

LABVIEW® SIMULATION OF AN ELECTRIC VEHICLE WITH FORCE FEEDBACK CONTROL

Narciso J. Gomez

Florida International University
Miami, Florida, USA

Sabri Tosunoglu

Florida International University
Miami, Florida, USA

ABSTRACT

A LabVIEW® simulation computer program was developed, that allows the user, by interacting through the computer mouse with a virtual control, to move, with adjustable speeds, a “vehicle” in a two-dimensional space that includes an inner “obstacle”, presented in the computer screen. The slider-type control implemented has a simulated force feedback capability, which increases linearly the “force” applied on the user’s “fingers”, according to the distance from the vehicle to the “walls” of the space. The program presents four virtual oscilloscopes, displaying simultaneously with the movement of the vehicle, the following signals: the speed-controlling supplied at its two “motors”, the output of its two “distance sensors”; and, at the virtual control, the force-controlling at its “linear motor”, and the output of its “position sensor”. This paper reports the details of the simulation program and the basic physical design of both the vehicle and the force feedback control.

OBJECTIVE

The objective of this project was to develop a program to simulate the functioning of a system consisting of an electric vehicle and its remote control. The minimum functions that this program must accomplish are:

- To display the animated two-dimensional motions of a simple drawing (the “vehicle”) in the computer screen.
- The orientation, direction, and speed of these motions are to be controlled by the user, by using the mouse to interact with graphical controls that resemble actual devices. The location and speed of the vehicle is to be continuously displayed in the screen.
- The space where the vehicle is to be moved must be limited by “walls” and include an inner “obstacle”.
- The user-commanded control must have a simulated “force feedback” capability, inversely proportional to the distance from the vehicle to an obstacle in front of it.
- All the electric signals existing at the “components” of both the vehicle and the “control”, are to be shown in the screen, updated continuously according to their status.

Also, to avoid simulating a vehicle and a control that would not be possible to bring into reality, part of this pro-

ject is also to design both devices to a sufficient extent as to demonstrate its feasibility.

The purposes of this report are:

- To present the details of the development, use and limitations of this simulation program.
- To describe the conceptual design of the simulated vehicle and the remote control, including the details of the method that would be used to generate the required force feedback at the control.

INTRODUCTION

The following concepts are involved in this project.

Teleoperation: Technology applied when the controls of a machine are not directly attached to it. The method of communication between them are either a cable or, more recent and convenient, a wireless transmission [1, 2].

Remote Control Vehicle: A teleoperated mobile device. An RCV is not a robot, because it is not capable of operating without external commands. The autonomic operation is a necessary condition, but not sufficient, for a machine to be classified as a robot: its working needs also to be modifiable within certain range of alternatives, thus, a clockwork toy is not a robot [2].

Telesensation: A capability of certain teleoperation systems, in which the controlled machine is able to send certain information back to the distant control in real-time, which is also prepared to use this information to present to the senses of the operator some stimuli, related to the current environment where the machine is located [1].

Force Feedback Control: A subcategory of “haptic” (pertaining to the sense of touch) telesensation, where the stimulus is a force applied to the part of the body of the operator interacting with the control [1, 2].

LabVIEW: A program development software, where the code is not written but rather created by assembling a flowchart using blocks representing a wide variety of functions, connected by lines that indicate their inputs and outputs. The user of the program so developed, interacts with it through *virtual instruments*, and the program can then interact with the real world, via modules to receive measured data sent by instruments and/or to send control commands to external equipment [2].

Virtual Instruments: Simulated representation of real indicators, like numerical displays, dials, and graphs resembling oscilloscope screens; and controls, such as knobs and switches, in the computer screen, where the user can interact with them, visually reading the measured quantity from the indicators, or setting the controls using the computer mouse or keyboard [3].

Linear Motor: An AC motor that generates a linear force instead of a torque, as the conventional rotational motors [2]. The motor selected for this project, consists of a round-section straight shaft containing permanent high-energy magnets; this bar generates the magnetic field and guides the linear displacement of a rectangular block with a circular hole, called the forcer because it generates the thrust by encircling the shaft with its energized coil [4].

Hall Effect Sensor: “A transducer that varies its output voltage in response to changes in a magnetic field” [2]. It can be used to detect and measure several variables but, in this project, it is integrated inside the linear motor to provide the positioning information of the forcer.

DESCRIPTION OF THE SIMULATED SYSTEM

The program developed in this project, simulates a system consisting of a vehicle and its control, reproducing their interdependent operation. It will be shown that the configuration of the latter was defined specifically according to the features of the former.

The simulated vehicle (Fig. 1), consists basically of a rectangular plate with short sides rounded along circular arcs, of colors green and red, which are the front and rear ends (not respectively, as will be explained later) during the travels. This shape was defined for the plate, due to the ease it offers to be animated in the computer screen, but it also looks appropriate for an actual application. On top of this plate, there is space to install the distance sensors, circuits for control and communication, and the active elements needed according to the function assigned to the vehicle. On the bottom surface of the plate, two electric motors, to drive the wheels, and two ball casters [5] are attached. These ball casters are not driven, they act as

passive weight-supporting elements. These four parts are symmetrically located.

In order to achieve a simple design for the vehicle, which also leads to a simple control, no vehicle-turning orientation mechanism is used for its wheels. Therefore, the only way to turn this vehicle, is by generating a non-zero relative rotational velocity between its two motors, as seen from a single coordinate system. For simplicity, for this project it was selected that, while turning the vehicle, both motors would rotate, as seen from their own coordinate systems, at the same speed and direction (because they are located facing opposite directions), thus the voltages supplied to them would be identical during turns. This turning method keeps, theoretically, the coordinates of the center of the vehicle constant, with the advantage of allowing the operator to achieve accurate positioning and orientation, but reducing its agility. For this kind of turns, the type of passive support selected, the ball caster, working like the ball of a computer mouse, offers less resistance during the turns than a regular wheel that can rotate freely around a vertical axis besides the horizontal axis.

This vehicle would use two DC motors, which can develop adjustable rotational speeds within the same ranges in both directions. Effectively, the vehicle does not have a forward and reverse motion, because both can be done at the same maximum speed, and its plate is symmetrical; the direction of travel is rather defined by the color of the round edge, either green or red, that is in the front of the vehicle when it moves. To achieve these straight travels, opposite-sign voltages of the same magnitude are supplied to both motors simultaneously, and the inversion of travel direction is done by the inversion of the supply polarity to each motor.

The control of this vehicle (Fig. 2), consists of a block that the user displaces, by means of a handle that protrudes above the guiding plate of the control, to modify the speed of the vehicle: from the central point of its straight travel, which corresponds to zero speed, the farther the block is pushed toward the green arrow (where the slot, which defines the travel of the block, ends), the

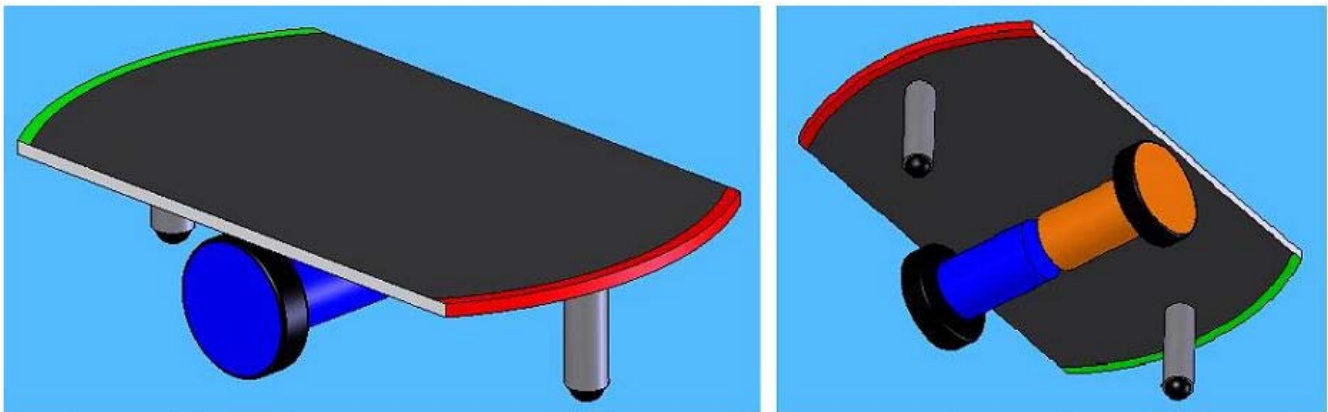


Figure 1. Top and Bottom Views of the Simulated Vehicle

higher the speed of the vehicle will be, moving with the green round end forward. In the same way, in order to move the vehicle with the red round end in front, the handle of the block has to be located between the neutral central point and the red arrow, achieving the maximum speed in this direction when the handle is at this end of the slot. In conclusion, this slider controls two of the three attributes of the velocity vector of the vehicle: its magnitude and direction. The displacement of the block is sensed, as will be described later, and this position information is then proportionally translated into a command to be communicated to the vehicle; this order will modify appropriately the voltages (regarding their magnitude and sign), that have to be supplied to both motors, so the speed of the vehicle will stabilize at the value selected by the user, which is the signed value indicated by the position of the handle respect to the scale adjacent to the guiding slot.

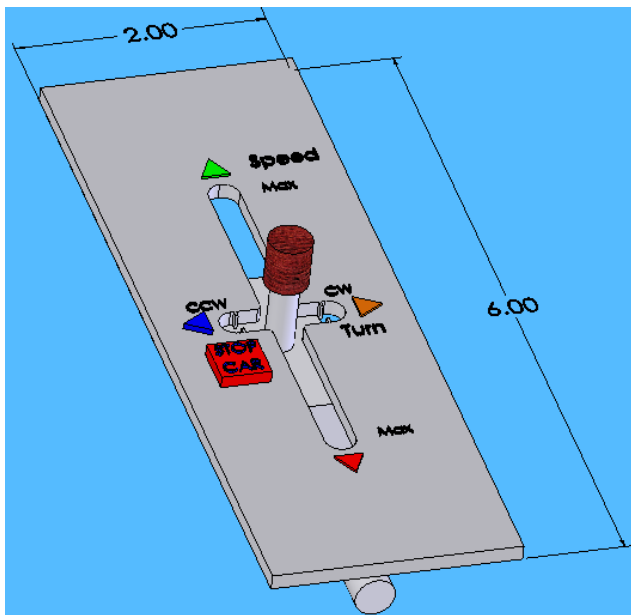


Figure 2. Simulated Control (dimensions in inches)

To set the third and last attribute of the vehicle velocity vector, this is, its orientation, the control gives the user the means to rotate the vehicle in either clockwise (CW) or counterclockwise (CCW) directions around a vertical axis that passes through the vehicle's center. The vehicle is not able to move while turning, as was explained before, so, to set the rotation direction, the control has, at the neutral central point of the speed-controlling slot, another slot, perpendicular to the other one and of much shorter length, both slots drawing a bisymmetrical cross (Fig. 2). Then, to make a turn while traveling, the user is forced by the shape of the cross slot to, first, reduce the speed to zero, and then, to select the direction of rotation, CW or CCW, by turning the handle accordingly, due to the freedom the system allows to this motion, as will be seen later. When

the handle reaches either end of the turn-controlling slot, the metal of the handle closes a circuit between two normally open contacts located at each end, as shown in this Fig. 2. In this moment, the closed circuit sends the appropriate commands from the control to the vehicle, making its motors to rotate in the directions required to achieve the direction of rotation selected. Evidently, in this scheme the vehicle would rotate around its vertical axis, in each direction, always at the same rotational speed; however the CW speed does not have to be necessarily equal to the CCW speed, but it is difficult to imagine an application where it would be advantageous to do so. The operator would keep the handle inclined, in contact with those terminals, until the desired orientation is reached. In this moment, the handle would be returned to the central neutral position, where a new speed and direction can be selected.

Several options were considered to select the method to generate the feedback force at the control's slider, ranging from a double action pneumatic cylinder to electric motors. Among them, the two alternatives more promising were a rotary electric motor driving a screw, which would push and pull a guided nut, and a linear magnetic motor. The latter was the selected solution due to its compact dimensions, low weight, low power consumption and simplicity. Other advantages are its noiseless and frictionless operation, because there is no contact between forcer and shaft. At any location of the forcer along the shaft, a force parallel to it can be developed in any direction and in any magnitude inside its designed range. The selected model, S080D from Nippon Pulse Motors [4], has a weight of 0.1 kgf, developing 1.8 N maximum constant thrust when consuming 0.8 A. This type of motor allows to freely rotate the forcer around the shaft, which is a necessary feature in order to select the direction of rotation of the vehicle as was explained before. Also mentioned previously, is the need to detect the position of the forcer in order to set the speed and direction of the vehicle, and this function is achieved by a Hall Effect sensor that is integrated inside the motor, thus eliminating the need of an external position sensor. The amount and direction of the force that is generated by this motor, is controlled by the current supplied to it. For this application, this supply is determined by the control according to the signal sent by the distance sensors located at the vehicle, so when it is approaching an obstacle, the operator will feel in the fingers that are moving the handle of the control, a force directed toward the central neutral position of the speed-controlling slot. This force feedback increases as the distance from the vehicle to the obstacle decreases, but would not be generated, regardless of how small is this distance, if the vehicle is not moving toward that object. For example, if it is very close to a wall and moving parallel to it, no force feedback would be felt. The operator would be able to overcome this resistance and maintain or even increase the approaching speed without damaging the motor.

The other function available in the control, is simply a button to stop the vehicle as fast as possible, by interrupting the power supply to its motors at the same time.

It is considered that, with the description and figures already given for the vehicle and the control, their feasibility has been sufficiently demonstrated.

DESCRIPTION OF THE SIMULATION PROGRAM

During this project, a simulation computer program was developed (Fig. 3), which allows the user to move, at adjustable speed, a simple drawing representing a vehicle, along any orientation and direction inside a bidimensional space displayed in the program's graphic user interface. To command this vehicle, the user interacts, by utilizing the mouse, with a virtual remote control with several functions, included also in this screen. Taking, as inputs, the vectorial information of the vehicle velocity (orientation, direction and speed) specified by the user, the simulation program generates two types of outputs: numerically, the x, y coordinates of the center point of the vehicle, and the orientation angle of its centerline; and graphically, using

four virtual oscilloscopes, the electric signals “generated by”, or “supplied to”, the components installed at both the vehicle (the “driving motors” and the “distance sensors”) and at the “remote control” (the “position sensor” and the “feedback motor”), simultaneously with the movement of the vehicle.

The virtual space where the vehicle can be moved, includes within its boundaries a rectangular obstacle that the vehicle cannot traverse. The insertion of this obstacle allowed, to the developer of this program, to study and mathematically model the effect of convex corners (as opposed to the concave corners created by the boundaries of the space) over the motions of the vehicle, over the signals generated by its distance sensors, and over the force-controlling signal supplied to the control's linear motor; and allows the user to experiment with the generation and application of these signals in all possible situations, as will be detailed later. The space where the vehicle can be moved displays, as references, the coordinates of the top left (which is the origin of the coordinate system for this software) and bottom right corners of both this space and

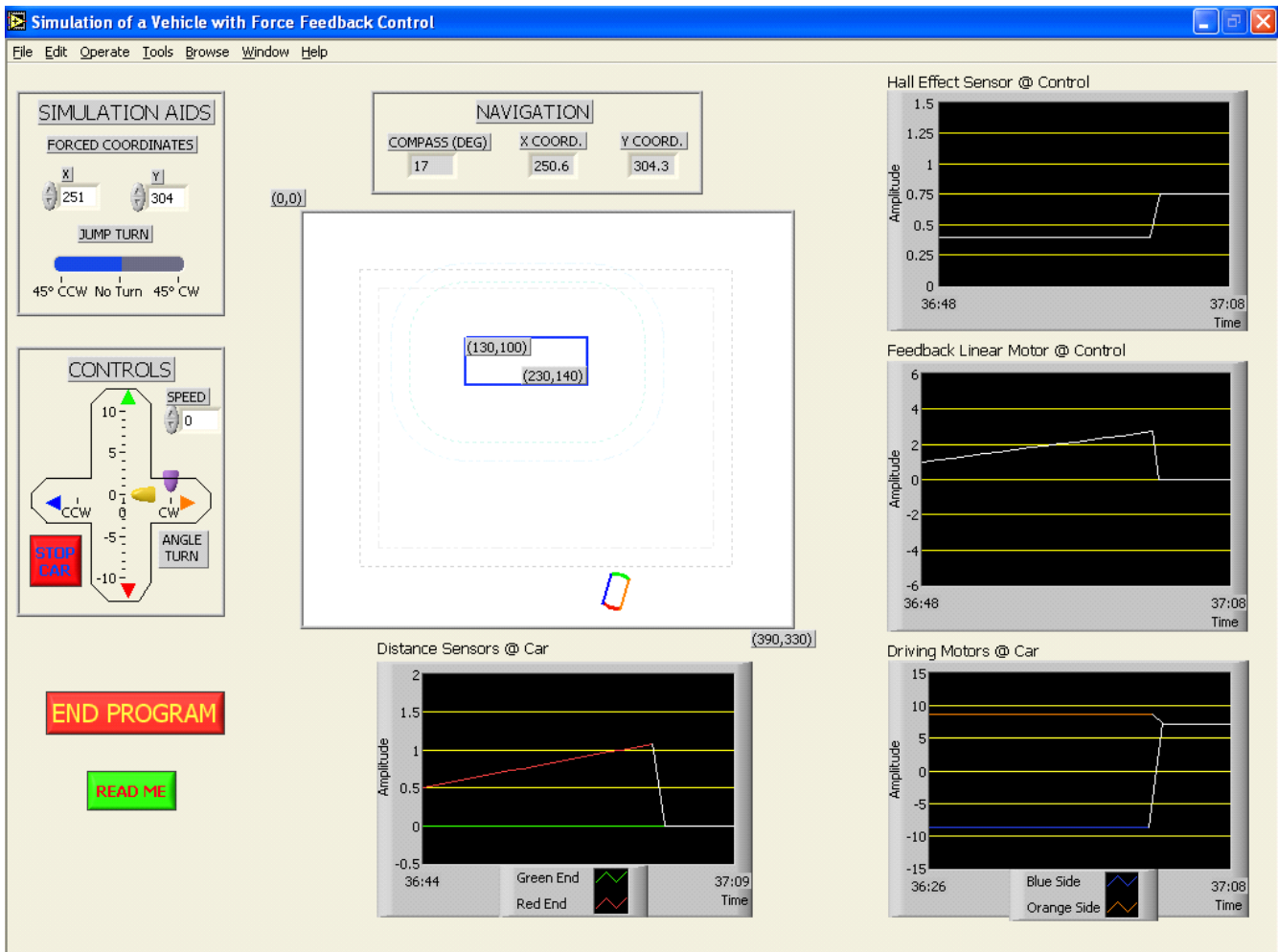


Figure 3. Graphic User Interface of the Simulation Program (displaying a travel at 17° followed by a turn)

the obstacle, in the format (x, y), being x and y the horizontal and vertical distances, respectively, from the origin to the point being identified, measured in “points”, an unit of measure particular to this coordinate system.

Navigation of the vehicle is performed with the cross-shaped control, to set all three characteristics of its velocity. The use of this control is the same as was described before for the real system, with the added function that an accurate value of speed can be written in the small window, or modified with the arrows. This feature is useful in certain situations, for example, setting a very slow speed (even in the order of less than 0.0001) allows to generate force feedback during extended periods of time for its study. The actual speed with which the animation travels, depends on the performance of the computer, so no attempt was made to calibrate this magnitude, and the speed scale (from 0 to 10) is just an indication of relative speed between them (e.g., the only certain fact is that at 6 of speed, the vehicle travels double as fast respect to 3, but those numbers are not directly related to any unit of length per second). Regarding the rotational speed, it was programmed intentionally to be rather slow, facilitating to achieve any specific orientation with a 1° resolution. To compensate for the time-consuming operation involved in achieving specific orientation angle and coordinates for the vehicle by using this control, two “Simulation Aids” were implemented on the simulation, that are not replicated from the capabilities that the actual system would have. The first one, a control bar identified as “Jump Turn” in the front panel, forces the vehicle to rotate instantaneously, in the selected CW or CCW direction, up to the next of the eight preset orientations, which are spaced every 45° starting from 0° . For example, if the compass reading is 274° and the user clicks on the right half of the control bar, indicating that the selected direction of the Jump Turn is CW, the vehicle will make the large turn up to 315° , but if at 274° the user clicks on the left side of the Jump Turn bar, thus selecting the CCW direction, the vehicle will just turn to 270° . In this situation, a second click on the right (CW) end, would bring the vehicle to 360° , while a second click on the left (CCW) end, would make the turn to 225° . The reason to implement this aid in the simulation, was to help the user to achieve large rotations faster, due to the rather slow turning velocity of the vehicle using the “Angle Turn” slider, as was designed to improve the accuracy of the orientation that can be achieved with it. Using the eight preset orientations and then adjusting with the Angle Turn, the maximum slow turning needed to reach any orientation is only 22° . It is not considered that this capability would be implemented in the actual vehicle, because this would require a digital compass to provide a signal with the actual orientation, and a closed loop control to keep the motors running until reaching the commanded orientation angle, thus complicating the system for the very little help that a user, who is using

the vehicle to actually explore a space, would get from this function.

The other simulation aid implemented, which are the numeric controls labeled as “Forced Coordinates” in the front panel, is indeed impossible to realize in the actual vehicle, because it allows the user of the program to displace the vehicle in any direction along only one of either x or y axis orientations, without modifying the other coordinate, which is something that can be done only by dragging manually the vehicle itself. The displacement can be done by incrementing or decrementing, one by one, the coordinates using the little arrows of the numeric controls, or by direct input of a desired coordinate in the corresponding field, and then clicking outside of the window (the latter method does not always work, possibly because it depends of the program block being executed in the instant that the mouse click is done). The purpose of this navigation aid is to help the user to achieve quickly a specific position, after approaching near to this target by running regularly the vehicle. Typically, this feature would be used in front of a corner, to investigate the influence of this singularity over the feedback and distance signals.

Regarding the outputs indicated at the screen, they are of two types. Numerically, the “Navigation” panel shows the x, y coordinates (in “points”) of the common center of the vehicle’s circular arcs, and the orientation angle (in degrees) of its longitudinal center line, measured in CCW direction from the vertical line. When the program is started, the vehicle is at 0° , with the center of the green arc facing up to the obstacle. The graphical outputs are six plots displayed in four virtual oscilloscopes. The plots for the voltages generated at the distance sensors and required at the vehicle motors, are color coded (green and red for the distance sensors, blue and orange for the motors) to match those colors used at the animated diagram, which are also represented in the drawing shown in Fig. 1, to facilitate the study of the behavior of a particular component. When those two oscilloscopes plot a single white line, is because both signals are identical, a situation that happens when neither sensor is detecting a wall (both voltages are zero), and when the vehicle is turning slowly (motors’ voltages are 7.2 V, either positive or negative).

For example, the screen shot of Fig. 3 was taken after approaching the bottom boundary wall at a constant speed, followed by a slow CW turn. The effect of these maneuvers can be seen by the difference in patterns between the first and second parts of the plots in all four oscilloscopes (on all oscilloscopes, the most recent signal is at the right edge and the oldest at the left edge). This figure also demonstrates that while turning the vehicle, the distance sensors are disabled. This simplified the program, because the calculation of the distances is based, among other factors, on the speed direction. Probably, this would be the strategy also on a real application, because the resulting sinusoidal variation would be of little use, being

used mostly for calculation of the force feedback, which is not working either anyway because the speed is zero while turning. This is also the reason why the Hall Effect sensor signal goes to the center of its range, equivalent to the neutral central position of the speed slider. Thus the only oscilloscope active in this situation is the corresponding to the vehicle-driving motors. Also, during the use of both Simulation Aids, there is no oscilloscope active.

To help the user locating the distance and feedback signals, the areas close to both boundaries and obstacle walls, are marked with dashed lines parallel to them, with two different shades of gray (one for boundaries, the other for the obstacle) and patterns (one for the distance sensor, the other for the feedback motor). The line farthest from any wall (at 60 points), identifies the farthest position where a distance sensor starts to detect it, so its voltage signal departs from zero there. The line closest to any wall (at 45 points), identifies the location where the force feedback motor starts working, signaled by a jump in the corresponding oscilloscope. The difference between these two areas, is that the distance sensor closest to the wall

works as long as its edge is inside it, while the feedback motor only works when the vehicle is approaching that wall, but not when going away or moving parallel respect to it, as was explained before. The position of all these guiding lines was determined considering that the approach orientation of the vehicle would be perpendicular to the wall so, as the angle between these two lines increases, the distance from any wall to the position where either transition actually occurs decreases.

To give the user the possibility to experiment with all the possible wall-detecting and force feedback situations, the asymmetric location of the obstacle, serves the purpose of creating three distinct “influence fields”. These are the regions where the distance between a boundary and an obstacle wall is so limited, that the vehicle is under the influence of them while traveling inside these fields for at least a significant portion of the time, causing a signal to be generated at the feedback linear motor and/or the vehicle’s distance sensors. The region under the obstacle, where the starting position of the vehicle is located, which is (195, 240), cannot be considered one of these

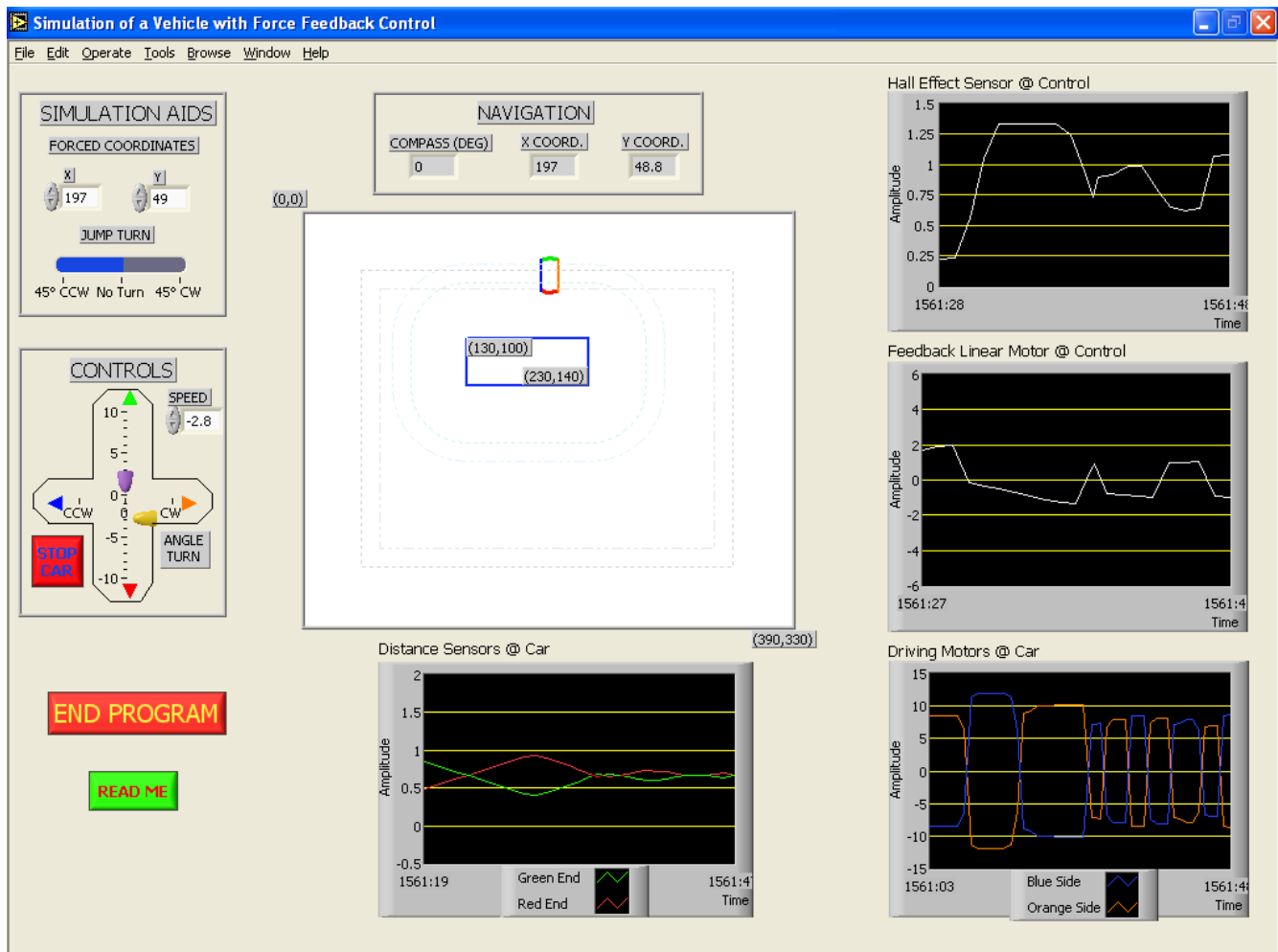


Figure 4. Screen of the program after two different sequences of maneuvers inside the strong influence field

three influence fields, even if feedback and distance signals are generated there, because there is enough free space there, both vertically (the vertical distance from obstacle to wall is $330 - 140 = 190$ units) and horizontally, to allow the user to move the vehicle for a relatively large travel without sensing any wall. However, to the right of the obstacle, a “limited influence field” exists, because there the free horizontal distance is 160 units so, with the length of the vehicle (dimension in the moving direction), which is 30 units, this field leaves only 10 units ($160 - 2 \times 60 - 30$) of space where the vehicle can move with the distance sensors detecting nothing. To the left of the obstacle, the free horizontal distance is only 130 units, thus an “intermediate influence field” is created: if the vehicle is moved there in an horizontal, or close-to-horizontal, direction, the distance sensors will be working all the time. Above the obstacle is where the “strong influence field” is located, because the free vertical distance is just 100 units, so both the distance sensors and the feedback motor will be generating signals continuously when the vehicle is traveling vertically, or close to that line, in this region. By quickly inverting the direction of motion of the vehicle, in the middle of the mild and strong influence fields, signals resembling an inverting square wave (or, an inverting saw tooth wave, if the inverting motion of the vehicle is slower) will be seen in the virtual oscilloscopes assigned to the distance sensors and the feedback motor. This can be seen in Fig. 4, which is a screen shot taken after moving the vehicle at a slow speed while letting it get close to the walls during the first part of the plot, and then inverting rapidly the direction of motion while keeping the vehicle near the center of the strong influence field. The effect of these two situations over the system can be seen in all four oscilloscopes, by how the pattern of the plots changed from the first to the second half of them.

Finally, the screen of the simulation program includes a “Read me” button. By clicking on it, the user will have access to an information screen, where the drawings for the vehicle and the control (Fig. 1 and 2), and a summary of this document, are presented.

DEVELOPMENT OF THE SIMULATION PROGRAM

The selected software to prepare the simulation program is LabVIEW®, due to the ease it offers to create, to link with other program elements, and to display, virtual controls and oscilloscopes of high graphic quality.

The relationships between input and output variables for the simulated system, are much more complicated than for the real system, because the inputs of the latter have also to be calculated in the former, and also because several factors interact to calculate the simulated outputs. Figure 5 shows a comparison between these two systems.

The functions executed by this simulation are:

a) Monitoring of the program screen to detect the selected type of vehicle motion (there are four alternatives, plus the sudden stop) and its parameters, commanded by the

user. Also, when one of these alternatives is selected, the other three are brought to zero, as required by the cross shape of the slot at the actual control.

b) Calculation of either the coordinates or the orientation angle of vehicle required for the next time interval, in order to animate its motion. This requires different methods depending on the user’s selection (respect to movement, by specifying the velocity vs. forced coordinates, respect to rotation, by ordering the slow turning vs. the jump turn).

c) Updating in the screen the current coordinates and orientation angle of the vehicle.

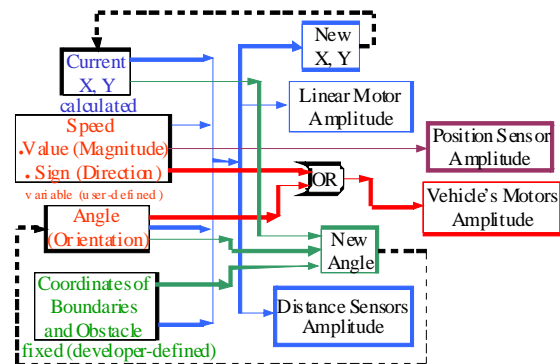


Diagram for the Simulation Program

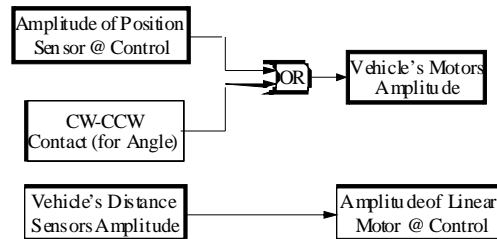


Diagram for an Actual Vehicle

Inputs Outputs

Figure 5. Comparison between Actual and Simulated Systems

d) Calculation of the coordinates of the closest position that the vehicle can be respect to the walls (both of the boundary and the obstacle) depending on the approach angle, so its travel and rotation beyond them are obstructed. This requires a set of eight different calculations, depending on which of the eight walls the vehicle is approaching.

e) Calculation of the distances from both the red and green ends of the vehicle to the boundaries and obstacle walls, measured along its central longitudinal line. To organize this task, the space where the vehicle is moved was divided in eight regions, depending on their positions respect to the obstacle, because each one has its own set of coordinates for the boundaries to be considered, which can be two or four simultaneously. Then, considering together the direction and orientation of movement, there

are eight possibilities for the vehicle to travel along (the positive and negative halves of both coordinate axes, and the four quadrants between them). Finally, there are two ends, the green and the red to calculate, so there are 128 ($8 \times 8 \times 2$) cases. Frequently, these calculations require discriminating with which one of two walls that meet at a right angle the center line is intersecting, and this depends of the orientation angle of the vehicle and its position.

- f) With the distances from function “e”, calculation of the corresponding signals generated by distance sensors.
- g) Using the distances from function “e” above, calculation of the voltage required at the feedback linear motor.
- h) Using the position of the speed control from function “a”, calculation of the voltage required at both motors of the vehicle to achieve the selected travel or rotation.
- i) Using the position of the speed control from function “a” above, calculation of the voltage generated by the Hall position sensor at the control.
- j) Presentation of six curves in four graphical outputs, controlling the changes of colors for four of those.
- k) Management of approximately 45 variables, which implies keeping track of their value for some of them, and reinitializing others after every cycle.

The complete program, including the actual equations that were used for these functions, is available for downloading and running it in LabVIEW, by accessing the National Instruments website [6].

Several assumptions, that do not affect the validity of the program’s logic, were adopted regarding the voltages for all the motors and sensors. The most important supposition is that all their operation curves (working parameter vs. voltage) were simulated as being linear. Typical voltage ranges for all these components were assumed, and uniform conventions for direction of movement of motors according to supply polarity were implemented.

INACCURACIES OF THE SIMULATION PROGRAM

The inaccuracies that have been detected in the animation of the vehicle are as follows.

- a) The behavior of the vehicle, after hitting the obstacle in a non-perpendicular direction and sliding along that wall, is not correct after clearing the corner, because the vehicle keeps sliding for a short length along that line, before continuing the inclined travel.
- b) Due to the simulation of only one distance sensor at each end of the vehicle, and also because its detection range is modeled as a single line instead of the actual narrow triangle, when it approaches a wall at a very acute angle, no signal is generated at these sensors or the feedback motor, because the separation distance sensed is too large.
- c) The middle of the round edges of the vehicle do not get in contact with the corners of the obstacle, when approaching them directly.

CONCLUSIONS

1) The program developed in this project, represents adequately the animated, user-controlled motions of a “vehicle” in the computer screen, together with the real-time plots of the electric signals that would exist at this mobile device and its control, which is equipped with a simulated force feedback capability.

2) The complexity of this program was greatly increased by the inclusion of a rectangular obstacle inside the space where this vehicle is moved.

3) In future projects, the focus will be in solving the inaccuracies found regarding the trajectory of the vehicle after clearing convex corners. Also, the program would allow the user to customize its working parameters during runtime; each end of the vehicle would have three distance sensors instead of one; the real working curves of both position and distance sensors would be implemented, and the detection range for the latter would be modeled as a narrow triangle instead of a single line. Even the position of the gray lines that indicate the threshold positions for the distance sensors and the feedback motors, could be programmed to adapt according to the approaching angle of the vehicle. Finally, other obstacles, of circular and triangular shapes, would also be included to study their detection modeling.

4) A practical application of the system simulated in this work, would be for a safety feature of the accelerator pedal of standard vehicles: distance sensors installed at their front and rear ends, would activate a force feedback preprogrammed to start when those distances to other moving vehicles (for the front sensors) and to stationary objects (for the rear sensors) decrease beyond triggering limits. The difference respect to this project would be that in this application the user would not be able to overcome the force feedback, but rather it would bring the pedal partially or totally up with front or rear detections.

REFERENCES

- [1] Batsomboon, P., Tosunoglu, S., Repperger, D., 2000, “A Survey of Telesensation and Teleoperation Technology with Virtual Reality and Force Reflection Capabilities,” International Journal of Modeling and Simulation, pp. 79–88, Volume 20, No. 1.
- [2] <http://en.wikipedia.org/wiki/>
- [3] National Instruments, 2003, “LabVIEW® User Manual.”
- [4] <http://www.nipponpulse.com>
- [5] <http://www.mcmaster.com>
- [6] Labview site, Nargo, Force Feedback Control. <http://decibel.ni.com/content/community/zone>