# Module -- Stability Compensation

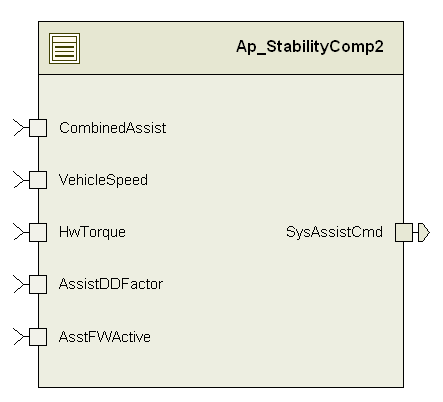
# High-Level Description

This function provides in-vehicle stability of EPS behavior. To maximize steering feel, the function blends between two different tunings based on vehicle speed and low frequency handwheel torque. Because system gains may get multiplied by various scale factors, either from serial communications or from other software functions, this function provides a second blending feature.

Some vehicle programs require a secondary, or "systematic", calculation of stability compensation followed by a correlation check. The systematic calculations usually must reside in a separate Memory Partition Unit (MPU), and switching between MPUs negatively impacts system throughput. This model is designed using the best-known implementation at this time.

# Figures

## Component Diagram



## Variable Data Dictionary

For details on module input / output variable, refer to the Data Dictionary for the application. Input / output variable names are listed here for reference.

|  |  |  |
| --- | --- | --- |
| Module Inputs | Module Outputs | |
| AssistDDFactor\_Uls\_f32 | | SysAssistCmd\_MtrNm\_f32 |
| HwTorque\_HwNm\_f32 | |  |
| VehicleSpeed\_Kph\_f32 | |  |
| CombinedAssist\_MtrNm\_f32 | |  |
| AsstFWActive\_Uls\_f32 | |  |

## Module Internal Variables

This section identifies the name, range and resolutions for module specific data created by this module. If there are no range restrictions on the variable, the term “FULL” is placed into the table for legal range.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variable Name | Resolution | Legal Range  (min) | Legal Range  (max) | Software Segment |
| StCmp1Out2\_MtrNm\_D\_f32 | Single Precision Float | -2312 | 2312 | STABILITYCOMP2\_START\_SEC\_VAR\_CLEARED\_32 |
| StCmp2Out2\_MtrNm\_D\_f32 | Single Precision Float | -2312 | 2312 | STABILITYCOMP2\_START\_SEC\_VAR\_CLEARED\_32 |
| StCmp3Out2\_MtrNm\_D\_f32 | Single Precision Float | -2312 | 2312 | STABILITYCOMP2\_START\_SEC\_VAR\_CLEARED\_32 |
| StCmp4Out2\_MtrNm\_D\_f32 | Single Precision Float | -2312 | 2312 | STABILITYCOMP2\_START\_SEC\_VAR\_CLEARED\_32 |
| HwTorqueSV\_HwNm\_M\_f32 | Single Precision Float | -10 | 10 | STABILITYCOMP2\_START\_SEC\_VAR\_CLEARED\_32 |
| VehicleSpeedSV\_Kph\_M\_f32 | Single Precision Float | 0 | 511.9921875 | STABILITYCOMP2\_START\_SEC\_VAR\_CLEARED\_32 |
| AssistDDFactorSV\_Uls\_M\_f32 | Single Precision Float | 1 | 2 | STABILITYCOMP2\_START\_SEC\_VAR\_CLEARED\_32 |
| CombAstNFSV2\_Cnt\_M\_Str\* | N/A | N/A | N/A | STABILITYCOMP2\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| CombAstNFSV2\_Cnt\_M\_Str[].SV1\_Uls\_f32 | Single Precision Floating Point | -2077 | 2077 |  |
| CombAstNFSV2\_Cnt\_M\_Str[].SV2\_Uls\_f32 | Single Precision Floating Point | -1981 | 1981 |  |
| CombAstNFSV2\_Cnt\_M\_Str[].Out\_Uls\_f32 | Single Precision Floating Point | -2312 | 2312 |  |
| CombAstNFSV2\_Cnt\_M\_Str[].KPtr\_Cnt\_Str->A1\_Uls\_f32 | Single Precision Floating Point | -2 | 0 |  |
| CombAstNFSV2\_Cnt\_M\_Str[].KPtr\_Cnt\_Str->A2\_Uls\_f32 | Single Precision Floating Point | -1 | 1 |  |
| CombAstNFSV2\_Cnt\_M\_Str[].KPtr\_Cnt\_Str->B0\_Uls\_f32 | Single Precision Floating Point | 0 | 2300 |  |
| CombAstNFSV2\_Cnt\_M\_Str[].KPtr\_Cnt\_Str->B1\_Uls\_f32 | Single Precision Floating Point | -4500 | 0 |  |
| CombAstNFSV2\_Cnt\_M\_Str[].KPtr\_Cnt\_Str->B2\_Uls\_f32 | Single Precision Floating Point | -4 | 2300 |  |
| HwTrqSV2\_HwNm\_M\_Str | N/A | N/A | N/A | STABILITYCOMP\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED |
| HwTrqSV2\_HwNm\_M\_Str.K\_Uls\_f32 | Single Precision Floating Point | 0.000125 | 0.4666 |  |
| HwTrqSV2\_HwNm\_M\_Str.SV\_Uls\_f32 | Single Precision Floating Point | 0 | 10 |  |

\* See unit testing notes in section 2.3.1 for range considerations of Notch Filter

### Notch filter Unit testing considerations

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.



### 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) |
| None |  |  |  |  |

# Constant Data Dictionary

## Calibration Constants

This section lists the calibrations used by the module. For details on calibration constants, refer to the Data Dictionary for the application.

|  |
| --- |
| Constant Name |
| t\_StCmpADDFBlendX\_Uls\_u2p14[ ] |
| t\_StCmpADDFBlendY\_Uls\_u2p14[ ] |
| t\_StCmpBlendSpdBS\_Kph\_u9p7[ ] |
| t\_StCmpBlend12Trq\_HwNm\_u8p8[ ] |
| t2\_StCmpBlend12TblY\_Uls\_u2p14[ ][ ] |
| t2\_StCmpBlend02TblY\_Uls\_u2p14[ ][ ] |
| t\_StCmpBlend34Trq\_HwNm\_u8p8[ ] |
| t2\_StCmpBlend34TblY\_Uls\_u2p14[ ][ ] |
| t2\_StCmpBlend04TblY\_Uls\_u2p14[ ][ ] |
| t\_StCmpNFK\_Str[ ] |
| t\_StCmpNFK\_Str[ ].A1\_Uls\_f32 |
| t\_StCmpNFK\_Str[ ].A2\_Uls\_f32 |
| t\_StCmpNFK\_Str[ ].B0\_Uls\_f32 |
| t\_StCmpNFK\_Str[ ].B1\_Uls\_f32 |
| t\_StCmpNFK\_Str[ ].B2\_Uls\_f32 |
| k\_StCmpHwTrqLPFKn\_Hz\_f32 |
|  |
|  |

## Program (fixed) Constants

### Embedded Constants

All embedded constants whose values are provided in Eng units will be evaluated to the equivalent counts by using the FPM\_InitFixedPoint\_m() macro within the #define statement.

#### Local

|  |  |  |  |
| --- | --- | --- | --- |
| Constant Name | Resolution | Units | Value |
| D\_COMPINLIMIT\_MTRNM\_F32 | Single Precision Floating Point | MtrNm | 10.275 |
| D\_NUMNOTCHFILT\_CNT\_U08 | UINT08 | Cnt | (TableSize\_m(t\_StCmpNFK\_Str)) |
| D\_NUMNFBLENDS\_CNT\_U08 | UINT08 | Cnt | 4 |
| D\_ASSISTINIT\_MTRNM\_F32 | Single Precision Floating Point | MtrNm | 0 |
| D\_COMPBLND12IDX\_CNT\_U08 | UINT08 | Cnt | 0 |
| D\_COMPBLND34IDX\_CNT\_U08 | UINT08 | Cnt | 1 |
| D\_COMPBLND02IDX\_CNT\_U08 | UINT08 | Cnt | 2 |
| D\_COMPBLND04IDX\_CNT\_U08 | UINT08 | Cnt | 3 |
| D\_ASSTSCLMT\_MTRNM\_F32 | Single Precision Floating Point | MtrNm | 8.8 |

#### Global

This section lists the global constants used by the module. For details on global constants, refer to the Data Dictionary for the application.

|  |
| --- |
| Constant Name |
| D\_ZERO\_ULS\_F32 |
| D\_2MS\_SEC\_F32 |

### Module specific Lookup Tables Constants

(This is for lookup tables (arrays) with fixed values, same name as other tables)

|  |  |  |  |
| --- | --- | --- | --- |
| Constant Name | Resolution | Value | Software Segment |
| None |  |  |  |

# Functions/Macros used by the Sub-Modules

## Library Functions / Macros

The library and functions / Macros that are called by the various sub modules are identified below,

1. FPM\_FloatToFixed\_m
2. Limit\_m
3. NF\_Init\_f32
4. FPM\_FixedToFloat\_m
5. BilinearXYM\_u16\_u16Xu16YM\_Cnt
6. IntplVarXY\_u16\_u16Xu16Y\_Cnt
7. Blend\_f32
8. Abs\_f32\_m

## Data Hiding Functions

1. <None>

## Global Functions/Macros Defined by this Module

None

## Local Functions/Macros Used by this MDD only

### ApplyStabilityComp2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | ApplyStabilityComp2 | | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | Assist\_MtrNm\_T\_f32 | | float32 | -10.275 | 10.275 |  |
|  | FiltSVPtr\_Uls\_T\_Str\* | | Pointer to NotchFiltSV\_Str |  |  |  |
|  |  | .SV1\_Uls\_f32 | float32 | -2077 | 2077 | 0.000489 |
|  |  | .SV2\_Uls\_f32 | float32 | -1981 | 1981 | 0.000489 |
|  |  | .Out\_Uls\_f32 | float32 | -2312 | 2312 | 0.000489 |
|  |  | .KPtr\_Cnt\_Str->A1\_Uls\_f32 | float32 | -2 | 0 | 0.00005 |
|  |  | .KPtr\_Cnt\_Str->A2\_Uls\_f32 | float32 | -1 | 1 | 0.00005 |
|  |  | .KPtr\_Cnt\_Str->B0\_Uls\_f32 | float32 | 0 | 2300 | 0.00005 |
|  |  | .KPtr\_Cnt\_Str->B1\_Uls\_f32 | float32 | -4500 | 0 | 0.00005 |
|  |  | .KPtr\_Cnt\_Str->B2\_Uls\_f32 | float32 | -4 | 2300 | 0.00005 |
|  | CompBlnd\_Uls\_T\_u2p14 | | Pointer to uint16 | 0 | 1 |  |
|  | ADDFBlend\_Uls\_T\_u2p14 | | uint16 | 0 | 1 |  |
|  | WriteDisplayVars\_Cnt\_T\_lgc | | boolean | FALSE | TRUE |  |
| **Return Value** | FilteredOutput\_MtrNm\_T\_s4p11 | | sint16 (s4p11\_T) | -8.8 | 8.8 | 4.89E-04 |

#### \* See unit testing notes in section 2.3.1 for range considerations of Notch Filter

#### Description



# Software Module Implementation

## Runtime Environment (RTE) Initial Values

This section lists the initial values of data written by this module but controlled by the RTE. After RTE initialization, the data in this table will contain these values.

|  |  |
| --- | --- |
| Data | Value |
| AssistDDFactor\_Uls\_f32 | 1 |
| HwTorque\_HwNm\_f32 | 0 |
| CombinedAssist\_MtrNm\_f32 | 0 |
| AsstFWActive\_Uls\_f32 | 0 |
| VehicleSpeed\_Kph\_f32 | 0 |

## Initialization Functions

### Init: StabilityComp2\_Init1

#### Design Rationale

None



## Periodic Functions

### Per: StabilityComp\_Per1

#### Design Rationale

#### Program Flow Start

Rte\_Call\_StabilityComp2\_Per1\_CP0\_CheckpointReached()

#### Store Module Inputs to Local copies

AssistDDFactor\_Uls\_T\_f32 = Rte\_IRead\_StabilityComp2\_Per1\_AssistDDFactor\_Uls\_f32();

AsstFWActive\_Uls\_T\_f32 = Rte\_IRead\_StabilityComp2\_Per1\_AsstFWActive\_Uls\_f32();

CombinedAssist\_MtrNm\_T\_f32 = Rte\_IRead\_StabilityComp2\_Per1\_CombinedAssist\_MtrNm\_f32();

HwTorque\_HwNm\_T\_f32 = Rte\_IRead\_StabilityComp2\_Per1\_HwTorque\_HwNm\_f32();

VehicleSpeed\_Kph\_T\_f32 = Rte\_IRead\_StabilityComp2\_Per1\_VehicleSpeed\_Kph\_f32();





#### Store Local copy of outputs into Module Outputs

Rte\_IWrite\_StabilityComp2\_Per1\_SysAssistCmd\_MtrNm\_f32(SysAssistCmd\_MtrNm\_T\_f32);

#### Program Flow End

Rte\_Call\_StabilityComp2\_Per1\_CP1\_CheckpointReached()

## Fault Recovery Functions

## Shutdown Functions

## Interrupt Functions

## Serial Communication Functions

# Execution Requirements

## Execution Sequence of the Module

(Describe in words relevant details about the execution sequence of the different sub modules.)

## Execution Rates for sub-modules called by the Scheduler

This table serves as reference for the Scheduler design

|  |  |  |
| --- | --- | --- |
| Function Name | Calling Frequency | System State(s) in which the function is called |
| StabilityComp2\_Per1 | 2ms | ALL STATES |
| StabilityComp2\_Init1 | STARTUP | COLD INIT |

## Execution Requirements for Serial Communication Functions

|  |  |
| --- | --- |
| Function Name | Sub-Module called by (Serial Comm Function Name) |
| <None> |  |

# Memory Map Definition Requirements

## Sub Modules (Functions)

This table identifies the software segments for functions identified in this module.

|  |  |
| --- | --- |
| Name of Sub Module | Software Segment |
| StabilityComp2\_Per1 |  |

## Local Functions

This table identifies the software segments for local functions identified in this module.

|  |  |
| --- | --- |
| Name of Sub Module | Software Segment |
| ApplyStabilityComp2 |  |

# Known Issues / Limitations With Design

1. INLINE functions defined in GlobalMacro.h are not unit tested

# Revision Control Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item #** | **Rev #** | **Change Description** | **Date** | **Author Initials** |
| 1 | 1.0 | Initial MDD | 27 Feb 2013 | SP |
| 2 | 2.0 | Anomaly 4719 fixes | 05 Apr 2013 | SP |
| 4 | 4.0 | Anomaly 4886 Updates. Range updates and notes about unit testing the notch filter. | 27 Apr 2013 | LWW |
| 5 | 5.0 | Range updates on HwTrq filter SV. Added tolerance for pointer values on local function. | 06 June 2013 | LWW |
| 6 | 6 | Added cal ptr for parameter in function call to NF\_FullUpdate\_f32 | 05-Dec-13 | VT |