**Содержание**

Введение

1. Требования к системе управления движением

2. Постановка задачи

3. Первый этап моделирования. Формирование математической модели

4. Второй этап моделирования. Реализация машинной модели

4.1 Блок схема.Основная схема

4.2 Листинг программы

5. Модель движения поезда рассмотренная в MatLab

Заключение

**Введение**

Американский математик Н. Винер, основатель науки кибернетики, отмечал, что XVIII столетие – это век часов, XIX столетие – это век паровых машин, настоящее время есть век связи и управления. «Техника нашего времени характеризуется использованием сложных комплексных систем, в которых переплетаются многочисленные и разнообразные материальные, энергетические и информационные потоки, требующие координации, управления и регулирования с быстротой и точностью, не достижимыми для внимания и памяти человека» [1]. Реализация таких задач управления возможна только с использованием технических средств автоматизации на базе вычислительной техники. Микропроцессоры стали входить в состав отдельных средств автоматики и контроля. Цифровая передача данных между отдельными устройствами сделала вычислительную сеть основой построения систем управления.

В настоящее время автоматизация большинства технологических процессов осуществляется на базе универсальных микропроцессорных контроллерных средств, которые в России получили название программно-технические комплексы (ПТК).[2]

ПТК представляют собой совокупность микропроцессорных средств автоматизации (микропроцессорных контроллеров, устройств связи с объектом), дисплейных пультов оператора. ПТК предназначены, в первую очередь, для создания распределенных систем управления технологическими процессами различной информационной мощности (от десятков входных/выходных сигналов до сотни тысяч) в самых разных отраслях промышленности.

Все универсальные микропроцессорные ПТК подразделяются на классы [2,3], каждый из которых рассчитан на определенный набор выполняемых функций и соответствующий объем получаемой и обрабатываемой информации об объектах управления.

1. контроллер на базе персонального компьютера;
2. локальный программируемый контроллер;
3. сетевой комплекс контроллеров;
4. распределенные мало масштабные системы управления;
5. полно масштабные распределенные системы управления.

В архитектуре автоматизированных систем управления технических программ контроллеры занимают место между уровнем датчиков и исполнительных механизмов и системами верхнего уровня управления процессом. Основная функция контроллеров в системе – сбор, обработка и передача на верхний уровень первичной информации, а также выработка управляющих воздействий на исполнительные механизмы.

Большинство современных контроллеров изготавливается по секционно-блочному принципу. Каждый логический модуль представляет собой отдельный блок. Что при необходимости позволяет заменять лишь отдельные модули без изменения архитектуры всей системы.

Основными функциональными элементами контроллеров являются:

* корпус;
* источник питания;
* процессорный модуль;
* модули ввода-вывода;
* модули связи и интерфейсов;
* специализированные модули.

Все современные программируемые логические контроллеры (ПЛК) обладают развитыми программными средствами. Несмотря на существования международного стандарта на языки программирования программируемых логических контроллеров многие производители снабжают свои контроллеры технологическими языками собственного производства. Технологические языки программирования позволяют проводить опрос входов и инициализацию выходов, обрабатывать арифметические и логические инструкции, управлять таймерами-счетчиками, осуществлять связь с другими программируемыми логическими контроллерами и компьютером.

Ввод программы в память контроллера осуществляется с помощью специальных программаторов или через интерфейс компьютера. Почти каждый производитель вместе с контроллерами поставляет пакет программ для создания и отладки контроллерного ПО на компьютере. После отладки программ контроллеры могут сохранять их в энергонезависимых ПЗУ, из которых программа перегружается в ОЗУ после включения питания или инициализации контроллера.

Целью данного курсового проекта является рассмотрение задачи управления модели железной дороги.

Определение границ проблемной области решения об управлении движением поездов необходимо принимать автоматически, и производить контроль за всеми элементами железной дороги с помощью компьютера. Такие автоматические и полуавтоматические системы сегодня существуют в Швеции, Великобритании, Германии, Франции и Японии. Эффект от каждой из этих систем был и экономический, и социальный; результатом их внедрения стало снижение эксплуатационных затрат, повышение эффективности использования ресурсов, безопасность.

Система управления движением выполняет две главные функции: выбор маршрутов железнодорожных перевозок и контроль систем, обеспечивающих перевозки. Эти функции включают: планирование перевозок, контроль местонахождения поездов, контроль за перевозками, предотвращение конфликтов, регистрацию всех операций. На эксплуатационном уровне система управления движением может содержать сотни компьютеров: по одному на каждый поезд, по одному на каждый блок интерфейса путевых устройств и по два на каждый диспетчерский центр. Система управления движением должна взаимодействовать с разнообразными датчиками и исполнительными механизмами.

Из краткого проблемного анализа четырех главных подзадач мы видим, что существуют три высокоуровневые ключевые абстракции:

* маршрут
* длина маршрута
* время прохождения маршрута
* предотвращения столкновений
* регистрацию всех операций

Каждый поезд характеризуется текущим положением на путях и может иметь только один активный план движения. Аналогично, в каждой точке пути может быть самое большое один поезд. Каждый план относится только к одному поезду, но ко многим точкам пути.

Данную задачу рассмотрения модели железных дорого можно осуществить с помощью представления ее указателями. Эта задача относится к классу задач представления графа.

Для разработки данного курсового проекта необходимо

ЗНАТЬ:

- основные положения теории вероятностей, законы распределения и числовые характеристики случайных величин, законы распределения случайных событий на железнодорожном транспорте;

- основы математической статистики и математической обработки результатов наблюдений, методы определения параметров эмпирических формул, основы теории информации;

- содержание линейного и динамического программирования транспортных процессов;

- теорию сетевого планирования и управления;

- основные положения теории массового обслуживания и ее применения при решении задач, возникающих в процессе функционирования железнодорожных транспортных систем;

- процедуру математического моделирования транспортных процессов.

УМЕТЬ:

- обрабатывать статистические данные, проверять гипотезы о законах распределения случайных величин по критериям согласия, проводить корреляционный анализ;

- решать задачи оптимизации транспортных процессов с использованием линейного и динамического программирования;

- применять теорию вероятностей и теорию массового обслуживания при решении транспортных локальных задач;

- строить сетевые графики и линейные диаграммы транспортных процессов, рассчитывать их основные параметры;

- разрабатывать элементарные математические модели описывающие отдельные подсистемы транспортного комплекса.

БЫТЬ ОЗНАКОМЛЕН:

- с основами теории исследования операций при анализе функционирования и оптимизации процессов, происходящих на транспорте;

- с основами теории надежности транспортных систем;

- с возможностями использования ПЭВМ при решении оптимизационных транспортных задач, включая моделирование транспортных процессов;

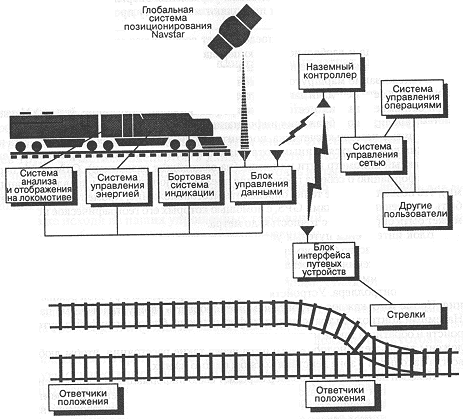
- с процедурой разработки программного обеспечения при решении транспортных задач.

В настоящее время имеется большое количество прикладных программ, позволяющих использовать их для автоматизированного проектирования систем. Одной из таких программ является прикладная программа MatLab+Simulink. Прикладная программа Simulink и её подпрограмма Control System Toolbox – инструментарий систем управления – предназначенный для моделирования, анализа и проектирования непрерывных автоматических систем. Пакет реализует методы исследования динамических систем, в основу которых положены функции и графические модели.

**1. Требования к системе управления движением**

Система управления движением выполняет две главные функции: выбор маршрутов железнодорожных перевозок и контроль систем, обеспечивающих перевозки. Эти функции включают: планирование перевозок, контроль местонахождения поездов, контроль за перевозками, предотвращение конфликтов, прогнозирование нарушении, регистрацию всех операций.

Рис.1. Система управления движением



Система анализа и отображения информации на локомотиве состоит из множества дискретных и аналоговых датчиков для контроля за такими параметрами, как температура, давление масла, количество топлива, напряжение и сила тока на генераторе, число оборотов вала двигателя в минуту, температура воды, тяговая мощность. Значения параметров с датчиков поступают к машинисту через дисплейную систему, а к диспетчеру и обслуживающему персоналу вне поезда - через сеть. Предупреждение или сигнал тревоги выдается и регистрируется всякий раз, когда показания датчика выходят за пределы нормального режима. Журнал показаний датчиков используется при проведении эксплуатационных работ и для управления расходом топлива.

Система управления энергией в режиме реального времени подсказывает инженеру поезда, как наиболее эффективно использовать установку. Входными данными для этой системы являются: профиль и качество пути, ограничения по скорости, расписание, загрузка поезда, максимальная развиваемая мощность. Исходя из этих данных, система может определить оптимальный по расходу топлива режим работы двигательных установок, согласующийся с заданным расписанием и требованиями безопасности. Рекомендации системы, профиль и качество пути, местоположение и скорость поезда могут отображаться с помощью бортовой системы индикации.

Бортовая система индикации обеспечивает человеко-машинный интерфейс для машиниста. На нее может выводиться информация из системы анализа и отображения информации на локомотиве, системы управления энергией и блока управления данными. Специальные клавиши позволяют машинисту просматривать различные данные.

Блок управления данными представляет собой шлюз между всеми бортовыми системами поезда и глобальной сетью передачи данных, к которой подключены все поезда, диспетчеры и прочие пользователи.

Отслеживание маршрутов движения поездов осуществляется с помощью подключенных к сети передачи данных ответчиков местоположения и глобальной спутниковой системы указания местоположения. Система анализа и отображения информации на локомотиве может вычислять пройденный путь с помощью счетчика, подсчитывающего число оборотов колеса. Эта информация дополняется данными ответчиков местоположения, которые размещены через каждый километр пути или чаще (на важнейших развилках). Ответчики передают информацию о себе на проходящие поезда (используя блок управления данными), что позволяет более точно определить местоположение.

Блок интерфейса путевых устройств размещается там, где есть какое-либо управляемое устройство (например, стрелка), или датчик (например, инфракрасный датчик для обнаружения перегрева подвесок колес). Каждый блок интерфейса получает команды (например, команды на включение и выключение сигнала) от локального наземного контроллера. Устройства могут быть переведены в ручной режим управления. Кроме того, каждое устройство может сообщать свои установочные параметры. Наземный контроллер транслирует информацию на блоки интерфейса путевых устройств и обратно, а также на проходящие мимо поезда и обратно. Контроллеры расположены вдоль железнодорожного пути через такие расстояния, чтобы любой поезд всегда находился в зоне действия хотя бы одного из них.

Каждый наземный контроллер передает свою информацию на объединенную систему управления сетью. Связь между системой управления сетью и наземным контроллером может осуществляться по радио в микроволновом диапазоне, по наземным линиям или по оптоволокну в зависимости от удаленности данного контроллера. Система управления сетью обеспечивает функционирование всей сети. Она может автоматически направлять информацию по другому маршруту в сети, если на одном из путей произойдет отказ оборудования.

Система управления сетью, в свою очередь, подсоединяется к одному или нескольким диспетчерским центрам, которые объединены в систему управления операциями. Система управления сетью соединена и с другими пользователями. В системе управления операциями диспетчеры могут задавать маршруты поездов и отслеживать их передвижение. Для управления различными участками выделяются отдельные диспетчеры; каждая диспетчерская управляющая консоль отвечает за одну или несколько территорий. Маршрутизация поездов подразумевает выдачу инструкций для автоматического перевода поезда с пути на путь, установку ограничения скорости, управление пропуском автомобилей на переездах, разрешение и запрещение движения поезда в зависимости от занятости определенных участков пути. Диспетчеры могут наблюдать за состоянием путей впереди по маршруту поезда и передавать эту информацию машинисту. Поезда могут быть остановлены системой управления операциями, когда обнаруживается опасность (выход поезда из графика, повреждение пути, возможность столкновения). Диспетчеры могут также вызвать на экран любую информацию, доступную машинистам отдельных поездов, разослать распоряжения по движению, установить параметры путевых устройств и пересмотреть план движения.

Расположение путей и путевое оборудование могут со временем меняться. Число поездов и маршруты их движения могут изменяться ежедневно. Система должна обеспечивать возможность подключения новых датчиков, сетей и оборудования, выполненных по более совершенным технологиям.

На врезке сформулированы основные требования к системе управления движением поездов. Очевидно, они сильно упрощены. На практике детальные требования к большой системе вырабатываются после демонстрации жизнеспособности программного решения проблемы. При этом анализ отменяет сотни человеко-месяцев труда с участием экспертов в данной области и пользователей системы. В конечном счете требования к системе могут состоять из тысяч страниц документации, специфицирующей не только базовое поведение, но и такие детали, как макеты форм интерфейса.

В данном курсовом проекте мы рассмотрим не все требования, а только их малую часть.

**2. Постановка задачи**

Спроектировать на основе микропроцессора контроллер для модели железной дороги. Рассмотреть путь, состоящий из замкнутого кольца и тупиков со стрелками на разветвлениях, причем считайте, что на путях находиться один поезд с мотором постоянного тока. Контроллер должен управлять маршрутами поезда в соответствии с поступающими с клавиатуры командами. Должны быть предусмотрены, например, такие команды:

1. Проследовать кратчайшим маршрутом из данной точки в указанный пункт назначения и остановиться.
2. Уйти в тупик.
3. Выдать длину маршрута.

Начальное положение поездов можно считать фиксированным и известным микропроцессору. В дальнейшем он должен следить за положением поезда.

Исходные данные:

1. количество остановок на пути
2. расстояние между остановками
3. начальная координата поезда
4. конечная координата поезда
5. тупик (расстояние до тупика)
6. место положения стрелки

**3. Первый этап моделирования. Формирование модели**

На первом этапе моделирования формулируется модель, строится ее формальная схема и собственно решается вопрос об ее эффективности и целесообразности моделирования системы на вычислительной машине.

Рассмотрим путь, состоящий из замкнутого кольца и тупиков со стрелками на разветвлениях, причем считаем что на путях находится один поезд с мотором постоянного тока. Спроектируем на основе микропроцессора контроллер для модели железной дороги.

Для того, чтобы определить положение поездов и управлять их движением, все пути разбиваются на участки. Присутствие состава на участке можно определить по потребляемому этим участком току. Для этого достаточно в цепь питания рельса включить резистор. Если есть потребление на участке, то на резисторе возникает падение напряжения, которое с помощью специального усилителя будет превращено в логический сигнал. В нашей модели присутствие поезда на данном участке будет определяться по метке.

В интерфейс между микропроцессором и железной дорогой будут входить порт вывода для подачи напряжения на каждый участок и бит порта ввода для управления стрелкой.

Начальное положение поездов можно считать фиксированным и известным микропроцессору. В дальнейшем он должен следить за их положением, управляя их перемещениями, в соответствии с поступающими с клавиатуры командами. Программа микропроцессора должна уметь находить кратчайшие маршруты.

Графически наше замкнутое кольцо с тупиками можно представить в виде взвешенного графа.

1u

3

2

1

6

5

4

w

e

r

t

y

x

Рис.1

6u

2u

3u

4u

5u

Эффективность и целесообразность моделирования системы представленной графом, дает возможность рассмотреть полностью структуру системы движения поезда по пути, не запуская его экспериментально на самом деле, и связи между элементами системы. Обход данного графа, возможно, провести несколькими способами расчета: аналитическим, имитационным. Аналитическое моделирование-процесс функционирования элементов системы записываются в виде некоторых функциональных соотношений или логических условий. Имитационное моделирование реализует модель, алгоритм воспроизводит процесс функционирования системы во времени, имитируются элементарные явления, составляющие процесс, с сохранением их логической структуры и последовательности протекания во времени, что позволяет по исходным данным получит сведения о состояниях процесса в определенные моменты времени, дающие возможность оценить характеристики системы. Имитационный метод включает все элементы реальной системы.

Методологическую основу построения маршрутов составляет расширенная граф-модель, реализующая событийное управление, позволяющая проводить адаптацию алгоритмов построения маршрутов, основанную на формальных манипуляциях с объектами без изменения исходной граф-модели. Для графа зависимости по управлению  на каждом конкретном шаге процесса управления для узла, из которого исходят несколько дуг, возникновение коллизии на одной из них определяет выбор стратегии (алгоритма) управления. Активизация некоторого события зависит от состояния транспортной инфраструктуры, определяемого ее текущими характеристиками.

Введем множество предикатов  для реализации событийного управления на графе управлений  и поставим их в соответствие дугам графа . Алгоритм, реализующий переход → на графе , инициируется, если объект на текущем шаге находится в состоянии , и предикат , помечающий данный переход, принимает «разрешающее» значение. При этом на дугах, исходящих из одной вершины, возможна ситуация появления значения истинности одновременно нескольких предикатов.

Вершины данного графа 1, 2, 3, 4, 5, 6 есть остановки; ребра-ветви железной дороги; e, r,t,y,x,w,1u,2u..- длина пути ветвей; ответвления от замкнутого кольца – тупики, вместе ответвления расположена стрелка. В данном курсовом проекте рассмотрена упрощенная схема пути, но на ее основе можно будет рассмотреть модернизацию в сторону усложнения пути и добавления поездов.

Процесс функционирования системы может быть реализован с помощью различных алгоритмов.

Построение математической модели. Под математической моделью будем понимать процесс установления соответствия данному реальному объекту некоторого математического объекта и исследование этой модели, позволяющее получать характеристики рассматриваемого реального объекта. Модель объекта содержит:

* совокупность входных воздействий - задача начальной и конечной точки отправления;
* совокупность воздействий внешней среды;
* совокупность внутренних параметров - длины путей(скорость, свободна или занята ветвь по которой собирается двигаться поезд);
* совокупность выходных характеристик – длина всего пройденного пути (время его прохождения, и вывод кратчайшего пути).

1. Задаем длины отдельных ветвей графа, они могут быть представлены вектором:

P[ e, r,t,y,x,w], все элементы больше нуля.

1. Задаем начальную координату или вершину отправления, она должна быть истинной для заданного графа: номер вершины – nach, где 1<=nach<=6 т.е. для нашего графа число натуральное целое не более числа вершин.
2. Задаем конечную координату или вершину прибытия: номер вершины – kon, где 1<=kon<=6, натуральное и целое.
3. Задаем длину до тупика: u, где u=1u=2u=…=6u, u>0.
4. Выбери направление обхода: по часовой стрелки или против часовой.
5. Ищем длину пути: алгоритм – по очередное суммирование всех длин пройденных ветвей, т.е.элементов вектора Р

S=[nach]+[nach+1]+…+[nach+n], где nach+n=kon.

1. Задаем уйдет ли поезд в тупик. Если уходит то к общему полученному пути прибавляем длину ветви тупика. Skon=S+u, где u элемент из вектора U[1u,2u,3u,4u,5u,6u]
2. Задаем среднюю скорость прохождения маршрута поездом: V=v.
3. Вычисляем время, затраченное на весь маршрут: T= Skon/v

**4. Второй этап моделирования. Реализация машинной модели**

На втором этапе математическая модель воплощается в машинную модель, т.е. решается проблема алгоритмизации модели, ее рационального разбиения на блоки и организации интерфейса между ними. Реализация алгоритма представлена на языке турбо паскаль.

**4.1 Блок схема**

Основная схема

m1:

начало

nash>n

goto m1

m2:

kon>n

gotom2

Неверный ввод

kon

Неверный ввод

нет

да

да

нет

nash

new(tek)

perv:=tek

kon:=1

perv^.zn:=false

perv^.zn:=true

tek^.s

tek^.zln:=nil

tek^.lln:=nl

pzed:=perv

pzev1:=perv

i:=2, n

new(tek)

pzed^.Rln:=tek

tek^.lln:=pzed

tek^.s

tek^.Rln:=nil

pzed:=tek

i=kon

pezv1:=tek

i=nash

Tek^.zn:=false

Tek^.zn:=true

dl

да

нет

да

нет

да

нет

tek^.Rln:=pezv

pzed^.:=tek

m3:

zn

zn

poisk1

Poisk2

‘Y’,y

‘N’,n

m3:

znash

Неверный ввод

else

tek:=pezv1

znash:=0

tek^.zn:=false

znash:=tek^.s+znash

tek:=tek^.zln

znash:=znash+dl

tek:=pezv1

znash:=0

cx:=0

tek^.zn:=false

znash:=tek^.lln^.s+znash

tek:=tek^.lln

cx:=xx+1

znash:=znash+dl

Poisk1:

Poisk2:

**4.2 Листинг программы**

|  |  |
| --- | --- |
| program u;  uses crt;  label m1,m2,m3;  const n=5;  type  uk=^zap;  zap=record  s:integer;  zn:boolean;  Lln:uk;  Rln:uk;  end;  var  zn:char;  perv,tek,pred,perv1:uk;  met:boolean;  i,nash,kon,znash,dl:integer; | Блок описаний: Задание названия программы, задание существующих меток, констант (n=5) – задание точного числа остановок на пути, задание типов переменных: целочисленных данных, логических переменных, логические метки, символьные переменные. |

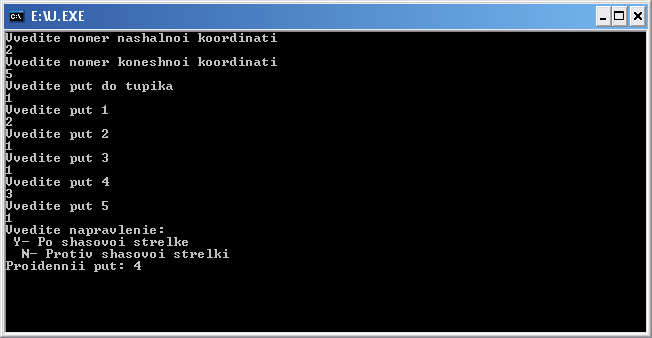
|  |  |
| --- | --- |
| procedure poisk1;  begin  tek:=perv1;  znash:=0;  while tek^.zn=false do  begin  znash:=tek^.s+znash;  tek:=tek^.rln;  end;  writeln(' ');  znash:=znash+dl;  end; | Процедура поиска и задания начальной координаты, и нахождения суммарного пути по часовой стрелке. |

|  |  |
| --- | --- |
| procedure poisk2;  var  cx:integer;  begin  tek:=perv1;  znash:=0;  cx:=0;  tek:=perv1;  while tek^.zn=false do  begin  cx:=cx+1;  znash:=tek^.lln^.s+znash;  tek:=tek^.lln;  end;  znash:=znash+dl;  end; | Процедура поиска и задания начальной координаты, и нахождения суммарного пути против часовой стрелки. |

|  |  |
| --- | --- |
| begin  clrscr;  m1:writeln('Vvedite nomer nashalnoi koordinati');readln(nash);  if nash>n then  begin  writeln('Nevernii vvod');  goto m1;  end;  m2:writeln('Vvedite nomer koneshnoi koordinati');readln(kon);  if kon>n then  begin  writeln('Nevernii vvod');  goto m2;  end;  writeln('Vvedite put do tupika');readln(dl); | Вводим начальные координаты  Если начальная координата больше общего числа возможных остановок на пути, выводится ошибка.  Ввод конечной координаты  Если конечная координата больше общего числа возможных остановок на пути, выводится ошибка.  Ввод расстояния до тупика |

|  |  |
| --- | --- |
| new(tek);  perv:=tek;  if kon=1 then perv^.zn:=true else perv^.zn:=false;  writeln('Vvedite put 1');readln(tek^.s);  tek^.Rln:=nil;  tek^.Lln:=nil;  pred:=perv;  perv1:=perv;  for i:=2 to n do  begin  new(tek);  pred^.Rln:=tek;  tek^.lln:=pred;  writeln('Vvedite put ',i);readln(tek^.s);  tek^.Rln:=nil;  pred:=tek;  if i=nash then perv1:=tek;  if i=kon then tek^.zn:=true else tek^.zn:=false;  end;  tek^.Rln:=perv;  perv^.Lln:=tek; | Здесь задается вопрос о величине каждой в отдельности ветви.  Задание величины длины пути до тупика при переходе по стрелке |

|  |  |
| --- | --- |
| m3:writeln('Vvedite napravlenie:');  writeln(' Y- Po shasovoi strelke');  writeln(' N- Protiv shasovoi strelki');  zn:=readkey;  case zn of  'Y','y':poisk1;  'N','n':poisk2;  else  begin  writeln('Nevernii vvod,povtorite ese ras');  goto m3;  end;  end;  writeln('Proidennii put: ',znash);  readkey;  end. | Вводится направления движения которое считается более оптимальным, то есть задается движение поезда по часовой стрелки или против часовой.  Y-по часовой  X-против часовой  При ошибки ввода неправильного значения выдается просьба ввода еще раз.  Вывод длины пройденного пути |



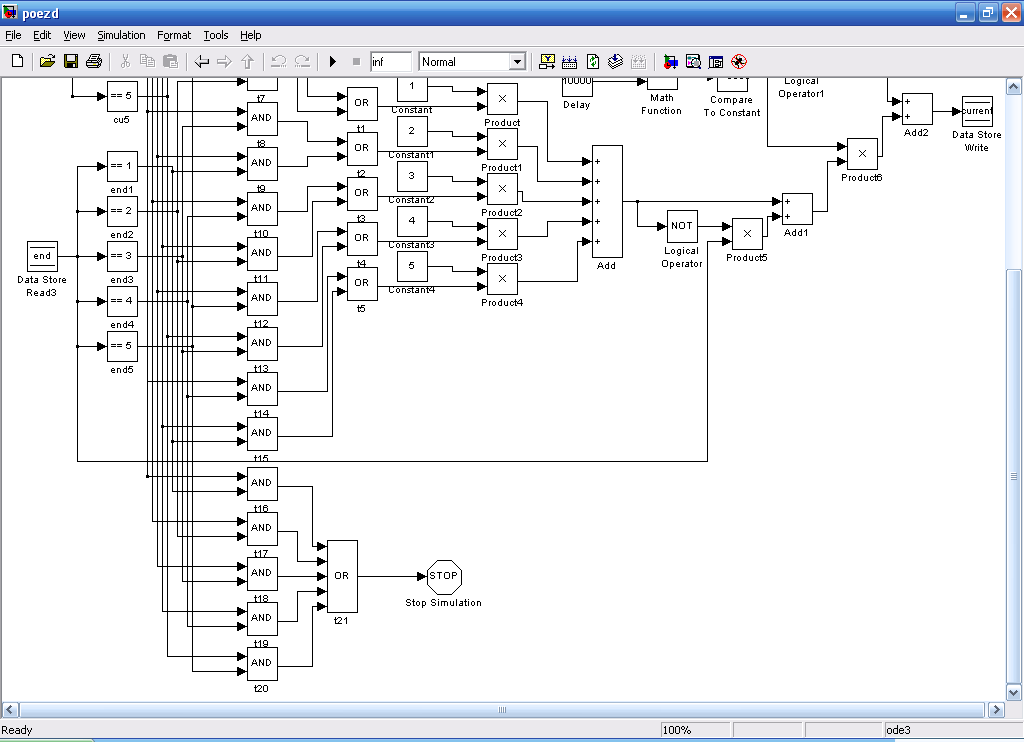
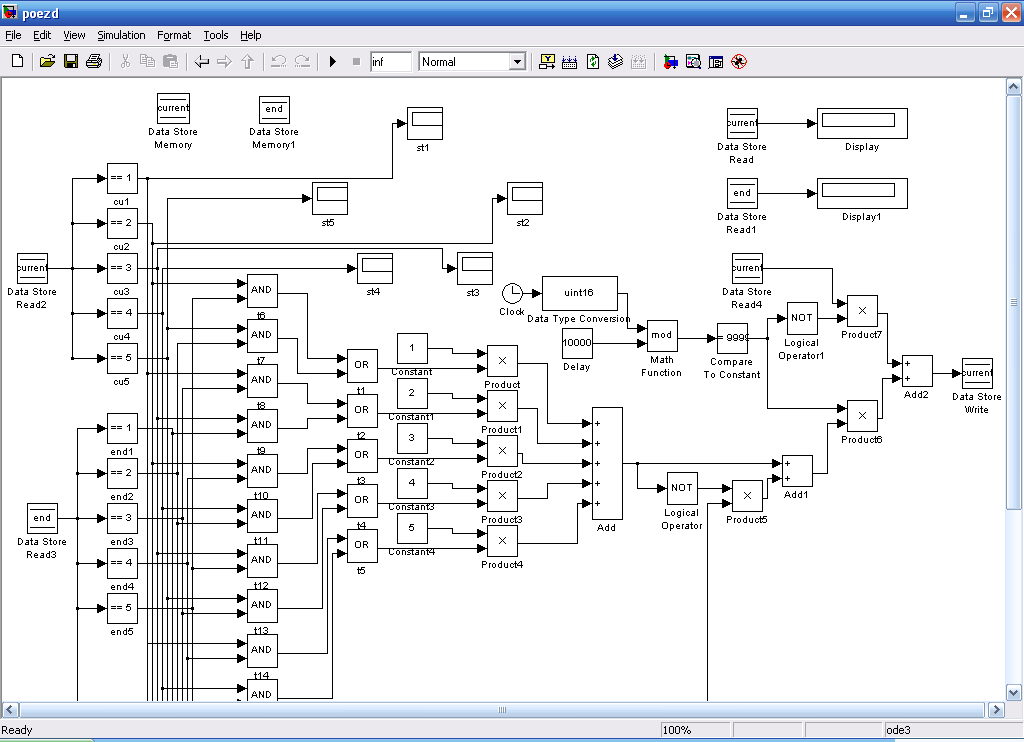
(Рис.2)

Это результат работы программы, в которой рассмотрена задача нахождения пути, переключение направления движения, а также работа стрелки.

**5. Модель движения поезда рассмотренная в MatLab**

Для того чтоб знать расположение в данный момент поезда на маршруте, используем пакет Simulink в программе MatLab. Рассмотрим граф с переходящей меткой, стоящей в той вершине, в которой находиться поезд в конкретный момент времени. Данная модель является динамическим объектом. Динамический объект – это объект, поведение (выход) которого зависит не только от текущего значения входных воздействий (сигналов), но и от их значений в предыдущие моменты времени. Модель находит кратчайший путь и идет по нему. То есть в данную модель включено цифровое устройство – счетчик времени.

Модель обнаружения поезда на путях (логическая схема).



Данная логическая схема обеспечивает удобный и наглядный визуальный контроль за поведением виртуальной модели железной дороги. Достаточно лишь запустить Simulink в режиме симуляции созданной модели системы. После запуска программа определяет начальную и конечную станцию, находит кратчайший маршрут движения, запускается счетчик времени, и производиться перемещения метки поезда по вершинам до конечной станции. Для изменения конечной и начальной станции достаточно редактировать лишь два блока: Data Store Read – начальная вершина, Data Store Read1- конечная вершина

Входные данные к логической схеме:

е1 — конечнная станция 1

е2 — конечнная станция 2

е3 — конечнная станция 3

е4 — конечнная станция 4

е5 — конечнная станция 5

с1 — текущая станция 1

с2 — текущая станция 2

с3 — текущая станция 3

с4 — текущая станция 4

с5 — текущая станция 5

s — конечная станция.

Поступают на входы в виде двоичного сигнала

Логическая функция реализованной модели

с1 = е1  ( c2 v c5 ) v e2  c5 v e5 c2

с2 = е2  ( c1 v c3 ) v e1 c3 v e3 c1

с3 = е3  ( c2 v c4 ) v e2  c4 v e4 c2

с4 = е4  ( c3 v c5 ) v e3 c5 v e5  c3

с5 = е5  ( c4 v c1 ) v e4  c1 v e1  c4

s = c1  e1 v c2  e2 v c3  e3 v c4  e4 v c5  e5

**Заключение**

В ходе выполнения курсового проекта по теме модель железной дороги были реализованы поставленные задачи прохождения поездом железно дорожного пути. Сделанная виртуальная модель способна проходить путь от заданной начальной, до заданной конечной точки за определенное время, переключать стрелку, находить кратчайший маршрут движения и считать длину пройденного пути. Разработанная виртуальная модель помогает в наблюдении и изучении движения поезда. Усложнения данной модели может идти в сторону увеличения числа поездов, появление шлагбаумов.

Использование данного проекта может быть реализовано в управлении движения поездов. В главе Требования к системе управления движением, рассмотрены основные проблемы и данная работа поможет в решении некоторых из них.

**Список использованной литературы**

1. Винер Н. Кибернетика, или управление и связь в животном и машине. М.: «Советское радио».1958. 215с.
2. Ицкович Э.Л. Как выбирать контроллерные средства. http://archive.expert.ru
3. Дьяконов В.П. SIMULINK 5/6/7 . М.:«ДМК Пресс», 2008
4. Моргун А.Н. Справочник по Turbo Pascal для студентов. М.: «Диалектика», 2006.
5. Эллиот Б. Коффман Turbo Pascal. М.: «Вильямс», 2005.
6. http://pascalstudy.narod.ru/
7. http://www.pascalguru.com/