

Разработка в MATLAB-Simulink модели визуализации в виртуальной реальности управляемого пространственного движения подводного аппарата

77-30569/234292

10, октябрь 2011

Скобелев М. М.

УДК 004.946; 519.876.5; 629.584

МГТУ им. Н.Э. Баумана
mskb1v@mail.ru

Введение

Как правило, разработка моделей реалистичной визуализации динамики различных сложных технических устройств в рабочей обстановке очень трудоемка и требует достаточно искусных навыков программирования на языках высокого уровня, таких, например, как Visual C++ с применением библиотеки OpenGL или, например, средства управления и рисования графических элементов DirectX. Существует множество различных программ, с помощью которых можно делать достаточно сложные 3D-сцены виртуальной рабочей обстановки, например Vega, LynX, OpenGVS, OpenGL Performer, и далее подключать к моделям динамики опять же посредством программного интерфейса на VisualC++.

Для упрощенного решения учебных и презентационных задач нет необходимости и зачастую даже возможности идти таким непростым путем. В то же время для разработчика (но не программиста!) технического устройства была бы кстати модель визуализации динамики этого устройства в виртуальной рабочей обстановке (хотя бы для ускоренной демонстрационной проверки правильности работы модели динамики устройства). В качестве варианта решения этой актуальной задачи средствами упрощенной (полностью непрограммистской) разработки в рамках одной среды MATLAB автору представляется приложение Virtual Reality Toolbox с редактором V-Realm Builder2 (сокращенно VRBuilder2) (для модели визуализации) и приложение Simulink (для модели динамики объекта) [1].

На кафедре СМ-11 «Подводные роботы и аппараты» факультета «Специальное машиностроение» МГТУ им. Баумана пакет MATLAB и приложение Simulink успешно используются в учебном процессе и НИР уже на протяжении многих лет. Однако

приложение Virtual Reality Toolbox с редактором VRBuilder2 было практически задействовано в программе кафедральной учебной дисциплины «Моделирование и исследование роботов и робототехнических систем» для старших курсов специальности кафедры СМ-11 сравнительно недавно. При этом автор, ведущий данную учебную дисциплину, зарегистрировал в Роспатенте на МГТУ им. Н.Э.Баумана свою авторскую программу для ЭВМ – программу имитационной модели системы управления движением автономного необитаемого подводного аппарата с визуализацией движения [2].

1. Создание 3D-сцены.

1.1 Редактор VRBuilder2

Желаемая 3D-сцена создается с помощью команд редактора VRBuilder2, который устанавливается на компьютер при установке пакета MATLAB. В VRBuilder2 можно как самому создавать 3D-модели, так использовать уже готовые встроенные библиотеки элементов объектов, текстур и материалов. Следует заметить, что создание сложных 3D-моделей непосредственно в VRBuilder2 достаточно долгий и неудобный процесс. Гораздо легче дополнять библиотеку VRBuilder2 моделями объектов из пакетов CATIA, 3D Studio Max, AutoCAD, а далее размещать их в созданной 3D-сцене. Для дополнения библиотеки VRBuilder2 собственными моделями из других пакетов необходимо экспортировать их файлы с предварительным конвертированием, например, конвертором 3D Explorator, так как редактор VRBuilder2 работает только с расширением файлов VRML.

Простой понятный интерфейс VRBuilder2 в совокупности со всеми необходимыми инструментами и некоторыми достаточно продвинутыми функциями дают возможность в самые короткие сроки по времени создать полноценную 3-D сцену из элементов библиотек материалов, текстур и объектов.

Запуск редактора VRBuilder2 осуществляется следующим образом: нужно запустить файл `vrbuild2.exe`. Он находится в директории `MatLab6p5\toolbox\vr\vrealm\program\vrbuild2.exe`. Или можно запустить данный редактор из окна настроек блока VR Sink (о том, как это сделать, будет сказано ниже). Внешний вид редактора с готовой 3D-сценой виртуального подводного мира и моделью подводного аппарата приведён на рис. 1.

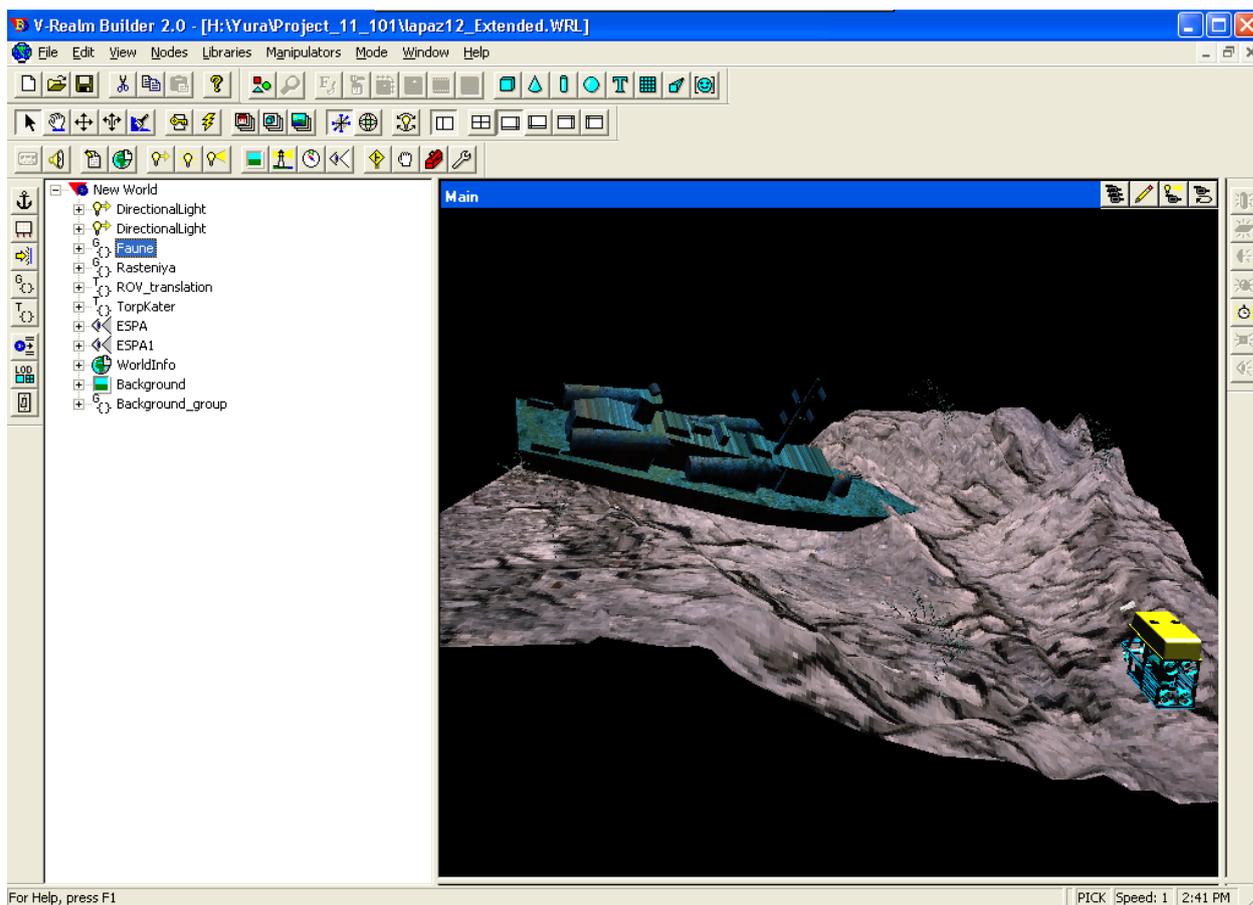


Рис. 1

Из достаточно большого разнообразия команд и функций можно выделить 2 типа: свойства самих объектов и общие функции и элементы. К собственным свойствам объекта можно отнести координаты, масштабный коэффициент и т.д., то есть те, которые принадлежат непосредственно конкретному объекту (их можно увидеть на рис. 2 – в иерархическом дереве 3-D сцены). Общие свойства и элементы – это камеры, источники света, средства управления туманом, базовые элементы рисования и т.д. они находятся на панели инструментов редактора. Основными для решения визуализационных задач представляются функции управления перемещением/вращением объектов как всей сцены, так и отдельных её частей, а также добавление объектов, текстур, материалов, управление видами. Также есть функции добавления и управления туманом, видимостью и освещенностью объектов. Все эти команды были использованы в первую очередь при создании 3D-сцены виртуального подводного мира (дно, источники света, подводный аппарат – все объекты, расположенные в пространстве друг относительно друга образуют 3D-сцену).

1.2 Параметры модели объекта устройства, добавленного в 3-D сцену.

Далее к готовой сцене следует добавить модель объекта устройства, в данном конкретном случае модель подводного аппарата Аква-ЧС. 3D-модель подводного аппарата был предварительно создана в пакете AutoCad, отредактирована в 3D Studio Max и далее сохранена (прямо в пакете 3D Studio Max) как файл с именем ROV_translation и расширением VRML.

На рис. 2 представлены параметры объекта ROV_translation, которые можно изменять в VRBuilder2 и позднее управлять ими, используя Virtual Reality Toolbox.

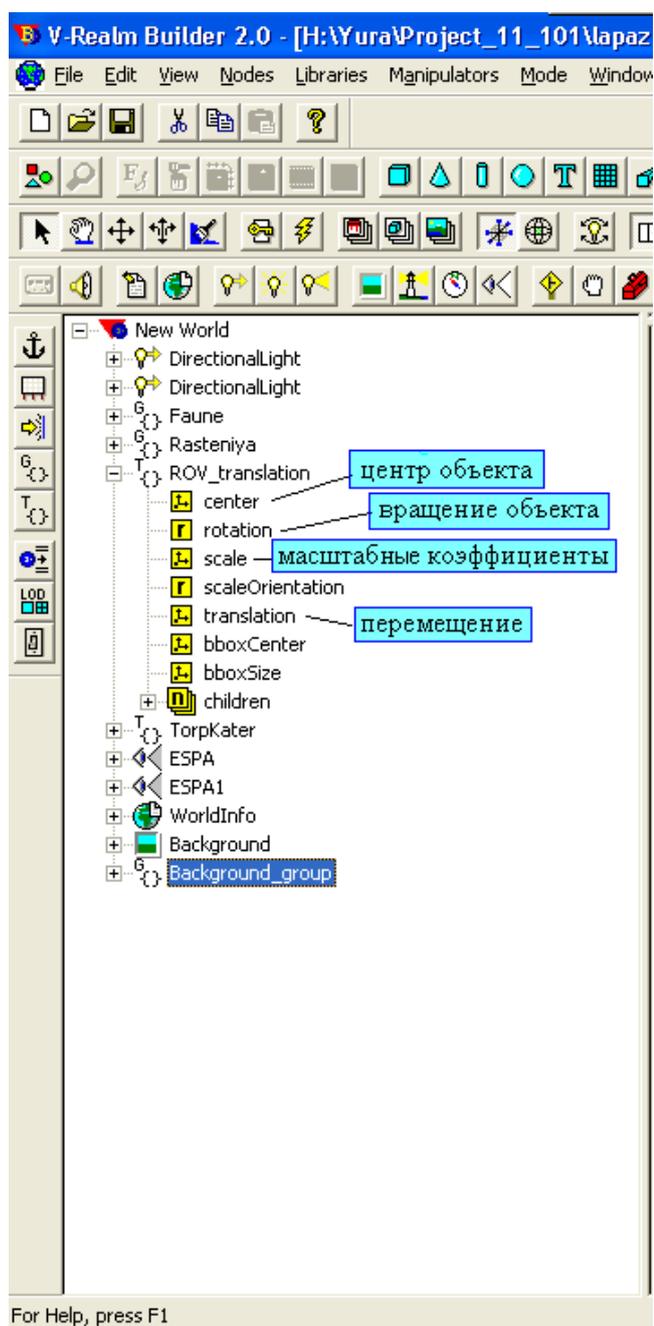


Рис. 2

На рис. 3 показан сам подводный аппарат Аква-ЧС, разработанный в НИИ специального машиностроения МГТУ им Н.Э. Баумана.

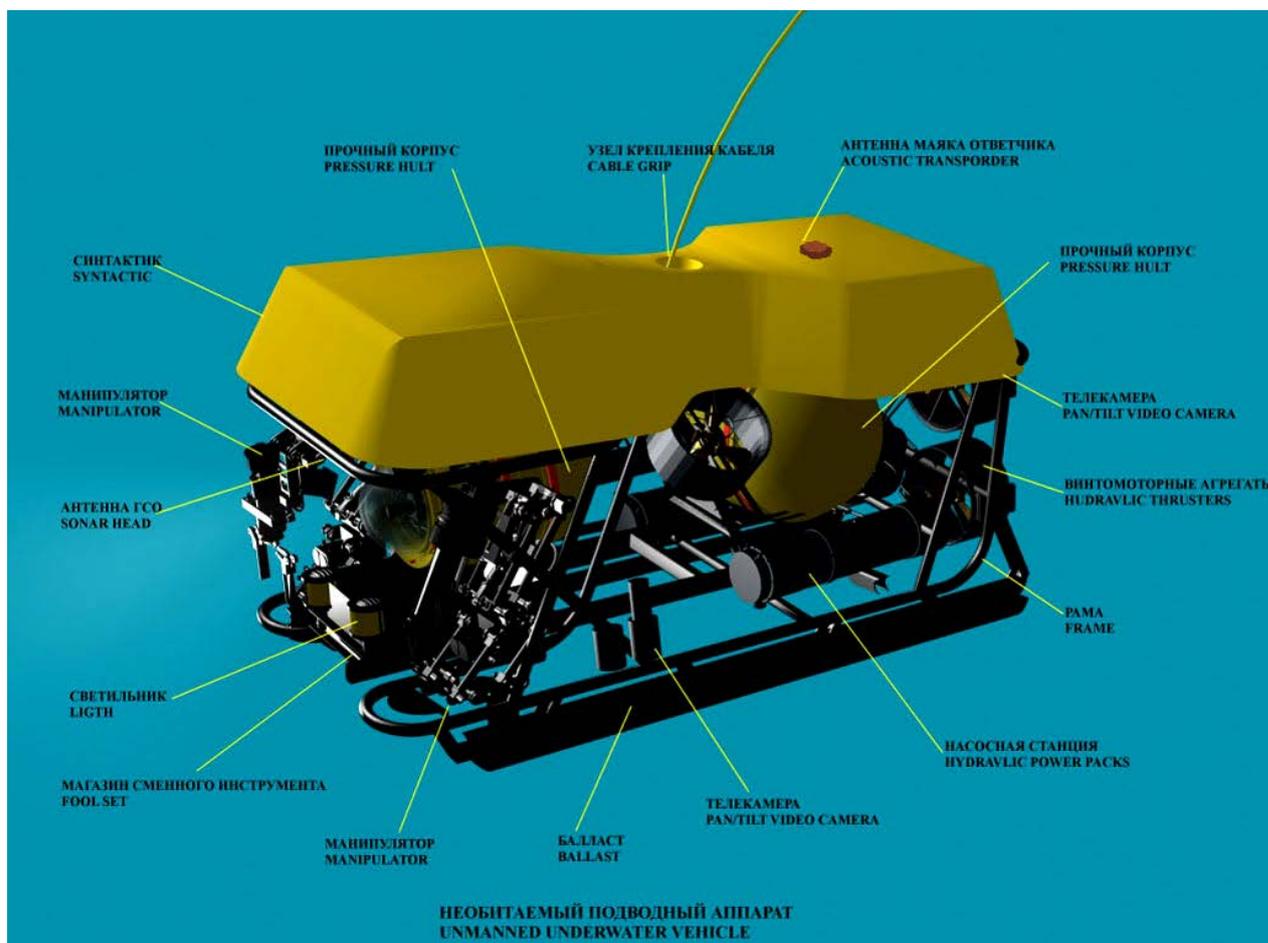


Рис. 3

У объекта ROV_translation существует собственная иерархия его частей (эти части на рис. 3 не показаны, они находятся в “свернутом” состоянии: ROV_translation → children. Они не играют важной роли, так как это просто набор компонентов, из которых состоит данный подводный аппарат). Также есть некоторые общие параметры, меняя которые, можно менять параметры всего объекта, что как раз и требуется:

- Center - координаты на 3D-сцене центра объекта по осям X, Y, Z. Каждый объект упрощенно представляется редактором как параллелепипед, и геометрический центр этого параллелепипеда и есть центр объекта.
- Rotation – величины углов, относительно которых повернут объект, вокруг осей X, Y, Z.
- Scale – масштабные коэффициенты по осям X, Y, Z.
- Translation – изменение положения объекта.

Во всех параметрах объекта, которые перечислены выше, есть возможность изменения координат по 3-м осям: X, Y, Z. В случае Rotation координаты по осям X, Y, Z задают радиус-вектор вращения объекта относительно центра Center. Эти (вышеперечисленные) параметры являются внешними переменными для модели визуализации, именно это и даёт возможность управлять изменением положения объектов сцены с помощью моделей динамики в Simulink через интерфейс Virtual Reality Toolbox.

Итак, при наличии 3D-сцены далее встаёт вопрос о том, как её подключить к модели динамики в Simulink. Для этого используется приложение Virtual Reality Toolbox.

2. Приложение Virtual Reality Toolbox

Приложение Virtual Reality Toolbox позволяет подключить 3D-сцены с моделями объектов к моделям динамики объектов, созданным и работающим в Simulink.

Библиотека vrlib приложения Virtual Reality Toolbox включает в себя 6 основных блоков (рис. 4): VR Sink; VR Source; VR Placeholder; VR Signal Expander; Joystick Input; Magellan Space Mouse;

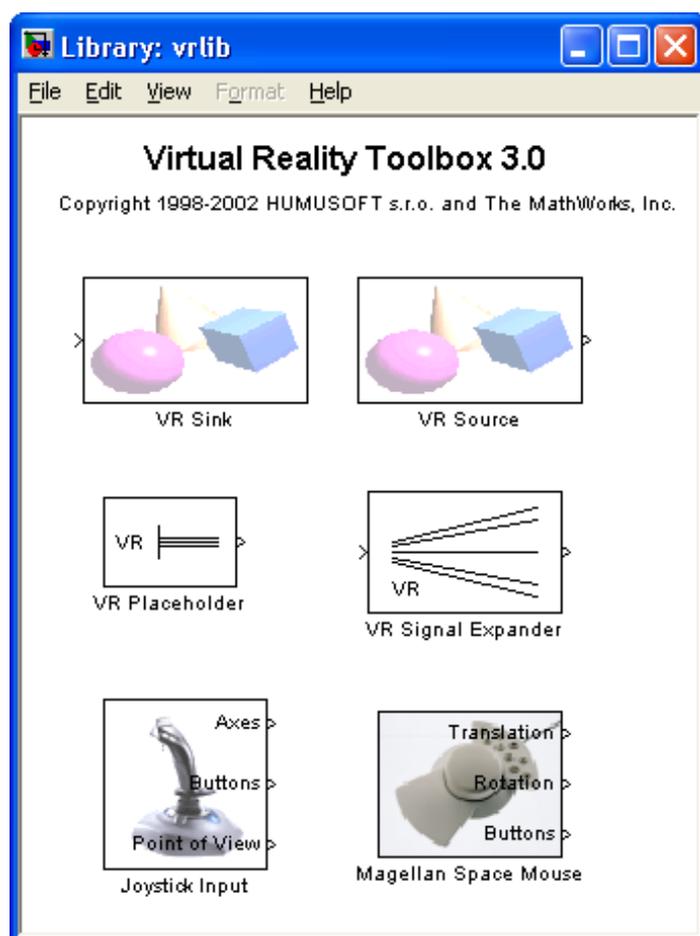


Рис. 4

Информацию по назначению и использованию вышеперечисленных блоков можно найти в электронной документации Help пакета MATLAB.

3. Подключение модели визуализации к модели динамики в Simulink

Подключение можно разделить на 2 части (этапа):

- настройка графической части,
- создание и настройка математической части.

3.1 Настройка графической части

Для настройки графической части, чтобы указать путь размещения файла с созданной моделью визуализации, нужно перетащить блок VR Sink из библиотеки vrlib в окно модели в Simulink. Блок VR Sink даёт возможность непосредственно состыковать модель динамики и модель визуализации. Настройка графической части происходит в настройках блока VR Sink. После двойного клика левой кнопкой мыши появляется окно настроек (рис. 5).

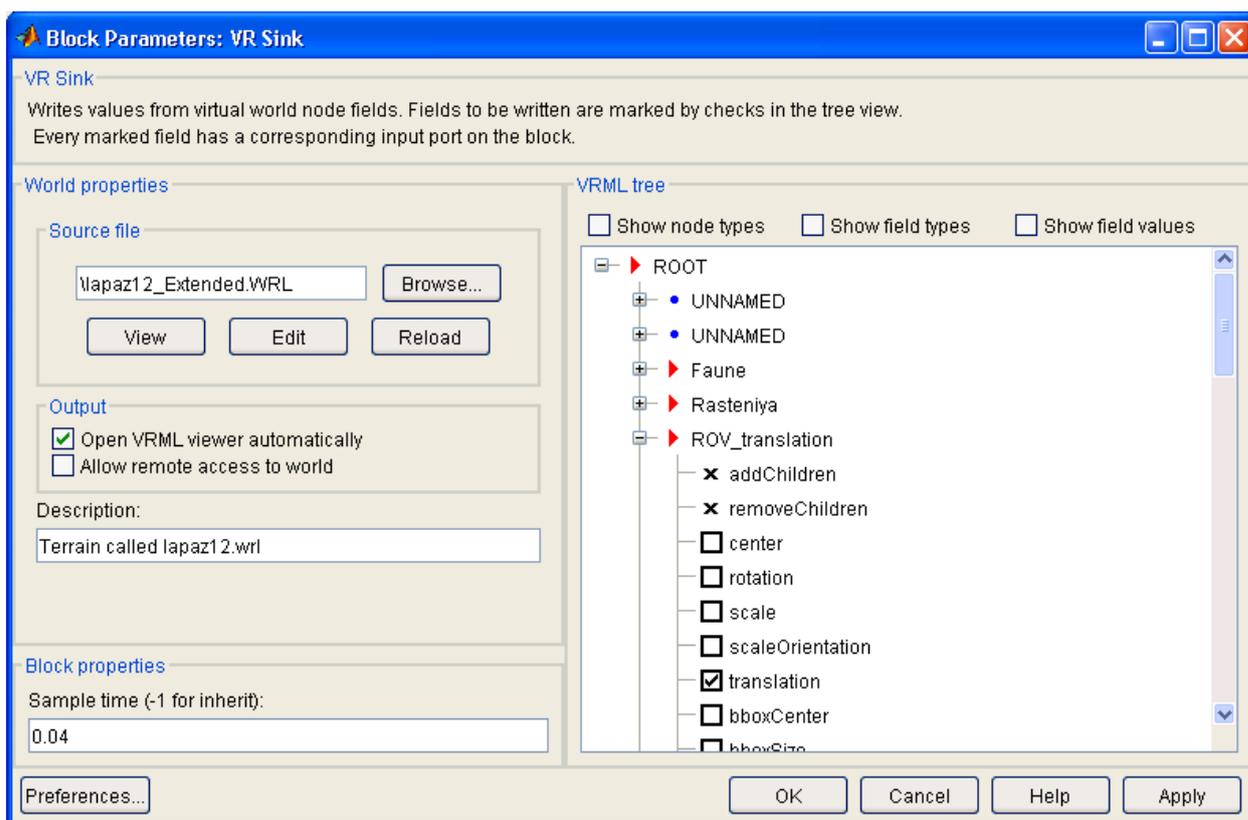


Рис. 5

В окне главными настройками являются:

- Source file – местонахождение файла с моделью визуализации;

- Block properties – Sample time - время дискретизации, $0.04 = 25$ кадров в секунду в секунду.

VRML Tree – иерархическая структура модели визуализации, которая включает все объекты. Квадратиками (рис. 5) обозначены параметры, которые можно изменять, подключая модель визуализации к модели динамики в Simulink. Галочками в квадратиках отмечены параметры, которые должны меняться. Наличие галочек в тех или иных квадратиках соответствует тому, что эти параметры будут запрашиваться из вне, т.е. у блока VR Sink появятся входы, соответствующие этим параметрам. Изменения значений линейных и угловых координат различных объектов на 3D-сцене в процессе симуляции формируют в графическом окне визуализацию перемещений объектов.

- Названия, стоящие против галочек – это входы блока VR_Sink. Иерархическая структура полностью соответствует структуре созданной модели визуализации в редакторе VRBuilder2.
- Output – галочка напротив “Open VRML viewer automaticaly” для того, чтобы окно с 3D-сценой появлялось автоматически при открытии окна модели в Simulink из файла с расширением mdl. (рис. 6)

Как упоминалось выше, редактор VRBulder2 можно запустить и из окна настроек блока VR Sink. Для этого нужно нажать кнопку Edit в окне настроек блока VR Sink (рис. 5).

Когда произведены настройки блока VR Sink, при открытии файла Simulink будет автоматически появляться окно с созданной моделью визуализации (см. рис. 6).

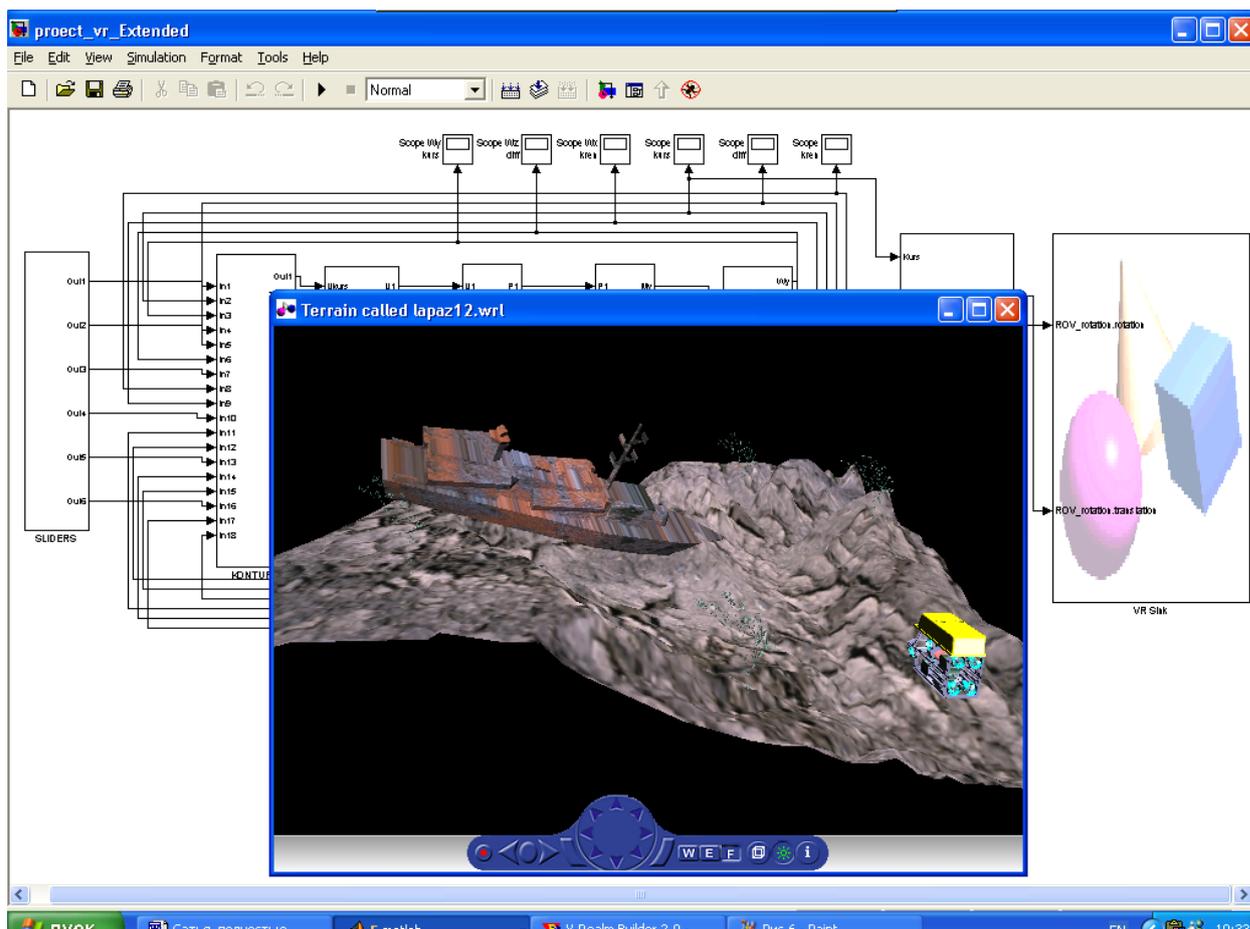


Рис. 6

Внизу графического окна расположен своеобразный дополнительный пульт управления 3D-сценой, который появляется автоматически при открытии графического окна. Это инструмент управления графикой сцены, виртуальными камерами обзора сцены, ее освещением и т.п. Например, с помощью него можно отключать/включать текстуры или материалы, переключаться с одной камеры на другую и т.д.

3.2 Создание и настройка математической части.

Итак, при наличии модели динамики и 3D-сцены встает вопрос об их взаимодействии для решения задачи визуализации. Как было сказано выше, у блока VR Sink есть входы, а значения, подаваемые на эти входы, как раз и будут соответствовать значениям тех или иных переменных, вследствие изменения которых мы будем видеть в графическом окне (рис.6) движение объектов друг относительно друга.

Для корректной работы графической части существуют блоки VR Placeholder и VR Signal Expander. Блок VR Signal Expander позволяет управлять вектором входных значений для блока VR Sink. Вектор значений для управления каждым параметром нашей модели через блок VR Sink имеет совершенно определённый вид. А вектор значений в

Simulink может содержать достаточную информацию для управления, но имеющую другой вид – например последовательность координат. VR Signal Expander является своего рода переходником для корректного взаимодействия моделей динамики и визуализации. Рассмотрим пример, если надо управлять изменением координат объекта

по 3-м осям координат, а исходный и требуемый вектора соответственно имеют вид $\begin{pmatrix} x \\ z \\ y \end{pmatrix}$ и

$\begin{pmatrix} x \\ y \\ z \end{pmatrix}$, то необходимо задать параметры в блоке VR Signal Expander, как показано на рис. 7.

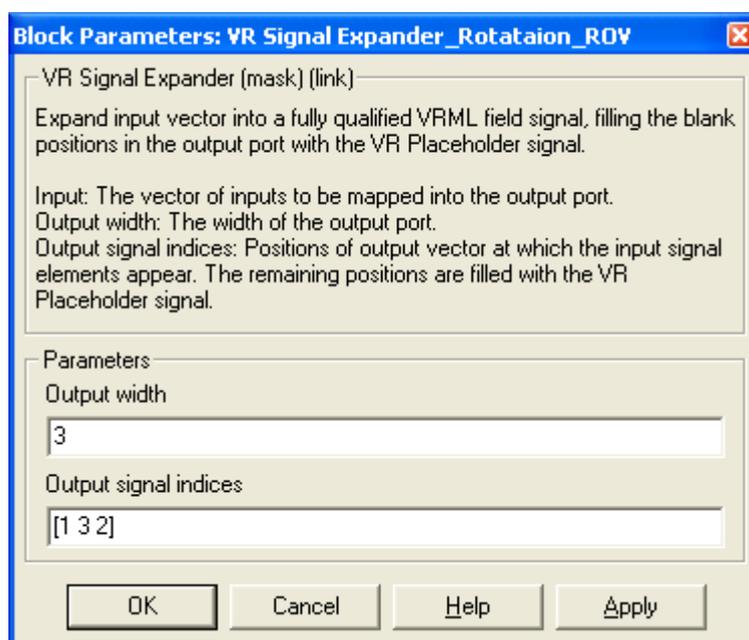


Рис. 7

Когда надо поменять всего одно значение в векторе координат или каком-либо другом, например векторе масштабных коэффициентов, при условии неизменности всех остальных (например нужно менять одну координату из 3-х), используется блок VR Placeholder.

В данном примере использовалась готовая модель динамики подводного аппарата с системой автоматического управления его движением, заранее разработанная в Simulink.

В рассматриваемом случае на выходе модели динамики в виде сигналов имеются значения элементарных приращений координат центра аппарата и координат вращения - углов в зависимости от входных воздействий. Т.е. задача сводится к тому, чтобы разложить эти элементарные приращения по координатам 3D-сцены и затем сложить их.

В итоге получим движение подводного аппарата по соответствующей траектории. Реализация математической части показана на рис. 8, 9, 10.

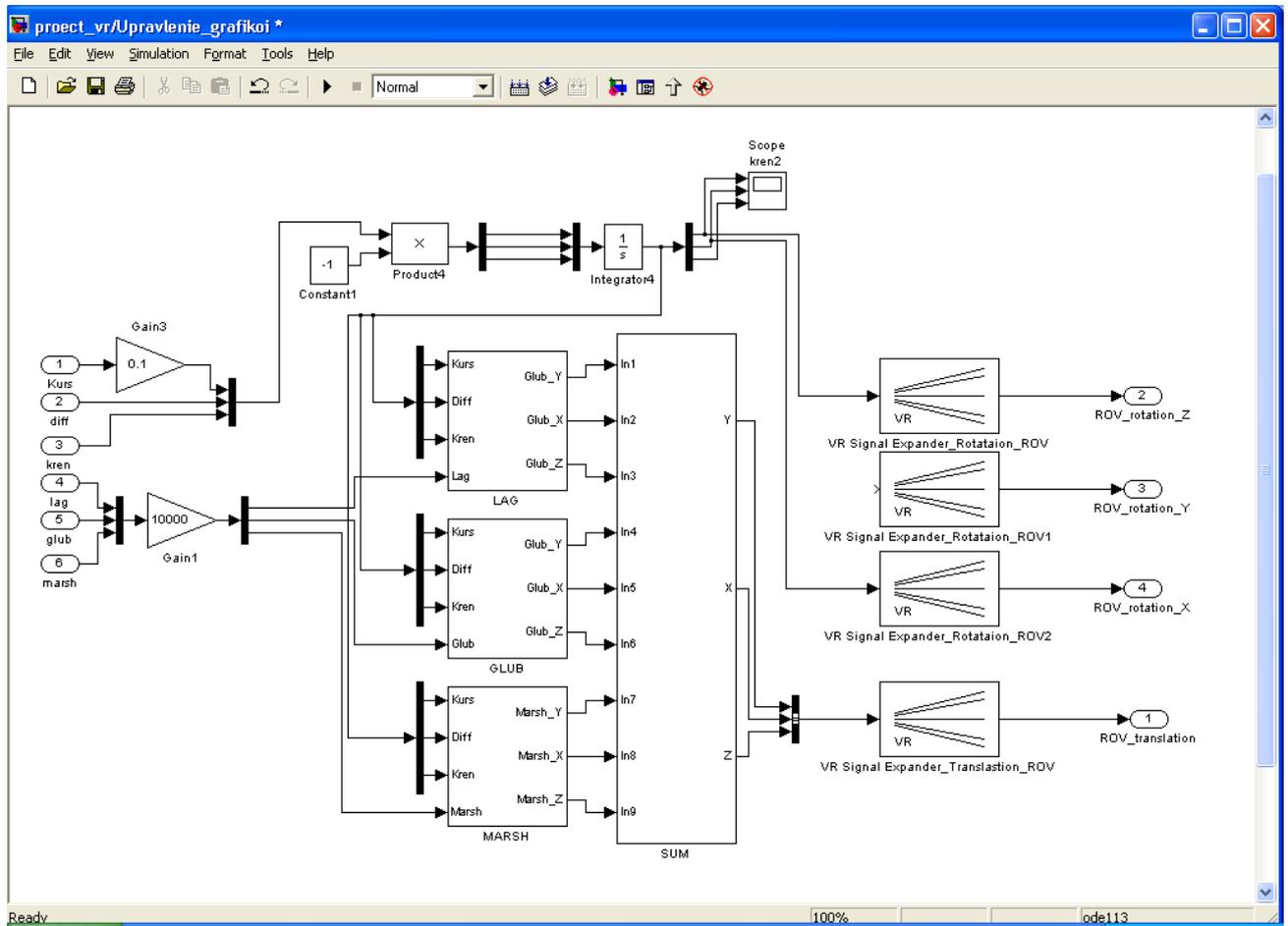


Рис. 8

На рис. 8 показана общая схема подключения модели визуализации к модели динамики. 6 входов – приращения по 6-ти координатам подводного аппарата. Далее (см. также рис. 9) раскладка этих приращений на оси 3D-сцены и интегрирование, далее (см. также рис. 10) все перемещения по одинаковым осям складываются и через блоки VR Signal Expander поступают на выход (и дальше непосредственно в блок VR Sink). Также на рис. 8 можно видеть 3 подсистемы : Lag, Glub, Marsh. В этих подсистемах происходит раскладка приращений перемещения подводного аппарата на координаты 3-D сцены. Ниже, на рис. 9 показана одна из 3-х выше перечисленных подсистем под названием Marsh.

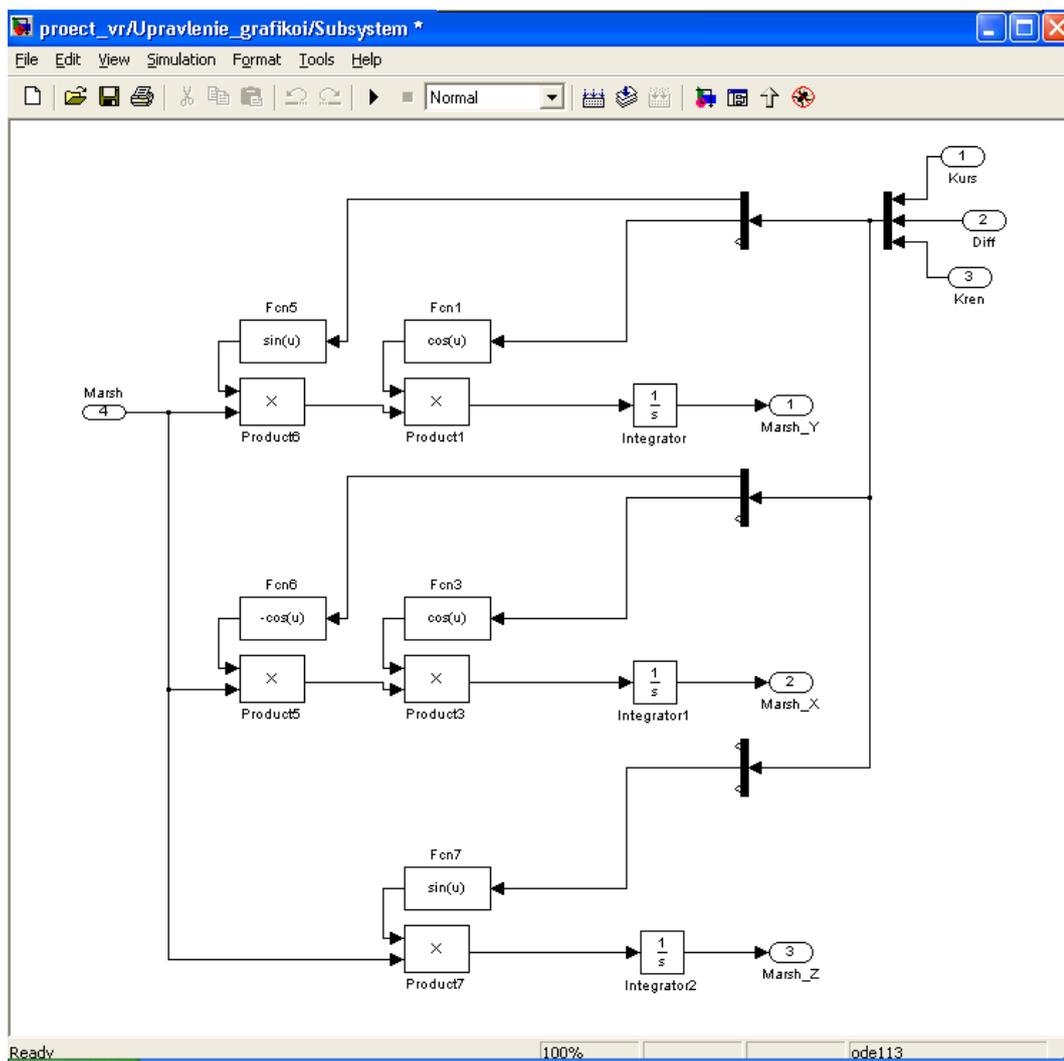


Рис. 9

На рис. 9 приведена схема раскладки маршевого движения подводного аппарата на оси X, Y и Z созданной 3D-сцены.

Далее, в подсистеме Sum (см. рис. 8) составляющие от продольного, поперечного и вертикального перемещений (Marsh, Lag, Glub) соответственно суммируются, и далее подаются на входы блоков VR Signal Expander.

4. Варианты формирования задающих сигналов для системы управления движением объекта по сцене

Задачу формирования задающих сигналов для системы управления движением объекта по сцене можно решить двумя путями: или подключить к системе реальное внешнее задающее устройство типа “джойстик” или сформировать виртуальное задающее устройство.

Для подключения джойстика следует перенести в модель в Simulink иконку Joystick Input из библиотеки vrlib (рис. 4), а перед началом работы с моделью в Simulink драйвер джойстика должен быть установлен в разделе “Игровые устройства” ОС Windows.

У блока Joystick Input 3 выхода, 2 из них являются векторами, третий – скалярной величиной (для решения выше поставленных задач не используется). Первый выход Axes – вектор сигналов с осей джойстика, второй Buttons – сигналы с кнопок джойстика.

На рис. 10 показано подсоединение джойстика к модели динамики.

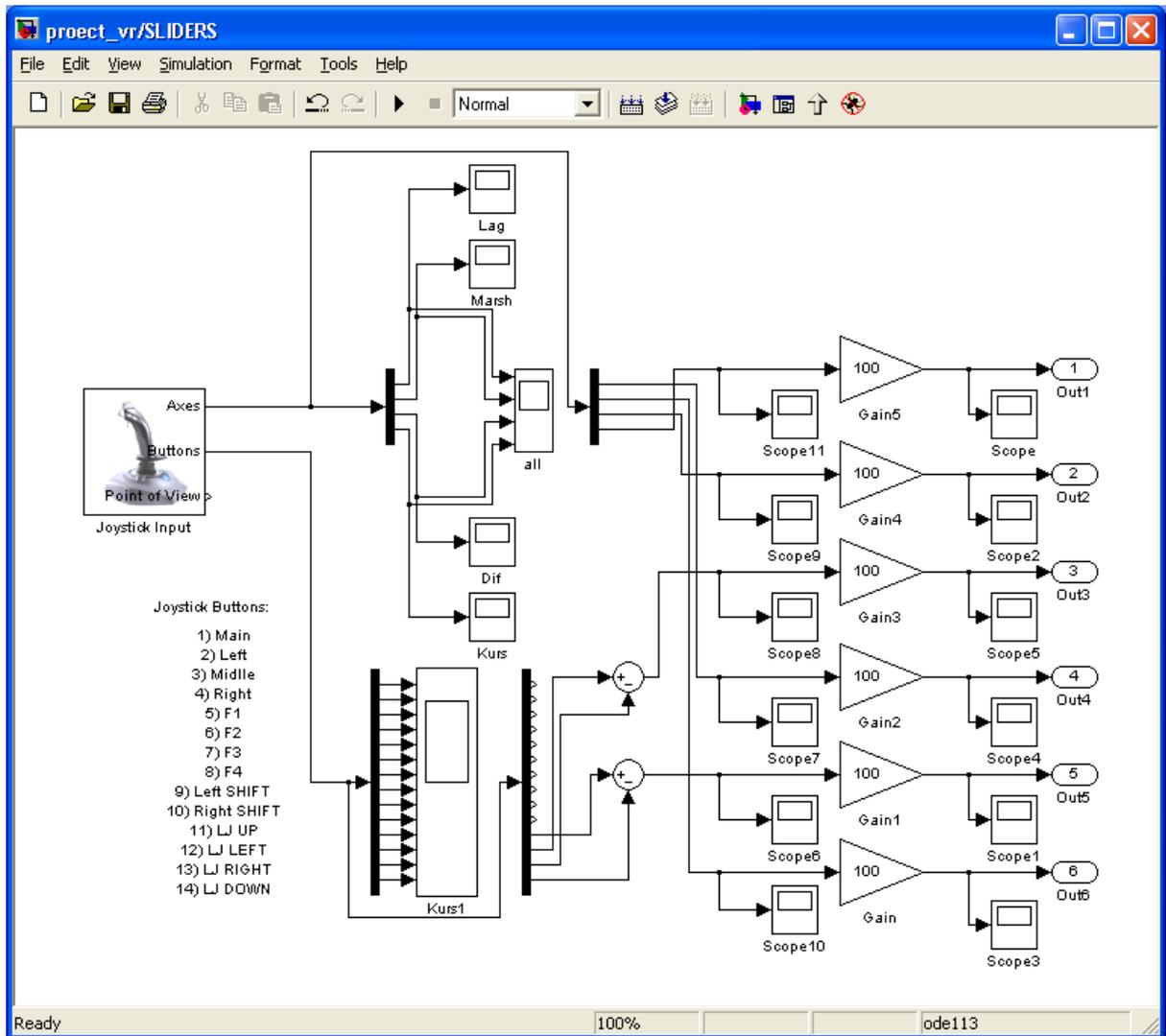


Рис. 10

На рис. 11 показано альтернативное джойстику (на случай его отсутствия) виртуальное устройство формирования задающих сигналов, использующее 6 блоков Slider Gain из раздела Math Operations базовой библиотеки Simulink.

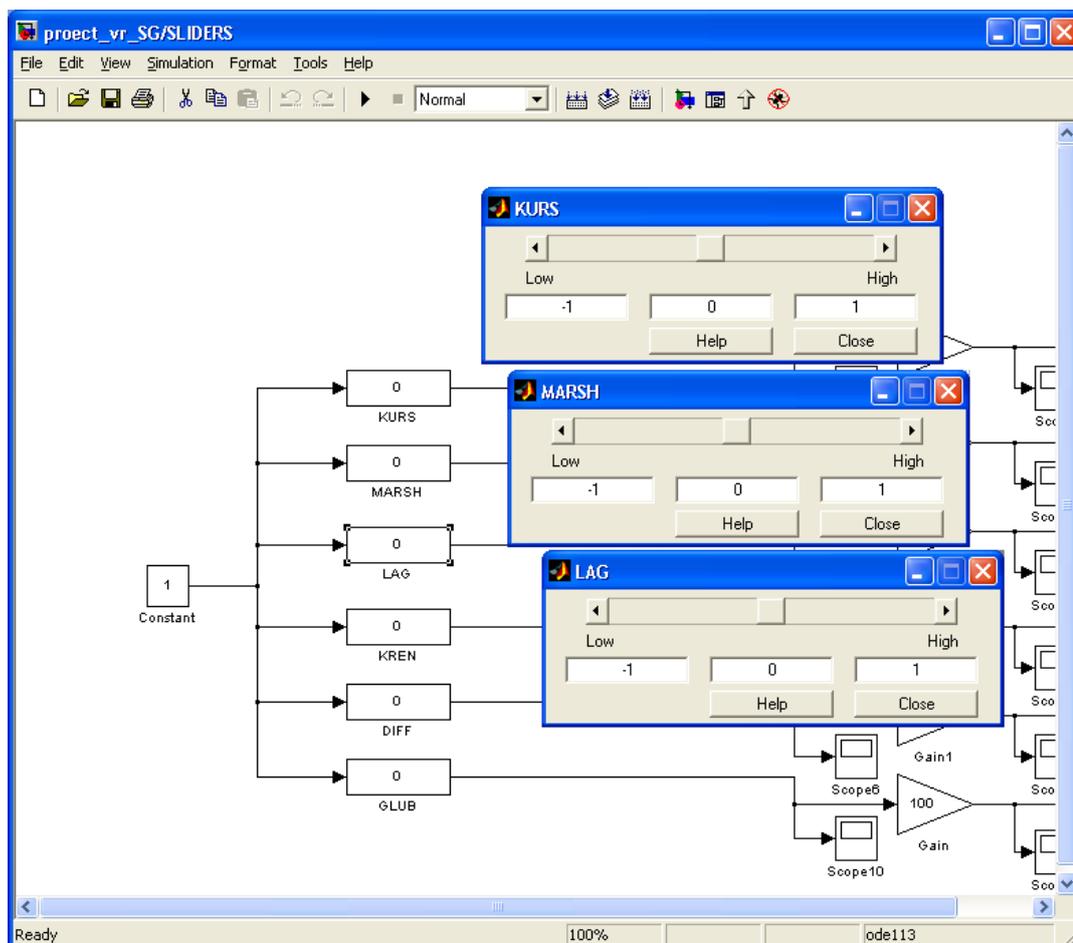


Рис. 11

Заключение

В результате подключения модели визуализации к модели динамики и управления подводного аппарата сформирована система визуализации и динамики движения подводного аппарата в виртуальном подводном мире. Запуск и управление ее работой производится в обычных режимах моделирования в Simulink.

Полученная модель динамики и визуализации удобна для быстрого прототипирования и применения в учебных, презентационно-демонстрационных и научно-исследовательских целях. Примечательно, что создание такой комплексной модели вполне по силам разработчикам-непрограммистам, практически не владеющим знаниями и навыками программирования на каких-либо языках программирования.

Недостаток данного пути разработки модели визуализации заключается в том, что создавать сложные 3-D сцены с учётом взаимодействия объектов достаточно трудоёмко в редакторе VRBuilder2. И компьютерных ресурсов для этих моделей требуется больше, чем, например, при реализации моделей в VisualC++. В этом плане с целью сокращения требуемых компьютерных ресурсов можно, например, оптимизировать работу

полученной системы, состоящей из модели визуализации в виртуальной реальности, и модели динамики, с помощью приложения RealTimeWorkshop системы MATLAB.

Литература:

1. Дьяконов В.П. Simulink 5/6/7: Самоучитель. – М.: ДМК-Пресс, 2008. – 784 с.
2. Скобелев М.М. Программа имитационной модели системы управления движением автономного необитаемого подводного аппарата с визуализацией движения. – Депонированная рукопись в ФГУ ФИПС Роспатента. - Свидетельство о государственной регистрации программы для ЭВМ № 2009615905. Зарегистрировано в Реестре программ для ЭВМ 23 октября 2009 г. Правообладатель: Государственное образовательное учреждение высшего профессионального образования «Московский государственный технический университет имени Н.Э. Баумана».