Universal Lat/Lon Controller
User's Guide

Point of Contact:
support@dataspeedinc.com

Contents

1 Overview

The Dataspeed ADAS Kit provides a robust and easy-to-use interface to brake, throttle, steering and shifting actuators. However, controlling these actuators usefully in a self-driving vehicle can be challenging. Additionally, developing such systems can consume large amounts of time, without adding significant value to higher-level projects. The Dataspeed Universal Lat/Lon Controller (ULC) system is designed to alleviate this issue, thereby making it even easier to quickly develop autonomous vehicle concepts.

The ULC is designed to provide equivalent interface and control behavior across all Dataspeed drive-by-wire platforms. Therefore, engineers using a particular vehicle platform can easily port existing software to another platform. The vehicle platforms and corresponding firmware versions that are available with the ULC are shown in Table 1.

Table 1: Supported Drive-by-Wire Platforms and Firmware Versions

Vehicles	Platform	Firmware Version
Lincoln MKZ / Ford Fusion / Ford Mondeo	FORD CD4	2.1.0 and later
Ford F150	FORD P5	Not supported
Chrysler Pacifica	FCA RU	1.0.0 and later
Jeep Grand Cherokee	FCA WK2	Not supported

1.1 Features

- **Embedded in ADAS Kit Firmware** – The system is fully integrated with the existing Steering-Shifter control module firmware and is configured by CAN messages. This allows any computing hardware already controlling the ADAS kit actuators to make use of the higher level of control offered by the ULC, without restricting the user to a particular software development environment.
- **Full-Range Speed Control** – The control of the throttle and brake pedals is designed to maintain a full range of speeds, from 0.45 m/s (1 MPH) to 45 m/s (100 MPH).
- **Automatic Shifting** – If the particular vehicle is capable of drive-by-wire shifting, the ULC will automatically shift into the appropriate gear to reach the specified speed. When a negative speed command is received, the vehicle comes to a stop, shifts into Reverse, and goes to the negative target speed. Likewise, the system stops and shifts into Drive when a positive speed command is received. The ULC can also start in Park and shift into Drive or Reverse depending on the sign of the speed command.
- **Natural Set Speed Transitioning** – If the speed command changes sharply, such as when changing a set speed, the system internally generates and tracks a smooth speed profile that feels natural to human passengers. The amount of acceleration used to transition to the target speed is configurable by the user.
- **Arbitrary Speed Trajectory Tracking** – In addition to set speed transitions, the ULC can track a gradually-changing speed command signal with minimal latency. This can be helpful for implementing a time-sensitive trajectory generated by a path planning algorithm.
- **Kinematic Yaw Rate and Curvature Control** – Using a kinematic model of the vehicle, steering wheel angle commands are generated from either a yaw rate or curvature input signal.
- **Configurable Dynamics Limits** – The aggressiveness of the speed and steering control can be configured by specifying maximum linear and lateral acceleration limits.

- **Independent Subsystem Disabling** – Using dedicated bits in the command CAN message, speed control can be disabled while leaving steering control active, and vice versa. Likewise, automatic shifting can be disabled completely, or merely restricted from shifting out of Park.
- **Versatile Integration with Other Controllers** – Because the speed, steering, and shifting components of the ULC can be independently switched on and off, integration with other systems is possible. For example, when speed control is disabled, the ULC does not send any commands to the throttle or brakes, leaving them controllable by another system. Likewise, steering can be controlled independently while the ULC is regulating speed and acceleration with the throttle and brake pedals.

1.2 Use Cases

This section describes some example scenarios in which the ULC could be used.

- **Testing a Lane Centering System** – In this scenario, the user is interested in testing a system that generates steering wheel commands to keep the vehicle in the center of a lane, but wants the vehicle to maintain a constant speed. To configure the ULC to support this, the user would enable the speed control subsystem of the ULC and send the desired speed, while disabling the steering control subsystem such that the output of the system under test has full control of the steering wheel.
- **Implementing Kinematics-Based Path Following** – By enabling the steering control subsystem of the ULC, the user can directly request yaw rate or curvature, and the system generates the appropriate steering wheel angle command based on the kinematic properties of the vehicle. Using this, path following controllers based on kinematics models that output yaw rate or turning radius can be easily integrated with the drive-by-wire system.
- **Testing an Emergency Braking System** – In this scenario, the speed control subsystem of the ULC could be used to set up repeatable initial speed conditions for an emergency braking test by sending a constant target speed. Meanwhile, the emergency braking system could monitor its criteria for triggering a braking event, and once triggered, start sending brake actuator commands to the drive-by-wire system.

  Because of the preemption capability of the ULC, when the brake actuator commands start being received, the speed control subsystem of the ULC would be preempted and complete control of the brakes would be given to the emergency braking system. Further, by disabling the steering control subsystem of the ULC, the human test driver would be able to steer manually without disrupting the test.

2 Quick Start Guide

This section describes how to run an out-of-the-box demonstration of the speed control capability of the ULC using ROS.

2.1 Downloading or Installing the ROS Interface

Dataspeed provides an open-source ROS interface for the ULC similar to the ROS interfaces for the Ford and FCA ADAS kits. The repository for the ROS interface can be found at:

https://bitbucket.org/dataspeedinc/dataspeed_ulc_ros

Binaries of the packages in this repository can also be installed using apt:

```
sudo apt install ros-$ROS__DISTRO-dataspeed-ulc
```

2.2 Configuring the Demonstration

There are two Python ROS nodes in the dataspeed_ulc_can package that demonstrate the speed control capability of the ULC:

- speed_square_wave.py
- speed_sine_wave.py

Both of the speed control demonstration nodes can be configured with the parameters outlined in Table 2. The speed_square_wave.py node generates a square wave speed command that changes between ‘v1’ and ‘v2’ with a total period of ‘period’. The speed_sine_wave.py node starts by reaching the speed specified in ‘v1’, then waits for 3 seconds before executing a sine wave that oscillates between ‘v1’ and ‘v2’ at the specified period.

Table 2: Speed Control Demonstration Node Parameters

ROS Parameter	Description	Units
v1	Velocity 1	m/s
v2	Velocity 2	m/s
period	Waveform period	s
accel_limit	Maximum allowed acceleration	m/s^2
decel_limit	Maximum allowed deceleration	m/s^2
enable_shifting	Allow ULC to shift gears	boolean

2.3 Running the Demonstration Programs

To execute the demonstration, follow this procedure:

1. Power on the drive-by-wire hardware and connect the Dataspeed USB / CAN converter tool to the computer.
2. Launch the ROS drive-by-wire interface for the appropriate platform:

   roslaunch dbw_mkz_can dbw.launch
   - or -
   roslaunch dbw_fca_can dbw.launch
3. Run either speed_square_wave.py or speed_sine_wave.py with the parameters in Table 2 set to the desired values.
4. Enable drive-by-wire control by either pressing the steering wheel buttons or publishing to the /vehicle/enable topic, which is of type std_msgs/Empty.
5. Observe the /vehicle/ulc_cmd topic to see the ROS message sent from the demonstration node to the ROS interface node.
6. Observe or record the /vehicle/ulc_report topic to see feedback from the ULC.

A node graph of the demonstration system is shown in Figure 1.

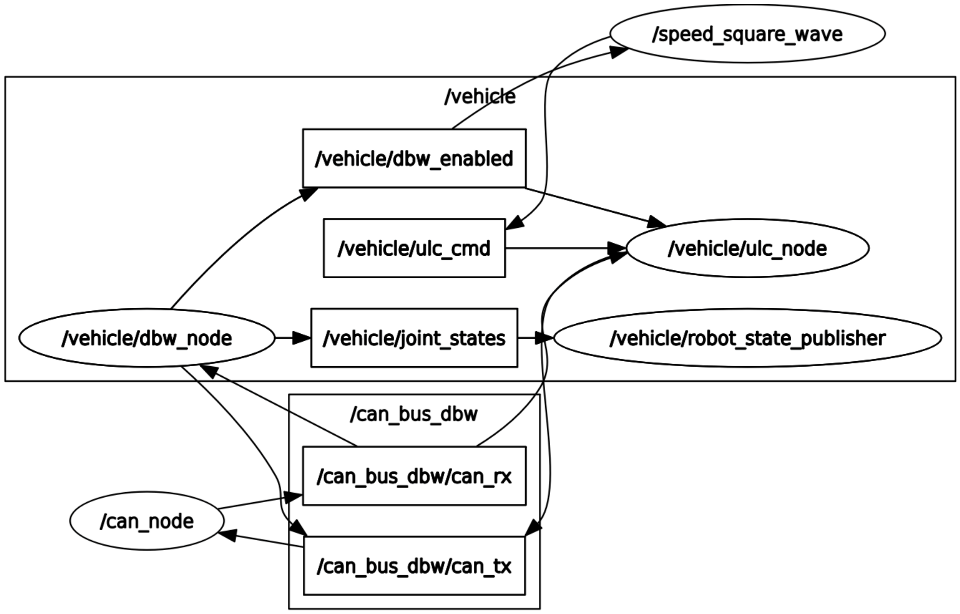


Figure 1: Runtime node graph of the speed control demonstration

3 Interfacing with the Universal Lat/Lon Controller

The user controls the ULC by transmitting a command and configuration CAN message on the Dataspeed CAN bus (CAN IDs 0x076 and 0x77, respectively). Information about the ULC is present in the ULC report message with CAN ID 0x78. This section describes how to populate the command and configuration messages to enable and configure the system, as well as which criteria must be satisfied before the ULC will activate. See Tables 4 and 5 in Section 6 for the complete structure of the CAN messages.

3.1 Control Activation Conditions

The following conditions must be satisfied before the ULC will send actuator commands to the drive-by-wire system:

- The most recent command CAN message must be received within a 100 ms timeout. However, the underlying control system runs at a sample time of 20 ms, so **it is recommended to send the command messages every 20 ms as well**.
- All driver overrides on any actuator of the overall drive-by-wire system must be cleared.
- To enable throttle and brake commands to regulate speed, the SPEED bit of the command CAN message must be set.
- To enable steering control, the STEER bit of the command CAN message must be set.

3.2 Control Preemption

The speed and steering subsystems of the ULC are automatically preempted if external actuator commands are being received.

- If another system is sending throttle or brake commands while speed control is active, the ULC will disable its speed control system as if the command CAN message's SPEED bit were cleared.
- If another system is sending steering actuator commands while steering control is enabled, the ULC will disable its steering control system as if the STEER bit were cleared.
- Speed control is automatically re-enabled when the stream of external throttle or brake commands stops for longer than 70 ms, provided the SPEED bit is set in the command message.
- Steering control is automatically re-enabled when the stream of external steering commands stops for longer than 70 ms, provided the STEER bit is set in the command message.
- The PRE_SP and PRE_ST bits in the ULC report message indicate whether or not the speed and/or steering control subsystems are being preempted. These bits are set when the particular ULC subsystem would otherwise be active if it weren't for the preemption condition.
- **Important:** The ULC is preempted by external actuator commands even if the enable bits of those actuator commands are cleared. The only way to avoid preempting the ULC is to stop sending the actuator commands entirely.

3.3 Clearing Driver Overrides

During regular ADAS Kit operation, clearing a driver override flag requires setting the CLEAR bit of the command message corresponding to the particular vehicle actuator. When using the ULC however, it is anticipated that not all users' software will be equipped to send command messages to the individual drive-by-wire subsystems, instead opting to just send command messages to the ULC.

Therefore, the ULC command message also includes a CLEAR bit. Sending a ULC command CAN message with the CLEAR bit set automatically requests a driver override clear on each of the throttle, brake, and steering subsystems. This means that whenever the driver intervenes with either the brake pedal, throttle pedal, or steering wheel, the ULC can be re-enabled by setting the CLEAR bit of the command.

3.4 Configuration CAN Message

The ULC configuration CAN message is used to specify maximum values for the linear acceleration and deceleration implemented by the speed control component, as well as maximum values for the lateral and angular acceleration implemented by the steering control component. To set these acceleration limits using the configuration message:

- The most recent configuration CAN message must be received within a 1 second timeout threshold. After timeout, the acceleration limits revert to their built-in default values. If it is desired to use non-default acceleration limits, it is recommended to send this message every 200 ms.
- Setting individual acceleration limit fields to zero instructs the ULC to use the built-in default values. Nonzero values override the defaults and are saturated between their minimum and maximum values.

The range and default value of each of the acceleration limits, along with their corresponding CAN signal names are shown in Table 3. The implications of different acceleration limits on speed tracking behavior is discussed in Section 4.2. The way lateral and angular acceleration limits are used to constrain steering angle and rate is discussed in Sections 5.3 and 5.4. The structure of the CAN configuration message where these limits are set is shown in Section 6.2.

Table 3: Range and Default Values of Acceleration Limits

Limit Type	CAN Name	Min Value	Max Value	Built-in Default
Linear Acceleration	LIN_ACCEL	0.3 m/s^2	3.0 m/s^2	Lookup table (See Section 4.2)
Linear Deceleration	LIN_DECEL	0.3 m/s^2	6.0 m/s^2	1.5 m/s^2
Lateral Acceleration	LAT_ACCEL	1.0 m/s^2	12.75 m/s^2	4.0 m/s^2
Angular Acceleration	ANG_ACCEL	0.5 rad/s^2	5.1 rad/s^2	1.0 rad/s^2

4 Regulating Speed and Acceleration

4.1 Input Tracking Modes

The ULC speed control system inputs a raw commanded speed received in the LIN_VEL signal of the command CAN message, and generates a target reference speed and acceleration referred to as the “Internal Reference Speed” and “Internal Reference Acceleration”. These internal references are generated by enforcing acceleration and jerk limitations on the raw commanded speed to produce a smooth speed profile that is comfortable for passengers.

The internal reference speed and acceleration can be generated in two modes: Loose Tracking Mode and Tight Tracking Mode. These two modes use fundamentally different methods to compute the internal references, and the mode that is active at any given time is dependent on the state of the system. Regardless of the active input tracking mode, the speed and acceleration references being generated are the inputs that propagate downstream for the controllers to regulate.

Loose Tracking Mode

While Loose Tracking Mode is active, the system generates the internal reference speed and acceleration signals to transition to the current command in a comfortable fashion, as shown in Figure 2. This mode is designed to close large gaps between the commanded speed and current internal reference speed.

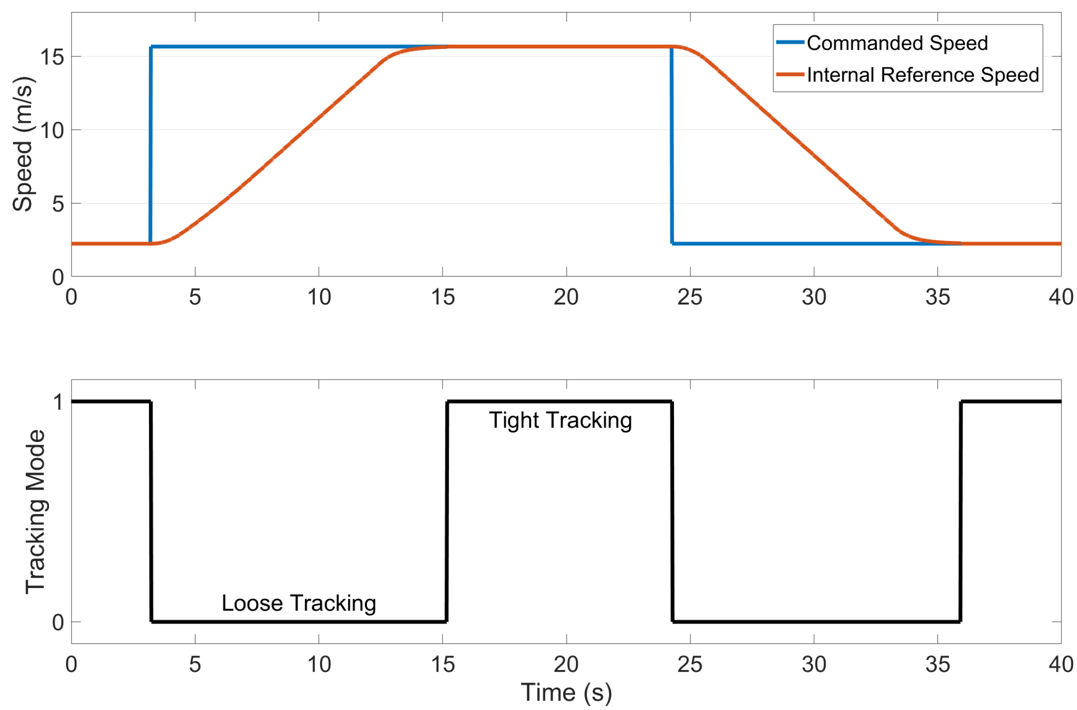


Figure 2: System transitions to Loose Tracking Mode to close large differences smoothly.

Tight Tracking Mode

While Tight Tracking Mode is active, the system analyzes the value and rate of change of the commanded speed to generate the internal reference speed and acceleration. Due to the nature of how the commanded speed is processed to estimate its rate of change, the internal reference speed is delayed from the user’s input by 250 ms, as shown in Figure 3. This mode is designed to closely follow a gradually changing speed command trajectory.

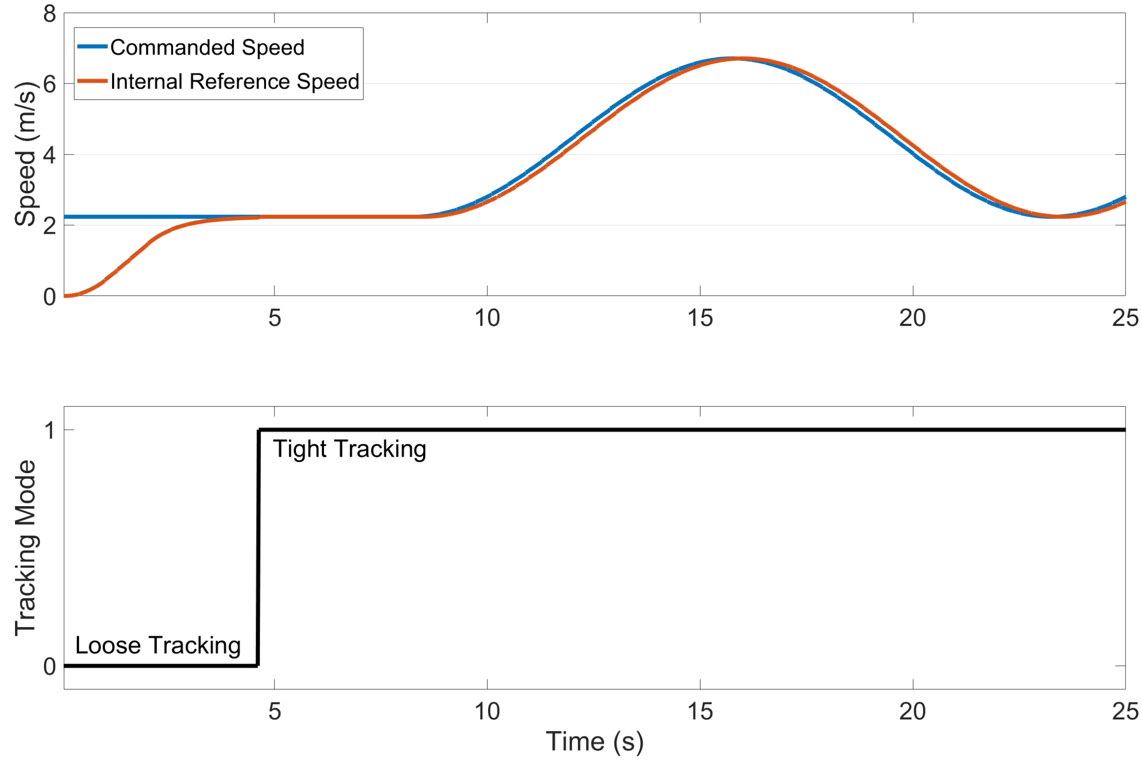


Figure 3: System stays in Tight Tracking Mode while tracking a gradually changing signal.

Transitioning Between Input Tracking Modes

The system automatically switches between Loose Tracking Mode and Tight Tracking Mode when the criteria listed in Figure 4 are met.

Loose Tracking to Tight Tracking:

1. No one-sample change in commanded speed greater than 0.2 m/s in the last 0.5 seconds
2. Rate of change of the commanded speed is within the acceleration and deceleration limits
3. Current internal reference speed is within 0.1 m/s of the commanded speed

Tight Tracking to Loose Tracking:

1a. A one-sample change in commanded speed greater than 0.2 m/s is detected

– or –

1b. Rate of change of the commanded speed exceeds either the acceleration or deceleration limit

Figure 4: Criteria for switching between the two input tracking modes.

4.2 Acceleration and Deceleration Limits

Default Loose Tracking Mode Limits

When the LIN_ACCEL or LIN_DECEL signals in the configuration CAN message are set to 0x00, built-in default values are used for each. The default deceleration limit is a constant 1.5 m/s^2, whereas the default acceleration limit is generated from a lookup table that is a function of current vehicle speed.

The acceleration limit lookup table is shown in Figure 5. This lookup table is designed to yield comfortable transitions at all speeds. Figure 6 shows an example of internal speed and acceleration references while default limits are being used in Loose Tracking Mode.

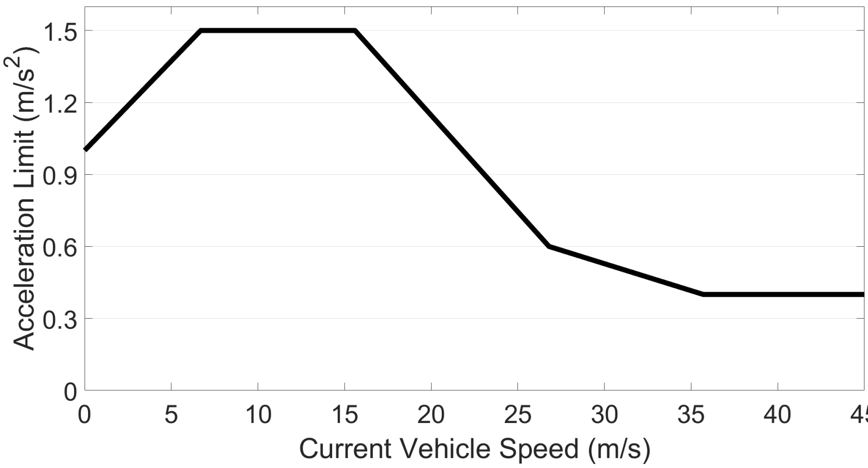


Figure 5: Default acceleration limit lookup table; acceleration vs. current speed.

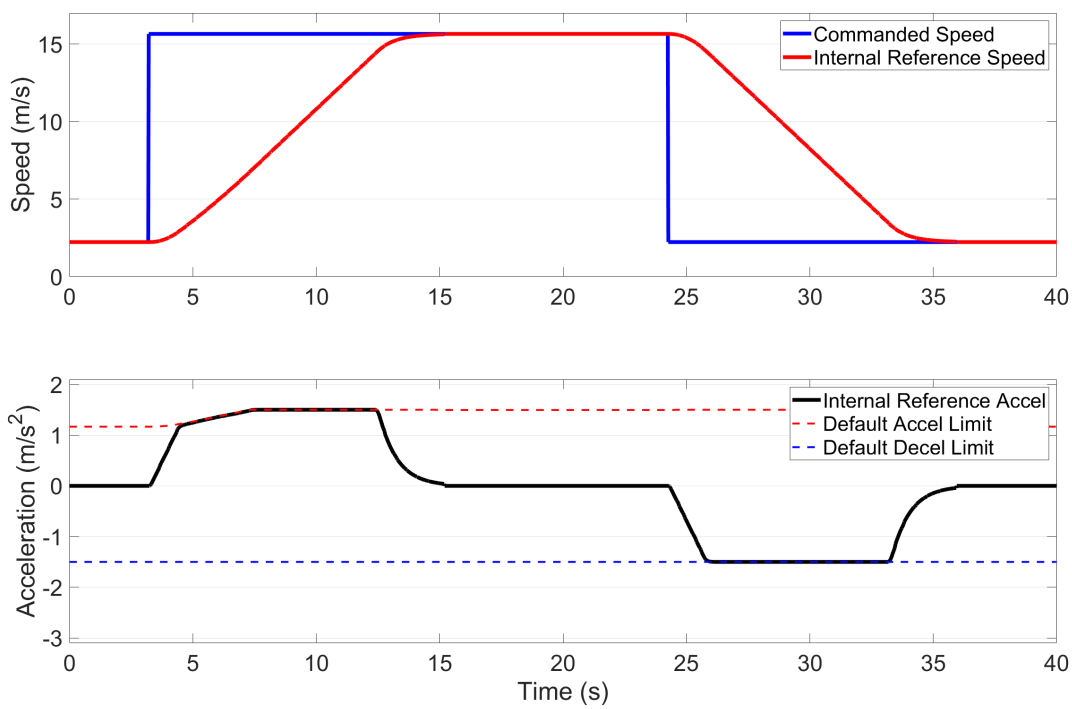


Figure 6: Loose Tracking Mode using default acceleration and deceleration limits.

User-Specified Loose Tracking Mode Limits

Setting the LIN_ACCEL or LIN_DECEL signals to non-zero values overrides the defaults. The internal acceleration reference still changes gradually, but saturates at the specified values. An example of internal speed and acceleration references in Loose Tracking Mode with non-zero limit values is shown in Figure 7.

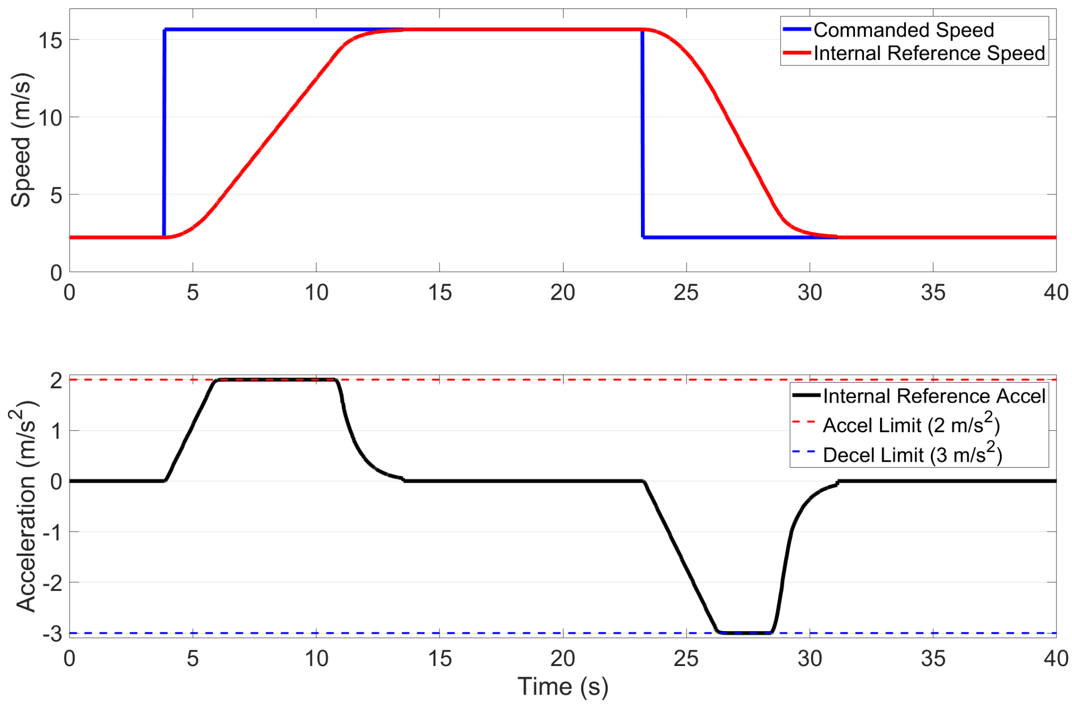


Figure 7: Loose Tracking Mode with user-specified acceleration and deceleration limit values.

Acceleration Limits in Tight Tracking Mode

When the system is in Tight Tracking Mode, acceleration and deceleration limits specified in LIN_ACCEL and LIN_DECEL are used for a different purpose. In this case, they are used to govern the thresholds for transitioning to Loose Tracking Mode instead of regulating the internal references.

As discussed in Section 4.1, if the rate of change of the commanded speed signal is greater than the acceleration limit or less than the negative of the deceleration limit, Loose Tracking Mode becomes active until the three criteria listed in Figure 4 are met again.

Figure 8 shows an example where the rate of change of a smooth commanded speed signal stays within the acceleration and deceleration limits of 2 m/s^2. In this case, the system remains in Tight Tracking Mode because the acceleration and deceleration limits are never exceeded.

Figure 9 is an example with the same commanded speed signal, but with the acceleration and deceleration limits reduced 1 m/s^2. In this case, the internal reference acceleration is clamped at the limits by transitioning to Loose Tracking Mode when the rate of change in the command exceeds them.

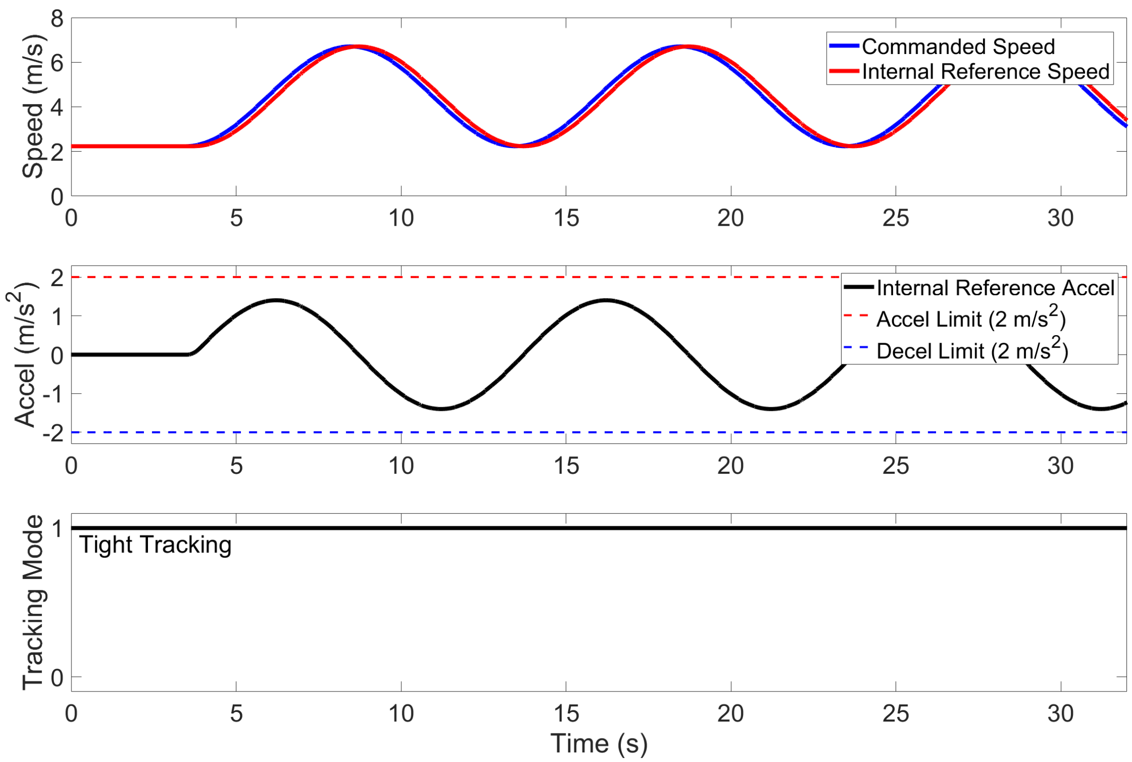


Figure 8: System stays in Tight Tracking Mode when input signal rate remains within limits.

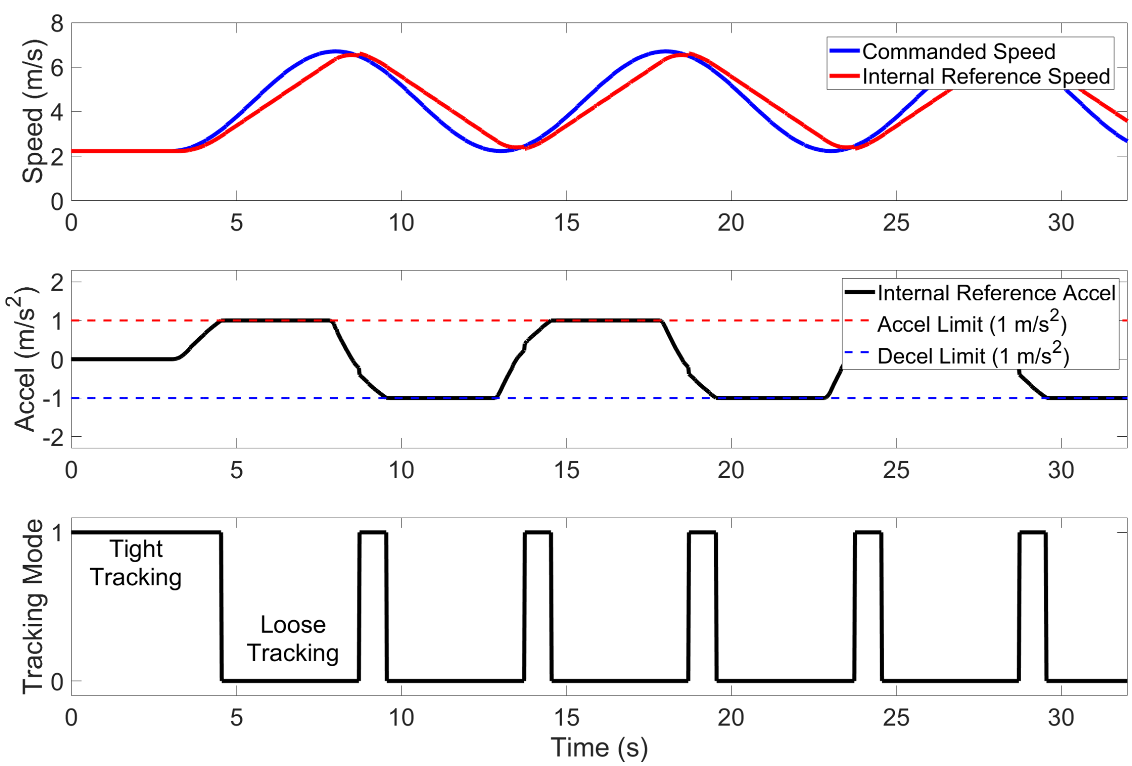


Figure 9: System transitions to Loose Tracking Mode when input signal rate exceeds limits.

4.3 Automatic Shifting Control

The ULC supports negative commanded speed inputs, where negative values indicate the desire to move backward. To achieve this, the shifter is controlled according to the following behavior, provided the user's particular vehicle is capable of drive-by-wire shifter control:

- If the vehicle is moving forward in Drive and a negative speed command is received, the vehicle first comes to a stop as if the commanded speed were zero. Then, the vehicle shifts into Reverse and tracks the negative speed command the same way it would as if the command were positive.
- The opposite case is true as well. If the vehicle is moving backward in Reverse and a positive speed command is received, the vehicle stops as if the commanded speed were zero, then shifts into Drive.
- If the vehicle is stopped and the commanded speed is zero, no gear shift is commanded.

See the automatic gear shift test results in Section 7.1.7 for an example of speed control with automatic shifting.

5 Kinematic Steering Control

The steering component of the ULC uses a kinematic model of the vehicle's steering geometry to generate open-loop steering wheel angle commands. These open-loop steering wheel angle commands can be used to control yaw rate or turning radius.

The control mode can be switched between curvature and yaw rate by setting or clearing the CURV bit in the command message. In curvature mode, the YAW_CMD signal in the command message is interpreted as a desired curvature in 1/m, and in yaw rate mode it is interpreted as a desired yaw rate in rad/s.

5.1 Turning Radius / Curvature

In curvature mode, steering wheel angle commands (α_s) are computed from a desired curvature (κ) according to:

$$\alpha_s = \gamma \tan^{-1} (L\kappa) \tag{1}$$

where the curvature is the inverse of turning radius, γ is the gear ratio between the steering wheel and the steering rack, and L is the wheelbase of the vehicle.

5.2 Yaw Rate

In yaw rate mode, steering wheel angle commands (α_s) are computed from a desired yaw rate ($\dot{\psi}$) according to:

$$\alpha_s = \gamma \tan^{-1} \left(\frac{L\dot{\psi}}{v} \right) \tag{2}$$

where v is the current speed of the vehicle.

At low speed, small changes in the yaw rate input result in large steering wheel angle outputs because the arctangent argument in (2) is obtained by dividing by vehicle speed. Therefore, it is recommended to use curvature mode for low-speed steering maneuvers instead. The speed measurement (v) used for computing steering wheel angle commands is saturated to a minimum of 0.5 m/s to ensure numerical stability.

5.3 Lateral Acceleration Limit

The user can limit the maximum allowed steering wheel angle ($\alpha_{s_{\max}}$) by specifying a lateral acceleration limit ($a_{y_{\max}}$) in the LAT_ACCEL signal of the configuration message. This maximum allowed steering wheel angle is computed kinematically, and is dependent on vehicle speed according to:

$$\alpha_{s_{\max}} = \gamma \tan^{-1}\left(\frac{L a_{y_{\max}}}{v^2}\right) \tag{3}$$

If LAT_ACCEL is set to 0x00, a default of 4 m/s^2 is used. The current maximum steering wheel angle is available to the user in the MAX_ANG signal of the report message.

At low speed, the computed maximum steering wheel angle can exceed the physical limit of the steering wheel. In this case, the reported value in MAX_ANG saturates at the physical limit to indicate that there is no maximum angle restriction based on the lateral acceleration constraint specified in LAT_ACCEL.

5.4 Angular Acceleration Limit

The user can limit the maximum allowed angular rate of the steering wheel ($\dot{\alpha}_{s_{\max}}$) by specifying a yaw angular acceleration limit ($\ddot{\psi}_{\max}$) in the ANG_ACCEL signal of the configuration message. This maximum allowed rate is computed kinematically, and is dependent on the current vehicle speed and the current steering wheel angle according to:

$$\dot{\alpha}_{s_{\max}} = \frac{\gamma L}{v} \cos^2\left(\frac{\alpha_s}{\gamma}\right) \ddot{\psi}_{\max} \tag{4}$$

If ANG_ACCEL is set to 0x00, a default of 1 rad/s^2 is used. The current maximum angular rate for the steering wheel is avaliable to the user in the MAX_RATE signal of the report message.

At low speed, the computed maximum angular rate of the steering wheel can exceed the physical limit of the steering wheel. Just as with the lateral acceleration limit, the MAX_RATE value saturates at the physical limit to indicate that the rate is not being restricted by the yaw angular acceleration constraint specified in ANG_ACCEL.

6 CAN Message Structure

6.1 Command

Message ID: 0x076
Receive Rate: 20ms
Receive Timeout: 100ms

Table 4: Universal Lat/Lon Controller Command CAN Message Description.

Byte	Bits	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	7:0	LIN_VEL<7:0>							
1	15:8	LIN_VEL<15:8>							
2	23:16	YAW_CMD<7:0>							
3	31:24	YAW_CMD<15:8>							
4	39:32	—	—	CLEAR	PEDALS	STEER	SHIFT	PARK	CURV
5	47:40	—	—	—	—	—	—	—	—
6	55:48	—	—	—	—	—	—	—	—
7	63:56	—	—	—	—	—	—	—	—

bit 0-15 **LIN_VEL:** Desired vehicle speed
Units: m/s
Resolution: 0.0025 m/s / lsb
Type: int16
Saturated Minimum: 0xF510 = -7 m/s
Saturated Maximum: 0x4650 = 45 m/s

bit 16-31 **YAW_CMD:** Desired steering (yaw rate or curvature, depending on the CURV bit setting)
Units: rad/s if CURV == 0, 1/m if CURV == 1
Resolution: 2.5×10^{-4} rad/s / lsb if CURV == 0, 6.1×10^{-6} 1/m / lsb if CURV == 1
Type: int16
Minimum: 0x8000 (full right turn) = -8.192 rad/s if CURV == 0, -0.1999 1/m if CURV == 1
Maximum: 0x7FFF (full left turn) = 8.1915 rad/s if CURV == 0, 0.1999 1/m if CURV == 1

bit 32 **CURV:** Steering mode switch
0 = Yaw rate mode
1 = Curvature mode

bit 33 **PARK:** Enable shifting out of Park
0 = disable
1 = enable

bit 34 **SHIFT:** Enable control of the shifter
0 = disable
1 = enable

bit 35 **STEER:** Enable control of steering
0 = disable
1 = enable

bit 36 **PEDALS:** Enable control of the brake and throttle pedals to regulate speed
0 = disable
1 = enable

bit 37 **CLEAR:** Clear driver override flag
0 = normal operation
1 = request clear of driver override

bit 38-63 **Unimplemented:** Set to ‘0’

6.2 Configuration

Message ID: 0x077
Receive Rate: 200ms
Receive Timeout: 1000ms

Table 5: Universal Lat/Lon Controller Configuration CAN Message Description.

Byte	Bits	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	7:0	LIN_ACCEL<7:0>							
1	15:8	LIN_DECEL<7:0>							
2	23:16	LAT_ACCEL<7:0>							
3	31:24	ANG_ACCEL<7:0>							
4	39:32	—	—	—	—	—	—	—	—
5	47:40	—	—	—	—	—	—	—	—
6	55:48	—	—	—	—	—	—	—	—
7	63:56	—	—	—	—	—	—	—	—

bit 0-7 **LIN_ACCEL:** Maximum linear acceleration
Units: m/s^2
Resolution: 0.025 m/s^2 / lsb
Type: uint8
Default: 0x00 = Use built-in speed-dependent LUT to limit acceleration
Saturated Minimum: 0x0C = 0.3 m/s^2
Saturated Maximum: 0x78 = 3.0 m/s^2

bit 8-15 **LIN_DECEL:** Maximum linear deceleration
Units: m/s^2
Resolution: 0.025 m/s^2 / lsb
Type: uint8
Default: 0x00 = 1.5 m/s^2
Saturated Minimum: 0x0C = 0.3 m/s^2
Saturated Maximum: 0xF0 = 6.0 m/s^2

bit 16-23 **LAT_ACCEL:** Maximum lateral acceleration to limit steering angle
Units: m/s^2
Resolution: 0.05 m/s^2 / lsb
Type: uint8
Default: 0x00 = 4.0 m/s^2
Saturated Minimum: 0x14 = 1.0 m/s^2
Maximum: 0xFF = 12.75 m/s^2

bit 24-31 **ANG_ACCEL:** Maximum angular acceleration to limit steering rate
Units: rad/s^2
Resolution: 0.02 rad/s^2 / lsb
Type: uint8
Default: 0x00 = 1 rad/s^2
Saturated Minimum: 0x19 = 0.5 rad/s^2
Maximum: 0xFF = 5.1 rad/s^2

bit 32-63 **Unimplemented:** Set to ‘0’

6.3 Report

Message ID: 0x078
Transmit Rate: 20ms

Table 6: Universal Lat/Lon Controller Report CAN Message Description.

Byte	Bits	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	7:0	SPEED_REF<7:0>							
1	15:8	MODE	PEDALS	TMOUT	SPEED_REF<12:8>				
2	23:16	SPEED_MEAS<7:0>							
3	31:24	CURV	STEER	OVERRIDE	SPEED_MEAS<12:8>				
4	39:32	ACCEL_REF							
5	47:40	ACCEL_MEAS							
6	55:48	—	MAX_ANG						
7	63:56	PRE_SP	PRE_ST	MAX_RATE					

bit 0-12 **SPEED_REF:** Internal speed reference being tracked
Units: m/s Resolution: 0.02 m/s / lsb Type: int16

bit 13 **TMOUT:** Command timeout status
0 = Command being received
1 = Command timed out after 100 ms

bit 14 **PEDALS:** Status of throttle and brake signals being sent by the speed control system
0 = Throttle and brake signals are not being sent
1 = Throttle and brake signals are being sent

bit 15 **MODE:** Input tracking mode currently active (see Section 4.1 for details)
0 = Loose Tracking Mode
1 = Tight Tracking Mode

bit 16-28 **SPEED_MEAS:** Speed control feedback value
Units: m/s Resolution: 0.02 m/s / lsb Type: int16

bit 29 **OVERRIDE:** Driver override status
0 = No driver overrides latched
1 = One or more driver overrides latched

bit 30 **STEER:** Status of steering angle signal being sent by the steering control system
0 = Steering signals are not being sent
1 = Steering signals are being sent

bit 31 **CURV:** Steering mode status
0 = Yaw rate mode
1 = Curvature mode

bit 32-39 **ACCEL_REF:** Internal acceleration reference being tracked
Units: m/s^2 Resolution: 0.05 m/s^2 / lsb Type: int8

bit 40-47 **ACCEL_MEAS:** Acceleration control feedback value
Units: m/s^2 Resolution: 0.05 m/s^2 / lsb Type: int8

bit 48-54 **MAX_ANG:** Maximum allowed steering angle given LAT_ACCEL signal in command
Units: degrees Resolution: 5 degrees / lsb Type: uint8

bit 55 **Unimplemented:** Set to ‘0’

bit 56-61 **MAX_RATE:** Maximum allowed steering velocity given ANG_ACCEL signal in command
Units: deg/s Resolution: 8 deg/s / lsb Type: uint8

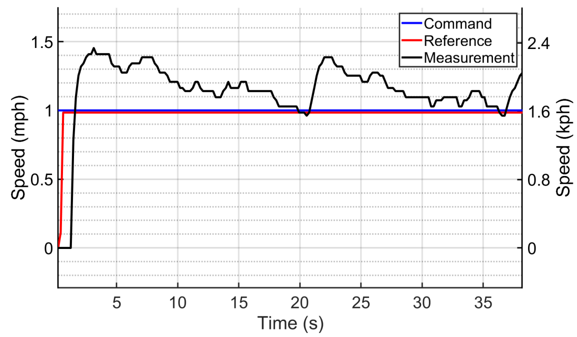
bit 62 **PRE_ST:** Steering preemption status
0 = Not being preempted
1 = Steering control would otherwise be active, but is being preempted

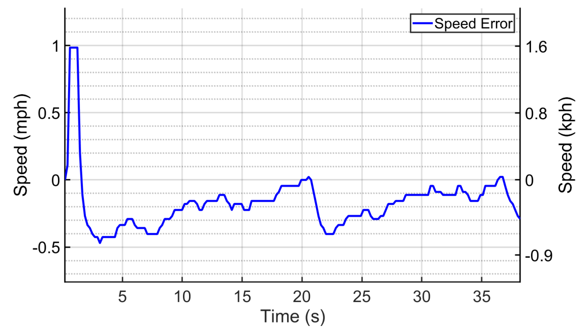
bit 63 **PRE_PD:** Pedal preemption status
0 = Not being preempted
1 = Speed control would otherwise be sending pedal commands, but is being preempted

7 Speed Control Test Cases

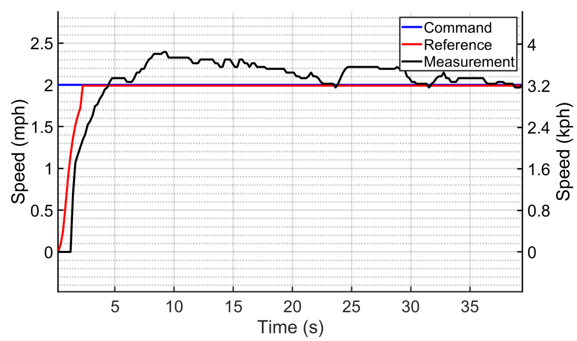
7.1 Low Speed Tests

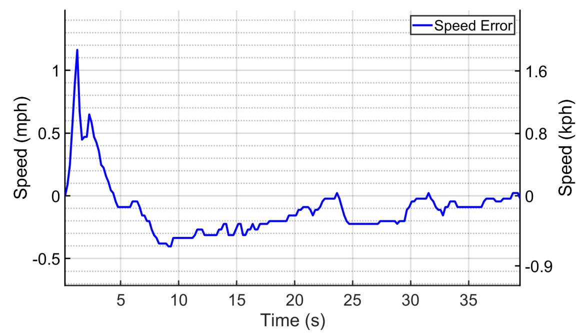
7.1.1 Constant 1 MPH



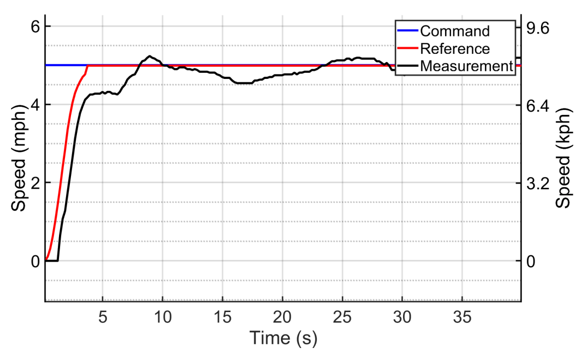


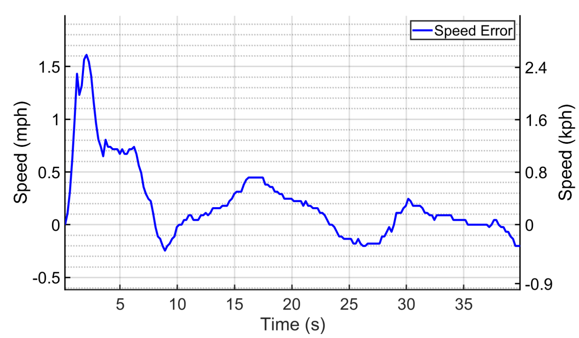
7.1.2 Constant 2 MPH



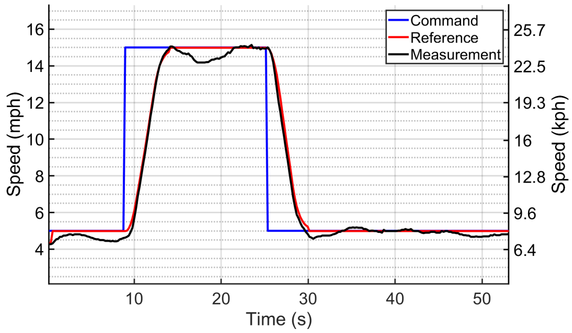


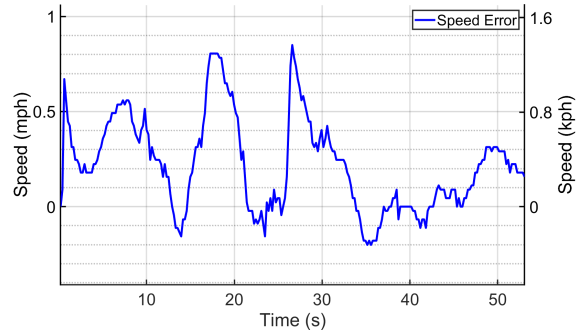
7.1.3 Constant 5 MPH



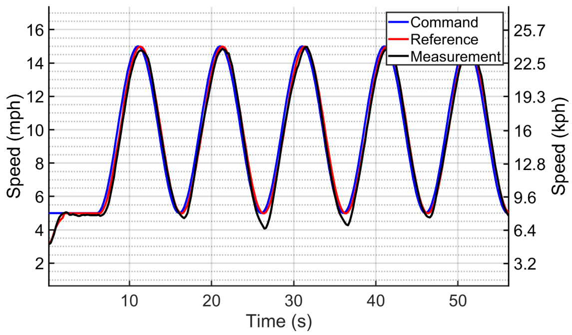


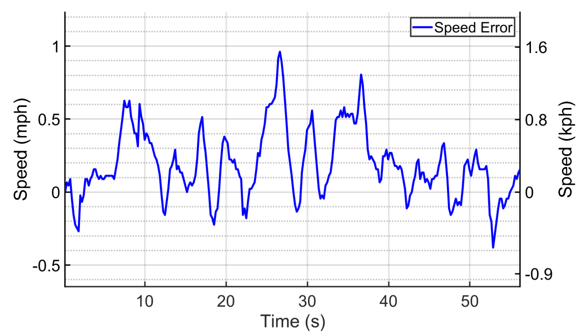
7.1.4 Step Transition (5 MPH – 15 MPH)

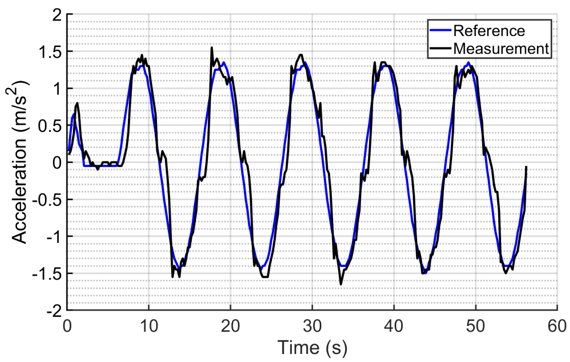


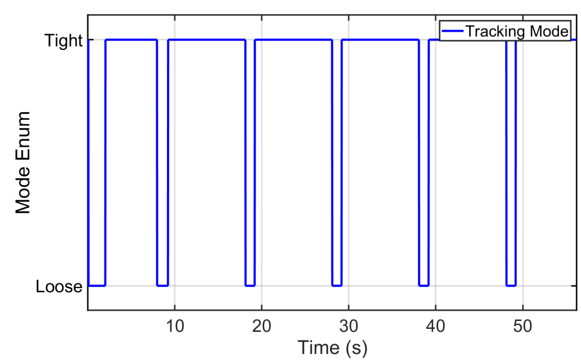


7.1.5 Sine Wave Tracking (10s Period)

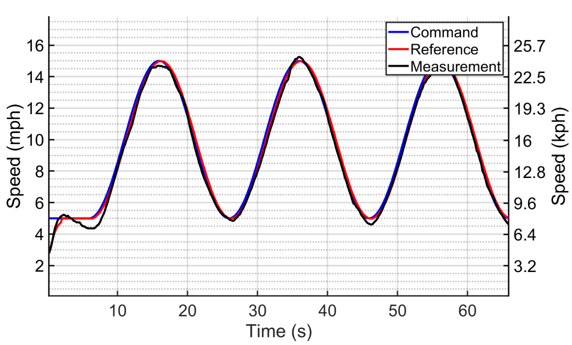


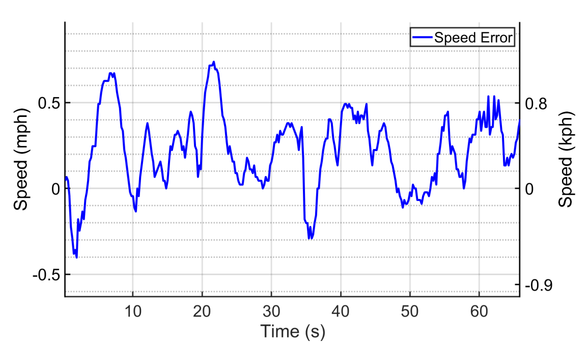


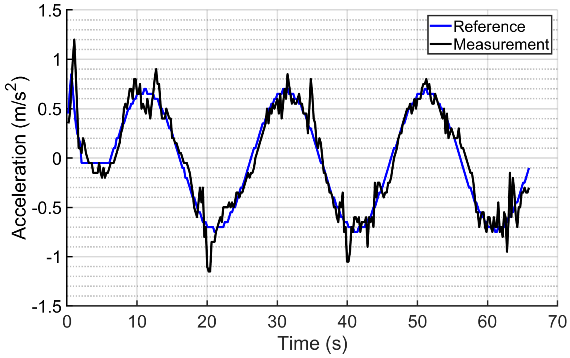


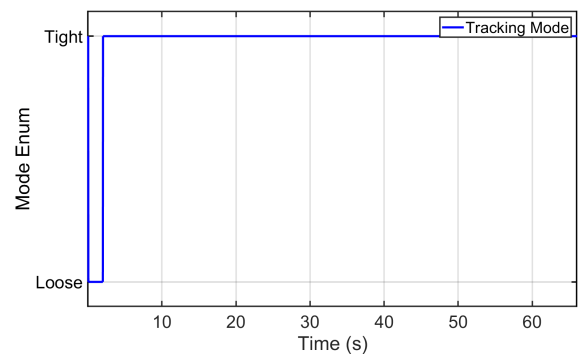


7.1.6 Sine Wave Tracking (20s Period)

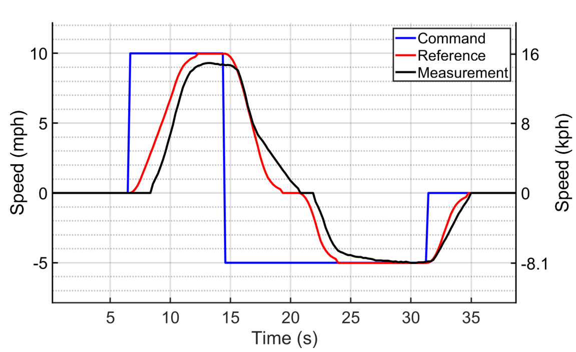


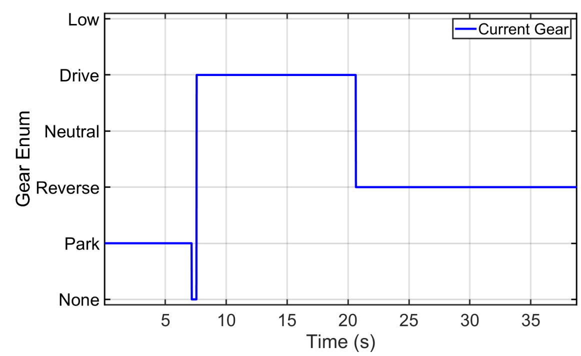




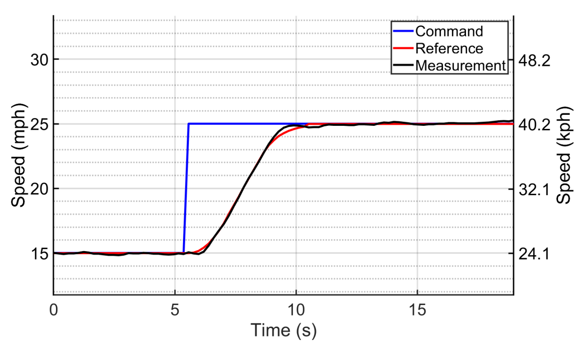


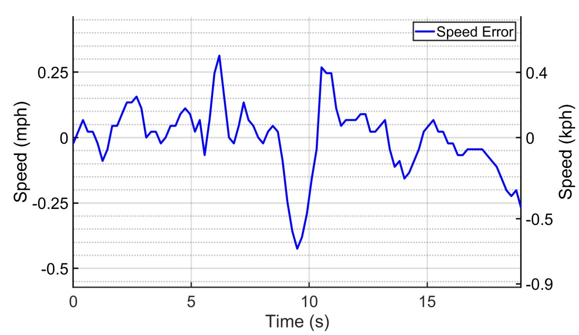
7.1.7 Automatic Gear Shift

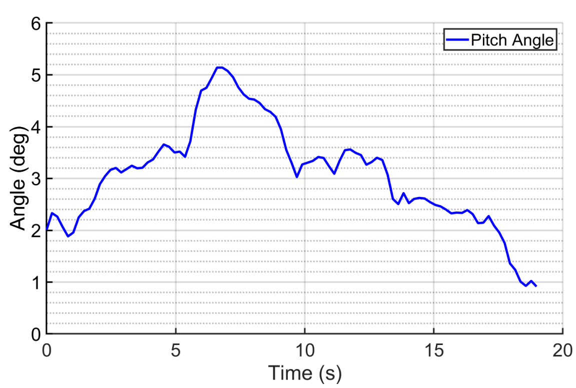




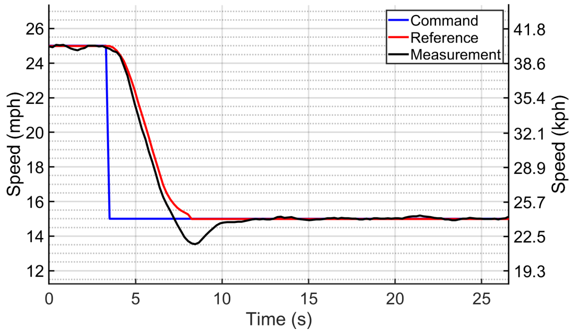
7.1.8 Uphill Step Transition (15 MPH – 25 MPH)

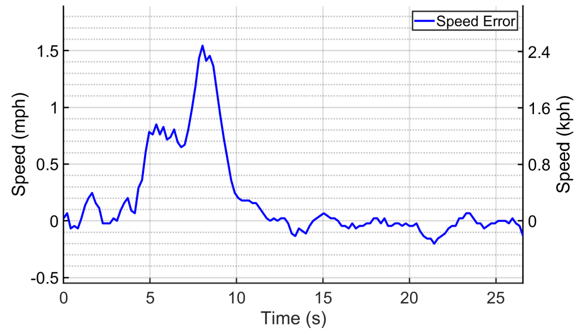


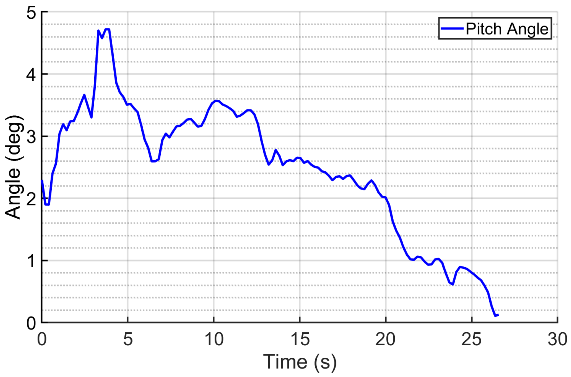




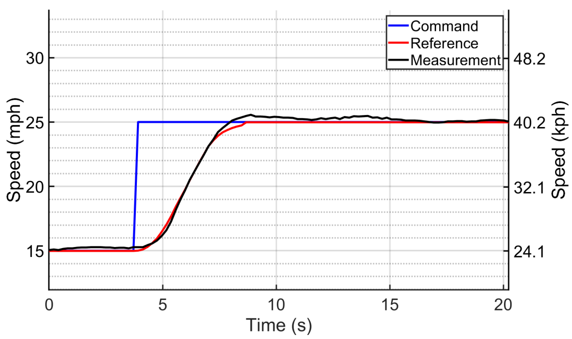
7.1.9 Uphill Step Transition (25 MPH – 15 MPH)

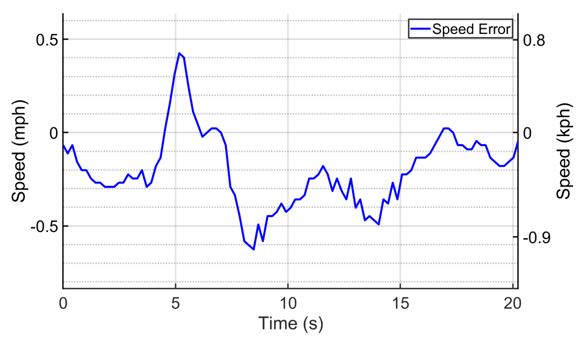


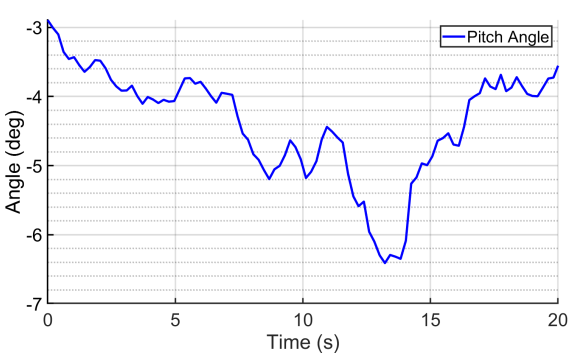




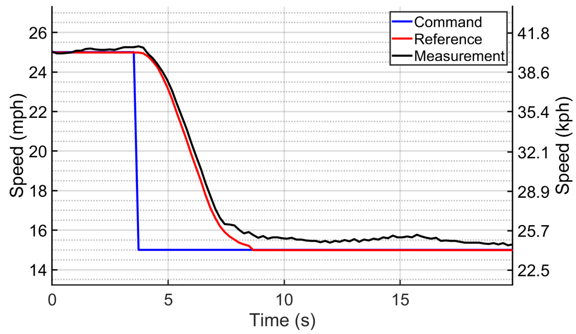
7.1.10 Downhill Step Transition (15 MPH – 25 MPH)

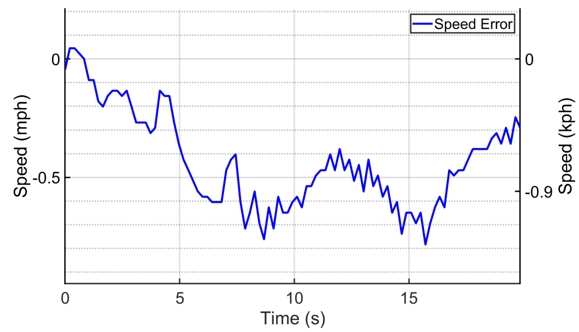


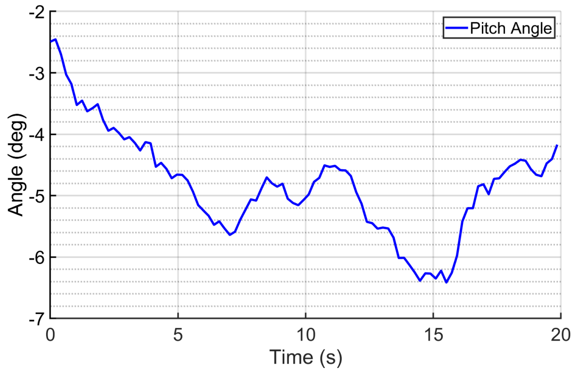




7.1.11 Downhill Step Transition (25 MPH – 15 MPH)

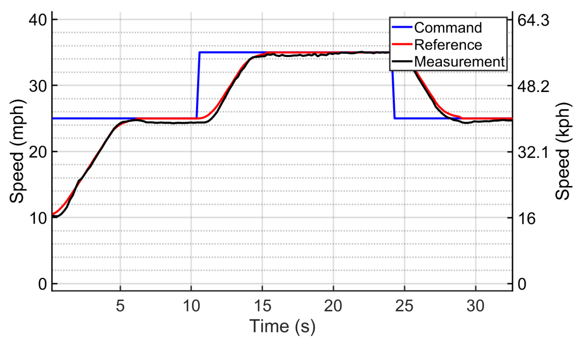


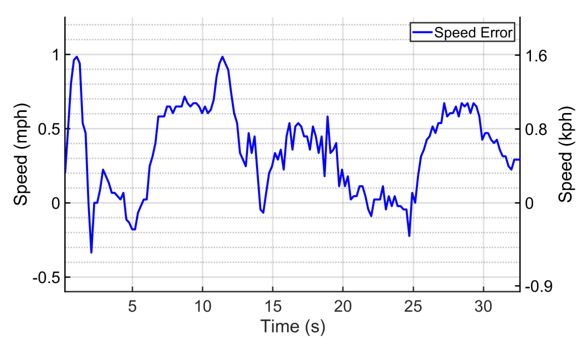




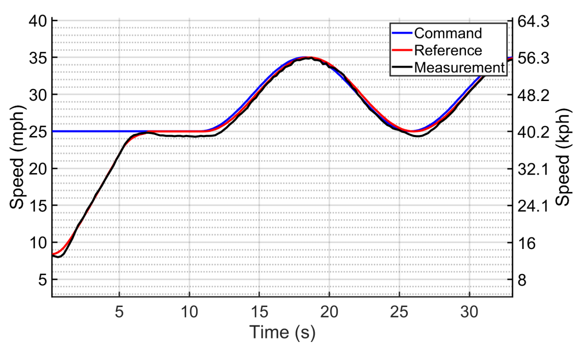
7.2 Medium Speed Tests

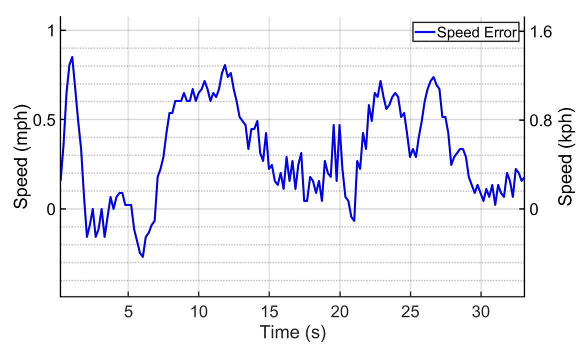
7.2.1 Step Transition (25 MPH – 35 MPH)

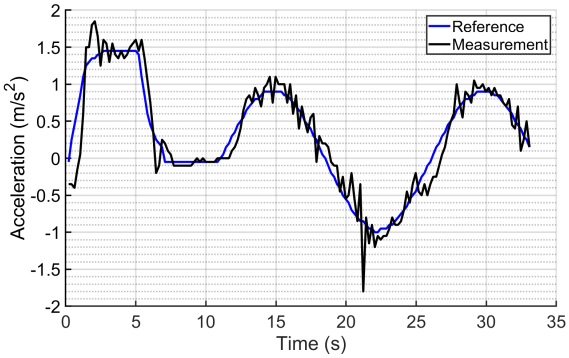


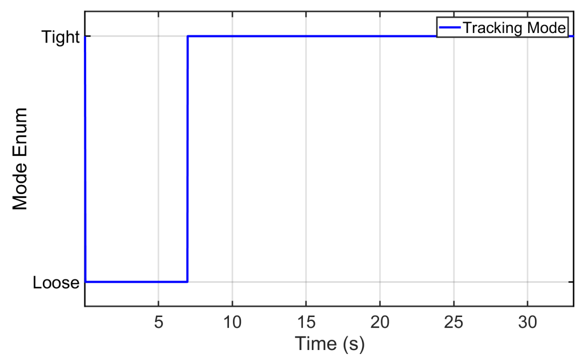


7.2.2 Sine Wave Tracking (15s Period)

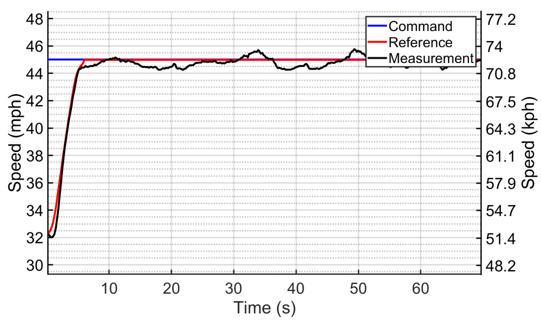


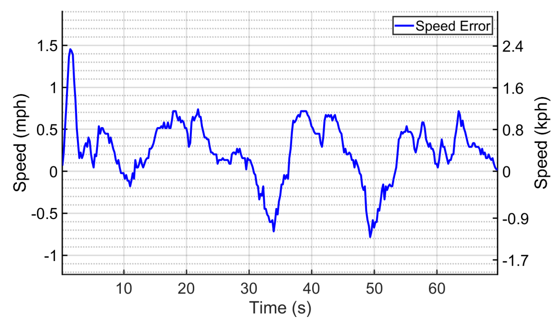


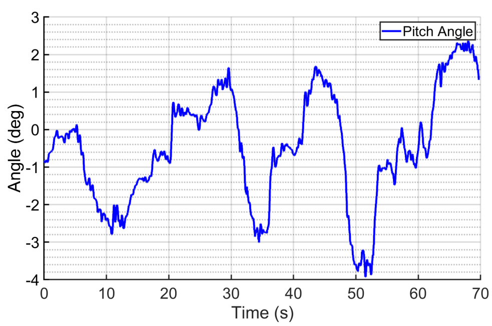




7.2.3 Maintain Speed with Variable Inclines



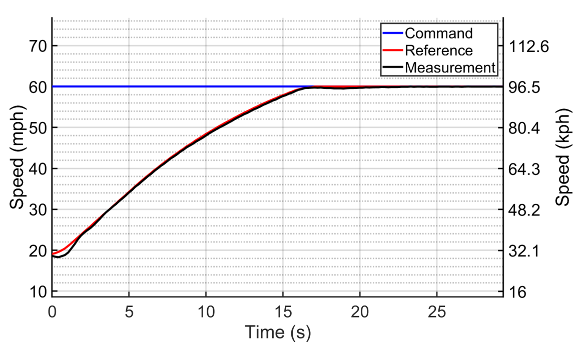


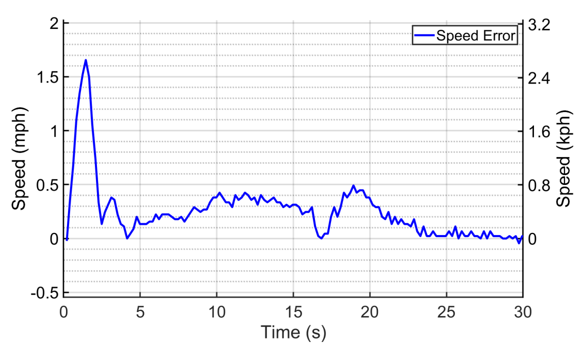


7.3 Highway Speed Tests

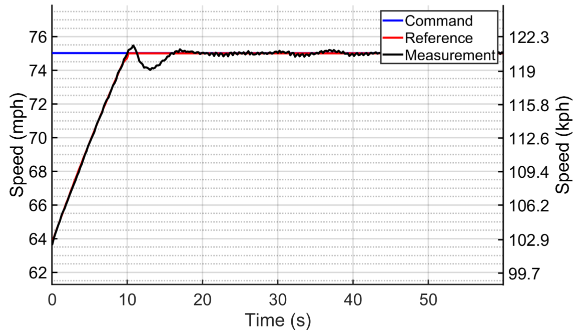
The test scenarios presented in this section represent tests performed at highway speeds using the default acceleration limit LUT and the default deceleration limit of 1.5 m/s^2.

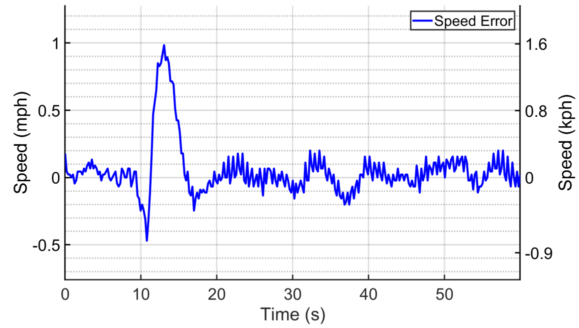
7.3.1 Reach and Maintain 60 MPH



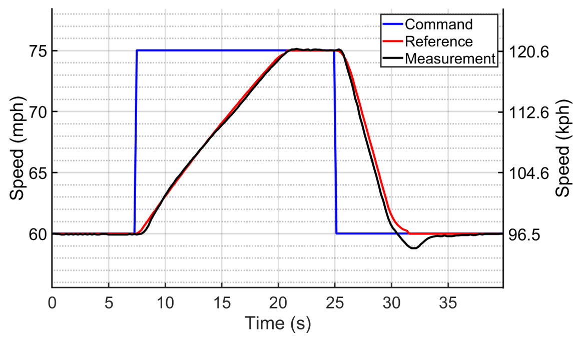


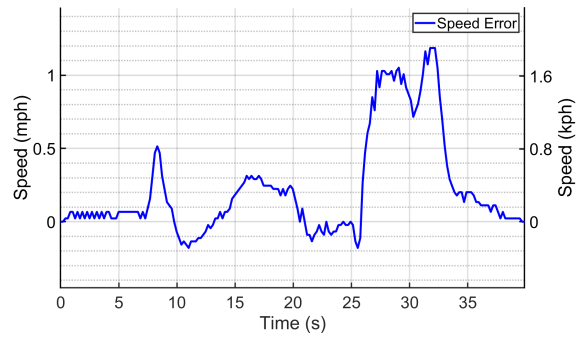
7.3.2 Reach and Maintain 75 MPH

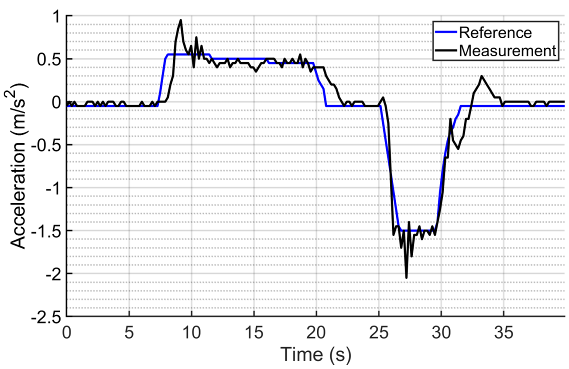


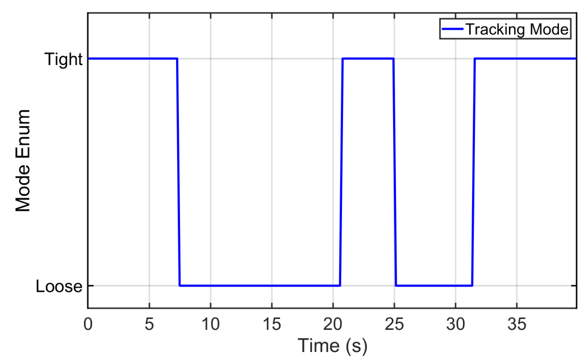


7.3.3 Step Transition (60 MPH – 75 MPH)







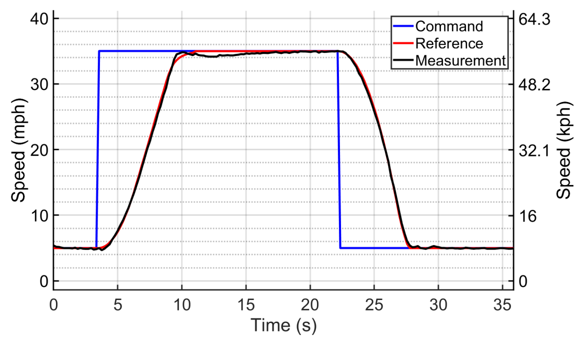


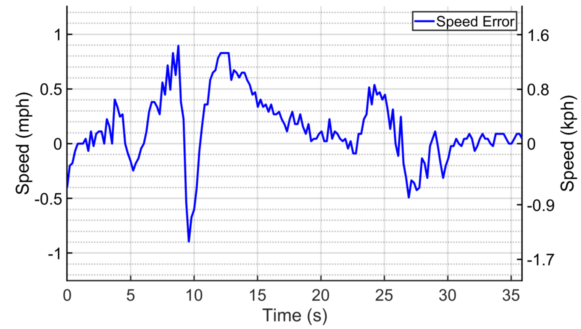
7.4 High Limit Tests

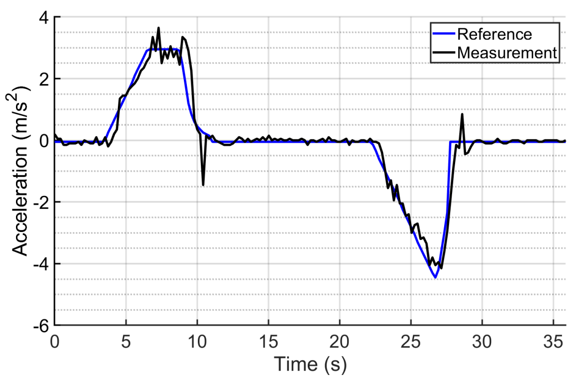
These test scenarios demonstrate the behavior of the ULC in situations where the default acceleration and deceleration limits are overridden with higher values.

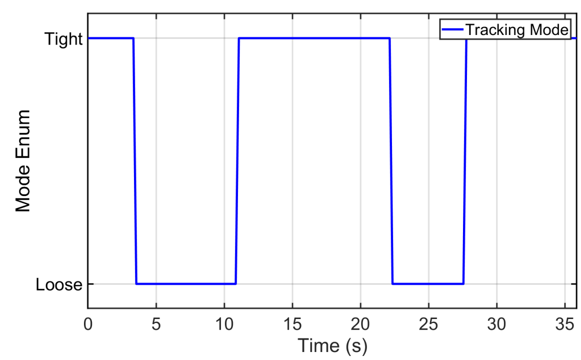
7.4.1 Step Transition (5 MPH – 35 MPH)

During this test, the acceleration limit was set to 3.0 m/s^2, and the deceleration limit was set to 5.0 m/s^2.



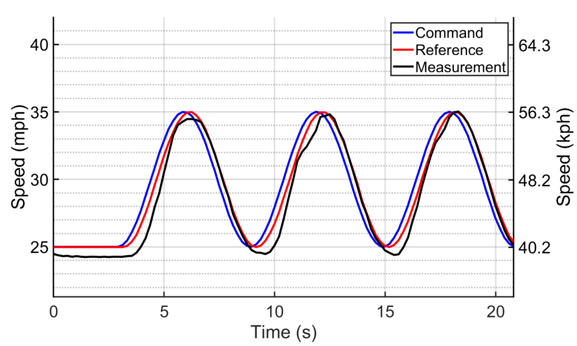


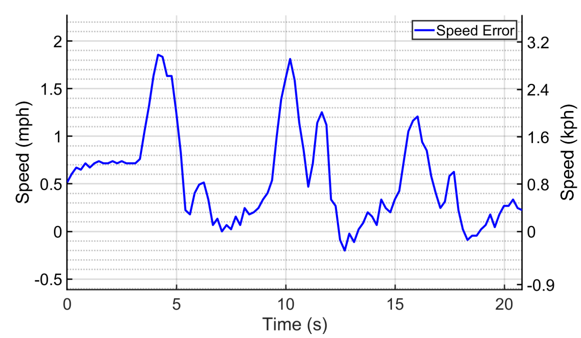


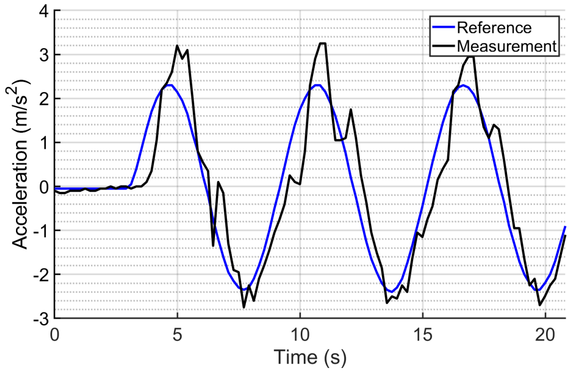


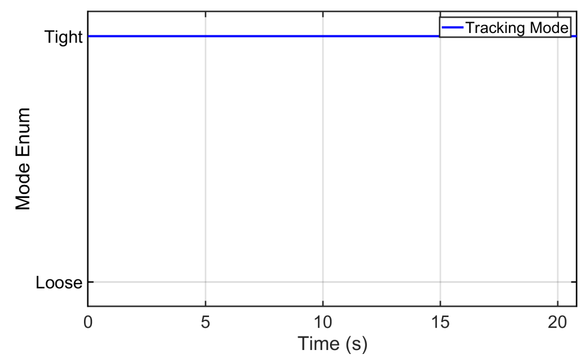
7.4.2 High-Frequency Sine Wave Tracking (6s Period)

During this test, the acceleration and deceleration limits were set to 3.0 m/s^2.



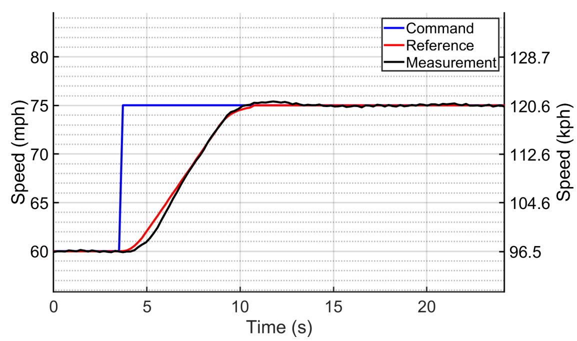


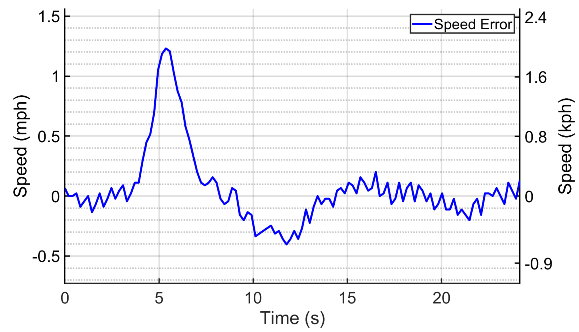


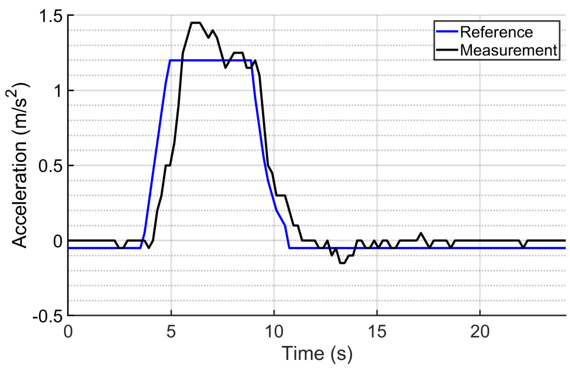


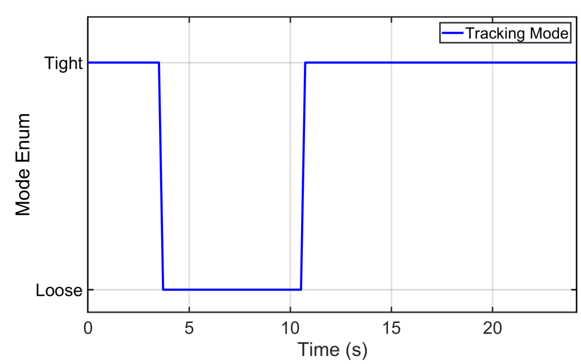
7.4.3 Step Transition (60 MPH – 75 MPH)

During this test, the acceleration limit was set to 1.2 m/s^2.









APPENDIX A: REVISION HISTORY

Revision A-01 (November 2018)
Modifications:

1. Initial release of the Universal Lat/Lon Controller.