ELSIROS

Выпуск 0.0.1

Azer Babaev, Egor Davydenko, Ilya Ryakin, Vladimir Litvinenko,

сент. 23, 2021

Содержание:

Глава 1

Введение в ELSIROS

Игра в футбол человекоподобными роботами (гуманоидами) становится популярным в университетском и научном сообществе. Этот вид соревнований является источником для большого количества научно – прикладных исследований. Для учеников школ, колледжей и студентов в общем можно сформулировать 3 основных препятствия для создания и поддержания команды из роботов-гуманоидов-футболистов:

1. Слишком высокая цена роботов – гуманоидов,
2. Слишком сложные алгоритмические задачи, которые необходимо решить на начальном этапе включения робота в игру,
3. Маленькое число команд в одном населенном пункте – означает, что каждаигра возможна только в комбинации с поездкой на дальние расстояния с относительно большими расходами на командировку.

Все 3 препятствия можно избежать, если новая команда начинает приобретать опыт футбола гуманоидов с ELSIROS. Используя ELSIROS, можно воспользоваться следующими преимуществами:

1. ELSIROS полностью бесплатный и с открытым кодом.
2. Самые трудные части освоения гуманоидных роботов, которые выходят за рамки программы школы и колледжей, такие как обратная кинематика, генератор ходьбы, планирование траекторий, детекция мяча и препятствий и локализация обеспечены в готовом виде с предоставлением исходного кода. Пример стратегии игры, который был доказан как успешный на играх реальных роботов предоставляется для изучения и дальнейшего совершенствования.
3. Для участия в соревнованиях и товарищеских играх нет необходимости в командировках. Команды – участники могут скомпилировать свой исходный код в исполняемый бинарный код, который обеспечен защитой от копирования алгоритмов, и выгрузить его на сервер судьи.
4. Команды не испытывают проблем с люфтами сервомоторов, разрегулировкой, раскалибровкой, так как виртуальная модель предоставляется без люфтов, отрегулированной и откалиброванной.
5. Модули стратегии, которые будут создавать команды в дальнейшем готовы к использованию на физических роботах, которые команды могут захотеть построить сами или приобрести готовыми.

6. Для симуляции тренировочных игр не требуется мощных серверов, симуляцию игры можно запустить даже на портативном компьютере.

ELSIROS создана командой гуманоидного футбола «Старкит» из МФТИ послезавоевания чемпионского титула на мировом чемпионате Robocup 2021ELSIROS – бесплатная с открытым кодом платформа, претендующая с одной стороны помочь новым исследовательским кружкам войти в мир футбола гуманоидных роботов, с другой стороны стать основой для виртуальных соревнований. ELSIROS состоит из следующих компонентов:

1. Симулятор Webots (скачивается с сайта разработчика);
2. Первоначальная модель робота, которая является виртуальной моделью физически существующего робота – победителя международных соревнований футбола гуманоидных роботов в 2019, 2020 и 2021 годах, использовавшийся командой «Robokit»;
3. Виртуальное окружение симулятора в виде футбольного поля;
4. Программа судьи, действующего Автономно или под управлением человека, а также программа управления игрой (game controller);
5. Пакет программ управления виртуальным роботом, обеспечивающий игру в футбол;

Вновь образованные команды могут использовать для участия в соревнованиях виртуальную модель робота «Robokit-1» а также использовать его для изучения основ программирования стратегии игры в футбол роботами-гуманоидами. Это удобный инструмент для изучения Искусственного Интеллекта в школах, колледжах и университетах. Программирование роботовподдерживается на языке Python 3. При необходимости также поддерживаются языкиC, C++, Java, MATLAB для вашего удобства

Структура пакета управляющей программы построена для 4 уровней программирования робота: Начальный, Средний, Продвинутый и Экспертный.Раработчики начального уровня могут изменять файл strategy.py с цельюизменения поведения робота на самом высоком уровне логики. Предлагаемыйисходный код представляет стратегию игры, использовавшуюся лидирующейРоссийской командой на нацилнальном чемпионате 2021. Видео этой игры доступнопо ссылке:

```
https://youtu.be/AmfKpkL2MUc
```

Программисты среднего уровня могут пробовать изменять модуль launcher.py. Этот модуль отвечает за детекцию состояния игры, состояния игрока, управление ролями игроков и их начальных позиций. Продвинутые разработчикимогут попробовать улучшить другие модули исходного кода, которые отвечаютза инверсную кинематику, дыижени, локализацию, планирование пути робота.

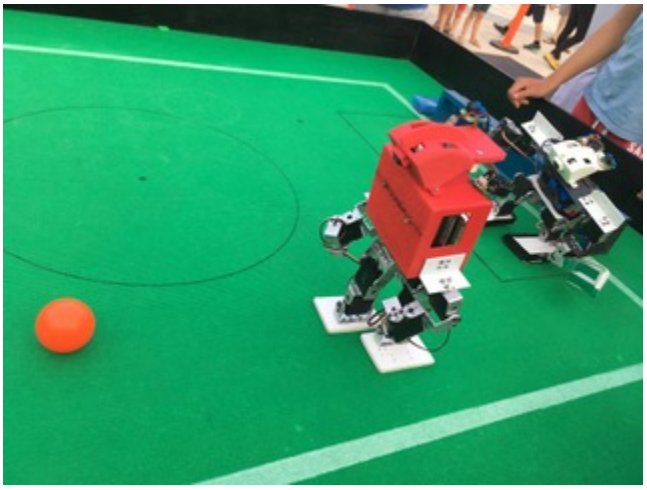
Специалисты экспертного уровня могут собрать свою модель робота и использовать свои программы управления роботом, решая сами пользоваться или не пользоваться исходным кодом, который предоставлен разработчиками ELSIROS. Для допуска к соревнованиям команда, предоставляющая свою собственную виртуальную модель робота, должна пройти квалификацию модели робота. Основное требование к модели робота состоит в том, что модель должна иметь такие же технические характеристики, как и физический робот. Специальные требования к PROTO модели, которые следуют исходя из свойств виртуальной среды, могут быть высланы по запросу.

Если вы руководитель потенциальной команды, которая хочет играть в футбол гуманоидными роботами, то вы можете самостоятельно начинать без особых трудностей. Необходимо скачать и установить ELSIROS на ваш компьютер. На выбор есть версия в исполняемых файлах для Windows 10 или версия для самостоятельной сборки для Linux. Следуйте инструкциям по установке и программа, приведенная как пример, начнет играть в футбол сразу после установки. Используйте вашу предпочитаемую среду разработки программ Python 3 для улучшения исходного кода игры игрока и вы уже готовы к виртуальным соревнованиям в футбол гуманоидных роботов в симуляции. Вы можете проводить тренировочные игры на своем переносном компьютере.

Глава 2

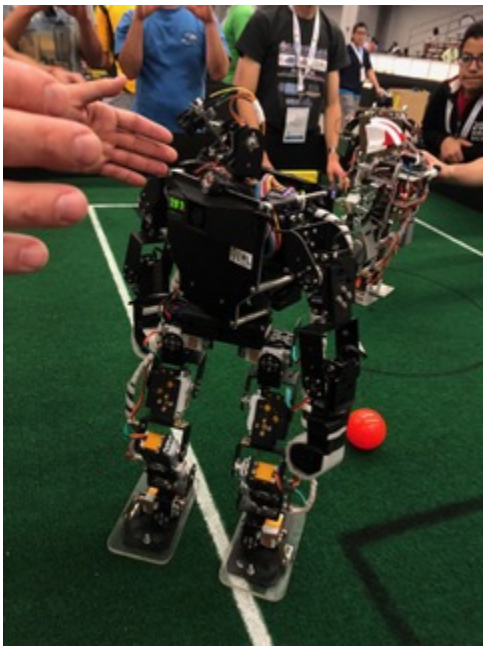
История юношеских гуманоидных футбольных игр

Известные ранние примеры футбола гуманоидных роботов, построенных и запрограммированных юниорами для Robocup относятся к 2016 году.

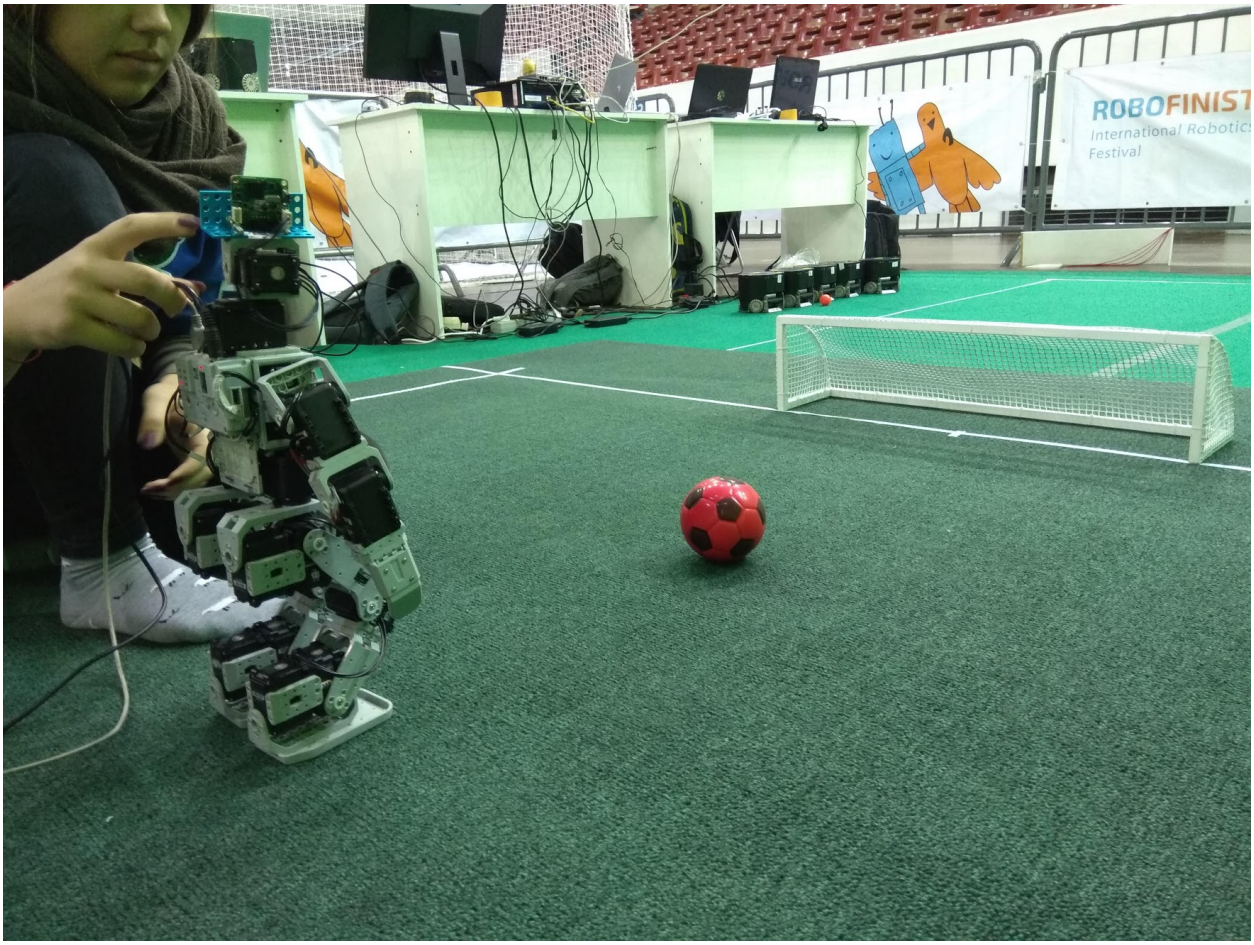


Попытки играть в футбол гуманоидным роботом командой из Израиля были сделаны в 2016 - 2018 годах

Команды из Израиля и Италии сыграли первый Международный матч в качестве демонстрационной игры на Robocup - 2018 в Монреале



Параллельно 2 команды из Томска и Москвы сыграли первую игру на соревнованиях Робофинист в Санкт-Петербурге 2018



Те же команды сыграли еще раз в Москве в 2019 году

Первый полноценный турнир состоялся в Томске в 2019 году с участием 3-х команд - участников Российского этапа Robocup


ый Российский этап чемпионата

С 2019 года игра была включена в официальную программу Robocup Asia - Pacific.

2.1 Игровое поле

Игровое поле сделано из ковролина 3мм размером 3 X 4 м

С воротами изготовленными из сантехнических труб и обклееных голубым и желтым скотчем

2.2 Мяч

Мяч оранжевого цвета из поролона диаметром 80 мм

2.3 Роботы

Сейчас 2 типа роботов используются командами

2.3.1 Модели Bioloid с надстроенной камерой и компьютером

2.3.2 Robokit - робот спроектированный в МФТИ с использованием японских сервомоторов KONDO

Последний робот, который показал наилучшие результаты став чемпионом Азиатско-Тихоокеанских игр в 2019 и 2020 годах был использован в качестве прототипа для виртуальной модели стандартного робота в платформе ELSIROS

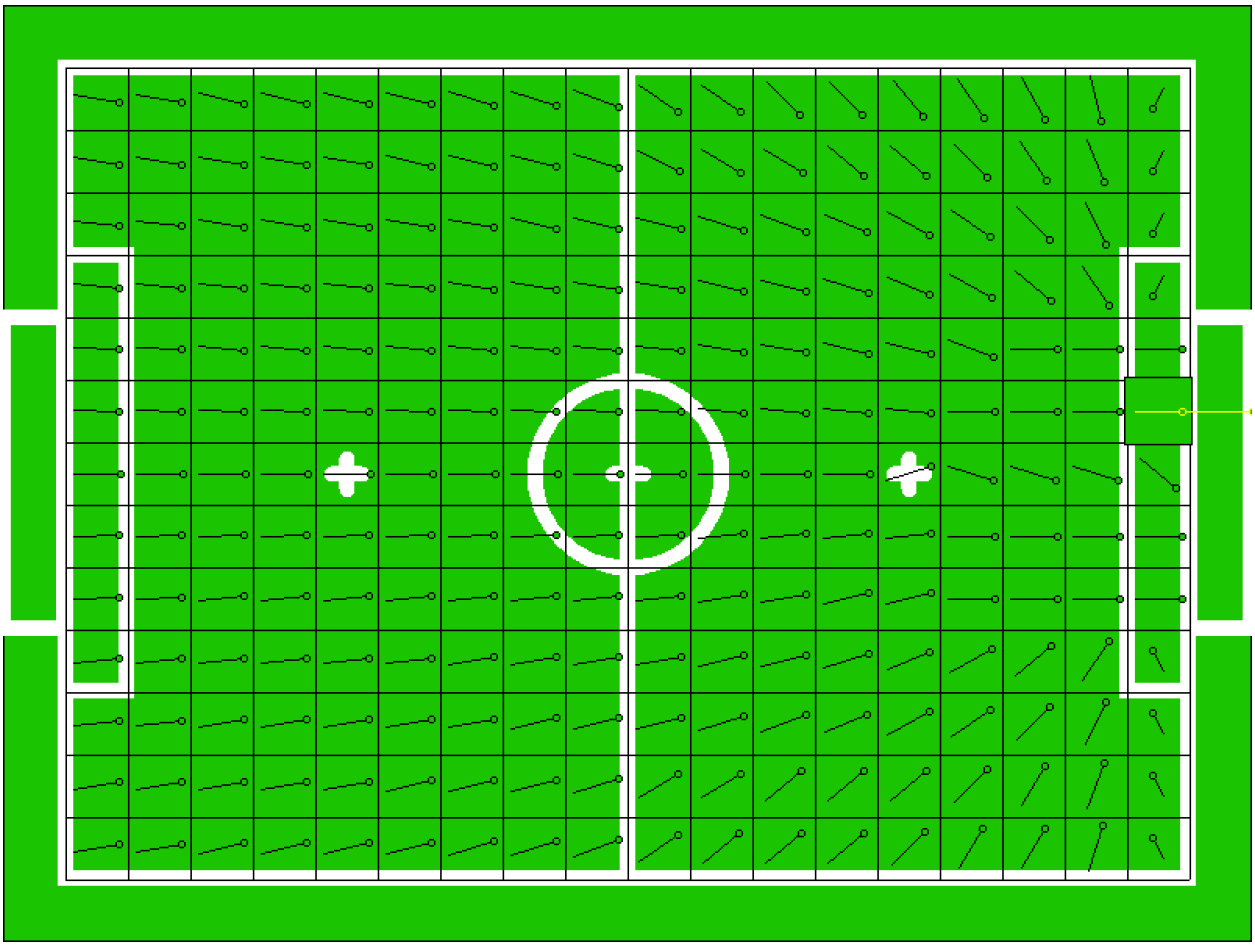
Глава 3

Руководство программирования Робота Robokit-1

1. Не рекомендуется играть игру между полностью одинаковыми командами. В большинстве случаем вы будете наблюдать неинтересную игру с малым количеством голов и безрезультатной борьбой. Поэтому мы вам предоставляем 2 разных алгоритма игры: нормальный и старого стиля, который использовался в 2020 году.
2. Реальные роботы используют смарт камеру OpenMV H7 в качестве визуального сенсора и в качестве вычислительного модуля. Это одноядерный контроллер со средой программирования Micropython без операционной системы. Это означает, что контроллер не позволяет одновременно использовать функцию хождения и получения видеокадра. Для обновления информации о нахождении мяча и само-локализации робот должен остановиться в стабилизированной вертикальной позиции и поворачивать голову в разные сектора обзора для осмотра окрестностей. При поворотах головы положение мяча может быть считано, если мяч оказался в видимом секторе камеры. Апертура камеры 46 градусов.
3. В процессе обзора камерой в вертикальной позиции робот также получает информацию, полезную для локализации, такую как стойки ворот, разметка поля и граница зеленого поля. Чем больше кадров обработано с разных ракурсов, тем точнее информация о локализации.
4. Робот может обновлять информацию о препятствиях из полученных кадров. В алгоритм построения траекторий включен алгоритм обхода препятствий, который не совсем совершенен. Нет планирования направления удара с учетом препятствий на пути мяча. Список self.glob.obstacles содержит информацию об обнаруженных и обновленных данных о препятствиях.
5. Коммуникация между игроками разрешена по правилам через UDP сообщения WIFI. Коммуникация не реализована в текущей стратегии, но команды могут её разрабатывать. Коммуникация внутри команды может помочь организовать более эффективную игру. ELSIROS API содержит средства коммуникации между членами команды.
6. Координатное пространство, используемое в стратегии, отличается от абсолютного координатного пространства в симуляции. Для целей стратегии собственные ворота располагаются в части поля с отрицательными значениями координаты X, ворота противника располагаются в части поля с положительными значениями координаты X. Положительные значения Y расположены на левом фланге атаки, отрицательные значения Y расположены на правом фланге атаки. Азимут yaw имеет нулевое значение, если направлен из центра поля к воротам противника. Yaw

изменяется от 0 к math.pi при вращении вокруг вертикальной оси влево от нулевого направления. Yaw изменяется от 0 к -math.pi при вращении вокруг вертикальной оси вправо от нулевого направления. Координата Z направлена вверх с нулём на поверхности поля.

7. Нормальный игрок 'forward' использует предварительно запрограммированную стратегию, заложенную в матрицу векторов. Матрица закодирована в файле strategy_data.json . Этот файл является читаемым и редактируемым так же как и обычный текстовый файл. В файле содержится словарь с одним ключом 'strategy_data'. Под ключом 'strategy_data' содержится список с числом элементов по умолчанию 234. Каждый элемент списка представляет квадратный сектор поля размерами 20cmX20cm. Каждому сектору назначен вектор представляющий из себя направление удара в ситуации когда мяч оказался в этом секторе. Сила удара закодирована в виде ослабляющего коэффициента: 1 – стандартный удар, 2 – удар ослабленный в 2 раза, 3- удар ослабленный в 3 раза. Каждый элемент списка закодирован следующим: [столбец, ряд, сила, yaw]. Футбольное поле разбито на 13 рядов и 18 столбцов. Столбец 0 находится возле своих ворот, столбец 17 находится возле ворот противника. Ряд 0 находится в области с положительной Y координатой, ряд 12 находится в области с отрицательной Y координатой.



1. В процессе игры игрок может принимать 4 разные роли: 'forward', 'goalkeeper', 'penalty_Shooter', 'penlaty_Goalkeeper'. Для каждой роли программа стратегии разная. Модуль launcher выбирает роль игрока для запуска в зависимости от номера игрока и вторичного состояния игры. В случае если номер игрока 1 назначаемая роль будет 'goalkeeper' или 'penalty_Goalkeeper'. В случае, если номер игрока отличается от 1, тогда назначаемая роль 'forward' или 'penalty_Shooter'. В случае secondary game state = 'STATE_PENALTYSHOOT' тогда игрок с номером 1 будет назначен 'penlaty_Goalkeeper', а игрок с номером, отличающимся от 1 будет назначен на роль 'penalty_Shooter'. Котроллер игры, поставляемый по умолчанию будет

назначать игроку с номером 1 ‘goalkeeper’, а игроку с номером, отличающимся от 1 роль ‘forward’ во всех остальных значениях secondary game state. Команды могут менять стратегию игры путем назначения различных ролей в зависимости от величины secondary game state, поставляемой геймконтроллером. Текущий геймконтроллер может поставлять следующие значения secondary game state: STATE_NORMAL=0, STATE_PENALTYSHOOT=1, STATE_OVERTIME=2, STATE_TIMEOUT=3, STATE_DIRECT_FREEKICK=4, STATE_INDIRECT_FREEKICK=5, STATE_PENALTYKICK=6, STATE_CORNERKICK=7, STATE_GOALKICK=8, STATE_THROWIN=9, DROPBALL=128, UNKNOWN=255

Глава 4

Правила

RoboCup Junior Humanoid Soccer Правила и Настройки

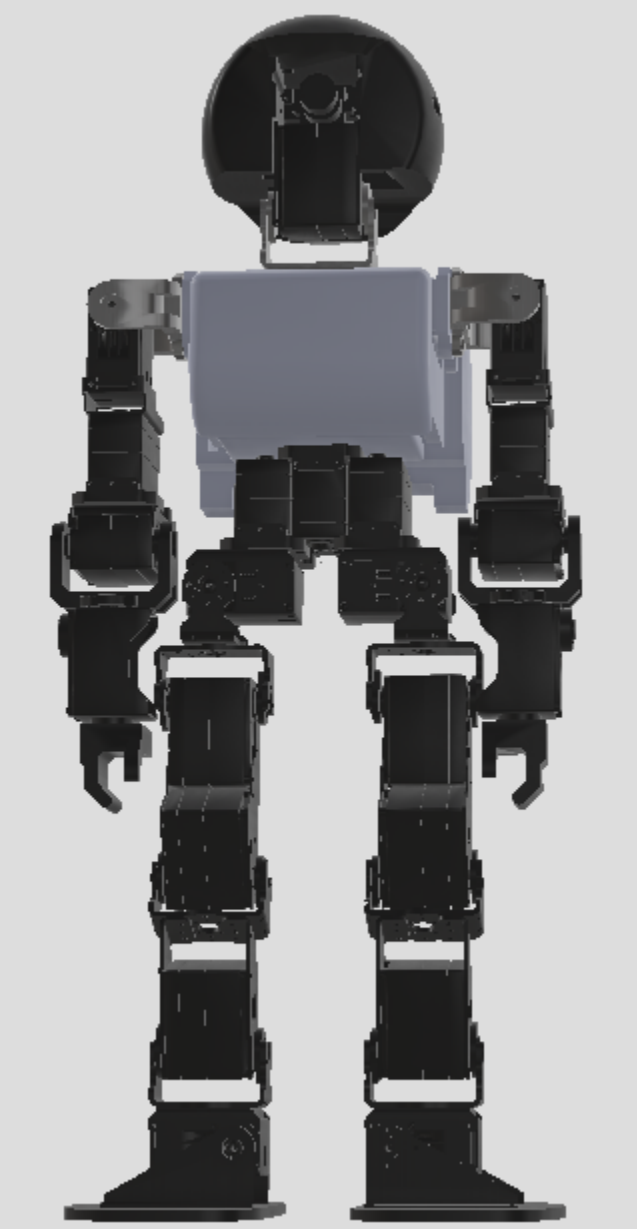
для виртуальных соревнований в симуляторе в 2021 году

Составитель Азер Бабаев (обновлено 14.09.21, Москва), на основе версии 2007 года правил Humanoid Soccer

Глава 5

Дизайн робота

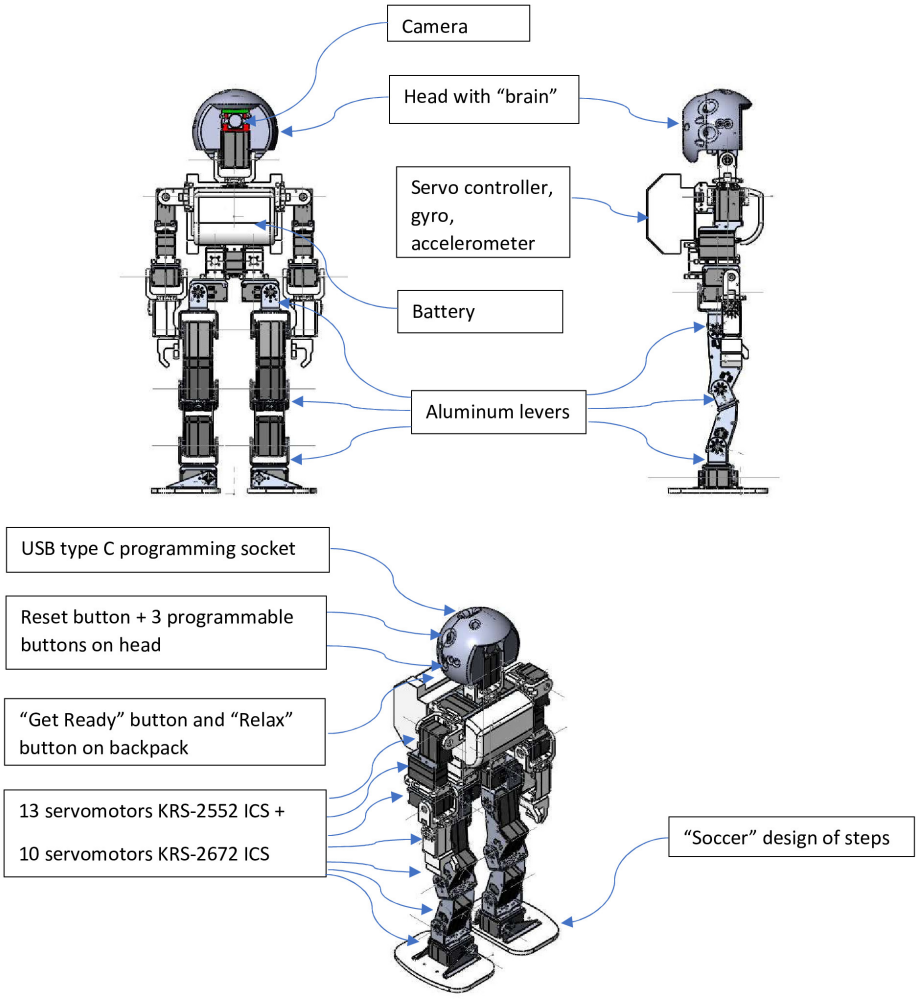
Standard robot proto used for simulation is Robokit1.proto. It is designed as virtual copy of real robot. In order to represent real robot all technical features of virtual model were copied from technical description of real robot including servo torque, servo speed, servo mass, location of camera, camera resolution, aperture of camera lens, number and location of accelerometers, weight distribution throughout of body.



Virtual model has 2 IMUs named «imu_head» and «imu_body». Real robot is capable to recognize coordinate of itself at soccer field with accuracy ±15cm, it is capable to recognize obstacles and ball. Recognition of distance and course to ball and obstacles is made through machine vision algorithms. Robot detectds distance and course to object with accuracy mainly dependent on camera accuracy. Good calibrated robot detects course to object with accuracy ± 0.01 Radian. Accuracy of distance recognition non-linearly depends on distance to object, with best accuracy at minimum distances which is ± 5mm. Distance measurement accuracy at distance 1m is ± 25mm, at distance 2m is ± 100mm High accuracy of distance and course measurement are provided due to equipment and smart algotihms used for direct measurements. Localization on soccer field is provided by measurements of cource and distence to goal posts, field marking, green field border, odometry, measurement of IMU. All localization data is accumulated with historical data and processed by patricle filter algorithm in order to achieve better accuracy than measurement of individual object. Simulation procedure in Webots is orgaised in way when simulation of physics and motion is provided together with camera image scanning. Procedure of image scanning and transfer to outside from Webots greatly reduces simulation speed. In order to keep simulation speed at acceptable rate it was decided to skip procedure of scanning images by camera in simulation and replace it with direct request- report

procedure. Following data are reported to Robokit1 robot from simulation by request: distance and course from robot to ball, distance and course from robot to other players, self coordinate and orientation on field. Requested data before reporting to robot are mixed with random values in order to provide same accuracy which usually real robot detects from camera image. Above procedure is called «blurrer». Above procedure provide increasing is simulation speed up to 10 times. This know-how of ELSIROS provides games to be played at laptop computer with nearly realtime speed.

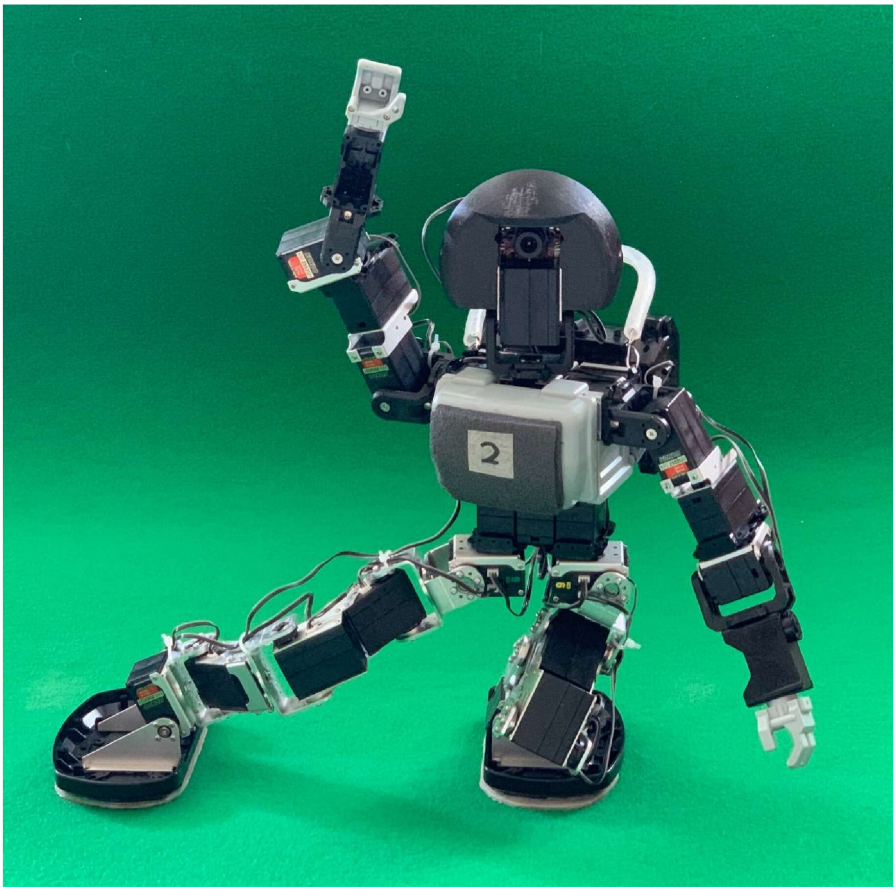
Below is technical description of real Robokit robot.



Robokit Robot

ROBOKIT Robot Specification:

- Height 45 cm
- Weight 1.9 kg
- Battery voltage 12 V
- 23 DOF: 13 servomotors KRS-2552 ICS +10 servomotors KRS-2672 ICS
- Main controller: OpenMV H7 with 32-Bit Arm Cortex-M7 operating at 400MHz with 1Mb SRAM
- Motion controller: Kondo RCB-4HV
- Programming language: Micropython.
- Sensors: OV7725 640x480 camera, 6D digital IMU BNO055, 2D analogue Gyro, 3D analogue Accelerometer



Глава 6

API