

УДК 004.896

# МОБИЛЬНЫЕ РОБОТЫ В МЕЖДУНАРОДНОМ СТАНДАРТЕ ROBOCUP SSL: ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

**Б.В. Викторов**

*Президентский физико-математический лицей № 239 / СПбГУ*

Россия, 191028, Санкт-Петербург, ул. Кирочная, 8 / Россия, 199034, Санкт-Петербург,  
Университетская наб., 7–9  
E-mail: boris@victorov.com

**А.Д. Мещеряков**

*Президентский физико-математический лицей № 239 / Национальный исследовательский  
университет ИТМО*

Россия, 191028, Санкт-Петербург, ул. Кирочная, 8 / Россия, 197101, Санкт-Петербург,  
Кронверкский пр., 49, лит. А  
E-mail: sashamesheriyakov919@gmail.com

**М.М. Липкович**

*Институт проблем машиноведения РАН / СПбГУ*

Россия, 199178, Санкт-Петербург, Большой проспект Васильевского острова, 61 / Россия, 199034,  
Санкт-Петербург, Университетская наб., 7–9  
E-mail: lipkovich.mikhail@gmail.com

**А.М. Ярмолинский**

*Институт Проблем Машиноведения РАН / СПбПУ Петра Великого*

Россия, 199178, Санкт-Петербург, Большой проспект В.О., 61 / Россия, 195251, Санкт-Петербург,  
ул. Политехническая, 29  
E-mail: yarmolinskiyam@gmail.com

**Ключевые слова:** Программное обеспечение, футбол роботов, RoboCup, мультиагентное управление.

**Аннотация:** В докладе дается обзор программного обеспечения для автономной игры в футбол по правилам RoboCup SSL, разработанного участниками команды SPbUnited, включающей учащихся и преподавателей вузов Санкт-Петербурга и Президентского физ.-мат. лицея №239. Описаны основные элементы ПО, связь между ними и логика стратегии.

## 1. Введение

Соревнования RoboCup Small Size League (SSL) являются состязанием роботов футболистов. Наибольший интерес в данных соревнованиях вызывают задачи мультиагентного управления. Для реализации и тестирования различных методов и алгоритмов был разработан комплекс программ, описываемых в данном докладе. Роботы в данной лиге представляют собой омни-колесных роботов, размеры которых должны быть ограничены цилиндром 0.18 м в ширину и 0.15 м в высоту. В игре участвуют две команды, в каждой из которых должно быть не более шести роботов, один из которых должен быть вратарем. Правила похожи на правила обычного футбола с рядом корректировок под особенности механики роботов [1]. Одной из особенностей рассматриваемой лиги является высокая скорость игры: скорость мяча ограничена сверху 6.5 м/с, что создает потребность в разработке программного обеспечения, способного быстро обрабатывать входящие сигналы с видеокамер, закрепленных над полем, и синтезировать управление на основе распознанной с кадров информации о положениях игроков и мяча. Помимо данных с видеокамер, на сервер команд также приходят данные от рефери, которые должны быть обработаны с самым высоким приоритетом, так как неисполнение команд от рефери приводит к нарушению правил.

## 2. Схема взаимодействия элементов ПО

Программное обеспечение команды SPbUnited состоит из 4-х элементов: SSL-Vision, LARCmaCS, Strategy-bridge и кода, реализующего стратегию игры. Они взаимодействуют следующим образом. От SSL-Vision получается пакет данных. LARCmaCS получает эти пакеты и передаёт их strategy-bridge. strategy-bridge отдаёт данные коду стратегии. Код стратегии после получения новых данных и подсчёта управлений для роботов передаёт их обратно в strategy-bridge откуда они переходят обратно в LARCmaCS, который передаёт их по сети на адрес станции связи. Станция связи подключена к сети и имеет фиксированный адрес. Станция обновляет локальную информацию роботов, при получении пакета данных (рис. 1).

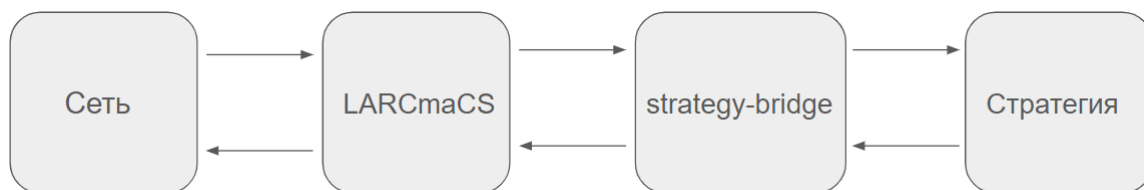


Рис. 1. Схема взаимодействия элементов ПО

## 3. Элементы ПО

### 3.1. SSL-Vision

Данные с видеокамер поступают на сервер организаторов, где из этих изображений извлекается информация об игре. Сервер отправляет пакеты с

данными о роботах, мяче, состоянии игры и геометрии поля на сервера участникам. Является open-source разработкой [9].

### 3.2. LARCmaCS

LARCmaCS - компонент, написанный на C++, который занимается получением пакетов с информацией об игре от организаторов и передаёт её компоненту strategy-bridge для синтеза управления. Далее LARCmaCS осуществляет передачу полученных от strategy-bridge управляющих команд на самих роботов. Таким образом, LARCmaCS отвечает за все низкоуровневое взаимодействие, инкапсулируя его от разработчиков стратегии. Помимо этого, LARCmaCS предоставляет пользователю графический интерфейс пользователя (рис. 2).



Рис. 2. Графический интерфейс LARCmaCS

### 3.3. Strategy-bridge

С целью предоставления для разработчиков инструментов для высокоуровневого описания стратегии был реализован компонент, названный strategy-bridge (далее «бридж»), написанный на языке Python. Основная абстракция бриджа - это процессор, представляющий собой вычислительную единицу, осуществляющую заданные пользователем действия в бесконечном цикле с настраиваемым шагом дискретизации. Например, может быть процессор, который потребляет пакеты SSL-Vision от LARCmaCS и преобразовывает их или процессор, который отправляет результирующие управляющие сигналы на LARCmaCS с целью отправки их роботам. Каждый процессор может потреблять данные, которые были созданы другими процессорами и делиться своими собственными. Были реализованы шаблоны процессоров, которые позволяют вызывать код написанный на языке MATLAB, часто используемым исследователями в области теории управления. Результирующие управляющие сигналы публикуются в топике ZeroMQ с использованием библиотеки ruzmq [7] для последующего использования модулем связи с роботом.

Все данные, которые должны быть поделены между различными процессорами проходят через структуру в памяти, называемую «шиной данных» (data bus). Все хранимые данные разбиваются на топики, где каждый топик представляет собой

очередь ограниченного размера. Пользователь может настраивать размер каждого конкретного топика. Когда топик достигает предельного размера, происходит вытеснение старых данных.

Каждый процессор может писать в один или несколько топиков, принадлежащих ему, а также читать из любых других топиков.

Было реализовано несколько режимов работы процессоров: с помощью асинхронных задач и библиотеки `asuncio` [8], а также в режиме многопроцессорности, где каждый процессор выполняется на отдельном процессе операционной системы. Выбор того или иного режима должен быть обусловлен ресурсами сервера, на котором происходит синтез управления, количества процессоров, которые должны выполняться одновременно, времени исполнения одного шага вычислений процессора, а также допустимыми задержками. Также имеет значение объем данных, который пишется и читается процессорами, поскольку в случае многопроцессорного режима происходит передача информации между различными процессами, занимающая больше времени чем передача информации внутри одного процесса.

### 3.4. Код стратегии

Получает на вход информацию о состоянии игры, а на выходе даёт управление для роботов. Изначально использовался Matlab, но позже по ряду соображений был выбран python. Есть 2 основных состояния: атака и защита. При защите роботы стараются максимально уменьшить количество возможных действий для соперника. Один из роботов пытается перехватить мяч, остальные либо перекрывают возможность паса, либо стоят в «стенке» у ворот (рис. 3 (левый)). При атаке роботы стоят в некоторой формации, как показано на (рис. 3 (правый)). В зависимости от ситуации атакующий робот с мячом либо нанесёт удар по воротам (рис. 4 (левый)), либо отдаст пас (рис. 4 (правый)).

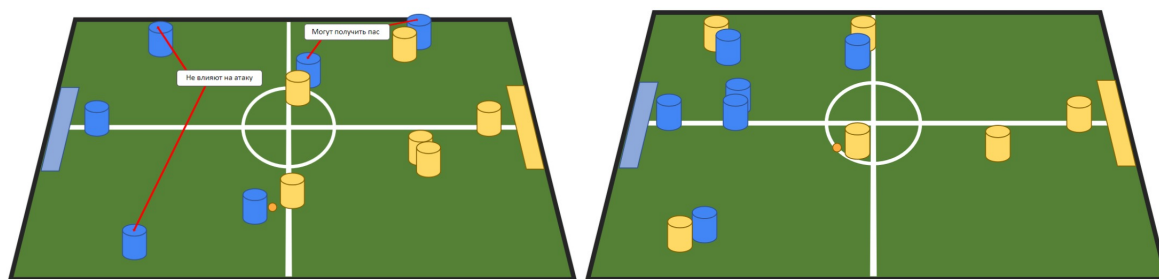


Рис. 3. Формации защиты и атаки

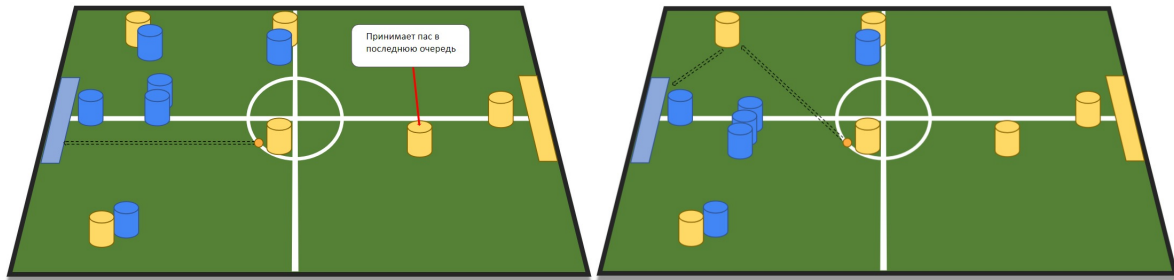


Рис. 4. Варианты атакующего робота с мячом

## 4. Заключение

В данной работе были рассмотрены элементы ПО, которые были созданы для исследований методов и алгоритмов мультиагентного управления. Команда SPbUnited благодарит за поддержку ПАО «Газпром нефть», ИПМаш РАН, благотворительный фонд «Финист», ООО «НПО «СтарЛайн».

## Список литературы

1. About RoboCup Small Size League. <https://ssl.robocup.org/about/> (дата обращения: 26.12.2023.)
2. RoboCup Small Size League (SSL) Principles and Goals. <https://robocup-ssl.github.io/ssl-goals/sslgoals.pdf> (дата обращения: 26.12.2023.)
3. RoboCup Small Size League (SSL) Rules. <https://ssl.robocup.org/rules/> (дата обращения: 26.12.2023.)
4. Команда SPbUnited приняла участие в открытом чемпионате по робофутболу в Бразилии. <https://239.ru/robot/tpost/rmrepiur1-nasha-komanda-prinyala-uchastie-v-otkrit> (дата обращения: 26.12.2023.)
5. SPbUnited, CBRobotica, Brazil, 2023. Записи игр. <https://www.youtube.com/playlist?list=PLdtGqMuKqq4xnZ6U0cP74L7y8fWG9zagE> (дата обращения: 26.12.2023.)
6. SPbUnited, репозиторий с исходным кодом на GitHub. <https://github.com/SPbUnited/strategy> (дата обращения: 26.12.2023.)
7. Библиотека для работы с очередью сообщений. <https://pyzmq.readthedocs.io/en/latest/> (дата обращения: 26.12.2023.)
8. Библиотека для написания асинхронного кода с использованием синтаксиса `async/await`. <https://docs.python.org/3.9/library/asyncio.html> (дата обращения: 26.12.2023.)
9. Система компьютерного зрения для RoboCup SSL. <https://github.com/RoboCup-SSL/ssl-vision> (дата обращения: 26.12.2023.)
10. Липкович М., Лучин Р., Сохарев А. Учебно-исследовательский стенд для мультиагентных систем управления (на примере футбола роботов). Материалы XIII Конференции молодых ученых «Навигация и управление движением», 2011.
11. Iarosh D., Reneva G., Kornilova A., Konovalov P. Multiagent System of Mobile Robots for Robotic Football // 2019 26th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS). St. Petersburg, Russia, 2019. P. 1-3. DOI: 10.23919/ICINS.2019.8769365. <https://ieeexplore.ieee.org/document/8769365> (дата обращения: 26.12.2023.)