



## **Joystick Teleoperation Demo**

Documentation of the joystick teleoperation demonstration model and corresponding ControlDesk front end.

Point of Contact:  
([support@dataspeedinc.com](mailto:support@dataspeedinc.com))

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Simulink Model</b>	<b>3</b>
2.1	Brake Look-up Table . . . . .	3
2.2	Steering Control . . . . .	4
2.3	Shifter Control . . . . .	5
2.4	Turn Signal Control . . . . .	6
2.5	Vehicle Data . . . . .	6
2.6	System Enable Logic . . . . .	7
<b>3</b>	<b>ControlDesk Front End</b>	<b>8</b>
3.1	Main . . . . .	8
3.1.1	Brake . . . . .	8
3.1.2	Throttle . . . . .	9
3.1.3	Steering . . . . .	9
3.1.4	Shifter . . . . .	9
3.1.5	Faults . . . . .	9
3.1.6	System Enabled . . . . .	9
3.2	Vehicle Data . . . . .	9
3.3	Joystick Instrument . . . . .	10

---

## DISCLAIMER:

This application is intended for demonstration purposes only. In no event shall Dataspeed Inc. be liable for any direct, indirect, punitive, incidental, special consequential damages, to property or life, whatsoever arising out of or connected with the use or misuse of its applications.

## 1 Introduction

This document describes the demo model **joystick\_teleop.mdl** and its corresponding ControlDesk project front end. The purpose of this demo model is to provide an example application-level Simulink model and ControlDesk front end that utilizes the Dataspeed drive-by-wire dSPACE interface blocks ([library\\_blocks\\_doc.pdf](#)). The main functions of this demo are:

- The joystick signals are inputted to ControlDesk over USB using the built-in 'Steering Controller' instrument.
- The joystick signals are sent to the MicroAutoBox that is running the **joystick\_teleop.mdl** Simulink model.
- The Simulink model parses the joystick signals and uses the provided Dataspeed drive-by-wire interface library blocks to communicate with the drive-by-wire hardware modules over CAN.
- Vehicle CAN network data that is re-transmitted by the drive-by-wire hardware modules are received by the Simulink model using the provided Dataspeed vehicle data library blocks.
- The received vehicle data is sent to ControlDesk for visualization.
- The system is enabled and disabled using buttons on the steering wheel, and is also disabled if the driver intervenes by taking control of the steering, pedals, or shifter. This behavior is implemented using the **System\_Enable\_Logic** block provided in the Dataspeed block library.

The joystick used to control the vehicle is shown in Figure 1. The demo uses the labeled buttons and axes, with each of their functions described in Table 1.

Table 1: Functions of each joystick button and axis.

Button/Axis	Function
Left Control Stick	The left/right axis is used to control steering
Right Control Stick	The left/right axis is used to control steering
Back	Hold down to amplify the range of the steering joysticks for low-speed maneuvering
Start	Hold down to amplify the range of the steering joysticks for low-speed maneuvering
Y	Shifts into park
B	Shifts into reverse
X	Shifts into neutral
A	Shifts into drive
D-Pad Left	Toggle left turn signal
D-Pad Right	Toggle right turn signal
Left Trigger	Controls the brake
Right Trigger	Controls the throttle

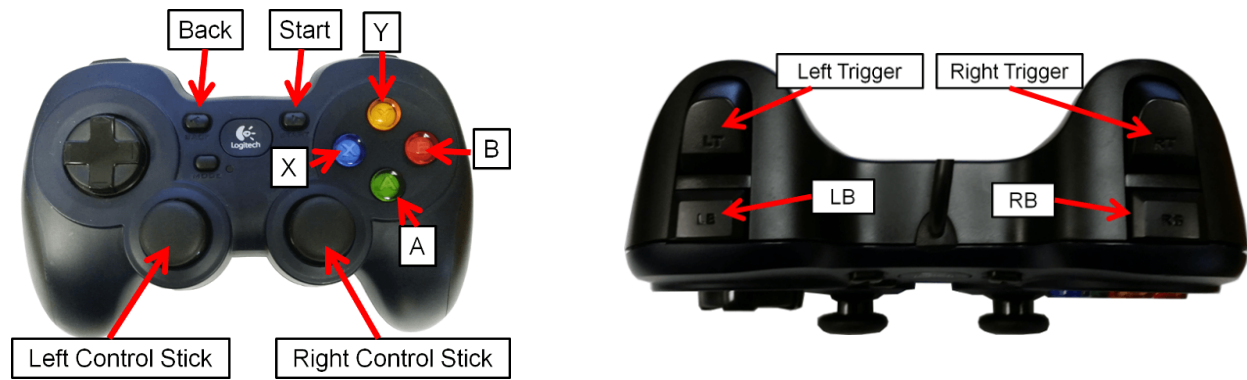


Figure 1: Video game joystick used to control the drive-by-wire modules.

## 2 Simulink Model

In the top level block of **joystick\_teleop.mdl**, the dSPACE RTICAN configuration block, along with the **Throttle\_Interface**, **Brake\_Interface**, **Steering\_Interface**, **Shifter\_Interface**, **Turn\_Signal\_Interface** and **System\_Enable\_Logic** blocks are links to the corresponding Dataspeed library blocks documented in [library\\_blocks.doc.pdf](#). This section outlines the rest of the blocks in the demo model that are specific to the joystick teleoperation functionality.

### 2.1 Brake Look-up Table

The actual brake pedal on the vehicle does not apply braking in a linear fashion. Early stages of brake pedal travel yield a slower increase in braking output, and at a certain pedal position, the rate of increase becomes greater. This behavior is simulated for the joystick by implementing a piecewise linear look-up table with a single break point, as shown in Figure 2.

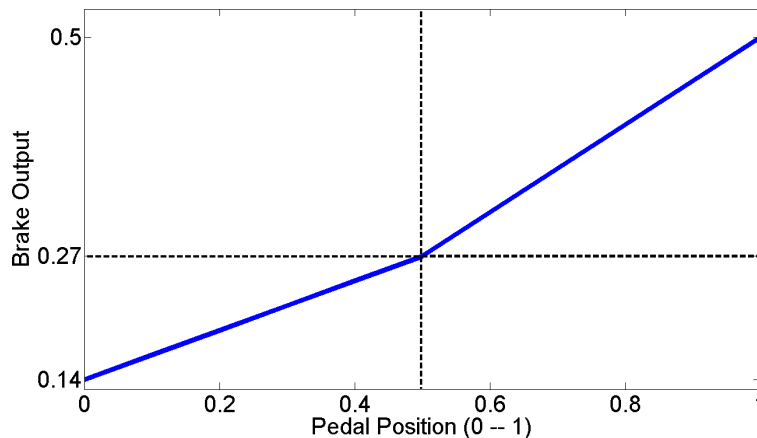


Figure 2: Graph of the lookup-table applied to the brake command from the USB joystick.

The break point in the look-up table is configurable using the **input\_LUT** and **output\_LUT** inputs, where **input\_LUT** is the joystick input corresponding to the break point, and **output\_LUT** is the braking output corresponding to the break point. Figure 2 shows the default values of **input\_LUT** = 0.5 and **output\_LUT** = 0.27.

The provided ControlDesk project can alter the look-up table break point values at runtime (see Section 3.3). The implementation of the brake look-up table is shown in Figure 3, and the I/O ports are tabulated in Table 2.

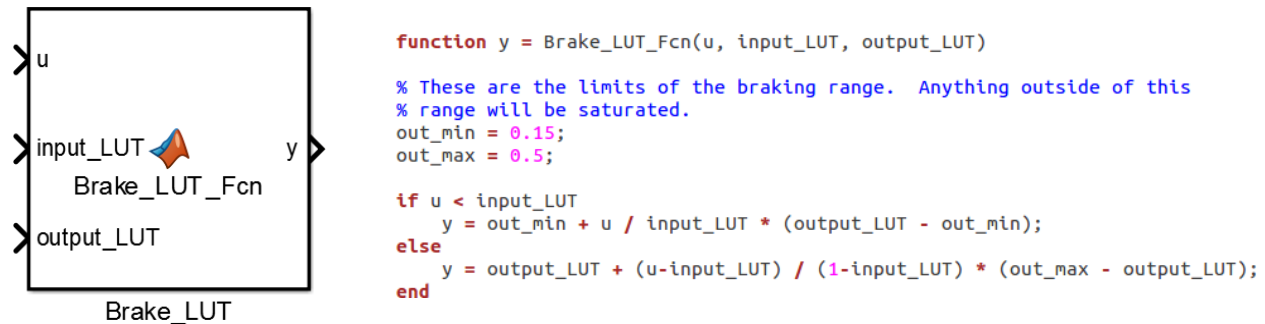


Figure 3: Embedded MATLAB function implementing the brake pedal look-up table.

Table 2: Port I/O of the brake look-up table block.

	Port Name	Data Type	Range	Description
<b>Inputs</b>	u	double	0 — 1	Pedal position input from joystick
	input_LUT	double	0 — 1	Joystick input corresponding to break point
	output_LUT	double	0.15 — 0.5	Brake output corresponding to break point
<b>Outputs</b>	y	double	0.15 — 0.5	Brake output from look-up table

## 2.2 Steering Control

When driving the vehicle with the USB joystick at very low speeds, it is desirable to have control over the full range of the steering wheel angle to make necessary maneuvers. However, when traveling at low to medium speed, it is better to have finer control of the steering to avoid oscillations. The **Steering\_Control** block, shown in Figure 4, implements the following processing of the raw USB joystick signals:

- The maximum input being received from either the left or right control sticks is passed through as the command. This is so the user can use whichever control stick is preferred.
- To change the gain on the steering signal, the user can press either the **Back** or **Start** buttons.
- Without pressing any buttons, the control stick covers half of the steering range ( $-235^\circ \Rightarrow 235^\circ$ ).
- When either **Back** or **Start** is pressed, the control stick covers the full range ( $-470^\circ \Rightarrow 470^\circ$ ).

The I/O of the block is described in Table 3.

Table 3: Port I/O of the steering control block.

	Port Name	Data Type	Range	Description
<b>Outputs</b>	Steering_Command	double	-470 — 470	Desired steering wheel angle in degrees

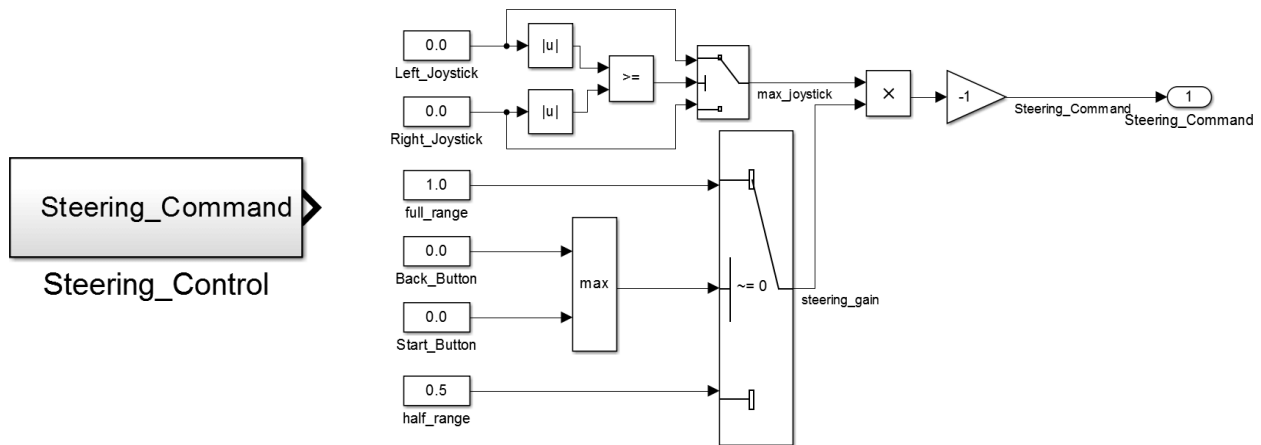


Figure 4: Steering control block that processes joystick signals and outputs a steering command.

## 2.3 Shifter Control

Gear shift commands are input using either the USB joystick, or the gear shift instrument in ControlDesk (see Section 3.3). The four buttons on the right front of the joystick are used to change gears, where the particular mappings are described in Table 1. The **Shifter\_Control** block, shown in Figure 5, implements the following behavior:

- If a gear selection is made using the ControlDesk gear shift instrument, the selected value is routed to the **Gear\_Command** output.
- If the user presses any of the shifting buttons listed in Table 1, then that particular gear is routed to the output.
- If no change is detected from the ControlDesk instrument, and no shifting buttons are pressed, then a value of '0' is routed to the output.

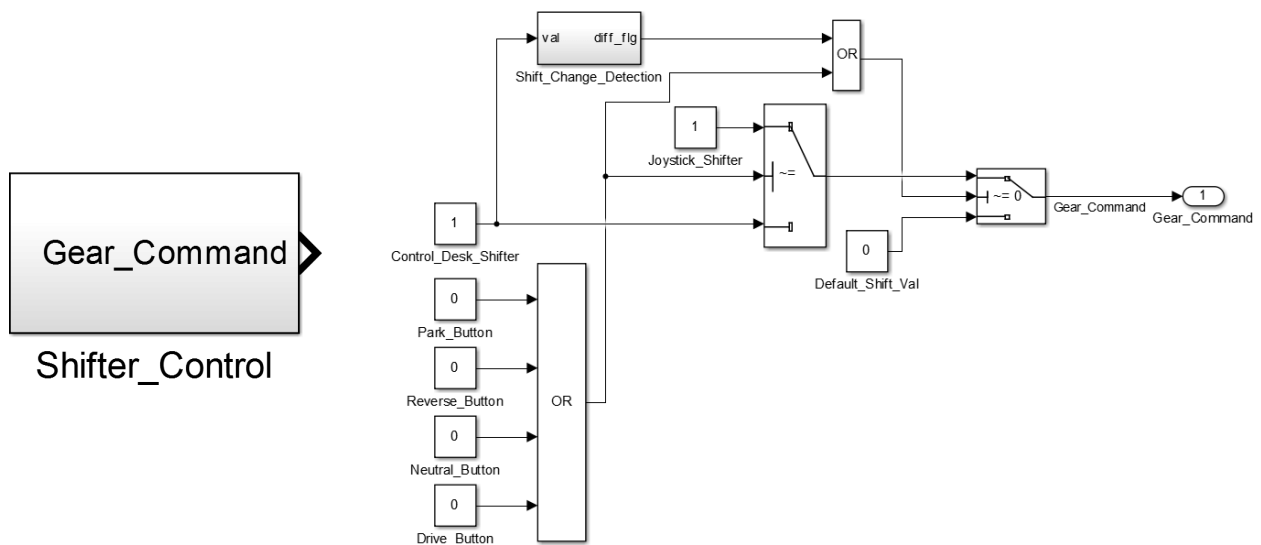


Figure 5: Shifter control block that combines a ControlDesk gear shift instrument with the USB joystick to output a gear shift command.

## 2.4 Turn Signal Control

The turn signals can be controlled with the left and right d-pad buttons on the USB joystick, as indicated in Table 1. The control behavior is:

- If no turn signal is active, left and right presses will activate the appropriate turn signal.
- Pressing the same direction as the active turn signal will turn off the signal.
- Pressing the opposite direction as the active turn signal will immediately switch the active turn signal.

## 2.5 Vehicle Data

The **Vehicle\_Data** block, shown in Figure 6, contains the library blocks that receive the sensor data that is available on the drive-by-wire bus. The I/O of this block is described in Table 4.

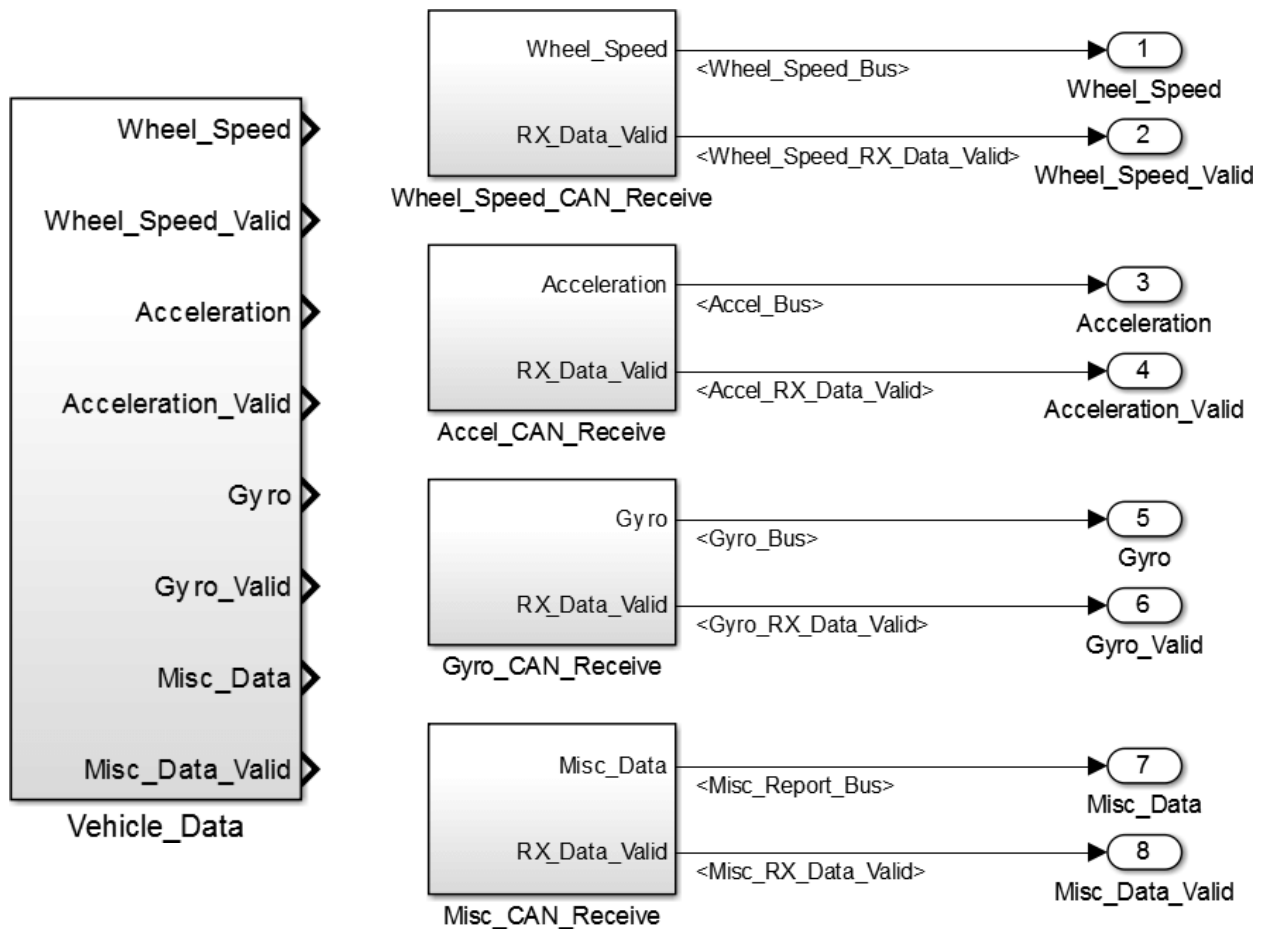


Figure 6: Simulink subsystem that contains the individual vehicle data library blocks.

Table 4: Port I/O of the vehicle data block.

	Port Name	Data Type	Range	Description
<b>Outputs</b>	Wheel_Speed	bus	–	Bus containing speed data from each of the four wheels
	Wheel_Speed_Valid	bool	–	Flag indicating that the wheel speed report message is being received
	Acceleration	bus	–	Bus containing longitudinal, lateral and vertical acceleration
	Acceleration_Valid	bool	–	Flag indicating that the acceleration report message is being received
	Gyro	bus	–	Bus containing roll and yaw rates
	Gyro_Valid	bool	–	Flag indicating that the gyro report message is being received
	Misc_Data	bus	–	Bus containing the data in the Miscellaneous Report message (ID = 0x69). See the <a href="#">Steering-Shifter datasheet.pdf</a>
	Misc_Valid	bool	–	Flag indicating that the misc data report message is being received

## 2.6 System Enable Logic

The system is enabled and disabled using the steering wheel buttons and the current status of the driver overrides. This is done by using the **System\_Enable\_Logic** library block, and connecting its inputs and outputs in an appropriate manner. A screenshot of how the **System\_Enable\_Logic** block is used in this model is shown in Figure 7.

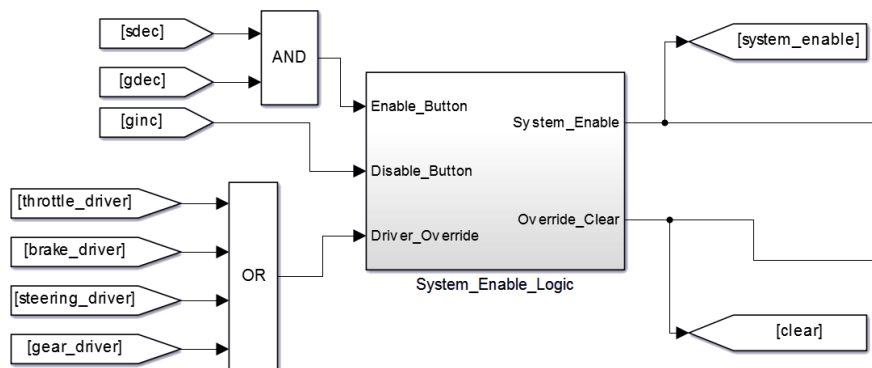


Figure 7: Use of the system enable logic block in the joystick teleoperation demo model.

In order to enable the system, the driver has to press the ACC decrement speed button and the ACC increment gap distance button simultaneously. The signals from these two buttons are therefore logically ANDed and sent to the **Enable\_Button** input. To disable the system, the driver can press the ACC increment gap distance button.

The system is also disabled if any driver override condition is present. Therefore, all the individual **DRIVER** signals from each drive-by-wire module are logically ORed together and sent to the **Driver\_Override** input.



## 3 ControlDesk Front End

A ControlDesk project called **joystick\_teleop\_controldesk** accompanies the Simulink model to operate the vehicle with a joystick. This section describes the different layout tabs present in the project and what each instrument reports and does.

### 3.1 Main

This layout contains the displays and controls required to operate the drive-by-wire interface. The different groups of instruments are:

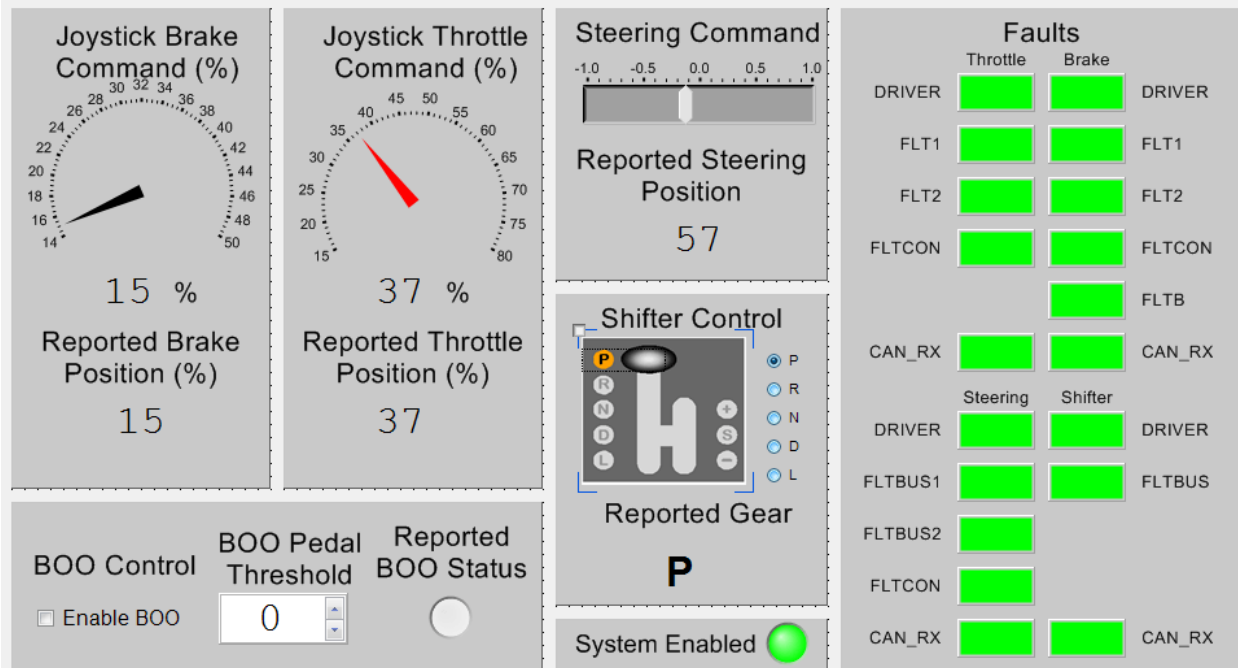


Figure 8: Main ControlDesk layout that displays signals relevant to the drive-by-wire functionality.

#### 3.1.1 Brake

- **Joystick Brake Command** – The gauge and display instruments both indicate the current joystick command for the braking signal being sent to the drive-by-wire hardware.
- **Reported Brake Position** – The current brake pedal position being reported by drive-by-wire hardware.
- **BOO Control** – Check box allowing the user to override the BOO signal and force it high.
- **BOO Pedal Threshold** – Numeric input to adjust the brake pedal position threshold past which the BOO signal is automatically set high. Refer to the library block documentation for the implemented behavior of this threshold parameter ([library\\_blocks\\_doc.pdf](#)).
- **Reported BOO Status** – Red light will be off when the actual BOO status being reported by drive-by-wire hardware is low, and on when it is high.

---

### 3.1.2 Throttle

- **Joystick Throttle Command** – The gauge and display instruments both indicate the current joystick command for the throttle signal being sent to the drive-by-wire hardware.
- **Reported Throttle Position** – The current throttle pedal position being reported by drive-by-wire hardware.

### 3.1.3 Steering

- **Steering Command** – Slider bar indicating the relative steering command, where the left and right sides represent the maximum amount of steering wheel angle input in that particular direction.
- **Reported Steering Position** – Current steering wheel angle as reported by drive-by-wire hardware.

### 3.1.4 Shifter

- **Shifter Control** – The graphical gear shifter instrument can be used to shift gears by clicking on it. The radio buttons report the last gear that was requested from the joystick.
- **Reported Gear** – The current gear being reported by drive-by-wire hardware.

### 3.1.5 Faults

The faults are an assortment of LEDs to visually determine system state and data validity.

- **DRIVER** – Will be green when the computer is controlling the specific drive-by-wire system. Will turn yellow when the driver intervenes. To return to computer control, the drive-by-wire systems must be disabled and then enabled again. This can be done with the enable check boxes on the main layout tab, or by pressing the **LB** button on the USB joystick to disable, followed by pressing the **RB** button to enable again.
- **CAN\_RX** – Will be green when receiving the appropriate subsystem's report CAN message. Will be red if no messages are being received.
- **Other Fault LEDs** – Will be green when no fault exists, and will be red if a fault does exist. Refer to the individual product datasheets for more details regarding these faults ([throttle\\_brake\\_datasheet.pdf](#), [steering\\_shifter\\_datasheet.pdf](#))

### 3.1.6 System Enabled

Indicator of the status of the **System\_Enable** output from the system enable logic block. The indicator LED will be off when system is disabled, and green when it is enabled.

## 3.2 Vehicle Data

This layout displays the vehicle data being received over CAN.

- **Wheel Speeds** – The gauges and displays show the individual wheel speeds in rad/s.
- **Acceleration** – The slider bars and displays show the longitudinal, lateral and vertical acceleration in  $\text{m/s}^2$ .

- **Roll** – The graphic shows a horizontal line under zero roll rate, and rotates in response to a non-zero roll rate. The display indicates the roll rate in rad/s.
- **Yaw** – The gauge and display show the yaw rate in rad/s.
- **Turn Signals** – The currently-requested turn signal command is shown.
- **CAN\_RX** – Will be green if the corresponding vehicle data message is being received, and will be red otherwise.

### 3.3 Joystick Instrument

This layout contains the steering controller instrument that receives the inputs from the USB joystick, and also provides numerical inputs for adjusting the brake pedal look-up table discussed in Section 2.1.

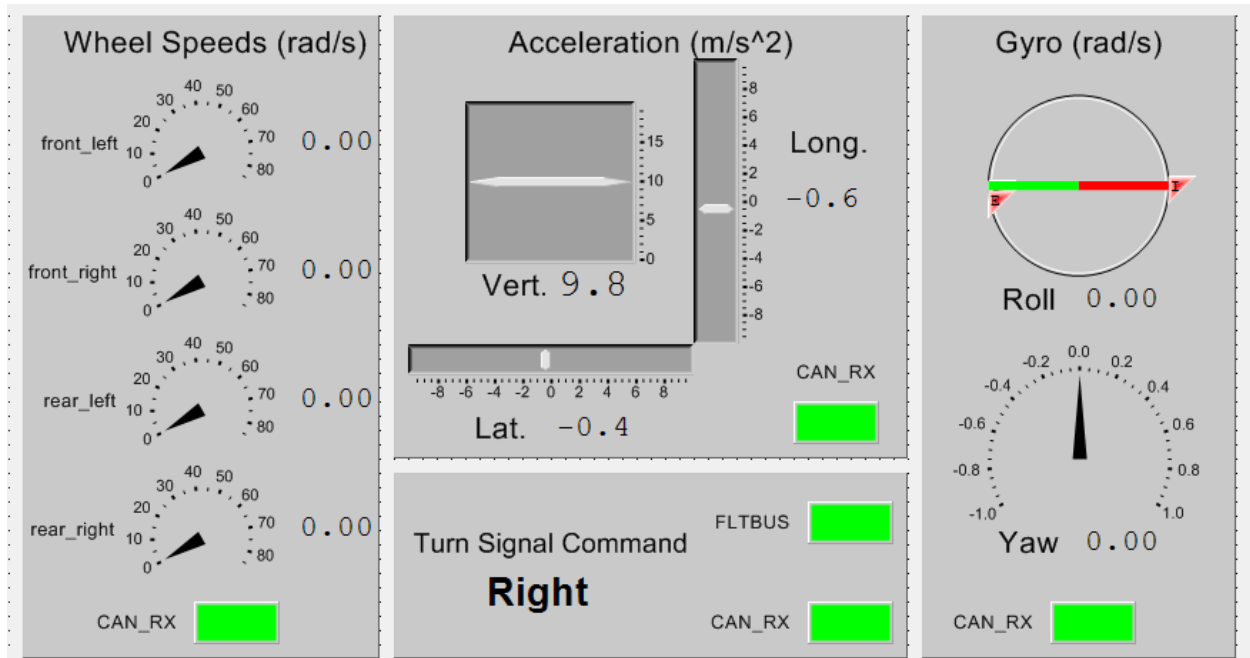


Figure 9: ControlDesk layout that displays received vehicle data.

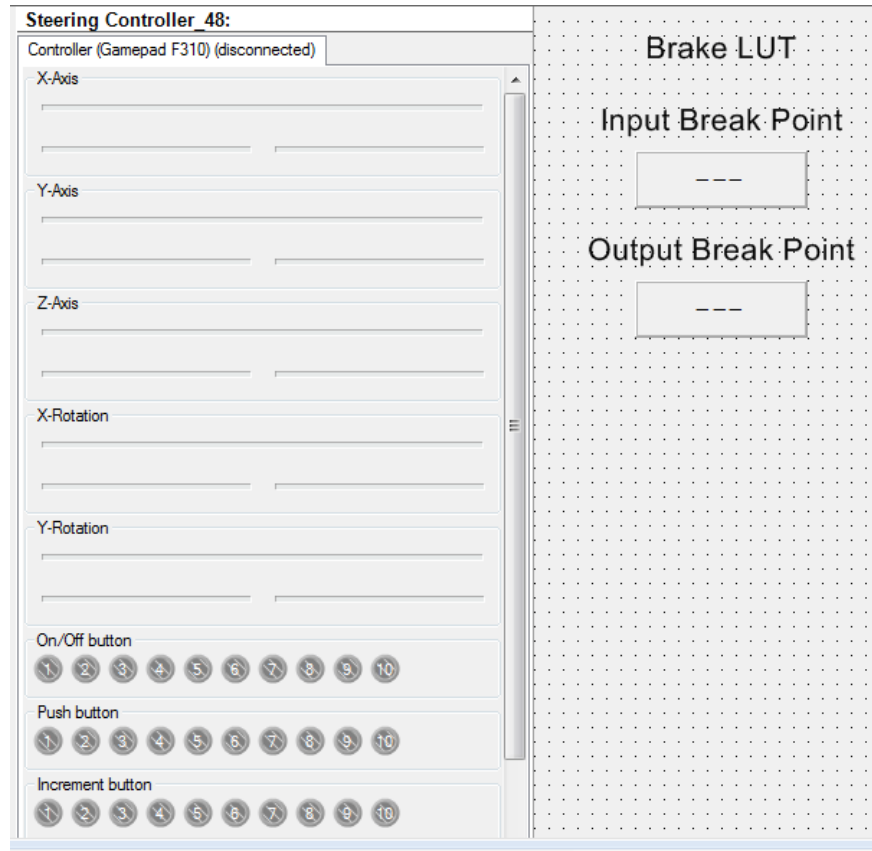


Figure 10: ControlDesk layout that contains the USB joystick instrument and adjusts the brake pedal look-up table.