Altair

HyperWorks

Altair MotionView 2019 Tutorials

MV-8000: Open Loop Events

altairhyperworks.com

MV-8000: Open Loop Events

In this tutorial, you will learn how to:

- Assemble the vehicle for full vehicle simulations using Assembly Wizard
- Attach the driver using Task Wizard

What is Altair Advanced Driver?

The Altair Driver is a set of MotionView models and libraries that allows MotionView users to control and script vehicle events. Example events include:

- Constant Radius Cornering
- Single Lane Change
- Double Lane Change
- User Defined Path Following
- Fish Hook Event, etc.

This is achieved by creating an interface to the five vehicle inputs:

- Steering
- Throttle
- Gear
- Brake
- Clutch

Using Altair Driver you can simulate any number of full vehicle events using these features:

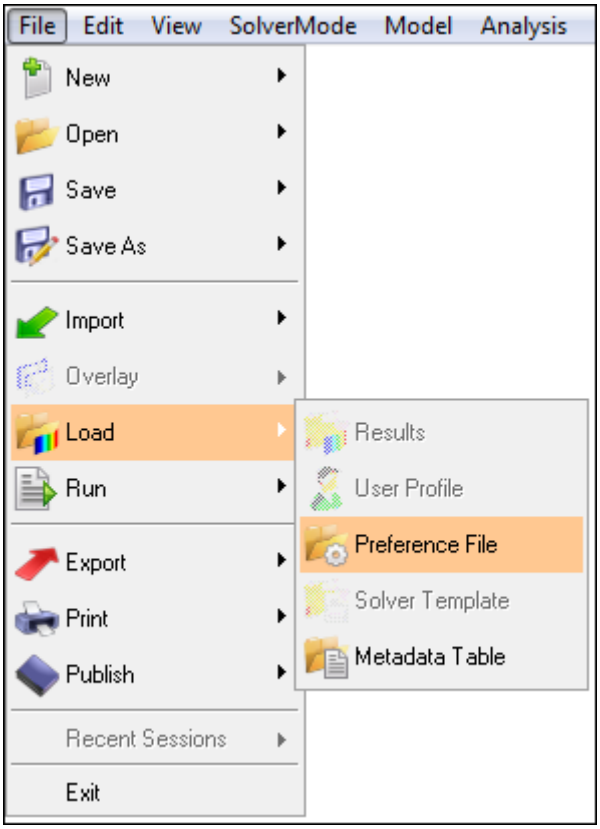
- Scripting – break up the simulation into different maneuvers; select the controllers for vehicle inputs and define conditions that end each maneuver.
- Open-loop, closed-loop, and user-defined controllers to control.
  - Longitudinal speed or acceleration.
  - Vehicle path or lateral acceleration.
- Switching controllers during a simulation.
- Defining path and speed profiles parametrically, in a table, or by referencing a data file.

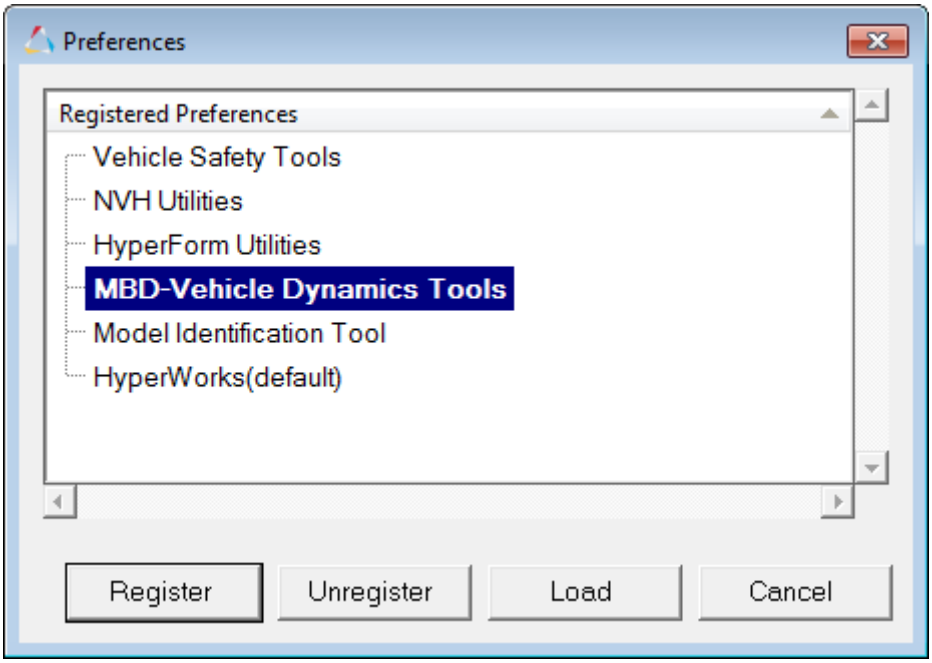
Exercise

Step 1: Assembling the vehicle.

1. Start a new MotionView session.

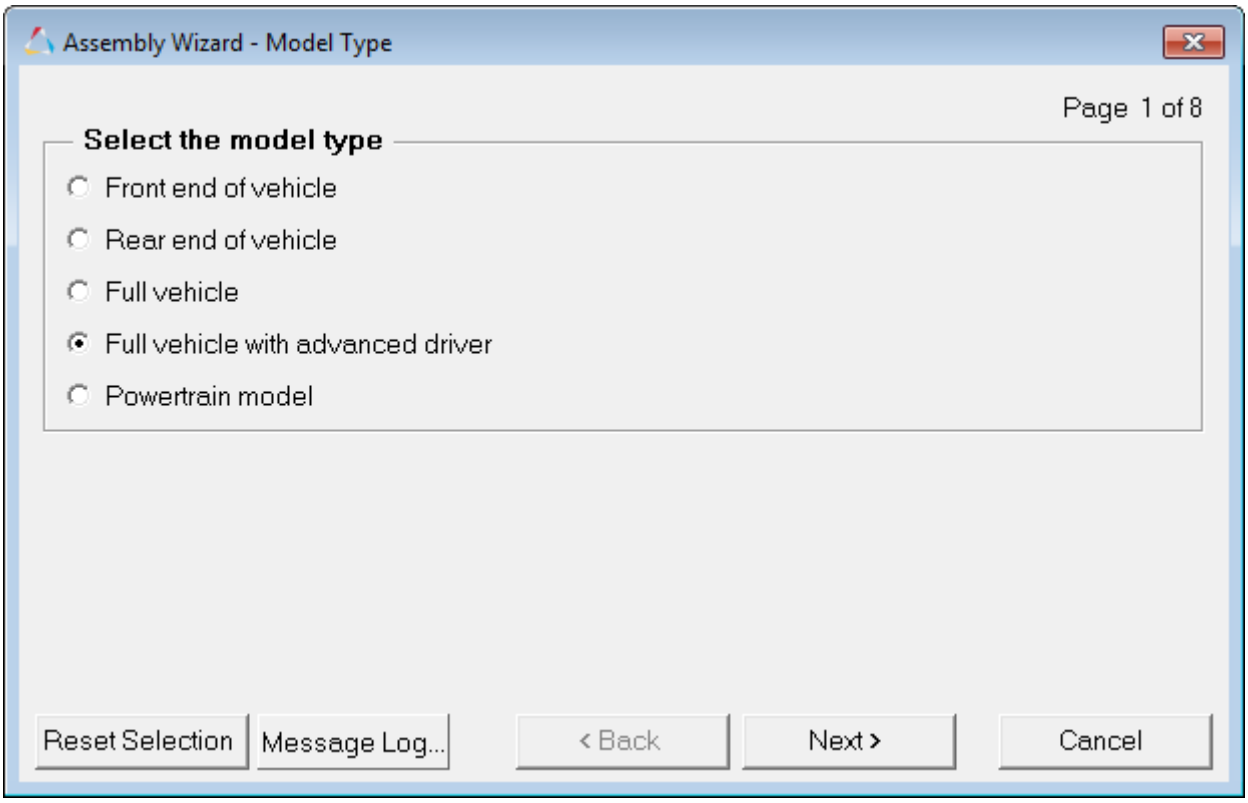
2. Make sure that the **MBD-Vehicle Dynamics Tools** preference file loaded is for all of the MotionView functionality of the Advanced Driver to work properly.





3. Use the Assembly Wizard to assemble the vehicle model. In the first example, a linear torque map powertrain will be used to avoid the complexity of adding controllers for gear and clutch.

4. Choose Full vehicle with advanced driver option. This takes care of all the dependencies of the advanced driver. Click Next.



5. Choose the default selections in the following Assembly Wizard pages.

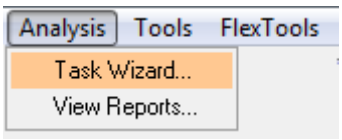
Page	Label	Selection	Default (Yes/No)
1	Model type	Full vehicle with advanced driver	No
2	Driveline configuration	Front wheel drive	Yes

Page	Label	Selection	Default (Yes/No)
3	Vehicle body	Body	Yes
3	Front suspension	Frnt macpherson susp (1 pc. LCA)	Yes
3	Steering linkages	Rackpin steering	Yes
3	Rear subframe	None	Yes
3	Rear suspension	Rear quadlink susp	Yes
3	Powertrain	Linear torque map powertrain	Yes
3	Signal generator	Driver signal generator	Yes
3	Tires	FIALA/HTIRE	Yes
4	Steering column	Steering column 1 (not for abaqus)	Yes
4	Steering boost	None	Yes
5	Front struts	Frnt strut (with inline jts)	Yes
5	Front stabilizer bars	None	Yes
5	Rear struts	Rear strut (with inline jts)	Yes
5	Rear stabilizer bars	None	Yes
6	Front jounce bumpers	None	Yes
6	Front rebound bumpers	None	Yes
6	Rear jounce bumpers	None	Yes
6	Rear rebound bumpers	None	Yes
7	Disk brakes	Disk brakes	Yes
7	Front driveline	Independent fwd	Yes
8		Next	No
9		Finish	No

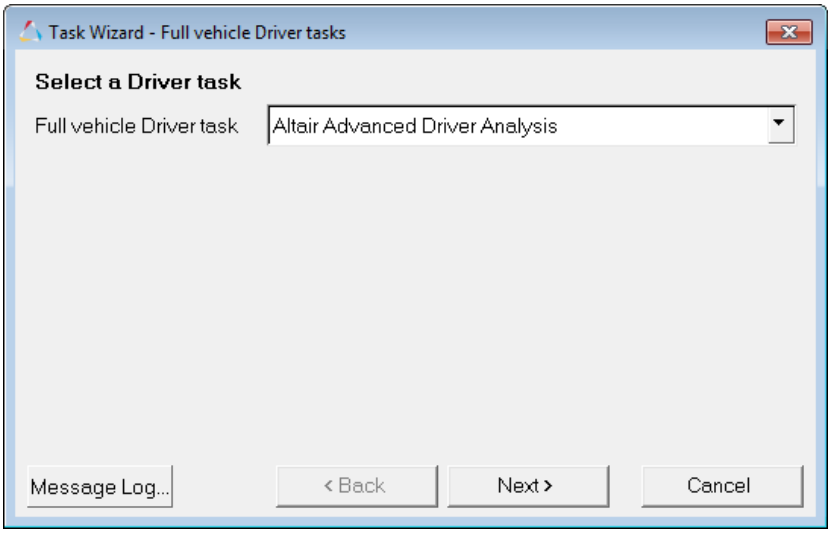
Step 2: Adding driver analysis.

Use the Task Wizard to load the driver analysis.

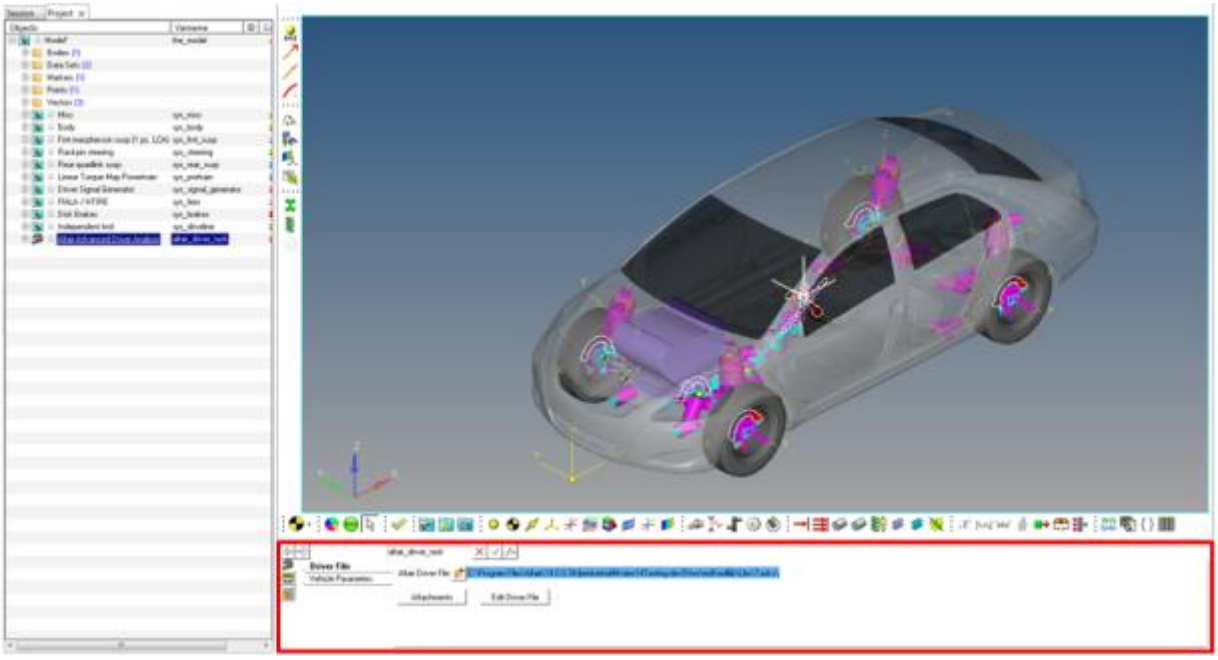
1. From the **Analysis** menu, select ***Task Wizard***.



2. Select ***Altair Advanced Driver Analysis*** from the **Full vehicle Driver task** drop-down menu.



The Altair Advanced Driver with  icon is added to the browser tree. Selecting the driver icon will open up the Altair Driver panel. This automatically resolves all of the vehicle attachments for the Advanced Driver.

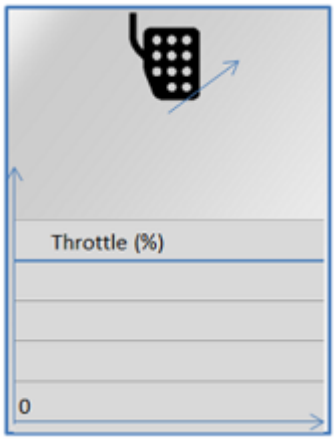


Step 3: Writing an Altair Driver File (ADF) driving event .

Driver requires an event script or the Altair driver file (ADF) to run any driving event. ADF can be edited using any text editor or by clicking the **Edit Driver File** button on the driver panel.

Example #1 Open Loop Acceleration Event

The objective of the example is to script an event with 50% throttle, 0% brake and 0 steering angle. An event can be broken down into smaller sub-events or maneuvers. For the sake of simplicity we will model this as a single maneuver event.



1. Open any text editor and copy and paste the following text into it. **Important: All blank lines must be removed prior to saving the file!** Be sure to read through the comments for a better understanding on what is written in the ADF.

```
$----------------------------------------------------------------ALTAIR_HEADER

$ This block is required for version control

[ALTAIR_HEADER]

FILE_TYPE                = 'ADF'

FILE_VERSION    = 1.0

FILE_FORMAT     = 'ASCII'

$-------------------------------------------------------------------------UNITS

$In this block we specify the units in which this file should be read

[UNITS]

(BASE)

{length  force            angle      mass   time}

'meter'   'newton'        'radians'  'kg'      'sec'

$-----------------------------------------------------------------VEHICLE_IC

$In this block we specify the initial conditions specifically initial speed of the

$vehicle with respect to the vehicle IC marker in the driver attachments
```

[VEHICLE_INITIAL_CONDITIONS]

VX0 = -20.0

VY0 = 0.0

VZ0 = 0.0

$----------STEERING_STANDARD

$This block specifies the saturation and cutoff frequency for the low pass filter for

$steering output signal. These signals are global and are active for the entire event

[STEER_STANDARD]

MAX_VALUE = 3.141593

MIN_VALUE = -3.141593

SMOOTHING_FREQUENCY = 10.0

INITIAL_VALUE = 0.0

$----------THROTTLE_STANDARD

$This block specifies the saturation and cutoff frequency for the low pass filter for

$throttle output signal

[THROTTLE_STANDARD]

MAX_VALUE = 1.0

MIN_VALUE = 0.00

SMOOTHING_FREQUENCY = 10.0

INITIAL_VALUE = 0.5

$----------BRAKING_STANDARD

$This block specifies the saturation and cutoff frequency for the low pass filter for

$brake output signal

[BRAKE_STANDARD]

MAX_VALUE = 1.0

MIN_VALUE = 0.0

SMOOTHING_FREQUENCY = 10.0

INITIAL_VALUE = 0.0

$----------MANEUVERS_LIST

```
$This block provides the list of all the maneuvers, simulation time for each
maneuver
$maximum solver step size (hmax) and print interval
[MANEUVERS_LIST]
{ name                  simulation_time  h_max  print_interval}
'MANEUVER_1'   5.0              0.01     0.01
$-----------------------------------------------------------------MANEUVER_1
[MANEUVER_1]
$This block provides the ties controllers to each driver output
TASK = 'STANDARD'
(CONTROLLERS)
{DRIVER_SIGNAL      PRIMARY_CONTROLLER      ADDITIONAL_CONTROLLER}
 STEER              OL_STEER_0              NONE
 THROTTLE           OL_0.5                  NONE
 BRAKE              OL_0                    NONE
$-----------------------------------------------------------OL_STEER
$This is controller block containing all the information required by
$the driver to construct the controller. Different controllers have
$different requirements. Here we are using open loop constant type
$of controller.
[OL_STEER_0]
TAG        = 'OPENLOOP'
TYPE       = 'CONSTANT'
VALUE      = 0.0
$-----------------------------------------------------------OL_BRAKE
[OL_0]
TAG        = 'OPENLOOP'
TYPE       = 'CONSTANT'
VALUE      = 0.0
$-----------------------------------------------------------OL_THROTTLE
```

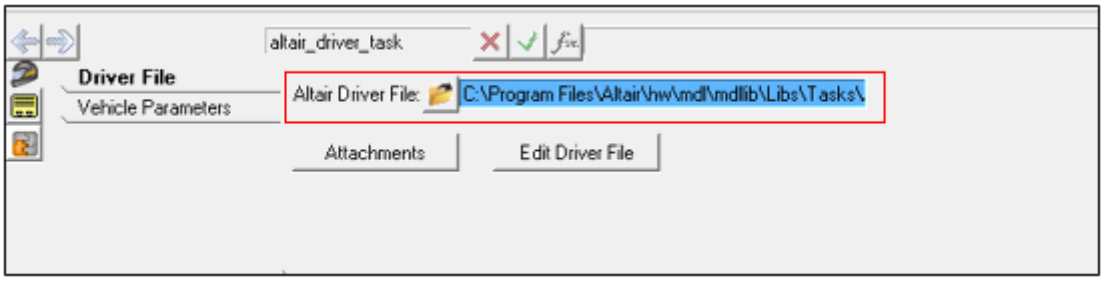
[OL_0.5]

TAG = 'OPENLOOP'

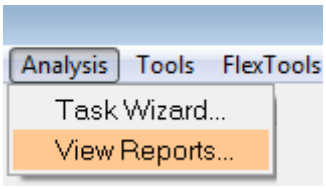
TYPE = 'CONSTANT'

VALUE = 0.5

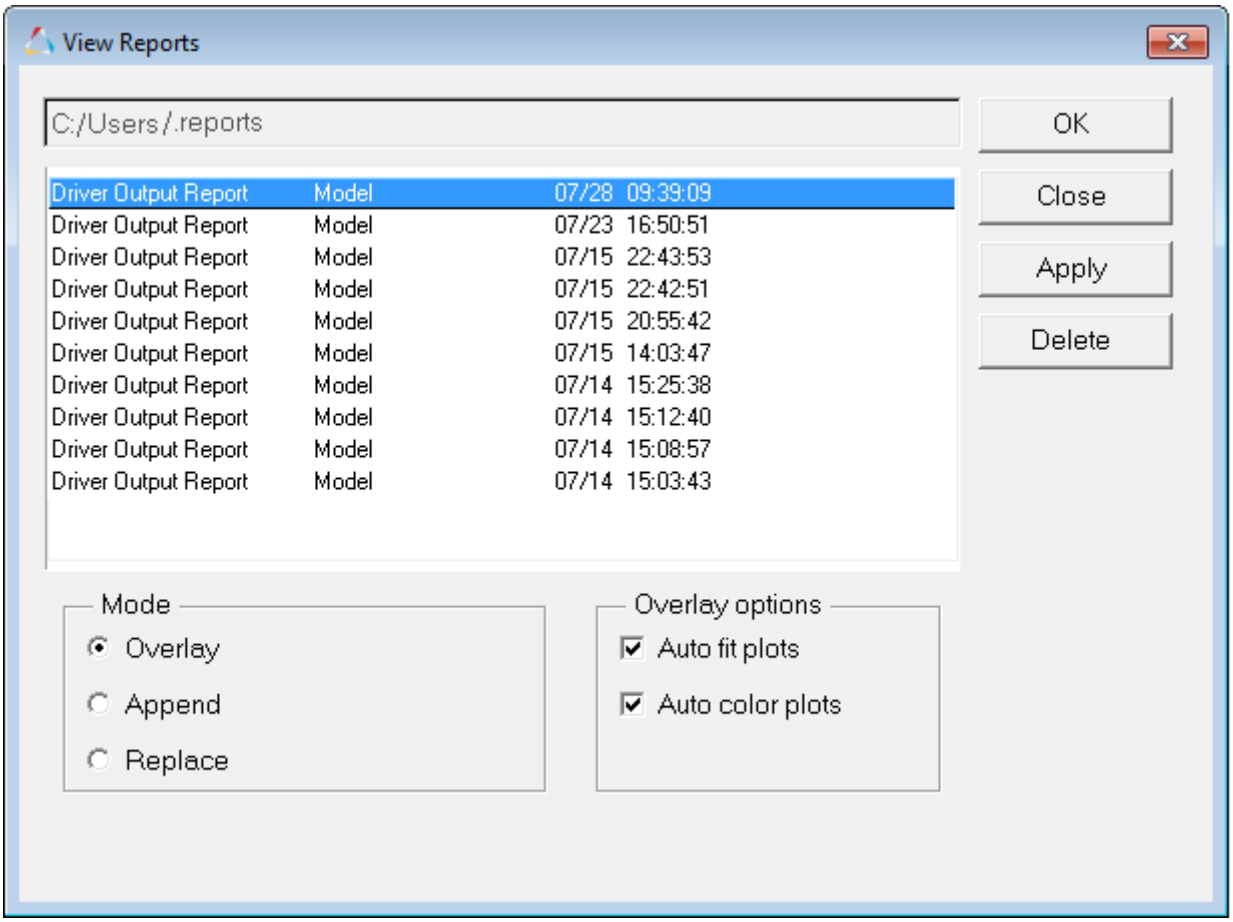
2. ***Save*** the file.
3. Source the file in the **Altair Driver File** browser.



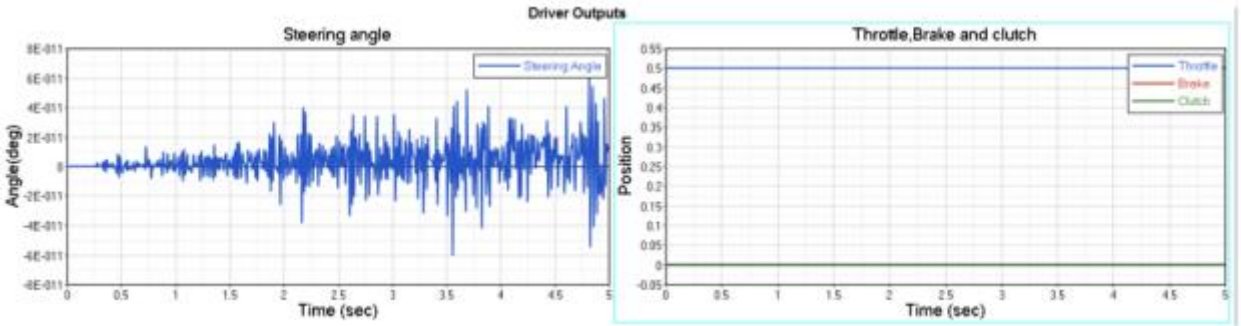
4. The **Edit driver file** button can be used for editing the file.
5. ***Run*** the simulation  .
6. After the simulation run is over and the solver creates h3d and plt files, click ***View Reports...*** for standard plots.



7. Select the recent run in the **View Reports** dialog and click ***OK***.



8. The noise in the steering is numerical error of negligible magnitude – 0 for all practical purposes. Throttle is constant at 0.5 (driver throttle, brake and clutch outputs are normalized so, 50%) and brake is constant at 0.



9. Next let's try slightly different initial conditions. We will start the throttle at 0 and brake at 100%. Click the ***Edit Driver File*** button to open up the file editor.

10. Change the ***INITIAL_VALUE*** attribute in the **THROTTLE_STANDARD** block in the ADF, from **0.5** to ***0***.

[THROTTLE_STANDARD]

MAX_VALUE = 1.0

MIN_VALUE = 0.00

SMOOTHING_FREQUENCY = 10.0

INITIAL_VALUE = ~~0.5~~ 0

11. Similarly, change the ***INITIAL_VALUE*** attribute in the **BRAKE_STANDARD** block in the ADF, from **0.0** to ***1.0***.

[BRAKE_STANDARD]

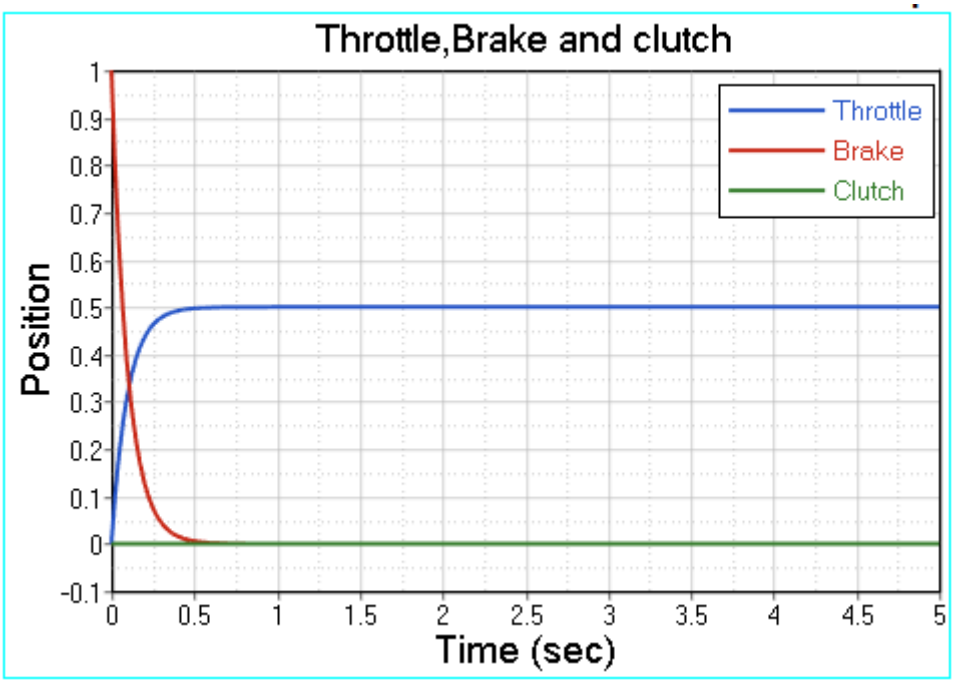
MAX_VALUE = 1.0

MIN_VALUE = 0.0

SMOOTHING_FREQUENCY = 10.0

INITIAL_VALUE = ~~0.0~~ 1.0

12. Save the ADF.
13. Run the simulation and study the results.
14. Now, we see that throttle and brake start from respective initial values and step up to the controller outputs. The time taken to step up is roughly (5x1/SMOOTHING_FREQUENCY).



Example #2 Open Loop Sinusoidal Steering Event

In this example we will model a simple event with constant 20% throttle, constant 0% throttle and sinusoidal steering input with amplitude of 60 degrees (∏/3 radians) and frequency of 0.5 Hz.

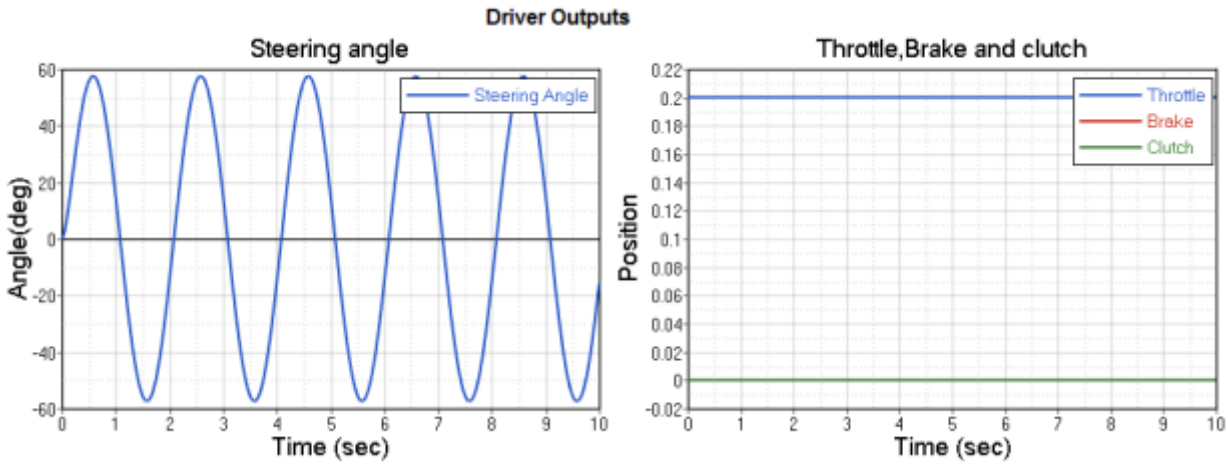
1. Open any text editor and copy/paste the following text into it. **Important: All blank lines must be removed prior to saving the file!**

```
$-------------------------------------------------------------ALTAIR_HEADER

[ALTAIR_HEADER]

FILE_TYPE                = 'ADF'

FILE_VERSION    = 1.0

FILE_FORMAT     = 'ASCII'

$-----------------------------------------------------------------------UNITS

[UNITS]

(BASE)

{length  force           angle      mass   time}

'meter'   'newton'       'radians'  'kg'     'sec'

$-------------------------------------------------------------------VEHICLE_IC

[VEHICLE_INITIAL_CONDITIONS]

VX0 = -20.0

VY0 = 0.0

VZ0 = 0.0

$-----------------------------------------------------------STEERING_STANDARD

[STEER_STANDARD]

MAX_VALUE                = 3.141593

MIN_VALUE                = -3.141593

SMOOTHING_FREQUENCY      = 10.0

INITIAL_VALUE            = 0.0

$-----------------------------------------------------------THROTTLE_STANDARD

[THROTTLE_STANDARD]

MAX_VALUE                = 1.0

MIN_VALUE                = 0.00
```

```
SMOOTHING_FREQUENCY     = 10.0
INITIAL_VALUE           = 0.2
$-------------------------------------------------------------BRAKING_STANDARD
 [BRAKE_STANDARD]
MAX_VALUE               = 1.0
MIN_VALUE               = 0.0
SMOOTHING_FREQUENCY     = 10.0
INITIAL_VALUE           = 0.0
$-----------------------------------------------------------------MANEUVERS_LIST
 [MANEUVERS_LIST]
{ name                  simulation_time  h_max  print_interval}
'MANEUVER_1'            10.0             0.01   0.01
$-----------------------------------------------------------------------MANEUVER_1
[MANEUVER_1]
$This block provides the ties controllers to each driver output
TASK = 'STANDARD'
(CONTROLLERS)
{DRIVER_SIGNAL          PRIMARY_CONTROLLER      ADDITIONAL_CONTROLLER}
 STEER                  OL_STEER                NONE
 THROTTLE               OL_THROTTLE             NONE
 BRAKE                  OL_BRAKE                NONE
$-----------------------------------------------------------OL_STEER.
$SIGNAL_CHANNEL tells the driver which solver variable in Signal Generator to over-
ride
$with the EXPRESSION value. The EXPRESSION should be consistent with
MOTIONSOLVE.
[OL_STEER]
TAG                     = 'OPENLOOP'
TYPE                    = 'EXPRESSION'
SIGNAL_CHANNEL          = 0
```

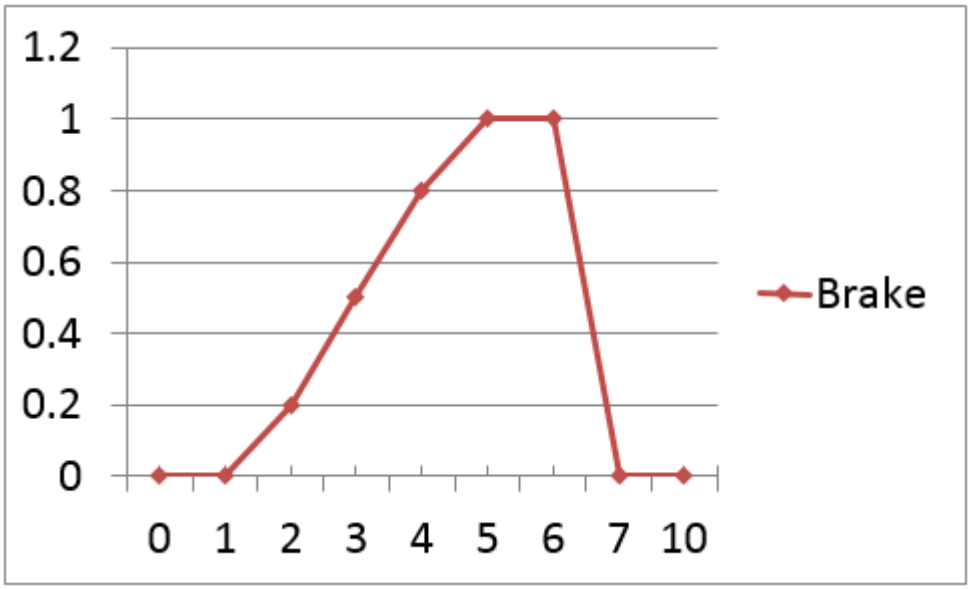
```
EXPRESSION              = 'DTOR(60)*SIN(2*PI*0.5*TIME)'

$-------------------------------------------------------OL_THROTTLE

[OL_THROTTLE]

TAG         = 'OPENLOOP'

TYPE        = 'CONSTANT'

VALUE       = 0.2

$-------------------------------------------------------OL_BRAKE

[OL_BRAKE]

TAG         = 'OPENLOOP'

TYPE        = 'CONSTANT'

VALUE       = 0.0
```

2. Run the simulation and study the results.



Example #3 Open Loop Curve Driven Braking Event

In this example we will model a simple event with braking signal as a curve.



1. Open text editor and copy/paste the following text into it. **Important: All blank lines must be removed prior to saving the file!**

```
$-------------------------------------------------------------ALTAIR_HEADER

[ALTAIR_HEADER]

FILE_TYPE                = 'ADF'

FILE_VERSION    = 1.0

FILE_FORMAT     = 'ASCII'

$-----------------------------------------------------------------------UNITS

[UNITS]

(BASE)

{length  force            angle      mass   time}

'meter'   'newton'        'radians'  'kg'     'sec'

$-----------------------------------------------------------VEHICLE_IC

[VEHICLE_INITIAL_CONDITIONS]

VX0 = -20.0

VY0 = 0.0

VZ0 = 0.0

$-------------------------------------------------------STEERING_STANDARD
```

```
 [STEER_STANDARD]
MAX_VALUE                = 3.141593
MIN_VALUE                = -3.141593
SMOOTHING_FREQUENCY      = 10.0
INITIAL_VALUE        = 0.0
$-------------------------------------------------------------THROTTLE_STANDARD
 [THROTTLE_STANDARD]
MAX_VALUE                = 1.0
MIN_VALUE                = 0.00
SMOOTHING_FREQUENCY      = 10.0
INITIAL_VALUE            = 0.0
$-------------------------------------------------------------BRAKING_STANDARD
 [BRAKE_STANDARD]
MAX_VALUE                = 1.0
MIN_VALUE                = 0.0
SMOOTHING_FREQUENCY      = 10.0
INITIAL_VALUE            = 0.0
$-------------------------------------------------------------MANEUVERS_LIST
 [MANEUVERS_LIST]
{ name                   simulation_time  h_max  print_interval}
'MANEUVER_1'             10.0             0.001    0.01
$-------------------------------------------------------------MANEUVER_1
[MANEUVER_1]
$This block provides the ties controllers to each driver output
TASK = 'STANDARD'
(CONTROLLERS)
{DRIVER_SIGNAL       PRIMARY_CONTROLLER      ADDITIONAL_CONTROLLER}
 STEER               OL_STEER                NONE
 THROTTLE            OL_THROTTLE             NONE
```

BRAKE OL_BRAKE NONE

$----------OL_STEER.

$SIGNAL_CHANNEL tells the driver which solver variable in Signal Generator to over-ride

$with the EXPRESSION value. The EXPRESSION should be consistent with MOTIONSOLVE.

[OL_STEER]

TAG = 'OPENLOOP'

TYPE = 'CONSTANT'

VALUE = 0.0

$----------OL_THROTTLE

[OL_THROTTLE]

TAG = 'OPENLOOP'

TYPE = 'CONSTANT'

VALUE = 0.0

$----------OL_BRAKE

[OL_BRAKE]

TAG = 'OPENLOOP'

TYPE = 'CURVE'

BLOCK = 'BRAKE_CRV'

$----------CURVE_DATA

[BRAKE_CRV]

INDEPENDENT_VARIABLE = 'TIME'

DEPENDENT_VARIABLE = 'BRAKE_SIGNAL'

INTERPOLATION = 'LINEAR'

{TIME BRAKE_SIGNAL}

0 0

1 0

2 0.2

3 0.5

4 0.8

5 1.0

6 1.0

7 0

10 0

2. Run the simulation and study the results.