Simulation System for Assistance in Driving using Force Feedback on Direction and Acceleration Commands

PAUL ROMERO, GABRIEL LOPEZ, NELSON SOTOMAYOR, DANilo CHAVEZ
Escuela Politecnica Nacional
Departamento de Automatizacion y Control
Ladron de Guevara E11 - 253. Quito
ECUADOR
paul_romero_b@outlook.com, gabriel.lopezl@yahoo.com
nelson.sotomayor@epn.edu.ec, danilo.chavez@epn.edu.ec

Abstract: In this project, the design and construction of a simulator to avoid potential driving collisions is presented. These potential driving collisions are avoided by generating a force feedback through acceleration and direction commands. With this objective, a testing cockpit has been designed taking into consideration aspects related to ergonomics and setup of elements like a real car. The simulator was created using the free version of the software “Unity”, additionally, algorithms of path tracking and collision probability evaluation have been developed. These algorithms are responsible for the calculation of force feedback outputs, which allows users to acknowledge the possible collisions.

Key–Words: Simulator, Force Feedback, Micro controlled systems.

1 Introduction

With the invention of the auto-mobile, inevitably came car accidents and related injuries. Since the creation of the auto-mobile, there have been many advances in the automotive industry to improve efficiency and speed. The initial concern of the companies’ vehicles was not security, but to build a better and more powerful vehicle, therefore, they became faster and more dangerous. Finally, almost 150 years of the creation of the first auto-mobile security elements began to play a leading role in the auto-mobile industry. Thus driving assistance systems were born, aiming to avoid possible impacts due to the distraction or inattention of the driver [1].

Driving simulators can reproduce the process of driving a vehicle in a virtual environment. They currently are used for research and to train drivers in different categories: novice drivers, improvement of professional drivers, drivers with special needs and older drivers. Usually, 3D displays and 3D sound devices are enough to create a sense of immersion, but in more complex applications the user is given the possibility to interact with the virtual environment [2].

Haptic devices are those that allow users to touch, feel or handle simulated objects in virtual environments and teleoperated systems. They establish a two-way transfer of information between the user and the virtual environment in real time [3].

2 Software

2.1 Architecture

Unity is one of the most important components in this project because through it, the virtual environments are created, as well as the simulation of car movements, rendering, path tracking and the calculation of force feedback outputs. It also allows the user to communicate through the interface with the simulation program. A microcontroller aims to generate control signals for the force feedback system over the acceleration pedal, according to data received from the virtual environment. Figure 1

2.2 Virtual Environment Creation

In the simulator[11], a road and city-type environment has been created. Within these, there is a user controlled car and several obstacle cars following predefined paths. The cars behave according to the car-like dynamic model [6], ie: rear-wheel traction, front steering and braking on all four tires. NVIDIA PhysX
Engine simulates the behavior of these elements considering the masses, torques and forces applied. Figure 2.

2.3 Path Following Algorithm [4]

This algorithm ensures that the obstacle cars in the simulation of the scenes move along the established path. The steering angle of the front tires is constantly estimated so that the cars can follow the path appropriately.

The route followed by the cars is defined by a set of points located at various positions on the road. Furthermore, the “Current Path Point” is defined by the point at which the car should be oriented at that moment.

First the “SteerVector” is calculated to obtain the steering angle, whose origin is in the car and ends at the “Current Path Point”. Figure 3.

Then, the “Steer” factor is calculated by 1. The value 1 represents the deviation of the car to the “Current Path Point”, and it can be positive or negative.

Finally, the steer angle of the front tires is calculated using the equation 2:

\[ Steer = \frac{SteerVector_x}{\|SteerVector\|} \] (1)

\[ Steer\ Angle = Steer \times Max\ Steer\ Angle \] (2)

Where:
SteerAngle = Front Tires’ Steering Angle.  
MaxSteerAngle = Maximum Steering Wheel Angle.

In this simulator, the MaxSteerAngle is 40°, which means:

\[-40° \leq \text{SteerAngle} \leq 40°\]  \hspace{1cm} (3)

2.4 Collision Probability Algorithm [10] [8] [9]

A LIDAR (Light Detection and Ranging) sensor has been simulated and it determines the distance from a laser transmitter to any object [7]. It is available for purchase (Ibeo LUX 8L) and perfectly fits the needs of distance and field of view for this driving assistant simulator. Figure 4.

To detect obstacles in the simulator, the LIDAR sensor scans the field and the rays that make contact with the obstacles are accumulated, and then averaged, to find the relative position, making the obstacle one defined point. Figure 5.

To calculate the speed of the obstacles, the relative position is evaluated in two different times: \(t_0\) and \(t_1\), where:

\[n = \text{Number of Obstacles.}\]
\[X = \text{Relative Position of the Obstacle.}\]
\[t_0 = \text{Time of the Previous Sweep.}\]
\[t_1 = \text{Time of the Current Sweep.}\]

\[
\vec{v}_n = \frac{X_{n1} - X_{n0}}{t_1 - t_0} \quad \hspace{1cm} (4)
\]

It is known that \(t_1 - t_0 = \Delta t = 0.16\text{s}\), which corresponds to the sample time of the LIDAR sensor [5], therefore:

\[
\vec{v}_n = \frac{X_{n1} - X_{n0}}{0.16\text{s}} \quad \hspace{1cm} (5)
\]

With the current speed and position of each obstacle, 20 trajectory points that each obstacle will follow are estimated within a period of 4 seconds.

\[
X_{n\text{m}} = X_{n0} + \vec{v}_n \times m \times 0.2\text{s} \quad \hspace{1cm} (6)
\]

\[m = 1, 2, 3 \ldots 20\]
Among all the points on the trajectory calculated by the equation above, the set of them which represents the greatest danger to the car is found, according to the expression:

\[ A = \left\{ X_{ntm}^- \mid \min (\|X_{ntm}^-\| \leq 2m) \right\} \quad (7) \]

And from these obstacles, the one which would cause a collision in the shortest time is selected to calculate the force feedback.

### 2.5 Force Feedback Functions

Relevant parameters were considered in attempt to evade obstacles, such as: the current position of the obstacle, the approximate time at which the collision would occur, and how close to the car’s center the obstacle would impact.

#### 2.5.1 Steering Wheel Force Feedback

- **Current position of the obstacle on the x axis:** This factor allows the algorithm to easily discern the direction that should have the force feedback, i.e: if the obstacle is on the right, then the force exerted by the wheel must be counter-clockwise, telling the user that the steering wheel needs to turn to the left to avoid the obstacle.

\[ f(x) = \begin{cases} 
-1 & X_{obs,x} \geq 0 \\
1 & X_{obs,x} < 0 
\end{cases} \quad (8) \]

- **Obstacle Distance:** An implemented function that returns a low percentage of feedback if the obstacle is far away (40m), and a higher percentage of feedback if the obstacle is near the vehicle.

\[ f(d) = -0.025 \times \|X_{obs}^-\| + 1 \quad (9) \]

- **Approximated time in which a collision could occur:** Similarly, this function provides a low percentage of feedback if the estimated time that a potential collision could occur is long (4s), and a higher percentage of feedback if this time is short.

\[ f(t) = -5 \times \frac{t_{X_{obs}^-}}{0.2s} + 100 \quad (10) \]

Finally, the total output is the product of these three functions by a factor of 2, and a saturator at 100 % is added to prevent feedback percentages outside the valid range. Figure 6.

\[ Output = 2 \times f(x) \times f(d) \times f(t) \quad (11) \]

![Figure 6: Steering Wheel Force Feedback Function](image)

#### 2.5.2 Accelerator Pedal Force Feedback

- **Ultimately, it responds to the estimated time in which a potential collision could happen, sending a high percentage if the time is short, or sending a low percentage if the time is long.**

\[ Output = 4 \times (-5 \times \frac{t_{X_{obs}^-}}{0.2s} + 100) \quad (12) \]

This function does not require many considerations, like the steering wheel, because it performs only in one direction.

In addition to this function, a saturator at 100 % is added to allow only admitted values.

### 2.6 Microcontroller

The electronic system designed is intended to control the DC motor output torque, which allows the force feedback on the accelerator pedal. This is done through a current ripple control, for this purpose a DC-DC buck converter has been designed.

There is a microcontroller that is responsible for performing this function, it receives data from the simulator and set the minimum and maximum limits of the current values.
3 Hardware

3.1 Cockpit

The structure is made of metal. It consists of a car seat, the steering wheel with force feedback, the gearshift, a display, pedals and the system for the force feedback in the accelerator. This structure is protected with metal sheet to safeguard the system. Figure 8.

3.2 Steering Wheel Force Feedback System

The Logitech G27 [12] was selected. This commercial gaming wheel allows to generate the force feedback via scripting [13] and it also generates forces to recreate the feeling of driving a real car.

3.3 Accelerator Pedal Force Feedback System

This system allows to generate the forces over the accelerator pedal. First a DC motor generates the torque, then it is multiplied by a set of pulleys and finally it is converted into a force through a pinion-rack coupling.

4 Results

4.1 Hardware Tests

4.1.1 Control Board

A series of tests have been performed with the DC motor. The work of each of the components forming the circuit was checked, and the output current ripple was verified in accordance with the values sent from the computer.

The Figure 9 shows the motor’s current waveform and the control signals.

4.2 Survey

A survey was carried out with the participation of different users, they aged from 18 to 53 years. A total of 53 people participated. Regarding gender, the sample consisted of 27 women and 26 men. The driving experience of the sample goes from zero to regular driver.

After making sure the participant felt comfortable in the cockpit, they were given an explanation of how to use the simulator. The test lasted about 10 minutes giving the participant the chance to try the simulator for a reasonable time. During the test, the steering wheel and accelerator force feedback were enabled.

4.3 Results

In the Figures 10, 11 and 12 are the answers given by the users after trying the driving assistant.
5 Conclusions

The driving assistant designed, with obstacle detection algorithms and force feedback in the commands of direction and acceleration, allows the driver to take the necessary actions to avoid a possible collision.

The use of path tracking algorithm to move the obstacle cars on the track, allows users to test the system and force feedback in a similar environment to the one that might be found in real life.

The electromechanical system that generates the force over the accelerator pedal, has been designed to easily perceive the force by the drivers. Further, it is contemplated that the action of this force decreases progressively, so that the driver will not feel startled.

The use of forces on the steering wheel to recreate the feeling when driving a real car, facilitated that the time needed to adapt to the simulator was short.

The driver should be the one that controls the vehicle all the time, for this reason the system never acts on the brake pedal and force feedback warnings are designed only to alert. It is the same driver who takes the corrective action.

The use of a haptic-type response, means the user can perceive with his senses, in this case the touch, the direct interaction with the virtual environment. This allows users to have an experience where warning interface is transparent and forms part of the same system that allows driving and controlling the vehicle.

Offering the users the possibility of enabling and disabling the force feedback on the accelerator pedal or in the steering wheel allows each user to configure it according to their particular needs.

References:


