

Lcc - Lambda Correction Calculation

1 [Lambda Correction Calculation] Lambda Correction Calculation

1.1 [Overview]

Figure 1: [Lcc Function Overview]

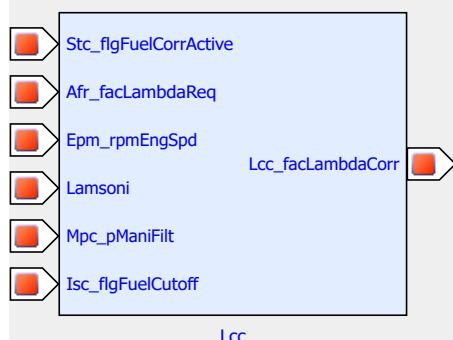
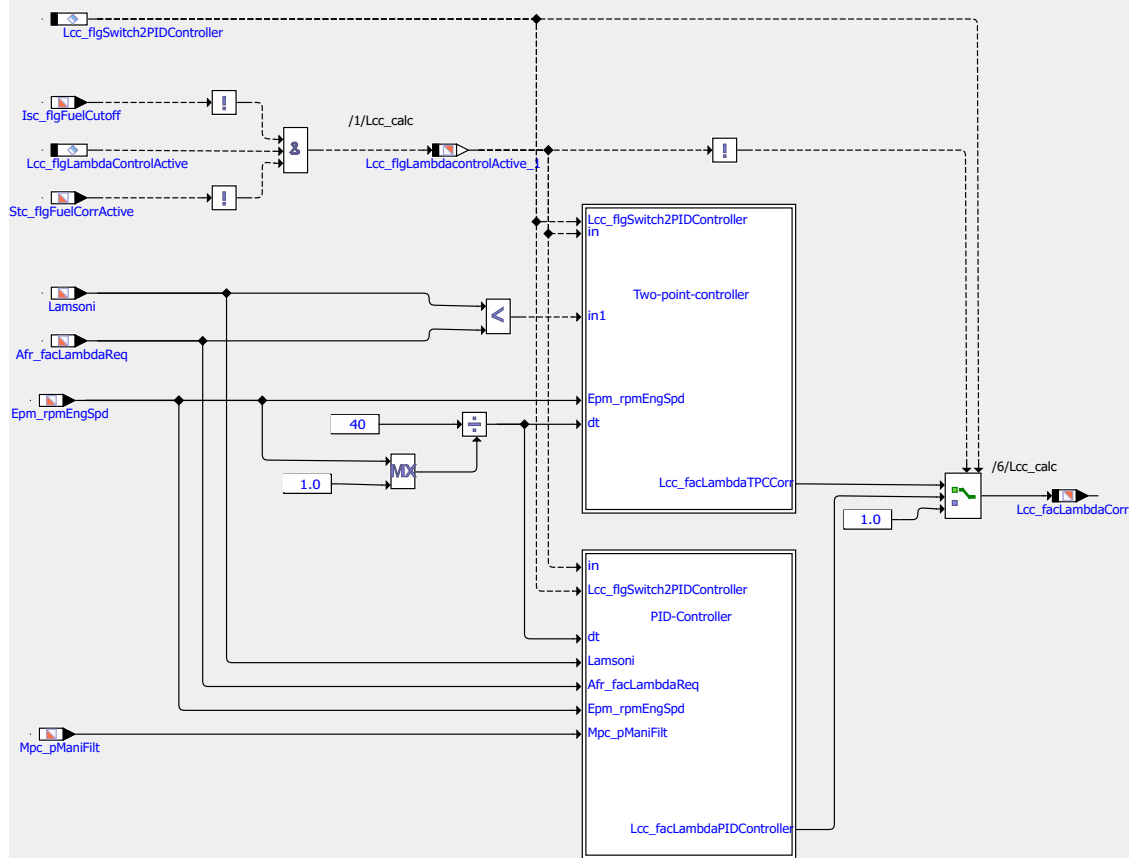


Figure 2: [LambdaControl.Main]



1.2 [Lambdakorrektur]

In die Motorsteuerung werden 2 Arten von Lambdaregelungen integriert. Zum Einen ein Zweipunktregler und zum Anderen ein stetiger Regler, als PID-Regler ausgeführt. Es werden zwei Regelkonzepte getestet, um das bestmögliche Ergebnis zu erzielen. Ein PID-Regler ist zudem schwer zu parametrieren, da die Regelstrecke eine Totzeit beinhaltet. Die verwendete Lambdasonde ist eine Breitbandsonde, die auch eine fette und magere Gemisch- Zusammensetzung erkennt. So ist es auch mit dem Zweipunktregler möglich fett und mager zu fahren.

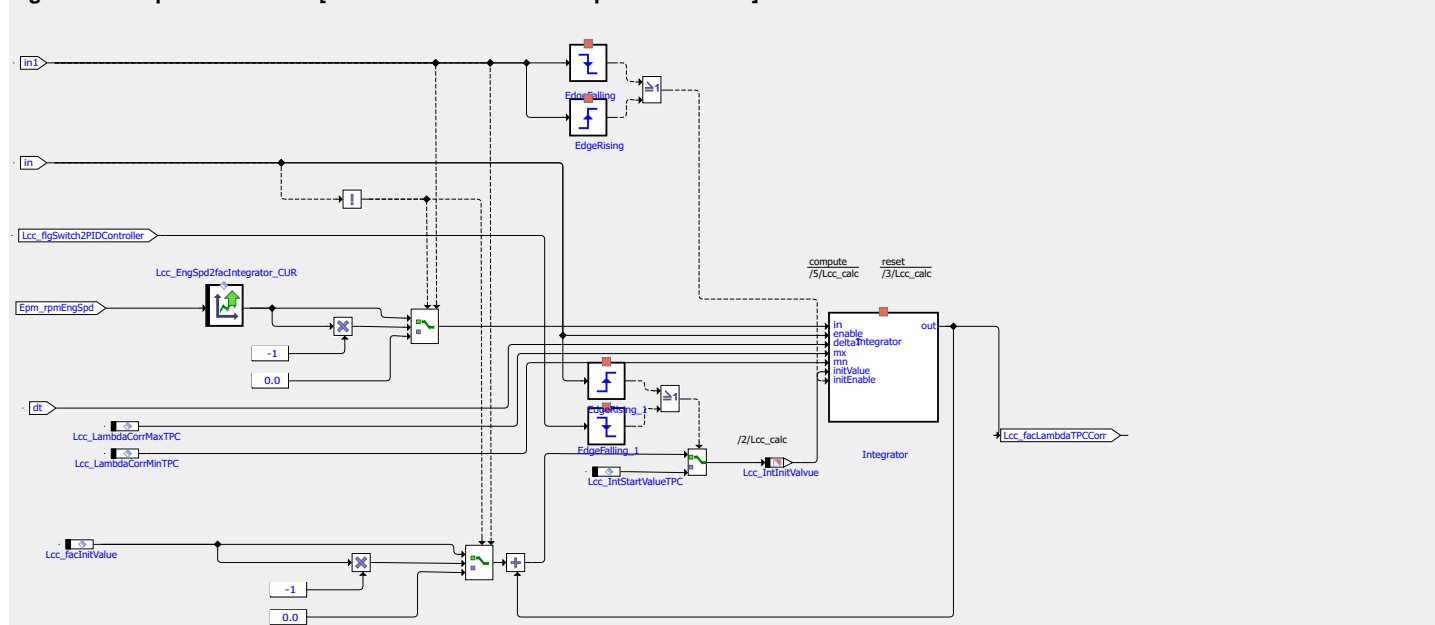
Lambdasondenerfassung:

Dieses Modul dient ausschließlich zur Inbetriebnahme der Lambdaregelung. Zur Erfassung dient ein Analogeingang, bis die eigentliche Lambdaerfassung durch die Basissoftware unterstützt wird. Dieses analoge Spannungssignal wird im Kennfeld zu einem Lambdawert umgerechnet und in die Variable **Lamsoni** geschrieben.

Bedingungen für aktive Lambdaregelung:

Die Lambdaregelung ist im Normalbetrieb aktiv. Ausnahmen sind der Schubetrieb (`lsc_flgFuelCutoff`), bei dem keine Einspritzung stattfindet und während des Startvorgangs (`Stc_flgFuelCorrActive`). Bei diesem wird erstens die Einspritzmenge gesteuert und zweitens muss die Lambdasonde vorgeheizt werden, damit sie betriebsbereit ist. Die Umschaltung zwischen PID-Regelung und Zweipunktregelung erfolgt über den Parameter `Lcc_flgSwitch2PIDController`.

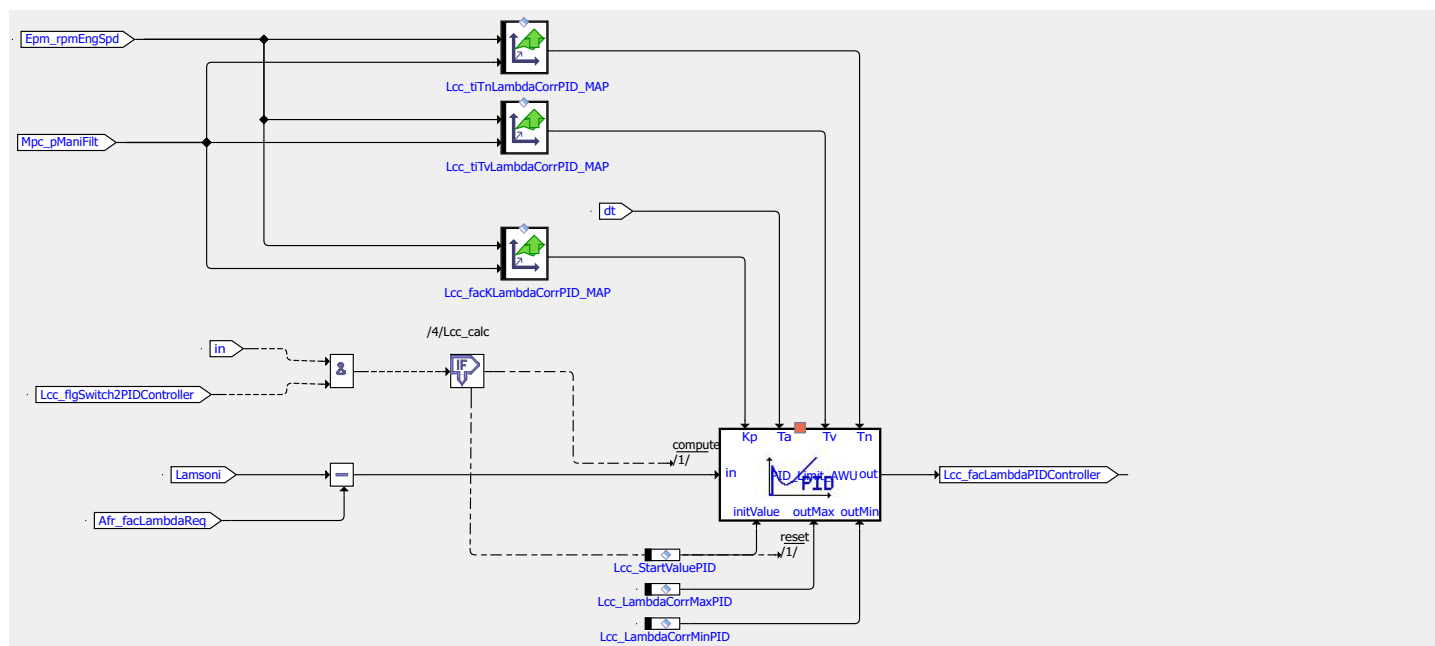
Figure 3: Two-point-controller [LambdaControl.Main.Two-point-controller]



Zweipunktregler:

Der Zweipunktregler funktioniert über einen Integrator. Wird zum Beispiel Lambda 1 geregelt und die Lambdasonde gibt einen mageren Wert aus (Lambda > 1) wird der Korrekturfaktor solange erhöht, bis die Einspritzung so groß ist, dass ein fetter Lambdawert vorliegt (Lambda kleiner 1). Danach wird der Korrekturfaktor nach unten korrigiert bis der Lambdawert wieder mager ist. Im Programm ist dies folgendermaßen angelegt: Der Lambdawert wird mit dem Lambda-Sollwert verglichen. Ist der Lambdawert größer, wird eine logische 0 ausgegeben, andernfalls eine logische 1. Sobald der Zustand wechselt, wird der verwendete Integrator initialisiert und es wird ein Startwert gesetzt. Dieser Startwert entspricht der Summe aus Integratorausgang und einem Korrekturfaktor (`Lcc_facInitValue`). Der Korrekturfaktor erhöht die Dynamik der Regelung. Der Integratoreingang wird aus dem Kennfeld `Lcc_EngSpd2facIntegrator_CUR` gebildet und ist abhängig von der Drehzahl. Der Ausgang des Integrators ist begrenzt durch `Lcc_LambdaCorrMinTPC` und `Lcc_LambdaCorrMaxTPC`.

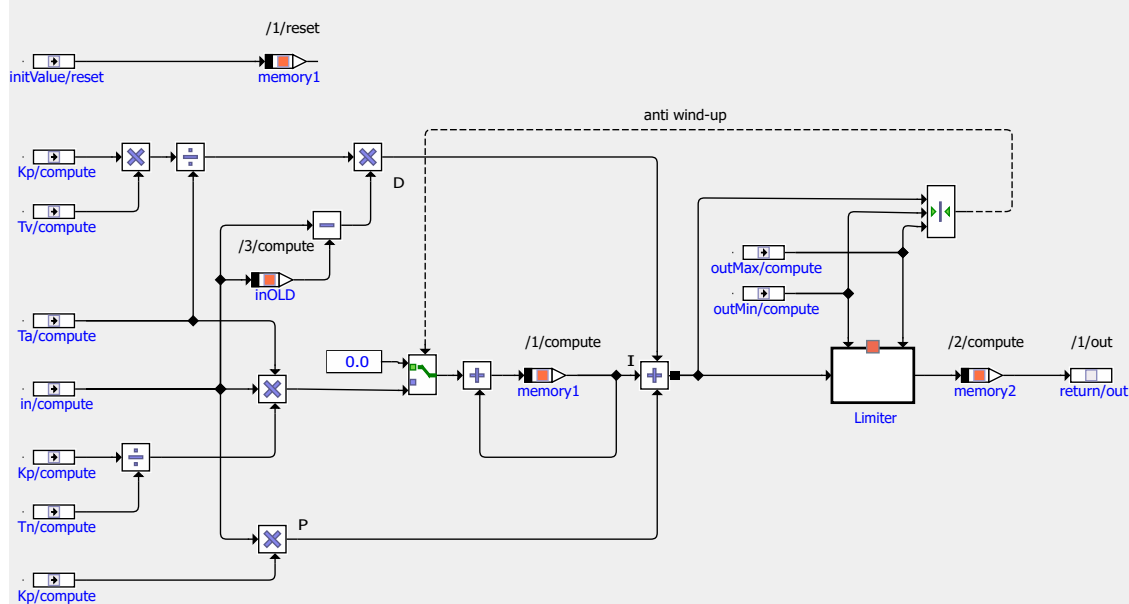
Figure 4: PID-Controller [LambdaControl.Main.PID-Controller]



PID-Regler:

Der Eingang des PID-Reglers wird aus der Differenz zwischen Lambdawert und Lambda-Sollwert gebildet. P-, I- und D-Anteil sind abhängig von der Drehzahl und des Saugrohrdrucks. Die Verstärkungsfaktoren sind in `Lcc_facKLambdaCorrPID_MAP` (P-Anteil), `Lcc_tiTnLambdaCorrPID_MAP` (I-Anteil) und `Lcc_tiTvLambdaCorrPID_MAP` (D-Anteil) hinterlegt. Auch der PID-Regler ist begrenzt durch `T2t_ThrValveReqCorrMin` und `T2t_ThrValveReqCorrMax` den Korrekturfaktor.

Figure 5: PID Limit [PID_Limit_AWU.Main]



Die Auswertung der Funktion `Lcc_calc` erfolgt im Synchron-Raster.

2 [C-Code Source]

2.1 [Code Listing]

```
/* BEGIN: ASCET REGION "Generation Information" */
/*****
 * BEGIN: Generation Information
 *-----
 * Component:.....Module
 * Name:....."LambdaControl"
 * Implementation:....."Impl"
 *****/
```

```

* Dataset:....."Data"
* Specification:.....Block Diagram
* Version:.....<empty String>
* Library Path:....."smartml60\Project_SmartM160\Function_Modules"
*-----
* Project Name:....."FlexECU_M160"
* Project Library Path:....."smartml60\Project_SmartM160\"
*-----
* Generation Date:.....03.12.2014
* Generation Time:.....13:41:34
*-----
* ASCET Version:.....V6.1.4 RB-DGS 2.3
* ASCET-MD Version:.....V6.1.4
* ASCET-RP Version:.....V6.1.4
* ASCET-SE Version:.....V6.1.4.28.19 CID[610]
*-----
* END: Generation Information
*****
/* END: ASCET REGION "Generation Information" */

/* BEGIN: ASCET REGION "Project Options" */
/*****
* BEGIN: Project Options "Build"/"Code"
*-----
*      Build
*-----
* Code Generator:.....Object Based Controller Implementation
* Compiler:.....Microsoft Visual C++ 2008
* Operating System:.....GENERIC_OSEK
* Target:.....ANSI-C
*-----
*      Code
*-----
* Add Comment with Generation Information for each Component [true]: true
* Add Comment with Implementation Information for each Assignment Statement [true]: true
* Add Comment with Specification Source for each Statement [true]:..true
* Add parenthesis for readability [false]:.....false
* Casting [MISRA]:.....MISRA
* Force Parenthesis for Binary Logical Operators [false]:.....false
* Generate Define Directives for Enum Values [false]:.....false
* Prefix for Component Names [<empty String>]:.....<empty String>
* Protected against division by zero [true]:.....true
* Protected Division against Signed Overflow [true]:.....true
* Protected Vector Indices [true]:.....true
*-----
*      Code.Compiler
*-----
* Division truncation direction [Zero (T-division)]:.....Zero (T-division)
* Inline directive [__inline]:.....__inline
* Integer Bit Size [32]:.....32
* Private directive [static]:.....static
* Public directive [<empty String>]:.....<empty String>
*-----
*      Code.FixedPoint
*-----
* Allow Double bit Size for Division Numerators [true]:.....true
* Allow Limit Service for Assignment Limitation [true]:.....true
* Arithmetic Service Set [<None>]:.....<None>
* Generate Limiters (may be changed locally) [true]:.....true
* Generate Round Operation on float to integer Assignment [true]:..true
* Maximum bit Length (float) [64]:.....64
* Maximum bit Length (int) [32]:.....32
* Result on Division by Zero [numerator]:.....numerator
* Temp Vars always 32 bit (integer) [false]:.....false
* Use power of 2 approximations of literals [false]:.....false
* Use SHIFT Operation on Signed Values instead of DIV Operation [true]: true
* Use SHIFT Operation on Signed Values instead of MUL Operation [true]: true
*-----
*      Code.Optimizations
*-----
* Auto-inline private methods (Smaller code-size - may be changed locally) [false]: false
* Generate well-formed switch [false]:.....false
* Hierarchical Code-Generation (may be changed locally) [false]:..false
* Initialise history variable with zero [false]:.....false
* Optimize Direct Access Methods (Multiple Levels) [false]:.....false
* Optimize Direct Access Methods (One Level) [false]:.....false
* Optimize Static Actions (Restricted Modelling) [false]:.....false
* Outline Generated Methods (may be changed locally) [false]:.....false
*-----
*      Code.Production
*-----
* Add Implementation Definitions [true]:.....true
* Generate Access Macros for [(variables, messages)]:.....(variables, messages)
* Generate Access Methods for dT (Alternative: use OS dT directly) [true]: true
* Generate Data Structures [USELOCAL]:.....USELOCAL

```

```

* Generate Map File [true]:.....true
* Generate OS Configuration [true]:.....true
* -----
*      Station.Build
* -----
* Use Customized Data Type Names [false]:.....false
* -----
* END: Project Options "Build"/"Code"
*****/
/* END: ASCET REGION "Project Options" */
/* BEGIN: ASCET REGION "ASCET-SE AddOn Options" */
*****
* BEGIN: ASCET-SE AddOn Options
* -----
*      Code
* -----
* checkMemSectionVolatility [true]:.....false
* checkMultipleSend [false]:.....false
* distribVarMemClass ["DISTRAM"]:....."RAM"
* genAlwaysInitValues [false]:.....true
* genLogicElementsAs [PACKED_BITFIELD]:.....PACKED_BITFIELD
* genObjList [false]:.....false
* implInfoComments [true]:.....true
* initTaskMemClass ["ASD_INIT_TASK_MEM"]:....."ASD_INIT_TASK_MEM"
* isrMemClass ["ASD_ISR_MEM"]:....."ASD_ISR_MEM"
* mainMemClass ["ASD_EXT_CODE_MEM"]:....."ASD_EXT_CODE_MEM"
* optimizeUnusedCode [true]:.....true
* paramAsSysConst [false]:.....false
* pragmaMemClassAtDecl [false]:.....false
* pragmaMemClassEnabled [true]:.....false
* referenceMemClass ["REFRAM"]:....."RAM"
* shortNames [false]:.....false
* taskMemClass ["ASD_TASK_MEM"]:....."ASD_TASK_MEM"
* virtualParameterMemClass ["VIRT_PARAM"]:....."VIRT_PARAM"
* -----
*      Code.Appearance
* -----
* braceLineFeed [true]:.....true
* genDate [<undef>]:.....<undef>
* genTime [<undef>]:.....<undef>
* generateSignatureDecorationComments [true]:.....true
* lineFeedPosition [LEFT]:.....LEFT
* maxIdentLength [0]:.....40
* maxRightLength [60]:.....60
* minLeftLength [8]:.....8
* preventIndentStructInit [true]:.....true
* -----
*      OS
* -----
* Os-Config-C_gen_declaration_alarms [false]:.....false
* Os-Config-C_gen_declaration_appmodes [false]:.....false
* Os-Config-C_gen_dt_calc [false]:.....true
* Os-Config-C_gen_initCOM [false]:.....false
* Os-Config-C_gen_inittask [false]:.....true
* Os-Config-C_gen_main [false]:.....false
* Os-Config-C_gen_process_container [false]:.....true
* Os-Config-C_gen_startuphook [false]:.....false
* asd_exclusive_area ["ASD_EXCLUSIVE_AREA"]:....."ASCET_exclusive_area"
* messageDoInit [false]:.....false
* messageExternalMessageCopies [false]:.....false
* messageGenOSEKDeclarations [true]:.....false
* messageIgnoreUsageInInitTask [false]:.....false
* messageOverloadInitValues [<undef>]:.....<undef>
* messageUsageVariant [OPT_COPY]:.....NON_OPT_COPY
* modularMessageUse [false]:.....false
* osAppModePattern ["%name%"]:....."appmode_%name%"
* osStartupFunction [<undef>]:.....<undef>
* -----
*      OS.OIL
* -----
* OIL-COOP-RESOURCE-name ["ASD_Cooperative_Res"]:....."ASD_Cooperative_Res"
* OIL-outputFile ["temp.oil"]:....."temp.oil"
* -----
*      SERAP
* -----
* SERAPRefPageMemoryClass ["SERAP_REF"]:....."SERAP_REF"
* SERAPWorkPageMemoryClass ["SERAP_WORK"]:....."SERAP_WORK"
* serap [false]:.....false
* serapEmbedded [true]:.....true
* -----
*      Virtual Address Tables
* -----
* addressTable [true]:.....false
* addressTableMemoryClass ["VATROM"]:....."VATROM"
* -----

```

```

* END: ASCET-SE AddOn Options
*****

/* END: ASCET REGION "ASCET-SE AddOn Options" */
/* BEGIN: ASCET REGION "Module Data Definitions" */

/*****
* BEGIN: DEFINITION OF SUBSTRUCT VARIABLE 'Lcc_RAM'
* -----
* memory class:.....'RAM'
* model name:.....'Lcc'
* data set:.....'LAMBDACONTROL_IMPL_Data'
* -----*/
struct LAMBDACONTROL_IMPL_RAM_SUBSTRUCT Lcc_RAM = {
  /* substruct: Lcc_RAM.EdgeFalling (modeled as:'EdgeFalling.Lcc') */
  {
    /* struct element:'Lcc_RAM.EdgeFalling.buffer1' (modeled as:'buffer1.EdgeFalling.Lcc') */
    false,
    /* struct element:'Lcc_RAM.EdgeFalling.oldSignal1' (modeled as:'oldSignal1.EdgeFalling.Lcc') */
    false
  },
  /* substruct: Lcc_RAM.EdgeFalling_1 (modeled as:'EdgeFalling_1.Lcc') */
  {
    /* struct element:'Lcc_RAM.EdgeFalling_1.buffer1' (modeled as:'buffer1.EdgeFalling_1.Lcc') */
    false,
    /* struct element:'Lcc_RAM.EdgeFalling_1.oldSignal1' (modeled as:'oldSignal1.EdgeFalling_1.Lcc') */
    false
  },
  /* substruct: Lcc_RAM.EdgeRising (modeled as:'EdgeRising.Lcc') */
  {
    /* struct element:'Lcc_RAM.EdgeRising.buffer' (modeled as:'buffer.EdgeRising.Lcc') */
    false,
    /* struct element:'Lcc_RAM.EdgeRising.oldSignal' (modeled as:'oldSignal.EdgeRising.Lcc') */
    true
  },
  /* substruct: Lcc_RAM.EdgeRising_1 (modeled as:'EdgeRising_1.Lcc') */
  {
    /* struct element:'Lcc_RAM.EdgeRising_1.buffer' (modeled as:'buffer.EdgeRising_1.Lcc') */
    false,
    /* struct element:'Lcc_RAM.EdgeRising_1.oldSignal' (modeled as:'oldSignal.EdgeRising_1.Lcc') */
    true
  },
  /* substruct: Lcc_RAM.Integrator (modeled as:'Integrator.Lcc') */
  {
    /* struct element:'Lcc_RAM.Integrator.memory' (modeled as:'memory.Integrator.Lcc') */
    0.0
  },
  /* substruct: Lcc_RAM.PID_Limit_AWU (modeled as:'PID_Limit_AWU.Lcc') */
  {
    /* struct element:'Lcc_RAM.PID_Limit_AWU.inOLD' (modeled as:'inOLD.PID_Limit_AWU.Lcc') */
    0.0,
    /* struct element:'Lcc_RAM.PID_Limit_AWU.memory1' (modeled as:'memory1.PID_Limit_AWU.Lcc') */
    0.0,
    /* struct element:'Lcc_RAM.PID_Limit_AWU.memory2' (modeled as:'memory2.PID_Limit_AWU.Lcc') */
    0.0
  }
};
/* -----
* END: DEFINITION OF SUBSTRUCT VARIABLE 'Lcc_RAM'
*****

/*****
* BEGIN: DEFINITION OF COMPONENT VARIABLE 'Lcc'
* -----
* memory class:.....'ROM'
* model name:.....'Lcc'
* data set:.....'LAMBDACONTROL_IMPL_Data'
* -----*/
const struct LAMBDACONTROL_IMPL Lcc = {
  /* substruct: Lcc.EdgeFalling (modeled as:'EdgeFalling.Lcc') */
  {
    /* type descriptor pointer 'EDGEFALLING_IMPL_RAM' for memory class substruct for 'RAM' */
    &Lcc_RAM.EdgeFalling
  },
  /* substruct: Lcc.EdgeFalling_1 (modeled as:'EdgeFalling_1.Lcc') */
  {
    /* type descriptor pointer 'EDGEFALLING_IMPL_RAM' for memory class substruct for 'RAM' */
    &Lcc_RAM.EdgeFalling_1
  },
  /* substruct: Lcc.EdgeRising (modeled as:'EdgeRising.Lcc') */
  {
    /* type descriptor pointer 'EDGERISING_IMPL_RAM' for memory class substruct for 'RAM' */
    &Lcc_RAM.EdgeRising
  },
  /* substruct: Lcc.EdgeRising_1 (modeled as:'EdgeRising_1.Lcc') */

```

```

{
    /* type descriptor pointer 'EDGERISING_IMPL_RAM' for memory class substruct for 'RAM' */
    &Lcc_RAM.EdgeRising_1
},
/* substruct: Lcc.Integrator (modeled as: 'Integrator.Lcc') */
{
    /* type descriptor pointer 'INTEGRATOR_IMPL_RAM' for memory class substruct for 'RAM' */
    &Lcc_RAM.Integrator
},
/* substruct: Lcc.PID_Limit_AWU (modeled as: 'PID_Limit_AWU.Lcc') */
{
    /* type descriptor pointer 'PID_LIMIT_AWU_IMPL_RAM' for memory class substruct for 'RAM' */
    &Lcc_RAM.PID_Limit_AWU
}
};
/* -----
* END: DEFINITION OF COMPONENT VARIABLE 'Lcc'
***** */

/* END: ASCET REGION "Module Data Definitions" */

/* BEGIN: ASCET REGION "Exported Data Definitions" */

*****
* BEGIN: DEFINITION OF CHARACTERISTIC TABLE 'Lcc_EngSpd2facIntegrator_CUR'
* -----*/
const struct CharTable1_real64_7_real64_LAMBDACONTROL_IMPL_TYPE Lcc_EngSpd2facIntegrator_CUR = {
    7,
    {
        400.0, 1000.0, 2000.0, 3000.0, 4000.0, 5000.0, 6000.0
    },
    {
        0.15, 0.22, 0.3, 0.45, 0.52, 0.6, 0.68
    }
};
/* result: min=-oo, max=+oo, ident, limit=no */
/* x axis: min=-oo, max=+oo, ident */
/* -----
* END: DEFINITION OF CHARACTERISTIC TABLE 'Lcc_EngSpd2facIntegrator_CUR'
***** */

*****
* BEGIN: DEFINITION OF VARIABLE 'Lcc_IntInitValvue'
* -----*/
real64 Lcc_IntInitValvue = 0.0;
/* min=-oo, max=+oo, ident, limit=yes */
/* -----
* END: DEFINITION OF VARIABLE 'Lcc_IntInitValvue'
***** */

*****
* BEGIN: DEFINITION OF VARIABLE 'Lcc_IntStartValueTPC'
* -----*/
const real64 Lcc_IntStartValueTPC = 1.0;
/* min=-oo, max=+oo, ident, limit=yes */
/* -----
* END: DEFINITION OF VARIABLE 'Lcc_IntStartValueTPC'
***** */

*****
* BEGIN: DEFINITION OF VARIABLE 'Lcc_LambdaCorrMaxPID'
* -----*/
const real64 Lcc_LambdaCorrMaxPID = 2.0;
/* min=-oo, max=+oo, ident, limit=yes */
/* -----
* END: DEFINITION OF VARIABLE 'Lcc_LambdaCorrMaxPID'
***** */

*****
* BEGIN: DEFINITION OF VARIABLE 'Lcc_LambdaCorrMaxTPC'
* -----*/
const real64 Lcc_LambdaCorrMaxTPC = 2.0;
/* min=-oo, max=+oo, ident, limit=yes */
/* -----
* END: DEFINITION OF VARIABLE 'Lcc_LambdaCorrMaxTPC'
***** */

*****
* BEGIN: DEFINITION OF VARIABLE 'Lcc_LambdaCorrMinPID'
* -----*/

```

```

const real64 Lcc_LambdaCorrMinPID = 0.5;
/* min=-oo, max=+oo, ident, limit=yes */
/* -----
* END: DEFINITION OF VARIABLE 'Lcc_LambdaCorrMinPID'
*****

/*****
* BEGIN: DEFINITION OF VARIABLE 'Lcc_LambdaCorrMinTPC'
* -----*/
const real64 Lcc_LambdaCorrMinTPC = 0.5;
/* min=-oo, max=+oo, ident, limit=yes */
/* -----
* END: DEFINITION OF VARIABLE 'Lcc_LambdaCorrMinTPC'
*****

/*****
* BEGIN: DEFINITION OF VARIABLE 'Lcc_StartValuePID'
* -----*/
const real64 Lcc_StartValuePID = 1.0;
/* min=-oo, max=+oo, ident, limit=yes */
/* -----
* END: DEFINITION OF VARIABLE 'Lcc_StartValuePID'
*****

/*****
* BEGIN: DEFINITION OF VARIABLE 'Lcc_facInitValue'
* -----*/
const real64 Lcc_facInitValue = 0.03;
/* min=-oo, max=+oo, ident, limit=yes */
/* -----
* END: DEFINITION OF VARIABLE 'Lcc_facInitValue'
*****

/*****
* BEGIN: DEFINITION OF CHARACTERISTIC TABLE 'Lcc_facKLambdaCorrPID_MAP'
* -----*/
const struct CharTable2_real64_12_real64_10_real64_LAMBDACONTROL_IMPL_TYPE Lcc_facKLambdaCorrPID_MAP = {
    12,
    10,
    {
        500.0, 1000.0, 1500.0, 2000.0, 2500.0, 3000.0, 3500.0, 4000.0, 4500.0, 5000.0,
        5500.0, 6000.0
    },
    {
        20000.0, 50000.0, 70000.0, 90000.0, 110000.0, 130000.0, 150000.0, 170000.0,
        190000.0, 210000.0
    },
    {
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
    }
};
/* result: min=-oo, max=+oo, ident, limit=no */
/* x axis: min=-oo, max=+oo, ident */
/* y axis: min=-oo, max=+oo, ident */
/* -----
* END: DEFINITION OF CHARACTERISTIC TABLE 'Lcc_facKLambdaCorrPID_MAP'
*****

/*****
* BEGIN: DEFINITION OF VARIABLE 'Lcc_flgLambdaControlActive'
* -----*/
const uint8 Lcc_flgLambdaControlActive = false;
/* min=0, max=1, Identity, limit=yes */
/* -----
* END: DEFINITION OF VARIABLE 'Lcc_flgLambdaControlActive'
*****

/*****

```



```

/* result: min=-oo, max=+oo, ident, limit=no */
/* x axis: min=-oo, max=+oo, ident */
/* y axis: min=-oo, max=+oo, ident */
/* -----
* END: DEFINITION OF CHARACTERISTIC TABLE 'Lcc_tiTvLambdaCorrPID_MAP'
*****/

/* END: ASCET REGION "Exported Data Definitions" */

/* -----
* BEGIN: DEFINITION OF MESSAGES
* -----
* Total size is [bytes]:.....2
* -----*/
/* messages of memory class:.....'RAM' */
/* messages of size [bytes]:.....2 */
/* modelled as 'Lcc_facLambdaCorr' */
uint16 Lcc_facLambdaCorr;
/* -----
* END: DEFINITION OF MESSAGES
*****/

#define _EdgeFalling Lcc.EdgeFalling
#define _EdgeFalling_1 Lcc.EdgeFalling_1
#define _EdgeFalling_1_REF_ (&(Lcc.EdgeFalling_1))
#define _EdgeFalling_REF_ (&(Lcc.EdgeFalling))
#define _EdgeRising Lcc.EdgeRising
#define _EdgeRising_1 Lcc.EdgeRising_1
#define _EdgeRising_1_REF_ (&(Lcc.EdgeRising_1))
#define _EdgeRising_REF_ (&(Lcc.EdgeRising))
#define _Integrator Lcc.Integrator
#define _Integrator_REF_ (&(Lcc.Integrator))
#define _Lcc_EngSpd2facIntegrator_CUR Lcc_EngSpd2facIntegrator_CUR
#define _Lcc_EngSpd2facIntegrator_CUR_REF_ (&(Lcc_EngSpd2facIntegrator_CUR))
#define _Lcc_facInitValue Lcc_facInitValue
#define _Lcc_facKLambdaCorrPID_MAP Lcc_facKLambdaCorrPID_MAP
#define _Lcc_facKLambdaCorrPID_MAP_REF_ (&(Lcc_facKLambdaCorrPID_MAP))
#define _Lcc_flgLambdaControlActive Lcc_flgLambdaControlActive
#define _Lcc_flgLambdaControlActive_1 Lcc_flgLambdaControlActive_1
#define _Lcc_flgSwitch2PIDController Lcc_flgSwitch2PIDController
#define _Lcc_IntInitValvue Lcc_IntInitValvue
#define _Lcc_IntStartValueTPC Lcc_IntStartValueTPC
#define _Lcc_LambdaCorrMaxPID Lcc_LambdaCorrMaxPID
#define _Lcc_LambdaCorrMaxTPC Lcc_LambdaCorrMaxTPC
#define _Lcc_LambdaCorrMinPID Lcc_LambdaCorrMinPID
#define _Lcc_LambdaCorrMinTPC Lcc_LambdaCorrMinTPC
#define _Lcc_StartValuePID Lcc_StartValuePID
#define _Lcc_tiTnLambdaCorrPID_MAP Lcc_tiTnLambdaCorrPID_MAP
#define _Lcc_tiTnLambdaCorrPID_MAP_REF_ (&(Lcc_tiTnLambdaCorrPID_MAP))
#define _Lcc_tiTvLambdaCorrPID_MAP Lcc_tiTvLambdaCorrPID_MAP
#define _Lcc_tiTvLambdaCorrPID_MAP_REF_ (&(Lcc_tiTvLambdaCorrPID_MAP))
#define _PID_Limit_AWU Lcc.PID_Limit_AWU
#define _PID_Limit_AWU_REF_ (&(Lcc.PID_Limit_AWU))

/* BEGIN: ASCET REGION "Component Functions" */
/* -----
* BEGIN: FUNCTIONS OF COMPONENT
*****/

/* BEGIN: ASCET REGION "Process Definition 'Lcc_calc'" */
/* -----
* BEGIN: DEFINITION OF PROCESS 'LAMBDACONTROL_IMPL_Lcc_calc'
* -----
* model name:.....'Lcc_calc'
* memory class:.....'CODE'
* -----*/
//#if defined(COMPILE_UNUSED_CODE) || defined(COMPILE_UNUSED__LAMBDACONTROL_IMPL_Lcc_calc)
/* messages used by this process */

/* public Lcc_calc [] */

void LAMBDACONTROL_IMPL_Lcc_calc (void)
{
    /* temp. variables */
    real64 _treal64;

    /* define local message copies */
    uint16 Afr_facLambdaReq__LAMBDACONTROL_IMPL_Lcc_calc;
    sint16 Epm_rpmEngSpd__LAMBDACONTROL_IMPL_Lcc_calc;
    uint8 Isc_flgFuelCutoff__LAMBDACONTROL_IMPL_Lcc_calc;
    uint16 Lamsoni__LAMBDACONTROL_IMPL_Lcc_calc;
    uint16 Lcc_facLambdaCorr__LAMBDACONTROL_IMPL_Lcc_calc;
    uint16 Mpc_pManiFilt__LAMBDACONTROL_IMPL_Lcc_calc;
    uint8 Stc_flgFuelCorrActive__LAMBDACONTROL_IMPL_Lcc_calc;
    /* receive messages implicitly */

```

```

{
    DisableAllInterrupts();
    Afr_facLambdaReq__LAMBDACONTROL_IMPL_Lcc_calc = Afr_facLambdaReq;
    Epm_rpmEngSpd__LAMBDACONTROL_IMPL_Lcc_calc = Epm_rpmEngSpd;
    Isc_flgFuelCutoff__LAMBDACONTROL_IMPL_Lcc_calc = Isc_flgFuelCutoff;
    Lamsoni__LAMBDACONTROL_IMPL_Lcc_calc = Lamsoni;
    Lcc_facLambdaCorr__LAMBDACONTROL_IMPL_Lcc_calc = Lcc_facLambdaCorr;
    Mpc_pManiFilt__LAMBDACONTROL_IMPL_Lcc_calc = Mpc_pManiFilt;
    Stc_flgFuelCorrActive__LAMBDACONTROL_IMPL_Lcc_calc = Stc_flgFuelCorrActive;
    EnableAllInterrupts();
}
/* Lcc_calc: sequence call #1 */
_Lcc_flgLambdacontrolActive_1
= !Isc_flgFuelCutoff__LAMBDACONTROL_IMPL_Lcc_calc && _Lcc_flgLambdaControlActive && !
Stc_flgFuelCorrActive__LAMBDACONTROL_IMPL_Lcc_calc;
/* Lcc_calc: sequence call #2 */
/* assignment to Lcc_IntInitValvue: min=-oo, max=+oo, hex=phys, limit=n.a., zero incl.=true */
_Lcc_IntInitValvue
= ((EDGERISING_IMPL_compute(_EdgeRising_1_REF_, _Lcc_flgLambdacontrolActive_1)
|| EDGEFALLING_IMPL_compute(_EdgeFalling_1_REF_, _Lcc_flgSwitch2PIDController)) ?
_Lcc_IntStartValueTPC : (((!_Lcc_flgLambdacontrolActive_1) ? 0.0 : ((Lamsoni__LAMBDACONTROL_IMPL_Lcc_calc
< Afr_facLambdaReq__LAMBDACONTROL_IMPL_Lcc_calc) ? -_Lcc_facInitValue : _Lcc_facInitValue))) +
INTEGRATOR_IMPL_out(_Integrator_REF_));
/* Lcc_calc: sequence call #3 */
INTEGRATOR_IMPL_reset(_Integrator_REF_, _Lcc_IntInitValvue, EDGEFALLING_IMPL_compute(_EdgeFalling_REF_,
Lamsoni__LAMBDACONTROL_IMPL_Lcc_calc < Afr_facLambdaReq__LAMBDACONTROL_IMPL_Lcc_calc)
|| EDGERISING_IMPL_compute(_EdgeRising_REF_, Lamsoni__LAMBDACONTROL_IMPL_Lcc_calc <
Afr_facLambdaReq__LAMBDACONTROL_IMPL_Lcc_calc));
/* Lcc_calc: sequence call #4 */
if (_Lcc_flgLambdacontrolActive_1 && _Lcc_flgSwitch2PIDController)
{
    /* If-block: sequence call #4/Then #1 */
    _tlreal64
    = (((real64)Epm_rpmEngSpd__LAMBDACONTROL_IMPL_Lcc_calc == 0.0) ? 80.0 : 80.0 /
(real64)Epm_rpmEngSpd__LAMBDACONTROL_IMPL_Lcc_calc);
    PID_LIMIT_AWU_IMPL_compute(_PID_Limit_AWU_REF_, ((real64)Lamsoni__LAMBDACONTROL_IMPL_Lcc_calc
- (real64)Afr_facLambdaReq__LAMBDACONTROL_IMPL_Lcc_calc) * 0.00048828125,
CharTable2_getAt_r64r64r64(_Lcc_facKLambdaCorrPID_MAP_REF_, (real64)Epm_rpmEngSpd__LAMBDACONTROL_IMPL_Lcc_calc *
0.5, (real64)Mpc_pManiFilt__LAMBDACONTROL_IMPL_Lcc_calc * 4.0),
CharTable2_getAt_r64r64r64(_Lcc_tiTvLambdaCorrPID_MAP_REF_, (real64)Epm_rpmEngSpd__LAMBDACONTROL_IMPL_Lcc_calc *
0.5, (real64)Mpc_pManiFilt__LAMBDACONTROL_IMPL_Lcc_calc * 4.0),
CharTable2_getAt_r64r64r64(_Lcc_tiTnLambdaCorrPID_MAP_REF_, (real64)Epm_rpmEngSpd__LAMBDACONTROL_IMPL_Lcc_calc
* 0.5, (real64)Mpc_pManiFilt__LAMBDACONTROL_IMPL_Lcc_calc * 4.0), ((_tlreal64 <= 40.0) ? _tlreal64 : 40.0),
_Lcc_LambdaCorrMaxPID, _Lcc_LambdaCorrMinPID);
}
else
{
    /* If-block: sequence call #4/Else #1 */
    PID_LIMIT_AWU_IMPL_reset(_PID_Limit_AWU_REF_, _Lcc_StartValuePID);
} /* end if */
/* Lcc_calc: sequence call #5 */
_tlreal64
= (((real64)Epm_rpmEngSpd__LAMBDACONTROL_IMPL_Lcc_calc == 0.0) ? 80.0 : 80.0 /
(real64)Epm_rpmEngSpd__LAMBDACONTROL_IMPL_Lcc_calc);
INTEGRATOR_IMPL_compute(_Integrator_REF_, ((_Lcc_flgLambdacontrolActive_1) ? 0.0 :
((Lamsoni__LAMBDACONTROL_IMPL_Lcc_calc < Afr_facLambdaReq__LAMBDACONTROL_IMPL_Lcc_calc) ? -
CharTable1_getAt_r64r64(_Lcc_EngSpd2facIntegrator_CUR_REF_, (real64)Epm_rpmEngSpd__LAMBDACONTROL_IMPL_Lcc_calc *
0.5) : CharTable1_getAt_r64r64(_Lcc_EngSpd2facIntegrator_CUR_REF_,
(real64)Epm_rpmEngSpd__LAMBDACONTROL_IMPL_Lcc_calc * 0.5))), _Lcc_flgLambdacontrolActive_1, ((_tlreal64 <=
40.0) ? _tlreal64 : 40.0), _Lcc_LambdaCorrMaxTPC, _Lcc_LambdaCorrMinTPC);
/* Lcc_calc: sequence call #6 */
_tlreal64
= (((!_Lcc_flgLambdacontrolActive_1) ? 2048.0 : (((_Lcc_flgSwitch2PIDController) ?
PID_LIMIT_AWU_IMPL_out(_PID_Limit_AWU_REF_) : INTEGRATOR_IMPL_out(_Integrator_REF_))) * 2048.0);
_tlreal64 = ((_tlreal64 < 0.0) ? _tlreal64 - 0.5 : _tlreal64 + 0.5);
/* assignment to Lcc_facLambdaCorr: min=0, max=4096, hex=2048phys+0, limit=(maxBitLength: true, assign:
true), zero incl.=true */
Lcc_facLambdaCorr__LAMBDACONTROL_IMPL_Lcc_calc
= ((_tlreal64 >= 0.0) ? ((_tlreal64 <= 4096.0) ? (uint16)_tlreal64 : 4096U) : 0U);
/* send messages implicitly */
{
    Lcc_facLambdaCorr = Lcc_facLambdaCorr__LAMBDACONTROL_IMPL_Lcc_calc;
}
}
/* -----
* END: DEFINITION OF PROCESS 'LAMBDACONTROL_IMPL_Lcc_calc'
*****
#endif
/* END: ASCET REGION "Process Definition 'Lcc_calc'" */

/* *****
* END: FUNCTIONS OF COMPONENT
*****
/* END: ASCET REGION "Component Functions" */

```

