

РЕАЛИЗАЦИЯ ЦИФРОВОЙ МОДЕЛИ ПЛОСКОГО ДВИЖЕНИЯ АВТОМОБИЛЯ НА БАЗЕ MATLAB REALTIME WINDOWS TARGET

Ульянов Р.В.

*ФГАОУ ВПО Казанский (Приволжский) федеральный университет,
420008, г. Казань, ул. Кремлевская, д.18*

e-mail: lntyrd@gmail.com

поступила в редакцию 11 декабря 2014 года

Аннотация

В работе рассмотрены различные методы обеспечения функционирования цифровых моделей в режиме реального времени на примере модели плоского движения грузового автомобиля, также проведен сравнительный анализ этих методов.

Ключевые слова: математическая модель, цифровая модель, динамическая система, режим реального времени, движение автомобиля.

Целью работы является исследование способов обеспечения функционирования цифровых моделей динамических систем в режиме реального времени.

Предметом исследования является математическая модель и её цифровая реализация, описывающая процесс плоского движения грузового автомобиля в режиме реального времени.

Моделирование систем в режиме реального времени является актуальной задачей, применяется в инженерной деятельности для исследований и испытаний технических систем.

Система реального времени (СРВ) – Любая система, в которой существенную роль играет время генерации выходного сигнала. Это обычно связано с тем, что входной сигнал соответствует каким-то изменениям в физическом процессе, и выходной сигнал должен быть связан с этими же изменениями. Временная задержка от получения входного сигнала до выдачи выходного сигнала должна быть небольшой, чтобы обеспечить приемлемое время реакции. Примерами систем реального времени являются системы управления физическими процессами с применением вычислительных машин, системы торговых автоматов, автоматизированные системы контроля и автоматизированные испытательные комплексы [1].

Цифровая модель плоского движения грузового автомобиля реализована в среде MATLAB Simulink, в связи с этим в данной работе рассмотрены 2 способа обеспечения функционирования этой модели в режиме реального времени:

1. Замедление времени моделирования и приближение его к реальному при помощи S-функции;

2. Генерация программного кода на языке C из Simulink модели при помощи программного продукта Realtime windows target.

Для достижения цели работы необходимо решить следующие задачи:

1. Разработать цифровую модель плоского движения автомобиля на основе соответствующей математической модели;

2. Обеспечить функционирование цифровой модели в режиме реального времени за счет S-функции;

3. Обеспечить функционирование цифровой модели в режиме реального времени за счет программного продукта Realtime windows target;

4. Провести сравнительный анализ методов.

Описание математической модели плоского движения автомобиля.**Допущения и ограничения.**

- Каждое колесо имеет свою линейную и угловую скорость и рассчитывается отдельно.
- При повороте учитывается перераспределение нормальных реакций между наружными и внутренними колёсами.
- При повороте учитывается крен автомобиля вокруг продольной оси симметрии.
- Учитывается динамика подвески: она моделируется упругим элементом с демпфированием. При этом рассматривается только поворот вокруг оси X (прочие повороты и деформации считаются малыми и не учитываются).
- Моделируется работа межколёсного дифференциала на заднем мосту. Дифференциал считается коническим симметричным, блокировка дифференциала не рассматривается и не моделируется.

Уравнения модели. Уравнения динамики корпуса автомобиля выражаются следующими дифференциальными уравнениями:

$$m \frac{dv_x}{dt} = R_{x21} + R_{x22} + R_{x11} \cos \theta + R_{x12} \cos \theta - R_{y11} \sin \theta - R_{y12} \sin \theta - kSv_x^2 + mv_y \omega_z,$$

$$m \frac{dv_y}{dt} = R_{y21} + R_{y22} + R_{x11} \sin \theta + R_{x12} \sin \theta + R_{y11} \cos \theta + R_{y12} \cos \theta - mv_x \omega_z,$$

$$J_z \frac{d\omega_z}{dt} = -(R_{y21} + R_{y22})l_2 - (R_{x21} - R_{x22})\frac{b}{2} + (-R_{x11} \cos \theta + R_{y11} \sin \theta)\frac{a}{2} +$$

$$+ (R_{x11} \sin \theta + R_{y11} \cos \theta)l_1 + (R_{x12} \cos \theta - R_{y12} \sin \theta)\frac{a}{2} + (R_{x12} \sin \theta + R_{y12} \cos \theta)l_1.$$

Уравнения, описывающие вращательное движение ведомых колес, имеют вид:

$$J_k \frac{d\omega_{k1}}{dt} = -R_{x11}r - R_{z11}fr,$$

$$J_k \frac{d\omega_{k2}}{dt} = -R_{x12}r - R_{z12}fr.$$

Совокупность уравнений для описания вращения ведущих колес имеет вид:

$$\left(\frac{2J_k + J_{mp} + u_{mp}^2 n_{mp} J_{\delta s}}{r} \right) \frac{d\omega}{dt} = \frac{\beta M_e u_{mp} n_{mp}}{r} - R_{x21} - R_{x22} - fR_{z21} - fR_{z22},$$

$$2J_k \frac{d\omega^*}{dt} = (R_{x21} - R_{x22})r + (R_{z21} - R_{z22})fr,$$

$$\omega_{z1} = \omega - \omega^*, \quad \omega_{z2} = \omega + \omega^*.$$

Продольное и боковое ускорения корпуса автомобиля вычисляются по формулам:

$$a_x = \frac{dv_x}{dt} - v_y \omega_z; \quad a_y = \frac{dv_y}{dt} + v_x \omega_z.$$

Боковой крен корпуса при наличии боковых ускорений описывается следующими уравнениями:

$$\frac{d\lambda}{dt} = \omega_\lambda,$$

$$J_\lambda \frac{d\omega_\lambda}{dt} = -(b_{\lambda 1} + b_{\lambda 2})\omega_\lambda - (c_{\lambda 1} + c_{\lambda 2} - m_n g h_\lambda)\lambda + h_\lambda m_n a_y.$$

Моменты сопротивления деформации подвески для передней и задней осей представлены уравнениями:

$$M_1 = b_{\lambda 1}\omega_\lambda + c_{\lambda 1}\lambda; \quad M_2 = b_{\lambda 2}\omega_\lambda + c_{\lambda 2}\lambda.$$

Нормальные реакции дороги для всех колес рассчитываются по формулам:

$$R_{z11} = \frac{R_{z1}}{2} - \frac{(mR_{z1}r + m_n R_{z1}(h_\lambda - r))a_y + mgM_1}{mga},$$

$$R_{z12} = \frac{R_{z1}}{2} + \frac{(mR_{z1}r + m_n R_{z1}(h_\lambda - r))a_y + mgM_1}{mga},$$

$$R_{z21} = \frac{R_{z2}}{2} - \frac{(mR_{z2}r + m_n R_{z2}(h_\lambda - r))a_y + mgM_2}{mgb},$$

$$R_{z22} = \frac{R_{z2}}{2} + \frac{(mR_{z2}r + m_n R_{z2}(h_\lambda - r))a_y + mgM_2}{mgb},$$

$$R_{z1} = \frac{h_g l_2 - h_g k S v_x^2 - m a_x h_c}{l_1 + l_2},$$

$$R_{z2} = \frac{h_g l_1 + h_g k S v_x^2 + m a_x h_c}{l_1 + l_2}.$$

Программная реализация математической модели. Реализация рассматриваемой модели в среде Simulink частично представлена на рисунках 1-6.

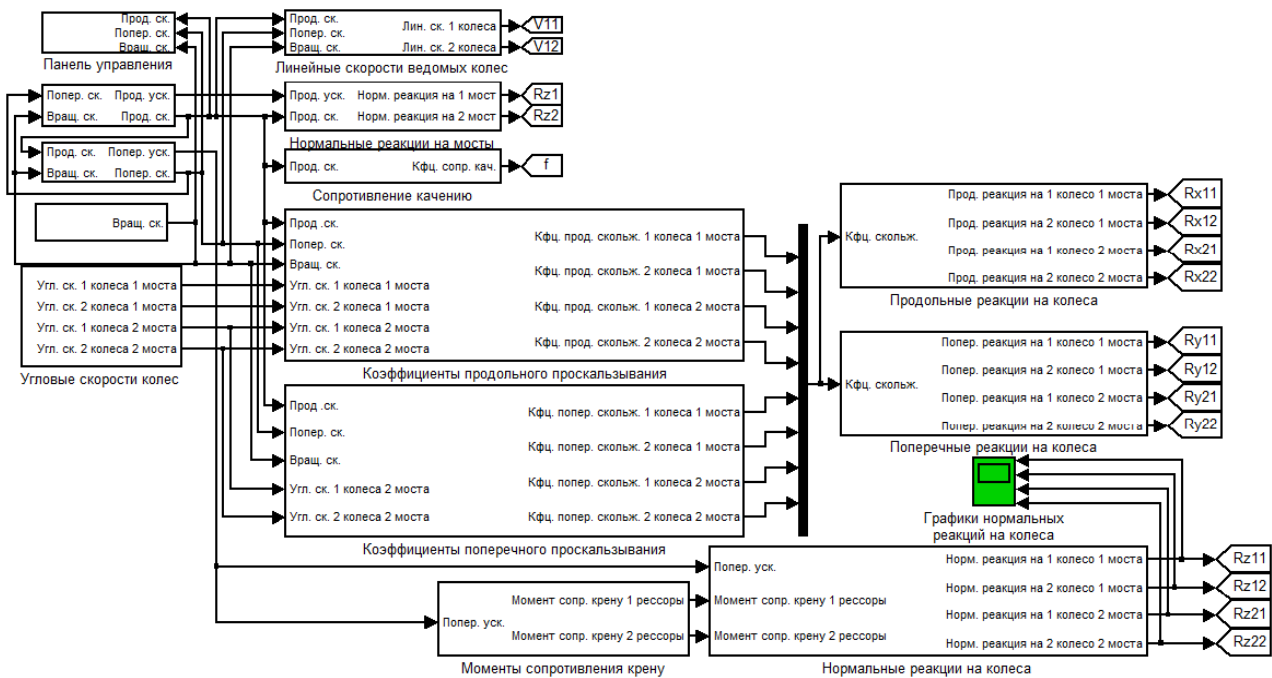


Рисунок 1. – Модель движения автомобиля с проскальзыванием колёс.

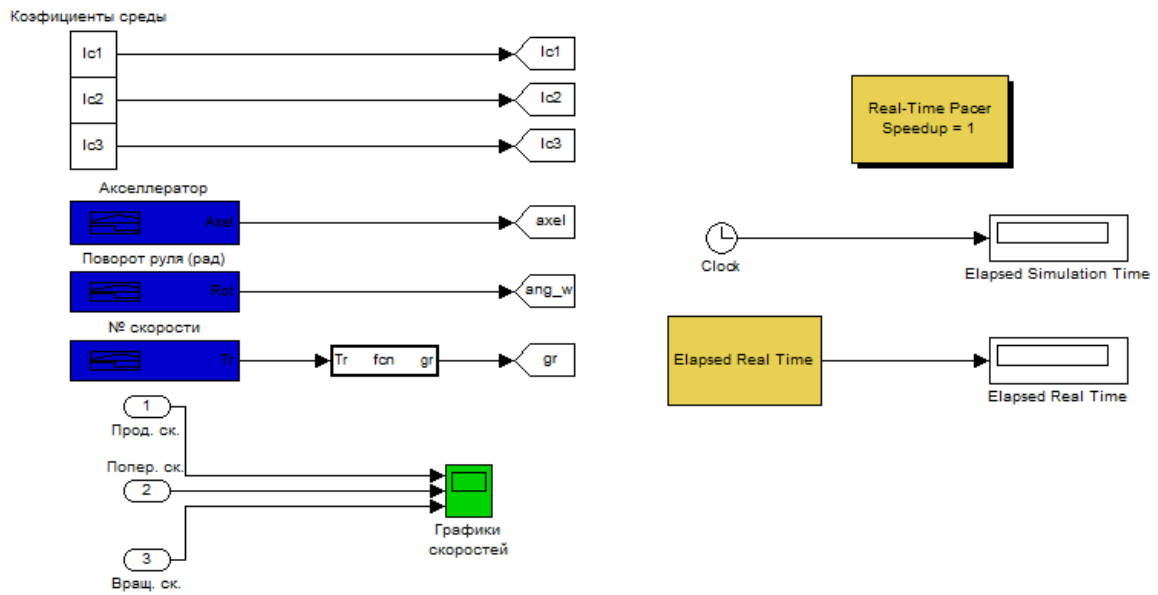


Рисунок 2. – Панель управления.

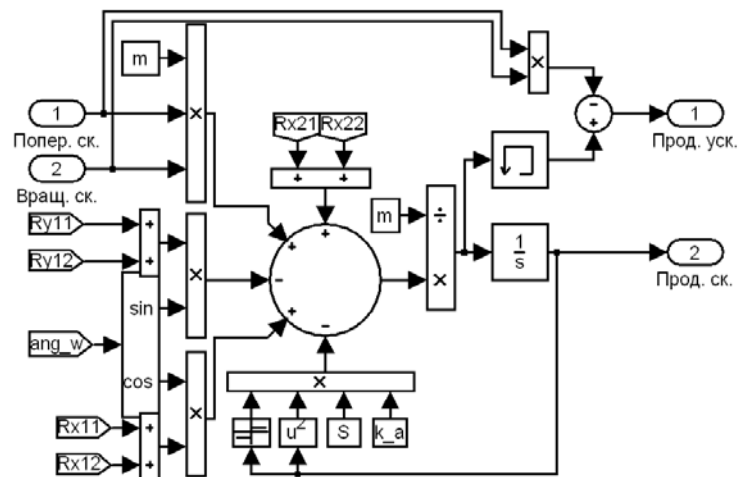


Рисунок 3. – Программная реализация алгоритма расчёта продольной скорости центра масс автомобиля.

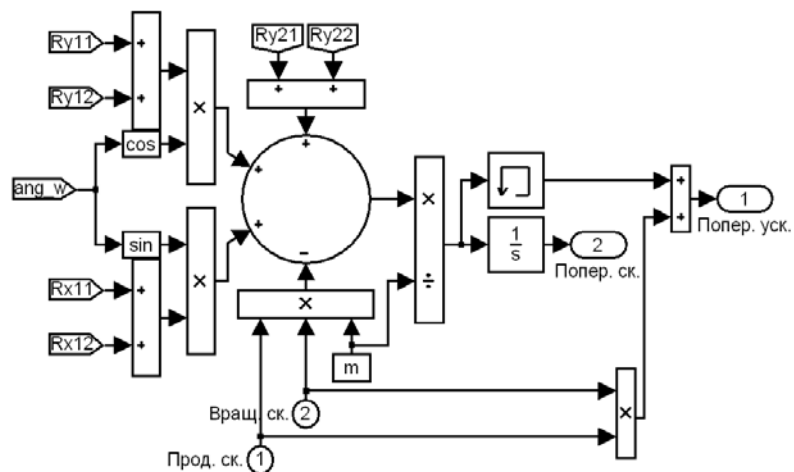


Рисунок 4. – Расчёт поперечной скорости центра масс автомобиля.

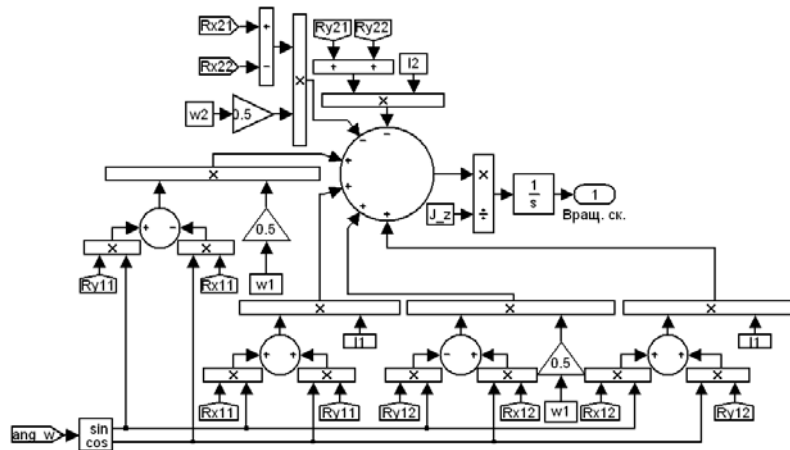


Рисунок 5. – Программная реализация алгоритма расчёта угловой скорости центра масс автомобиля.

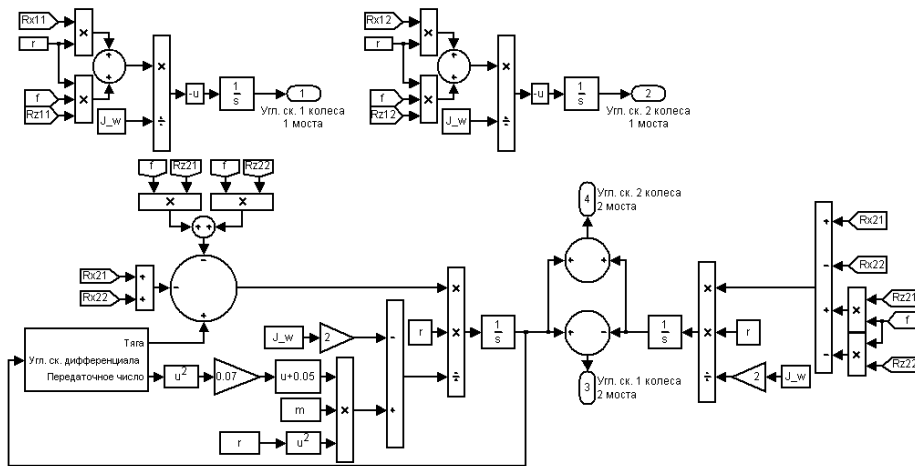


Рисунок 6. – Программная реализация алгоритма расчёта угловых скоростей колёс.

Обеспечение функционирования цифровой модели в режиме реального времени.

Использование S-функции RealTime Pacer. Принцип работы этой функции заключается в том, что на начальном этапе моделирования вычисляется, сколько времени требуется на один шаг интегрирования модели, затем она замедляет время моделирования, создавая паузы. Таким образом, она приближает модельное время к реальному. Если же время моделирования изначально отстаёт от реального, никакого эффекта эта функция не даст. На рисунке 7 показана разница между временем моделирования и реальным.

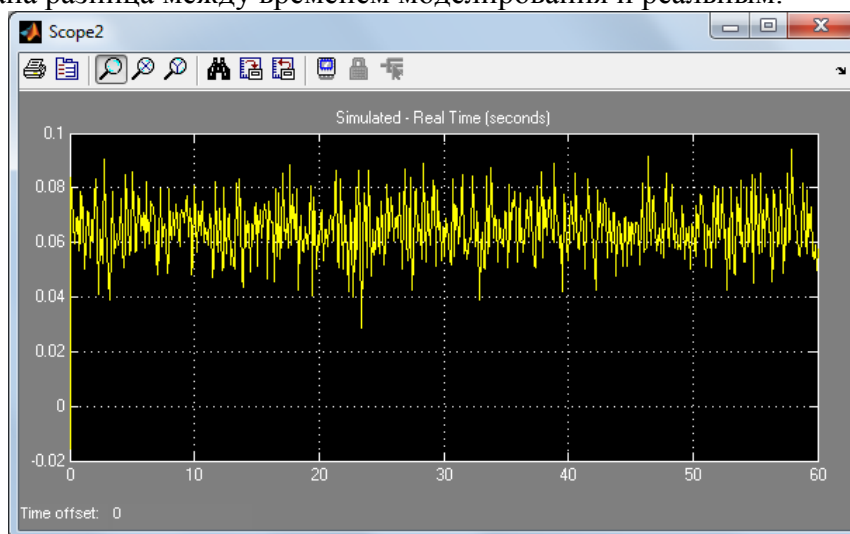


Рисунок 7. – Разница между временем симуляции и реальным временем в течении одной минуты моделирования.

Это называется режимом квазиреального времени (soft real time), когда допустимы небольшие рассогласования между временем моделирования и реальным временем [1].

Использование RealTime Windows Target. Real-Time Windows Target – это специальный программный продукт, обеспечивающий выполнение модели Simulink в реальном времени непосредственно на ОС Windows для быстрого прототипирования или программно-аппаратного тестирования систем управления и обработки сигналов [4].

С помощью этого инструмента был сгенерирован код на языке C:

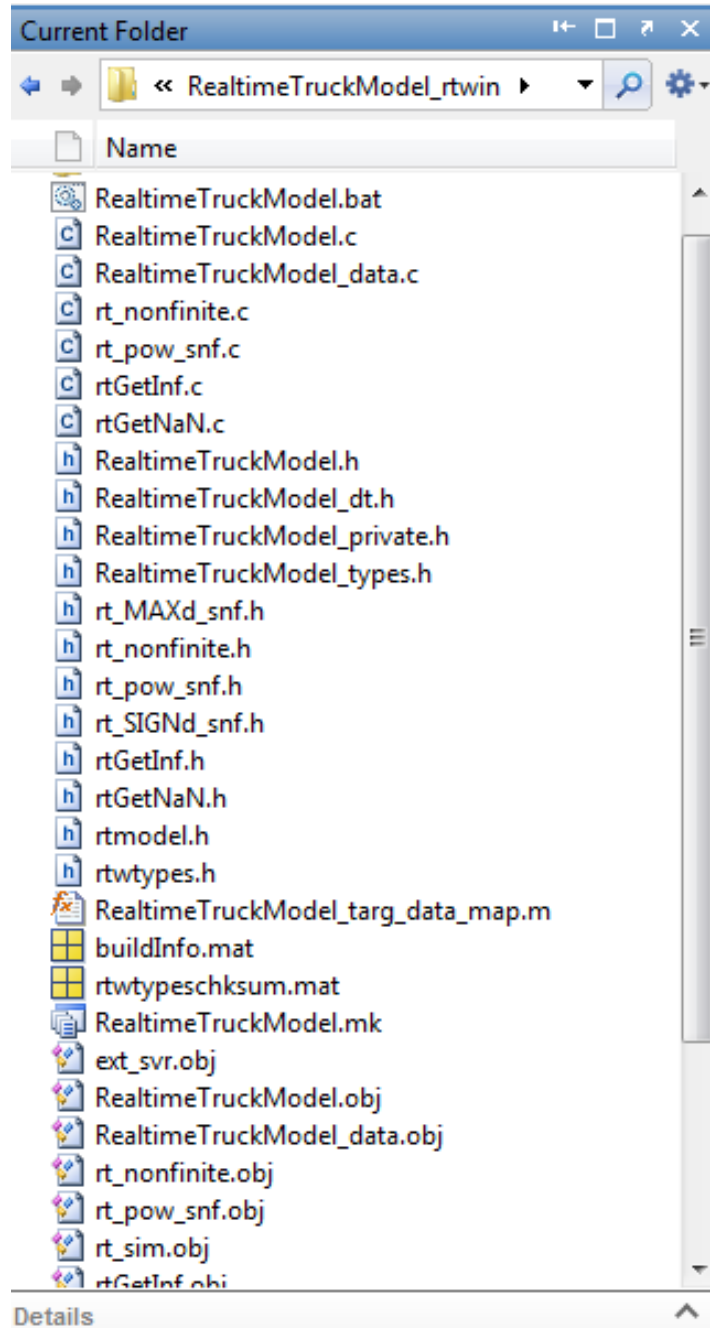


Рисунок 8. – файлы с программным кодом, сгенерированные при помощи Realtime Windows Target.

Запустить этот код можно также при помощи Simulink, либо собрав его в один проект в Microsoft visual studio, можно также использовать другой компилятор. На рисунке 9 изображена разница между временем моделирования и реальным.

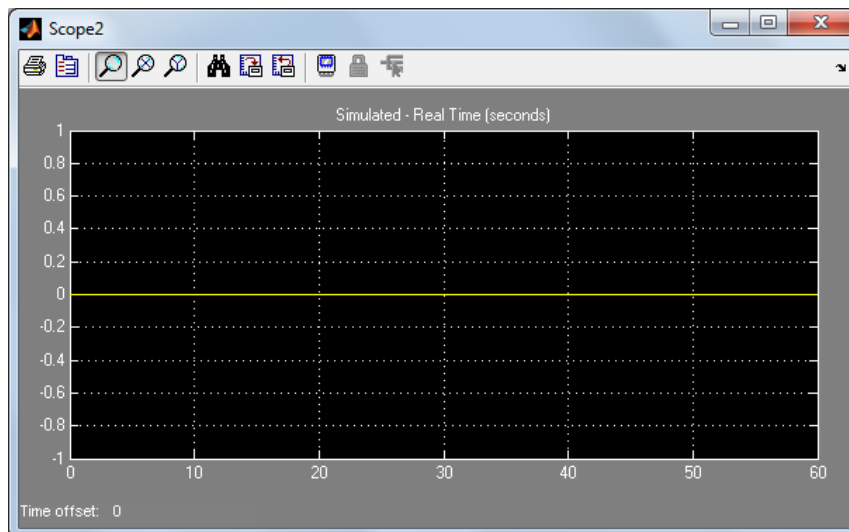


Рисунок 9. – Разница между модельным временем и реальным.

Как видно из графика, рассогласование между величинами здесь намного меньше, время моделирования практически соответствует реальному. Однако такой метод требует намного больше вычислительных ресурсов, чем предыдущий, и также не даёт никакого эффекта при слишком сложных моделях, в которых время моделирования отстаёт от реального.

Моделирование систем в режиме реального времени является актуальной задачей, применяется в инженерной деятельности для исследований и испытаний технических систем.

В результате выполнения работы были успешно решены следующие задачи:

1. Изучена математическая модель плоского движения грузового автомобиля;
2. Разработана Simulink модель плоского движения грузового автомобиля;
3. Проведены исследования по вычислению разработанной цифровой модели в режиме реального времени различными способами.

Исходя из полученных результатов, делается вывод, что использование Realtime Windows Target для обеспечения функционирования цифровой модели в режиме реального времени приемлемее, чем использование S-функции.

В перспективе созданная цифровая модель может получить развитие по следующим направлениям:

1. Добавление графической реализации;
2. Добавление возможности менять входные сигналы во время работы модели;
3. Замена плоской поверхности на трёхмерную.

Список литературы

- 1) Интернет-ресурс: Что такое функционирование в "Реальном масштабе времени". http://old.ci.ru/inform15_97/kovrigin.htm.
- 2) Зотов В.М., Зотов Н.М., Федин А.В. Проблемы численного моделирования динамических процессов в реальном времени и возможные пути их решения // Математика. Компьютер. Образование: сб. науч. трудов. М.: 2000. Вып.7, ч.2. С.597-603.
- 3) Муха Ю.П., Авдеюк О.А. Конспект лекций по системам реального времени. Часть 1: Учебное пособие. Волгоград: Изд-во ВолгГТУ, 1999. 80 с.
- 4) Интернет-ресурс: Real-Time Windows Target – моделирование в реальном времени для Simulink. <http://matlab.ru/products/real-time-windows-target>.