Document Title	Explanation of Application Interfaces of the Chassis Domain
Document Category	Explanation (EXP)
Document UID	270
Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R26-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2026-11-26	R26-11	AUTOSAR Release Management	• Splitting of the former chapter 2 into new chapters 2, 3 , and 4 • Editorial corrections
2025-11-27	R25-11	AUTOSAR Release Management	• No content changes
2024-11-27	R24-11	AUTOSAR Release Management	• No content changes
2023-11-23	R23-11	AUTOSAR Release Management	• No content changes
2022-11-24	R22-11	AUTOSAR Release Management	• No content changes
2021-11-25	R21-11	AUTOSAR Release Management	• No content changes
2020-11-30	R20-11	AUTOSAR Release Management	• No content changes
2019-11-28	R19-11	AUTOSAR Release Management	• No content changes • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• Editorial changes
2017-12-08	4.3.1	AUTOSAR Release Management	• Editorial changes

△

2016-11-30	4.3.0	AUTOSAR Release Management	• Editorial changes
2014-10-31	4.2.1	AUTOSAR Release Management	• Changing Status->state; current, actual -> consolidate after Harmonization with Engine
2013-10-31	4.1.2	AUTOSAR Release Management	• Add reference to Sensor Actuator Design Pattern (in chapter 2.5.4.1) • Delete old description for Internal state sensors (in chapter 2.5.4.1)
2013-03-15	4.1.1	AUTOSAR Administration	• Including new "Parking Aid" component • Harmonisation of names and descriptions (for components, interfaces etc.) in the whole document, update of related pictures • Clearing up of obsolete content, which is not valid for the next release
2010-02-02	3.1.4	AUTOSAR Administration	• Description of SW-ComponentTirePMon (Tire Pressure Monitoring) inserted (new in R4.0) • Description of SW-Component DtTqDistbn (Drivetrain Torque Distribution) inserted (new in R4.0) • Major extensions of the Description SW-Component CrsCtrlAndAcc (Cruise Control and Adaptive Cruise Control) inserted (decomposition missing in R3.0) • Chassis domain structure outlook including scalability and outlook on signal qualifer concept • Update of SW-Components (e.g. Higher-level VSS) inserted • Legal disclaimer revised
2009-02-04	3.1.2	AUTOSAR Administration	• Initial release
2008-08-13	3.1.1	AUTOSAR Administration	• Legal disclaimer revised
2007-12-21	3.0.1	AUTOSAR Administration	• Initial release

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Purpose of this Document 6

2 Acronyms and abbreviations 7

3 Description of terms and concepts 8
- 3.1 Concepts . . . 8
  - 3.1.1 Axis System . . . 8
  - 3.1.2 Definitions . . . 8
    - 3.1.2.1 Methodology for Optimization of Interface Definition in the Chassis Domain . . . 8
    - 3.1.2.2 Definitions / Explanations . . . 9
    - 3.1.2.3 Naming consistency rules . . . 11
  - 3.1.3 Centre of gravity of passenger car . . . 12
  - 3.1.4 Definition of the car surroundings (e.g. Used for ACC) . . . 13
- 3.2 General remarks . . . 18
  - 3.2.1 Differences between SW-Components and ECUs . . . 18
  - 3.2.2 Functional safety . . . 18
  - 3.2.3 Concept of core-, conditional- and optional- ports . . . 19
  - 3.2.4 Concept of sensors . . . 19
    - 3.2.4.1 Internal state sensors . . . 19
    - 3.2.4.2 Surrounding sensors . . . 20
  - 3.2.5 Limitations . . . 20

4 Related Documentation 21

5 Architecture Overview 22

6 Description of the Chassis Domain Software Compositions and Components 25
- 6.1 Software Composition CrsCtrlAndAcc (Cruise Control and Adaptive Cruise Control) . . . 25
  - 6.1.1 AccSnsrDataFusion (ACC Sensor Data Fusion) . . . 28
  - 6.1.2 AccObjTarSeln (ACC Object Target Selection) . . . 28
  - 6.1.3 CrsCtrlFree (Free Cruise Control) . . . 28
  - 6.1.4 FolwCtrlAndAArbn (Follow Control and Acceleration Arbitration) . . . 29
  - 6.1.5 CrsCtrlAndAccStCtrl (Cruise Control and ACC State Machine) . . . 29
  - 6.1.6 CrsCtrlAndAccObsvrForVehSt (Cruise Control and ACC Vehicle State Observer) . . . 29
- 6.2 Software Component Esc (Electronic Stability Control) . . . 30
- 6.3 Software Component Ssm (Stand Still Manager) . . . 31
- 6.4 Software Component Epb (Electronic Parking Brake) . . . 31
- 6.5 Software Component Vlc (Vehicle Longitudinal Control) . . . 33
- 6.6 Software Component Rsc (Roll Stability Control) . . . 33
- 6.7 Software Component Steer (Steering System) . . . 34

6.8 Software Component SteerDrvrAsscSys (Steering Driver Assistance System) . . . 35
6.9 Software Component SteerVehStabyCtrl (Steering Vehicle Stabilizing Control) . . . 35
6.9.1 Superposition steering angle actuation . . . 36
6.9.2 Superposition steering torque actuation . . . 37
6.10 Software Component Susp (Suspension System) . . . 37
6.11 Software Component TirePMon (Tire Pressure Monitoring) . . . 38
6.12 Software Component DtTqDistbn (Drivetrain Torque Distribution) . . . 39
6.12.1 Hang-on Coupling . . . 40
6.12.2 Active Differentials . . . 41
6.12.3 Torque Vectoring Device . . . 42
6.13 Software Component SurrndgsSnsr (Surroundings Sensor) . . . 43
6.14 Software Component ChassisSnsr (Chassis Sensor) . . . 43
6.15 Software Component PrkgAid (Parking Aid) . . . 44
7 Outlook 45
7.1 Chassis Domain Structure . . . 45
7.2 Scalability . . . 46
7.2.1 Use Case: ESC, Powertrain, and Tire Pressure Monitoring (TirePMon) 46
7.2.2 Use Case: Cruise Control . . . 47
7.2.3 Use Case: Steering DAS Superposition Steering Torque . . . 47

1 Purpose of this Document

This document explains all design decisions that lead to the Application Interface Table [1] contents relevant to the Chassis Domain.

2 Acronyms and abbreviations

Abbreviation:	Description:
ABS	Antilock Braking System
ACC	Adaptive Cruise Control
BAS	Brake Assist
BRWS	Basic Rear Wheel Steering
BSTS	Basic Steering Torque Superposition
BSAS	Basic Steering Angle Superposition
CBC	Cornering Brake Control
CoG	Center of Gravity
DAS	Driver Assistance System
DTC	Regulation of the Drag Torque
EBD	Electronic Brake Force Distribution
ECU	Electronic Control Unit
EPB	Electronic Parking Brake
ESC	Electronic Stability Control
FA	Front Axle
HDC	Hill Decent Control
HHC	Hill Hold Control
HMI	Human Machine Interface
NVH	Noise, Vibration, Harshness
RA	Rear Axle
RSC	Roll Stability Control
SR	Situation Recognition
SSM	Stand Still Manager/Management
TCS	Transaction Control System
VGR	Variable Gear Ratio
VLC	Vehicle Longitudinal Control
VM	Vehicle Model
YRC	Yaw Rate Control

Table 2.1: Abbreviations used in the scope of this Document

3 Description of terms and concepts

This document refers to the formulation of unified Application Inter-faces of the Domain Chassis. The goal is to define and publish functional catalogues of all unified interfaces and functions. This includes interfaces for passenger cars and trucks and also covers driver assistance systems (DAS). The results of the Domain Chassis shall be aligned with the other domains, e.g. Body, Powertrain, Occupant and Pedestrian Safety. The Application Interface Table represents a good basis for this alignment and detection of conflicts.

3.1 Concepts

This document refers to the formulation of unified Application Inter-faces of the Domain Chassis. The goal is to define and publish functional catalogues of all unified interfaces and functions. This includes interfaces for passenger cars and trucks and also covers driver assistance systems (DAS). The results of the Domain Chassis shall be aligned with the other domains, e.g. Body, Powertrain, Occupant and Pedestrian Safety. The Application Interface Table represents a good basis for this alignment and detection of conflicts.

3.1.1 Axis System

The standard coordinate system used by the Chassis Domain refers to the International Standard ISO 8855 [2].

Whether a fixed geometry point (e.g. near to the average Centre of Gravity (CoG) of all variants of a vehicle type) is used as a reference point for basic Centre of Gravity has to be decided on project level.

3.1.2 Definitions

3.1.2.1 Methodology for Optimization of Interface Definition in the Chassis Domain

3.1.2.1.1 Intended optimization

Harmonize the use of keywords in names & descriptions for:

- indication, state, request, mode, consolidate, target, confirmation, demand, command, information, ..
- and how to differentiate ’fault’, ’failure’, ’error’, ’ state’ and ’mode’

3.1.2.1.2 Overview of realization possibilities for "flow" and "state"

3.1.2.1.2.1 Control Flow versus Signal Flow

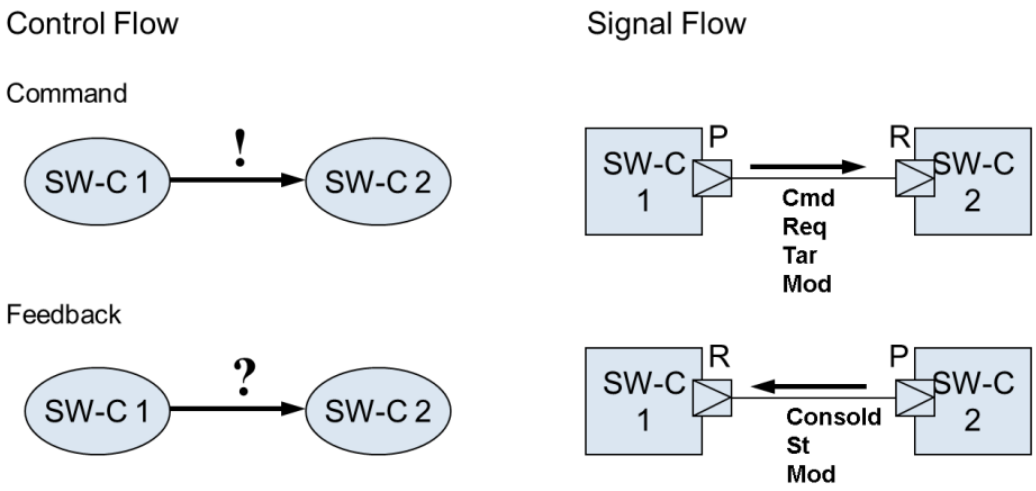


Figure 3.1: Overview of Control Flow v. Signal flow (Chassis Domain)

3.1.2.2 Definitions / Explanations

3.1.2.2.1 State

- The present status of a system or entity:
- State (physics) (list of 8 definitions), particularly in thermodynamics, statistical physics, and dynamical systems and chaos theory
- State (computer science), a unique configuration of information in a program or machine

3.1.2.2.2 Modus

- a "working mode", set by commutation via diagnosis, driver or other
  Example: suspension => Comfort or Sport Modes
- an adjustment of a procedure : can be sometimes considered as request
  Example: reset of TPMS function => it's more a request than a mode

3.1.2.2.3 Mode

- Kind, manner of s.th.

3.1.2.2.4 Indication

- A signal that informs external systems of a relevant state of the system providing the signal. In general, the final destination of the signal is expected to be the driver or other operator.

3.1.2.2.5 Target

- Set point, set value that an automatic control system, will aim to reach

3.1.2.2.6 Request

- Default wording for the target value of a controller

3.1.2.2.7 Meaning of "active":

- it is on => leave as "active"

3.1.2.2.8 Meaning of "intervention":

- active intervention at the moment => changed to "intervention"
  (to be used when a function can be on, but not intervening)
  Keyword: Intv

3.1.2.2.9 Fault:

- cause / reason
- ISO/WD 26262-1 [3]: A fault is the cause of an error; it can either be systematic or random. NOTE Do not use: defect, mistake.

3.1.2.2.10 Failure:

- means doesn't work, but is not necessarily the reason
- ISO/WD 26262-1 [3]: A failure is the inability (of a component or system) to perform a required function.

3.1.2.2.11 Error:

- temporary / permanent, same as failure

- ISO/WD 26262-1 [3]: An error is the discrepancy between a computed, observed or measured value or condition and the specified correct value.

Decision:

- Use failure or fault instead of error based on the definition above
- Differentiation from State? => Failure is a state!

3.1.2.3 Naming consistency rules

	Command	Feedback
Controller value, actuator value **Request:** default wording for the target value of a controller **Reference:** in the case of arbitration, the word shall be used to diffentiate from the word "request" **Target:** try to replace it with "reference" or "request", using request when we want to denote the direction	**Request** Axiliary examples: dynamic consign from other system or driver (e.g. BrakeAssistRequest) also for **Request**Mode_xy (e.g. sportiv) static switch, switchable power off, by diagnosis or complex HMI interface « **Mode** » is part of the request. **Keyword: Req**	"….." No additional keyword (like: consolidated, present, ..) (e.g. "velocity" is always the current value.) But add to the description, that this is the consolidated value of the signal. In special cases target value and current value should be used (e.g. Tpms)
Information, Flag	**Request / Indication** see also for descriptions (e.g. Confirmation of …) **Indication** : only for HMI or Lamps **Keyword: Req/Indcn**	"….." / State / Mode **State** : as active, failure, in progress, in(active)…., e.g.: State: Control Active => Current information about system **Mode**: Suspension « comfort, sport », ⇒Choice, Imposition, Tuning **Keyword: St/Mod**

Figure 3.2: Overview of naming consistency rules (domain chassis)

3.1.2.3.1 Rules to optimize descriptions (for ports, PortInterfaces & Data Elements, -Types)

- Start with a Noun (avoid words "information, indicates, ...")
- It must be consistent with the LongName (e.g. if "Mode" is in the LongName it should also be in the description)
- Neither sending nor receiving components should be named (like SSM, ESC). The purpose can be named (e.g. for display to ..)
- Positive example taken from AI Table:
  LongName: Steering Control Tuning Request
  Description: Driver's requested steering controller tune

As opposed to (bad example): Indicates HMI's request for controller tune to "Steer"- SWC.

3.1.3 Centre of gravity of passenger car

The car's chassis-fixed coordinate system's origin lies in the middle of the Front Axle (FA) see Figure 3.3. The car's centre of gravity is defined as the origin of a chassis-free coordinate-system that is positioned relative to the FA co-ordinate system. The orientations of both these co-ordinate systems' axes are defined as follows:

With the car moving forward:

- x is positive to the front of the car.
- y is positive to the left of the car.
- z is positive to the top of the car.
- Roll is positive in a left-hand bend, when the car body tips to the right-hand side.
- Pitch is positive when a car is braking and the car body tips forward.
- Yaw is positive in a left-hand bend.

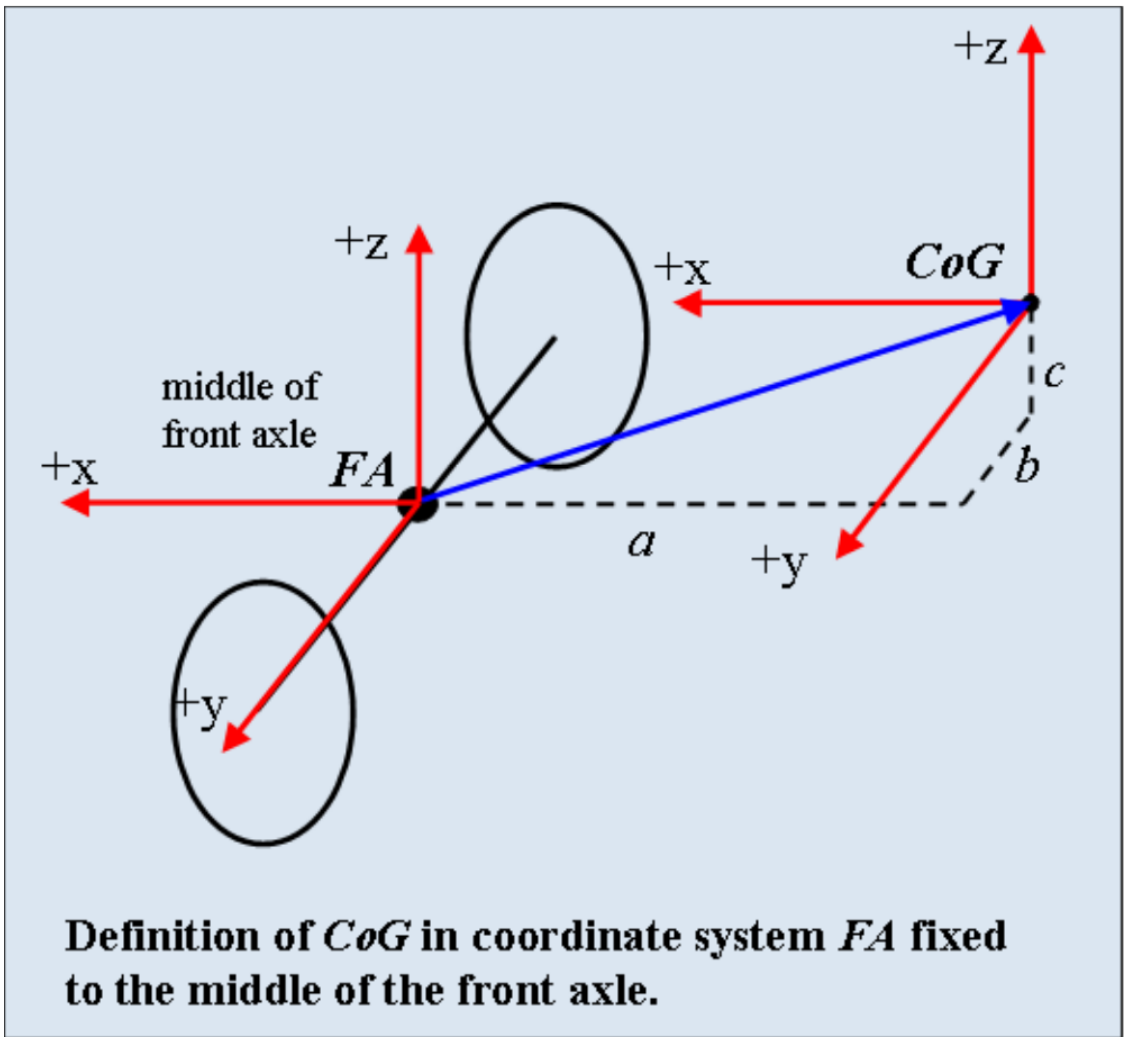


Figure 3.3: Definition of a car's Center of Gravity (CoG)

3.1.4 Definition of the car surroundings (e.g. Used for ACC)

The car's surrounding for ACC and predicted course are defined as Figure 3.4 and Figure 3.5.

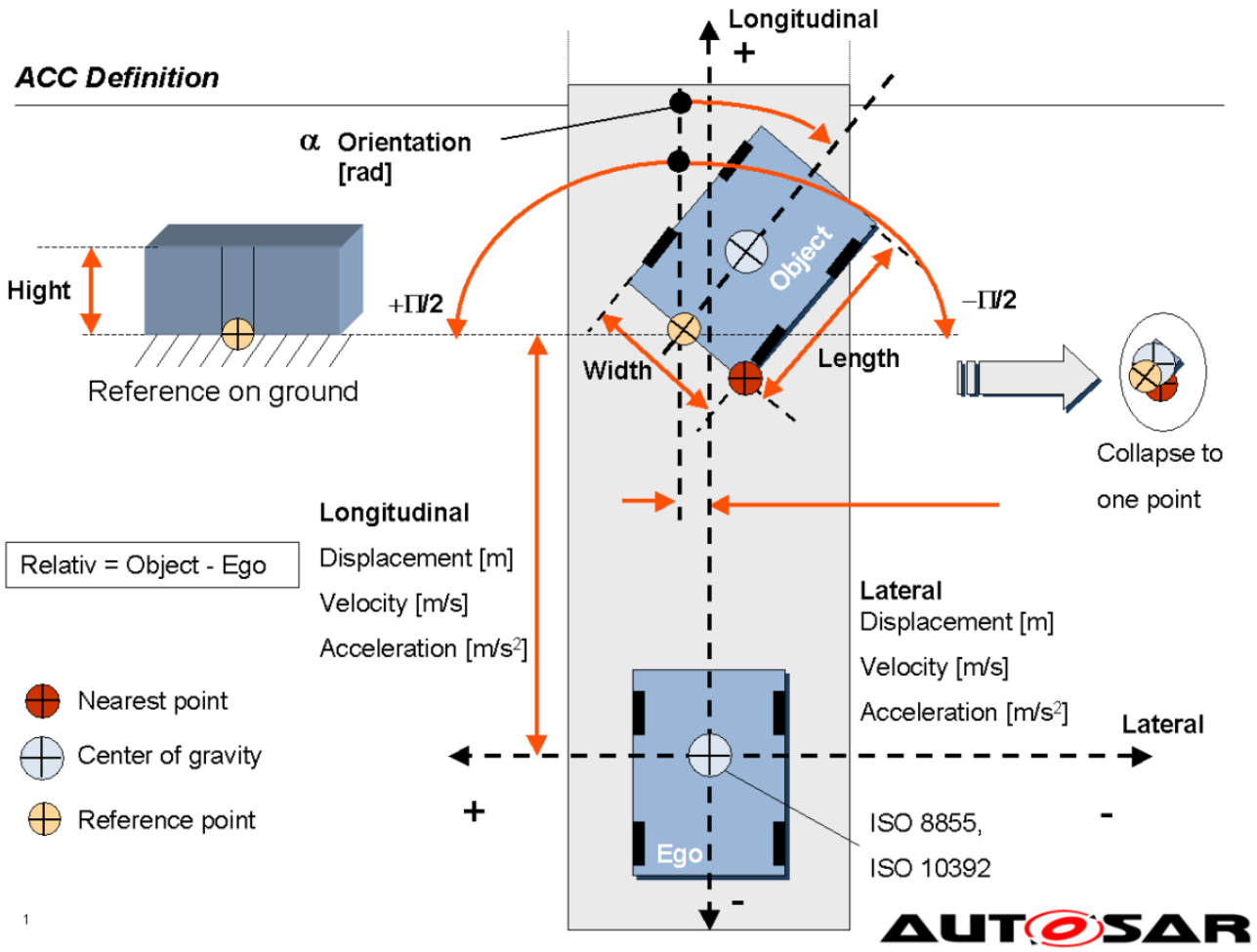


Figure 3.4: Definition of car surrounding for ACC

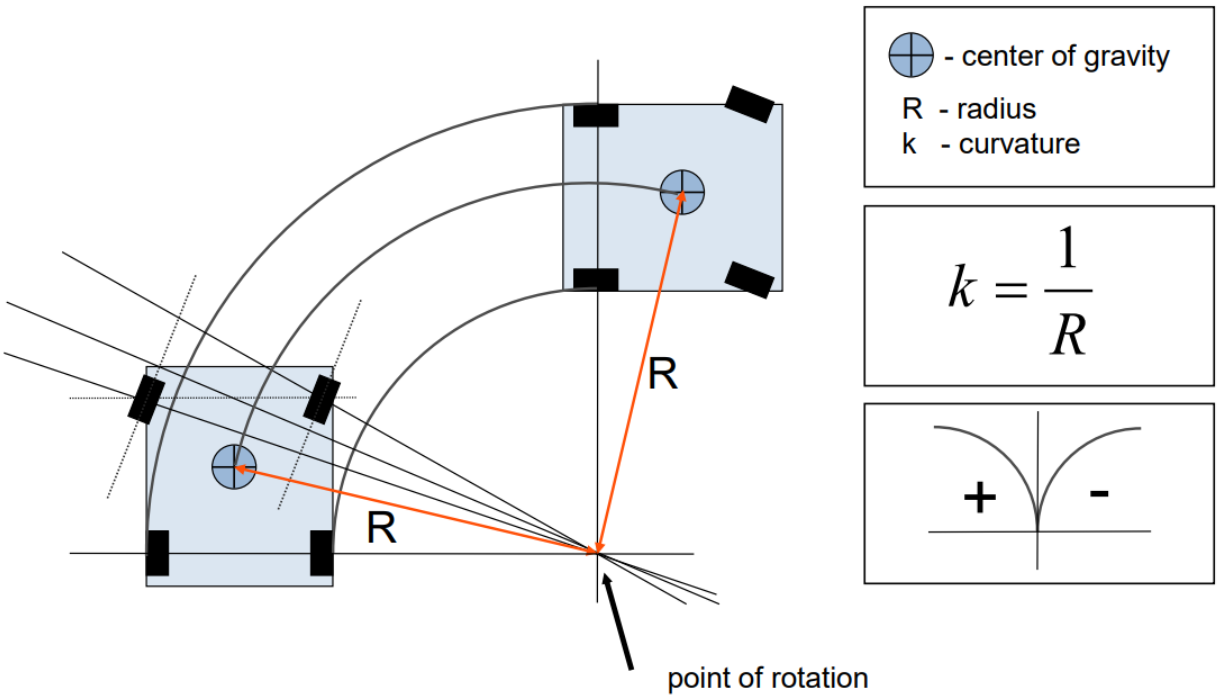


Figure 3.5: Definition of predicted course

The Object list for ACC is defined in the following table.

Record Type Name	Long Name	Description	Name	Comment
ObjData1	ObjectData	Cluster of object attributes detected by surroundings sensors. This information is used e.g. for Adaptive Cruise Control (ACC), but is not limited to this use case only	DplLgt	<Core> Distance between the sensor and the object.
			DplLat	<Core> Lateral distance between the center of the sensor and the object.
			SpdLgtRel	<Core> Longitudinal speed difference between the ego vehicle and the object.

△

			DynPpty	<Core> Status which indicates the dynamic property of the object (e.g. stationary,moving, etc). Standing: has never been detected moving. Stopped: has been detected moving, but is stationary now. Moving: object has absolute speed in any direction. The definition is made according to the earth-fixed axis system as defined in ISO 8855 [2].
			StOfObjMeasd	<Core> Status which indicates whether the object was measured in current cycle, or only tracked.
			ObjId	<Core> Unique number for each object which has been tracked. The object keeps the same ID as long as it is output in the list. When an object dies, the ID can be reused for a new object. When a new object appears, for one cycle the attribute "object is new" is set. ID=0 is reserved for "no object".
			SpdLatRel	<Opt> Lateral speed difference between the ego vehicle and the object.
			ALgtRel	<Opt> Longitudinal acceleration difference between the ego vehicle and the object.
			ALatRel	<Opt> Lateral acceleration difference between the ego vehicle and the object.
			Width	<Opt> Lateral length of the object.
			Hei	<Opt> Vertical length of the object from the ground.
			Len	<Opt> Longitudinal length of the object.
			Orntn	<Opt> Yaw angle of the object. Coordinate system according to ISO 8855 [2].
			ObjClassn	<Opt> Class which indicates vehicle types (e.g. truck, passenger car, etc.)

▽

△

			DataQly	<Core> Quality of object data. This value decreases if the quality is poor (e.g. because of weather condition, object reflectivity). Scaling will be dependent on type of sensor and signal processing algorithm, therefore requires application of parameters on receiving function side.
ObjDataRlvForFol-wCtrl1	ObjDataRelevant-ForFollowControl	Cluster of relevant object attributes for Follow Control.	DplLgt	<Core> Distance between front edge of the ego vehicle and the object.
			SpdLgtRel	<Core> Longitudinal speed difference between the ego vehicle and the object.
			DynPpty	<Core> Status which indicates the dynamic property of the object (e.g. stationary, moving, etc). Standing: has never been detected moving. Stopped: has been detected moving, but is stationary now. Moving: object has absolute speed in any direction. The definition is made according to the earth-fixed axis system as defined in ISO 8855 [2].
			ObjId	<Core> Unique number for each object which has been tracked. The object keeps the same ID as long as it is output in the list. When an object dies, the ID can be reused for a new object. When a new object appears, for one cycle the attribute "object is new" is set. ID=0 is reserved for "no object".
			DplLat	<Opt> Lateral distance between the center of the ego vehicle and the object.
			SpdLatRel	<Opt> Lateral speed difference between the ego vehicle and the object.
			ALgtRel	<Opt> Longitudinal acceleration difference between the ego vehicle and the object.
			StOfObjMeasd	<Opt> Status which indicates whether the object was measured in current cycle, or only tracked.

▽

△

			ObstrcnProblty	<Opt> Probability that obstacle can not be overrun, or can not be driven over it.
			LaneProblty	<Opt> Probability of the potential target object to be located in the driving corridor, or predicted vehicle course.
			CutInProblty	<Opt> Probability of the potential target object for cutting in the driving corridor, or predicted vehicle course.
			ObjRlvc	<Opt> Measure of the plausibility as target object for Adaptive Cruise Control (ACC), which means that this measure increases if LaneProbability keeps high.
			ObjRlvcUpprLim	<Opt> Upper limit of ObjectRelevance. Above this limit the object becomes target object for Adaptive Cruise Control (ACC).
			ObjRlvcLowrLim	<Opt> Lower limit of ObjectRelevance. Below this limit the object is not the target object for Adaptive Cruise Control (ACC).
			DstMaxForFol-wCtrl	<Opt> Maximum distance for the relevant object to become the target object for Follow Control.

Table 3.1: Table Object list for ACC

<Core> Longitudinal speed difference between the ego vehicle and the object.

3.2 General remarks

The definitions of SW-Composition and SW-Component are given in [4].

3.2.1 Differences between SW-Components and ECUs

The SW-Components defined below are not to be misunderstood with ECU functionalities.

For example, an ESC ECU may contain the Esc SW-Component and other components like Ssm SW-Component, SteervehStabyCtrl SW-Component, etc.

3.2.2 Functional safety

- Most of the chassis domain signals are considered as safety relevant. It is assumed that in AUTOSAR, reliable methods of communication are available. No specification of reliable communication is done in the Chassis Domain.
- Note that these values are meant as minimum requirements, i.e. optimization within the communication layer is possible.
- Diagnostics, timing, and safety concepts haven't been considered yet in the Chassis Domain. In order to prove that the discussed use cases are safety compliant (in accordance to the definition of the upcoming ISO 26262), a joint discussion across all application domains will be necessary in the future.
- There is no safety concept provided for active chassis systems. This has to be done on project level. This means that the specified interfaces have to be checked to fulfil the safety requirements on each specific project.

3.2.3 Concept of core-, conditional- and optional- ports

Within the Chassis Domain, port attributes "core", "cond" and "opt" have been defined, which indicate whether a provider or receiver port is seen today to be more a core or optional interface for the SW-C or its usage is depending on additional conditions. As all interfaces can be used in different configurations and there is no mandatory usage of single interfaces (see also chapter 5), this is for information purpose only. The attribute is given in the EXCEL version of the Application Interface Table [1]. This attribute is not used from all Application SW domains and will not appear in the ARXML-File of the Application Interfaces. The values of the port attribute are:

- Core provider and receiver ports
- Conditional provider and receiver ports
- Optional provider and receiver ports

This definition is not consistent with the document AUTOSAR_Powertrain_AI_Explanation. Information concerning Core/Cond/Optional attributes in the AUTOSAR_CP_MOD_AISpecification [1] shall be ignored.

3.2.4 Concept of sensors

In the Chassis Domain, a concept of different signal processing levels has been defined for the chassis Application Interface definitions.

Remark1: Chassis Domain used only sender receiver interfaces for sensors.

Remark2: Complex device drivers can also be used to transmit sensor signals.

3.2.4.1 Internal state sensors

- Raw signal
  Raw sensor data w/o preprocessing. Not to be used as an interface since these raw signals are hardware dependent. For example YawRate and PitchRate.
- Base signal
  Pre-processed sensor data. Model-free. Abstraction of sensor hardware. For example YawRate and PitchRate.
- Standard signal
  Derived from Base signal(s) according to open AUTOSAR specification (Filtering, offset compensation, model-based plausibility check, etc.). The SW-Compositions should provide standard signals from their sensor information to other SW- Compositions in order to avoid duplication of signal processing algorithms. Please refer to the Application Interface Table [1] for definition of standard signals. For example YawRate and PitchRate.
- Fused signal
  Several sensor signals are processed to derive the fused signal. For example vehicle speed or body side slip.

3.2.4.2 Surrounding sensors

Only fused signals are standardized. This value is related to the physical output of any used sensors. For example Object list.

3.2.5 Limitations

There are some fundamental limitations. These limitations can be listed as follows:

- The objective of the Application Interface Table [1] is not to make any software architecture proposal. Any architecture shown on any of the AUTOSAR documents are only used for clarification and have to be treated as "not binding".
- Interfaces are defined on vehicle related physical values like torques or forces, and not on actuator specific interfaces like current or PWM (Pulse Width Modulation) duty cycle.

4 Related Documentation

[1] XML Specification of Application Interfaces
AUTOSAR_CP_MOD_AISpecification

[2] ISO 8855:2011, Road vehicles – Vehicle dynamics and road-holding ability – Vocabulary
https://www.iso.org

[3] ISO 26262:2018 Road vehicles -– Functional Safety
https://www.iso.org

[4] Software Component Template
AUTOSAR_CP_TPS_SoftwareComponentTemplate

[5] Virtual Functional Bus
AUTOSAR_CP_TR_VFB

5 Architecture Overview

In the following, a rough explanation of the Chassis Domain is given in the context of the other AUTOSAR Application Domains - Body Electronics, Powertrain, Occupant and Pedestrian Safety and Multimedia, Telematics and HMI.

The task of Chassis Domain is focused on data description of functional domain Chassis within the AUTOSAR Application Layer.

The SW-Components of the Chassis Domain are shown. These SW-Components may communicate between each other or with other domains, like Body Electronics, Powertrain, Occupant and Pedestrian Safety, HMI, or the basic software. In order to define the Application Interfaces independently of their physical implementation, the Virtual Function Bus (VFB) serves as an abstraction of the AUTOSAR Software Components' interconnections, see [5].

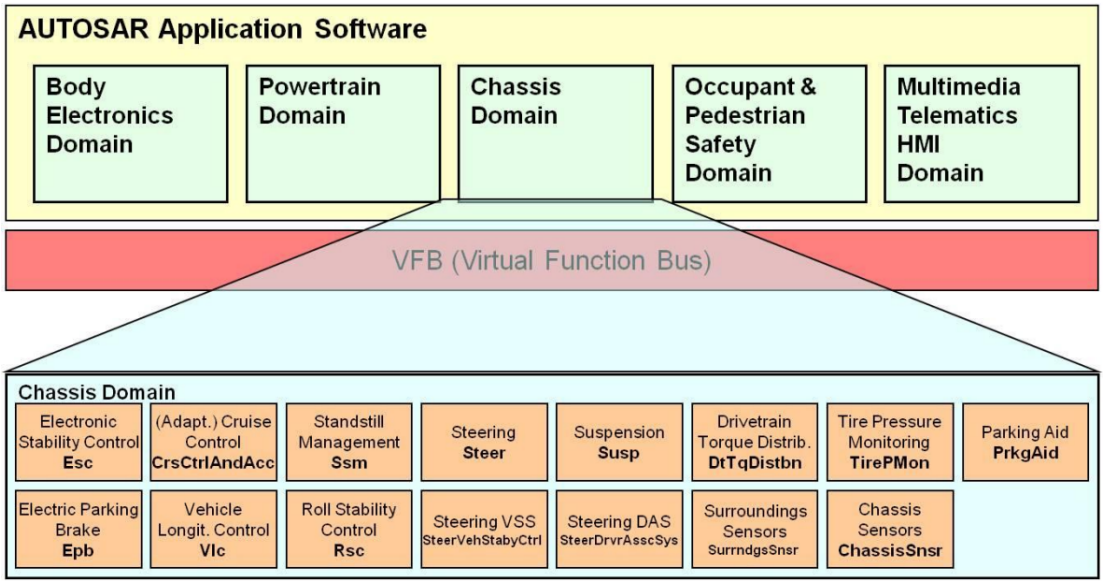


Figure 5.1: Chassis Domain overview

An example of a Chassis Domain structure built from these SW-Components is given in Figure 5.1. Based on this structure, the Application Interfaces for single SW-Components have been defined. This structure is an example among many other possible structures. Therefore, this structure is intended to be scalable and different configurations can be derived from this structure. All SW- Components may have their own control- and application specific vehicle models and situation recognitions. Neither all SW-Components have to be present at the same time, nor all interconnections between SW-Components have to be used in the given manner.

The implemented structure has to be defined on project level. In the future, more complex and powerful structures may be used, see Chapter 7.

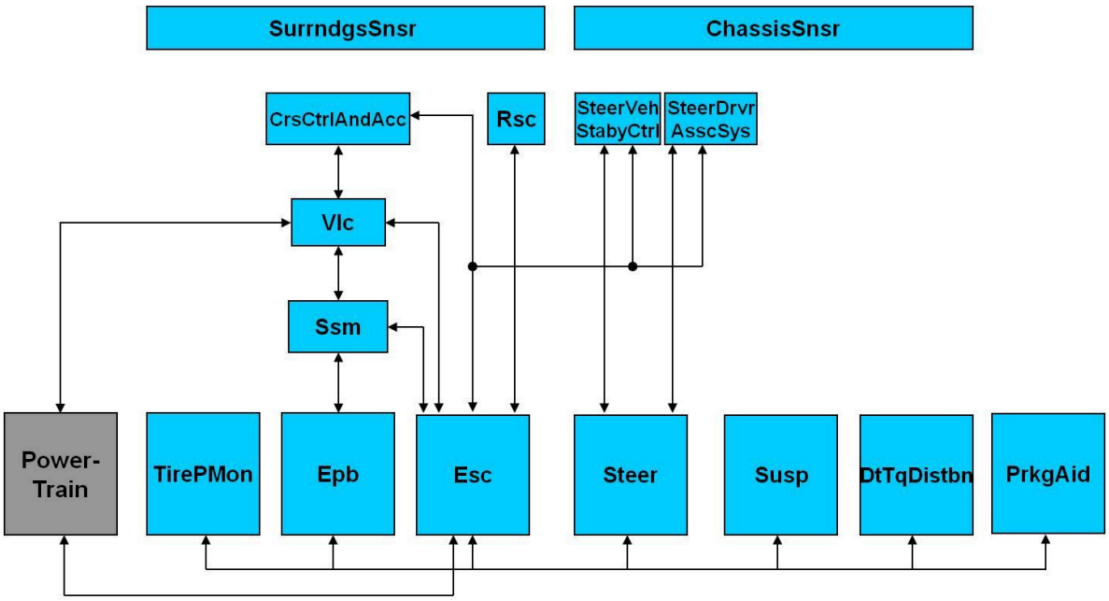


Figure 5.2: Example of a Chassis Domain Structure

In Figure 5.2 the inter domain dependencies between the Chassis Domain and the other Application Interfaces are depicted.

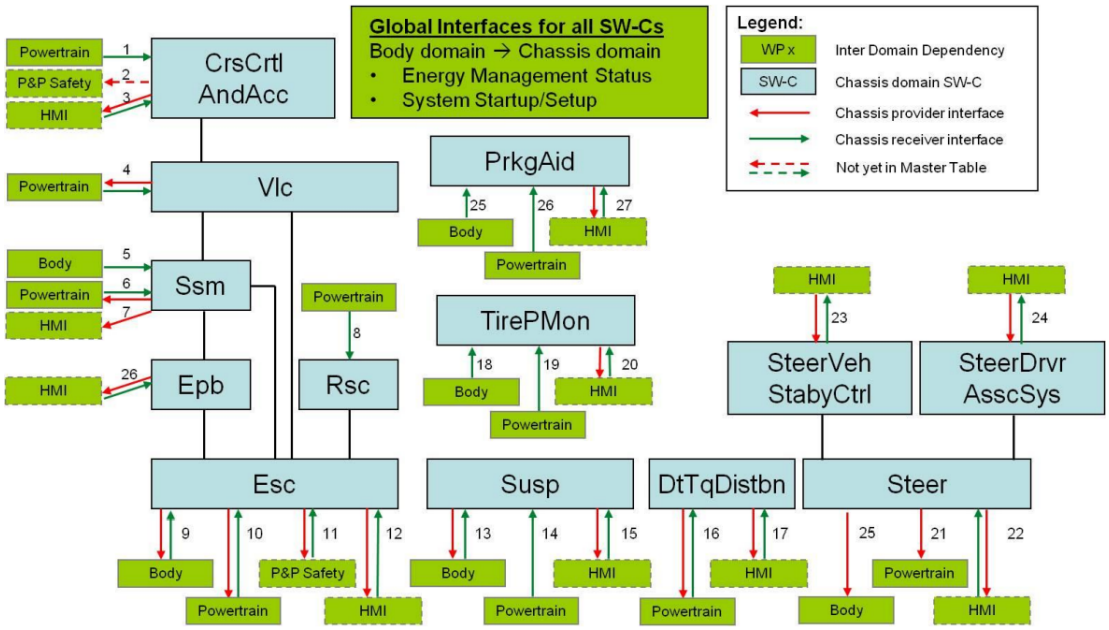


Figure 5.3: Chassis Domain overview (Inter domain dependencies)

The dependencies from each single SW-Component to the other Application Interfaces are listed below and refer to the numbered connections in Figure 5.2.

1. Basic driveline settings (CrsCtrlAndAcc)
2. Crash prediction and environment information (CrsCtrlAndAcc)
3. Display and controls (CrsCtrlAndAcc)

4. Torque management (Vlc)
5. Door, window and trailer status (Ssm)
6. Basic driveline settings and park request (Ssm)
7. Parking brake display (Ssm)
8. Basic driveline settings and torque information (Rsc)
9. Brake, driving and trailer status (Esc)
10. Wheel based torque control (Esc)
11. Driving dynamics (e.g. velocities) (Esc)
12. Display and controls (Esc)
13. Door, window and trailer status (Susp)
14. Basic driveline settings and torque information (Susp)
15. Display and controls (Susp)
16. Torque management (DtTqDistbn)
17. Display and controls (DtTqDistbn)
18. Environment temperature (TirePMon)
19. Torque information (TirePMon)
20. Display and controls (TirePMon)
21. Power steering load (Steer)
22. Display and controls (Steer)
23. Display and controls (SteerVehStabyCtrl)
24. Display and controls (SteerDrvrAsscSys)
25. Road wheel angle front (Steer)
26. Activation Button & Feedback (Epb)

6 Description of the Chassis Domain Software Compositions and Components

The Chassis Domain contains the following SW-Compositions and Components:

SW-Composition	Component
CrsCtrlAndAcc	Cruise Control and Adaptive Cruise Control
Esc	Electronic Stability Control (formerly known as 'ESP')
Ssm	Stand-still Manager
Epb	Electronic Parking Brake
Vlc	Vehicle Longitudinal Control
Rsc	Roll-Stability Control
Steer	Steering System
SteerDrvrAsscSys	Steering Driver Assistance System (DAS)
SteerVehStabyCtrl	Steering Vehicle Stabilizing Control
Susp	Suspension System
TirePMon	Tire Pressure Monitoring System
DtTqDistbn	Drivetrain Torque Distribution (formerly known as 'AWD')
SurrndgsSnsr	Surroundings Sensor
ChassisSnsr	Chassis Sensor
PrkgAid	Parking Aid

Table 6.1: SW-Composistions and their Components

6.1 Software Composition CrsCtrlAndAcc (Cruise Control and Adaptive Cruise Control)

The SW-Composition CrsCtrlAndAcc controls the vehicle velocity and distance to vehicles or other obstacles in front of the vehicle down to and from a reference speed until standstill. The used sensor signals can for instance be derived out of a sensor (e.g. radar, lidar or camera). The output of CrsCtrlAndAcc SW-Composition is acceleration command values to Vlc SW-C (ACC definition as in ISO 15622). Stand-alone free cruise control (without follow control) is also considered in the CrsCtrlAndAcc SW-Composition for the current AUTOSAR release. It is located in CrsCtrlAndAcc SW-Composition in the Chassis Domain rather than within the Powertrain Domain.

In the current AUTOSAR release, the essential SW-Component which provides information relevant to preceding traffic participants to SW-Composition CrsCtrlAndAcc is the SW-C SurrndgsSnsr, see section 6.13.

CrsCtrlAndAcc SW-Composition is decomposed into SW-Components:

- AccSnsrDataFusion (ACC Sensor Data Fusion)
- AccObjTarSeln (ACC Target Object Selection)
- CrsCtrlFree (Free Cruise Control)

- FolwCtrlAndAArbn (Follow Control and Acceleration Arbitration)
- CrsCtrlAndAccStCtrl (Cruise Control and ACC State Machine)
- CrsCtrlAndAccObsvrForVehSt (Cruise Control and ACC Vehicle State Observer)

This decomposition, shown in Figure 6.1, enables the realization of variable business cases between an OEM and a supplier with different border of responsibility on the functionalities inside the CrsCtrlAndAcc SW-Composition.

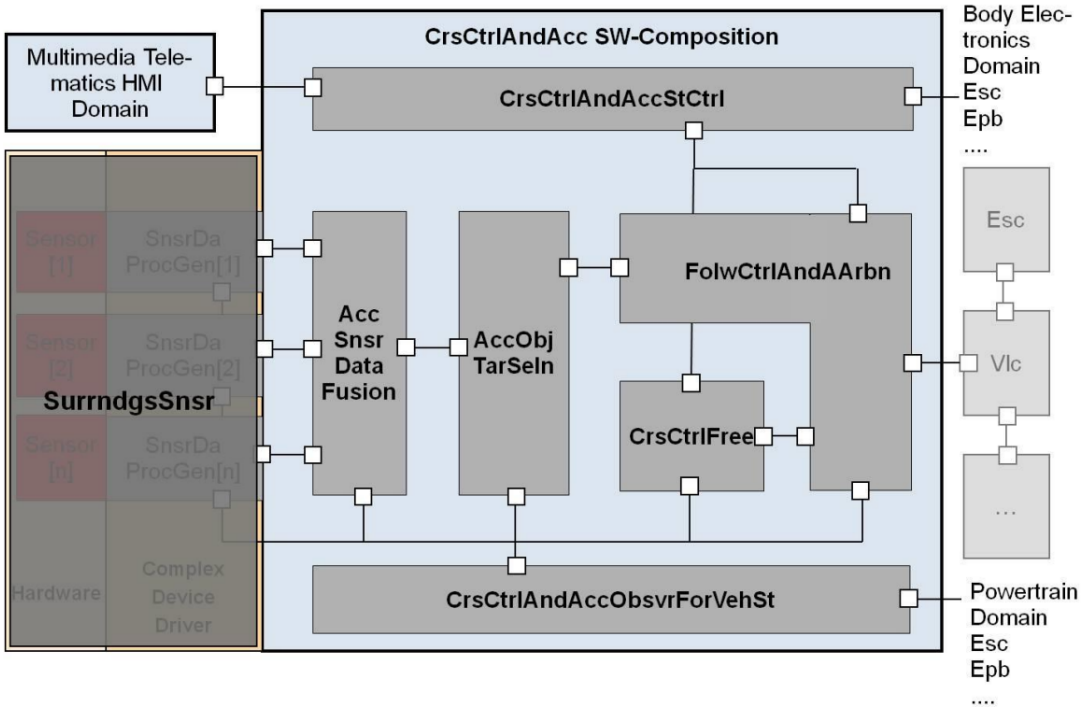


Figure 6.1: SW-Composition CrsCtrlAndAcc

Please note that the number and type of surroundings sensors used for this SW-Composition should be determined at the project level. Additionally, the interface between the SW-C SurrndgsSnsr and the SW-Composition CrsCtrlAndAcc should be defined at project level too. In the current release, an interface on object level has been defined, but there are also other interfaces possible, e.g. on raw signal level. In the current state of the art, only one surroundings sensor is used typically. The SW-Composition in this case might not involve the SW-C AccSnsrDataFusion because no sensor data fusion is necessary, see Figure 6.2:

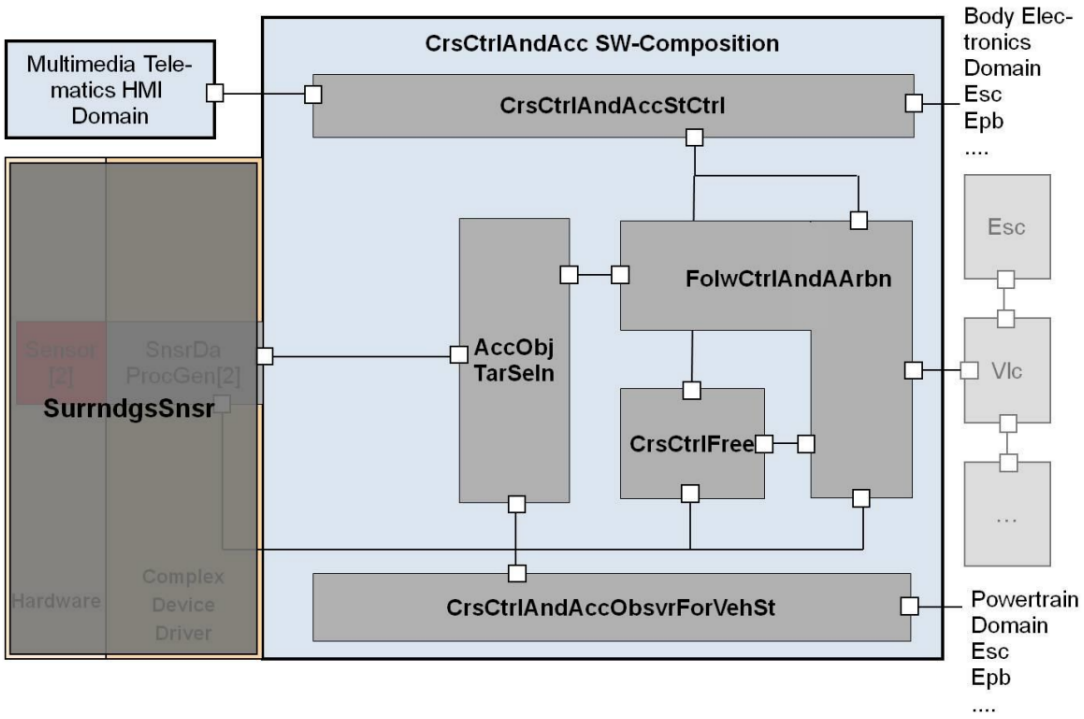


Figure 6.2: Example of SW-Composition CrsCtrlAndAcc in case of one SurrndgsSnsr

In addition, this decomposition is sufficiently flexible to permit the definition of a stand-alone free cruise control feature without the ACC feature, which is more common for the time being, as shown in Figure 6.3. In the case of the stand-alone free cruise control feature only, the CrsCtrlAndAcc SW- Composition consists of the SW-Components:

- CrsCtrlFree (Free Cruise Control)
- CrsCtrlAndAccStCtrl (Cruise Control and ACC State Machine)
- CrsCtrlAndAccObsvrForVehSt (Cruise Control and ACC Vehicle State Observer)

Here, if the signal conditioning conducted by the SW-C CrsCtrlAndAccObsvrForVehSt is not required, it may be removed from a project's SW-Composition.

Therefore, the CrsCtrlAndAcc Application Interface is designed in order to be able to handle these several cases. Note that Figure 6.1, Figure 6.2 and Figure 6.3 are examples of possible SW-Compositions.

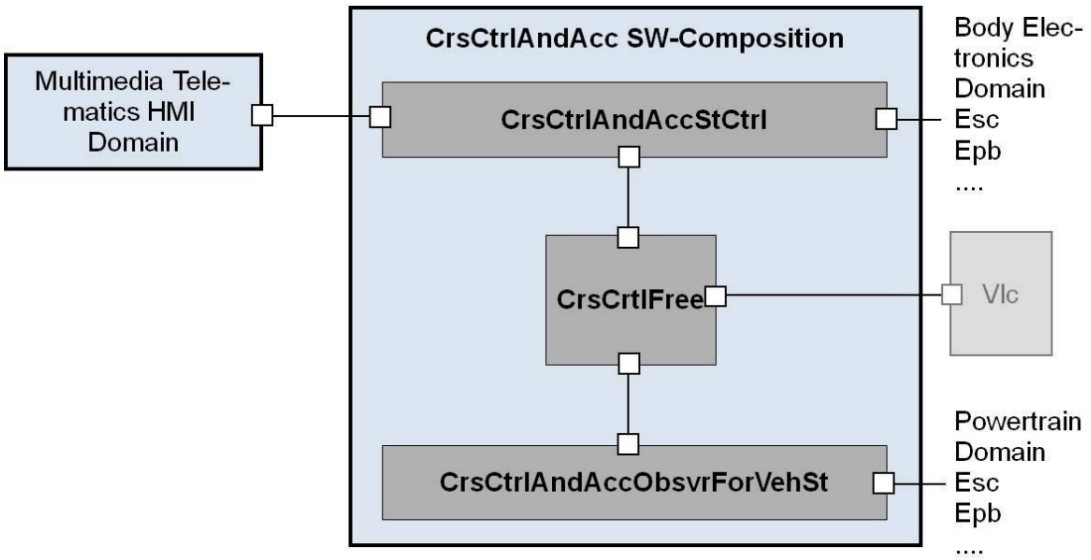


Figure 6.3: Example of SW-Composition CrsCtrlAndAcc in case of stand-alone free cruise control

6.1.1 AccSnsrDataFusion (ACC Sensor Data Fusion)

This SW-Component has a task to carry out multiple sensor fusion which consolidates all the input information from the SW-C SurrndgsSnsr if several surroundings sensors are involved.

6.1.2 AccObjTarSeln (ACC Object Target Selection)

The SW-C AccObjTarSeln has the role to select ACC relevant target object(s) and their attributes, which are necessary to control vehicle velocity and the distance to the vehicle in front of the sensing vehicle. Note that ACC relevant target can be several objects. The tasks of this SW-C consist of, for example:

- Determination of ACC relevant attributes (e.g. Lane probability)
- Determination of ACC relevant object(s)

6.1.3 CrsCtrlFree (Free Cruise Control)

The SW-C CrsCtrlFree has the role to calculate the target control value of vehicle acceleration according to the set speed provided by the HMI. The task of this SW-C is, e.g.:

- Determination of target acceleration for free cruise control

In the case of the ACC feature, the output of this SW-Component goes to the SW-C FolwCtrlAndAArbn, and is arbitrated. In the case of the stand-alone free cruise

control feature, the output of this component directly goes to the SW-C Vlc (Vehicle Longitudinal Control).

6.1.4 FolwCtrlAndAArbn (Follow Control and Acceleration Arbitration)

The SW-C FolwCtrlAndAArbn has the role to calculate target control value of vehicle acceleration according to the behaviour of ACC relevant object(s) in front of the sensing vehicle. The other role is to arbitrate the target acceleration values, which are provided from the following control functionality and from the free cruise control functionality. The tasks to be done in this SW-Component are, for example:

- Determination of target acceleration for follow control and standstill
- Arbitration of the follow control and free cruise control

Note that arbitration between control and driver intervention is handled by the SW-C Vlc.

6.1.5 CrsCtrlAndAccStCtrl (Cruise Control and ACC State Machine)

The SW-C CrsCtrlAndAccStCtrl has the role to determine the system control mode, which includes system activation status and HMI related information. Since the functionality of this SW-Component may depend on each OEM, AUTOSAR does not standardize its functionality and interfaces beyond the ones defined in the ISO standard 15622. The functionality includes diagnostics or failure detection of the system. The tasks of this SW-C consist of, for example:

- Interpretation of driver input through instruments
- Determination of state machine
- Diagnostics

6.1.6 CrsCtrlAndAccObsvrForVehSt (Cruise Control and ACC Vehicle State Observer)

The SW-C CrsCtrlAndAccObsvrForVehSt has a role to provide information relevant to the own vehicle behaviour. The tasks of this SW-Component consist of, e.g.:

- Prediction of vehicle course based on yaw rate signal
- Calculation of ACC specific vehicle dynamics data (e.g. yaw rate, vehicle longitudinal speed, and so on)

Note that the predicted vehicle course may be corrected by the SW-C SurrndgsSnsr or the SW-C AccSnsrDataFusion in order to achieve better accuracy with object information.

6.2 Software Component Esc (Electronic Stability Control)

The SW-C Esc controls the stability of vehicle motion on the ground plane, by controlling the wheels' longitudinal slip, vehicle yaw rate and vehicle side slip. The used sensor signals among others can for instance be derived out of a sensor cluster, delivering for example yaw rate and lateral acceleration.

In the current AUTOSAR release the SW-C Esc controls applied wheel individual braking forces and/or the total powertrain torque.

The SW-C Esc is comprised of at least the following sub-functions, as also shown in:

- Controlling the longitudinal slip
  - ABS Antilock Braking System
  - TCS Traction Control System
  - EBD Electronic Brake Force Distribution
  - DTC Regulation of the Drag Torque
- YRC Yaw Rate Control
- VM Vehicle Model
- SR Situation Recognition
- Actuator related driver assistance functions (due to performance, NVH reasons) e.g.
  - BAS Brake Assist
  - HDC Hill Decent Control
  - HHC Hill Hold Control
  - CBC Cornering Brake Control
  - ...
- Arbitration of the ESC-internal controllers and inputs to the actuator SW
- Actuator SW (HW driver)

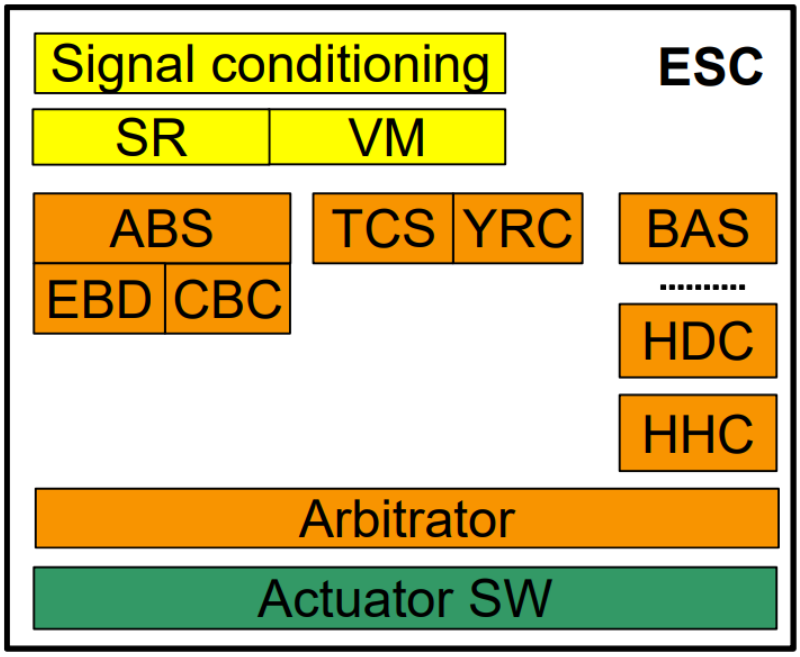


Figure 6.4: SW-C Esc content

6.3 Software Component Ssm (Stand Still Manager)

The SW-Component Ssm manages stand still and supports driver during drive off. It also keeps the vehicle parked (long term, e.g. without a driver in the car, ignition off) and decelerates vehicle under driver request. To achieve this, it relies on the Esc SW-C (service brake) and the Epb SW-C (Parking Brake) by sending them braking requests.

The motivation to create this Ssm SW-C comes from the fact that on vehicles, these "SSM functions" may be included either in the ESC ECU or the EPB ECU.

6.4 Software Component Epb (Electronic Parking Brake)

The SW-C Epb controls the parking brake actuator. Its functions are:

- Realizing the requests from Ssm by controlling the EPB actuator.
- Interpretation of the driver's wish for the Chassis Domain when pushing the EPB button.

Current market implementations of the SW-C Epb are comprised of 3 main versions:

EPB stand-alone: EPB does not work in conjunction with ESC; minimal number of interfaces are used to avoid conflicts.

EPB master-ESC slave: the EPB unit controls the ESC as an actuator.

ESC master-EPB slave: the ESC controls the EPB actuator; some backup modes are implemented at EPB Actuator unit in case of ESC fault.

The AUTOSAR definition of EPB/ESC interfaces supports all 3 variants.

Table 6.2 shows an overview of features that can be realized by typical EPB configurations. All conclusions derived from the following table that are inconsistent with the Application Interface Table [1] content should not be considered. This table is just presented here as an illustration.

Feature	**Feature Description**	**Configuration**		
		I	II	III
		EPB/ESC Cooperation	**EPB stand-alone** (or Config I w/ brake actuator failure)	**Config I w/ brake ECU or bus failure**
EPB Driver Request Interpretation	Switch processing and generation of the driver command (apply/release)	SSM (opt. VLC for Dynamic Apply)	SSM	EPB
Active Vehicle Deceleration	Arbitration and control of deceleration requests from driver assistance systems using hydraulic brake and powertrain	ACC -> VLC -> ESC/PTC	n/a	n/a
EPB Dynamic Apply	Vehicle deceleration using hydraulic or mechanical brake in case EPB switch is applied while driving	EPB -> SSM opt. VLC -> ESC	EPB -> SSM -> EPB	EPB
EPB Static Apply	Hold/park the vehicle using hydraulic or mechanical brake in case EPB switch is applied while vehicle is in standstill	EPB -> SSM -> ESC/EPB	EPB -> SSM -> EPB	EPB
ACC Stop & Go	Hold/park the vehicle using hydraulic or mechanical brake on request from ACC	ACC -> VLC -> SSM -> ESC/EPB	n/a	n/a
EPB Drive-away Release	Release hydraulic and/or mechanical brake for drive-away	SSM -> ESC/EPB	SSM -> EPB	n/a
EPB Auto Apply	Automatic application without driver action on EPB control (e.g. vehicle at stand still and engine or ignition off)	SSM -> ESC/EPB	SSM -> EPB	n/a
Rolling / Skidding detection / prevention	System behaviour in case vehicle starts to move unintendly	SSM -> ESC/EPB	SSM -> EPB	Possibly EPB
EPB Anti-Lock braking	Prevention of wheel locking due to mechanical brake apply while driving	n/a	EPB	n/a
Mechanical Actuator Control	Force control considering all actuator-specific features, e.g. re-clamping due to cooling down	SSM -> EPB or EPB only	SSM -> EPB or EPB only	Possibly EPB

▽

△

Hydraulic Actuator Control	Brake torque control considering all actuator-specific features, e.g. compensation for valve leakages	ESC	n/a	n/a

Table 6.2: List of features of different EPB configurations

6.5 Software Component Vlc (Vehicle Longitudinal Control)

In the current release, the SW-C Vlc gets the longitudinal acceleration command from the SW-C CrsCtrlAndAcc and distributes torque-commands to the powertrain and/or brake-system to achieve this longitudinal acceleration command.

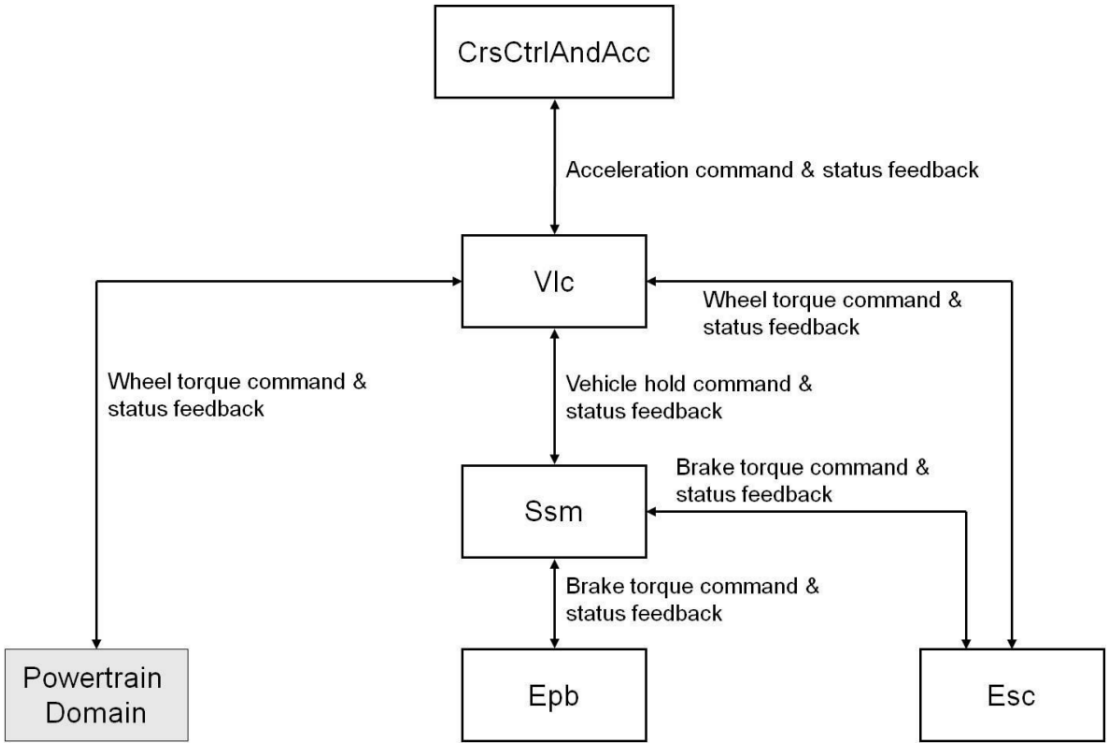


Figure 6.5: Overview of the SW-C Vlc

6.6 Software Component Rsc (Roll Stability Control)

The goal of the Rsc SW-C is to prevent rollover of the vehicle. For that, it controls the vehicle roll motion, i.e. control of body roll angle, by reducing the lateral forces at the tyres' contact patches. The reduction of these forces can be realized by requesting brake torques from the Esc SW-C and/or by requesting a powertrain torque from the Powertrain Domain via the Esc SW-C.

6.7 Software Component Steer (Steering System)

The three main functions of the Steer SW-C (each one of them to be achieved by a specific actuator) are:

- Basic Steering Torque Superposition (BSTS): It provides steer torque to support the driver in steering manoeuvres. This is the assistance torque, i.e. power steer assist or torque overlay function.
- Basic Steering Angle Superposition (BSAS): It provides additional front steer angle to the driver's steering wheel angle to support the driver in steering manoeuvres. This comprises VGR function (Variable Gear Ratio) and "Lead Lag" compensation functions.
- Basic Rear Wheel Steering (BRWS): It generates rear road wheel steer angles to support the driver in steering manoeuvres.

The conversion between pinion angle and the road wheel angle is part of this Steer SW-C. This road wheel angle is provided to other SW-Cs, e.g. Esc SW- C. The Steer SW-C receives requests (either angle or torque requests) from high level steering functions described below. The Steer SW-C arbitrates and dispatches the requests to the different steering actuators, as shown in Figure 6.6.

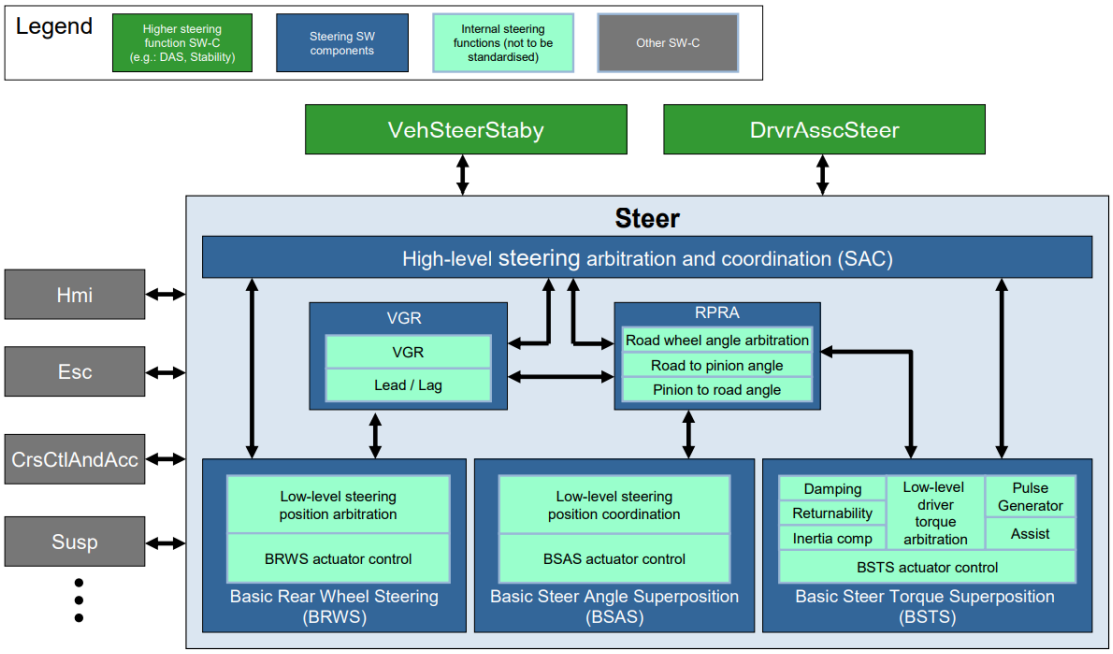


Figure 6.6: The complete Steer SW-C and its interfaces

A possible physical implementation of this architecture can be seen in Figure 6.7 below.

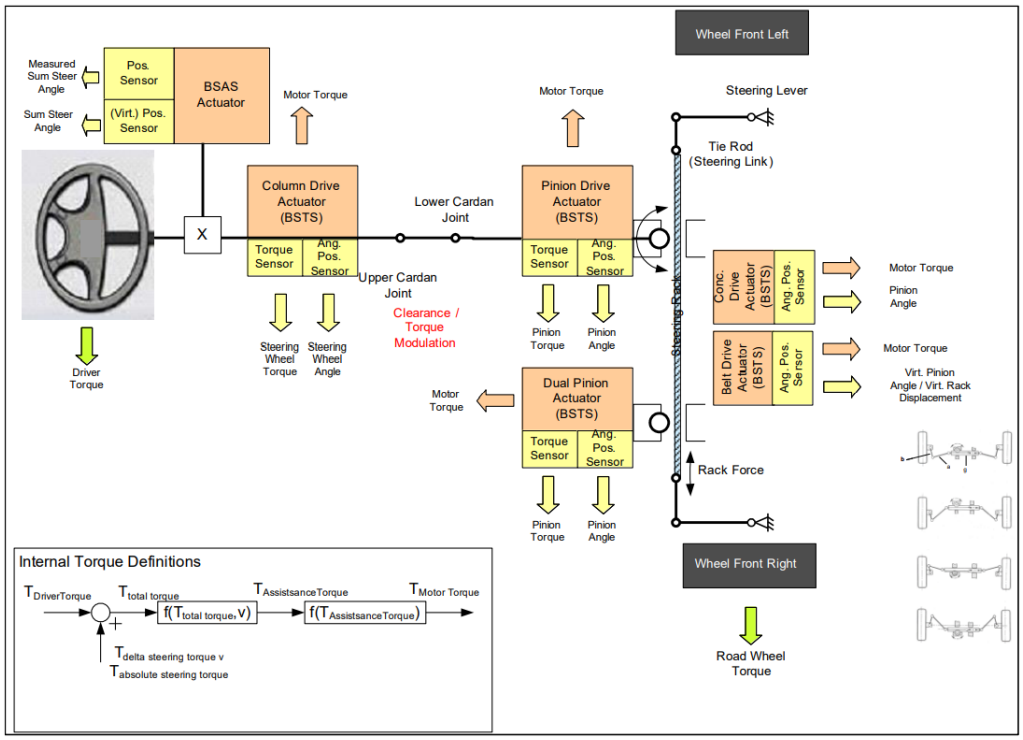


Figure 6.7: Possible physical steering architecture

6.8 Software Component SteerDrvrAsscSys (Steering Driver Assistance System)

In the current release, the SteerDrvrAsscSys SW-C is a place-holder for high- level vehicle comfort functions which send requests to the Steer SW-C. The nature of these requests (angle or torque) depends on the steering system(s) onboard the vehicle.

Examples of these high-level steering driver assistance system functions are:

- Lane Keeping
- Parking Aid
- Haptic Feedback Request

6.9 Software Component SteerVehStabyCtrl (Steering Vehicle Stabilizing Control)

In the current release, the SW-C SteerVehStabyCtrl is an additional function to the ESC stabilizing system, which acts on the active steering actuators

- Superposition steering angle actuator (front and/or rear axle)
- Superposition steering torque actuator (front axle)

The intention is to improve the vehicle stability in a more comfortable way than the ESC internal YRC in special driving situations:

- Oversteering support
- Split-μ braking support

Because the functionality in detail depends on the actuator capabilities, which differ from superposition steering angle and torque actuators, the following descriptions of the use cases is organized according to the used actuators.

6.9.1 Superposition steering angle actuation

The intention of using superposition steering angle actuator is to improve the vehicle stability in a more comfortable way than the ESC internal YRC by reducing the driver workload in stabilizing the vehicle in special driving situations:

- Oversteering support:
  - The oversteering control function computes a steering angle request for automatic countersteering, when it detects an oversteering situation.
  - The function stabilizes the vehicle according to the driver intend and the physical limitations by using the superposition steering angle device.
  - An oversteering situation is detected from the difference between target yaw rate and measured yaw rate.
  - Since steering interventions are more comfortable for the driver, they can be started earlier than brake interventions by ESC and unstable driving situations are controlled easier. This is done as unperceivable to the driver as possible and with a minimized loss of velocity.
- Split-μ support
  - When braking or accelerating on a road with friction differences between left and right side (split-μ), an unwanted yaw torque is generated by the unequal wheel forces. This yaw torque is compensated by an automatic counter steering angle intervention requested by the SteerVehStabyCtrl SW-C. This leads to a largely reduced driver steering effort.
  - Since automatic counter steering is faster than counter steering by the driver, a quicker build-up of the yaw torque from the unequal forces can be tolerated. By modifying the parameters of the ESC internal yaw torque build-up delay, the braking distance can be reduced, too.

6.9.2 Superposition steering torque actuation

The intention of using superposition steering torque actuation is to improve the vehicle stability in a more comfortable way than the ESC internal YRC and increase the safety feeling of the driver especially in special driving situations:

- Over steering
  - The over steering control function computes a steering torque command to support the driver in counter steering, when it detects an over steering situation. An over steering situation is detected e.g. from the difference between target yaw rate and measured yaw rate.
  - Since steering interventions are more comfortable for the driver, they can be started earlier than brake interventions and unstable driving situations are controlled more sophisticated.
- Split-μ support
  - When braking or acceleration on a road with friction differences between left and right side (split-μ), the driver has to counter steer against the yaw torque generated by the unequal wheel forces. The split-μ control function computes a steering torque request to support the driver in counter steering.
  - Since assisted counter steering is faster than counter steering by the driver alone, a quicker build-up of the yaw torque from the unequal wheel forces can be tolerated. By modifying the parameters of the ESC internal yaw torque build-up delay, the braking distance can be reduced.

6.10 Software Component Susp (Suspension System)

The SW-C Susp controls besides others:

- The vehicle body ride height (level control)
- The vehicle body pitch during braking and acceleration
- The vehicle body roll during cornering and straight ahead driving
- Ride control and wheel force control on uneven roads via suspension
- Roll behaviour via suspension

The different suspension actuators onboard could be:

- Semi-active dampers
- Switchable dampers
- Active stabilizers
- Switchable stabilizers (Semi-active stabilizers)

- Switchable springs
- Body level actuator

For describing the use case Suspension, a unique methodology was established: a feature versus technology matrix, see Table 6.3. Figure 6.8 gives an overview of the SW-C Susp.

	semi-active/switchable **damper**	active/switchable **stabilizer**	Switchable **spring Body level actuator** (only body height control)
body ride height control	no	no	partly
ride control & wheel force control	yes	no	partly
body pitch control	yes	no	yes
body roll control	yes	yes	yes

Table 6.3: Feature Matrix Suspension

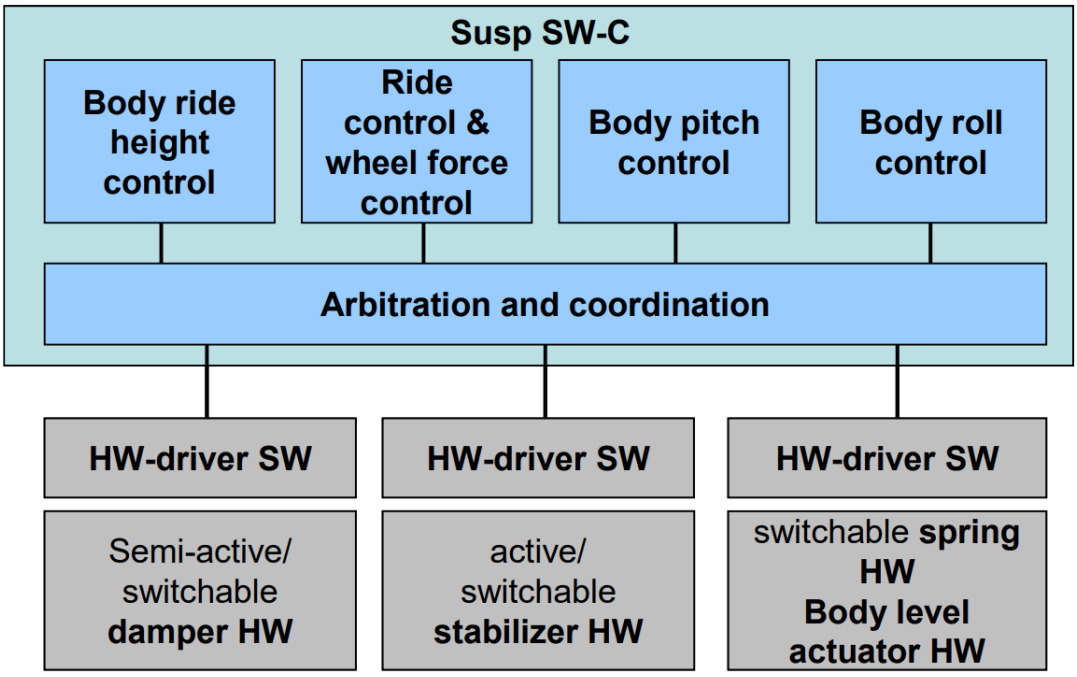


Figure 6.8: Example of the structure of SW-C Susp

6.11 Software Component TirePMon (Tire Pressure Monitoring)

The goal of the SW-C TirePMon is to monitor and identify a significant loss of tire pressure. Two different methods are known:

- Direct measuring systems measure the tire-pressure and temperature with a built-in tire or rim sensor and transfer the information via RF transmission to the receiver.

- Indirect measuring systems calculate the tire-pressure based on wheel- speed information. The detailed description that follows covers only the indirect systems.

Physical principals of indirect measuring systems:

- **Circumference based method**: The tire pressure is a function of the changing circumference.
  - Compares the current circumference of each wheel with a learned default value
  - Detection of pressure loss at 1,2 or 3 wheels
  - 30% drop in tire pressure leads to an increase in wheel speed of 0.2% to 0.5%.
- **Frequency based method**: The tire pressure is a function of the tire resonance frequency.
  - Compares the "resonance frequency" (typically around 42 Hz) of each wheel with a learned default value
  - Detection of simultaneous pressure loss at all wheels
- Combinations of these physical principles are also possible.

There are two main operation modes:

- Calibration
  - The driver "resets" the system e.g. due to change of tires
  - The TirePMon calibrates (direct measuring systems) the pressure to a preset value or learns (indirect measuring systems) the relative wheel speed differences with nominal tire pressure
- Detection
  - The TirePMon monitors the tire pressure and detects significant loss of tire pressure by direct measuring or comparing the current wheel speed information with the values learned during calibration (indirect measuring systems) respectively.

6.12 Software Component DtTqDistbn (Drivetrain Torque Distribution)

The SW-C DtTqDistbn includes control of all propulsion torque distributing active drive train components except engine, gearbox and clutch with manual gearboxes in order to influence the lateral vehicle dynamics, e.g.

- Limited slip differentials

- Torque vectoring devices
- Hang-on couplings

It also includes axle-dependent interventions for single front or rear wheel drives.

It does not contain the actuator control. The DtTqDistbn SW-C uses the mentioned drive train components as smart actuators.

Use Cases of DtTqDistbn SW-C to influence the vehicle dynamics:

- Basic torque distribution
- Traction control
- Split-μ support
- Yaw rate control
- Handling balance

In order to control a variety of drive train components in numerous drive train configurations, interfaces to three basic drive train elements have been defined:

1. Hang-on couplings
2. Active Differentials
3. Torque Vectoring Devices

The principle layout of these actuators is show in Figure 6.9.

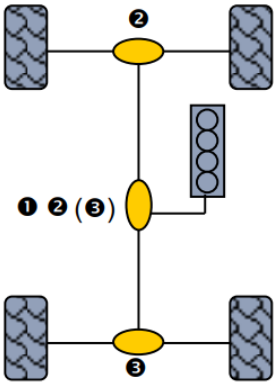


Figure 6.9: Drivetrain Torque Distribution actuator layout principle

6.12.1 Hang-on Coupling

A hang-on coupling is a speed converter. It transfers a torque from input to output shaft. Its block diagram is shown in Figure 6.10.

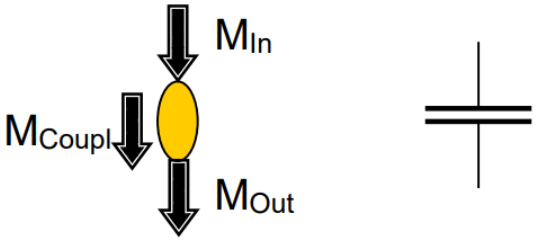


Figure 6.10: Hang-on coupling

- M_{Coupl} can only be applied if there is a speed difference between input and output shaft
- Controls:
  - $M_{Coupl} = M_{out} = f$(coupling force, difference velocity)
  - Coupling force limited by clamping force
  - For controls, additionally to the current applied coupling force, the "actuator potential" indicates the maximum torque which can (potentially) be transferred and is limited by e.g. overheat, wear, design. This actuator potential can vary in time.
- The range from the output torque is between zero and M_{In}: $M_{out}/M_{in} = [0, 1]$
- Sign convention: from faster to slower shaft, is positive

6.12.2 Active Differentials

An active differential is a device to distribute an incoming torque to two output shafts running on different speeds. It usually consists of a bevel gear and a coupling which can be electrically or hydraulically controlled. Its block diagram is shown in Figure 6.11.

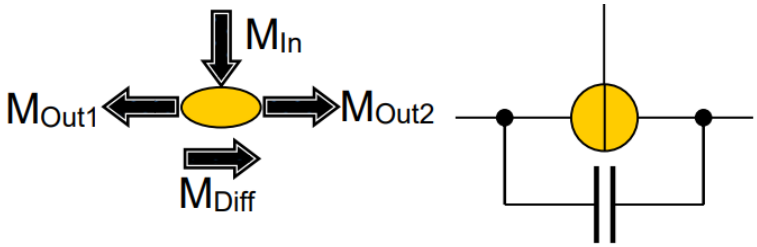


Figure 6.11: Block diagram, Active Differential

- M_{Diff} is an absolute value of the differential torque between left and right, it can be controlled electronically or hydraulically.
- In order to transfer a torque from one side to the other, there is a clutch, e.g. on split-μ, the low-μ side will spin. To avoid this, the clutch can transfer a torque from one side to the other.

- Prerequisite for the torque transfer with an active differential is a difference of the rotating velocity of both output shafts. Driving ahead, there is a 50%/50%-distribution to left and right. No torque can be transferred by the clutch.
- Special effects like torque bias ratio, friction losses etc. remain unconsidered.
- Because there might be a transfer ratio between input and output shafts, M_{in} has to be considered on the same level as outputs $M_{Out1,2}$.
- Sign convention:
  - M_{Diff} coupling torque: from faster to slower side, is always positive
- Controls:
  - Consolidated differential torque $M_{Diff} = f(\text{clutch force, difference velocity})$
  - Because the differential torque can only be applied if there is a difference velocity, a "maximum torque" indicates the potentially transferable torque if there would be a difference velocity. This torque is limited by the clamping force of the clutch.
  - For controls, additionally to the current applied coupling force and the maximum torque, the "actuator potential" indicates the maximum which can (potentially) be transferred and which is limited by e.g. overheat, wear, design. This actuator potential can vary in time.

6.12.3 Torque Vectoring Device

As an active differential, a torque vectoring device distributes an incoming torque to two output shafts by means of hydraulic or mechanical devices. In contrast to an active differential, a torque vectoring device "produces" the necessary difference velocities of the shafts internally by means of suitable gearsets. That means, a torque vectoring device can build-up a torque difference even if there is no difference velocity of the output shafts. Its block diagram is shown in Figure 6.12.

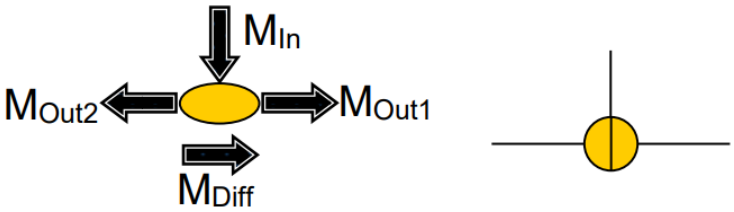


Figure 6.12: Block Diagram, Torque Vectoring Device

- Torque distribution:
  - $M_{Out1} = 1/2 * (M_{In} + M_{Diff})$
  - $M_{Out2} = 1/2 * (M_{In} - M_{Diff})$
  - $M_{Out1} + M_{Out2} = M_{In}$

  - $M_{\text{Out1}} - M_{\text{Out2}} = M_{\text{Diff}}$
  - 1 = right or front; 2 = left or rear
- Because there might be a transfer ratio between input and output shafts, M_{In} has to be considered on the same level as outputs $M_{\text{Out1,2}}$.
- M_{Diff} is an absolute value of the differential torque between 1 and 2, it can be controlled
- Sign convention:
  - M_{Diff} coupling torque: positive to the right or front
- Controls:
  - Differential torque M_{Diff}
  - For controls, additionally to the current applied differential torque, the "actuator potential" indicates the maximum which can (potentially) be transferred and which is limited by e.g. overheat, wear, design. This actuator potential can vary in time.

6.13 Software Component SurrndgsSnsr (Surroundings Sensor)

The SW-C SurrndgsSnsr has the role to provide information on other local traffic participants and on stationary objects with relevance for the ego vehicle motion, as e.g. lane markings. Note that the information on the traffic participants are aimed to be used not only for ACC but also for other potential use cases e.g. occupant safety systems.

The tasks of the SW-Component consist of, e.g.:

- Sensor signal processing
- Sensor hardware monitoring (blindness detection)
- Determination and compensation of sensor misalignment
- Object tracking and clustering
- Determination of object attributes and classification

6.14 Software Component ChassisSnsr (Chassis Sensor)

The SW-C ChassisSnsr provides sensor information to be used among SW-Components in the Chassis Domain. For example, following sensor information are delivered by the SW-C ChassisSnsr.

- Longitudinal/ lateral/ vertical acceleration
- Ride height level

- Yaw/ roll/ pitch rate
- Roll/ pitch angle
- Steering wheel angle
- Steering wheel torque

Note that those sensor information could be used in other domains.

Comment: these sensors belong to the already mentioned internal state sensors.

6.15 Software Component PrkgAid (Parking Aid)

The SW-C PrkgAid provides the distances to obstacles detected in different park segments measured by ultrasonic sensors. A typical use for ultrasonic sensors is depicted in Figure 6.13: relevant park segments for 8 ultrasonic sensors, detection in front and rear are shown.

These distances are highly suitable for other driver assistance functions as use for plausibility checks. The distances are combined with status information of these signals (enums) in a record for each of the 8 segments.

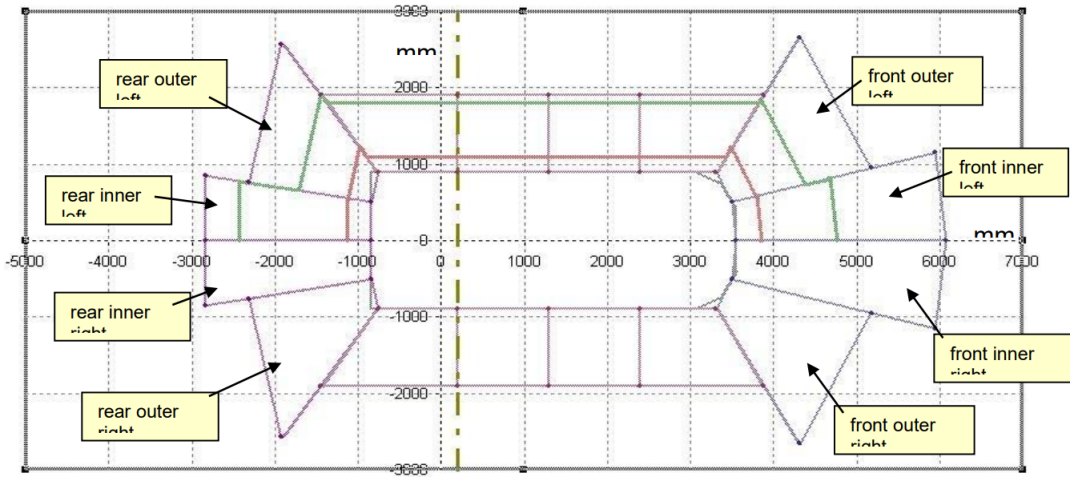


Figure 6.13: Relevant park segments for 8 ultrasonic sensor detection in front and in rear

7 Outlook

7.1 Chassis Domain Structure

For AUTOSAR Release R4.0 the definition of SW-Cs and Application Interfaces is based on an assumed Chassis Domain Structure as described in chapter 5. This structure is an example among many other possible structures and configurations. During AUTOSAR Phase II, several structures have been discussed. An example for a more complex and powerful structure has been worked out, see Figure 7.1.

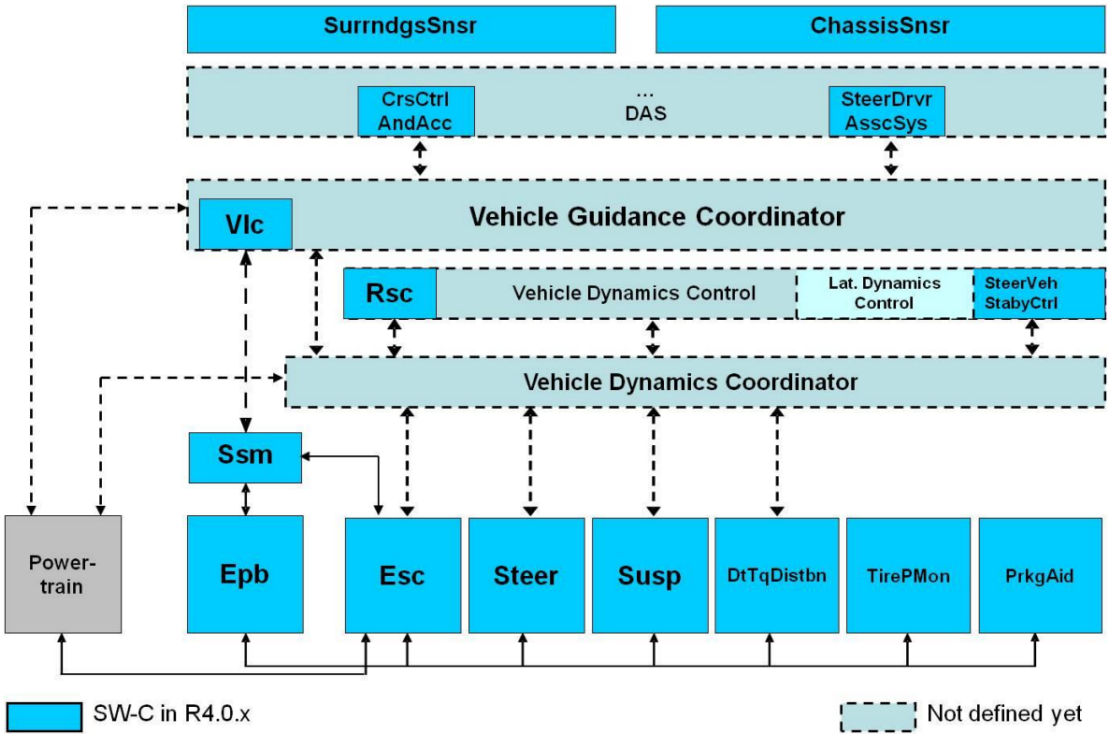


Figure 7.1: Example of Chassis Domain Structure (Outlook)

This structure consists of three layers:

- Vehicle guidance layer including DAS functions and Vehicle Guidance Coordinator,
- Vehicle stability layer including Vehicle Dynamics Control functions like RSC, lateral Dynamics control, and Steering VSS and the Vehicle Dynamics Coordinator,
- Actuator systems with associated control functions.

Vehicle Guidance Coordinator

- Goal of the Vehicle Guidance Coordinator is to arbitrate the vehicle longitudinal and lateral trajectory requests from the vehicle guidance functions and distribute them to the actuators systems.
- Sends requests to the Vehicle Dynamics Coordinator and the Powertrain Domain.

Vehicle Dynamics Coordinator

- Goal of the Vehicle Dynamics Coordinator is to arbitrate the vehicle stabilization requests and distribute them to the actuator systems with their associated control functions

7.2 Scalability

Also an example of a more complex structure has to cope with a large variety of configurations. Examples for scalability of this structure are given below in Figure 7.2 to Figure 7.4.

7.2.1 Use Case: ESC, Powertrain, and Tire Pressure Monitoring (TirePMon)

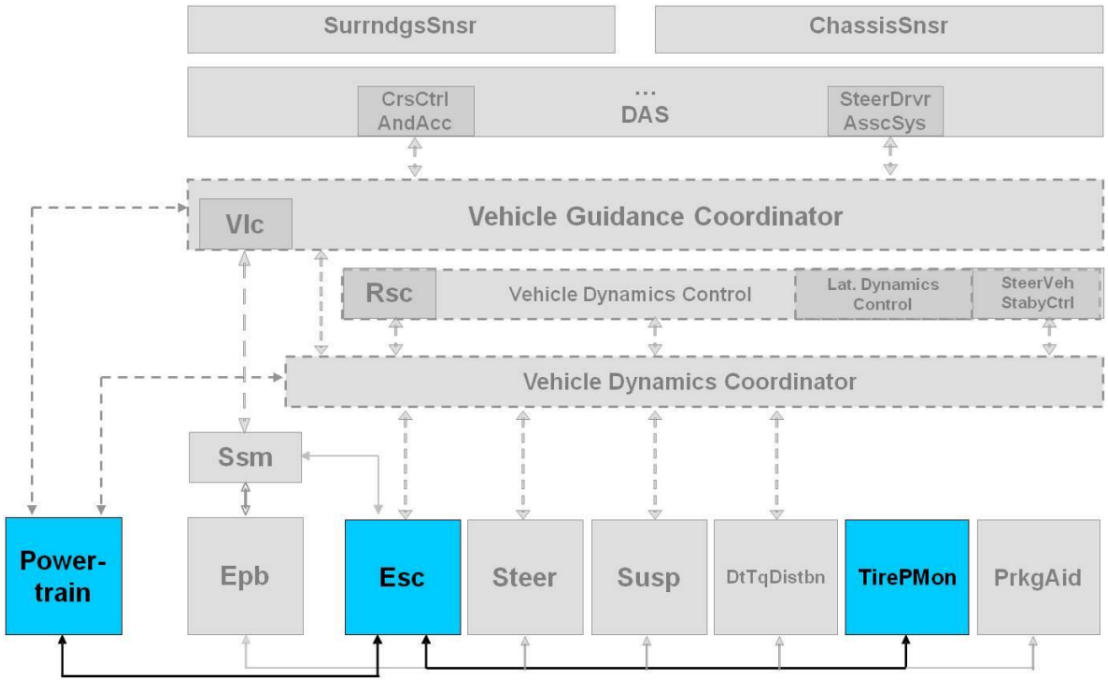


Figure 7.2: Scalability Use Case ESC, Powertrain, and Tire Pressure Monitoring

7.2.2 Use Case: Cruise Control

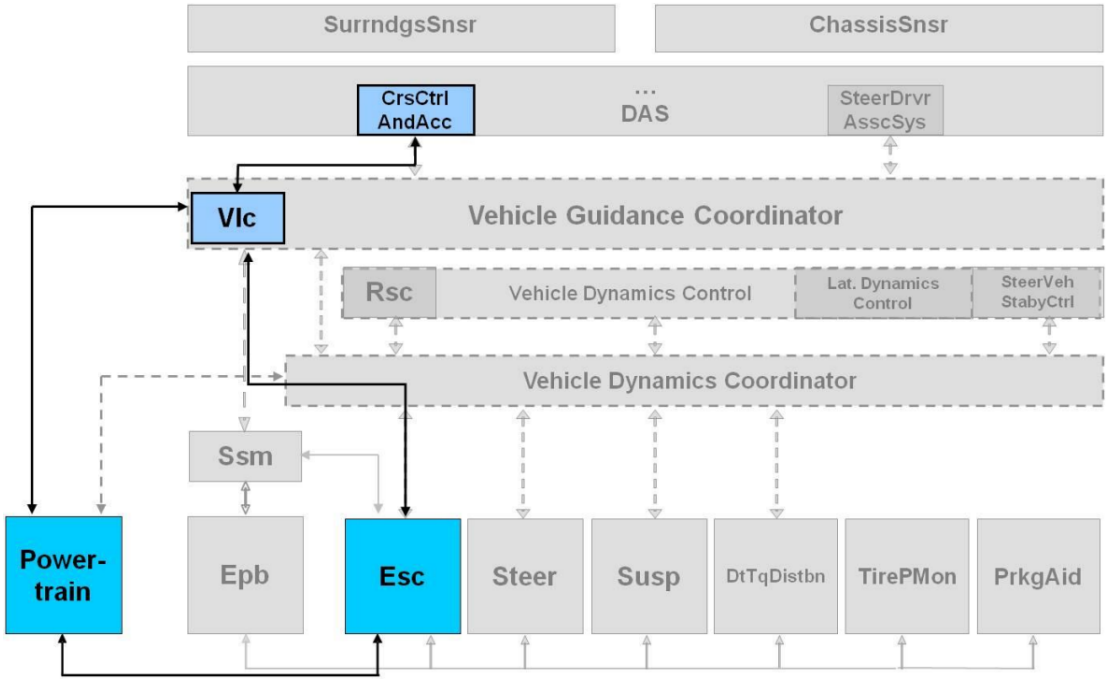


Figure 7.3: Scalability Use Case Cruise Control

7.2.3 Use Case: Steering DAS Superposition Steering Torque

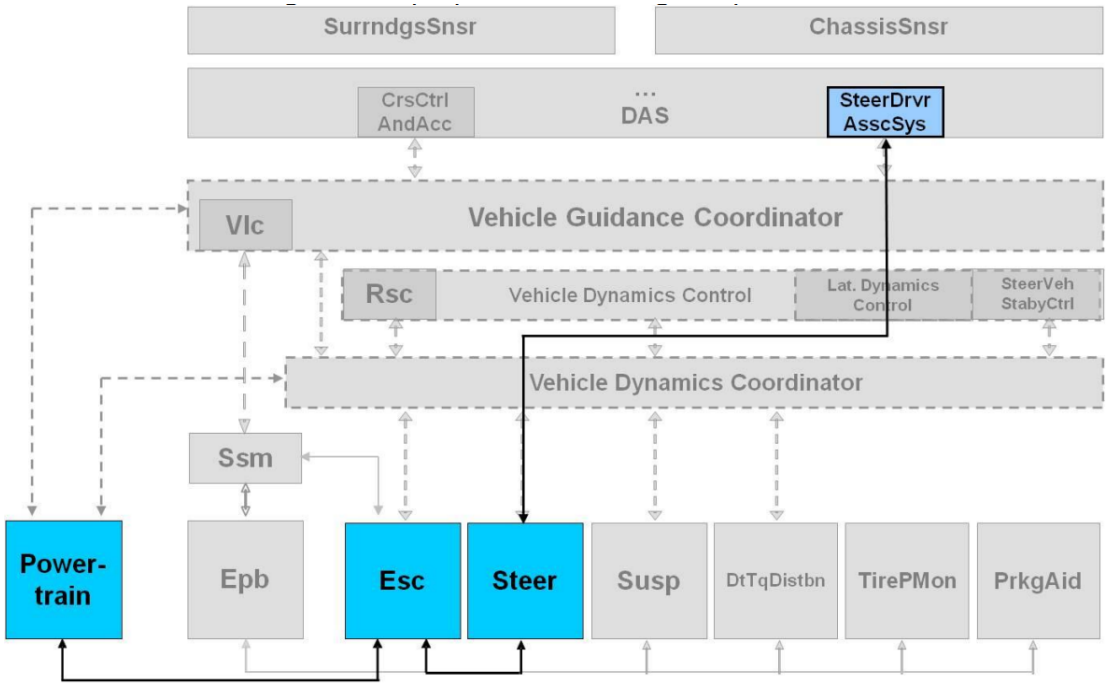


Figure 7.4: Scalability Use Case Steering DAS Superposition Steering Torque