Test Report for Pedal Interpretation

Test Report for Pedal Interpretation:

Published 09-Jan-2008 22:20:22

Table of Contents

List of Figures

Chapter 1. Summary

Table of Contents

Model

The Model section provides information about the current model

```
Model Name : LCTC_vTD8_Test_TC0_ref6_2007a

Model Location:
'C:\profiles\XUX\Eigene Dateien\MATLAB\
021007_PedInt_TestDataVariants\LCTC_vTD8_Test_TC0_ref6_2007a.mdl'

Environment: Windows XP, MATLAB 2007a, Simulink 2007a

SUT Name : PedalInterpretation

Amount of SUT Input Signals : 3
SUT Input Signals Names:
v_Fzg,phi_Acc,phi_Brake

Amount of SUT Output Signals :  4
SUT Output Signals Names:
Acc Pedal,Brake Pedal,T_des Drive,T_des_Brake
```

Report

The Report section provides information about the current report.

```
Report File Name :  TestReport_8

Report Location :
'C:\profiles\XUX\Eigene Dateien\MATLAB\
021007_PedInt_TestDataVariants\TestReport_8.rpt'
```

Chapter 2. Test Control

Table of Contents

Partition testing method is used for test data selection. Combination strategies define ways to select values for individual input parameters and combine them to form complete test cases. Hence, this section includes variants management and test cases management.
(More details about Partition testing see Test Coverage section.)

Three kinds of Variants Management methods (Minimal Combination, One factor at a time, and n-wise Combination.) are provided.
One of them is chosen by the tester before test data generation:

Minimal combination A ++ B: Each class in A and B are considered at least once.

One factor at a time: it uses a default normal condition as the starting point. The test cases are constructed by changing only one parameter at a time, assuming there is no interaction among parameters.

n-wise-combinations n-wise(A1, A2, ..., Am): Each possible combination of n classes in A1, A2, ...,Am (1<=n <=m) is considered at least once.

General Infomation about the testing model :

Amount of Test Cases : 4

Maximum Amount of Variants : 4

Variants Management

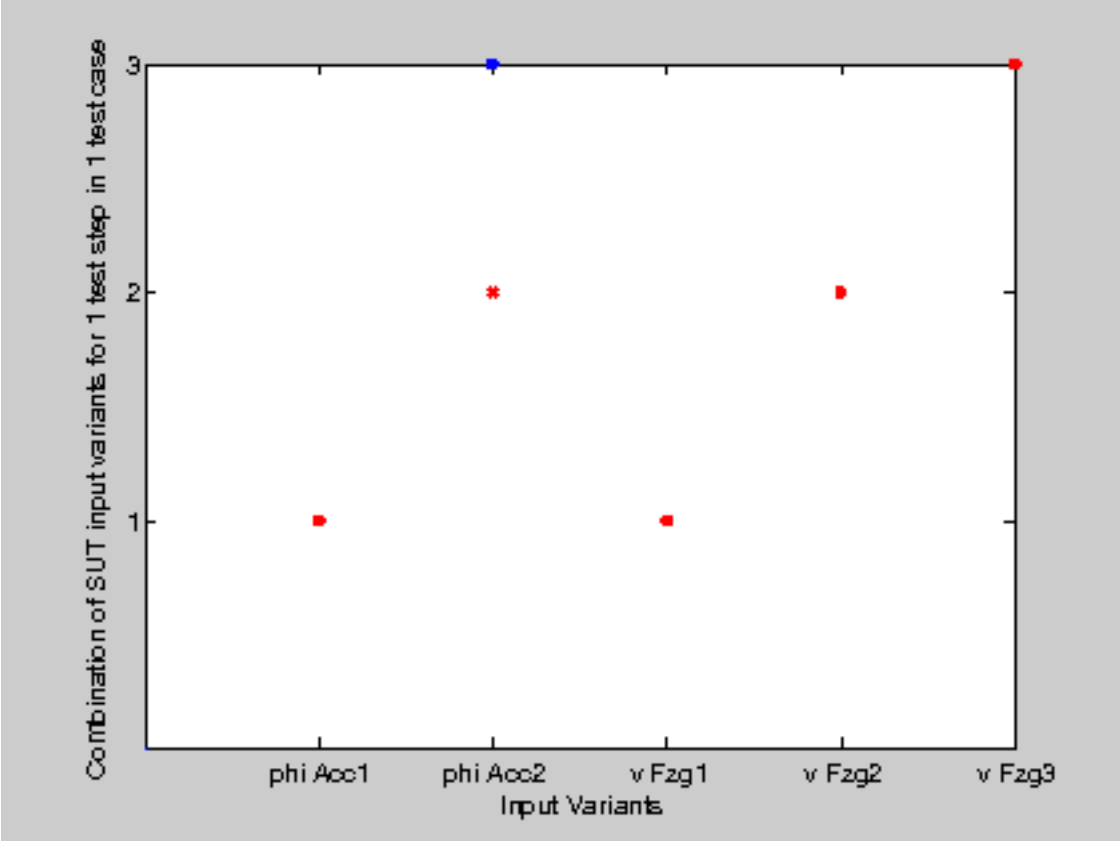
Variants Management in this model adopts 'Minimal Combination' method

Test Sequence is exemplified by the Figure2.1 below :
(based on the precondition :
v = const & phi_Acc decreases & T_des_Drive >= 0)

input_SUT : phi Acc -- generate decrease-- 2 variant(s)
(The signal range of phi Acc is [0,100], partition point : 2)

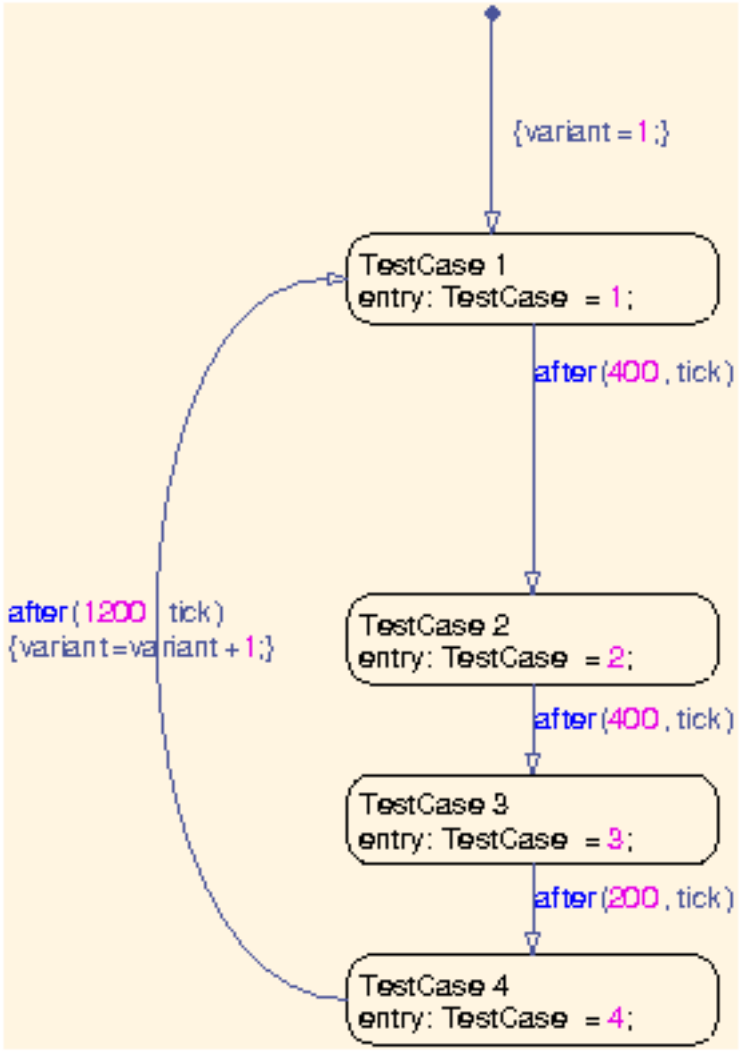
input_SUT : v Fzg -- generate constant-- 3 variant(s)
(The signal range of phi Acc is [-10,70], partition point : 0, 2)

Figure 2.1. Matrix Table



Test Cases Management

Figure 2.2. Stateflow Chart




```
Test Case 1 :
SR-PI-01.1
Brake
Pedal recognition
Running Time : 400 second


Test Case 2 :
SR-PI-01.3
Acceleration
Pedal recognition
Running Time : 400 second


Test Case 3 :
SR-PI-02.1
Brake pedal
interpretation
Running Time : 200 second
```

Test Case 4 :
SR-PI-02.2
Acceleration pedal
interpretation
Running Time : 1200 second

Variants management is set by testing level, which means every test case will be executed 'maximum number of variants' times.

Testing performs 4 times iteration

One Cycle Time = 22 second

Test Execution Time (total) = 88 second

Chapter 3. Test Data & Results

Table of Contents

Test Case 1

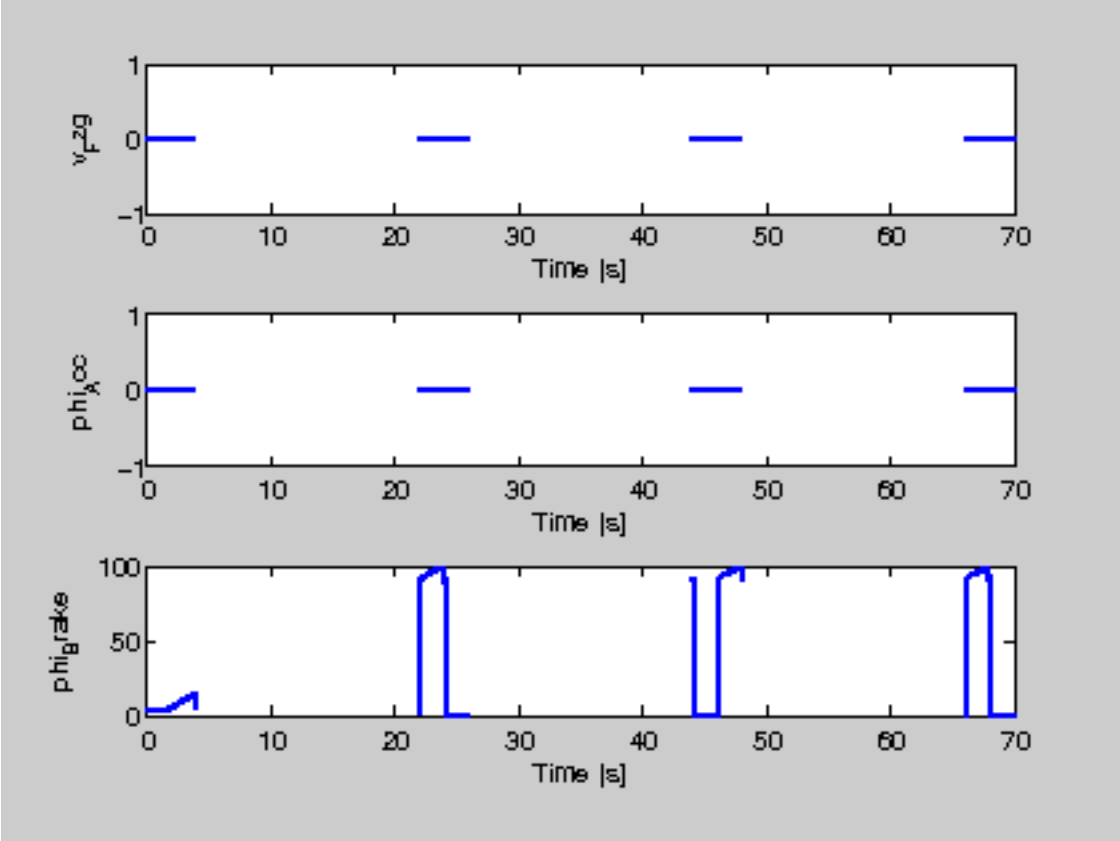
In the requirement : SR-PI-01.1 Brake Pedal recognition

```
Test Data in the test case: SR-PI-01.1 Brake Pedal recognition
v_Fzg  minimum value:0
v_Fzg  maximum value:0

Test Data in the test case: SR-PI-01.1 Brake Pedal recognition
phi_Acc  minimum value:0
phi_Acc  maximum value:0

Test Data in the test case: SR-PI-01.1 Brake Pedal recognition
phi_Brake  minimum value:0.0025
phi_Brake  maximum value:99.9525
```

Figure 3.1. Test Data in iterations of the test case SR-PI-01.1 Brake Pedal recognition



Precondition 1 . 1

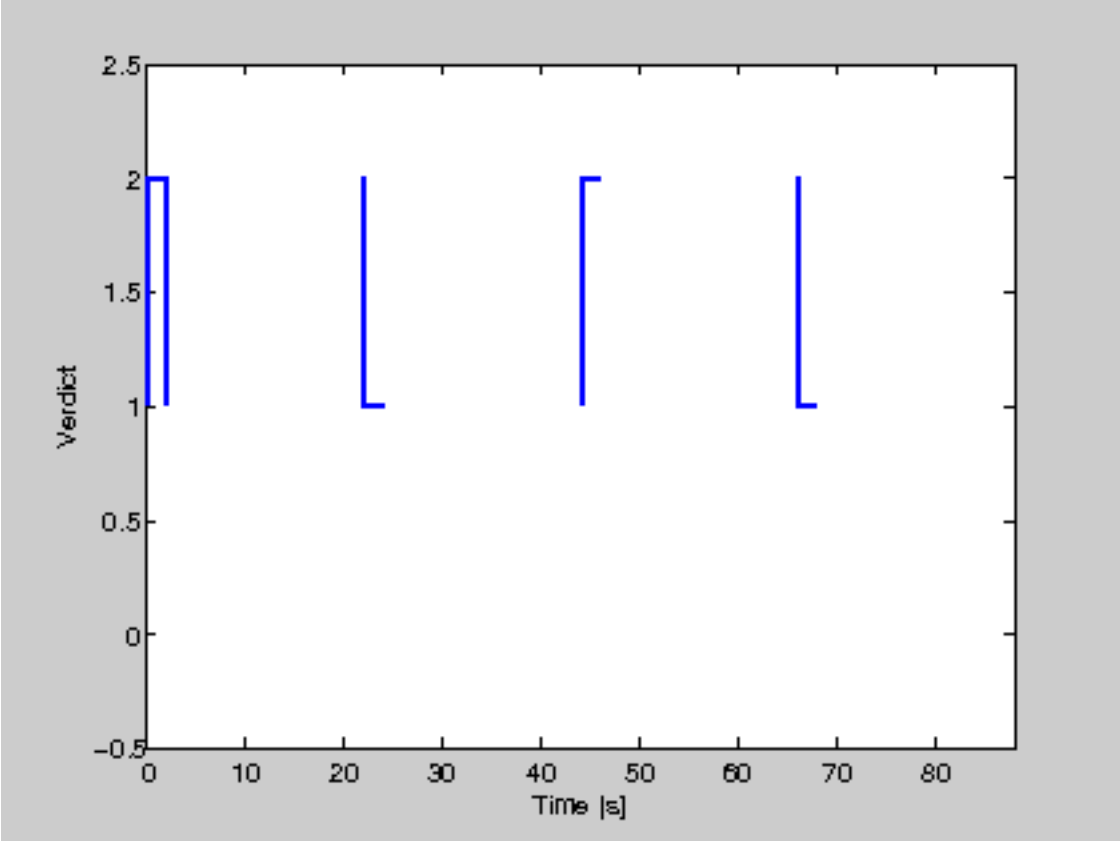
```
the Precondition : phi_Brake < ped_min
```

```
Validation Results
```

```
Overall verdict: PASS
```

- FAIL: 0% (0/8801)
- PASS: 63% (5604/8801)
- NONE: 36% (3197/8801)

Figure 3.2. Validation Result



Precondition 1 . 2

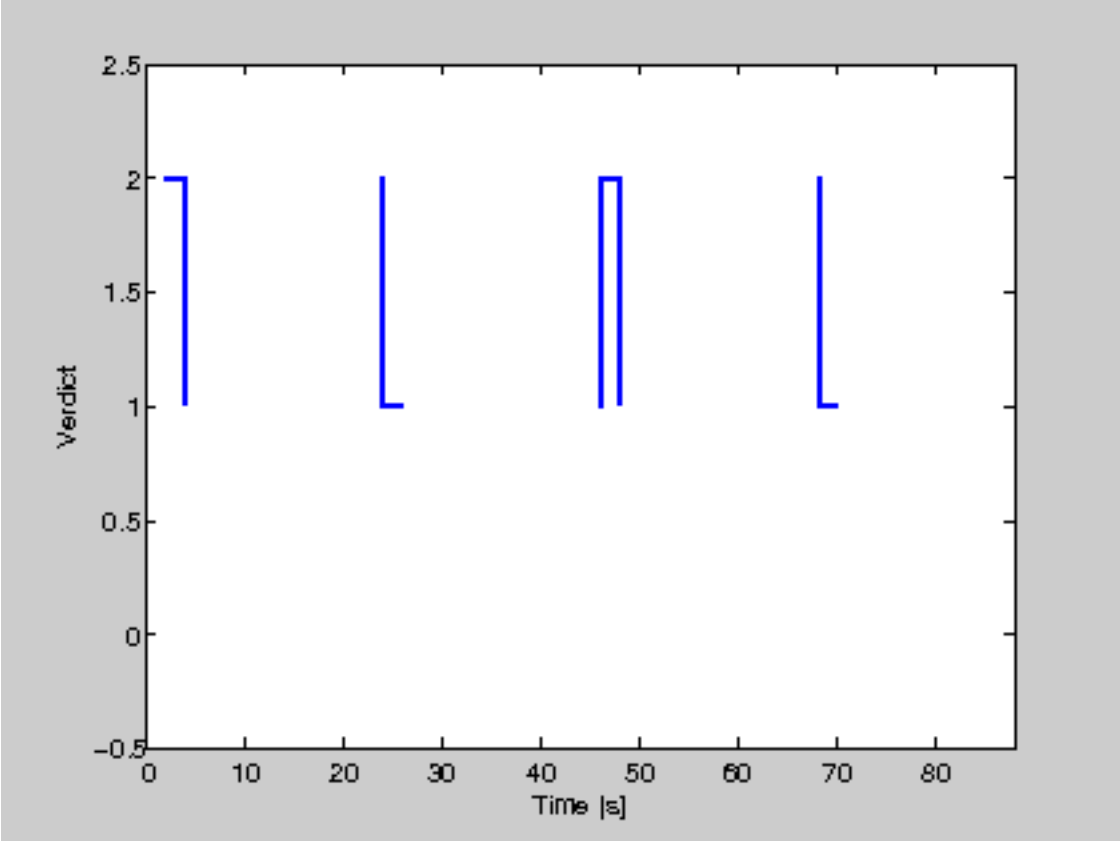
```
the Precondition : phi_Brake >= ped_min
```

```
Validation Results
```

```
Overall verdict: PASS
```

- FAIL: 0% (0/8801)
- PASS: 36% (3197/8801)
- NONE: 63% (5604/8801)

Figure 3.3. Validation Result



Test Case 2

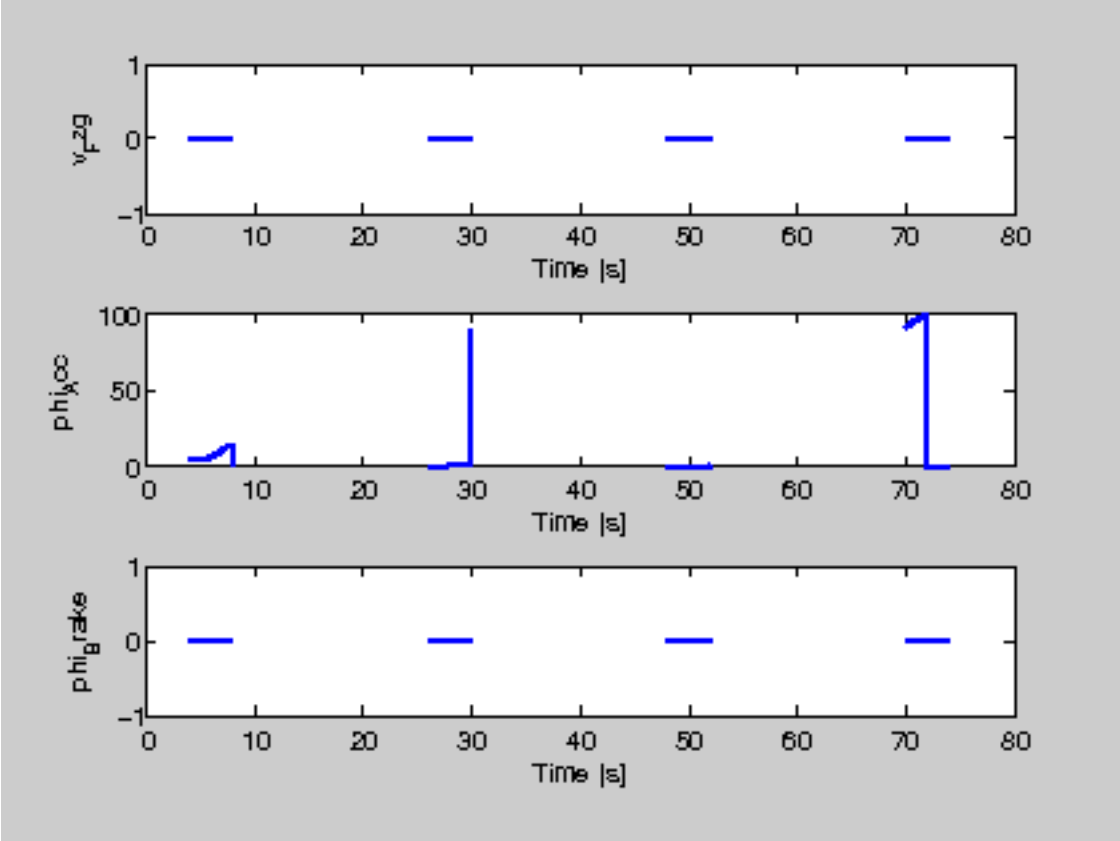
In the requirement : SR-PI-01.3 Acceleration Pedal recognition

```
Test Data in the test case: SR-PI-01.3 Acceleration Pedal recognition
v_Fzg  minimum value:0
v_Fzg  maximum value:0

Test Data in the test case: SR-PI-01.3 Acceleration Pedal recognition
phi_Acc  minimum value:0
phi_Acc  maximum value:99.9525

Test Data in the test case: SR-PI-01.3 Acceleration Pedal recognition
phi_Brake  minimum value:0
phi_Brake  maximum value:0
```

Figure 3.4. Test Data in iterations of the test case SR-PI-01.3 Acceleration Pedal recognition



Precondition 2 . 1

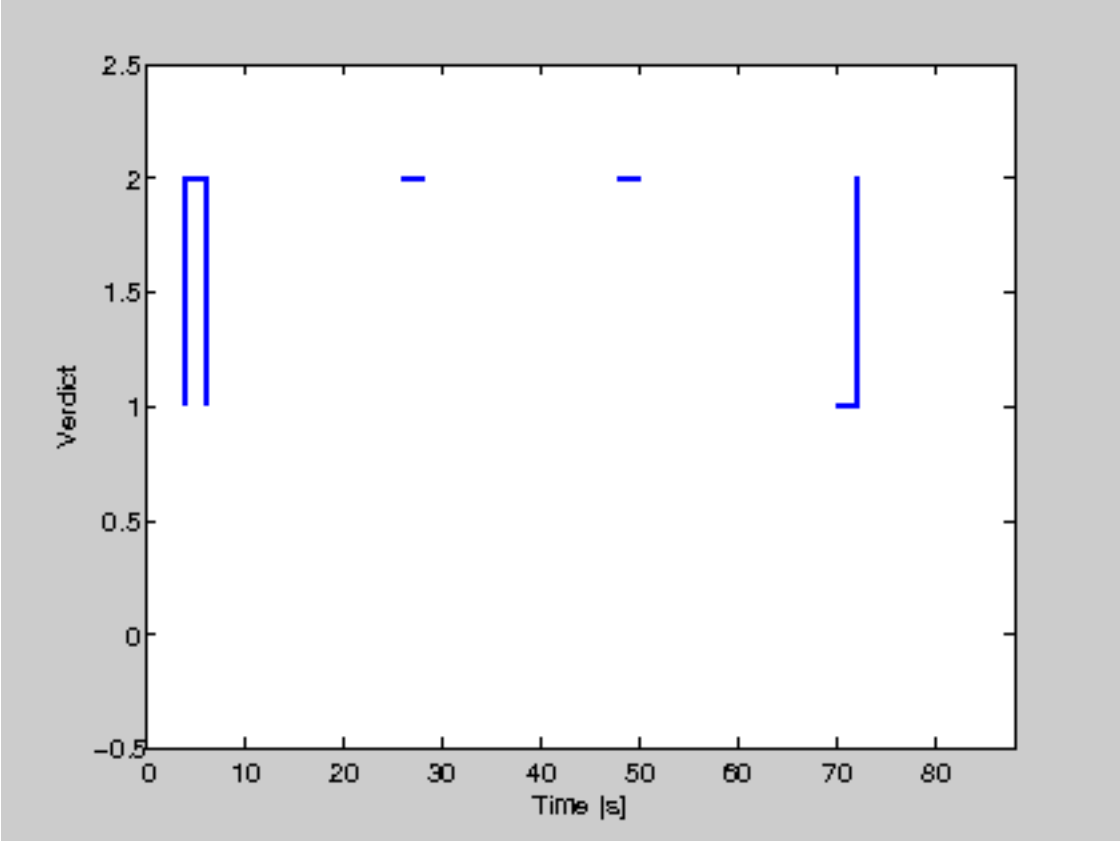
```
the Precondition : phi_Acc < ped_min
```

```
Validation Results
```

```
Overall verdict: PASS
```

- FAIL: 0% (0/8801)
- PASS: 48% (4304/8801)
- NONE: 51% (4497/8801)

Figure 3.5. Validation Result



Precondition 2 . 2

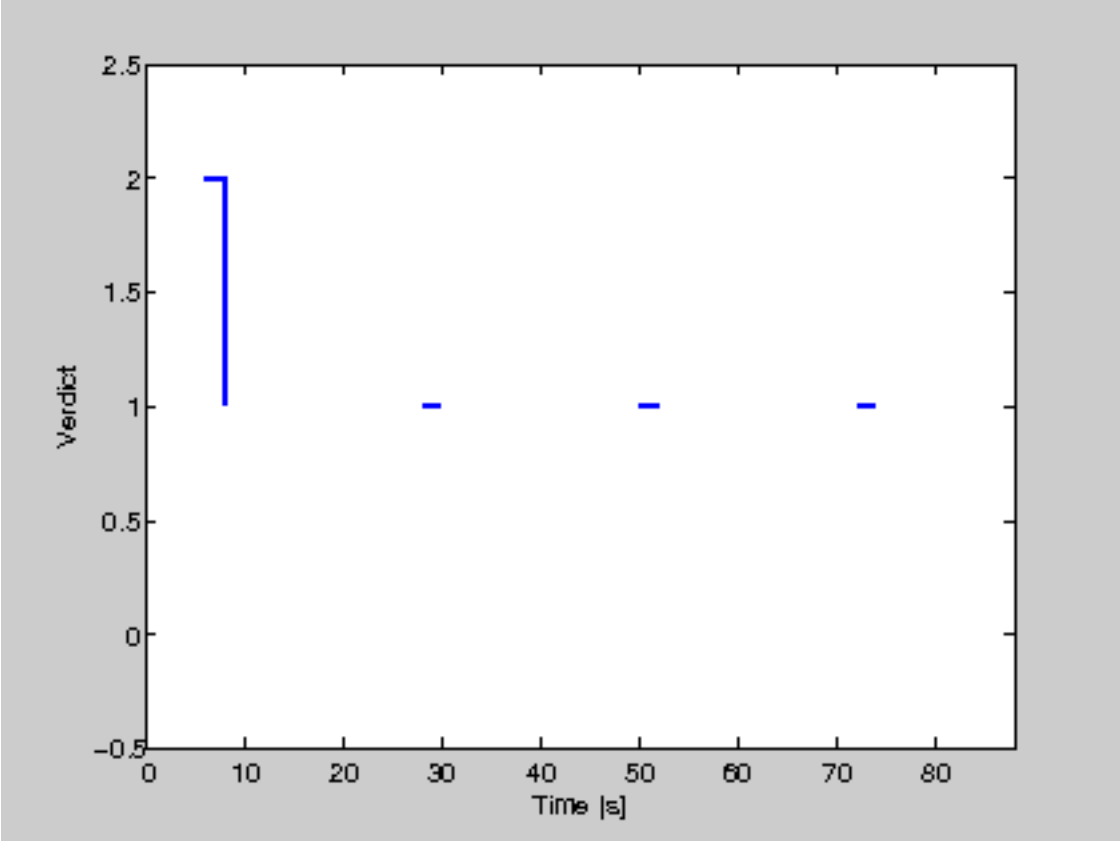
```
the Precondition : phi_Acc >= ped_min
```

```
Validation Results
```

```
Overall verdict: PASS
```

- FAIL: 0% (0/8801)
- PASS: 51% (4497/8801)
- NONE: 48% (4304/8801)

Figure 3.6. Validation Result



Test Case 3

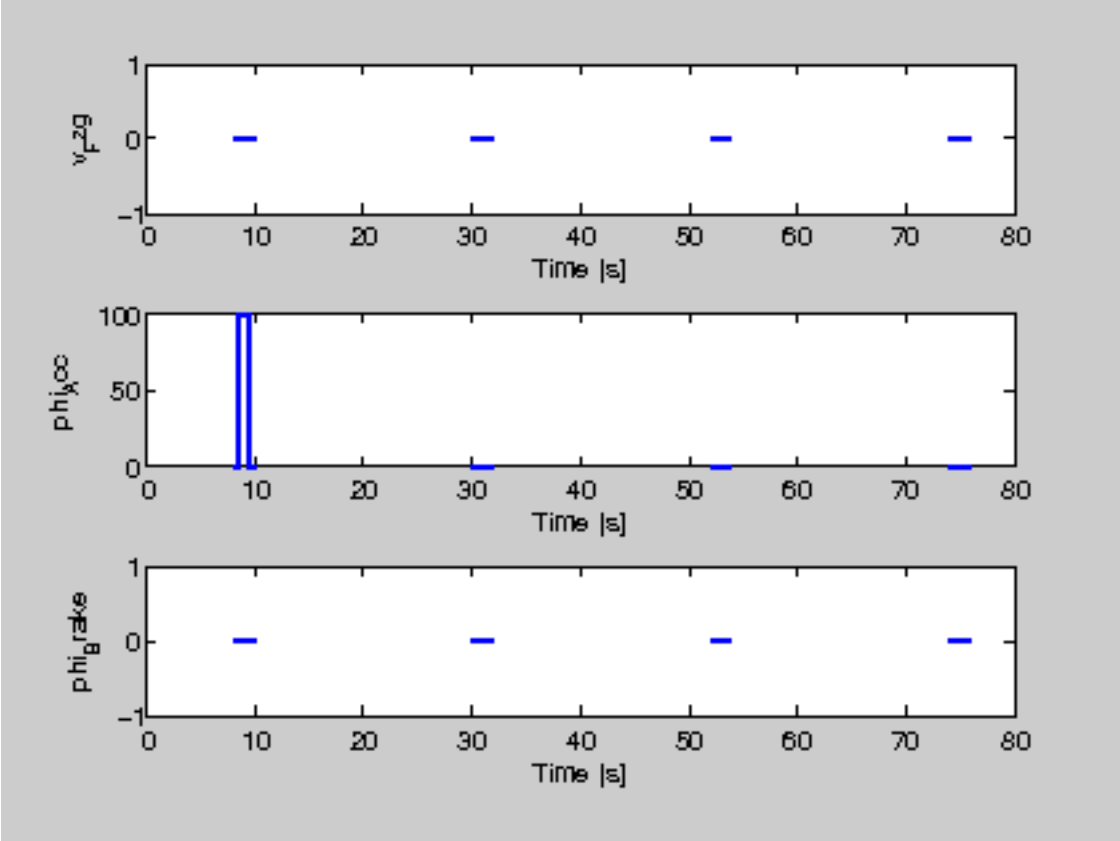
In the requirement : SR-PI-02.1 Brake pedal interpretation

```
Test Data in the test case: SR-PI-02.1 Brake pedal interpretation
v_Fzg  minimum value:0
v_Fzg  maximum value:0

Test Data in the test case: SR-PI-02.1 Brake pedal interpretation
phi_Acc  minimum value:0
phi_Acc  maximum value:99

Test Data in the test case: SR-PI-02.1 Brake pedal interpretation
phi_Brake  minimum value:0
phi_Brake  maximum value:0
```

Figure 3.7. Test Data in iterations of the test case SR-PI-02.1 Brake pedal interpretation



Precondition 3 . 1

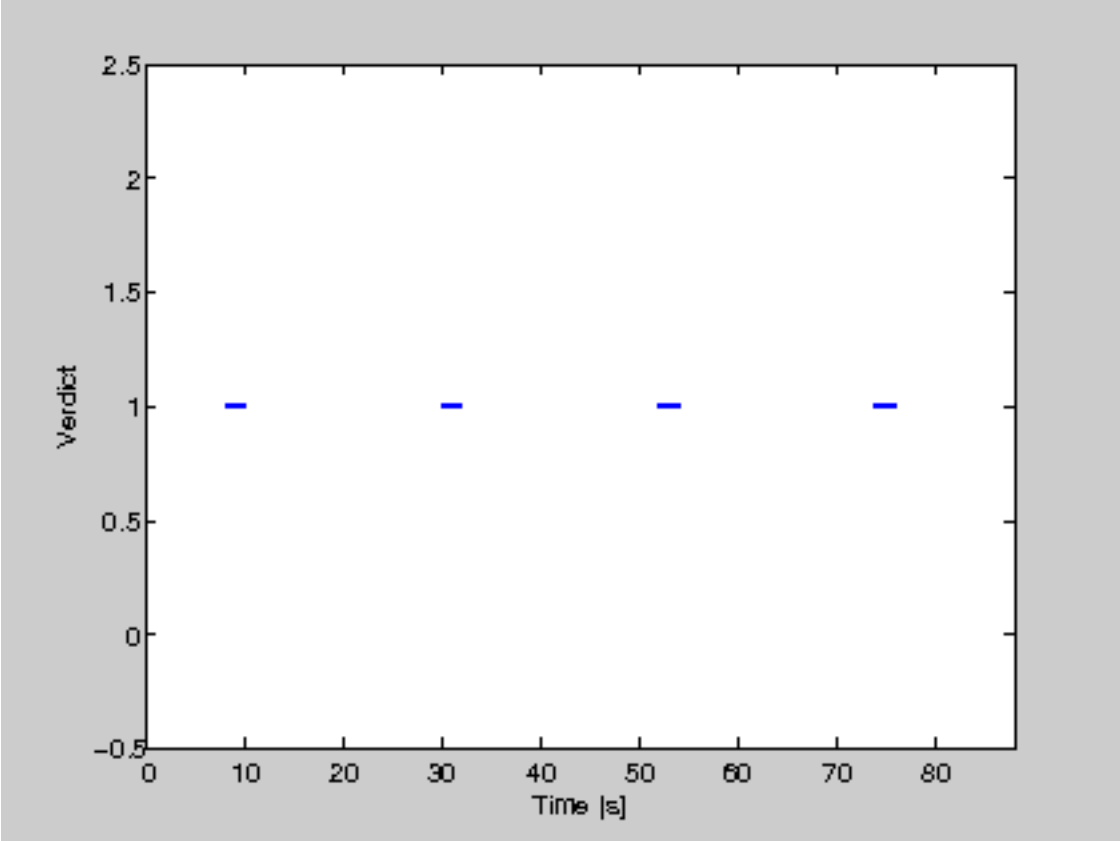
```
the Precondition : true


Validation Results

Overall verdict: PASS
```

- FAIL: 0% (0/8801)
- PASS: 100% (8801/8801)
- NONE: 0% (0/8801)

Figure 3.8. Validation Result



Test Case 4

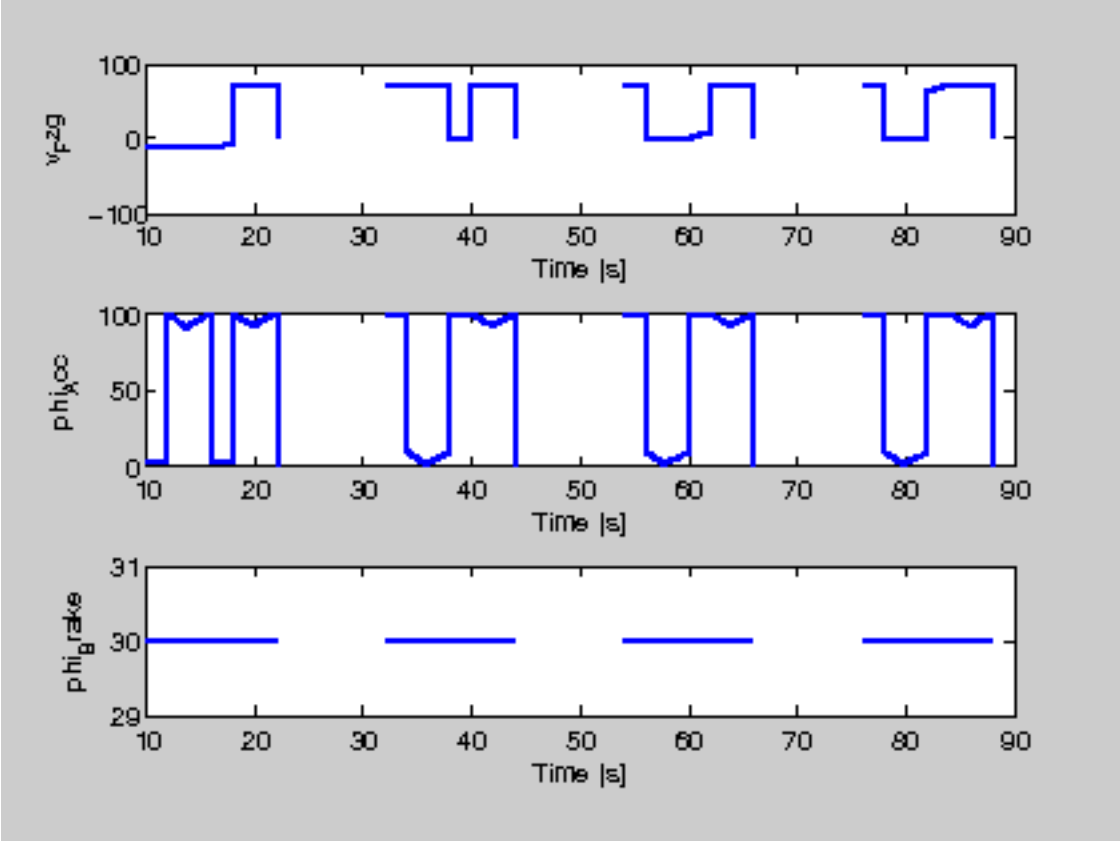
In the requirement : SR-PI-02.2 Acceleration pedal interpretation

```
Test Data in the test case: SR-PI-02.2 Acceleration pedal interpretation
v_Fzg  minimum value:-10
v_Fzg  maximum value:70

Test Data in the test case: SR-PI-02.2 Acceleration pedal interpretation
phi_Acc  minimum value:0
phi_Acc  maximum value:100

Test Data in the test case: SR-PI-02.2 Acceleration pedal interpretation
phi_Brake  minimum value:30
phi_Brake  maximum value:30
```

Figure 3.9. Test Data in iterations of the test case SR-PI-02.2 Acceleration pedal interpretation



Precondition 4 . 1

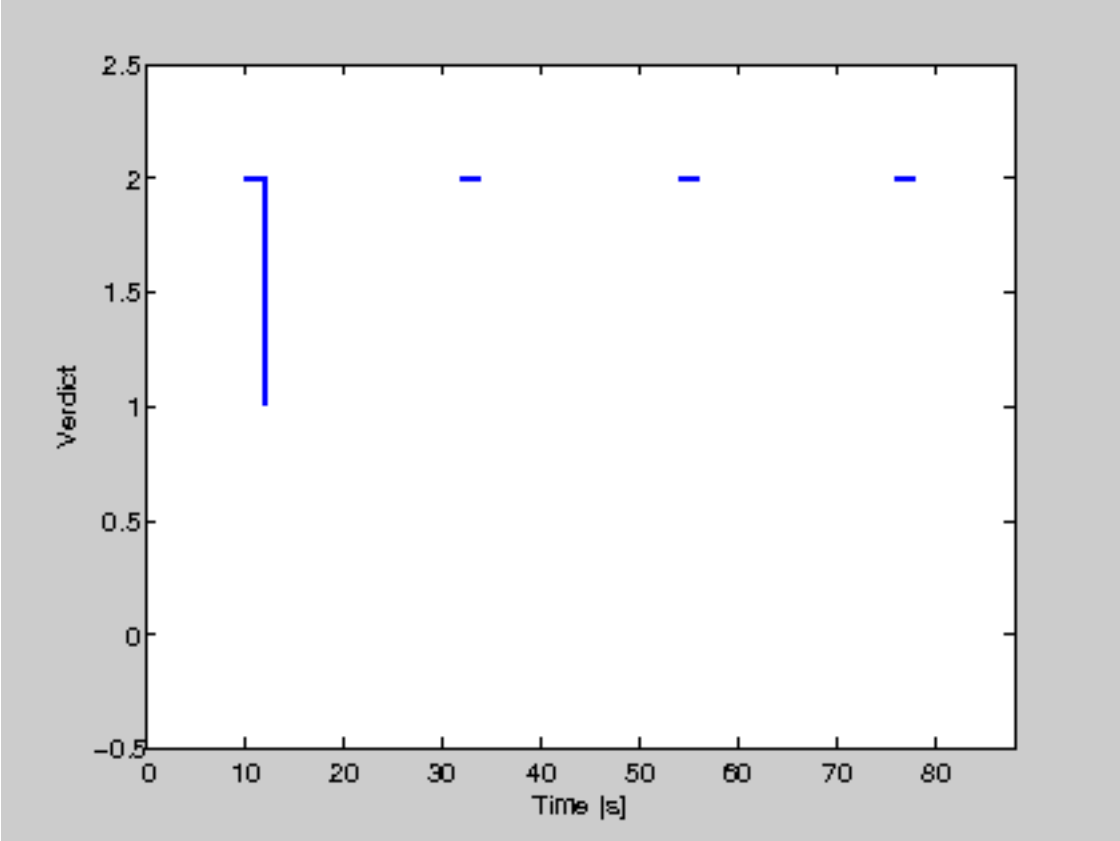
```
the Precondition : v = const & phi_Acc = const
```

```
Validation Results
```

```
Overall verdict: PASS
```

- FAIL: 0% (0/8801)
- PASS: 13% (1205/8801)
- NONE: 86% (7596/8801)

Figure 3.10. Validation Result



Precondition 4 . 2

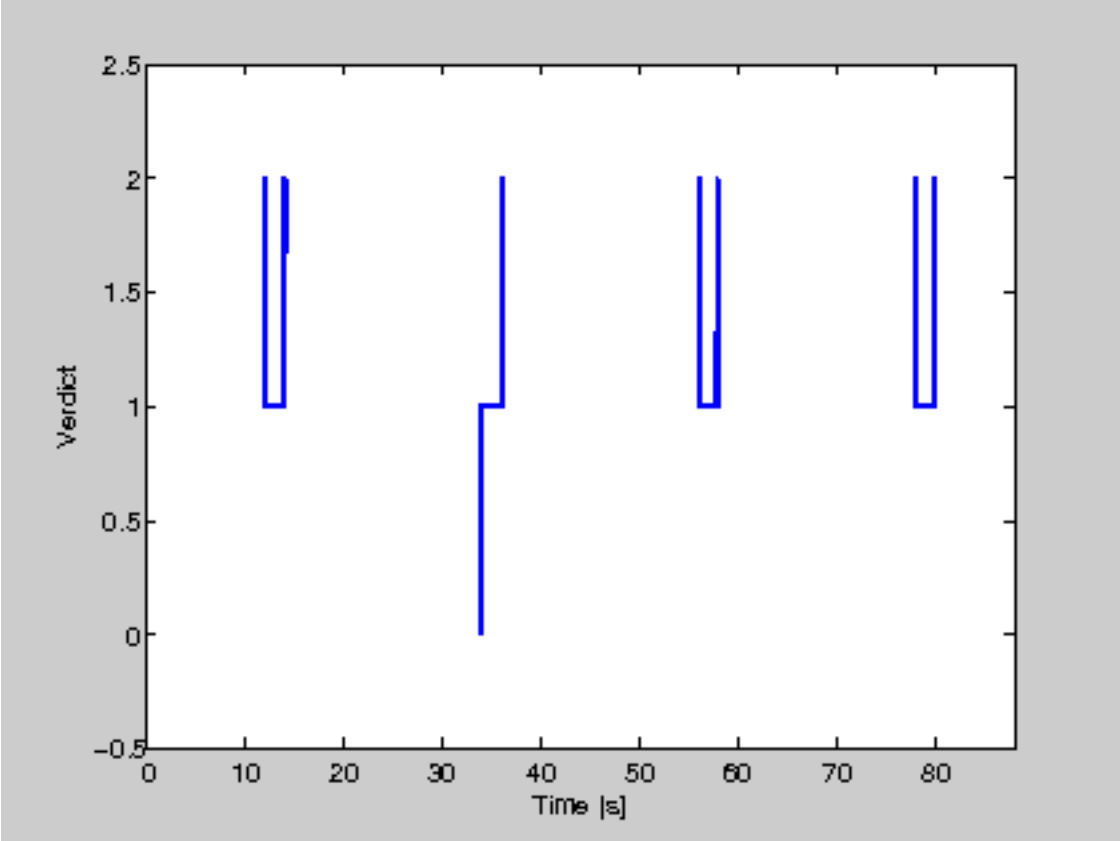
```
the Precondition : v = const & phi_Acc decreases & T_des_Drive < 0
```

```
Validation Results
```

```
Overall verdict: FAIL
```

- FAIL: 0% (1/8801)
- PASS: 13% (1199/8801)
- NONE: 86% (7601/8801)

Figure 3.11. Validation Result



Precondition 4 . 3

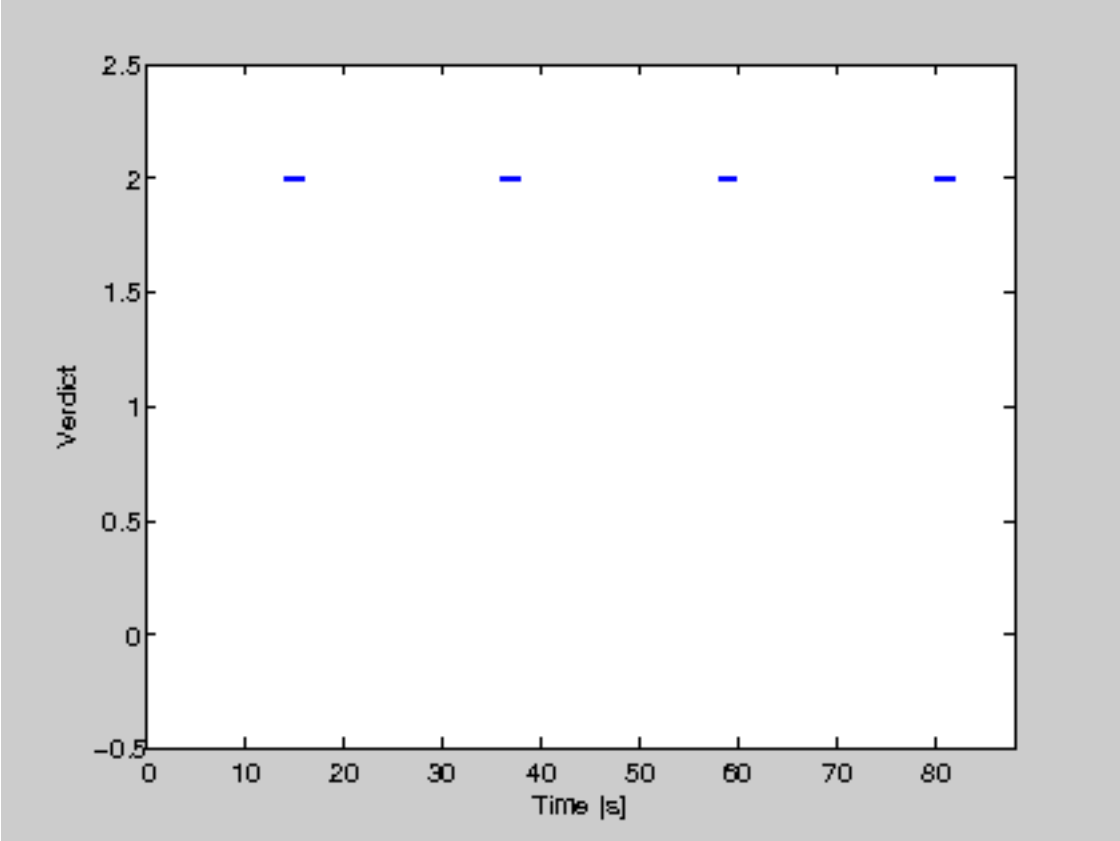
```
the Precondition : v = const & phi_Acc decreases & T_des_Drive >= 0
```

```
Validation Results
```

```
Overall verdict: PASS
```

- FAIL: 0% (0/8801)
- PASS: 35% (3091/8801)
- NONE: 64% (5710/8801)

Figure 3.12. Validation Result



Precondition 4 . 4

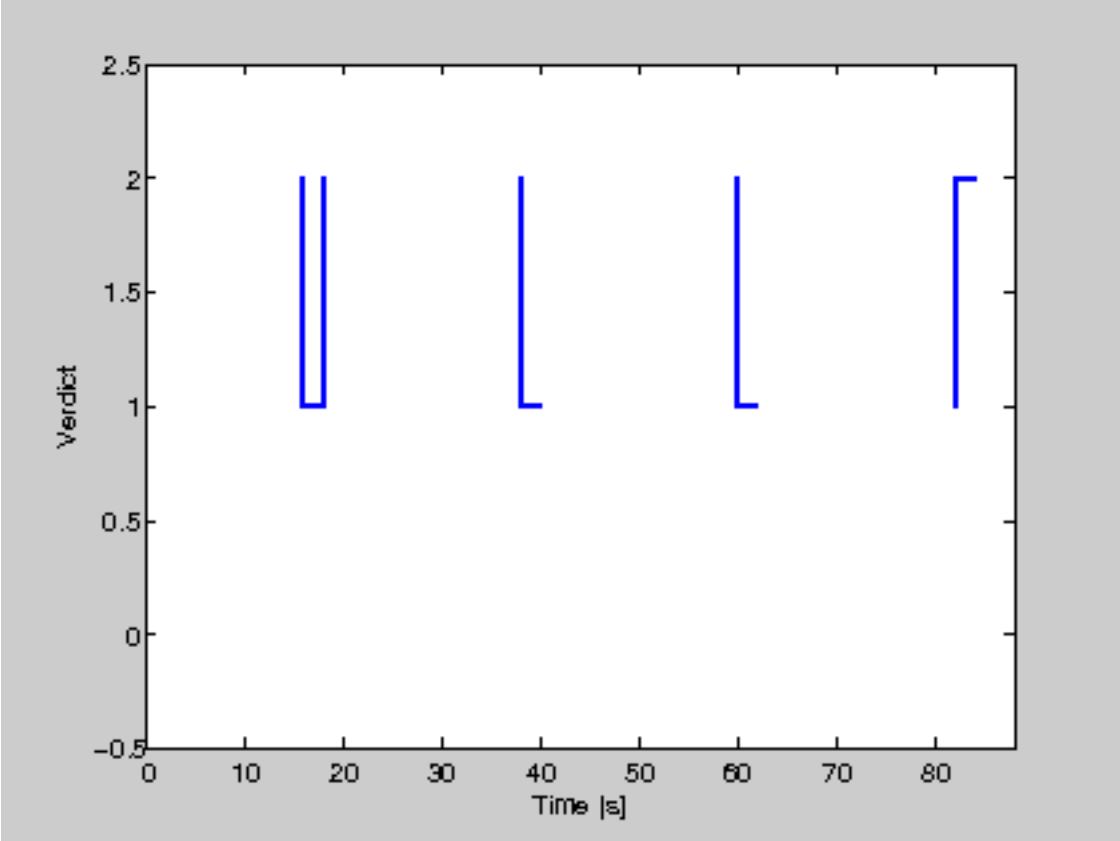
```
the Precondition : v = const & phi_Acc increases & T_des_Drive < 0
```

```
Validation Results
```

```
Overall verdict: PASS
```

- FAIL: 0% (0/8801)
- PASS: 6% (600/8801)
- NONE: 93% (8201/8801)

Figure 3.13. Validation Result



Precondition 4 . 5

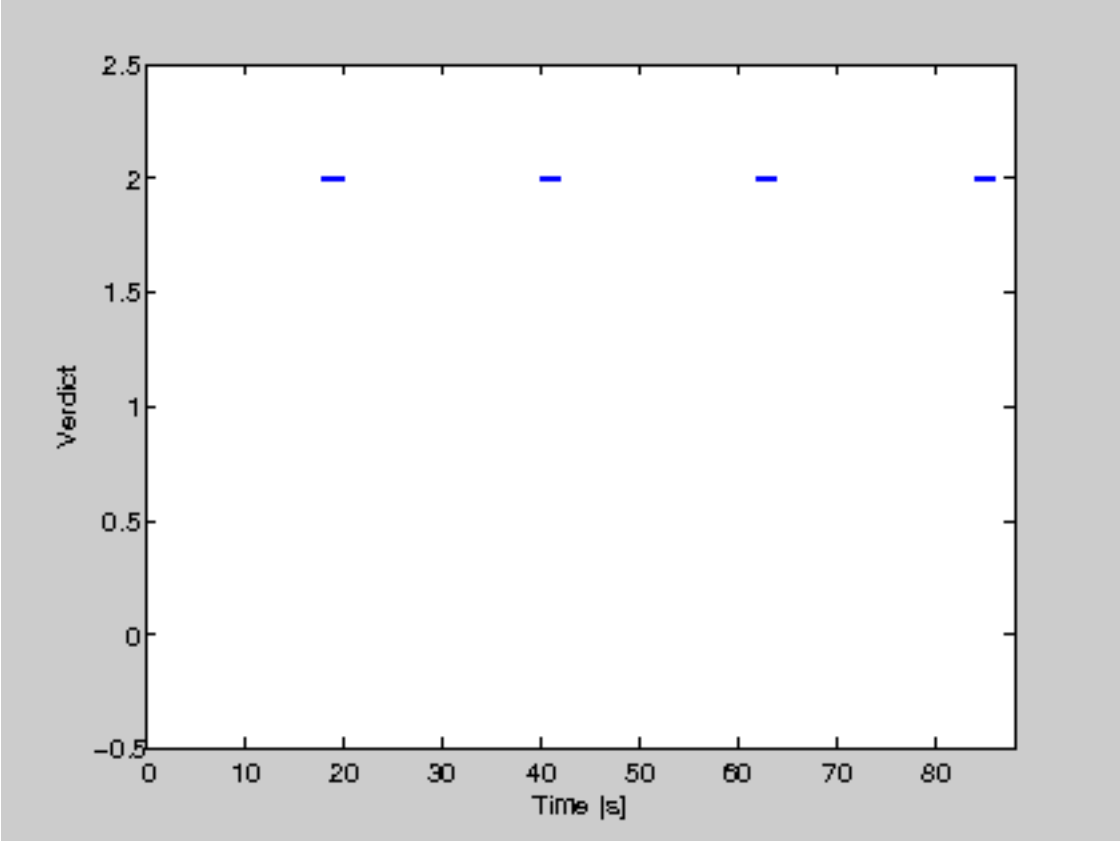
```
the Precondition : v = const & phi_Acc increases & T_des_Drive >= 0
```

```
Validation Results
```

```
Overall verdict: PASS
```

- FAIL: 0% (0/8801)
- PASS: 9% (796/8801)
- NONE: 90% (8005/8801)

Figure 3.14. Validation Result



Precondition 4 . 6

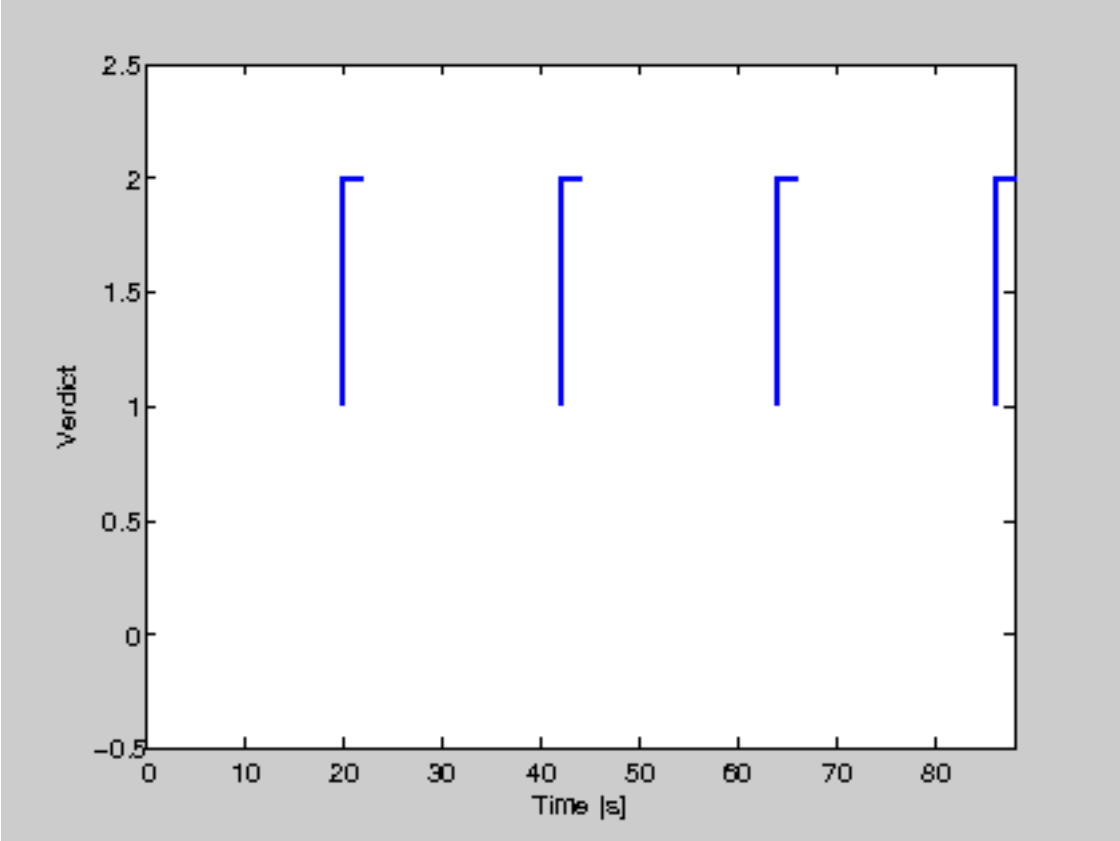
```
the Precondition : v increases & phi_Acc = const & T_des_Drive >= 0
```

```
Validation Results
```

```
Overall verdict: PASS
```

- FAIL: 0% (0/8801)
- PASS: 9% (800/8801)
- NONE: 90% (8001/8801)

Figure 3.15. Validation Result


2.5
2
1.5
1
0.5
0
−0.5
Verdict
0
10
20
30
40
50
60
70
80
Time [s]

Chapter 4. Test Efficiency

Table of Contents

Describe the efficiency of the test data as demonstrated by the tests. Find the output signal coverage via a test coverage chart

Input Partitions

```
SUT Input signal : v_Fzg
Signal Range : [-10,70]
Partition Point(s) : {0}
Expected Signal Range is divided into:
[-10,0][0,70]
Actual Test Signal: [-10,70]
input coverage is: 2/2- 100%
------------------------
SUT Input signal : phi_Acc
Signal Range : [0,100]
Partition Point(s) : empty
Actual Test Signal: [0,100]
input coverage is: 1/1- 100%
/
------------------------
SUT Input signal : phi_Brake
Signal Range : [0,100]
Partition Point(s) : empty
Actual Test Signal: [0,99.9525]
input coverage is: 1/1- 100%
/
------------------------
```

Output Coverage

Section 1 : Acc Pedal

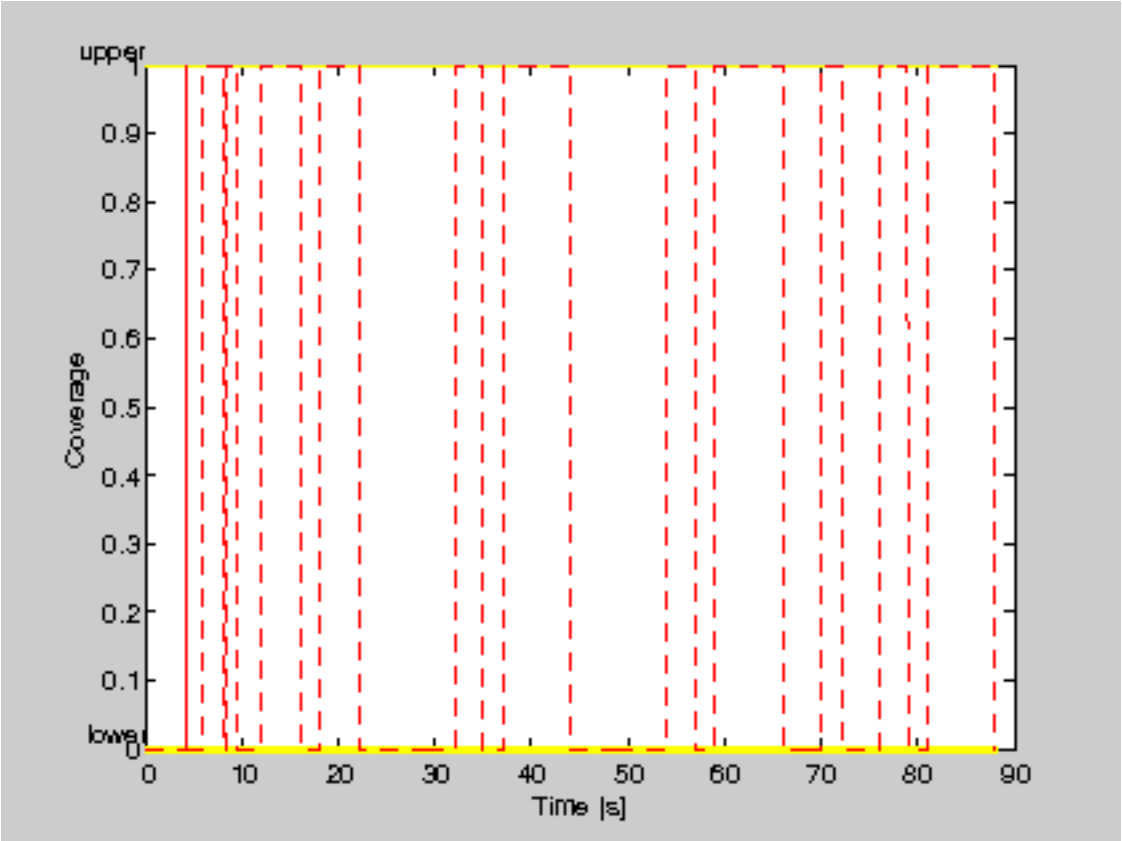
Test Coverage of output variable: Acc Pedal is: 100 %.

```
Partitions Point(s) : []

Output Signal Range : [0 1]

Actual Output Domain : [0 1]
```

Figure 4.1. Test Coverage Chart for Acc Pedal

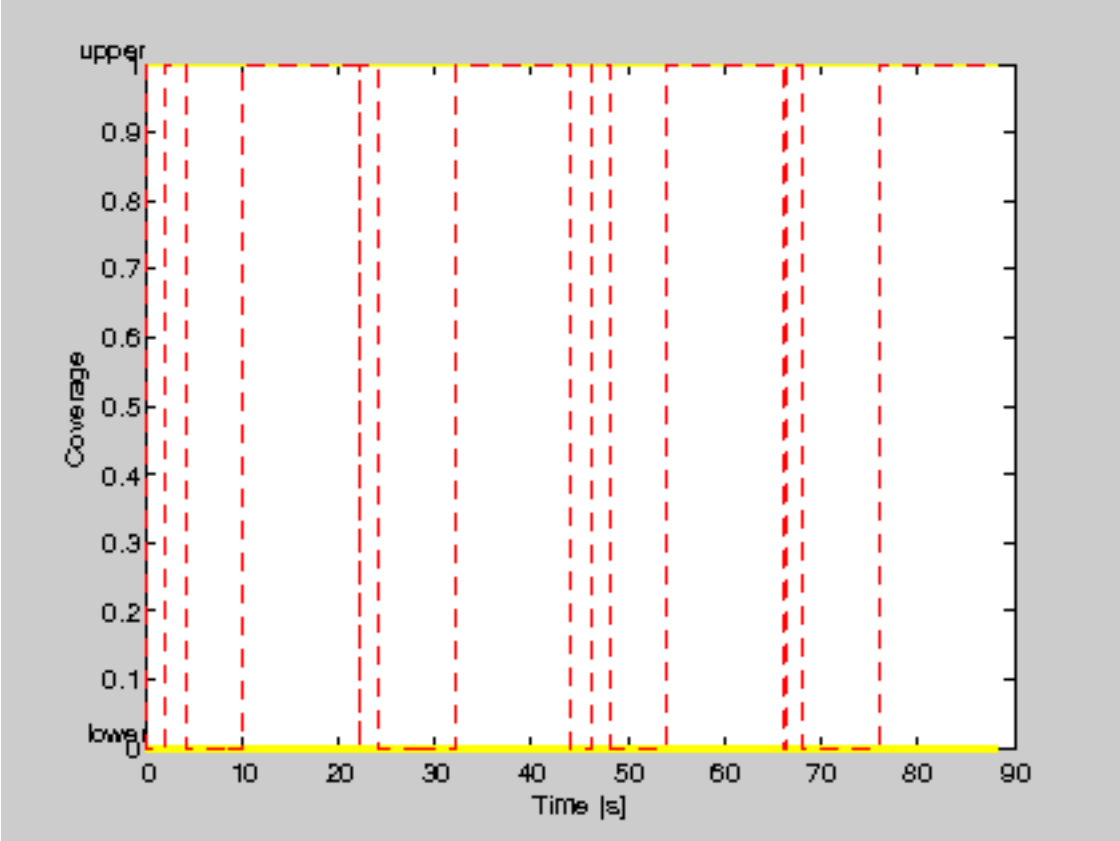


Section 2 : Brake Pedal

Test Coverage of output variable: Brake Pedal is: 100 %.

```
Partitions Point(s) : []
Output Signal Range : [0 1]
Actual Output Domain : [0 1]
```

Figure 4.2. Test Coverage Chart for Brake Pedal

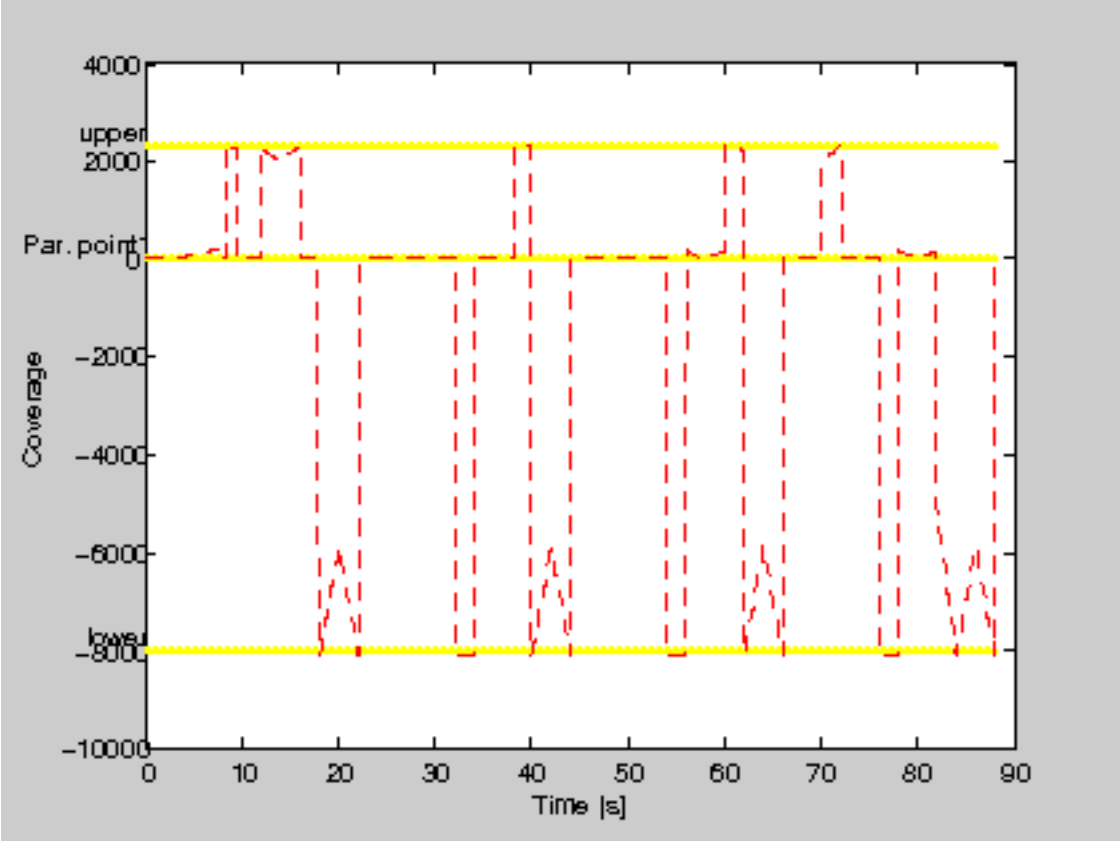


Section 3 : T_des Drive

Test Coverage of output variable: T_des Drive is: 100 %.

```
Partitions Point(s) : 0
Output Signal Range : [-8000 2300]
Actual Output Domain : [-8.0996e+003 2300]
```

Figure 4.3. Test Coverage Chart for T_des Drive

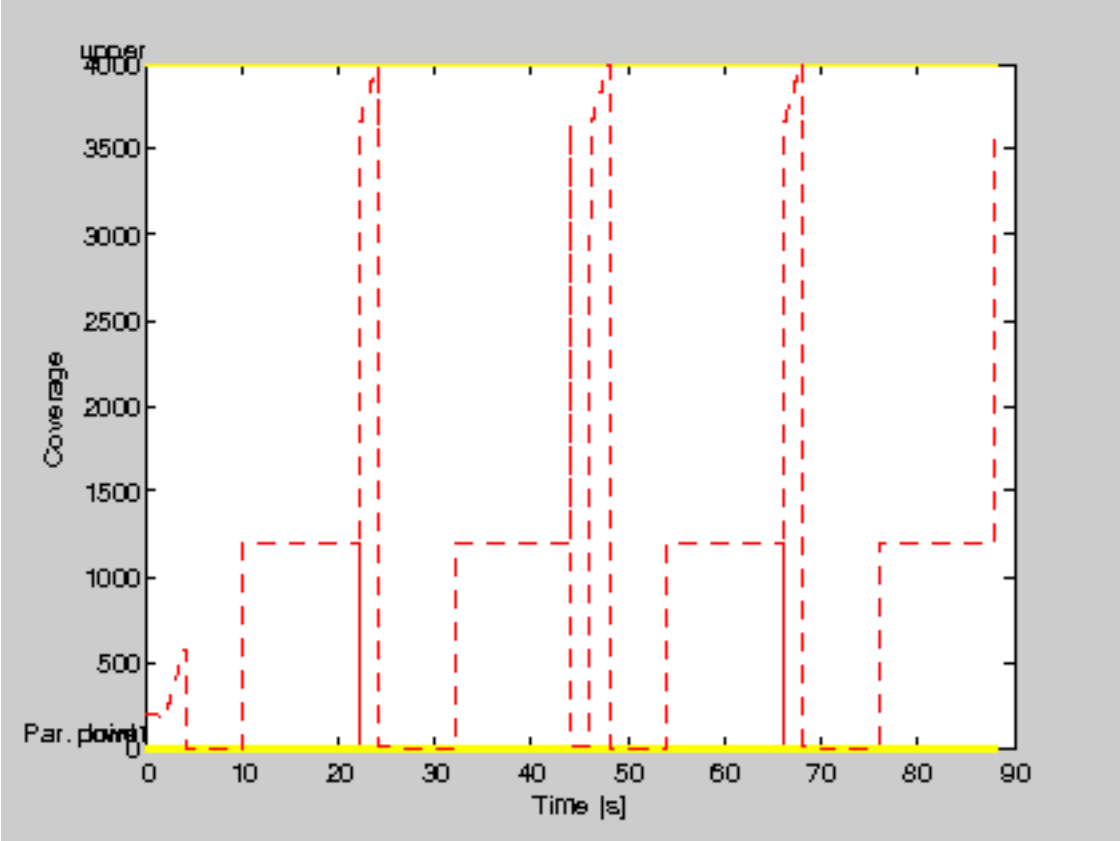


Section 4 : T_des_Brake

Test Coverage of output variable: T_des_Brake is: 100 %.

```
Partitions Point(s) : 0
Output Signal Range : [0 4000]
Actual Output Domain : [0 3.9981e+003]
```

Figure 4.4. Test Coverage Chart for T_des_Brake


upper
4000
3500
3000
2500
2000
1500
1000
500
0
Coverage
Par. point1
0
10
20
30
40
50
60
70
80
90
Time [s]