Project: MSS54

Module: Idle control in moment structure

Changes:

version	Date	comment
r300	1.6.04	takeover of MSS54 project
r360	January 23, 2005	document revised
r360	21.08.05	rm: P-part of the LL controller changed to 4-quadrant operation

Table of contents

FUNCTIONALITY

The module described here describes all functions related to idle control via torque manager.

1 OVERVIEW IDLE CONTROL

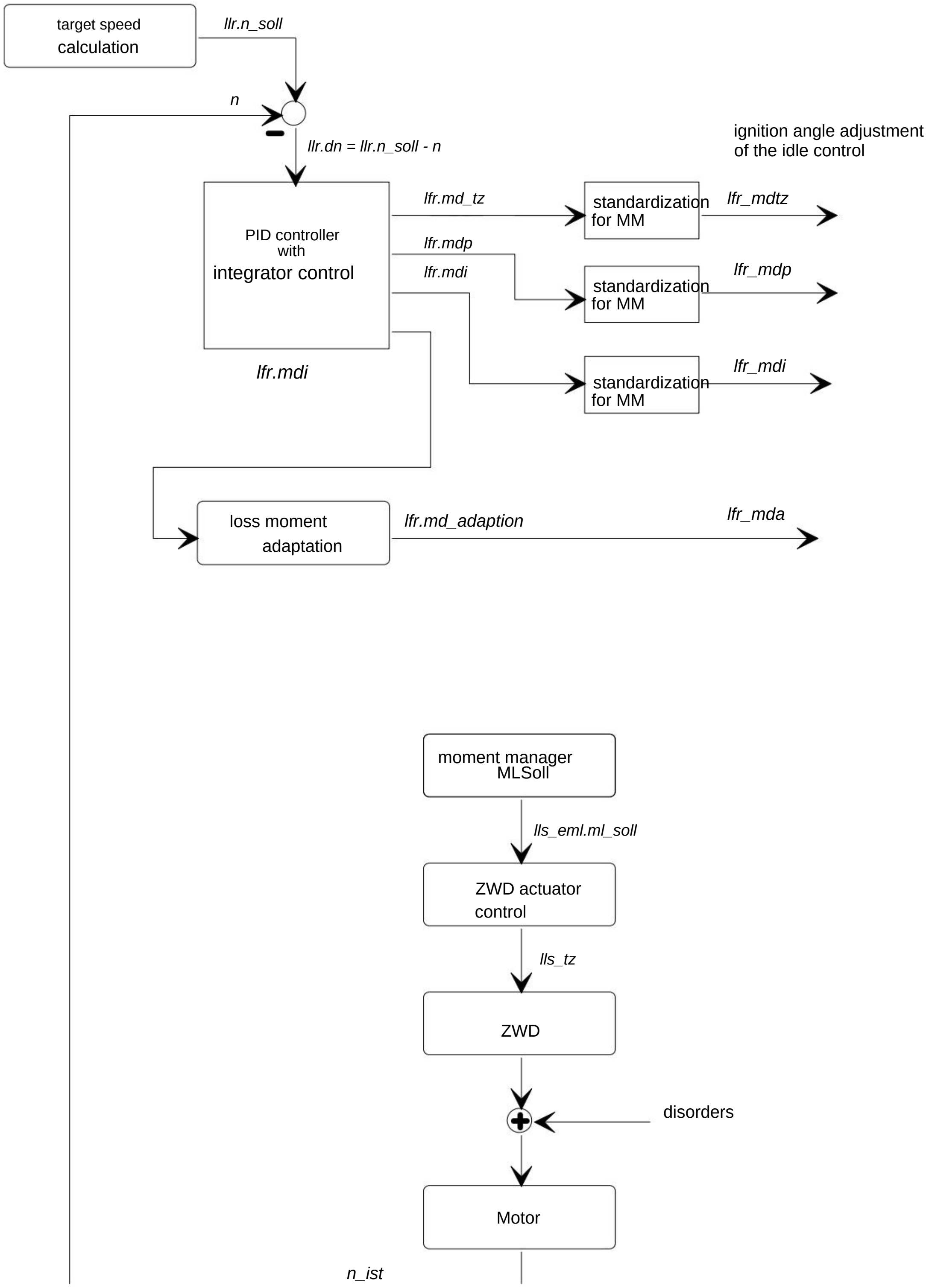
The entire idle control is shown schematically in Figure 1 - Overview of idle control - shown.

It consists of the submodules

- Target speed calculation
- idle control
- Adaptation loss torque
- ZWD control

Unless explicitly stated in the description of the sub-modules, the idle control is active in all operating states of the MSS54. The interpolation of the characteristic curves/characteristic maps with slowly changing input variables takes place in the background.
Otherwise, the idle control is calculated synchronously in 20 ms intervals.

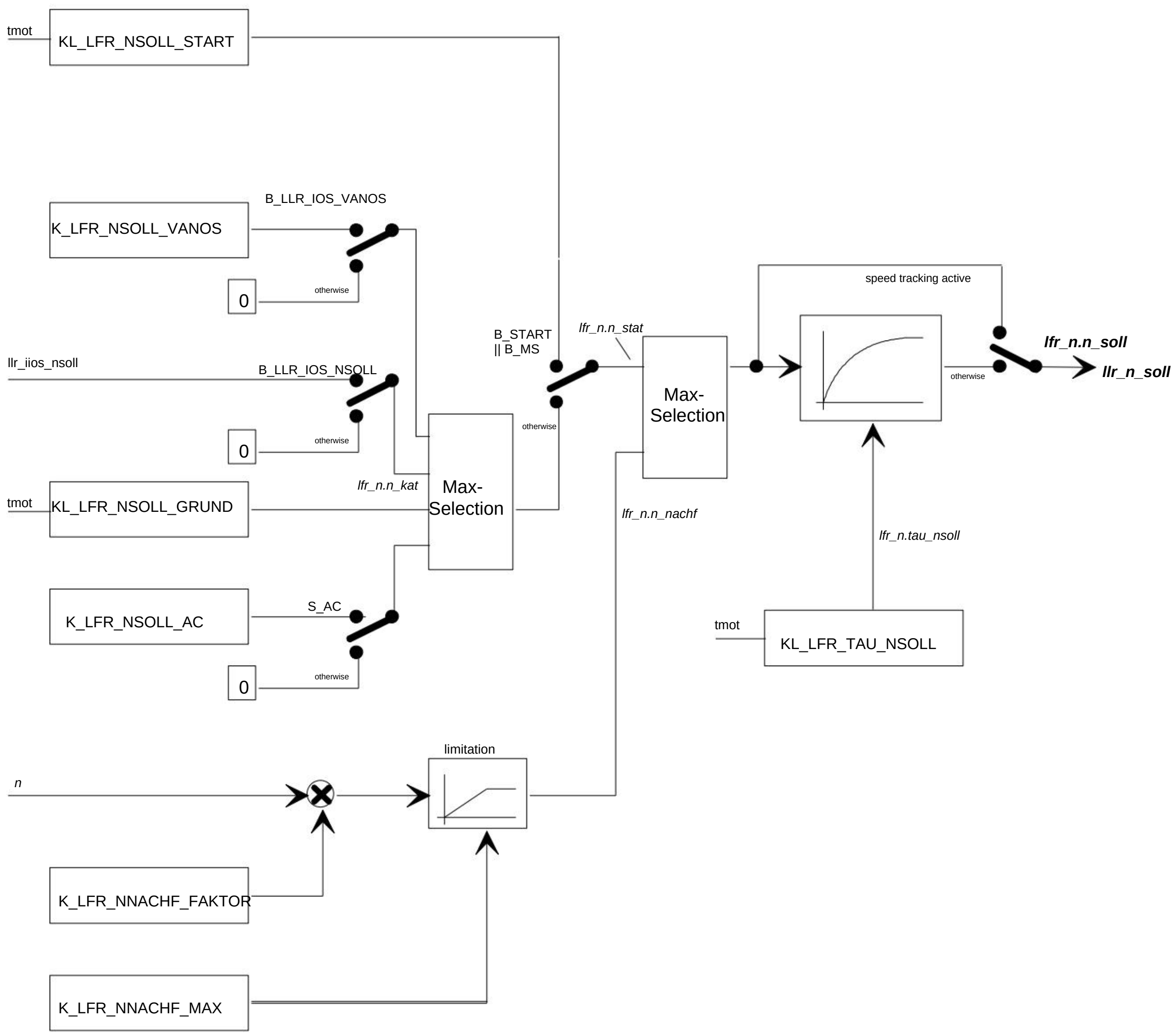
Figure 1: Overview of idle control



2 TARGET SPEED CALCULATION

The target speed is the reference variable for the PID controller of the idle speed control. Figure 2 gives an overview of the target speed calculation.

Figure 2: Calculation of the target speed



The target speed is the maximum of the stationary target speed "lfr_nstat" and the tracked target speed "lfr_nnachf".

The stationary target speed is calculated as follows:

In the operating state "Engine_standstill" or "Start"
lfr_nstat =KL_LFR_NSOLL_START ; Target speed during start = f(tmot)

In all other operating conditions
lfr_nstat =Maximum from

KL_LFR_NSOLL_GRUND ; Basic characteristic curve target speed = f(tmot)
K_LFR_NSOLL_AC The ; Target speed for air conditioning readiness
tracked target speed corresponds to the current engine speed weighted with the factor K_LFR_NNACHF_FAKTOR, whereby the factor can be between 0 and 0.997. The tracked target speed is limited to the value K_LFR_NNACHF_MAX.

If the stationary speed is greater than the tracked speed, it is filtered via a pt1 filter with the filter time constant lfr_znsoll, which is calculated from the characteristic curve KL_LFR_TAU_NSOLL. If the tracked speed is used as the target speed, this filter is bypassed.

cat heaters

The idle speed offset after starting is determined from the characteristic curve KL_LFR_N_OFFSET over the engine temperature at start. The offset remains constant for K_T_SEIT_START, after which it is ramped down to zero.

data of the target speed calculation
Description of the variables:

Name	Description	Type Resolution
lfr_n.n_stat	stationary target speed	uw 1 rpm
llr_ios_nsoll	Target speed via DS2	uw 1 rpm
lfr_n.n_soll	resulting, filtered target speed	uw 1 rpm
lfr_n.tau_nsoll	time constant for target speed filter	ub 1 rpm

Description of the application data:

name	type	Dim. x-axis		y-axis
KL_LFR_NSOLL_START	KL	3 x 1	tmot - cooling water temp.	--
KL_LFR_NSOLL_GRUND	KL	4 x 1	tmot - cooling water temp.	--
K_LFR_NSOLL_AC	K		--	--
K_LFR_NSOLL_VANOS	K		--	--
K_LFR_NNACHF_FAKTOR	K		--	--
K_LFR_NNACHF_MAX	K		--	--
KL_LFR_TAU_NSOLL	KL	1 1 1 1 4 x 1	tmot - coolant temp.	--

3 IDLE CONTROLLER

The idle speed control is carried out via torque interventions, ie the output of the controller is a torque request and no longer a filling request.

The idle controller is a PID controller (proportional, integral, differential controller), whereby the PD component is contained in the new 4-quadrant map KF_LFR_PD.

The I-controller remains structurally the same as MSS50, with the addition of falling speed and n < lfr_nsoll+K_LFR_DN_EINGEREGELT an I-component lead, which can increase the I-component of the controller once by lfr_i_auf = f(dn). The input variable of the controller is the deviation of the actual speed from the target speed.

speed difference lfr_dn = target speed - lfr_nsoll - actual speed n
=

A positive speed difference means that the engine speed is too low in relation to the target speed. A negative speed difference means that the engine speed is too high.

To support the control of the idle speed via the air supply, the LFR also intervenes in the ignition angle path using the PD component. In addition, ignition angle stabilization from the idle control can be taken into account.

3.1 PD COMPONENT OF THE IDLE CONTROLLER

The PD component is calculated from the KF_LFR_PD map and depends on the speed difference between the target and actual speed as well as the speed gradient "d_n" (1st derivative). It is calculated synchronously in 20 ms increments.

The previous D-controller as well as the associated characteristic curve KL_LFR_I_AUF and the old P-component via the characteristic curve KL_LFR_DQP_POS are no longer required!

The applicable constant K_LFR_CONTROL can be used to control the various controller components activated or deactivated:

7	6	5	4	3	2	1	0
			TZ	D	Adapt	I	P
			$10	$08	$04	$02	$01

1 - aktiv
0 - deaktiv

symbolic constants:

PID-active
PD-active
PID+Adapt.active
PID+Adapt.+TZ
I-active
I+Adapt.active
I+Adapt.+TZ active

Application note P(D) controller:

The P component is calibrated after the I component is calibrated. At speeds lower than the idle target speed (lfr.dn>0), the torque is increased (if LFR_MDPD>0); at speeds higher than the idle target speed, the torque is reduced.

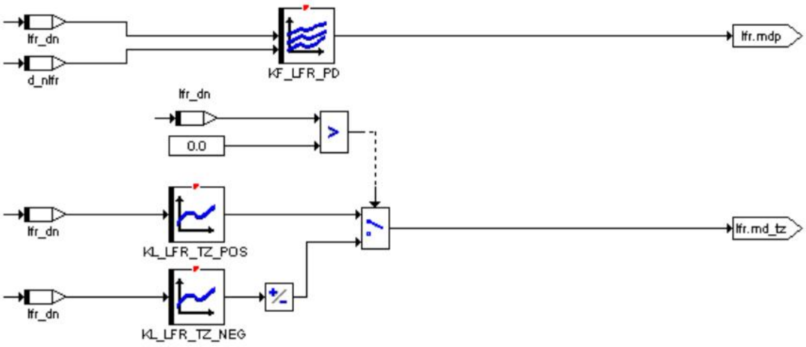
Figure 3: Overview of idle control

Machine Translated by Google

############

\# P-Regler #

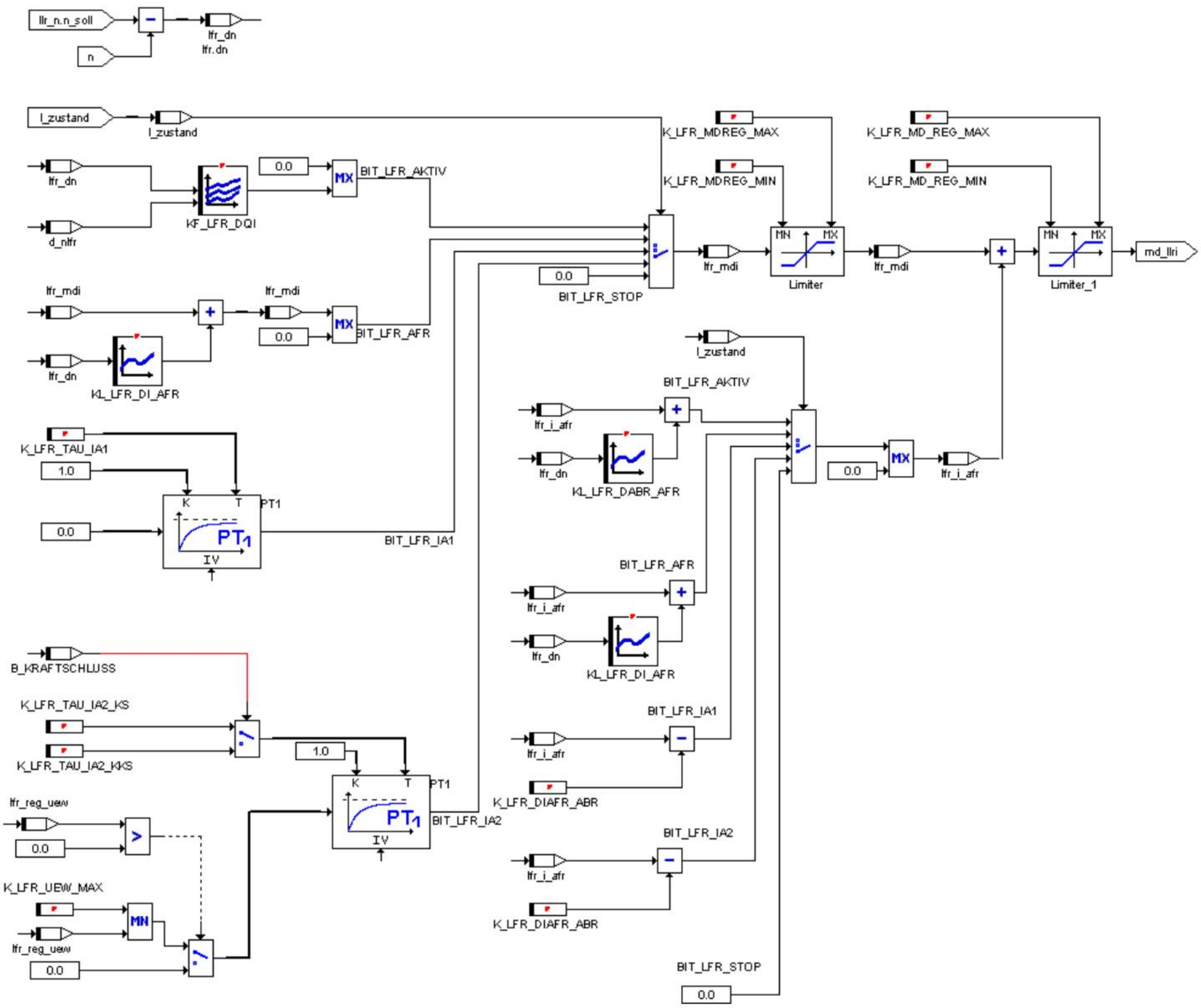
############



############

\# I-Regler #

############



3.2 I-PART OF THE IDLE SPEED CONTROLLER

When calculating the I component, a distinction must be made between different operating states of the I controller. These are the states:

- I-controller stop:	B_LFR_STOP The I component is set to zero.
- Idle control:	B_LFR The I controller is active lfr_mdi t = lfr_mdi t-1 + dmdi additionally lfr_i_afr = lfr_i_afr-KL_DABR_AFR throttled down to zero
- Starting control:	B_AFR The part from the start-up control: lfr_i_afr = lfr_i_afr + KL_LFR_DI_AFR (d_nlfr) The I-controller portion is for n<lfr_nsoll frozen otherwise throttled down lfr_mdi = lfr_mdi - f(dn)
- Integrator control range 1:	B_IA1 The I component is set to zero via a pt1 filter with the time constant K_LFR_TAU_IA1. The part from the start-up control is regulated: lfr_i_afr = lfr_i_afr - K_LFR_DIAFR_ABR
- Integrator control area 2:	B_IA2 The I component is guided to the maximum of lfr_uew and zero via a pt1 filter with the time constant K_LFR_TAU_IA2. The variable lfr_reg.uew is the I component at the time of the state transition from idle control to start-up control. The time constant depends on S_KS: set at S_KS (power connection) K_LFR_TAU_IA2 = K_LFR_TAU_IA2_KS not set at S_KS (no frictional connection) K_LFR_TAU_IA2 = K_LFR_TAU_IA2_KKS The part from the start-up control is regulated: lfr_i_afr = lfr_i_afr - K_LFR_DIAFR_ABR

Within the state B_LFR there are special cases:

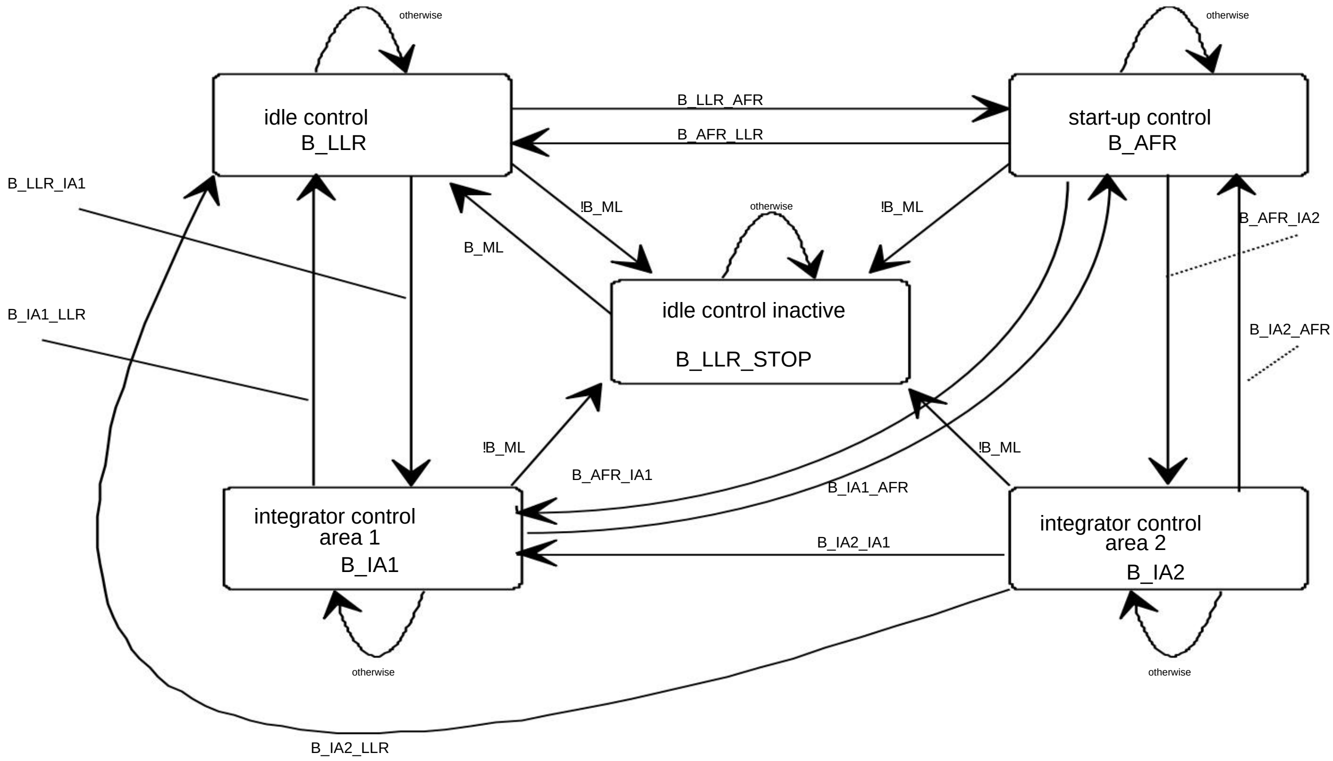
- the idle speed is undercut by the value K_LFR_NDIFF_RESET and is too
  If the I component is negative at this point, it is immediately set to zero
  (B_LFR_RESET).

- the filling rf falls below the minimum filling threshold lfr_reg.rf_min, calculated from
  KL_LFR_RF_MIN = f(tmot), a further reduction of the I component is blocked
  (B_LFR_NEGSTOP).

- to prevent the speed from diving after the engine starts,
  If B_MD_NACHSTART is set, a reduction of the I component is also blocked.

Figure 4 shows the state diagram and the transition conditions for the idle controller.

Figure 4: Idle speed controller state diagram



State transitions of the idle controller:

Transition idle control ÿ start-up control

B_LFR_AFR = (B_TL
or B_VL
or (B_LL and S_GANG))
and !B_SA

Transition from start-up control to idle control

B_AFR_LFR = B_LL
and !S_GANG)
and !B_SA

Transition start-up control ÿ Integrator de-control area 1

B_AFR_IA1 = !B_ML
and n > lfr.n_soll + K_LFR_DN_HYS

Transition start-up control ÿ Integrator de-control area 2

B_AFR_IA2 = !B_ML
and n > lfr.n_soll + K_LFR_DN_HYS

Transition Integrator de-control area 1ÿ Start-up control

B_IA1_AFR = B_ML
and n < lfr.n_soll + K_LFR_DN_HYS

Transition Integrator de-control area 2ÿ Start-up control

B_IA2_AFR = and B_ML
n < lfr.n_soll + K_LFR_DN_HYS

Transition Integrator Control Area 2 ÿ Area 1

B_IA2_IA1 = B_ML
and B_SA

Transition Integrator de-control area 2 ÿ Idle control

B_IA2_LFR = B_LL
and !S_GANG
and !B_SA

Transition Integrator de-control Range 1 ÿ Idle control

B_IA1_LFR = B_LL
and !S_GANG
and !B_SA

Transition idle control ÿ Integrator de-control area 1

B_LFR_IA1 = B_SA

The I component can be set to zero for test or application purposes using BIT_I_REGLER_ON (bit 1) in the constant K_LFR_CONTROL.

3.3 IGNITION ANGLE STABILIZATION OF THE IDLE REGULATOR

The TZ component is calculated from the characteristic curve KL_LFR_TZ_POS/NEG and depends on the amount of the speed difference between the target and actual speed. It is calculated synchronously in 20 ms. It provides a torque correction lfr.md_tz. proportional to the detected speed gradient.

The TZ component of the idle controller is active under the following conditions:

BIT_TZ_REGLER_ON (BIT 4) set in K_LFR_CONTROL

and Operating state = Motor_running (B_ML)

and (S_GANG = no power transmission

or S_GANG = power transmission and v <= K_LFR_V_MAX)

When calculating the TZ share, two cases must be distinguished:

- Speed too low (lfr_dn > 0)

lfr.md_tz = f (KL_LFR_TZ_POS)

- Speed too high (lfr_dn < 0)

Engine state idling:

lfr.md_tz = f (KL_LFR_TZ_NEG)

otherwise lfr.md_tz = 0

idle speed controller data

Description of the variables:

name	**Description** Speed	**Type Resolution**
lfr_dn	difference P-component	sw 1 rpm
lfr_mdp	I-	sw [1/10*8] Nm
lfr_mdi		sw [1/10*16] Nm
lfr_i_afr	component Component from the AFR	sw [1/10*16] Nm
lfr_i_auf	control I-component derivative with rapidly falling	sw [1/10*16] Nm
md_lfri	n I-component of the LL controller for torque m.	sw [1/10] Nm
md_lfrp	PD component of the LL controller for torque m.	sw [1/10] Nm
md_lfr_tz	ZW--part of the LL controller for torque m.	sw [1/10] Nm
lfr_zustand	status information LFR Bit 0: B_LFR_STOP 1: B_LFR 2: B_ALFR 3: B_IA1 4: B_IA2	uc --
lfr_flags	internal flags of the LFR Bit 0: Flag for starting air mass (Qvs) 1: Time monitoring B_KO active 2: B_KO was last active 4: B_LFR_NEGSTOP 5: B_LFR_HALT	uc --
lfr_uew	I-component at the end of the idle control	sw 1/256 kg/h
lfr_tlmin	minimum load for negative stop	uw 1 µs/rev.

Description of the application data:

name	**type**	**Dim. x-axis**		**y-axis**
KL_LFR_DMDP_POS	KL	16 x 1	lfr.dn - speed deviation	--
KL_LFR_DI_AFR	**KL**	**16 x 1 lfr.dn -** speed deviation. **16 x 1 lfr.dn -**		**--**
KL_LFR_TZ_POS	KL	**speed deviation. 16 x 1 lfr.dn - speed**		--
KL_LFR_DABR_AFR	**KL**	**deviation.**	**15 x 8 PD component** 16 x 1 lfr.dn	**--**
KF_LFR_PD	**KF**	- speed deviation. 15 x 8		**--**
KL_LFR_TZ_NEG	KL	lfr.dn - speed deviation.		--
KF_LFR_DMDI	KF			d_n40 speed gradient
K_LFR_MDREG_MIN	K	1	**Min. limit LFR-I share**	--
K_LFR_MDREG_MAX	K		**max. limit. LFR-I share**	--
K_LFR_MD_REG_MIN	**K**	1 1	**min. limit. LFR-I +AFR portion**	
K_LFR_MD_REG_MAX	**K**		**Max.Limit LFR-I +AFR-Share --**	
K_LFR_MDAFR_MAX	**K**		**Max. limit AFR share**	**--**
KL_LFR_RF_MIN	KL	1 1 4 x	tmot - cooling water temp.	--
K_LFR_NDIFF_RESET	K		--	--
K_LFR_DIAFR_ABR	**K**	1 1 1	**--**	**--**
K_LFR_DN_EINGEREGELT	**K**	**1**	**--**	**--**
K_LFR_UEW_MIN	K		--	--
K_LFR_TAU_IA1	K	1 1	--	--
K_LFR_TAU_IA2_KS	K	1	--	--
K_LFR_TAU_IA2_KKS	K	1	--	--
K_LFR_V_MAX	K	1	--	--
KL_LFR_TZ_NEG	KL	16x1	lfr.dn - speed deviation lfr.dn -	--
KL_LFR_TZ_POS	KL	16x1	speed deviation	--

4 adaptation loss moment

The demand adaptation learns the different friction losses of the engine + auxiliary units system. For the adaptation, only the air conditioning compressor is included in the auxiliary units. Since the engine is sometimes operated without the air conditioning compressor switched on, the adaptation must take the different load conditions into account.

When idling without the air conditioning compressor, an adaptation is carried out which only learns the change in the load torque of the engine (base load change of the engine corresponds to a change in the internal friction of the engine due to, for example, wear or different oil viscosity).

If the air conditioning compressor is switched on, the base load change remains the same; new load changes are mainly caused by the air conditioning compressor. Therefore, a further factor is now adapted in addition to the base load change.

This adaptation will usually be significantly faster than the base load adaptation, since the compressor load in controlled compressors (depending on the quality of the load feedback of the compressor) changes more quickly than the engine base load (determined by mechanical friction and oil viscosity).

4.1 adaptation conditions

To activate demand adaptation, the following conditions must be met:

```
B_LFRA      =      B_LFR                        ; Idle control active state
                                                  (see LFR state machine)
            and tmot > K_LFR_TMOT_ADAPT ; engine temperature greater than threshold
            and !B_TMOT_FEHLER and !            ; error-free tmot recording
            B_LFR_IBEGR                         ; Integrator is not in a
                                                  limitation
```

In the following documentation, these conditions are summarized in the condition B_LFRA (LFR demand adaptation).

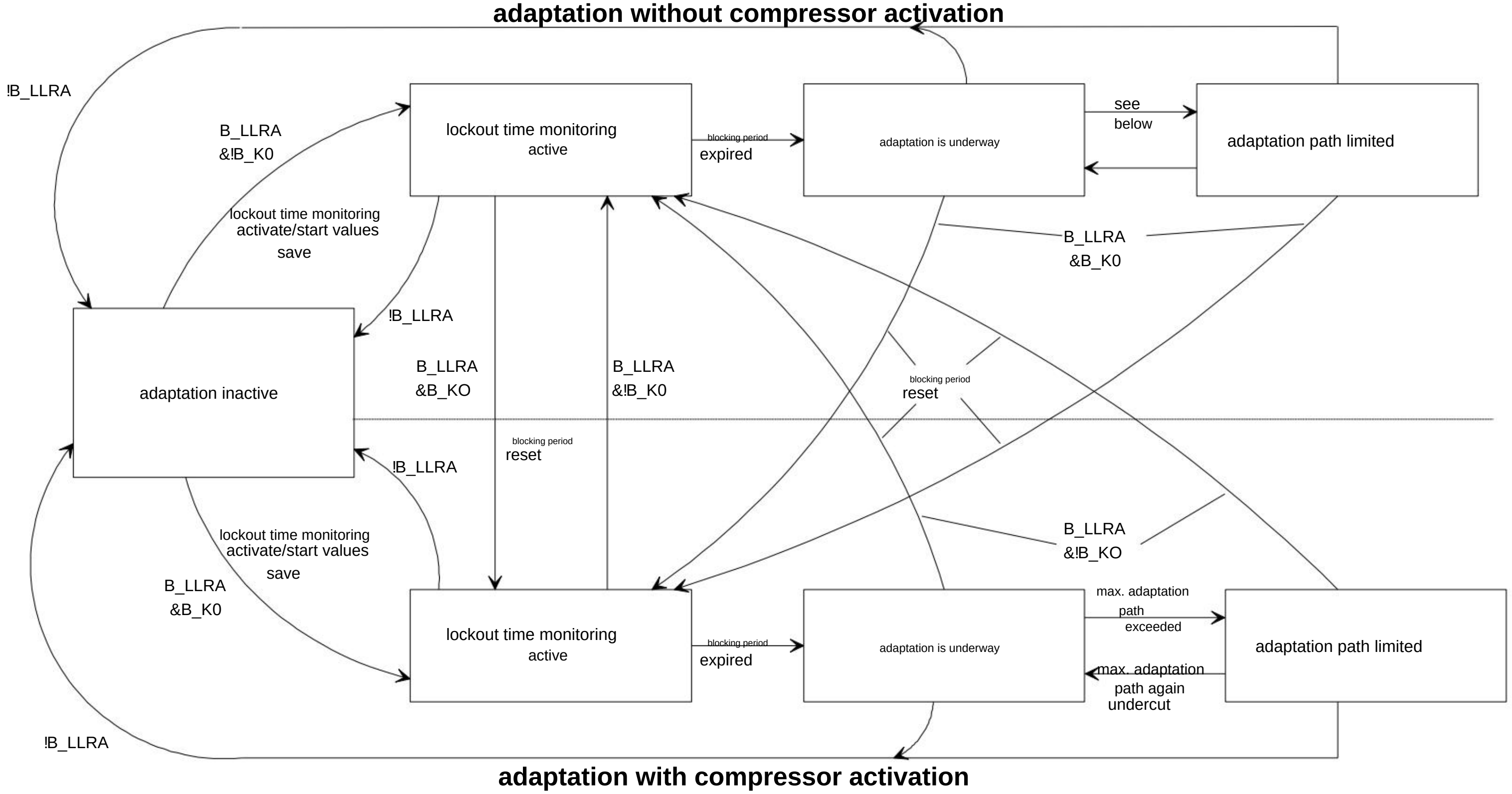
During the development and test phase, the complete demand adaptation (B_LFRA_ENABLED) can be switched off via bit 2 in the control byte K_LFR_CONTROL.
All adaptation values are then equal to zero.

B_LFRA_ENABLE = 1	demand adaptation released
B_LFRA_ENABLE = 0	demand adaptation switched off

4.2 States of demand adaptation

The control of demand adaptation can be described as a state machine with seven states.

Figure 5: State machine of demand adaptation



adaptation inactive

Condition: B_LFRA not fulfilled

Mark: lla_flags = 0 (inactive)
lla_kflags = 0 (inactive)

Adaptation values: lla_mdadapt (t) = lla_mdadapt (t - 20 ms)
lla_kmdadapt (t) = lla_kmdadapt (t - 20 ms)

Lock time monitoring for LFRA without K0 active

Condition: B_LFRA fulfilled
and !B_KO
and lla_timer !=0
(Lockout period not yet expired

Mark: lla_flags = 1 (lock time)
lla_ko_flags = 0 (inactive)

Adaptation values: lfra_mdadapt (t) = lla_mdadapt (t - 20 ms)
lla_kmdadapt (t) = lla_kmdadapt (t - 20 ms)

Adaptation running (without K0)

Condition: B_LFRA
and !B_K0
and lla_timer == 0 (lock time expired)
and | lla_mdadapt - lla_mdstart | ÿ K_LFR_DMDADAPT_MAX
(adaptation path not limited)

Mark: lla_flags = 3 (adapted)
lla_kflags = 0 (inactive)

Adaptation values: lla_mdadapt (t) = lla_mdadapt (t - 20 ms) + (lfr_mdi(t - 20 ms) + K_LFR_MDADAPT_OFFSET) * K_LFR_TAU_ADAPT
(without taking into account any limitation)
lla_kmdadapt (t) = lla_kmdadapt (t - 20 ms)

adaptation value (without K0) limited

Condition: B_LFRA
and !B_KO
and lla_timer == 0
and | lla_mdadapt - lla_mdstart | > K_LFR_DMDADAPT_MAX
(adaptation path limited)

Mark: lla_flags = 7 (limited)
lla_kflags = 0 (inactive)

Adaptation values: lla_mdadapt (t) = lla_mdstart ÿ K_LFR_DMDADAPT_MAX
lla_kmdadapt (t) = lla_kmdadapt (t - 20 ms)

Note: If the difference between the calculated adaptation value and the starting value at the beginning of the adaptation phase becomes smaller than the maximum adaptation path, the system switches back to the "Adaptation in progress" state.

Lock time monitoring for LFRA with K0 active

Condition: B_LFRA fulfilled
and !B_KO
and lla_timer !=0
(Lockout period not yet expired

Mark: lla_flags = 1 (lock time)
lla_kflags = 0 (inactive)

Adaptation values: lla_mdadapt (t) = lla_mdadapt (t - 20 ms)
lla_kmdadapt (t) = lla_kmdadapt (t - 20 ms)

Adaptation running (with K0)

Condition: B_LFRA
and !B_K0
and lla_timer == 0 (lock time expired)
and | lla_mdadapt - lla_mdstart | ÿ K_LFR_DMDADAPT_MAX
(adaptation path not limited)

Mark: lla_flags = 3 (adapted)
lla_kflags = 0 (inactive)

Adaptation values: lla_ko_mdadapt (t) = lla_kmdadapt (t - 20 ms) +
K_LFR_MDADAPT_OFFSET) * K_LFR_TAU_ADAPT
(without taking into account any limitation)
lla_mdadapt (t) = lla_mdadapt (t - 20 ms) + (lfr.mdi(t - 20 ms)

adaptation value (with K0) limited

Condition: B_LFRA
and !B_KO
and lla_timer == 0
and | lla_mdadapt - lla_mdstart | > K_LFR_DMDADAPT_MAX
(adaptation path limited)

Mark: lla_flags = 7 (limited)
lla_kflags = 0 (inactive)

Adaptation values: lla_kmdadapt (t) = lla_kmdadapt (t - 20 ms)

lla_mdadapt (t) = lla_mdstart ÿ K_LFR_DMDADAPT_MAX

For all states applies

Initial value of demand adaptation:

lfr_mdadaption (t) = lfra_mdadapt(t) = , if !B_KO

lla_mdadapt(t)

+ lla_kmdadapt (t) , if B_KO

Correction of the integrator component lfr.mdi of the idle control

when compressor activation is inactive

lfr_mdi (t) = lfr_mdi (t)

- (lla_mdadapt (t) - lla_mdadapt (t - 20 ms))

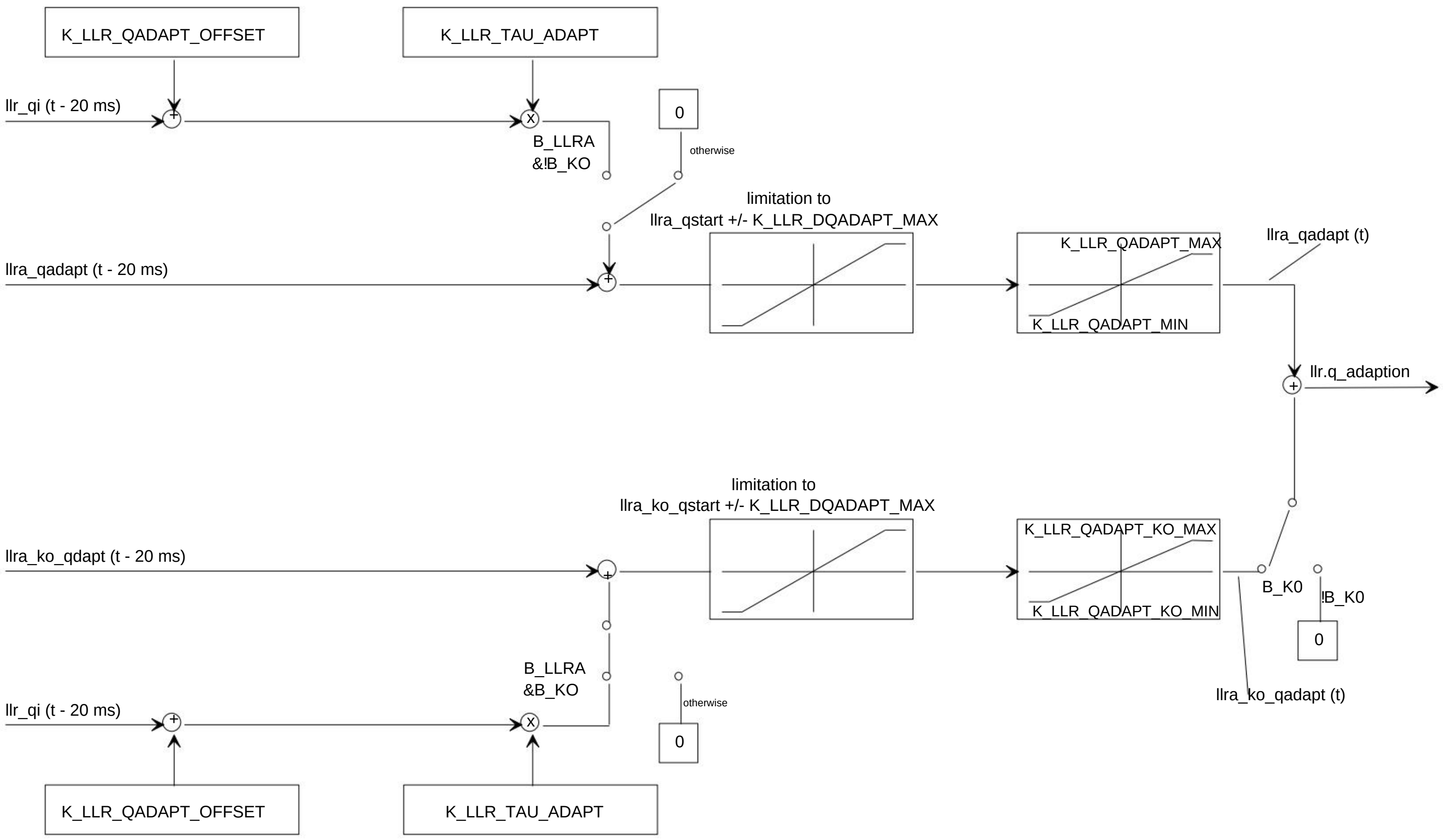
when compressor activation is active

lfr_mdi (t) = lfr_mdi (t)

- (lla_kmdadapt (t) - lla_kmdadapt (t - 20 ms))

4.3 Calculation steps of demand adaptation

Figure 6: Block diagram of the LFR adaptation



integrator

If the active condition for the adaptation integrator is fulfilled (B_LFRA fulfilled and blocking time expired), a new integration cut for the current adaptation mode (B_K0 or !B_K0) is calculated synchronously every 20 ms:

lmdadapt (t) = mdadapt (t-20 ms)
+ (lfr_mdi (t - 20 ms) + K_LFR_MDADAPT_OFFSET)
K_LFR_TAU_ADAPT

limiting the adaptation path

A maximum adaptation path of ÿ K_LFR_DMDADAPT_MAX is possible per adaptation phase. An adaptation phase begins when the condition B_LFRA = fulfilled is recognized and ends as soon as this condition is no longer fulfilled. A change in the condition B_K0 or a retriggering of the blocking time does not lead to a new adaptation phase.

At the beginning of the adaptation phase, the two adaptation values lfra_mdadapt and lla_kmdadapt are saved in the variables lla_mdstart and lla_kmdstart. During the adaptation phase, the current adaptation value is then set to the value ..._mdstart ÿ
K_LFR_DMDADAPT_MAX limited.

limiting the adaptation values

The resulting adaptation value for inactive compressor activation is set to -
K_LFR_DMDADAPT_MAX, K_LFR_DMDADAPT_MIN which for active compressor switching is limited to the values K_LFR_MDADAPT_K0_MAX and K_LFR_MDADAPT_K0_MIN.

initial value of demand adaptation

The output value of the adaptation lfr_mdadaption, which is added to the pre-control value of the idle control, is always calculated - regardless of the condition B_LFRA and is composed as follows:

if B_K0 inactive
lfr_mdadaption (t) = lla_mdadapt (t)

if B_K0 active
lfr_mdadaption(t) = lla_mdadapt(t) + lla_kmdadapt(t)

Correction of the integration component of the idle controller

The LFR demand adaptation must not change the air specification of the idle control lfr_mdsoll, but only transfer a correction offset from the I component of the idle controller lfr_mdi to the adaptation value lfr_mdadaption. This means that with each change in the adaptation value, the I component lfr_mdi must be corrected by this amount.

if B_K0 inactive
lfr_mdi (t) = lfr_mdi (t)
- (lla_mdadapt (t) - lla_mdadapt (t - 20 ms))

if B_K0 active
ll_mdi (t) = lfr_mdi (t)
- (lla_kadapt (t) - lla_kmdadapt (t - 20 ms))

4.4 Data of demand adaptation

Description of the variables:

name	Description	type	resolution
lla_timer	remaining adaptation blocking time	uw	0.02 sec.
lla.mdadapt	Value of the adaptation integrator without compressor activation	sw	1/256 kg/h
lla.mdstart	Value of the adaptation integrator at the beginning of a new adaptation phase (without K0)	sw	1/256 kg/h
lla.flags	Flags for adaptation without compressor activation Value 0: Adaptation inactive Value 1: Lockout time running Value 3: adapted Value 7: Adaptation path limited	uc	--
lla_kmdadapt	Value of the adaptation integrator with compressor activation	sw	1/256 kg/h
lla_kmdstart	Value of the adaptation integrator at the beginning of a new adaptation phase (with K0)	sw	1/256 kg/h
lla_kflags	Flags for adaptation with compressor activation Value 0: Adaptation inactive Value 1: Lockout time running Value 3: adapted Value 7: Adaptation path limited	uc	--

Description of the application data:

name	type	Meaning
K_LFR_MDADAPT_OFFSET	FW	adaptation offset for integrator component
K_LFR_TAU_ADAPT	FW	time constant for demand adaptation
K_LFR_DMDADAPT_MAX	FW	max. adaptation path per adaptation phase
K_LFR_T_ADAPT	FW	adaptation blocking time
K_LFR_MDADAPT_MIN	FW	lower adaptation value limit (without K0)
K_LFR_MDADAPT_MAX	FW	upper adaptation value limit (without K0)
K_LFR_MDADAPT_KO_MIN	FW	lower adaptation value limit (with K0)
K_LFR_MDADAPT_KO_MAX	FW	upper adaptation value limit (with K0)
K_LFR_TMOT_ADAPT	FW	temperature threshold for demand adaptation

4.5 Non-volatile storage

In the control unit's run-on phase, the current values

lla_mdadapt
and lla_kmdadapt

the demand adaptation is stored non-volatilely in the E²PROM of the control unit

In the initialization phase, the current adaptation values are preset with the stored values. If data is lost from the E²PROM, the adaptation values are preset with the value zero.