

Визуальное моделирование углового движения космических конструкций 77-30569/239583

10, октябрь 2011

А. В. Кавинов

УДК 28.17.19; 004.94

МГТУ им. Н.Э. Баумана
mathmod@bmstu.ru

1. Введение

Современные средства математического моделирования редко обходятся без подсистем, реализующих визуальное представление результатов вычислений. В настоящее время развитие компьютерной техники позволяет создавать реалистичные изображения моделируемых объектов, в том числе — объемные и стереоскопические. Разработанный на кафедре ФН-12 «Математическое моделирование» МГТУ им. Н.Э.Баумана программный комплекс КОКОН объединяет в себе как средства создания трехмерных моделей, так и средства визуализации полученных результатов моделирования в различных представлениях.

В первую очередь разработанное программное обеспечение предназначено для моделирования вращения космических аппаратов вокруг центра масс под воздействием управления, но оно может быть использовано и в других задачах трехмерного моделирования.

2. Структура и состав разработанного программного обеспечения

В основу концепции ПО был положен принцип модульности, что позволило логически отделить программные модули, моделирующие движение КА под воздействием управления, от модуля, выполняющего одни и те же функции вне зависимости от конкретного закона управления. Код, включенный в этот «общий» модуль позволяет создавать и редактировать визуальные модели конструкций трехмерных КА, задавать и вычислять их геометрические, оптические и инерционно-массовые характеристики, а также представлять в удобном для восприятия виде результаты моделирования движения. Эти результаты могут быть представлены в виде графиков, а также визуализированы при помощи средств трехмерной компьютерной графики как в обычном, так и в стереоскопическом режиме.

В общий программный модуль не входят средства собственно моделирования углового движения КА под воздействием того или иного управления. Код ПО для такого моделирования выделен в отдельные программные модули, которые могут разрабатываться отдельно

от основного общего модуля. Эти модули подключаются к основному автоматически при помощи механизма динамической компоновки библиотек операционных систем семейства Windows.

Таким образом, разработанное ПО представляет собой расширяемый программный комплекс, который может быть дополнен программными модулями, реализующими различные модели углового движения.

3. Визуальное конструирование трехмерных объектов

Одна из функций оболочки состоит в предоставлении пользователю возможности визуального конструирования геометрических моделей КА. Объект сложной геометрической формы представляется составленным из более простых объектов примитивной формы, или примитивов. Задав для каждого из них форму, размеры и расположение в пространстве относительно некоторой связанной с объектом точки, мы получим полное геометрическое описание составного объекта.

Перечислим типы примитивов, поддерживаемые оболочкой:

- 1) цилиндр;
- 2) параллелепипед;
- 3) шар;
- 4) усеченный шар;
- 5) конус и усеченный конус;
- 6) пирамида с правильным основанием;
- 7) система параметрически заданных поверхностей.

Кроме того, имеются два специальных служебных типа псевдопримитивов — точка присоединения и модуль, о применении которых будет сказано ниже.

Форма каждого примитивного объекта задается его типом и набором числовых значений параметров, определяющих его размеры. Состав параметров зависит от типа примитива. Так, для параллелепипеда такими параметрами являются его ширина, длина и высота, для шара — его радиус и т.д. Наиболее общим примитивным типом является тип «система поверхностей». Примитивы всех прочих типов могут быть заменены в конструкции элементами этого типа. Отметим, что такая замена не была бы оправдана с точки зрения удобства, простоты и интуитивности процесса редактирования; поэтому рекомендуется использовать этот тип примитивов в тех случаях, когда форма конструкции не может быть промоделирована с использованием примитивов остальных типов.

Опишем процесс задания формы примитива типа «система поверхностей». Примитивы всех остальных типов можно рассматривать как частные случаи, которые мы отдельно описывать не будем. Примитив такого типа можно представить себе как тело, ограниченное совокупностью n подмножеств поверхностей. Каждая i -я поверхность параметрически за-

дается уравнениями

$$\begin{cases} x = x_i(u, v) \\ y = y_i(u, v) \\ z = z_i(u, v) \end{cases}, \quad u \in [u_{i \min}; u_{i \max}], v \in [v_{i \min}; v_{i \max}], \quad (1)$$

где x, y, z — координаты точек поверхностей в некоторой связанной с примитивом строительной системе координат. Функции x_i, y_i, z_i , а также значения $u_{i \min}, u_{i \max}, v_{i \min}, v_{i \max}$ задаются пользователем при редактировании свойств примитива (рис. 1).

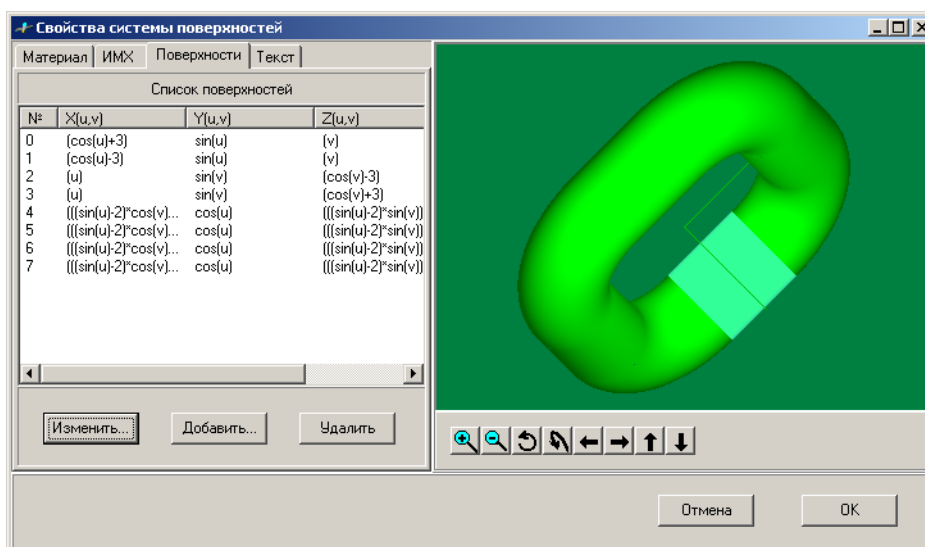


Рис. 1. Окно редактирования свойств системы поверхностей

Геометрические свойства поверхностей, из которых составлена система, могут быть отредактированы для каждой из них по отдельности (рис. 2). Чтобы обеспечить правильное визуальное представление поверхности, необходимо также задать ее светоотражательные

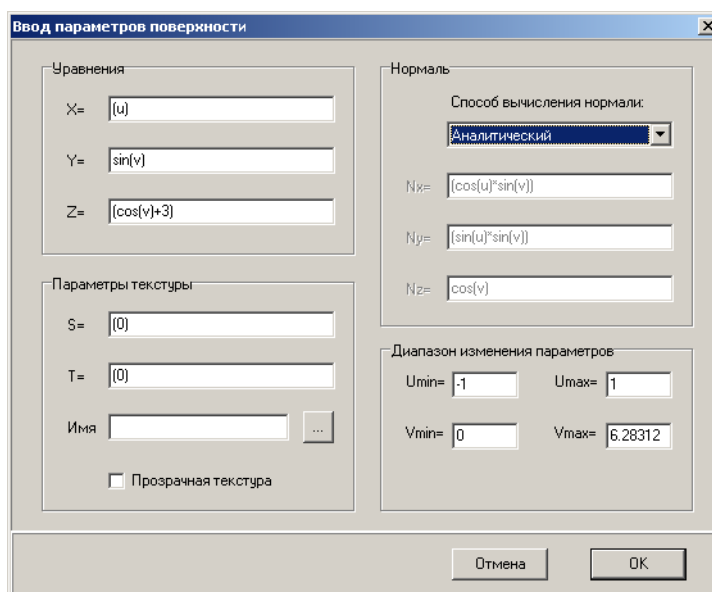


Рис. 2. Окно редактирования геометрических свойств поверхности

свойства и закон изменения вектора нормали в каждой точке в зависимости от параметров u и v . Кроме того, на каждую поверхность может быть наложена текстура. Текстурное изображение и закон соответствия координат u и v с текстурными координатами также может быть задан пользователем.

Оптические (светоотражательные) свойства задаются для всего примитива в целом (рис. 3). Пользователем задаются цвета рассеянного, диффузного и отраженного светового излучения материала, а также степень зеркальности отражения света от поверхности.

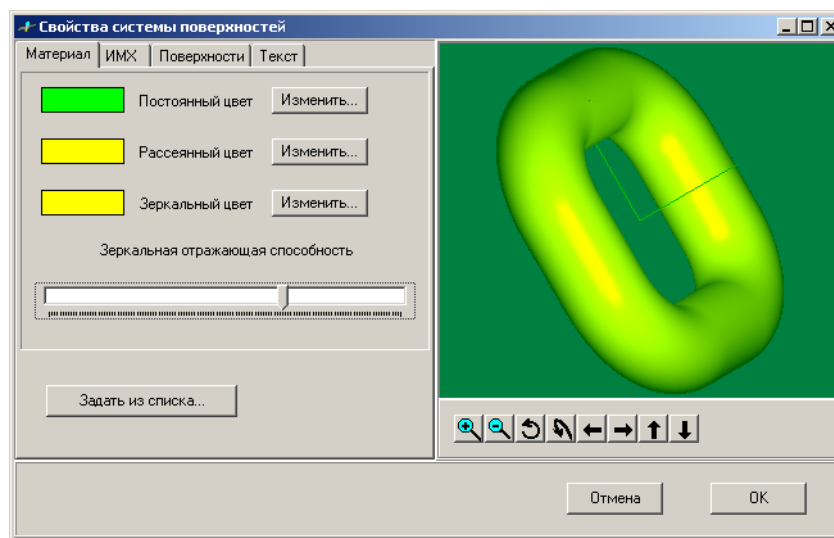


Рис. 3. Окно редактирования оптических свойств поверхности

Для более эффективного производства вычислений, связанных с отображением графических объектов и расширением возможностей оболочки по моделированию конструкций в нее интегрирован компилятор элементарных математических выражений. Он преобразует текстовые строки, содержащие математические выражения наподобие $\sin(u)+3*\cos(v+u*\exp(v-u))$ в машинный код семейства процессоров Pentium, вычисляющий эти выражения при заданных значениях параметров. Благодаря этому в процессе отображения объектов не требуется постоянного разбора выражений для вычисления координат точек поверхностей, нормалей и текстурных координат. Разбор всех выражений производится один раз при редактировании или при загрузке модели с жесткого диска. Кроме этого, компилятор включает в себя элементы символьной математики. Он позволяет аналитически выражать частные производные введенных выражений, что делает необязательным явное задание при редактировании координат нормалей к поверхностям.

Для каждого примитивного объекта могут быть заданы его инерционно-массовые характеристики (рис. 4). Чтобы вычислить матрицу тензора инерции совокупности объектов, достаточно иметь информацию о массе, расположении центра масс и тензоре инерции для каждого из этих объектов. Если такая информация имеется, то совокупный тензор инерции может быть вычислен на основании этих данных. Решение этой задачи изложено в [4].

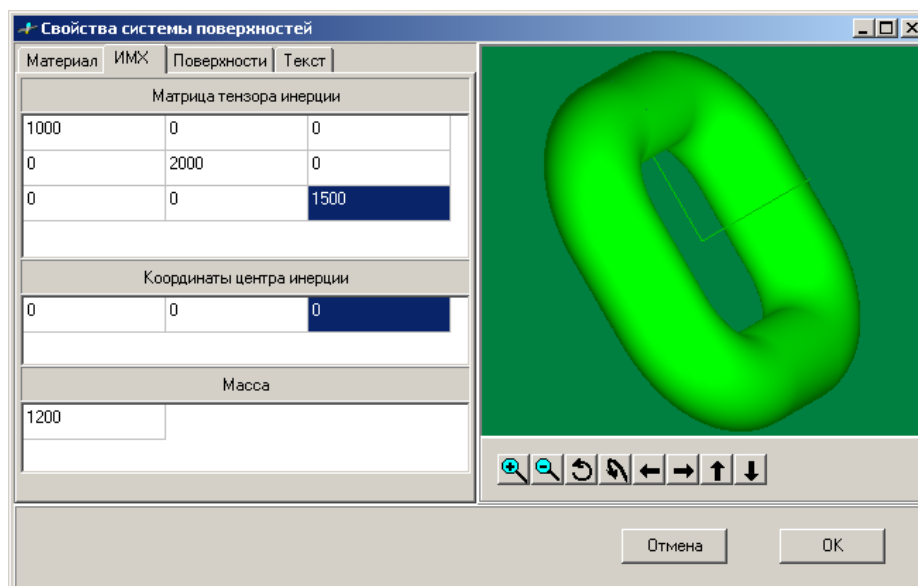


Рис. 4. Окно редактирования инерционно-массовых характеристик свойств поверхности

Перейдем теперь к описанию процесса сборки модели КА из примитивов. Этот процесс состоит в последовательном присоединении примитивов друг к другу. Первым шагом пользователя в этом процессе должно стать введение в модель базового примитива. Положение всех примитивов, составляющих объект, в пространстве друг относительно друга задается с использованием концепции дерева присоединений. Все составляющие объект примитивы организованы в ориентированное дерево, корнем которого является базовый примитив. Положение в пространстве элемента-потомка задается не в общей для всех объектов системе координат, а относительно строительной системы координат элемента-предка (исключением является базовый примитив). Тот факт, что наряду с геометрической информацией о положении примитивов хранится информация о предках и потомках каждого элемента, позволяет легко оперировать группами примитивов, соединенными в ветвь дерева присоединений. При этом само дерево присоединений может быть переконфигурировано в процессе редактирования. В частности, любой из элементов дерева может быть сделан базовым.

Чтобы присоединить примитивы друг к другу «склейкой по поверхности», пользователь должен указать по одной точке склейки на каждой поверхности и один дополнительный параметр, который задает поворот присоединяемой конструкции вокруг нормалей в точке склейки. Этот процесс имеет визуальный характер, то есть пользователь может наблюдать результат присоединения в зависимости от выбора точек и параметра в процессе редактирования. Пользователь также может задать смещение присоединяемого объекта относительно полученного положения.

Чтобы обеспечить удобство сборки конструкций, содержащих повторяющиеся элементы, а также отделить процесс задания форм различных частей КО от их соединения в одну конструкцию, в оболочке реализована концепция модуля.

Конфигурация космической станции, находящейся на орбите, не может быть изменена произвольно. Станция составлена из одного или более модулей, соединенных друг с другом через специальные стыковочные узлы. Их расположение не произвольно, так же, как и форма самих стыкуемых объектов. Разумным представляется ограничить процедуру сборки выбором некоторых из возможных модулей и соединения их одним из нескольких заранее заданных способов.

Итак, модуль — это совокупность примитивов, положение которых задано друг относительно друга. С точки зрения пользователя модуль является специальным типом примитива, который может быть включен в конструкцию наряду с другими примитивами, некоторые или все из которых, возможно, также являются модулями. Однако, процедура присоединения модуля к другому примитиву несколько отличается от описанной выше.

Чтобы модуль можно было использовать в конструкции, в нем должно присутствовать не менее одной точки присоединения. Точка присоединения — специальный примитив, не отображающийся при визуализации движения и предназначенный для конструирования. Она задает точку, к которой может осуществляться присоединение других объектов, и направление присоединения. Таким образом, если присоединение к примитиву может осуществляться относительно любой точки любой его поверхности, то присоединение к модулю может происходить только через заранее заданные точки. Отметим, что это никак не ограничивает возможности по расположению модулей в пространстве, а лишь упрощает процедуру его задания.

На рис. 5 показан возможный вид модуля. Отрезками прямых обозначены точки присоединения. При присоединении модуля пользователю предлагается выбрать одну из этих

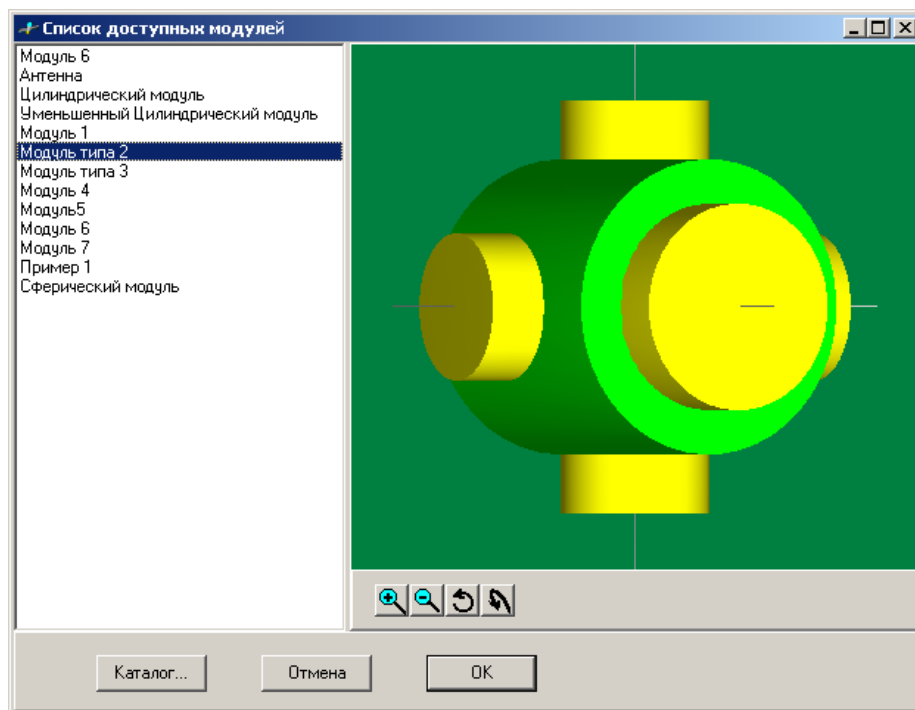


Рис. 5. Модуль и точки присоединения

точек. Окно выбора точки присоединения показано на рис. 6. Объемными серыми стрелками на проекциях обозначено направление присоединения.

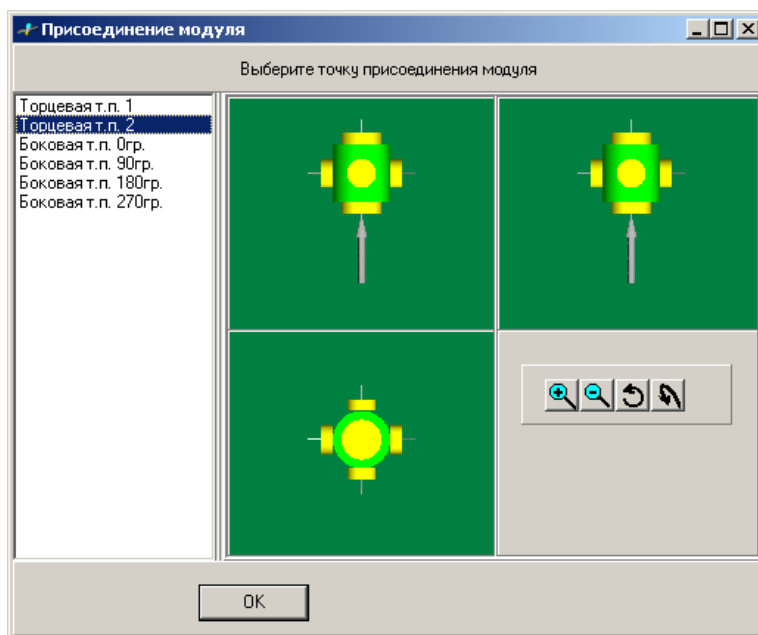


Рис. 6. Выбор точки присоединения

В результате сборки пользователь получает визуальную модель некоторого КА (рис. 7). Собранная модель может быть сохранена в файл для дальнейшей работы.

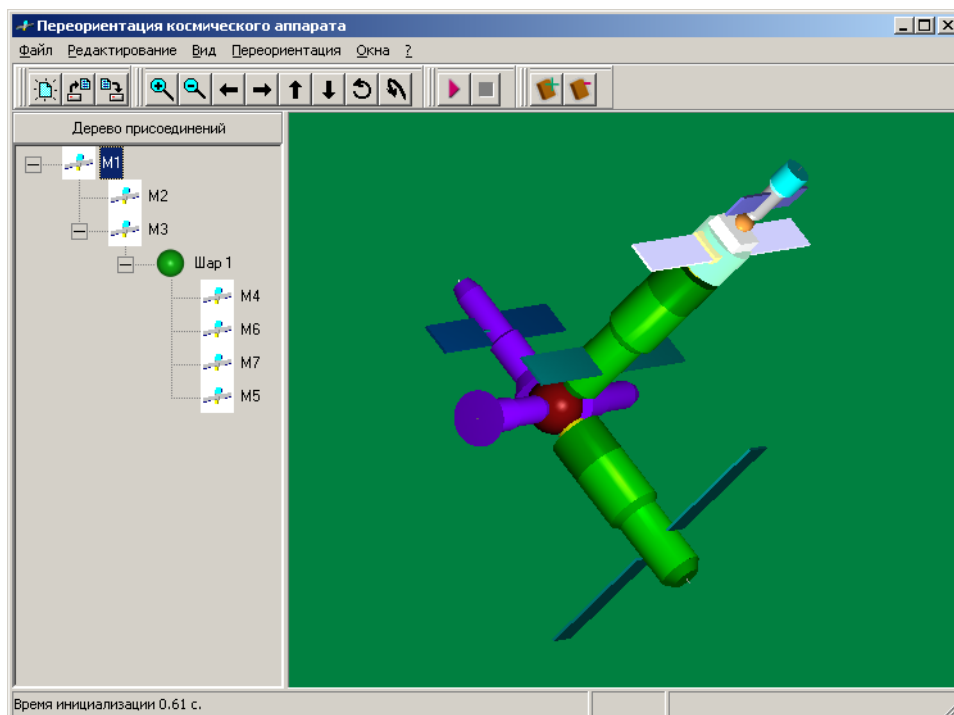


Рис. 7. Визуальная модель КА

4. Моделирование процесса управления угловым движением космического аппарата

В результате сборки пользователь также получает модель КА с заданными или вычисленными инерционно-массовыми характеристиками. Как уже было сказано, оболочка позволяет визуализировать угловое движение модели под воздействием различных законов управления. Угловое движение можно представить как последовательность кватернионов, задающих положение тела в определенные моменты времени. Таким образом, программный код, моделирующий движение, должен выдавать оболочке массив значений компонентов кватернионов вращения. Получать же от оболочки он должен данные о начальном состоянии объекта, о цели управления, а также некоторые дополнительные параметры моделирования и управления, которые различаются в зависимости от алгоритма управления. Пользователь перед запуском того или иного программного модуля моделирования движения задает начальное и желаемое конечное состояние КА, а также параметры конкретного алгоритма управления.

Код моделирования движения не входит в состав оболочки, а подключается к ней динамически. Для этого DLL-модуль, экспортирующий соответствующий протоколу взаимодействия функции для вычислений и обмена данными, помещается в специальный каталог файловой системы. Оболочка при загрузке анализирует содержимое этого каталога, вследствие чего пользователю предоставляется возможность выбора одного из алгоритмов управления для моделирования (рис. 8).

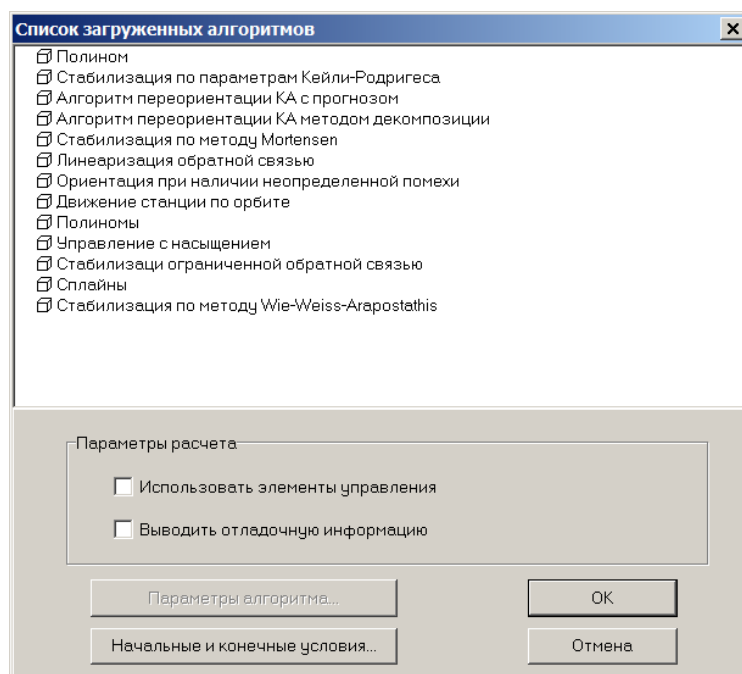


Рис. 8. Окно выбора алгоритма управления

Сотрудниками и студентами кафедры был разработан ряд модулей, вычисляющих управление по известным алгоритмам [5, 6, 7, 8] и др.

Результаты моделирования углового движения могут быть представлены программным комплексом как в виде графиков, так и в виде трехмерной анимации, в том числе объемной (стереоскопической).

5. Заключение

Таким образом, разработанный сотрудниками и студентами кафедры ФН-12 (в том числе и автором этих строк) программный комплекс позволяет производить математическое моделирование процесса управления угловым движением космических аппаратов, а также сборку их трехмерных моделей и стереоскопическую визуализацию процесса их вращения.

Описанное программное обеспечение используется кафедрой при подготовке презентаций, докладов, а также в курсовом проектировании студентов.

Работа выполнена при поддержке грантов РФФИ №10-07-00617 и Программы Президента РФ по государственной поддержке ведущих научных школ (грант № НШ-4144.2010.1).

Список литературы

1. Механика больших космических конструкций / Н.В. Баничук, И.И. Карпов, Д.М. Климов и др. М.: Факториал, 1997. 302 с.
2. Attitude Control Design for Spacestation with Variable Structure / A.P. Krishchenko, S.B. Tkachev, A.V. Kavinov, M.A. Velishchanskii // 13 Int. Conf. on Process Control'01 June 11-14, Strbske Pleso. Summaries Volume, SUT. Bratislava, 2001. P. 36.
3. Разработка программного обеспечения для проектирования облика сборных космических станций и разработка алгоритма управления ориентацией: Отчет о НИР (заключ.) / МГТУ. НУК ФН; Руководитель А.П. Крищенко. ФН 12/01; №ГР 01990008630; Инв. №02200103252. М., 2000. 123 с.
4. Лурье А. И. Аналитическая механика. М.: Гос. изд-во физ.-мат. лит., 1961. 824 с.
5. Mortensen R. A globally stable linear attitude regulator // International journal of control. 1968. V. 8. P. 297–302.
6. Tsiotras P. New control laws for the attitude stabilization of rigid bodies // 13th IFAC Symposium on automatic control in aerospace. Palo Alto (California), 1994. P. 316–321.
7. Wie B., Weiss H., Arapostathis A. Quaternion feedback regulator for spacecraft eigenaxis rotations // Journal of guidance, control and dynamics. 1989. V. 12., № 3. P. 375–380.
8. Ермошина О. В. , Крищенко А. П. Синтез программных управлений ориентацией космического аппарата методом обратных задач динамики // Изв. АН. Теория и системы управления. 2000. № 2. С. 155–162.

Visual modeling of spaceships angular motion**77-30569/239583**

10, October 2011

A. V. Kavinov

Bauman Moscow State Technical University

mathmod@bmstu.ru

The paper describes KOKON software package developed at the Mathematical modeling Department (FN-12) of Bauman Moscow State Technical University. The software allows to construct the visual three-dimensional models of spacecrafts and spacestations. Realistic 3D Stereo visualization is also included. The process of spacecraft motion and, in particular, the angular rotation of the spacecraft under the influence of the attitude control can be simulated and visualized. The developed software also contains tools for visual design of three-dimensional models.

References

1. Mehanika bol'shikh kosmicheskikh konstrukcij / N.V. Banichuk, I.I. Karpov, D.M. Klimov i dr. M.: Faktorial, 1997. 302 s.
2. Attitude Control Design for Spacestation with Variable Structure / A.P. Krishchenko, S.B. Tkachev, A.V. Kavinov, M.A. Velishchanskii // 13 Int. Conf. on Process Control'01 June 11-14, Strbske Pleso. Summaries Volume, SUT. – Bratislava, 2001. – P. 36.
3. Razrabotka programmnogo obespecheniya dlya proektirovaniya oblika sbornykh kosmicheskikh stancij i razrabotka algoritma upravleniya orientaciej: Otchet o NIR (zakljuch.) / MGTU. NUK FN; Rukovoditel' A.P. Krishchenko. - FN 12/01; N GR 01990008630; Inv. N 02200103252. M., 2000. - 123 s.
4. Lur'e A. I. Analiticheskaya mehanika. M.: Gos. izd-vo fiz.-mat. lit., 1961. 824 s.
5. Mortensen R. A globally stable linear attitude regulator // International journal of control. 1968. – V. 8. – P. 297–302.
6. Tsiotras P. New control laws for the attitude stabilization of rigid bodies // 13th IFAC Symposium on automatic control in aerospace. – Palo Alto (California), 1994. – P. 316–321.
7. Wie B., Weiss H., Arapostathis A. Quaternion feedback regulator for spacecraft eigenaxis rotations // Journal of guidance, control and dynamics. – 1989. – V. 12., № 3. – P. 375–380.
8. Ermoshina O. V., Krishchenko A. P. Sintez programmyh upravlenij orientaciej kosmicheskogo apparata metodom obratnyh zadach dinamiki // Izv. AN Teoriya i sistemy upravleniya. 2000. N 2. S. 155-162.