**Module Design Document**

**For**

**InertiaCmpVel**

**Jul 14, 2016**

**Prepared For:**

**Software Engineering**

**Nexteer Automotive,**

**Saginaw, MI, USA**

**Prepared By:**

**Krishna Anne,**

**Nexteer Automotive,**

**Saginaw, MI, USAChange History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SNo** | **Description** | **Author** | **Version** | **Date** |
| 1 | Initial Version | SB | 1.0 | 23-Jul-2015 |
| 2 | Updated to version 1.3.0 of design | SB | 2.0 | 11-Mar-2016 |
| 3 | Updated to version 1.7.0 and 1.8.0 of design | KK | 3.0 | 21-Jun-2016 |
| 4 | Updated to version 1.9.0 of design | KK | 4.0 | 14-Jul-2016 |

**Table of Contents**

[1 Introduction 4](#_Toc425521158)

[1.1 Purpose 4](#_Toc425521159)

[1.2 Scope 4](#_Toc425521160)

[2 InertiaCmpVel & High-Level Description 5](#_Toc425521161)

[3 Design details of software module 6](#_Toc425521162)

[3.1 Graphical representation of InertiaCmpVel 6](#_Toc425521163)

[3.2 Data Flow Diagram 7](#_Toc425521164)

[3.2.1 Component level DFD 7](#_Toc425521165)

[3.2.2 Function level DFD 7](#_Toc425521166)

[4 Constant Data Dictionary 8](#_Toc425521167)

[4.1 Program (fixed) Constants 8](#_Toc425521168)

[4.1.1 Embedded Constants 8](#_Toc425521169)

[5 Software Component Implementation 9](#_Toc425521170)

[5.1.1 Sub-Module Functions 9](#_Toc425521171)

[5.1.2 Interrupt Service Routines 9](#_Toc425521172)

[5.1.3 Server Runnable Functions 9](#_Toc425521173)

[5.1.4 Module Internal (Local) Functions 9](#_Toc425521174)

[5.1.5 Transition Functions 11](#_Toc425521175)

[6 Known Limitations with Design 12](#_Toc425521176)

[7 UNIT TEST CONSIDERATION 13](#_Toc425521177)

[Appendix A Abbreviations and Acronyms 14](#_Toc425521178)

[Appendix B Glossary 15](#_Toc425521179)

[Appendix C References 16](#_Toc425521180)

# Introduction

## Purpose

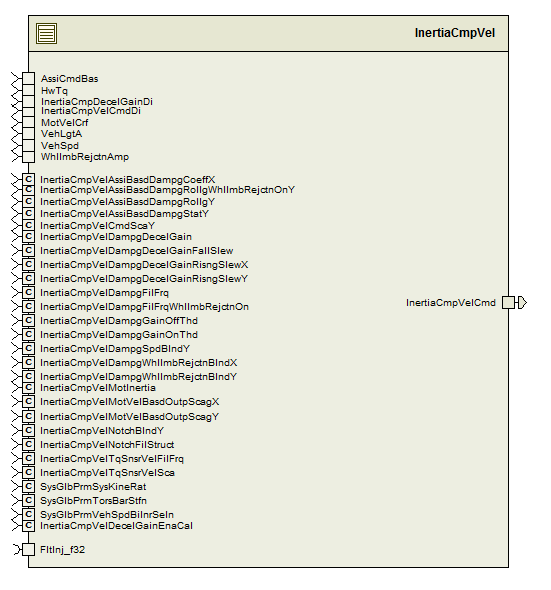
## Scope

# InertiaCmpVel & High-Level Description

*Refer FDD*

# Design details of software module

## Graphical representation of InertiaCmpVel

**

## Data Flow Diagram

### Component level DFD

Refer FDD

### Function level DFD

Refer FDD

# Constant Data Dictionary

## Program (fixed) Constants

### Embedded Constants

#### Local Constants

None

#### Global Constants

Refer .m file

#### User defined typedef definition/declaration

This section documents any user types uniquely used for the module.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Typedef Name | Element Name | User Defined Type | Legal Range  (min) | Legal Range  (max) |
| typedef struct FilCoeffRec | b0\_Uls\_f32 | Float32 | FULL | FULL |
|  | b1\_Uls\_f32 | Float32 | FULL | FULL |
|  | b2\_Uls\_f32 | Float32 | FULL | FULL |
|  | a0\_Uls\_f32 | Float32 | FULL | FULL |
|  | a1\_Uls\_f32 | Float32 | FULL | FULL |
|  | a2\_Uls\_f32 | Float32 | FULL | FULL |

# Software Component Implementation

### Sub-Module Functions

#### Initialization sub-module InertiaCmpVelInit1()

**Design Rational:**

**Init function is not present in the model but in reference to the Init.txt text file Low pass filter and Notch filter are initialized.**

**For Low pass filter standard EA4 LPF implementation from NxtrFil.h is followed and for Notch filter initialization, EA3 implementation is followed.**

#### Periodic sub-module InertiaCmpVelPer1()

### Interrupt Service Routines

None

### Server Runnable Functions

None

### Module Internal (Local) Functions

#### Calculate Driver Velocity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | DrvrVelCalc | Type | Min | Max |
| **Arguments Passed** | HwTq\_HwNwtMtr\_T\_f32 | float32 | -10 | 10 |
|  | MotVelCrf\_MotRadPerSec\_T\_f32 | float32 | -1350 | 1350 |
|  | VehSpd\_Kph\_T\_f32 | float32 | 0 | 511 |
| **Return Value** | ScadDrvrVel\_MotRadPerSec\_T\_f32 | float32 | -1350 | 1350 |

#### Calculate ADD Coefficient

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ADDCoeffCalc | Type | Min | Max |
| **Arguments Passed** | AssiCmdBas\_MotNwtMtr\_T\_f32 | float32 | -8.8 | 8.8 |
|  | WhlImbRejctnAmp\_MotNwtMtr\_T\_f32 | float32 | 0 | 8.8 |
|  | VehSpd\_Kph\_T\_f32 | float32 | 0 | 511 |
| **Return Value** | ADDCoeffCalc\_MotNwtMtrSpRad\_T\_f32 | float32 | 0.0 | 0.00007 |

#### Calculate Gain

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | DecelGain | Type | Min | Max |
| **Arguments Passed** | VehLgtA\_KphPerSec\_T\_f32 | float32 | -35 | 35 |
|  | MotVelCrf\_MotRadPerSec\_T\_f32 | float32 | -1350 | 1350 |
| **Return Value** | DecelGain\_Uls\_T\_f32 | float32 | 0 | 1 |

#### Calculate Filter Coefficients

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | FilCoeffCalc | | Type | Min | Max |
| **Arguments Passed** | ADDCoeff\_MotNwtMtrPerMotRadPerSec\_T\_f32 | | float32 | 0.0 | 0.041306 |
|  | WhlImbRejctnAmp\_MotNwtMtr\_T\_f32 | | float32 | 0 | 8.8 |
|  | VehSpd\_Kph\_T\_f32 | | float32 | 0 | 511 |
| **Return Value** | \*FilCoeff\_T\_Rec | b0\_Uls\_f32 | float32 | -2.74156205240179 | 0 |
|  |  | b1\_Uls\_f32 | float32 | 0.0 | 0.330448 |
|  |  | b2\_Uls\_f32 | float32 | -0.160083862455113 | 2.41111405240179 |
|  |  | a0\_Uls\_f32 | float32 | 0.5525885 | 3.9498924 |
|  |  | a1\_Uls\_f32 | float32 | -7.9996842 | -4.8417266 |
|  |  | a2\_Uls\_f32 | float32 | 4.0504234 | 10.6056849 |

#### Generate Command

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | GenFddIcCmd | | Type | Min | Max |
| **Arguments Passed** | ScadDrvrVel\_MotRadPerSec\_T\_f32 | | float32 | -7226.652 | 7226.652 |
|  | \*FilCoeff\_T\_Rec | b0\_Uls\_f32 | float32 | -2.74156205240179 | 0 |
|  | b1\_Uls\_f32 | float32 | 0.0 | 0.330448 |
|  | b2\_Uls\_f32 | float32 | -0.166262133009164 | 2.41111405240179 |
|  | a0\_Uls\_f32 | float32 | 0.5525885 | 3.9498924 |
|  | a1\_Uls\_f32 | float32 | -7.9996842 | -4.8417266 |
|  | a2\_Uls\_f32 | float32 | 4.0504234 | 10.6056849 |
| **Return Value** | InertiaCmp\_MotNwtMtr\_T\_f32 | | Float | -8.8 | 8.8 |

#### NotchCmp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | NotchCmp | Type | Min | Max |
| **Arguments Passed** | VehSpd\_Kph\_T\_f32 | float32 | 0 | 511 |
|  | InertiaCmp\_MotNwtMtr\_T\_f32 | float32 | -8.8 | 8.8 |
| **Return Value** | NotchCmp \_MotNwtMtr\_T\_f32 | float32 | -8.8 | 8.8 |

#### FilNotchFullUpdOutp\_f32

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | FilNotchFullUpdOutp\_f32 | Type | Min | Max |
| **Arguments Passed** | Inp | float32 | See unit test consideration | |
|  | FilNotchStRecPtr | FilNotchStRec1 |
|  | FilNotchGainRecPtr | FilNotchGainRec1 |
| **Return Value** | None |  |

##### Description

Notch filter output calculation implemented based on ‘Inertia Comp Notch’ block functionality.

#### FilNotchInit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | FilNotchInit | Type | Min | Max |
| **Arguments Passed** | Inp | float32 | See unit test consideration | |
|  | FilNotchStRecPtr | FilNotchStRec1 |
|  | FilNotchGainRecPtr | FilNotchGainRec1 |
| **Return Value** | FilOut | float32 |

##### Description

Notch filter initialization function implemented based on EA3 design.

### Transition Functions

None

# Known Limitations with Design

None

# UNIT TEST CONSIDERATION

1. Since the notch filter implementation used in this module is dynamic in nature, absolute ranges are difficult to determine without pre-defined knowledge on the combination of coefficient values (A1, A2, B0, B1, B2). Because of this, the systems group ran simulations on 10 different combinations of coefficients (2 with defined default calibrations, 8 considered extreme cases of notch filters) and logged the ranges of the filter state variables and outputs during a frequency sweep. The ranges given throughout this module were taken as the worst case results of all of the given test cases.

To provide useful cases for unit testing, the boundary checks tested during unit testing should be altered to test the state variable minimum and maximum for each of the 10 test cases with the given coefficients set to the values given in that test case. In the case where the default values of the coefficients are used in a vector, the unit tester should not test the corresponding state variables with values over the range defined for that set of coefficients. See attached simulation results.



1. GenFddIcCmd function is designed to work with argument values from the calling function as used with the other functions in the module, and outputs may be out of the expected range if tested with arbitrary combinations of input values. Unit testing of this function should use only passed argument value combinations coming from the calling function.

Abbreviations and Acronyms

| **Abbreviation or Acronym** | **Description** |
| --- | --- |
|  |  |
|  |  |

Glossary

**Note**: Terms and definitions from the source “Nexteer Automotive” take precedence over all other definitions of the same term. Terms and definitions from the source “Nexteer Automotive” are formulated from multiple sources, including the following:

* ISO 9000
* ISO/IEC 12207
* ISO/IEC 15504
* Automotive SPICE® Process Reference Model (PRM)
* Automotive SPICE® Process Assessment Model (PAM)
* ISO/IEC 15288
* ISO 26262
* IEEE Standards
* SWEBOK
* PMBOK
* Existing Nexteer Automotive documentation

| **Term** | **Definition** | **Source** |
| --- | --- | --- |
| MDD | Module Design Document |  |
| DFD | Data Flow Diagram |  |

References

| **Ref. #** | **Title** | **Version** |
| --- | --- | --- |
| 1 | AUTOSAR Specification of Memory Mapping (Link:[AUTOSAR\_SWS\_MemoryMapping.pdf](http://www.autosar.org/download/R4.0/AUTOSAR_SWS_MemoryMapping.pdf)) | v1.3.0 R4.0 Rev 2 |
| 2 | MDD Guideline | EA4 01.00.00 |
| 3 | [Software Naming Conventions.doc](http://misagweb01.nexteer.com/eRoomReq/Files/erooms8/NextGeneration/0_fc55f/Software%20Naming%20Conventions%2003x(In%20Work).doc) | 2.0 |
| 4 | [Software Design and Coding Standards.doc](http://eroom1.nexteer.com/eRoomReq/Files/erooms8/NextGeneration/0_1a67a9/Software%20Design%20and%20Coding%20Standards.doc) | 2.1 |
| 5 | FDD – SF014A\_InetiaCmpVel\_Design | See Synergy Sub project version |