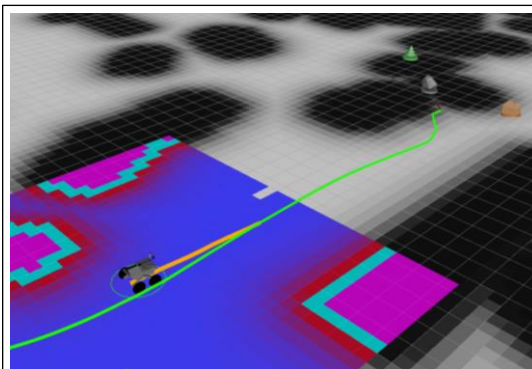


Рис. 4. Перемещение модели робота к недоступному объекту «камень» (серая фигура), интерпретируемое как перемещение в достаточно близкую к нему достижимую точку

Fig. 4. The movement of the robot model to an unreachable object "stone" is interpreted as moving to a sufficiently close reachable point

действия должны формировать по предикату *after* однонаправленный связный список.

### Языковой интерфейс робота

На вход робот принимает сообщения в виде списка RDF-триплетов, описывающих команду или информационное сообщение для робота. При этом для корректной интерпретации роботом должны соблюдаться определенные требования к содержанию сообщения: в частности, информационные сообщения (добавляющие новые факты в модель мира робота) не могут содержать действия робота, информация в любом сообщении должна быть непротиворечивой, должны использоваться понятия, содержащиеся в специальном словаре модели мира (другие понятия в принципе допустимы, но сообщение может быть некорректно обработано из-за их отбрасывания). Исходные данные для языкового интерфейса образуют специальным образом описанную семантическую сеть, представленную в формате JSON, что делает их весьма удобными для преобразования [12].

Обработка результатов лингвистического разбора проходит в два этапа. На первом этапе выделяются только те данные, которые в дальнейшем могут использоваться в правилах. Другие данные, например, неиспользуемые идентификаторы и метаданные, отбрасываются для облегчения дальнейшей обработки. Некоторым числам сопоставляются словесные представления, используемые на следующих этапах обработки. Данный этап можно рассматривать как перевод команды из одного представления в другое без потерь информации, значимой для интерпретации команды. В качестве целевого представления используется модель данных RDF в формате N-Triples.

Второй этап заключается в последовательном пополнении набора RDF-троек на основе набора правил интерпретации фигурирующих в них сущностей в понятную роботу команду. Этот этап необходим по той причине, что робот не способен воспринимать лингвистический разбор как команду: ему необходимо перевести языковые конструкции в непосредственные объекты, типы команд и их аргументы, причем на основе известных ему понятий. Для этого применяются правила к RDF-записям для получения команды в RDF-формате.

Каждое правило выражает соответствие между понятиями, используемыми при лингвистическом разборе, и понятиями, используемыми в системе команд робота. Правило имеет вид: «Если в наборе имеется некоторое множество троек  $A$  и не имеется никаких троек из множества  $B$ , то следует добавить некоторое множество троек  $C$  и убрать некоторое множество троек  $D$ ». Например, для получения сведений о некоторой постройке может быть применено правило: «Если в наборе есть тройка  $\langle X, lemma, строение \rangle$  и нет тройки  $\langle X, type, house \rangle$ , следует добавить тройки  $\langle X, type, house \rangle$  и  $\langle X, type, object \rangle$ ».

Процесс обработки выполняется итеративно: сначала правила из заданного набора применяются ко всем тройкам, поступившим на обработку, затем как к ним, так и к полученным на предыдущем шаге тройкам и т.д. до того момента, когда новых троек получить не удастся. Результатом этого этапа становятся все тройки, добавленные к начальному набору на любой итерации, и только они (исходные тройки, полученные на этапе 1, к ним не добавляются, хотя и используются в процессе вывода новых).

Созданное ПО поддерживает получение команд с несколькими вариантами разбора (альтернативами), что может быть полезно в ситуациях, когда семантический разбор неоднозначен. Обработка альтернатив поддерживается только для команд (информационные сообщения нельзя передавать в виде альтернатив). Выбор альтернативы осуществляется на стороне робота исходя из известных ему данных. В простейшем случае выбирается альтернатива с наибольшим значением пригодности разбора или при его отсутствии может быть выбран произвольный разбор. Значение пригодности разбора определяется на стороне лингвистического сервера во время разбора текста команды и передается в специальном RDF-триплете как часть сообщения, обозначающая пригодность всей альтернативы.

### Модель мира робота

Модель мира робота состоит из знаков, которые соответствуют объектам среды и выполняемым роботом действиям. Интерпретация модели выполняется сторонними модулями, которые также отвечают за подачу на нее тех видов информации, логический вывод

которых затруднен. Такой подход позволяет единым образом обрабатывать команды оператора, автономную реакцию робота на изменения в мире, информационные сообщения от человека, высокоуровневый выход модулей обработки данных с датчиков (распознанные объекты, данные о взгляде пользователя). Подробное описание использованного метода моделирования приведено в [13].

В модель входят все типы известных роботов объектов, причем для любой сущности можно указать более одного типа с помощью выражений *rdf:type* (или интерпретируемого аналогично *ki:type*). Типы образуют иерархию с помощью связей *rdf:subclass*. Операции переноса меток типов на объекты осуществляются автоматически специальными правилами модели мира.

Модуль исполнения команд *instruction\_processor* по описанию текущей задачи отдает нужные команды подсистемам управления роботом, а также контролирует успешность их выполнения. Переключение задач осуществляется при этом обычными правилами модели мира без создания особых механизмов.

Модуль генерации пространственных отношений *space\_rel\_generator* пришлось выделить, потому что хранение их полного набора резко замедляло распознавание новых команд на модели робота (до времени порядка 10–20 сек.). Интерфейс этого модуля с точки зрения семиотической системы состоит из набора

- постоянных отношений: близость робота к важному объекту *near\_important*; оно же может быть удалено модулем при необходимости;

- временных отношений: все отношения относительного и абсолютного расположения объектов: *to\_north\_from*, *behind\_of*, *near* и др. (при этом реально генерируются только те отношения, которые имеются в пришедшем запросе, то есть могут повлиять на результат).

Генерация осуществляется по известной карте объектов, хранимой в базе знаний семиотической системы. Для определения отношений используется клеточное представление окрестностей объектов: каждому из них сопоставляется клеточное поле, в котором степень релевантности ячейки запросу показывается величиной от 0 до 1. Операции с такими полями реализованы в специальной библиотеке. Результатом операции может быть вывод либо о наличии отношения

(«А действительно расположен к северу от В»), либо о координатах требуемой точки («Точка, расположенная за объектом С, имеет координаты (14,13)»). Все временные пространственные отношения, полученные таким образом, не сохраняются в базе знаний.

Система поддерживает работу с дополнительным признаком объектов, связанным с наличием на них взгляда пользователя. Для передачи ей такой информации следует отправить сообщение об объекте типа *gaze\_data*:

```
<KI:X> <rdf:type> <ki:gaze_data> .
<KI:X> <ki:x> "1.0" .
<KI:X> <ki:y> "-11.1" .
<KI:X> <ki:z> "0.3" .
```

Координаты линии взгляда пользователя при этом задаются относительно камеры (сенсора, через который оператор должен видеть данную информацию). Модуль *gaze\_relations\_generator* автоматически создает у объектов, находящихся на этой линии, признак *ki:by\_gaze\_line*, а у ближайшего из них – *ki:gaze\_on*. Этот признак может быть далее использован, например, для выбора объекта в эвристике.

Вся подаваемая на модель информация считается истинной. Проверки противоречивости выполняются только специальными сторонними модулями. Таким образом, например, обновляются координаты объектов и точек взгляда. Неоднозначности в информации разрешаются отдельной подсистемой эвристического выбора цели, интегрированной в ядро [14].

Предложенный подход позволяет достичь полной унификации по данным для любых типов входных и выходных данных. Кроме того, он легко интерпретируется пользователем и основан на широко распространенных известных подходах, что облегчает первоначальное освоение системы.

### Имитационная модель мобильной платформы

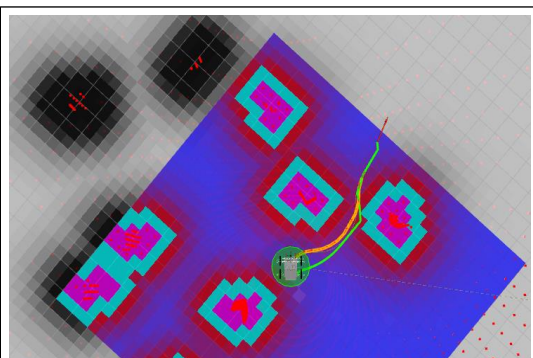
Описанная модель мира реализована в виде приложения, разработанного на базе фреймворка ROS (Robot Operating System) – распространенной системы для интеграции различных программных модулей роботов. Для симуляции робота используется среда моделирования Gazebo. Конфигурация всех систем выполнена с расчетом на то, чтобы в будущем обеспечить возможность легкого



переноса ПО, использованного для модели робота, на аппаратный прототип робота.

В качестве проблемы при отработке системы рассматривалось удаленное управление с поддержкой полуавтономного режима, в ходе которого робот должен выполнять различные задачи, такие как перемещение по маршруту, сбор данных о местности и выполнение целевых измерений в ключевых точках.

Робот функционирует в модели большого открытого окружения, имитирующего естественный ландшафт. Навигация осуществляется штатными средствами узла ROS move\_base [15], то есть парой планировщиков – локальным и глобальным. Глобальный планировщик строит примерный маршрут для достижения указанной точки, а перемещение к ней выполняет локальный планировщик TEB (Timed-Elastic-Band) [16], оптимизирующий траекторию движения с учетом ограничений допустимой скорости и ускорения для обхода препятствий при следовании по маршруту (рис. 5).



*Рис. 5. Использование локального планировщика TEB для оптимизации траектории вблизи препятствий*

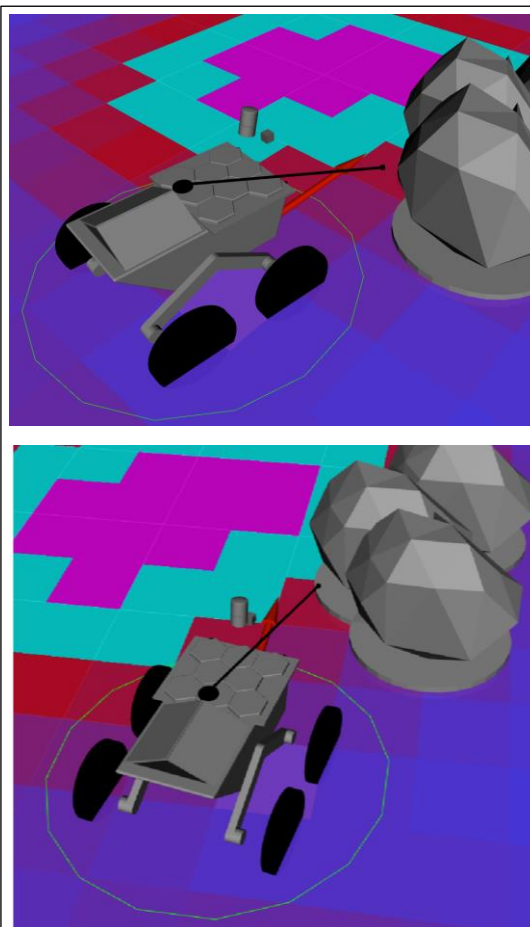
*Fig. 5. Using the TEB local planner to optimize the trajectory in proximity to obstacles*

Позиция робота определяется абсолютно точно с использованием специального плагина Gazebo, хотя для реального робота для позиционирования потребовалась бы разработка специальной системы с применением разных классов сенсорных данных и знаний об окружении. Навигационная подсистема обладает начальной картой препятствий, но в процессе работы робота она пополняется сведениями сканирующего дальномера (модуля velodyne\_3d) и данными о распознанных объектах с установленной на робота камеры.

Для взаимодействия с окружающим миром робот оборудован манипулятором с двумя степенями свободы, способным имитировать исследование объектов путем выдвижения к их поверхности штока с установленной моделью видеокамеры (рис. 6).

Для тестирования работы робота необходимо дополнительно промоделировать выполнение им некоторых понятных пользователю сложных поведенческих процедур. Для этого реализованы операции поиска объекта и их фотографирования.

Разработан набор типовых сценариев применения робота, позволяющих оценить удобство его использования и качество исполнения команд. Каждый сценарий представляет собой словесную постановку решаемой роботом задачи, а также описание ожидаемых команд оператора для ее выполнения и ожидаемых действий робота при получении команд



*Рис. 6. Процесс исследования камней манипулятором модели робота*

*Fig. 6. The process of stone probing by the robot model manipulator*



от оператора. В тестовый набор сценариев входят:

- использование команд тактического управления: движение вперед-назад на заданное расстояние или непрерывно, повороты налево-направо на заданный угол или непрерывно, повороты и движения по абсолютному направлению (азимуту) («двигайся вперед десять метров», «поверни на северо-запад», «поворачивай налево»);

- перемещение к объектам по описанию; в описании могут входить свойства как самих объектов («поезжай к дереву»), так и других объектов, связанных с нужными («поезжай к лесу возле дома»);

- перемещение к объектам по взгляду; реализуется так же, как предыдущий вариант, но для цели указывается специфический признак *gaze\_on(me, Goal)*;

- патрулирование и исследование местности; робот при этом двигается по заданному маршруту, отвлекаясь на объекты, имеющие признак *important(Object, true)*;

- исследование объектов манипулятором; робот при этом приближает свой манипулятор к объекту, имитируя сбор образцов или фотографирование с близкого расстояния.

Данные сценарии были протестированы, и реализованный подход показал свою работоспособность. В частности, при экспериментах были отмечены следующие преимущества избранного подхода по сравнению с традиционным:

- большая пригодность для движения по неровному грунту; при традиционных интерфейсах управления движение по настолько неровной местности требует постоянного внимания оператора и мелких коррекций, поэтому очень некомфортно, что усугубляется укачиванием;

- более простая навигация при отсутствии ориентиров и однообразном окружении;

- оператору не нужно помнить детали об устройстве мира – вместо точного пути к объекту можно указать его характерный признак;

- не нужно использовать руки для управления.

В то же время были отмечены и некоторые недостатки созданной системы:

- значительные проблемы при указании однозначных признаков объектов, малое количество распознаваемых признаков;

- сложности с выбором признаков, различных для системы распознавания объек-

тов робота и в то же время достаточно простых, универсальных и различных при нахождении на местности;

- замедление работы при слишком большом количестве объектов: для примененной тестовой платформы число объектов больше 30 приводило ко времени обработки одной команды более 1 сек., что ощущалось как некомфортная задержка.

## Заключение

Были разработаны система управления и внешний интерфейс оператора на основе знаковой модели мира робота с использованием модели данных RDF. Разработанная модель мира робота и алгоритмы ее обработки позволяют совместить обработку данных и команд как от голосового интерфейса оператора, так и от автономных элементов системы. При этом обеспечиваются поддержка ограниченного ряда человеческих понятий о некоторых объектах, действиях, атрибутах и отношениях между объектами, доуточнение команды другими данными (например, информацией о положении взгляда оператора), использование логического вывода для дополнения модели мира и поддержка команд с пространственными отношениями.

Алгоритмы апробированы на имитационной модели мобильного робота, работающего на открытой местности, в среде Gazebo и продемонстрировали возможность управления роботом с помощью голосового интерфейса при отдаче команд движения к объектам и в области, автономную реакцию робота на особые ситуации, отработку протяженных поведенческих операций, устранение неоднозначностей команды за счет разрешения пространственных отношений и учета направления взгляда оператора. Из недостатков системы следует отметить малое количество практически применимых признаков объектов и замедление работы при необходимости обработки большого количества объектов. Система в текущей реализации не поддерживает обработку противоречивой информации. Ожидается, что запланированный перевод ядра системы на когнитивную архитектуру Soag позволит повысить производительность операций логического вывода.

*Работа выполнена при частичной финансовой поддержке НИЦ «Курчатовский институт» (приказ № 1055 от 02.07.2020).*

**Литература**

1. Zander S., Ahmed N., Frank M.T. A survey about the usage of semantic technologies for the description of robotic components and capabilities. Proc. SamI40 Workshop at I-KNOW, 2016. URL: <http://ceur-ws.org/Vol-1793/paper2.pdf> (дата обращения: 29.05.2021).
2. Saeed N.T.M., Kazerouni M.F., Fathi M., Kuhnert K.-D. Robot semantic protocol (RoboSemProc) for semantic environment description and human–robot communication. *Int. J. of Social Robotics*, 2020, vol. 12, no. 2, pp. 599–612. DOI: 10.1007/s12369-019-00580-5.
3. Dolgova E.V., Fayzrakhmanov R.A., Kurushin D.S. Decision-making in autonomous mobile robot control system based on active semantic network. Proc. XX Int. Conf. SCM, 2017, pp. 221–223. DOI: 10.1109/SCM.2017.7970543.
4. Joo S.-H., Manzoor S., Rocha Yu.G., Bae S.-H. et al. Autonomous navigation framework for intelligent robots based on a semantic environment modeling. *Applied Sciences*, 2020, vol. 10, p. 3219. DOI: 10.3390/app10093219.
5. Sowa J.F. Semantic networks. In: *Encyclopedia of Artificial Intelligence*, 1987. URL: <http://www.jfsowa.com/pubs/semnet.pdf> (дата обращения: 29.05.2021).
6. Zhao C., Mei W., Pan W. Building a grid-semantic map for the navigation of service robots through human–robot interaction. *Digital Communications and Networks*, 2015, vol. 1, no. 4, pp. 253–266. DOI: 10.1016/j.dcan.2015.09.002.
7. Olszewska J.I. et al. Ontology for autonomous robotics. Proc. Int. Symposium RO-MAN, 2017, pp. 189–194. DOI: 10.1109/ROMAN.2017.8172300.
8. Savage J. et al. Semantic reasoning in service robots using expert systems. *Robotics and Autonomous Systems*, 2019, vol. 114, pp. 77–92. DOI: 10.1016/j.robot.2019.01.007.
9. Шенк Р. Обработка концептуальной информации; [пер. с англ.]. М.: Энергия, 1980. 360 с.
10. W3C. RDF 1.1 Concepts and Abstract Syntax. URL: <https://www.w3.org/TR/rdf11-concepts/> (дата обращения: 29.05.2021).
11. Осипов Г.С., Панов А.И., Чудова Н.В., Кузнецова Ю.М. Знаковая картина мира субъекта поведения. М.: Физматлит, 2018. 264 с.
12. Батура Т.В. Семантический анализ и способы представления смысла текста в компьютерной лингвистике // Программные продукты и системы. 2016. Т. 29. № 4. С. 45–57. DOI: 10.15827/0236-235X.116.045-057.
13. Карпов В.Э., Малахов Д.Г., Московский А.Д., Ровбо М.А., Сорокоумов П.С., Величковский Б.М., Ушаков В.Л. Архитектура системы управления для инвалидной коляски: на пути к многофункциональному роботу с нейробиологическими интерфейсами // Современные технологии в медицине. 2019. Т. 11. № 1. С. 90–102. DOI: 10.17691/stm2019.11.1.11.
14. Sorokoumov P.S., Rovbo M.A., Moscovsky A.D., Malyshev A.A. Robotic wheelchair control system for multimodal interfaces based on a symbolic model of the world. In: *Smart Electromechanical Systems*, 2021, pp. 163–183. DOI: 10.1007/978-3-030-68172-2\_14.
15. ROS.org. Navigation. URL: <http://wiki.ros.org/navigation?distro=jade> (дата обращения: 29.05.2021).
16. Rösmann C., Hoffmann F., Bertram T. Integrated online trajectory planning and optimization in distinctive topologies. *Robotics and Autonomous Systems*, 2017, vol. 88, pp. 142–153. DOI: 10.1016/j.robot.2016.11.007.

**Semiotic control system for a mobile robotic platform**

**M.A. Rovbo**<sup>1</sup>, *Research Engineer, rovbo@gmail.com*

**P.S. Sorokoumov**<sup>1</sup>, *Research Engineer, petr.sorokoumov@gmail.com*

<sup>1</sup> *National Research Centre “Kurchatov Institute”, Moscow, 123182, Russian Federation*

**Abstract.** The paper considers the problem of combining a control system based on a semiotic model of a robot’s world and human-machine interfaces, in particular, a voice interface, into a single system that processes both user commands and the robot’s autonomous behavior.

The developed system allows controlling a robot using a voice interface, executing extended behaviors, eliminating command ambiguities by resolving spatial relationships and taking into account the operator's gaze direction. This is achieved by integrating all the information necessary for making a decision and choosing the robot's action into the world model, applying logical inference to supplement it, and using heuristics to eliminate ambiguities in the speech command when there are not enough known facts for this. The decision-making system based on this world model also provides the robot's autonomous response to special situations.

The paper describes the developed system architecture and demonstrates the applicability of this approach using a simulated model of a mobile robot in Gazebo. The simulation showed the possibility of controlling a mobile platform using the developed system and the indirect control proved to be more ergonomic. Although, it should be noted that the computational load increases substantially when it is necessary to process a large number of objects in the world and it is impossible to process conflicting information in the current implementation. The computational load is partially compensated by algorithms that make it possible to infer only the information necessary to process the current command.

**Keywords:** semiotic models; control systems; robotics; human-machine interfaces; voice control.

**Acknowledgements.** *The work was partially supported by NRC "Kurchatov Institute" (order no. 1055, 02.07.2020).*

### References

1. Zander S., Ahmed N., Frank M.T. A survey about the usage of semantic technologies for the description of robotic components and capabilities. *Proc. SamI40 Workshop at I-KNOW*, 2016. Available at: <http://ceur-ws.org/Vol-1793/paper2.pdf> (accessed May 29, 2021).
2. Saeed N.T.M., Kazerouni M.F., Fathi M., Kuhnert K.-D. Robot semantic protocol (RoboSemProc) for semantic environment description and human-robot communication. *Int. J. of Social Robotics*, 2020, vol. 12, no. 2, pp. 599–612. DOI: 10.1007/s12369-019-00580-5.
3. Dolgova E.V., Fayzrakhmanov R.A., Kurushin D.S. Decision-making in autonomous mobile robot control system based on active semantic network. *Proc. XX Int. Conf. SCM*, 2017, pp. 221–223. DOI: 10.1109/SCM.2017.7970543.
4. Joo S.-H., Manzoor S., Rocha Yu.G., Bae S.-H. et al. Autonomous navigation framework for intelligent robots based on a semantic environment modeling. *Applied Sciences*, 2020, vol. 10, p. 3219. DOI: 10.3390/app10093219.
5. Sowa J.F. Semantic networks. In: *Encyclopedia of Artificial Intelligence*, 1987. Available at: <http://www.jfsowa.com/pubs/semnet.pdf> (accessed May 29, 2021).
6. Zhao C., Mei W., Pan W. Building a grid-semantic map for the navigation of service robots through human-robot interaction. *Digital Communications and Networks*, 2015, vol. 1, no. 4, pp. 253–266. DOI: 10.1016/j.dcan.2015.09.002.
7. Olszewska J.I. et al. Ontology for autonomous robotics. *Proc. Int. Symposium RO-MAN*, 2017, pp. 189–194. DOI: 10.1109/ROMAN.2017.8172300.
8. Savage J. et al. Semantic reasoning in service robots using expert systems. *Robotics and Autonomous Systems*, 2019, vol. 114, pp. 77–92. DOI: 10.1016/j.robot.2019.01.007.
9. Schank R.C. *Conceptual Information Processing*. Elsevier Publ., 1975, 382 p. (Russ. ed.: Moscow, 1980, 360 p.).
10. *W3C. RDF 1.1 Concepts and Abstract Syntax*. Available at: <https://www.w3.org/TR/rdf11-concepts/> (accessed May 29, 2021).
11. Osipov G.S., Panov A.I., Chudova N.V., Kuznetsova Yu.M. *Sign-Based World Model for a Behavior Subject*. Moscow, 2018, 264 p. (in Russ.).
12. Batura T.V. Semantic analysis and methods of text meaning representation in computer linguistics. *Software and Systems*, 2016, vol. 29, no. 4, pp. 45–57. DOI: 10.15827/0236-235X.116.045-057 (in Russ.).
13. Karpov V.E., Malakhov D.G., Moscovsky A.D., Rovbo M.A., Sorokoumov P.S., Velichkovsky B.M., Ushakov V.L. Architecture of a wheelchair control system for disabled people: Towards multi-functional robotic solution with Neurobiological interfaces. *Modern Technologies in Medicine*, 2019, vol. 11, no. 1, pp. 90–102. DOI: 10.17691/stm2019.11.1.11 (in Russ.).
14. Sorokoumov P.S., Rovbo M.A., Moscovsky A.D., Malyshev A.A. Robotic wheelchair control system for multimodal interfaces based on a symbolic model of the world. In: *Smart Electromechanical Systems*, 2021, pp. 163–183. DOI: 10.1007/978-3-030-68172-2\_14.

15. *ROS.org. Navigation*. Available at: <http://wiki.ros.org/navigation?distro=jade> (accessed May 29, 2021).

16. Rösmann C., Hoffmann F., Bertram T. Integrated online trajectory planning and optimization in distinctive topologies. *Robotics and Autonomous Systems*, 2017, vol. 88, pp. 142–153. DOI: 10.1016/j.robot.2016.11.007.

#### Для цитирования

Ровбо М.А., Сорокоумов П.С. Знаковая система управления мобильной платформой // Программные продукты и системы. 2021. Т. 34. № 4. С. 543–554. DOI: 10.15827/0236-235X.136.543-554.

#### For citation

Rovbo M.A., Sorokoumov P.S. Semiotic control system for a mobile robotic platform. *Software & Systems*, 2021, vol. 34, no. 4, pp. 543–554 (in Russ.). DOI: 10.15827/0236-235X.136.543-554.