ASM Environment

Reference

For ASM Environment Blockset 4.14 and ASM Environment Operator Blockset 4.14

Release 2021-A – May 2021

dSPACE

How to Contact dSPACE

Mail:	dSPACE GmbH Rathenaustraße 26 33102 Paderborn Germany
Tel.:	+49 5251 1638-0
Fax:	+49 5251 16198-0
E-mail:	info@dspace.de
Web:	http://www.dspace.com

How to Contact dSPACE Support

If you encounter a problem when using dSPACE products, contact your local dSPACE representative:

- Local dSPACE companies and distributors: http://www.dspace.com/go/locations
- For countries not listed, contact dSPACE GmbH in Paderborn, Germany.
  Tel.: +49 5251 1638-941 or e-mail: support@dspace.de

You can also use the support request form: http://www.dspace.com/go/supportrequest. If you are logged on to mydSPACE, you are automatically identified and do not need to add your contact details manually.

If possible, always provide the relevant dSPACE License ID or the serial number of the CmContainer in your support request.

Software Updates and Patches

dSPACE strongly recommends that you download and install the most recent patches for your current dSPACE installation. Visit http://www.dspace.com/go/patches for software updates and patches.

Important Notice

Contents

About This Reference

Content

This reference introduces you to the features provided by the ASM Environment Model (driver, road, and maneuver). It describes the structure and parts of the model, its physical background, and the data required for parameterization.

Symbols

dSPACE user documentation uses the following symbols:

Symbol	Description
⚠ DANGER	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
⚠ WARNING	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
⚠ CAUTION	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates a hazard that, if not avoided, could result in property damage.
Note	Indicates important information that you should take into account to avoid malfunctions.
Tip	Indicates tips that can make your work easier.
	Indicates a link that refers to a definition in the glossary, which you can find at the end of the document unless stated otherwise.
	Precedes the document title in a link that refers to another document.

Naming conventions

dSPACE user documentation uses the following naming conventions:

%name% Names enclosed in percent signs refer to environment variables for file and path names.

< > Angle brackets contain wildcard characters or placeholders for variable file and path names, etc.

Special folders

Some software products use the following special folders:

Common Program Data folder A standard folder for application-specific configuration data that is used by all users.

`%PROGRAMDATA%\dSPACE\<InstallationGUID>\<ProductName>`

or

`%PROGRAMDATA%\dSPACE\<ProductName>\<VersionNumber>`

Documents folder A standard folder for user-specific documents.

`%USERPROFILE%\Documents\dSPACE\<ProductName>\<VersionNumber>`

Local Program Data folder A standard folder for application-specific configuration data that is used by the current, non-roaming user.

`%USERPROFILE%\AppData\Local\dSPACE\<InstallationGUID>\<ProductName>`

Accessing dSPACE Help and PDF Files

After you install and decrypt dSPACE software, the documentation for the installed products is available in dSPACE Help and as PDF files.

dSPACE Help (local) You can open your local installation of dSPACE Help:

- On its home page via Windows Start Menu
- On specific content using context-sensitive help via **F1**

dSPACE Help (Web) You can access the Web version of dSPACE Help at www.dspace.com/go/help.

To access the Web version, you must have a *mydSPACE* account.

PDF files You can access PDF files via the icon in dSPACE Help. The PDF opens on the first page.

Overview of the Environment Library

Environment Library

Introduction

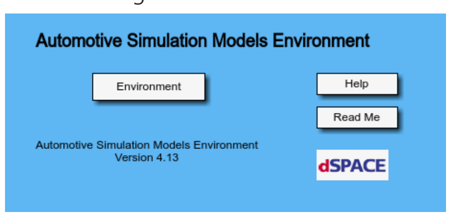
This topic gives you an overview of the ASM Environment Library.

Opening the library

You can open the library in MATLAB/Simulink. Refer to How to Open an ASM Library (ASM Vehicle Dynamics Model Description 📖).

Contents

The following illustration shows the first level of the library.



The library has one main subsystem.

Environment The Environment subsystem contains all the Simulink blocks necessary to model the vehicle environment.

It contains blocks to model the:

- Driver, refer to Driver on page 12.
- Maneuver, refer to Maneuver on page 62.
- Road, refer to Road on page 77.

Environment

Where to go from here

Information in this section

Driver

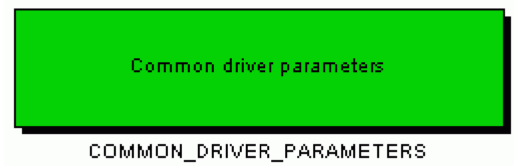
Where to go from here

Information in this section

Common Driver Parameters

Description

The driver model needs a number of vehicle parameters to function properly. The COMMON_DRIVER_PARAMETERS block contains parameters which are used more than once in the driver model and provides central access to them.



Parameters

The following table shows all common driver-related parameters:

Name	Unit	Description
Const_Angle_Max_SteeringWheel	[deg]	Maximum steering angle

Related topics

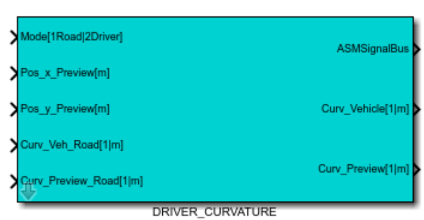
References

Common Driver Parameters (ModelDesk Parameterizing 📖)

Curvature

Description

The DRIVER_CURVATURE block provides the curvature for the reference velocity calculation.



Inports

The following table shows the inports:

Name	Unit	Description
Curv_Preview_Road	[1/m]	Curvature of the road at the position of the preview points.
Curv_Veh_Road	[1/m]	Curvature of the road at the position of the ASM vehicle.

Name	Unit	Description
Mode	[1|2]	The block supports the following modes: ▪ 1 ROAD: The block adopts the curvature values from the ASM road. The ASM road calculates the curvature on the basis of the road network geometry. The curvature calculated by the **ROAD** is fed to the last two inports of the **DRIVER_CURVATURE** block. ▪ 2 DRIVER: The block calculates the curvature on the basis of the prescribed trajectory of the ASM vehicle. For this purpose, the block calculates cubic hermite spline segments which connect the different preview points. The block calculates the curvature by using these cubic hermite spline segments.
Pos_x_Preview	[m]	x-positions of the preview points in the earth coordinate system.
Pos_y_Preview	[m]	y-positions of the preview points in the earth coordinate system.

Outports

The following table shows the outports:

Name	Unit	Description
Curv_Preview	[1/m]	Curvature at the position of the preview points. Depending on the mode at the inport, either the curvature from the ROAD (Curv_Preview_Road) is output, or the curvature is derived from the x- and y- coordinates of the preview points.
Curv_Vehicle	[1/m]	Curvature at the position of the ASM vehicle. Depending on the mode at the inport, either the curvature from the ROAD (Curv_Veh_Road) is output, or the curvature is derived from the x- and y- coordinates of the preview points.

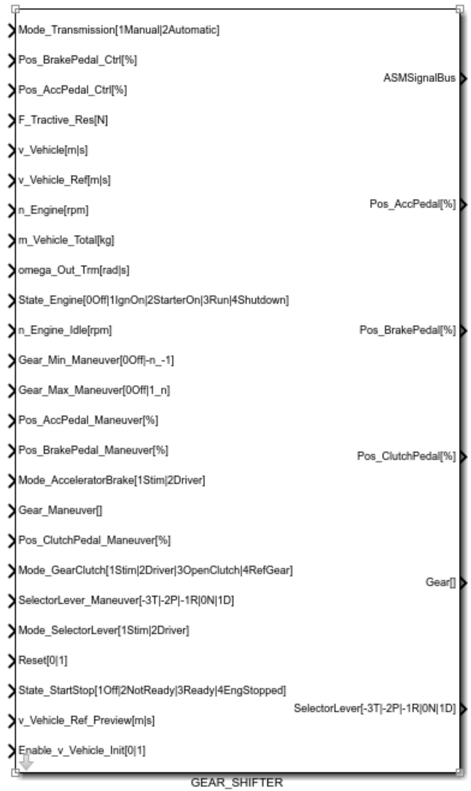
Gear Shifter

Description

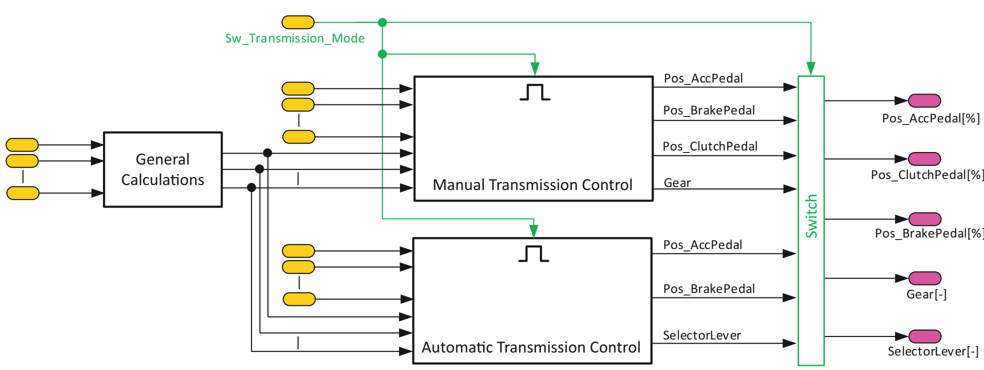
The **GEAR_SHIFTER** block simulates a part of the longitudinal driver and can be used with vehicle models equipped with manual and automatic transmissions.

In case of manual transmission, the block shifts the gear and controls the clutch pedal. Additionally, it ensures that the vehicle starts smoothly and prevents the engine from being stalled.

If it is used with automatic transmission, it controls the selector lever. The gearshift itself is performed by the SOFT_ECU_TRANSMISSION block.



The GEAR_SHIFTER block is divided into two main control subsystems. One subsystem builds the longitudinal behavior of the driver with manual transmission and the other the driver with automatic transmission. The following illustration shows the first level of the GEAR_SHIFTER block:



The Sw_Trm_Mode[1Manual|2Automatic] inport can be used to switch between the two control subsystems.

For more information on the two control subsystems, refer to the following topics:

- Gear Shifter - Manual Transmission Control on page 18
- Gear Shifter - Automatic Transmission Control on page 27

Inports

The following table shows the inports:

Name	Unit	Description
Enable_v_Vehicle_Init	[0|1]	Drivetrain initialization flag for scenarios with non-zero vehicle initial velocity
F_Tractive_Res	[N]	Tractive resistance
Gear_Maneuver	[]	Stimulus gear from maneuver
Gear_Max_Maneuver	[0Off|1_n]	Maximum allowed gear from maneuver
Gear_Min_Maneuver	[0Off|-n_-1]	Minimum allowed gear from maneuver
m_Vehicle_Total	[kg]	Total vehicle mass
Mode_AcceleratorBrake	[1|2]	Mode signal for accelerator and brake pedal: ▪ 1: Stimulus ▪ 2: Driver
Mode_GearClutch	[1|2|3|4]	Mode signal for gear and clutch pedal: ▪ 1: Stimulus ▪ 2: Driver ▪ 3: Open clutch ▪ 4: Reference gear
Mode_SelectorLever	[1|2]	Mode signal for selector lever: ▪ 1: Stimulus ▪ 2: Driver
n_Engine	[rpm]	Engine speed
n_Engine_Idle	[rpm]	Engine idle speed
omega_Out_Trm	[rad/s]	Transmission output speed
Pos_AccPedal_Ctrl	[%]	Calculated accelerator pedal position from the longitudinal controller
Pos_AccPedal_Maneuver	[%]	Accelerator pedal position from maneuver
Pos_BrakePedal_Ctrl	[%]	Calculated brake pedal position from the longitudinal controller
Pos_BrakePedal_Maneuver	[%]	Brake pedal position from maneuver
Pos_ClutchPedal_Maneuver	[%]	Clutch pedal position from maneuver
Reset_States	[0|1]	Reset of states
SelectorLever_Maneuver	[-3|-2|-1|0|1]	Selector lever position from maneuver: ▪ -3: TipShift ▪ -2: Park ▪ -1: Reverse ▪ 0: Neutral ▪ 1: Drive
State_Engine	[0|1|2|3|4]	Engine state: ▪ 0: Engine off

Name	Unit	Description
		▪ 1: Ignition on ▪ 2: Ignition on and starter activated ▪ 3: Engine is running ▪ 4: Ignition is switched off, shutdown active
State_StartStop	[1|2|3|4]	State of start-stop system: ▪ 1: Switched off ▪ 2: Not ready (conditions are not fulfilled) ▪ 3: Ready (waiting for driver action) ▪ 4: Engine actively stopped by the system
Sw_Trm_Mode	[1|2]	Switch for manual or automatic transmission: ▪ 1: Manual ▪ 2: Automatic
v_Vehicle	[m/s]	Vehicle velocity
v_Vehicle_Ref	[m/s]	Reference vehicle velocity
v_Vehicle_Ref_Preview	[m/s]	Preview reference vehicle velocity

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 📖).
Gear	[]	Gear
Pos_AccPedal	[%]	Accelerator pedal position
Pos_BrakePedal	[%]	Brake pedal position
Pos_ClutchPedal	[%]	Clutch pedal position
SelectorLever	[-3|-2|-1|0|1]	Selector lever position: ▪ -3: TipShift ▪ -2: Park ▪ -1: Reverse ▪ 0: Neutral ▪ 1: Drive

Parameters

The following table shows the parameters:

Name	Unit	Description
Const_a_Tractive_Res_UpLim	[m/s^2]	Tractive resistance acceleration threshold for startup
Const_a_x_Vehicle_LowLim	[m/s^2]	Hard braking threshold for gear hold
Const_a_x_Vehicle_UpLim	[m/s^2]	Acceleration threshold for startup
Const_D_Gain_StartUp	[s]	D gain of accelerator pedal startup controller
Const_I_Gain_ClutchPedal	[1/s]	I gain of clutch pedal controller
Const_I_Gain_StartUp	[1/s]	I gain of accelerator pedal startup controller

Name	Unit	Description
Const_n_Engine_StartUp_Set	[rpm]	Engine speed setpoint during startup
Const_P_Gain_ClutchPedal	[]	P gain of clutch pedal controller
Const_P_Gain_StartUp	[]	P gain of accelerator pedal startup controller
Const_t_Gear_Neutral	[s]	Time of gear passing through neutral during gear change
Const_t_Hold_ClutchPedal	[s]	Disengaged clutch hold time
Const_v_x_Vehicle_LowLim	[m/s]	Low velocity threshold
Map_Gear_StartUp	[]	Startup gear
Map_i_Ratio	[]	Gear ratio = f(gear)
Map_n_Engine_Downshift_Lim	[rpm]	Down shift engine speed limit = f(Pos_AccPedal)
Map_n_Engine_Upshift_Lim	[rpm]	Up shift engine speed limit = f(Pos_AccPedal)
Map_omega_Out_Trm_Shift_Bias	[rpm]	Transmission output speed shift bias = f(Tractive resistance acceleration)
Map_Pos_AccPedal	[0_1]	Accelerator pedal factor during gear change = f(Pos_ClutchPedal)
Map_Pos_ClutchPedal_Disenagement	[%]	Clutch pedal disengagement position not during startup = f(t)
Map_Pos_ClutchPedal_Disenagement_StartUp	[%]	Clutch pedal disengagement position during startup = f(t)
Map_Pos_ClutchPedal_Enagement	[%]	Clutch pedal engagement position not during startup = f(t)
Map_Pos_ClutchPedal_Enagement_StartUp	[%]	Clutch pedal engagement position during startup = f(t)
StepSize	[s]	Simulation step size
Sw_ClutchPedal_EngineOff	[0|1]	Switch to disengage the clutch if engine is not running
Sw_Upshift_Gear_Skipping	[1|2]	Switch for gear skipping: ▪ 1: On ▪ 2: Off

Related topics

References

Gear Shifter (ModelDesk Parameterizing 🕮)
Gear Shifter - Automatic Transmission Control.. 27
Gear Shifter - Manual Transmission Control... 18

Gear Shifter - Manual Transmission Control

Description

In the Control_ManualTransmission subsystem, the gearshift process and the clutch pedal are controlled according to the driving situation.

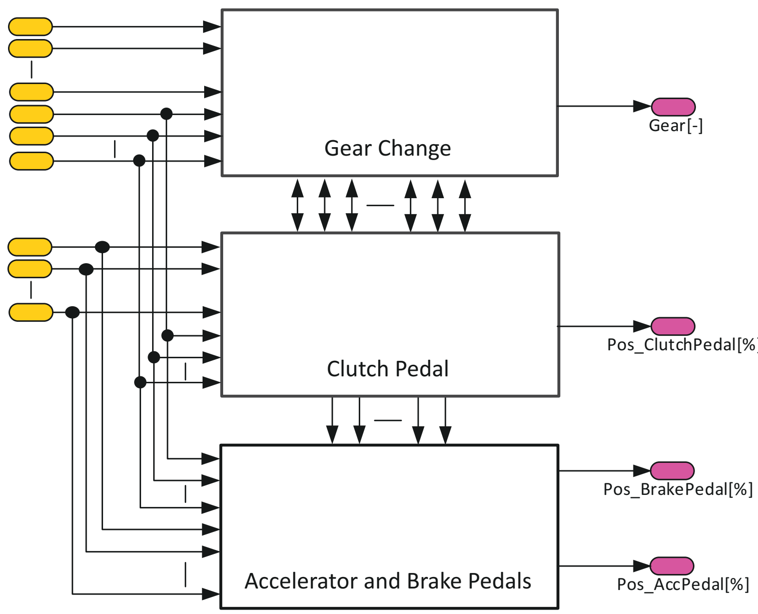
The Control_ManualTransmission subsystem ist part of the GEAR_SHIFTER block. Refer to Gear Shifter on page 14.

The driving situation is analyzed and a gear is calculated. This gear is sent as a request for clutch disengagement. When the clutch is completely disengaged, the requested gear is set and the clutch is engaged afterwards.

The manual transmission control is implemented in three subsystems:

- Clutch_Pedal: controls the clutch pedal
- Gear_Change: simulates the gearshift
- Accelerator_And_Brake_Pedals: controls the accelerator pedal during startup

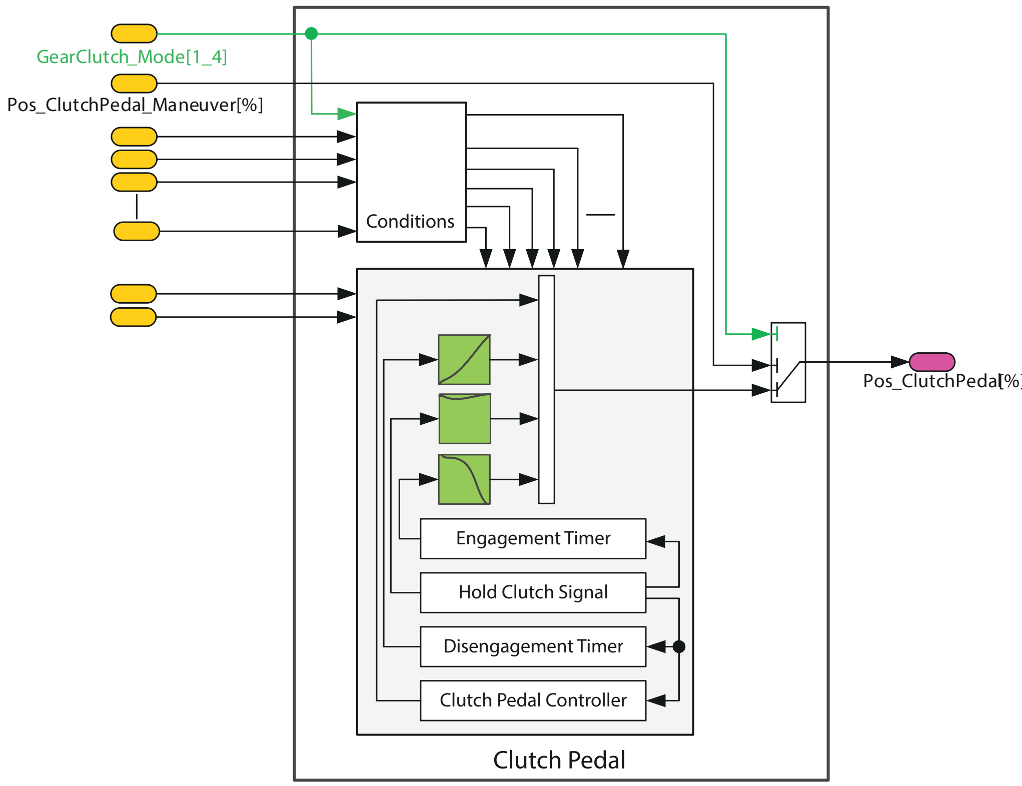
The following illustration shows how these subsystems are interconnected:



The calculations inside each subsystem are described below.

Clutch_Pedal subsystem

In the Clutch_Pedal subsystem, the clutch pedal activation and deactivation are triggered according to the requested gear and other conditions. The clutch pedal position during actuation is defined via look-up tables. It can also be controlled in special driving situations. Furthermore, the clutch pedal position can be stimulated by using external maneuver signals. The following figure illustrates this subsystem:



In addition to the gearshifting request, there are several conditions that control the actuation of the clutch pedal. Most of these conditions are related to user-defined parameters. Refer to the following table:

Condition Name	Description
Low_Engine_Speed	Indicates whether a low engine speed has been reached. Parameter: none Calculation unit: [rpm] True: {engine speed} ≤ {current engine idle speed + 100} False: {engine speed} ≥ {current engine idle speed + 500}
Engine_On	Indicates whether the engine is on. Parameter: none Calculation unit: [] The signal is calculated from the State_Engine[0|1|2|3|4] input signal and is active if the engine state equals 3. Otherwise, it is inactive.
Hard_Braking	Indicates whether the driver is braking hard. Parameter: Const_a_x_Vehicle_LowLim Calculation unit: [m/s^2] True: {reference vehicle acceleration} < {parameter} False: otherwise
Small_Velocity	Indicates whether there is a small reference vehicle velocity. Parameter: Const_v_x_Vehicle_LowLim Calculation unit: [m/s]

Condition Name	Description
	True: {reference vehicle velocity} ≤ {parameter} False: { reference vehicle velocity} ≥ {parameter + 0.4167}
Startup	Indicates whether there is a startup. Parameter: none Calculation unit: [m/s] True: ({vehicle reference velocity} > {0.05}) & (gear = neutral) False: otherwise
Standstill	Indicates whether the vehicle is standing still. Parameter: none Calculation unit: [m/s] True: {vehicle reference velocity} ≤ {0.05} False: otherwise
Initial_Velocity	Indicates whether the maneuver is defined with initial vehicle velocity. Parameter: none Calculation unit: [m/s] True: {vehicle velocity change in one simulation step} ≥ {3} False: otherwise
Big_Acceleration	Indicates whether a high acceleration is required during the startup to follow the reference vehicle velocity. Parameter: Const_a_x_Vehicle_UpLim Calculation unit: [m/s^2] True: {reference vehicle acceleration} > {parameter} False: otherwise
Big_Tractive_Resistance_Acceleration	Indicates whether a high driving resistance has to be overcome during startup. Parameter: Const_a_Tractive_Res_UpLim Calculation unit: [m/s^2] True: {tractive resistance acceleration} > {parameter} False: otherwise

The actuation of the clutch pedal can be divided into three phases:

1. Clutch pedal activation: clutch disengagement
2. Clutch pedal hold: clutch hold
3. Clutch pedal deactivation: clutch engagement

These phases are normally carried out in succession. However, clutch pedal activation always has the highest priority. It means the clutch pedal can be activated at any time regardless of its current position.

Each phase is activated if certain conditions are fulfilled.

Clutch pedal activation The clutch pedal is activated in the following cases:

- A gear change is requested.
- The reference vehicle velocity becomes low (`Small_Velocity` condition is true).

- The driver is braking hard (`Hard_Braking` condition is true) and a low engine speed has been reached (`Low_Engine_Speed` condition is true).

Clutch pedal hold The clutch pedal is held down after the clutch pedal activation ended and as long as at least one of the following conditions is fulfilled:

- A small reference vehicle velocity is tracked (`Small_Velocity` condition is true).
- The driver continues to brake hard (`Hard_Braking` condition is true) and the engine speed is low (`Low_Engine_Speed` is true).
- There is a startup with slipping clutch where no high acceleration is required (`Big_Tractive_Resistance_Acceleration` condition and `Big_Acceleration` condition are false).

If the clutch pedal activation ended and none of the above conditions is true, then the clutch pedal is held down while the gear is being changed for a user-defined time, i.e., with the Const_t_Hold_ClutchPedal parameter.

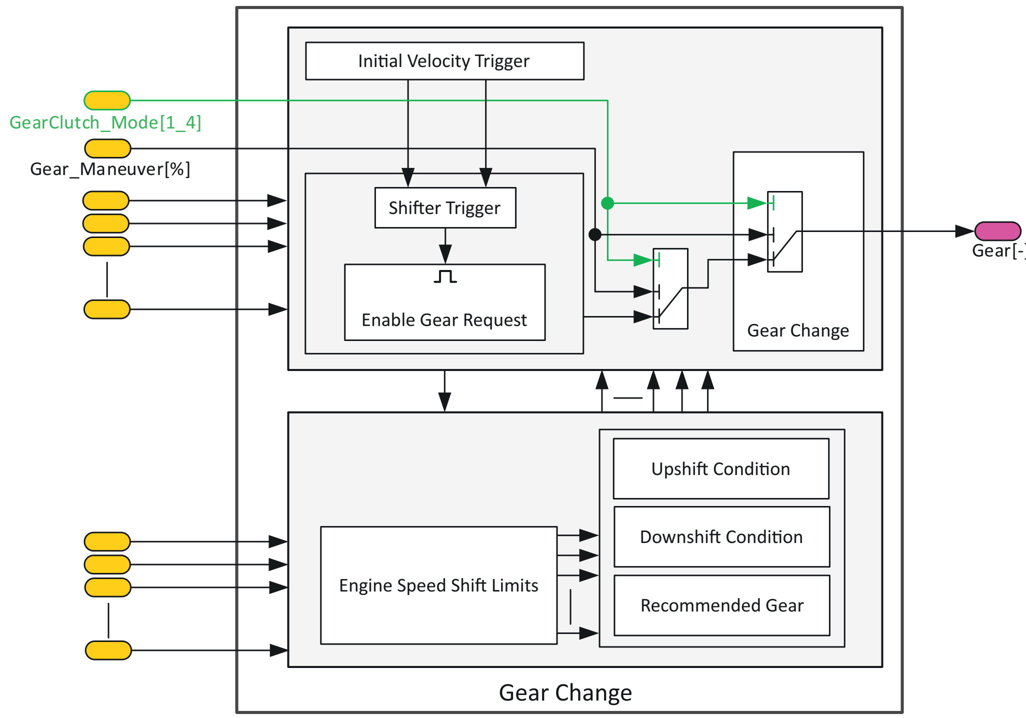
Clutch pedal deactivation After the clutch pedal hold ends, the clutch pedal is deactivated and the clutch is engaged.

The clutch pedal position is sometimes also controlled by a PI controller. The controller is active during the hold and deactivation phases of the clutch pedal after the activation phase ended and if at least one of the following conditions is fulfilled:

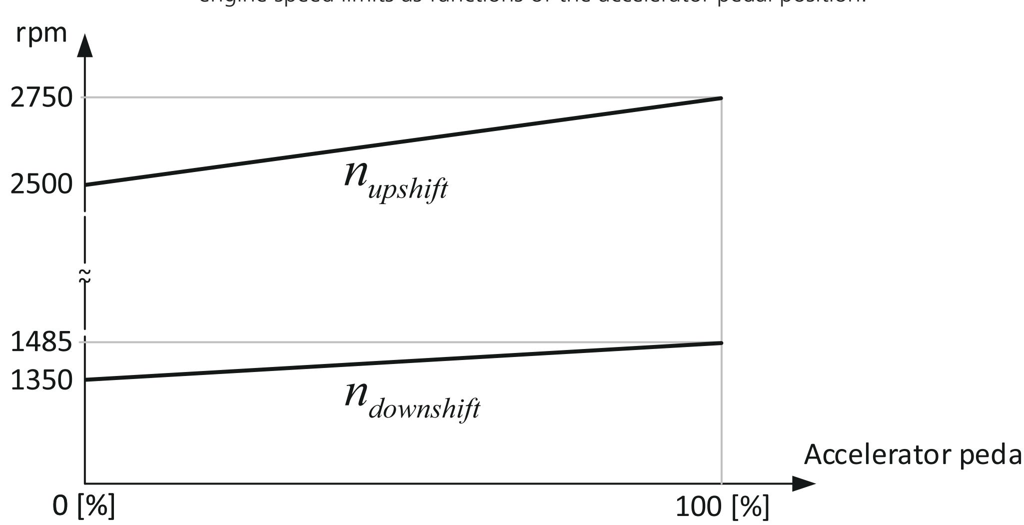
- A small reference vehicle velocity is tracked (`Small_Velocity` condition is true).
- There is a startup with slipping clutch where no high acceleration is required (`Big_Tractive_Resistance_Acceleration` condition and `Big_Acceleration` condition are false).

Gear_Change subsystem

The Gear_Change subsystem simulates the requested gear as well as the gearshift process. The requested gear can be externally stimulated or calculated according to the driving situation. The following illustration shows the subsystem.



First, the engine speed limits for the shift process and the ideal engine speed for the current and possible gears are calculated. The engine speed limits are parameterized via look-up tables as functions of the accelerator pedal position (Map_n_Engine_Upshift_Lim and Map_n_Engine_Downshift_Lim parameters). The following illustration shows exemplary values of the diesel engine speed limits as functions of the accelerator pedal position:



To calculate the ideal engine speed, the gearbox ratios and the transmission output speed are needed, with:

$$n_{current,\,ideal} = i_{current}\,\omega_{Gear,\,Out}$$

$$\bar{n}_{possible,ideal} = \bar{i}_{possible}\,\omega_{Gear,Out}$$

where:

$n_{current,ideal}$	is the ideal engine speed of the current gear
$i_{current}$	is the gearbox ratio of the current gear
$\omega_{Gear,Out}$	is the transmission output speed
$\bar{n}_{possible,ideal}$	is a vector of ideal engine speeds of all possible gears
$\bar{i}_{possible}$	is a vector of gearbox ratios of all possible gears

By using the user-defined engine speed limits for the shift process, the ideal engine speeds of the possible gears are filtered so only certain allowed gears are considered for the further calculations. The ideal engine speeds of the allowed gears have to fulfill the following condition:

$$n_{upshift} \geq \bar{n}_{allowed,ideal} \geq \left(n_{downshift} + n_{shift,bias}\right)$$

where:

$n_{upshift}$	is the up shift engine speed limit
$n_{downshift}$	is the down shift engine speed limit
$\bar{n}_{allowed,ideal}$	is a vector of the ideal engine speeds of the allowed gears
$n_{shift,bias}$	is the engine speed shift bias

The engine speed shift bias is a function of the tractive resistance acceleration and is defined in a look-up table. The value is used to avoid oscillations in the calculated allowed gears. The transmission output speed is being dropped as the driver disengages the clutch pedal, causing a drop in the calculated ideal engine speed as shown in the equations above.

Moreover, in the mentioned look-up table, the engine speed drop after the change to a lower gear is also considered, i.e., the look-up table delivers nonzero values for zero tractive resistance acceleration.

Afterwards, the recommended gear for the current driving situation is calculated from the allowed gears. That is, it is checked whether the current driver behavior indicates accelerating or braking demand. In the case of acceleration demand, the maximum allowed gear is selected. In the case of braking demand, the minimum allowed gear is selected. Thus, this calculation provides the optimal use of the available gears in many situations.

The up shift request is triggered if the ideal engine speed of the current gear becomes greater than or equals the up shift limit, as illustrated in the following equation:

$$request_{upshift} = \begin{cases} 1, & n_{current,ideal} \geq n_{upshift} \\ 0, & otherwise \end{cases}$$

The down shift request is triggered if the ideal engine speed of the current gear becomes smaller than or equals the down shift limit, as illustrated in the following equation:

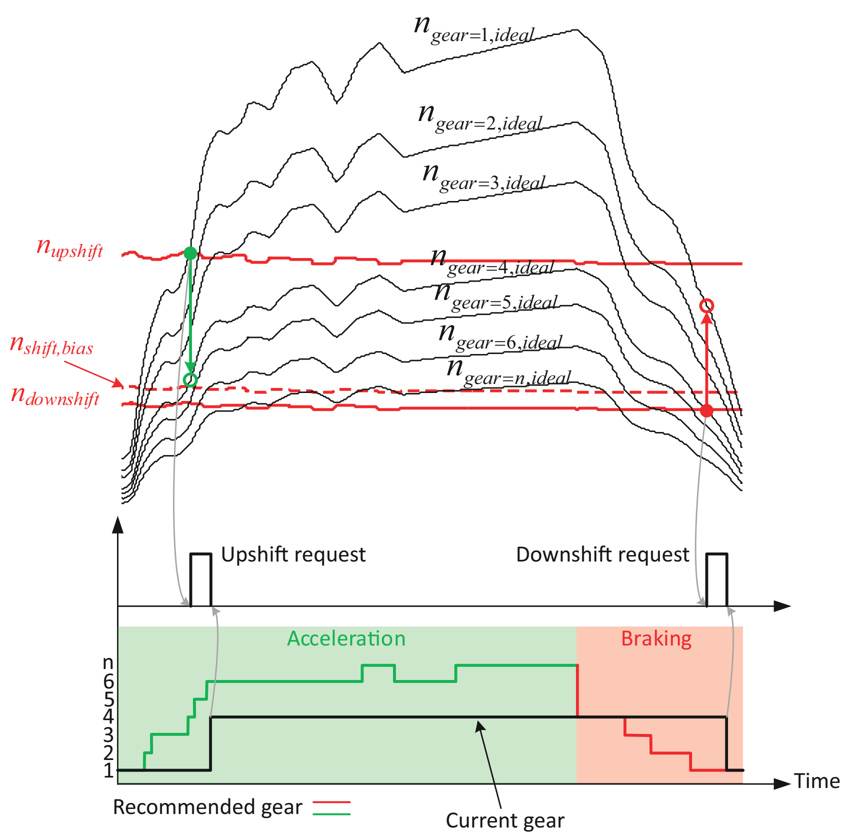
$$request_{downshift} = \begin{cases} 1, & n_{current,ideal} \leq n_{downshift} \\ 0, & otherwise \end{cases}$$

The up shift or down shift request triggers a gear change request which is sent to demand clutch disengagement. However, depending on the driving situation, these requests might be disabled. This can be the case, if the driver is braking hard (`Hard_Braking` condition is true). In this case, the gear is not changed and a shift request is sent once a low engine speed has been reached (`Low_Engine_Speed` condition is true), as if there was a real driver in the vehicle.

The gear change request then causes the clutch to be disengaged. Once the clutch is completely disengaged, a gearshift occurs.

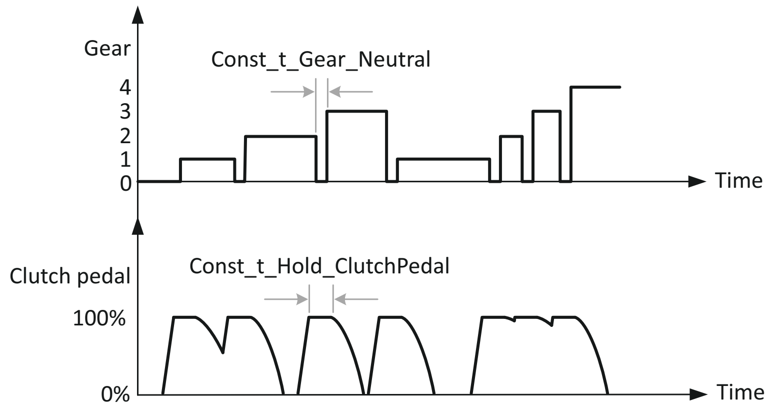
In the gear down shift process, it is also allowed to skip gears. This can happen, for example, in braking maneuvers. However, during the gear up shift, you can activate or deactivate this feature by using the Sw_Upshift_Gear_Skipping parameter.

Refer to the following illustration for a visualization of the gear calculation:



During the gearshift, you can activate the feature of passing the gear through neutral by using the Const_t_Gear_Neutral parameter. This parameter defines the time for which the gear passes through neutral during the gearshift process, thus simulating a realistic gearshifting behavior. If the parameter is zero, this feature is inactive.

Because the gear passes through neutral only when the clutch is completely open, the Const_t_Gear_Neutral parameter should always be less than or equal the time the clutch pedal remains completely depressed, i.e., less then or equal to the Const_t_Hold_ClutchPedal parameter. If this condition is not fulfilled, the minimum time of both parameters is used. The following illustration shows the feature of passing the gear through neutral during the gearshift process for both parameters.

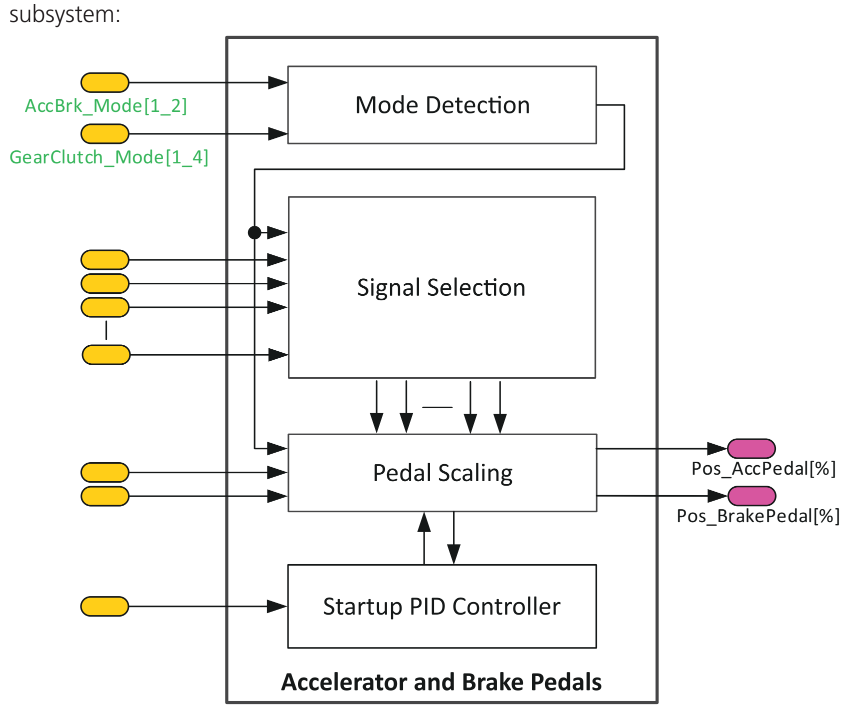


You can also force the driver to select a certain gear value at any time. This is done via the GearClutch_Mode[1Stim|2Driver|3OpenClutch|4RefGear] and Gear_Maneuver[] inports.

- If GearClutch_Mode is set to 1, the Gear_Maneuver value is used as the current gear.
- If GearClutch_Mode is set to 2, the calculated gear is used.

Accelerator_And_Brake_Pedals subsystem

The following illustration shows the Accelerator_And_Brake_Pedals subsystem:



The Accelerator_And_Brake_Pedals subsystem implements the startup functionality and the contribution of the gearshift to the accelerator and brake pedals. A PID controller is used to modify the accelerator pedal position to reach a user-defined engine speed (Const_n_Engine_StartUp_Set parameter) during vehicle startup.

The startup controller is activated during startup only if at least one of the following conditions is fulfilled:

- There is high tractive resistance (`Big_Tractive_Resistance_Acceleration` condition is true)
- The maneuver is defined with initial velocity (`Initial_Velocity` condition is true)
- High acceleration is needed to follow the reference vehicle velocity (`Big_Acceleration` condition is true)

The brake pedal position is disabled if the startup controller is active.

Because this block ensures a correct startup, you might need to change the defined engine speed during startup if, for example, the vehicle is to start on a steep hill.

The accelerator pedal position is scaled during clutch actuation. For the scaling, a user-defined look-up table is used as a function of the clutch pedal position (Map_Pos_AccPedal parameter).

If the accelerator and brake pedals are stimulated via external signals, they are routed out without intervention.

Related topics

References

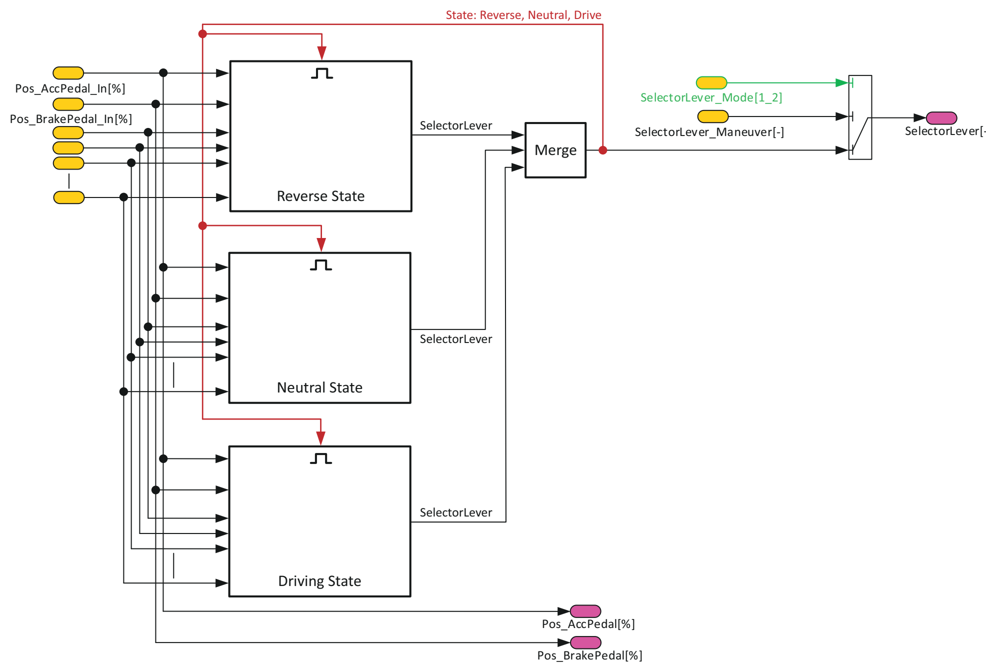
Gear Shifter.. 14
Gear Shifter - Automatic Transmission Control.. 27

Gear Shifter - Automatic Transmission Control

Description

In the Control_AutomaticTransmission subsystem, the selector lever is simulated for automatic transmission. The following illustration shows the structure of the block.

The Control_AutomaticTransmission subsystem is part of the Gear_Shifter block. Refer to Gear Shifter on page 14.

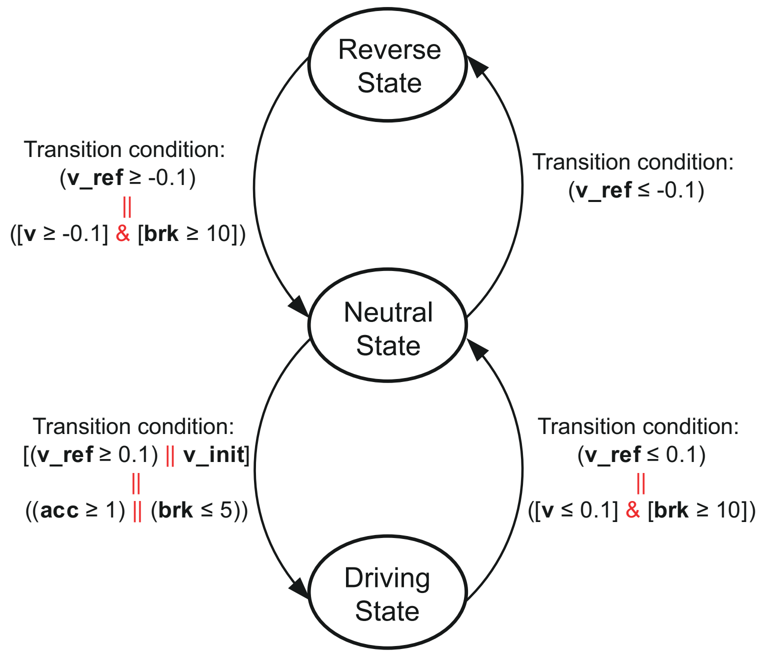


A state machine is implemented to continuously check the driving situation and set selector lever to the right position. Only one state can be active at a time. The transition from one state to another is conditionally driven and only occurs in a rational manner. The selector lever can also be stimulated via external signals, i.e., gear or selector lever. The stimulated gear can force the selector lever to change position. If the selector lever is externally stimulated, the calculated value is overwritten.

The following illustration displays the simulated states as well as the transitions between them:

Where,

||	: logical OR
&	: logical AND
~	: logical NOT
v	: vehicle speed [m|s]
v_ref	: reference vehicle speed [m|s]
brk	: brake pedal position [%]
acc	: accelerator pedal position [%]
v_init	: initial velocity condition [0|1]



While the selector lever is set in this block, the shift itself is performed by the SOFT_ECU_TRANSMISSION block.

Related topics

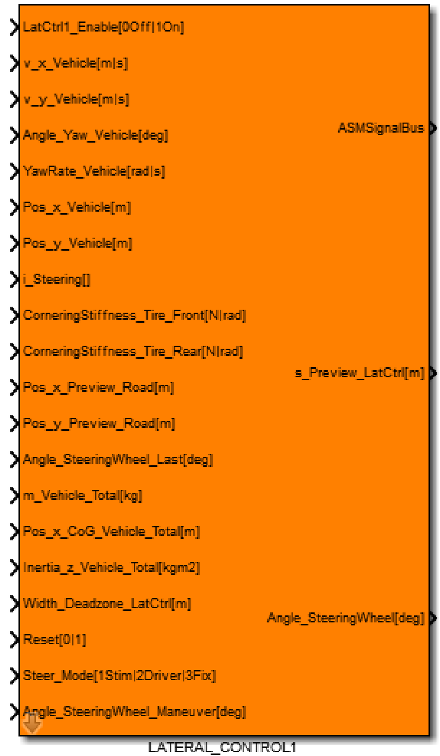
References

Gear Shifter.. 14
Gear Shifter - Manual Transmission Control.. 18

Lateral Controller 1

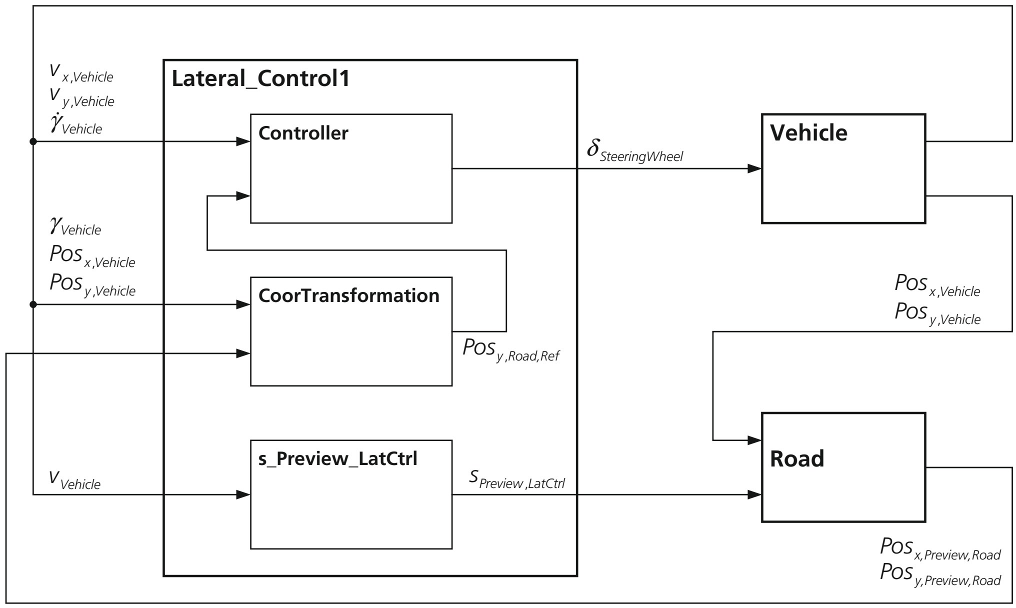
Description

The Lateral_Control1 subsystem keeps the vehicle on the road by controlling the steering wheel. This lateral controller provides optimal results for driving on roads. The controller concept was proposed by [MacA80] and [MacA81] and is based on linear optimal control theory. The controller uses preview information and contains a linear single-track model.



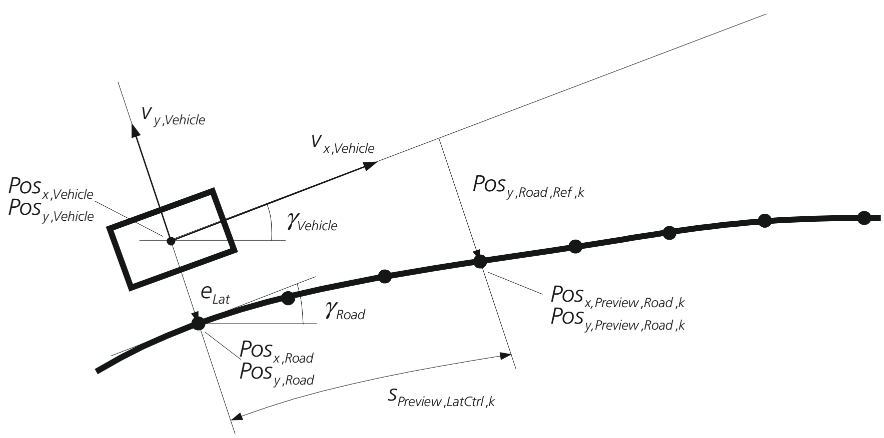
The control algorithm described below calculates the steering angle at the wheels. This is converted to steering wheel angle by means of the steering ratio. This value is provided as a signal from the vehicle dynamics model. The maximum steering wheel angle is provided by the common driver parameters block. The steering wheel angle signal is delayed by a first-order delay.

The following illustration shows the main signal flow inside the subsystem and the interconnection to the road and vehicle models.



Preview

The controller uses preview information for the control law. The following illustration is a schematic of the preview concept.



The preview distance is calculated as the distance the vehicle moves in the preview time $t_{Preview}$. The preview time is set to 1 s.

$$s_{Preview,LatCtrl} = v_{Vehicle} t_{Preview}$$

For low vehicle velocities, a minimum preview distance is used. There are ten preview points evenly spaced along the preview distance.

$$s_{Preview,LatCtrl,k} = 0.1\,k\,s_{Preview,LatCtrl} \qquad k = [1...10]$$

Reference position

The road block provides x and y coordinates for the preview points in earth coordinates according to the preview distance,

$$\begin{bmatrix} Pos_{x,Preview,Road,k} \\ Pos_{y,Preview,Road,k} \end{bmatrix} = Road\_LatData\left(s_{current} + s_{Preview,LatCtrl,k}\right)$$

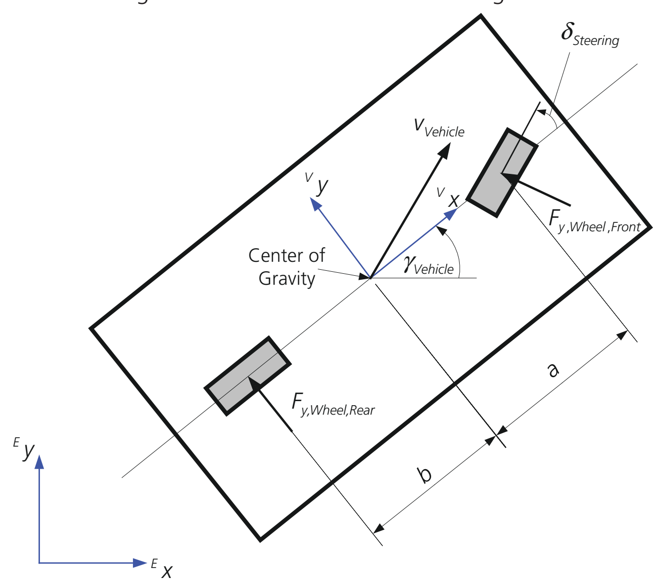
The controller uses the y-component of the reference positions in vehicle coordinates.

$$Pos_{y,Road,Ref,k} = \left(Pos_{y,Preview,Road,k} - Pos_{y,Vehicle}\right)\cos\gamma_{Vehicle} - \left(Pos_{x,Preview,Road,k} - Pos_{x,Vehicle}\right)\sin\gamma_{Vehicle}$$

Linear single-track model

The controller algorithm is based on a linear single-track model (bicycle model) in state space form.

The following illustration is a schematic of a single-track model.



The components of vehicle velocity $V_{x,Vehicle}$ and $V_{y,Vehicle}$, and the vehicle position $Pos_{y,Vehicle}$ are described in vehicle coordinates in this subchapter. The value of $V_{x,Vehicle}$ is treated as fixed in the following calculations.

The sideslip angles of the wheels are given by

$$\alpha_{Wheel,Front} = \delta_{Steering} - \arctan\left(\frac{V_{y,Vehicle} + a\dot{\gamma}_{Vehicle}}{V_{x,Vehicle}}\right)$$

The tire forces are assumed to be linear.

$$\alpha_{Wheel,Rear} = \arctan\left(\frac{-\left(V_{y,Vehicle} - b\dot{\gamma}_{Vehicle}\right)}{V_{x,Vehicle}}\right)$$

$$F_{y,Wheel,Front} = 2\,C_F\,\alpha_{Wheel,Front}$$

$F_{y,Wheel,Rear} = 2\,C_R\,\alpha_{Wheel,Rear}$

where:
C_F is the cornering stiffness for front wheels
C_R is the cornering stiffness for rear wheels

The balance of forces and torques of the single track vehicle is

$$m_{Vehicle}\dot{V}_{y,Vehicle} = F_{y,Wheel,Front}\cos\delta_{Steering} + F_{y,Wheel,Rear} - V_{x,Vehicle}\dot{\gamma}_{Vehicle}m_{Vehicle}$$

$$J_{Vehicle}\ddot{\gamma}_{Vehicle} = aF_{y,Wheel,Front}\cos\delta_{Steering} - bF_{y,Wheel,Rear}$$

After linearization, the model is written in state space form.

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

where:

$x = \left[Pos_{y,Vehicle}\ V_{y,Vehicle}\ \dot{\gamma}_{Vehicle}\ \gamma_{Vehicle}\right]^T$ is the state vector
$y = Pos_{y,Vehicle}$ is the output signal

The linearized model equations are

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & V_{x,Vehicle} \\ 0 & \dfrac{-2(C_F + C_R)}{m_{Vehicle}V_{x,Vehicle}} & \dfrac{2(bC_R - aC_F)}{m_{Vehicle}V_{x,Vehicle}} - V_{x,Vehicle} & 0 \\ 0 & \dfrac{2(bC_R - aC_F)}{J_{Vehicle}V_{x,Vehicle}} & \dfrac{-2\left(a^2C_F + b^2C_R\right)}{J_{Vehicle}V_{x,Vehicle}} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \dfrac{2C_F}{m_{Vehicle}} \\ \dfrac{2aC_F}{J_{Vehicle}} \\ 0 \end{bmatrix} \delta_{Steering}$$

$$y = \ [1\ 0\ 0\ 0]x$$

Control Law

The solution to any linear system

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

at discrete time points kT is known to be

$$y(kT) = C\Phi(kT)\,x(0) + C\Gamma(kT)Bu(0)$$

where

$$\Phi(kT) = e^{AkT}\ and\ \Gamma(kT) = \int_0^{kT} \Phi(r)\,dr$$

[MacA80] proposed a controller which minimizes the following objective function

$$J = \sum_{k=1}^{m} \left(y_{Ref}(kT) - y\left(kT\right)\right)^2 W_k$$

The weighted difference between output signal and reference signal is summed at a number of m preview points.

The solution to the linear system equations is inserted in the objective function.

$$J = \sum_{k=1}^{m} \left(y_{Ref}(kT) - C\Phi(\mathrm{kT})x(0) - \mathrm{C}\Gamma(\mathrm{kT})\mathrm{B}u\big(0\big)\right)^2 W_k$$

After derivation of *J* with respect to $u(0)$

$$\frac{dJ}{du(0)} = 2\sum_{k=1}^{m} \left(y_{Ref}(kT) - C\,\Phi(kT)x(0) - C\Gamma(kT)Bu(0)\right)\left(-C\Gamma(kT)BW_k\right)$$
$$= 0 = \sum_{k=1}^{m} \left(y_{Ref}(kT) - C\,\Phi(kT)x\big(0\big)\right)\left(-C\Gamma(kT)BW_k\right)$$
$$+ \sum_{k=1}^{m} (-C\Gamma(kT)Bu(0))(-C\Gamma(kT)BW_k)$$

the control law can be calculated in closed form

$$u(0) = \frac{\sum_{k=1}^{m} \left(y_{Ref}(kT) - C\Phi(kT)x\big(0\big)\right)\left(C\Gamma(kT)BW_k\right)}{\sum_{k=1}^{m} \left((C\Gamma(kT)B)^2 W_k\right)}$$

In this approach it is assumed that the control input is kept constant over the whole preview interval.

In the implemented controller, a number of m = 10 preview points are used. With a preview time $t_{Preview}$ of 1 s, this results in an internal sample time T = 1 s / 10 = 0.1 s.

The previewed reference positions $y_{Ref}(kT)$ are provided by the road subsystem. The values are transformed to vehicle coordinates as described above.

$$y_{Ref}(kT) = Pos_{y,Road,Ref,k}$$

With $u(0) = \delta_{Steering}$the control law is

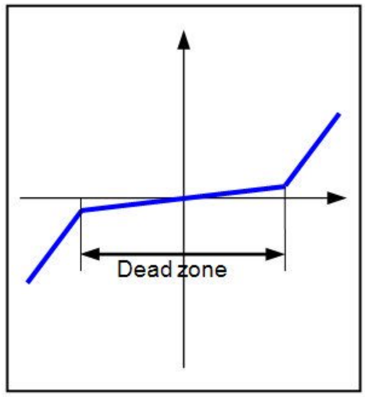
$$\delta_{Steering} = \frac{\sum_{k=1}^{10} \left(Pos_{y,Road,Ref,k} - C\Phi(kT)x\big(0\big)\right)\left(C\Gamma(kT)BW_k\right)}{\sum_{k=1}^{10} \left((C\Gamma(kT)B)^2 W_k\right)}$$

The weights W_k are set to unity.

Tolerance zone

The lateral controller uses road preview information and calculates the optimal steering angle based on the position control. The controller will always have a forceful influence to the steering as long as the vehicle does not drive on the given trajectory. If active lane assist systems are tested, this behavior conflicts with the *Lane Keeping Actuator* since the lane keeping actuator will try to keep the vehicle at the lane center line.

To avoid this, an additional lateral controller tolerance zone (`Width_Deadzone_LatCtrl` parameter) is included. Within this tolerance zone, the lateral error curve (see the following illustration) has a small slope which means that the lateral controller has only a slight influence to the steering.



The width of the tolerance zone can be defined in ModelDesk's Scenario Editor in Follow Road segments. Refer to Lateral Type Properties (Maneuver) (ModelDesk Scenario Creation 📖).

Implementation

The control law was implemented in an S-function and optimized for real-time simulation.

Inports

The following table shows the inports:

Name	Unit	Description
Angle_SteeringWheel_Last	[deg]	Last steering wheel angle (for smooth signal transitions)
Angle_SteeringWheel_Maneuver	[deg]	Steering wheel angle stimulus from maneuver scheduler
Angle_Yaw_Vehicle	[deg]	Vehicle yaw angle
CorneringStiffness_Tire_Front	[N/rad]	Front tire cornering stiffness
CorneringStiffness_Tire_Rear	[N/rad]	Rear tire cornering stiffness
i_Steering	[]	Steering ratio (angle of wheel to steering wheel angle)
Inertia_z_Vehicle_Total	[kgm^2]	Element (3.3) of total inertia tensor of vehicle body
LatCtrl1_Enable	[0|1]	Enable switch for calculation of steering angle ▪ 0: Off ▪ 1: On
m_Vehicle_Total	[kg]	Total vehicle mass (with additional loads)
Pos_x_CoG_Vehicle_Total	[m]	x-element of position vector of vehicle center of gravity
Pos_x_Preview_Road	[m]	Vector of previewed road positions in x direction (earth coordinates)
Pos_x_Vehicle	[m]	Vehicle position in x direction in earth coordinates
Pos_y_Preview_Road	[m]	Vector of previewed road positions in y direction (earth coordinates)
Pos_y_Vehicle	[m]	Vehicle position in y direction in earth coordinates
Reset	[0|1]	Reset of all integrators and internal states
Steer_Mode	[1|2|3]	Mode signal for steering wheel angle: ▪ 1: Stimulus ▪ 2: Driver

Name	Unit	Description
		▪ 3: Fix
v_x_Vehicle	[m/s]	Vehicle velocity (x direction of vehicle coordinate system)
v_y_Vehicle	[m/s]	Vehicle velocity (y direction of vehicle coordinate system)
Width_Deadzone_LatCtrl	[m]	Defines the width of a tolerance zone around the reference trajectory. Within this tolerance zone, the lateral controller ignores deviations from the reference trajectory. This is used for tests of lane assist systems. The value is set by the maneuver and can be defined in 'Follow Road' segments.
YawRate_Vehicle	[rad/s]	Vehicle yaw rate

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 📖).
Angle_SteeringWheel	[deg]	Steering wheel angle
s_Preview_LatCtrl	[m]	Preview distance for lateral control (Vector)

Parameters

The following table shows the block parameters:

Name	Unit	Description
Const_t_Preview_LatCtrl	[s]	Preview time for lateral controller
Const_t_PT1_LatCtrl	[s]	Delay time for lateral control signal (PT1)
Const_WheelBase	[m]	Wheel base
StepSize	[s]	Step size of model

Note

The steering wheel angle parameter is provided by the common driver parameters subsystem, see Common Driver Parameters on page 13.

Related topics

HowTos

How to Shift Gears in a Maneuver (ModelDesk Scenario Creation 📖)

References

Lateral Control 1 (ModelDesk Parameterizing 📖)
Lateral Type Properties (Maneuver) (ModelDesk Scenario Creation 📖)

Lateral Controller 2

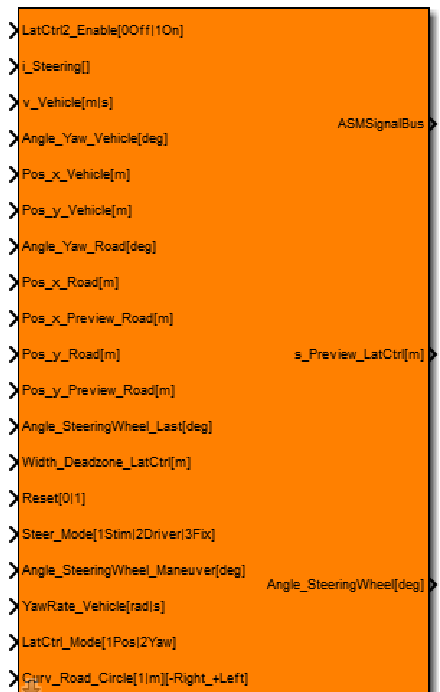
Description

The Lateral_Control2 subsystem keeps the vehicle on the road by controlling the steering wheel. This controller is recommended for stationary driving situations such as driving in a circle at a given lateral acceleration. The concept of this controller was proposed by [Sha00].

There are two different control modes to select from:

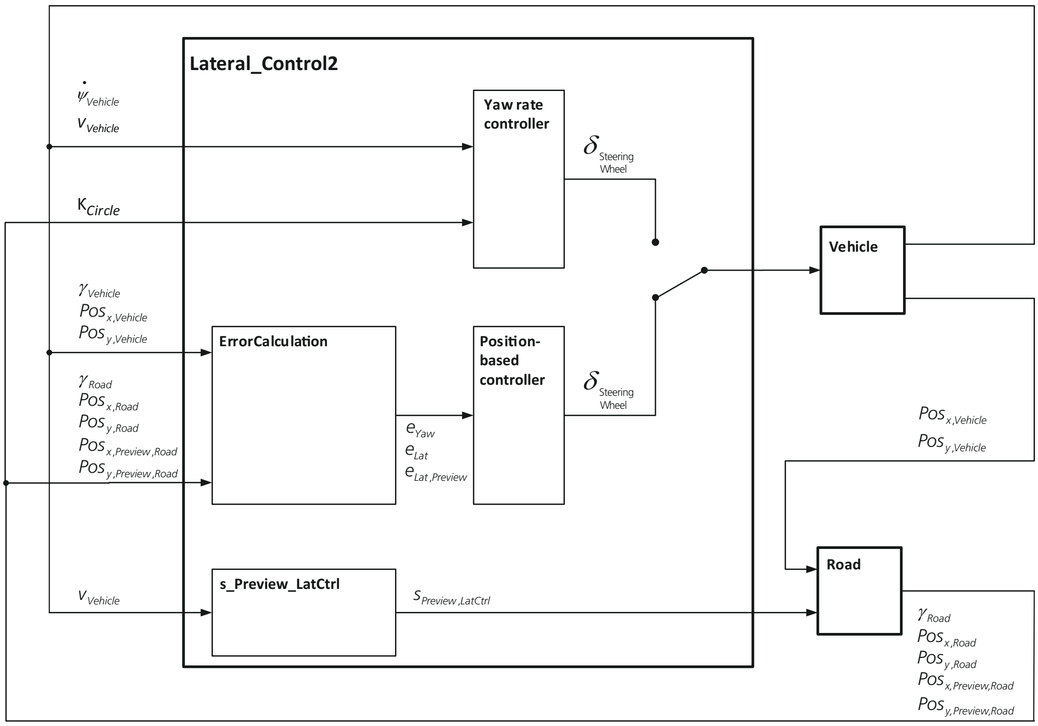
1. Position-based control with preview. This is suitable for all maneuvers without road (i.e. with basic road).
2. Yaw rate control. This control is without preview and is intended only for the circle lateral type, in particular for steady-state maneuvers.

An integral component was added to the concept proposed by [Sha00] to prevent a control offset between vehicle and road position.



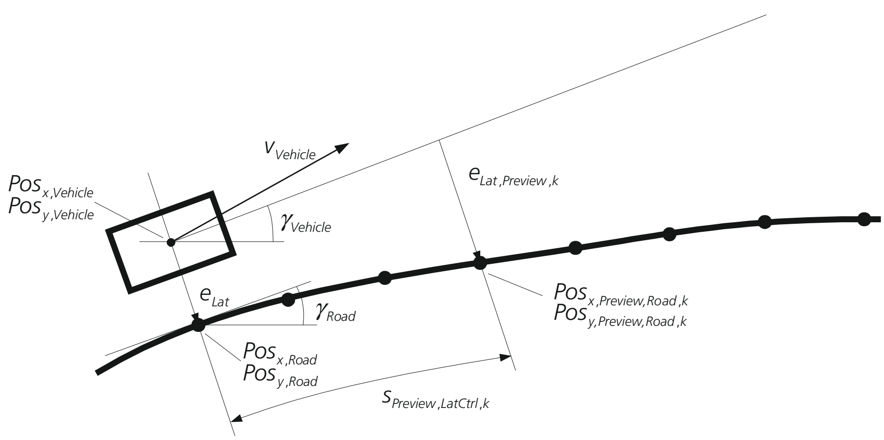
The control algorithm described below calculates the steering angle at the wheels. This is converted to steering wheel angle by means of the steering ratio. This value is provided as a signal from the vehicle dynamics model. The maximum steering wheel angle is provided by the common driver parameters block. The steering wheel angle signal is delayed by a first-order delay.

The following illustration shows the main signal flow inside the block and the interconnection to the road and vehicle models.



Preview

If the position-based control is activated, the controller uses preview information for the control law. The following illustration is a schematic of the preview concept.



The preview distance is calculated as the distance the vehicle moves in the preview time $t_{Preview}$. The preview time is set to 1 s.

$$s_{Preview,\, LatCtrl} = v_{Vehicle} t_{Preview}$$

There are seven non equally spaced preview points along the preview interval. The position of each preview point in the preview interval is defined by a vector.

$$s_{Preview,\,LatCtrl,\,Spacing} = [0.1, 0.2, 0.3, 0.4, 0.6, 0.8, 1.0]$$

$$s_{Preview,\,LatCtrl,\,k} = s_{Preview,\,LatCtrl,\,Spacing}(k)\, s_{Preview,\,LatCtrl} \qquad k = [1 \ldots 7]$$

The road block provides x and y coordinates and the direction of the current reference position in earth coordinates.

$$\begin{bmatrix} Pos_{x,\,Road} \\ Pos_{y,\,Road} \\ \gamma_{Road} \end{bmatrix} = Road\_LatData\,(s_{current} + 0)$$

x and y coordinates are provided for the preview points.

$$\begin{bmatrix} Pos_{x,\,Preview,\,Road,\,k} \\ Pos_{y,\,Preview,\,Road,\,k} \end{bmatrix} = Road\_LatData(s_{current} + s_{Preview,\,LatCtrl,\,k})$$

ErrorCalculation for position-based control

The lateral controller algorithm uses position error signals for current and previewed vehicle positions and the current yaw angle error.

The position error corresponds to the y coordinate of the road positions in the vehicle coordinate system.

$$e_{Lat} = (Pos_{y,\,Road} - Pos_{y,\,Vehicle})\cos\gamma_{Vehicle} - (Pos_{x,\,Road} - Pos_{x,\,Vehicle})\sin\gamma_{Vehicle}$$

The previewed position errors are calculated by

$$e_{Lat,\,Preview,\,k} = (Pos_{y,\,Preview,\,Road,\,k} - Pos_{y,\,Vehicle})\cos\gamma_{Vehicle}$$

$$-(Pos_{x,\,Preview,\,Road,\,k} - Pos_{x,\,Vehicle})\sin\gamma_{Vehicle}$$

The yaw angle error is calculated by

$$e_{Yaw} = (\gamma_{Road} - \gamma_{Vehicle})$$

Control Law for position-based control

The control law

$$\delta_{Steering} = k_{Yaw}e_{Yaw} + k_{Lat}e_{Lat} + \sum_{k=1}^{7} k_{Lat,\,Preview,\,k}\,e_{Lat,\,Preview,\,k}$$

where

$$K_{Lat,\,Preview,\,k} = K_{Lat,\,Preview}K_{Lat,\,Preview,\,Weight}(k)$$

$$K_{Lat,\,Preview,\,Weight} = [1.0\;\; 0.6\;\; 0.2\;\; 0.08\;\; 0.016\;\; 0.004\;\; 0.001]$$

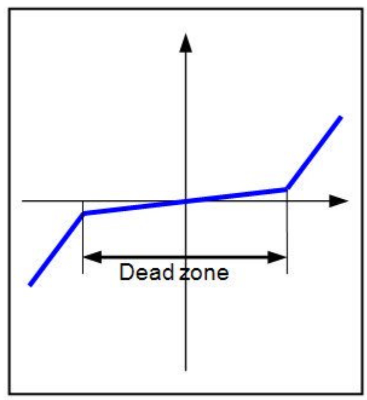
was proposed by [Sha00]. To improve road following in steady state maneuvers, dSPACE added an integral component to the controller.

$$\delta_{Steering} = k_{Yaw}e_{Yaw} + k_{Lat}e_{Lat} + k_{Lat,\,i}\int e_{Lat}\,dt + \sum_{k=1}^{7} k_{Lat,\,Preview,\,k}\,e_{Lat,\,Preview,\,k}$$

Tolerance zone

The lateral controller uses road preview information and calculates the optimal steering angle based on the position control. The controller will always have a forceful influence to the steering as long as the vehicle does not drive on the given trajectory. If active lane assist systems are tested, this behavior conflicts with the *Lane Keeping Actuator* since the lane keeping actuator will try to keep the vehicle at the lane center line.

To avoid this, an additional lateral controller tolerance zone (`Width_Deadzone_LatCtrl` parameter) is included. Within this tolerance zone, the lateral error curve (see the following illustration) has a small slope which means that the lateral controller has only a slight influence to the steering.



The width of the tolerance zone can be defined in ModelDesk's Scenario Editor in Follow Road segments. Refer to Lateral Type Properties (Maneuver) (ModelDesk Scenario Creation 📖).

Controller parameters for position-based control

The default parameter settings are chosen such that the controlled system is stable for most maneuvers and vehicles. In some cases it may be necessary to fine-tune the parameters to increase robustness or performance.

You can access the k_{Yaw}, k_{Lat}, $k_{Lat,\,i}$ and $k_{Lat,\,Preview}$ parameters in the subsystem mask. The following rules shall are a guide to handling the controller parameters.

- Increasing k_{Yaw} increases system stability.
- Increasing k_{Lat} increases the performance of the system. If this value is too large, the system may become instable.
- As a rule of thumb, the values of k_{Lat}, $k_{Lat,\,i}$ and $k_{Lat,\,Preview}$ should be in the same range.
- The ratio between k_{Lat} and $k_{Lat,\,Preview}$ describes the weighting of the current position error to the previewed position errors.
- $k_{Lat,\,i}$ is the gain for the integral component of the controller. This value influences the time the controller needs to reach the reference curve. The robustness of the controller decreases as $k_{Lat,\,i}$ increases.

Control law for yaw rate control

If the yaw rate controller is used, the control law includes an anti-windup control:

$$\delta_{Steering} = K_{P,YawRate} \cdot e_{YawRate} + K_{I,YawRate} \cdot \int e_{YawRate} - 0{,}1\left(\delta_{Steering} - \delta_{Steering,Sat}\right)dt$$

where:

$$\delta_{Steering,Sat} = \begin{cases} \delta_{Steering,max} & if\ \delta_{Steering} \geq \delta_{Steering,max} \\ \delta_{Steering} & if\ -\delta_{Steering,max} < \delta_{Steering,max} < \delta_{Steering,max} \\ -\delta_{Steering,max} & if\ \delta_{Steering} \leq -\delta_{Steering,max} \end{cases}$$

Inports

The following table shows the inports:

Name	Unit	Description
Angle_SteeringWheel_Last	[deg]	Last steering wheel angle (for smooth signal transitions)
Angle_SteeringWheel_Maneuver	[deg]	Steering wheel angle stimulus from maneuver scheduler
Angle_Yaw_Road	[deg]	Road yaw angle (tangent to center line)
Angle_Yaw_Vehicle	[deg]	Vehicle yaw angle
Curv_Road_Circle	[1/m]	Road curvature of circular road defined by maneuver
i_Steering	[]	Steering ratio (Angle_Steering --> Angle_SteeringWheel)
LatCtrl_Mode	[1|2]	Defines the control mode: ▪ 1: Position-based control ▪ 2: Yaw rate control
Pos_x_Preview_Road	[m]	Vector of previewed road positions in x direction (earth coordinates)
Pos_x_Road	[m]	Current road position in x direction in earth coordinates
Pos_x_Vehicle	[m]	Vehicle position in x direction in earth coordinates
Pos_y_Preview_Road	[m]	Vector of previewed road positions in y direction (earth coordinates)
Pos_y_Road	[m]	Current road position in y direction in earth coordinates
Pos_y_Vehicle	[m]	Vehicle position in y direction in earth coordinates
Reset	[0|1]	Reset of all integrators and internal states
Steer_Mode	[1|2|3]	Mode signal for steering wheel angle: ▪ 1: Stimulus ▪ 2: Driver ▪ 3: Fix
v_Vehicle	[m/s]	Vehicle velocity
YawRate_Vehicle	[rad/s]	Vehicle yaw rate
Width_Deadzone_LatCtrl	[m]	Defines the width of a tolerance zone around the reference trajectory. Within this tolerance zone, the lateral controller ignores deviations from the reference trajectory. This is used for tests of lane assist systems. The value is set by the maneuver and can be defined in follow road segments.

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 📖).
Angle_SteeringWheel	[deg]	Steering wheel angle
s_Preview_LatCtrl	[m]	Preview distance for lateral control (Vector)

Parameters

The following table shows the block parameters:

Name	Unit	Description
Const_Ki_YawRate_Ctrl	[]	Yaw rate controller gain Ki (integral)
Const_Kp_YawRate_Ctrl	[]	Yaw rate controller gain Kp (proportional)
Const_LatCtrl_kLat	[]	Controller gain k_{Lat} for position error (proportional)
Const_LatCtrl_kLat_i	[]	Controller gain $k_{\mathrm{Lat,i}}$ for position error (integral)
Const_LatCtrl_kLat_Preview	[]	Controller gain $k_{\mathrm{Lat,Preview}}$ for previewed position error
Const_LatCtrl_kYaw	[]	Controller gain k_{Yaw} for yaw angle error
Const_t_Preview_LatCtrl	[s]	Preview time for lateral controller
Const_t_PT1_LatCtrl	[s]	Delay time for lateral control signal (PT1)
StepSize	[s]	Step size of model

Note

The maximum steering wheel angle parameter is provided by the common driver parameters subsystem (see Common Driver Parameters on page 13).

Related topics

HowTos

How to Shift Gears in a Maneuver (ModelDesk Scenario Creation 📖)

References

Lateral Control 2 (ModelDesk Parameterizing 📖)
Lateral Type Properties (Maneuver) (ModelDesk Scenario Creation 📖)

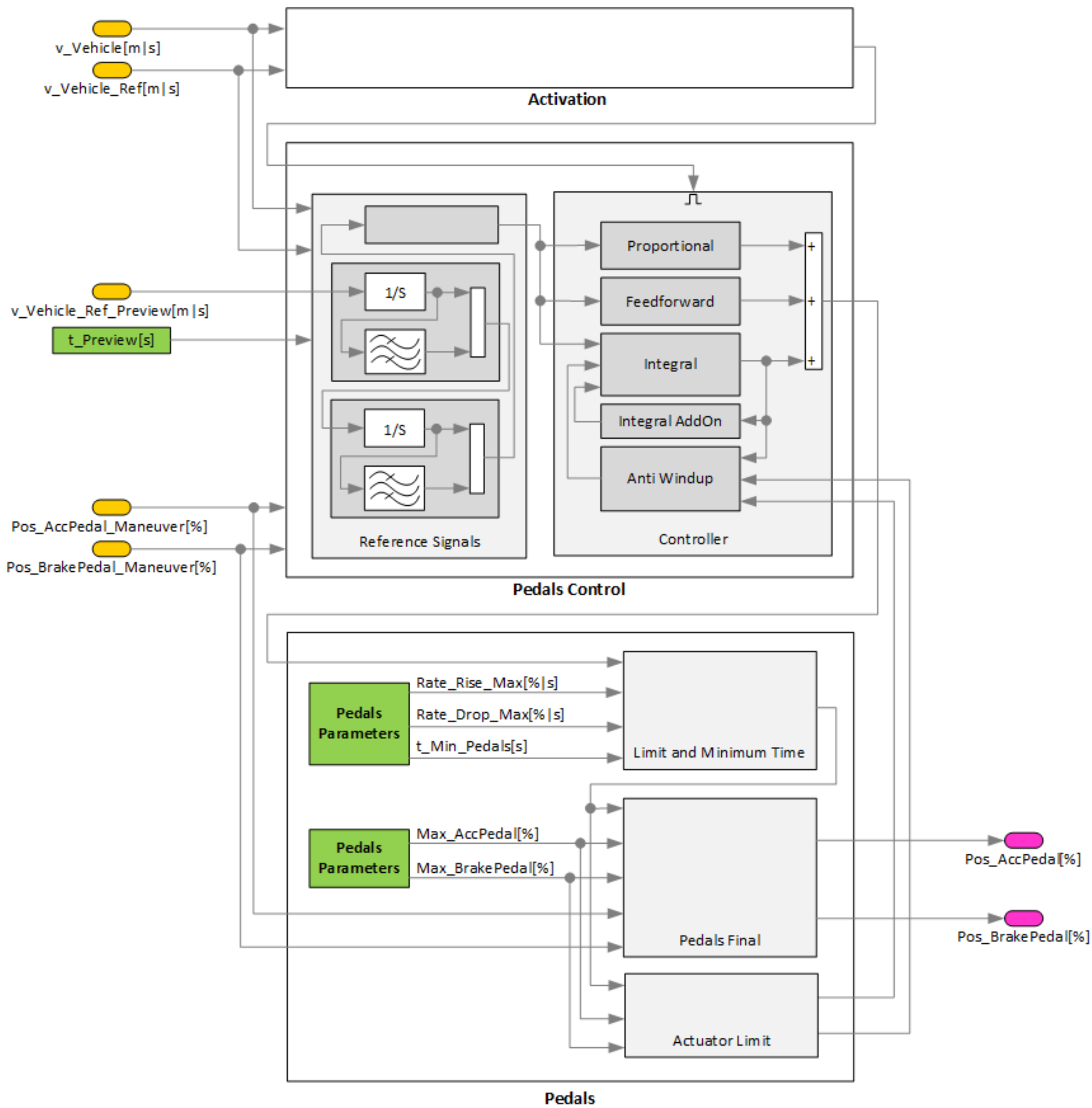
Longitudinal Control

Description

The LONGITUDINAL_CONTROL block controls the accelerator pedal and the brake pedals to follow a specified reference velocity.

The block is mainly modeled as a PI plus feed-forward controller. User parameters, preview information of the reference velocity, and the conditions of the speed course significantly influence the controller.

The following illustration shows the schematic of the block.



The model can simulate realistic pedal behavior. It lets you define parameters, such as the minimum time to switch between different pedals, the maximum and the minimum allowed pedal actuation rate, etc.

Inports

The following table shows the inports:

Name	Unit	Description
Enable_v_Vehicle_Init	[0|1]	Drivetrain initialization flag for scenarios with non-zero vehicle initial velocity
Mode_AcceleratorBrake	[1|2]	Mode signal for accelerator and brake pedal ▪ 1: Stimulus ▪ 2: Driver
Pos_AccPedal_Maneuver	[%]	Accelerator pedal position stimulus from the maneuver scheduler
Pos_BrakePedal_Maneuver	[%]	Brake pedal position stimulus from the maneuver scheduler
Reset	[0|1]	Reset of states
State_Engine	[0|1|2|3|4]	Engine state ▪ 0: Engine off ▪ 1: Ignition on ▪ 2: Ignition on and starter activated ▪ 3: Engine is running ▪ 4: Ignition is switched off, shutdown active
v_Vehicle	[m/s]	Vehicle velocity
v_Vehicle_Ref	[m/s]	Reference vehicle velocity
v_Vehicle_Ref_Preview	[m/s]	Preview reference vehicle velocity

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 📖).
Pos_AccPedal	[%]	Accelerator pedal position
Pos_BrakePedal	[%]	Brake pedal position
t_Preview	[s]	Preview time on reference vehicle velocity

Parameters

The following table shows the parameters:

Name	Unit	Description
Const_Max_AccPedal	[%]	Maximum value of accelerator pedal
Const_Max_BrakePedal	[%]	Maximum value of brake pedal
Const_P_FeedForward_aCtrl	[]	Feed-forward P gain of reference acceleration
Const_Pos_BrakePedal_Standstill	[%]	Brake pedal actuation in standstill
Const_Rate_Drop_Max_AccPedal	[%/s]	Minimum drop rate of accelerator pedal

Name	Unit	Description
Const_Rate_Drop_Max_BrakePedal	[%/s]	Maximum drop rate of brake pedal
Const_Rate_Rise_Max_AccPedal	[%/s]	Maximum rise rate of accelerator pedal
Const_Rate_Rise_Max_BrakePedal	[%/s]	Minimum rise rate of brake pedal
Const_t_Min_Pedals	[s]	Minimum time needed to switch between different pedals
Const_t_Preview	[s]	Reference speed preview time
Map_I_vCtrl	[]	I gain of velocity controller = f(v_Vehicle)
Map_P_vCtrl	[]	P gain of velocity controller = f(v_Vehicle)
StepSize	[s]	Simulation step size

Related topics

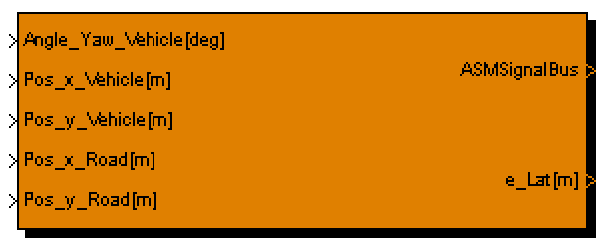
References

Longitudinal Control (ModelDesk Parameterizing)

Position Errors

Description

The POSITION_ERRORS subsystem provides data on the deviation between vehicle position and current road position.



The position error between vehicle position and current road position is calculated by

$$e_{Lat} = \left(Pos_{y,Road} - Pos_{y,Vehicle}\right)\cos\gamma_{Vehicle} - \left(Pos_{x,Road} - Pos_{x,Vehicle}\right)\sin\gamma_{Vehicle}$$

This signal can be interpreted as the y component of the current road position in the vehicle coordinate system. The signal can have positive and negative values.

The absolute value of the e_{Lat} signal is provided in the $e_{Lat,abs}$ signal.

$$e_{Lat,abs} = abs\left(e_{Lat}\right)$$

The total position error is calculated by

$$e_{Lat,total} = \sqrt{\left(Pos_{x,Road} - Pos_{x,Vehicle}\right)^2 + \left(Pos_{y,Road} - Pos_{y,Vehicle}\right)^2}$$

The signals $e_{Lat,abs}$ and $e_{Lat,total}$ are contained in the ASMSignalBus. They are not provided as block outputs.

Inports

The following table shows the inports:

Name	Unit	Description
Angle_Yaw_Vehicle	[deg]	Vehicle yaw angle
Pos_x_Road	[m]	Current road position in x direction in earth coordinates
Pos_x_Vehicle	[m]	Vehicle position in x direction in earth coordinates
Pos_y_Road	[m]	Current road position in y direction in earth coordinates
Pos_y_Vehicle	[m]	Vehicle position in y direction in earth coordinates

Outports

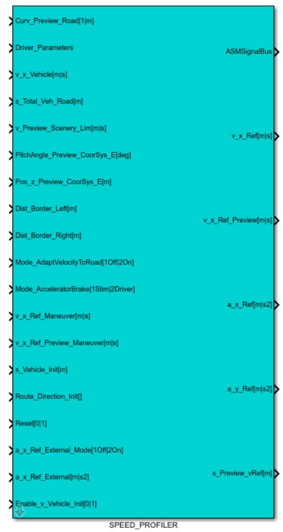
The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 📖).
e_Lat	[m]	Lateral offset between current road position and vehicle position

Speed Profiler

Description

The **SPEED_PROFILER** block lets you generate the driver reference speed in real time as the vehicle is driven on the road. A preview distance is used to collect information about the upcoming road and subsequently find the maximum velocity to cover this distance according to the user-specified driver parameters.



The road information is calculated in the environment model and provided to the driver upon request. The requested preview distance is calculated using the following equation:

$$s_{preview,\,LongCtrl} = \frac{{v_{x,\,veh}}^2}{2 \cdot |a_{x,\,min}|}$$

where:

$v_{x,veh}$ is the longitudinal vehicle velocity

$a_{x,min}$ is the maximum longitudinal driver deceleration, which has a negative sign

The driving style depends on the driver parameters and takes effect when the vehicle is approaching a stop or a turn, during cornering, after leaving a turn, and when accelerating from lower to higher vehicle speeds. At this point, a *g-g diagram* is used in an idealized form to characterize the driver. This adapted form of the *g-g diagram* is written as follows:

$$\left(\frac{a_x}{a_{x,\,max/min}}\right)^{n_{GG}} + \left(\frac{a_y}{a_{y,\,max}}\right)^{n_{GG}} \leq 1$$

where:

a_x is the longitudinal acceleration

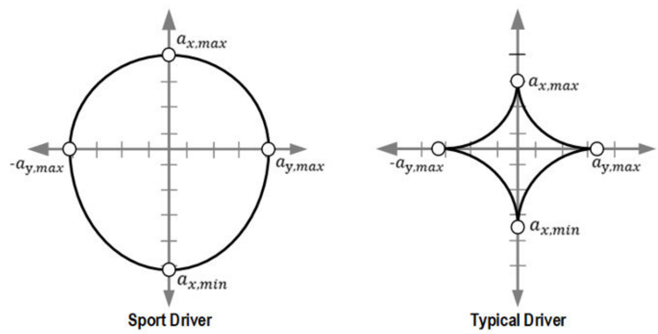
a_y is the lateral acceleration generated by the curved path

$a_{x,max/min}$ is the maximum longitudinal driver acceleration as well as deceleration

$a_{y,max}$ is the maximum lateral driver acceleration

n_{GG} is the *g-g diagram* exponent used to form different driving patterns (performance envelope)

Adapting $a_{x,max/min}$, $a_{y,max}$, and n_{GG} produces different driver characters as shown in the following illustration:



Velocity profile generation starts by determining the velocity limits for road turns (horizontal road profile) and slopes (vertical road profile) ahead. For simplification purposes, the road friction and bank angle are neglected, which leads to the following equation:

$$v_{x,\,lim} = \sqrt{\frac{a_{y,\,max}}{\kappa_y}}$$

where:

a_y is the lateral acceleration generated by the curved path
v_x is the tangential component of the driving velocity
k_y is the road horizontal curvature

The generated reference speed is also adapted according to the vertical road profile according to the following equation:

$$v_{x,\,lim} = \sqrt{\frac{a_{z,\,max/min} \cdot cos(\beta_{road})}{\kappa_z}}$$

where:

$a_{z,max/min}$ is the maximum driver deflection and rebound vertical acceleration
k_z is the vertical road curvature
β_{road} is the road pitch angle at the preview position

From the above equations, the following differential equation is formulated:

$$dv_{x,ref}\left(s\right) = \pm\frac{a_{x,\,max/min}}{v_{x,ref}\left(s\right)} \cdot \sqrt[n_{GG}]{1 - \left(\frac{v_{x,ref}\left(s\right)^2 \cdot \kappa_y\left(s\right)}{a_{y,\,max}}\right)^{n_{GG}}} \cdot ds$$

The reference speed is then found by integrating the last equation numerically:

$$v_{x,ref}\left(s_n + 1\right) \approx v_{x,ref}\left(s_n\right) \pm \frac{a_{x,\,max/min}}{v_{x,ref}\left(s_n\right)} \cdot \sqrt[n_{GG}]{1 - \left(\frac{v_{x,ref}\left(s\right)^2 \cdot \kappa_y\left(s\right)}{a_{y,\,max}}\right)^{n_{GG}}} \cdot \Delta s$$

Driver parameters

The following table shows the driver parameters:

Name	Unit	Description
a_x_Max	[m/s^2]	Maximum allowed longitudinal acceleration.
a_x_Min	[m/s^2]	Maximum allowed longitudinal deceleration.
a_y_Max	[m/s^2]	Maximum allowed lateral acceleration.
a_z_Max_Deflect	[m/s^2]	Maximum allowed vertical acceleration in the positive direction.
a_z_Max_Rebound	[m/s^2]	Maximum allowed vertical acceleration in the negative direction.
Dist_OutsideRoad_Margin	[m]	Margin distance from the road border if the vehicle is no longer on the road.
n_GG_Acceleration	[]	Acceleration exponent of the driver performance envelope.
n_GG_Deceleration	[]	Deceleration exponent of the driver performance envelope.
Sw_Observe_v_Scenery_Lim	[1|2]	Switch to observe the speed limit defined in the scenery: ▪ 1: Off ▪ 2: On
t_Preview_vRef	[s]	Preview time of the driver on the reference speed.
v_x_Ref_OutsideRoad	[m/s]	Reference speed when the vehicle is driving outside the road.

Modes

There are five different modes for the SPEED_PROFILER block.

Adapting the velocity to curves To activate this mode, set the AdaptVelocityToRoad_Mode inport to *On*. In this mode, the driver adapts the vehicle velocity to the curves using the following parameters:

Name	Description
a_x_Max	Describes the maximum acceleration capability after exiting curves and during vehicle launch.
a_x_Min	Describes the maximum braking capability before entering curves and when braking to a standstill.
a_y_Max	Describes the maximum turning capability.
n_GG_Acceleration	Describes the acceleration exponent of the driver performance envelope. This parameter shapes the velocity development when the driver turns and accelerates, e.g., while exiting road curves.

Name	Description
n_GG_Deceleration	Describes the deceleration exponent of the driver performance envelope. This parameter shapes the velocity development when the driver turns and brakes, e.g., while entering road curves.

Speed reduction outside the road In this mode, the driver detects when the vehicle leaves the road and reduces the velocity to keep the vehicle under control. The Dist_OutsideRoad_Margin parameter is used to adjust the reference speed when the vehicle is driving outside the road. If the vehicle leaves the road by more than Dist_OutsideRoad_Margin, the driver uses v_x_Ref_OutsideRoad as the reference speed.

Observation of speed limit in scenery If the Sw_Observe_v_Scenery_Lim parameter is set to *On*, the driver observes the speed limit determined in the scenery which is set by the maneuver. In this case, the driver accelerates and decelerates so that these limits are not exceeded.

External acceleration request If the a_x_Ref_External_Mode input is set to *On*, the vehicle reference speed is additionally adjusted according to the requested a_x_Ref_External input.

Adjusting the velocity when driving uphill and downhill If the AdaptVelocityToRoad_Mode parameter is set to *On*, the driver reduces the vehicle speed if an uphill or downhill slope is ahead to keep as much road contact as possible. For this purpose, the driver uses the a_z_Max_Deflect and a_z_Max_Rebound parameters. In the simulation, the driver uses these parameters only as a reference and they can be exceeded according to the road profile. To deactivate this feature, set both parameters to high values.

Note

The SPEED_PROFILER block provides reasonable reference signals only if the vehicle velocity is close to the reference velocity. A reasonable parameterization of the longitudinal controller is therefore required.

Inports

The following table shows the inports:

Name	Unit	Description
a_x_Ref_External	[m/s^2]	Externally requested reference acceleration.
a_x_Ref_External_Mode	[1|2]	Switch for externally requested acceleration mode: ▪ 1: Off ▪ 2: On
AccBr_Mode	[1|2]	Mode signal for accelerator and brake pedal: ▪ 1: Stimulus ▪ 2: Driver

Name	Unit	Description
AdaptVelocityToRoad_Mode	[1|2]	Switch for velocity adaptation: ▪ 1: Off ▪ 2: On
Curv_Preview_Road	[1/m]	Road curvature at previewed road positions.
Dist_Border_Left	[m]	Distance from the vehicle reference coordinate system to the left road border. A negative value means the vehicle has left the road at the left side.
Dist_Border_Right	[m]	Distance from the vehicle reference coordinate system to the right road border. A negative value means the vehicle has left the road at the right side.
Driver_Parameters	[]	Vector signal containing the driver parameters.
Enable_v_Vehicle_Init	[0|1]	Drivetrain initialization flag for scenarios with non-zero vehicle initial velocity.
PitchAngle_Preview_CoorSys_E	[deg]	Road pitch angle at the preview point.
Pos_z_Preview_CoorSys_E	[m]	Road z-position at the preview point.
Reset	[0|1]	Reset of states.
Route_Direction_Init	[]	Direction on the route for all fellow vehicles.
s_Total_Veh_Road	[m]	Total driven distance *s* along the route at the position of the vehicle.
s_Vehicle_Init	[m]	Initial distance *s* along the road.
v_Preview_Scenery_Lim	[m/s]	Speed limit in the scenery at the preview point.
v_x_Ref_Maneuver	[m/s]	Scenario reference speed provided by the maneuver.
v_x_Ref_Preview_Maneuver	[m/s]	Preview of scenario reference speed provided by the maneuver.
v_x_Vehicle	[m/s]	Vehicle velocity in the forward direction.

Outports

The following table shows the outports:

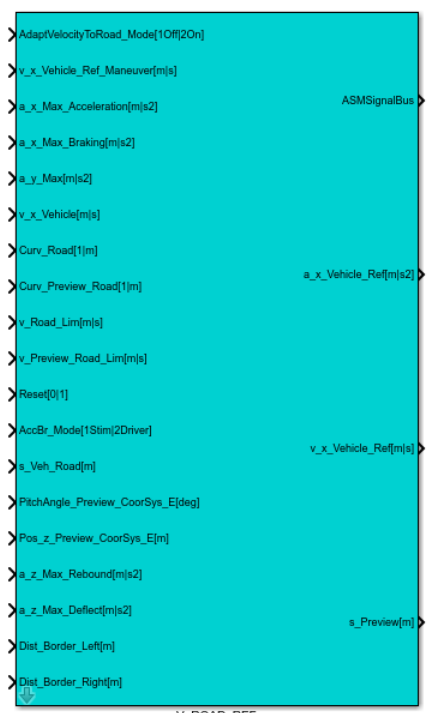
Name	Unit	Description
a_x_Ref	[m/s^2]	Vehicle reference acceleration in the longitudinal direction.

Name	Unit	Description
a_y_Ref	[m/s^2]	Vehicle reference acceleration in the lateral direction (only positive values).
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 📖).
s_Preview_vRef	[m]	Vector of preview distance *s* points for reference velocity calculation.
v_x_Ref	[m/s]	Vehicle reference speed in longitudinal direction.
v_x_Ref_Preview	[m/s]	Preview on the vehicle reference speed in longitudinal direction.

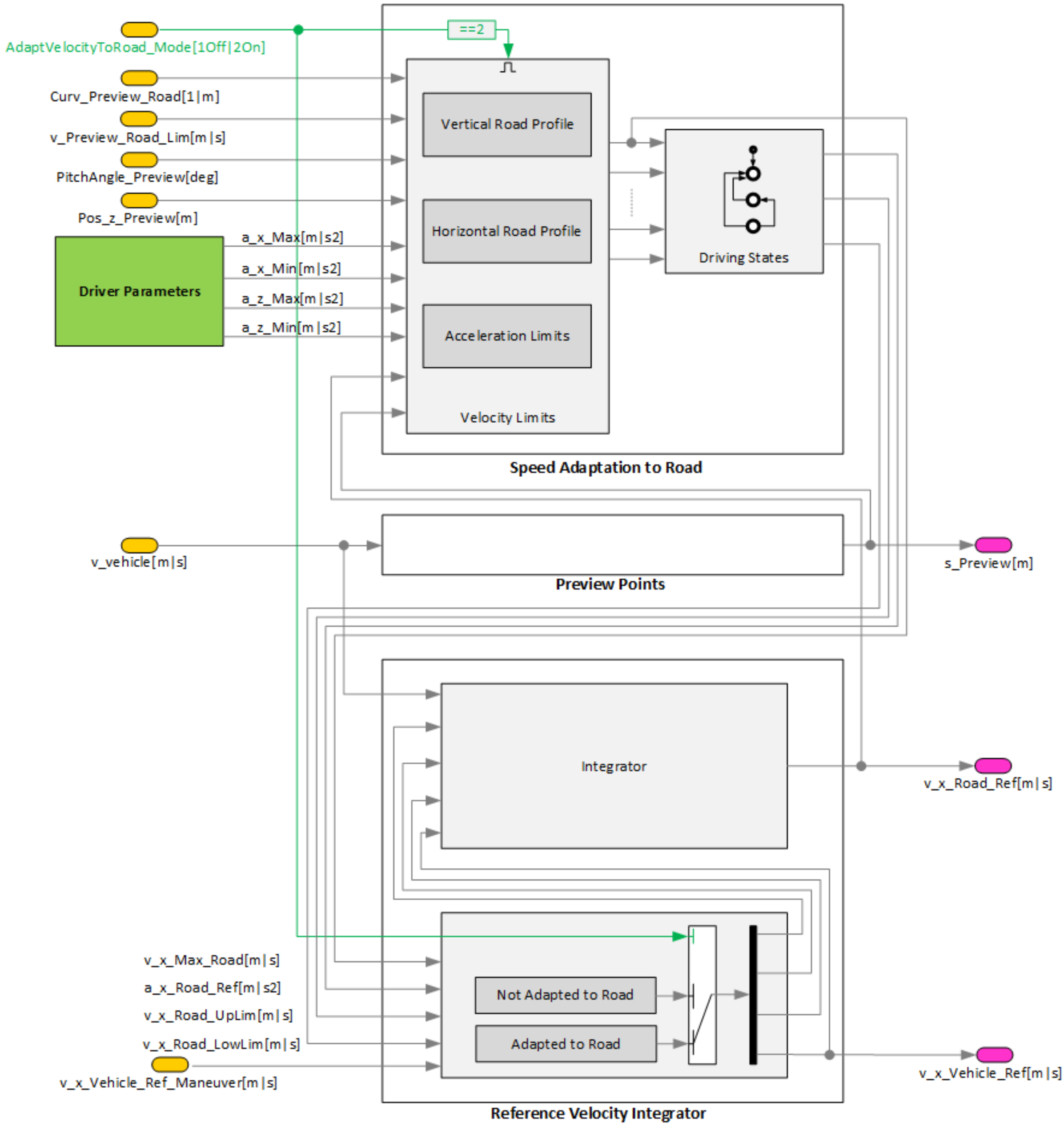
Road Reference Velocity

Description

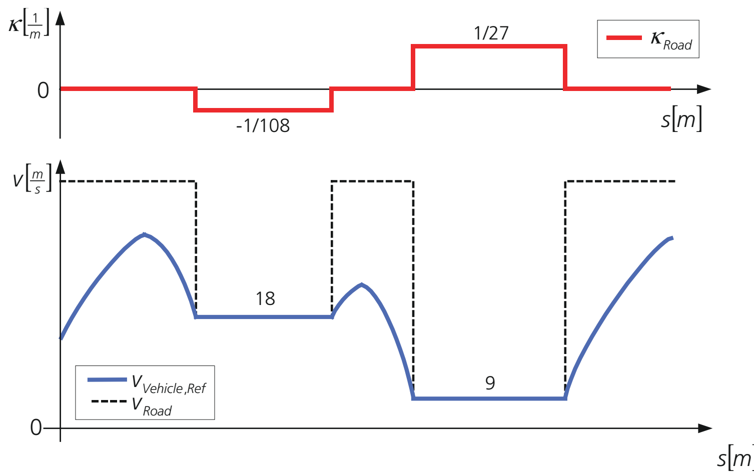
The V_ROAD_REF block provides the reference velocity for driving on roads. The reference velocity provided by the maneuver is automatically adapted to the road according to the defined driver characteristics. These parameters are set by the maneuver. The reference velocity is valid only as long as the vehicle is on the road. Once the vehicle leaves the road, the V_ROAD_REF block assumes a relatively small reference velocity. This assumption helps the driver find the road reference line when the vehicle has slipped off the road. The concept for reference velocity calculation was inspired by [Maj97] and [Kie05].



The following illustration shows the main signal flow inside the V_ROAD_REF block.



The reference velocity provided by the maneuver is adapted according to the horizontal and vertical road profiles. For example, the following illustration shows the adaptation according to the horizontal road profile:



The upper subplot shows the road curvature. At the beginning, there is a straight segment, which is followed by a circular segment with a radius of 108 m. This is followed by a straight segment, a circular segment with radius 27 m, and a straight segment. The maximum permitted vehicle velocity on the road V_{Road} is calculated according to road curvature and maximum permitted lateral acceleration. As can be seen in the second subplot, V_{Road} is not continuous. Using preview information, the V_ROAD_REF block provides a smooth vehicle reference velocity. The vehicle velocity is prevented from exceeding the velocity that is permitted for the road. The subsystem functionality is described in more detail below.

Modes

The v_Road_Ref block provides the following modes that can be set by the maneuver with the `AdaptVelocityToRoad_Mode` mode signal.

- On: The reference velocity provided by the maneuver is adapted to the road. Cornering or driving uphill or downhill too fast is prevented.
- Off: The reference velocity provided by the maneuver is fed through. The signal is not modified.

For driving on roads it is recommended to provide a constant reference velocity via the maneuver and set the `AdaptVelocityToRoad_Mode` to On.

Example The vehicle must move along a road at a velocity of 90 km/h on a road. The reference velocity provided by the maneuver is set to a constant value of 90 km/h and the `AdaptVelocityToRoad_Mode` is set to On. The reference velocity provided by the maneuver is automatically decreased before and in curves.

Maneuver signals

The block does not contain mask parameters. The following driver characteristics parameters are provided by the maneuver via signal lines.

- a_x_Max_Acceleration is the maximum allowed longitudinal acceleration, i.e., it describes the acceleration capability after exiting curves and during vehicle launch.

- a_x_Max_Braking is the maximum allowed longitudinal deceleration, i.e., it describes the braking capability before entering curves and when braking to a standstill. This value is also used for the preview s-distance calculation.
- a_y_Max is the maximum allowed lateral acceleration, i.e., it describes the curve turning capability.
- a_z_Max_Deflect is the maximum allowed vertical acceleration in the positive direction, i.e., it describes the braking capability while driving uphill or downhill.
- a_z_Max_Rebound is the maximum allowed vertical acceleration in the negative direction, i.e., it describes the braking capability while driving uphill or downhill.

Preview points

For reference velocity calculation, the V_ROAD_REF block uses information on the road curvature at certain preview points. The preview distance is chosen such that it is possible to stop the car completely within the preview distance using a defined braking acceleration. There are ten preview points evenly spaced along the preview distance.

$$s_{preview}(i) = (0.1*i)*\frac{{v_{vehicle}}^2}{2(|a_{x,braking}| + 0.5)} \qquad i = 1{:}10$$

Example The vehicle moves at a velocity of 90 km/h. The maneuver provides a braking acceleration threshold of $a_{x,Brake,Thr} = -2\frac{m}{s^2}$. The preview distance is 125 m. The preview points are located at 25, 50, 75, 100, and 125 m.

Horizontal preview (road curvature)

The Road subsystem provides the curvature values at the current vehicle position and at the preview points according to the preview distance.

$$\kappa_{road} = Road\_vRefData(s_{current})$$

$$\kappa_{road}(i) = Road_{cRefData}(s_{current} + s_{preview}(i)) \qquad i = 1{:}10$$

Vertical preview (height profile)

The Road subsystem provides the z-position and pitch angle at the preview points according to the preview distance.

$$z_{road}(i) = Road_{vRefData}(s_{current} + s_{preview}(i)) \qquad i = 1{:}10$$

$$\beta_{road}(i) = Road_{vRefData}(s_{current} + s_{preview}(i)) \qquad i = 1{:}10$$

Velocity limits

The road preview information is processed and the permitted velocity is calculated according to the provided driver characteristics parameters. The calculation is performed with respect to the horizontal road profile (curvature) and the road height profile (uphill and downhill). At the end, the lowest value is considered.

Velocity limits according to road curvature The maximum allowed velocity when driving on curves is calculated for the current vehicle position as well as for the preview points by using the follow equation:

$$v_{max,road}(i) = \sqrt{\frac{a_{y,max}}{\kappa_{road}(i)}} \qquad i = 0{:}10$$

Velocity limits according to road height profile Road vertical curvature is approximated from the preview points. The calculated information is then used to define the maximum permitted velocity to avoid losing road contact and guarantee a stable vehicle guidance.

The vertical curvature is approximated by using the s-distance and the z-position of the preview points. Curvature can be calculated from three preview points. The following illustrates the calculation: The three points r_1, r_2 and r_3 are given. The information of the s-distance and the z-position is known for each point.

$$a = r_2 - r_1 = \begin{bmatrix} a_x \\ a_z \end{bmatrix}$$

$$b = r_3 - r_2 = \begin{bmatrix} b_x \\ b_z \end{bmatrix}$$

$$c = 2 \cdot (a_x b_z - a_z b_x)$$

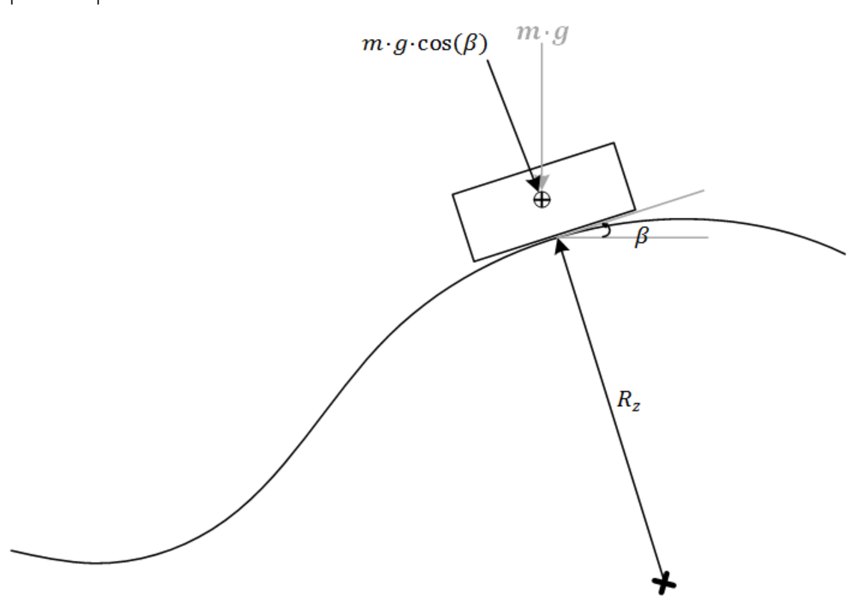
$$p = c \cdot r_1 - \begin{bmatrix} b_z & -a_z \\ -b_x & a_x \end{bmatrix} \cdot \begin{bmatrix} r_2^T \cdot r_2 - r_1^T \cdot r_1 \\ r_3^T \cdot r_3 - r_2^T \cdot r_2 \end{bmatrix}$$

$$\kappa_z = \frac{c}{\sqrt{(p^T \cdot p)}}$$

Where κ_z is the approximated vertical curvature of three preview points.

The velocity limit at each preview point can then be calculated by using the approximated curvature and additional preview information.

Refer to the following illustration, where the calculations are based on three preview points.



$F_n = m \cdot g \cdot \cos(\beta)$

Where:
F_n is the normal force acting on the vehicle
m is the vehicle mass
b is the earth gravitation acceleration
β is the pitch angle

The generated centrifugal force due to the curvature can be calculated as follows:

$$F_c = m \cdot \frac{v^2}{R_z}$$

Where:
v is the resulting tangent speed
R_z is the vertical curvature at the contact point

To ensure a stable guidance across the road profile, the normal force must be larger than the centrifugal force:

$$m \cdot \frac{v^2}{R_z} \leq m \cdot g \cdot \cos(\beta)$$

$$v \leq \sqrt{g \cdot R_z \cdot \cos(\beta)}$$

$$Or\ v \leq \sqrt{\frac{g \cdot \cos(\beta)}{\kappa_z}}$$

With

$$\kappa_z = \frac{1}{R_z}$$

If you specify a maximum allowed vertical acceleration as a parameter, the resulting equation can be rewritten as:

$$v_{max,road} \leq \sqrt{\frac{a_{z,max} \cdot \cos(\beta)}{\kappa_z}}$$

The calculations for the other preview points are performed in a similar manner.

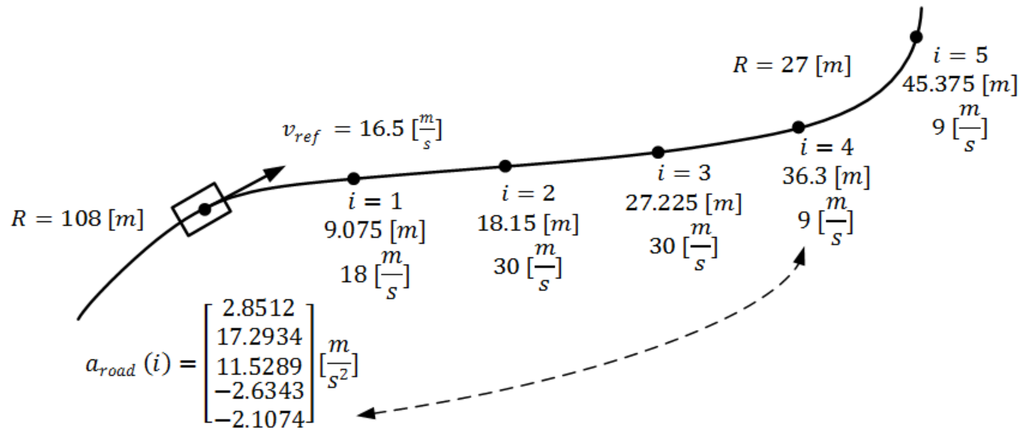
Acceleration limits The resulting velocity limits from the previous calculation are compared and the lowest value is considered.

Afterwards, the constant acceleration, which is required to accelerate the vehicle from the current reference velocity to the permitted velocity is calculated as follows:

$$a_{road}(i) = \frac{v_{max,road}(i)^2 - v_{ref}{}^2}{2{*}s_{preview}(i)} \qquad i = 1{:}10$$

Example: Consider a road with a curvature profile as shown in the following illustration and assume five preview points. The following parameters and settings are assumed:

- The vehicle is moving at the reference velocity.
- Maximum allowed velocity limit on road = 30 [m/s]
- $V_{ref} = v_{vehicle} = 16.5\ [m/s]$
- $a_{x,braking} = 2.5\ [m/s^2]$
- $a_{x,acceleration} = 3\ [m/s^2]$



Variable	Current vehicle position	Preview points				
		***i* = 1**	***i* = 2**	***i* = 3**	***i* = 4**	***i* = 5**
s	0	9.075	18.15	27.225	36.3	45.375
k_{road}	1/108	1/108	0	0	1/27	1/27
$v_{max,road}$	18	18	30	30	9	9
a_{road}	-	2.8512	17.2934	11.5289	-2.6343	-2.1074

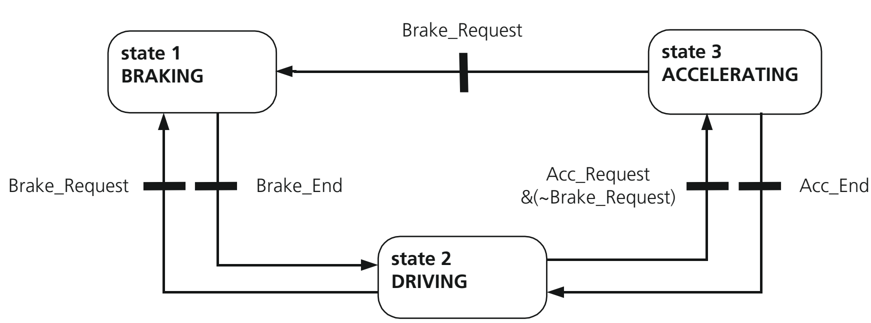
Driving states

As can be seen in the illustration above, the values for a_{Road} can be outside a reasonable range. The Driving States subsystem has the task of calculating appropriate reference accelerations. The processed preview information from the Velocity Limits subsystem is used to define the current driving state. Three states are used: braking, driving, and accelerating.

- In the driving state the reference acceleration is zero.
- In the braking state the reference acceleration is negative.
- In the accelerating state the reference acceleration is positive.

During driving on clothoids, the reference acceleration is adapted to the road curvature.

The following Stateflow chart shows the three states and corresponding state transitions.



The following functionality is implemented for state changes.

Braking is initiated (Brake_Request) if one of the following conditions applies:

- One of the previewed acceleration values exceeds the limit given by parameter $a_{x,braking}$.
- None of the acceleration values exceeds the limit given by $a_{x,braking}$, but the acceleration value corresponding to the nearest preview point is negative and the minimum of all previewed acceleration values. This ensures braking for small velocity differences even if the braking threshold is not exceeded.

Braking is stopped (Brake_End) if

- The required end velocity is approached.

Accelerating is initiated (Acc_Request) if

- The acceleration values corresponding to the first two preview points are positive and the current permitted vehicle velocity is larger than the reference velocity.

Accelerating is ended (Acc_End) if

- The required end velocity is approached.

Integrator

The **Integrator** block integrates the reference acceleration to provide a smooth reference velocity. Anti-windup functionality is ensured by values for upper and lower limits for the integrator, which are provided by the **Driving States** subsystem according to the driver state.

Note

- The reference velocity provided by the **V_ROAD_REF** block cannot exceed the road elements speed limits, i.e., the driver cannot drive faster than these speed limits.
- The **V_ROAD_REF** block provides reasonable reference velocities only if the vehicle velocity is close to reference velocity. A good parameterization of the longitudinal controller is required.
- Due to a discrete number of preview points, the actual braking acceleration may be smaller than the parameter $a_{x,braking}$. The difference may be larger in roads which use only circular and straight segments. In realistic road models, the transition between straight and circular segments is done by clothoids. In this case the resulting braking acceleration is close to the given parameter value.

Inports

The following table shows the inports:

Name	Unit	Description
a_x_Max_Acceleration	[m/s^2]	Maximum allowed longitudinal acceleration
a_x_Max_Braking	[m/s^2]	Maximum allowed longitudinal deceleration
a_y_Max	[m/s^2]	Maximum allowed lateral acceleration
a_z_Max_Deflect	[m/s^2]	Maximum allowed vertical acceleration in positive direction

Name	Unit	Description
a_z_Max_Rebound	[m/s^2]	Maximum allowed vertical acceleration in negative direction
AccBr_Mode	[1|2]	Mode signal for accelerator and brake pedal: ▪ 1: Stimulus ▪ 2: Driver
AdaptVelocityToRoad_Mode	[1|2]	Switch for velocity adaptation: ▪ 1: Off ▪ 2: On
Curv_Preview_Road	[1/m]	Road curvature at preview points
Curv_Road	[1/m]	Road curvature at current vehicle position
Dist_Border_Left	[m]	Distance from the vehicle reference coordinate system to the left road border. A negative value means the vehicle has left the road from the left side.
Dist_Border_Right	[m]	Distance from the vehicle reference coordinate system to the right road border. A negative value means the vehicle has left the road from the right side.
PitchAngle_Preview_CoorSys_E	[deg]	Road pitch angle at preview points
Pos_z_Preview_CoorSys_E	[m]	Road z-position at preview points
Reset	[0/1]	Reset of states
s_Veh_Road	[m]	S-distance along the route at the position of the vehicle
v_Preview_Road_Lim	[m/s]	Road speed limit at preview points
v_Road_Lim	[m/s]	Road speed limit at current vehicle position
v_x_Vehicle	[m/s]	Vehicle velocity
v_x_Vehicle_Ref_Maneuver	[m/s]	Vehicle reference velocity provided by maneuver

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 📖).
a_x_Vehicle_Ref	[m/s^2]	Vehicle reference acceleration
s_Preview	[m]	Preview distance for reference velocity calculation (vector)
v_x_Vehicle_Ref	[m/s]	Vehicle reference velocity

Parameters

The following table shows the block parameters:

Name	Unit	Description
StepSize	[s]	Simulation step size

Related topics

References

Maneuver

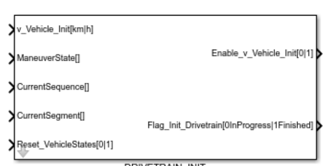
Where to go from here

Information in this section

Drivetrain Initialization

Description

The DRIVETRAIN_INIT block is used to enable drivetrain initialization in scenarios with initial vehicle velocity.



The block generates the following signals:

- Enable_v_Vehicle_Init: When the scenario starts with an initial velocity, this signal is set to *1* for a certain period to allow the drivetrain initialization.

- Flag_Init_Drivetrain: This flag remains *0* during the drivetrain initialization phase. Once the initialization has been finished, the flag becomes active. This gives an indication to the scenario to start.

Inports

The following table shows the inports:

Name	Unit	Description
CurrentSegment	[]	Current segment number
CurrentSequence	[]	Current sequence number
ManeuverState	[]	Maneuver state
Reset_VehicleStates	[0|1]	Reset of vehicle states
v_Vehicle_Init	[km/h]	Initial vehicle velocity

Outports

The following table shows the outports:

Name	Unit	Description
Enable_v_Vehicle_Init	[0|1]	Drivetrain initialization flag for scenarios with non-zero vehicle initial velocity
Flag_Init_Drivetrain	[0|1]	Status flag of drivetrain initialization for scenarios with non-zero vehicle initial velocity: ▪ 0: In progress ▪ 1: Finished

Parameters

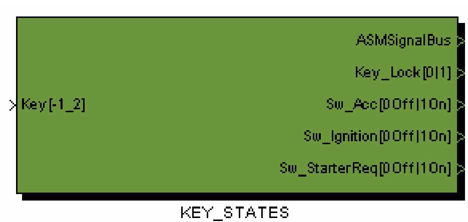
The following table shows the parameters:

Name	Unit	Description
StepSize	[]	Simulation step size

Key States

Description

The KEY_STATES block calculates the ignition and starter request according to the key-position.



Inports

The following table shows the inports:

Name	Unit	Description
Key	[-1|0|1|2]	Key-position: ▪ -1: Off ▪ 0: Park ▪ 1: Ignition on ▪ 2: Starter request

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 📖).
Key_Lock	[0|1]	Key locked signal: ▪ 0: Key is not locked ▪ 1: Key is locked
Sw_Acc	[0|1]	State of accessories: ▪ 0: Accessories are on ▪ 1: Accessories are off
Sw_Ignition	[0|1]	Ignition request ▪ 0: Ignition is off ▪ 1: Ignition is on
Sw_StarterReq	[0|1]	Starter request ▪ 0: Starter is off ▪ 1: Starter is on

Related topics

References

Maneuver Function Call Generator

Description

The MANEUVER_FCN_CALL_GENERATOR block generates function call events separated by a sample-based gap. If specific maneuver state transitions occur, this block internally creates a reset pulse to synchronize the generation of function call events with the start of the maneuver.

The NumSamples parameter specifies the number of time steps between two function call events. For example, if the step size of the model is set to `1 ms` and the block parameter is set to `20`, the block generates function call events every `20 ms`.

Inports

The following table shows the inports:

Name	Unit	Description
ManeuverState	[]	Maneuver state signal used to trigger a reset signal for function call generation.

Outports

The following table shows the outports:

Name	Unit	Description
FcnCallEvent	[]	Function call event generated depending on the NumSamples parameter.

Parameters

The following table shows the parameters:

Name	Unit	Description
NumSamples	[]	Number of time steps between two function call events.

Maneuver Scheduler

Description

The MANEUVER_SCHEDULER block is the core of the Maneuver subsystem. It controls the vehicle, road, and driver models to perform maneuvers by setting mode signals and reference signals.

In normal operation, the maneuver scheduler controls the model in such a way that the vehicle performs a given maneuver. It also provides additional features for improved model handling. These features are:

- Predefined maneuvers
- Maneuver start and stop flags
- Vehicle model reset flag
- Manual driving

Predefined maneuvers

There are two possibilities to switch between maneuvers:

- You can download maneuvers from ModelDesk (*Maneuver_ModelDesk*). This is the standard procedure.

- The model contains five predefined maneuver definitions (*Maneuver_2-6*), which you can select via the Sw_ManeuverSelect variable. This variable has a default value of *1*, which corresponds to the maneuver accessed by ModelDesk (*Maneuver_ModelDesk*). Every time when ModelDesk downloads a maneuver, Sw_Maneuver_Select is set to *1* (in case of Simulink simulation) or to *-1* (in case of VEOS and real-time simulation). The predefined maneuvers which are contained in the model are loaded during model start-up and are specified in the `asm_road_maneuver_ini.m` file.

Maneuver start and stop flags

The `ManeuverStart` variable is used to start the maneuver. The maneuver is started if `ManeuverStart` has a value of 1. This value is recommended as the default value for offline simulations. The maneuver is then started immediately after simulation start. In HIL applications, you may want to start the maneuver manually. A default value of *0* is recommended for this.

The `ManeuverStop` variable has a default value of *0*. If the maneuver stop flag has a value of *1*, the current maneuver is quit.

Vehicle model reset flag

The `Reset_VehicleStates` variable is used to reset the vehicle model. In HIL applications it is recommended to use the `ManeuverStop` variable in general. In this case, the vehicle is braked correctly and the ECU receives valid signals. In case of strange model behavior it is recommended to use the `Reset_VehicleStates` variable.

The `ManeuverStop` variable is set, every time the `Reset_VehicleStates` variable is activated. A running maneuver is interrupted automatically this way.

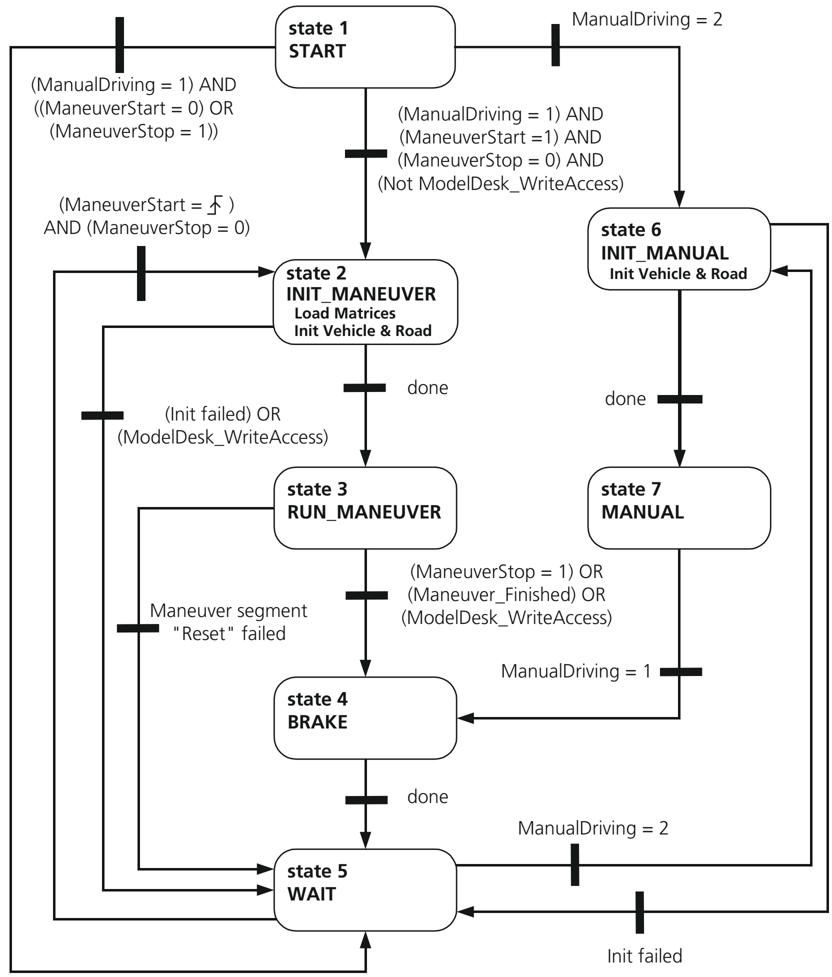
Manual driving

In some cases manual driving is required. You can use the `Sw_ManualDriving` variable to switch off the maneuver scheduler. A value of *1* means the maneuver scheduler is active. A value of *2* means the maneuver scheduler is switched off and the signals for manual driving with ControlDesk (suffix _CDesk) are fed through. It is not possible to activate the manual mode when a maneuver is running. The running maneuver must be stopped before. See the following Stateflow chart for details.

Maneuver states and transitions

Internally, the maneuver scheduler uses a state machine to run maneuvers defined in ModelDesk and for manual driving.

The following stateflow chart shows the maneuver states and possible state transitions.



The states on the left show the normal operation of the maneuver scheduler with maneuvers defined in ModelDesk. The states on the right hand are used for manual driving. To avoid inconsistent simulation results, the current maneuver is quit if ModelDesk downloads new data. The `ModelDesk_WriteAccess` variable is used for this functionality.

Inports

The following table shows the inports:

Name	Unit	Description
a_y_Vehicle	[m/s^2]	Vehicle lateral acceleration.
Dist_Refline_Response	[m]	Distance to the road reference line according to the maneuver's distance definition to the road reference line.
Dist_Response	[m]	Distance according to the maneuver's distance definition.
External_SegmentEnd	[0|1]	Flag for user defined segment transition (vector of 25 values).

Name	Unit	Description
ExternalSignals	[]	Signal bus with external driver signals, such as accelerator pedal, brake pedal and others (for a description of the signals, see below).
Keep_VehiclePosition	[0|1]	Flag to disable all integrators in the model in order to keep the vehicle at its current position.
LastValues	[]	Signal bus with driver output signals to ensure hold functionality (for a description of the signals, see below).
ManeuverSettings	[]	Signal bus with maneuver settings and parameters (for a description of the signals, see below).
ManeuverStart	[0|1]	Maneuver start flag from MdlUserInterface used to manually start running maneuvers.
ManeuverStop	[0|1]	Maneuver stop flag from MdlUserInterface used to manually stop running maneuvers.
ManualControl	[]	Signal bus with signals used for manual driving (for a description of the signals, see below).
ModelDesk_WriteAccess	[]	Flag is set to nonzero values if ModelDesk downloads parameters to the model.
r_Tire	[m]	Tire radius.
Reset_VehicleStates	[0|1]	Vehicle model reset flag from MdlUserInterface used to manually quit the running maneuver and reset all vehicle integrators.
RoadState	[]	Road state signal used for communication between maneuver and road.
s_Road	[m]	Current vehicle position on road (starts with 0 in new lap).
s_Total_Road	[m]	Current vehicle position on road.
s_Vehicle	[m]	Traveled vehicle distance.
Sw_ManeuverSelect	[1_6]	Select maneuver switch.
Sw_ManualDriving	[1|2]	Manual driving switch.
t_v_Ref_Preview	[s]	Preview time for reference speed calculation.
v_Vehicle	[m/s]	Vehicle velocity.

ExternalSignals The ExternalSignals group contains driver reference values which are calculated in the model. These signals can be activated in the maneuver segments during running maneuvers. This group can be used to insert customer specific calculations.

Name	Unit	Description
Angle_SteeringWheel	[deg]	Steering wheel angle.
d_Ref	[m]	Reference value for vehicle lateral offset. This value is added to the lateral offset which is derived from the reference lane.
Gear	[]	Gear
LaneIdx_Ref	[]	Reference value for the lane the vehicle shall drive on.
Pos_AccPedal	[%]	Accelerator pedal position.
Pos_BrakePedal	[%]	Brake pedal position.
Pos_ClutchPedal	[%]	Clutch pedal position.
SelectorLever	[]	Selector lever position.

Name	Unit	Description
Trq_Steering	[Nm]	Steering torque.
v_Vehicle_Ref	[m/s]	Vehicle reference velocity.

LastValues The LastValues group contains the driver output signals of the last time step and the information of the lateral vehicle position on the road. These signals are used to ensure hold functionality.

Name	Unit	Description
Angle_SteeringWheel_Last	[deg]	Steering wheel angle from driver.
d_Total_Veh_Road_Last	[m]	Lateral vehicle offset from road reference line.
Gear_Last	[]	Gear from driver.
LaneIdx_Total_Veh_Road_Last	[]	Index of current lane, the vehicle drives on. Floating values are used to describe intermediate positions.
Pos_AccPedal_Last	[%]	Accelerator pedal position from driver.
Pos_BrakePedal_Last	[%]	Brake pedal position from driver.
Pos_ClutchPedal_Last	[%]	Clutch pedal position from driver.
SelectorLever_Last	[]	Last selector lever position.
Trq_SteeringWheel_Last	[Nm]	Steering torque from steering stimulus.

ManeuverSettings The ManeuverSettings group contains parameters which influence the maneuver output signals in certain maneuver states or segments.

Name	Unit	Description
Pos_BrakePedal_Seg_BrakeUntilStop	[%]	Brake pedal position which is used in *Brake Until Stop* segments.
Pos_BrakePedal_Seg_Standstill	[%]	Brake pedal position which is used in *Standstill* segments.
Pos_BrakePedal_State_Brake	[%]	Brake pedal position which is used in Maneuver State *Brake*.
Pos_BrakePedal_State_Wait	[%]	Brake pedal position which is used in Maneuver State *Wait*.
Pos_ClutchPedal_Seg_Standstill	[%]	Clutch pedal position which is used in *Standstill* segments.
Rate_Angle_SteeringWheel_State_Wait	[deg/s]	The steering wheel angle is smoothly set to zero in maneuver state *Brake*. This value defines the transition rate.

ManualControl The ManualControl group contains the driver reference signals to manually control the vehicle model. These signals can be activated in the maneuver segments during running maneuvers. Additionally these signals are used in maneuver state 'Manual'

Name	Unit	Description
Angle_SteeringWheel	[deg]	Steering wheel angle.
d_Ref	[m]	Reference value for vehicle lateral offset. This value is added to the lateral offset which is derived from the reference lane.
Gear	[]	Gear.
LaneIdx_Ref	[]	Reference value for the lane the vehicle shall drive on.
Pos_AccPedal	[%]	Accelerator pedal position.
Pos_BrakePedal	[%]	Brake pedal position.

Name	Unit	Description
Pos_ClutchPedal	[%]	Clutch pedal position.
SelectorLever	[]	Selector lever position.
Trq_Steering	[Nm]	Steering torque.
v_Vehicle_Ref	[m/s]	Vehicle reference velocity.

Outports

The maneuver output signals are grouped. The following table shows the groups for the outports:

Name	Unit	Description
Info	[]	Signal bus containing information on maneuver internal data like the current maneuver state.
InitialVehStates	[]	Signal bus containing initial vehicle states.
MdlControl	[]	Signal bus containing signals for general model control.
ModeSignals	[]	Signal bus containing mode signals.
Parameters	[]	Signal bus containing parameters.
RefSignals	[]	Signal bus containing reference signals.
UserSignals	[]	Signal bus containing user signals.

Note

All signals are contained in the ASMSignalBus. The following tables show the maneuver scheduler output signal groups.

Info The Info group contains data on internal signals used in the maneuver scheduler.

Name	Unit	Description
CurrentSegment	[]	Current segment number.
CurrentSequence	[]	Current sequence number.
ErrorFlag	[]	Error flag.
ManeuverState	[]	Current maneuver state (see illustration above).
ManeuverTime	[s]	Time after maneuver start.
s_local	[m]	Distance the vehicle is driven in the current maneuver segment.
t_local	[s]	Time since start of the current maneuver segment.

InitialVehStates The InitialVehStates bus is used to set the initial vehicle position on the road.

Name	Unit	Description
Angle_Yaw_Vehicle_Rel_Init	[deg]	Initial vehicle orientation relative to road.
d_Vehicle_Init	[m]	Initial lateral offset from preferred lane.
LaneIdx_Vehicle_Init	[]	Initial lane index.

Name	Unit	Description
Pos_z_Vehicle_Rel_Init	[m]	Initial vehicle height relative to road.
Route_Direction_Init	[]	Direction of travel route.
Route_Id_Init	[]	ID of route for maneuver.
s_Vehicle_Init	[m]	Initial vehicle position on route.
v_Vehicle_init	[km|h]	Initial vehicle velocity.

MdlControl The **MdlControl** group is used for model initialization and state transitions.

Name	Unit	Description
Get_VehPos_Trigger	[]	Current vehicle position is used as initial position for straight and circle segments. This flag is used to control the basic road block. It is used in maneuvers without road only.
Keep_VehiclePosition	[0|1]	Partial reset of vehicle model to keep vehicle at current position.
Reset_VehicleStates	[0|1]	The reset vehicle flag is used to reset all vehicle integrators. It is activated in case of a manual reset request from the MdlUserInterface or in case of model initialization and maneuver initialization.
RoadControl	[]	Used to control the road model (reload road data).

ModeSignals The **ModeSignals** group controls main switches in the model.

Name	Unit	Description
AccBr_Mode	[]	Controls whether accelerator and brake pedal positions are stimulus signals or controlled by driver.
AdaptVelocityToRoad_Mode	[]	Controls whether reference velocity calculation on roads is active.
Clutch_Mode	[]	Controls whether clutch pedal position is stimulus signal or controlled by driver.
CurvatureCalculation_Mode	[1|2]	Specifies the curvature calculation mode: ▪ 1: The ASM driver uses the curvature calculated by the ASM Road. ▪ 2: The ASM driver calculates the curvature based on the position of the preview points.
ForceToRoad_Mode	[]	Controls the interpretation of lane index in the road calculations, if not existing lanes are specified. This is only used during maneuvers with road.
Gear_Mode	[]	Controls whether gear is stimulus signal or controlled by driver.
LatDriver	[]	Controls whether lateral_control1 or lateral_control2 is active.
Road_Mode	[]	Controls whether vehicle drives on flat area, on straight or circular road or on the road which is created with ModelDesk's **Road Generator**.
SelectorLever_Mode	[1|2]	Controls the selector lever position: ▪ 1: Stimulus signal ▪ 2: Driver control
Steer_Mode	[]	Controls whether the steering wheel angle is controlled by stimulus signal or by driver.

Parameters The Parameters group contains parameter values which are set by the maneuver scheduler.

Name	Unit	Description
a_x_Brake_Threshold	[m/s^2]	Parameter used for reference velocity calculation
a_x_Max	[m/s^2]	Parameter used for reference velocity calculation
a_y_Max	[m/s^2]	Parameter used for reference velocity calculation
Curv_Road_Circle	[1/m]	Curvature for circular roads
Gear_Max	[]	Maximum gear for the gearshift
v_Lat	[m/s]	Parameter to control the road calculations. Defines the lateral velocity during lane changes.
Width_Deadzone_LatCtrl	[m]	Defines the width of a tolerance zone around the reference trajectory. Within this tolerance zone, the lateral controller ignores deviations from the reference trajectory. This is used for tests of lane assist systems.

RefSignals The RefSignals group contains reference signals for pedal stimulus and for vehicle velocity.

Name	Unit	Description
Angle_SteeringWheel_Maneuver	[deg]	Reference signal for steering wheel angle
d_Maneuver	[m]	Reference signal for lateral offset from road reference line
Dist_Def_Maneuver	[]	3x1 vector containing signals for distance calculation performed in road module
Dist_Def_Refline_Maneuver	[]	4 x 1 vector containing signals for distance calculation performed in the traffic module
Gear_Maneuver	[]	Reference signal for gear stimulus
LaneIdx_Maneuver	[]	Reference signal for road lane the vehicle shall drive on
Pos_AccPedal_Maneuver	[%]	Reference signal for accelerator pedal stimulus
Pos_BrakePedal_Maneuver	[%]	Reference signal for brake pedal stimulus
Pos_ClutchPedal_Maneuver	[%]	Reference signal for clutch pedal stimulus
SelectorLever_Maneuver	[]	Reference signal for the selctor lever position
Sw_Steering_Mode_Maneuver	[1|2]	Reference signal for steering angle calculation: ▪ 1: Angle ▪ 2: Torque
Trq_Steering_Maneuver	[Nm]	Reference torque for the steering angle calculation
v_Vehicle_Ref_Maneuver	[m/s]	Reference signal for vehicle velocity
v_Vehicle_Ref_Preview_Maneuver	[m|s]	Reference signal for preview speed calculation
v_Vehicle_Ref_Preview_State	[0|1|2]	Reference signal to indicate which road segment is evaluated for the preview speed calculation: ▪ 0: Invalid ▪ 1: Current segment ▪ 2: Next segment

UserSignals The UserSignals group contains 30 signals defined with the Scenario Editor, refer to How to Select and Specify Maneuver User Signals (ModelDesk Scenario Creation 📖).

Name	Unit	Description
User1	[]	User signal defined with the Scenario Editor
...	...	...
User30	[]	User signal defined with the Scenario Editor

Parameters

The following table shows the block parameters:

Name	Unit	Description
Maneuver_ModelDesk	[]	Maneuver_ModelDesk: Definition of the maneuver sequences and their segments (vector). This vector is accessed by ModelDesk to change the maneuvers online and offline.
Maneuver 2	[]	Maneuver 2: Definition of maneuver sequences and their segments (vector).
Maneuver 3	[]	Maneuver 3: Definition of maneuver sequences and their segments (vector).
Maneuver 4	[]	Maneuver 4: Definition of maneuver sequences and their segments (vector).
Maneuver 5	[]	Maneuver 5: Definition of maneuver sequences and their segments (vector).

Related topics

HowTos

How to Accelerate and Brake in a Maneuver (ModelDesk Scenario Creation)

Maneuver Select Switch

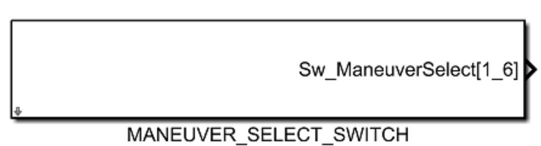
Description

The MANEUVER_SELECT_SWITCH block is used to switch between a ModelDesk maneuver and other predefined maneuvers.

The model contains six different maneuvers:

- A maneuver that is downloaded from ModelDesk (Maneuver_ModelDesk)
- Five predefined maneuvers (Maneuver_2 - Maneuver_6). These are loaded during model startup and are specified in the `asm_road_maneuver_ini.m` file.

You can select one of the maneuvers via the SW_Maneuver_Select outport. The maneuver is then interpreted by the MANEUVER_SCHEDULER block.



Outports

The following table shows the outports:

Name	Unit	Description
Sw_ManeuverSelect	[1_6]	Selector used to switch between the ModelDesk maneuver and other predefined maneuvers.

Related topics

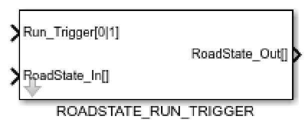
References

Maneuver Scheduler.. 65

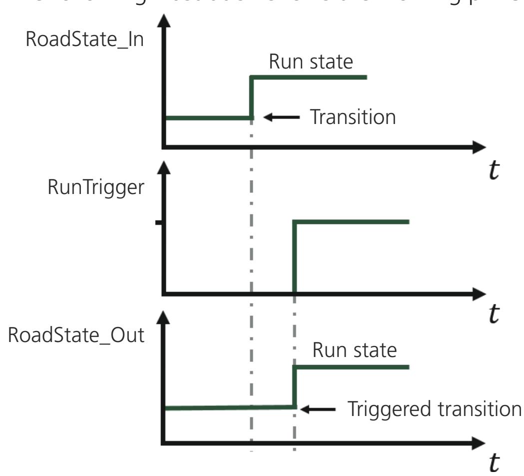
Road State Run Trigger

Description

The ROADSTATE_RUN_TRIGGER block triggers the transition of the road state to the run state, on the basis of the input trigger signal.



The following illustration shows the working principle of the block:



Inports

The following table shows the inports:

Name	Unit	Description
RoadState_In	[]	State of the road S-function.
Run_Trigger	[0|1]	Trigger to enable the transition of the road state to the run state: ▪ 0: Disabled ▪ 1: Enabled

Outports

The following table shows the outports:

Name	Unit	Description
RoadState_Out	[]	State of the road S-function.

Road Select Switch

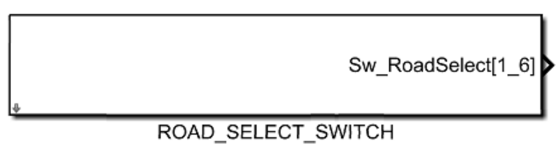
Description

The ROAD_SELECT_SWITCH block is used to switch between the ModelDesk road and other predefined roads.

The model contains six different roads:

- A road model that is downloaded from ModelDesk (Road_ModelDesk).
- Five predefined road definitions (Road_2 - Road_6). These are loaded during model startup and are specified in the `asm_road_maneuver_ini.m` file.

You can select one road via the SW_Road_Select outport. The road is then interpreted by the ROAD block.



Outports

The following table shows the outports:

Name	Unit	Description
Sw_RoadSelect	[1_6]	Selector switch used to switch between the ModelDesk road and other predefined roads.

Road

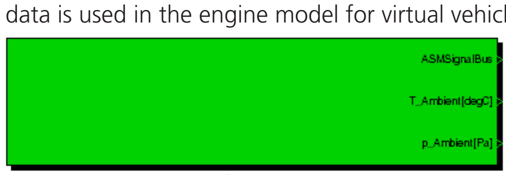
Where to go from here

Information in this section

Ambient

Description

The AMBIENT block provides data on ambient pressure and temperature. This data is used in the engine model for virtual vehicle simulation.



Inports

No inports are required for the functionality of the AMBIENT block.

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 📖).
p_Ambient	[Pa]	Ambient pressure
T_Ambient	[°C]	Ambient temperature

Parameters

The following table shows the block parameter:

Name	Unit	Description
Const_pressure	[mbar]	Ambient pressure
Const_temperature	[°C]	Ambient temperature

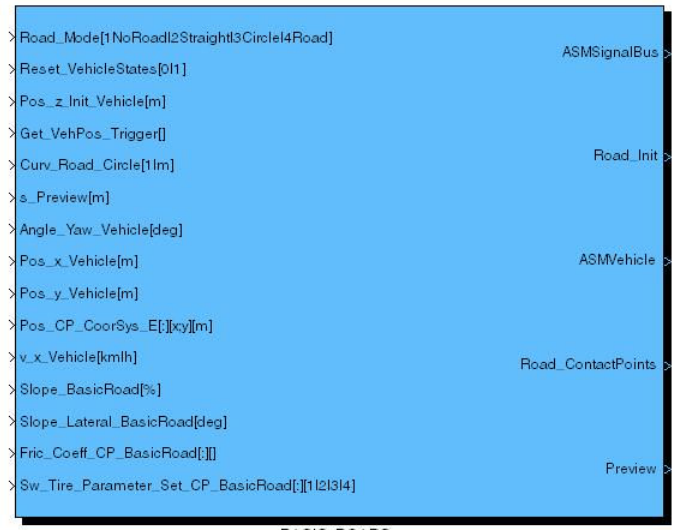
Related topics

References

Basic Road

Description

The BASIC_ROADS block implements simple road segments, which can be straight or circular. The generated road signals correspond to driving on a flat plane. If no road is required for a certain application, the BASIC_ROADS block provides the default road signals to let the vehicle dynamics simulation run properly.



BASIC_ROADS

The straight and the circular road segments are implemented to describe a simple reference route for the driver model. The driver model sends preview points to the road as incremental distances on the road ahead of the vehicle. The road model calculates the reference coordinates at these distances and returns them to the driver model.

For straight road, the coordinates can be calculated as follows:

$$\kappa_{Road} = 0.0$$

$$\gamma_{Road} = \gamma_0$$

$$\begin{bmatrix} x_{Road} \\ y_{Road} \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} s_{preview}\cos(\gamma_0) \\ s_{preview}\sin(\gamma_0) \end{bmatrix}$$

where:

κ_0 is the curvature of the road, which is zero in case of straight segment

y_0 is the vehicle start orientation in the plane

x_0, y_0 are the vehicle start coordinates

For circular road, the coordinates can be calculated as follows:

$$\kappa_{Road} = \kappa_0$$

$$\gamma_{Road} = \kappa_0\, s$$

$$\begin{bmatrix} x_{Road} \\ y_{Road} \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \cos(\gamma_0) & -\sin(\gamma_0) \\ \sin(\gamma_0) & \cos(\gamma_0) \end{bmatrix} \begin{bmatrix} \frac{1}{\kappa_0}\sin(\gamma_{Road}) \\ \frac{1}{\kappa_0}(1-\cos(\gamma_{Road})) \end{bmatrix}$$

where:

κ_0 is the curvature of the road, which is constant in case of circular segment

y_0 is the vehicle start orientation in the plane

x_0, y_0 are the vehicle start coordinates

The slope, lateral slope, friction coefficients and tire parameter set values for the basic road are provided as inputs to the BASIC_ROADS block. They can be modified in
<ModelName>/MDLUserInterface/Environment/MDL_PAR/BasicRoad

Inports

The following table shows the inports:

Name	Unit	Description
Angle_Yaw_Vehicle	[deg]	Vehicle yaw angle
Curv_Road_Circle	[1/m]	Road curvature of circular road defined by maneuver
Fric_Coeff_CP_BasicRoad[:]	[]	Friction coefficient for each tire
Get_VehPos_Trigger	[]	Trigger to read in the current vehicle position which is used as initial position for straight road and circle
Pos_CP_CoorSys_E[:][x;y]	[m]	Position of contact points in earth coordinates
Pos_x_Vehicle	[m]	Vehicle position in x direction in earth coordinates
Pos_y_Vehicle	[m]	Vehicle position in y direction in earth coordinates
Pos_z_Init_Vehicle	[m]	Initial vehicle height relative to the road
Reset_VehicleStates	[0|1]	Reset signal from maneuver scheduler
Road_Mode	[1|2|3|4]	Controls whether vehicle drives on flat area, on a straight line, on a circle or on a road defined by ModelDesk ▪ 1: No road ▪ 2: Straight ▪ 3: Circle ▪ 4: Road from ModelDesk
s_Preview	[m]	Vector with preview distances for lateral control and vehicle reference velocity calculation
Slope_BasicRoad	[%]	Slope for basic road
Slope_Lateral_BasicRoad	[deg]	Lateral slope for basic road
Sw_Tire_Parameter_Set_CP_BasicRoad[:]	[1|2|3|4]	Tire parameter set for each tire on basic road

Outports

The following table shows the outports:

Name	Unit	Description
RoadSignals_Circle	[]	Mixed signal containing data for circular basic road
RoadSignals_Common	[]	Mixed signal containing common data for basic road
RoadSignals_NoRoad	[]	Mixed signal containing data for basic road
RoadSignals_Straight	[]	Mixed signal containing data for straight basic road

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide).
ASMVehicle	[]	Signal bus containing signals of the ASM vehicle
Preview	[]	Signal bus containing preview signals for the driver
Road_ContactPoints	[]	Signal bus containing road property signals such as height and friction for all wheel contact points
Road_Init	[]	Signal bus containing initial position and angle of the ASM vehicle

Note

All signals are contained in the ASMSignalBus. The following tables show the basic road output signal groups.

ASMVehicle The ASM Vehicle group contains road signals and properties at the position of the vehicle:

Name	Unit	Description
Angle_Yaw_Veh_Road	[deg]	Yaw angle of the reference line of the road at the current distance s of the vehicle
Curv_Veh_Road	[1/m]	Curvature of the road at the current distance s of the vehicle
d_Total_Veh_Road	[m]	The total lateral distance of the vehicle to the reference line of the road.
LaneIdx_Total_Veh_Road	[]	Lane index of the vehicle on the road. Since the basic roads do not provide lanes, the index is always zero.
Pos_x_Ref_Veh_Road	[m]	x-position of the reference line of the road at the current distance s of the vehicle. The value also depends on the target lateral offset and lane index.
Pos_y_Ref_Veh_Road	[m]	y-position of the reference line of the road at the current distance s of the vehicle. The value also depends on the target lateral offset and lane index.
s_Total_Veh_Road	[m]	The total distance along the road.
s_Veh_Road	[m]	The distance s along the road at the position of the vehicle. Reset to zero after each turn on a closed road.
Slope_Lateral_Veh_Road	[deg]	Lateral slope of the road at the current distance s of the vehicle
Slope_Veh_Road	[%]	Slope of the road at the current distance s of the vehicle
v_wind_Coor_Sys_E[x;y;z]	[m/s]	Velocity of the wind in the earth coordinate system

Road_ContactPoints The road contact point group contains signals that define the properties of the road at the contact points of the wheels. All signals are vectors whose size depends on the number of wheel position vectors connected to the inports.

Name	Unit	Description
Fric_Coeff_CP[:]	[]	Friction coefficient of the road at the contact point. This value can also be set externally.
Pos_z_CP[:]	[m]	Vertical (z) position of the contact point in the earth coordinate system.

Name	Unit	Description
Sw_Tire_Parameter_Set_CP[:]	[1|2|3|4]	Switch to select one of four sets of tire parameters. This value can also be set externally.
UnitVec_z_CP[:][x;y;z]	[m]	Unit vector perpendicular to the road surface at the contact point, described in earth coordinate system.

Road_Init The road init group contains signals that define the initial position and orientation angles of the vehicle on the road.

Name	Unit	Description
Angle_Vehicle_Init_CoorSys_E[x;y;z]	[rad]	Initial vehicle orientation angles
Pos_Vehicle_Init_CoorSys_E[x;y;z]	[m]	Initial vehicle position

Preview The preview group contains preview signals for the driver.

Name	Unit	Description
Curv_Preview_Road	[1/m]	Curvature of the road at the preview point positions
Pitch_Preview_Road	[deg]	Pitch angle of the road at the preview point positions
Pos_x_Preview_Road	[m]	Vector of previewed road positions in x-direction (earth coordinates)
Pos_y_Preview_Road	[m]	Vector of previewed road positions in y-direction (earth coordinates)
Pos_z_Preview_Road	[m]	Vector of previewed road positions in z-direction (earth coordinates)
Roll_Preview_Road	[deg]	Roll angle of the road at the preview point positions
Speed_Limit_Preview_Road	[km/h]	Speed limit at the preview point positions

Parameters

The following table shows the block parameters:

Name	Unit	Description
StepSize	[s]	Simulation Step Size

Related topics

References

History of the BASIC_ROADS Block.. 175

GPS Position

Description

The GPS_POSITION block converts a given (x,y)-position to GPS longitude and latitude.

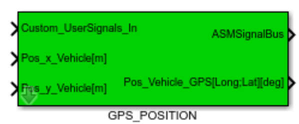
The following options are available:

- No conversion takes place. Longitude and latitude are set to zero.
- The conversion algorithm uses a simple spherical geodetic reference system.
- The conversion algorithm uses the UTM conversion based on the World Geodetic System 1984 (WGS 84). The algorithm is based on the formula in Map Projections - A Working Manual by J.P. Snyder (1987) on page 63.

You can set the conversion method by configuring the road network user signal 1. Its alias is GPS method[0xy|1GPS|2GPS_WGS84]:

- 0: The algorithm does not compute a conversion.
- 1: The algorithm uses a simple spherical reference system during the conversion.
- 2: The algorithm performs a UTM conversion on the basis of the WGS 84 system.

The road network user signals 2 and 3 set the start coordinates for latitude and longitude. Their aliases are start coordinate latitude[deg] and start coordinate longitude[deg].



Heading angle

The block also calculates the heading angle of the vehicle. The calculation is done with the following equation:

$$A = \cos(lat_2) \cdot \sin(long_2 - long_1)$$

$$B = \cos(lat_1) \cdot \sin(lat_2) - \sin(lat_1) \cdot \cos(lat_2) \cdot \cos(long_2 - long_1)$$

$$heading = \mathrm{atan2}(\mathrm{A},\ \mathrm{B})$$

Where

$long_i$,lat_i are the longitude and latitude of point *i*.

Inports

The following table shows the inports:

Name	Unit	Description
Custom_UserSignals_In	[m]	Signal bus that contains customized user signals of the loaded road.
Pos_x_Vehicle	[m]	X-position of the vehicle in earth coordinate system.
Pos_y_Vehicle	[m]	Y-position of the vehicle in earth coordinate system.

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 📖).
Pos_Vehicle_GPS[Long;Lat]	[deg]	(Longitude,Latitude) vehicle position corresponding to the (x,y) input values.

Related topics

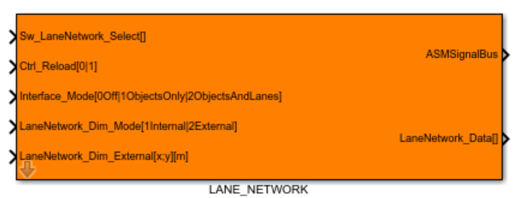
References

Lane Network

Description

The LANE_NETWORK block provides static road data, such as lane boundaries and the lane topology calculated by ModelDesk. The road data is assigned to different network sections. The dimensions of the network sections can be parameterized.

The block is intended for use with the OSI Groundtruth Interface block. Refer to OSI Groundtruth Interface (ASM Traffic Reference 📖).



Inports

The following table shows the inports:

Name	Unit	Descption
Ctrl_Reload	[0|1]	Signal to trigger a reload of the lane network parameter
Interface_Mode	[0|1|2]	Signal to control the LANE_NETWORK block (mode is set by the ground truth interface) ▪ 0: Interface is off ▪ 1: Interface mode for *objects only* (LANE_NETWORK not used) ▪ 2: Interface mode for objects and lanes
LaneNetwork_Dim_External	[]	Dimensions of lane network sections for LaneNetwork_Dim_Mode = 2 (external)

Name	Unit	Desciption
LaneNetwork_Dim_Mode	[1|2]	Signal to control the dimension mode: ▪ 1: Dimensions of lane network sections set internally by the LANE_NETWORK parameters ▪ 2: Dimensions of lane network sections set externally (via LANE_NETWORK input)
Sw_LaneNetwork_Select	[]	Selector to chose one of the six lane networks that are set as block parameters

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 📖).
LaneNetwork_Data	[]	Signal with reference to internal lane network data

ASMSignalBus output

The following table shows the signals in the ASMSignalBus output:

Name	Unit	Description
Data_Valid	[]	Flag that shows whether the output data is valid and can be used: ▪ 0 : Invalid (before or while reading data or if an error occurred) ▪ 1 : Valid
LaneNetwork_State	[]	State of the block: ▪ 1: Sleep ▪ 2: Reading parameter vector ▪ 3: Normal output of data ▪ 4: Error ▪ 5: Reset

Parameters

The following table shows the parameters:

Name	Unit	Description
Const_Max_Num_Lb_Per_Section	[]	Maximum number of lane boundaries within one lane network section.
Const_Max_Num_Lanes_Free_Per_Step	[]	Maximum number of lanes freed within one simulation step.
Const_Max_Num_Lanes_Per_Step	[]	Maximum number of lanes loaded within one simulation step.
Const_Max_Num_Lb_Free_Per_Step	[]	Maximum number of lane boundary points freed within one simulation step.
Const_Max_Num_Pnts_Per_Step	[]	Maximum number of lane boundary points loaded within one simulation step.
Const_Section_Dim_x	[m]	Longitudinal dimension of one lane network section.
Const_Section_Dim_y	[m]	Lateral dimension of one lane network section.

Name	Unit	Description
LaneNetwork_2_6	[]	Five additional lane network vectors. ModelDesk does not access these parameters, but they can be selected via the Sw_LaneNetwork_Select inport.
LaneNetwork_ModelDesk	[]	This vector contains the lane network definition. This parameter is accessed by ModelDesk when you work offline in Simulink.

Related topics

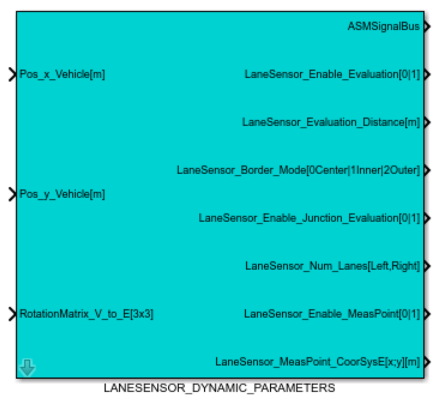
References

OSI Groundtruth Interface (ASM Traffic Reference)

Lane Sensor Dynamic Parameters

Description

The LANESENSOR_DYNAMIC_PARAMETERS block defines the parameters needed by the Lane Sensor Dynamic.



Inports

The following table shows the inports:

Name	Unit	Description
Pos_x_Vehicle	[m]	X-position of the vehicle in the earth coordinate system
Pos_y_Vehicle	[m]	Y-position of the vehicle in the earth coordinate system
RotationMatrix_V_to_E	[]	Rotation matrix to transfer from the vehicle to the earth coordinate system [3x3]

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide).
LaneSensor_Enable_Evaluation	[0|1]	Flag for the lane sensor evaluation at each lane sensor evaluation distance: ▪ 0: Enable lane sensor evaluation at each lane sensor evaluation distance ▪ 1: Disable lane sensor evaluation at each lane sensor evaluation distance
LaneSensor_Evaluation_Distance	[m]	Vector of distances from the vehicle in the s-direction at which the lane sensor information is evaluated
LaneSensor_Border_Mode	[0|1|2]	Switch to define how the sensor interprets lane borders: ▪ 0: Lane borders are at the centerline of lane markings ▪ 1: Lane borders are at the inner edge of lane markings ▪ 2: Lane borders are at the outer edge of lane markings
LaneSensor_Enable_Junction_Evaluation	[0|1]	Flag to determine if lane sensor outputs on junctions are evaluated: ▪ 0: No sensor output on junctions ▪ 1: The current lane information is interpolated by using the previous and next road
LaneSensor_Num_Lanes	[]	Number of lanes to the right and left of the current lane to be considered by the lane sensor [Left,Right]
LaneSensor_Enable_MeasPoint	[0|1]	Flag for the evaluation of each measurement point: ▪ 0: Enable the evaluation of each measurement point ▪ 1: Disable the evaluation of each measurement point
LaneSensor_MeasPoint_CoorSys_E	[m]	Measurement point positions in the earth coordinate system [x,y]

Parameters

The following table shows the parameters:

Name	Unit	Description
Sw_LaneSensor_Enable	[0|1]	Flag for the lane sensor evaluation at each lane sensor evaluation distance ▪ 0: Enable the lane sensor evaluation at each lane sensor evaluation distance ▪ 1: Disable the lane sensor evaluation at each lane sensor evaluation distance
Const_Dist_LaneSensor	[m]	Vector of distances from the vehicle in the s-direction at which the lane sensor information is evaluated
Sw_LaneSensor_Border_Mode	[0|1|2]	Switch to specify how the sensor interprets lane borders: ▪ 0: Lane borders are at the centerline of lane markings ▪ 1: Lane borders are at the inner edge of lane markings ▪ 2: Lane borders are at the outer edge of lane markings

Name	Unit	Description
Sw_LaneSensor_Junction_Enable	[0|1]	Flag to specify if lane sensor outputs on junctions are evaluated: ▪ 0: No sensor output on junctions ▪ 1: The current lane information is interpolated by using previous and next road
Const_NumLanes_Left	[]	Number of lanes to the left of the current lane to be considered by the lane sensor
Const_NumLanes_Right	[]	Number of lanes to the right of the current lane to be considered by the lane sensor
Const_Pos_MeasPoint_1	[m]	x- and y-coordinates of the first measurement point in the vehicle coordinate system [x,y]
Const_Pos_MeasPoint_2	[m]	x- and y-coordinates of the second measurement point in the vehicle coordinate system [x,y]
Const_Pos_MeasPoint_3	[m]	x- and y-coordinates of the third measurement point in the vehicle coordinate system [x,y]
Const_Pos_MeasPoint_4	[m]	x- and y-coordinates of the fourth measurement point in the vehicle coordinate system [x,y]

Related topics

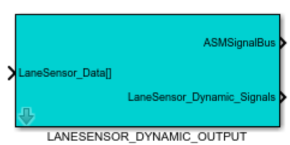
References

Lanesensor Dynamic Parameters (ModelDesk Parameterizing)

Lane Sensor Dynamic Output

Description

The LANESENSOR_DYNAMIC_OUTPUT block receives the lane information from the road network and outputs the Lane Sensor Dynamic values.



Inports

The following table shows the inports:

Name	Unit	Description
LaneSensor_Data	[]	Signal with a reference to the Lane Sensor Dynamic data that is defined in the road network

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 📖).
LaneSensor_Dynamic_Signals	[]	Signal bus containing all sensor output signals.

Parameters

The following table shows the parameters:

Name	Unit	Description
Const_SensorPoints_OutputDimension	[]	Number of lane sensor evaluation distances. Specifies the output dimensions of the block.
Const_MeasPoints_OutputDimension	[]	Number of measurement points. Specifies the output dimensions of the block.
Const_LanesLeft_OutputDimension	[]	Number of lanes to the left of the current lane to be considered by the lane sensor. Specifies the output dimensions of the block.
Const_LanesRight_OutputDimension	[]	Number of lanes to the right of the current lane to be considered by the lane sensor. Specifies the output dimensions of the block.

Note

Because these parameters specify the output dimensions, they must not be changed during run time. Therefore, their values are declared in the `asm_road_maneuver_ini.m` file, which is executed only during initialization.

LaneSensor_Dynamic_Signals bus

The following table shows the signals of the LaneSensor_Dynamic_Signals bus:

Name	Unit	Description
SFunction_Info	[]	Signal bus that contains general S-function information.
Network_Item	[]	Signal bus that contains information on the current network item.
CurrentLane	[]	Signal bus that contains information that is relevant for the current lane.
Lanes_Left	[]	Signal bus that contains information on the lanes left to the current lane.
Lanes_Right	[]	Signal bus that contains information on the lanes right to the current lane.
Measurement_Points	[]	Signal bus that contains information that is relevant to the measurement points.

SFunction_Info bus

The following table shows the signals of the SFunction_Info bus:

Name	Unit	Description
Sfcn_State	[0|1]	Indicates if the S-function is in a valid state: ▪ 0: S-function is in an invalid/error state. ▪ 1: S-function is in a valid state.
Input_Data_Valid	[0|1]	Indicates if the input data is valid: ▪ 0: Input data is invalid. ▪ 1: Input data is valid.
Data_Valid	[0|1]	Indicates if the input data and S-function are in valid states: ▪ 0: Data is invalid. ▪ 1: Data is valid.

Network_Item bus

The following table shows the signals of the Network_Item bus:

Name	Unit	Description
Network_Item_ID	[]	ID number of the road on which the lane sensor is located on. You can find the ID in the Road Generator in ModelDesk
Network_Item_Type	[]	Type of network item (as defined for the direct route) at the position of the lane sensor. This is independent of the driving direction on the route: ▪ 1: Object drives on the road element in the positive direction (according to the direct route definition). ▪ 2: Object drives on the road element in the negative direction (according to the direct route definition). ▪ 3: Junction.
Network_Item_Type_Absolute	[]	Type of network item at the position of the lane sensor. This depends on the driving direction on the route: ▪ 1: Object drives on the road element in the positive direction. ▪ 2: Object drives on the road element in the negative direction. ▪ 3: Junction.
Preferred_Lane_Direct	[]	Preferred lane in direct driving direction. This parameter is Zero if driving across a junction.
Preferred_Lane_Oncoming	[]	Preferred lane in oncoming driving direction. The signal is zero if you drive across a junction.

Note

All signals are vectors with dimensions that are equal to the number of the lane sensor evaluation distances specified in the parameters. If an evaluation distance is not enabled, the outputs are default values.

CurrentLane bus

The following table shows the signals of the CurrentLane bus:

Name	Unit	Description
Common	[]	Signal bus that containts the current lane information that is not relevant to right or left lane markings
LeftSide	[]	Signal bus that containts the current lane information on the left lane markings of the current lane.
RightSide	[]	Signal bus that containts the current lane information on the right lane markings of the current lane.

Common bus

The following table shows the signals of the Common bus:

Name	Unit	Description
Curvature	[1/m]	Curvature at the reference line of the current road.
LaneCurvature	[1/m]	Curvature at the center line of the current lane. The signal is -1 if there is no lane.
LaneDirection_Absolute	[]	Defined driving direction on the lane relative to the road element: ▪ -1: Object can drive on the lane in the negative direction of the road element. ▪ 0: Object can drive on the lane in any direction of the road element. ▪ 1: Object can drive on the lane in the positive direction of the road element. The signal is zero if there is no lane.
LaneDirection_Relative	[]	Defined driving direction on the lane relative to the vehicle: ▪ -1: Object can drive on the lane in opposite driving direction of the observed vehicle. ▪ 0: Object can drive on the lane in any direction of the observed vehicle. ▪ 1: Object can drive on the lane in the positive driving direction of the observed vehicle. The signal is zero if there is no lane.
LaneID_PrefLane	[]	Maneuver lane ID (corresponds to the preferred lane and driving direction) of the current lane.
LaneID_Absolute	[]	Road lane ID (as defined in the road) of the current lane.
LaneWidth	[m]	Lane width of the current lane. The signal is zero if there is no lane.

Note

All signals are vectors with dimensions that are equal to the number of the lane sensor evaluation distances specified in the parameters. If an evaluation distance is not enabled, the outputs are default values.

LeftSide bus

The following table shows the signals of the LeftSide bus:

Name	Unit	Description
Pos_Left	[m]	Position of the left lane marking in earth coordinate system at each enabled lane sensor evaluation distance[x,y,z].

Name	Unit	Description
Yaw_Angle_Left	[deg]	Yaw angle of the left lane marking in earth coordinate system.
Yaw_Angle_RelVehicle_Left	[deg]	Yaw angle of the left lane marking relative to the vehicle yaw angle.
Curvature_Left	[1/m]	Curvature of the left lane marking of the current lane.
MarkerType_Left	[]	Type of the left lane marking. The signal is -1 if there is no marker.
MarkerWidth_Left	[m]	Width of the left lane marking. The signal is zero if there is no marker
MarkerColor_Left	[]	Color of the left lane marking. The signal is -1 if there is no marker
LineID_1_Left	[]	Line ID (corresponds to the Line Sensor output) of the first line of left lane marking. The signal is -99 if there is no marker.
LineID_2_Left	[]	Line ID (corresponds to the Line Sensor output) of the second line of left lane marking. The signal is -99 if there is no marker or if the marker contains only one line.
Distance_Marker_Left	[m]	Lateral distance from lane sensor position to the left lane marking.

Note

All signals are vectors with dimensions that are equal to the number of the lane sensor evaluation distances specified in the parameters. If an evaluation distance is not enabled, the outputs are default values.

RightSide bus

The following table shows the signals of the RightSide bus:

Name	Unit	Description
Pos_Right	[m]	Position of the left lane marking in earth coordinate system at each enabled lane sensor evaluation distance[x,y,z].
Yaw_Angle_Right	[deg]	Yaw angle of the right lane marking in earth coordinate system.
Yaw_Angle_RelVehicle_Right	[deg]	Yaw angle of the right lane marking relative to the vehicle yaw angle.
Curvature_Right	[1/m]	Curvature of the right lane marking of the current lane.
MarkerType_Right	[]	Type of the right lane marking. The signal is -1 if there is no marker.
MarkerWidth_Right	[m]	Width of the right lane marking. The signal is zero if there is no marker.
MarkerColor_Right	[]	Color of the right lane marking. The signal is -1 if there is no marker.
LineID_1_Right	[]	Line ID (corresponds to the Line Sensor output) of the first line of right lane marking. The signal is -99 if there is no marker.
LineID_2_Right	[]	Line ID (corresponds to the Line Sensor output) of the second line of right lane marking. The signal is -99 if there is no marker or if the marker contains only one line.
Distance_Marker_Right	[m]	Lateral distance from the lane sensor position to the right lane marking.

Note

All signals are vectors with dimensions that are equal to the number of the lane sensor evaluation distances specified in the parameters. If an evaluation distance is not enabled, the outputs are default values.

Lanes_Left bus

The following table shows the signals of the Lanes_Left bus:

Name	Unit	Description
LaneID_PrefLane	[]	Lane ID of the lanes to the left of the current lane, described relative to the preferred lane.
LaneID_Absolute	[]	Road lane ID of the lanes to the left of the current lane, as defined in the road.
LaneDirection_Absolute	[]	Defined driving direction on the lanes to the left of the current lane, described relative to the road element: ▪ 1: Object can drive on the lane in the positive direction of the road element. ▪ -1: Object can drive on the lane in the negative direction of the road element. ▪ 0: Object can drive on the lane in any direction of the road element. The signal is zero if there is no lane.
LaneDirection_Relative	[]	Defined driving direction on the lanes to the left of the current lane, described relative to the vehicle: ▪ 1: Object can drive on the lane in the positive driving direction of the observed vehicle. ▪ -1: Object can drive on the lane in the opposite driving direction of the observed vehicle. ▪ 0: Object can drive on the lane in any direction of the observed vehicle. The signal is zero if there is no lane.
LaneWidth	[m]	Lane widths of the lanes to the left of the current lane. The signal is zero if there is no lane.
Lane_Curvature	[1/m]	Curvatures at center lines of lanes to the left of the current lane. The signal is -1 if there is no lane.
Pos_Left	[m]	Positions of the left lane markings of the lanes to the left of the current lane. Each position is described in the earth coordinate system at each enabled lane sensor evaluation distance [x,y,z].
Yaw_Angle_Left	[deg]	Yaw angles of the left lane markings of the lanes to the left of the current lane. Each yaw angle is described in the earth coordinate system.
Yaw_Angle_RelVehicle_Left	[deg]	Relative yaw angles of the left lane markings of the lanes to the left of the current lane. Each yaw angle is described relative to the vehicle yaw angle in the earth coordinate system.
Curvature_Left	[1/m]	Curvatures of the left lane markings of the lanes to the left of the current lane.
MarkerWidth_Left	[m]	Marker widths of the left lane markings of the lanes to the left of the current lane. The signal is zero if there is no lane marking.
MarkerType_Left	[]	Marker types of the left lane markings of the lanes to the left of the current lane. The signal is -1 if there is no lane marking.
MarkerColor_Left	[]	Marker colors of the left lane markings of the lanes to the left of the current lane. The signal is -1 if there is no lane marking.
LineID_1_Left	[]	Line IDs of the first line of the left lane markings of the lanes to the left of the current lane. The signal is -99 if there is no lane marking.
LineID_2_Left	[]	Line IDs of the second line of the left lane markings of the lanes to the left of the current lane. The signal is -99 if there is no lane marking or only one line.

Name	Unit	Description
Distance_Marker_Left	[m]	Lateral distances from the lane sensor position to the left lane markings of the lanes to the left of the current lane.
Pos_Right	[m]	Positions of the right lane markings of the lanes to the left of the current lane. Each position is described in the earth coordinate system at each enabled lane sensor evaluation distance [x,y,z].
Yaw_Angle_Right	[deg]	Yaw angles of the right lane markings of the lanes to the left of the current lane. Each yaw angle is described in the earth coordinate system.
Yaw_Angle_RelVehicle_Right	[deg]	Relative yaw angles of the right lane markings of the lanes to the left of the current lane. Each yaw angle is described relative to the vehicle yaw angle in the earth coordinate system.
Curvature_Right	[1/m]	Curvatures of the right lane markings of the lanes to the left of the current lane.
MarkerType_Right	[]	Marker types of the right lane markings of the lanes to the left of the current lane. The signal is -1 if there is no lane marking.
MarkerWidth_Right	[m]	Marker widths of the right lane markings of the lanes to the left of the current lane. The signal is zero if there is no lane marking.
MarkerColor_Right	[]	Marker colors of the right lane markings of the lanes to the left of the current lane. The signal is -1 if there is no lane marking.
LineID_1_Right	[]	Line IDs (correspond to the Line Sensor output) of the first line of the right lane markings of the lanes to the left of the current lane. The signal is -99 if there is no lane marking.
LineID_2_Right	[]	Line IDs (correspond to the Line Sensor output) of the second line of the right lane markings of the lanes to the left of the current lane. The signal is -99 if there is no lane marking or only one line.
Distance_Marker_Right	[m]	Lateral distances from the lane sensor position to the right lane markings of the lanes to the left of the current lane.

Note

All signals are vectors with dimensions that are equal to the number of the lane sensor evaluation distances multiplied by the number of considered lanes to the left of the current lane (as specified in the parameters). If an evaluation distance is not enabled, the outputs are default values.

Example Consider the following example for a vector with the following parameters of the road:

Parameter	Value
Number of considered lanes	3
Number of sensor evaluation distances	2
First sensor distance	a
Second sensor distance	b

Then: Lanes_Left.LaneWidth = [a_1, a_2, a_3, b_1, b_2, b_3]

Lanes_Right bus

The following table shows the signals of the Lanes_Right bus:

Name	Unit	Description
LaneID_PrefLane	[]	Lane ID of the lanes to the right of the current lane, described relative to the preferred lane.
LaneID_Absolute	[]	Road lane ID of the lanes to the right of the current lane, as defined in the road.
LaneDirection_Absolute	[]	Defined driving direction on the lanes to the right of the current lane, described relative to the road element: ▪ 1: Object can drive on the lane in the positive direction of the road element. ▪ -1: Object can drive on the lane in the negative direction of the road element. ▪ 0: Object can drive on the lane in any direction of the road element. The signal is zero if there is no lane.
LaneDirection_Relative	[]	Defined driving direction on the lanes to the right of the current lane, described relative to the vehicle: ▪ 1: Object can drive on the lane in the positive direction of the observed vehicle. ▪ -1: Object can drive on the lane in the opposite driving direction of the observed vehicle. ▪ 0: Object can drive on the lane in any direction of the observed vehicle. The signal is zero if there is no lane.
LaneWidth	[m]	Lane widths of the lanes to the right of the current lane. The signal is zero if there is no lane.
Lane_Curvature	[1/m]	Curvatures at the center lines of lanes to the right of the current lane. The signal is -1 if there is no lane.
Pos_Left	[m]	Positions of the left lane markings of the lanes to the right of the current lane. Each position is described in the earth coordinate system at each enabled lane sensor evaluation distance [x,y,z].
Yaw_Angle_Left	[deg]	Yaw angles of the left lane markings of the lanes to the right of the current lane. Each yaw angle is described in the earth coordinate system.
Yaw_Angle_RelVehicle_Left	[deg]	Relative yaw angles of the left lane markings of the lanes to the right of the current lane. Each yaw angle is described relative to the vehicle yaw angle in the earth coordinate system.
Curvature_Left	[1/m]	Curvatures of the left lane markings of the lanes to the right of the current lane.
MarkerType_Left	[]	Marker types of the left lane markings of the lanes to the right of the current lane. The signal is -1 if there is no lane marking.
MarkerWidth_Left	[m]	Marker widths of the left lane markings of the lanes to the right of the current lane. The signal is zero if there is no lane marking.
MarkerColor_Left	[]	Marker colors of the left lane markings of the lanes to the right of the current lane. The signal is -1 if there is no lane marking.
LineID_1_Left	[]	Line IDs of the first line of the left lane markings of the lanes to the right of the current lane. The signal is -99 if there is no lane marking.

Name	Unit	Description
LineID_2_Left	[]	Line IDs of the second line of the left lane markings of the lanes to the right of the current lane. The signal is -99 if there is no lane marking or only one line.
Distance_Marker_Left	[m]	Lateral distances from the lane sensor position to the left lane markings of the lanes to the right of the current lane.
Pos_Right	[m]	Positions of the right lane markings of the lanes to the right of the current lane. Each position is described in the earth coordinate system at each enabled lane sensor evaluation distance [x,y,z].
Yaw_Angle_Right	[deg]	Yaw angles of the right lane markings of the lanes to the right of the current lane. Each yaw angle is described in the earth coordinate system.
Yaw_Angle_RelVehicle_Right	[deg]	Relative yaw angles of the right lane markings of the lanes to the right of the current lane. Each yaw angle is described relative to the vehicle yaw angle in the earth coordinate system.
Curvature_Right	[1/m]	Curvatures of the right lane markings of the lanes to the right of the current lane.
MarkerType_Right	[]	Marker types of the right lane markings of the lanes to the right of the current lane. The signal is -1 if there is no lane marking.
MarkerWidth_Right	[m]	Marker widths of the right lane markings of the lanes to the right of the current lane. The signal is zero if there is no lane marking.
MarkerColor_Right	[]	Marker colors of the right lane markings of the lanes to the right of the current lane. The signal is -1 if there is no lane marking.
LineID_1_Right	[]	Line IDs (correspond to the Line Sensor output) of the first line of the right lane markings of the lanes to the right of the current lane. The signal is -99 if there is no lane marking.
LineID_2_Right	[]	Line IDs (correspond to the Line Sensor output) of the second line of the right lane markings of the lanes to the right of the current lane. The signal is -99 if there is no lane marking or only one line.
Distance_Marker_Right	[m]	Lateral distances from the lane sensor position to the right lane markings of the lanes to the right of the current lane.

Note

All signals are vectors with dimensions that are equal to the number of the lane sensor evaluation distances multiplied by the number of considered lanes to the right of the current lane (as specified in the parameters). If an evaluation distance is not enabled, the outputs are default values.

Example Consider the following example for a vector with the following parameters of the road:

Parameter	Value
Number of considered lanes	3
Number of sensor evaluation distances	2

Parameter	Value
First sensor distance	a
Second sensor distance	b

Then: Lanes_Left.LaneWidth = [a_1, a_2, a_3, b_1, b_2, b_3]

Measurement_Points bus

The following table shows the signals of the Measurement_Points bus:

Name	Unit	Description
Distance_MeasPnt_Marker_Left	[m]	Distance from each measurement point to the left marker in d-direction.
Distance_MeasPnt_Marker_Right	[m]	Distance from each measurement point to the right marker in d-direction.
Velocity_MeasPnt_Marker_Left	[m/s]	Velocity from each measurement point towards the left marker in d-direction.
Velocity_MeasPnt_Marker_Right	[m/s]	Velocity from each measurement point towards the right marker in d-direction.

Note

All signals are vectors with dimensions equal to the number of measurement points specified in the parameters. If a measurement point is not enabled, the outputs are default values.

Marker Colors/Types

You can define line properties in the ModelDesk Road Generator. The outputs are two-digit numbers that define the marker color and marker type. The outputs can be interpreted as follows:

- First digit: marker type/color of the first line
- Second digit: mraker type/color of the second line

The following table shows the possible line type values:

Value	Line Type
1	(Available only for the second line:) None
2	Solid
3	Dashed
4	Bicolored

The following table shows the possible line color values:

Value	Line Color
0	(Available only for the second line:) None
1	Custom
2	White
3	Yellow

Examples The following table shows some exemplary line type and color coding:

Number of lines	Line 1 type	Line 2 type	Line 1 color	Line 2 Color	Line type code	Line color code
1	Solid	–	White	–	21	20
1	Dashed	–	Yellow	–	31	30
2	Solid	Dashed	White	Yellow	23	23
2	Bicolored	Solid	Custom	White	42	12

Related topics

References

Lanesensor Dynamic Parameters (ModelDesk Parameterizing 📖)

Road

Description

With the **Road** block, you can use the complex road models generated in the ModelDesk Road Generator. The generated roads are used as parameters to the **Road** block, or can be downloaded directly, either to a Simulink simulation or to a dSPACE platform.

Basics on road models

Road network In ASM Vehicle Dynamics, the road contains junctions and road elements that are connected to form a road network. Each junction can be connected to several road elements via connection points. Road elements can also be connected to other road elements. Each road element is defined by a reference line in the horizontal plane. The reference line is created by consecutively adding segments of four different types. For the road network to be valid, the properties of the junction and road elements, such as lane properties or height, have to match at each connection point.

Trajectories Roads and junctions can contain multiple trajectory shapes for the definition of a vehicle path, which does not match the course of the reference line.

Routes The movement of vehicles on the road network is defined with routes, that are part of the road network. A route is an ordered list of road and junction elements that describes the path a vehicle takes through the road network. If roads or junction elements contain trajectory shapes, the corresponding route contains the information whether to take the default path or the path defined by a trajectory shape. The routes are selected in the Road Generator in ModelDesk.

Open/closed road Roads can be either open or closed. In the latter case the end point is connected to the start point and the vehicle can drive many laps around the road.

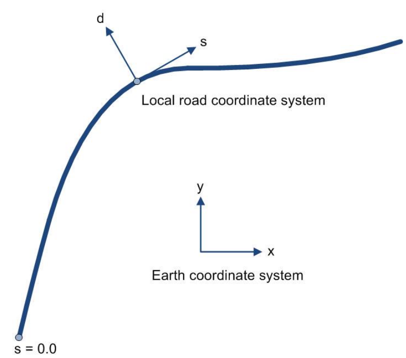
Driving direction As long as all road elements in a route can be driven in both directions, the route can also be driven in both directions. A road can be driven in both directions if it has driving lanes in both directions.

Junction elements On junction elements, the road properties are described as a function of the local x and y coordinates on the junction plate.

The following properties can be defined for junction elements:

- Shape of the junction border line
- Lane properties of connection points
- Height of the junction plate
- Special surface properties such as low friction areas or bumps
- Position markers

Road elements On road elements, the position of any point near the road is described by the distance s traveled along the reference line, measured from the start point of the road, and the distance d perpendicular to the reference line.



All properties of road elements are defined as a function of this distance s along the reference line, independently of the particular segments that define the horizontal profile.

The following properties can be defined for each road element:

- Height profile
- Lateral slope profile
- Lane sections
- Road shapes for additional road-independent lines, continuous objects and trajectory shapes
- Road scenery sections
- Special surface properties such as low friction areas or bumps

- Position markers can be defined in road and also in junction elements. The distance to position markers is evaluated as an event trigger in the Scenario Editor in ModelDesk.

These properties provide plenty of scope for defining and constructing user-specific test tracks. For details on the Road Generator, refer to Introduction to the Road Generator (ModelDesk Road Creation 📖).

It is also possible to create segment-based ASM roads out of measured road data using a MATLAB-based tool, named ASM RoadConverter. For details, refer to ASM RoadConverter (ASM User Guide 📖).

Road S-function

The road S-function carries out the mathematical calculations which are required in the vehicle dynamics simulation.

The road calculations are divided into several modular parts:

- Wheel contact point properties required for the vehicle dynamics calculations
- Vehicle initialization
- Preview point calculations required for the driver
- Traffic fellow calculations
- Lane sensor calculations
- Calculation of the distance between vehicles and position markers in the road network

  Each calculation evaluates the properties depending on the current position on the road or junction element.

Wheel contact point properties required for vehicle dynamics calculations

For the vehicle, the following variables are required at each tire-road contact point:

- Road height
- Friction coefficient
- Unit vector in the direction of road normal
- Surface condition

These values are calculated as a function of the x- and y-position of the tire contact point on the horizontal plane.

On junction elements, the properties are calculated in the local coordinate system of the junction plate.

Since for road elements the properties are defined as a function of the distance s along the road, the road s-function first calculates the distance s and the lateral offset d perpendicular to the road element that corresponds to the x- and y-position, i.e.:

$$[s_{CP}, d_{CP}] = f(x_{CP}, y_{CP})$$

where:

x_{CP} is the x-position of the contact point

y_{CP} is the y-position of the contact point

s_{CP} is the corresponding distance along the road

d_{CP} is the corresponding lateral offset to the road reference line

The road element properties at the tire-road contact point are then calculated using the s and d values, i.e.

$$\begin{aligned} z_{CP} &= (s_{CP}, d_{CP}) \\ e_{zCP} &= (s_{CP}, d_{CP}) \\ \mu_{CP} &= (s_{CP}, d_{CP}) \\ SC &= (s_{CP}, d_{CP}) \end{aligned}$$

where:

z_{CP} is the road height at the contact point

e_{zCP} is unit vector in direction of road normal at the contact point

μ_{CP} is the road friction coefficient the contact point

SC is the surface condition at the contact point, which is used to switch between tire parameters sets

Vehicle initialization

At simulation start the integrators of the vehicle coordinates in the vehicle dynamics model have to be initialized to a start position.

The start position is defined in the Scenario Editor in ModelDesk. The following values can be set:

- The route that the vehicle should drive on (each route is chosen via an ID number)
- The direction of travel on the route
- The initial distance along the route
- The initial lateral offset
- The initial lane
- The initial yaw angle relative to the route reference line
- The initial height (typically the tire radius)

  If the chosen route is not available in the road network, the initialization will fail and an error message will be printed to the log window. This can happen if the executed maneuver does not match the road.

  For example, maybe the route cannot be driven in the chosen direction because some of its road elements only have lanes.

Preview point calculations required for the driver

The road gets the starting point definition from the maneuver and calculates

- x and y coordinates of 10 preview points
- curvature of 5 preview points

If the ASM vehicle drives on a trajectory shape on a road element, the curvature will be calculated from the course of the road element and not from the course of the trajectory shape. As a result, if the vehicle drives on a trajectory shape and the option "Adapt velocity to road" has been checked in the ASM maneuver the speed of the ASM vehicle might not be decreased enough to keep the ASM vehicle on the trajectory shape. On junctions, the curvature is always calculated for the selected junction trajectory, regardless of whether it is the default trajectory or a trajectory shape .

Traffic calculations

This calculation is used only in combination with the ASM Traffic library. The traffic scenarios are defined using the Scenario Editor in ModelDesk.

The traffic scheduler provides the following signals to the road for all the fellow vehicles:

- The path s along the route
- The lateral offset d to the route reference line
- The index of the lane that the fellow vehicle should drive on
- The lateral velocity to be used in case of a lane change (optional)
- The force-to-road flag that defines whether the traffic fellow should be forced to drive on the road if the target lane is not available in the current lane section.

  Using these signals the Road calculates the following values:

  - The x, y and z coordinates of all traffic fellow vehicles
  - The three orientation angles of all traffic fellow vehicles
  - Additional information on the properties of the road or junction element at the traffic fellow vehicle positions

Interpretation of lateral deviation and lane index in the road model

For each lane section of a road element, it is possible to define preferred lanes for each direction of travel. A preferred lane is the lane that a vehicle will drive on by default.

The lane index selected in the Scenario Editor in ModelDesk is evaluated relative to the preferred lane in the current direction of travel, i.e., selecting the lane with index zero means that the vehicle will drive on the preferred lane. Selecting a positive lane index means that the vehicle will drive on a lane to the left of the preferred lane, as seen from the direction of travel.

For example, in right-hand traffic a vehicle will normally drive on the outer right-side lane of the road, hence the outer lanes should be chosen as the preferred lanes for each direction of travel. The vehicle will then drive on different lanes depending on the direction of travel on the road.

The lane index is usually given as integer number. However, in certain maneuvers non-integer indices may occur. Non-integer lane indices are interpreted as linear interpolation between the centers of the two lanes.

Example A road has three lanes, the center lane (index 0) with width 4 m, a first left lane (index 1) with width 3.5 m and a second left lane (index 2) with width 4 m.

Lane Index	Width	Lateral Position of Lane Center
0	5.0 m	0.00 m
1	3.5 m	4.25 m
2	4.0 m	8.00 m

A lane index of 0.8 would correspond to a lateral offset of
$d = 0.8 \cdot 4.25\ m = 3.40\ m$.

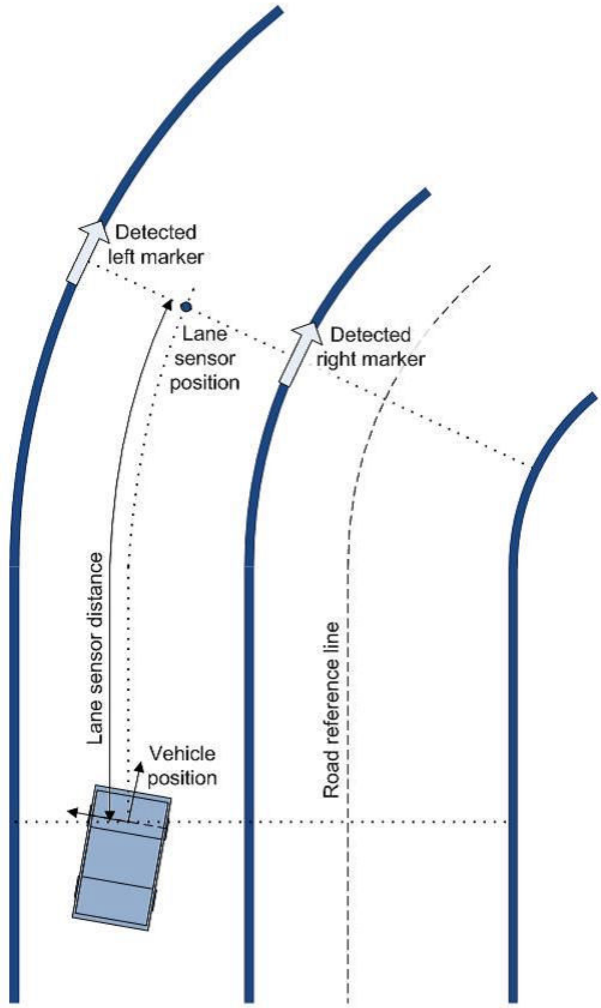
A lane index of 1.4 would correspond to a lateral offset of
$d = 4.25\ m + (8.00\ m - 4.25\ m) \cdot 0.4 = 5.75\ m$.

Lane Sensor Dynamic

The Lane Sensor Dynamic provides information on the number, position, size, and type of lanes and markers on the road at a certain position in front or behind the vehicle. The sensor position is defined with the distance to the vehicle, which is always measured in parallel to the road reference line.

An instance of the LANESENSOR_DYNAMIC_PARAMETERS block provides up to five lane sensor positions. You have to adjust the parameters of the LANESENSOR_DYNAMIC_OUTPUT block accordingly.

The lane sensor detects the right and left markers of the lane on which the vehicle is currently driving from all available road markers. For these two markers, the sensor calculates the x-, y- and z-position, the yaw angle, and provides information on the marker width, type, color, and the corresponding line IDs as given by the line sensor. If the vehicle is outside of the road, then only the marker at the road border is found and returned. If no markers are available, the sensor outputs default values.



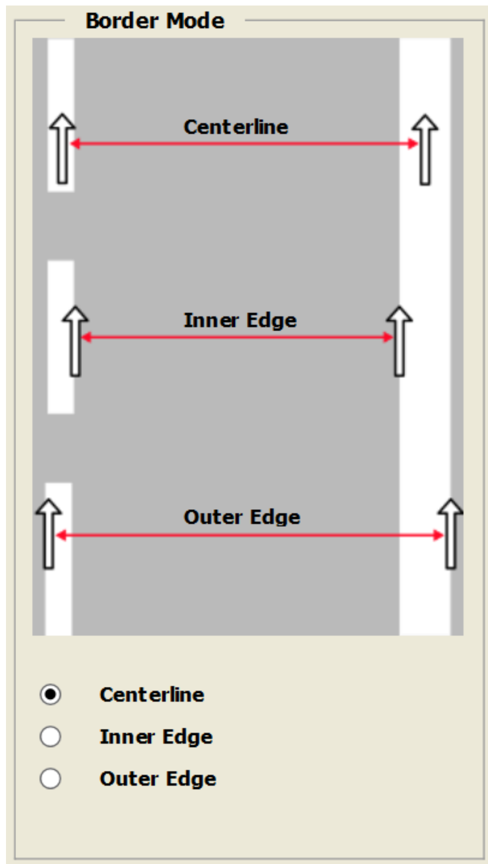
In addition to the two markers nearest to the right and to the left of the vehicle, the lane width, marker width, type, color, and line IDs information is available for a configurable number of lanes left and right of the current lane.

The Lane Sensor Dynamic interpolates the current lane information on junctions by using the lane widths of the incoming and outgoing roads. The sensor can optionally be configured to output default values on junctions.

Border mode You can configure the lane borders interpretation in the lane sensor ModelDesk parameter page. The following modes are available:

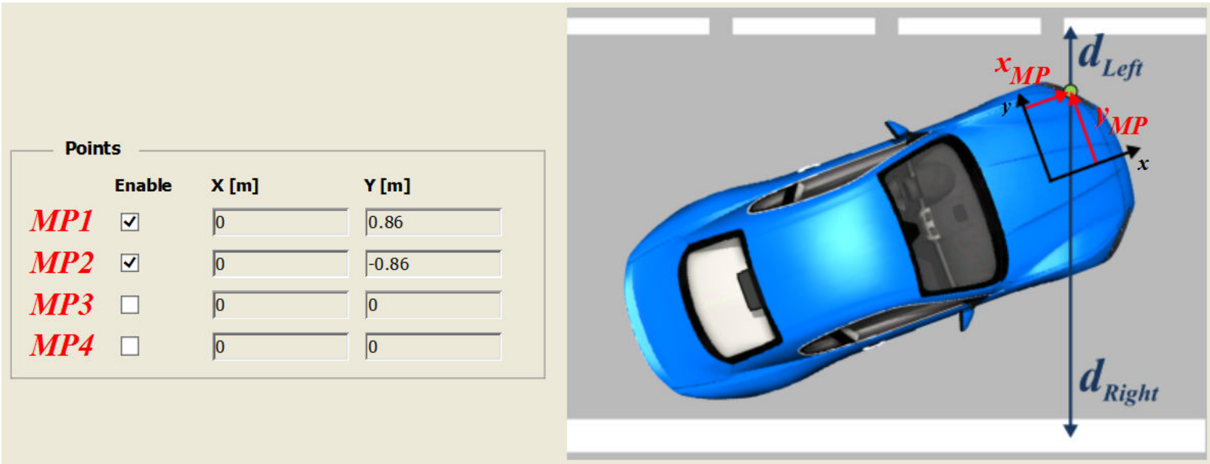
1. Lanes borders are at the centerline of the lane markings.
2. Lane borders are at the inner edge of the lane markings.
3. Lane borders are at the outer edge of the lane markings.

This setting affects all calculations executed by the sensor.



Measurement points In addition to the lane information, the lane sensor also calculates the distance and velocity to the lane markings in the d-direction from configurable points on the vehicle. The position of the measurement points is specified in vehicle coordinates. For each point, the distance and velocity to both the right and left marker are calculated. By default, two measurement points are specified and enabled: one at the outer edge of each forward wheel.

An instance of the LANESENSOR_DYNAMIC_PARAMETERS block provides up to five measurement points. You have to adjust the parameters of the LANESENSOR_DYNAMIC_OUTPUT block accordingly.



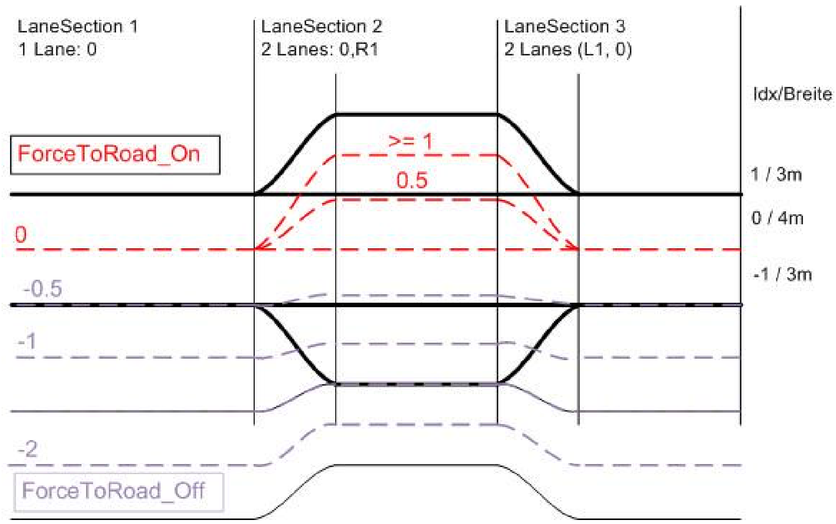
Force-to-road flag

It can happen in maneuvers and scenarios that the target lane is not available in the lane section that the vehicle is driving on. If the lane is defined as non-drivable, it will be treated as a non-available lane.

Example In the current lane section the road has a center lane (index 0) and one lane to the right (lane index -1). If the maneuver sets the target lane index to 1, then the target lane is not available in this section.

In this case there are two options for the vehicle behavior that can be chosen with the force-to-road flag.

Force-to-road flag	Behavior
Off (0)	The vehicle drives outside of the road or on a lane with an opposite direction of travel if available. Outside the road the lateral offset is linearly extrapolated from the width of the outermost lane.
On (1)	The vehicle drives on the outermost available drivable lane in the direction of travel of the vehicle. The vehicle cannot drive on non-drivable lanes or bypass them in order to reach the other side.



Example In the current lane section the road has a center lane (index 0) of width 4 m and one lane to the right (lane index -1) of width 3.5 m. Depending on the target lane index and the force-to-road flag the resulting lateral offset is then calculated as follows:

Target Lane Index	Force-to-road = 0	Force-to-road = 1
-2.0 (lane not available)	-7.25 m (extrapolated with width 3.5m)	3.75 (force vehicle to road)
-1.5 (lane not available)	-5.5 m (extrapolated with width 3.5 m)	3.75 m (force vehicle to road)
-1.0 (lane available)	-3.75 m	3.75 m
-0.5 (lane available)	-1.875 m (interpolated)	
0.0 (lane available)	0.00 m	0.00 m
1.0 (lane not available)	+4.00 m (extrapolated with width 4 m)	0.00 m (force vehicle to road)
2.0 (lane not available)	+8.00 m (extrapolated with width 4 m)	0.00 m (force vehicle to road)

Inports The following table shows the inports:

Name	Unit	Description
Angle_Yaw_Vehicle_CoorSys_E	[deg]	Vehicle yaw angle
Angle_Yaw_Vehicle_Rel_Init	[deg]	Initial vehicle yaw angle
d_Fellows_Ref	[m]	Lateral position on the route for fellow vehicles
d_Ref_Preview	[m]	Target lateral position on the route for preview points
d_Vehicle_Init	[m]	Initial lateral position on the route
Dist_Def_Mnv	[]	Info vector from the maneuver that defines two objects (fellows, vehicle or marker). The road calculates the distance between the two objects.
Dist_Def_Trf	[]	Info vector from the traffic scheduler that defines between which objects (fellows, vehicle, marker) the distances must be calculated.

Name	Unit	Description
Dist_LaneSensor	[m]	Distance of the lane sensor along the road relative to the ASM vehicle. This input refers to the Lane Sensor in former versions. Refer to Lane Sensor on page 157.
FellowsUsed	[0|1]	Enable flags for all fellow vehicles
ForceToRoad_Fellows_Ref	[0|1]	Enable flag of force-to-road functionality for all fellow vehicles
ForceToRoad_Ref	[0|1]	Enable flag of force-to-road functionality for ASM vehicle
LaneIdx_Fellows_Ref	[]	Target lane index for all fellow vehicles
LaneIdx_Ref_Preview	[]	Target lane index for preview points of the ASM vehicle
LaneIdx_Vehicle_Init	[]	Initial lane index for the ASM vehicle
LaneSensor_Border_Mode	[0|1|2]	Switch to define how to interprete lane borders: ▪ 0: Lane borders are at the centerline of the lane markings. ▪ 1: Lane borders are at the inner edge of the lane markings. ▪ 2: Lane borders are at the outer edge of the lane markings.
LaneSensor_Enable	[0|1]	Lane sensor enable flag. This input refers to the Lane Sensor in former versions. Refer to Lane Sensor on page 157.
LaneSensor_Enable_Evaluation	[]	Flag to enable or disable the lane sensor evaluation at each lane sensor evaluation distance
LaneSensor_Enable_Junction_Evaluation	[0|1]	Flag to determine if lane sensor outputs on junctions are evaluated: ▪ 0: No sensor output on junctions. ▪ 1: The current lane information is interpolated by using the previous and the next road.
LaneSensor_Enable_MeasPoint	[]	Flag to enable or disable the evaluation of each measurement point
LaneSensor_Evaluation_Distance	[m]	Vector of distances from vehicle in s-direction at which the Lane Sensor Dynamic information is evaluated
LaneSensor_MeasPoint_CoorSysE	[m]	Measurement point positions in the earth coordinate system [x,y]
LaneSensor_Num_Lanes	[]	Number of lanes that are to the right and to the left of the current lane to be considered by the lane sensor [Left,Right].
Pos_CP_CoorSys_E[:][x;y]	[m]	Position of contact points in earth coordinates
Pos_x_Vehicle	[m]	Vehicle position in x direction in earth coordinates
Pos_y_Vehicle	[m]	Vehicle position in y direction in earth coordinates
Pos_z_Vehicle_Rel_Init	[m]	Initial z-position of the ASM vehicle
RelativeLane_Mode	[0|1]	Flag which defines the reference lane used for the target lane index **LaneIdx_Ref_Preview[]** of the preview points of the ASM vehicle. ▪ 0: Center lane (lane zero). ▪ 1: Preffered lane.
RelativeLane_Mode_Fellows	[0|1]	Flag which defines the reference lane used for the target lane index **LaneIdx_Ref_Preview[]** of the preview points of all fellow vehicles: ▪ 0: Center lane (lane zero). ▪ 1: Preffered lane.

Name	Unit	Description
RoadCtrl	[]	Road block execution control input
Route Direction_Fellows_Init	[-1|1]	Direction on the route for all fellow vehicles
Route_Direction_Init	[]	Direction on the route for the ASM vehicle
Route_Id_Fellows_Init	[-1|1]	ID number of the route that the fellow vehicles will drive on
Route_Id_Init	[]	ID number of the route that the ASM vehicle will initialize and drive on
s_Fellows_Ref	[m]	Distance s along the route for all fellow vehicles
s_Preview	[m]	Vector with preview distances for lateral control and vehicle reference velocity calculation
s_Vehicle_Init	[m]	Initial distance s along the route for ASM vehicle
Sw_RoadSelect	[]	Select switch to chose one of the six roads that are set as block parameters
v_Lat_Fellows_Ref	[m/s]	Lateral velocity during lane change for all fellow vehicles
v_Lat_Ref	[m/s]	Lateral velocity during lane change of ASM vehicle

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 🕮).
ASMVehicle	[]	Signal bus containing signals of the ASM vehicle
ContactPoints	[]	Signal bus containing road property signals, such as height and friction for all wheel contact points
Info	[]	Signal bus containing general information on the properties and state of the loaded road
Init	[]	Signal bus containing the initial position and angle of the ASM vehicle
LaneSensor	[]	Signal bus containing all lane sensor signals. This output refers to the Lane Sensor in former versions. Refer to Lane Sensor on page 157.
LaneSensor_Data	[]	Signal with the reference to the Lane Sensor Dynamic data that is defined in the road network.
Preview	[]	Signal bus containing preview signals for the driver
TrafficFellows	[]	Signal bus containing traffic fellow signals
TrafficObject_Data	[]	Signal with reference to the traffic object data defined in the road network
UserSignals	[]	Signal bus containing user signals of the loaded road

Note

All signals are contained in the ASMSignalBus. The following tables show the road output signal groups.

ASMVehicle The ASMVehicle group contains road signals and properties at the position of the vehicle.

Name	Unit	Description
Angle_Yaw_Veh_Road	[deg]	Yaw angle of the reference line of the road at the current position of the vehicle
Curv_Veh_Road	[1/m]	Curvature of the road reference line at the current position of the vehicle
d_Total_Veh_Road	[m]	Lateral distance of the vehicle regarding the preferred lane
LaneIdx_Total_Veh_Road	[]	Lane index of the vehicle on the road relative to the preferred lane in the current direction of travel
Pos_x_Ref_Veh_Road	[m]	x-position of the reference line of the road at the current position of the vehicle
Pos_y_Ref_Veh_Road	[m]	y-position of the reference line of the road at the current position of the vehicle
s_Total_Veh_Road	[m]	The total distance along the route
s_Veh_Road	[m]	The distance s along the route at the position of the vehicle. Reset to zero after each turn on a closed route.
Slope_Lateral_Veh_Road	[deg]	Lateral slope of the road at the current position of the vehicle
Slope_Veh_Road	[%]	Slope of the road at the current distance s of the vehicle
v_wind_Coor_Sys_E[x;y;z]	[m/s]	Velocity of the wind in the earth coordinate system

Info The road info group contains signals that provide information on the state and properties of the road.

Name	Unit	Description
d_Veh_Ref_Line	[m]	Lateral distance of the vehicle regarding the road reference line
Dist_Border_Left	[m]	Distance from the vehicle reference coordinate system to the left road border. A negative value means the vehicle has left the road from the left side.
Dist_Border_Right	[m]	Distance from the vehicle reference coordinate system to the right road border. A negative value means the vehicle has left the road from the right side.
Dist_Response_Maneuver	[m]	Distance between the two objects (vehicle or position marker) selected at the Dist_Def_Mnv input The distance in the xy plane (Distance_xy) and the distance in z (Distance_z) is calculated separately
Flag_RoadClosed_Veh	[0|1]	This flag is set to 1 if the road is closed
LaneSection_Veh	[]	The index of the lane section of the road element that the vehicle is driving on. Zero if driving on junction.
Network_Item_Id_Veh	[]	ID number of the road or junction element the ASM vehicle is driving on. The ID is visible in the Road Generator in ModelDesk.
Network_Item_Type_Veh	[1|2|3]	Type of network item at the position of the ASM vehicle: ▪ 1: road element in positive direction ▪ 2: road element in negative direction ▪ 3: junction element
Network_Load_Idx	[]	Signal indicating the load state of the network during initialization phase
RoadLength_Veh	[m]	The length of the loaded road
RoadSegment_Veh	[]	The index of the horizontal profile segment of a road element that the vehicle is driving on
RoadState	[]	The road S-function execution state

Name	Unit	Description
RoadWidth_Veh	[]	The total road width, including the outside borders but without the batter, at the position of the vehicle. Zero if driving on a junction.
Speed_Limit_Veh	[km/h]	Speed limit as defined in the scenery section of the road element at the position of the ASM vehicle

ContactPoints The road contact point group contains signals that define the properties of the road at the contact points of the wheels. All signals are vectors whose size depends on the number of wheel position vectors connected to the inports.

Name	Unit	Description
Fric_Coeff_CP[:]	[]	Friction coefficient of the road at the contact point.
Pos_z_CP[:]	[m]	Vertical (z) position of the contact point in the earth coordinate system.
Sw_Tire_Parameter_Set_CP[:]	[1|2|3|4]	Switch to select one of four sets of tire parameters.
UnitVec_z_CP[:][x;y;z]	[m]	Unit vector perpendicular to the road surface at the contact point, described in earth coordinate system

Init The road init group contains signals that define the initial position and orientation angles of the vehicle on the road.

Name	Unit	Description
Angle_Vehicle_Init_CoorSys_E[x;y;z]	[rad]	Initial vehicle orientation angles
Pos_Vehicle_Init_CoorSys_E[x;y;z]	[m]	Initial vehicle position

Preview The road preview group contains preview signals for the driver.

Name	Unit	Description
Curv_Preview_Road	[1/m]	Curvature of the road at the longitudinal postions of the preview points. Also depends on the current lateral offset and lane index.
Pitch_Preview_Road	[deg]	Pitch angle of the road at the positions of the preview points
Pos_x_Preview_Road	[m]	Vector of the previewed road positions in the x-direction (earth coordinates)
Pos_y_Preview_Road	[m]	Vector of the previewed road positions in the y-direction (earth coordinates)
Pos_z_Preview_Road	[m]	Vector of the previewed road positions in the z-direction (earth coordinates)
Roll_Preview_Road	[deg]	Roll angle of the road at the positions of the preview points
Speed_Limit_Preview_Road	[km/h]	Speed limit at the positions of the preview points, as defined in the scenery section of the road element. On junctions, the last value is kept constant.
x_Preview_DriverCurv	[m]	Vector of the previewed road positions in the x-direction (earth coordinates) for curvature calculation.
y_Preview_DriverCurv	[m]	Vector of the previewed road positions in the y-direction (earth coordinates) for curvature calculation.

TrafficFellows The fellow vehicles group contains the position, orientation and road properties for all fellow vehicles. Each signal is a vector whose size is equal to the number of fellow vehicles.

Name	Unit	Description
Angle_Pitch_Fellows_Road	[deg]	Pitch angle of the fellow vehicles in earth coordinates
Angle_Roll_Fellows_Road	[deg]	Roll angle of the fellow vehicles in earth coordinates
Angle_Yaw_Fellows_Road	[deg]	Yaw angle of the fellow vehicles in earth coordinates
Curv_Fellows_Road	[1/m]	Curvature of the road at the fellow position
d_Fellows_RefLine	[m]	Lateral distance to the road reference line
d_Total_Fellows_Road	[m]	Lateral distance to the preferred lane of the road
Dist_Response_Traffic	[m]	Distances between the objects (fellows, vehicle, marker) selected at the Dist_Def_Trf input. The distances in the xy plane (Distance_xy) and the distances in z (Distance_z) are provided separately.
Flag_RouteClosed_Fellows[	[0|1]	Flag indicating if the assigned route is a closed route
LaneIdx_Total_Fellows_Road	[]	Lane index corresponding to the lateral distance to the road reference line
LaneSection_Fellows	[]	Index of the lane section that the traffic fellow is driving on
Network_Item_Id_Fellows	[]	ID number of the road or junction element that the traffic fellow is driving on. The ID is visible in the Road Generator in ModelDesk.
Network_Item_Type_Fellows	[1|2|3]	Type of network item at the position of the traffic fellow: ▪ 1: road element in positive direction ▪ 2: road element in negative direction ▪ 3: junction element
Pos_x_Fellows_Road	[m]	x-position of the fellow vehicles in earth coordinates
Pos_y_Fellows_Road	[m]	y-position of the fellow vehicles in earth coordinates
Pos_z_Fellows_Road	[m]	z-position of the fellow vehicles in earth coordinates
RoadSegment_Fellows	[]	Index of the horizontal profile segment that the traffic fellow is driving on
RouteLength_Fellows	[m]	The length of the route the fellow vehicle is driving on.
Speed_Limit_Fellows_Road	[km/h]	The speed limit at the position of the fellow vehicles, as defined in the scenery section of the road element. On junctions the last value is kept constant.

UserSignals The user signals contains 10 signals defined with the Road Generator in ModelDesk, refer to How to Configure User Signals for a Road (ModelDesk Road Creation 📖).

Name	Unit	Description
User1	[]	Constant user signal 1
…	…	…
User10	[]	Constant user signal 10

Parameters

The following table shows the parameters:

Name	Unit	Description
Road_ModelDesk	[]	This vector contains the road definition. This parameter is accessed by ModelDesk when working offline in Simulink
Road_2-6	[]	Five additional road vectors contain the definition of five roads. ModelDesk does not access these parameters, but you can select one of them using the Sw_RoadSelect switch.

Related topics

Basics

ASM RoadConverter (ASM User Guide 🕮)
Basics of the Scenario Editor (ModelDesk Scenario Creation 🕮)
Introduction to the Road Generator (ModelDesk Road Creation 🕮)
Introduction to the Scenario Editor (ModelDesk Scenario Creation 🕮)

Blocks from Former Versions

Introduction

The following topics provide information on blocks that were used in former library versions.

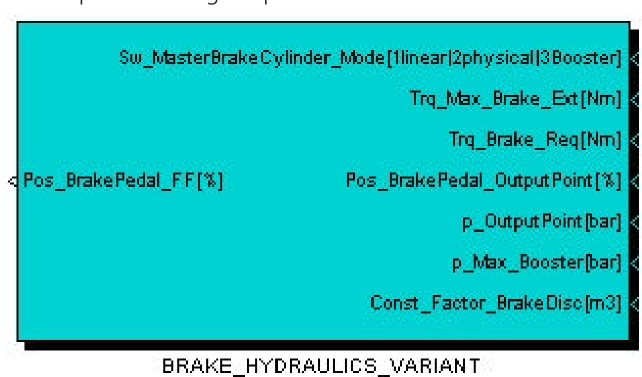
Where to go from here

Information in this section

Brake Hydraulics Variant_5_0

Description

The brake hydraulics variant contains the variant-dependent inverse model of the brake system. It provides the feed forward control for the brake pedal according to a required braking torque.



Several brake system models can be used in the vehicle dynamics model, for example, a brake disc model or a detailed brake hydraulics model.

The brake hydraulics variant subsystem contains a simplified inverse model of the brake system. It calculates the required brake pedal position according to a required braking torque.

The inports, outports, and parameters are shown below. A different brake variant may be used in your model.

Inports

The following table shows the inports:

Name	Unit	Description
Const_Factor_BrakeDisc	[m^3]	Factor to calculate the brake disc torque from brake disc pressure.
p_Max_Booster	[bar]	Maximum master brake cylinder pressure that can be generated by the booster and the brake pedal
p_OutputPoint	[bar]	Brake pressure master brake cylinder output point
Pos_BrakePedal_OutputPoint	[%]	Brake pedal position at output point
Sw_MasterBrakeCylinder_Mode	[1|2|3]	Master brake cylinder mode ▪ 1: Modeled linearly ▪ 2: Modeled physically ▪ 3: Brake booster
Trq_Brake_Req	[Nm]	Braking torque requested from feed forward controller
Trq_Max_Brake_Ext	[Nm]	Maximum braking torque at 100% brake pedal position

Note

The maximum braking torque is provided as a signal for each wheel in the ASM_VehicleDynamics model. The sum of the signals is used as input to the brake hydraulics variant block.

Outports

The following table shows the outports:

Name	Unit	Description
Pos_BrakePedal_FF	[%]	Brake pedal position corresponding to requested braking torque

Parameters

The following table shows the block parameters:

Name	Unit	Description
Map_BrakeForce_Inv_Normalized	[%]	Inverse normalized BrakeForce Map [%] = f(Trq_Req/Trq_Max) for physical modeling of master brake cylinder

Related topics

References

Master Brake Cylinder (ModelDesk Parameterizing)

Brake Pneumatics Variant_4_0

Description

The brake pneumatics variant contains the variant-dependent inverse model of the brake system. It provides the feed forward control for the brake pedal according to a required braking torque.

The BRAKE_PNEUMATICS_VARIANT block contains a simplified inverse model of the brake system. It calculates the required brake pedal position according to a required braking torque.

Note

There is a similar block for brake hydraulics, BRAKE_HYDRAULICS_VARIANT, which is used as a default in the demo models. To make it easy to replace the brake pneumatics with the brake hydraulics in the demo models, both blocks have the same input signals. Except for Trq_Brake_Req and Trq_Max_Brake, these signals do not have a function and are only used for compatibility with BRAKE_HYDRAULICS_VARIANT.

Inports

The following table shows the inports:

Name	Unit	Description
Const_Factor_BrakeDisc	[m^3]	No meaning. Only for compatibility with brake hydraulics.
p_Max_Booster	[bar]	No meaning. Only for compatibility with brake hydraulics.
p_OutputPoint	[bar]	No meaning. Only for compatibility with brake hydraulics.

Name	Unit	Description
Pos_BrakePedal_OutputPoint	[%]	No meaning. Only for compatibility with brake hydraulics.
Sw_MasterBrakeCylinder_Mode	[1|2|3]	No meaning. Only for compatibility with brake hydraulics.
Trq_Brake_Req	[Nm]	Braking torque requested by feed forward controller
Trq_Max_Brake	[Nm]	Maximum braking torque at 100% brake pedal position

Outports

The following table shows the outports:

Name	Unit	Description
Pos_BrakePedal_FF	[%]	Brake pedal position corresponding to requested braking torque

Parameters

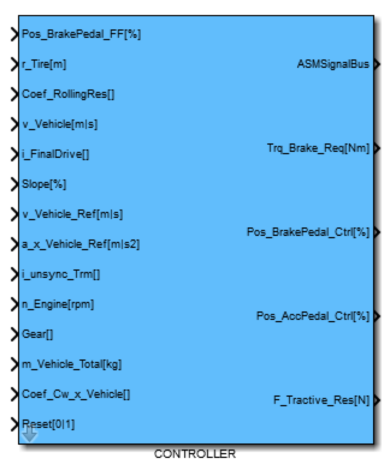
The following table shows the parameters:

Name	Unit	Description
Map_BrakeForce_Inv_Normalized	[%]	Inverse normalized BrakeForce Map = f(Trq_Req/Trq_Max)

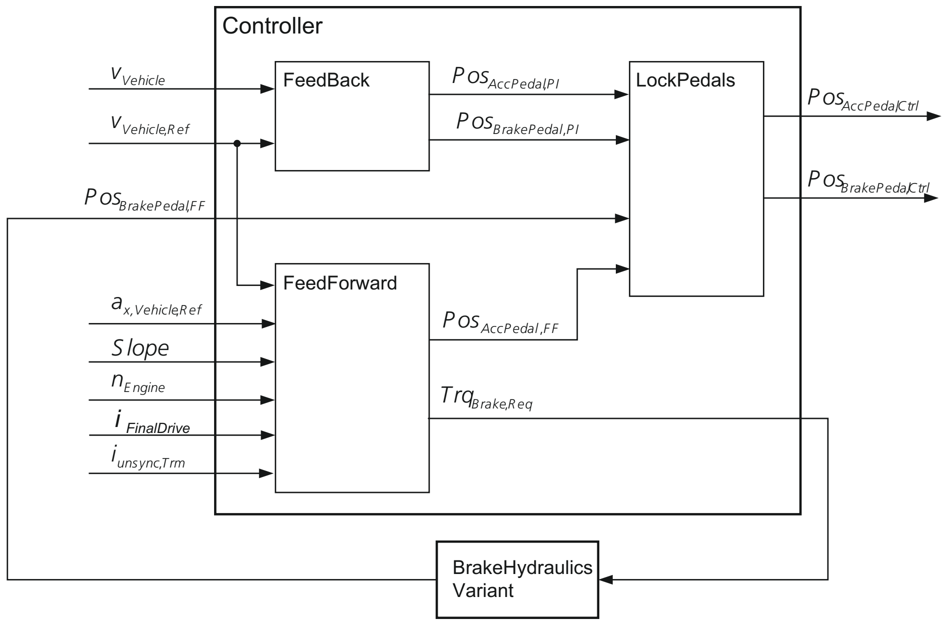
Controller_10_0

Description

The controller (CONTROLLER block) controls accelerator pedal and brake pedal according to a reference velocity. An internal logic ensures that only one pedal is actuated at a time. The controller comprises the variant-independent part of the feed forward controller and a feedback controller.



The following illustration shows the main signal flow in the controller subsystem and the interconnection to the variant-dependent inverse brake hydraulics model which is outside the controller subsystem.



FeedForward

Using a simplified inverse vehicle model, the feed forward controller calculates the requested accelerator pedal position and brake pedal position according to a reference velocity and reference acceleration.

Internally, the feed forward subsystem comprises the calculation of the requested tire torque, the calculation of the total gear ratio, the calculation of the available engine braking torque, and an inverse engine map.

First the required tire torque is calculated from reference velocity and reference acceleration. Afterwards, the required engine torque and required braking torque are calculated. Two options are implemented for this calculation. The option to be used, is defined by the **Sw_FeedForward_Mode** parameter.

1. Torque based calculation: The required engine and brake torques are calculated using the current transmission ratio. Using this calculation gear changes result in steps in the required engine torque and thus in the accelerator pedal position. Steps in the accelerator pedal after gear changes may cause problems in closed loop simulation with transmission ECUs.
2. Power based approach: The required engine and brake torques are calculated via a power-based approach. In this calculation direct steps in the accelerator pedal position due to gear changes are avoided.

Using the inverse engine map, the accelerator pedal position is calculated from required engine torque and engine speed. The brake pedal position is calculated from requested braking torque in the external variant-dependent block BRAKEHYDRAULICS_VARIANT.

The feed forward controller actuates only one pedal, accelerator or brake, at a time.

Feedback

The feedback controller controls accelerator pedal and brake pedal to decrease the error between reference and vehicle velocity by closing the control loop. The feedback controller comprises two PI controllers, one for the accelerator pedal and one for the brake pedal. The parameters of the PI controllers can be set via mask parameters. An anti-windup functionality is implemented in the integrators.

To ensure anti-windup, the integrator for accelerator pedal control is reset if one of the following conditions applies.

- The vehicle velocity is very small. This functionality is used during vehicle start-up.
- The difference between reference and vehicle velocity is large.
- A gearshift is performed.
- The brake pedal is activated.

To ensure anti-windup, the integrator for brake pedal control is reset if one of the following conditions applies.

- The difference between vehicle velocity and reference velocity is large.
- The accelerator pedal is activated.

LockPedals

The feed forward and feedback controllers provide accelerator pedal and brake pedal positions. The resulting accelerator and brake pedal positions are calculated as the sum of the two controller signals.

In some cases it can happen that the preceding controllers actuate both pedals simultaneously. The LockPedals subsystem contains an internal logic which ensures that only one pedal is actuated at a time. The logic has the following features.

- If feed forward is present for the accelerator pedal, it is activated. If feed forward is present for the brake pedal, it is actuated. The feed forward controller provides feed forward for only one pedal at a time.
- If no feed forward is present, the brake pedal has priority for reference accelerations below a certain threshold. The accelerator pedal has priority if the reference acceleration exceeds a certain value.

Inports

The following table shows the inports:

Name	Unit	Description
a_x_Vehicle_Ref	[m/s^2]	Reference vehicle acceleration
Coef_Cw_x_Vehicle	[]	Longitudinal aerodynamic coefficient Cw
Coef_RollingRes	[]	Rolling resistance coefficient
Gear	[]	Current gear
i_FinalDrive	[]	Total transmission ratio of final drive line (between transmission output and wheel speed)
i_unsync_Trm	[]	Current transmission ratio

Name	Unit	Description
m_Vehicle_Total	[kg]	Total vehicle mass (with additional loads)
n_Engine	[rpm]	Engine speed
Pos_BrakePedal_FF	[%]	Brake pedal position from feed forward controller
r_Tire	[m]	Dynamic tire radius
Reset	[0|1]	Reset of all integrators and internal states
Slope	[%]	Current road slope
v_Vehicle	[m/s]	Vehicle velocity
v_Vehicle_Ref	[m/s]	Reference vehicle velocity

Note

Some parameters for the controller model are provided as signal lines from the vehicle model. The tire radius r_tire and the rolling resistance coefficient Coef_RollingRes are provided as signals for each wheel. The controller uses averaged values of these signals.

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 📖).
F_Tractive_Res	[N]	Tractive resistance force
Pos_AccPedal_Ctrl	[%]	Accelerator pedal position
Pos_BrakePedal_Ctrl	[%]	Brake pedal position
Trq_Brake_Req	[Nm]	Requested braking torque

Parameters

The following table shows the block parameters:

Name	Unit	Description
Const_A_x_Vehicle	[m^2]	Longitudinal shadow area of the vehicle
Const_Dens_Air	[kg/m^3]	Air density
Const_I_Gain_Acc	[]	Controller gain ki for accelerator pedal controller (integral)
Const_I_Gain_Brake	[]	Controller gain ki for brake pedal controller (integral)
Const_P_Gain_Acc	[]	Controller gain kp for accelerator pedal controller (proportional)
Const_P_Gain_Brake	[]	Controller gain kp for brake pedal controller (proportional)
Map_Trq_Engine_Inv	[%]	Inverse engine map: Pos_AccPedal = f(n_Engine, Trq_Engine_Req)
Map_Trq_Fric_Engine	[Nm]	Engine friction torque = f(n_Engine)
Sw_FeedForward_Mode	[]	Feed forward calculation mode ▪ 1: Torque based

Name	Unit	Description
		▪ 2: Power based (default)

Related topics

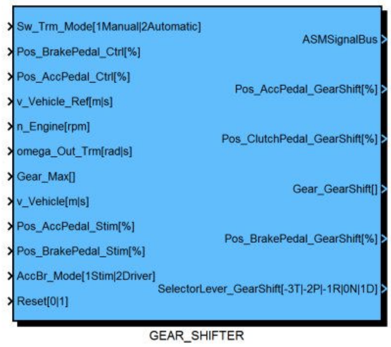
References

Controller V10 (ModelDesk Parameterizing 📖)

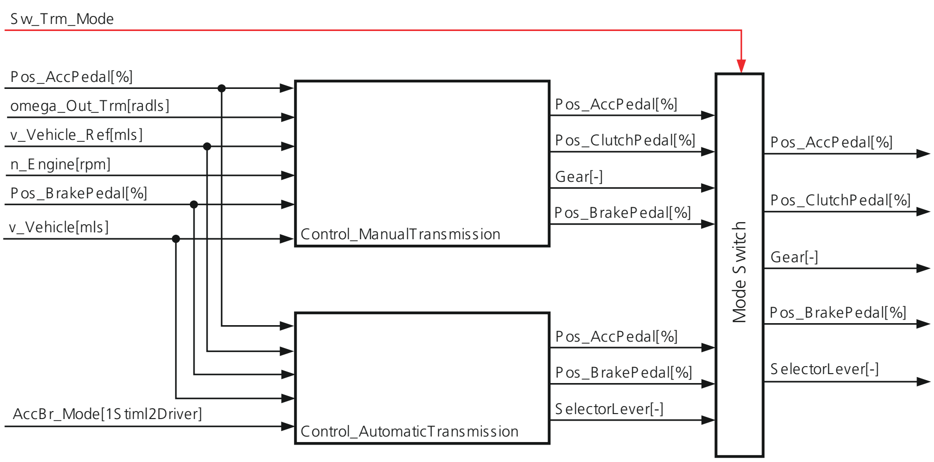
Gear Shifter (Version 9.0 and Earlier)

Description

The GEAR_SHIFTER block shifts the gears according to the current driving situation for an automatic or manual transmission.



The following illustration presents an overview of the different modes of the gearshift.



The gearshift can handle two different shift modes

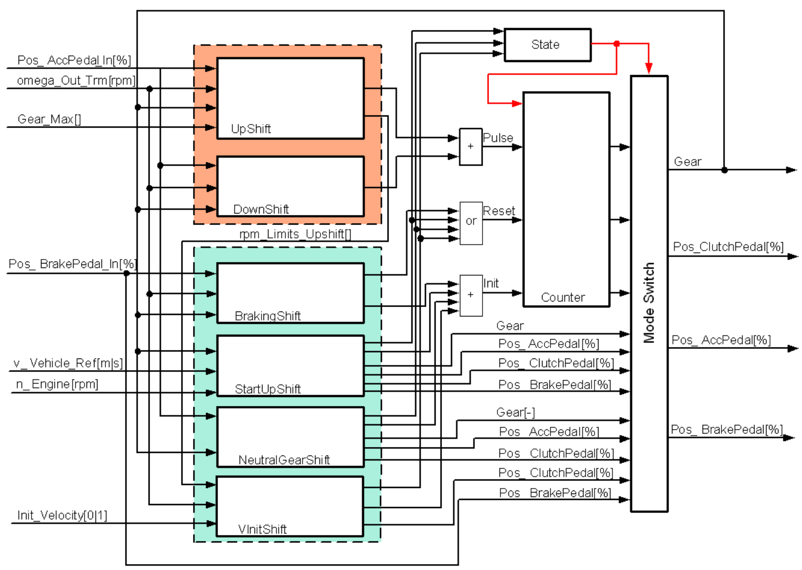
- Shifting of a manual transmission
- Shifting of an automatic transmission

Shifting the manual transmission is done by the Control_ManualTransmission function, which engages the correct gear for the current driving situation.

Shifting automatic transmission is done by engaging the driving position of an automatic transmission if the reference velocity is greater than zero. Neutral position is engaged if the reference velocity is set to zero. Shifting itself is done by the Soft ECU transmission. This situation is handled by the Control_AutomaticTransmission function.

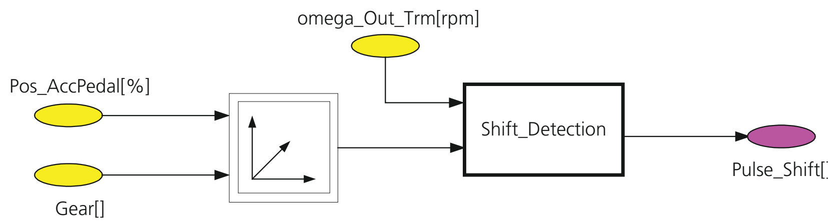
In automatic transmission mode the gearshift sets the selector lever position. This is done in the Control_AutomaticTransmission subsystem. The selector lever is set to drive position (D) if the reference velocity exceeds a limit of 0.1 m/s or if the accelerator pedal in stimulus mode is exceeds 5%. The selector lever is set to reverse position (R) if the reference velocity is smaller than -0.1 m/s.

The following illustration shows an overview of the Control_ManualTransmission function.

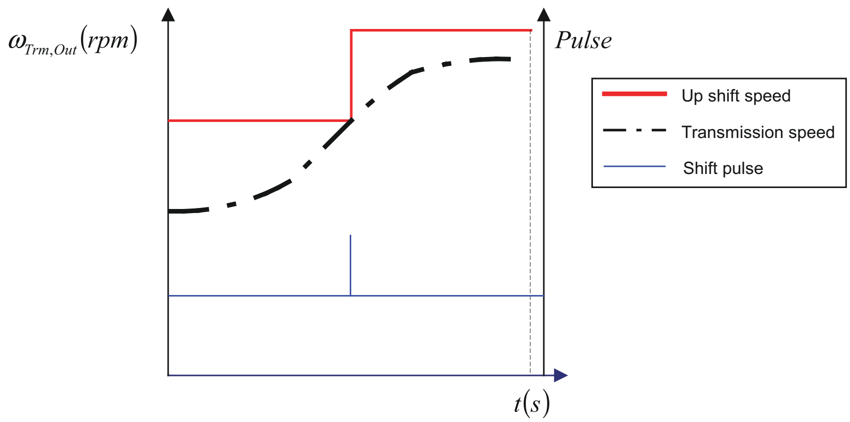


The model is divided into three sections. The light red section contains all the function which shift a gear up or down. The cyan subsystem contains functions for particular driving situations. Finally, the counter section is the gearshift itself, which generates the gear for the transmission and the pedal positions. The mode switch section enables the correct signals for the current driving situation. The following three paragraphs describe the different parts of the Gear_shifter block.

Shifting section (light red) Generally speaking, the shifting functionality compares the current transmission output speed with an up shift or down shift speed. If the transmission output speed crosses the u pshift or down shift speed for the current driving situation, a pulse is generated. Inside the gearshift, this pulse is used to engaged the clutch, to decrease the accelerator pedal, and to increase or decrease the current gear position.



The next illustration shows a typical up shift functionality:

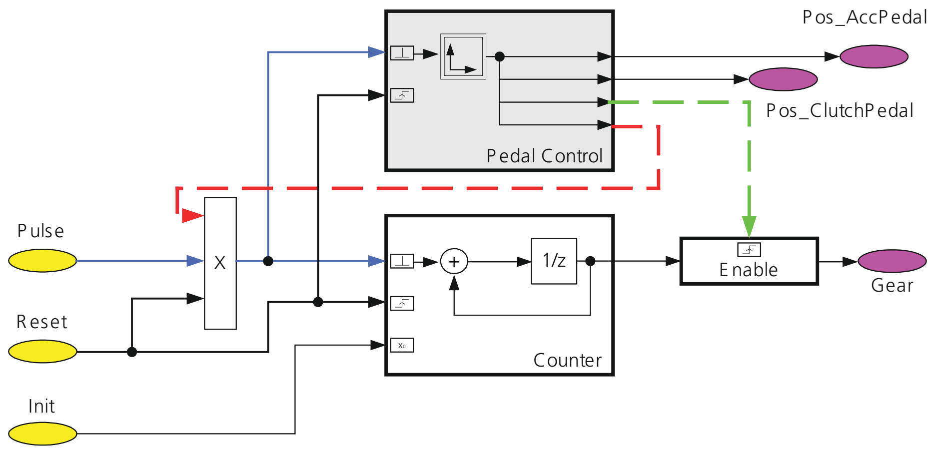


If the transmission output speed crosses the up shift signal, a shift pulse is generated. This forces the transmission to shift up a gear.

Special driving situations (cyan) The models for particular driving situation are used to shift to a certain gear position. These models do not shift via pulses but via gear positions which are generated according to the current driving situation. Currently four functions are implemented:

- The neutral gear function resets the gear position to the neutral position if the vehicle reaches zero velocity. This avoids an engine cut off.
- The brake shift function checks whether the shifting functions have set the current gear correctly for a specific brake pedal position. In certain driving situations, the driver may brake the vehicle very fast, which leads to rapidly decreasing vehicle speed. Because a gearshift takes some time, it may happen that the shifting section does not set the correct gear for the current vehicle speed. This can happen because although a second gearshift is necessary, it cannot be executed because the first gearshift has not finished. The brake shift function therefore compares the current gear with the recommended gear position during a braking maneuver. If a difference occurs, the recommended gear position is engaged.
- StartUpShift is a special function which engages the first gear and controls the clutch for vehicle startup. During startup it is assumed that an accelerator pedal position defined by `Const_Pos_AccPedal_Startup_FF` (default: 5%) is sufficient to reach the engine speed defined by `Const_n_Engine_StartUp_Set` (default: 1100 rpm). If this is not the case the engine startup will fail. Most likely your engine parameterization must be modified in this case. As a workaround the parameter `Const_Pos_AccPedal_Startup_FF` may be increased.
- If the vehicle is initialized with a specific start velocity, V_Init_Shift calculates the gear which fits to this initial velocity.

Gearshift The gearshift itself evaluates each shift request and transforms it into a corresponding gear and pedal position. The next illustration is a schematic of the gearshift:



The gearshift has of three functions. The counter is responsible for the gear position. The pedal control controls the pedal and decides if a pulse or reset is possible. It also controls the enable block, which allows the current gear position in the counter to be sent to the transmission model.

Inports The following table shows the inports:

Name	Unit	Description
AccBr_Mode	[1|2]	Mode signal for accelerator and brake pedal ▪ 1: Stimulus ▪ 2: Driver
Gear_Max	[]	Gear setpoint defined by maneuver input
n_Engine	[rpm]	Engine speed
omega_Out_Trm	[rad/s]	Transmission output speed
Pos_AccPedal_Ctrl	[%]	Accelerator pedal position
Pos_AccPedal_Stim	[%]	Accelerator pedal position stimulus (from maneuver scheduler)
Pos_BrakePedal_Ctrl	[%]	Brake pedal position
Pos_BrakePedal_Stim	[%]	Brake pedal position stimulus (from maneuver scheduler)
Reset	[0|1]	Reset of all integrators and internal states
Sw_Trm_Mode	[]	Switch for enabling manual of automatic transmission
v_Vehicle	[m/s]	Vehicle speed
v_Vehicle_Ref	[m/s]	Vehicle reference speed

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 📖).
Gear_GearShift	[]	Gear position for manual transmission
Pos_AccPedal_GearShift	[%]	Accelerator pedal position
Pos_BrakePedal_GearShift	[%]	Brake pedal position
Pos_ClutchPedal_GearShift	[%]	Clutch pedal position
SelectorLever_Gearshift	[]	Selector lever for automatic transmission: ▪ -3: TipShift ▪ -2: Park ▪ -1: Reverse ▪ 0: Neutral ▪ 1: Drive

Parameters

The following table shows the block parameters:

Name	Unit	Description
Const_I_Gain_StartUp	[]	I Gain for startup PI controller
Const_n_Engine_StartUp_Set	[rpm]	Engine speed setpoint during startup
Const_P_Gain_StartUp	[]	P Gain for startup PI controller
Const_Pos_AccPedal_Startup_FF	[%]	Accelerator pedal position at begin of startup in manual transmission mode
Const_t_Shift	[s]	Shift time
Map_omega_DownShift	[rpm]	Table for the transmission output shaft speed for a gear down shift f(Gear, Pos_AccPedal)
Map_omega_UpShift	[rpm]	Table for the transmission output shaft speed for a gear up shift f(Gear, Pos_AccPedal)
Map_Pos_Clutch_Pedal_StartUp	[%]	Clutch pedal position when starting
StepSize	[s]	Step size of model

Related topics

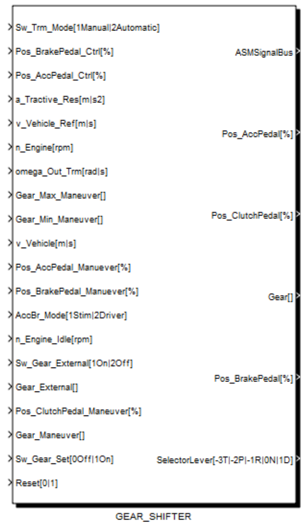
References

Gear Shifter V9 (ModelDesk Parameterizing 📖)

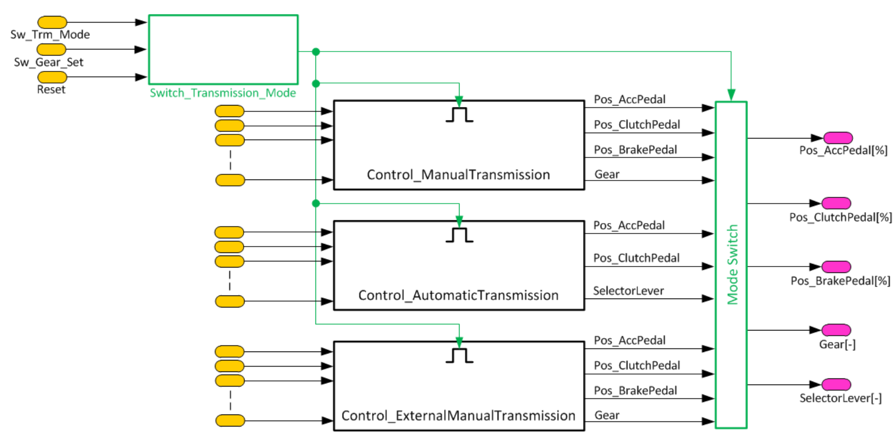
Gear Shifter (Version 10.0)

Description

The **GEAR_SHIFTER** block shifts the gears and controls the clutch pedal for an automatic or manual transmission according to the driving situation. There is also a user-defined mode for you to define the gear and the clutch pedal position yourself.



The following illustration shows the structure of the **GEAR_SHIFTER** block and gives an overview of the different modes.



The **GEAR_SHIFTER** block can handle three different modes:

- **Manual transmission shifting** (refer to Gear Shifter - Manual Transmission Shifting on page 129): This mode is active if the **Sw_Trm_Mode[1Manual|2Automatic]** inport = 1.
- **Automatic transmission shifting** (refer to Gear Shifter - Automatic Transmission Shifting on page 138): This mode is active if the **Sw_Trm_Mode[1Manual|2Automatic]** inport = 2.

- External manual transmission shifting (refer to Gear Shifter - External Manual Transmission Shifting on page 139): This mode is active if the Sw_Trm_Mode[1Manual|2Automatic] inport = 1 AND the Sw_Gear_Set[0Off|1On] inport = 1.

Related topics

References

Gear Shifter - Automatic Transmission Shifting.......138
Gear Shifter - External Manual Transmission Shifting.......139
Gear Shifter - Manual Transmission Shifting.......129
Gear Shifter Basic V10 (ModelDesk Parameterizing 🕮)
Gear Shifter V10 (ModelDesk Parameterizing 🕮)

Gear Shifter - Manual Transmission Shifting

Description

In this mode,the driver shifts the gear and controls the pedals according to the driving situation.

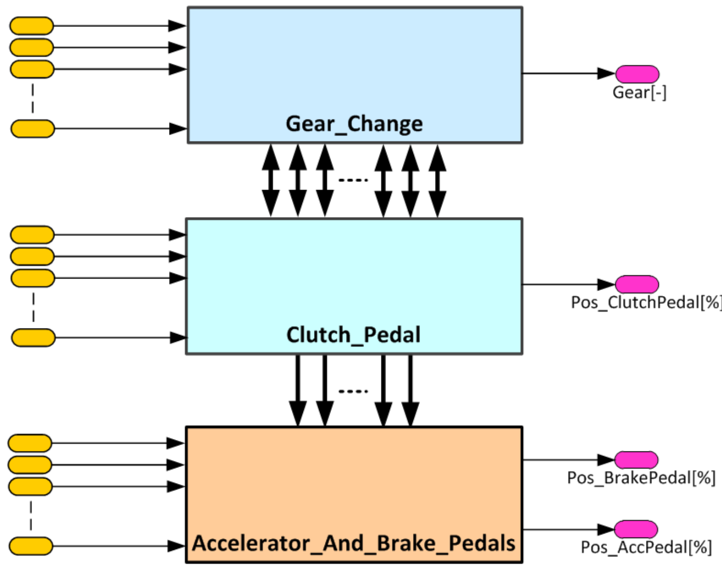
The manual transmission shifting belongs to the **Gear_Shifter 10.0** block. Refer to Gear Shifter (Version 10.0) on page 128.

A gearshift process begins with analyzing the driving situation and calculating the recommended gear. This gear is then sent as a request for clutch disengagement. When the clutch is completely disengaged, the requested gear is set and the clutch is engaged again. Other driving situations might require holding the gear, gear skipping, fast up shift or down shift, etc.

These functionalities are implemented in three subsystems:

- Gear_Change: Simulates gearshifts.
- Clutch_Pedal: Controls the clutch pedal.
- Accelerator_And_Brake_Pedal: Controls the accelerator pedal during startup.

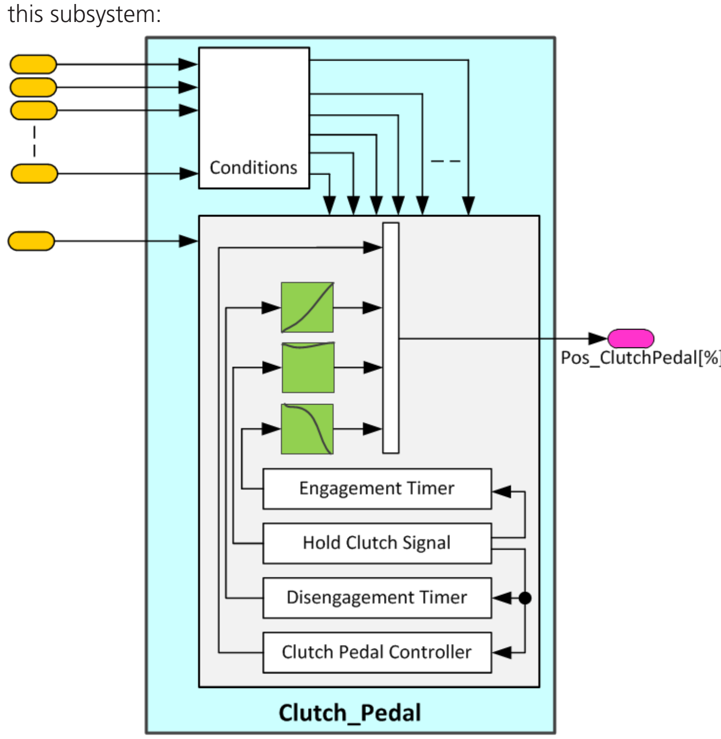
The following illustration shows how these subsystems are interconnected:



The calculations made inside each subsystem are described below.

Clutch_Pedal subsystem

In this subsystem the disengagement and engagement of the clutch pedal are triggered according to the requested gear and other conditions. You can define the clutch pedal positions during actuation via look-up-tables. The clutch pedal position is also controlled in special situations. The following illustration shows this subsystem:



Besides the gearshifting request, there are several other implemented conditions which control the actuation of the clutch pedal. Most of these conditions are related to user-defined parameters. See the following table.

Condition	**Description**
Low_Engine_Speed	Indicates whether a low engine speed has been reached.
	Parameter: None Calculation unit: [rpm] True: {engine speed} ≤ {1.8*current engine idle speed} False: {engine speed} ≥ {1.9*current engine idle speed}
Engine_On	Indicates whether the engine is on.
	Parameter: None Calculation unit: [rpm] Relay is implemented with input = engine speed, switch on point = 200 (true), switch off point = 25 (false)
Hard_Braking	Indicates whether the driver has to brake hard.
	Parameter: Const_a_x_Vehicle_LowLim Calculation unit: [m/s^2] True: {reference vehicle acceleration} < {parameter} False: Otherwise
Small_Velocity	Indicates whether there is a small reference vehicle velocity.
	Parameter: Const_v_x_Vehicle_LowLim Calculation unit: [m/s] True: {reference vehicle velocity} ≤ {parameter} False: {reference vehicle velocity} ≥ {parameter + 0.4167}
Startup	Indicates whether there is a startup.
	Parameter: None Calculation unit: [m/s] True: ({vehicle reference velocity} > {0.05}) & (gear = neutral) False: Otherwise
Standstill	Indicates whether the vehicle is at a standstill.
	Parameter: None Calculation unit: [m/s] True: {vehicle reference velocity} ≤ {0.05}

Condition	Description
	False: Otherwise
Initial_Velocity	Indicates whether the maneuver is defined with initial vehicle velocity.
	Parameter: None Calculation unit: [m/s] True: {vehicle velocity change in one simulation step size} ≥ {3} False: Otherwise
Big_Acceleration	Indicates whether a big acceleration is required during startup to follow the reference vehicle velocity.
	Parameter: Const_a_x_Vehicle_UpLim Calculation unit: [m/s^2] True: {reference vehicle acceleration} > {parameter} False: Otherwise
Big_Tractive_Resistance_Acceleration	Indicates whether a big driving resistance has to be overcome during startup.
	Parameter: Const_a_Tractive_Res_UpLim Calculation unit: [m/s^2] True: {tractive resistance acceleration} > {parameter} False: Otherwise

The actuation of the clutch pedal can be divided into three phases:

- Disengagement
- Hold
- Engagement

These phases are normally carried out in succession. However, disengagement always has the highest priority. It means the clutch pedal can always be disengaged at any time regardless of its actual position. Each of these phases is activated if certain conditions are fulfilled.

The clutch pedal is disengaged in the following cases:

- A gear change is requested.
- Reference vehicle velocity becomes small (Small_Velocity condition is true).
- The driver is braking hard (Hard_Braking condition is true) and a low engine speed is reached (Low_Engine_Speed condition is true).

The clutch pedal is held completely opened if clutch disengagement has ended and if at least one of the following conditions is fulfilled:

- Tracking small reference vehicle velocity (Small_Velocity condition is true). In this case, the driver follows the reference velocity with controlled clutch pedal position.

- The driver continues to brake hard (Hard_Braking condition is true) and there is low engine speed (Low_Engine_Speed is true).
- Startup with slipping clutch where no big acceleration is required (Big_Tractive_Resistance_Acceleration condition and Big_Acceleration condition are false)

If clutch disengagement ends and none of the conditions above is true, the clutch pedal is held completely open while the gear is being changed for a user-defined time, i.e., parameter Const_t_Hold_ClutchPedal.

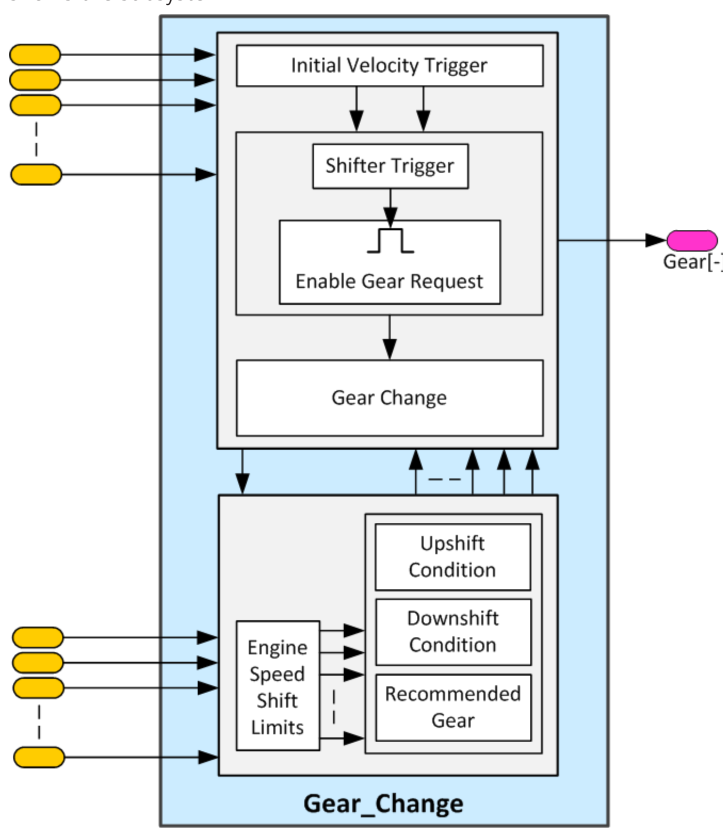
When clutch hold ends, clutch engagement is performed unconditionally.

The clutch pedal position is also controlled in special situations using a PI controller. The controller is active during the hold and engagement phases of the clutch pedal, if the disengagement phase has been ended and at least one of the following conditions is fulfilled:

- Tracking small reference vehicle velocity (Small_Velocity condition is true)
- Startup with slipping clutch where no big acceleration is required (Big_Tractive_Resistance_Acceleration condition and Big_Acceleration condition are false)

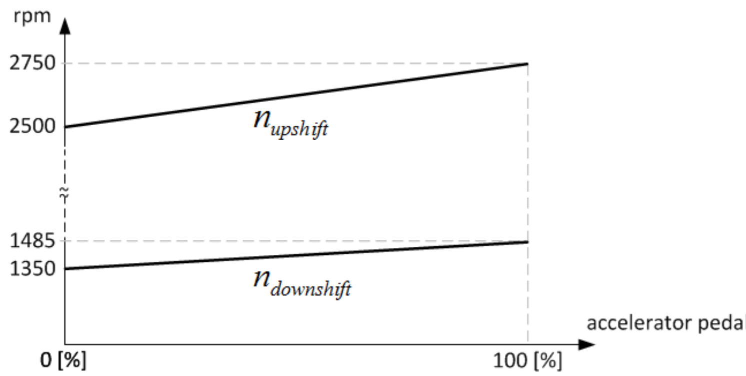
Gear_Change subsystem

The Gear_Change subsystem calculates the recommended gear according to the driving situation and simulates the gearshifts. The following illustration shows the subsystem.



First, the engine speed limits for the shift process and the ideal engine speed for the current and possible gears are calculated. You can parameterize engine speed limits as functions of the accelerator pedal position (parameters Map_n_Engine_Upshift_Lim and Map_n_Engine_Downshift_Lim) via look-up-tables.

The following illustration shows example values of the diesel engine speed limits for the up shift and down shift as functions of the accelerator pedal position.



For the calculation of the ideal engine speed, the gearbox ratios and the transmission output speed are needed:

where:

$$n_{current,\,ideal} = i_{current}\omega_{Gear,\,Out}$$

$$\overline{n}_{possible,\,ideal} = \overline{i}_{possible}\omega_{Gear,\,Out}$$

where:

$n_{current,\,ideal}$	is the ideal engine speed of the current gear
$i_{current}$	is the gearbox ratio of the current gear
$\omega_{Gear,\,Out}$	is the transmission output speed
$\overline{n}_{possible,\,ideal}$	is a vector of ideal engine speeds of all possible gears
$\overline{i}_{possible}$	is a vector of gearbox ratios of all possible gears

If the user-defined engine speed limits are used for the shift process, the ideal engine speeds of the possible gears are filtered, so that only certain gears are considered for further calculations. The ideal engine speeds of these gears should fulfill the following condition:

$$n_{upshift} \geq \overline{n}_{allowed,\,ideal} \geq \left(n_{downshift} + n_{shift,\,bias}\right)$$

where:

$n_{upshift}$	is the up shift engine speed limit
$n_{downshift}$	is the down shift engine speed limit
$\overline{n}_{allowed,\,ideal}$	is a vector of the ideal engine speeds of the allowed gears
$n_{shift,\,bias}$	is the engine speed shift bias

The engine speed shift bias is a function of the tractive resistance acceleration and defined in a look-up-table. This value is used to avoid oscillations in the calculated allowed gears, because the transmission output speed drops as the driver disengages the clutch pedal, causing a drop in the calculated ideal engine speeds stated in the equations above.

The look-up-table also includes the engine speed drop after lowering the gear, i.e., it provides a nonzero value for zero tractive resistance acceleration, and thus gear hunting can be avoided.

Afterwards, the recommended gear for the current driving situation is calculated from the allowed gears. That is, a check is made whether the current driver behavior indicates accelerating or braking demand. If there is acceleration, the maximum allowed gear is selected, whereas if there is braking, the minimum allowed gear is selected. In other words, this simulates an eco-friendly driving manner.

The up shift request is triggered as the ideal engine speed of the current gear becomes greater than or equals the up shift limit, as shown below:

$$request_{upshift} = \begin{cases} 1, & n_{current,ideal} \geq n_{upshift} \\ 0, & otherwise \end{cases}$$

The down shift request is triggered as the ideal engine speed of the current gear becomes smaller than or equals the down shift limit, as shown below:

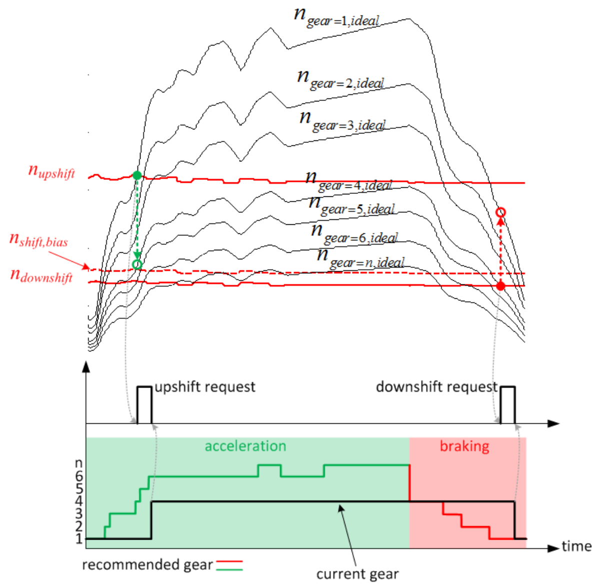
$$request_{downshift} = \begin{cases} 1, & n_{current,ideal} \leq n_{downshift} \\ 0, & otherwise \end{cases}$$

The up shift or down shift request triggers a gear change request for clutch disengagement. Depending on the driving situation, these requests might be disabled. For example, this happens if the driver recognizes that hard braking is required (Hard_Braking condition is true). In this case, the up shift and down shift requests are deactivated and the gear is held. Meanwhile, once the driver recognizes a low engine speed (Low_Engine_Speed condition is true), the clutch is disengaged to avoid an engine stall just like the real driver would do.

The gear change request causes the clutch to be disengaged. Once the clutch is completely disengaged, a gearshift occurs.

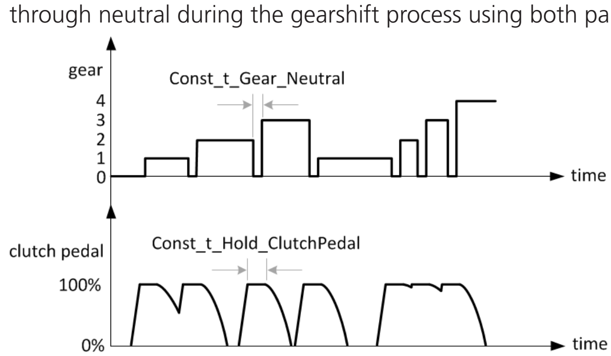
In braking maneuvers which are not recognized as hard braking (Hard_Braking condition is false), the driver can skip gears during the down shift process whenever this is necessary.

The illustration below shows the calculation of the gearshift in accelerating and braking maneuvers:



During the gearshift, you can activate the feature for passing the gear through neutral using the parameter Const_t_Gear_Neutral. This parameter defines the time that the gear passes through neutral during the gearshift process and provides a realistic gearshifting behavior. If the parameter is zero, this feature is inactive.

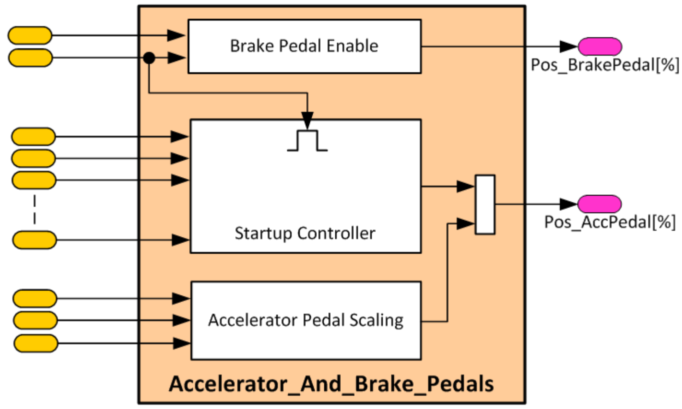
Since the gear passes through neutral only when the clutch is completely opened, the parameter Const_t_Gear_Neutral should always be less than or equal to the time the clutch pedal remains completely open, i.e., parameter Const_t_Hold_ClutchPedal. If this condition is not fulfilled, the minimum time from both parameters is used. The illustration below shows the gear passing through neutral during the gearshift process using both parameters.



You can also force the driver to select a certain gear value at any time during the simulation. This feature is available by using the inports Sw_Gear_External[1On|2Off] and Gear_External[]. If Sw_Gear_External is set to 1, the Gear_External value is used as the current gear. If Sw_Gear_External is set back to 0, the calculated gear is used as mentioned in the calculations above. To use this feature, the inport Sw_Trm_Mode[1Manual|2Automatic] should be 1.

Accelerator_And_Brake_Pedal subsystem

The following illustration shows the structure of the subsystem:



This subsystem implements the startup functionality. A PI controller is used to modify the accelerator pedal position to reach a user-defined engine speed (parameter Const_n_Engine_StartUp_Set) during vehicle startup.

The startup controller is activated during startup only if at least one of the following conditions is fulfilled:

- There is big tractive resistance (Big_Tractive_Resistance_Acceleration condition is true)
- The maneuver or the vehicle velocity are initialized with a nonzero value (Initial_Velocity condition is true or reference vehicle velocity starts with a nonzero value).
- Big acceleration is needed to follow the reference vehicle velocity (Big_Acceleration condition is true)

The brake pedal position is disabled if the startup controller is active.

Since this block ensures a correct startup, you should consider changing the defined engine speed during startup, for example, if the vehicle has to start on a steep hill.

Another block is used to scale the accelerator pedal position during a gear change. For the scaling, a user-defined look-up-table is used as a function of the inverse of the normalized clutch pedal position (parameter Map_Pos_AccPedal).

Related topics

References

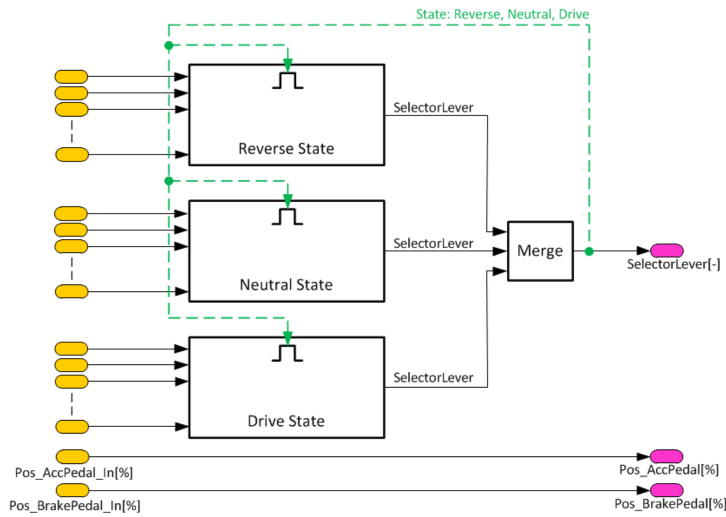
Gear Shifter - Automatic Transmission Shifting ... 138
Gear Shifter - External Manual Transmission Shifting ... 139
Gear Shifter (Version 10.0) ... 128

Gear Shifter - Automatic Transmission Shifting

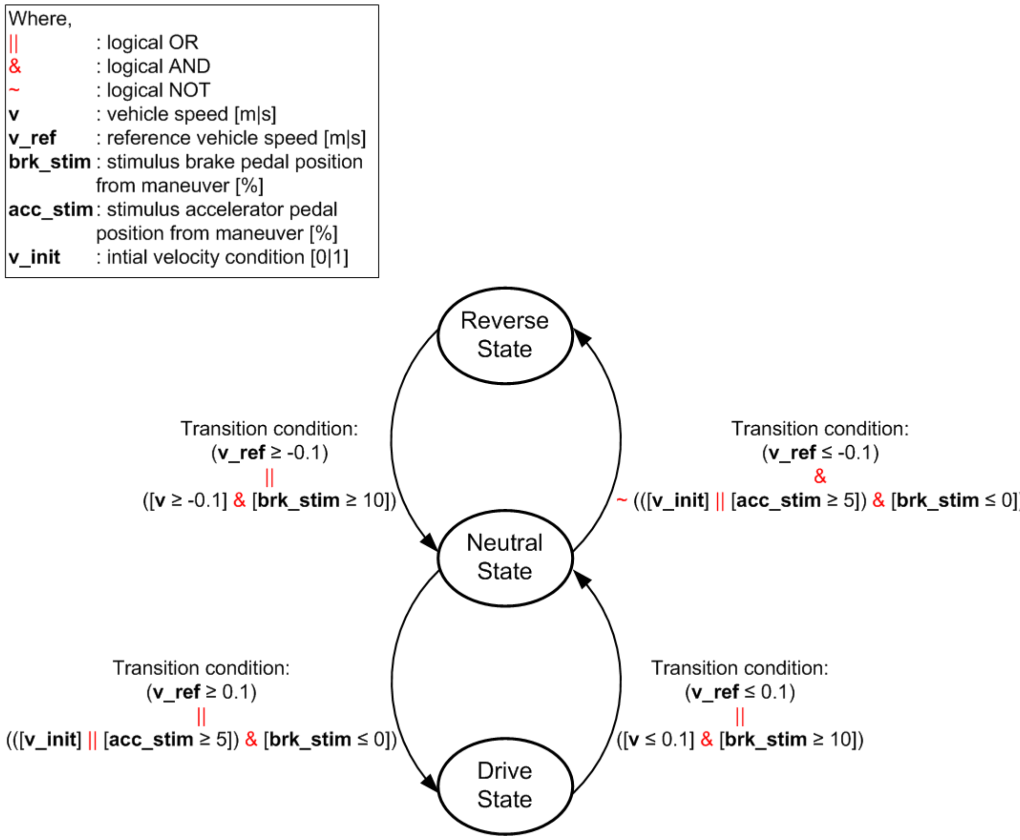
Description

In this mode, the selector lever is simulated for an automatic transmission. The following illustration shows the structure of the block.

The automatic transmission shifting belongs to the **Gear_Shifter 10.0** block. Refer to Gear Shifter (Version 10.0) on page 128.



A state machine has been implemented to check the driving situation continuously and set the correct selector lever, where only one state can be active at a time. The transition from one state to another is conditionally driven and only occurs in a rational manner. The following figure illustrates the simulated states as well as the transitions between them.



While the selector lever is set in this block, the shifting itself is done by the SOFT_ECU_TRANSMISSION block.

Related topics

References

Gear Shifter - External Manual Transmission Shifting

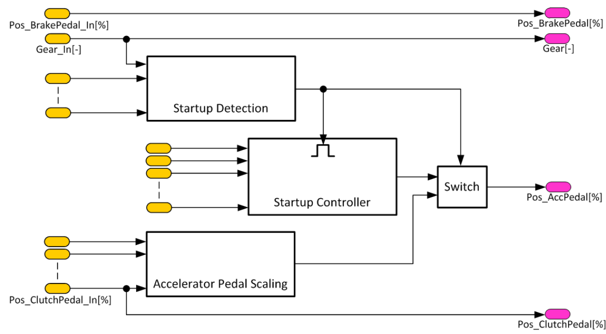
Description

The external manual transmission shifting belongs to the Gear_Shifter 10.0 block. Refer to Gear Shifter (Version 10.0) on page 128.

This user-defined mode handles maneuvers with a predefined gear change set and the corresponding clutch pedal position. The values of the accelerator and brake pedals are controlled in the longitudinal controller.

The user-predefined gear and clutch pedal position are forwarded to the outports. Additionally, when a startup is detected, a PI controller controls the accelerator pedal position to reach a user-defined engine speed (parameter Const_n_Engine_StartUp_Set).

In all other cases the accelerator pedal position is forwarded to the outport and scaled only if a gear change occurs. For the scaling, a simplified look-up-table is used as a function of the inverse of the normalized clutch pedal position.



Inports

The following table shows the inports:

Name	Unit	Description
a_Tractive_Res	[m/s^2]	Tractive resistance acceleration
AccBr_Mode	[1|2]	Mode signal for accelerator and brake pedal ▪ 1: Stimulus ▪ 2: Driver
Gear_External	[]	User-defined external gear
Gear_Maneuver	[]	Stimulus gear from maneuver
Gear_Max_Maneuver	[]	Maximum gear from maneuver
Gear_Min_Maneuver	[]	Minimum gear from maneuver
n_Engine	[rpm]	Engine speed
n_Engine_Idle	[rpm]	Engine idle speed
omega_Out_Trm	[rad/s]	Transmission output speed
Pos_AccPedal_Ctrl	[%]	Calculated accelerator pedal position from the longitudinal controller
Pos_AccPedal_Maneuver	[%]	Stimulus accelerator pedal position from maneuver
Pos_AccBrake_Maneuver	[%]	Stimulus brake pedal position from maneuver
Pos_BrakePedal_Ctrl	[%]	Calculated brake pedal position from the longitudinal controller
Pos_ClutchPedal_Maneuver	[%]	Stimulus clutch pedal position from maneuver
Reset_States	[0|1]	Reset
Sw_Gear_External	[1|2]	Switch for user-defined external gear ▪ 1: On ▪ 2: Off
Sw_Gear_Set	[0|1]	Switch for stimulus gear from maneuver ▪ 0: Off

Name	Unit	Description
		▪ 1: On
Sw_Trm_Mode	[1|2]	Switch for enabling manual or automatic transmission ▪ 1: Manual ▪ 2: Automatic
v_Vehicle	[m/s]	Vehicle velocity
v_Vehicle_Ref	[m/s]	Reference vehicle velocity

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 📖).
Gear	[]	Gear
Pos_AccPedal	[%]	Accelerator pedal position
Pos_BrakePedal	[%]	Brake pedal position
Pos_ClutchPedal	[%]	Clutch pedal position
SelectorLever	[-3|-2|-1|0|1]	Selector lever position ▪ -3: Tip shift ▪ -2: Park ▪ -1: Reverse ▪ 0: Neutral ▪ 1: Drive

Parameters

The following table shows the parameters:

Name	Unit	Description
Const_a_Tractive_Res_UpLim	[m/s^2]	Tractive resistance acceleration threshold for startup
Const_a_x_Vehicle_LowLim	[m/s^2]	Hard braking threshold for gear hold
Const_a_x_Vehicle_UpLim	[m/s^2]	Acceleration threshold for startup
Const_I_Gain_ClutchPedal	[1/s]	I gain of clutch pedal controller
Const_I_Gain_StartUp	[1/s]	I gain of accelerator pedal startup controller
Const_n_Engine_StartUp_Set	[rpm]	Engine speed setpoint during startup
Const_P_Gain_ClutchPedal	[]	P gain of clutch pedal controller
Const_P_Gain_StartUp	[]	P gain of accelerator pedal startup controller
Const_Pos_AccPedal_Startup_FF	[%]	Accelerator pedal position for startup
Const_t_Gear_Neutral	[s]	Time of gear passing through neutral during gear change
Const_t_Hold_ClutchPedal	[s]	Disengaged clutch hold time
Const_v_x_Vehicle_LowLim	[m/s]	Small velocity threshold for driving with slipping clutch
Map_i_Ratio	[]	Gear ratio = f(gear)

Name	Unit	Description
Map_n_Engine_Downshift_Lim	[rpm]	Down shift engine speed limit = f(Pos_AccPedal)
Map_n_Engine_Upshift_Lim	[rpm]	Up shift engine speed limit = f(Pos_AccPedal)
Map_omega_Out_Trm_Shift_Bias	[rpm]	Transmission output speed shift bias = f(tractive resistance acceleration)
Map_Pos_AccPedal	[0_1]	Accelerator pedal factor during gear change = f(Pos_ClutchPedal)
Map_Pos_ClutchPedal_Disengagement	[%]	Clutch pedal disengagement position not during startup = f(t)
Map_Pos_ClutchPedal_Disengagement_StartUp	[%]	Clutch pedal disengagement position during startup = f(t)
Map_Pos_ClutchPedal_Engagement	[%]	Clutch pedal engagement position not during startup = f(t)
Map_Pos_ClutchPedal_Engagement_StartUp	[%]	Clutch pedal engagement position during startup = f(t)
StepSize	[s]	Simulation step size
Sw_Upshift_Gear_Skipping	[1|2]	Gear skipping switch during up shift ▪ 1: On ▪ 2: Off

Handling of unused inports

Depending on the demo model and the library, some of the inport signals might not be used. In this case, these inports cannot be connected to dummy values. They should be connected to constant values instead.

These are defined in the following table:

Inport Name	Not Used If
AccBr_Mode	Its value = 0
Pos_AccPedal_Maneuver	Its value = 0
Pos_BrakePedal_Maneuver	Its value = 0
Gear_External	Sw_Gear_External = 2 OR Sw_Trm_Mode ≠ 1
Gear_Max_Maneuver	Its value = 0
Sw_Gear_Set	Its value = 0
Sw_Gear_Set	Its value = 0

Related topics

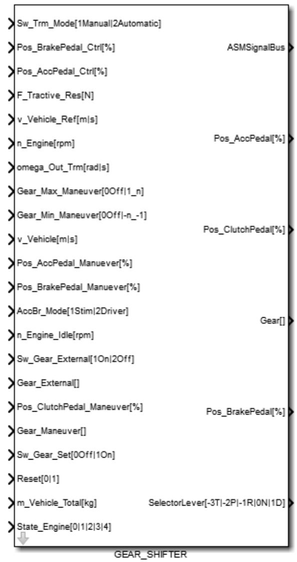
References

Gear Shifter - Automatic Transmission Shifting........ 138
Gear Shifter - Manual Transmission Shifting........ 129
Gear Shifter (Version 10.0)........ 128

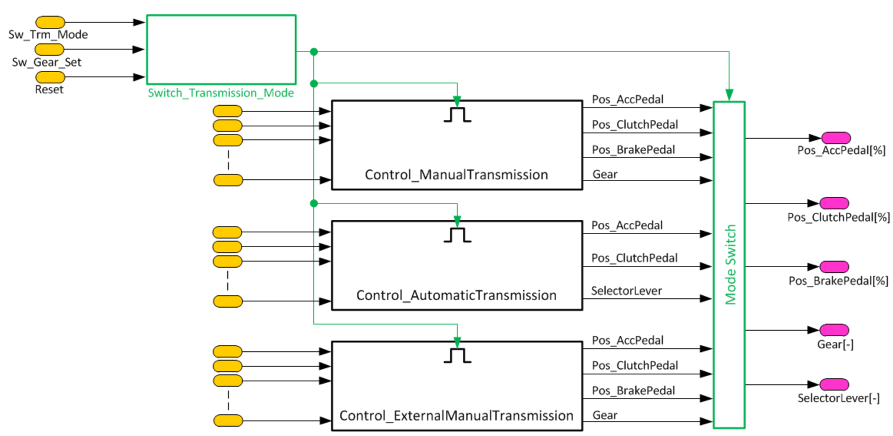
Gear Shifter (Version 11.0 - 13.0)

Description

The GEAR_SHIFTER block shifts the gears and controls the clutch pedal for an automatic or manual transmission according to the driving situation. There is also a user-defined mode for you to define the gear and the clutch pedal position yourself.



The following illustration shows the structure of the GEAR_SHIFTER block and gives an overview of the different modes.



The GEAR_SHIFTER block can handle three different modes:

- Manual transmission shifting (refer to Gear Shifter - Manual Transmission Shifting on page 144): This mode is active if the Sw_Trm_Mode[1Manual|2Automatic] inport = 1.
- Automatic transmission shifting (refer to Gear Shifter - Automatic Transmission Shifting on page 152): This mode is active if the Sw_Trm_Mode[1Manual|2Automatic] inport = 2.

- External manual transmission shifting (refer to Gear Shifter - External Manual Transmission Shifting on page 153): This mode is active if the Sw_Trm_Mode[1Manual|2Automatic] inport = 1 AND the Sw_Gear_Set[0Off|1On] inport = 1.

Related topics

References

Gear Shifter V13 (ModelDesk Parameterizing 📖)

Gear Shifter - Manual Transmission Shifting

Description

In this mode,the driver shifts the gear and controls the pedals according to the driving situation.

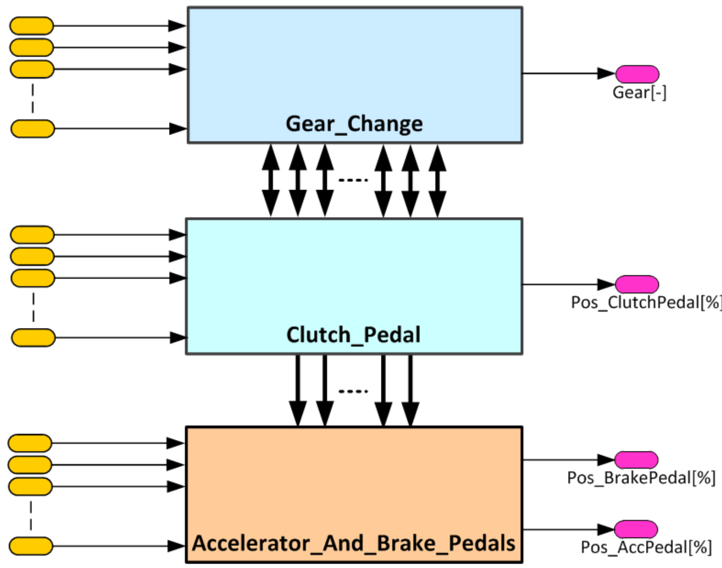
The manual transmission shifting belongs to the Gear_Shifter 11.0 block. Refer to Gear Shifter (Version 11.0 - 13.0) on page 143.

A gearshift process begins with analyzing the driving situation and calculating the recommended gear. This gear is then sent as a request for clutch disengagement. When the clutch is completely disengaged, the requested gear is set and the clutch is engaged again.

These functionalities are implemented in three subsystems:

- Gear_Change: Simulates gearshifts.
- Clutch_Pedal: Controls the clutch pedal.
- Accelerator_And_Brake_Pedal: Controls the accelerator pedal during startup.

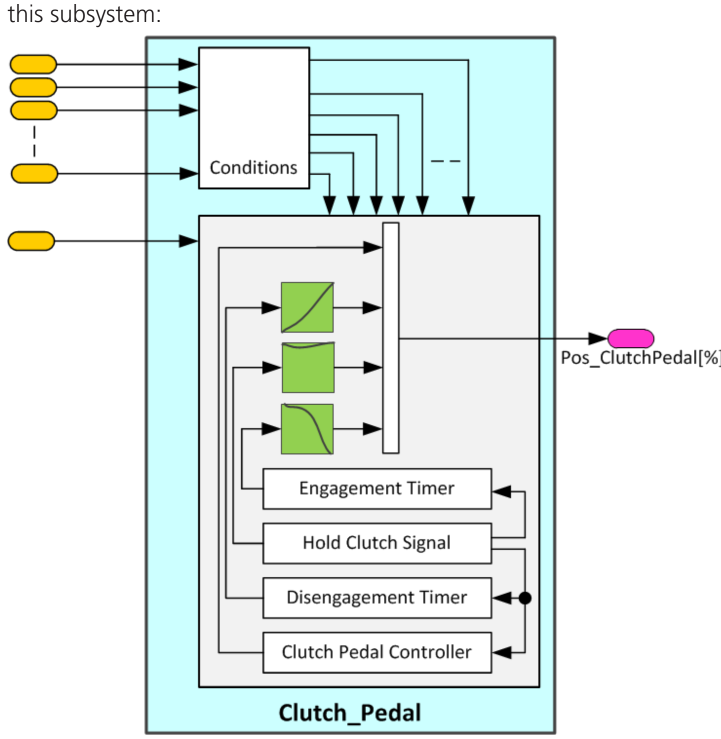
The following illustration shows how these subsystems are interconnected:



The calculations made inside each subsystem are described below.

Clutch_Pedal subsystem

In this subsystem the disengagement and engagement of the clutch pedal are triggered according to the requested gear and other conditions. You can define the clutch pedal positions during actuation via look-up-tables. The clutch pedal position is also controlled in special situations. The following illustration shows this subsystem:



Besides the gearshifting request, there are several other implemented conditions which control the actuation of the clutch pedal. Most of these conditions are related to user-defined parameters. See the following table.

Condition	**Description**
Low_Engine_Speed	Indicates whether a low engine speed has been reached.
	Parameter: None Calculation unit: [rpm] True: {engine speed} ≤ {current engine idle speed + 100} False: {engine speed} ≥ {current engine idle speed + 500}
Engine_On	Indicates whether the engine is on.
	Parameter: None Calculation unit: [] The signal is calculated from the State_Engine[0|1|2|3|4] input signal and is active if the engine state equals 3, otherwise it is inactive.
Hard_Braking	Indicates whether the driver has to brake hard.
	Parameter: Const_a_x_Vehicle_LowLim Calculation unit: [m/s^2] True: {reference vehicle acceleration} < {parameter} False: Otherwise
Small_Velocity	Indicates whether there is a small reference vehicle velocity.
	Parameter: Const_v_x_Vehicle_LowLim Calculation unit: [m/s] True: {reference vehicle velocity} ≤ {parameter} False: {reference vehicle velocity} ≥ {parameter + 0.4167}
Startup	Indicates whether there is a startup.
	Parameter: None Calculation unit: [m/s] True: ({vehicle reference velocity} > {0.05}) & (gear = neutral) False: Otherwise
Standstill	Indicates whether the vehicle is at a standstill.
	Parameter: None Calculation unit: [m/s] True: {vehicle reference velocity} ≤ {0.05} False: Otherwise
Initial_Velocity	Indicates whether the maneuver is defined with initial vehicle velocity.
	Parameter: None Calculation unit: [m/s] True: {vehicle velocity change in one simulation step size} ≥ {3} False: Otherwise
Big_Acceleration	Indicates whether a big acceleration is required during startup to follow the reference vehicle velocity.
	Parameter: Const_a_x_Vehicle_UpLim Calculation unit: [m/s^2]

Condition	Description
	True: {reference vehicle acceleration} > {parameter} False: Otherwise
Big_Tractive_Resistance_Acceleration	Indicates whether a big driving resistance has to be overcome during startup.
	Parameter: Const_a_Tractive_Res_UpLim Calculation unit: [m/s^2] True: {tractive resistance acceleration} > {parameter} False: Otherwise

The actuation of the clutch pedal can be divided into three phases:

- Disengagement
- Hold
- Engagement

These phases are normally carried out in succession. However, disengagement always has the highest priority. It means the clutch pedal can always be disengaged at any time regardless of its actual position. Each of these phases is activated if certain conditions are fulfilled.

The clutch pedal is disengaged in the following cases:

- A gear change is requested.
- Reference vehicle velocity becomes small (Small_Velocity condition is true).
- The driver is braking hard (Hard_Braking condition is true) and a low engine speed is reached (Low_Engine_Speed condition is true).

The clutch pedal is held completely opened if clutch disengagement has ended and if at least one of the following conditions is fulfilled:

- Tracking small reference vehicle velocity (Small_Velocity condition is true).
- The driver continues to brake hard (Hard_Braking condition is true) and there is low engine speed (Low_Engine_Speed is true).
- Startup with slipping clutch where no big acceleration is required (Big_Tractive_Resistance_Acceleration condition and Big_Acceleration condition are false)

If clutch disengagement ends and none of the conditions above is true, the clutch pedal is held completely open while the gear is being changed for a user-defined time, i.e., parameter Const_t_Hold_ClutchPedal.

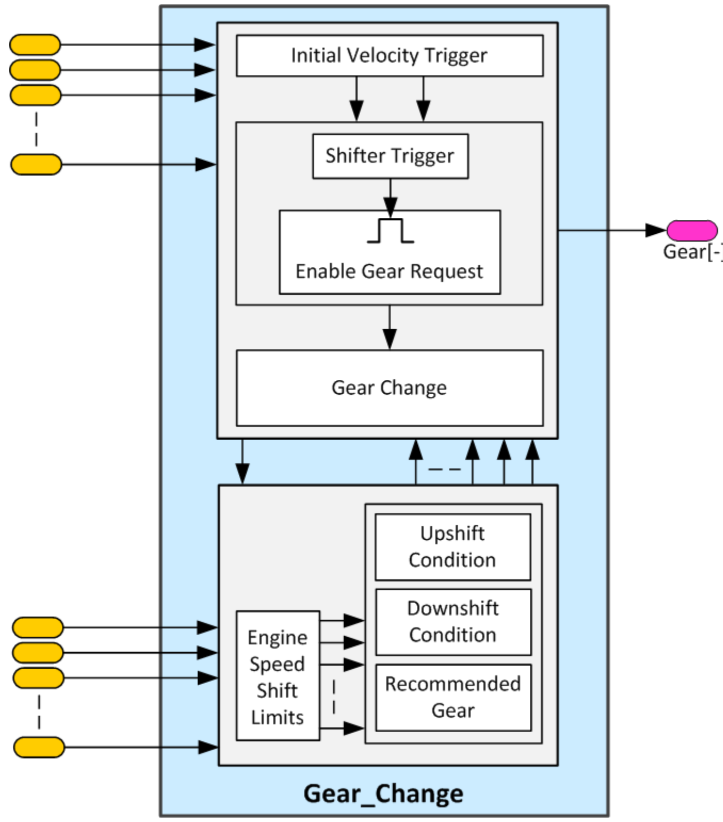
When clutch hold ends, clutch engagement is performed unconditionally.

The clutch pedal position is also controlled in special situations using a PI controller. The controller is active during the hold and engagement phases of the clutch pedal, if the disengagement phase has been ended and at least one of the following conditions is fulfilled:

- Tracking small reference vehicle velocity (Small_Velocity condition is true)
- Startup with slipping clutch where no big acceleration is required (Big_Tractive_Resistance_Acceleration condition and Big_Acceleration condition are false)

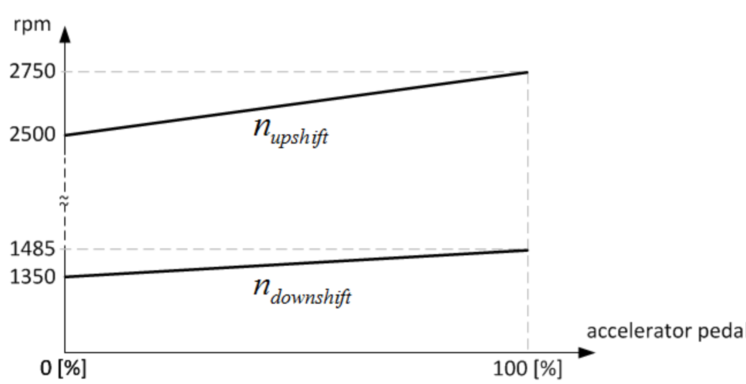
Gear_Change subsystem

The Gear_Change subsystem calculates the recommended gear according to the driving situation and simulates the gearshifts. The following illustration shows the subsystem.



First, the engine speed limits for the shift process and the ideal engine speed for the current and possible gears are calculated. You can parameterize engine speed limits as functions of the accelerator pedal position (parameters Map_n_Engine_Upshift_Lim and Map_n_Engine_Downshift_Lim) via look-up-tables.

The following illustration shows exemplary values of the engine speed limits for the up shift and down shift as functions of the accelerator pedal position.



For the calculation of the ideal engine speed, the gearbox ratios and the transmission output speed are needed:

where:

$$n_{current,\,ideal} = i_{current}\omega_{Gear,\,Out}$$

$$\overline{n}_{possible,\,ideal} = \overline{i}_{possible}\omega_{Gear,\,Out}$$

where:

$n_{current,\,ideal}$	is the ideal engine speed of the current gear
$i_{current}$	is the gearbox ratio of the current gear
$\omega_{Gear,\,Out}$	is the transmission output speed
$\overline{n}_{possible,\,ideal}$	is a vector of ideal engine speeds of all possible gears
$\overline{i}_{possible}$	is a vector of gearbox ratios of all possible gears

If the user-defined engine speed limits are used for the shift process, the ideal engine speeds of the possible gears are filtered, so that only certain gears are considered for further calculations. The ideal engine speeds of these gears should fulfill the following condition:

$$n_{upshift} \geq \overline{n}_{allowed,\,ideal} \geq \left(n_{downshift} + n_{shift,\,bias}\right)$$

where:

$n_{upshift}$	is the up shift engine speed limit
$n_{downshift}$	is the down shift engine speed limit
$\overline{n}_{allowed,\,ideal}$	is a vector of the ideal engine speeds of the allowed gears
$n_{shift,\,bias}$	is the engine speed shift bias

The engine speed shift bias is a function of the tractive resistance acceleration and defined in a look-up-table. This value is used to avoid oscillations in the calculated allowed gears, because the transmission output speed drops as the driver disengages the clutch pedal, causing a drop in the calculated ideal engine speeds stated in the equations above.

Afterwards, the recommended gear for the current driving situation is calculated from the allowed gears. That is, a check is made whether the current driver behavior indicates accelerating or braking demand. If there is acceleration, the maximum allowed gear is selected, whereas if there is braking, the minimum allowed gear is selected. In other words, this simulates an eco-friendly driving manner.

The up shift request is triggered as the ideal engine speed of the current gear becomes greater than or equals the up shift limit, as shown below:

$$request_{upshift} = \begin{cases} 1, & n_{current,\,ideal} \geq n_{upshift} \\ 0, & otherwise \end{cases}$$

The down shift request is triggered as the ideal engine speed of the current gear becomes smaller than or equals the down shift limit, as shown below:

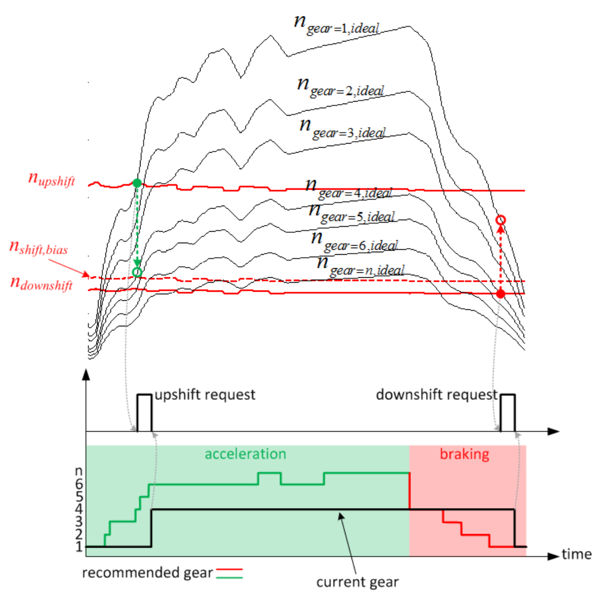
$$request_{downshift} = \begin{cases} 1, & n_{current,\,ideal} \leq n_{downshift} \\ 0, & otherwise \end{cases}$$

The up shift or down shift request triggers a gear change request. Depending on the driving situation, these requests might be disabled. For example, this happens if the driver recognizes that hard braking is required (Hard_Braking condition is true). In this case, the up shift and down shift requests are deactivated and the gear is held. Meanwhile, once the driver recognizes a low engine speed (Low_Engine_Speed condition is true), the clutch is disengaged to avoid an engine stall just like the real driver would do.

The gear change request causes the clutch to be disengaged. Once the clutch is completely disengaged, a gearshift occurs.

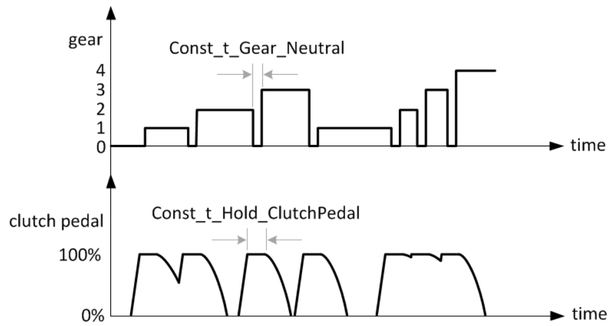
In braking maneuvers which are not recognized as hard braking (Hard_Braking condition is false), the driver can skip gears during the down shift process whenever this is necessary. However, during the gear up shift you can activate or deactivate this feature using the Sw_Upshift_Gear_Skipping parameter. This also includes the selection of the startup gear, which is selected depending on the load conditions.

The illustration below shows the calculation of the gearshift in accelerating and braking maneuvers:



During the gearshift, you can activate the feature for passing the gear through neutral using the parameter Const_t_Gear_Neutral. This parameter defines the time that the gear passes through neutral during the gearshift process and provides a realistic gearshifting behavior. If the parameter is zero, this feature is inactive.

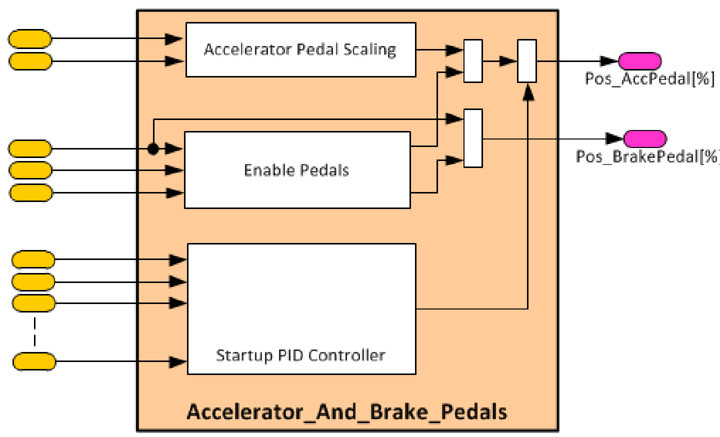
Since the gear passes through neutral only when the clutch is completely opened, the parameter Const_t_Gear_Neutral should always be less than or equal to the time the clutch pedal remains completely open, i.e., parameter Const_t_Hold_ClutchPedal. If this condition is not fulfilled, the minimum time from both parameters is used. The illustration below shows the gear passing through neutral during the gearshift process using both parameters.



You can also force the driver to select a certain gear value at any time during the simulation. This feature is available by using the inports Sw_Gear_External[1On|2Off] and Gear_External[]. If Sw_Gear_External is set to 1, the Gear_External value is used as the current gear. If Sw_Gear_External is set back to 0, the calculated gear is used as mentioned in the calculations above. To use this feature, the inport Sw_Trm_Mode[1Manual|2Automatic] should be 1.

Accelerator_And_Brake_Pedal subsystem

The following illustration shows the structure of the subsystem:



This subsystem implements the startup functionality. A PID controller is used to modify the accelerator pedal position to reach a user-defined engine speed (parameter Const_n_Engine_StartUp_Set) during vehicle startup.

The startup controller is activated during startup only if at least one of the following conditions is fulfilled:

- There is big tractive resistance (Big_Tractive_Resistance_Acceleration condition is true)
- The maneuver or the vehicle velocity are initialized with a nonzero value (Initial_Velocity condition is true or reference vehicle velocity starts with a nonzero value).
- Big acceleration is needed to follow the reference vehicle velocity (Big_Acceleration condition is true)

The brake pedal position is disabled if the startup controller is active.

Since this block ensures a correct startup, you should consider changing the defined engine speed during startup, for example, if the vehicle has to start on a steep hill.

Another block is used to scale the accelerator pedal position during a gear change. For the scaling, a user-defined look-up-table is used as a function of the inverse of the normalized clutch pedal position (parameter Map_Pos_AccPedal).

Related topics

References

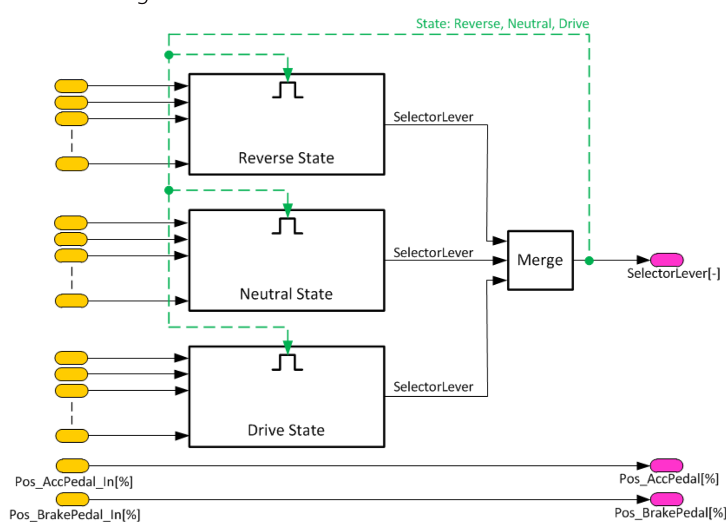
Gear Shifter - Automatic Transmission Shifting

Description

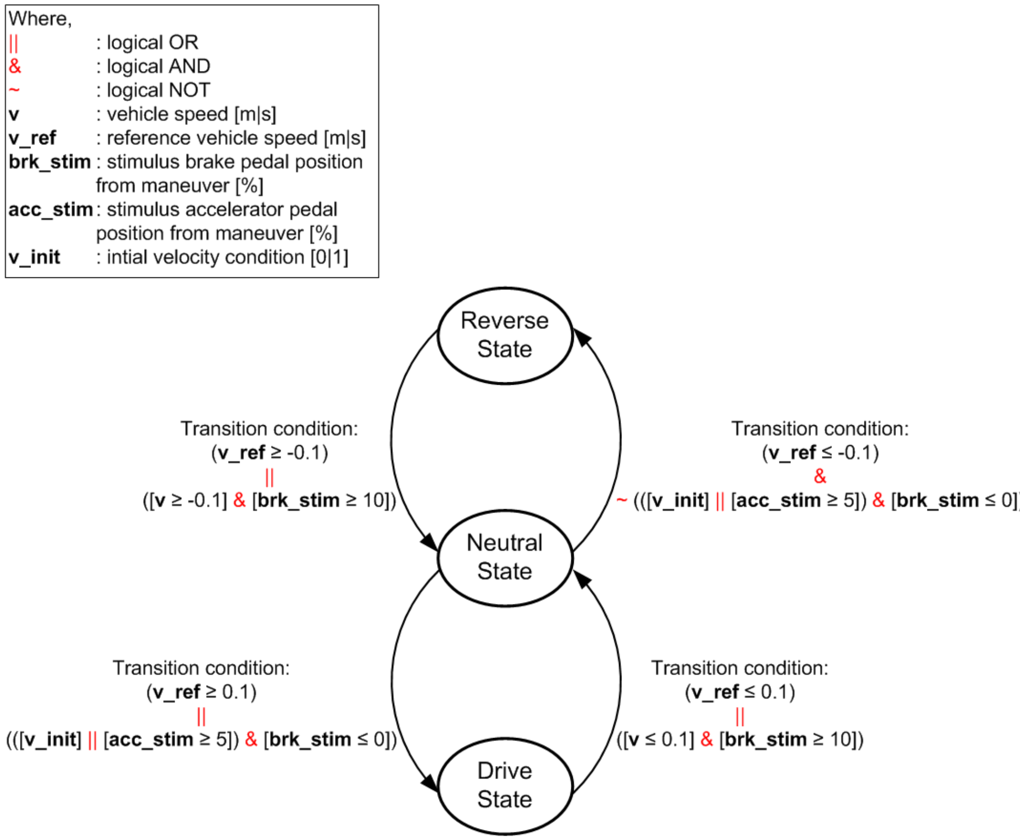
In this mode, the selector lever is simulated for an automatic transmission.

The automatic transmission shifting belongs to the **Gear_Shifter 11.0** block. Refer to Gear Shifter (Version 11.0 - 13.0) on page 143.

The following illustration shows the structure of the block.



A state machine has been implemented to check the driving situation continuously and set the correct selector lever, where only one state can be active at a time. The transition from one state to another is conditionally driven and only occurs in a rational manner. The following figure illustrates the simulated states as well as the transitions between them.



While the selector lever is set in this block, the shifting itself is done by the SOFT_ECU_TRANSMISSION block.

Related topics

References

Gear Shifter - External Manual Transmission Shifting 153
Gear Shifter - Manual Transmission Shifting 144
Gear Shifter (Version 11.0 - 13.0) 143

Gear Shifter - External Manual Transmission Shifting

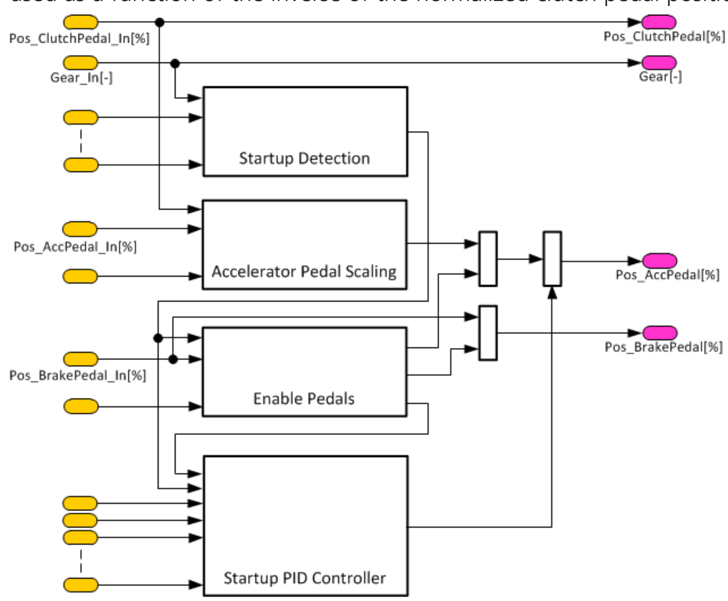
Description

The external manual transmission shifting belongs to the Gear_Shifter 11.0 block. Refer to Gear Shifter (Version 11.0 - 13.0) on page 143.

This user-defined mode handles maneuvers with a predefined gear change set and the corresponding clutch pedal position. The values of the accelerator and brake pedals are controlled in the longitudinal controller.

The user-predefined gear and clutch pedal position are forwarded to the outports. Additionally, when a startup is detected, a PID controller controls the accelerator pedal position to reach a user-defined engine speed (Const_n_Engine_StartUp_Set parameter).

In all other cases the accelerator pedal position is forwarded to the outport and scaled only if a gear change occurs. For the scaling, a simplified look-up-table is used as a function of the inverse of the normalized clutch pedal position.



Inports

The following table shows the inports:

Name	Unit	Description
AccBr_Mode	[1|2]	Mode signal for accelerator and brake pedal ▪ 1: Stimulus ▪ 2: Driver
F_Tractive_Res	[N]	Tractive resistance force
Gear_External	[]	User-defined external gear
Gear_Maneuver	[]	Stimulus gear from maneuver
Gear_Max_Maneuver	[0Off|1_n]	Maximum gear from maneuver
Gear_Min_Maneuver	[0Off|-n_-1]	Minimum gear from maneuver
m_Vehicle_Total	[kg]	Vehicle mass
n_Engine	[rpm]	Engine speed
n_Engine_Idle	[rpm]	Engine idle speed
omega_Out_Trm	[rad/s]	Transmission output speed
Pos_AccPedal_Ctrl	[%]	Calculated accelerator pedal position from the longitudinal controller
Pos_AccPedal_Stim	[%]	Stimulus accelerator pedal position from maneuver
Pos_BrakePedal_Ctrl	[%]	Calculated brake pedal position from the longitudinal controller

Name	Unit	Description
Pos_BrakePedal_Stim	[%]	Stimulus brake pedal position from maneuver
Pos_ClutchPedal_Maneuver	[%]	Stimulus clutch pedal position from maneuver
Reset_States	[0|1]	Reset
State_Engine	[0|1|2|3|4]	Engine state: ▪ 0: Engine off/ Error ▪ 1: Ignition on ▪ 2: Starter on ▪ 3: Engine running ▪ 4: Engine shutdown
Sw_Gear_External	[1|2]	Switch for user-defined external gear ▪ 1: On ▪ 2: Off
Sw_Gear_Set	[0|1]	Switch for stimulus gear from maneuver ▪ 0: Off ▪ 1: On
Sw_Trm_Mode	[1|2]	Switch for enabling manual or automatic transmission ▪ 1: Manual ▪ 2: Automatic
v_Vehicle	[m/s]	Vehicle velocity
v_Vehicle_Ref	[m/s]	Reference vehicle velocity

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 🕮).
Gear	[]	Gear
Pos_AccPedal	[%]	Accelerator pedal position
Pos_BrakePedal	[%]	Brake pedal position
Pos_ClutchPedal	[%]	Clutch pedal position
SelectorLever	[-3|-2|-1|0|1]	Selector lever position ▪ -3: Tip shift ▪ -2: Park ▪ -1: Reverse ▪ 0: Neutral ▪ 1: Drive

Parameters

The following table shows the parameters:

Name	Unit	Description
Const_a_Tractive_Res_UpLim	[m/s^2]	Tractive resistance acceleration threshold for startup
Const_a_x_Vehicle_LowLim	[m/s^2]	Hard braking threshold for gear hold
Const_a_x_Vehicle_UpLim	[m/s^2]	Acceleration threshold for startup
Const_D_Gain_Startup	[s]	D gain of accelerator pedal startup controller
Const_I_Gain_ClutchPedal	[1/s]	I gain of clutch pedal controller
Const_I_Gain_StartUp	[1/s]	I gain of accelerator pedal startup controller
Const_n_Engine_StartUp_Set	[rpm]	Engine speed setpoint during startup
Const_P_Gain_ClutchPedal	[]	P gain of clutch pedal controller
Const_P_Gain_StartUp	[]	P gain of accelerator pedal startup controller
Const_t_Gear_Neutral	[s]	Time of gear passing through neutral during gear change
Const_t_Hold_ClutchPedal	[s]	Disengaged clutch hold time
Const_v_x_Vehicle_LowLim	[m/s]	Small velocity threshold for driving with slipping clutch
Map_Gear_Startup	[]	Startup gear
Map_i_Ratio	[]	Gear ratio = f(gear)
Map_n_Engine_Downshift_Lim	[rpm]	Down shift engine speed limit = f(Pos_AccPedal)
Map_n_Engine_Upshift_Lim	[rpm]	Up shift engine speed limit = f(Pos_AccPedal)
Map_omega_Out_Trm_Shift_Bias	[rpm]	Transmission output speed shift bias = f(tractive resistance acceleration)
Map_Pos_AccPedal	[0_1]	Accelerator pedal factor during gear change = f(Pos_ClutchPedal)
Map_Pos_ClutchPedal_Disengagement	[%]	Clutch pedal disengagement position not during startup = f(t)
Map_Pos_ClutchPedal_Disengagement_StartUp	[%]	Clutch pedal disengagement position during startup = f(t)
Map_Pos_ClutchPedal_Engagement	[%]	Clutch pedal engagement position not during startup = f(t)
Map_Pos_ClutchPedal_Engagement_StartUp	[%]	Clutch pedal engagement position during startup = f(t)
StepSize	[s]	Simulation step size
Sw_Upshift_Gear_Skipping	[1|2]	Gear skipping switch during up shift ▪ 1: On ▪ 2: Off

Handling of unused inports

Depending on the demo model and the library, some of the inport signals might not be used. In this case, these inports cannot be connected to dummy values. They should be connected to constant values instead.

These are defined in the following table:

Inport Name	Not Used If
AccBr_Mode	Its value = 0
Pos_AccPedal_Stim	Its value = 0
Pos_BrakePedal_Stim	Its value = 0
Gear_External	Sw_Gear_External = 2 OR Sw_Trm_Mode ≠ 1
Gear_Max_Maneuver	Its value = 0
Gear_Min_Maneuver	Its value = 0
Sw_Gear_Set	Its value = 0

Related topics

References

Gear Shifter - Automatic Transmission Shifting........152
Gear Shifter - Manual Transmission Shifting........144
Gear Shifter (Version 11.0 - 13.0)........143

Lane Sensor

Introduction

Describes the lane sensor calculations that were performed by the ROAD subsystem and the former LANESENSOR_ENABLE and LANESENSOR_PARAMETERS blocks.

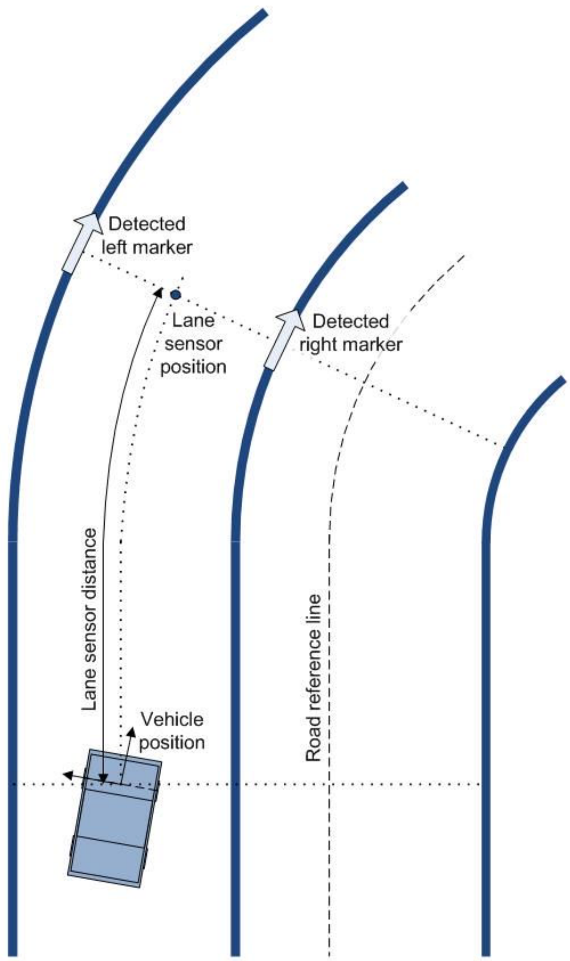
Lane sensor calculations

The lane sensor provides information on the number, position, size and type of lanes and markers on the road at a certain position in front or behind the vehicle.

The sensor position is defined with the distance to the vehicle, which is always measured in parallel to the road reference line.

You can define multiple lane sensors by specifying the lane sensor position input Pos_LaneSensor[m] as a vector signal.

Among all available road markers the lane sensor detects those two markers, one to the right and one to the left, that are nearest to the vehicle lateral offset. For these two markers the sensor calculates the x-, y- and z-position and the yaw angle, and provides information on the marker width and type. If the vehicle is outside of the road, then only the marker at the road border is found and returned. If no markers are available the sensor outputs default values.



In addition to the two markers nearest to the right and left of the vehicle the width and type information for all lanes and markers is available at the sensor output signal bus.

The lane sensor is not evaluated if the vehicle is driving on a junction, or if the lane sensor is located on a junction.

Details of lane sensor outputs

The lane widths calculated by the Road block do not include the width of the road markers or lines that separate the lanes. The lateral position of road markers is always given with respect to the center of the marker or line.

Details of lane sensor outputs

The lane widths calculated by the Road block do not include the width of the road markers or lines that separate the lanes. The lateral position of road markers is always given with respect to the center of the marker or line.

The line properties that can be defined in the ModelDesk Road Generator are described by a three-digit number, where each digit describes one line property:

Digit	Description
1.	Line type of first line
2.	Line type of second line
3.	Color

The following table shows the possible line type values:

Value	Line type
1	None
2	Solid
3	Dashed
4	Bicolored

The following table shows the possible color values:

Value	Color
1	Custom
2	Yellow
3	White

Examples The following table shows some exemplary line type codings:

Number of lines	Line 1	Line 2	Color	Line type code
1	solid	–	white	213
1	dashed	–	yellow	312
1	bicolored	–	custom	411
2	solid	solid	white	223
2	solid	dashed	custom	231
2	dashed	bicolored	custom	341

Former outports of the ROAD subsystem

LaneSensor The LaneSensor group contains signals that provide all lane sensor signals. This refers to the Lane Sensor in former versions.

Name	Unit	Description
Curvature	[1/m]	Road curvature at the lane sensor position The curvature is calculated for the road reference line.
Distance_Left	[m]	Lateral distance from lane sensor position to the nearest left line.
Distance_Right	[m]	Lateral distance from lane sensor position to the nearest right line.
LaneWidth_0	[m]	Center lane width (lane index 0)
LaneWidth_Left1	[m]	First left lane width (lane index -1). Zero if there is no lane.
LaneWidth_Left2	[m]	Second left lane width (lane index -2). Zero if there is no lane.
LaneWidth_Left3	[m]	Third left lane width (lane index -3). Zero if there is no lane.
LaneWidth_Left4	[m]	Fourth left lane width (lane index -4). Zero if there is no lane.

Name	Unit	Description
LaneWidth_Right1	[m]	First right lane width (lane index 1). Zero if there is no lane.
LaneWidth_Right2	[m]	Second right lane width (lane index 2). Zero if there is no lane.
LaneWidth_Right3	[m]	Third right lane width (lane index 3). Zero if there is no lane.
LaneWidth_Right4	[m]	Fourth right lane width (lane index 4). Zero if there is no lane.
LineWidth_Left	[m]	Line width of the left line
LineWidth_Right	[m]	Line width of the right line
Line_Type_Left	[]	Line type of the left line
Line_Type_Right	[]	Line type of the right line
MarkerWidth_Left4	[m]	Line width of line at left of fourth left lane. Zero if there is no marker.
MarkerWidth_Left3	[m]	Line width of line at left of third left lane. Zero if there is no marker.
MarkerWidth_Left2	[m]	Line width of line at left of second left lane. Zero if there is no marker.
MarkerWidth_Left1	[m]	Line width of line at left of first left lane. Zero if there is no marker.
MarkerWidth_Left0	[m]	Line width of line at left of center lane. Zero if there is no marker.
MarkerWidth_Right0	[m]	Line width of line at right of center lane. Zero if there is no marker.
MarkerWidth_Right1	[m]	Line width of line at right of first right lane. Zero if there is no marker.
MarkerWidth_Right2	[m]	Line width of line at right of second right lane. Zero if there is no marker.
MarkerWidth_Right3	[m]	Line width of line at right of third right lane. Zero if there is no marker.
MarkerWidth_Right4	[m]	Line width of line at right of fourth right lane. Zero if there is no marker.
MarkerType_Left4	[]	Line type of line at left of fourth left lane. -1 if there is no marker
MarkerType_Left3	[]	Line type of line at left of third left lane. -1 if there is no marker
MarkerType_Left2	[]	Line type of line at left of second left lane. -1 if there is no marker
MarkerType_Left1	[]	Line type of line at left of first left lane. -1 if there is no marker
MarkerType_Left0	[]	Line type of line at left of center lane. -1 if there is no marker
MarkerType_Right0	[]	Line type of line at right of center lane. -1 if there is no marker
MarkerType_Right1	[]	Line type of line at right of first right lane. -1 if there is no marker
MarkerType_Right2	[]	Line type of line at right of second right lane. -1 if there is no marker
MarkerType_Right3	[]	Line type of line at right of third right lane. -1 if there is no marker
MarkerType_Right4	[]	Line type of line at right of fourth right lane. -1 if there is no marker
Network_Item_Id	[]	ID number of the road the lane sensor is located on. The ID is visible in the Road Generator in ModelDesk.
Network_Item_Type	[1|2|-1]	Type of network item at the position of the lane sensor: ▪ 1: road element in positive direction ▪ 2: road element in negative direction ▪ -1: invalid (either lane sensor or ASM vehicle is located on junction)
xPos_Left	[m]	x-position of left road marker in the earth coordinate system.
xPos_Right	[m]	x-position of right road marker in the earth coordinate system.
YawAngle_Left	[deg]	Yaw angle of the left line in earth coordinates
YawAngle_Left_RelVehicle	[deg]	Yaw angle of the left line relative to vehicle yaw angle
YawAngle_Right	[deg]	Yaw angle of the right line in earth coordinates

Name	Unit	Description
YawAngle_Right_RelVehicle	[deg]	Yaw angle of the right line relative to vehicle yaw angle.
yPos_Left	[m]	y-position of left line at the position of the lane sensor in the earth coordinate system.
yPos_Right	[m]	y-position of right line at the position of the lane sensor in the earth coordinate system.
zPos_Left	[m]	z-position of left road marker in the earth coordinate system.
zPos_Right	[m]	z-position of left road marker in the earth coordinate system.

Related topics

References

Lane Sensor Enable.. 161
Lane Sensor Parameters.. 162

Lane Sensor Enable

Description

The LANESENSOR_ENABLE block enables the lane sensor calculations. The lane sensor calculations are performed in the Road block. The LANESENSOR_ENABLE block is located at /MDLUserInterface/Environment/MDL_PAR.

Parameters

The following table shows the parameters:

Name	Unit	Description
Sw_LaneSensor_Enable	[0|1]	Enable lane sensor calculations

Related topics

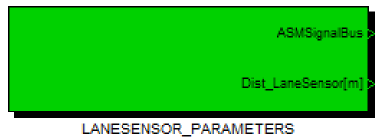
References

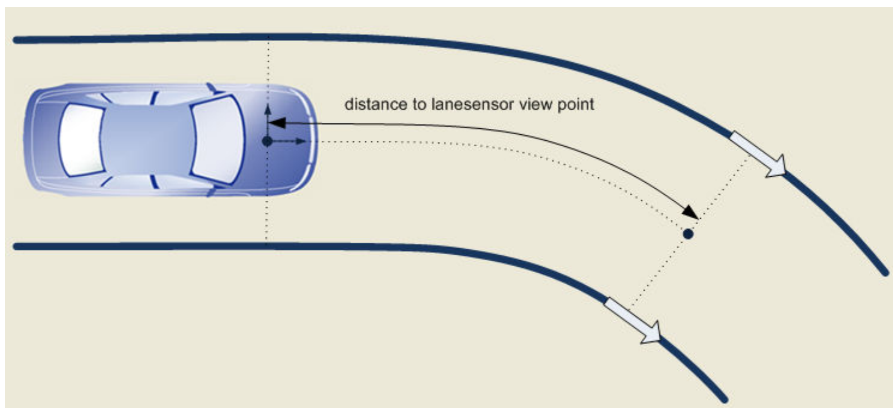
Lane Sensor.. 157
Lane Sensor Enable (ModelDesk Parameterizing 🕮)

Lane Sensor Parameters

Description

The LANESENSOR_PARAMETERS block defines the distance to the lane sensor view point.





Outports

The following table shows the outports.

Name	Unit	Description
Dist_LaneSensor	[m]	Distance to lane sensor view point

Parameters

The following table shows the parameters.

Name	Unit	Description
Const_Dist_LaneSensor	[m]	Distance to lane sensor view point

Related topics

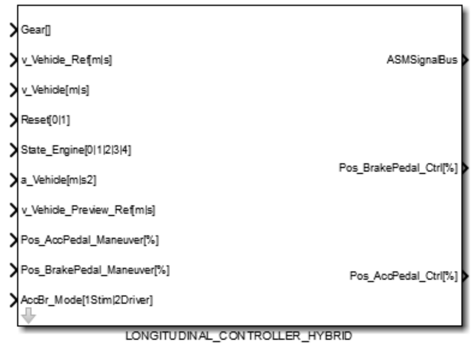
References

Lane Sensor..157
Lane Sensor Parameters (ModelDesk Parameterizing 📖)

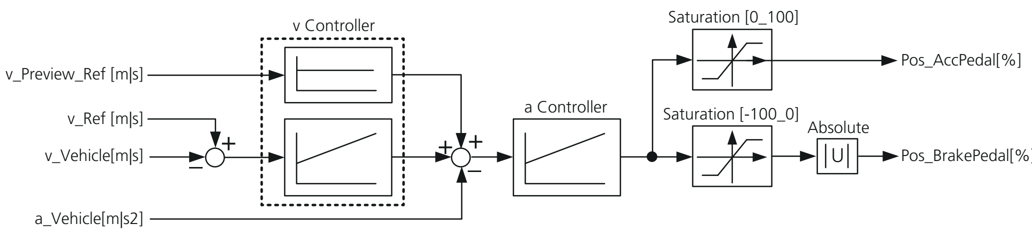
Longitudinal Controller Hybrid 5.0

Description

The LONGITUDINAL_CONTROLLER_HYBRID block controls the accelerator and brake pedals to follow a given reference velocity.



The block is modeled as a cascade controller with two PI controllers. The following figure shows the schematic of the block.



The velocity controller generates an acceleration set signal, which is used to generate an error signal for the acceleration controller. The control signal of the acceleration controller is then interpreted as an acceleration or brake demand depending on its sign.

The pedals are controlled in such a way that one pedal is pressed at a time. The minimum time needed to switch between different pedals can be parameterized. To simulate a more realistic pedal actuation, you can limit the maximum allowed rate of pedal actuation. Moreover, the driver can tolerate a specific velocity difference. This tolerated value is set as a function of the reference speed. If the actual relative velocity lies within the parameterized limits, the driver does not contribute actively to the pedals actuation.

Inports

The following table shows the inports:

Name	Unit	Description
a_Vehicle	[m/s^2]	Vehicle acceleration
AccBr_Mode	[1|2]	Mode signal for accelerator and brake pedal: ▪ 1. Stimulus ▪ 2. Driver
Gear	[]	Current gear

Name	Unit	Description
Pos_AccPedal_Maneuver	[%]	Accelerator pedal position from maneuver
Pos_BrakePedal_Maneuver	[%]	Brake pedal position from maneuver
Reset	[0|1]	Reset of states
State_Engine	[0|1|2|3|4]	Engine state: ▪ 0: Engine off/Error ▪ 1: Ignition on ▪ 2: Starter on ▪ 3: Engine running ▪ 4: Engine shutdown
v_Vehicle	[m/s]	Vehicle velocity
v_Vehicle_Preview_Ref	[m/s]	Preview vehicle reference velocity
v_Vehicle_Ref	[m/s]	Vehicle reference velocity

Outports

The following table shows the outports:

Name	Unit	Description
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 📖).
Pos_AccPedal_Ctrl	[%]	Controlled accelerator pedal position
Pos_BrakePedal_Ctrl	[%]	Controlled brake pedal position

Parameters

The following table shows the parameters:

Name	Unit	Description
Const_I_aCtrl_Pedals	[1/s]	I gain of acceleration controller
Const_I_vCtrl_Pedals	[1/s]	I gain of velocity controller
Const_P_aCtrl_Pedals	[]	P gain of acceleration controller
Const_Pos_BrakePedal_Standstill	[%]	Brake pedal actuation in standstill
Const_Preview_vCtrl_Pedals	[]	Preview gain of velocity controller
Const_P_vCtrl_Pedals	[]	P gain of velocity controller
Const_Rate_Max_AccPedal	[%/s]	Maximum allowed rate of accelerator pedal
Const_Rate_Max_BrakePedal	[%/s]	Maximum allowed rate of brake pedal
Const_t_Min_Pedals	[s]	Minimum time needed to switch between different pedals
Map_v_Dead_Driver	[km/h]	Tolerated relative velocity = f(v_Vehicle_Ref)
StepSize	[s]	Simulation step size
Sw_v_Dead_Driver	[0|1]	Switch to activate the tolerated relative velocity: ▪ 0: Off ▪ 1: On

Related topics

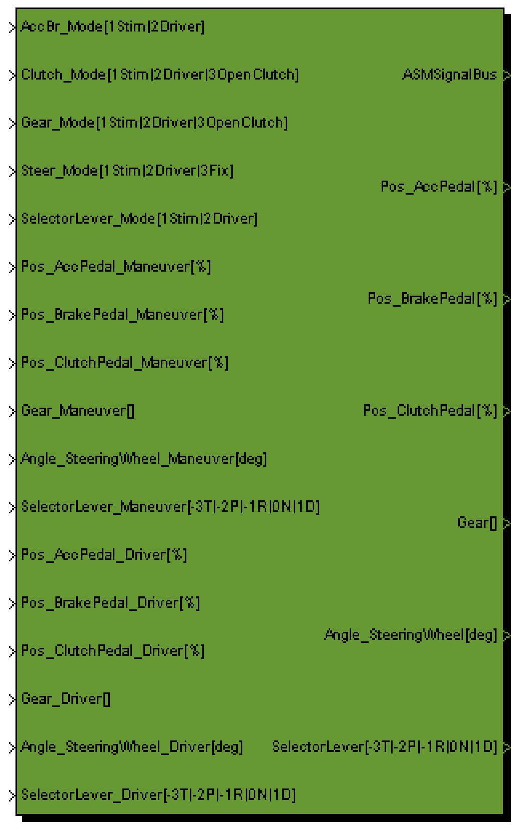
References

Long Controller Hybrid V5 (ModelDesk Parameterizing)

Signal Selection

Description

The SIGNAL_SELECTION subsystem is the interface between the maneuver, driver, and vehicle models. Depending on the maneuver definition the pedals, gear, and steering wheel are actuated by stimulus or controlled by the driver.



SIGNAL_SELECTION

The pedal positions, gear, and steering wheel angle from the driver or from the maneuver (stimulus) are transmitted to the engine, drivetrain, and vehicle dynamics models according to mode signals from the maneuver. The mode signals are described below.

- The AccBr_Mode signal defines the mode for the accelerator and brake pedals. These are the modes:
  - Stimulus (1): The stimulus signals from the maneuver are transmitted to the engine and vehicle dynamics models.

- Driver (2): The driver signals from the longitudinal controller are transmitted to the engine and vehicle dynamics models.
- The **Clutch_Mode** signal defines the mode for the clutch pedal. These are the modes:
  - Stimulus (1): The stimulus signal from the maneuver is transmitted to the drivetrain model.
  - Driver (2): The driver signal from the longitudinal controller is transmitted to the drivetrain model.
  - OpenClutch (3): The clutch is opened by the maneuver.
- The **Gear_Mode** signal defines the mode for the gear signal. These are the modes:
  - Stimulus (1): The stimulus signal from the maneuver is transmitted to the drivetrain model.
  - Driver (2): The driver signal from the longitudinal controller is transmitted to the drivetrain model.
  - OpenClutch (3): The last gear is held.
- The **SelectorLever_Mode** signal defines the mode for the selector lever for automatic transmission. These are the modes:
  - Stimulus (1): The stimulus signal from the maneuver is transmitted to the soft ECU model.
  - Driver (2): The driver signal from the gearshift model is transmitted to the soft ECU model.
- The **Steer_Mode** signal defines the mode for the steering wheel angle. These are the modes:
  - Stimulus (1): The stimulus signal from the maneuver is transmitted to the vehicle dynamics model.
  - Driver (2): The driver signal from the lateral controller is transmitted to the vehicle dynamics model.
  - Fix (3): The last steering wheel angle is held.

Inports

The following table shows the inports:

Name	Unit	Description
AccBr_Mode	[1|2]	Mode signal for accelerator and brake pedal ▪ 1: Stimulus ▪ 2: Driver
Angle_SteeringWheel_Driver	[deg]	Steering wheel angle from lateral controller
Angle_SteeringWheel_Maneuver	[deg]	Steering wheel angle stimulus from maneuver scheduler
Clutch_Mode	[1|2|3]	Mode signal for clutch pedal ▪ 1: Stimulus ▪ 2: Driver ▪ 3: OpenClutch
Gear_Driver	[]	Gear from longitudinal controller
Gear_Maneuver	[]	Gear stimulus from maneuver

Name	Unit	Description
Gear_Mode	[1|2|3]	Mode signal for gear ▪ 1: Stimulus ▪ 2: Driver ▪ 3: OpenClutch
Pos_AccPedal_Driver	[%]	Accelerator pedal position from longitudinal controller
Pos_AccPedal_Maneuver	[%]	Accelerator pedal position stimulus from maneuver scheduler
Pos_BrakePedal_Driver	[%]	Brake pedal position from longitudinal controller
Pos_BrakePedal_Maneuver	[%]	Brake pedal position stimulus from maneuver scheduler
Pos_ClutchPedal_Driver	[%]	Clutch pedal position from longitudinal controller
Pos_ClutchPedal_Maneuver	[%]	Clutch pedal position stimulus from maneuver scheduler
SelectorLever_Driver	[]	Selector lever position from driver
SelectorLever_Maneuver	[]	Selector lever position from maneuver
SelectorLever_Mode	[1|2]	Mode signal for selector lever ▪ 1: Stimulus ▪ 2: Driver
Steer_Mode	[1|2|3]	Mode signal for steering wheel angle ▪ 1: Stimulus ▪ 2: Driver ▪ 3: Fix

Outports

The following table shows the outports:

Name	Unit	Description
Angle_SteeringWheel	[deg]	Steering wheel angle
ASMSignalBus	[]	Signal bus that contains signals of ASM components. Refer to ASMSignalBus (ASM User Guide 📖).
Gear	[]	Gear
Pos_AccPedal	[%]	Accelerator pedal position
Pos_BrakePedal	[%]	Brake pedal position
Pos_ClutchPedal	[%]	Clutch pedal position
SelectorLever	[]	Selector lever position for automatic transmission: ▪ -3: TipShift ▪ -2: Park ▪ -1: Reverse ▪ 0: Neutral ▪ 1: Drive

Parameters

This block does not contain mask parameters.

New Features/Migration History of the ASM Environment Blockset

Introduction

The following topics provide an overview of the changes to the ASM products in the previous Releases.

For an overview of the new features and migration of the current Release, refer to Automotive Simulation Models (ASM) (New Features and Migration 📖).

Where to go from here

Information in this section

General Changes to the ASM Environment Blockset

Release 2019-B

Geometrically optimal trajectory With a new Python script, you can automatically calculate an optimal geometrical trajectory of a road based on the road geometry. In a graphical user interface, you can plot the calculated trajectory and then export it as free trajectory to the corresponding road in ModelDesk.

Maneuver models ModelDesk provides the *compatibility mode* for working with maneuvers created with Release 2017-B and earlier.

With Release 2019-B, the compatibility mode is provided for the last time. It will be discontinued with Release 2020-A. Therefore, it is mandatory to migrate your projects from Releases 2017-B and earlier with the current Release.

Release 2018-B

The new GPS_POSITION block converts a given (x,y)-position to GPS longitude and latitude.

Release 2018-A

ASM Driver A new longitudinal driver model based on a different control strategy is introduced. The new model simulates a more realistic pedal actuation. The driver model also adapts the vehicle velocity to the road height profile. This ensures a stable vehicle guidance when driving uphill and downhill without losing road contact. Refer to Driver on page 12.

ASM Road The LANE_SENSOR block was replaced with a new LANE_SENSOR_DYNAMIC block. The new block has several new features, such as:

- Configurable interpretation of lane borders.
- Detection of virtual lanes on junctions.
- Configurable measurement points on the vehicle from which the distances to the lane markings are accurately measured.

The new block also lets you specify the number of lanes to the left and to the right of the current lane that are to be considered by the lane sensor.

ASM Maneuver ASM Maneuver supports the new ModelDesk Scenario Editor. For more information, refer to ModelDesk Scenario Creation 📖.

With the new ASM Maneuver subsystem, the following additional features are available:

- Stimulus of the selector lever position.
- Maneuver user signals support a time delay of the specified signal values.
- Multiple segment transition conditions can be concatenated with the OR operator.v

Release 2017-A

ASM Driver The driver is now able to automatically drive backwards by using the manual transmission. In case of a negative reference velocity course, the driver controls the clutch and the gear to follow the velocity course.

Moreover, the driver can now be used for the simulation with a start-stop system.

ASM Road The lane sensor now provides additional information about the direct and oncoming preferred lanes.

The signal bus of the LaneSensor outport has been extended by two new signals for the preferred lanes in the direct and oncoming direction.

Release 2016-B

Look-up table migration The discontinued Simulink blocks Lookup and Lookup2D in the ASM library blocks were updated to the new standard Simulink Look-up table (n-D) block. Refer to Changes to all ASM Products (ASM User Guide 📖).

The look-up tables were updated in the following blocks within this library:

- BRAKE_HYDRAULICS_VARIANT
- BRAKE_PNEUMATICS_VARIANT
- CONTROLLER
- GEAR_SHIFTER_9_0

License check of ASM Utils blocks The ASM_UTILS license was discontinued. The ASM Utils blocks now check the license of the ASM blockset in which they are used.

The Utils blocks in the following blocks within this library were updated:

- BASIC_ROADS
- GEAR_SHIFTER_9_0
- LATERAL_CONTROL1
- MANEUVER_SCHEDULER
- ROAD

Release 2013-B

Several blocks have a new reset inport. During migration of the model, the new inports are connected to Constant blocks and therefore the reset functionality cannot be used yet. If the reset functionality is required, the inport must be connected to the reset output from the maneuver scheduler block that is available on the ASMSignalBus.

Release 2013-A

The ASM Environment road model now supports the definition of road networks consisting of roads and junctions. Vehicles drive on routes on the road network. Typical application scenarios are the development and testing of advanced driver assistance systems such as intersection assistants and Car2x applications.

Road networks are created using ModelDesk's Road Generator. The road network is visualized with MotionDesk.

Example road networks are included in the demo models.

The maneuver now supports driving on road networks. The ASM vehicle is initialized on a certain route on the road network. Certain distance calculations can be used to trigger segment transitions in maneuvers. A new segment type for longitudinal movement makes it easy to define a stop at a junction. An initial velocity can be set at maneuver start.

The format of the maneuver MAT files has changed. The maneuver MAT files are automatically migrated during the migration process. To migrate maneuver MAT files manually, use the script asm_migrate_maneuver.

The format of the road MAT files has changed. The road MAT files are automatically migrated during the migration process. To migrate road MAT files manually, use the script `asm_migrate_road`.

Release 7.4

The lane sensor can now be parameterized using ModelDesk.

Release 7.3

ASM Environment now supports the definition of a variable number of lanes on the road. Complex maneuvers and traffic scenarios in multiple lanes can be defined. Typical application scenarios are the development and testing of advanced driver support systems, such as adaptive cruise control and lane departure warning systems.

The road supports up to five lanes of variable width, which can be set with freely configurable lane sections along the road.

When maneuvers and traffic scenarios are defined, movement in the available lanes, such as a smooth lane change, can be defined for each vehicle by setting a target lane index. If the target lane is not available on the current section of the road, you can choose whether the vehicle is allowed to drive outside of the road or must stay in the outermost available lane.

The new features are supported by all the dSPACE tools: Roads with lanes can be easily created in ModelDesk and the simulation result animated in MotionDesk.

Lines of different type, color and size can be set between the lanes on the road. A lane sensor model provides the information on these lines along the road at various adjustable distances from the ASM vehicle.

The road definition has changed from a segment-based approach, where all road properties such as height or friction are set for each road segment, to a more flexible approach, where road properties can be set independently of the road segments that define the profile. In the new approach, road properties such as height, lateral slope, lanes and road environment are defined as a function of the path s along the road. Height and lateral slope profiles can be imported with MAT files.

In previous releases, local road surface properties, such as bumps or a different friction value, were defined with four strips on each road segment. The new road definition format does not use the four strips. Instead, local surface properties are defined with patches that can be placed anywhere on the road.

The maneuver now supports driving in lanes. The initial vehicle height can now be set as an offset to the tire radius. The steering wheel angle is set smoothly to zero after the maneuver is finished. Some previously internal definitions, for example, brake pedal position in certain maneuver states, can be made in the model. The number of maneuver user signals has increased to 30. The number of trigger inputs has increased to 25.

To avoid the driver model interfering with active steering ECUs such as a lane keeping assistant, a tolerance zone has been added to the driver model. Within this tolerance zone, the controller has only a small influence on the steering, while outside it, the controller behaves as usual and keeps the vehicle on the target course. If the width of this tolerance zone is set to zero, the lateral controller controls the vehicle as usual.

To implement the set of new features in this release, the **Environment** subsystem of the demo models has been substantially restructured. Due to the structural changes, not all new features can be added automatically to the migrated models. To obtain the full functionality of all new features for migrated models, it is recommended to replace the **Environment** subsystem in the migrated model with the **Environment** subsystem in the demo model, and adapt the model and ModelDesk experiment accordingly to reflect the structural changes.

Release 7.0

A tool to create ASM roads from road measurement data or from GPS data is provided.

Release 6.2	Now, this blockset is available as operator version. This is a model variant specifically designed for offline simulation with Simulink®. The model offers the same functionality, simulation quality and parameterization options as the standard simulation package. The operator version is compatible with the standard model (developer version) and can be parameterized in ModelDesk, the parameterization software. The fundamental difference with this model is the way the library components are implemented: They are encapsulated in separate systems to ensure good performance during Simulink simulation. The systems are accessible in the model so that their input/output behavior can be studied.

History of the AMBIENT Block

Release 6.4

The AMBIENT block was added to the ASM_Environment_lib. In previous releases this block existed without any library link. No migration steps are necessary.

Related topics

References

Ambient..77

History of the BRAKE_PNEUMATICS_VARIANT Block

Release 2018-A

This block is obsolete. During migration, the links to the BRAKE_PNEUMATICS_VARIANT block are changed to the former version.

History of the BASIC_ROADS Block

Release 2018-A

The dimensions of one inport and one outport were changed. During migration, these changes are compensated for so that the old block behavior remains unchanged.

Release 2015-B

There are changes due to Simulink diagnostics.

Release 2013-A

Additional outports have been added to match the ROAD block.

Release 7.3

The output of the BASIC_ROADS block has been restructured in the same way as for the ROAD block. Both blocks now have the same output bus structure.

Release 7.0

The rate-limiter sample modes at slope and lateral slope signals have been changed from "continuous" to "inherited". Thus, the signal is of discrete sample time and not of continuous one. Now the ASM_Truck model can be used in combination with ASM_VehicleDynamics Operator.

Release 6.4

This block is now dynamically sized to be able to handle more tires for truck and trailer.

The calculation of the z-position was wrong if the lateral and longitudinal slopes were used. This bug has been fixed.

Some inports have been renamed.

Old Inport Name	New Inport Name
Pos_CP_CoorSys_E[FL;FR;RL;RR][x;y][m]	Pos_CP_CoorSys_E[:][x;y][m]
Fric_Coeff_BasicRoad[FL;FR;RL;RR][]	Fric_Coeff_CP_BasicRoad[:][]
Sw_Tire_Parameter_Set_BasicRoad[FL;FR;RL;RR][1|2|3|4]	Sw_Tire_Parameter_Set_CP_BasicRoad[:][1|2|3|4]

Release 6.2

The slope, lateral slope, friction coefficients and tire parameters have been moved from internal constant values to block inputs. They can therefore be used in the model without breaking the link to the library block.

During migration from previous versions, the subsystem migrate60_newinports is inserted before the BASIC_ROADS block. This subsystem contains default values for all new input ports. All inputs are connected to this subsystem.

Related topics

References

Basic Road.. 78

History of the BRAKE_HYDRAULICS_VARIANT Block

Release 2018-A

This block is obsolete. During migration, the links to the BRAKE_HYDRAULICS_VARIANT block are changed to the former version.

Release 6.6

The Const_Factor_BrakeDisc[FL;FR;RL;RR][m3] inport was renamed to Const_Factor_BrakeDisc[m3]. The dimension was modified from 4 to 1. During the migration process a sum block is added before this inport.

Release 6.5

The block supports connection with the brake booster model from the ASM Vehicle Dynamics Blockset.

The block contains four new inports. During migration, the new inports are connected to constant blocks with default values.

Related topics

References

Brake Hydraulics Variant_5_0 116

History of the COMMON_DRIVER_PARAMETERS Block

Release 6.2

The driver now detects information on additional loads. This is especially helpful for truck simulation. The driver parameterization no longer needs adapting when the vehicle mass changes. The required information is transmitted to the driver via signal lines. The old Const_m_Vehicle, Const_Inertia_Tensor_Vehicle, and Const_PosVec_CoG_Vehicle driver parameters were removed.

The Const_m_Vehicle parameter was removed. The necessary information is now provided to the relevant blocks via signal lines.

Related topics

References

Common Driver Parameters 13

History of the CONTROLLER Block

Release 2018-A

During migration, the links to the CONTROLLER block are changed to the former version.

Release 2015-A	The new CONTROLLER block calculates the tractive resistance force instead of the tractive resistance acceleration. During migration, this calculation is compensated to keep the old behavior unchanged.
Release 2014-A	Calculation of the tractive resistance acceleration has been added and routed as an outport. It is used by the GEAR_SHIFTER block as an indication of the effort needed to overcome the driving resistances.
Release 2013-B	A reset inport has been added.
Release 7.4	Braking during slow acceleration after startup is now avoided.
Release 7.1	The feed forward part of the controller now also contains a power-based approach. This calculation avoids steps in the accelerator pedal after gear changes. The new Sw_FeedForward_Mode parameter was added.
Release 7.0	The Const_dens_Air parameter was renamed to Const_Dens_Air. The Const_Coef_Cw_x_Vehicle parameter was removed. This information is now provided as an input signal. The new Coef_Cw_x_Vehicle[] inport was added. During migration a subsystem with a constant block is connected to the new input and its constant value is set to the value of MDL.Environment .Driver.LongitudinalController.Controller.Const_Coef_Cw_x_Vehicle.v. For correct simulation results, the VehicleDynamics.Aerodynamics.Coefficients.Cw_x[] signal from the ASMSignalBus should be connected to the new Coef_Cw_x_Vehicle[] inport. This must be done manually. For models created with dSPACE Release 6.6, refer to the ASM VehicleDynamics demo model as an example of how to connect the bus signal using the Environment interface block (ASM_VehicleDynamics/MDL/Environment/EnvironmentInterface_In). If you use ModelDesk automation, note that the Environment.CONTROLLER.Const_Coef_Cw_x_Vehicle parameter was removed. The Environment.CONTROLLER.Const_dens_Air parameter was renamed to Environment.CONTROLLER.Const_Dens_Air.
Release 6.2	The driver now detects information on additional loads. This is especially helpful for truck simulation. The driver parameterization no longer needs adapting when

the vehicle mass changes. The required information is transmitted to the driver via signal lines. The old Const_m_Vehicle, Const_Inertia_Tensor_Vehicle, and Const_PosVec_CoG_Vehicle driver parameters were removed.

The vehicle mass information was previously provided by a from-goto connection from the COMMON_DRIVER_PARAMETERS block.

It must now be provided as an input signal via a signal line. The mass signal is connected automatically during the migration process. If problems occur with automatic connection, refer to the ASM_VehicleDynamics default model or to the following description:

- The VehicleMovement.VehicleMassAndAdditionalLoads.m_TotalVehicle[kg] bus signal must be selected using a Bus Selector block from the VehicleDynamics_Signals bus. This selected signal must be connected to the m_Vehicle_Total[kg] block input.

Related topics

References

Controller_10_0 ... 119

History of the DRIVER_CURVATURE Block

Release 2018-B

The new DRIVER_CURVATURE block provides the curvature for the reference velocity calculation.

The block supports two modes:

1. ROAD: The curvature is calculated on the basis of the geometric form factor of the road network.
2. DRIVER: The curvature is calculated on the basis of the desired trajectory of the ASM vehicle.

History of the DRIVETRAIN_INIT Block

Release 2020-B

This block is new. You can use it to trigger drivetrain initialization for scenarios with an initial vehicle speed.

History of the GEAR_SHIFTER Block

Release 2020-B	The block has a new inport: Enable_v_Vehicle_Init. To keep the model behavior unchanged, it is connected to constant block during migration. The new inport is used to enable drivetrain initialization for scenarios with an initial vehicle speed.
Release 2018-A	The new v_Vehicle_Ref_Preview[m|s] inport was added. During migration, this inport is connected to a Constant block so that the previous block behavior remains unchanged.
Release 2017-A	A new inport for the start-stop system status has been added: State_StartStop[1Off|2NotReady|3Ready|4EngStopped]. During migration, this inport is connected to a Constant block.
Release 2016-B	The block has a new parameter (Sw_ClutchPedal_EngineOff) to actuate the clutch pedal while the engine is not running. The behavior in hard braking scenarios where the clutch should not be disengaged if the current gear is neutral has been improved.
Release 2016-A	The block has been restructured to offer new features and make it more compatible with other ASM blocks. For example, it is now possible to use the stimulus accelerator and brake pedal signals while the GEAR_SHIFTER block controls the selector lever or the gear as well as the clutch position. Due to the considerable changes in this block, it cannot be migrated automatically. Therefore, during the migration, the link to the GEAR_SHIFTER block is changed to the former implementation version: `FormerVersions/GEAR_SHIFTER_13_0`. The former version of the block still contains some new features. To use the new GEAR_SHIFTER block, add it to your model from the ASM Environment Library. In this case, you must manually adapt the inports and outports.
Release 2015-B	The behavior after reset has been corrected.
Release 2015-A	A startup gear other than 1 is now possible. This builds up a more realistic behavior for the simulation of truck driving. In addition, the startup behavior has been improved by introducing a new PID controller instead of the previous feed forward plus PI controller. The accelerator

pedal actuation during the startup phase is now smoother and the reference engine speed can be reached more efficiently.

The block cannot be automatically migrated, due to major changes in the inports and the block parameters. Therefore, during migration, the link to the GEAR_SHIFTER block is changed to the former implementation in `FormerVersions/GEAR_SHIFTER_10_0`.

To use the new implementation, drag the GEAR_SHIFTER block from the ASM Environment Library to the model. However, a hand adaptation of the inports, outports and parameters is necessary.

Release 2014-A

The block has been revised and new approaches have been developed. The revised block ensures better handling of different driving situations, realistically simulates several behaviors, and offers more possibilities for future extension. It also gives you more control to change the behavior via user-defined parameters.

The new GEAR_SHIFTER offers the following features:

- Fully parameterizable implementation
- Flexible clutch actuation according to the driving situation
- Gear skipping during dow nshift, e.g., emergency brake
- Gear skipping during up shift, e.g., truck application
- Manual modification of the current gear via external gear triggering
- The driver can now also shift the selector lever to reverse in automatic transmission
- Tracking of small velocities with slipping clutch
- Consideration of the effort needed to overcome driving resistances during the startup
- Detection of engine stall
- Realistic simulation of behavior when the gear passes through neutral during gear shifting

The block cannot be migrated automatically due to the major changes in the inports as well as in the block parameters. Therefore, during migration, the link to GEAR_SHIFTER is changed to the former implementation, which is located in FormerVersions-GEAR_SHIFTER_9_0.

To use the new implementation, you can drag the GEAR_SHIFTER block from the ASM_DrivetrainBasic_lib library to the model. The inports, outports and parameters must be adapted manually. ROAD block.

Release 2013-B

A reset inport has been added.

The simulation step size has been added as parameter.

Release 2013-A

An initial gear is set in case of an initial velocity at maneuver start. The block interface in Simulink has not been modified.

Release 7.0	It is now possible to set the Implement logic signals as Boolean data (vs. double) option in the Operator Simulink models.
Release 6.2	The GEAR_SHIFTER block now contains a new parameter Const_Pos_AccPedal_Startup_FF. This defines the accelerator pedal position during the startup process. The driver parameterization was adapted to handle up to 25 gears. The dimension of the Map_omega_UpShift and Map_omega_DownShift shift tables was increased to handle up to 25 gears. The ASM_SHIFTABLE_LOOKUP block from the ASM Utilities library is used to evaluate the shift tables. This avoids multiple use of local mask variables. The GEAR_SHIFTER block now contains the new Const_Pos_AccPedal_Startup_FF parameter. During startup it is assumed that an accelerator pedal position defined by Const_Pos_AccPedal_Startup_FF (default: 5%) is sufficient to reach the engine speed defined by Const_n_Engine_StartUp_Set (default: 1100 rpm). If this is not the case, engine startup will fail. Most likely the engine parameterization must be modified in this case. As a workaround, the Const_Pos_AccPedal_Startup_FF parameter can be increased.
Release 6.1	If the brake and accelerator were depressed at the same time in driver mode, the selector lever position toggled between Neutral and Drive state. This bug has been fixed.
Related topics	References Gear Shifter........ 14

History of the GPS_POSITION Block

Release 2020-A	This block now also calculates the heading angle of the vehicle. Vectorized calculation is now also enabled. This block is now able to handle vectorized input signals.
Release 2019-B	The GPS calculation for the southern hemisphere has been improved.
Release 2018-B	The new GPS_POSITION block converts the x/y-position information to GPS coordinates depending on the specified conversion method (spherical or based

on the WGS 84 reference system). Additional position offset information provided by the ROAD block via user signals is also processed.

The GPS_POSITION block is included in all ASM demo models linked to the ASM Environment Library.

Related topics

References

GPS Position 82

History of the KEY_STATES Block

Release 6.4

An inport was renamed.

Old Inport Name	New Inport Name
Sw_Key[0off|1on]	Key[-1_2]

Related topics

References

Key States 63

History of the LANE_NETWORK Block

Release 2020-A

Three inports were added to the LANE_NETWORK block. With these new inports, the LANE_NETWORK block can be controlled by the new OSI_GROUNDTRUTH_INTERFACE block of the ASM Traffic Library.

Release 2019-B

The communication vector has been updated.

Release 2019-A

The ASM Environment library was extended by the initial version of the LANE_NETWORK block. You can use the block to download static road information from ModelDesk. The static road information can be used with the OSI Groundtruth Interface block.

History of the LANESENSOR_DYNAMIC_OUTPUT Block

Release 2019-A

The block provides further information about the vehicle's current lane and the adjacent lanes.

The new outputs include:

- Curvatures of lanes and curvatures of related lane markings.
- Defined driving direction on lanes, or the direction of lanes.
- Detailed information on adjacent lanes, such as the position of corresponding lane markings and lane IDs. Thus, the information is as detailed as the information on the current lane.

History of the LANESENSOR_PARAMETERS Block

Release 2018-A

During migration, the links to the LANESENSOR_PARAMETERS block are changed to the former version.

Related topics

References

Lane Sensor Parameters..162

History of the LATERAL_CONTROL1 Block

Release 2020-A

The vehicle behavior when driving backwards at low velocities was improved.

Release 2018-A

The upper limit saturation of the cornering stiffness was reduced.

Release 2016-A

The block now has new inports to switch between the different angle steering modes internally: Steer_Mode[1Stim|2Driver|3Fix] and Angle_SteeringWheel_Maneuver[deg]. The new inports are connected to the related signals during the migration process.

Release 2015-B

The steering wheel angle will be reset at a global reset.

Release 2013-B	A reset inport has been added.
Release 7.3	The LATERAL_CONTROL1 block has a new Width_Deadzone_LatCtrl[m] input. A constant block with default value zero is added to the new input during migration.
Release 7.0	Setting the preview time to zero previously caused invalid simulation results (division by zero). This has been corrected.
Release 6.4	The lateral controller had problems with steering trucks. This has been fixed now. No migration steps are necessary.
Release 6.2	The driver now detects information on additional loads. This is especially helpful for truck simulation. The driver parameterization no longer needs adapting when the vehicle mass changes. The required information is transmitted to the driver via signal lines. The old Const_m_Vehicle, Const_Inertia_Tensor_Vehicle, and Const_PosVec_CoG_Vehicle driver parameters were removed.
Release 6.2	The Const_Inertia_Tensor_Vehicle and Const_PosVec_CoG_Vehicle parameters were removed. The necessary information is now provided via signal lines. The vehicle mass information was previously provided by a from-goto connection from the COMMON_DRIVER_PARAMETERS block. It must now be provided as an input signal via a signal line. The new block inputs must be connected to signals from a bus selector which is connected to the VehicleDynamics_Signals bus. The signals are connected automatically during the migration process. If problems occur with automatic connection, refer to the ASM_VehicleDynamics default model or to the following description: ▪ The m_Vehicle_Total[kg] block input must be connected to the VehicleMovement.VehicleMassAndAdditionalLoads.m_TotalVehicle[kg] bus signal. ▪ The Pos_x_CoG_Vehicle_Total[m] block input must be connected to the first element of the bus signal VehicleMovement.VehicleMassAndAdditionalLoads.PosVec_CoG_TotalVehicle[x;y;z][m].

- The Inertia_z_Vehicle_Total[kgm2] block input must be connected to element [3,3] of the bus signal VehicleMovement.VehicleMassAndAdditionalLoads.Inertia_Tensor_Total Vehicle[3x3][kgm2].

Related topics

References

Lateral Controller 1..29

History of the LATERAL_CONTROL2 Block

Release 2016-A

The LATERAL_CONTROL2 block has been enhanced with an additional yaw rate controller. It has no preview functionality and is intended for use during steady-state cornering maneuvers. It can be activated via the maneuver definition in ModelDesk only during a circle maneuver. In all other cases, the position controller with preview is used.

The block now has new inports to switch internally between the different angle steering modes: Steer_Mode[1Stim|2Driver|3Fix] and Angle_SteeringWheel_Maneuver[deg]. The new inports are connected to the related signals during the migration process.

There are also new inports for the new yaw rate control functionality: YawRate_Vehicle[rad|s], LatCtrl_Mode[1Pos|2Yaw] and Curv_Road_Circle[1|m][-Right|_+Left]. During migration, the new inports are connected to dummy values and only the position controller is used.

Moreover, the yaw rate controller parameters (Const_Kp_YawRate_Ctrl and Const_Ki_YawRate_Ctrl) are promoted to mask parameters, which are added and initialized during the migration process.

Release 2015-B

The steering wheel angle will be reset at a global reset.

Release 2013-B

A reset inport has been added.

Release 7.3

The LATERAL_CONTROL2 block has a new Width_Deadzone_LatCtrl[m] input. A constant block with default value zero is added to the new input during migration.

Release 6.2

The controller parameters in the default parameterization have been modified to ensure better stability on straight roads. The following table shows the old and new parameter values.

Parameter	Old Value	New Value
Const_LatCtrl_kLat	2	1
Const_LatCtrl_kLat_i	2	2
Const_LatCtrl_kLat_Preview	1	0.2
Const_LatCtrl_kYaw	35	30

Note

These parameter changes are not automatically transferred to the parameterization. You have to do this manually if necessary.

Related topics

References

Lateral Controller 2.. 37

History of the LONGITUDINAL_CONTROL Block

Release 2020-B

The block has a new inport: Enable_v_Vehicle_Init. To keep the model behavior unchanged, it is connected to constant block during migration.

The new inport is used to enable drivetrain initialization for scenarios with an initial vehicle speed.

Release 2019-A

The issue of increased turnaround time with active preview time was solved by using a customized variable time delay block.

History of the LONGITUDINAL_CONTROLLER_HYBRID Block

Release 2019-A

The block was discontinued and shifted to the former versions. During migration, the link to the block is changed to the former version LONGITUDINAL_CONTROLLER_HYBRID_5_0.

Release 2017-A

Three new inports have been added to the block:

- Pos_AccPedal_Maneuver[%]
- Pos_BrakePedal_Maneuver[%]
- AccBr_Mode[1Stim|2Driver]

These inports are used to keep the driver inactive during stimulus maneuvers. Moreover, they enable a smooth pedal transition when switching between stimulus and driver maneuvers.

During migration, these inports are connected to Constant blocks.

Release 2016-B

The block has several new features:

- It is now possible to limit the actuation rate of both pedals.
- The minimum time needed to switch between different pedals can be set to a nonzero value.
- The driver can tolerate certain relative velocity.
- A new controller parameter for the preview reference speed has been added as a preparation for improved controller strategy.

For this, the following new parameters are added during migration:

- Const_Rate_Max_AccPedal and Const_Rate_Max_BrakePedal: Pedals maximum rate of change
- Cont_t_min_Pedals: Minimum time needed to switch between different pedals
- Map_v_Dead_Driver and Sw_v_Dead_Driver: Tolerated relative velocity
- Const_Preview_vCtrl_Pedals: Controller parameter for the preview reference speed

These parameters are initialized in `asmmigratepost` so that the previous behavior is kept unchanged.

Release 2016-A

The actual vehicle acceleration is used instead of the approximated value. The vehicle acceleration is fed to the block via a new inport which is connected with the corresponding signal during the migration.

Release 2015-A

This block is new. It simulates the driver behavior when following a reference velocity by controlling the accelerator and brake pedals. The block does not use the vehicle parameters and can be used as a modular block.

Related topics

References

Longitudinal Controller Hybrid 5.0 163

History of the MANEUVER_SCHEDULER Block

Release 2020-A

The MANEUVER_SCHEDULER block now supports the usage of absolute lane indices.

The block now also supports lane changes of the ego-vehicle to a lane that is defined relative to the current lane of the ego-vehicle.

Release 2019-B

The longitudinal types *stop within distance* and *final velocity* support the new *ignore driver parameters* option. If enabled, the driver parameters in the *follow road* lateral type are overridden by the longitudinal parameters.

Use the `asm_migrate_scenario` MATLAB script to perform the manual migration of maneuver and scenario MAT files created with previous dSPACE Releases.

Release 2019-A

Segments can be triggered using a new transition condition. The new transition condition evaluates the distance between the bounding boxes of two vehicles on the road reference line. This feature is available only with an ASM Traffic license.

Tables that use the distance *s* along the route can be used. In previous Releases, only tables that use the driven distance in the current segment could be used.

Use the`"asm_migrate_scenario"` MATLAB script to perform the manual migration of maneuver and scenario MAT files created with previous dSPACE Releases. The `"asm_migrate_maneuver"` script has been discontinued.

Release 2018-A

The ASM maneuver scheduler is automatically migrated to support the new ModelDesk Scenario Editor. The number of inports and outports of the ASM maneuver scheduler block is the same. However, the content of some signals has changed.

- The ExternalSignals, LastValue, and ManualControl inports, which are all vector signals, expect one additional signal at the end of the vector: the selector lever position.
- The Info outport contains the additional CurrentSequence[] signal.
- The ModeSignals outport contains the new SelectorLever_Mode[1Stim|2Driver] signal. This signal controls whether the selector lever position is a stimulus signal or controlled by the driver.
- The RefSignals outport contains the new SelectorLever_Maneuver[] signal.

Note

ModelDesk features two different modes for working with maneuvers:
- *Default maneuver mode* (recommended). This mode supports maneuvers in the ModelDesk Scenario Editor.
- *Maneuever compatibility mode* (recommended only if the migration of the maneuver automation is to be postponed). This mode supports maneuvers in the ModelDesk Maneuver Editor, as in Release 2017-B and earlier.

The maneuver compatibility mode offers the same ModelDesk user interface and automation API for maneuvers as earlier dSPACE Releases. With the new ModelDesk Scenario Editor, which is the default editor for specifying maneuvers as of Release 2018-A, the ModelDesk automation API for maneuvers has changed. Therefore, you have to adapt all existing maneuver automation scripts if you use the new ModelDesk Scenario Editor for maneuver creation. However, if ModelDesk's Maneuver Compatibility Mode is active, you can continue to use the existing automation scripts. The compatibility mode was introduced to let you determine when to perform the migration to the default maneuver mode and, subsequently, the migration of legacy maneuver automation scripts.

Note

- The maneuver compatibility mode will exist only for a limited time. The last dSPACE Release to support this mode is Release 2019-B.
- With dSPACE Releases that do not feature the compatibility mode anymore (Release 2020-A and later), it will not be possible to migrate ModelDesk maneuvers from the compatibility mode to the new ModelDesk Scenario Editor. Therefore, you must migrate maneuvers with dSPACE Releases earlier than Release 2020-A.
- Up to and including Release 2019-B, you can easily switch between the maneuver compatibility mode and the new Scenario Editor in ModelDesk. However, you have to manually adapt the ASM model if you use the compatibility mode. If the model was prepared for the maneuver compatibility mode and you want to use the default maneuver mode, you have to adapt the ASM model again.
- All new maneuver features are based on the new ModelDesk Scenario Editor. In addition, the new maneuver features stated above for Release 2018-A have been implemented for the maneuver compatibility mode.

Release 2017-B

The maneuver scheduler is improved so that it is possible to set the steering torque stimulus in addition the steering wheel angle.

Release 2015-B

There are changes due to Simulink diagnostics.

Release 2013-A	To support driving on road networks, new inports and outports are added during the migration process.
Release 7.4	During initialization of the manual driving state, a fixed value was used to initialize the vehicle height. This has been replaced by a small offset to the tire radius.
Release 7.3	The interface of the MANEUVER_SCHEDULER block has been restructured to accommodate the new signals required to control the movement of the ASM vehicle in multiple lanes. Some signal names have been changed to reflect their new usage. To reduce the number of input ports, the input signals have been grouped as bus signals: ▪ ManualControl ▪ ExternalSignals ▪ LastValues ▪ ManeuverSettings In addition output signals have been merged with the unchanged output bus signal structure. During migration, the MANEUVER block from the previous release is replaced by a subsystem that contains the new MANEUVER block and performs the mapping from the old interface to the new. Thus, the model path to the MANEUVER_SCHEDULER block is modified.
Release 6.4	In the MANEUVER_SCHEDULER block, only minor bug fixes have been done. There was a wrong unit conversion in hold segments. The maneuver segment length sometimes differed by 1 ms. User signals are now held in brake state. No migration steps are necessary.
Release 6.3	Tables and MAT files are now interpreted correctly when user maneuvers are used.
Release 6.1	The matrix dimensions for the user maneuver were decreased. It contains only data and segments which are actually used.

Related topics	References	

History of the ROAD Block

Release 2020-A	The ROAD block now supports the usage of incoming absolute lane indices.
Release 2019-B	The ROAD block was extended by several internal calculations and data structures for use with the new traffic driver feature.
Release 2018-B	The ROAD block now supports junctions that feature height gradients. Until now, the junction plate had to be level.
Release 2018-A	Seven new inports and one new outport were added. The dimensions of three inports and one outport were changed. During migration, these changes are compensated for so that the old block behavior remains unchanged.
Release 2017-B	The ROAD block supports the usage of trajectory shapes for the definition of trajectories for the ASM vehicle and traffic fellows. According to the new road functionalities, the format of the road MAT files was changed. They are automatically migrated during standard model migration. The road MAT files can also be migrated separately by using the asm_migrate_road function.
Release 2016-A	The ROAD block has been adapted for the new line definition and shapes features. The modifications are implemented within the S-function. There are only minor changes to the block interface. The signal bus at the Info outport has been extended by an additional signal: d_Veh_Ref_Line[m]. The signal bus at the TrafficFellows outport has been extended by an additional signal: d_Fellows_RefLine[m]. According to the new road functionalities, the format of the road MAT files has been changed. They are automatically migrated during standard model

migration. The road MAT files can also be migrated separately by using the `asm_migrate_road` function.

Release 2015-B	The distance calculations are executed for a dynamic number of traffic objects. Thus, the corresponding ports of the ROAD block have a dynamically sized width. There are changes due to Simulink diagnostics.
Release 2015-A	The ROAD block now supports dynamically-sized distance calculations to allow independent movement of the fellow vehicles in ASM Traffic. In models containing ASM Traffic, the signal dimension for distance calculation is increased.
Release 2014-A	The road now supports static traffic objects such as traffic signs. The traffic object data is output at the new outport TrafficObjects_Data[], which is evaluated by the new sensors added to the Traffic model. During migration, a Goto block is added to the new outport TrafficObjects_Data[]. The road definition files are updated to the new format.
Release 2013-A	To support road networks, new inports and outports have been added during the migration process.
Release 7.4	The calculation of the length inside spline segments has been modified.
Release 7.3	The interface of the ROAD block has been restructured to accommodate the new signals required for the lane and lane sensor features. To make the increased number of output signals more easily accessible, they are organized in a bus structure. Some signal names have changed to reflect their new usage. During migration, the ROAD block from the previous release is replaced by a subsystem that contains the new ROAD block and performs the mapping from the old interface to the new. Thus the model path to the ROAD block is modified.
Release 7.1	The ASM Road now provides user signals allowing important information about GPS coordinates or others to be handled. Phase-shifted bumps can be defined by an initial distance offset.

The matrix size of the road block parameter was changed. During migration, the road MAT files in the `<ProjectRoot>\Simulation.current\IniFiles\Road` folder are migrated to the new format.

To migrate further road MAT files, type `asm_migrate_road` in the MATLAB command window.

The new UserSignals outport was added.

Release 7.0

It is now possible to download an unlimited number of road segments.

The calculation of the contact points in the ROAD Simulink S-function has been improved. This improvement also reduces the time taken by contact point calculation.

Release 6.4

This block is now dynamically sized to be able to handle more tires for truck and trailer. In previous versions the ASM_Enviroment_lib and ASM_Traffic_lib contained a separate version of the ROAD block. Now the ROAD block of the ASM_Environment_lib can handle the traffic features as well.

It is now possible to define a certain lateral offset from the road centerline. Outputs for lateral offset, road width and current road segment were added.

An inport signal was renamed.

Old Inport Name	New Inport Name
Pos_CP_CoorSys_E[FL;FR;RL;RR][x;y][m]	Pos_CP_CoorSys_E[:][x;y][m]

Some outport signals were renamed.

Old Inport Name	New Inport Name
Pos_z_CP_CoorSys_E[FL,FR,RL,RR][m]	Pos_z_CP_CoorSys_E[:][m]
UnitVec_z_CP_CoorSys_E[FL,FR,RL,RR][x y z]	Fric_Coeff_CP[:][]
UnitVec_z_CP_CoorSys_E[FL,FR,RL,RR][x y z]	UnitVec_z_CP_CoorSys_E[:][x;y;z]
v_Wind_Coor_Sys_E[x,y,z][m|s]	v_Wind_Coor_Sys_E[x;y;z][m|s]
Sw_Tire_Parameter_Set[FL,FR,RL,RR][1|2|3|4]	Sw_Tire_Parameter_Set_CP[:][1|2|3|4]

The block has 3 new inports: LateralOffset_Vehicle[m], s_Fellows[m], d_Fellows[m].

The block has 3 new outports: Fellow_Signals, Slope_Lateral_Road[deg], RoadInfo.

Release 6.3

In some cases, task overruns occurred in driving on the road. The clothoid segment calculation was modified to avoid these overruns.

Release 6.1	The ROAD block can now simulate user roads with more than 25 segments. The matrix dimensions for the user roads were decreased. It contains only data and road segments which are actually used.

Related topics

References

Road..98

History of the ROADSTATE_RUN_TRIGGER Block

Release 2019-B	This block is used for the traffic serialization feature. It provides the option to trigger the road state transitions into the run state.

History of the SIGNAL_SELCTION Block

Release 2016-A	The SIGNAL_SELECTION block is not used in new ASM demos anymore. Therefore, it has been moved to the Former versions sublibrary. During migration, the link to the block is changed to the former implementation version: `FormerVersions/SIGNAL_SELECTION_1_0`.

History of the SPEED_PROFILER Block

Release 2020-B	The block has a new inport: Enable_v_Vehicle_Init. To keep the model behavior unchanged, it is connected to constant block during migration. The new inport is used to enable drivetrain initialization for scenarios with an initial vehicle speed.
Release 2019-B	There is a new SPEED_PROFILER block. It can be used to include the driving style characterization (driver's performance envelope) in the reference speed calculation. The driving style depends on the driver's parameters and appears while approaching a stop as well as a turn, during cornering and after leaving a turn as well as accelerating from lower to higher vehicle speeds.

History of the V_ROAD_REF Block

Release 2018-B	The driver behavior when the vehicle slips off the road has been improved. Until now, when the vehicle was no longer on the road, the driver kept trying to find the road with the same maneuver speed. In most cases, this speed was too high and led to an instable steering behavior, which caused the driver to oscillate about the road reference line. The new model assumes the maneuver speed to be valid only on the road. Once the vehicle leaves the road, a relatively small speed is assumed. Two new inports were added to the block. During migration, these ports are connected to dummy values to maintain the previous behavior.
Release 2018-A	Six new inports were added. The dimensions of two inports as well as one outport were changed. During migration, these changes are compensated for so that the old block behavior remains unchanged.
Release 2013-B	A reset inport has been added. The simulation step size has been added as parameter.
Release 2013-A	Evaluation of speed limit from road scenery sections has been implemented.
Release 7.4	Braking at maneuver segment transition between two consecutive FinalVelocity segments is now avoided.
Release 7.0	It is now possible to set the Implement logic signals as Boolean data (vs. double) option in the Operator Simulink models.
Related topics	References Road Reference Velocity..52

History of the Environment Demo Model

Release 2013-A	To implement the set of new features in this release, the Environment subsystem of the demo models has been substantially restructured.

Due to the structural changes, not all new features can be added to the migrated models automatically.

To obtain the full functionality of all new features for migrated models, it is recommended to replace the Environment subsystem in the migrated model with the Environment subsystem in the demo model, and adapt the model and ModelDesk experiment accordingly to reflect the structural changes.

Appendix

Bibliography

List of literature

The following literature provides more details:

[Guo93] Guo K., Guan H.: Modelling of driver/vehicle directional control system. Vehicle System Dynamics, 22(3-4) pp. 141-184, 1993.

[Kie05] Kiencke U., Nielsen L.: Automotive Control Systems - For Engine, Driveline and Vehicle. Berlin: Springer, 2005.

[Kri10] Kritayakirana K. and Gerdes J. C.: Autonomous Cornering at the Limits: Maximizing a “g-g” Diagram by Using Feedforward Trail-Braking and Throttle-on-Exit, in: 6th IFAC Symposium Advances in Automotive Control, Munich, Germany, 2010.

[MacA80] MacAdam C.C.: An Optimal Preview Control for Linear Systems. Journal of Dynamic Systems, Measurement and Control, Trans. of the ASME, Vol. 102, 1980.

[MacA81] MacAdam C. C.: Application of an Optimal Preview Control for Simulation of Closed-Loop Automobile Driving. IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-11, No.6, 1981.

[Maj97] Majjad R.: Hybride Modellierung und Identifikation eines Fahrer-Fahrzeug Systems. Dissertation, TH Karlsruhe, 1997.

[Mil95] Milliken F. W. and Milliken D. L.: ""g-g" Diagram," in: Race Car Vehicle Dynamics, SAE International, 1995, pp. 345-366.

[Sha00] Sharp R.: A Mathematical Model for Driver Steering Control with Design, Tuning and Performance Results. Vehicle System Dynamics 33, pp. 289-326, 2000.