

ОСОБЕННОСТИ МОДЕЛИРОВАНИЯ ДИНАМИКИ ЭКСПЕРИМЕНТАЛЬНОЙ ТРОСОВОЙ СИСТЕМЫ СОВРЕМЕННЫМИ КОМПЬЮТЕРНЫМИ ПРОГРАММАМИ

На примере модели экспериментальной тросовой системы проведен сравнительный анализ использования пакетов MATLAB, Maple и MapleSim для исследования динамики системы и выработаны рекомендации по рациональному использованию этих программных средств.

На прикладі моделі експериментальної тросової системи проведено порівняльний аналіз використання пакетів MATLAB, Maple і MapleSim для дослідження динаміки системи і вироблено рекомендації з раціонального використання цих програмних засобів.

The comparative analysis of MATLAB, Maple, and MapleSim packages for studying the system dynamics is carried out using the model of an experimental tethered system, and recommendations for rational using this software are made.

Одним из перспективных направлений современной космической техники является создание космических тросовых систем (КТС) – систем, состоящих из спутников, соединенных протяженными гибкими нитями [1 – 3]. Математические модели динамики КТС, как правило, достаточно громоздки, поэтому верификация компьютерных программ, реализующих эти модели, представляет собой серьезную проблему [4]. Надежность результатов расчетов может быть обеспечена благодаря использованию различных программных средств и различных подходов к моделированию динамики подобных систем. В связи с этим возникает задача выбора программных средств, позволяющих создавать математические модели и исследовать динамику КТС с наименьшими затратами усилий на рутинную работу.

Рассмотрим систему, состоящую из твердого тела, подвешенного на невесомой упругой нити (рис. 1). Она представляет собой частный случай экспериментальной тросовой системы [5] и может служить для проверки программ расчета динамики последней.

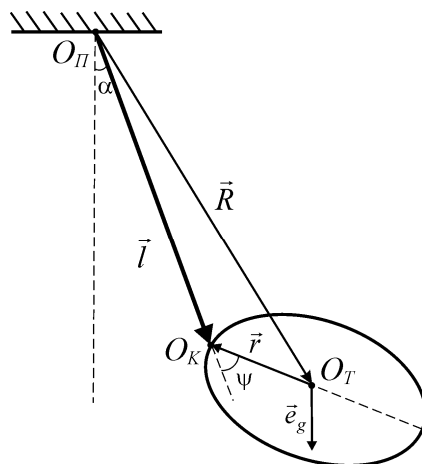


Рис. 1

В качестве программных средств моделирования динамики рассматриваются: MATLAB¹ (разработчик: The MathWorks, США), Maple (Waterloo Maple Inc., Канада) и MapleSim² (Waterloo Maple Inc., Канада)³. Целью настоящей статьи является сравнительный анализ особенностей моделирования рассматриваемой системы каждым из программных средств и выработка рекомендаций по рациональному использованию этих средств.

Последовательность этапов моделирования динамики рассматриваемой системы представлена на рис. 2.

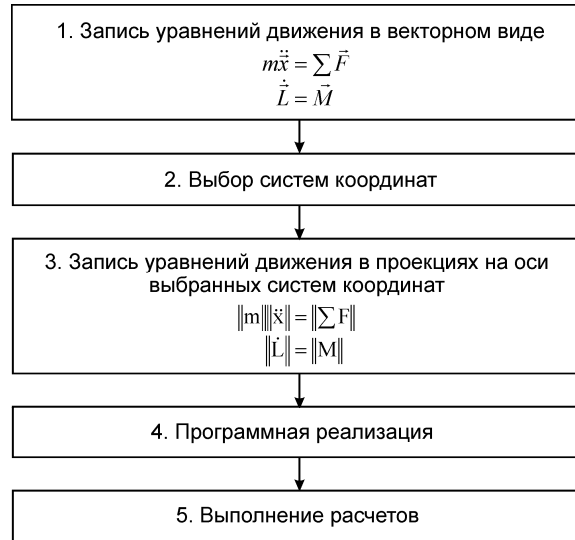


Рис. 2

Запишем уравнения движения системы. Предполагая, что массой нити в сравнении с массой тела можно пренебречь, получим

$$\begin{aligned}
 m\ddot{\vec{R}} &= \vec{F}_{mp} + mg\vec{e}_g, \\
 \dot{\vec{L}} &= \vec{r} \times \vec{F}_{mp},
 \end{aligned}
 \tag{1}$$

где m – масса тела; \vec{R} – радиус-вектор центра масс тела (точка O_T) относительно точки подвеса нити O_{II} ; \vec{F}_{mp} – сила натяжения нити; \vec{e}_g – единичный вектор по направлению действия ускорения земного тяготения \vec{g} ; $\vec{r} = \overrightarrow{O_T O_K}$ (рис. 1); \vec{L} – кинетический момент движения тела относительно центра масс.

Предположим далее, что упругие свойства нити описываются законом Гука, а рассеивание энергии движения системы происходит за счет вязкого трения в материале нити

¹ Ограниченные рамки статьи не позволили рассмотреть SimMechanics – библиотеку Simulink для физического моделирования механических систем. Подробнее о SimMechanics можно узнать в [6].

² MapleSim является дополнением к Maple и использует возможности последнего для выполнения расчетов, однако предлагает другой способ построения математических моделей и потому заслуживает отдельного рассмотрения.

³ В работе над статьей использовались trial-версии указанных программ.

$$\vec{F}_{mp} = - \left[c \frac{\vec{l} (l-d)}{l} + \chi \dot{l} \frac{\vec{l}}{l} \right] \delta, \quad \delta = \begin{cases} 1 & l > d, \\ 0 & l \leq d, \end{cases} \quad (2)$$

где \vec{l} – вектор, направленный из точки O_{II} в точку крепления нити к телу O_K : $\vec{l} = \overrightarrow{O_{II}O_K}$, $l = |\vec{l}|$, $\dot{l} = (\vec{l}, \dot{\vec{l}}) / |\vec{l}|$; d – номинальная длина нити; c – коэффициент жесткости; χ – коэффициент демпфирования.

Необходимые для описания движения системы координат (СК), а также кинематические уравнения, замыкающие систему (1), (2), приведены в [5].

Рассмотрим теперь особенности моделирования динамики системы каждым из программных средств. Полный код рассматриваемых ниже программ доступен по адресу [7].

MATLAB. Перед тем как приступить к использованию MATLAB, необходимо спроецировать уравнения (1), (2), дополненные кинематическими уравнениями [5], на оси выбранных СК, записать их в удобном для численного интегрирования виде (форме Коши), а также выполнить ряд вспомогательных вычислений (в частности, найти равновесную длину нити l_0). Как правило, эти действия выполняются вручную или с использованием сторонних программ.

В результате получим уравнения движения, записанные в матричном виде

$$\|\dot{\mathbf{x}}\| = \|\mathbf{F}(\mathbf{x}, t)\|, \quad (3)$$

которые затем реализуются в программном коде (\mathbf{x} – обобщенные координаты системы).

Блок-схема MATLAB-программы (рис. 3) типична для всех языков программирования высокого уровня, поскольку ее основой является численное интегрирование уравнений движения.

Преимущества MATLAB при выполнении расчетов динамики систем твердых тел заключаются в удобстве операций с матрицами, благодаря чему программный код становится значительно компактнее; богатых графических возможностях и большом количестве встроенных функций. Это позволяет значительно быстрее, в сравнении с другими языками, приступить непосредственно к моделированию динамики.



Рис. 3

Наибольшие трудности при использовании MATLAB представляет собой «ручной» вывод уравнений движения в матричном виде. Поскольку матрицы перехода между СК являются ортогональными (т. е. матрица, обратная данной, получается транспонированием), то достаточно просто, перепутав знаки в элементах матрицы, перепутать и направление преобразования координат. Вероятность таких ошибок увеличивается с ростом числа элементов системы.

При внесении в систему изменений необходимо вновь вручную выводить уравнения движения. При этом неизбежно возникают ошибки, как в ходе вывода уравнений, так и при переводе полученных формул в программный код.

Решением этих проблем в рамках MATLAB может служить разработка специального пакета расширения (toolbox), позволяющего создавать системы на основе типовых элементов. Такой подход реализован в пакете Neweul-M², разработанном в Институте инженерной и вычислительной механики при университете г. Штутгарт [8]. Другим возможным решением является использование Simulink и его расширения SimMechanics [6].

MATLAB обладает весьма ограниченными возможностями по документированию работы. Запись уравнений в векторном виде, чертеж системы и введенных СК, ход вывода уравнений в матричном виде, наконец, графики полученных результатов – все это необходимо создавать и хранить с помощью сторонних программ (например, MS Word), что не всегда удобно.

Maple. В отличие от MATLAB, ориентированного, в первую очередь, на выполнение численных расчетов, основное назначение Maple – выполнение символьных преобразований, т. е. преобразований формул.

Maple можно использовать, начиная с самых ранних этапов исследования системы (рис. 2) – записи уравнений движения (рис. 4) и выражений основных сил и моментов в векторном виде. Причем записанные выражения можно использовать в дальнейших преобразованиях.

$$\begin{aligned}
 &> \mathbf{m} * \text{Diff}(\mathbf{R}_-, \mathbf{t}, \mathbf{t}) = \mathbf{F}_-[\mathbf{r}_p] + \mathbf{m} * \mathbf{g} * \mathbf{e}_-[\mathbf{g}]; \\
 &\quad \text{Diff}(\mathbf{L}_-, \mathbf{t}) = \mathbf{r}_- \ \& \times \ \mathbf{F}_-[\mathbf{r}_p]; \\
 & \\
 & \quad m \frac{\partial^2}{\partial t^2} \vec{R} = \vec{F}_{mp} + m \mathbf{g} \vec{e}_g \\
 & \quad \frac{\partial}{\partial t} \vec{L} = \vec{r} \times \vec{F}_{mp}
 \end{aligned}$$

Рис. 4

Все необходимые матрицы перехода вычисляются автоматически перемножением матриц поворотов осей СК (рис. 5).

```

> A[x] := <1, 0, 0;
           0, cos(phi(t)), -sin(phi(t));
           0, sin(phi(t)), cos(phi(t))>;
> A[yprime] := <cos(theta(t)), 0, sin(theta(t));
                0, 1, 0;
                -sin(theta(t)), 0, cos(theta(t))>;
> A[zprime] := <cos(psi(t)), -sin(psi(t)), 0;
                sin(psi(t)), cos(psi(t)), 0;
                0, 0, 1>;
> A[nw] := A[x] . A[yprime] . A[zprime];

```

$$A_{nw} := \begin{pmatrix} \cos(\theta(t)) \cos(\psi(t)) & -\cos(\theta(t)) \sin(\psi(t)) & \sin(\theta(t)) \\ \sin(\phi(t)) \sin(\theta(t)) \cos(\psi(t)) + \cos(\phi(t)) \sin(\psi(t)) & -\sin(\phi(t)) \sin(\theta(t)) \sin(\psi(t)) + \cos(\phi(t)) \cos(\psi(t)) & -\sin(\phi(t)) \cos(\theta(t)) \\ -\cos(\phi(t)) \sin(\theta(t)) \cos(\psi(t)) + \sin(\phi(t)) \sin(\psi(t)) & \cos(\phi(t)) \sin(\theta(t)) \sin(\psi(t)) + \sin(\phi(t)) \cos(\psi(t)) & \cos(\phi(t)) \cos(\theta(t)) \end{pmatrix}$$

Рис. 5

Кроме того, в документ Maple можно импортировать графические файлы, например чертеж системы (рис. 1). Таким образом, все этапы, предваряющие исследование системы с помощью MATLAB, можно осуществлять средствами Maple, сохраняя результаты работы в едином документе.

Подставляя в исходные уравнения (1) выражения действующих сил и моментов и выполняя ряд вспомогательных преобразований, в частности вычисление производных во вращающихся СК (рис. 6),

```
> Dif := proc(x, w)
    map(x -> diff(x, t), x) + w &x x;
end proc;
```

Рис. 6

получим уравнения движения в проекциях оси заданных СК.

Полученные уравнения необходимо упростить, приводя подобные члены и удаляя тривиальные соотношения вида $0 = 0$. Последние появляются, например, когда рассматриваются частные случаи движения системы (например, случай движения системы в плоскости наблюдения – рис. 7).

```
> eq[1] := subsop(1 = Dif(Dif(1[n], w[n]), w[n]) - A[nr].Dif(Dif(r[r], w[r]), w[r]), eq1);
eq[2] := subsop(1 = Dif(J[t].w[r], w[r]), eq2);
```

$$eq_1 := \begin{bmatrix} \ddot{l}_n(t) - w_n(t)^2 l_n(t) + \cos(\psi(t)) w_l(t)^2 r_1 + \sin(\psi(t)) \dot{w}_l(t) r_1 \\ \dot{w}_n(t) l_n(t) + 2 w_n(t) \dot{l}_n(t) + \sin(\psi(t)) w_l(t)^2 r_1 - \cos(\psi(t)) \dot{w}_l(t) r_1 \\ 0 \end{bmatrix} = \begin{bmatrix} -\frac{c(l_n(t) - d)}{d} - \chi l_1(t) + \cos(\alpha(t)) m g \\ -\sin(\alpha(t)) m g \\ 0 \end{bmatrix}$$

$$eq_2 := \begin{bmatrix} 0 \\ 0 \\ J_3 \dot{w}_l(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -r_1 \sin(\psi(t)) \left(-\frac{c(l_n(t) - d)}{d} - \chi l_1(t) \right) \end{bmatrix}$$

Рис. 7

Затем уравнения движения приводятся к форме Коши (рис. 8).

$$\begin{bmatrix} \dot{l}_n(t) \\ \dot{\alpha}(t) \\ \dot{l}_1(t) \\ \dot{w}_n(t) \\ \dot{\psi}(t) \\ \dot{w}_l(t) \end{bmatrix} = \begin{bmatrix} l_1(t) \\ w_n(t) \\ w_n(t)^2 l_n(t) - \cos(\psi(t)) w_l(t)^2 r_1 - \frac{c(l_n(t) - d)}{d} - \chi l_1(t) + \cos(\alpha(t)) m g + \frac{\sin(\psi(t))^2 r_1^2 \left(-\frac{c(l_n(t) - d)}{d} - \chi l_1(t) \right)}{J_3} \\ \frac{-\sin(\psi(t)) w_l(t)^2 r_1 - \sin(\alpha(t)) m g}{l_n(t)} - \frac{\cos(\psi(t)) r_1^2 \sin(\psi(t)) \left(-\frac{c(l_n(t) - d)}{d} - \chi l_1(t) \right)}{J_3 l_n(t)} - \frac{2 w_n(t) l_1(t)}{l_n(t)} \\ \frac{w_l(t) - w_n(t)}{J_3} \\ \frac{r_1 \sin(\psi(t)) \left(-\frac{c(l_n(t) - d)}{d} - \chi l_1(t) \right)}{J_3} \end{bmatrix}$$

Рис. 8

Далее можно конвертировать полученные уравнения в программный код на языке программирования высокого уровня (в частности, MATLAB) и дальнейшие вычисления выполнять вне Maple. Однако возможности для проведения численных расчетов, которыми располагают современные версии Maple, позволяют выполнять с его помощью и этот этап моделирования.

Работа с Maple и другими системами аналитических вычислений (CAV) имеет ряд особенностей по сравнению с использованием систем, ориентированных на выполнение численных расчетов.

Как мы уже видели, исследование задачи с помощью Maple может начинаться «раньше», чем становится возможным программирование в традиционном смысле. Вывод уравнений движения и их приведение к форме

Коши осуществляются не на бумаге, а с помощью САВ. При этом центральную роль в создаваемой программе играет не численное интегрирование уравнений, а процедуры их преобразования к необходимому для исследований виду.

При традиционном программировании, последовательность выполнения команд, реализующих математические выражения, диктуется не физическим смыслом или значимостью этих выражений для рассматриваемой модели, а логикой вычислений: чтобы получить численное значение выражения, необходимо вычислить значения всех входящих в него величин. В САВ, помимо этого подхода, может быть реализован и другой, близкий к традиционной работе исследователя «с карандашом и бумагой». Вначале записываются (на языке САВ) наиболее общие уравнения, описывающие поведение системы. Затем входящие в них величины расписываются подробно, делаются оценки и упрощения, результаты которых подставляются в исходные уравнения.

Кроме того, Maple обладает развитыми возможностями по оформлению результатов расчетов и подготовке научных публикаций, в частности возможностью экспорта подготовленных документов в форматы PDF, TeX, RTF и HTML.

По сути, Maple является не только средством создания расчетных программ, но и инструментом, который можно использовать на всех этапах моделирования, сведя «бумажную» работу к минимуму.

MapleSim. До сих пор уравнения движения рассматриваемой системы в векторном виде необходимо было получать вручную. В MapleSim эти уравнения формируются автоматически, на основе механической схемы системы.

Моделирование физических процессов в MapleSim заключается в создании блок-схемы рассматриваемой системы. Библиотека готовых компонентов (блоков) MapleSim позволяет исследовать механические, электротехнические, гидравлические и тепловые системы, а также системы управления. Связи между компонентами имитирует физические соединения в реальной системе.

В нашем случае используются компоненты из раздела Multibody (Системы твердых тел). Они делятся на пять видов:

- Bodies and Frames (Тела и системы отсчета) – для моделирования систем отсчета, абсолютно твердых тел и гибких балок;
- Forces and Moments (Силы и моменты) – имитируют кинематические пары, образованные телами рассматриваемой системы;
- Joints and Motions (Связи и виды перемещений) – для имитации приложенных к системе внешних сил и моментов;
- Sensors (Датчики) – для имитации измерительных устройств параметров движения;
- Visualization (Визуализация) – имитируют внешний вид системы.

Блок-схема рассматриваемой системы состоит из следующих компонентов (рис. 9).

Fixed Frame – задает расположение неподвижной СК. Каждая механическая модель MapleSim должна содержать хотя бы один подобный компонент.

Universal – представляет собой шарнир с двумя вращательными степенями свободы вокруг взаимно перпендикулярных осей. Параметры одного из подобных компонентов (Universal 2) представлены на рис. 10. Помимо направлений осей вращения (\mathcal{E}_1 и \mathcal{E}_2) можно задавать также начальные усло-

вия: угол отклонения $\bar{\theta}_0$ и угловую скорость $\bar{\omega}_0$. Эти условия могут применяться к компоненту обязательно (когда в выпадающем меню $IC_{\theta,\omega}$ выбрано Strictly Enforce); применяться, если это возможно (Treat As Guess) или игнорироваться (Ignore).

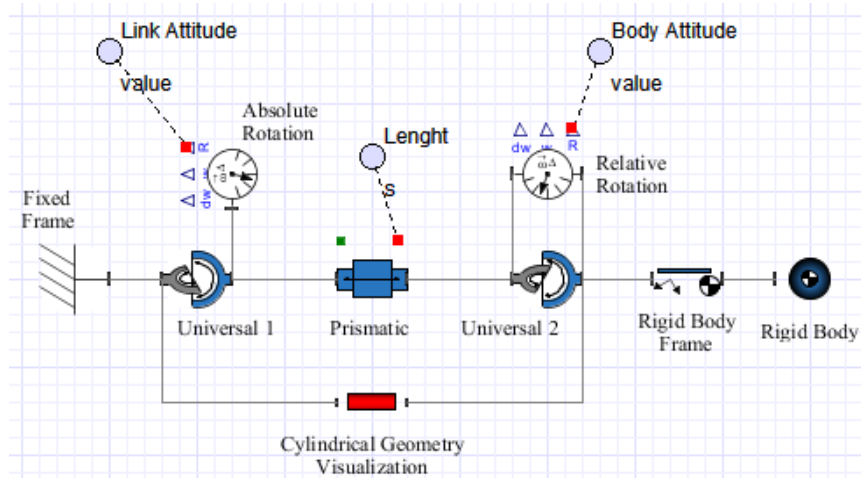


Рис. 9

Prismatic (рис. 11) – шарнир с одной поступательной степенью свободы (движением вдоль заданной оси \hat{e}_1) – имеет следующие параметры: K_s – коэффициент жесткости пружины; L_0 – длина нерастянутой пружины; K_d – коэффициент демпфирования. Кроме того, можно задать начальные условия: перемещение s_0 и линейную скорость v_0 .

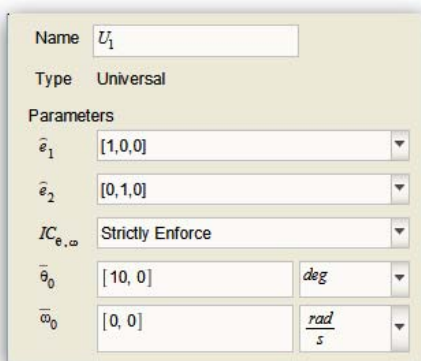


Рис. 10

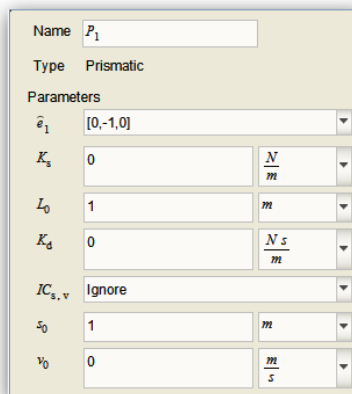


Рис. 11

Компонент Rigid Body имитирует твердое тело с заданной массой (m) и тензором инерции (I). Начало связанной с телом СК расположено в центре масс тела. Параметр $Type_v$ (рис. 12) указывает, относительно какой из СК отсчитывается начальная скорость: связанной с телом (inboard) или внешней по отношению к телу (outboard), в данном случае – неподвижной СК. \bar{r}_0 задает начальное смещение центра масс тела относительно начала неподвижной СК, измеряемое в координатах связанной СК. v_0 задает начальную линейную скорость центра масс. $IC_{r,v}$, $IC_{\theta,\omega}$ определяют применимость на-

чальных условий для линейных и угловых характеристик движения соответственно. $Type_0$ задает последовательность поворотов, определяющих ориентацию неподвижной СК (outboard) по отношению к СК, связанной с телом (inboard). Например, запись [1,2,3] говорит о том, что повороты производятся последовательно вокруг осей x , y и z – последовательность, соответствующая углам Брайнта [9]. $\bar{\theta}_0$ задает начальную ориентацию тела (точнее, неподвижной СК относительно связанной СК). $Type_\omega$ указывает, задается ли начальная угловая скорость в проекциях на оси связанной СК или на оси неподвижной СК (выбор значения Euler означает, что начальная угловая скорость задается как матрица производных углов ориентации тела).

Rigid Body Frame – система координат с заданным смещением и ориентацией относительно связанной с телом СК. Этот компонент удобен для задания точек крепления в системах связанных тел. Параметры Rigid Body Frame (рис. 13) имеют следующий смысл: \bar{r}_{XYZ} – смещение относительно осей связанной СК (в нашем случае 0,1 м); $Type_R$ указывает, определяется ли ориентация компонента при помощи матрицы ориентации $[R]$ (Rotation Matrix) или последовательностью поворотов на углы ζ, η, ξ (Euler Angles) соответственно; $Type_0$ имеет тот же смысл, что и в Rigid Body (принятая по умолчанию последовательность поворотов [1,2,3] соответствует углам Брайнта, хотя в MapleSim эти углы называются Euler Angles); $\bar{\theta}_{\zeta, \eta, \xi}$ – углы поворота компонента.

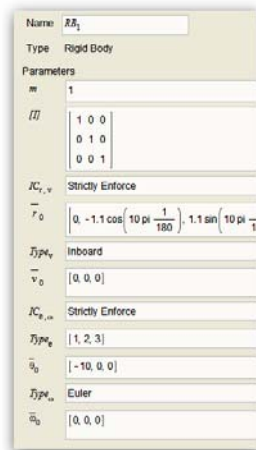


Рис. 12

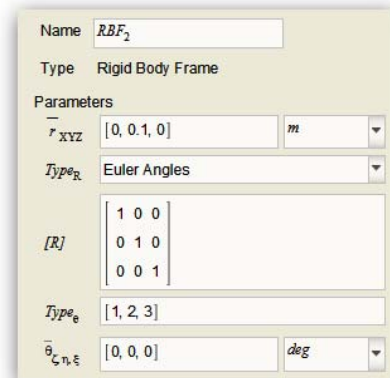


Рис. 13

Компонент Cylindrical Geometry предназначен для визуализации цилиндров заданного цвета и радиуса и служит в нашем случае для отображения троса.

Кроме возможностей, предоставляемых компонентами Multibody / Visualization, MapleSim позволяет загружать данные о геометрии рассматриваемых систем из CAD-программ, обеспечивая высокую реалистичность изображений.

Кроме того, на схеме (рис. 9) присутствуют два датчика (Sensors): Absolute Rotation и Relative Rotation, измеряющие ориентацию (R), угловую скорость (w) и угловое ускорение (dw). Absolute Rotation измеряет характери-

стики вращения тела относительно неподвижной СК, Relative Rotation — характеристики вращения внешней по отношению к телу СК относительно связанной СК. Ориентация тела описывается матрицей ориентации R .

Для вывода результатов показаний датчиков на графики используются зонды (probes). В нашей модели их три: Link Attitude и Body Attitude выводят характеристики вращательного движения троса и тела, а Length — изменение длины троса.

MapleSim полностью интегрирован со средой Maple, так что можно использовать команды, компоненты, инструменты для построения графиков и многие другие возможности Maple для анализа и управления поведением моделей MapleSim (или отдельных подсистем этих моделей). Например, можно получить доступ к сгенерированным автоматически уравнениям движения системы, а затем преобразовать их с помощью Maple.

Для работы с MapleSim-моделью в Maple используются так называемые «шаблоны» (templates). Шаблоны представляют собой рабочие документы Maple с встроенными в них средствами решения различных задач. Доступ к шаблонам в MapleSim осуществляется через меню View/Create Attachment. Далее, из появившегося списка выбирается требуемый шаблон.

Шаблоны, среди прочего, позволяют решать следующие задачи:

- доступ к сгенерированным уравнениям движения (шаблон Equations);
- линеаризация уравнений (Linearization);
- анализ линейных систем (Analysis);
- оптимизация параметров системы (Optimization) (позволяет использовать возможности Global Optimization Toolbox из Maple.)
- анализ чувствительности (Sensitivity Analysis)
- трансляция кода модели в язык C (Code Generation);
- создание пользователем собственных компонент MapleSim (Custom Component).

Следует отметить, что уравнения движения, сгенерированные MapleSim, весьма неудобны для дальнейшего анализа, даже когда речь идет о сравнительно простой системе. Принятая система обозначений компонентов ориентирована на удобство выполнения автоматических преобразований, но не на понимание смысла полученных уравнений (для этого необходимо выполнить ряд нетривиальных замен обозначений). Таким образом, преимущество MapleSim перед другими средствами физического моделирования, не позволяющими выводить уравнения (в частности, перед SimMechanics из MATLAB) оказывается весьма незначительным.

Результаты расчетов. Тестовые расчеты проводились при следующих значениях параметров системы: $m = 1$ кг; моменты инерции тела $J_x = J_y = J_z = 1$ кг·м²; $d = 1$ м; $c = 100$ Н; $\chi = 0$; $\vec{r} = [-0,1; 0; 0]$ (проекция вектора \vec{r} на оси связанной с телом СК).

Каждым из рассматриваемых пакетов строились графики зависимостей углов α (угол отклонения вектора \vec{l} от местной вертикали в плоскости, перпендикулярной вектору мгновенной угловой скорости движения тела; начальное значение $\alpha(0) = 10^\circ$) и ψ (угол отклонения главной центральной оси X инерции тела от направления \vec{l} ; $\psi(0) = 0^\circ$), а также длины нити l

($l(0) = 1$ м) от времени. Результаты расчетов, выполненных в MATLAB, приведены на рис. 14 – 16.

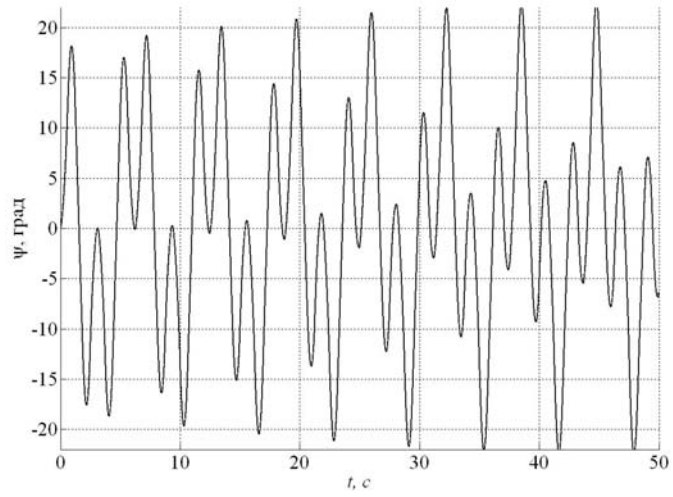


Рис. 14

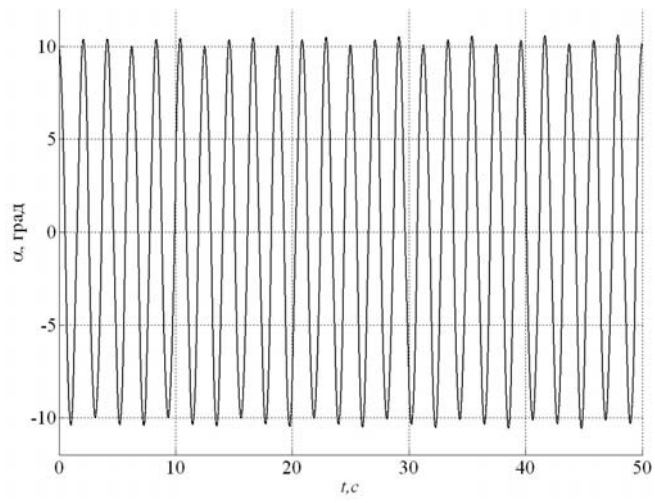


Рис. 15

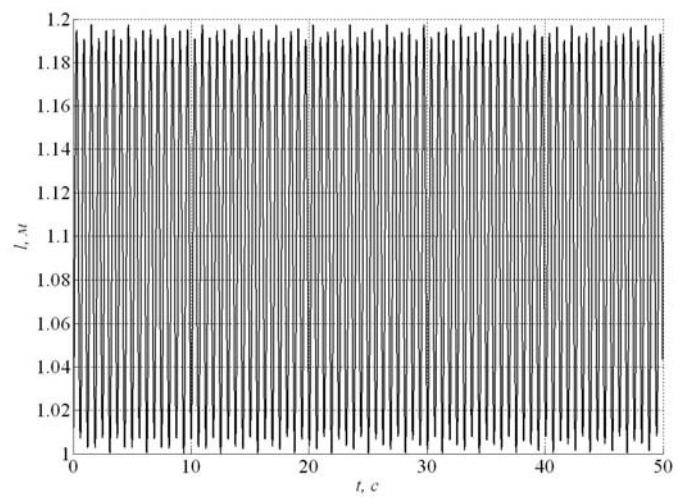


Рис. 16

Результаты расчетов, выполненных в Maple и MapleSim, практически совпали с приведенными выше: максимальная относительная погрешность не превышала относительной погрешности численного интегрирования.

Для сопоставления результатов, полученных в MapleSim, с результатами, полученными в других пакетах, необходимо выполнить некоторые дополнительные преобразования. Дело в том, что датчики, сообщающие данные об ориентации троса и тела, выводят не значения углов, а элементы матриц ориентации. Поэтому необходимо было осуществить переход от элементов матрицы R к углам ориентации.

Помимо графиков, в MapleSim движение системы демонстрируется на видео (кадр из такого видео приведен на рис. 17).

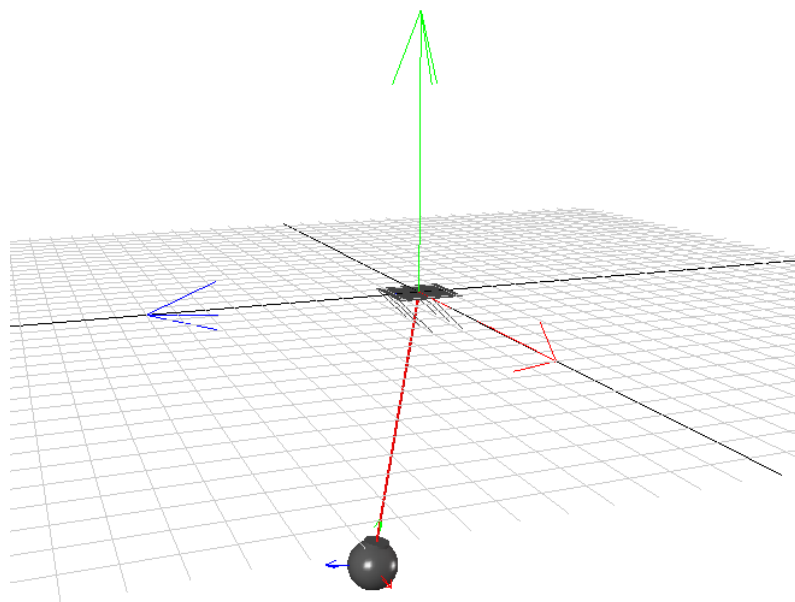


Рис. 17

Выводы. Преимущество MATLAB перед двумя другими рассмотренными пакетами заключается в более низком «пороге вхождения»: работа в MATLAB мало чем отличается от использования других языков программирования. В случае, когда уравнения движения приведены к виду, удобному для программирования, MATLAB позволяет быстрее выполнять численные расчеты⁴. Представляется целесообразным использовать MATLAB для выполнения оценочных расчетов, целью которых является принципиальная проверка работоспособности модели, а также в тех случаях, когда механическая схема системы уже устоялась и не потребует в будущем существенных изменений.

Преобразование уравнений движения с помощью Maple, как правило, требует разработки процедур упрощения исходных уравнений. Однако затраченное на это время окупается при проведении многократных расчетов, требующих внесения изменений в модель. К тому же уравнения движения в ска-

⁴ Это преимущество существенно сократилось благодаря реализации в современных версиях Maple более эффективных численных алгоритмов, а также возможности использовать сторонние библиотеки алгоритмов, в частности библиотеку NAG.

лярном виде, полученные с помощью Maple, могут затем быть использованы в других программных средствах (в частности, в MATLAB).

Сфера возможных применений Maple не ограничивается проведением расчетов, но охватывает все этапы исследования, начиная с вывода уравнений математической модели и проведения оценочных расчетов и кончая оформлением полученных результатов в виде публикации в печатных изданиях или Интернет.

MapleSim позволяет быстро создавать модели механических систем, в том числе и весьма сложных. Особенно удобно применение MapleSim в тех случаях, когда модели создаются из готовых элементов, имеющихся в библиотеке системы.

Особое внимание при моделировании с использованием MapleSim необходимо уделять механической схеме исследуемой системы, так как вывод уравнений движения и дальнейшее изучение поведения системы осуществляется на основе именно этой схемы. Поскольку работа идет на более «высоком» уровне по сравнению с непосредственным выводом уравнений, то и цена ошибки здесь выше. Хотя MapleSim и отсеивает случаи, когда уравнения движения невозможно составить (проверяет синтаксические ошибки, если воспользоваться аналогией с трансляторами), но вполне возможно получить в итоге уравнения, описывающие поведение системы, отличной от задуманной (т. е. совершить логическую ошибку). Для уменьшения риска появления подобных ошибок можно формировать систему постепенно, небольшими обозримыми блоками, каждый из которых должен быть доступен непосредственной проверке.

1. *Белецкий В. В.* Динамика космических тросовых систем / В. В. Белецкий, Е. М. Левин. – М. : Наука, 1990. – 329 с.
2. *Lorenzini E. C.* Tethers in Space Handbook : 3rd edition. / E. C. Lorenzini., M. L. Cosmo. – Smithsonian Astrophysical Observatory, 1997. – 241 p.
3. *Levin E. M.* Dynamic analysis of space tether missions / E. M. Levin. – San Diego : American Astronautical Society, 2007. – 453 p.
4. *Храмов Д. А.* Использование пакета символьных вычислений Maple для моделирования динамики космической тросовой системы со сферическим шарниром // Системные технологии. – 2004. – № 3 (32) – С. 110 – 116.
5. *Волошенко О. Л.* Модель процессов стабилизации движения концевых тел вращающейся космической тросовой системы в наземных экспериментах / О. Л. Волошенко, А. В. Пироженко // Техническая механика. – 2010. – № 3. – С. 106 – 116.
6. Modeling Flexible Bodies in SimMechanics and Simulink / V. Chudnovsky, D. Kennedy, A. Mukherjee, J. Wendlandt // [Электронный ресурс] – Режим доступа : <http://www.mathworks.com/company/newsletters/digest/2006/may/simmechanics.html>
7. *Dkhramov.dp.ua* // Режим доступа : http://dkhramov.dp.ua/uploads/Sci/HomePage/test3paks_code.rar
8. *Neweul-M²*: software package for the dynamic analysis of mechanical systems in Matlab // Режим доступа : http://www.itm.uni-stuttgart.de/research/neweul/neweulm2_en.php.
9. *Виттенбург Й.* Динамика систем твердых тел / Й. Виттенбург. – М. : Мир, 1980. – 292 с.

Институт технической механики
НАН Украины и НКА Украины,
Днепропетровск

Получено 29.06.2011,
в окончательном варианте 06.09.2011