

УДК 29.113.073

DOI: 10.46960/1816-210X_2023_2_79

О ПУТЯХ ВИЗУАЛИЗАЦИИ ДОРОЖНОЙ ОБСТАНОВКИ В СМЕШАННОЙ РЕАЛЬНОСТИ ПРИ МОДЕЛИРОВАНИИ АВТОМОБИЛЯ

В.Г. МихайловORCID: 0009-0008-4767-7293 e-mail: sapr7@mail.ru*Минск, Республика Беларусь*

Проанализированы пути компьютерной визуализации дорожной обстановки при моделировании движения, колебаний и управляемости автомобиля с использованием смешанной реальности. Отмечена сложность реализации дорожной обстановки при смешанной реальности в пакетах *Unity*, *Unreal Engine*, где имеет место нереалистичная картина дороги, не учитывается макро- и микропрофиль дороги, и проблематична синхронизация пройденного пути с изображением видео. Предложен метод реализации визуализации дорожной обстановки на основе записанного видео на фиксированной скорости движения, которое воспроизводится в шлеме виртуальной реальности в соответствии скорости моделирования автомобиля в пакете *Simulink* на основе разработанной на C++ программы видео и полученной зависимости задержки от скорости для управления видео. Получено приемлемое расхождение (3-4 %) по времени прохождения контрольной точки пути и видео при использовании предложенного метода, подтверждающее его корректность.

Ключевые слова: моделирование автомобиля, программное обеспечение, виртуальная, смешанная реальность.

ДЛЯ ЦИТИРОВАНИЯ: Михайлов, В.Г. О путях визуализации дорожной обстановки в смешанной реальности при моделировании автомобиля / В.Г. Михайлов // Труды НГТУ им. Р.Е. Алексеева. 2023. № 2. С. 79-88.

DOI: 10.46960/1816-210X_2023_2_79

METHODS OF TRAFFIC SITUATION VISUALIZATION USING MIXED REALITY DURING MOTOR VEHICLE MODELING

V.G. MikhailovORCID: 0009-0008-4767-7293 e-mail: sapr7@mail.ru*Minsk, Belarus Republic*

Abstract. The study analyzed methods of computer visualization of traffic situation during simulation of movement, vibrations and steerability of a motor vehicle using mixed reality. The author notes difficulty in modeling traffic situation in mixed reality of *Unity*, *Unreal Engine* packages having unrealistic road pattern and missing road macro and micro-profiles, where it is unclear how to achieve synchronization of the path traveled with the video image. The proposed is a method of traffic situation visualization based on a fixed-rate recorded video playing on a VR Helmet matching the speed of a simulated motor vehicle in *Simulink* package which uses C++-based video software and the obtained delay/speed dependency to control the video. The correctness of the proposed method is confirmed by the resulted acceptable offset between the control point passing times on a simulated route and in a video. For the purposes of simulation of movement, vibrations and steerability, the use of *Unreal Engine* package can be eliminated. However, the tasks of ergonomics modeling require the use of this package as well as trackers with their correspondent software.

Key words: motor vehicle modeling, software, virtual, mixed reality.

FOR CITATION: Mikhailov V.G. Methods of traffic situation visualization using mixed reality during motor vehicle modeling. Transactions of NNSTU n.a. R.E. Alekseev. 2023. № 2. С. 79-88. DOI: 10.46960/1816-210X_2023_2_79

Введение

Зарубежные исследователи начала XXI в. уделяют большое внимание совершенствованию методов имитационного моделирования автомобиля [1, 2]; пройдены этапы от имитационного моделирования на вибростенде с экранами для воспроизведения дорожной обстановки до создания движущихся по рельсам симуляторов, в которых размещаются весь автомобиль или его часть для создания ощущения у водителя нахождения в дороге и управления им (рис. 1) т.е. учета человеческого фактора, существенно влияющего на режим движения автомобиля и нагруженность его узлов.



Рис. 1. Развитие методов имитационного моделирования автомобиля

Fig. 1. Method development for motor vehicle simulation modeling

Ведущие зарубежные автомобильные фирмы интенсивно используют методы виртуальной (VR) и смешанной реальности (MR), позволяющие снизить затраты на такие стенды (рис. 1, справа) [3-5]. Виртуальная и смешанная реальность представляет собой компьютерную технологию погружения человека в искусственный моделируемый мир, где он управляет виртуальным объектом (в данном случае – автомобилем, движущимся по дороге). Это достигается путем использования специального шлема, звукового воздействия, «искусственной руки», вибростенда и моделирования автомобиля. Лидеры автомобильной промышленности для этих целей в основном используют предназначенный для создания игр, мультимедиа, киноиндустрии пакет *Unreal Engine (UE)* [6]. Примерами его использования являются игра *Euro Truck Simulator*, рекламные материалы *BMW* [7] и других фирм (рис. 2).

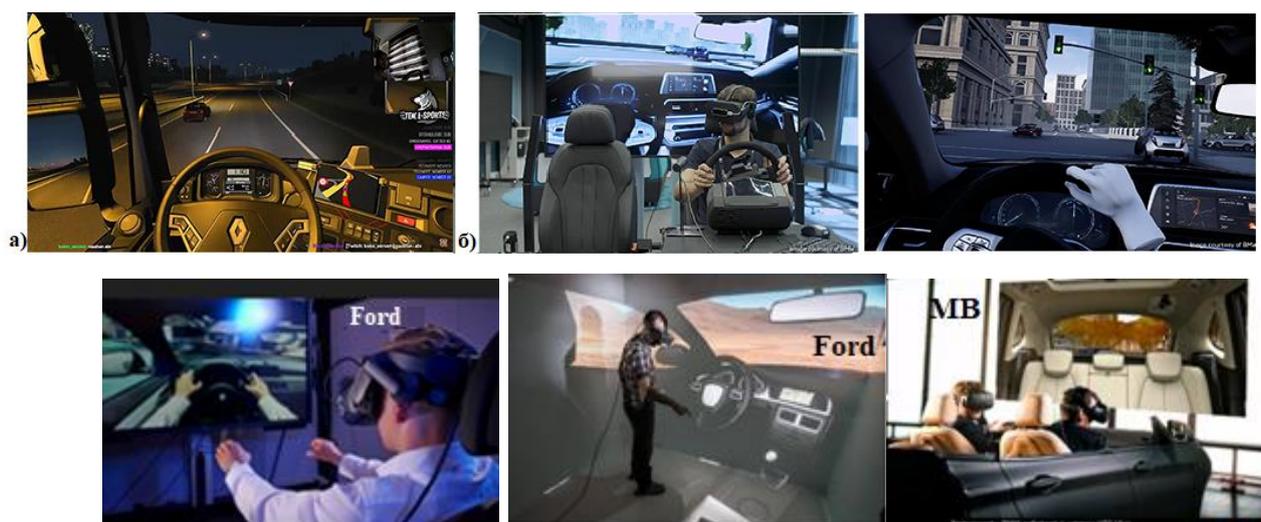


Рис. 2. Использование смешанной реальности в Unreal Engine в игре Euro Truck Simulator (а) и фирмами BMW (б), Ford, MB

Fig. 2. Use of mixed reality within Unreal Engine package in Euro Truck Simulator game (a) and by BMW (б), Ford, MB companies

Технология разработки игр *Unreal Engine* основывается на создании сцены, в качестве которой используется ландшафт изменяемой местности, где методом вытягивания участка можно создать возвышенности, низины, горы, проложить дорогу и разместить другие объекты, придав тем самым обстановке большую реалистичность. Но возможно также использовать готовые шаблоны. Далее следует создать 3D-модель автомобиля, конвертировать ее в формат *.fbx Unreal Engine*, разместить на трассе, расположив в автомобиле человека и камеру показа в объекте. Далее путем установки связей и настройки визуальных компонентов можно активизировать их действия управления автомобилем. Для этого потребуется программирование компонентов в *MS Visual Studio 2017* на *C/C++*, включая снятие информации с рулевого колеса, педалей газа, тормоза, рычага переключения КПП.

В настоящее время, в связи с отсутствием специальных исследований, нельзя с определенностью утверждать, какими средствами лучше это решать: *Luforia* или *Unreal Engine*. Преимуществом *UE* является возможность смоделировать различные условия эксплуатации и проработать варианты внешнего вида, внутреннего интерьера и эргономики машины на компьютерных моделях без реализации в металле, тем самым снизив затраты. Именно на это направлено основное внимание зарубежных специалистов в *VR/MR*. На видео [7] (рис. 2) показано использование пакета *Unreal Engine 4.1* на *BMW*: демонстрируется внутренний дизайн автомобиля, возможности движения рук, моделирование движения на стенде при езде по городу. Однако такое моделирование дает слишком плавные колебания. Следует указать, что в 2018 г. еще не было трекеров (датчиков перемещений) и ПО для них: они появились только в 2020 г. Таким образом, преждевременно оценивать важную для эргономики корректность определения перемещений руки.

В версии *UE 5.1* нет возможности задания вибраций и можно предположить, что воздействие было задано из внешнего источника (другого компьютера). Перемещение автомобиля задается в пакете и связано с трассой. При этом неясно, определяет ли *BMW* на основе *UE* долговечность машины, которая связана со скоростью движения (режимом движения) и микропрофилем дороги либо все что делается только для проработки дизайна и эргономики. Отметим, что из-за падения производительности при передаче информации на стенды по сети *CAN BUS*, моделирование в режиме реального времени требует разнесения вычислений на несколько компьютеров (рис. 3).

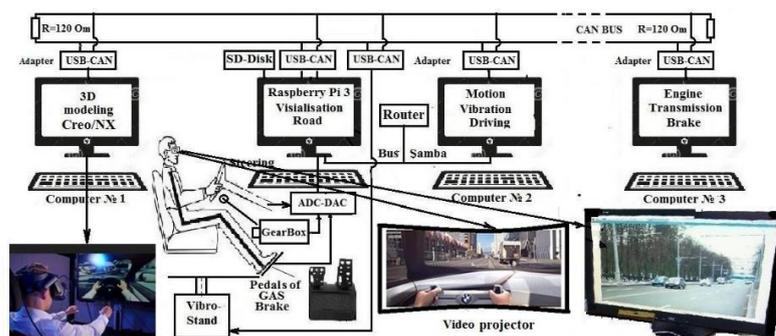


Рис. 3. Предлагаемая схема реализации моделирования с MR-реальности

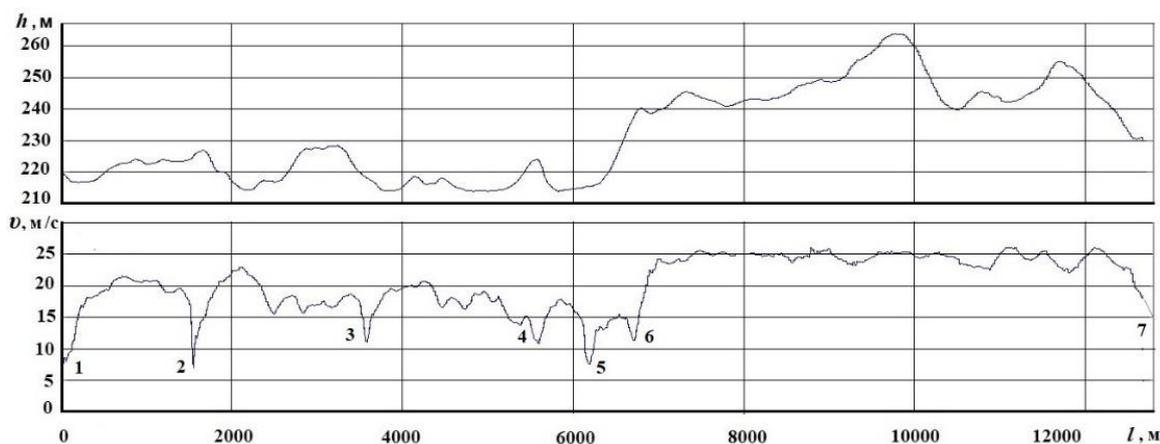
Fig. 3. Proposed schematic view of simulation using MR reality

Как показывает анализ, пакет *Unreal Engine*, используется в основном для визуализации внешней и внутренней обстановки машины. Для визуализации внутренней обстановки используются 3D-модели основных *CAD*-пакетов: *Autodesk Inventor*, *Autodesk 3ds Max*, *Dassault Systèmes CATIA*, *PTC Creo*, *Siemens NX* и др., а для внешней – контент гоночной трассы, городских условий. Но пакет очень сложен в освоении и требует работы в команде 5-7 чел.: проектировщиков, программистов и дизайнеров. Возможности пакетов *3D max* с *Unity*

(на C#) слабее, в них нет точных размеров. Для моделирования целесообразно использовать один язык C/C++ для всех компонентов цепочки.

Использование шлема и простейшего рабочего места водителя в сочетании с вибростендом позволяет максимально погрузить водителя в среду реального движения по дороге. Без шлема, вибростенда, рабочего места и звукового воздействия нет полного ощущения реального движения. Реализация такого комбинированного моделирования требует применения более совершенных моделей колебаний, движения и управляемости автомобиля с использованием реального совместного макро и микропрофиля дороги [2]. Важную роль играет компьютерная визуализация дорожной обстановки, являющаяся ключевым моментом имитационного моделирования. Она влияет на режим движения автомобиля, его нагруженность, долговечность, но она должна рассматриваться в сочетании с движением по реальной, а не идеальной по ровности и профилю дороги, реализуемой в *Unreal Engine* у *BMW* и других фирмах. Иными словами, только если должен будет задействован реальный макро- и микропрофиль дороги, можно оценить реальные параметры автомобиля: вибронгруженность, тягово-скоростные свойства, топливную экономичность, долговечность с учетом фактора водителя. Тогда на их основе путем моделирования в стендовых условиях можно отработать машину.

Одной из сложных проблем является более реалистичское компьютерное воспроизведение видео дорожной обстановки в шлеме в соответствии переменной скорости движения, зависящей от условий дорожной обстановки, характера дороги и возможностей автомобиля (рис. 3), имеющих сложный характер. В пакете *Unreal Engine* есть возможность показа видео, но не известно, можно ли подключить собственную программу управляемого видео и как объединить сцену с видео.



1 – старт, 2 – искусственное препятствие, 3 – поворот, 4 – ограничение v_a , 5 – выбоина, 6 – узкий мост

Рис. 4. Изменение скорости движения автомобиля от пройденного пути l , высоты (h) продольного профиля дороги и дорожной обстановки

Fig. 4. Change in the speed of a vehicle based on the distance traveled (l), the height (h) of the longitudinal profile of the road and the traffic situation

Как показало изучение возможностей шлема НТС, за счет настройки его программного обеспечения можно воспроизводить в шлеме видео дорожной обстановки, транслируемое с компьютера [8, 9]. Нерешенным вопросом остается реализация управления видео и воспроизведение его в соответствии со скоростью моделирования автомобиля. В стандартных пакетах видео таких возможностей нет, и нельзя однозначно утверждать, можно ли обойтись без пакета *Unreal Engine*.

Целью данной работы является анализ вопросов визуализации имитационного моделирования на основе создания программы управляемого видео при использовании комбинированного подхода смешанной реальности и моделирования.

1. Подходы к реализации видео.

Видео является важным компонентом реализации имитационного моделирования, поскольку водитель выбирает скорость движения и управляет автомобилем, исходя из его возможностей и дорожных условий, включая визуальные:

- наличие полос движения, потока машин, проведение ремонта дорог;
- ограничение скорости на отдельных участках, в населенных пунктах;
- повороты дороги, подъемы, спуски, при которых приходится снижать скорость;
- движение встречных и попутных автомобилей;
- состояние дороги (неровности, ее микропрофиль);
- ситуацию на обочине, движение пешеходов и др.

Для реализации видео возможны два пути.

1. Использование компьютерной графики, имеющейся в игровых, мультимедийных пакетах *Unity*, *Unreal Engine* [6, 7]. В них можно, привлекая художников, создать свою трассу или проложить трассу на основе вначале изменяемого пустынного переменного ландшафта, по обочинам трассы расставить деревья, различные объекты или использовать предлагаемую кольцевую гоночную трассу либо на основе городских условий США [7]. *Unreal Engine* довольно мощный, но и сложный пакет объемом до 21 Гб. Для реализации задач моделирования автомобиля, потребуется еще программирование в C# либо C++ в зависимости от пакета помимо использования их визуальных компонентов, а также моделирование в *Simulink*.

В обоих упомянутых пакетах отсутствуют возможность задания продольного и микропрофиля дороги и синхронизации их с дорожной обстановкой, что необходимо для комплексного моделирования автомобиля. Отметим, что условия на дорогах в СНГ значительно отличаются: преобладают дороги с невысоким качеством, двухполосным движением, перепадами высот, что является важным фактором, усложняющим моделирование.

На освоение этих игровых пакетов с нуля и реализацию программ с программированием в C# и C++ данных задач потребуется до двух лет. Кроме того, вне этих пакетов необходима отдельное моделирование движения, колебаний и управляемости автомобиля с учетом реального продольного и микропрофиля в *Simulink*, а также интеграция их с игровыми пакетами.

2. Запись видео дорожной обстановки при постоянной скорости движения автомобиля по дороге и с помощью разработанных программ для *S-Function Builder* в *Simulink* и видео, синхронизировать перемещение автомобиля при моделировании и воспроизведение видео. Для этого потребуется разработка специальной программы воспроизведения видео, управляемой с компьютера, осуществляющего моделирование движения в *Simulink*.

Учитывая ограничительные моменты в пакетах *Unity*, *Unreal Engine*: наличие нереалистической картины дороги, неясность синхронизации пройденного пути с изображением обстановки и сложность создания таких программ для решения задачи моделирования, – предпочтителен второй комбинированный подход. Он проще и использовался для создания программы реализации управляемого видео использован *MS Visual Studio 2022* с установкой на него библиотеки *Open CV 4.6.0*.

2. Выбор и установка программного обеспечения.

Первоначально в работе [2] для просмотра видео использовался *Raspberry Pi 3b* с программой, созданной на языке C++, где не было достаточно корректного управления. Это обеспечивало разрешение только 1280×720, а на его операционную систему нельзя было установить ПО шлема, необходимое для трансляции видео с компьютера. Поэтому для функционирования шлема использован компьютер с Windows 10, обеспечивающий разрешение 1920×1080 и работу ПО шлема. Дополнительно потребовалось создать новую программу просмотра видео в *Windows* с использованием библиотеки *OpenCV*. Эта библиотека функци-

онирует только на *C++* или *Python*. Поэтому для разработки программы на *C++* выбран пакет *MS Visual Studio 2022* версии 17.4, для которого имеется готовая библиотека *OpenCV 4.6.0* (*opencv-4.6.0-vc14_vc15.exe*) в виде бинарных файлов, что значительно упрощает ее установку и подключение к *MS Visual Studio* [9, 10].

Вначале необходимо загрузить *MS Visual Studio 2022* и библиотеку *Open CV 4.6.0* из интернета и воспользоваться рекомендациями работ [9, 10] по их установке. Далее следует воспользоваться процедурами, данные в инструкции на шлем НТС для приложения *Vive Home* для воспроизведения видео с компьютера [10], они приведены ниже.

1. Нажмите кнопку «Меню» и выберите «Объекты».
2. Выберите .
3. Выберите , а затем найдите желаемую видеозапись на компьютере.
4. Используйте элементы управления для воспроизведения или приостановки видео, регулировки громкости звука и т. д.
5. Выберите , чтобы зафиксировать видеопроигрыватель таким образом, чтобы он перемещался вместе с вашим взглядом.
6. Выберите , чтобы приглушить освещение во время просмотра видеозаписи.
7. Выберите , чтобы просматривать видеозапись с полным эффектом присутствия в виртуальной реальности.

Подобное описание можно найти в документации на шлем *Oculus*. На основе этого ПО осуществлялось реализация видео со смешанной реальностью в совокупности с моделированием движения, колебаний и управляемости автомобиля.

3. Разработка программы видео.

На основе анализа компьютерного моделирования [2], функций *Open CV* сделан вывод, что для управление скоростью воспроизведения видео целесообразно использовать функцию задержки *waitKey(zd)*. Особенностью этой функции является использование целого значения *zd*. Чтобы уменьшить погрешность при преобразовании *float* в *long*, берущей меньшее значение, использовалась функция *fmod(vd.zd,1)*, чтобы скорректировать значение до ближайшего целого числа. Скорость просмотра видео регулируется варьированием значений задержки. Предложено формировать бинарный файл обмена информации (пройденный путь, скорость, величина задержки) в *S-Function Builder*, который считывается в программе видео и регулирует скорость воспроизведения видео. Ниже приведен исходный текст разработанной программы воспроизведения видео на *C++* в *MS Visual Studio 2022*.

```
// CASHv.cpp – program of implementation of video for modeling of vehicle
```

```

#define _CRT_SECURE_NO_WARNINGS
#include <opencv2\opencv.hpp>
#include <opencv2\highgui\highgui.hpp>
#include <opencv2\imgproc.hpp>
#include <iostream>
#define _CRT_SECURE_NO_DEPRECATED
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <share.h>
using namespace std;
using namespace cv;

struct d {
    long dl; // length of a way
    float v; // speed of vehicle
    float zd; // delay in mc

```

```
};
struct d vd;
long ch = 0;
int main()
{
    struct d vd;

    FILE* fp;
    long FPS, zd, zd0;
    float sv;
    int a, zd5;
    long sumlv; // sum streng on speed of vehicle    delay in mc
    bool bSuccess;
    int FramesSkipped = 1;
    ch = sumlv = 0;
    sv = 0.0;
    vd.zd = 20; //inset preliminary values
    zd = 20;
    a = 1;

    VideoCapture cap("road.mp4"); // Creating VideoCapture object
//and opening input file of vidio
    if (!cap.isOpened()) //Checking of opening input file of Video
    {
        cout << "Error opening video stream or file" << endl;
        return -1;
    }
    Mat frame; // Creating Mat object
    a = 1;
    vd.zd = 40.0; // preliminary value
    while (a == 1)
    {
        if (ch == 10)
        {
            fp = _fsopen("ma.bt", "rb", _SH_DENYNO);
            fread(&vd, 12, 1, fp); // reading file of a time delay
            ch = 0;
// Set limits of valves
            if (vd.zd > 80.0)
                vd.zd = 80.0;
            if (vd.zd < 5.0)
                vd.zd = 5.0;
            fclose(fp);
            zd = zd0 = vd.zd;
            if (fmod(vd.zd,1)>=0.5) //define the closest integer of a delay
                zd = zd + 1;
        }
        ch++;
        cap >> frame; // Capture frame-by-frame
        if (frame.empty()) // If the frame is empty, break immediately
            break;
    }
}
```

```

imshow("roadFrame", frame); // Display the frame
char a = (char)waitKey(zd); //Press ESC on keyboard to exit from Video
if (a == 27)
    break;
} // End while
return 0;
}

```

В данной программе из файла обмена, формируемого в модуле *S-function Builder Simulink*, считываются целые значения величины задержки `waitKey(vd.zd)` в мс для обеспечения необходимой скорости воспроизведения видео. Дополнительно потребовалось установить зависимость значения задержки в мс, которая обеспечивает необходимую скорость вывода видео от скорости движения. Чтобы не загружать процессор для считывания этой информации, использован счетчик циклов `ch`, который после определенного числа обнуляется и производит заново дальнейший подсчет циклов. При достижении этого числа (интервал 0,3 с) происходит считывание данных пути, скорости и необходимой задержки.

Важным моментом является запись видео. Целесообразно проводить запись в разрешении 1920×080 при 30 fps при постоянной скорости 20 м/с . Выбор этой скорости обусловлен возможностью достижения верхних значений $30\text{-}40 \text{ м/с}$ при визуализации, который получается за счет времени задержки воспроизведения видео. Как выяснилось, на это влияет также объем файла. В силу этого, при первоначальном объеме файла $1,1 \text{ Гб}$, соответствующего участку длиной 12 км , на первом этапе не удалось обеспечить требуемый диапазон регулирования скорости ($v=30 \text{ м/с}$). Поэтому файл сжали в пакете *Movavi Video Converter* до 500 Мб . Экспериментальным путем обработки времени просмотра видео определенного участка пути установлена зависимость значения задержки в мс, которая обеспечивает необходимую скорость вывода видео (рис. 5). Моделирование движения, колебаний, управляемости, топливной экономичности автомобиля и видео без вибростенда осуществлялось в *Simulink (r15b)* с помощью блок-схемы (рис. 6) на базе отработанных решений [2].

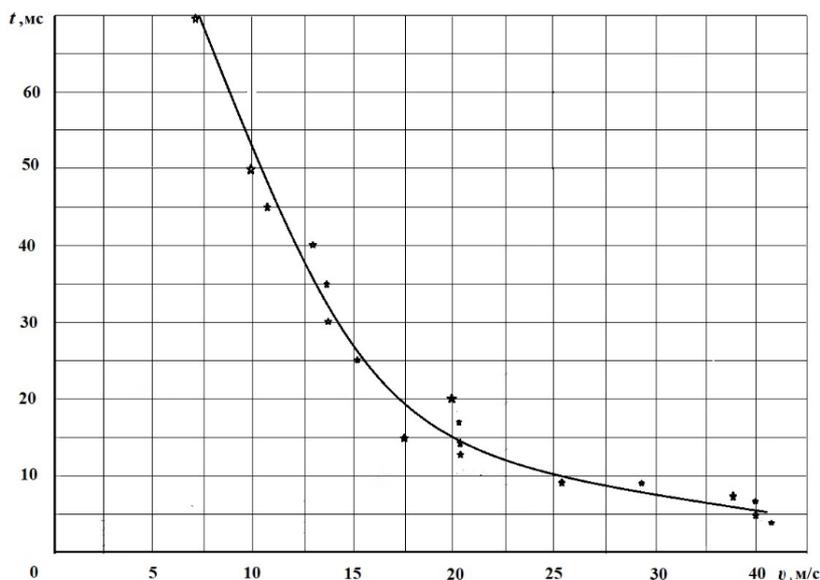


Рис. 5. График изменения времени задержки, необходимого для обеспечения соответствия просмотра видео скорости движения автомобиля

Fig. 5. Curve of the delay time needed to match the video to the speed of the vehicle

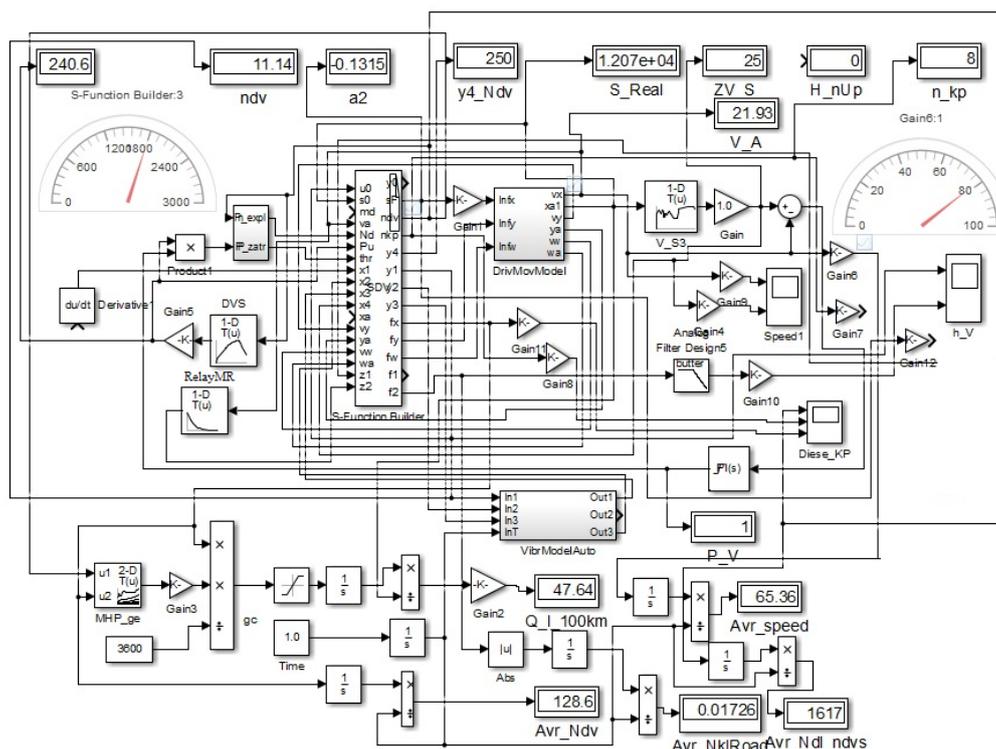


Рис. 6. Блок-схема реализации имитационного моделирования с AR/MR в Simulink

Fig. 6. Flowchart for implementing simulation with AR/MR in Simulink

Использовался компьютер в составе *AMD Risen 5 2600*, 3,4 ГГц, *ASUS B485 Plus*, ОП 16 Гб, *P5 Plus M.2 (6000 МГц)*, *GeForce 710*. В случае использования 6-DoF платформы на электроактуаторах в качестве вибростенда потребуется применение более быстродействующих компьютеров, разнесения моделирования на компьютеры в виду значительного падения производительности (до 1,5 раз) вследствие сети *CAN BUS*.

С помощью модуля *RelayMR* осуществлялось преобразование скорости в величину задержки, а через *S-Function Builder* производилась запись данных пройденного пути и задержки в виде бинарного файла, который затем считывался в программе видео, и на основе этих данных реализовалось управление скоростью просмотра видео.

Для подтверждения предложенного метода сравнивалось совпадение прохождения конечной точки пути и визуального объекта на ней, показавшее расхождение по времени 3-4 % при средней скорости 65 км/ч, расходе топлива 47 л/100 км, $m_a = 20\ 000$ кг, что вполне приемлемо для длины участка (12 км) и свидетельствует о корректности метода. Дополнительно проведено сравнение при уменьшении режима скоростей при $K=0,7$ ($v_a = 47,4$ км/ч, 42,7 л/100 км), показавшее снижение по v_a в 1,38 раза против 1,42 заданного, что также свидетельствует о корректности предложенного метода.

Проведенное исследование показало, что пакет *Unreal Engine* не является необходимым при моделировании движения, колебаний и управляемости с максимальным погружением испытуемого в обстановку движения. Однако решение эргономических задач требует использования *Unreal Engine* или *Luforia*, а также трекеров с их программным обеспечением.

Заключение

1. Проанализированы используемые пакеты для *MR*, пути реализации видео дорожной обстановки при моделировании движения, колебаний и управляемости автомобиля при использовании смешанной реальности.

2. Отмечена сложность реализации дорожной обстановки при смешанной реальности в пакетах *Unity*, *Unreal Engine*, где присутствуют нереалистическая картина дороги и трудности с визуализацией и синхронизацией пройденного пути и скорости движения.
3. Предложен программный метод визуализации дорожной обстановки на основе записанного в дороге и обработанного видео, которое воспроизводится в шлеме виртуальной реальности в соответствии со скоростью моделирования автомобиля в пакете *Simulink* с помощью разработанной программы управляемого видео на основе определенной функции значений задержки от скорости движения.
4. Установлена графическая зависимость значений задержки от скорости движения, обеспечивающая управление скоростью просмотра видео в соответствии скорости моделирования автомобиля.
5. Получено приемлемое расхождение (3-4 %) прохождения контрольной точки пути и видео при использовании предложенного метода, подтверждающее его корректность.

Библиографический список

1. Mercedes Benz TecDay 2010 Driving Simulator 2 [Электронный ресурс]. Режим доступа /https://www.youtube.com/watch?v=CxIo11_YsK0 Video: Mercedes-Benz Driving Simulator Is Industry Leader (дата обращения 15.01.2023).
2. **Михайлов, В.Г.** Компьютерное и имитационное моделирование автомобиля / В.Г. Михайлов. – Минск: Белнаука, 2020. – 316 с.
3. **Blissing, V.** Driving in Virtual Reality Requirements for Automotive Research and Development // Dissertations No. 2085, Linköping Studies in Science and Technology, Division of Machine Design Department of Management and Engineering Linköping University, SE-581 83 Linköping, Sweden, 2020 [Электронный ресурс] // Режим доступа: <https://www.diva-portal.org/smash/get/diva2:1459751/FULLTEXT01.pdf>. (дата обращения 15.01.2023).
4. **McGill, M.** Challenges in passenger use of mixed reality headsets in cars and other transportation / M. McGill, J. Williamson et al [Электронный ресурс] // Режим доступа: <https://link.springer.com/article/10.1007/s10055-019-00420-x> / Published: 18 December 2019.
5. Augmented Reality (AR) in Creo [Электронный ресурс] // Режим доступа: https://www.irisoft.ru/ptc_files/Datasheet-Augmented_Reality_In_Creon.pdf?ysclid=14qo3kxnp926975499 (дата обращения 15.01.2023).
6. Unreal Engine 5.1 Documentation [Электронный ресурс] // Режим доступа: <https://docs.unrealengine.com/5.1/en-US> (дата обращения 15.01.2023).
7. BMW brings mixed reality to automotive design with Unreal Engine [Электронный ресурс] // Режим доступа: <https://www.unrealengine.com/fr/spotlights/bmw-brings-mixed-reality-to-automotive-design-with-unreal-engine> / 24/04/2018 (дата обращения 15.01.2023).
8. Install and configure OpenCV-4.2.0 in Windows 10 – VC++ [Электронный ресурс] // Режим доступа: <https://towardsdatascience.com/install-and-configure-opencv-4-2-0-in-windows-10-vc-d132c52063a1> (дата обращения 15.01.2023).
9. How to Install Visual Studio 2022 [Электронный ресурс] // Режим доступа <https://www.c-sharpcorner.com/article/how-to-install-visual-studio-2022/> (дата обращения 15.01.2023).
10. Воспроизведение видео в приложении Vive Home [Электронный ресурс] // Режим доступа <https://mcgrp.ru/files/viewer/296494/70?ysclid=lbahqxvd5s791512283> (дата обращения 15.01.2023).

**Дата поступления
в редакцию: 28.02.2023**

**Дата принятия
к публикации: 02.05.2023**