

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
**"ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ"**

Утверждаю
Зав. отделением кафедры ЮНЕСКО

 Ю.М. Осипов

" _____ " _____ 2011 г.

**МОДЕЛИРОВАНИЕ РАБОТЫ МУЛЬТИКООРДИНАТНЫХ СИСТЕМ
ДВИЖЕНИЯ**

Методические указания к выполнению практических занятий и самостоятельной работы по дисциплинам: "Геометрические основы планирования движения микроэлектромехатронных систем (МЭМС)" для магистрантов по направлению 222000.68 «Инноватика» по магистерской программе «Управление инновациями в мехатронике и робототехнике» и "Методы и инструментарий геометрического обеспечения функционирования электромехатронных систем " для магистрантов по направлению 221000.68 «Мехатроника и робототехника» по магистерской программе «Проектирование и исследование мультикоординатных электромехатронных систем движения»



Томск 2012

УДК 621.865.8

Методические указания к выполнению практических занятий и самостоятельной работы по дисциплинам: "Геометрические основы планирования движения микроэлектромехатронных систем (МЭМС)" для магистрантов по направлению 222000.68 «Инноватика» по магистерской программе «Управление инновациями в мехатронике и робототехнике" и " Методы и инструментарий геометрического обеспечения функционирования электромехатронных систем " для магистрантов по направлению 221000.68 «Мехатроника и робототехника» по магистерской программе «Проектирование и исследование мультикоординатных электромехатронных систем движения»: Томск, 2012. – 37 с.

Составитель

доц., канд. техн. наук С.В. Щербинин

Рецензент проф. кафедры ЕНО ЮТИ ТПУ, д.т.н.

А.В. Градобоев

Методические указания рассмотрены на заседании кафедры 27.03.2012г., протокол № 8.

Зав. отд. кафедры ЮНЕСКО

проф., докт. тех. наук., докт. экон. наук



Ю.М. Осипов

ВВЕДЕНИЕ

Имитационное моделирование (ИМ) - один из самых мощных инструментов анализа при разработке сложных систем и анализе процессов их функционирования. Суть имитационного моделирования проста и в то же время интуитивно привлекательна. Его использование дает возможность экспериментировать с существующими или предлагаемыми системами в тех случаях, когда сделать это на реальных объектах практически невозможно или нецелесообразно.

Имитационное моделирование как метод научного исследования предполагает использование компьютерных технологий для имитации различных процессов или операций - моделирования. Устройства или процесс в дальнейшем будут называться системой. Для научного исследования системы применяют определенные допущения, касающиеся ее функционирования. Эти допущения, как правило выражаемые в виде математических зависимостей или логических отношений, представляют собой модель, с помощью которой можно изучать поведение рассматриваемой системы. Если выражения, входящие в модель, достаточно просты для получения точной информации при решении тех или иных вопросов, то можно использовать аналитические методы. Однако большинство реальных систем являются очень сложными, и создать их аналитическую модель не представляется возможным. Такие модели следует изучать путем ИМ: при этом для получения численных результатов, с помощью которых проводят расчет характеристик исследуемой системы, применяют компьютер.

1. ЦЕЛЬ И ЗАДАЧИ КУРСА

Цель изучения практической части курса – выработка у магистрантов навыков решения научных и практических задач связанных с повышением качества функционирования мультикоординатных систем за счет использования их имитационных моделей.

Основными задачами работы являются:

- закрепление теоретических знаний, полученных ранее;
- выработка умения анализировать исходные данные, выбирать и обосновывать методы решения при выполнении профессиональных задач;
- формирование навыков самостоятельной работы и творческого подхода к решению задач, связанных с профилем специальности;
- приобретение практических навыков использования нормативных документов и современных технических средств при решении профессиональных задач и оформлении результатов;
- ознакомиться пакетом расширения Simulink для моделирования механических систем SimMechanics;

– освоение основные принципы создания моделей механических систем.

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К РАБОТЕ

Мощным средством наглядного и эффективного составления программных моделей является приложение к системе Matlab— пакет визуального программирования Simulink.

Пакет Simulink позволяет осуществлять исследование (моделирование во времени) поведения динамических линейных и нелинейных систем, причем составление «программы» и ввод характеристик систем можно производить в диалоговом режиме, путем сборки на экране схемы соединений элементарных (стандартных или пользовательских) звеньев. В результате такой сборки получается модель системы (в дальнейшем будем называть ее S-моделью), которая сохраняется в файле с расширением .mdl. Такой процесс составления вычислительных программ принято называть визуальным программированием.

Создание моделей в пакете Simulink основывается на использовании технологии Drag-and-Drop. В качестве «кирпичиков» при построении S-модели применяются визуальные блоки (модули), которые сохраняются в библиотеках Simulink. S-модель может иметь иерархическую структуру, то есть состоять из моделей более низкого уровня, причем количество уровней иерархии практически не ограничено. В процессе моделирования есть возможность наблюдать за процессами, которые происходят в системе. Для этого используются специальные блоки («обзорные окна»), входящие в состав библиотеки Simulink. Библиотека может быть пополнена пользователем за счет разработки собственных блоков.

В основе блок-схем S-моделей лежат элементарные блоки, позволяющие связать блок-схему со средой MATLAB и обеспечить функционирование в ней S-модели как программы. Эти блоки размещены в главной библиотеке пакета Simulink, которая имеет то же название.

Начать работу с пакетом Simulink можно из командного окна MATLAB, щелкнув для этого на соответствующей кнопке панели инструментов. В результате на экране должно появиться окно SimulinkLibraryBrowser (Браузер библиотек Simulink) (рис. 1).

В левой половине окна браузера приведен перечень библиотек, включенных в состав Simulink, а в правой - перечень разделов соответствующей библиотеки либо изображения блоков соответствующего раздела.

Ядром пакета Simulink является библиотека Simulink, указанная в первой строке браузера. Остальные библиотеки пакета по необходимости включаются в состав общей библиотеки в соответствии с предпочтениями пользователя. Чтобы ознакомиться с составом какой-либо из библиотек следует дважды щелкнуть на ее имени. Библиотеку можно открыть и в отдельном ок-

не. Для этого следует щелчком правой кнопки мыши на имени библиотеки вызвать контекстное меню и выбрать команду открытия.

Чтобы начать сборку блок-схемы моделируемой системы, необходимо в командном окне MATLAB вызвать команду **Файл ► Новый ► Модель**. После этого на экране появится новое (пустое) окно untitled, в котором и будет осуществляться сборка S-модели (MDL-файла) (рис. 2).

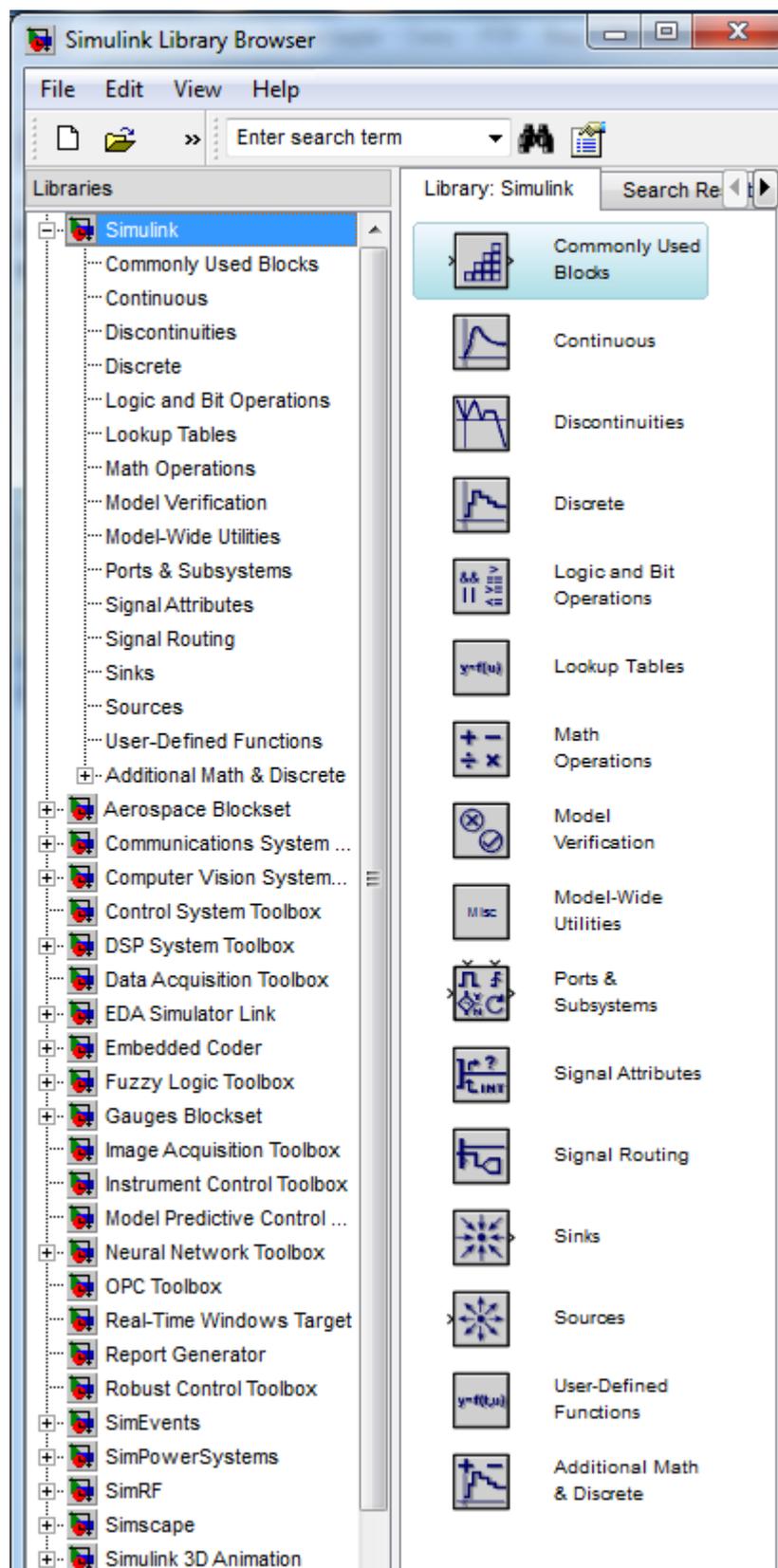


Рис. 1. Окно библиотеки Simulink

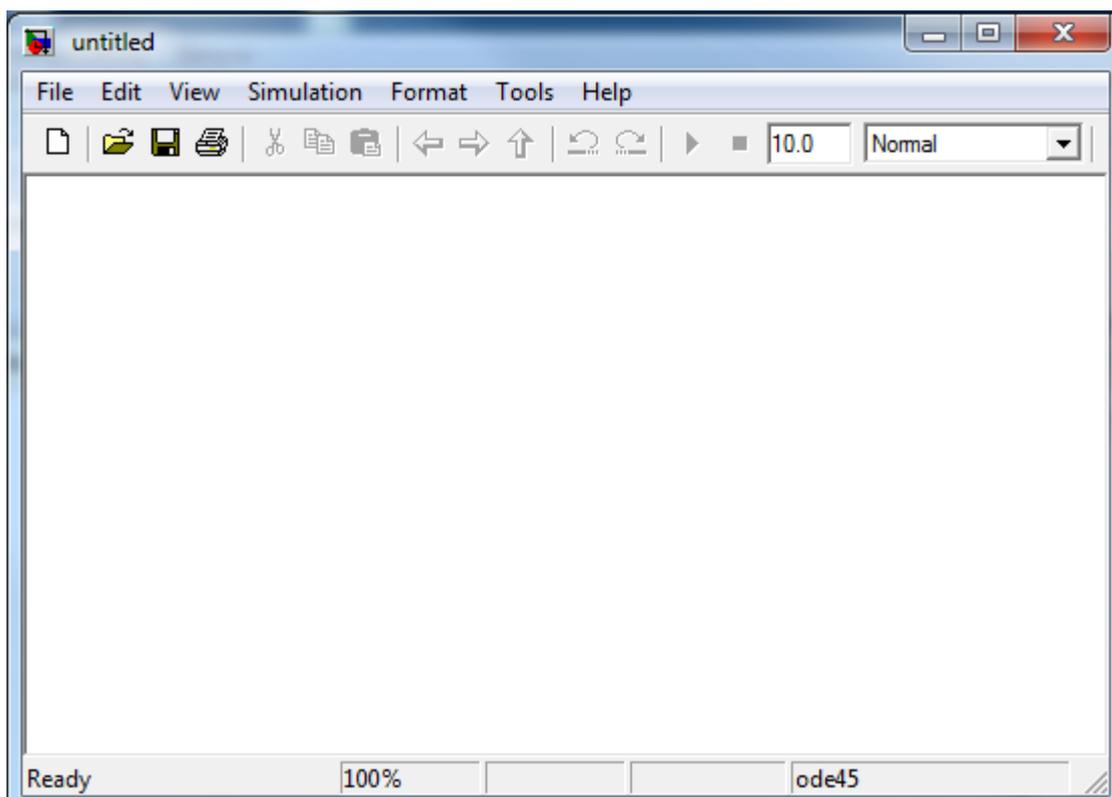


Рис. 2. Скриншот окна для сборки S-модели

Окно имеет строку меню, панель инструментов и рабочее поле. Меню File (Файл) содержит команды, предназначенные для работы с MDL-файлами; меню Edit (Правка) - команды редактирования блок-схемы; меню View (Вид) - команды изменения внешнего вида окна; меню Simulation (Моделирование) - команды управления процессом моделирования; меню Format (Формат) - команды редактирования формата (то есть команды, позволяющие изменить внешний вид отдельных блоков и блок-схемы в целом). Меню Tools (Инструменты) включает некоторые дополнительные сервисные средства, предназначенные для работы с S-моделью.

Сборка блок-схемы S-модели заключается в том, что графические изображения выбранных блоков с помощью мыши перетягиваются из окна раздела библиотеки в окно блок-схемы, а затем выходы одних блоков в окне блок-схемы соединяются со входами других блоков (также с помощью мыши). Соединение блоков выполняется следующим образом: указатель мыши подводят к определенному выходу нужного блока (при этом указатель должен приобрести форму крестика), нажимают левую кнопку и, не отпуская ее, перемещают указатель к нужному входу другого блока, а потом отпускают кнопку. Если соединение осуществлено верно, на входе последнего блока появится изображение черной стрелки.

Любая блок-схема моделируемой системы должна включать в себя один или несколько блоков-источников, генерирующих сигналы, которые, собственно, и вызывают «движение» моделируемой системы, и один или несколько блоков-приемников, которые позволяют получить информацию о выходных сигналах этой системы (увидеть результаты моделирования).

Библиотека блоков Simulink - это набор визуальных объектов, при использовании которых, соединяя отдельные модули между собой линиями связей, можно составлять функциональную блок-схему любого устройства.

Если с помощью контекстного меню вызвать библиотеку Simulink, на экране появится окно Library: simulink, в котором представлены графические обозначения разделов этой библиотеки (рис. 3).

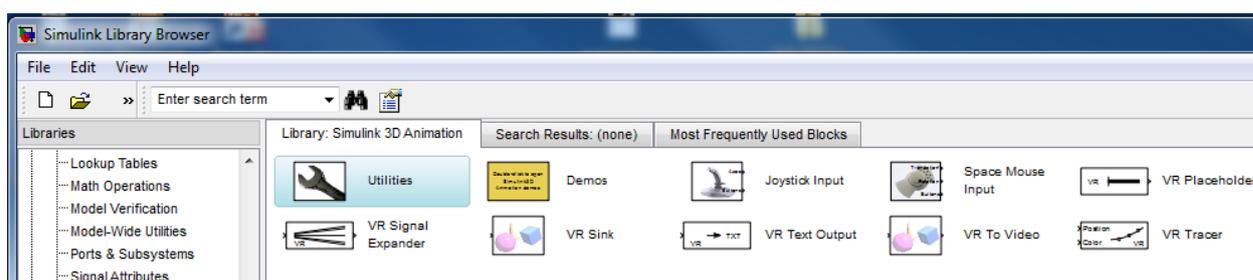


Рис. 3. Графические одного из разделов библиотеки *Simulink*

Моделирование механического движения механизмов и машин осуществляется с помощью библиотеки SimMechanics пакета Simulink. Идеология составления блок-схем в этой библиотеке, как и в ранее рассмотренной библиотеке SimPowerSystems, существенно отличается от идеологии составления функциональных блок-схем библиотеки Simulink. В блок-схеме SimMechanics отдельные блоки следует рассматривать как модели, имитирующие механическое движение одной части моделируемого механизма относительно другой. Входы и выходы блока фактически таковыми не являются, а имитируют «посадочное место» соответствующей части механизма. Линии соединения «входов» и «выходов» блоков имитируют жесткие соединения выходной части одного механизма с входной частью другого. Можно утверждать, что это соединение моделирует передачу силового воздействия между частями разных механизмов. Но поскольку в соответствии с третьим законом Ньютона сила действия равна силе противодействия, такую передачу силы нельзя рассматривать как однонаправленное воздействие. Поэтому в блок-схемах SimMechanics на линиях соединений механических блоков вы не встретите изображений стрелок, указывающих направление воздействия. По той же причине графические изображения «входов*» и «выходов*» механических блоков имеют вид не стрелок, а квадратов с диагоналями.

Как и в блоках библиотеки SimPowerSystems, «входы» и «выходы*» механических

блоков нельзя рассматривать в качестве источников и приемников каких бы то ни было сигналов. К их соединительным линиям нельзя подсоединить обычные S-блоки, а потому нельзя и сформировать с помощью последних заданные воздействия или вывести информацию о получаемых в результате движениях механизмов (например, в обзорные окна или непосредственно в среду MATLAB). Но так как любое моделирование механизмов невозможно осуществить без указания нужных исследователю воздействий и без вывода результатов моделирования в среду MATLAB, такая идеология построения блок-схем механизмов требует включения в библиотеку блоков, осуществляющих прямую и обратную связь S-блоков с механическими блоками. Связующие блоки должны иметь хотя бы один т-вход и один механический «выход» (для «восприятия» заданного воздействия и перевода его в механическое), либо механический «вход» и т-выход (для отображения результатов моделируемого механического движения в виде информационного сигнала).

Если с помощью контекстного меню открыть библиотеку SimMechanics из окна SimulinkLibraryBrowser (Браузер библиотек Simulink), то на экране появится окно, показанное на рис. 4.

В состав библиотеки входит шесть разделов:

– Bodies (Тела) - содержит блоки, моделирующие движения твердых тел;

– Joints (Сочленения) - включает блоки имитации механических сочленений, обеспечивающих требуемые степени свободы одной части механизма относительно другой;

– Constraints&Drivers(Связи) - состоит из блоков имитации ограничений на степени свободы механической системы;

– Sensors&Actuators(Датчики и приводы) - содержит блоки, имитирующие измерители параметров механического движения, и блоки, позволяющие задавать движение частей механизма;

– Utilities(Утилиты) - включает вспомогательные блоки, которые могут использоваться при создании модели механизма;

– Demos(Демонстрационные программы) позволяет вызвать на исполнение демонстрационные модели.

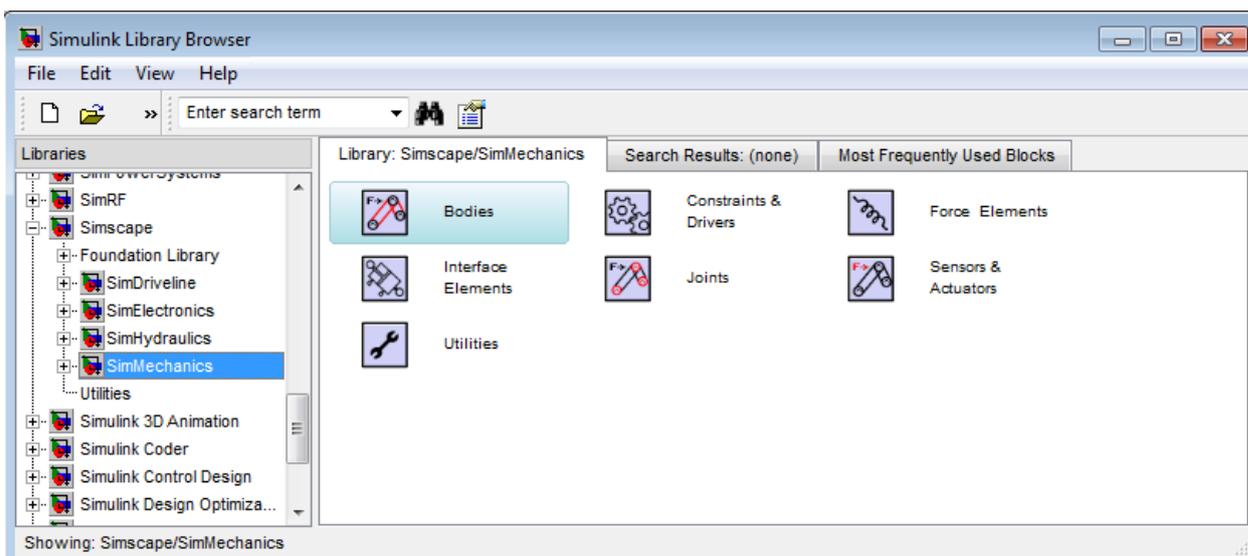


Рис. 4. Окно раздела Simmechanics

При составлении блок-схем механизмов следует принимать во внимание следующие особенности.

- Основу блок-схемы любого механизма составляет цепь типа Ground-Joint-Body-Joint-...-Body с открытой или закрытой топологией, где по крайней мере одно из тел представлено блоком Ground. Блоки Body могут быть соединены более чем с двумя блоками Joint с фиксацией разветвления указанной последовательности. Но каждое сочленение (блок Joint) должно быть подсоединено к двум и только двум телам.

- Блоки Body могут быть соединены и с помощью блоков Driver или Constraint и митирующих связи.

- Блоки Actuator и Sensor могут быть подсоединены к любому из блоков BodyJoint или Driver, но только через дополнительные порты, которые устанавливаются в окнах настройки этих блоков.

- Задать желаемый закон изменения во времени параметров движения можно с помощью блоков Actuator, а вывод результатов в рабочее пространство MATLAB - посредством блоков Sensor, которые связывают блоки библиотеки bimMechanics со средой Simulink.

Раздел Bodies

Раздел Bodies (Тела) содержит два блока: Ground и Body (рис. 5). Блок Ground(Основание) является обязательным при построении модели любого механизма.

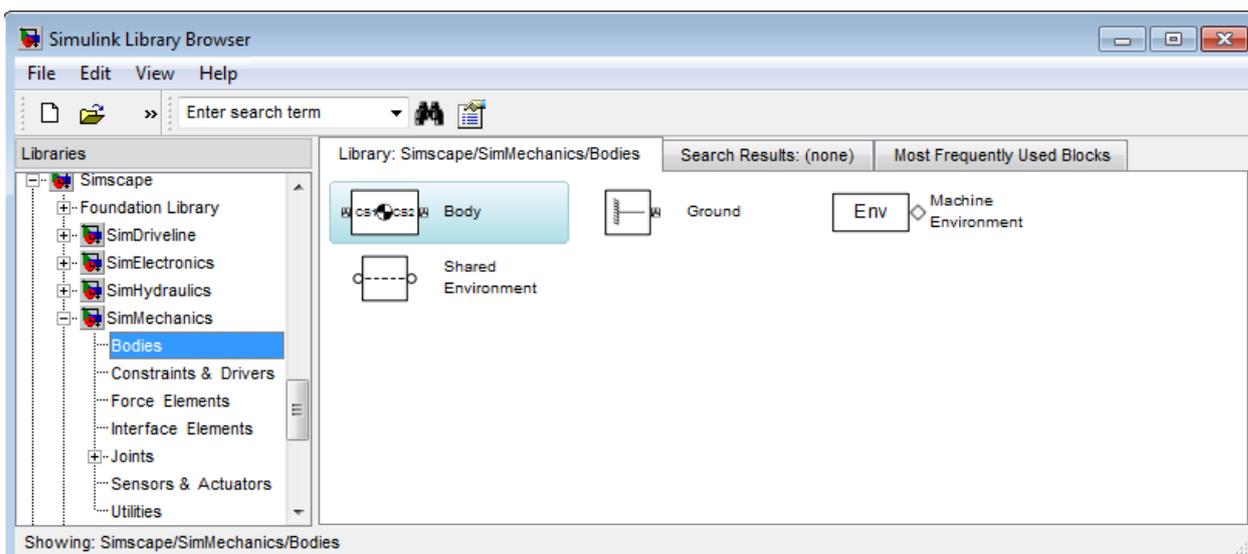


Рис. 5. Блоки раздела Bodies

Он представляет неизменные точки основания (Земли), неподвижные в абсолютном (инерциальном) пространстве. Движение отдельных частей механизма задается или определяется по отношению к системе координат, воплощаемой именно этим блоком.

В окне настройки блока Ground устанавливается лишь один параметр векторсмещения начала системы координат, связанной с неподвижной частью механизма относительно начала инерциальной системы координат.

Блок Body представляет отдельную часть механизма, рассматриваемую как твердое тело, движение которого моделируется. В окне настройки блока задаются масса и матрица инерции этого твердого тела, его начальное положение и ориентация (то есть положение и ориентация систем координат CS, жестко связанных с ним). В число систем координат, жестко связанных с телом, обязательно входит система CG (Center of Gravity), начало которой совмещено с центром тяжести тела. Именно относительно осей этой системы координат задается матрица моментов инерции тела. В поле Mass (Масса) указывается величина массы той части механизма, которая представлена как твердое тело. В поле Inertia (Моменты инерции) задается квадратная матрица (3x3) моментов инерции тела относительно ортогональных осей, жестко связанных с телом и проходящих через его центр тяжести. Область Body coordinates systems (Системы координат, связанные с телом) содержит две вкладки: Позиция и Ориентация. Во вкладке Позиция (рис. 6) расположена таблица ввода координат начала систем координат, связанных с телом. По умолчанию эта таблица содержит три строки и позволяет задать координатные системы, связанные с телом: CG, начало которой совмещено с центром тяжести тела, CS1, «привязанную» к левому порту («входу») блока Body и CS2, «привязанную» к правому порту («выходу») блока Body.

В каждой строке предусмотрен ввод семи характеристик соответствующей системы координат:

- Showport (Показать порт) - установка (или сброс) флажка в этом столбце позволяет ввести в изображение блока (или убрать из него) изображение порта, связанного с соответствующей системой координат тела;
- Portside (Сторона порта) - этот столбец позволяет установить изображение порта слева или справа на изображении блока;
- Имя - в этом столбце устанавливается идентификатор вводимой системы координат;
- Originpositionvector $[x, y, z]$ (Вектор положения начала) - этот столбец содержит вектор координат начала соответствующей системы координат;
- Единицы измерения - здесь устанавливаются единицы измерения для вектора координат начала соответствующей системы координат;
- Translatedfromoriginof (Отсчитывается от начала системы координат) - здесь указывается имя (идентификатор) системы координат, от начала которой отсчитываются координаты устанавливаемой системы координат;
- Componentsinaxesof (Компоненты в осях системы координат) - в этом столбце указывается имя (идентификатор) системы координат, относительно которой отсчитываются координаты начала устанавливаемой системы координат.

В раскрывающемся списке на вкладке Позиция (см. рис. 6) вы можете выбрать систему отсчета. В него входят инерциальная система отсчета WORLD, все системы координат, имеющиеся на вкладке, а также система ADJOINING, под которой понимается система координат, жестко связанная с тем сочленением, которое подсоединено к телу.

Вкладка Ориентация (рис. 7) устроена аналогичным образом, она позволяет установить начальную угловую ориентацию вводимой системы координат. Отличия заключаются в следующем. Вместо вектора координат начала в четвертом столбце вводится вектор углов поворота вводимой системы координат относительно исходной, имя которой указывается в шестом столбце. При этом принимаемая последовательность поворотов вокруг координатных осей выбирается в седьмом столбце.

Каждому месту соединения двух тел (блоков Body) соответствует своя отдельная система координат CS. Количество точек соединения тела с другими телами (а значит, и количество связанных с телом систем координат) можно увеличить или уменьшить, пользуясь кнопками, расположенными в правом верхнем углу области Bodycoordinatesystems.

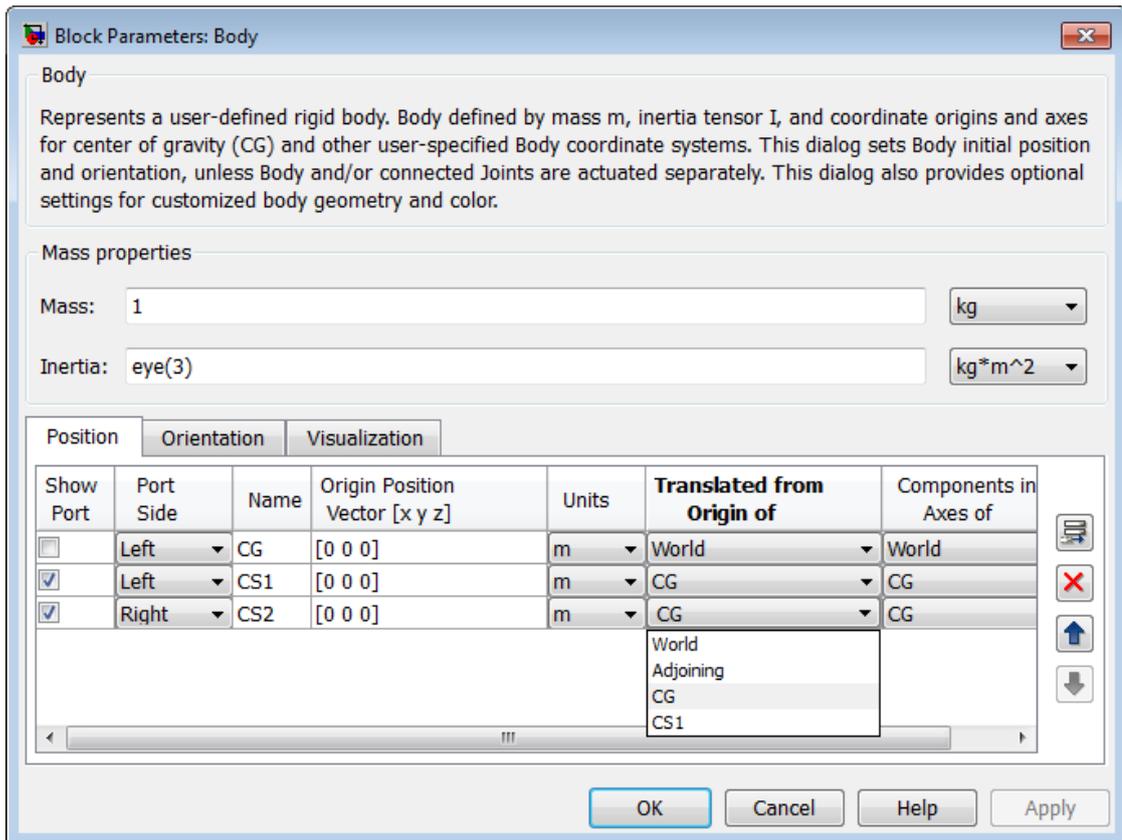


Рис. 6. Вкладка Позиция окна настройки блока Body

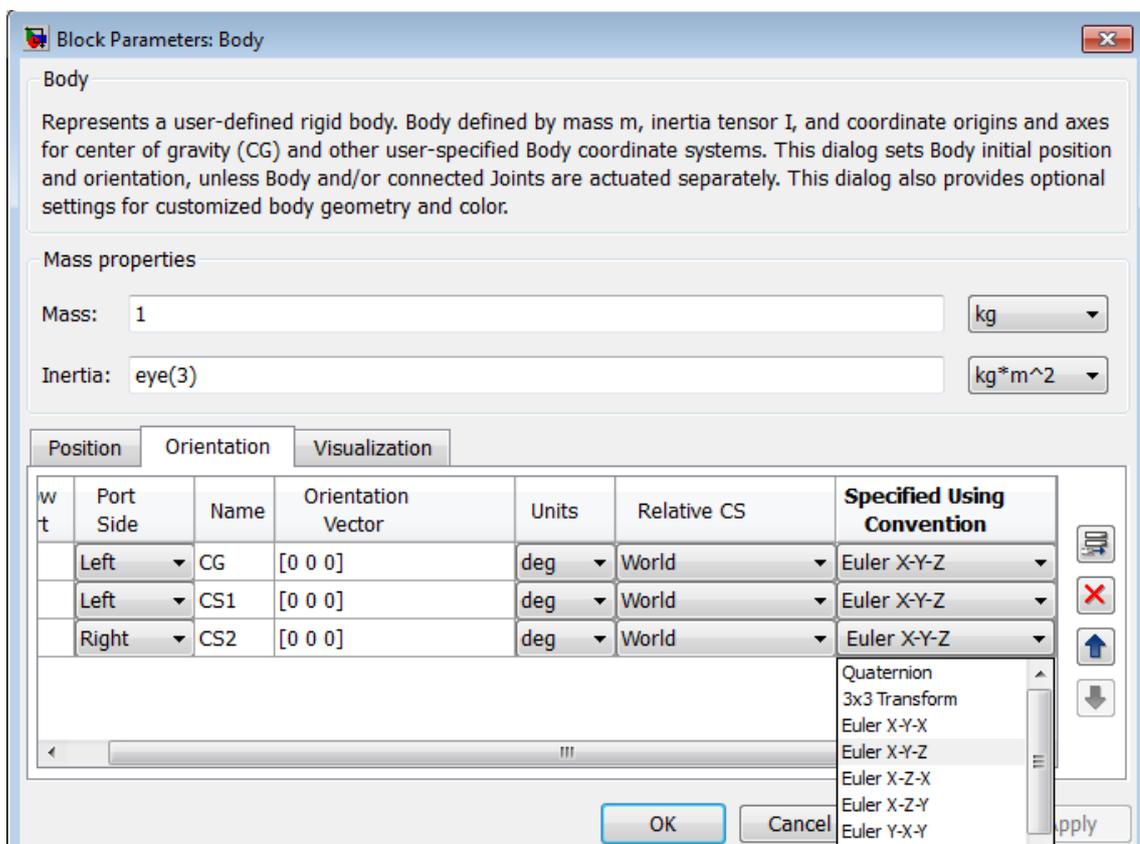


Рис. 7. Вкладка Ориентация окна настройки блока Body

Раздел Joints

Раздел Joints (Сочленения) содержит блоки, позволяющие обеспечить возможность относительных движений тел, представленных отдельными блоками Body, то есть необходимых степеней свободы.

Кроме 15 блоков, имитирующих различные виды сочленений, в разделе Joints находятся два подраздела - DisassembledJoints (Разобранные сочленения) и MasslessConnectors (Безинерционные соединители) (рис. 8).

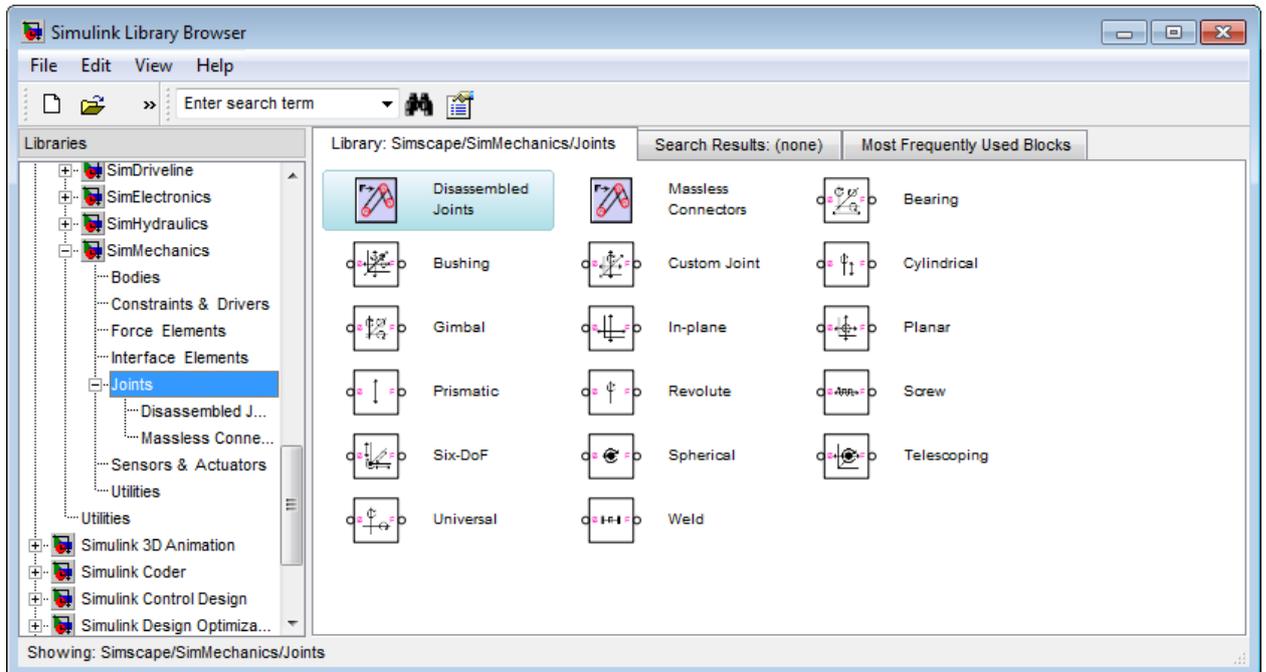


Рис. 8. Блоки раздела Joints

Рассмотрим некоторые наиболее простые и важные блоки сочленений. Блок Prismatic обеспечивает одну поступательную степень свободы вдоль оси, указанной во вкладке Оси окна его настройки. Такой осью свободного перемещения является третья ось (z) инерциальной системы координат (рис. 9). Как видно на рисунке, имеется возможность связать ось относительного перемещения также с одной из осей первого тела, с которым связан блок Prismatic (выбрав элемент Base из предлагаемого списка), либо с одной из осей системы координат, связанной со вторым телом (выбрав элемент Follower).

В поле Connectionparameters (Параметры соединения) естанавливаются три параметра: Currentbase (Текущая база), Currentfollower (Текущее ведомое тело и Numberofsensor/actuatorports (Число портов для измерителей и возбудителей движения). Первые два параметра (не устанавливаемые пользователем) указывают название блока тела, к которому подсоединен соответствующий порт блока Joint. Если блок Joint не подсоединен к телам, то напротив этих параметров появляется запись «notconnected» (не соединен). Третий параметр позволяет ввести дополнительные порты блока.

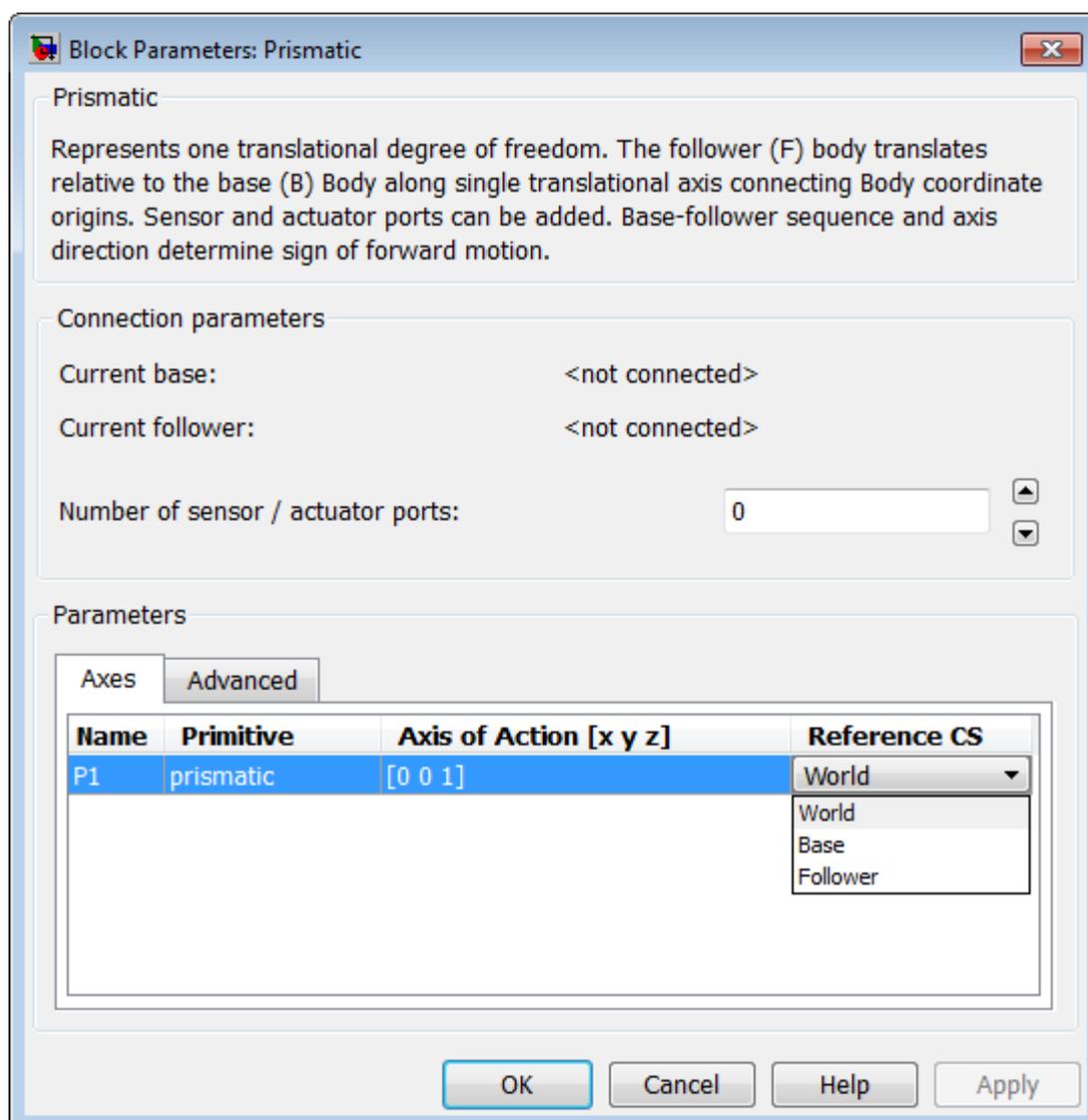


Рис. 9. Окно настройки блока Prismatic

Образует простейшую цепь из блоков Ground, Prismatic и Body (рис. 10), одновременно установив в блоке Prismatic два дополнительных порта для подсоединения блоков Actuator и Sensor. После этого окно настройки блока Prismatic будет выглядеть так, как показано на рис. 11.

Как видим, теперь параметр Currentbase имеет значение GND@Ground, а параметра Currentfollower - CS1@Body. Это означает, что база отождествлена с системой координат GND блока Ground, а ведомое тело Follower связано с системой координат CS1 блока Body. При этом на изображении блока Prismatic появились изображения двух дополнительных портов. С ними теперь можно соединить блоки измерителей (Sensor) и (или) возбудителей (Actuator) движения.

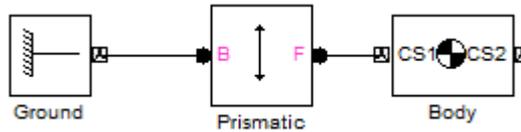


Рис. 10. Простейшая механическая цепь

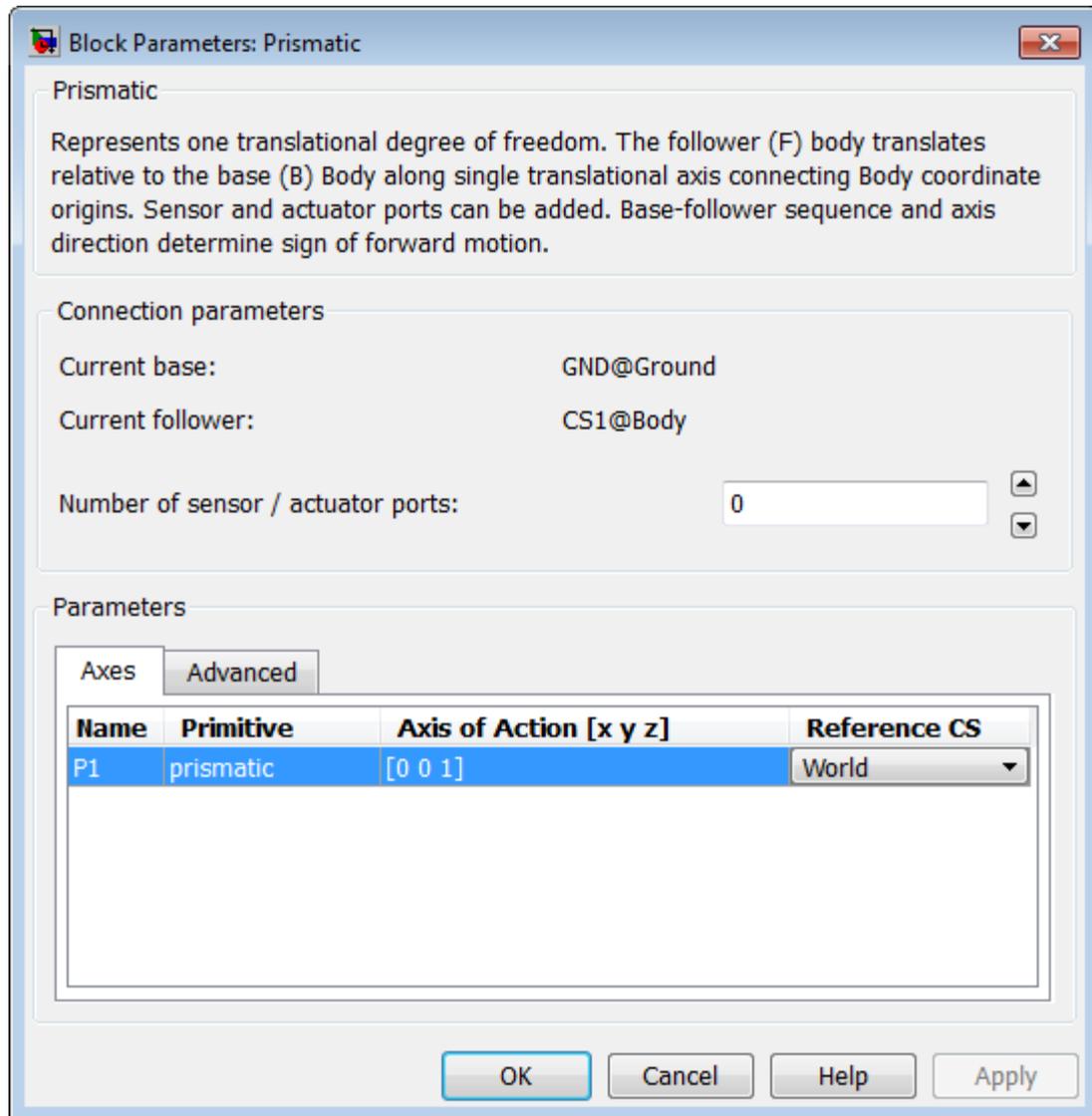


Рис. 11. Окно настройки подсоединенного блока Prismatic

В окне настройки блока Revolute (Цилиндрический шарнир), устанавливается направление оси вращения тела Follower относительно тела Base (рис. 12).

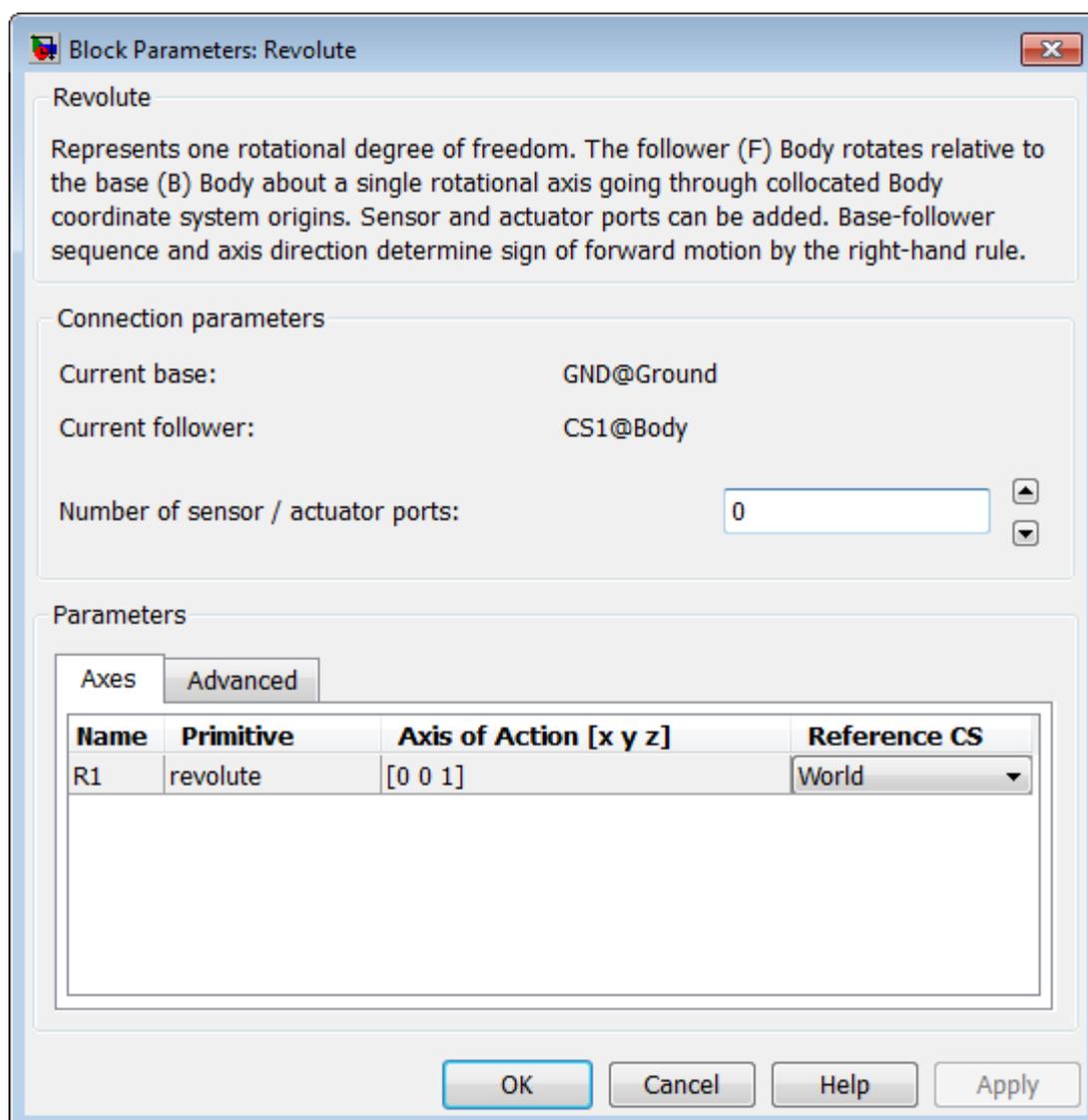


Рис. 12. Окно настройки блока Revolute

Следует заметить, что элементарное сочленение типа Prismatic имеет внутреннее обозначение «P*», а сочленение типа цилиндрического шарнира — обозначение «R*».

Блок In-plane обеспечивает свободу относительного поступательного движения двух тел в плоскости осей, направление которых установлено в окне настройки (рис. 13). Нетрудно убедиться, что блок представляет собой последовательносоединение двух элементарных сочленений типа Prismatic. Первое из них (P1) обеспечивает свободу перемещения второго (P2) вдоль оси, направление которой указывается в первой строке вкладки Оси окна настройки. Во второй строке той же вкладки устанавливается направление второй оси свободного перемещения.

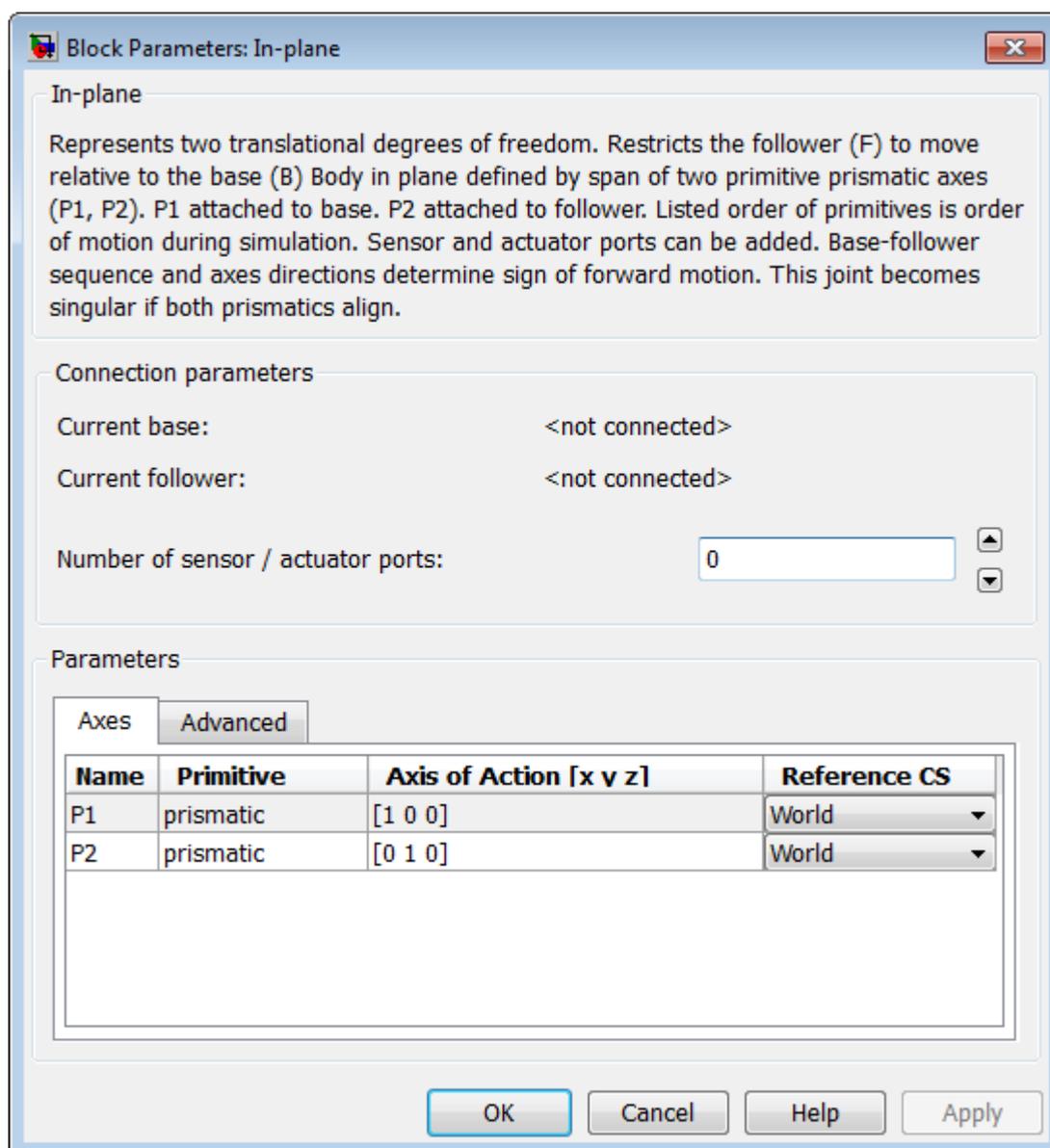


Рис. 13. Окно настройки блока In-Plane

Блок Universalобеспечивает свободу поворота тела Followerотносительно тела Baseвокруг двух осей, задаваемых в окне его настройки. Блок представляет собой последовательное соединение двух элементарных сочленений R1 и R2 типа Revolute.

Блок Gimbal(Карданов подвес) представляет собой последовательное соединениетрех элементарных сочленений R1, R2 и R3 тина Revoluteи обеспечивает свободууглового перемещения одного тела относительно другого вокруг трех, в общемслучае некомплаиарных, осей, указанных в окне настройки.

Блок Spherical(Сферический шарнир), как и предыдущий блок, обеспечивает три угловые степени свободы относительного перемещения двух тел. Отличия заключаются в следующем. Во-первых, в блоке Sphericalнет явно выраженных осей вращения,и поэтому они не устанавливаются в его окне настройки. Во-вторых, вследствиеэтой особенности, к блоку Sphericalне мо-

жет быть подсоединен блок возбуждения движения (Actuator), а параметры относительных поворотов тел не могут быть представлены в углах поворотов; они могут задаваться лишь в виде вектора составляющих кватерниона поворота второго тела (Follower) относительно первого (Base).

С помощью блока Planar(Плоское движение) можно обеспечить такое сочленение двух тел, когда одна из плоскостей как ведущего, так и ведомого тела сохраняется неизменной, а точки ведомого тела могут занимать любое положение в соответствующей плоскости ведущего тела. Блок представляет собой последовательное соединение трех элементарных сочленений двух призматических (P1 и P2) и одного цилиндрического шарнира (R1). Для обеспечения плоского движения необходимо, чтобы указанные в блоке настройки оси были некопланарными.

Блок Cylindrical(Цилиндрическое сочленение) позволяет имитировать такое соединение двух тел, которое допускает свободный поворот вокруг указанной оси одновременно поступательным перемещением вдоль этой же оси. Он представляет собой последовательное соединение сочленения P1 типа Prismatic и сочленения R1 типа Revolute.

Особое место занимает блок CustomJoint(Наборное сочленение). Он предоставляет пользователю возможность самому сконструировать произвольную последовательную цепь из элементарных сочленений. Для этого предназначен набор(в виде списка, расположенного в первой колонке вкладки Оси) из трех примитивов P1, P2, P3 типа Prismatic, трех сочленений R1, R2, R3 типа Revolute и одного примитива S-типа Spherical(рис. 14).

Ограничения состоят в следующем. Набор должен обеспечивать не более шести степеней свободы — три угловые и три поступательные. Никакие две оси примитивов Prismatic или две оси примитивов Revolute не должны быть параллельными (в этом случае сочленение вырождается).

Блок Weld(Жесткое соединение) служит для имитации жесткого соединения двух тел. Его удобно использовать для построения моделей поводковых механизмов.

Блок Bushing обеспечивает шесть степеней свободы (три поступательные и три угловые); он состоит из последовательно соединенных трех примитивов P1, P2, P3 типа Prismatic и трех сочленений R1, R2, R3 типа Revolute.

Подобную функцию выполняет блок Six-DoF(Шесть степеней свободы). Единственным отличием данного блока от блока Bushing является то, что для обеспечения трех угловых степеней свободы используется один примитив S-типа Spherical вместо трех примитивов типа Revolute.

Особенностью блока Screw(Винт), обеспечивающего степень свободы относительного перемещения двух тел по винту, является наличие дополнительно устанавливаемого параметра Pitch(Шаг винта) (рис. 15).

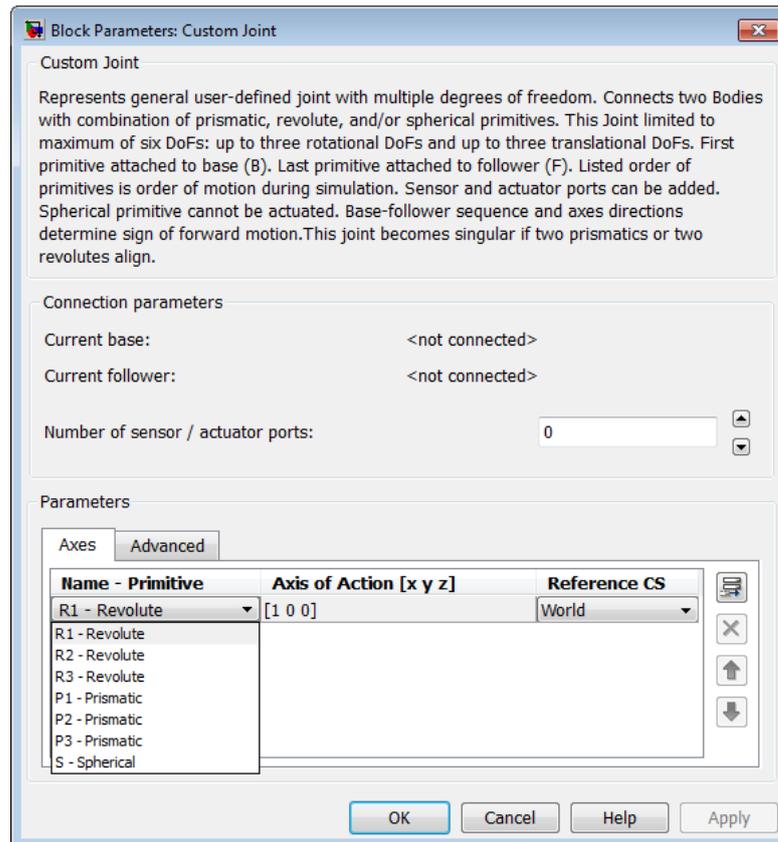


Рис. 14. Окно настройки блока CustomJoint

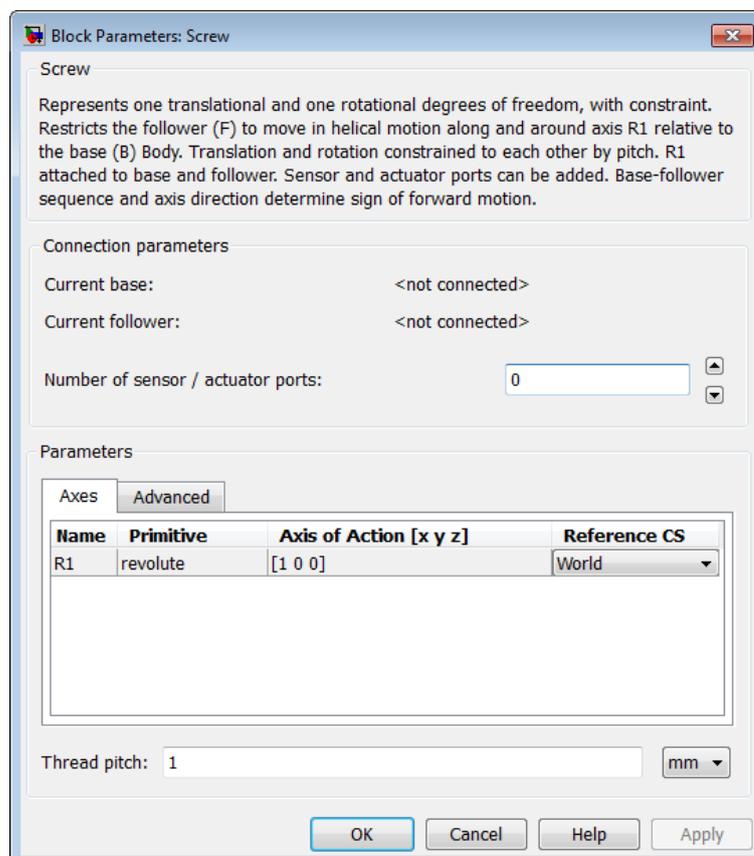


Рис. 15. ОкнонастройкиблокаScrew

Раздел Sensors & Actuators

Раздел Sensors&Actuators(Датчики и приводы) (рис. 16) включает блоки Sensors, которые служат для измерения относительных движений тел, и блоки Actuators, задающие относительные движения.

Как ранее было отмечено, блоки типа Joint(Сочленения) могут быть снабжены дополнительными портами для подсоединения к ним блоков Actuators и Sensor. Аналогичная операция возможна и по отношению к блокам Body, Driver и Constraint.

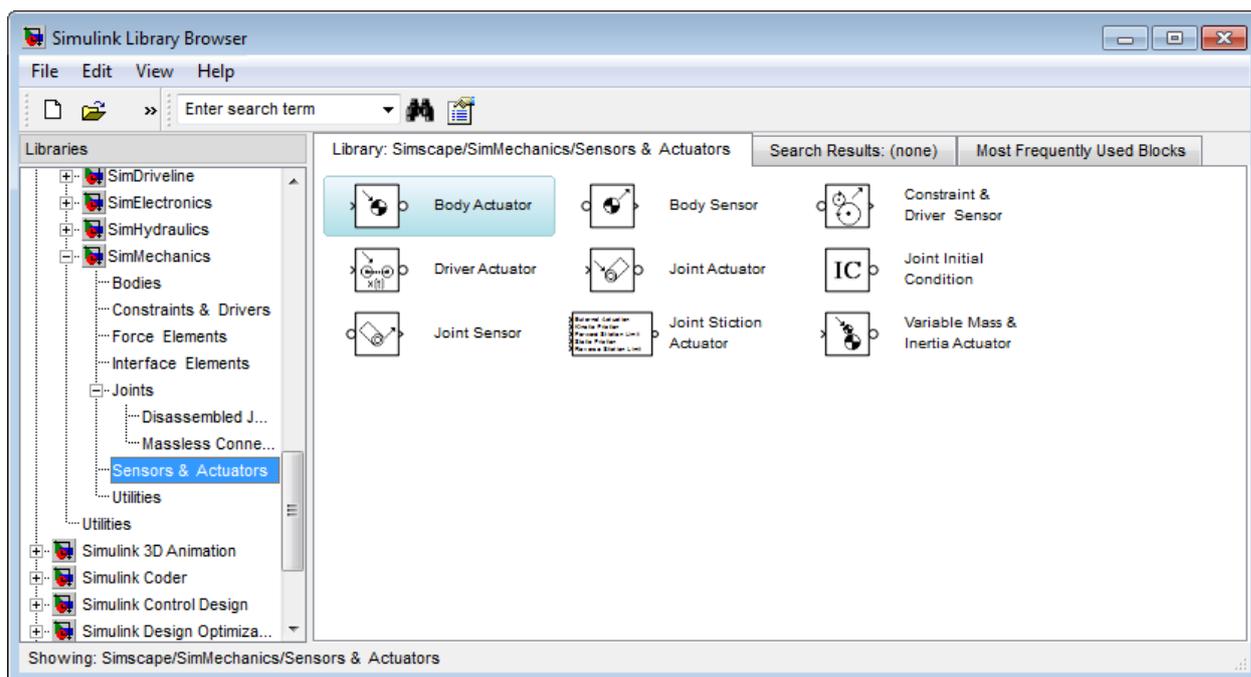


Рис. 16. Блоки раздела Sensors&Actuators

Особенность блоков Actuators и Sensors состоит в том, что они являются связующим между механическими блоками библиотеки SimMechanics и обычными S-блоками. Это позволяет использовать возможности библиотеки Simulink для формирования сигналов, их преобразования и перевода получаемых результатов в рабочем пространстве Simulink.

Блоки возбудителей (Actuator) и измерителей (Sensor) относительного движения разделены на три группы.

- Body Actuator (Возбудитель движения тела) и Body Sensor (Измеритель движения тела) - предназначены для подсоединения к блокам Body; задают или измеряют абсолютное движение жестко связанной с телом системы координат, к которой они подсоединены.

- Joint Actuator (Возбудитель движения сочленения) и Joint Sensor (Измеритель движения сочленения) - предназначены для подсоединения к блокам Joint; задают или измеряют относительное движение примитива, указанного в окне настройки.

- DriverActuator(Возбудитель нестационарной связи) и Driver&ConstraintSensor(Измеритель движения связи) предназначены для подсоединения к блокам Driverили Constraint; задают или измеряют относительное движение связи, с которой они соединены.

Вначале остановимся на особенностях использования блоков JointActuatorи Joint

Sensor. На рис. 17 и 18 приведены окна настройки блоков. Рассмотрев эти рисунки, можно сделать следующие выводы.

- С помощью блока JointActuatorможно в общем случае задать (как функцию времени) либо силовое взаимодействие между элементами примитива, имя которого указывается в верхнем поле ввода, либо относительное движение элементов этого примитива. Установка вида возбуждения осуществляется путем активизации соответствующего переключателя - Generalized-Forces(Обобщенные силы) или Motion(Движение).

- Относительное движение частей примитива задается в виде векторного сигнала из трех элементов, первый из которых определяет относительное перемещение, второй относительную скорость, а третий относительное ускорение частей указанного элементарного сочленения.

- Блок JointSensorпозволяет в общем случае измерить следующие характеристики относительного движения частей примитива, имя которого устанавливается в верхнем поле ввода окна настройки блока:

- Angle(Угол) — угол поворота выходной части примитива (Follower) относительно его части, соединенной со входом (Base);
- Angularvelocity(Угловая скорость) - относительная угловая скорость;
- Angularacceleration(Угловое ускорение) - относительное угловое ускорение;
- Computedtorque(Вычисленный момент) — полный момент сил, вызывающий относительное угловое ускорение;
- Позиция — перемещение выходной части примитива (Follower) относительно его части, соединенной со входом (Base);
- Velocity(Скорость) относительная скорость;
- Acceleration(Ускорение) - относительное ускорение;
- Computedforce(Вычисленная сила) - полная сила, вызывающая относительное ускорение;
- Quaternion(Кватернион) - вектор из четырех элементов, характеризующий текущее относительное угловое положение частей примитива;
- Quaternion, derivative(Производная от кватерниона по времени) - вектор из четырех элементов, являющихся производными по времени от соответствующих элементов кватерниона относительного поворота;
- Quaternion, secondderivative(Вторая производная от кватерниона по времени) — вектор из четырех элементов, являющихся вторыми производными по времени от соответствующих элементов кватерниона относительно поворота;

3. ПРИМЕРЫ СОЗДАНИЯ МОДЕЛЕЙ МАНИПУЛЯТОРОВ

Как правило, моделирование объектов, помимо чисто научных целей может иметь и прикладное значение. Для проектирования и анализа механических систем (например, различных кинематических цепей) давно разработан специальный физико-математический аппарат.

SimMechanics - пакет расширения системы Simulink для физического моделирования. Его цель - техническое проектирование и моделирование механических систем (в рамках законов теоретической механики). SimMechanics позволяет моделировать поступательное и вращательное движения в трех плоскостях. SimMechanics содержит набор инструментов для задания параметров звеньев (масса, моменты инерции, геометрические параметры), кинематических ограничений, локальных систем координат, способов задания и измерения движений.

SimMechanics позволяет создавать модели механических систем подобно другим Simulink-моделям в виде блок-схем. Встроенные дополнительные инструменты визуализации Simulink позволяют получить упрощенные изображения трехмерных механизмов как в статике, так и в динамике. Любой механизм можно представить в виде совокупности звеньев и сочленений. Например, двухзвенный физический маятник (см. рис. 1) представляет собой последовательное соединение следующих элементов:

- неподвижного звена (земли);
- шарнирного соединения (задающего 1-му звену одну степень свободы поворот вокруг оси z);
- первого звена (звено представляется как абсолютное твердое тело);
- шарнирного соединения между 1-ым и 2-ым звеньями (ограничивает степени свободы 2-го звена, оставляя также только поворот в плоскости xy);
- второго звена.

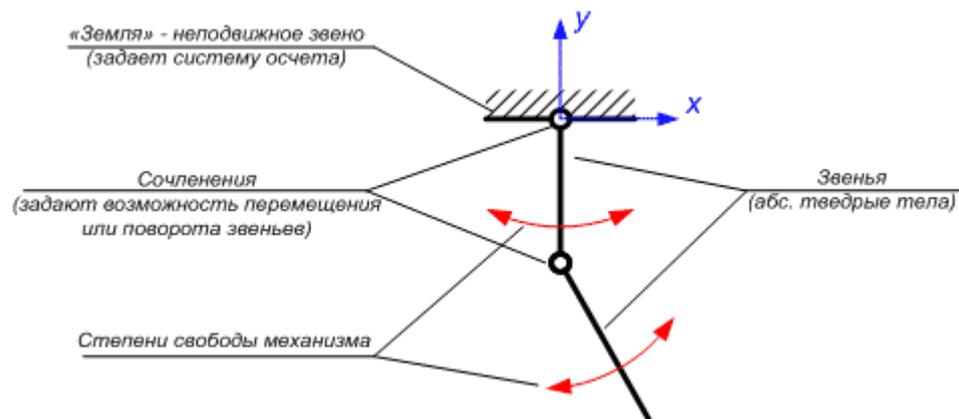
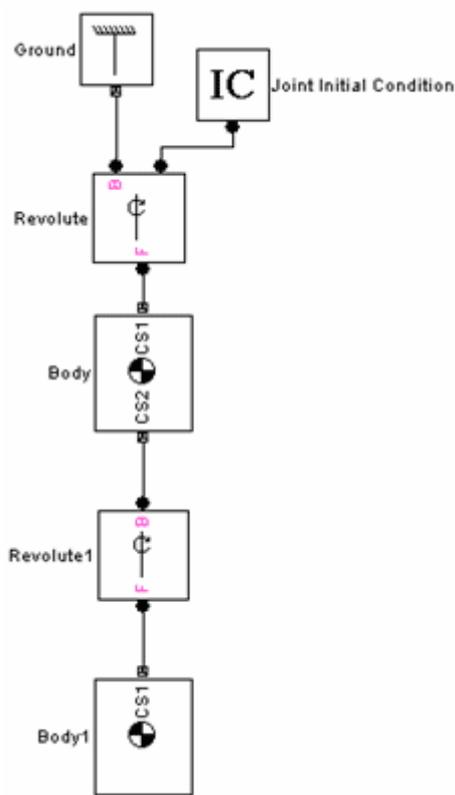


Рис. Модель двухзвенного физического маятника

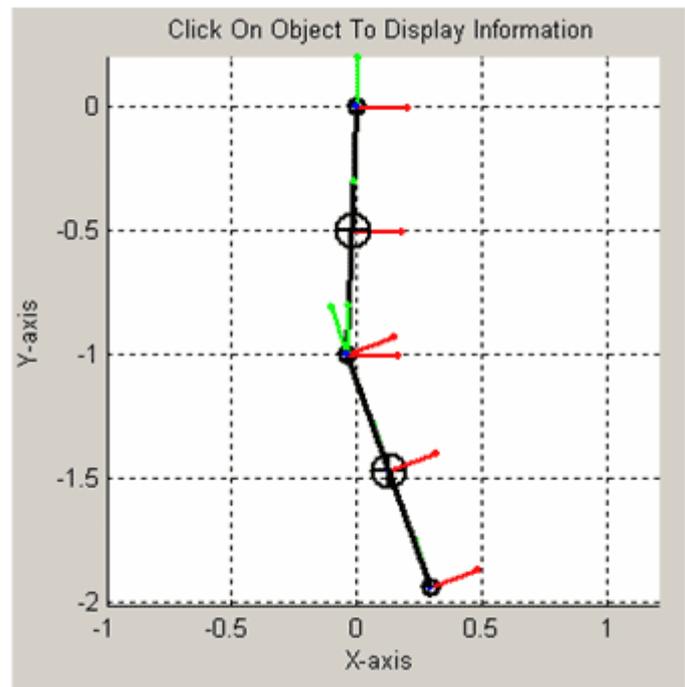
Simulink-модель такого механизма строится в аналогичной последовательности (см. рис. 2). Исходным элементом модели является звено Ground - земля. К нему присоединен элемент - Revolute (т.е. сопряжение, позволяющее следующему звену лишь поворачиваться вокруг указанной оси - z). Далее следует непосредственно звено физического маятника Body. В качестве параметров этого звена необходимо указать массу тела, моменты инерции относительно главных центральных осей симметрии, а также координаты верхнего, нижнего конца звена и его центра масс. При этом координаты можно задавать как в глобальной системе координат (ГСК), так и в локальной системе координат (ЛСК) звена.

Аналогично, к первому звену посредством шарнирного соединения Revolute 1 присоединяется второе звено Body 1.

Чтобы звенья спроектированного механизма начали движение необходимо либо добавить вынуждающую силу, либо задать начальные условия (например, начальное отклонение или сообщить начальную скорость). Для реализации последних используется блок InitialCondition.



а)



б)

Рис. Simulink-модель двухзвенного физического маятника (а) и моделью имитации движения (б)

На модели имитации отображаются звенья, колеблющиеся по законам классической механики (физики твердого тела). Там же отображаются ло-

кальные системы координат (ЛСК) звеньев. Вопрос о выборе той или иной системы координат (СК) является очень важным. Правильный выбор СК значительно облегчает моделирование механизма и интерпретацию результатов.

При моделировании данного механизма использовались следующие СК (рис. 3).

Неподвижная глобальная система координат ГСК Global находится в точке сопряжения неподвижного звена с верхним звеном (коленом маятника)². Задавать координаты точек верхнего звена маятника можно различными способами, в том числе, просто перечислив их значения в ГСК. Однако это не всегда удобно.

Верхний конец первого звена сопрягается с неподвижным звеном, и поэтому его координаты совпадают с началом ГСК. Его координаты действительно легко задать как Global $[0, 0, 0]$. Пусть звено имеет длину L и симметрию относительно ГЦОИ. Положение центра масс (ЦМ) звена удобно задавать уже не в ГСК а, в только что созданной ЛСК, где началом координат является верхний конец звена, т.е. в ЛСК CS1. Тогда координаты ЦМ можно задать как CS1 $[0, -L/2, 0]$. Аналогично нижний конец звена можно задать в ЛСК CS1 $[0, -L, 0]$.

Несмотря на то, что начало ЛСК CS1 совпадает с началом ГСК Global, следует иметь в виду что ЛСК CS1 принадлежит верхнему звену, а значит, может поворачиваться относительно точки Global $[0, 0, 0]$. Глобальная же система координат ГСК Global всегда неподвижна. Ее начало может и не совпадать с точкой сопряжения неподвижного звена (тем более, когда неподвижных звеньев в механизме несколько).

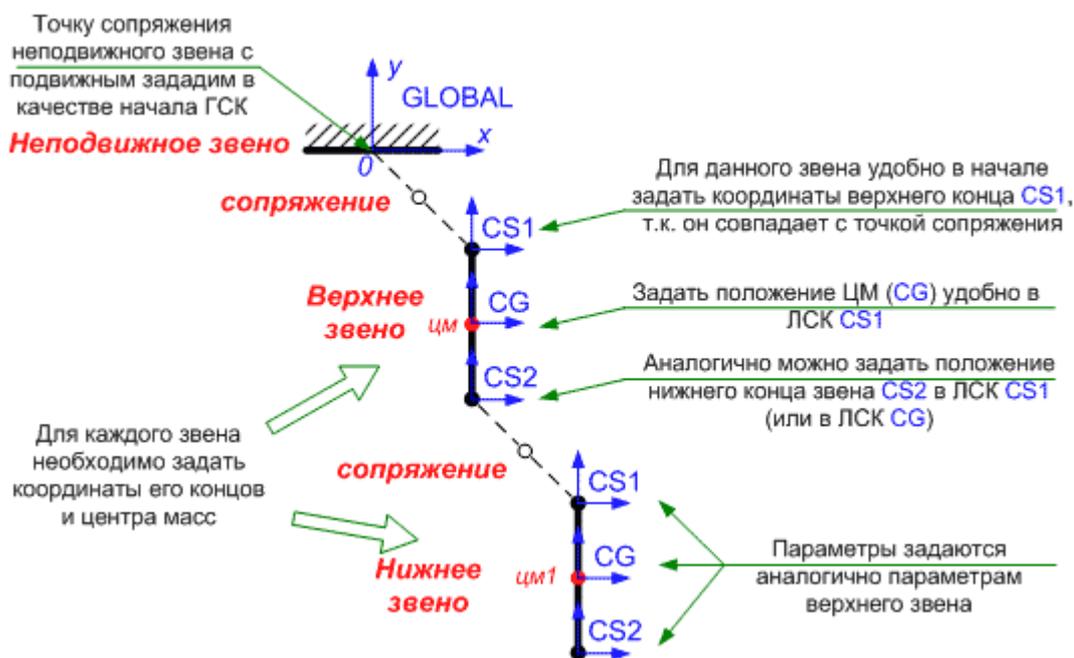


Рис. Системы координат двухзвенного физического маятника

Помимо визуального наблюдения за свободными (при задании начальных условиях) или вынужденными (при наложении внешней силы) можно анализировать законы движения любой точки механизма. Для этого необходимо при задании координат звеньев указать координаты интересующей точки и к выходу соответствующего Simulink-блока подключить блок-датчик (Sensor).

Датчики могут регистрировать как угловые колебания, так и линейные, причем как перемещение, так и скорость и ускорение. Выход с датчика обычно выводят на блок осциллографа Scope (см. рис. 4).

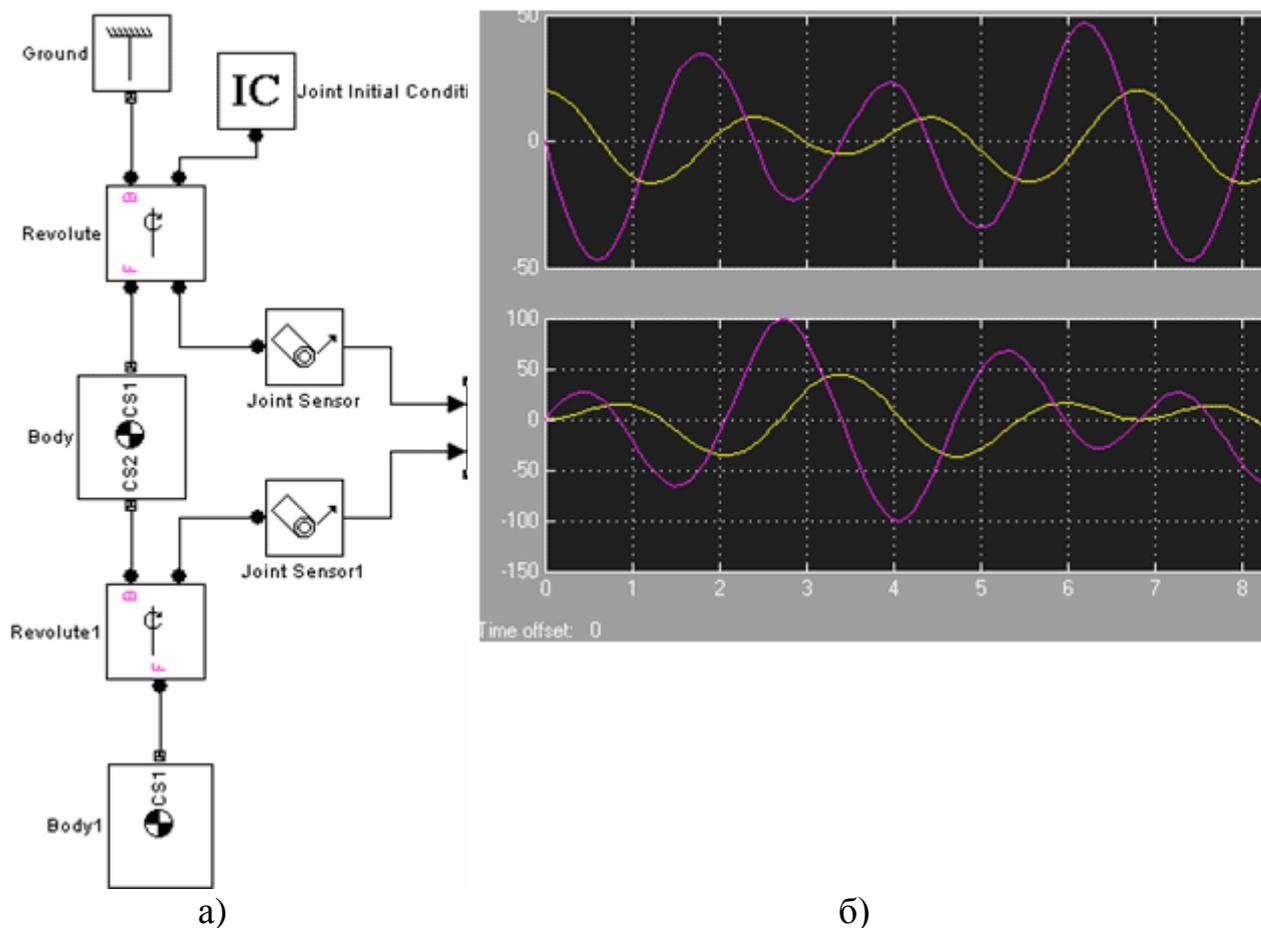


Рис. Модель двухзвенного физического маятника (а) для исследования законов движения его звеньев (б)

Порядок выполнения работы:

1. Ознакомление с пакетом SimMechanics.
 - 1.1. Запустите MatLab и Simulink.

1.2. Откройте файл Phys_pend2.mdl. Проанализируйте приведенную модель двухзвенного физического маятника. Откройте функциональные блоки и посмотрите на способы задания их параметров.

1.3. В меню Simulation выберите пункт Mechanical Environment... . Здесь можно задать параметры механического моделирования (значения ускорения свободного падения, точности расчетов, способ отображения работы механизма и пр.). Внесите изменения только во вкладку Visualisation (см. рис. 5).

1.4. Запустите модель, при этом наблюдая за колебаниями маятника.

1.5. Запустите модель с различными начальными условиями.

2. Моделирование кривошипно-шатунного механизма.

В данной лабораторной работе предлагается создать кинематическую модель кривошипно-шатунного механизма (рис. 6) и исследовать закон движения поршня.

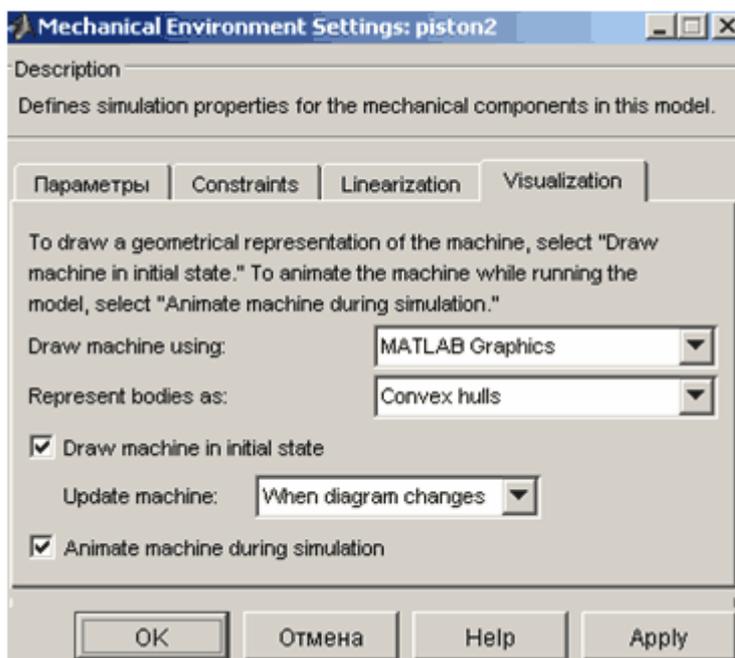


Рис. Задание параметров визуализации механизма

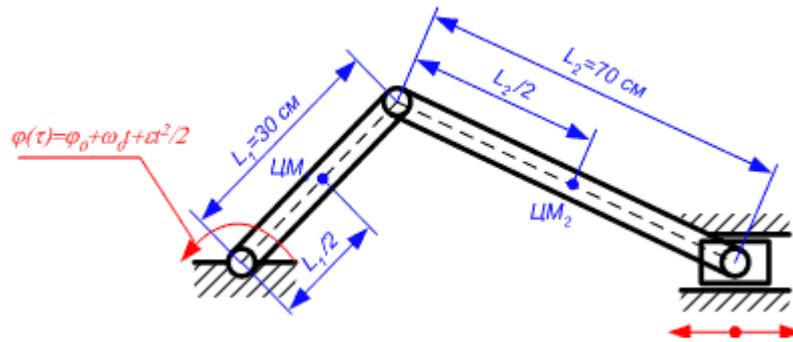


Рис. Кривошипно-шатунный механизм

Параметры кривошипа:

Масса - 1 кг;

Моменты инерции относительно ГЦОИ - $[I_x, I_y, I_z] = [50, 50, 300]$ г/см².

Параметры шатуна:

Масса - 1.5 кг;

Моменты инерции относительно ГЦОИ - $[I_x, I_y, I_z] = [80, 80, 500]$ г/см²

2.1 Создайте новую Simulink-модель.

2.2 Проектирование большинства механизмов начинается с неподвижного звена - земли. Для этого воспользуйтесь пакетом расширения SimMechanics - Bodies - Ground. В параметрах блока укажите координаты $[0, 0, 0]$, это будет означать, что координаты блока совпадают с началом ГСК.

2.3 Так как кривошип может совершать только вращение вокруг оси OZ, то возьмем в качестве блока сопряжения блок Joints - Revolute. В параметрах блока необходимо указать $[0, 0, 1]$, что будет означать возможность вращения вокруг оси OZ.

2.4 Добавьте кривошипное звено в модель механизма, для этого воспользуйтесь блоком Bodies - Body. Внесите все необходимые данные о звене в параметры блока (см. рис. 7).

Ввод значений моментов инерции необходимо осуществлять в матричном виде $[3 \times 3]$ (см. рис. 7).

2.5. К уже спроектированной части механизма можно добавить привод (элемент, заставляющий механизм двигаться по определенному закону). Добавим вращение к сопряжению Revolute. Для этого в параметрах блока Revolute необходимо указать вход/выход для привода/датчика. При этом блок Revolute будет иметь еще один вход/выход. К этому входу нужно подключить блок Sensors and Actuators - Joint Actuator.

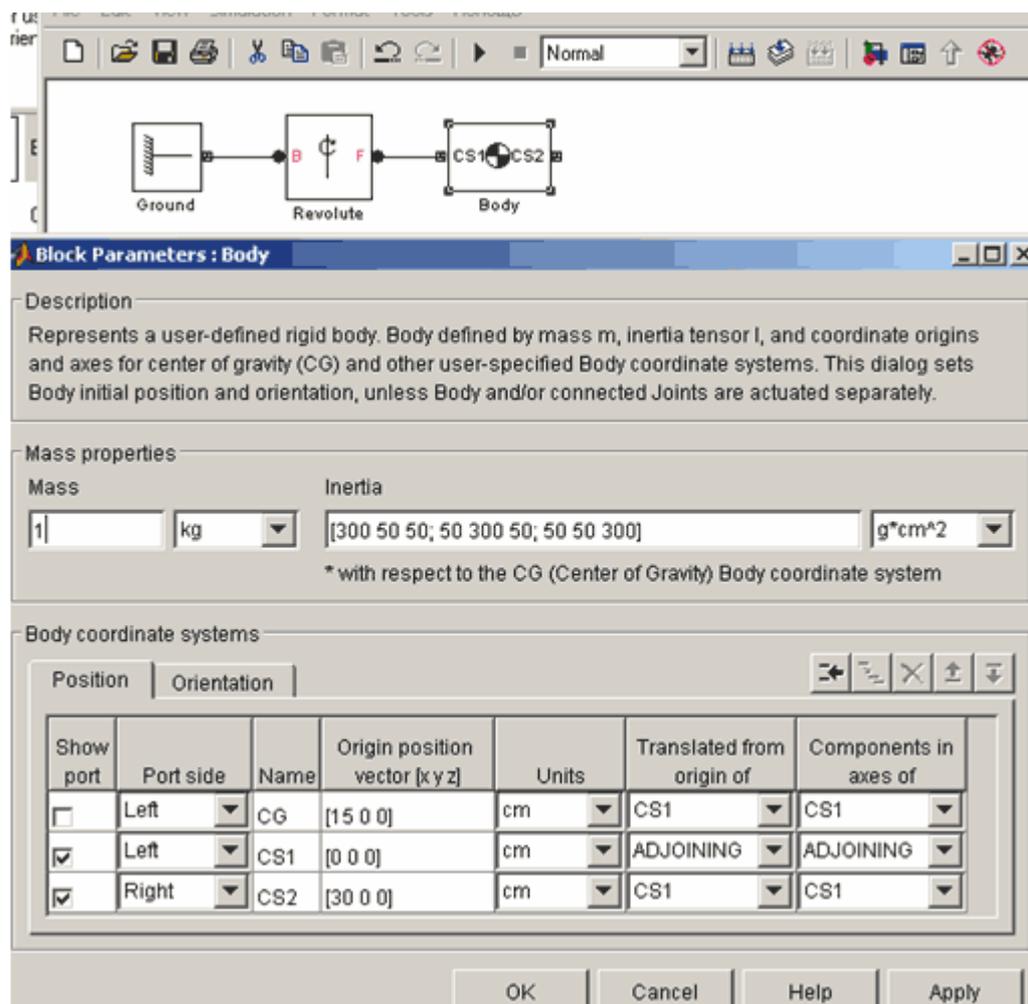


Рис. Параметры блока Body - при вводе данных о кривошипном звене

2.6. Для задания закона движения формируемого в звене JointActuator необходимо задать три параметра - вектора $\varphi(t)$, $\omega(t)$, $\varepsilon(t)$. Поэтому ко входу элемента JointActuator необходимо подать необходимые вектора. Так как величины φ_0 , ω_0 , ε - величины известные, то легко определить зависимости $\varphi(t)$, $\omega(t)$, $\varepsilon(t)$:

$$\varphi(t) = \varphi_0 + \omega_0 t + \frac{\varepsilon t^2}{2}; \quad \omega(t) = \omega_0 + \varepsilon t; \quad \varepsilon(t) = \varepsilon = const$$

Реализуйте приведенные зависимости в виде отдельной подсистемы, например, как показано на рис. 8.

2.7. Теперь можно запустить сформированную часть модели. Для этого необходимо выбрать в меню Simulation - MechanicalEnvironment ... и далее указать параметры, показанные на рис. 5 и нажать кнопку ОК.

2.8 Самостоятельно доработайте модель кривошипно-шатунного механизма (см. рис. 9).

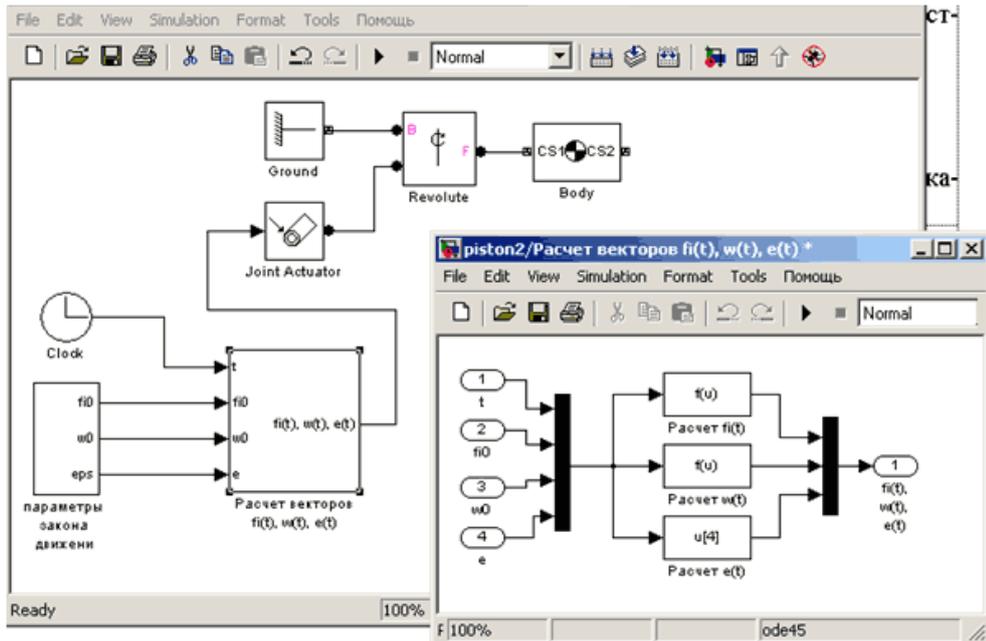


Рис. Добавление к модели привода и моделирование закона движения

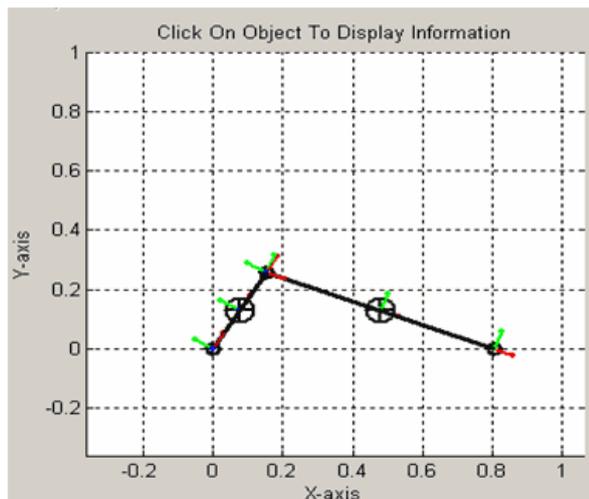
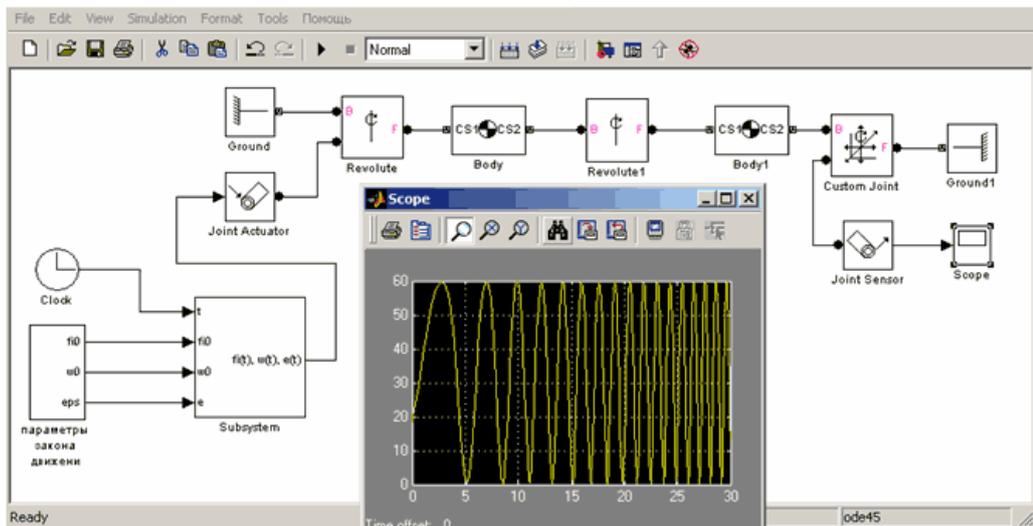


Рис. 9 Окончательная модель кривошипно-шатунного механизма

1. Для создания сопряжения между шатуном и неподвижным звеном (направляющей) воспользуйтесь блоком JointsCustomJoint, и укажите степени свободы:

Name - Primitive	Axis of action [x y z]	Reference csys
R1 - Revolute	[0 0 1]	Base
P1 - Prismatic	[1 0 0]	Follower

2. При указании координат второго неподвижного звена можно воспользоваться следующими значениями $[L_1+L_2; 0; 0]$

3. Для исследования закона движения поршня целесообразно добавить датчик перемещения SensorsandActuators - JointSensor, а его выход подключить к осциллографу Scope (см. рис. 9).

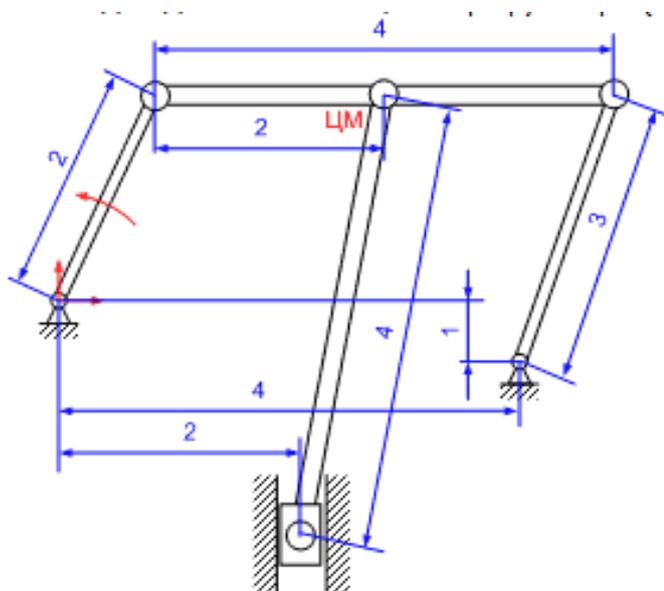


Рис. 10 - Механизм с одной степенью свободы

Создайте имитационную модель механизма, представленного на рис. 10. Размеры, указанные на эскизе механизма (рис. 10) условные. Определите, при какой длине правого кривошипа, ведущее левое звено не сможет совершить полный оборот. Вариант модели механизма (без датчиков) представлен на рис. 11.

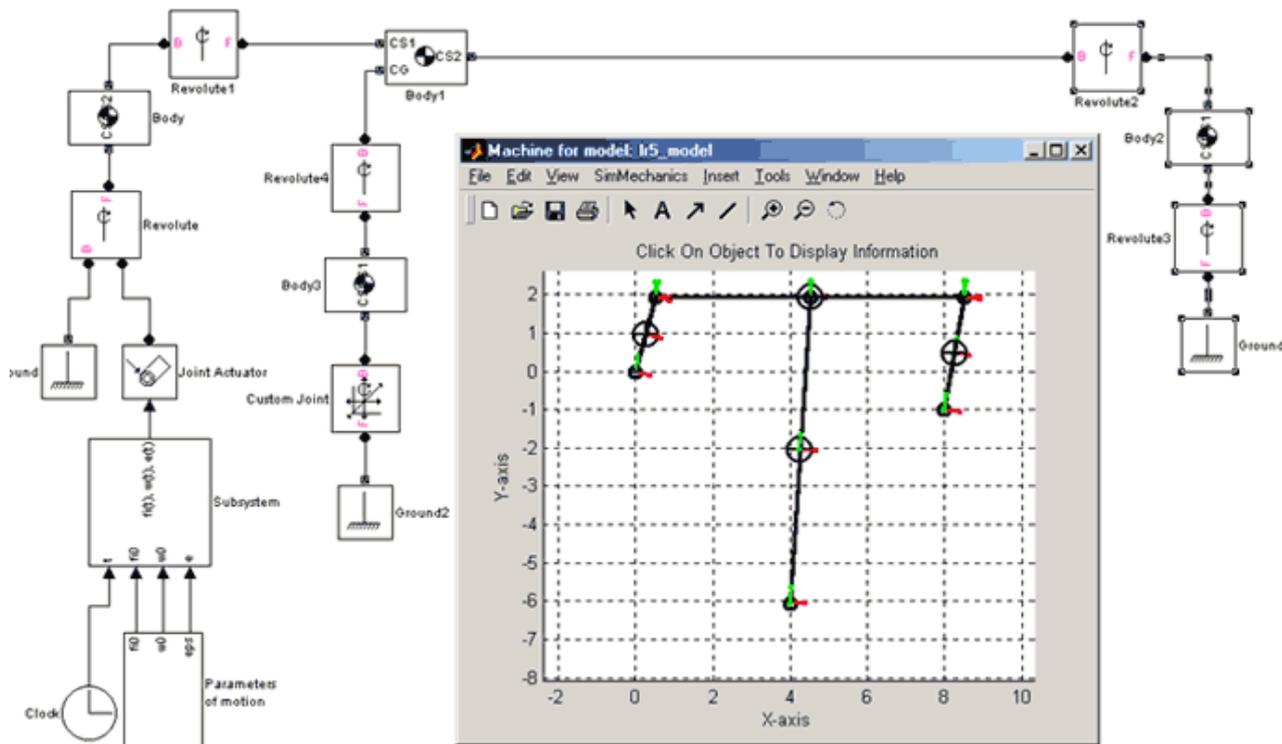


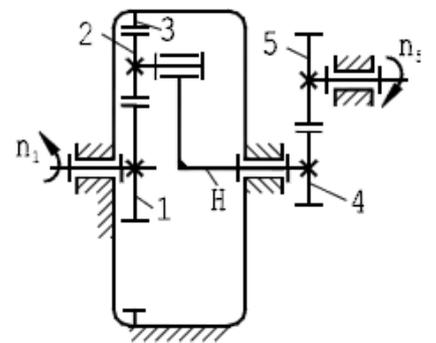
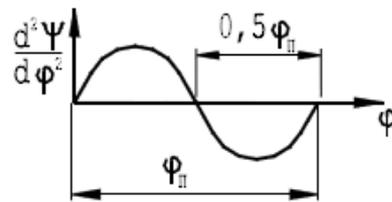
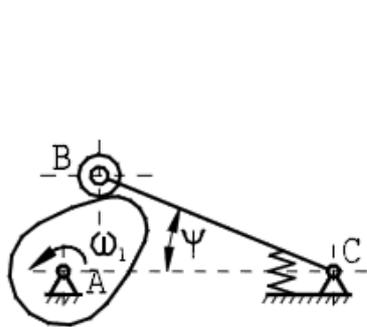
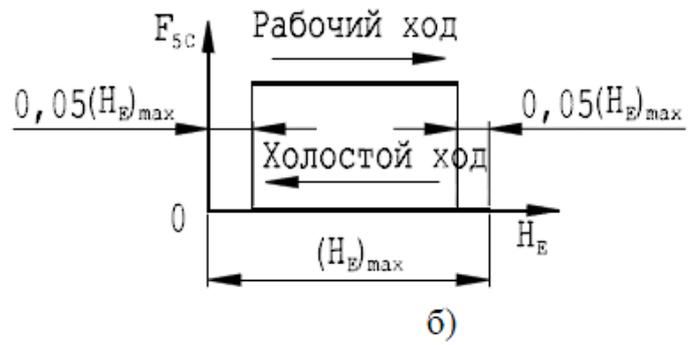
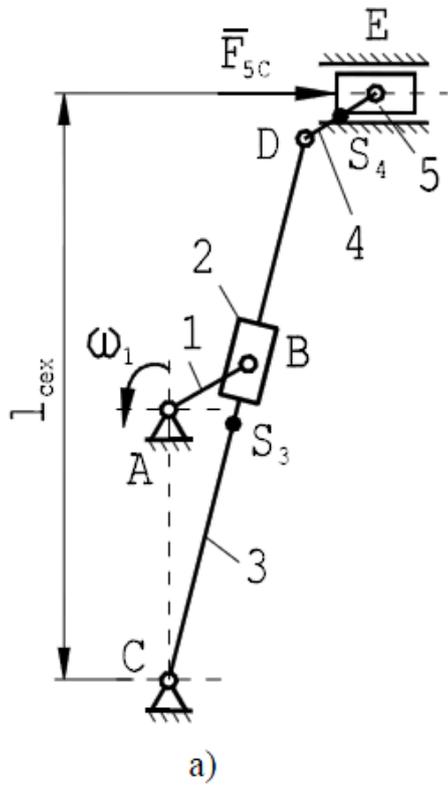
Рис. 11 - Вариант построения имитационной модели механизма, представленного на рис. 10

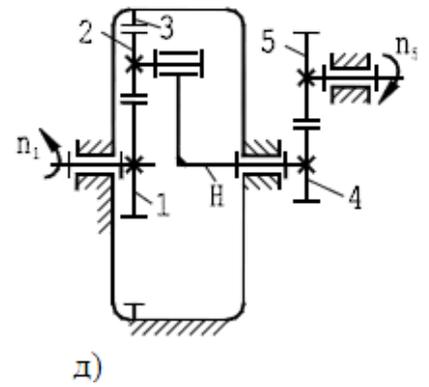
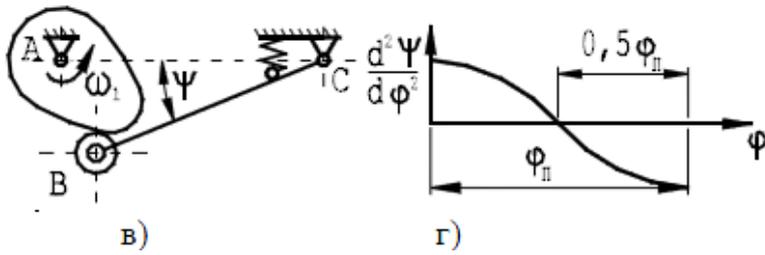
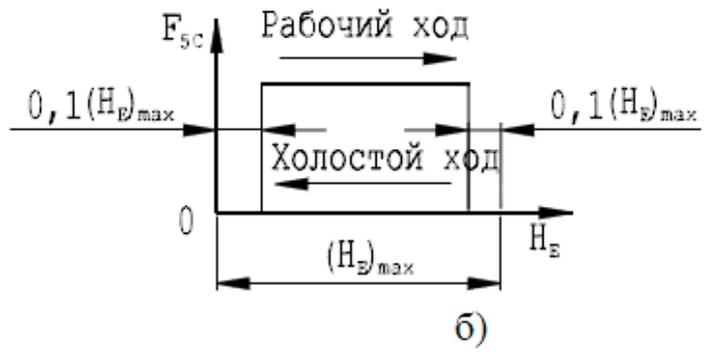
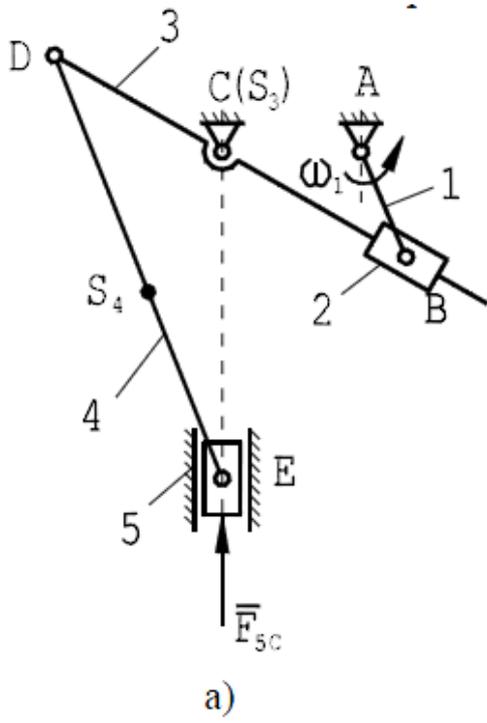
4. ВОПРОСЫ ДЛЯ ПОДГОТОВКИ К ЗАЩИТЕ РАБОТ

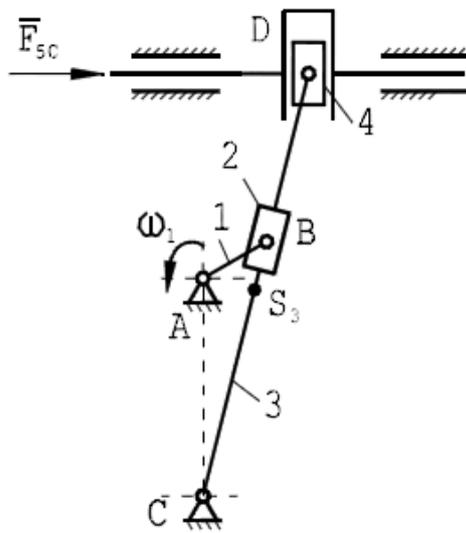
1. Что называют манипулятором?
2. В каких системах координат могут работать существующие типы манипуляторов?
3. В чем преимущества манипуляторов, работающих в сферической системе координат?
4. В какой системе координат работает исследуемый манипулятор?
5. Что представляет собой библиотека пакета SimMechanics?
6. Особенности имитационного моделирования кинематических механизмов в Simulink.
7. Глобальные и локальные системы координат механизмов.
8. Задание законов движения звеньям механизмов и их исследование.

ВАРИАНТЫ ЗАДАНИЙ

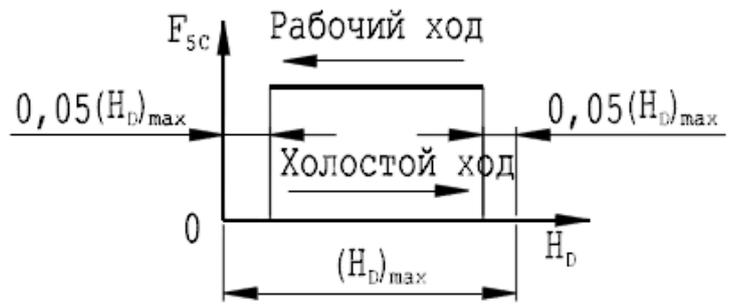
№ 1



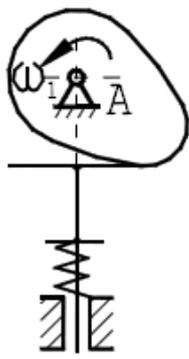




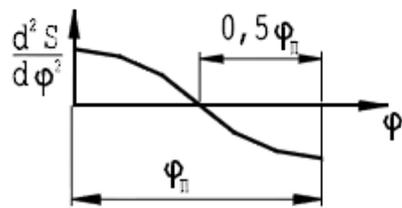
а)



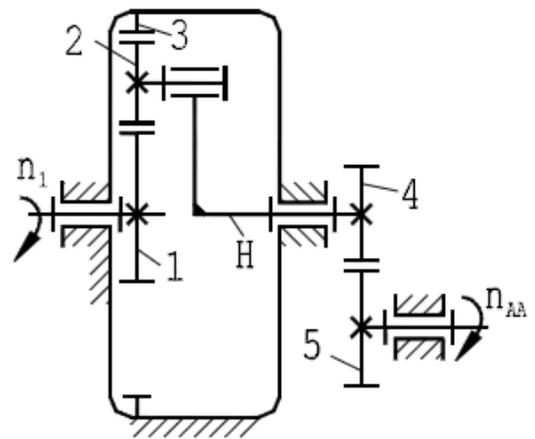
б)



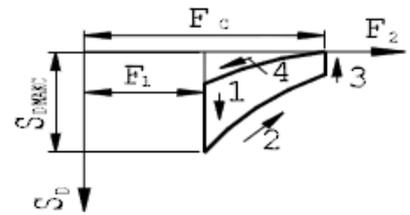
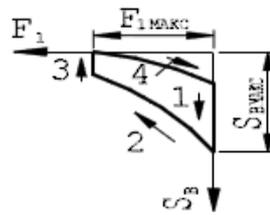
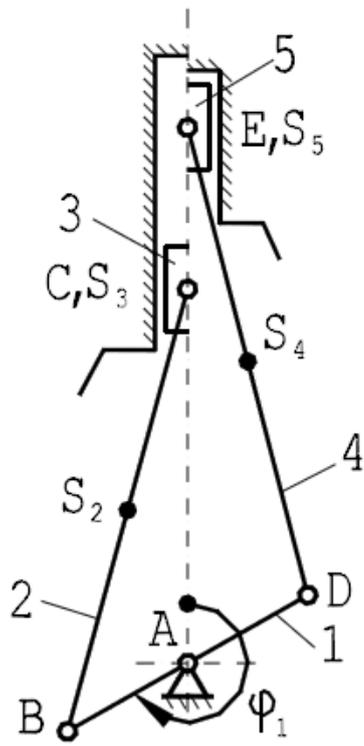
в)



г)

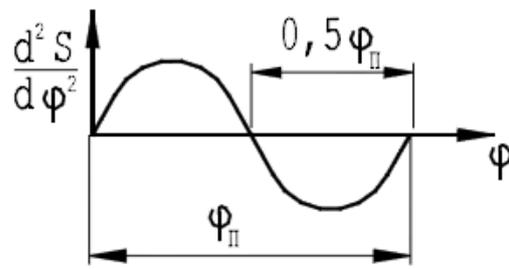
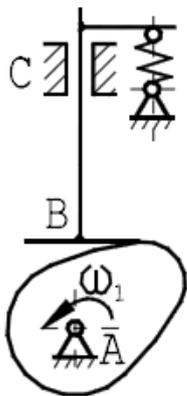


д)



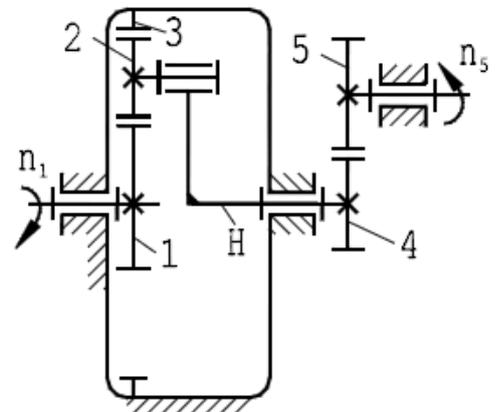
a)

б)



в)

г)



д)

ИМИТАЦИОННЫЕ МОДЕЛИ МУЛЬТИКООРДИНАТНЫХ СИСТЕМ ДВИЖЕНИЯ

Методические указания к выполнению практических занятий и самостоятельной работы по дисциплинам: "Геометрические основы планирования движения микроэлектромехатронных систем (МЭМС)" для магистрантов по направлению 222000.68 «Инноватика» по магистерской программе «Управление инновациями в мехатронике и робототехнике" и "Методы и инструментарий геометрического обеспечения функционирования электромехатронных систем " для магистрантов по направлению 221000.68 «Мехатроника и робототехника» по магистерской программе «Проектирование и исследование мультикоординатных электромехатронных систем движения»

Составители:

Щербинин Сергей Васильевич

Подписано к печати

Формат 60x84/16. Бумага офсетная

Печать RISO. Усл.печ.л. Уч.-изд.л.

Тираж 50 экз. Заказ . Бесплатно