



Руководство пользователя

Программа: «Демонстрационный программный комплекс для управления роботами, передачи пасов и игры по регламенту RoboCup SSL (SAFSAF)»



Содержание

1. Общие сведения	3
2. Использование совместно со студией для запуска стратегий	3
3. Использование студии PAsmaCS для работы с SAFSAF	4
4. Передача данных между SAFSAF и студией	5
5. Конфигурационные параметры	6
6. Запуск демонстрационной стратегии	7



1. ОБЩИЕ СВЕДЕНИЯ

Демонстрационный программный комплекс предназначен для стратегического управления командой мобильных роботов в соответствии с регламентом RoboCup Small Size League (SSL). Программное обеспечение реализует принятие решений в реальном времени, распределение ролей, построение маршрутов движения, расчёт и исполнение пасов, а также формирование управляющих воздействий для роботов. Работа комплекса основана на компоненте strategy-bridge, обеспечивающей взаимодействие модулей через шину данных.

2. ИСПОЛЬЗОВАНИЕ СОВМЕСТНО СО СТУДИЕЙ ДЛЯ ЗАПУСКА СТРАТЕГИЙ

Для функционирования демонстрационного программного комплекса требуется использование внешней студии, обеспечивающей:

- передачу данных системы компьютерного зрения и судейского модуля из локальной сети непосредственно в программный продукт;
- передачу управляющих воздействий от программного продукта к роботам.

По умолчанию поддерживаются следующие среды:

- «Программируемая мультиагентная кибернетическая студия для управления роботами и автономного запуска стратегии (PAsmaCS)»;
- LARCmaCS (общедоступная студия для запуска стратегий).

Программный комплекс может быть адаптирован для работы с конкретной студией, выбранной пользователем.

Следует учитывать, что использование LARCmaCS не позволяет использовать полный функционал SAFSAF. Для целей эксплуатации рекомендуется применять PAsmaCS, на основе которой построено последующее «Руководство пользователя».



3. ИСПОЛЬЗОВАНИЕ СТУДИИ РАСМАС для РАБОТЫ с SAFSAF

Для корректного использования SAFSAF рекомендуется применять программное обеспечение РАСМАС. В данном приложении реализована готовая обработка сообщений от модуля машинного зрения и модуля судьи, а также осуществляется вывод визуализации и отладочной информации (телеметрии). В частности, API РАСМАС предоставляет следующие функции:

- Получение данных с модуля машинного зрения.
- Получение данных с модуля судьи.
- Вывод визуализации программы.
- Вывод отладочной информации (телеметрии).

Подробная информация о программном интерфейсе (API) программного комплекса, а также о используемых сетевых портах для передачи данных и управляющих пакетов представлена в официальной документации программного комплекса. Пользователю рекомендуется обращаться к данной документации для получения актуальных сведений и параметров конфигурации.

Для начала работы SAFSAF с РАСМАС следует:

1. Скачать SAFSAF на рабочее устройство (Демонстрационный установочный файл доступен по ссылке: <https://drive.google.com/file/d/1o8EKNPtd2g7kvuIYoAS2omU5TwxzL4uD/view>);
2. Установить РАСМАС в соответствии с прилагаемой инструкцией (Демонстрационный установочный файл доступен по ссылке: <https://drive.google.com/file/d/1Xa8rbmxSyd4ihqGnENzuHiuD0-gC0-yT/view?usp=sharing>);
3. Запустить приложение и убедиться, что производится корректный вывод информации о поле или симуляторе.

Примечание: В случае отсутствия получаемых данных о поле или симуляторе программа РАСМАС не сможет передавать информацию в стратегический модуль SAFSAF, что приведет к некорректной работе стратегии.



4. ПЕРЕДАЧА ДАННЫХ МЕЖДУ SAFSAF И СТУДИЕЙ

При корректной установке всех компонентов программного комплекса, после их запуска на одном устройстве осуществляется автоматическая передача данных между ними.

В случае, если стратегический модуль выводит сообщения о отсутствии внешних данных и/или программный комплекс RAsmaCS не отображает в телеметрии данные, передаваемые стратегией, рекомендуется выполнить следующие действия:

1. Проверить корректность настройки сетевых параметров в соответствии с разделом «Порядок действий при настройке» из «Руководства по установке и настройке».
2. Повторно запустить программный комплекс и убедиться в корректной передаче данных между компонентами.

Данная процедура обеспечивает надёжное функционирование программного комплекса при изменении стандартной конфигурации или при модификациях, внесённых пользователем.



5. КОНФИГУРАЦИОННЫЕ ПАРАМЕТРЫ

При первом запуске программного комплекса автоматически создаётся файл `.env`, содержащий основные параметры конфигурации. В данном файле содержатся основные игровые параметры, включая цвет команды, полярность стороны, идентификационный номер вратаря и другие необходимые значения. Также через данный файл можно включить отладочный режим для проверки различных игровых состояний. К каждому параметру прилагается его описание и примеры возможных значений.

```
.env
1  # FILE WITH BASIC CONFIGURABLE CONSTANTS #
2
3
4  # Controlling robots in a simulator
5  IS_SIMULATOR_USED = True
6  # True | False
7
8
9
10 # The division in which the game will be played
11 DIV = B
12 # A | B | C
13
14 # The goal side of two teams
15 POLARITY = 1
16 # 1 = blue goal on "-x" | -1 = blue goal on "+x"
17
18 # The color of the team you play for
19 COLOR = BLUE
20 # BLUE | YELLOW
21
22 # Ally goalkeeper number (of the "COLOR" team)
23 GK = 0
24 # 0 | 1 | 2 | ... | 14 | 15
25
26 # Enemy goalkeeper number (of the "not COLOR" team)
27 ENEMY_GK = 0
28 # 0 | 1 | 2 | ... | 14 | 15
29
30
31
32 # Manually setting the game state
33 DEBUG_MODE = True
34 # True = use gamestate from DEBUG_GAME_STATE
35 # False = listen states from the Referee (HALT by default)
36
37 # Game state in DEBUG_MODE
38 DEBUG_GAME_STATE = DEBUG
39 # HALT | TIMEOUT | STOP | PREPARE_KICKOFF | BALL_PLACEMENT
40 # PREPARE_PENALTY | KICKOFF | FREE_KICK | PENALTY | RUN | DEBUG
41
42 # Active team in DEBUG_MODE
43 DEBUG_ACTIVE_TEAM = BLUE
44 # ALL = both teams | BLUE | YELLOW
45
46 # Time to prepare before FREE_KICK, KICKOFF, PENALTY (start state PREPARE_... before it)
47 DEBUG_PREPARATION_DELAY = 10.0
48 # Time in seconds
```

Рисунок 1. Файл конфигурации

Пользователь должен самостоятельно настраивать эти параметры перед проведением игры



или тренировочного занятия.

Изменение значений выполняется пользователем напрямую в файле .env с соблюдением комментариев и рекомендаций по каждому параметру.



6. ЗАПУСК ДЕМОСТРАЦИОННОЙ СТРАТЕГИИ

Программный комплекс поддерживает запуск демонстрационных стратегий, предназначенных для проверки и демонстрации функционала системы управления роботами. Выбор стратегии осуществляется при запуске программы с использованием аргумента командной строки.

Общий формат запуска

```
python main.py <название_стратегии>
```

где <название_стратегии> — одно из допустимых значений:

- zmeika
- random_goal
- passes
- game

После запуска программный комплекс, при необходимости, автоматически инициирует передачу управляющего сигнала в симулятор для выполнения корректной расстановки роботов на игровом поле. При вводе некорректного значения программа завершает работу с сообщением об ошибке.

Доступные стратегии

1. passes — стратегия передач

- На поле размещаются 4 робота под управлением программного обеспечения.
- Расстояние между любыми двумя роботами должно быть не менее 2 метров.
- В течение одной минуты роботы выполняют суммарно не менее 10 передач мяча.
- Запрещено возвращать мяч тому роботу, от которого была получена последняя передача.

Пример запуска:

```
python main.py passes
```




2. game — стратегия игры по регламенту RoboCup SSL

- 6 роботов распределяются по ролям: вратарь, защита ворот стенкой, перехват пасов, выполнение передач и забивание голов.
- Распределение ролей осуществляется динамически в зависимости от игровой ситуации.

Пример запуска:

```
python main.py game
```

3. random_goal — стратегия прямого удара по воротам

- На поле располагается мяч, неподвижные роботы противоположной команды и один управляемый робот.
- Задача: забить гол прямым ударом, не выпустив мяч за пределы поля.
- Гарантируется, что гол возможен после одного прямого удара.

Пример запуска:

```
python main.py random_goal
```

4. zmeika — стратегия «Змейка»

- На прямой линии выставляются 5 банок на расстоянии 0,5 м друг от друга.
- Робот стартует на расстоянии не менее 1 м от первой банки.
- Мяч устанавливается между роботом и ближайшей банкой на расстоянии 0,5 м от банки.
- Робот должен захватить мяч и выполнить движение по траектории «змейка» вокруг всех банок в прямом и обратном направлении, не теряя мяч.

Пример запуска:

```
python main.py zmeika
```