|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1.Введение**  В данной работе мною будут рассмотрены программы моделирования работы робототехнических систем на основании чего было решено создать набор классов для создания их аналогов: класс TZweno отвечающий за графическое отображение звеньев и производный от него класс TZmove отвечающий за ввод и хранение начальных параметров движения, вывод параметров зависимых от введенных и алгоритм движения для звеньев. Для каждого класса была составлена своя математическая модель: для TZweno - рассматривающая закономерности изменения координат звена на при его движения на плоскости, для TZmove – рассматривающая закономерности между зависимыми параметрами движения как для прямолинейного движения так и для вращательного. Все законы как построения так и движения были выведены в общем виде для возможности использования их при создании более сложных законов движения. Для лучшего понимания принципов работы разработанных классов, для каждого из них была составлена блок-схема. Кроме того была разработана спецификация для создания программы моделирования работы с манипулятором на основе созданных классов и ее интерфейс. | | | | | | | | |
|  |  |  |  | | ДР-2068752-221000-КБ-5-19-15 | | | |
| Должность. | Ф И О | Подпись | | Дата |
| Студент | Чуриков А.А. |  | |  | Разработка классов для информационной системы моделирования манипуляторов. | Лит. | Лист | Листов |
| Руководитель | Пименов А.В. |  | |  | У | 1 | 1 |
| Консультант |  |  | |  |  | | |
| Нормоконтр. |  |  | |  |  | | |
| Зав.кафедрой | Жуков Д.О. |  | |  |

**1.1 Обзор существующих программ**

**ADAMS**

Данная программная система предназначена для виртуального моделирования и анализа работы сложных машин и механизмов в реальных условиях, тем самым позволяя пользователю протестировать и сравнить различные реализации своего изделия с учетом их поведения в реальных условиях минуя необходимость сборки опытного образца, что в свою очередь уменьшает как стоимость разработки, так и время на разработку самого продукта.

Программы семейства ADAMS используются для разработки и совершенствования конструкций фактически всего, что движется от простых механических и электромеханических устройств до автомобилей и самолетов, железнодорожной техники, космических аппаратов и т.д.

Интерфейс ADAMS позволяет в одном пакете быстро разработать расчётную модель изделия, строя ее на базе геометрических примитивов, создаваемых непосредственно в препроцессоре или на базе геометрических моделей компонентов изделия, импортируемых из CAD-систем, задать связи компонентов модели (упругие, деформируемые, кинематические и многие другие.), приложить нагрузки, запустить расчет и проанализировать его результаты. Интерфейс пользовательского пакета включает эффективные средства анализа результатов, которые позволяют в сжатые сроки наметить пути к совершенствованию расчётной модели и добиться максимальной близости её свойств к характеристикам реального динамического процесса, изделия-прототипа или результатам испытаний физического образца разрабатываемой машины.

Основные модули программного пакета

1. **Препроцессор** ADAMS **\View**

Обеспечивает импорт геометрических примитивов из многих CAD систем, а также создание твердотельных моделей непосредственно в самой среде моделирования. Данный модуль также способен автоматизировать многие повторяющиеся задачи, экономя бесценное время. Он также позволяет создавать шаблоны и инструменты для настройки ваших решений.

1. **Основной решатель** ADAMS **\Solver**

Его основой являются системы дифференциальных уравнений, описывающих динамику исследуемого объекта. Разработчики постоянно прилагают усилия по повышению эффективности математической базы программного пакета, благодаря чему использование решателей ADAMS гарантирует получение результатов при минимальных усилиях со стороны пользователя.

1. **Постпроцессор** ADAMS **\Postprocessor**

Модуль анализа полученных результатов обеспечивает анимацию полученных результатов, построение графиков, вывод результатов в различном виде.

Вспомогательные модули ADAMS

Дополнительные модули, доступные в ADAMS позволяют пользователям интегрировать механические компоненты, пневматику, гидравлику, электронику и системы контроля технологии для создания и тестирования виртуальных прототипов, которые точно учитывают взаимодействие между этими подсистемами.

* ADAMS **\Flex**

Позволяет включать гибкие компоненты в вашу полноценную симуляцию. Добавление гибкости обеспечивает лучшее понимание поведения и работы проектов, позволяя делать более высококачественные продукты быстрее. Кроме того, данный модуль позволяет использование существующею линию моделей МКЭ, а также он тесно интегрирован с Nastran и Patran.

* ADAMS **\Durability**

Расширяет возможности Flex, он возвращает жесткость на гибком теле.

* ADAMS **\Vibration**

Позволяет изучать созданную вибрацию на ваших моделях Адамса, используя анализ области модели.

* ADAMS **\Engine**

Модуль для статического, кинематического, динамического анализа двигателей (поршневых). Моделирование всей механической схемы производится по подсистемам (идеология повторяет логику работы Car, Rail), которые в дальнейшем автоматически собираются программой в общую сборку. Каждый тип подсистемы (кулачки, пружины, поршни, клапаны и пр.) может быть создан в виде шаблонов и храниться в библиотеках. Осуществляется контроль соударений (заклиниваний), амплитуд движения, траекторий, предельных положений и прочих характеристик. В комплекте с Flex позволяет учитывать деформации шатунов, поршней и др. деталей при динамическом расчете.

* ADAMS **\Android**

Позволяет моделировать человеческое тело. Модуль создает реалистичную динамическую модель для исследования взаимодействия тела и изделия, позволяет получить нагрузки на суставы и части тела. Кроме перечисленных возможностей в нем имеются уже готовые модели мужчины, женщины и ребенка с возможностью модификации их характеристик пользователем (конкретные размеры, законы движения, воздействующие силы и перемещения).

* ADAMS **\Animation**

Осуществляет высококачественное анимационное воспроизведение результатов анализа. В большинстве случаев анимация производится в реальном масштабе времени, что позволяет безошибочно оценить результаты расчета. Специальные функции модуля: трассировка расстояния между заданными точками, установка множественных источников света, текстурирование поверхностей, задание прозрачности составных частей позволяюшее получать анимацию высочайшего качества.

* ADAMS **\Linear**

Линеаризует нелинейные уравнения для получения собственных частот и форм механической системы. Результирующая система уравнений дает возможность: установить взаимосвязь между большими перемещениями конструкции и малыми колебательными перемещениями ее частей, осуществить контроль построения системы, облегчить проектирование систем управления, облегчить построение систем с гибкими звеньями.

Наряду с развитием универсальных возможностей, разработчиками ADAMS созданы проблемно-ориентированные модули, обеспечивающие точное и быстрое моделирование таких сложнейших объектов как:

1. Зубчатые, Ремённые и Цепные передачи (пакет **Adams Machinery**, включающий модули Adams Gear, Adams Belt и Adams Chain);
2. Автомобили (пакет **Adams Car**, включающий модули Adams Car Plugin, Adams 3D Road, Adams Tire Handling, Adams Car Suspension, Adams SmartDriver, Adams Vehicle Solver, Adams Chassis, Adams Car Ride, Adams Driveline, Adams Tire Ftire);
3. Гусеничная техника (пакет Adams Tracked Vehicle – ATV).

Также важно отметить что программный пакет ADAMS предоставляет импорт геометрических моделей из CAD систем в форматах: Parasolid, IGES, STEP, DXF, DWG, VDAFS.

ADAMS помогает инженерам изучить динамику подвижных частей, как грузы и силы распределяются по всему механических систем. Механические, электрические и другие подсистемы, проверяются в отношении их конкретных требований в рамках систем технологического процесса, что позволяет избежать натурного моделирования, испытания реальных образцов и существенно сокращает как время, так и стоимость разработок. Кроме того, данная программа позволяет оценить и управлять сложными взаимодействиями между дисциплинами в том числе движением, структурой, приведением, и управлением, для лучшей оптимизации конструкции изделий в целях повышения производительности, безопасности и комфорта. Наряду с широкими возможностями анализа, данная программа оптимизирована для решения крупномасштабных проблем, пользуясь высокопроизводительными вычислительными средами. Силы, вычисленные ADAMS смоделированы с повышенной точностью путем предоставления более точной оценки того, как они изменяются в течение полного диапазона движения и операционных сред.

Плагин **Mechatronics** может быть использован для внедрения систем управления в механические модели. Он был разработан на основе функционала Control и содержит элементы моделирования и обмена информацией с системами управления. Например, с помощью Mechatronics в Car, можно быстро создавать сборки систем для транспортных средств, включая системы управления, а затем проанализировать их, чтобы понять их эффективность и поведение.

Основных возможности ADAMS

* Создавать компьютерную модель системы из жестких и деформируемых элементов, соединенных между собой различными связями и шарнирами.
* Создавать параметризованную модель на базе ядра твердотельного моделирования PARASOLID,а также обмениваться геометрическими моделями в форматах IGES, STEP, DXF, DWG, STL.
* Визуализировать модель конструкции мощными средствами трехмерной графики;
* Задавать вынужденные перемещения и движения элементов системы и прикладывать активные внешние силы и моменты;
* Проводить статический, модальный, динамический и кинематический анализ системы;
* Визуализировать движение системы и фиксировать (реагировать на) заданные события;
* Анализировать влияние вариаций конструктивных элементов на поведение системы;
* Оптимизировать изделие по заданным критериям;
* Получать результаты анализа в удобном для оценки и интерпретации виде (графики, таблицы, анимация);
* Производить двусторонний обмен информацией с программными комплексами автоматизированного проектирования, конечно-элементного анализа, анимации;
* Определять все параметры движения системы как из абсолютно жестких, так и из упругих звеньев;
* Учитывать деформируемость элементов механизмов за счет двустороннего обмена данными с ведущими пакетами конечно-элементного анализа и передавать данные о распределении нагрузок по времени;
* Применять деформируемые элементы - линейные и торсионные пружины/демпферы с линейными и нелинейными характеристиками, бушинги;
* Вычислять усилия в связях и реакции в опорах с полной историей изменения по времени, приходящие усилия на элементы управления;
* Определять взаимное перемещение составных частей, перемещения и углы поворота в шарнирах;
* Проводить статический и модульный анализ;
* Возможность настраивать комплекс под типовые задачи конкретного пользователя;
* Использование специализированных модулей, ориентированных на конкретные области техники;

Из перечисленного можно сделать вывод что данная программа является очень мощным редактором который позволяет разрабатывать расчётные модели исследуемых изделий, в максимальной степени учитывающие особенности их конструкции, включая высокую идентичность внешнего вида, что нередко облегчает как построение моделей, так и их отладку и анализ полученных результатов. Кроме того, собственный решатель, выполняющий расчёт параметров изделий, определяющих их работоспособность и точность (перемещения, скорости и ускорения компонентов изделия, действующие нагрузки, габариты пространства, необходимого для движущихся частей машины и т.п.), а также выполнять оптимизацию параметров изделия.

**MRDS**

**Microsoft Robotics Developer Studio** - Windows-ориентированная среда программирования для разработки приложений в области робототехники, симуляции и управления роботами.

Данную среду можно рассматривать в виде четырех взаимосвязанных компонентов:

1. **Visual Programming Language** (VPL)

Язык визуального программирования, созданный Microsoft специально для MRDS. Программы на VPL выглядят в виде диаграмм, в которых все элементы связаны между собой и каждый из них обладает своей функциональностью. Язык рассчитан на начинающих программистов, требует базовых знаний о переменных и принципах алгоритмизации.

1. **Concurrency and Coordination Runtime**

Библиотека для работы с асинхронными и параллельными потоками данных, опирающаяся на .NET Framework. С ее помощью упрощается создание кода для масштабирования и параллельного исполнения на многоядерных процессорах последнего поколения, что позволяет, например, роботам правильно реагировать на данные, одновременно поступающие от множества датчиков. Библиотека получила широкое распространение в сторонних проектах, существенно упрощая разработку многопоточных приложений.

1. **Decentralized Software Services**

Основанная на CCR облегченная среда для разработки распределенных приложений, предусматривающая управление различными сервисами, корректирующими поведение роботов в целом.

1. **Visual Simulation Environment**

Среда визуализации, имитирующая поведение роботов в трехмерном виртуальном мире. Позволяет экспериментировать с разными моделями, тестировать и отлаживать алгоритмы в том случае, если нет возможности использовать настоящего робота. Для создания реалистичности применяется технология NVIDIA PhysX.

Эти компоненты ориентированы на программистов различных уровней и включают в себя поддержку огромного количества аппаратного обеспечения.

Данный продукт совместим со следующими робототехническими платформами: Aldebaran Robotics, iRobot Create, Mindstorms NXT, CoroWare CoroBot, KUKA Robotics, Boe-Bot, Parallax Scribbler, Kondo KHR-1 и многими другими.

Для лучшего понимания возможностей и методов работы в данной программе рассмотрим доступные в ней инструменты работы:

1. Инструменты, позволяющие разрабатывать MRDS-приложения, включают в себя графическую среду (Microsoft Visual Programming Language: VPL);
2. Утилиты командной строки, позволяющие взаимодействовать с проектами Visual Studio (VS Express) на языках C#, C++ или VB.NET, а также осуществляет поддержку скриптовых языков, как например, IronPython; и инструменты симуляции 3D.
3. Visual Programming Language — графическая среда разработки, использующая каталог сервисов и действий. Они могут быть связаны графическим способом: сервис или действие представляются в виде блока, у которого есть ввод и вывод, и его нужно лишь перетащить из каталога на диаграмму. Связывание можно сделать при помощи мыши, это позволяет вам определить, являются ли сигналы синхронными или нет, позволяет вам выполнять операции, связанные с передачей значений и пр. VPL также позволяет вам генерировать код новых “макро”-сервисов на основе диаграмм, созданных пользователями. В нем реализована простая настройка сервисов для различных аппаратных элементов.
4. Среда симуляции 3D в MRDS позволяет вам симулировать поведение роботов в виртуальном мире, используя технологию NVIDIA PhysX, что позволяет использовать продвинутую физическую модель.
5. Множество примеров и руководств, доступных для различных инструментов, ускоряющих понимание и усваивание MRDS. В комплект добавлено несколько приложений, среди них, например, упоминавшиеся выше Maze Simulator или Soccer Simulation, разработанные корпорацией Microsoft.
6. Также в стандартном издании MRDS присутствуют 3 небольшие симуляционные среды, которые являются ограниченными версиями более крупных обобщенных симуляционных сред, разработанных компанией SimplySim:

* Комната
* Открытая местность
* Город

Дополнительные возможности

Также Microsoft Robotics Developer Studio поддерживает широкий набор робототехнических платформ, а также дает возможность работать непосредственно на платформе (если она имеет встроенный ПК с операционной системой Windows) или если управление роботом осуществляется из Windows ПК через каналы связи, такие как Wi-Fi или Bluetooth.

Мощное визуализация окружающей среды Моделирование (VSE) обеспечивает высококачественное моделирования окружающей среды на основе технологии NVIDIA PhysX для запуска игры качества 3D моделирования с реальными физическими взаимодействиями.

Для помощи начинающим разработчикам студия Robotics Developer содержит обширную документацию и большой набор образцов и обучающих программ, которые иллюстрируют, как писать приложения, начиная от простого "Hello Робот" до сложных приложений, которые одновременно работают на нескольких роботов.

**UM**

Программный комплекс **UM** (Универсальный механизм) предназначен для моделирования динамики и кинематики плоских и пространственных механических систем. Программа ориентирована на инженеров-практиков, студентов и преподавателей вузов, всех, кто сталкивается с проблемами исследования динамического поведения машин и механизмов. Механизмы описываются как системы твердых тел, шарниров и силовых элементов. Поддерживается непосредственная анимация движения вашей модели в процессе расчета. Для анализа доступны практически все необходимые величины: координаты, скорости, ускорения, силы реакций в шарнирах, усилия в пружинах и т.д.

Развитый постпроцессор дает возможность использовать: линейный анализ, статистический анализ, многовариантные расчеты и оптимизацию, экспорт результатов. Это эффективный инструмент для моделирования динамики различных машин и механизмов: космических конструкций, роботов и манипуляторов, железнодорожных экипажей, автомобилей, кабелей и т.д. Решение прямой и обратной задач кинематики и динамики.

UM включает в себя ядро и ряд дополнительных модулей, расширяющих функциональность ядра: модуль импорта из CAD программ (UM CAD Interfaces), модуль моделирования упругих тел (UM FEM), автомобильный модуль (UM Automotive), железнодорожный модуль (UM Loco), модуль моделирования гусеничных машин (UM Tracked Vehicle), модуль многовариантных расчетов и оптимизации UM Experiments, модуль связи с Matlab/Simulink UM Control и другие.

Основные возможности и характеристики UM

UM позволяет создавать тела произвольной формы и производить автоматический расчет инерционных параметров и положения центра масс в них. Данный программный комплекс поддерживает следующие виды шарниров и связей: вращательные, поступательные, цилиндрические, карданные, обобщенные, а также связь в виде невесомого стержня. UM позволяет использовать: общие, биполярные, контактные, специальные и другие описанные пользователем силы.

Графические элементы

Полиэдр, эллипс, параллелепипед, винтовая линия эллипсоид, конус, параметрический, профильный

Импортированные модели из ASC и 3DS файлов, КОМПАС, SolidWorks, Autodesk Inventor, Unigraphics NX и Pro/ENGINEER

Измеряемые величины

* Угловые и линейные координаты, скорости, ускорения
* Активные силы и моменты
* Силы реакции
* Другие величины, определенные пользователем

Управление моделированием

* Запуск, пауза, продолжение, остановка моделирования в любое время
* Контроль точности решения
* Программирование модулей пользователя

Поддержка разработки модулей пользователя с использованием компиляторов:

* MS Visual C++
* Borland C++ Builder
* Borland Delphi
* любые другие языки программирования, поддерживающие создание динамически загружаемых библиотек (DLL)

Даная программа поддерживает импорт из:

* 3DS файлов, ASC файлов,
* Детали и сборки из КОМПАС, SolidWorks, Autodesk Inventor, Unigraphics NX и Pro/ENGINEER (графические образы, инерционные параметры тел, сопряжения)
* Модели из Matlab/Simulink

Экспорт в:

* BMP файлы
* AVI файлы
* Анимированные GIF файлы
* Измеряемые величины в форматированный текстовый файл
* Таблицы и графики в MS Word и Excel

Наиболее важные модули для разработки моделей в данной среде.

1. **UM Subsystems**

Метод подсистем является основой моделирования объектов с особо большим числом степеней свободы, а также создания баз данных типовых элементов конструкций при моделировании технических систем. При использовании данного метода объект может быть представлен деревом подсистем, соединенных друг с другом посредством силовых элементов и связей. В качестве отдельной подсистемы выступает любой объект, предварительно созданный пользователем, который, в свою очередь, может содержать произвольное дерево подсистем.

1. **UM CAD Interfaces**

Современный подход к проектированию машин и механизмов предполагает создание модели в одной из программ твердотельного моделирования (CAD программ) с параллельным расчетом динамики системы и прочности элементов конструкции. Для упрощения импортирования таких моделей в программный комплекс "Универсальный механизм" мы разработали средства прямого импорта данных из ряда CAD программ. Эти программные средства объединены в отдельный модуль UM CAD Interfaces. В настоящей версии программы реализована поддержка импорта из следующих CAD программ: КОМПАС 3D, Autodesk Inventor, Unigraphics NX, SolidWorks и конвертация данных из файлов в форматах STEP, IGES, Parasolid, SAT, STL.

1. **UM Experiments**

Как правило, в инженерной практике при исследовании механических систем часто требуется проведение серийных численных экспериментов: для анализа поведения системы на различных режимах работы, проверки чувствительности по параметрам, подбора наилучших значений параметров. Интегрированный модуль сканирования и оптимизации включает в себя набор инструментов (сканирование, оптимизация, аппроксимация), предназначенных для расширенного анализа динамических свойств моделей.

1.2 Постановка задачи

На основе разработанных мною классов можно создавать аналоги многофункциональным системам, описанным мною выше, которые позволяет создавать модели кинематических схем манипуляторов для визуализации работы с данными схемами, тем самым позволяя пользователю рассмотреть принцип работы, заданной им схемы руки манипулятора.

Данная библиотека является её первой версией и в дальнейшем она может быть доработана для возможности создания в ней моделей алгоритмов работы робототехнических систем типа «руки манипулятора» с визуализацией схематичной работы данного алгоритма. В итоге - это позволит продемонстрировать работу схемы в целом и проанализировать правильность составления алгоритма по работе с ней, тем самым позволяя пользователю проанализировать работу придуманной им системы на примере визуальной модели помогая лучше понять возможности и сложность его манипулятора при помощи кинематической модели.

Также пользователь сможет дорабатывать библиотеки содержащих набор физических законов для решения конкретных проблем в проектировании и моделировании. Преимуществами моей системы можно назвать минимализм предоставляемых ею возможностей позволяющей разбираться в узконаправленных проблемах самому и использовать в их решениях свои методы, не опираясь на стандартный набор уже существующих. Тем самым пользователь сможет сам создавать новые решения стандартных проблем.

Использование визуализации по средствам имитации работы кинематических схем позволяет во время создания работы системы, изучать проблемы управления и сами принципы взаимодействия звеньев, не отвлекаясь как на форм-фактор, так и на прочие моменты реализации самого изделия. Так как моя библиотека является узкоспециализированной в области робототехники, затрагивая только модели манипуляторов, то все описанное выше позволяет пользователям максимально эффективно проверять работу своих систем.

Также стоит заметить, что реализация библиотек в среде LAZARUS – дает возможность создавать пользователям программы на их основе для разных ОС, что позволит использовать их и на мобильных устройствах. Все это благодаря описанному мной ранее минимализму функционала, для использования которого ненужно, как большой вычислительной мощности, так и большого объема памяти и в тоже время нечем не, ограничивая возможности работы.

Реализация проекта как класса с большим количеством переменных и открытом исходным кодом позволяет как дополнять класс новыми переменными для увеличения функционала, так и напрямую работать с уже существующими, позволяя данному проекту развиваться в нескольких направлениях...

1. Программы тестирования студентов
2. Программы презентации работы звеньев руки манипулятора
3. Программы по имитации управления манипулятором
4. Набор библиотек для любителей робототехники, помогающие им в создании моделей манипуляторов.

Несмотря на описанные мною достоинства моей библиотеки в ней на данный момент имеются и существенные недостатки такие как ограниченный набор звеньев, почти полное отсутствие библиотек физических законов, что было бы необходимо для использования программы в целях моделирования реальных систем, так как моей задачей основной задачей было создание классов, которые дадут возможности, описанные мою выше. Разработка физических библиотек и мультиплатформенность требует большего времени и ресурсов. Но еще в начале я описал что главной задачей моего проекта, является ознакомление с принципами как разработки решений, так и создания моделей робототехнических систем, а не создание высоко детализированных моделей роботов. Поэтому ее использование должно изначально носить учебно-ознакомительный характер по сравнению с ее аналогами.

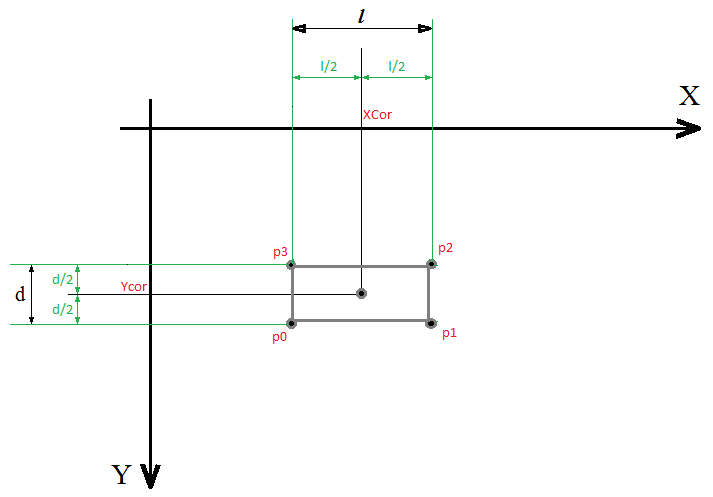
Теперь, когда у вас появилось примерное представление о целях и назначениях моей программы, а также об её первоначальных характеристиках, можно рассмотреть реализацию как самого функционала, так и архитектуру представленной далее мной системы…

2.Математическая модель для расчета законов движения звеньев TZweno.

Для построения звеньев мной была выведена следующая закономерность между соответствующими координатами X и Y Центра построение (Начальной точки построения) в двумерной системе координат и координатами точек геометрических примитивов, используемых для построения звеньев.

Таким образом, изменяя координаты центра построения, мы сможем изменять положение всего звена. Для звеньев, чье построение возможно под углом, кроме координат центра, также на положение будут влиять и координаты угла поворота.

2.1 Подвижное основание (Рисунок.2.1)



**Рисунок 2.1** - Подвижное основание.

Простейшим вариантом его изображения можно считать прямоугольник, состоящий из 4х точек, равно удаленных от центра.

Примем координаты центра равными ( XCor ; YCor) ,длину звена - l, а ширину - d.

Так как все точки равноудалены от центра их координаты будут отличатся от координат центра на половину расстояния между ними [ l/2 ] для оси Ох и [ d/2 ] для оси Оу .

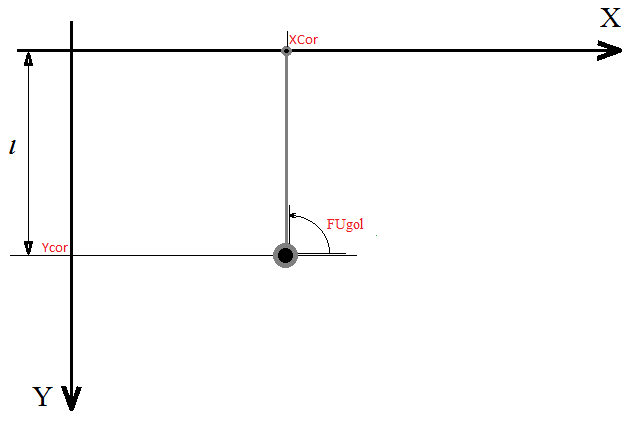
P0(XCor - l/2; YCor - d/2)

P1(XCor + l/2; YCor - d/2)

P2(XCor + l/2; YCor + d/2) ( 2.1)

P3(XCor - l/2; YCor + d/2)

2.2 Вращательная пара



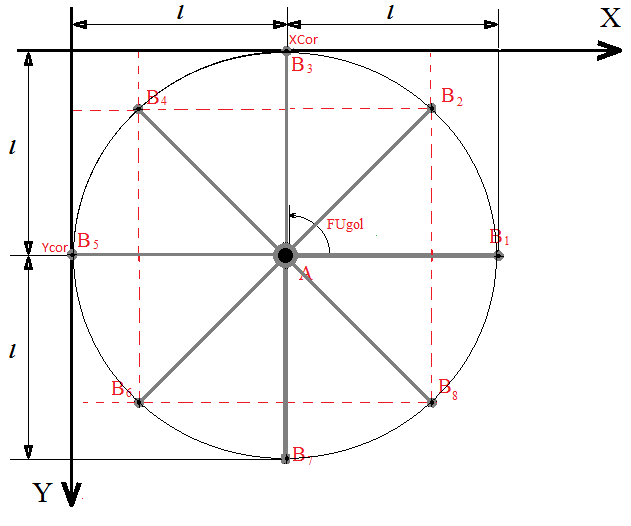
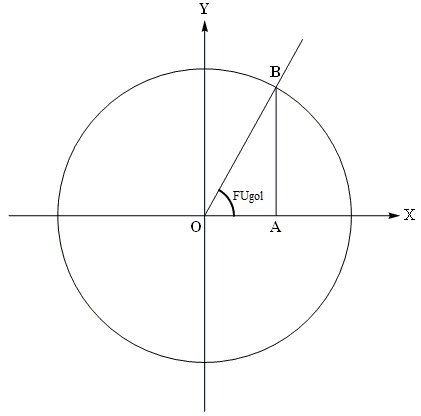
**Рисунок 2.2** - Звенья вращательной пары расположенные под углом π/2 (90о ) относительно оси оХ.

Простейшим ее вариантом ее изображения можно считать отрезок прямой ограниченный точками **A** и **B**.

Так как данное звено совершает вращательное движение с центром в точке **A**  ее координаты при различных значениях угла **FUgol** остаются неизменными (**XCor** ; **YCor**), а координаты конечной точки **B** изменяются по осям координат в зависимости от значения угла **FUgol ,** как показано на рисунке 2.2 , пределах:

oX: (XCor – L; XCor + L);

oY: (XCor – L; XCor + L); (2.2)

**Рисунок 2.3** - Движение Вращательной пары **Рисунок 2.4** - Определение

при изменении угла поворота. тригонометрических функций.

Так как с изменением угла поворота расстояние между точками **A** и **B** остается неизменным для учета изменения координат точки **B** нужно рассмотреть отношение между координатами неподвижной точки **А** и изменением проекции отрезка **AB (l)** на оси координат.

(2.3)

Из чего можно считать справедливой следующую тригонометрическую зависимость между точками **A** и **B**:

Bx:= XCor + l\*Sin(Fugol)

By:= YCor + l\*Cos(Fugol) (2.4)

Произведем проверку правильности вычислений расчетом положения точки **B** для значений угла поворота равных 90о и180о :

1) Fugol = 90 о (Рис.3.1)

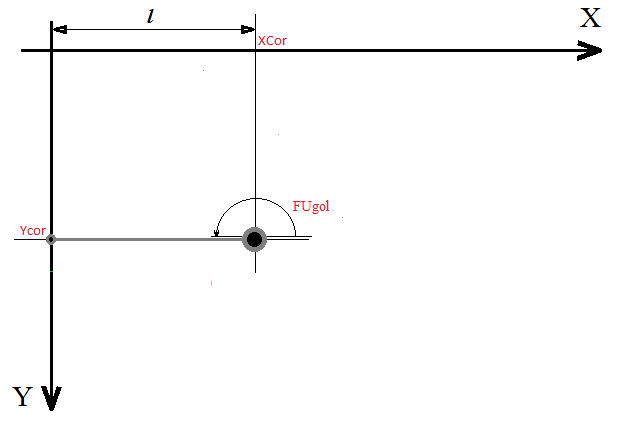
Bx := XCor + l\*Sin(90 о) = XCor + l

By := YCor + l\*Cos(90 о) = YCor (2.5)

2) Fugol =180о (Рис.2.3)

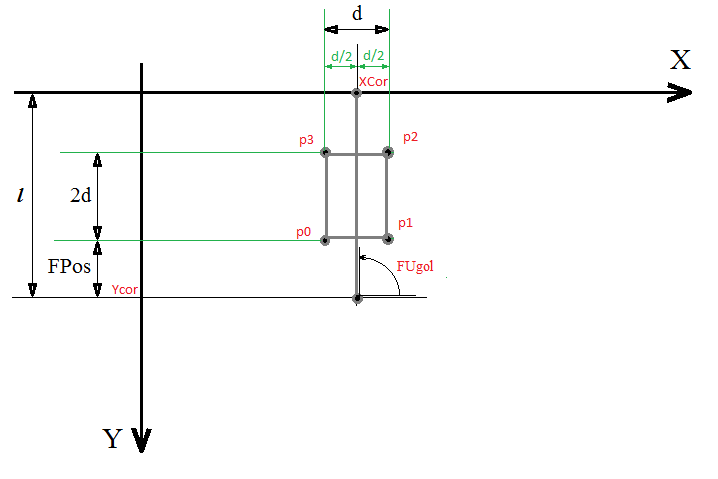
Bx := XCor + l\*Sin(180 о) = XCor

By := YCor + l\*Cos(180 о) = YCor + l (2.6)



**Рисунок 2.5** - Вращательная пара расположенная параллельно оси оХ (другими словами под углом π (180о ) относительно оси оХ.)

2.3 Поступательная пара (Рисунок.2.6)



**Рисунок 2.6 -** Поступательная пара расположенная под углом π/2 (90о ) относительно оси оХ.

Простейшим вариантом данного звена можно считать прямоугольник, состоящий из 4х точек осуществляющий передвижение по отрезку длиной l . Расчет изменения положения отрезка относительно угла поворота будет тем же что и для вращательного звена, поэтому произведем расчет только изменения положения прямоугольника на нем.

Примем координаты центра равными ( XCor ; YCor) , Длину звена l, Угол поворота относительно нормали FUgol.

Таким образом в общем виде зависимость координаты конца отрезка от угла поворота можно считать:

Bx := XCor + l \* Cos(Fugol)

By := YCor + l \* Sin(Fugol) (2.7)

Рассмотрим следующие виды зависимости положения прямоугольника от координат центра :

1)Для угла в 90o как показано на рисунке 2.6

Координаты точек p0,p1,p2,p3 можно обозначить как:

P0 (XCor - d/2; YCor + FPos)

P1 (XCor + d/2; YCor + FPos)

P2 (XCor + d/2; YCor + FPos + 2d) (2.8)

P3 (XCor - d/2; YCor + FPos + 2d)

2)Для угла в 180o как показано на рисунке 2.7

Координаты точек p1,p2,p3,p4 можно обозначить как :

P0 (XCor – FPos ; YCor- d/2)

P1 (XCor – FPos ; YCor + d/2)

P2 (XCor - FPos - 2d; YCor + d/2) (2.9)

P3 (XCor - FPos - 2d; YCor - 2d)

Таким образом можно отметить что при изменении угла поворота на координаты точек построения относительно координат начальной точки будут влиять разные параметры размера прямоугольника(ползуна). Степень их влияния, как и в примере с вращательной парой можно выразить с использованием законов синуса и косинуса:

1. P0

X := XCor + Pos\*Cos(FUgol) - (d/2)\*Cos(FUgol)

Y := YCor + Pos\*Sin(FUgol) - (d/2)\*Sin(FUgol) (2.10)

1. P1

X := XCor + Pos\*Cos(FUgol) + (d/2)\*Cos(FUgol)

Y := YCor + Pos\*Sin(FUgol) + (d/2)\*Sin(FUgol) (2.11)

1. P2

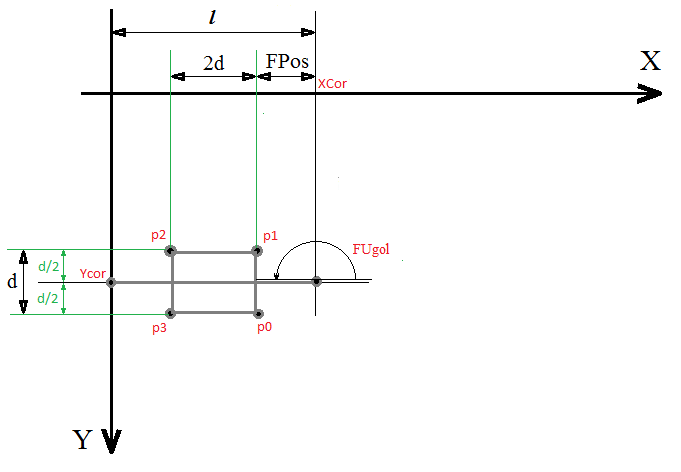
X := XCor + Pos\*Cos(FUgol) + (d/2)\*Cos(FUgol) - 2\*d\*Sin(FUgol)

Y := YCor + Pos\*Sin(FUgol) + (d/2)\*Sin(FUgol) - 2\*d\*Cos(FUgol) (2.12)

1. P3

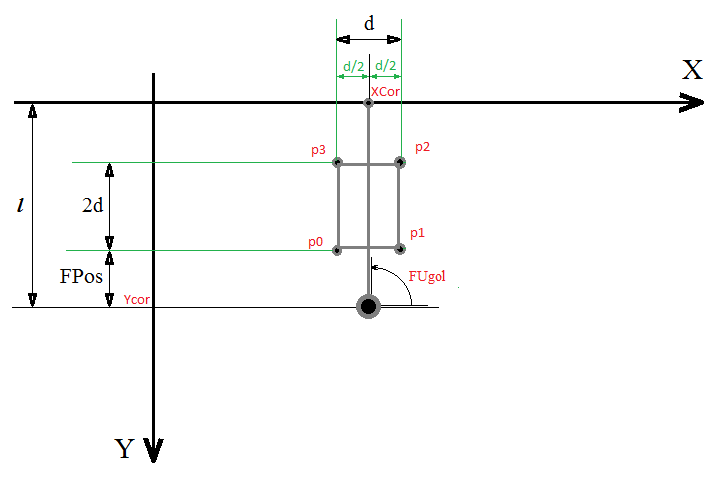
X := XCor + Pos\*Cos(FUgol) - (d/2)\*Cos(FUgol) - 2\*d\*Sin(FUgol)

Y := YCor + Pos\*Sin(FUgol) - (d/2)\*Sin(FUgol) - 2\*d\*Cos(FUgol) (2. 13)

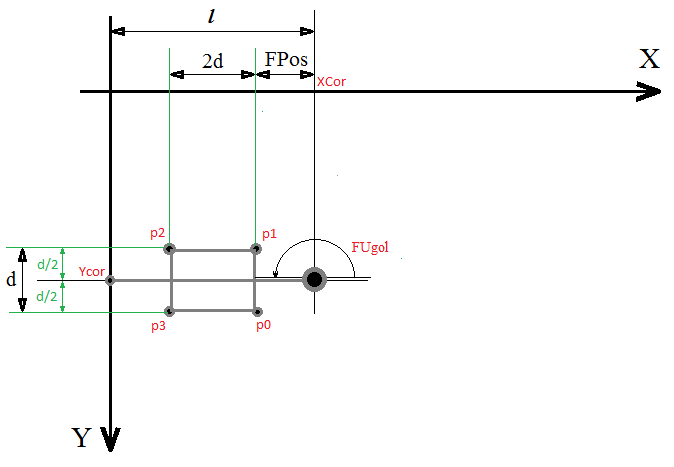


***Рисунок 2.7 -*** Поступательная пара расположенная параллельно оси оХ (другими словами под углом π (180о ) относительно оси оХ.)

2.4 Кулисная группа звеньев



***Рисунок 2.8*** - Кулисная группа звеньев расположенная под углом π/2 (90о ) относительно оси оХ

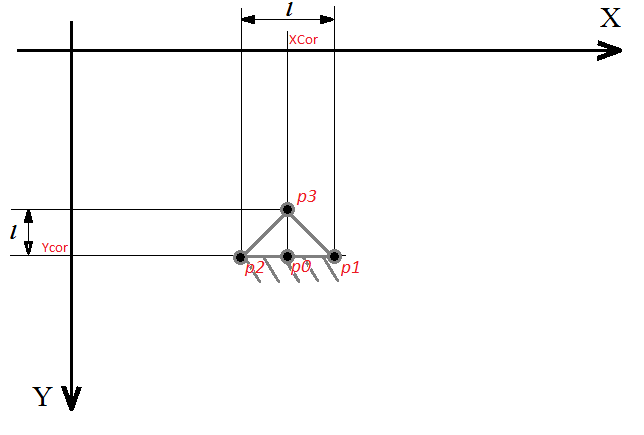


***Рисунок 2.9*** - Кулисная группа звеньев расположенная параллельно оси оХ

(под углом π (180о ) относительно оси оХ.)

Так как упрощенно Кулисную группу звеньев можно рассматривать как совокупность вращательной и поступательной пары то для ее построения будет требоваться тот же набор законов что и для данных пар.

2.5 Стойка



***Рисунок 2.10*** - Стойка.

Простейшим вариантом данного звена можно считать треугольник с центром построения в точке **p0** , и штриховкой у основания.

Примем координаты центра равными ( XCor ; YCor) , высоту звена l, ширину основания l.

Таким образом в общем виде зависимость координат точек построения от координат центра будет:

1. P1

X := XCor + l/2

Y := YCor (2.14)

1. P2

X := XCor – l/2

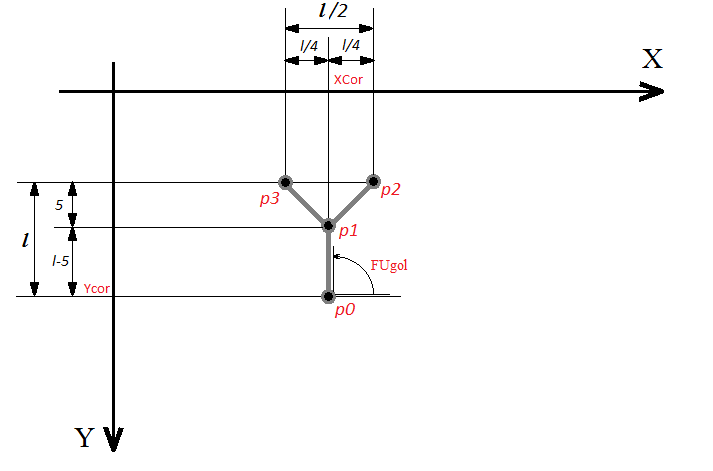
Y := YCor (2.15)

1. P3

X := XCor

Y := YCor + l (2.16)

2.6 Схват



**Рисунок 2.11** - Схват .

Простейшим вариантом данного звена можно считать совокупность отрезков, показанную на рисунке 6.

Примем координаты центра равными ( XCor ; YCor) , высоту звена l, ширину звена l/2 .

Таким образом в общем виде зависимость координат точек построения от координат центра будет:

1. P1

X := XCor + (l - 5)\*Cos(FUgol)

Y := YCor + (l - 5)\*Sin(FUgol) (2.17)

1. P2

X := XCor + (l - 5)\*Cos(FUgol) + l/4\*Cos(FUgol)

Y := YCor + (l - 5)\*Sin(FUgol) + l/4\* Sin(FUgol) (2.18)

1. P3

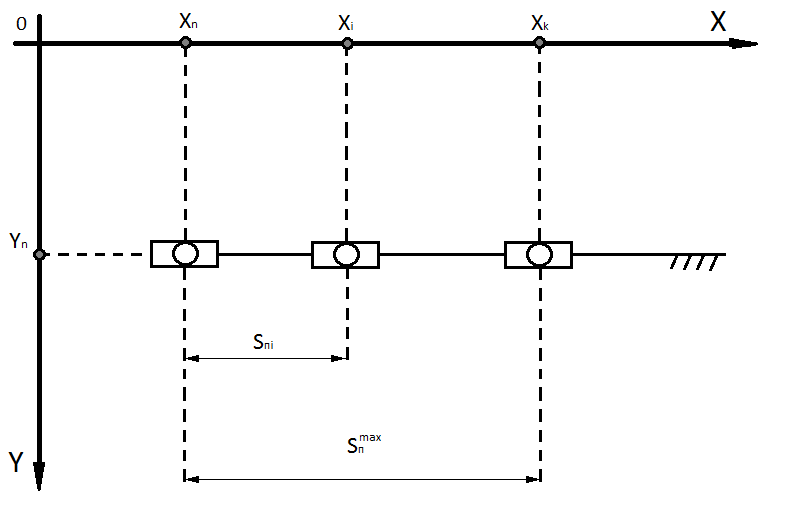
X := XCor + (l - 5)\*Cos(FUgol) – l/4\*Cos(FUgol)

Y := YCor + (l - 5)\*Sin(FUgol) – l/4\* Sin(FUgol) (2.19)

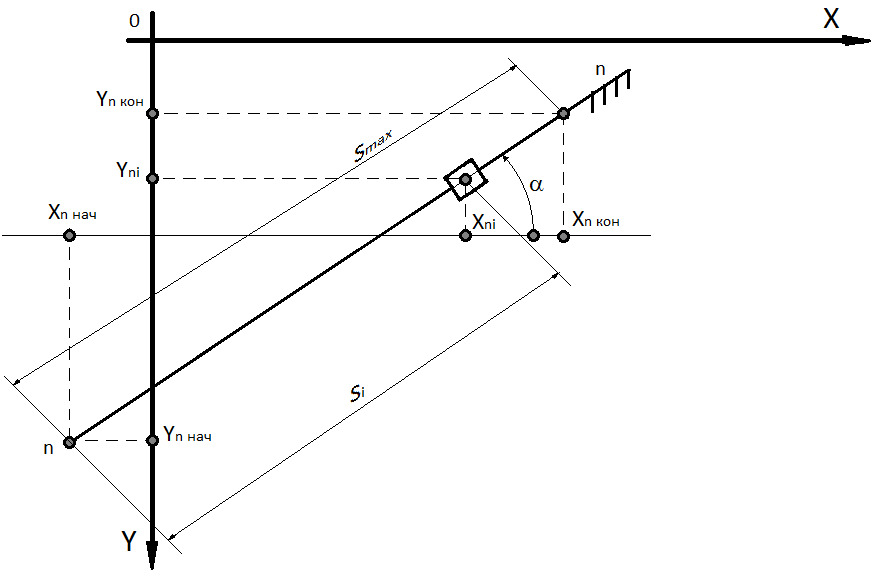
Для класса TZmove.

Для описания движения звеньев мною была выведена следующая закономерность между соответствующими величинами, характеризующими движение объекта.

Для Прямолинейного движения.



***Рисунок 3.1* -** Прямолинейное движение по горизонтали.



**Рисунок 3.2 -** Прямолинейное движение под углом.

На рисунке.3.1 изображен более общий, по сравнению с рисунком. 3.2 случай движения ползуна: линия перемещения ползуна «n-n» составляет некоторый угол α стойки с осью X

В этом случае:



 (3.1)

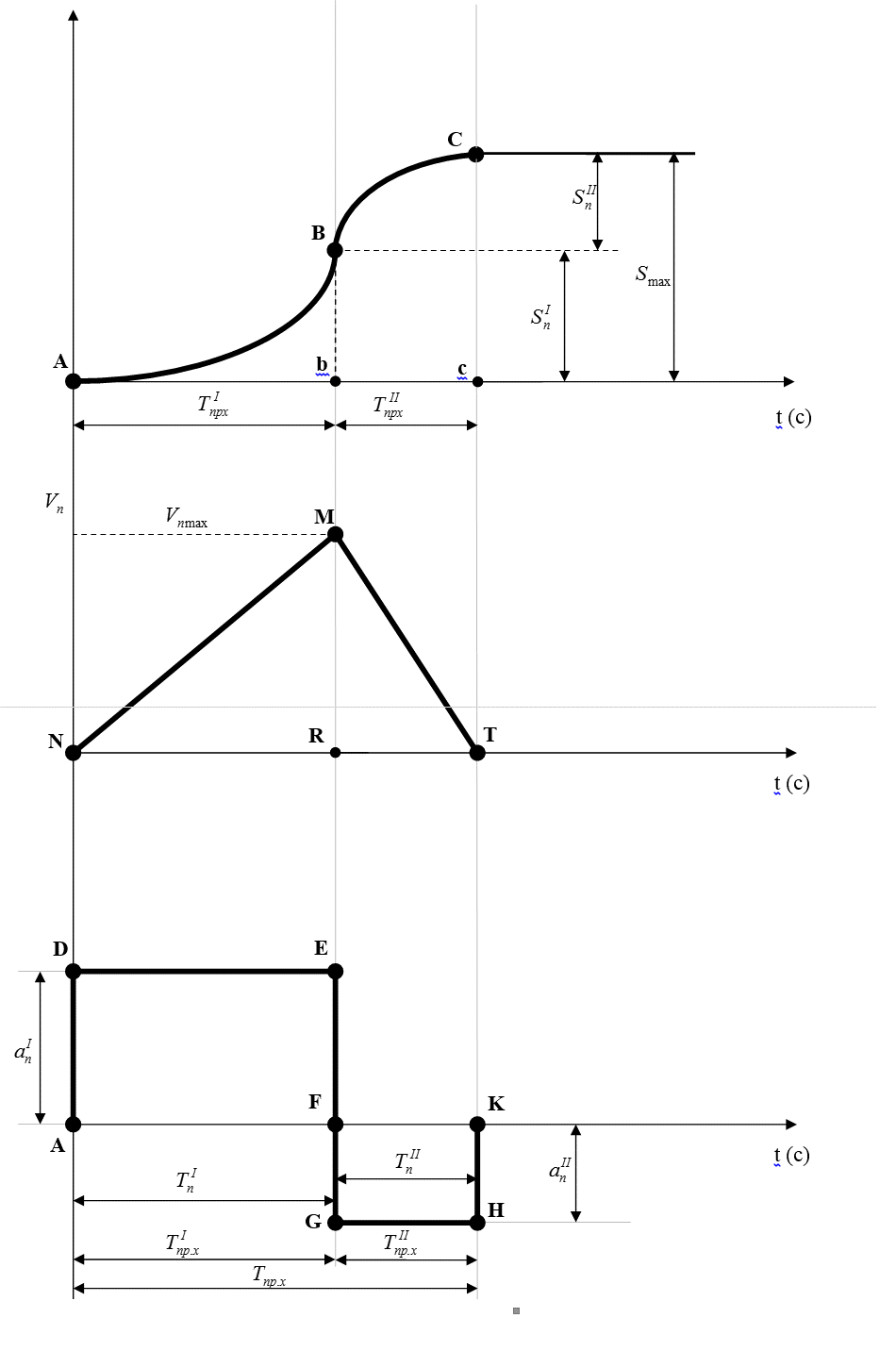
При этом необходимо учесть возможность движения как в направлении по возрастанию координат (Х;Y) так и обратное этому движение.

;

Однако, сначала будет рассмотрен вариант задачи, изображенной на рис.1, как более простой, тем более что первый вариант может быть получен путем поворота оси Х на угол α стойки против часовой стрелки.

Чтобы исследовать все характеристики рассматриваемого закона движения, рассмотрим его в аналитической форме.

В момент начала относительного движения звеньев манипулятора начальная скорость звеньев будет равна нулю. Рассмотрим законы движения для равноускоренного и равнозамедленного движения



***Рисунок 3.3 -*** Изменение Скорости, Ускорения и пройденного пути.

Рассмотрим прямой ход ползуна, соответствующий перемещению Smax. Время ti на этой фазе движения изменяется в следующих пределах:

Участок 1:  ti ≤ T 1 (3.2)

Участок 2: T 1 ≤ ti ≤ T (3.3)

Рассмотрим закон постоянного ускорения: зависимость для первого интервала времени (неравенства (1)) будет следующая:

Для первого участка пути для равноускоренного движения

*a*1 = const1, (3.4)

а для второго для равнозамедленного движения

*a*2 = - const2. (3.5)

Интегрируя дважды выражение для первого интервала времени движения ползуна, получаем выражения для скорости и перемещения рассматриваемого звена :

*V*1=  (3.6)

*S*1 =  (3.7)

Для решения данной задачи рассмотрим ее частный случай, когда “начальное перемещение” и начальная скорость равны нулю…

Начальные условия для определения Cи C: при t = 0, *V*п = 0,  C

Тогда

*Vп =;* (3.8)

*Sп =* (3.9)

При равнозамедленном движении (участке II) получаем обратную динамику изменения величин:

*Vп =* (3.10)

*S* (3.11)

Поэтому равнозамедленное движение при отсутствии начальной скорости можно рассматривать как равноускоренное движением в обратном направлении.

Рассмотрим условия, которым должны удовлетворять зависимости a(t) и V(t).

Так как, an в начале и в конце фазы движения равно нулю, то площади прямоугольников ADEF и FGHK (Рис. 9) должны быть равны, то есть

, (3.12)

или

*,*  (3.13)

Так как , то учтя предыдущее выражение для  можно вывести следующие закономерности :

, (3.14)

**  (3.15)

Перемещения  и  ползуна пропорциональны площади прямоугольников NMR и RMT (Рис. 9).

Тогда обозначим значение в скорости в точке М через , таким образом мы получаем следующие зависимости движения от максимальной скорости :

, (3.16)

 (3.17)

Делим равенство (12) на равенство (13):

 (3.18)

Приняв во внимание зависимости полученные в формуле (9) получаем :

 (3.19)

Так как , то из равенства (14) получаем:

 (3.20)

 (3.21)

Из равенства (7) следует:

 , (3.22)

 (3.23)

Подставляя в уравнения (3.22) и (3.23) значения для  и  из равенств (3.20), (3.21) и (3.14), (3.15), получаем

 (3.24)

 (3.25)

Далее, пользуясь равенствами (3.8), (3.24) и (3.25) имеем

 (3.26)

Из полученных равенств следует, что если задан полный ход ползуна S, время прямого хода T=Tn и коэффициент, то можно по формулам (3.14) и (3.15) определить время  и , по формуле (21) – максимальное значение и по (3.24) и (3.25) – значения ускорения  и .

Из равенства (3.26) следует, что максимальное значение скорости не зависит от коэффициента Kn , то есть от распределения времени прямого (или обратного) хода  по участкам положительного  и отрицательного  ускорений.

Если коэффициент Kn принят равным =1, то получаем симметричный закон движения, для которого на интервале  имеем

**  (3.27)

Принимая во внимание равенство (3.8) скорость для этого движения определяем так , (3.28)

а перемещение  из равенства (3.9) равно

 (3.29)

Для фазы обратного движения (фазы возврата), соответствующий отрезок времени обозначим То.д. Расчёт всех параметров движения может быть сделан по уже выведенным формулам с заменой коэффициента Kn для участка прямого хода на Ко.д. для участка обратного движения.

Второй участок движения

Рассмотрим подробнее второй участок:

На данном участке движения звено производит торможениепоэтому ускорение примем отрицательным:



Рассмотрим граничные условия:

При 

, отсюда 

тогда 

При 



Тогда выражения для скорости и перемещения рассматриваемого звена примут вид :

** (3.32)

 (3.33)

, (3.34)

Учтём, что 

 (3.35)

Объединив выведенные нами закономерности мы получаем:

 (3.36)

 (3.37)

Выведем зависимость участка пути при равнозамедленном движении от коэффициента Kn

 (3.38)

 (3.39)

 (3.40)

Окончательно:  (3.41)

Подставив значение :

 (3.42)

 (3.43)

Если К,  мы можем вывести аналогичную закономерность пути от максимальной скорости :

 (3.44)

Таким образом, при прямолинейном движение с постоянной скоростью, для нахождения незаданных параметров можно использовать следующие зависимости:

Для первого участка**: **

 (3.24)

 (3.8)

 (3.9)

 (3.14)

Для второго участка: 

 (3.15)

 (3.25)

 (

 (3.41)

где

 (3.19)

 (3.26)

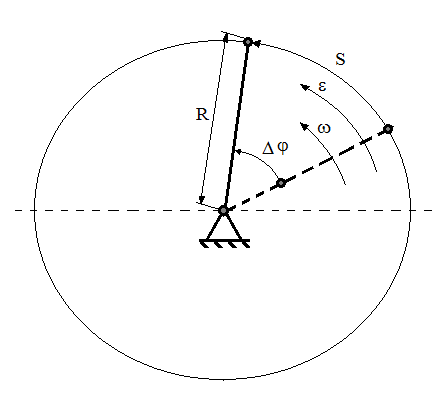
При заданных величинах:

1. - ход ползуна

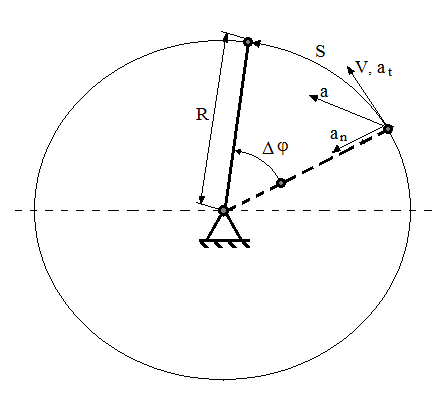
2.  - время прямого/обратного хода ползуна

3.  - отношения продолжительности участков с  и 

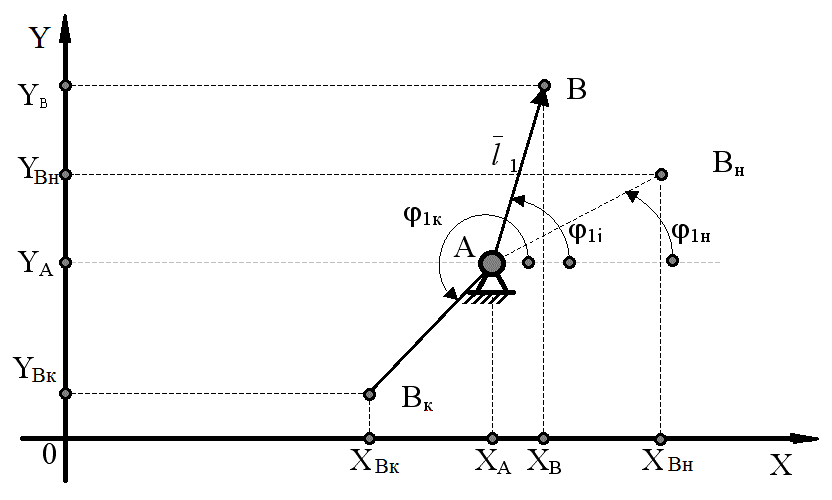
**3.2 Для вращательного движения.**

****

**Рисунок 3.4 -** Движение вращательного звена под действием угловой скорости и углового ускорения.



**Рисунок 3.5 -** Движение вращательного звена под действиемлинейной скорости и ускорения.



**Рисунок 3.6 -** Общая схема движения вращательного звена.

Рассмотрим движение по окружности.



(3.46)

(3.47)

Проведем аналогию между вращательным и равноускоренным движением по прямой, выражения для скорости и перемещения при прямолинейном движении будет иметь вид:

V = (3.48)

S= (3.49)

(3.50)

(3.51)

Рассмотрев зависимости (25) и (28) для скорости и угловой скорости , и зависимости (26) и (30) для ускорения и углового ускорения можно сделать вывод что та как для вращательного движения путь пройденный телом за время t будет равен длине дуги окружности **S** ограниченной углом поворота **** уравнение (28) примет вид :

(3.52)

Кроме того закономерность (32) позволяет считать справедливой зависимости параметров для равноускоренном движении выведенные в предыдущем разделе справедливыми и параметров вращательного движения.

Согласно формуле (27) изменение угла поворота будет равно :

(3.53)

Определим зависимость между угловой скоростью и линейной:

(3.54)

Определим зависимость между угловым ускорением и ускорением.

(3.55)

(3.56)

(3.57)

(3.58)

3.3 Разработка спецификации для информационной системы.

Данная программа позволяет создавать модель руки манипулятора из последовательно соединяемых поступательных, вращательных, а также в виде “кулиса-камень” звеньев. Пользователь задаёт силы и скорости для каждого звена для дальнейшей визуализации программой, движения руки манипулятора при данных параметрах.

Для Создания (построения) звена требуется в **меню** *Рисунок.5.2(2)* выбрать тип звена и в появившемся **окне создания звена** *Рисунок.5.3(1)* в зависимости от типа звена ввести его **параметры**\*(1) на выбор и нажать кнопку **создать** *Рисунок.5.2(2).*

**Звено** – это деталь или группа деталей, жестко соединенных между собой и движущихся как единое твердое тело.

**Кинематическая пара** – это соединение двух звеньев, обеспечивающее определённое относительное движение.

**Кинематическая цепь** – это связанная система объектов, образующих между собой кинематические пары.

Параметры (1):

1. **Вращательная пара. (***Одноподвижная, допускает лишь относительное вращательное движение звеньев вокруг оси. Элементы вращательной пары могут состоять из любых поверхностей вращения.)*

Угловая скорость, угловое ускорение или линейная скорость, ускорение. Сила\*(3).

а) Размер звена и угол поворота.

б) Координаты конечной точки.

1. **Поступательная пара. (***Одноподвижная , с геометрическим замыканием, допускает лишь прямолинейное поступательное относительное движение звеньев.* **)**

Ускорение, скорость. Сила\*(3).

а) Размер звена угол поворота и положение звена на направляющей.

б) Координаты конечной точки и положение звена на направляющей.

1. **Кулисная группа звеньев. (***Шарнирный механизм, в котором два подвижных звена кулиса и кулисный камень связаны между собой поступательной (иногда вращательной при дуговой кулисе) кинематической парой)*

Скорость, Ускорение (поступательного движения). Угловая скорость, угловое ускорение или Линейная скорость, ускорение (вращательное движение).

а) Размер звена угол поворота и положение звена на направляющей.

б) Координаты конечной точки и положение звена на направляющей.

**4) Схват. (***Устройство, в котором захватывание и удержание объекта производятся посредством относительного перемещения частей данного устройства***).**

Размер звена.

**5) Стойка.** *(звено, принимаемое за неподвижное.)*

Размер звена**.**

1. ***Подвижное основание***

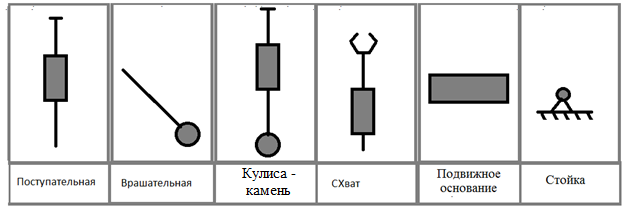
Скорость, Ускорение (поступательного движения).

Размер звена.

Для моделирования простых схем манипулятора этих наборов звеньев должно быть вполне достаточно. В следующих версиях программы появятся возможность расставлять ограничители движения(препятствия), а также будет дополнен и сам набор доступных звеньев.

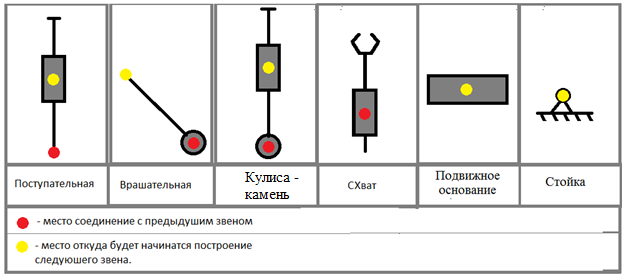
**Манипулятор** - это механизм для управления пространственным положением орудий, объектов труда и конструкционных узлов и элементов.

Для большей ясности какое звено будет построено кроме подписей у кнопок на самой кнопке будет изображение того звена, которое будет строится. *Рисунок.3.1*



**Рисунок 3.1** - Примерное изображение на кнопках выбора типа звена.

Первое звено строится из середины подвижной **платформы**.*Рисунок.5.2(3).* Более полная информация по принципу построению остальных звеньев показана на *рисунке 3.2*.



**Рисунок 3.2** - Принципы построения модели манипулятора из звеньев.

После создания звена предполагается что его размер будет невозможно поменять.

После нажатия на кнопку создать в меню состояния звеньев *Рисунок.5.2(7)* появится описание созданного звена (Тип звена, Номер, длина, угол) и его характеристики\*(2) *Рисунок.5.2(8).*

Характеристики(2):

Скорость, ускорение, угловая скорость, угловое ускорение.

При нажатии на кнопку **начать движение** *Рисунок.5.2(4)* звенья начнут свое перемещение по траектории движения исходя из введенных пользователем значений сил, ускорений, скоростей.

При нажатии на кнопку **остановить движение** *Рисунок.5.2(4)* звенья прекратят свое передвижение.

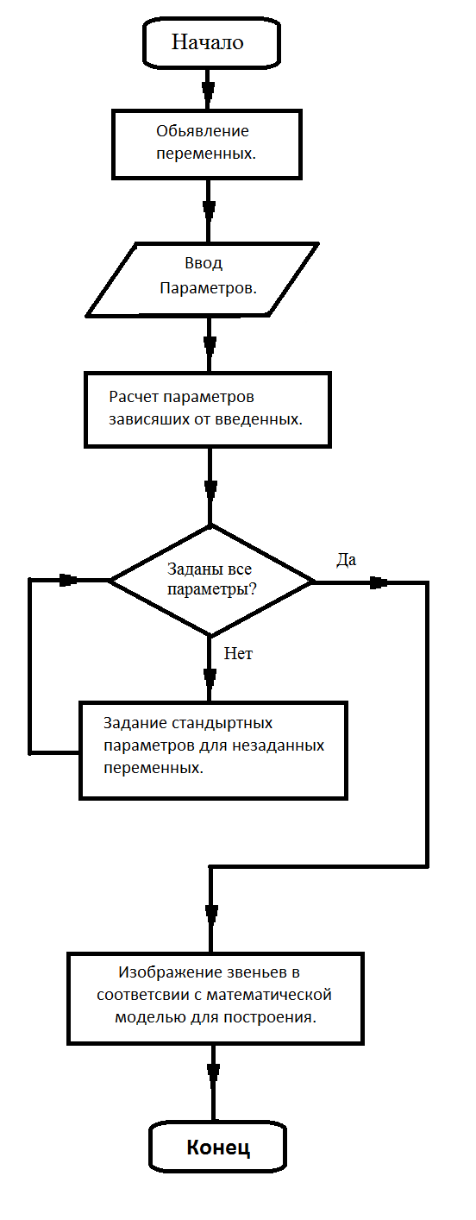
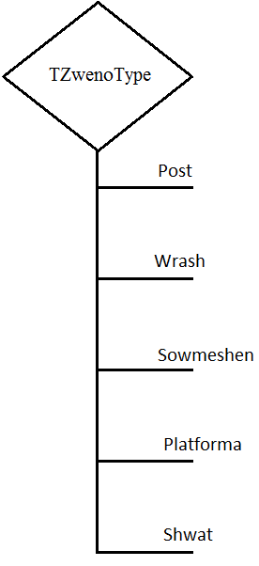
Для нанесения координатной сетки на поле отображения движения объекта *Рисунок.5.2(1)* нужно установить галочку в графе **Координатная разметка** *Рисунок.5.2 (5).*

При нажатии на кнопку **Выход** *Рисунок.5.2(6)* будет осуществлён выход из программы.

4 Описание классов

Для реализации работы программы будут использоваться:

a) **Класс звеньев TZweno**

***Рисунок 4.1*** *-* Упрошенная блок схема ***Рисунок 4.2*** *-* Блок схема

для класса **TZweno**. **TZwenoType**

Блок-схемы для данного класса представлены на рисунках 4.1 , 4.2 .

Данный класс содержат в себе основные параметры звена, такие как: его размер, вид, тип силы приложенный к нему, а также положение и скорости. Массив типа звена **TZwenoType (**Рисунок 4.2) содержит набор переменных в зависимости от которых будет осуществляется построение звена**.**

Для изображения на экране звена будет использоваться процедура **Paint** которая в зависимости от значения в массиве **Zweno** будет осуществлять различное построение (изображение) на экране:

1. **Post** - Поступательное звено.
2. **Wrash** - Вращательное звено.
3. **Sowmeshen** - **Кулисная группа звеньев** (Вращательно- поступательное звено.)
4. **Nomove** - Стойка
5. **Platforma** – Подвижное основание
6. **Shwat** – Схват

Так как реализация вывода графики на экран реализована через набор стандартных методов делфи-образного языка (**Lazarus**) работа с данным классом возможна на большинстве операционных систем.

Для каждого параметра звена создан специальный метод обращения к его характеристикам:

* Координаты положения начальной точки звена

**XCor** - Служит для задания/изменения координаты звена по оси Оy (для ориентации положения начальной точки относительно оси Oy).

**YCor** - Служит для задания/изменения координаты звена по оси Оx (для ориентации положения начальной точки относительно оси Ox).

* **Zweno –** Определяет тип звена (Если значение данного параметра не указывается перед объявлением (построением) звена звено строится как вращательное.
* **Dlina** – Служит для задания/изменения длины подвижной части звена.
* **Ugol** - Служит для задания/изменения угла поворота (вращательного или звена кулисной группы (вращательно поступательного с вращательным основанием)

Данные методы имеют свои процедуры обращения к параметрам звена что позволяет избежать сбоев в работе данного класса.

Кроме этого для осуществления визуализации движения составных частях звена относительно друг друга реализованы следующие переменные:

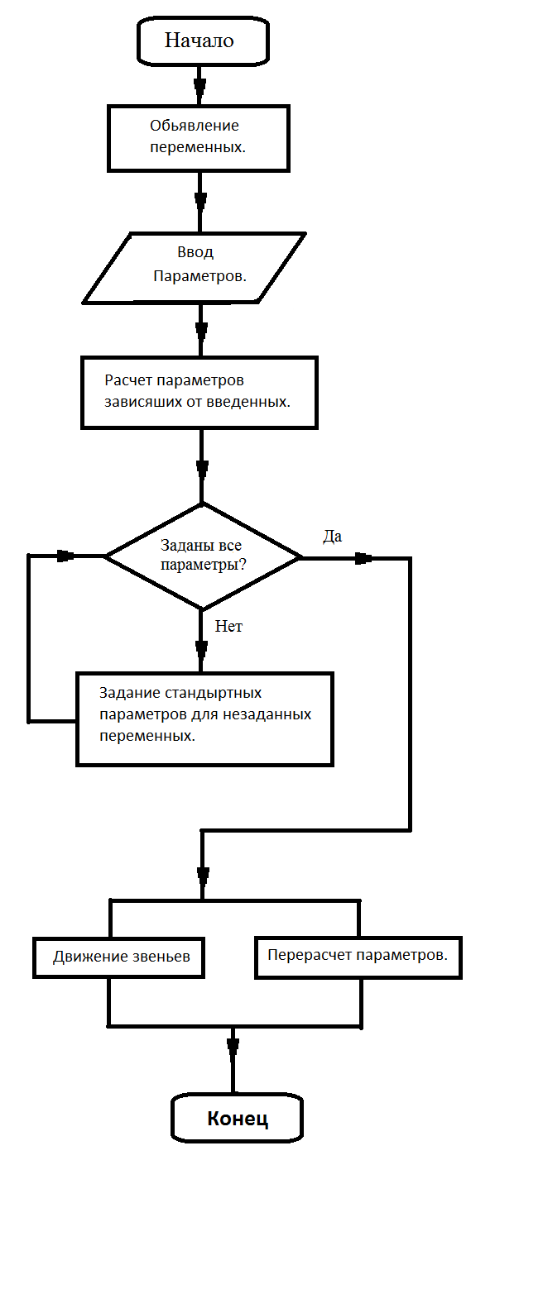
**Pos –** Для поступательного или кулисного звена определяет положение подвижной части звена совершающего относительное прямолинейное относительно точки соединения данного звена с основанием или звеном построенным ранее.

**D** – диаметр звена совершающего прямолинейное движение.

Подвижность компонентов звена осуществляется через перерисовку звена на канве для этого каждое звено содержит свою математическую модель построения. \*(3)

Кроме того, при реализации программ на основе данного класса реализована возможность выбора цвета звена через переменную **Pen** что позволит как расширить функционал программы (добавляя в нее функцию выбора цвета звена) так и управлять цветом звеньев при создании программ на основе данного класса.

b) **Класс движения**  **TZmove .**



***Рисунок 4.3*** Упрошенная блок-схема для класса **TZmove**.

Блок-схема для данного класса представлены на рисунках 4.3 .

Потомок класса **TZweno** реализующего графическое построение звеньев. Осуществляет работу по движению звеньев за счет расчета законов движения базирующихся на наборе параметров:

1. Общего времени движения.

(Расчет скорости и ускорения и торможения производится с расчетом что время делится на 3 равных части на каждый тип движения. (ускоренное, движение с постоянной скоростью, торможение)

1. Времени каждого типа движения (ускоренное, движение с постоянной скоростью, торможение), при вводе производится подсчет общего времени движения.
2. Параметрах движения (Путь(**S**), Скорость(**V**), Ускорение(**A**),Угловое ускорение(**E**),Угловая скорость(**W**)).

3.1- Для вращательной пары или кулисной группы звеньев предусмотрена зависимость между угловой скоростью и линейное при вводе параметров.

3.2-Для поступательной пары или подвижного основания перерасчета параметров не происходит.

Также данный класс содержит процедуру **newValue** которая позволяет вывести если это возможно недостающие параметры для движения звена.

Для избежание ошибок при перерасчете параметров созданы переменные **S2, V2, A2, W2,** **E2** контролирующие какой параметр был введен, а какой вычислен или задан автоматически.

Для контроля параметров относящихся к участкам равноускоренного или равнозамедленного движения были созданы следующие переменные:

1. **S3, V3, A3**- переменные характеризующие равноускоренное движение.
2. **S4, V4** - переменные характеризующие движение с постоянной скоростью.
3. **S5, V5, A5** - переменные характеризующие равнозамедленное движение.

**Vmax** – определяет максимальную скорость при ускоренном движении

**Kn** – Коэффициент отношения параметров равноускоренного и равнозамедленного движения.

Если какой-либо параметр движения не будет задан его значение по умолчанию будет равно нулю.

Кроме того данный класс осуществляет обращение к звеньям для их перемещения и расчета траектории их перемещения в зависимости от их положения и характеристик.

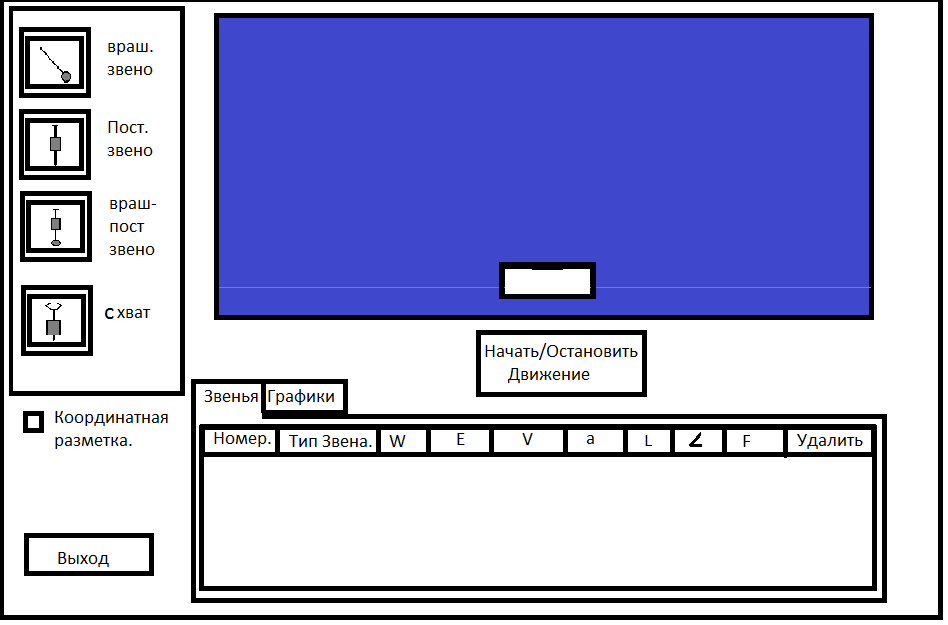
Правила

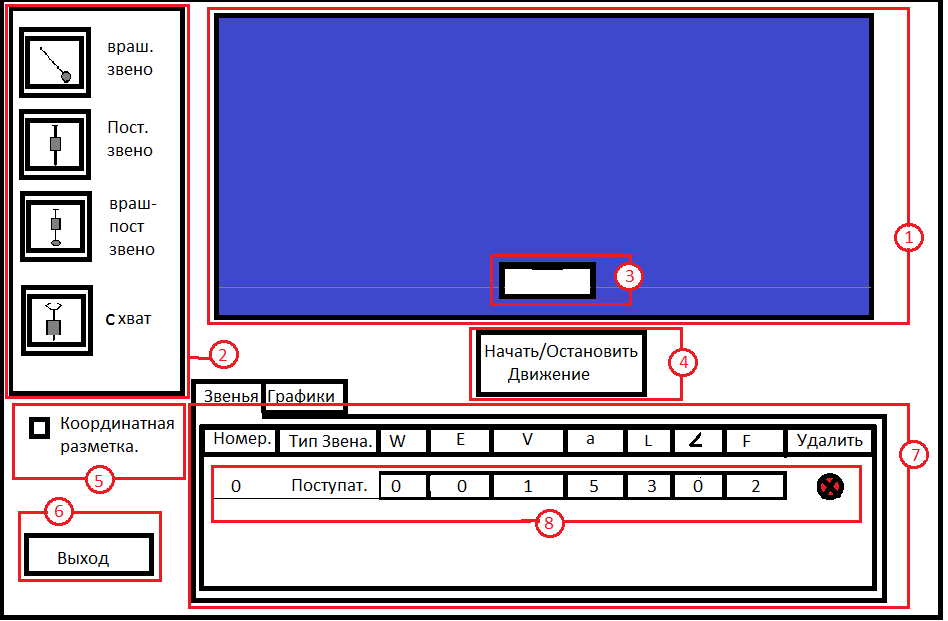
Угол поворота будет отсчитываться относительно горизонтали (оси параллельной Ох)

Среда функционирования продукта: Windows.

5 Создание Интерфейса системы

Описание интерфейса

**Рисунок 5.1 -** Начальный Экран работы приложения. (Примерный эскиз)



**Рисунок 5.2 -** Описание интерфейса программы

1) Поле Отображения движения объекта

2) Меню

3) Платформа

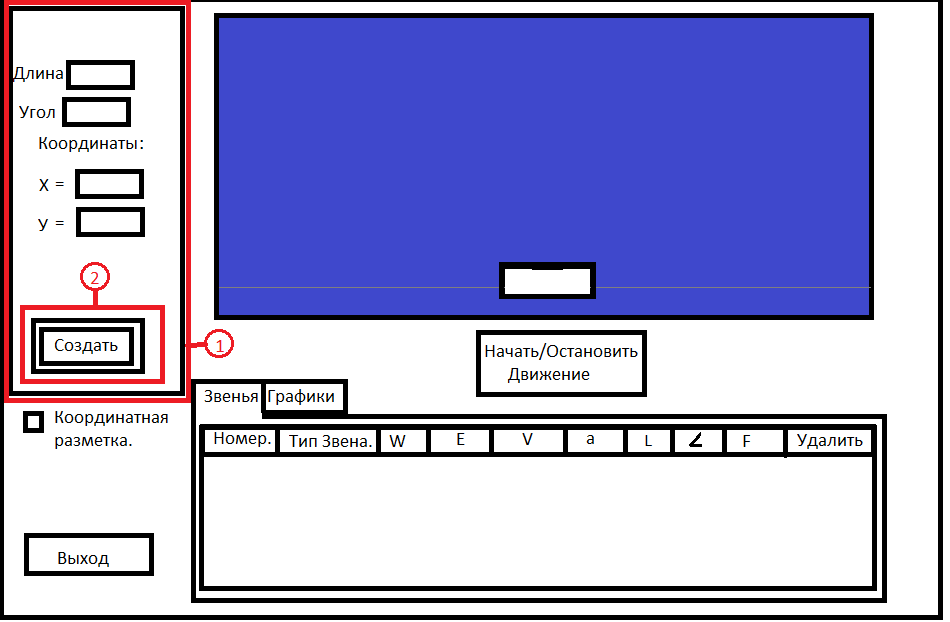
4) Кнопка начало остановки движения

5) Кнопка координатной разметки

6) Кнопка выхода

7) Меню состояния звеньев

8) Описание и характеристики созданного звена



**Рисунок 5.3 –** Окно ввода параметров звена

1) Окно построения звена (для вращательного звена)

2) Кнопка создания звена

1. Заключение

В данной работе мною были разработаны классы дающие возможность создавать на их основе программы имитирующие движение различных соединений звеньев, таких как вращательное звено, поступательное звено, стойка, подвижное основание, схват, кинематическая пара вида кулиса камень. Разработанные мною классы отвечают лищ за движение каждого звена по отдельности поэтому при создание программ на их основе потребуется учесть специфику последовательности их соединения. Для каждого класса была разработана своя математическая модель помогающая понять по какому принципу происходит то или иное движение, также была сделана примерная концепция программы с их использование в программе по моделированию движения манипуляторов и анализа их движения.

Кроме того мною были рассмотрены аналоги программ схожей направленности для ориентации в каком направление может в дальнейшем идти работа над данным проектом …

7 Список используемой литературы:

1. Артоболевский И.И Теория механизмов и машин.
2. Попов С.А., Тимофеев Г.А. Курсовое проектирование по теории механизмов и механике машин.
3. Осипов Д. Графика в проектах Delphi.
4. Сухарев М. Delphi Полное руководство. Включая версию 2010.
5. [www.mscsoftware.com/product/adams](http://www.mscsoftware.com/product/adams) (Официальный сайт разработчиков ADAMS).
6. ru.wikipedia.org/wiki/Microsoft\_Robotics\_Developer\_Studio (Интернет-энциклопедия).
7. www.umlab.ru/ (Официальный сайт разработчиков UM ).