Project: MSS54

Module: Idle control without moment structure

Table of contents

3. idle control

This chapter describes the calculation of the target air mass flow rate for idle control and the conversion of the Q_soll specification into a duty cycle for idle actuator control.

A two-winding rotary actuator ZWD is used. The opening and closing coils are controlled with an inverse duty cycle at a fixed frequency of 100 Hz.
The calculation of the duty cycle of the ZWD control refers to the coil to be opened.

3.1 Overview of idle control

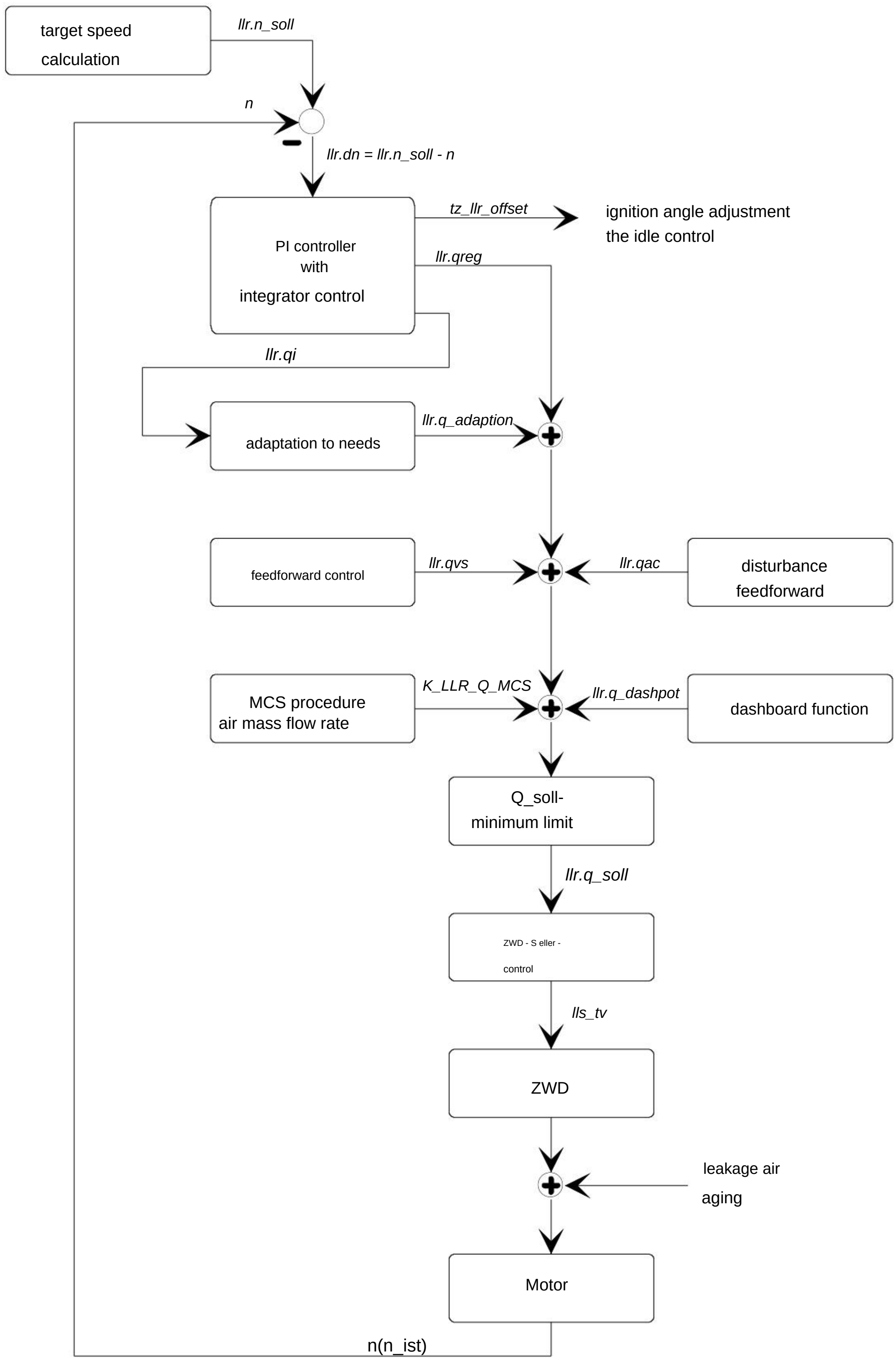
The entire idle control is shown schematically in Figure 3.1 - Overview of idle control - shown.

It consists of the submodules

- feedforward control
- disturbance feedforward control
- Dashpot function
- Target speed calculation
- idle control
- adaptation to needs
- Q_target calculation
- ZWD control

Unless explicitly stated in the description of the sub-modules, the idle control is active in all operating states of the MSS50. 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 3.1: **Overview of idle control**



3.2 Pre-control

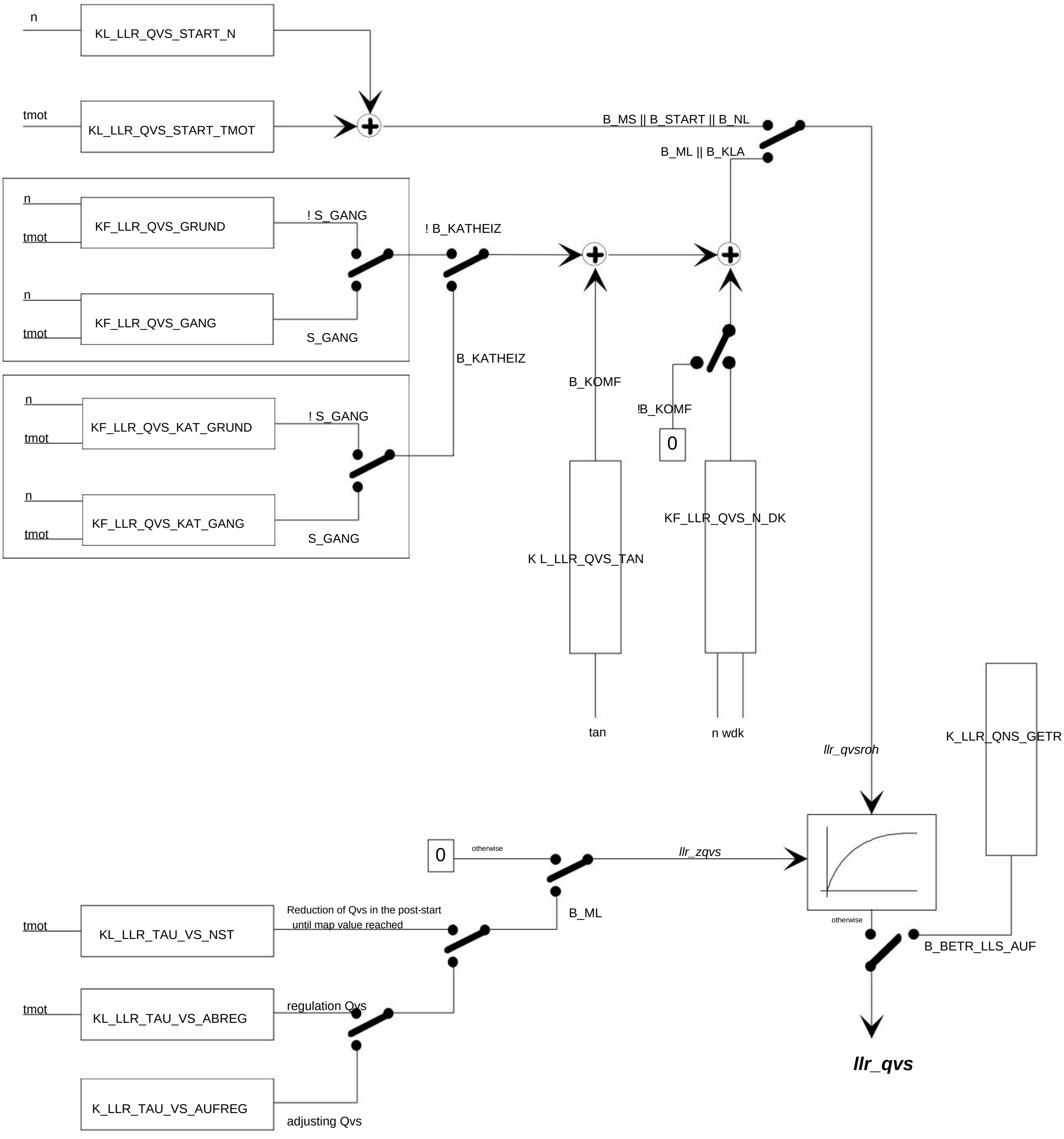
The feedforward control is active in every operating state of the MSS50 and calculates a base value for the air mass flow rate of the idle control.

The composition of the underlying "llr_qvsroh" differs depending on the Operating states and the conditions B_KRAFTS (power closure) and B_KATH_AKTIV (Cat heating function active)

3.2.1 Pre-control at "Engine_standstill" or "Start" or "After-run"

3.2.2 Pre-control at "engine_running" or "terminal 15_off"

Figure 3.4: **idle speed control pre-control**



3.2.3 Filtering the input tax value:

The subsequent filter for the pre-control value has a pt1-like behavior. The filter time constant llr_zqvs is applicable and differs for four ranges:

while engine is stopped or starting: llr_zqvs = 0 (unfiltered)

in the post-start until the map value is reached: llr_zqvs = KL_LLR_TAU_VS_NS
= f(tmot)

thereafter:

if Qvs is to be regulated: llr_zqvs = KL_LLR_TAU_VS_ABR
= f(tmot)

if Qvs is to be regulated: llr_zqvs = K_LLR_TAU_VS_AUFREG

The filter time constant is standardized so that after one tau in a jump of the input value, starting from the value zero, approx. 65%, and after five tau approx. 99% of the input value is reached.

3.2.7 Gearbox intervention in LLR feedforward control

If the condition B_GETR_LLS_AUF (ASG downshifts in overrun mode) is met, the pilot control value of the idle control is switched to the value K_LLR_QVS_GETR. The pilot control filter is bridged in this case.

3.2.8 Pre-control data

Description of the feedforward control variables:

name	Description	Type	Resolution
llr_zqvs	current time constant for Qvs filter unfiltered	uc 5.12	sec / x
llr_qvsroh	value of the VS filtered value	uw 1/256	kg/h
llr_qvs	of the VS	uw 1/256	kg/h

Description of the application data:

name	type	Dim.	x-axis	y-axis
KL_LLR_QVS_START_N	KL	3 x 1	n - speed tmot	--
KL_LLR_QVS_START_TMOT	KL	4 x 1	- cooling water temp.	--
KF_LLR_QVS_KAT_GRUND	KF	6 x 4	n - speed	tmot - cooling water temp.
KF_LLR_QVS_KAT_GANG	KF	6 x 4	n - speed	tmot - cooling water temp.
KL_LLR_QVS_TAN	KL	4 x 1	tan - intake air temperature 10 x	--
KF_LLR_QVS_N_DK	KF	8	n - speed	wdk_adapt - DK angle related to LL
KL_LLR_TAU_VS_NST	KL	3 x 1	tmot - cooling water temp.	--
KL_LLR_TAU_VS_AUFREG	KL	3 x 1	tmot - cooling water temp.	--

K_LLR_TAU_VS_ABREG	K		--	--
K_LLR_QVS_GETR	K	1 1	--	--

3.3 Disturbance control air conditioning system

The air conditioning disturbance compensation has the task of compensating the load change reactions caused by the activation of the air conditioning compressor by means of an increased air supply.

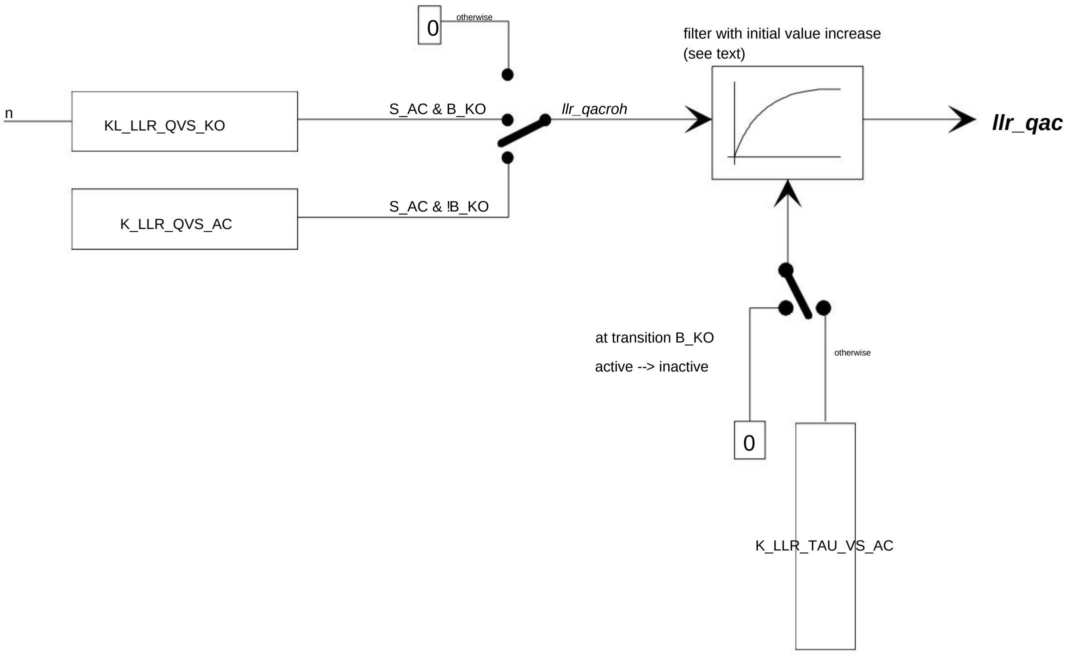
The additional air mass flow rate of the disturbance variable compensation is composed as follows:
together:

llr_qacroh = K_LLR_QVS_AC ; Pre-control value for air conditioning readiness (S_AC = active)

llr_qacroh = KL_LLR_QVS_KO ; Correction offset when compressor is switched on (B_KO = active) = f(n)

llr_qacroh = 0 ; otherwise no disturbance compensation is active

Figure 3.3: **disturbance control air conditioning**



3.3.1 Filtering of the disturbance feedforward

The result of the disturbance feedforward is, analogous to the feedforward control, by means of a pt1 similar filter element. The filter time constant is K_LLR_TAU_QAC.

If the air conditioning compressor is switched on within the time period K_LLR_T_AC after the air conditioning readiness has been detected (S_AC: inactive ÿ active), an initial filter increase is activated. This means that the filter output value is immediately set to the value K_LLR_DQKO. Switching on the air conditioning compressor outside of this time period does not result in an initial filter increase. The remaining active time for the initial value increase is stored in the variable "llr_tdqko".

When changing from active to inactive air conditioning compressor activation, the new filter input value is adopted unfiltered.

3.3.2 Data of the disturbance control

Description of the variables:

name	Description	Type	Resolution
llr_tdqko	Timer for time monitoring of the filter initial value increase	uc	0.02 sec
llr_qacroh	unfiltered value of the climate control	uw	1/256 kg/h
llr_qvs	filtered value of the air conditioning control	uw	1/256 kg/h

Description of the application data:

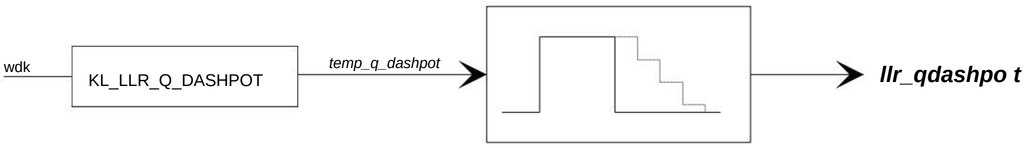
name	type	Dim.	x-axis	y-axis
KL_LLR_QVS_KO	KL	4 x 1	n - speed	--
K_LLR_QVS_AC	K	1	--	--
K_LLR_TAU_VS_AC	K	1	--	--
K_LLR_DQKO	K	1	--	--
K_LLR_T_AC	K	1	--	--

3.4 Dashpot function

The Dashpot function is designed to ensure a slower torque reduction during closing the throttle valve.

For this purpose, the dashpot function calculates an offset for the target specification of the filling (characteristic curve, which depends on the throttle valve position control KL_LLR_Q_DASHPOT). Increases in this dashpot offset are passed on without delay. If the offset value is reduced, however, a change limitation with the applicable change rate K_LLR_DQ_DASHPOT comes into effect.

Figure 3.4: **Dashpot function of the idle control**



temp_q_dashpot increases: llr.q_dashpot follows immediately
temp_q_daspot falls: llr.q_dashpot with cut-off speed K_LLR_DQ_DASHPOT

data from the Dashpot function

Description of the variables:

name	Description	Type	Resolution
llr_qdashpot	Q-Offset of the Dashpot function	uw	1/256 kg/h

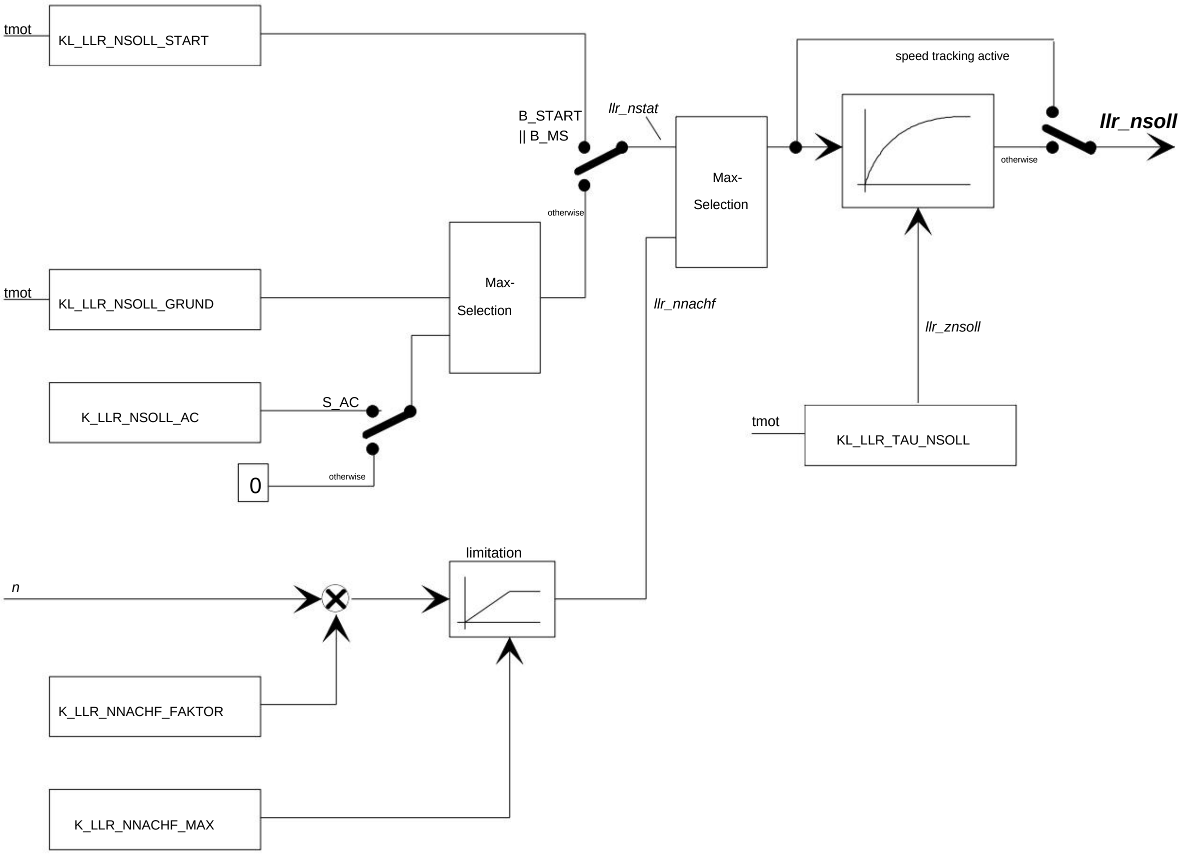
Description of the application data:

name	type	Dim.	x-axis	y-axis
KL_LLR_Q_DASHPOT	KL	6 x 1	wdk_adapt - LL related DK-position	--
K_LLR_DQ_DASHPOT	K	1	--	--

3.5 Target speed calculation

The target speed is the reference variable for the PI controller of the idle speed control. Figure 3.5 gives an overview of the target speed calculation.

Figure 3.5: **Calculation of the target speed**



The target speed is the maximum of the stationary target speed "llr_nstat" and the tracked target speed "llr_nnachf".

The stationary target speed is calculated as follows:

<u>In the operating state "Engine standstill" or "Start"</u>

llr_nstat =KL_LLR_NSOLL_START ;Target speed during start = f(tmot)

<u>In all other operating conditions</u>

llr_nstat =Maximum from

KL_LLR_NSOLL_GRUND ; Basic characteristic curve target speed = f(tmot)
K_LLR_NSOLL_AC ; Target speed for air conditioning readiness

The tracked target speed corresponds to the current engine speed weighted with the factor K_LLR_NNACHF_FAKTOR, whereby the factor can be between 0 and 0.997. The tracked target speed is limited to the value K_LLR_NNACHF_MAX.

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

data of the target speed calculation

Description of the variables:

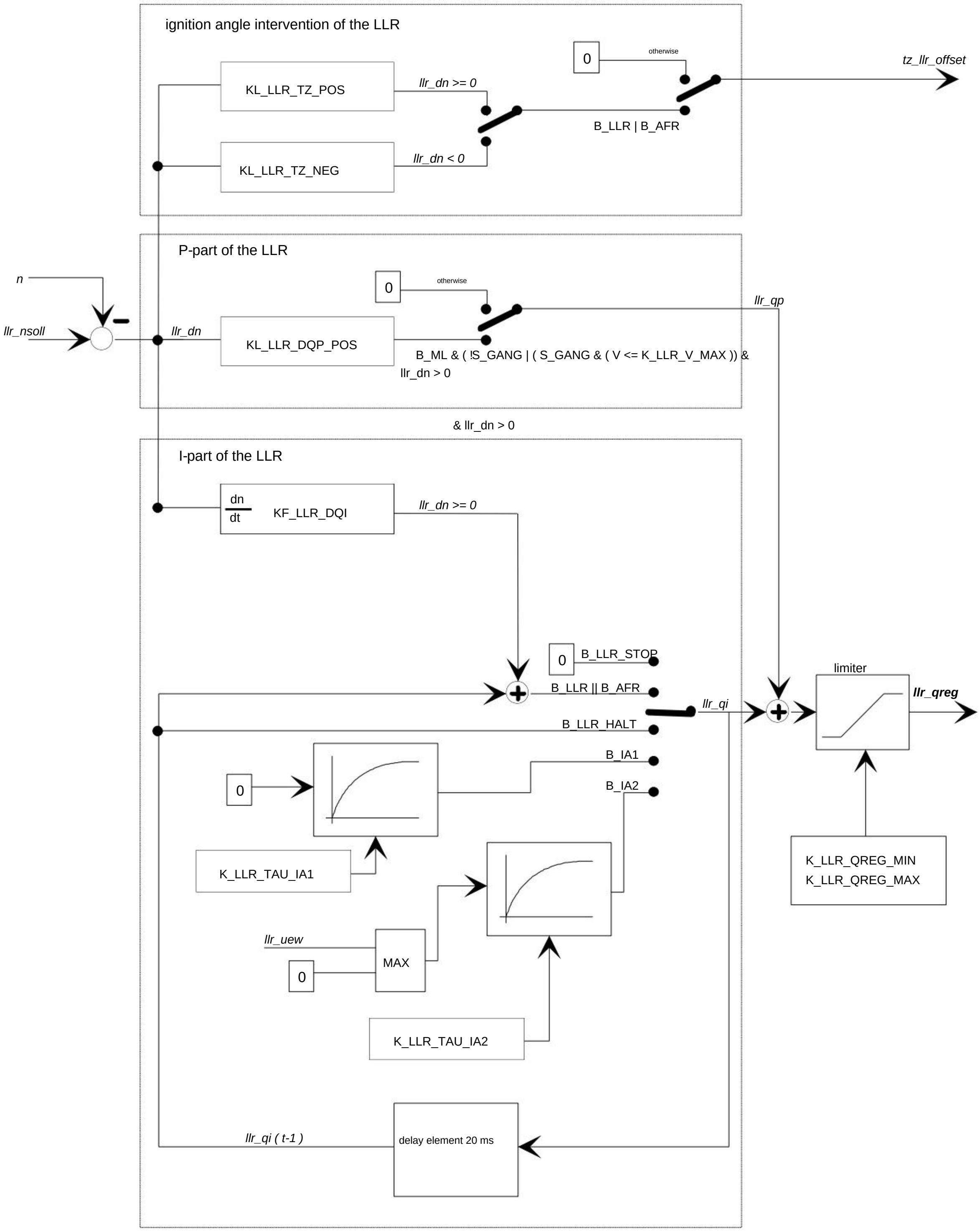
Name	Description	Type	Resolution
llr_nstat	stationary target speed	uw	1 rpm
llr_nnachf	tracked target speed resulting,	uw	1 rpm
llr_nsoll	filtered target speed	uw	1 rpm
llr_znsoll	time constant for target speed filter	uw	1 rpm

Description of the application data:

name	type	Dim.	x-axis	y-axis
KL_LLR_NSOLL_START	KL	3 x 1	tmot - cooling water temp.	--
KL_LLR_NSOLL_GRUND	KL	4 x 1	tmot - cooling water temp.	--
K_LLR_NSOLL_AC	K		--	--
K_LLR_NNACHF_FAKTOR	K		--	--
K_LLR_NNACHF_MAX	K		--	--
KL_LLR_TAU_NSOLL	KL	1 1 1 4 x 1	tmot - cooling water temp.	--

3.6 Idle control

Figure 3.6: **Overview of the idle control**



An idle speed controller is superimposed on the stationary Qsoll calculation, which is intended to compensate for deviations from the specified idle speed.

The idle speed controller is designed as a PI controller. The input variable of the controller is the deviation of the actual speed from the target speed.

speed difference llr_dn	=	target speed - llr_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 LLR also intervenes in the ignition angle calculation using the correction offset "tz_llr_offset".

3.6.1 P-component of the idle controller

The P component is calculated from the characteristic curve KL_LLR_DQP_POS and depends on the amount of the speed difference between the target and actual speed. It is calculated synchronously in 20 ms.

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

	BIT_P_REGLER_ON (BIT 0) set in K_LLR_CONTROL
and	Operating state = Motor_running (B_ML)
and	(S_GANG = no power transmission
	or S_GANG = power transmission and v <= K_LLR_V_MAX)
and	Speed too low (llr_dn > 0)

The P component can be set to zero for test or application purposes using BIT_P_REGLER_ON (bit 0) in the constant K_LLR_CONTROL.

3.6.2 I-component 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_LLR_STOP The I component is set to zero.
- Idle control: Starting control:	B_LLR or B_AFR The I controller is active llr_qi t = llr_qi t-1 + dqi
- Integrator control range 1:	B_IA1 The I component is set to zero via a pt1 filter with the time constant K_LLR_TAU_IA1.

- Integrator control area 2: B_IA2
  The I component is guided to the maximum of llr_uew and zero via a pt1 filter with the time constant K_LLR_TAU_IA2. The variable llr_reg.uew is the I component at the time of the state transition from idle control to start-up control.

Within the states B_LLR and B_AFR there are four special cases:

If the idle speed falls below the value K_LLR_NDIFF_RESET and the I component is negative at this time, it is immediately set to zero (B_LLR_RESET).

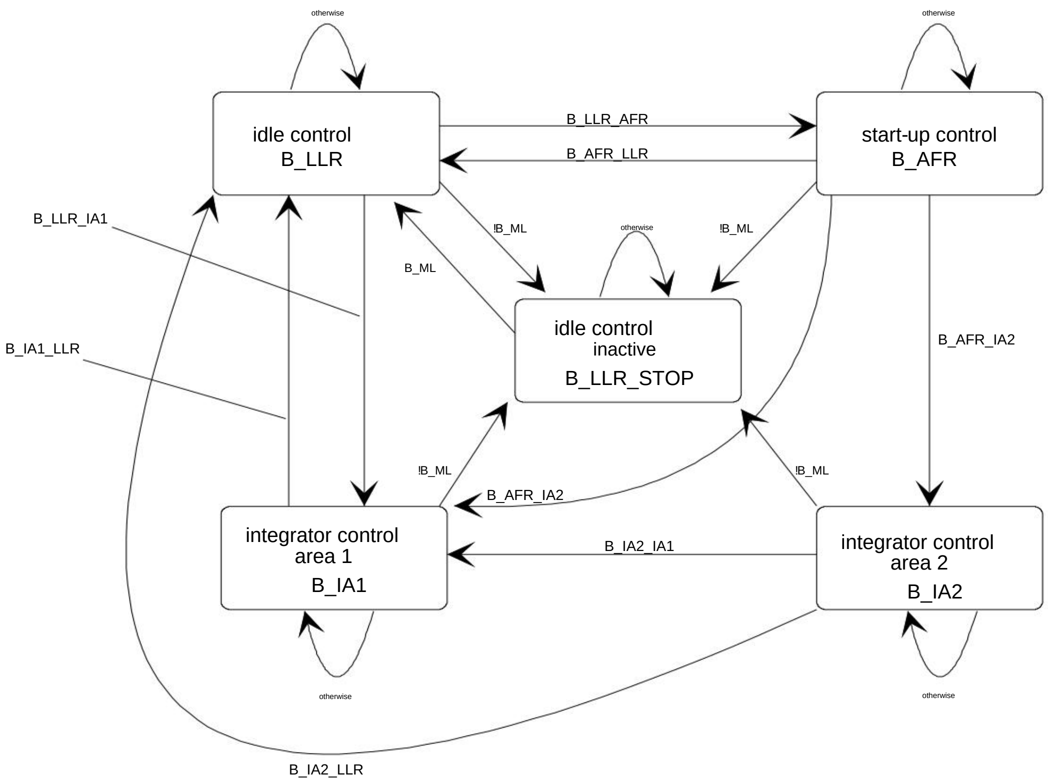
If the load tl falls below the minimum load threshold llr_reg.tl_min, calculated from KL_LLR_TL_MIN = f(tmot), a further reduction of the I-component is blocked (B_LLR_NEGSTOP).

If the sum of the P and I components is outside the controller limits defined with K_LLR_QREG_MIN and K_LLR_QREG_MAX, the I component is frozen (B_LLR_HALT).

The start-up control can only increase the I component, but not reduce it.

Figure 3.6 shows the state diagram and the transition conditions for the idle controller.

Figure 3.6: **idle speed controller state diagram**



State transitions of the idle controller:

Transition idle control ÿ start-up control
B_LLR_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_LLR = B_LL
and !S_GANG)
and !B_SA

Transition start-up control ÿ Integrator de-control area 1
B_AFR_IA1 = B_ML
and B_SA

Transition start-up control ÿ Integrator de-control area 2
B_AFR_IA2 = B_ML
and !(B_LLR and !S_GANG)
and !B_SA
and time K_LLR_TFBR expired

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_LLR = B_LL
and !S_GANG
and !B_SA

Transition Integrator de-control Range 1 ÿ Idle control
B_IA1_LLR = B_LL
and !S_GANG
and !B_SA

Transition idle control ÿ Integrator de-control area 1
B_LLR_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_LLR_CONTROL.

idle speed controller data

Description of the variables:

name	Description	Type	Resolution
llr_dn	speed difference	sw	1 rpm
llr_qp	P-component	sw	1/256 kg/h
llr_qi	I-part	sw	1/256 kg/h
llr_qreg	limited regulator contribution	sw	1/256 kg/h
llr_status	LLR status information Bit 0: B_LLR_STOP 1: B_LLR 2: B_AFR 3: B_IA1 4: B_IA2	uc	--
llr_flags	internal flags of the LLR Bit 0: Flag for starting air mass (Qvs) 1: Time monitoring B_KO active 2: B_KO was last active 4: B_LLR_NEGSTOP 5: B_LLR_HALT	uc	--
llr_uew	I-component at the end of the idle control	sw	1/256 kg/h
llr_tlmin	minimum load for negative stop	uw	1 us/rev.
llr_tfbr	Timer for underbrake release at AFR	uc	0.02 sec
llr_tz_offset	ignition angle offset of the idle speed controller	sw	0.1 °kW

Description of the application data:

name	type	Dim.	x-axis		y-axis
KL_LLR_DQP_POS	KL	16 x 1	llr.dn speed deviation	-	--
KF_LLR_DQI	KF	15 x 8	llr.dn speed deviation	-	d_n40 speed gradient
K_LLR_QREG_MIN	K		--		--
K_LLR_QREG_MAX	K		--		--
KL_LLR_TL_MIN	KL	1 1 4 x 1	tmot cooling water temp.	-	--
K_LLR_NDIFF_RESET	K	1	--		--
K_LLR_T_FBR	K	1	--		--
K_LLR_TAU_IA1	K	1	--		--
K_LLR_TAU_IA2	K	1	--		--
K_LLR_V_MAX	K	1	--		--
KL_LLR_TZ_NEG	KL	12	llr.dn speed deviation	-	--
KL_LLR_TZ_POS	KL	12	llr.dn speed deviation	-	--

3.6.3 Ignition angle intervention of the idle speed controller

To support idle control via the air supply, the controller also intervenes in the ignition angle calculation via the global variable "tz_llr_offset". The variable "tz_llr_offset" is calculated as follows:

Condition
B_LLR or B_AFR & ; Status of the LLR: LLR or AFR active
ll_dn ÿ 0 ; speed too low

tz_llr_offset = KL_LLR_TZ_POS ; ignition angle offset = f(llr.dn)

Condition
B_LLR or B_AFR & ; Status of the LLR: LLR or AFR active
llr_dn < 0 ; speed too high

tz_llr_offset = KL_LLR_TZ_NEG ; Ignition angle offset = f(-llr.dn)

otherwise

tz_llr_offset = 0 ; no ignition angle measures

The TZ intervention of the LLR is calculated segment-synchronously into the ignition angle path and is not subject to the ignition angle change limitation.

3.7 Demand adaptation

The purpose of demand adaptation is to correct a deviation in the pilot control of the idle air quantity, caused by manufacturing variations, air leakage and signs of aging, in relation to the air mass actually required at idle. This deviation is to be determined by demand adaptation and the pilot control values are to be shifted in parallel by this offset. A distinction is made between two adaptation ranges.

- Adaptation value with inactive air conditioning compressor activation
- Additional adaptation offset with active air conditioning compressor activation

3.7.1 Adaptation conditions

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

B_LLRA = B_LLR ; Idle control active state (see LLR state machine)
and tmot > K_LLR_TMOT_ADAPT ; engine temperature greater than threshold
and !B_TMOT_FEHLER and ! ; error-free tmot recording
B_LLR_IBEGR ; Integrator is not in a limitation

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

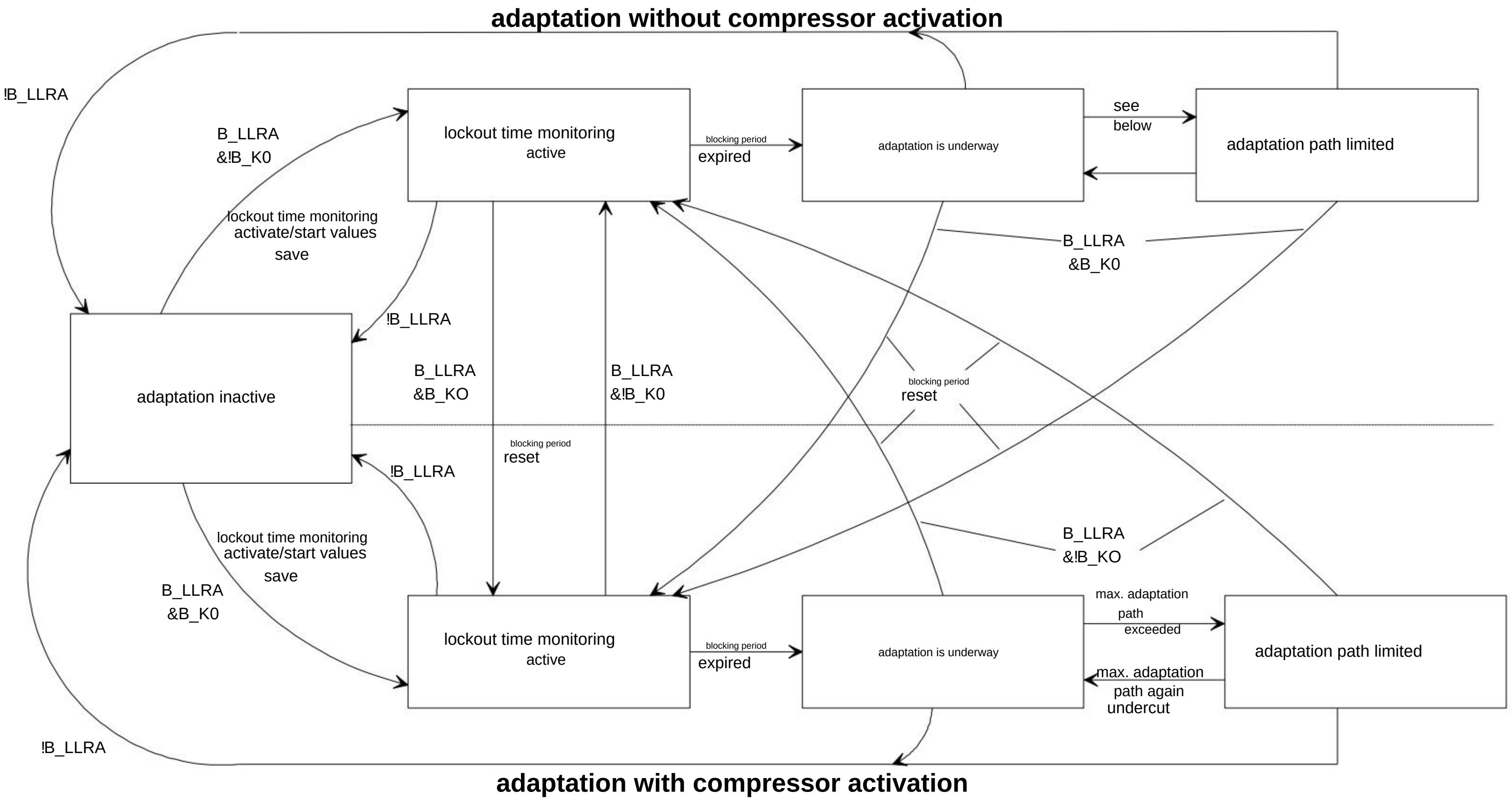
During the development and test phase, the complete demand adaptation (B_LLRA_ENABLED) can be switched off via bit 2 in the control byte K_LLR_CONTROL.
All adaptation values are then equal to zero.

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

3.7.2 States of demand adaptation

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

Figure 3.7.2: **state machine of demand adaptation**



adaptation inactive

Condition: B_LLRA not fulfilled

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

Adaptation values: lla_qadapt (t) = lla_qadapt (t - 20 ms)
lla_kqadapt (t) = lla_kqadapt (t - 20 ms)

Lock time monitoring for LLRA without K0 active

Condition: B_LLRA 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: llra_qadapt (t) = lla_qadapt (t - 20 ms)
lla_kqadapt (t) = lla_kqadapt (t - 20 ms)

Adaptation running (without K0)

Condition: B_LLRA
and !B_K0
and lla_timer == 0 (lock time expired)
and | lla_qadapt - lla_qstart | ÿ K_LLR_DQADAPT_MAX
(adaptation path not limited)

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

Adaptation values: lla_qadapt (t) = lla_qadapt (t - 20 ms) + (llr_qi(t - 20 ms) + K_LLR_QADAPT_OFFSET) * K_LLR_TAU_ADAPT
(without taking into account any limitation)
lla_kqadapt (t) = lla_kqadapt (t - 20 ms)

adaptation value (without K0) limited

Condition: B_LLRA
and !B_KO
and lla_timer == 0
and | lla_qadapt - lla_qstart | > K_LLR_DQADAPT_MAX
(adaptation path limited)

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

Adaptation values: lla_qadapt (t) = lla_qstart ÿ K_LLR_DQADAPT_MAX
lla_kqadapt (t) = lla_kqadapt (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 LLRA with K0 active

Condition: B_LLRA 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_qadapt (t) = lla_qadapt (t - 20 ms)
lla_kqadapt (t) = lla_kqadapt (t - 20 ms)

Adaptation running (with K0)

Condition: B_LLRA
and !B_K0
and lla_timer == 0 (lock time expired)
and | lla_qadapt - lla_qstart | ÿ K_LLR_DQADAPT_MAX
(adaptation path not limited)

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

Adaptation values: lla_ko_qadapt (t) = lla_kqadapt (t - 20 ms) +
K_LLR_QADAPT_OFFSET) * K_LLR_TAU_ADAPT
(without taking into account any limitation)
lla_qadapt (t) = lla_qadapt (t - 20 ms) + (llr.qi(t - 20 ms)

adaptation value (with K0) limited

Condition: B_LLRA
and !B_KO
and lla_timer == 0
and | lla_qadapt - lla_qstart | > K_LLR_DQADAPT_MAX
(adaptation path limited)

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

Adaptation values: lla_kqadapt (t) = lla_kqadapt (t - 20 ms)
lla_qadapt (t) = lla_qstart ÿ K_LLR_DQADAPT_MAX

For all states applies
Initial value of demand adaptation:
llr_qadaption (t) = llra_qadapt (t), if !B_KO
= lla_qadapt (t)
+ lla_kqadapt (t) , if B_KO

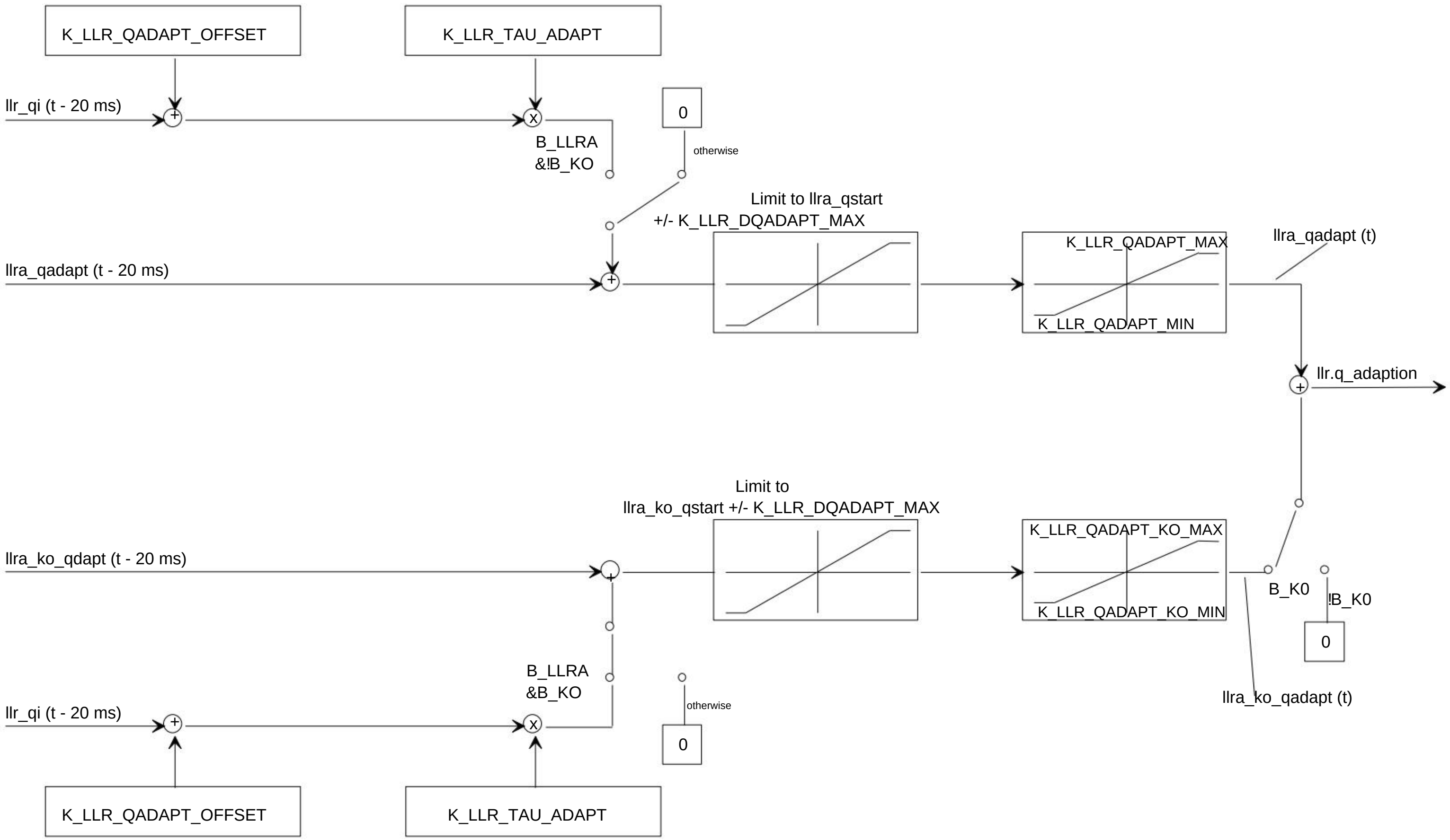
Correction of the integrator component llr.qi of the idle control

when compressor activation is inactive
llr_qi (t) = llr_qi (t)
- (lla_qadapt (t) - lla_qadapt (t - 20 ms))

when compressor activation is active
llr_qi (t) = llr_qi (t)
- (lla_kqadapt (t) - lla_kqadapt (t - 20 ms))

3.7.3 Calculation steps of demand adaptation

Figure 3.7.3: **Block diagram of the LLR adaptation**



integrator

If the active condition for the adaptation integrator is fulfilled (B_LLRA 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:

lqadapt (t) = qadapt (t-20 ms)
+ (llr_qi (t - 20 ms) + K_LLR_QADAPT_OFFSET)
K_LLR_TAU_ADAPT

limiting the adaptation path

A maximum adaptation path of ÿ K_LLR_DQADAPT_MAX is possible per adaptation phase.
An adaptation phase begins when the condition B_LLRA = 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 llra_qadapt and lla_kqadapt are saved in the variables lla_qstart and lla_kqstart. During the adaptation phase, the current adaptation value is then limited to the value ..._qstart ÿ K_LLR_DQADAPT_MAX.

limiting the adaptation values

The resulting adaptation value for inactive compressor activation is set to - K_LLR_DQADAPT_MAX, K_LLR_DQADAPT_MIN which limits the active compressor circuit to the values K_LLR_QADAPT_K0_MAX and K_LLR_QADAPT_K0_MIN.

initial value of demand adaptation

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

if B_K0 inactive
llr_qadaption (t) = lla_qadapt (t)

if B_K0 active
llr_qadaption(t) = lla_qadapt(t) + lla_kqadapt(t)

Correction of the integration component of the idle controller

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

if B_K0 is inactive
llr_qi (t) = llr_qi (t)
- (lla_qadapt (t) - lla_qadapt (t - 20 ms))

when B_k0 is active
ll_qi (t) = llr_qi (t)
- (lla_kadapt (t) - lla_kqadapt (t - 20 ms))

3.7.4 Data of demand adaptation

Description of the variables:

name	Description	type	resolution
lla_timer	remaining adaptation blocking time	uw	0.02 sec.
lla.qadapt	Value of the adaptation integrator without compressor activation	sw	1/256 kg/h
lla.qstart	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_kqadapt	Value of the adaptation integrator with compressor activation	sw	1/256 kg/h
lla_kqstart	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_LLR_QADAPT_OFFSET	FW	adaptation offset for integrator component
K_LLR_TAU_ADAPT	FW	time constant for demand adaptation
K_LLR_DQADAPT_MAX	FW	max. adaptation path per adaptation phase
K_LLR_T_ADAPT	FW	adaptation blocking time
K_LLR_QADAPT_MIN	FW	lower adaptation value limit (without K0)
K_LLR_QADAPT_MAX	FW	upper adaptation value limit (without K0)
K_LLR_QADAPT_KO_MIN	FW	lower adaptation value limit (with K0)
K_LLR_QADAPT_KO_MAX	FW	upper adaptation value limit (with K0)
K_LLR_TMOT_ADAPT	FW	temperature threshold for demand adaptation

3.7.5 Non-volatile storage

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

lla_qadapt
and lla_kqadapt

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.

3.8 Setpoint idle control

The setpoint for the air mass flow rate of the idle control is composed additively of the individual results of the described sub-modules.

llr_qsoll = llr_qvs ; basic value of the feedforward control
+ llr_qac ; Correction of the disturbance variable climate
+ llr_qdashpot ; Correction of the Dashpot function
+ llr_qadaption ; Correction of demand adaptation
+ llr_qreg ; Correction of the idle control
+ K_LLR_Q_MCS ; Q-intervention of the application system

The constant K_LLR_Q_MCS offers the application engineer the possibility to easily influence the air specification using the MCS system.

The minimum setpoint llr_qsoll is limited to the value K_LLR_QSOLL_MIN.

data of the Q setpoint calculation

Description of the variables:

name	Description	Type	Resolution
llr_qsoll	resulting output variable of the idle control	uw	1/256 kg/h

Description of the application data:

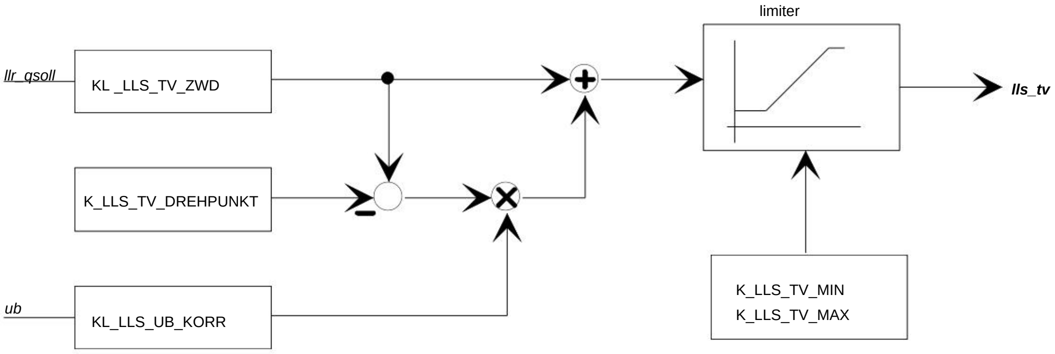
name	type	Dim.	x-axis	y-axis
K_LLR_Q_MCS	K		--	--
KL_LLR_QSOLL_MIN	K	1 1	--	--

3.9 ZWD control

The S50 motor uses a two-winding rotary controller ZWD with one opening and one closing winding as the idle speed controller. The windings are controlled with a clocked pulse width modulated signal. The PWM frequency is 100Hz. The PWM signal for the closing winding corresponds to the inverted signal of the opening winding.

The air flow rate specification llr_qsoll is converted into a duty cycle for the control signal of the idle speed actuator via the ZWD actuator characteristic curve, corrected depending on the on-board voltage and limited to the values K_LLS_TV_MIN or K_LLS_TV_MAX.

Figure 3.9: **Calculation of the duty cycle for the opening ZWD winding**



The duty cycle is stored as a high time in the variable lls_tv. The resolution is 2µs.

lls_tv = tv(f(llr_qsoll)) + (tv(f(llr.qsoll)) - K_LLS_TV_DREHPUNKT) * ub_korr(f(ub))

To improve the charge balance, the output stages for the ZWD activation are only switched through in the operating states "Start" and "Engine running" or when terminal 50 (starter) is active. In the operating states "Engine stopped", "Terminal 15 off" or "After-run" the output stages are switched off and the idle speed controller only releases the emergency running cross-section.

data of the actuator control

Description of the variables:

name	Description	Type	Resolution
lls_tv	high time of the control signal	uw	2 µs
status_lls	status information of the LLS Bit 0: Error in control 7: Power amplifiers switched off	uc	--

Description of the application data:

name	type	Dim.	x-axis	y-axis
KL_LLS_TV_ZWD	KL	28 x 1	llr.qsoll - air specification ub	--
KL_LLS_UB_KORR	KL	5 x 1 1	- vehicle electrical system voltage	--
K_LLS_TV_DREHPUNKT	K		--	--
K_LLS_TV_MIN	K		--	--
K_LLS_TV_MAX	K	1 1	--	--

3.10 Replacement value for switch S_GANG

Since the switch S_GANG for detecting a switched drive train has not yet been 100% tested and is not yet installed in all vehicles, it is possible to use the constant K_LLR_SGANG to switch to a substitute value for S_GANG, which is derived from the vehicle speed v.

K_LLR_SGANG = 0:
S_GANG = 0 if v <= K_LLR_V_MAX
S_GANG = 1 if v > K_LLR_V_MAX

K_LLR_SGANG != 0 (requires that switch is installed)
S_GANG = 0 if drive train not engaged
S_GANG = 1 if drive train is switched through

3.11 Possible modifications of the idle control