РАЗРАБОТКА КОНЦЕПЦИИ АЛГОРИТМА УПРАВЛЕНИЯ БЕСПИЛОТНОГО КОЛЁСНОГО ТЯГАЧА, ДВИЖУЩЕГОСЯ В КОЛОННЕ ЗА НАПРАВЛЯЮЩИМ

DEVELOPMENT OF CONTROL ALGORITHM CONCEPT FOR CONVOY-LED DRIVERLESS WHEEL TRUCK

УДК 004.942

Калинин Алексей Владимирович, сотрудник кафедры "Специальная робототехника и мехатроника" МГТУ им. Н.Э. Баумана, г. Москва

Малая Анастасия Геннадьевна, магистрант, Московский государственный технический университет имени Н.Э. Баумана, г. Москва

Kalinin A.V. kalinin_mvtu@mail.ru

Malaia A.G. malayaag@student.bmstu.ru

Аннотация

В данной статье разработана концепция алгоритма управления беспилотного колёсного тягача, движущегося в колонне за направляющим, на основе Pure Pursuit Controller. Для проверки и симуляция разработанного алгоритма была создана модель системы с реализованным алгоритмом управления в программной среде MATLAB, а также проведена визуализация в среде моделирования Unreal Engine от Epic Games.

Annotation

The article considers a new control algorithm concept, realized via Pure Pursuit Controller, had been developed for convoy-led driverless wheel trucks. For testing and simulation purposes, the algorithm was implemented into a model of the system designed in MATLAB numeric computing environment and visualized in UNREAL ENGINE (by Epic Games) simulation environment.

Ключевые слова: беспилотный грузовой автомобиль, алгоритм управления, Pure Pursuit Controller, движение автомобиля в колонне, модель системы, симуляция.

Keywords: driverless truck, control algorithm, Pure Pursuit Controller, self-driving truck in convoy following, system model, simulation.

Автомобильные концерны по всему миру удивляют своими новейшими разработками в области беспилотных технологий, предоставляя новые возможности для различных сфер и отраслей, в том числе и для грузового транспорта.

Главное преимущество автомобильного транспорта перед железнодорожным – доставка груза от двери до двери и точная прогнозируемость срока доставки. Однако, так же имеются следующие недостатки:

1. Необходимость остановок на сон, душ и приемы пищи,
2. Усталость водителя приводит к риску возникновения ДТП,
3. Нападение на водителей грузовиков с целью ограбления.

Решением, устраняющим упомянутые выше недостатки пилотируемых автомобильных тягачей, могут служить технологии автономных грузоперевозок.

Согласно исследователям из Массачусетского технологического института (MIT), автономные транспортные средства, следующие друг за другом колонной, позволяют сэкономить до 20 % топлива, а замыкающий колонну грузовик может сэкономить до 15 % топлива.[2]

Таким образом, метод выстраивания грузовой автомобилей колонной (рис. 1) позволяет значительно уменьшить экономической расходы на топливо, что является определяющим фактором в сфере логистики, а его основной задачей является минимизация расстояния между грузовиками, учитывая возможность выполнения всех необходимых маневров.



Рисунок 1 – Метод выстраивания грузовой автомобилей колонной

Для обеспечения движения беспилотного грузовика в колонне за впереедиидущим грузовиком необходимо разработать концепцию алгоритма управления ведомого грузовика, основной целью которого является обеспечение точного следования транспортного средства за направляющим тягачом, движущемся по заданному пути.

Для достижения этой цели алгоритм управления должен вычислять угол поворота колес, необходимый для следования в колонне, и отслеживать изменения направления движения направляющего большегруза по мере их появления. Таким образом, задачи, такого алгоритма:

1. свести к минимуму разницу в курсе ведомого транспортного средства и ведущего транспортного средства,
2. обеспечить движение ведомого транспортного средства на заданном расстоянии от ведущего транспортного средства,
3. минимизировать разницу скоростей между ведомым и ведущими транспортными средствами.

Таким образом, чтобы спроектировать алгоритм управления необходимо определить ошибку между положением ведомого и впередиидущим грузовиками, выбрать стратегию проектирования управления, которая сводит ошибки к нулю, при этом удовлетворяя ограничениям угла поворота колес.

Разработка концепции алгоритма управления

Исходя из существующих алгоритмов управления (Pure Pursuit Controller, Stanley Controller, Model Predictive Controller (MPC)) [3] сделан вывод, что на данный момент нет алгоритма, подходящего для решения поставленной задачи, а именно задачи следования беспилотным автомобилем за ведомым транспортным средством. Однако, стоит заметить, что концепция управления Pure Pursuit Controller наиболее близка к поставленной задаче, где в качестве целевой точки на опорном пути может быть взята характерная точка над центром задней оси впередиидущего грузовика. Как уже было сказано ранее алгоритм Pure Pursuit Controller опирается на кинематическую модель велосипеда, рассмотрим ее более детально.

Pure Pursuit Controller

Основная идея данного алгоритма состоит в том, что целевая точка размещена на опорном пути на фиксированном расстоянии впереди транспортного средства, к которой автомобиль стремиться двигаться, используя угол поворота рулевого колеса, который и необходимо вычислить.

В данном алгоритме целевая точка на траектории, центр задней оси, и мгновенный центр вращения образуют треугольник с двумя сторонами длины R и одной длины l_d (рис. 2).

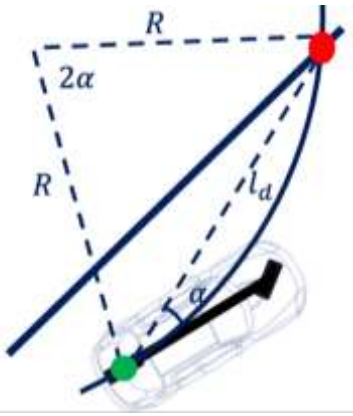

Рисунок 2 – Концепция мгновенного центра вращения в Pure Pursuit Controller

Основываясь на законе синусов, мы можем написать следующее уравнение:

$$\frac{l_d}{\sin 2\alpha} = \frac{R}{\sin(\frac{\pi}{2} - \alpha)}$$

$$\frac{l_d}{\sin \alpha} = 2R$$

Из кинематической модели велосипеда [6]:

$$\tan \delta = \frac{L}{R}$$

Сочетая это выражение с выражением для R, полученным ранее:

$$R = \frac{l_d}{2\sin \alpha}$$

Получим,

$$\tan \delta = \frac{L}{R} = \frac{2\ L\ \sin \alpha}{l_d}$$

Из полученного выражения можно выразить угол поворота δ, равный:

$$\delta = \tan^{-1}\left(\frac{2\ L\ \sin \alpha}{l_d}\right)$$

Концепция алгоритма управления ведомым грузовиком

Как отмечалось ранее, разрабатываемая концепция опирается на Pure Pursuit Controller, однако, в качестве целевой точки на опорном пути берется характерную точка над центром задней оси впередиидущего грузовика, как это показано на рисунке 3.

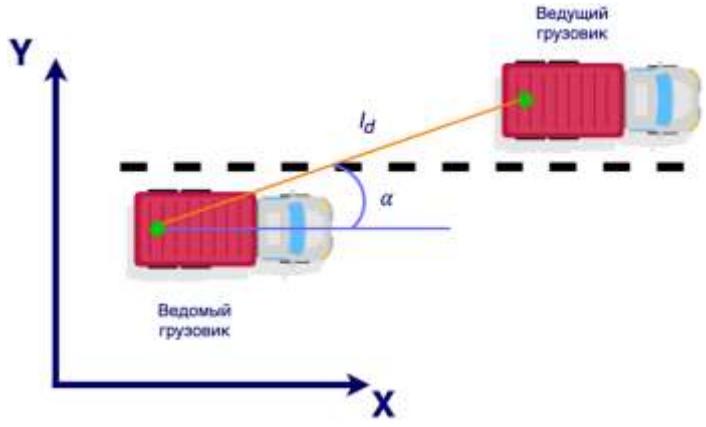

Рисунок 3 – Концепция алгоритма управления ведомым грузовиком

Расстояние между грузовиками l_d не должно превышать заранее заданного минимального расстояния между ведущим и ведомым грузовиками.

Создание и симуляция модели системы с разработанным алгоритмом в программной среде MATLAB

В данном разделе будет рассказано о создании системы в среде MATLAB, в состав которой войдут основные блоки:

1. Блок, задающий траекторию движения ведущего грузовика;
2. Блок, вычисления угла поворота колес ведомого грузовика;
3. Блок кинематической модели велосипеда;
4. Блок коррекции скорости;
5. Блоки визуализации в среде моделирования Unreal Engine от Epic Games.

Начнем проектирование модели системы с создания блока, задающего траекторию движения ведущего автомобиля, используя приложение Matlab Driving Scenario Designer (рис. 4).

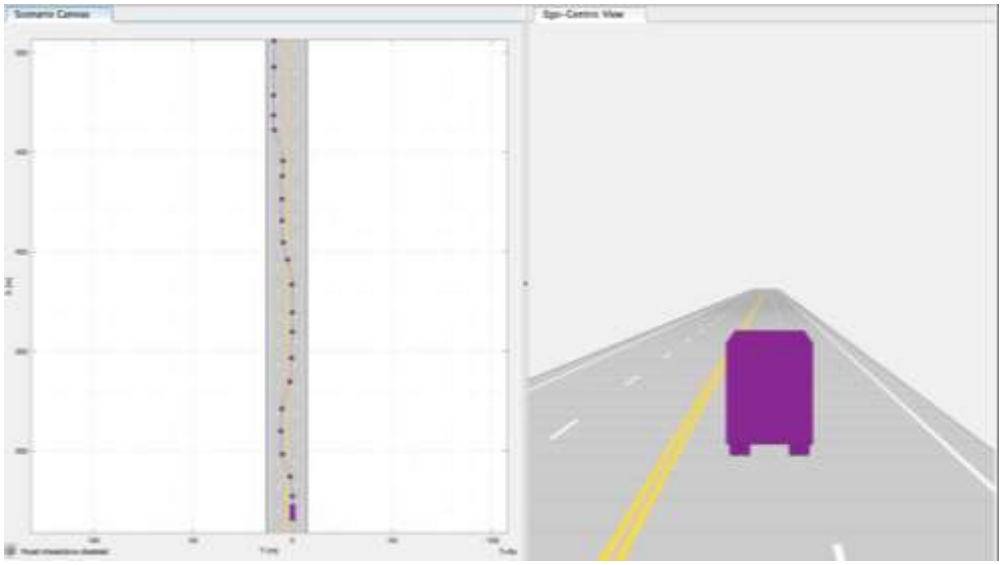

Рисунок 4 – Задание траектории движения ведущего грузовика

С помощью экспортированной функции создается три массива значений:

1. Координаты центра задней оси грузовика по глобальной оси X;
2. Координаты центра задней оси грузовика по глобальной оси Y;
3. Значения угла курса автомобиля;

Затем все три массива с помощью блоков “From Workspace” включаются в состав пользовательски-определенного блока “Заданная траектория движения” (рис. 5).

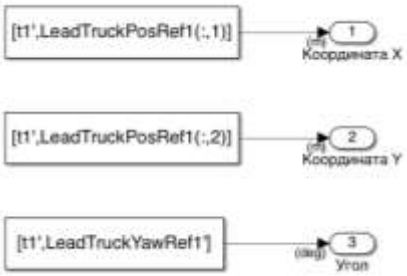

Рисунок 5 – Реализация блока вывода точек траектории ведущей машины

Также в отдельном скрипте задаем начальные состояния обеих машин (рис. 6):

```
stepTime = 0.01;
L = 8.2;
Lmin = 10;
%lead car initial coordinates
X_1 = 266;
Y_1 = 0;
%ego car initial coordinates
X_2 = 256;
Y_2 = 0;
psi_o = 0;
```

Рисунок 6 – Скрипт начальных состояний

В модели системы нахождение расстояния между грузовиками l_d и угла α между направлением кузова ведомого грузовика и расстоянием до контрольной точки ведущего грузовика с помощью стерео камеры заменено расчетом по координатам машин, как показано на рисунке 7.

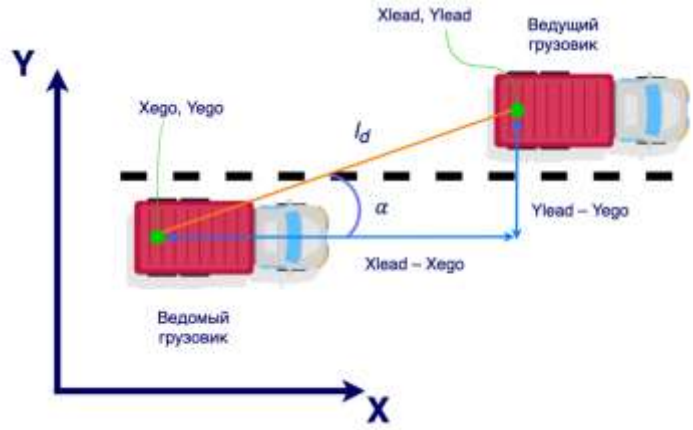

Рисунок 7 – Нахождение расстояния l_d и угла α по координатам

Тогда, $$l_d = \sqrt{(Y_{lead} - Y_{ego})^2 + (X_{lead} - X_{ego})^2};$$

$$\alpha = \tan^{-1}\frac{(Y_{lead} - Y_{ego})}{(X_{lead} - X_{ego})}$$

При вычислении угла отклонения направления ведомого грузовика от направления ведущего грузовика следует учитывать ситуацию, когда грузовики не сонаправлены, как например на рисунке 8.

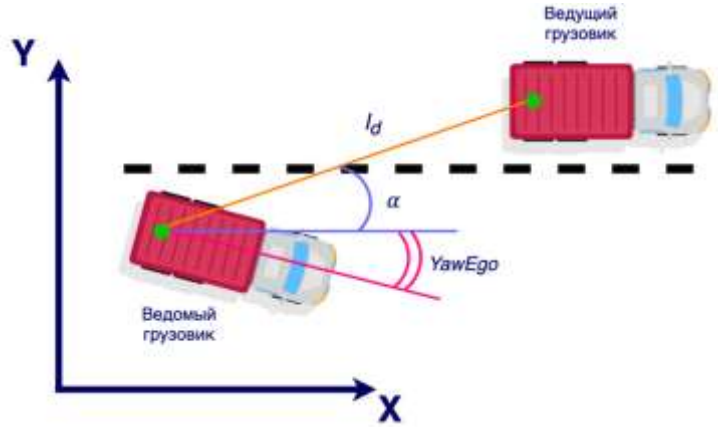

Рисунок 8 – Ситуация, когда грузовики не сонаправлены

Таким образом, угол отклонения направления ведомого грузовика от направления ведущего грузовика ψ равен (рис. 9):

$$\psi = \alpha - YawEgo$$

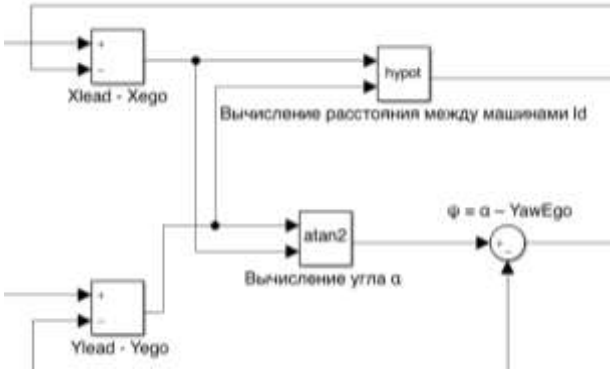

Рисунок 9 – Вычисление угла ψ и расстояния l_d

Блок вычисления поворота колес δ производит расчет по формуле, выведенной в Pure Pursuit Controller (рис. 10):

$$\delta = \tan^{-1}\left(\frac{2\ L\ \sin\alpha}{l_d}\right)$$

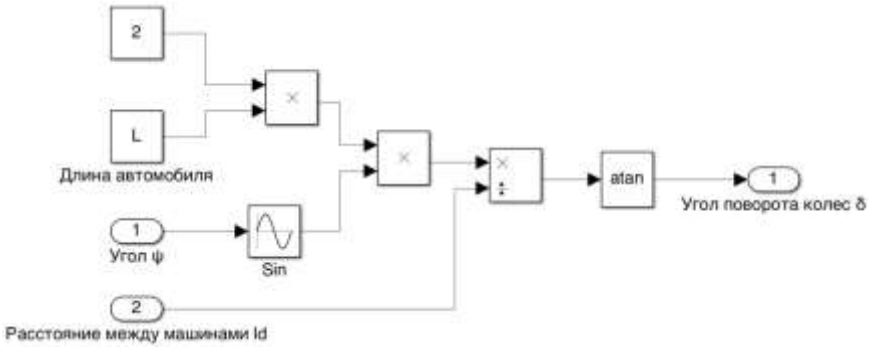

Рисунок 10 – Реализация вычислений в блоке вычисления поворота колес δ

Блок кинематической модели велосипеда представлен в виде стандартного блока Bicycle Kinematic Model из библиотеки Robotics System Toolbox.

Блок коррекции скорости ведомой машины осуществляет коррекцию скорости из учета расстояния между машинами l_d и минимального необходимого расстояния l_{min}, которое заранее задано, а также времени шага t (рис. 11):

$$v = \frac{l_d - l_{min}}{t}$$

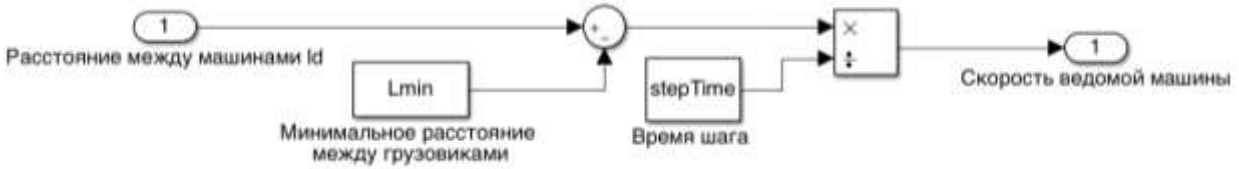

Рисунок 11 – Реализация вычислений в блоке коррекции скорости

Для определения скорости ведущей машины был создан блок вычисления скорости ведущего автомобиля по его координатам (рис. 12). Этот блок определяет скорость транспортного средства по его координатам в глобальной системе координат XOY. Для реализации этого блока необходимо найти матрицу перехода из одной системы координаты в другую.

Воспользуемся матрицей перехода из системы координат транспортного средства в глобальную:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \times \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

где:

$\dot{X}$ – скорость автомобиля по глобальной оси X;

$\dot{Y}$ – скорость автомобиля по глобальной оси Y;

v_x – продольная скорость автомобиля;

v_y – поперечная скорость автомобиля;

φ – угол курса автомобиля;

Тогда, $$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{bmatrix} \times \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix}$$

Тогда получим следующие уравнения для скоростей:

$$v_x = \dot{X}\cos\varphi + \dot{Y}\sin\varphi;$$
$$v_y = \dot{Y}\cos\varphi + \dot{X}\sin\varphi;$$

И скорости машины:

$$v = \sqrt{{v_x}^2 + {v_y}^2}$$

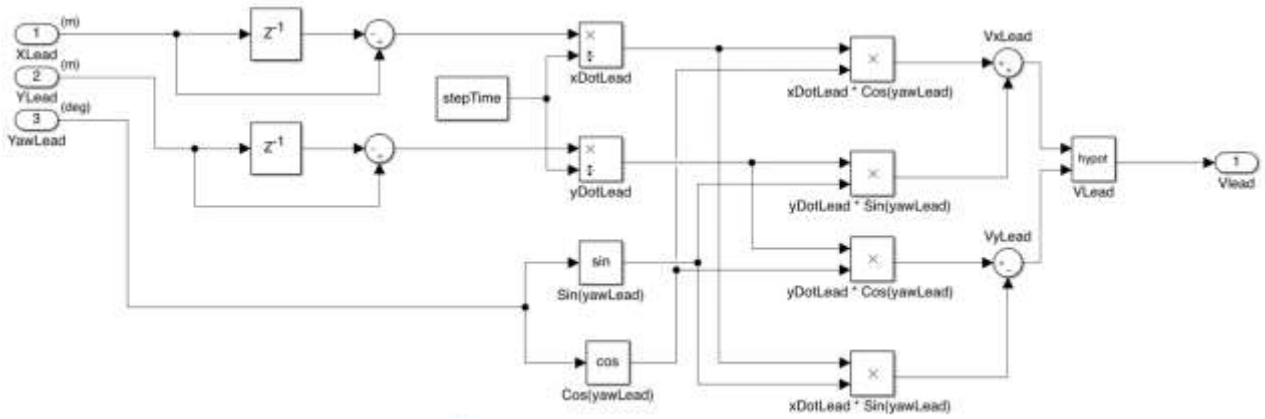

Рисунок 12 – Реализация блока вычисления скорости ведущего автомобиля

Блоки визуализации представляют собой стандартные блоки библиотеки Automated Driving Toolbox, предназначенные для визуализации в среде моделирования Unreal Engine от Epic Games.

Также в систему добавлены исполняющие устройства – электропривод рулевого механизма и тяговый привод. Электропривод и тяговый привод в данной системе представлены в виде простейших апериодических звеньев, электромеханические постоянные которых равны 0.65, и 0.05 соответственно. В результате моделирования получаем следующие графики (рис. 13 и рис. 14):

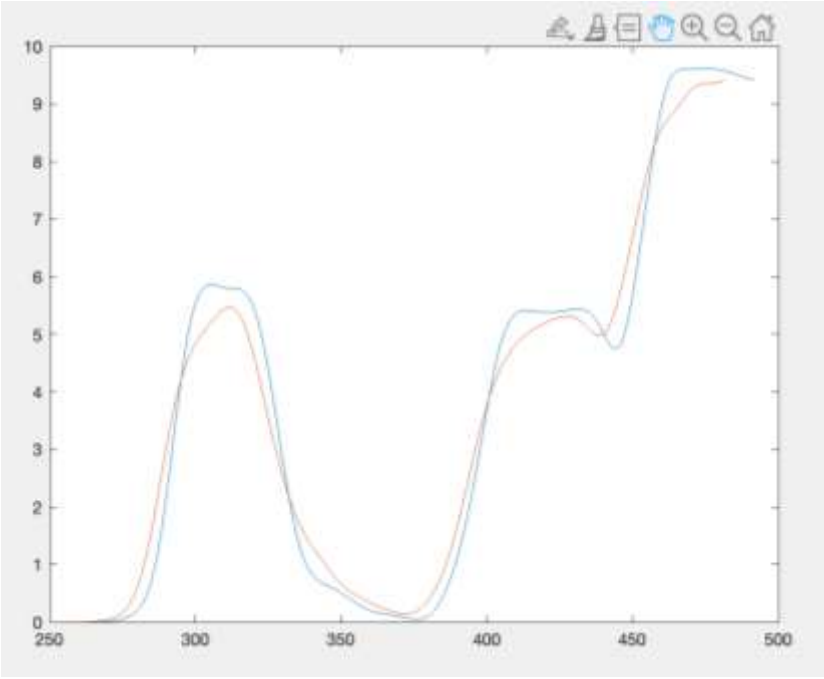

Рисунок 13 – Траектории ведущего (синий) и ведомого (оранжевый)

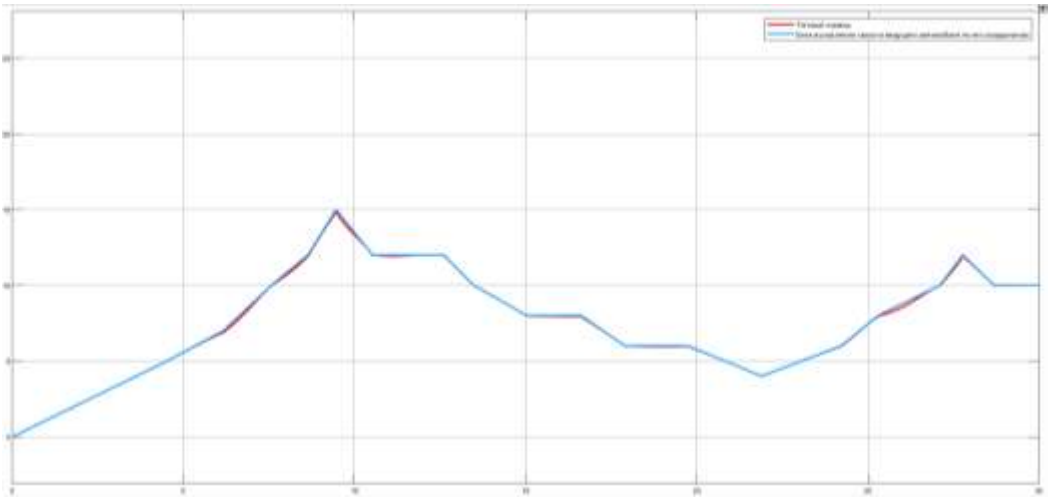

Рисунок 14 – Скорости ведущего (красный) и ведомого (голубой) грузовиков

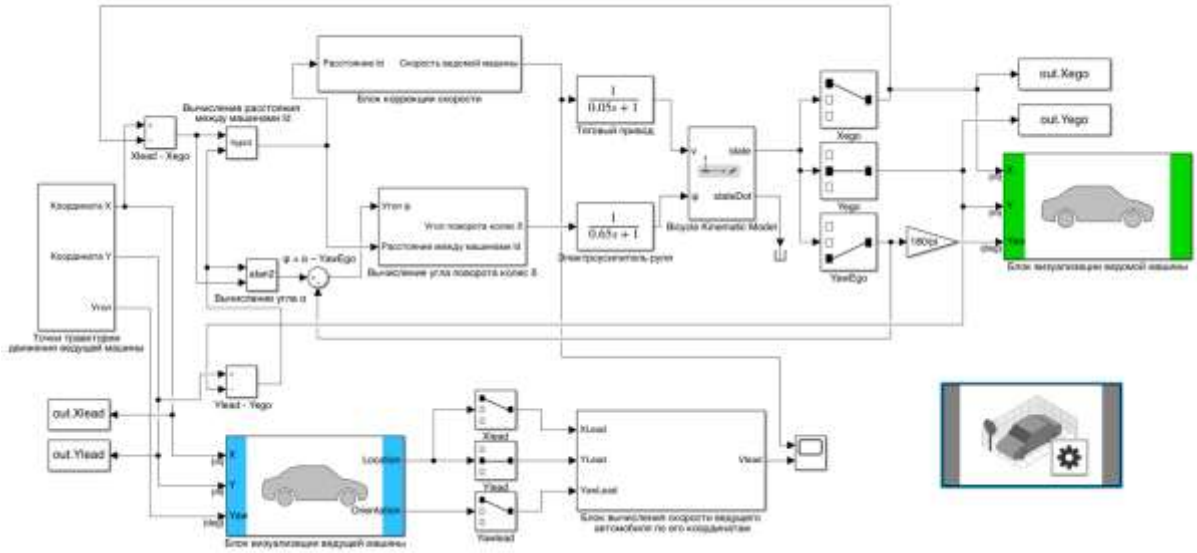

Рисунок 15 – Система в среде Simulink

Визуализация в среде моделирования Unreal Engine

Для наглядности была произведена визуализация симуляции разработанного алгоритма управления в среде моделирования Unreal Engine от Epic Games. Результат визуализации с разных ракурсов представлен на рисунках 16 – 17.



Рисунок 16 – Визуализация в среде моделирования Unreal Engine



Рисунок 17 – Визуализация в среде моделирования Unreal Engine

В результате выполнения данной работы была разработана концепция алгоритма управления беспилотного колёсного тягача, движущегося в колонне за направляющим, на основе Pure Pursuit Controller. Для проверки и симуляция разработанного алгоритма была создана модель системы с разработанным алгоритмом в программной среде MATLAB, а также проведена визуализация в среде моделирования Unreal Engine.

Литература

Literature

1. R. Rajamani, Vehicle Dynamics and Control, Mechanical Engineering Series, Rajesh Rajamani 2012.
2. https://news.mit.edu/2016/driverless-truck-platoons-save-time-fuel-1221
3. Jarrod M. Snider Automatic Steering Methods for Autonomous Automobile Path Tracking 2009.
4. Steven Waslander, Jonathan Kelly, “Introduction to Self-Driving Cars”, Coursera.
5. Gabriel M. Hoffmann, Claire J. Tomlin, “ Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing”, 2007.
6. R. Rajamani,Vehicle Dynamics and Control, Mechanical Engineering Series,DOI, 2012.