Научная статья
Original article
УДК 004.942

РАЗРАБОТКА МОДЕЛИ ИСПОЛНИТЕЛЬНЫХ ПРИВОДОВ, ЗАДЕЙСТВОВАННЫХ В УПРАВЛЕНИИ ПРОДОЛЬНЫМ ПЕРЕДВИЖЕНИЕМ БЕСПИЛОТНОГО ГОНОЧНОГО АВТОМОБИЛЯ ФОРМУЛА СТУДЕНТ ДЛЯ HILS СИСТЕМЫ

THE DEVELOPMENT OF THE MODEL OF ACTUATORS INVOLVED IN THE LONGITUDINAL CONTROL OF THE DRIVERLESS RACING FORMULA STUDENT CAR FOR HILS SYSTEM

Карасев Павел Дмитриевич, магистрант, МГТУ им. Н.Э. Баумана, (105005, Москва, 2-я Бауманская ул., д. 5, стр. 1) 8 (499) 263-639, karasevpd@student.bmstu.ru

Калинин Алексей Владимирович, сотрудник кафедры "Специальная робототехника и мехатроника", МГТУ им. Н.Э. Баумана, (105005, Москва, 2-я Бауманская ул., д. 5, стр. 1) 8 (499) 263-639, kalinin_mvtu@mail.ru

Pavel D. Karasev, master student, Bauman Moscow State Technical University, BMSTU, (105005, ul. Baumanskaya 2-ya, 5/1, Moscow) 8 (499) 263-639, karasevpd@student.bmstu.ru

Alexey V. Kalinin, employee of the "Special robotics and mechatronics" department, Bauman Moscow State Technical University, BMSTU, (105005, ul. Baumanskaya 2-ya, 5/1, Moscow) 8 (499) 263-639, kalinin_mvtu@mail.ru

Аннотация. В данной статье создана модель работы исполнительных приводов

автомобиля класса «Формула Студент», задействованных в продольном передвижении беспилотного гоночного автомобиля, для проведения тестирования разработанных алгоритмических решений для автономного движения на программно-аппаратной модели автомобиля (HILS системе). Созданный интерфейс внутри модели связывает тело модели (реализованное в среде Simulink) с алгоритмами управления беспилотным автомобилем и симулятором (реализованные на языке C++) с использованием ROS2.

Abstract. In this paper we created a model of the actuators of Formula Student car involved in the longitudinal motion of an unmanned racing car to test the developed algorithmic solutions for autonomous driving on a hardware-software model of the car (HILS system). The created interface inside the model links the model body (implemented in the Simulink environment) with the control algorithms of the unmanned vehicle and the simulator (implemented in C++) using ROS2. The created interface inside the model links the model body (implemented in the Simulink environment) with the control algorithms of the driverless vehicle and the simulator (implemented in C++) using ROS2.

Ключевые слова: беспилотный гоночный автомобиль, продольное движение, программно-аппаратная модель автомобиля, HILS система, Simulink, C++, ROS2.

Keywords: driverless race car, longitudinal motion, hardware-software model of the car, Hardware-in-the-Loop Simulation system, Simulink, C++, ROS2.

Стоимость автомобиля «Формула Студент», укомплектованного всем необходимым для автономного передвижения, легко может доходить до 250 000 $ и даже превышать эту сумму. В связи с этим тесты применяемых алгоритмических решений для распознавания объектов, планирования пути, управления и локализации автомобиля на ранних стадиях написания программного обеспечения, когда алгоритмы могут содержать множество как мелких, так и крупных ошибок необходимо проводить не на реальном автомобиле, а на программно-аппаратной модели автомобиля (HILS системе).

Схема всей системы программно-аппаратного моделирования автомобиля представлена на рисунке1.

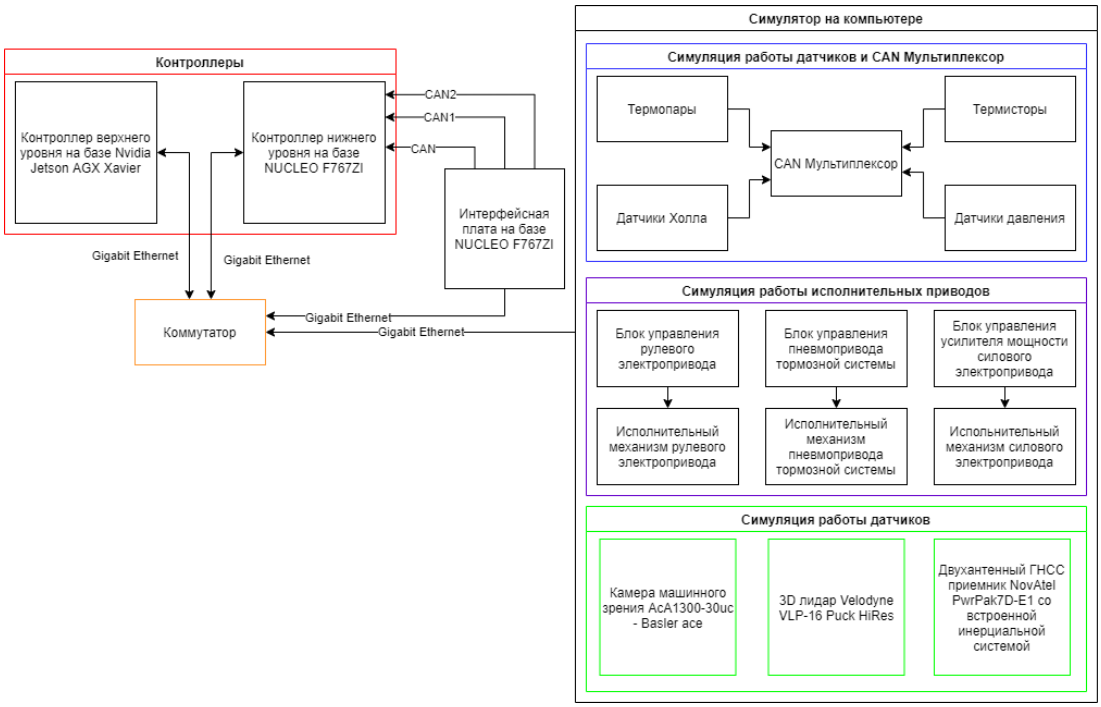

Рисунок 1 – Схема комплекса программно-аппаратного моделирования

Описание требований по подключению модели с исполнительными приводами к симулятору BRTSIM

В качестве симулятора для синтеза сигналов датчиков и имитации динамики автомобиля в процессе гонки используется симулятор BRTSIM на базе программного комплекса Gazebo, совмещенного с фреймворком ROS2.

Все алгоритмы управления беспилотного автомобиля: алгоритмы распознавания, построения пути, управления и локализации беспилотного автомобиля представляют собой отдельные модули, реализованные на языке C++, – узлы внутри фреймворка ROS2, связанные между собой через топики. Данные алгоритмы работают на контроллере верхнего уровня и взаимодействуют со средой симулятора через фреймворк ROS2.

В данной работе описана схема подключение модели к симулятору BRTSIM и алгоритмам управления автомобилем через фреймворк ROS2 с созданием дополнительных узлов (Nodes) внутри архитектуры системы управления, включающих в себя модели исполнительных приводов. Схема подключения модели с исполнительными приводами к архитектуре симулятора и алгоритмов

управления представлена на рисунке 2. Для удобного моделирования исполнительных приводов будут использоваться программные пакеты MATLAB, Simulink. Для связи модели из Simulink с фреймворком ROS2 предполагается использовать инструментарий из ROS Toolbox, который распространяется вместе с пакетом Simulink.

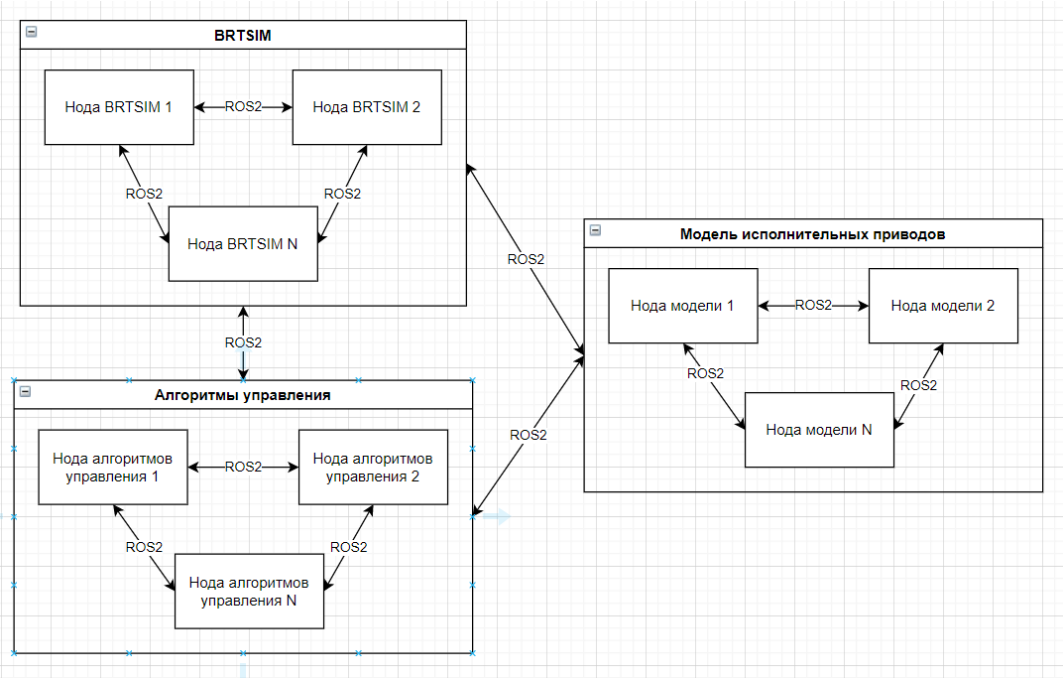

Рисунок 2 – Связь модели исполнительных приводов с BRTSIM и алгоритмами управления

Описание состава исполнительных приводов, задействованных в управлении продольным движением гоночного автомобиля

За управление продольным движением гоночного автомобиля команды МГТУ отвечает электрическая силовая установка и тормозная система.

В состав электрической силовой установки автомобиля входят:

1. Синхронный электродвигатель с ротором на постоянных магнитах (PMSM) EMRAX 228 HV CC;
2. Инвертор для преобразования постоянного тока с тяговой батареи в трехфазный Unitek Bamocar D3;
3. Силовая батарея на базе ячеек Li1x6pVTC6T;

В состав тормозной системы автомобиля входят:

1. Тормозные суппорта, разработанные командой BRT;

2. Тормозные гидроцилиндры AP-Racing;
3. Пневмосистема, собранная командой МГТУ для активации рабочей тормозной системы на базе компонентов компании FESTO;

Техническое задание на создание модели с исполнительными приводами, подключаемую к симулятору BRTSIM

Совокупность уравнений, необходимых для тела модели:

$$\begin{cases} F_{tractive} = D_{acc} \cdot M_m \cdot Gr/R_w - F_{brakes}; \\ \left[\begin{array}{l} F_{brakes}=D_{bregen}\cdot M_m\cdot\frac{G_r}{R_w}, \text{ если } F_{brakes}=< F_{regenerative_max}; \\ F_{brakes}= D_{bfric}\cdot\frac{2}{Rw}\cdot(M_{br}+M_{bf})+D_{bregen}\cdot M_m\cdot\frac{G_r}{R_w}, \text{ если } F_{brakes}> F_{regenerative_{max}}; \end{array}\right. \end{cases}$$

где

$F_{tractive}$ [Н] – сумарная сила на выходе модели (суммарная сила действия тормозной силы и разгоняющей силы от тягового электродвигателя);

D_{acc}– управляемая величина, изменяющаяся в диапазоне от 0 до 1 в зависимости от требуемого разгона с помощью электродвигателя;

M_m [Н*м] – максимальный крутящий момент, возникающий на валу электродвигателя;

G_r – главное передаточное число между ведущей звездой электродвигателя и ведомой звездой;

R_w[м] – радиус колеса;

F_{brakes}[Н] – суммарная тормозная сила, возникающая в результате воздействия рабочей тормозной системы и торможения электродвигателем;

D_{bfric}(0…1) – коэффициент равный степени торможения с участием тормозной системы;

D_{bregen}(0…1) – коэффициент равный степени торможения с участием электродвигателя;

$F_{regenerative_max}$[Н] – максимальная тормозная сила, полученная в результате торможения с помощью рекуперации, осуществляемой электродвигателем;

M_{br}[Н*м] – максимальный тормозной момент, возникающий в результате торможения на заднем колесе;

M_{bf}[Н*м] – максимальный тормозной момент, возникающий в результате торможения на переднем колесе;

Необходимые требования к модели:

1. Входами модели должны быть общая степень торможения D_b, изменяющая свое значение от 0 до 1 и степень разгона D_{acc}, изменяющая свое значение от 0 до 1.
2. Входные величины D_{acc} и D_b должны быть получены от алгоритма управления, используя фреймворк ROS2.
3. Выходом модели должна быть сила $F_{tractive}$, которая должна отправляться в симулятор через фреймворк ROS2.

Общая схема модели представлена на рисунке 3.

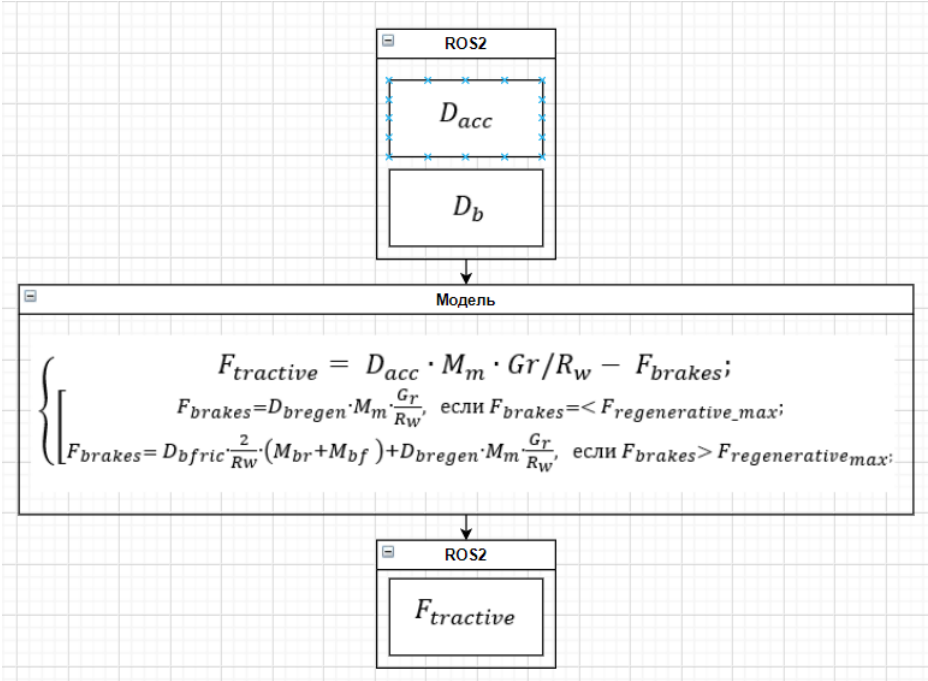

Рисунок 3 – Общая схема модели

Сборка модели осуществляется в программном пакетах MATLAB 2021b и Simulink. Модель будет представлена в виде совокупности моделей подмодулей, имеющих определенные наборы входных и выходных величин.

Создание подмодуля, определяющего величину требуемой тормозной силы от тормозной системы и величину требуемой тормозной силы от электродвигателя

На вход данного подмодуля поступает величина D_b от алгоритма управления. В процессе разработки подмодуля было решено использовать D_b, величину которая изменяет свое значение от -1 до 0 вместо того, чтобы изменять свое значение от 0 до 1, в связи с удобством построения модели.

Используя уравнения для вычисления суммарной тормозной силы:

$$\begin{bmatrix} F_{brakes} = F_{regen}, \text{ если } F_{brakes} =< F_{regenerative_max}; \\ F_{brakes} = F_{friction} + F_{regenerative_max}, \text{ если } F_{brakes} > F_{regenerative_{max}}; \end{bmatrix}$$

вычислим составные силы F_{brakes}: $F_{regenerative}$ и $F_{friction}$.

Таким образом, выходом данного подмодуля будут F_{regen}и $F_{friction}$.

Распишем:

$$F_{brakes} = D_b \cdot (F_{friction_max} + F_{regenerative_max});$$

$$F_{regenerative_max} = M_m \cdot \frac{Gr}{R_w};$$

$$F_{regenerative} = D_{bregen} \cdot M_m \cdot Gr/R_w;$$

$$F_{friction_max} = \frac{2}{R_w} \cdot (M_{br} + M_{bf});$$

$$F_{friction} = D_{bfric} \cdot \frac{2}{R_w} \cdot (M_{br} + M_{bf});$$

$$\begin{bmatrix} \begin{cases} D_{bregen} = \dfrac{F_{brakes}}{F_{regenerative_max}}; \\ D_{bfric} = 0; \end{cases} \text{ если } F_{brakes} =< F_{regenerative_max} \\ \begin{cases} D_{bregen} = 1; \\ D_{bfric} = \dfrac{F_{brakes} - F_{regenerative_max}}{F_{friction_max}}; \end{cases} \text{ если } F_{brakes} > F_{regenerative_max} \end{bmatrix}$$

Интегрировав уравнения, требуемые для вычисления $F_{friction}$ и $F_{regenerative}$ внутрь подмодуля, используя на входе общую степень торможения D_b, была получена следующая структура подмодуля (рисунок 4). В подмодуле реализовано условие, предотвращающее движения автомобиля назад по действием тормозной силы.

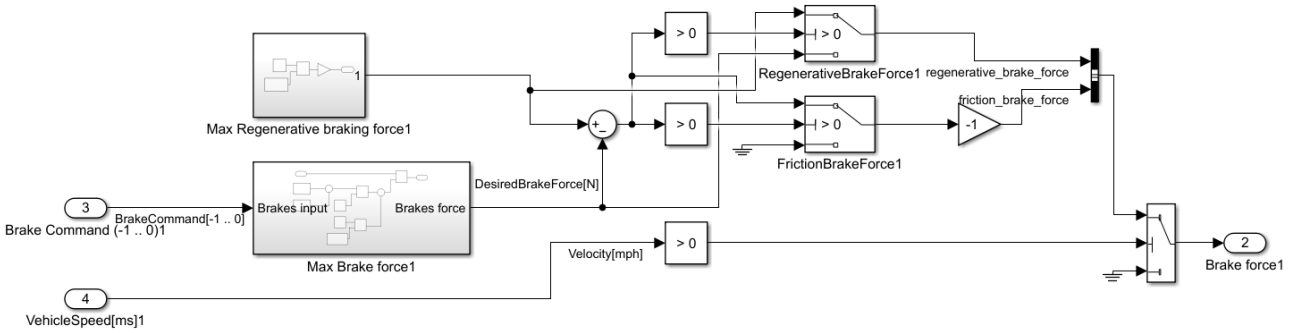

Рисунок 4 – Структура подмодуля

Для удобства читаемости структуры вычисления сил $F_{friction_max} + F_{regenerative_max}$ и $F_{regenerative_max}$ были свернуты в свои подмодули Max Brake force (рисунок 5) и Max Regenerative force (рисунок 6) соответственно.

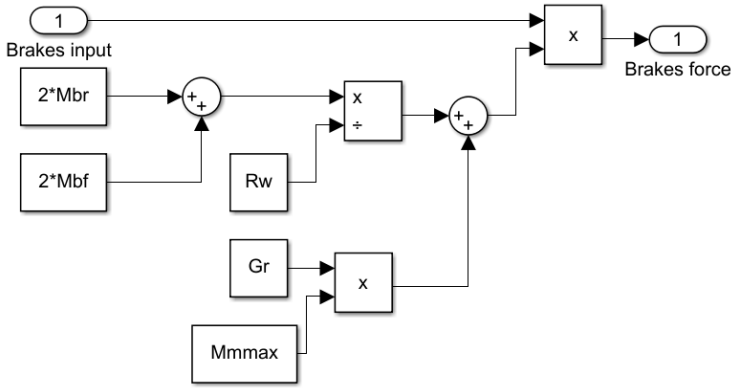

Рисунок 5 – Подмодуль для вычисления $F_{friction_max} + F_{regenerative_max}$

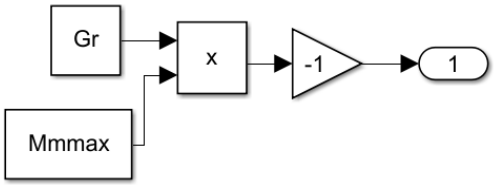

Рисунок 6 – Подмодуль для вычисления $F_{regenerative_max}$

Общий вид смоделированного подмодуля в общей модели выглядит следующим образом (рисунок 7):

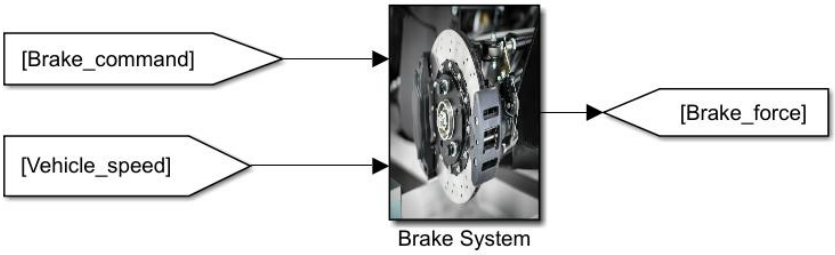

Рисунок 7 – Общий вид подмодуля

Создание подмодуля, определяющего крутящий момент на выходном валу электродвигателя

Так как электродвигатель может как разгонять, так и тормозить автомобиль, но в один и тот же момент времени он может выполнять только одну функцию, поэтому было принято создать один подмодуль, на вход которого поступают:

1) F_{regen} – команда о требуемой силе торможения с подмодуля Brake system;
2) D_{acc} – (0…1), приходящая на вход подмодуля из алгоритма управления;

Центром данного подмодуля является общий блок на инвертор и электродвигатель из библиотеки Simscape Electrical (рисунок 8), который реализует в себе функционал электродвигателя с ротором на постоянных магнитах (PMSM) и инвертора, то есть реализует функционал выбранного электродвигателя EMRAX 228 и инвертора Unitek Bamocar D3.

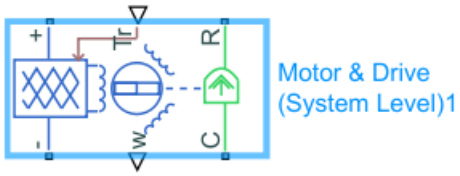

Рисунок 8 – Блок, реализующий функционал PMSM двигателя и инвертора

Внутри самого блока реализована обратная связь по моменту на выходном валу электродвигателя. Требуемый момент Tr, подводимый на вход блока, необходимо вычислить, имея в качестве входных переменных F_{regen} и D_{acc}.

Для этого на первом этапе необходимо получить F_m:

$$F_m = D_{acc} \cdot M_m \cdot Gr/R_w;$$

Далее необходимо получить требуемый момент на валу электродвигателя M:

$$M = \frac{F_m}{G_r} \cdot R_w;$$

Далее необходимо силу торможения F_{regen} преобразовать в момент M_{regen}:

$$M_{regen} = \frac{F_{regen}}{G_r} \cdot R_w;$$

Для получения суммарного момента Tr необходимо :

$$Tr = M + M_{regen};$$

Для того, чтобы подать на порт R обратную связь о скорости вращения выходного вала электродвигателя, ее необходимо получить из симулятора BRTSIM путем преобразования продольной скорости движения автомобиля к скорости вращения выходного вала электродвигателя:

$$w = \frac{v_x}{R_w} \cdot G_r;$$

где

w – скорость вращения выходного вала электродвигателя;

v_x- продольная скорость движения автомобиля;

Таким образом подмодуль имеет три входа: F_{regen}, D_{acc} и v_x и один выход M_t – момент M_t, снимаемый с вала электродвигателя с помощью идеального датчика момента (рисунок 9, 10).

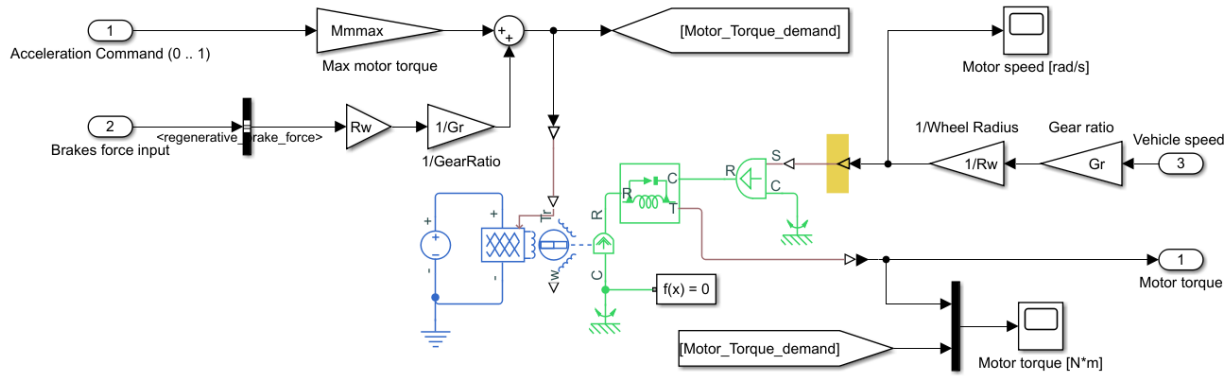

Рисунок 9 – Внутренний вид подмодуля

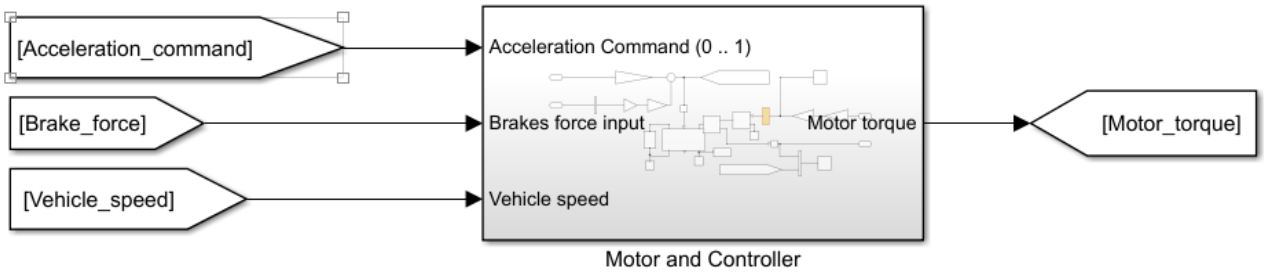

Рисунок 10 – Общий вид подмодуля

Создание подмодуля для расчета результирующей силы $F_{tractive}$

Основной задачей модели является расчет результирующей силы $F_{tractive}$:

$$F_{tractive} = F_m - F_{brakes};$$

где F_m – разгоняющая сила;

Так как от подмодуля Motor and Controller поступает крутящий момент электродвигателя M_t, а от подмодуля Brake system – $F_{friction}$, то сила $F_{tractive}$:

$$F_{tractive} = M_t \cdot \frac{G_r}{R_w} - F_{friction};$$

При этом входными величинами подмодуля будут $F_{friction}$, M_t, а выходной переменной будет $F_{tractive}$ (рисунок 11, 12).

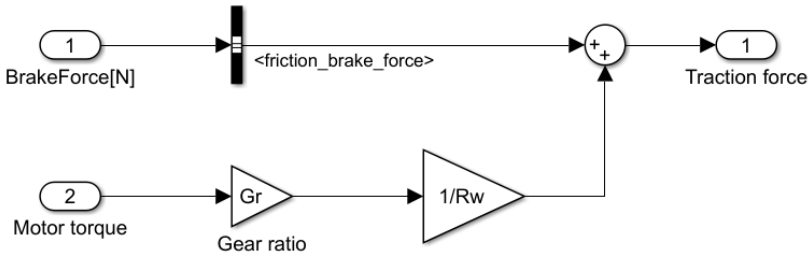

Рисунок 11 – Внутренний вид подмодуля

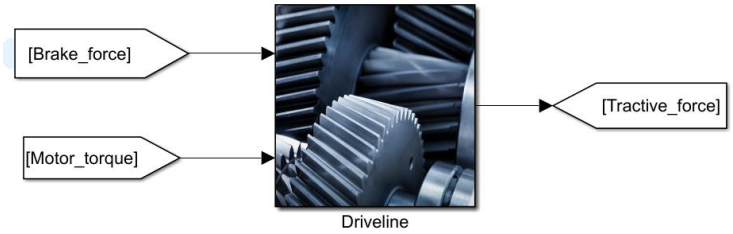

Рисунок 12 – Внешний вид подмодуля

Общий вид всей полученной модели, на вход которой приходят D_b, D_{acc}и v_x, на выходе которой получается $F_{tractive}$, представлен на рисунке 13.

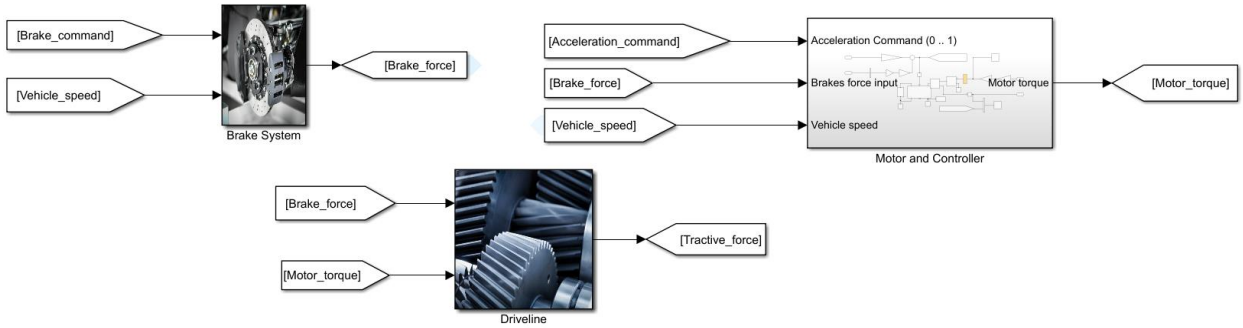

Рисунок 13 – Общий вид полученной модели

Создание интерфейса внутри модели, связывающего тело модели с алгоритмами управления беспилотным автомобилем и симулятором

На данном этапе необходимо создать интерфейс для связи модели, полученной в пункте, выполненном ранее, с симулятором BRTSIM и алгоритмами управления. Так как и симулятор BRTSIM и алгоритмы управления уже имеют

налаженную связь через фреймворк ROS2, то самым удобным способом будет связать полученную модель с симулятором и алгоритмами управления через фреймворк ROS2. Matlab и Simulink с недавнего времени имеют специальный ROS Tollbox для выполнения двусторонней связи между моделями, выполненными в их среде и ROS2.

Для отправки и приема данный в модель будут использоваться подмодули: подмодуль приема управляющего сигнала от алгоритмов управления, подмодуль приема продольной скорости автомобиля из симулятора BRTSIM и подмодуль отправки $F_{tractive}$.в симулятор BRTSIM.

Создание подмодуля для приема управляющего сигнала от алгоритмов управления

Подмодуль для приема управляющего сигнала состоит из элемента ROS Toolbox Subscribe, который принимает сообщения из фреймворка ROS2, отправленные алгоритмом управления в топик “/brt8d/control/pure_pursuit/car_command”, далее из данного сообщения с помощью Bus selector вычленяется информация о требуемом разгоне/ торможении и делится на два выхода: D_b и D_{acc} (рисунок 14, 15).

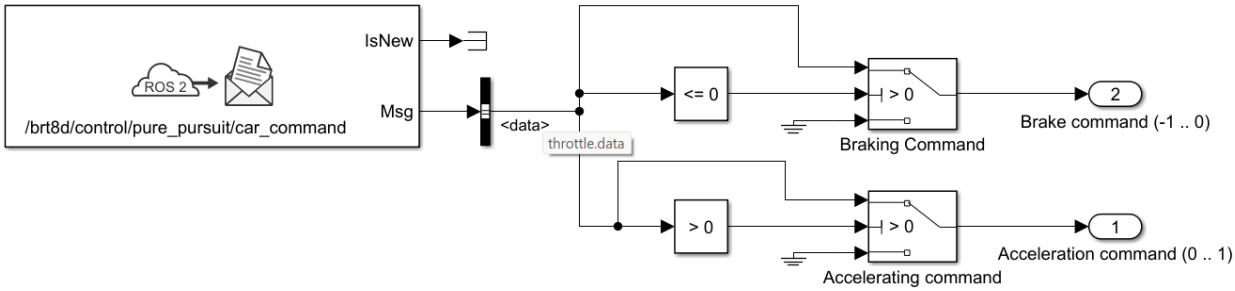

Рисунок 14 – Схема внутреннего устройства подмодуля

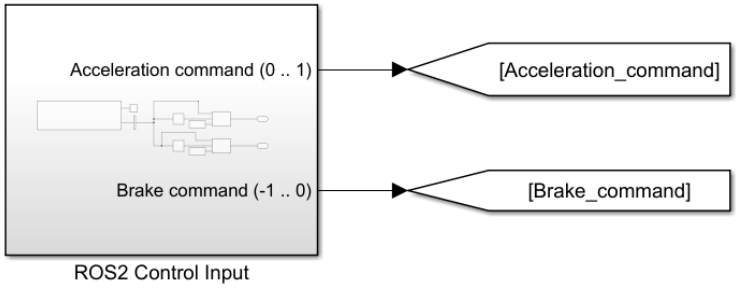

Рисунок 15 – Общий вид подмодуля на общей схеме

Создание подмодуля для приема продольной скорости от симулятора BRTSIM

Подмодуль для приема продольной скорости состоит из элемента ROS Toolbox Subscribe, который принимает сообщения из фреймворка ROS2, отправленные симулятором в топик “/brt8d/estimation/slam/state”, далее из данного сообщения с помощью Bus selector вычленяется информация о продольной скорости и отправляется на выход: v_x (рисунок 16, 17).

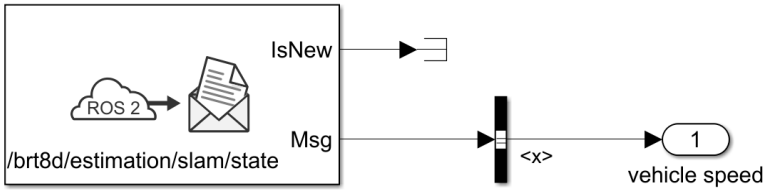

Рисунок 16 – Схема внутреннего устройства подмодуля

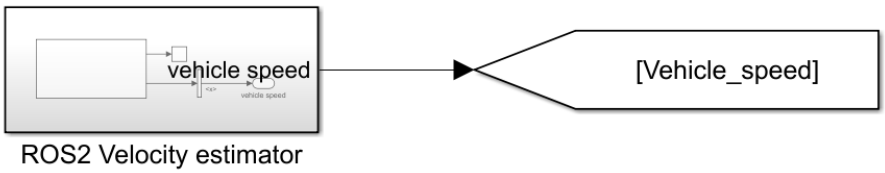

Рисунок 17 – Общий вид подмодуля

Создание подмодуля для отправки $F_{tractive}$в симулятор BRTSIM

Подмодуль для отправки $F_{tractive}$ состоит из элемента ROS Toolbox Publish, который отправляет созданные сообщения внутрь фреймворка ROS2 в топик “/matlab/tractive_force”. Перед этим необходимо с помощью блока ROS Toolbox создать пустое сообщение типа std_msgs/Float64 и с помощью блока Bus assignment заполнить его данными о $F_{tractive}$ (рисунок 18, 19).

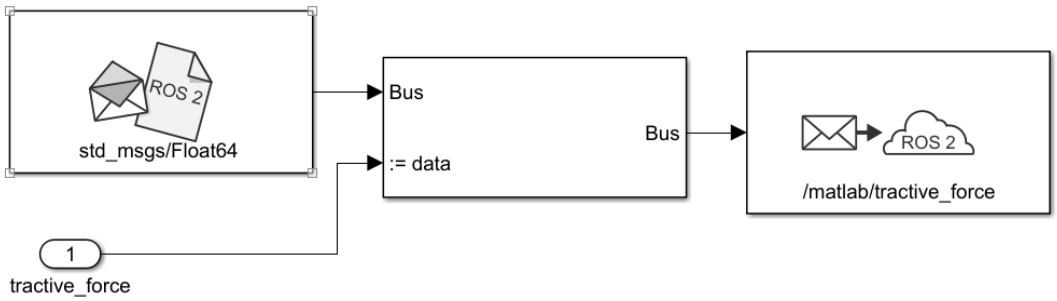

Рисунок 18 – Схема внутреннего устройства подмодуля

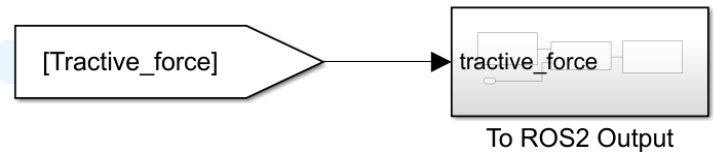

Рисунок 19 – Общий вид подмодуля

После создания интерфейса, связывающего модель, симулятор BRTSIM и алгоритмы управления, модель принимает следующий вид (рисунок 20).

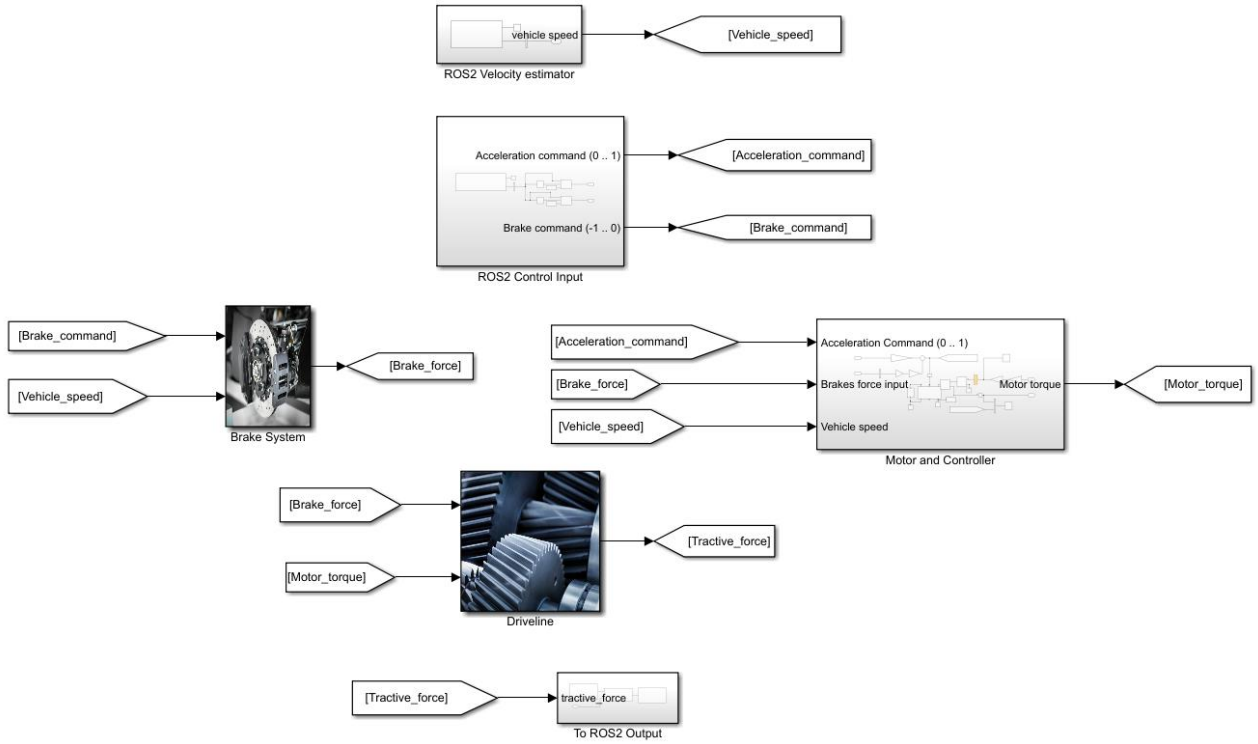

Рисунок 20 – Общий вид полученной модели с интерфейсом

Проверка работы связки симулятора, модели и алгоритмов управления

Для проверки работы связки симулятора, модели и алгоритмов управления необходимо разработать подмодуль, в котором можно будет наблюдать за изменением основных величин таких, как M_t, F_{brakes}, $F_{tractive}$. Помимо инструментов мониторинга в данном подмодуле можно реализовать аналог дополнительного тормозного усилия (блок Slider), останавливающего автомобиль. Результатом увеличения такого дополнительного тормозного усилия будет рост момента на валу электродвигателя и замедление автомобиля. движущегося в симуляторе.

Входами для данного подмодуля будут наблюдаемые величины: M_t, F_{brakes}, $F_{tractive}$, выходы – блоки Scope.

Схема внутреннего устройства подмодуля мониторинга представлена на рисунок 21, общий вид подмодуля для мониторинга представлен на рисунок 22.

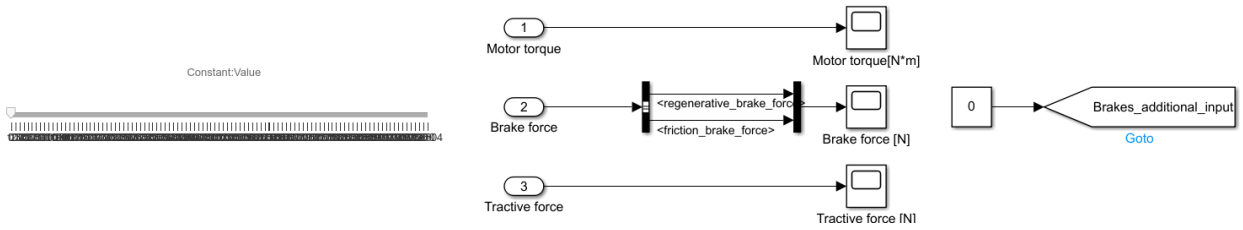

Рисунок 21 – Схема внутреннего устройства подмодуля

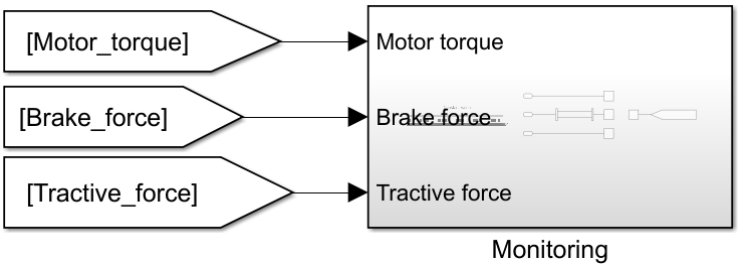

Рисунок 22 – Общий вид подмодуля

На данном этапе сборка модели полностью завершена и необходимо выполнить тестирование связки симулятора, алгоритмов управления и модели. Для этого необходимо запустить симулятор BRTSIM в связке с кинематическим способом управления pure_pursuit.

После выполнения запуска симулятора открывается окно Gazebo (рисунок 23) с автомобилем и виртуальным миром, а также окно визуализации данных Rviz2 (рисунок 24).

Сразу после запуска симулятора и алгоритмов управления они начинают выдавать и забирать информацию из соответствующих топиков, поэтому пока модель приводов выключена, автомобиль покоится на месте, так как сила $F_{tractive}$ равна при этом нулю.

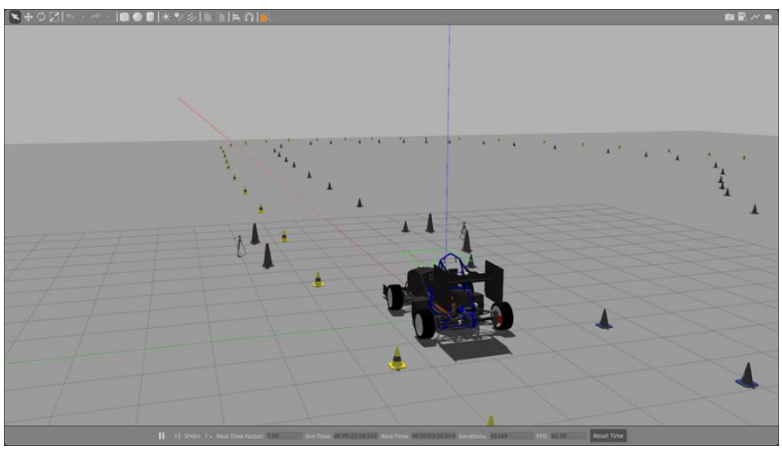

Рисунок 23 – Окно Gazebo с запущенным симулятором BRTSIM (автомобиль покоиться в начальной точке)

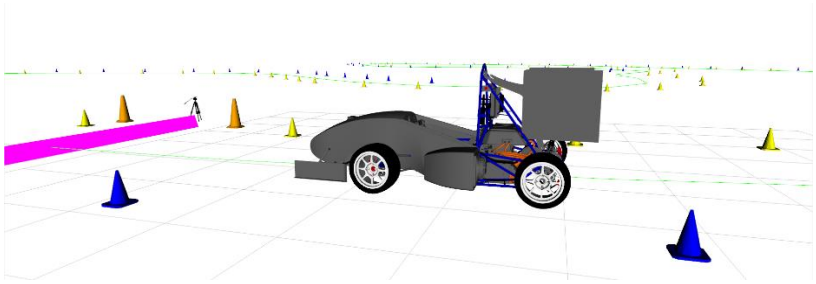

Рисунок 24 – Окно Rviz2 с визуализацией данных

Как только модель запускается, автомобиль начинает движение. Это видно в симуляторе и на блоках Scope подмодуля мониторинга (рисунок 25, 26).

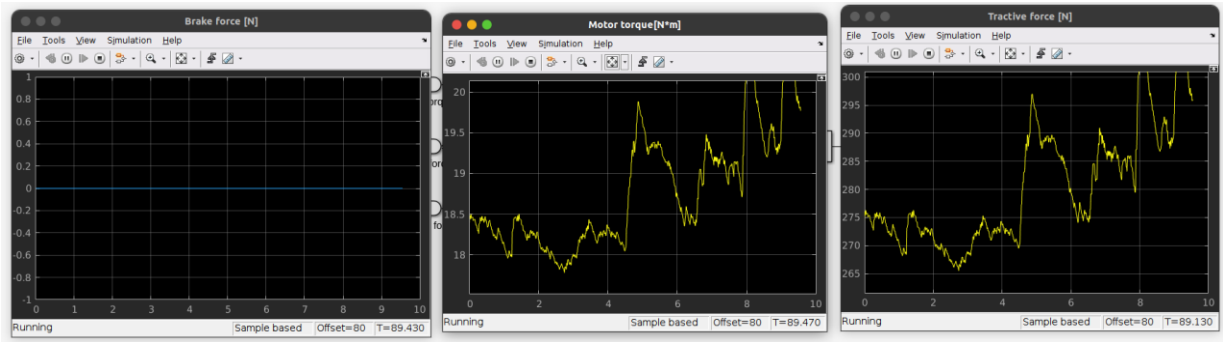

Рисунок 25 – Графики суммарной силы F_{brakes}, крутящего момента на выходном валу двигателя M_t, результирующей силы $F_{tractive}$

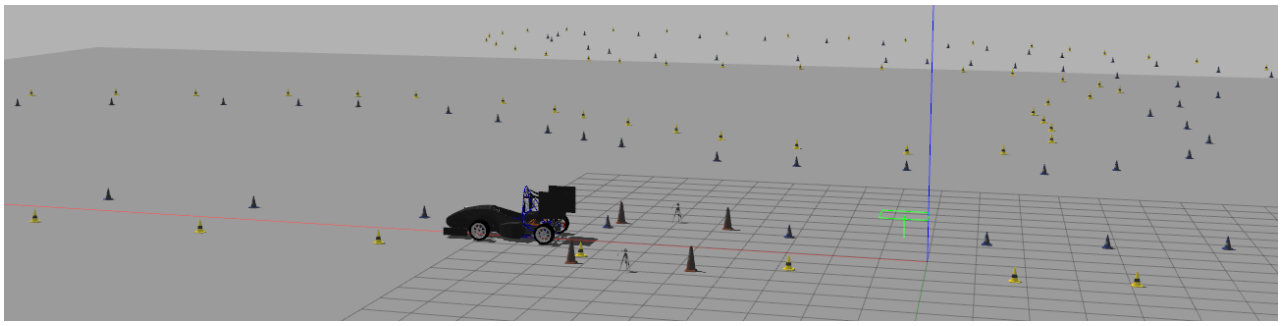

Рисунок 26 – Автомобиль в движении

При повышении тормозного усилия растет крутящий момент на электродвигателе (рисунок 27).

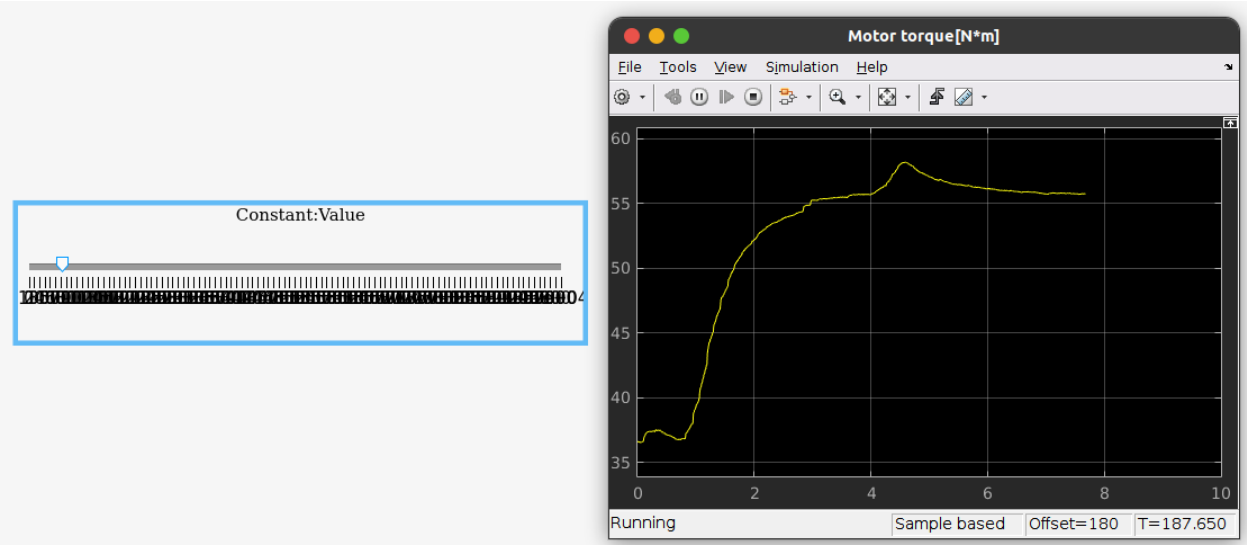

Рисунок 27 – Повышение крутящего момента на выходном валу электродвигателя с увеличением тормозной силы

В какой-то момент мощности электродвигателя перестает хватать на компенсацию все более возрастающей тормозной силы и автомобиль начинает замедляться. Таким образом можно сделать вывод о том, что связка симулятора, алгоритмов управления и модели в MATLAB Simulink устойчиво работает и выдает адекватный результат.

Литература

References

1. Formula Student rules 2020 – URL: https://www.formulastudent.de/fileadmin/user_upload/all/2020/rules/FS-Rules_2020_V1.0.pdf
2. ROS2 Galactic documentation – URL: https://index.ros.org/doc/ros2/
3. Gazebo documentation – URL: http://gazebosim.org/
4. Matlab ROS Toolbox documentation – URL: https://www.mathworks.com/products/ros.html
5. Matlab Simscape Electrical documentation – URL: https://www.mathworks.com/products/simscape-electrical.html
6. Matlab Motor & Drive (System Level) block documentation – URL: https://www.mathworks.com/help/physmod/sps/ref/motordrivesystemlevel.html

7. Matlab Powertrain Blockset documentation – URL: https://www.mathworks.com/products/powertrain.html
8. Matlab HEV PMSM Drive Test Harness documentation – URL: https://www.mathworks.com/help/physmod/sps/ug/hev-pmsm-drive-test-harness.html

© Карасев П.Д., Калинин А.В., 2022 Научно-образовательный журнал для студентов и преподавателей «StudNet» №6/2022.

Для цитирования: Карасев П.Д., Калинин А.В. РАЗРАБОТКА МОДЕЛИ ИСПОЛНИТЕЛЬНЫХ ПРИВОДОВ, ЗАДЕЙСТВОВАННЫХ В УПРАВЛЕНИИ ПРОДОЛЬНЫМ ПЕРЕДВИЖЕНИЕМ БЕСПИЛОТНОГО ГОНОЧНОГО АВТОМОБИЛЯ ФОРМУЛА СТУДЕНТ ДЛЯ HILS СИСТЕМЫ // Научно-образовательный журнал для студентов и преподавателей «StudNet» №6/2022.