

#09, сентябрь 2015

УДК 681.527

Методика разработки управляемых 3D моделей сложных электромеханических систем на примере цилиндрического редуктора

***Зубков А. Ю.**, студент*

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
кафедра «Специальная Робототехника и Мехатроника»*

***Пахомов Т. К.**, студент*

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
кафедра «Специальная Робототехника и Мехатроника»*

Научный руководитель: Перминова Е.А., к.т.н., доцент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
кафедра «Элементы приборных устройств»*

Научный руководитель: Бошляков А.А., к.т.н., доцент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
кафедра «Специальная Робототехника и Мехатроника»*

kafsm7@sm.bmstu.ru

Введение

На кафедре РЛ5 студенты выполняют лабораторную работу на тему: «Теоретическое и экспериментальное определение КПД многоступенчатого цилиндрического редуктора». Реальный стенд (Рис. 1.) находится в эксплуатации долгое время, периодически выходит из строя и требует частого обслуживания. Поэтому, было принято решение разработать виртуальную модель стенда, и проводить лабораторные работы на компьютере. Была поставлена задача создать компьютерную модель реальной лабораторной установки и на ее основе комплексной мультимедийной программы всей лабораторной работы.

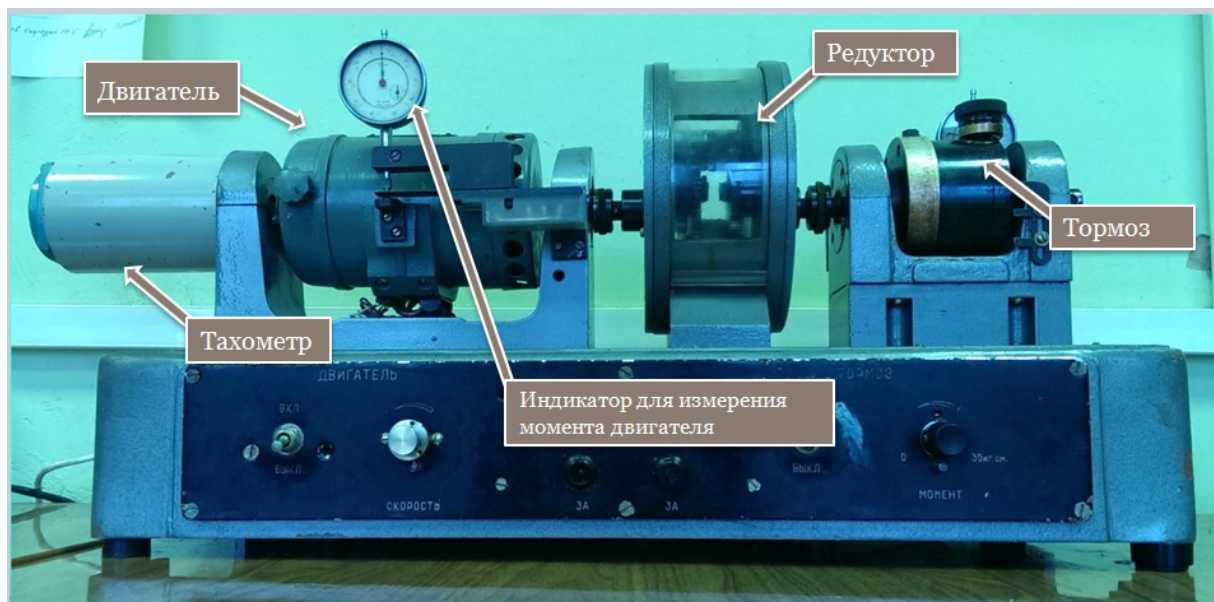


Рис. 1. Реальный стенд цилиндрической передачи

Вопрос построения 3D модели виртуального стенда в *MatLab* уже рассматривался в статье «Моделирование лабораторной работы по курсу ОКП «Исследование КПД прямозубого зубчатого редуктора»» [1], но данный метод требует очень больших ресурсов от компьютера. Так же аналогичная задача, но уже с использованием пакета MBTU, для визуализации лабораторных стендов планетарного и червячного редуктора рассмотрена в статье «Использование *Irrlicht Engine* для визуализации работы лабораторных стендов по исследованию КПД редукторов»[2]. В данной работе для решения проблемы визуализации предлагается использовался метод с использованием пакета MBTU, *SolidWorks* и графического движка *Irrlicht Engine*, который управляется с помощью MBTU. Целью данной работы является изложение методики построения виртуального стенда.

Дадим краткую характеристику программных пакетов, используемых для решения данной задачи. А также рассмотрим методику разработки управляемых 3D моделей:

1. Построение 3D моделей при помощи программных пакетов *SolidWorks* и *Autodesk 3ds Max*, с целью получения объектов для дальнейшей работы в *Irrlicht Engine*

SolidWorks

SolidWorks — программный комплекс, используемый для автоматизации работ промышленного предприятия на этапах конструкторской и технологической подготовки производства. Обеспечивает разработку изделий любой степени сложности и назначения.

В данной работе программа SolidWorks использовалась для создания конструкторской документации, а так же разработки на ее основе 3D модели (Рис.2), которая позже с использованием среды Autodesk 3ds Max была перенесена для использования в графическом движке Irrlicht Engine (Рис.3а, Рис.3б).

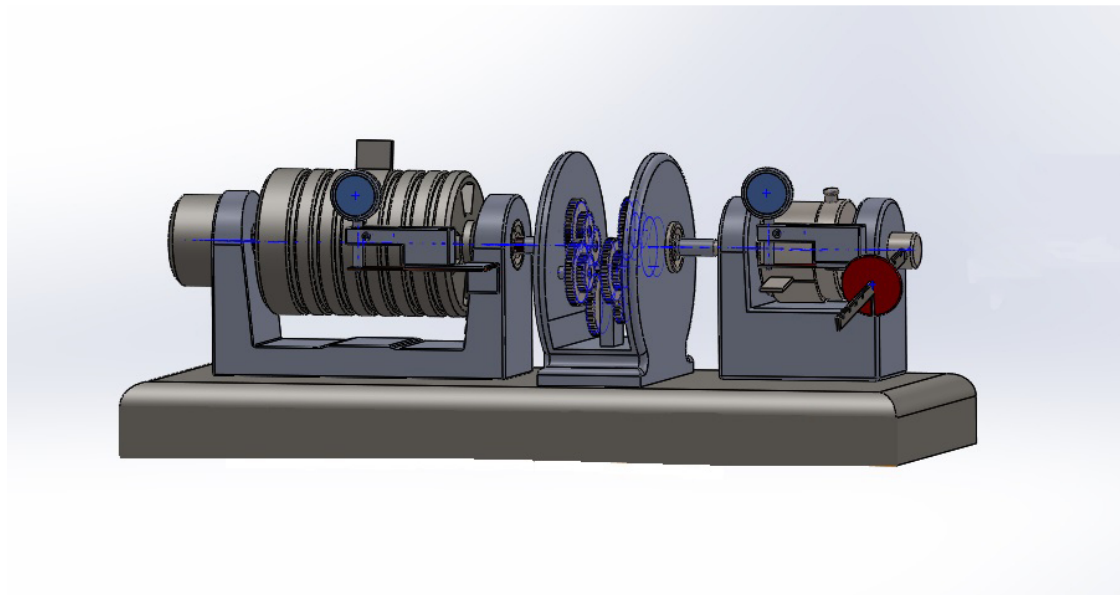


Рис.2. 3D-модель лабораторного стенда, разработанного в SolidWorks.

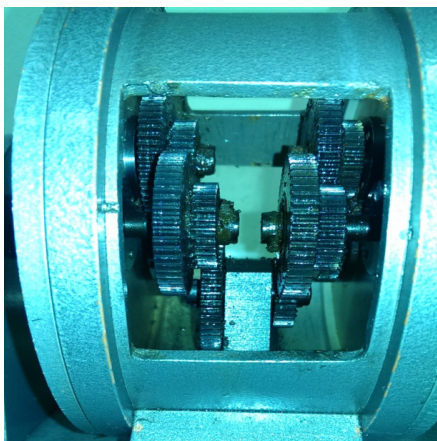


Рис.3а Реальный редуктор

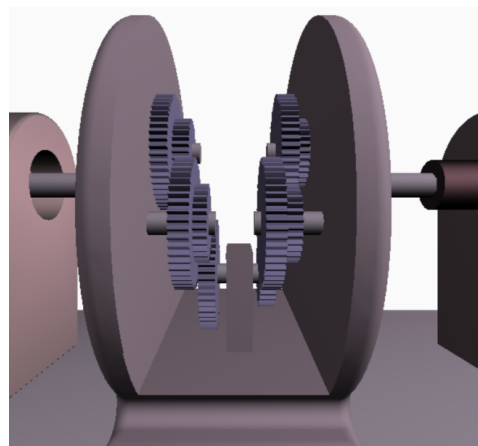


Рис. 3б Модель редуктора, разработанная в Irrlicht

Autodesk 3ds Max

Autodesk 3ds Max – это компьютерная программа предназначенная для моделирования объектов в 3D. Применяется данное программное обеспечение для реализации самых разнообразных задач. Используется для разработки компьютерных игр.

В состав программы входит большое количество различных библиотек изображений, текстур и объектов. С помощью данного программного обеспечения легко достигается реалистичность изображения. При этом не возникает потребности в многочисленной и длительной настройке. Содержит самые современные средства для художников и специалистов в области мультимедиа. Написана на C#.

В данной работе Autodesk 3ds Max использовался в качестве программы, преобразующей 3D-модели объектов, полученных в SolidWorks.

Над 3D-моделями производились следующие действия:

- задание начальных координат, которые в последствии использовались для работы в Irrlicht;
- задание ориентации 3D-модели в пространстве, что значительно облегчило повторную сборку виртуального стенда в Irrlicht;
- задание начального цвета поверхностей 3D-моделей. Таким образом обеспечивалась возможность однородной покраски стенда в программном коде, разработанном для визуального движка;
- также Autodesk 3ds Max позволяет перевести 3D-модели из формата STL, в котором сохраняет детали программа SolidWorks, в формат OBJ, требуемый для работы в Irrlicht.

2. Математическое моделирование лабораторной установки и реализация пользовательского интерфейса

МВТУ

Для создания математической модели стенда использовался программный комплекс МВТУ.

МВТУ (“Моделирование в технических устройствах”) - современная среда интеллектуального САПР, предназначенная для детального исследования и анализа нестационарных процессов в системах автоматического управления, в ядерных и тепловых энергоустановках, в следящих приводах и роботах, в любых технических системах, описание динамики которых может быть реализовано методами структурного моделирования. Является альтернативой программным продуктам SIMULINK, VisSim, MATRIX и др.

МВТУ обладает низкими системными требованиями, за счет чего достигается большое быстродействие на маломощных ПК, при этом данное ПО обладает большим набором возможностей.

На основе дифференциальных уравнений и анализа установки была составлена математическая модель станда (Рис.4), которая включает в себя двигатель постоянного тока, тормозную муфту и редуктор. В данной установке используется шестиступенчатый цилиндрический редуктор (Рис.3а,3б) с прямозубыми зубчатыми колесами, с числом зубьев $Z_1 = 31$ и $Z_2 = 53$.

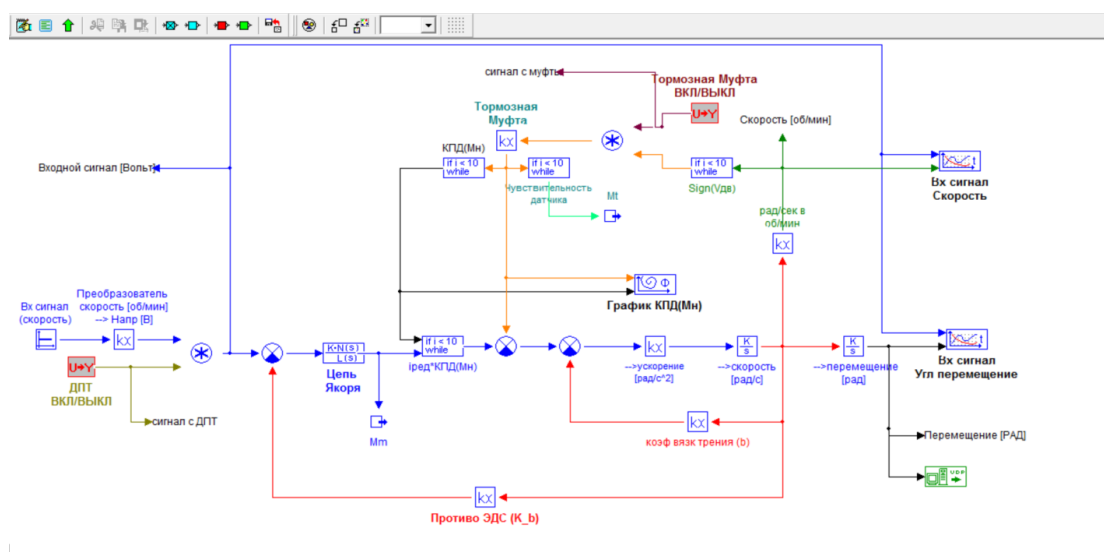


Рис. 4. Математическая модель станда

С помощью встроенного языка программирования был составлен блок, рассчитывающий КПД редуктора.

Так же был составлен блок для градуировки двигателя и тормоза, обработки экспериментальных данных, и проверки полученных значений(Рис.5).

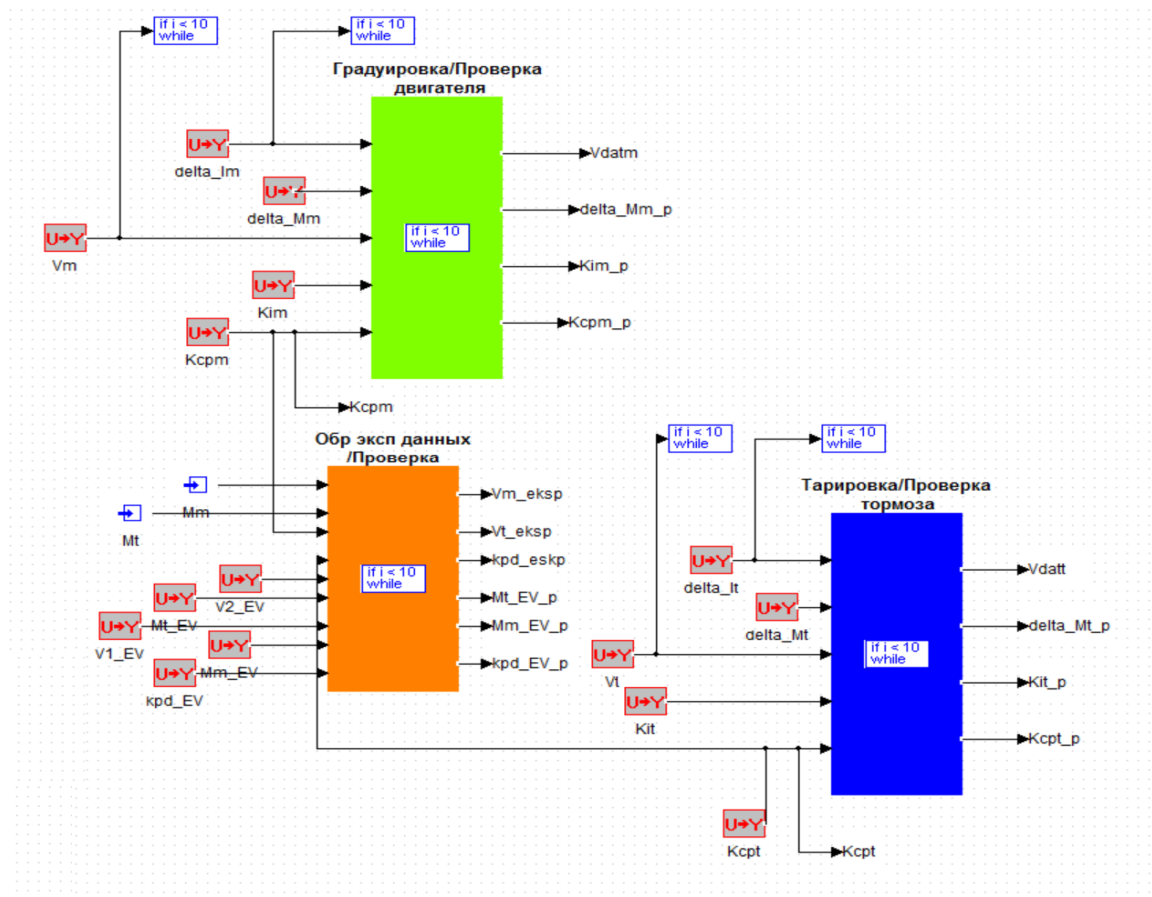


Рис. 5. Блок обработки данных, градуировки и проверки значений

Градуировка производится с помощью специальной интерактивной таблицы, в которую студент заносит необходимые значения и индикаторов момента (Рис. 6).

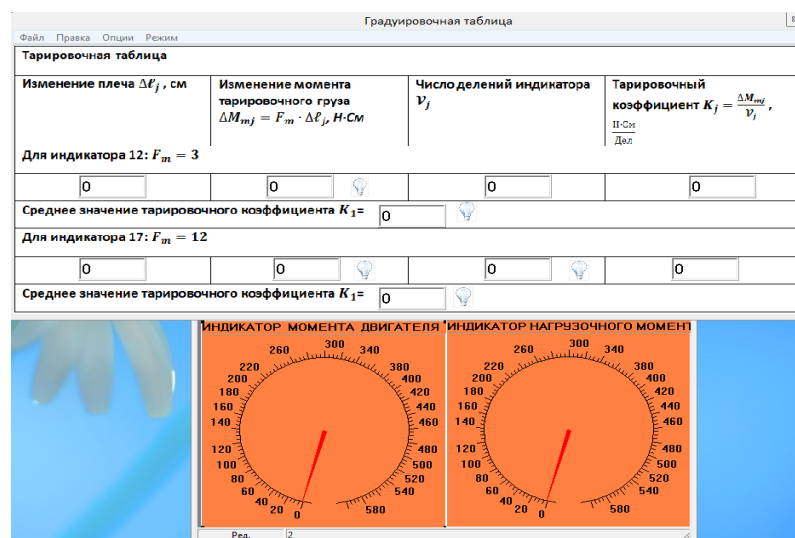


Рис. 6. Градуировочная таблица и индикаторы моментов

Взаимодействие с виртуальным стендом осуществляется при помощи специального пульта (Рис.7), на котором находятся элементы управления и наблюдения за показаниями соответствующих значений (тахометр, датчик момента двигателя, датчик момента тормоза).

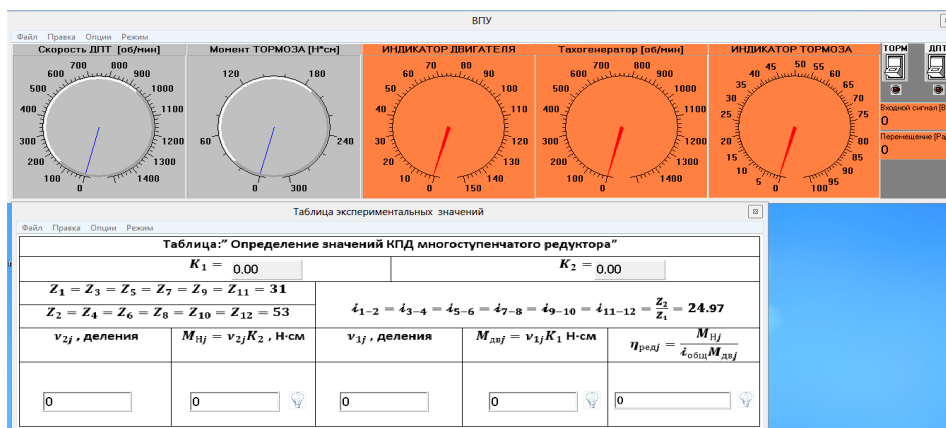


Рис. 7. Виртуальный пульт управления

3. *В графическом движке Irrlicht был реализован программный код, отвечающий за работу виртуальной лабораторной установки*

Irrlicht

Irrlicht (Irrlicht Engine) — высоко производительный трёхмерный графический движок, написанный на C++, который находится в свободном доступе и распространяется на условиях лицензии *zlib*. Содержит в себе мощное высокоуровневое *API* для создания 2D и 3D приложений использует возможности *OpenGL*, *DirectX* и нескольких собственных рендеров.

Главные возможности и особенности:

- Высокопроизводительная 3D визуализация посредством *Direct3D* и *OpenGL*.
- Платформенная независимость, запускается на *Windows*, *Linux*, *OSX*, *Solaris* и др.
- Импорт многих известных форматов моделей: *Maya (.obj)*, *3DStudio (.3ds)* и др.

Структура движка Irrlicht

Движок содержит следующие пространства имен (они же модули):

Модуль *irr* содержит в себе следующие пространства имен:

core — основные математические и вспомогательные функции такие как *vectors*, *planes*, *arrays*, *lists* и т. п.

gui — содержит полезные классы для упрощения создания графического интерфейса пользователя *GUI*.

scene — в этом модуле сосредоточено управление сценой: загрузка мешей (*Mesh*), специальные узлы сцены (так как *octrees* и *billboards*), камеры.

video — в этом модуле содержатся классы для доступа к видеодрайверу. Весь 2d и 3d рендлинг происходит здесь.

Основное понятие в *Irrlicht* – нода. Нод – это «объект». Объектом в *Irrlicht* является почти все – модели, камеры, источники света.

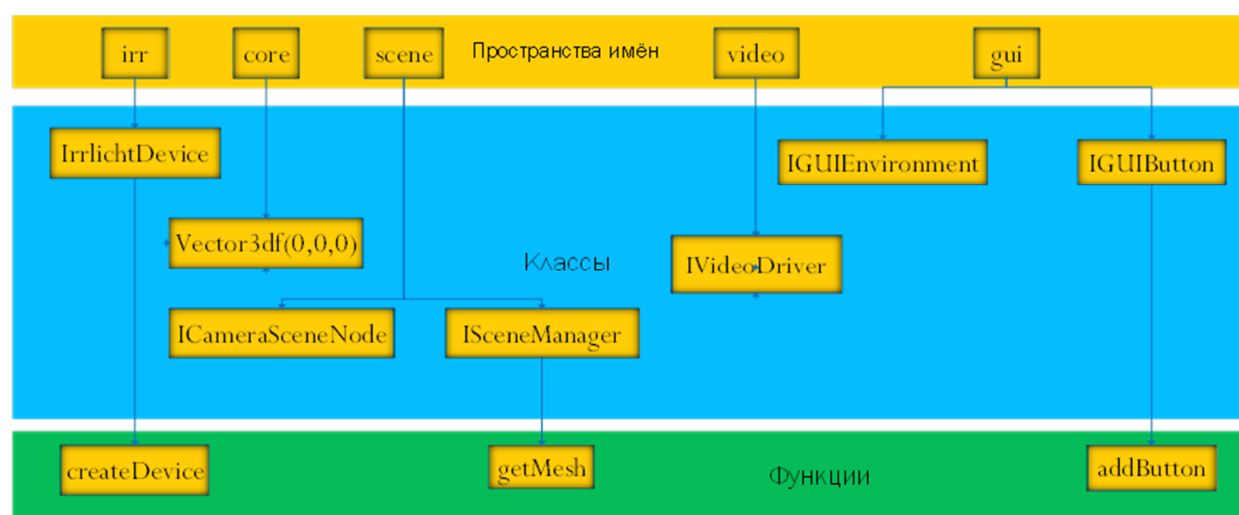


Рис. 8. Пространства имен, некоторые классы и функции *Irrlicht*

Самая главная функция движка – это *createDevice()*, ей создается корневой объект движка *IrrlichtDevice*. Функция *createDevice()* имеет 7 аргументов

deviceType: Тип устройства.

windowSize: размер окна в оконном режиме или разрешение экрана в полноэкранном режиме.

bits: Количество бит для цвета на один пиксель. Может быть 16 или 32. Параметр часто игнорируется в оконном режиме.

fullscreen: определяет в оконном или полноэкранном режиме будет запущено приложение

stencilbuffer: включает трафаретный буфер (для рисования теней).

vsync: включает вертикальную синхронизацию, полезна для полноэкранного режима.

eventReceiver: объект – обработчик событий. Если в нем нет необходимости, ставить равным нулю.

Пример данной функции:

```
device = createDevice( EDT_OPENGL,dimension2d<u32>(1270, 714), 32, false, false,  
false, 0);
```

Формирование сцены, освещения и создание интерфейса в разработанном ПО

Для создания управляемой 3D модели станда была написана программа на C++ с использованием классов и функций *Irrlicht*. Сначала была разработана модель в *SolidWorks* (Рис.2), затем каждая деталь по отдельности была загружена в рабочее пространство *Irrlicht*.

Первоначально был создан массив нод типа *IAnimatedMeshSceneNode*. Затем производилась загрузка детали из указанной папки.

С помощью функций графического движка устанавливалась позиция и ориентация деталей в пространстве. Для формирования лабораторного станда было загружено 60 деталей.

Для создания интерфейса был написан класс, который отслеживал, какое событие произошло: была нажата кнопка на клавиатуре или виртуальная кнопка. Было создано 8 виртуальных кнопок: одна отвечает за выход из программы, другая за просмотр тарифовочной таблицы и 6 кнопок переключают положение стационарной камеры.

Ниже представлены фотографии станда разработанного в *Irrlicht* (Рис. 9).

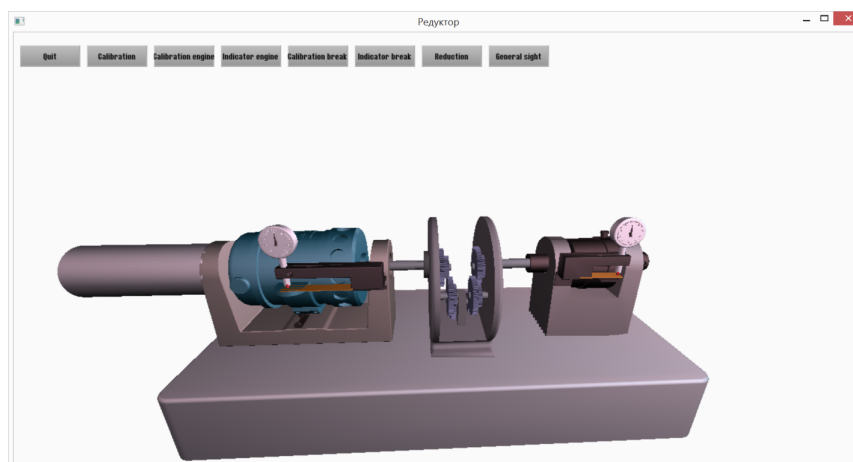


Рис. 9. Виртуальный стенд цилиндрической передачи в *Irrlicht*, основной вид.

Главный цикл программы, в котором происходит вывод на экран *while(device->run())*. В этом цикле выполнялись функции, которые позволяли получить данные из MBTU, а именно: угол поворота выходного вала редуктора и изменение плеча двигателя и тормоза при тарировке, а также показания индикаторов. Далее эти данные передавались в функцию, отвечающую за анимацию на сцене, и нужные детали двигались.

4. *Настройка информационного обмена между MBTU и Irrlicht с помощью протокола UDP и текстового файла*

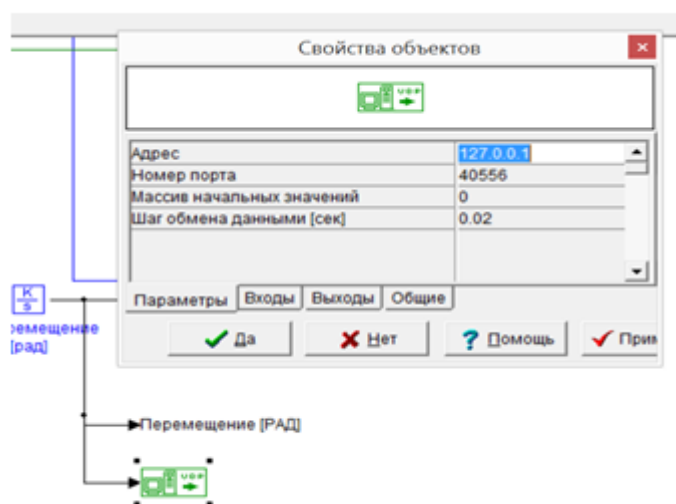


Рис. 10. Настройка *UDP* в MBTU

Связь MBTU и *Irrlicht*, осуществлялась с помощью протокола *UDP* и текстового файла. *UDP* — простой, основанный на сообщениях протокол без установления соединения, протокол такого типа не устанавливает выделенного соединения между двумя хостами. Связь достигается путем передачи информации в одном направлении от источника к получателю без проверки готовности или состояния получателя. По протоколу *UDP* передается угол поворота выходного вала редуктора. Для этого в окне “Свойства объектов” (Рис. 10) был изменен адрес и номер порта. Использование адреса 127.0.0.1 позволяет устанавливать соединение и передавать информацию для программ-серверов, работающих на том же компьютере, что и программа-клиент.

Изменение плеча двигателя и тормоза при тарировке, а также показания индикаторов, передавались через текстовый файл, так как этим данным не требуются низкие задержки при передаче данных.

Заключение

В ходе выполнения работы была изложена методика создания виртуального лабораторного стенда, на примере которого разработана 3D модель, визуализировано движение цилиндрической передачи, тарировка датчиков, работа индикаторов. Реализованная управляемая 3D модель лабораторного стенда не требует значительных ресурсов от компьютера, позволяет получить хорошую детализацию и дает возможность управлять 3D объектами с помощью математической модели.

Рекомендуемые требования, предъявляемые к компьютеру:

- ОС: *Windows* ® 8 / *Windows* ® 7 / *Vista* / *Vista64*
- Процессор: *Intel Core 2 Duo P8400* (2,26 ГГц)
- Оперативная память: 4 ГБ
- Видеокарта: *DirectX 9* совместимая видеокарта с памятью 512 МБ, *NVIDIA GeForce 9600M GT* или лучше
- Жёсткий диск: Не менее 200 МБ свободного места

Список литературы

1. Остапенко Д.Г., Мурзин И.А. Моделирование лабораторной работы по курсу ОКП «Исследование КПД прямозубого зубчатого редуктора» // Молодёжный научно-технический вестник. Электрон. журн. 2013. № 10. Режим доступа: <http://sntbul.bmstu.ru/doc/628122.html> (дата обращения 25.05.2015).
2. Бошляков И.А., Коновалов К.В., Метасов И.Е., Шереужев М.А. Использование *Irrlicht Engine* для визуализации работы лабораторных стендов по исследованию КПД редукторов // Молодёжный научно-технический вестник. Электрон. журн. 2013. № 10. Режим доступа: <http://sntbul.bmstu.ru/doc/733764.html> (дата обращения 25.05.2015).
3. Гебхардт Н. Обучающие материалы по графическому движку *Irrlicht Engine*. Режим доступа: http://irrlicht.ru/?page_id=23 (дата обращения 25.05.2015).
4. Шилдт Г. Полный справочник по C++: пер. с англ. / под ред. С.Н. Тригуб. М.: Издательский дом «Вильямс», 2006. 800 с. [Schildt H. C++: The Complite Reference New York : Academic Press, 2003.].