

## Лабораторная работа № 1

### Конструирование мобильного робота в среде CoppeliaSim

**Цель работы:** получить практические навыки использования встроенной CAD-системы среды CoppeliaSim для проектирования колесных роботов и настройки параметров их моделирования.

#### 1. Обзор приложения CoppeliaSim

##### Назначение

Развитие робототехники и технологий моделирования привело к появлению целого класса программного обеспечения – **робототехнических симуляторов**, которые имеют особую значимость при обучении робототехнике. Виртуальные симуляторы существенно отличаются от сред моделирования общего назначения. В них применяется подход, при котором в модели имеются компоненты, реализующие такие же функции, как и аналогичные компоненты в реальных роботах. Таким образом, изучая виртуальную модель можно получить практические навыки работы с электрическими приводами, механическими элементами и системой управления.

Тестирование является наиболее распространенной, но далеко не единственной целью разработки экспериментальных платформ. Большинство современных промышленных роботов имеют собственную систему оффлайн – программирования, в которую входит и система экспериментального моделирования. Подобная система выполняет все необходимые расчеты для моделирования и визуализации работы робота (рис. 1).

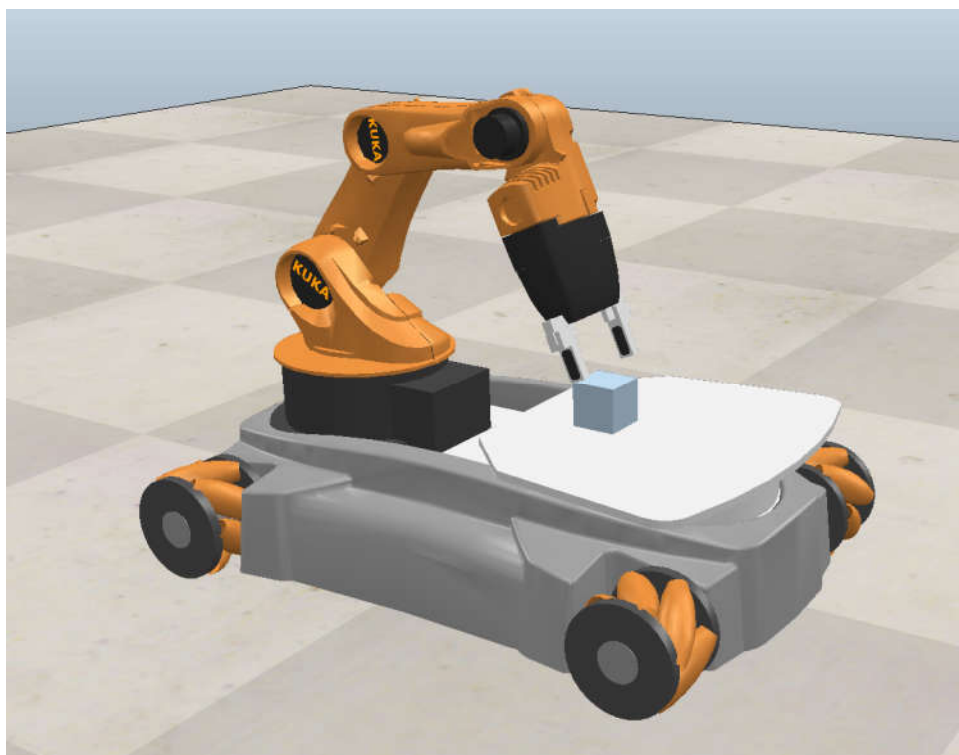


Рис. 1. Модель мобильного робота-автопогрузчика в CoppeliaSim

Цель моделирования зависит от поставленных задач, а при создании новых робототехнических решений, главным образом – от стадии разработки. Это может быть и проверка гипотезы, и оптимизация конструкции, и тестирование программного обеспечения, реализующего новые алгоритмы обработки сенсорной информации и управления поведением, и на более поздних этапах – отладка исполняемого кода до его запуска на контроллере робота. Процесс разработки – нелинейный. Поэтому хорошая среда компьютерного моделирования позволяет избежать многих проблем еще на ранней стадии, позволяя тестировать как отдельные узлы робота, так и упрощенные модели, реализующие отдельные функции.

### Особенности приложения

**Программная структура.** Среда CoppeliaSim имеет ряд особенностей, которые предоставляют разработчику широкие возможности для создания симуляций. Как основной компонент CoppeliaSim можно выделить технологию встроенных сценариев, которые выполняют функции контроллеров в симуляциях. При этом, наличие возможности привязки отдельных сценариев к компонентам робота позволяет реализовать четкую иерархию, обеспечивая портативность и масштабируемость.

Любая симуляция в CoppeliaSim по умолчанию имеет **основной сценарий**, который решает общие задачи обеспечения корректности данных при выполнении симуляции. Например, вызывает разные подсистемы для моделирования кинематики и динамики механических элементов системы.

**Дочерние сценарии**, в отличие от основного, прикрепляются к отдельным компонентам модели. Они являются неотъемлемой частью сценария симуляции и выполняются при каждой итерации моделирования. Сценарии являются исполняемыми файлами и не требуют предварительной компиляции. Дочерние сценарии могут быть **потокowymi** или **непотокowymi**. Симуляция – это итеративный процесс и перерасчет параметров моделируемой системы осуществляется через постоянный промежуток времени (шаг моделирования). В CoppeliaSim по умолчанию используется 50 миллисекунд.

**Потоковые сценарии** не зависят от итераций симуляции и выполняются один раз от начала до конца. **Непотокowe сценарии** регулярно запускаются с началом каждой итерации. Такие сценарии вызываются основным сценарием 2 раза за каждый шаг симуляции: при получении сенсорной информации и выработке управляющих воздействий.

**Модель «Клиент-Сервер».** Среда CoppeliaSim выступает сервером, на котором запущена симуляция, а управление осуществляется на клиентской стороне. При такой схеме взаимодействия в качестве клиента может выступать как другое программное обеспечение, так и любое устройство с подключением к серверу. Например, клиентом может быть реальный робот.

**API функции.** API CoppeliaSim включает более 400 функций на языке Lua. Remote API среды позволяет создавать сценарии симуляции на других языках программирования: C/C++, Python, Java, Matlab, Octave. CoppeliaSim имеет 4 физических движка моделирования: ODE, Bullet, Vortex и Newton. Все они хорошо ре-

шают общие задачи моделирования роботов, но их особенности способствуют лучшему решению при моделировании определенных задач.

**Встроенные модули для специализированных задач.** CoppeliaSim имеет мощный модуль вычисления обратной и прямой кинематики роботов. Данный пакет хорошо подходит для узкоспециализированных задач при работе с манипуляторами. Встроенный модуль для быстрой проверки столкновений объектов позволяет решать ряд задач, связанных с безопасностью. Также имеется модуль, который быстро и точно рассчитывает минимальное расстояние между объектами.

**Датчики разного типа.** В среду встроена поддержка датчиков приближения и фото-видео датчиков. В виде модулей в CoppeliaSim можно задействовать следующие сенсоры: 2D- и 3D-лазерные сканеры, GPS, гироскоп, датчика касания, лазерный указатель и др.

**Планирование движения.** В программе V-REP планирование движения выполняется с использованием открытой библиотеки планирования движения (Open Motion Planning Library), сокращенно OMPL.

**Запись и визуализация данных.** CoppeliaSim имеет весь необходимый инструментарий для записи и визуализации любых типов данных и сцен работы робототехнических систем в симуляциях.

**Встроенный редактор сетки.** Имеется редактор сетки модели, на основе которой рассчитываются физические параметры механических элементов симуляции. Также поддерживается несколько специальных режимов редактирования сетки объекта.

**Импорт и экспорт данных.** Загрузка и выгрузка данных CoppeliaSim выполняется через меню. Для этих задач имеется специальный инструмент в основном меню. Также приложение может получать данные через соединение с робототехническим устройством.

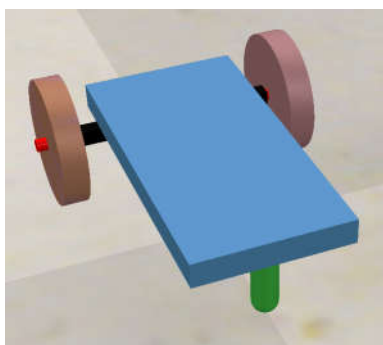
**Полнофункциональная иерархия моделей в сцене.** CoppeliaSim имеет удобный инструмент для установления взаимосвязи компонентов модели, который обладает широкими возможностями масштабирования.

**Доступность.** Распространяется приложение CoppeliaSim условно бесплатно. Для научных исследований и ведения образовательной деятельности данная программа имеет отдельные версии. В случае же использования в коммерческих проектах необходимо приобрести лицензию. CoppeliaSim является кроссплатформенной, т.е не зависит от используемой операционной системы.

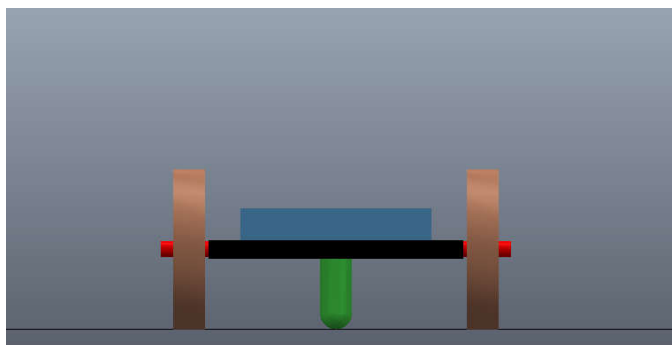
## 2. Задание

### 2.1. Обучающая часть

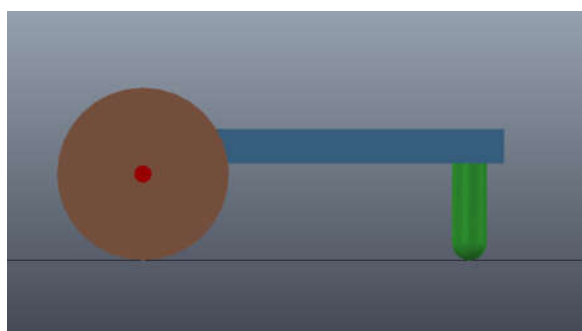
**Проект «Трехопорный мобильный робот».** Средствами встроенной CAD-системы приложения CoppeliaSim построить мобильную тележку, использующую два колеса с независимым приводом в задней части основания и неподвижную опору с полусферическим окончанием в передней части (рис. 1).



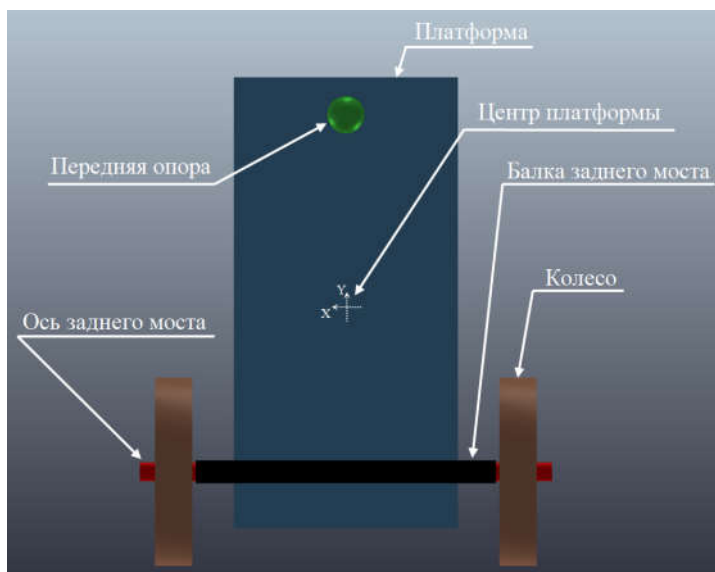
а)



б)



в)



г)

Рис. 1. Трехопорная тележка мобильного робота: а – общий вид; б – вид спереди; в – вид сбоку; г – вид снизу

### Технические характеристики элементов робота

#### **Платформа:**

- Размеры (X х Y х Z): **60 х 120 х 10 мм**;
- Плотность материала: **1250 кг/м<sup>3</sup>** (пластик).

#### **Балка заднего моста:**

- Размеры: **80 х 6 х 6 мм**;
- Смещение от центра платформы по оси Y ( $\Delta Y$ ): **-45 мм**;
- Плотность материала: **7800 кг/м<sup>3</sup>** (железо).

#### **Ось заднего моста:**

- Размеры (Диаметр х Длина): **5 х 10 мм**.

#### **Колесо:**

- Размеры (Диаметр х Ширина): **50 х 10 мм**;
- Плотность материала: **5000 кг/м<sup>3</sup>**.


#### **Передняя опора:**

- Диаметр сферического закругления: **10 мм**;


- Диаметр цилиндрической части опоры: **10 мм**;
- Длина: рассчитать самостоятельно;
- Смещение от центра платформы по оси Y ( $\Delta Y$ ): **+50 мм**;
- Плотность материала: **1250 кг/м<sup>3</sup>**.

1. Запустите приложение CoppeliaSim. По умолчанию будет создана новая сцена, которой следует дать название в соответствии с шаблоном: *Work1(<Фамилия студента-группа (на латинице)>-task1.ttt* и сохранить, используя меню «*File – Save scene as – CoppeliaScene...*». Например, студент Иванов из группы 11016 должен именовать файл сцены CoppeliaSim: *Work1(Ivanov-1101b)-task1.ttt*

### Платформа робота

2. Создайте платформу робота, к которой будут крепиться передняя опора и задний мост с колесами. Для этого щелкните правой кнопкой мыши в рабочей области (окно со сценой) и выберите меню «*Add – Primitive Shape - Cuboid*» (Добавить – Примитивные формы – Кубоид). В диалоговом окне «*Primitive Cuboid*» (Простой кубоид) укажите размеры, обозначенные на рисунке 1. В правом нижнем углу окна сцены показано текущее направление осей прямоугольной системы координат. В полях диалогового окна: «*X-size*», «*Y-size*» и «*Z-size*», укажите соответствующие значения в метрах: **6e-2** (означает  $6 \cdot 10^{-2}$  м (метр) = 6 см (сантиметр) = 60 мм (миллиметр)); **1.2e-1** и **1e-2**. Остальные поля диалогового окна оставьте без изменений. Щелкните кнопку «Ок».
3. Откройте окно иерархии объектов сцены, отметив пункт меню «*Tools – Scene Hierarchy*» (Инструменты – Иерархия сцены), или нажмите пиктограмму  на левой панели инструментов. Найдите объект с названием «*Cuboid*» и измените его название на «*Base*». Для этого дважды щелкните на названии объекта. В появившемся поле ввода желтого цвета укажите новое название.

**Примечание.** Название объектам сцены следует давать на латинице, без пробелов и специальных символов.

4. Центр платформы робота должен находиться над поверхностью сцены на высоте равной сумме половин высоты платформы и моста плюс радиус колеса. На рисунке 1 высота платформы 10 мм, высота моста 6 мм, а радиус колес составляет 25 мм. Таким образом, нужно поднять платформу на высоту  $(5+3+25)=33$  мм. Выделите объект «*Base*» на сцене, либо в окне иерархии и щелкните пиктограмму  на верхней панели инструментов. В появившемся диалоговом окне «*Object/Item Translation/Position*» (Объект/Элемент – Перемещение/Позиция) выберите закладку «*Translation*» (Перемещение), в поле «*Along Z*» (Вдоль Z) [m] запишите значение **3.3e-2** и нажмите кнопку «*Z-translate*» (Z-перемещение). Закройте диалог «*Object/Item Translation/Position*». Теперь центр платформы робота находится над поверхностью сцены на высоте 3.3 см.

5. Выделите объект «*Base*» и щелкните по нему правой кнопкой мыши. Во всплывающем меню выберите пункт «*Edit – Copy selected objects*» (Редактировать – Копировать выбранные объекты). Теперь щелкните правой кнопкой мыши вне объекта «*Base*» и выберите во всплывающем меню пункт «*Edit – Paste buffer*» (Редактировать – Вставить из буфера). В окне иерархии объектов сцены появится новый объект. Переименуйте его в «*Base\_visual*».
6. В окне иерархии сцены поместите курсор над объектом «*Base\_visual*» и нажав левую кнопку мыши переместите его на объект «*Base*». Отпустите левую кнопку мыши. Теперь объект «*Base\_visual*» должен стать дочерним по отношению к объекту «*Base*». В окне иерархии объектов «*Base\_visual*» станет подобъектом «*Base*» (рис. 2).

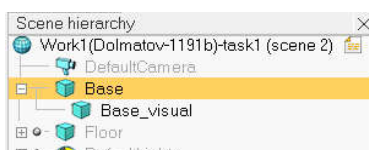


Рис. 2. Организация иерархии между объектами сцены

7. Объект «*Base*» будем использовать для настройки динамических качеств платформы робота, а «*Base\_visual*» для демонстрации ее визуальных характеристик и симуляции способности к детектированию датчиками разного типа. Выберите объект «*Base*» и пометьте пункт меню «*Tools – Scene object properties*». В появившемся диалоговом окне выберите закладку «*Common*» (Общие свойства). Установите флажок «*Camera visibility layers*» (Слои видимости камеры) для первой группы второго слоя (рис. 3). Это позволит отнести объект к скрытому (по умолчанию) слою, где, как правило, содержатся динамические и служебные объекты сцены. Выбор первой группы скрытого слоя выполнен произвольно на свое усмотрение разработчика.

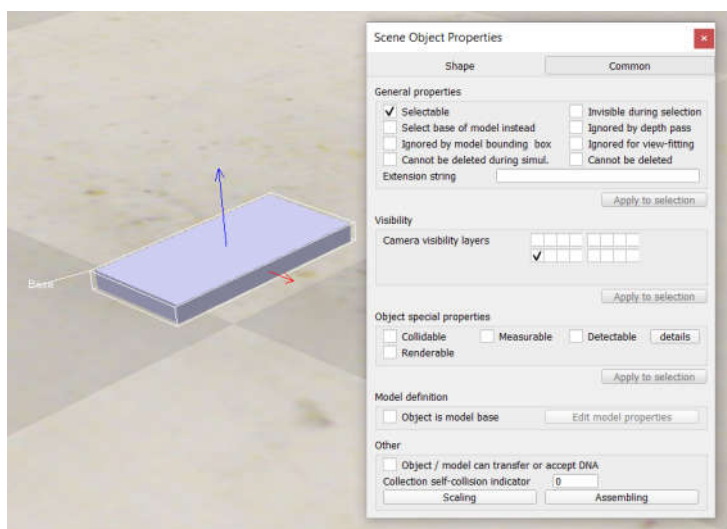


Рис. 3. Включение объекта «*Base*» в 1 группу скрытого слоя

Теперь выберите закладку «*Shape*» (Форма-объект) и нажмите кнопку «*Show dynamic properties dialog*» (Показать диалоговое окно динамических свойств). В появившемся диалоговом окне «*Rigid Body Dynamic Properties*» (Динамические свойства твердого тела) нажмите кнопку «*Compute mass & inertia proper-*



ties for selected convex shapes» (Расчет массы и моментов инерции выбранной выпуклой формы). В диалоговом окне «*Density body*» (Плотность тела) задайте плотность пластика PLA: **1250** кг/м<sup>3</sup>. Нажмите «Ok». Приложение рассчитает объем платформы, ее массу и моменты инерции, которые будут использоваться при симуляции движения. В диалоговом окне «*Rigid Body Dynamic Properties*» пометьте флажками пункты: «Body is responsible» (Тело взаимодействует с объектами сцены) и «Body is dynamic» (Тело используется для расчета динамики объектов сцены) (рис. 4).

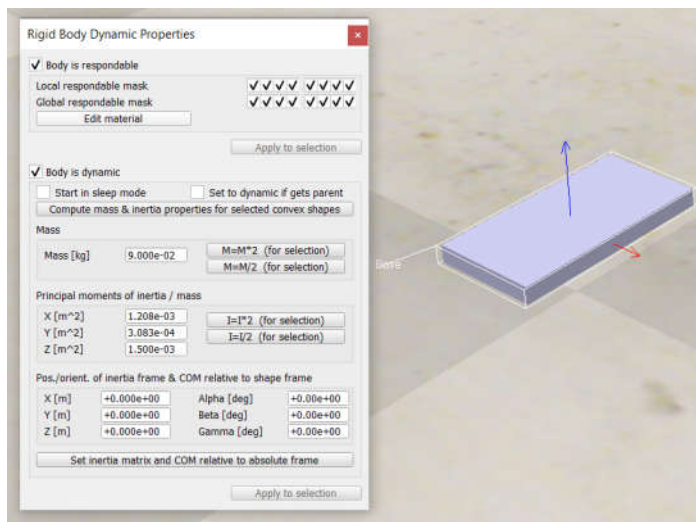


Рис. 4. Настройка динамических свойств объекта «Base»

Выберите объект «*Base\_visual*» и откройте его свойства с помощью меню «*Tools – Scene object properties*». В диалоговом окне «*Scene object properties*» выберите закладку «Common» (Общие свойства) и включите объект в первую группу видимого слоя (рис. 5).

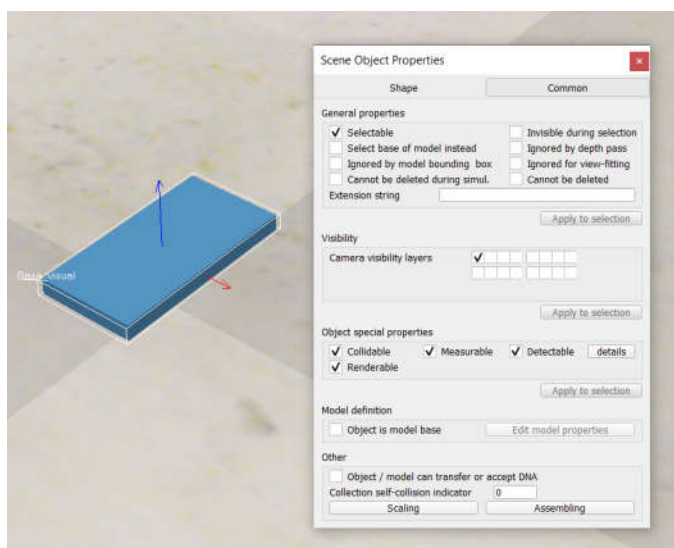


Рис. 5. Включение объекта *Base\_visual* в 1 группу видимого слоя

На этой же закладке в группе свойств «*Object special properties*» (Специальные свойства объекта) установите флажки у всех свойств: «*Collidable*» (Возможность проверки объекта на столкновение), «*Measurable*» (Возможность использования объекта для определения минимального расстояния до дру-

гого объекта), «*Detectable*» (Возможность обнаружения объекта датчиками приближения), «*Renderable*» (Возможность наблюдения объекта датчиками визуального контроля).

- В диалоговом окне «*Scene object properties*» выберите закладку «*Shape*» и нажмите кнопку «*Adjust color*» (Настроить цвет) в группе «*Visual properties*» (Визуальные свойства). Откроется диалоговое окно «*Shape*» (рис. 6). В нем можно настраивать: «*Ambient/Diffuse component*» (Цвет диффузно отраженной составляющей излучения объекта-формы), «*Specular component*» (Цвет зеркально отраженной составляющей излучения объекта-формы), «*Emissive component*» (Цвет излучения, генерируемого объектом-формой), «*Auxiliary component*» (Цвет вспомогательной компоненты излучения объекта-формы), «*Shininess*» (Степень отсутствия блеска объекта: 128 – максимально диффузное отражение излучения; 1 – максимально зеркальное отражение излучения), «*Opacity*» (Степень непрозрачности: 0 – 1). Когда флажок «*Opacity*» снят, объект-форма становится полностью непрозрачным. Произвольно настройте цвет каждого компонента излучения и параметров их смешения у объекта-формы (рис. 6).

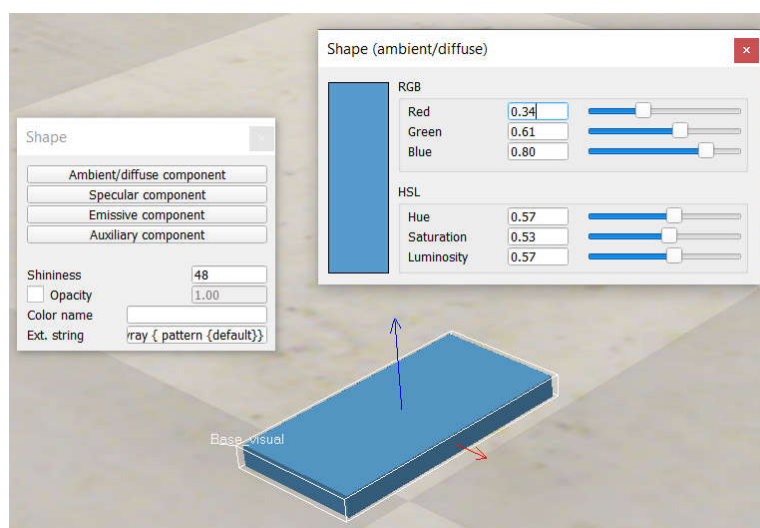




Рис. 6. Настройка спектральных параметров объекта-формы

Далее следует отключить динамические свойства у объекта «*Base\_visual*». Для этого откройте диалоговое окно настройки динамических свойств «*Rigid Body Dynamic Properties*» и уберите флажки «*Body is respondable*» и «*Body is dynamic*». Теперь объект «*Base\_visual*» отвечает только за отображение платформы робота, но не используется для расчета ее динамики и взаимодействия с другими объектами сцены. Однако в сцене данный объект является дочерним по отношению к объекту «*Base*». Поэтому во время симуляции он будет двигаться вслед за родительским объектом. Для демонстрации такой способности нажмите пиктограмму  на верхней панели инструментов. Запустится симуляция, в результате чего платформа робота упадет на поверхность клетчатого поля. Если флажок «*Body is respondable*» будет снят у обоих объектов «*Base*» и «*Base\_visual*», то платформа провалится



сквозь клетчатое поле. Отключите симуляцию пиктограммой  на верхней панели инструментов приложения CoppeliaSim.

### Задний мост

9. Постройте балку заднего моста, к которой будут крепиться колеса робота. Для чего создайте объект типа кубоид с размерами (x, y, z)[в метрах]: **(8e-2; 6e-3; 6e-3)**. Поднимите центр балки над поверхностью сцены на высоту 25 мм и сместите его вдоль оси Y на **-45** мм. Дайте созданному объекту сцены название «*BackBridge*».
10. Создайте копию балки заднего моста и назовите ее «*BackBridge\_visual*». Сделайте ее дочерним объектом для «*BackBridge*».
11. Используя диалоговое окно «*Scene object properties*» для объекта «*BackBridge*»: укажите принадлежность ко 2 группе скрытого слоя; настройте динамические свойства, опираясь на плотность железа – **7800** кг/м<sup>3</sup>.
12. Используя диалоговое окно «*Scene object properties*» для объекта «*BackBridge\_visual*»: настройте спектральные (цветовые) свойства; включите специальные свойства. Кроме того, данный объект следует отнести ко 2 группе визуального слоя и выключить у него динамические свойства.
13. Создайте оси для визуализации крепления колес и прикрепите их к балке моста. Для построения оси щелкните в рабочей области сцены правой кнопкой мыши и во всплывающем меню выберите пункт «*Add – Primitive shape – Cylinder*» (Добавить – Примитивная форма - Цилиндр). Диаметр (X=Y)[в метрах]: **5e-3**, а длина по оси Z: **1.5e-2**. Назовите первую ось «*RightAxis\_visual*». Поверните созданный объект вокруг оси Y на **90** градусов. Для этого на верхней панели инструментов нажмите пиктограмму . В появившемся диалоговом окне «*Object/Item/Orientation*» (Объект/Элемент – Ориентация) выберите закладку «*Rotation*» (Вращение). В поле «*Around Y*» (Вокруг Y) введите значение **90** и нажмите кнопку «*Y-rotate sel.*» (Y-вращение выбранной формы-объекта). Теперь нужно совместить ось с правым торцом задней балки. Используя диалоговое окно «*Object/Item Translation/Position*» с выбранной закладкой «*Position*» (Позиция) поместите центр правой оси по мировым (World) координатам в точку **(4.75e-2; -4.5e-2; 2.5e-2)**. Отключите динамические свойства объекта «*RightAxis\_visual*». Произвольно настройте его спектральные (цветовые) характеристики и включите все специальные свойства. Прикрепите объект «*RightAxis\_visual*» ко 2 группе визуального слоя. Создайте копию объекта «*RightAxis\_visual*». Назовите копию «*LeftAxis\_visual*» и поместите ее центр в точку **(-4.75e-2; -4.5e-2; 2.5e-2)** мировых координат. Если теперь с помощью диалогового окна «*Layers*» оставить включенными только 2 и 8 группу визуального слоя, то на сцене останутся визуальные объекты заднего моста (рис. 7). Созданные оси должны только создавать иллюзию вращения колес вокруг них, но не должны взаимодействовать с колесами как динамические объекты. Поэтому динамических копий объектов «*RightAxis\_visual*» и «*LeftAxis\_visual*» делать

не нужно. Чтобы все визуальные объекты заднего моста робота сделать дочерними объектами «*BackBridge*» нужно их сгруппировать. Для этого в окне иерархии объектов сцены, удерживая на клавиатуре кнопку «*Ctrl*», щелкните по каждому левой кнопкой мыши, что приведет к одновременному выбору всех трех объектов: «*LeftAxis\_visual*», «*BackBridge\_visual*» и «*RightAxis\_visual*».

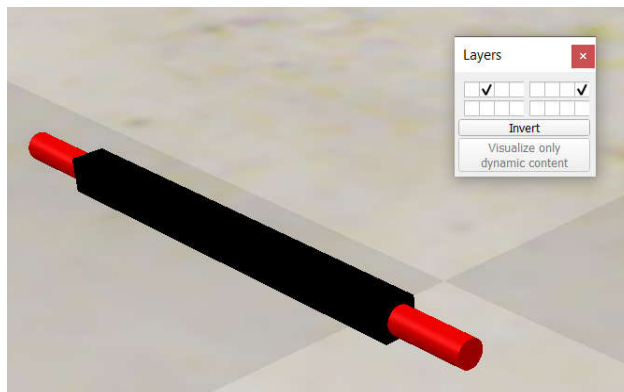


Рис. 7. Отображение визуальных объектов заднего моста робота

Теперь щелкните в рабочей области по любому из них правой кнопкой мыши и во всплывающем меню выберите «*Edit – Grouping/Merging – Grouping selected shapes*» (Редактировать – Группировка/Слияние – Группировка выбранных форм-объектов). После данной операции в окне иерархии вместо трех визуальных объектов останется один групповой объект. Назовите его «*BackBridge\_group\_visual*» и сделайте дочерним объектом «*BackBridge*» (рис. 8). Задний мост робота готов.



Рис. 8. Иерархия объектов заднего моста робота

### Соединение динамических объектов сцены

14. Динамические объекты в среде CoppeliaSim нельзя соединить на основе иерархии. Для их жесткого связывания, когда отсутствуют любые степени свободы (поступательные или вращательные) движения, используется логический объект типа «*Force Sensor*» (Силовой датчик). Он симулирует соединение с порогом усилия и/или крутящего момента. Если в ходе симуляции силовое воздействие на соединение превысит заданный порог силового датчика, то динамические элементы робота разъединятся.
15. Используйте элемент типа «*Force Sensor*» для соединения объектов «*Base*» и «*BackBridge*». Чтобы создать элемент типа «*Force Sensor*» щелкните в рабочей области сцены правой кнопкой мыши и во всплывающем меню выберите пункт «*Add – Force Sensor*» (Добавить – Силовой датчик). Назовите созданный объект «*BaseBridge*». Откройте для него окно «*Scene Object Properties*» и задайте свойство «*Object size*» (Размер (характерный) объекта) [в метрах] равное **0.005**. С помощью свойства «*Force threshold*» (Порог силы) задайте максимальное усилие удерживающее задний мост и платформу вместе

в Ньютонах:  $1e+4$ . Чтобы предотвратить в симуляции вращение этих объектов относительно друг друга по оси силового датчика, установите его свойство «*Torque threshold*» (Порог крутящего момента) равным значению:  $1e+5$  N\*m (Ньютон на метр). Поместите «*BaseBridge*» в точку с мировыми координатами (0; -4.5e-2; 2.8e-2) и отнесите с 3 группе скрытого слоя объектов. После включения 1, 2 и 3 групп скрытого слоя увидим соединяемые динамические объекты: «*Base*»; «*BackBridge*» и элемент типа «*Force Sensor*» - «*BaseBridge*» (рис. 9).

**Примечание.** Чтобы динамические объекты стали полупрозрачными настройте их свойство «*Opacity*». На рисунке 9 свойство «*Opacity*» для объектов «*Base*» и «*BackBridge*» имеет значение 0.6.

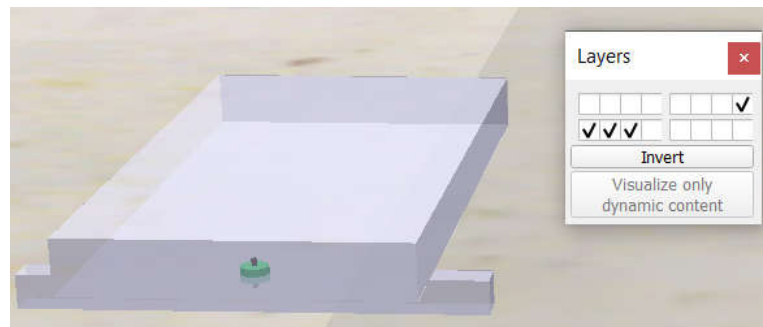


Рис. 9. Соединяемые динамические объекты и элемент типа «*Force Sensor*»

- 16.С помощью окна иерархии объектов сделайте объект «*BaseBridge*» дочерним по отношению к «*Base*», а объект «*BackBridge*» дочерним по отношению к «*BaseBridge*». В окне иерархии должна появиться взаимосвязь, показанная на рисунке 10.



Рис. 10. Связь платформы робота с задним мостом в окне иерархии

## Колеса робота

- 17.Используя всплывающее меню «*Add – Primitive shape - Cylinder*» создайте объект с размерами [в метрах]: диаметр ( $X=Y$ ) =  $5e-2$ ; высота ( $Z$ ) =  $1e-2$ . Назовите объект «*RightWheel\_visual*» (Правое колесо\_визуальное). Поверните колесо вокруг оси  $Y$  на  $90$  градусов и поместите в точку с мировыми координатами ( $4.6e-2$ ;  $-4.5e-2$ ;  $2.5e-2$ ). Отнесите объект к 4 группе визуального слоя. Выключите динамические свойства объекта, произвольно настройте спектральные характеристики и активируйте его специальные свойства. Создайте копию объекта и назовите ее «*LeftWheel\_visual*» (Левое колесо\_визуальное). Поместите объект «*LeftWheel\_visual*» в точку с мировыми координатами ( $-4.6e-2$ ;  $-4.5e-2$ ;  $2.5e-2$ ). На рисунке 11 показано визуальное представление робота после завершения работ на описанном этапе.

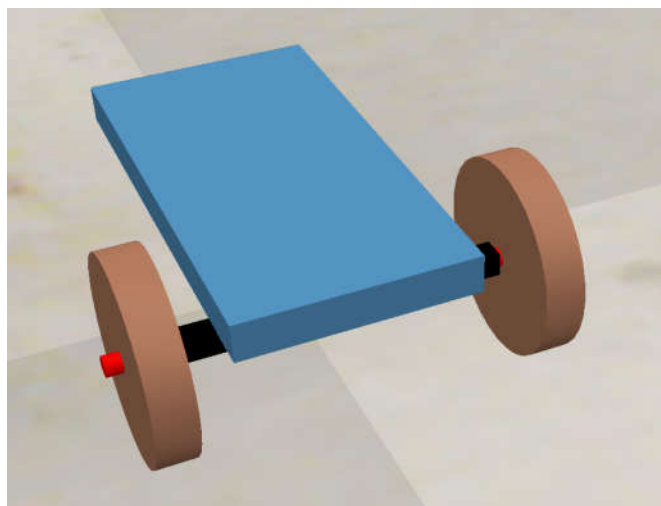


Рис. 11. Представление робота визуальными объектами после монтажа колес на задний мост

18. Теперь необходимо построить динамические объекты колес и связать их с «BackBridge» логическим объектом типа «Revolute». Создайте для каждого визуального объекта-колеса копию, у которой включите динамические свойства и выполните их настройку, используя материал с плотностью **5000** кг/м<sup>3</sup>. Новые объекты назовите соответственно «LeftWheel» и «RightWheel». Выключите у них специальные свойства и отнесите к 4 группе скрытого слоя. Щелкните в рабочей области правой кнопкой мыши и во всплывающем меню выберите пункт «Add – Joint - Revolute» (Добавить - Вращающийся шарнир). Найдите в окне иерархии сцены новый объект «Revolute\_joint» и измените его название на «Right\_joint». Для него откройте диалоговое окно «Scene object properties» и используя закладку «Common» отнесите объект к 4 группе скрытого слоя. Переключитесь на закладку «Joint» и в группе свойств «Visual properties» установите диаметр объекта «Diameter» [m]: **0.005**, а длину «Length» [m]: **0.01**. Затем нажмите кнопку «Show dynamic properties dialog» и сделайте данную ось активной, включив мотор с помощью флажка «Motor enabled». Установите максимальный крутящий момент «Maximum torque» [N\*m]: **2.5e+2**, а угловую скорость вращения колеса «Target velocity» [deg/s](градусы в секунду): **-400**. После чего диалоговое окно «Joint Dynamic Properties» и «Scene object properties» для объекта «Right\_joint» можно закрыть. Далее следует повернуть объект «Right\_joint» вокруг оси Y на **90** градусов и поместить в точку с мировыми координатами **(+4e-2; -4.5e-2; +2.5e-2)**. Сделайте копию «Right-joint». Назовите новый объект «Left\_joint» и поместите его в точку с мировыми координатами **(-4e-2; -4.5e-2; +2.5e-2)**. Осталось прикрепить динамические объекты колес к объекту «BackBridge» с помощью логических объектов «Left\_joint» и «Right\_joint». Для этого постройте их иерархию, аналогичную той, что показана на рисунке 12.

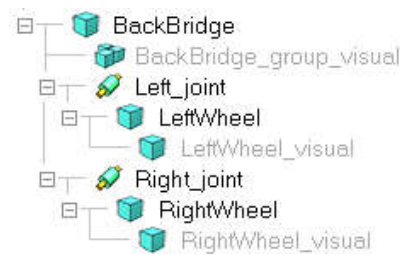


Рис. 12. Иерархия динамических и логических объектов заднего моста

На рисунке 13 показано представление робота динамическими и логическими объектами, которое должно получиться в результате полного монтажа колес на его задний мост.

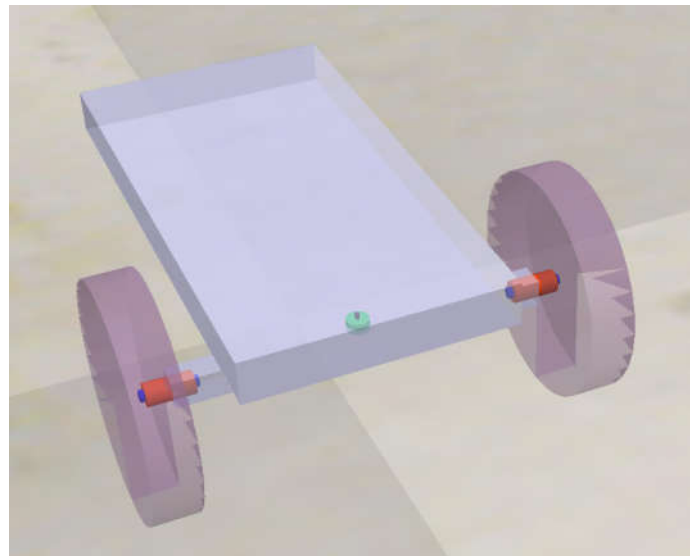


Рис. 13. Представление робота динамическими и логическими объектами после монтажа колес на задний мост

### Передняя опора

19. Платформа робота должна располагаться горизонтально. Поэтому высота передней опоры робота должна равняться сумме радиуса колеса и половины высоты балки заднего моста:  $25 + 3 = 28$  мм. В соответствии с техническими характеристиками передней опоры постройте ее с помощью 2 элементов: сферы диаметром 10 мм и цилиндра такого же диаметра. При этом высота цилиндра должна составить 23 мм.
20. Используя диалоговое окно управления слоями уберите со сцены все объекты, чтобы они не мешали построению передней опоры робота. С помощью всплывающего меню «Add – Primitive Shape - Sphere» (Добавить - Прimitives форма - Сфера) откройте диалоговое окно «Primitive Sphere» (Простая сфера) и введите диаметр сферы в поле «X-size»[m]: **1e-2**. Нажмите «Ok».
21. Добавьте в сцену цилиндр диаметром 10 мм и высотой 23 мм. Сместите его по оси Z на 5 мм. Используя окно иерархии сцены выделите одновременно новые сферический и цилиндрический объекты. На сцене щелкните по од-



ному из них правой кнопкой мыши и во всплывающем меню выберите «*Edit – Grouping/Merging – Merging selected shapes*» (Редактировать - Группировка/Слияние – Объединить выбранные формы). Появится предупреждение «*You are about to merge pure (and non-pure) shapes. Doing so will result in a non-pure shape. Do you want to proceed anyway?*» (Вы собираетесь объединить чистые (и нечистые) формы. Это приведет к нечистой форме. Вы все равно хотите продолжить?). Нажмите «Yes». В результате получится один объект сложной формы. Дайте ему название «*Support*». Отнесите его к 5 группе скрытого слоя. Настройте динамические свойства в соответствии с техническими характеристиками и поместите в точку с мировыми координатами (0; +5e-2; +1.4e-2). Сделайте копию объекта и назовите ее «*Support\_visual*». У нового объекта отключите динамические свойства. Поместите его в 5 группу визуального слоя. Произвольно задайте его спектральные (цветовые) характеристики и активируйте специальные свойства. Постройте иерархию: «*Support*» - родительский объект; «*Support\_visual*» - дочерний объект.

- 22.С помощью «Layers» отобразите на сцене динамические объекты «*Base*» и «*Support*». Добавьте в сцену новый объект типа «*Force Sensor*». Отнесите его к 5 группе скрытого слоя. Дайте название «*BaseSupport*», уменьшите диаметр до величины 1e-2 м и поместите в точку с мировыми координатами (0; +5e-2; +2.8e-2). Задайте значения параметров «*Force threshold*» и «*Torque threshold*» такие же как у «*BaseBridge*». Постройте иерархию объектов: «*Base*», «*BaseBridge*» и «*Support*» для крепления передней опоры к платформе робота. Окончательная иерархия объектов мобильного робота должна соответствовать рисунку 14.

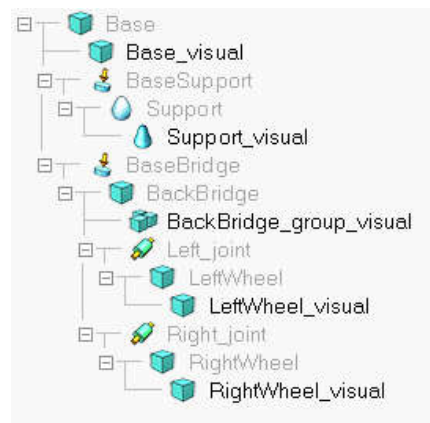


Рис. 14. Финальная иерархия элементов мобильного робота в CoppeliaSim

- 23.Включите отображение только объектов визуального слоя. Изображение мобильного робота должно быть подобно рисунку 1-а.

### Тестирование конструкции робота в движении

- 24.На верхней панели инструментов выберите тип движка симуляции «*ODE*» (рис. 15). Запустите симуляцию. Моторы левого и правого колеса робота настроены на скорость вращения 400 градусов в секунду.



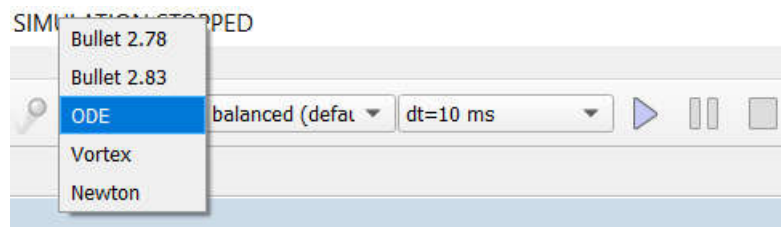


Рис. 15. Выбор движка симуляции «*ODE*» в среде CoppeliaSim

Колесо имеет диаметр  $D = 50$  мм. За один оборот колеса (360 градусов) робот должен перемещаться на расстояние  $L = \pi \cdot D \approx 157$  мм, а при повороте колеса на 400 градусов за 1 секунду линейная скорость робота должна составлять  $v = (157 \cdot 400)/360 \approx 174$  мм/с. В симуляции с движком «*ODE*» движение робота должно быть прямолинейным, т.к. колеса вращаются с одинаковой угловой скоростью, а возмущения внешней среды движок данного типа практически не учитывает. Попробуйте запустить симуляцию с другим типом движка: «*Bullet 2.78*», «*Bullet 2.83*», «*Vertex*», «*Newton*». Как движется робот при использовании других движков симуляции? При любом движке в процессе перемещения робота его конструкция не должна распадаться, а дочерние визуальные объекты должны следовать за динамическими. Во время симуляции откройте динамические свойства элементов «*Right\_joint*» и «*Left\_joint*» и попробуйте поменять абсолютную величину угловой скорости и ее знак. Установите разные угловые скорости вращения колес робота и попытайтесь определить как зависит радиус поворота робота от отношения угловых скоростей правого и левого моторов.

25. По завершении тестирования и корректировки конструкции робота в приложении CoppeliaSim сохраните окончательный проект и продемонстрируйте преподавателю работу Вами построенного робота.

26. Сохраните финальную версию проекта и загрузите на сайт курса в Eluniver.

## 2.2. Самостоятельная работа

### *Проект «Мобильный робот с вращающейся балкой переднего моста».*

1. Создайте новую сцену в среде CoppeliaSim. Дайте ей название в соответствии с шаблоном: `Work1(<Фамилия студента-группа (на латинице)>)-task2.ttt` и сохраните. Например, студент Иванов из группы 1101б должен именовать файл сцены CoppeliaSim: `Work1(Ivanov-1101b)-task2.ttt`
2. Переключитесь на сцену соответствующую **Обучающей части** задания к практической работе и выделите одновременно в окне иерархии все элементы трехопорного мобильного робота. Затем с помощью всплывающего меню скопируйте выбранные элементы в буфер обмена (пункт меню «Edit – Copy selected object»). Вернитесь обратно на сцену **Самостоятельной работы** и используя всплывающее меню вставьте в нее элементы мобильного робота из буфера.

3. Выделите в окне иерархии сцены все объекты (визуальные, динамические, логические), используемые при построении передней опоры, и удалите их из сцены.
4. Сделайте копию всех элементов заднего моста, включая колеса и их шарнирные соединения с балкой, кроме логического элемента типа «*Force Sensor*». Измените названия элементов обоих мостов так, чтобы в них отражалась принадлежность к переднему или заднему мосту.
5. Переместите родительский объект переднего моста в точку, у которой в мировой системе координат значение  $Y$  равно  $+4.5e-2$ . Все дочерние элементы переднего моста должны последовать за родителем.
6. Создайте на балке переднего моста визуальный объект имитирующий поворотный шарнир диаметром 15 мм, высота которого совпадает с высотой балки (рис. 16). Правильно выстройте иерархию объектов, чтобы новый элемент копировал движения передней балки моста.

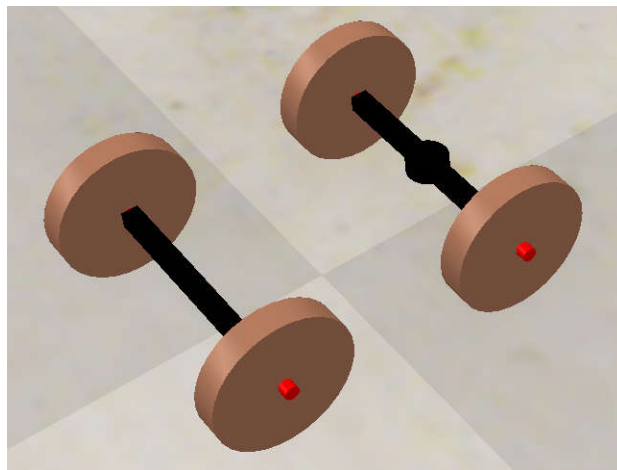


Рис. 16. Передний и задний мост мобильного робота

7. Соедините балку переднего моста с платформой робота шарниром, обеспечивающим вращение моста вокруг оси  $Z$ . Дайте название шарниру: «*Direct\_Motor\_joint*». Представление мобильного робота динамическими и логическими (связующими) объектами должно быть подобно рисунку 17.

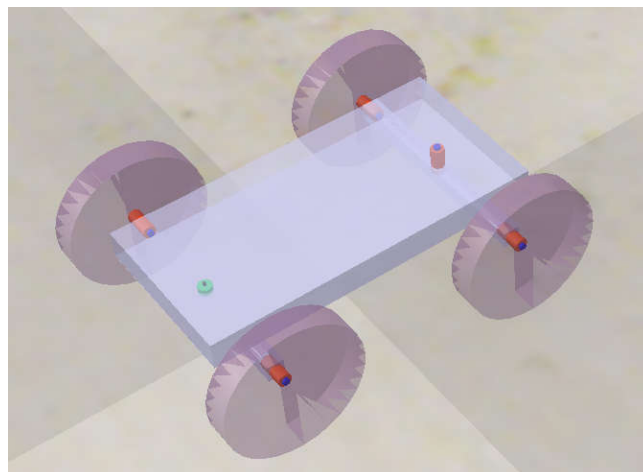


Рис. 17. Представление робота динамическими и логическими объектами

8. Отключите моторы колес заднего моста. Моторы колес переднего моста настройте на работу с одинаковой угловой скоростью.
9. Активируйте мотор шарнира «*DirectMotor\_joint*». Задайте у него нулевую угловую скорость («*Target velocity*»), а момент вращения («*Maximum torque*») задайте равным **250** Н·м. Установите флажок «Loop» (Петля обратной связи) для активации контроля угла поворота вала мотора. Значение параметра «*Target position*» [градусы] установите равным **15**. Это в начале симуляции приведет к повороту переднего моста относительно поперечной оси платформы (X) на угол 15 градусов и заставит двигаться робота по окружности.
10. Запустите симуляцию с движком «*ODE*» и оцените целостность конструкции робота во время движения. Протестируйте работу робота при использовании других движков симуляции. Откройте динамические свойства шарнира «*DirectMotor\_joint*», запустите симуляцию и в процессе ее работы поварьируйте значение параметра «*Target position*» в пределах от -20 до 20 градусов. Как при этом меняется движение робота? Что произойдет, если установить значение «*Target position*» вне обозначенного диапазона?
11. По завершении тестирования и корректировки конструкции робота в приложении CoppeliaSim сохраните окончательный проект и продемонстрируйте преподавателю работу Вами построенного робота.
12. Сохраните финальную версию проекта и загрузите на сайт курса в Eluniver.

### 3. Подготовка отчета, представление и оценка работы

#### Структура отчета

В качестве отчета по каждой части задания необходимо предоставить готовый проект CoppeliaSim. В отчете (проекте CoppeliaSim) оценивается точность параметров модели (названий, значений, иерархии и др.), для которых в задании определена уникальная величина или идентификатор. Произвольно определяемые объекты или их параметры могут иметь любое значение. В задании они оговариваются отдельно.

Загрузку проектов на сайт Eluniver следует выполнять после демонстрации всех частей задания преподавателю. Желательно загружать все проекты одновременно либо отдельными файлами, либо общим архивом.

#### Представление и защита работы

Представлением работы является ее демонстрация преподавателю. Каждый проект может быть представлен отдельно от других частей задания. В ходе представления преподаватель может задать вопрос по любому пункту соответствующей части задания или попросить выполнить какие-либо построения на основе навыков, полученных при разработке проекта. Оценка за представление задания выставляется на основе работоспособности проекта, правильности ответа студента

на вопросы по проекту и готовности выполнить дополнительное задание без использования методического материала.

Защита работы заключается в ответе на два контрольных вопроса, выбранных произвольно преподавателем из списка контрольных вопросов (п.4). Оценивается детальность и точность ответа. Во время ответа пользоваться методическим материалом нельзя. Возможность ответа на контрольные вопросы дается студенту после представления всех частей задания.

#### Структура оценки практической работы

№	Вид оценки	Максимальный балл
1.	Проект «Обучающая часть»	25
2.	Проект «Самостоятельная работа»	25
3.	Отчет «Обучающая часть»	10
4.	Отчет «Самостоятельная работа»	10
5.	Контрольный вопрос 1	15
6.	Контрольный вопрос 2	15
<b>Итого:</b>		<b>100</b>

#### **4. Контрольные вопросы**

1. Какие примитивные формы доступны во встроенной CAD-системе CoppeliaSim?
2. Как добавить новый объект в сцену CoppeliaSim?
3. Какие единицы измерения используются в CoppeliaSim для определения геометрических параметров элементов робота?
4. С помощью какого пункта меню осуществляется доступ к настройке свойств элементов робота?
5. Какие служебные объекты сцены обеспечивают ее отображение? К какому слою относятся такие объекты?
6. Сколько слоев объектов можно использовать при построении конструкции робота?
7. Как определяется радиус примитива типа «Sphere»?
8. Каким образом можно выполнить поворот объекта сцены? Продемонстрируйте на примере.
9. Где находится центр мировой системы координат сцены CoppeliaSim?
10. Каким образом осуществляется управления видом сцены CoppeliaSim?
11. Каким образом осуществляется управление отображением элементов робота в сцене CoppeliaSim?
12. Сколько групп объектов можно задать для каждого слоя?
13. Какой инструмент позволяет перемещать объекты в сцене CoppeliaSim?
14. Как осуществляется активация динамических свойств объектов?
15. Каким образом можно задать массу и моменты инерции элемента робота?
16. Какое свойство объекта нужно активировать, чтобы он мог взаимодействовать с другими элементами сцены?

17. Какое свойство объекта позволяет контролировать с помощью ультразвукового датчика?
18. При активации флажка «Renderable» каким свойством начинает обладать объект сцены?
19. Какие спектральные свойства объекта сцены CoppeliaSim Вам известны? Поясните их назначение.
20. Где следует искать пиктограмму управления иерархией сцены CoppeliaSim?
21. Для чего используется иерархия объектов в сценах CoppeliaSim?
22. С какими видами логических объектов Вы познакомились в рамках Практической работы №1?
23. Каким образом в среде CoppeliaSim осуществляется разделение визуальных объектов и динамических?
24. Зачем в среде CoppeliaSim осуществляется разделение объектов на визуальные и динамические?
25. К какому слою относятся динамические объекты сцены?
26. Как заставить визуальный объект копировать движения динамического объекта в сцене CoppeliaSim?
27. Для чего предназначены объекты типа «ForceSensor»?
28. Какие геометрические характеристики имеет объект типа «ForceSensor»?
29. Где осуществляется настройка свойства «Torque threshold»? Каково его назначение?
30. Как в CoppeliaSim можно учесть максимальное усилие, превышение которого приведет к разрушению робототехнической системы?
31. Каким образом в среде CoppeliaSim можно создавать объекты сложной формы?
32. Для чего осуществляется группировка объектов сцены? Каков должен быть алгоритм действий по группировке объектов?
33. Можно ли задать разные спектральные свойства у элементов объекта, построенного в результате группировки? Продемонстрируйте на примере?
34. Как осуществляется слияние нескольких примитивных форм в CoppeliaSim? Продемонстрируйте на примере.
35. После слияния форма построенного объекта является простой или сложной, правильно или неправильной?
36. Какая форма динамических объектов некорректно обрабатывается симулятором CoppeliaSim?
37. Какие виды шарнирных соединений используются в среде CoppeliaSim?
38. Как активировать мотор шарнирного соединения?
39. Какое свойство отвечает за скорость вращения шарнира?
40. Для чего используется флажок «Loop» и какие возможности он открывает для управления шарнирным соединением?
41. Что понимается под движком симуляции? Какими типами движков Вы пользовались при выполнении заданий настоящей практической работы?
42. Какие параметры управляют симуляцией в среде CoppeliaSim?