SRT / SRAE SOFTWARE DOCUMENTATION SUPER-USER

12. PROPORTIONAL or ON/OFF LAMBDA REGULATION

chapter 12

Release : 2.0

REVISIONS DOCUMENT

Release	*Author*	*Date*	*Modifications*
1.0	**M.Mersier**	**09/19/2004**	• **Creation**
2.0	**F.Charbonnel**	**01/07/2009**	• **Proportional lambda regulation update**

12.1 Richness regulation choice

Following maps allow regulation selection and global conditions:

Both regulations (ON/OFF or Proportional):

- **EE.LbP.Nloop**
  Control Calculation Time: Control update time in milliseconds.(response time of the sensor)

- **EE.LbP.Signe_loop**
  Sign of the Error Calculation:
  0: AF increases with increasing richness,
  1: AF decreases with increasing richness (A/F)

- **EE.LbP.Gain_Int**
  Gain Integrator
  Gain applied to the error for integration.

Proportional lambda sensor used:

- **EE.CfgU.LbdaProp_EN**
  Selection of Richness Regulation Function with Proportional Lambda:
  0: Disables the function
  1: Enables the function.

- **EE.LbP.ChxRegulLBDA**
  Type of proportional richness regulation.
  Allows choice of the used regulation
  0: Dyno Bench
  1: Car.

Lambda ON/OFF sensor used:

- **EE.CfgU.LbdOnOff_EN**
  Selection of Lambda ON/OFF Regulation Function:
  0: Disables the function
  1: Enables the function.

12.2 Lambda sensor heating

The following maps allow temperature control of lambda sensor.

- **<u>EE.LBDheat.TimeOnReset</u>** : Lambda heat time after an ECU reset.During this time and while engine stalled, a 100% duty cycle is applied to the lambda heater output.
  After and if engine stays stalled, the lambda heat will stop.

- **<u>EE.LBDheat.Rco</u>** : Lambda heater duty cycle as a function of battery voltage, applied when engine runs

12.3 Proportional lambda sensor

12.3.1 Car richness regulation

- Regulation uses as input channel **Richesse** (Index 288).This channel comes from EE.Acq.air_fuel map interpolated using FD.Acq.Lambda (Index 113).

- Output channel is **GainLoop** (Index 219). It is a multiplicative correction for injection time.

12.3.1.1 Car regulation conditions

To allow richness regulation, some values must be within functional ranges and engine must be stable enough.

<u>Functional ranges</u>
<u>*Minimum coolant temperature*</u>
- **<u>EE.LbP.S_TeauLBDA</u>**
  Coolant temperature threshold in °C above which the lambda regulation is enabled.

<u>*Minimum A/F*</u>
- **<u>EE.LbP.S_minLBDAregul</u>**
  A/F input threshold above which the lambda regulation is enabled.

<u>*Maximum A/F*</u>
- **<u>EE.LbP.S_maxLBDAregul</u>**
  A/F value threshold below which the lambda regulation is enabled..

<u>Stability criterion</u>

- **<u>EE.LbP.KFILT_RegLBDA</u>** : RPM filter for Lambda regulation purpose
  0.004 -> Strong filter.
  1 -> No filter.

- **<u>EE.LbP.Delta_RegLBDA</u>** : Filtered Rpm Stability Band

- **EE.LbP.Tmin_RegStabLBDA** : Filtered Rpm stability duration to decide Rpm is stable.

Throttle/Rpm mode used :

- **EE.LbP.KFILT_PapLBDA** : Throttle filter for Lambda regulation purpose
  0.004 -> Strong filter.
  1 -> No filter.

.

- **EE.LbP.Delta_PapLBDA** : Filtered Throttle Stability Band

- **EE.LbP.Tmin_PapStabLBDA** : Filtered throttle stability duration to decide Throttle is stable.

Stability is detected when both Rpm and Throttle are stables.

Pressure/Rpm mode used

- **EE.LbP.KFILT_PadmLBDA** : PAdm filter for Lambda regulation purpose
  0.004 -> Strong filter.
  1 -> No filter.

- **EE.LbP.Delta_PadmLBDA** : Filtered PAdm Stability Band.

- **EE.LbP.Tmin_PadmStabLBDA** Filtered PAdm stability duration to decide PAdm is stable.

Stability is detected when both Rpm and PAdm are stables.

12.3.1.2 Car richness regulation

Target when Throttle/Rpm mode used :

- **EE.LbP.tab_ConsLBDA** Richness target f(Rpm,TPS)
  Map Richness Target for Control as a function of RPM and Throttle position
  - "Enable" property of this table must be ticked for RPM/throttle
  - "Enable" property of this table must be unticked for RPM/inlet pressure
  (scale must be the same as AF)

Target when Pressure/Rpm mode used :

- **EE.LbP.tab_ConsLBDA1** Richness target f(Rpm,inletP)
  Map Richness Target for Control as a function of RPM and the inlet air pressure.
  - "Enable" property of this table must be unticked for RPM/throttle
  - "Enable" property of this table must be ticked for RPM/Pair
  (scale must be the same as AF)

- **EE.LbP.MinLoopCar** f(Rpm)
  Minimum Gain Clamp: Minimum clamp for control correction.
- **EE.LbP.MaxLoopCar** f(Rpm)
  Maximum Gain Clamp: Maximum clamp for control correction.

Error evaluation:

- **EE.LbP.Signe_loop** =0 erreur_loop = Target - FD.Acq.AF;
- **EE.LbP.Signe_loop** <>0 erreur_loop = FD.Acq.AF - Target

Gain Loop calculation:

GainLoop = previous GainLoop + erreur_loop * **EE.LbP.Gain_Int**

GainLoop is confined between **EE.LbP.MinLoopCar** and **EE.LbP.MaxLoopCar**

12.3.2 Dyno richness regulation

12.3.2.1 Dyno regulation conditions

To allow dyno proportional regulation, one must set **FD.LbP.bouclage** to 1 using Vision.

12.3.2.2 Dyno richness regulation

Used Maps

- **EE.LbP.MinLoop**
  Minimum Gain Clamp: Minimum clamp for control correction.

- **EE.LbP.MaxLoop**
  Maximum Gain Clamp: Maximum clamp for control correction.

- **EE.LbP.Objectif**
  Richness Target for Control: Same scale as AF

Error evaluation:

- **EE.LbP.Signe_loop** =0 erreur_loop = Target - FD.Acq.AF;
- **EE.LbP.Signe_loop** <>0 erreur_loop = FD.Acq.AF - Target

Gain Loop calculation:

GainLoop is a multiplicative correction for injection time.

GainLoop = previous GainLoop + erreur_loop * **EE.LbP.Gain_Int**

GainLoop is confined between **EE.LbP.MinLoop** and **EE.LbP.MaxLoop**

Lambda ON/OFF Regulation

12.3.3 Richness Target (ON/OFF only)

12.3.3.1 Richness Target Map

A map of richness target is used when in closed loop with an ON/OFF oxygen sensor.
To ensure a correct regulation by the oxygen sensor, the base injection map must have been previously filled-in a manner to obtain a richness of 1, for all engine operating points (the base map with richness of 1 must be generated, at the dyno, prior to using the closed loop regulation).
The richness target map then allows the application of a gain f(RPM & throttle), between 0.5 and 1.5, in order to enrich or enlean specific zones.
Since an ON/OFF lambda oxygen sensor can only be used to regulate around a richness of 1, if the regulation is active, it works only in the zones of the map where the richness is 1 (gain = 1.00). In the areas of the target map where the gain is different from 1.00, the gain is applied, but there can't be any closed loop regulation.

12.3.3.2 Mode of Operation

The correction CorrLambda is calculated, or not, according to the following parameter:

EE.Lbd.cx_rich = 0. Function with richness target map and closed loop.

$$\boxed{\begin{array}{l} \text{IF EE.Ti.tab\_Kobj}, f(RPM, Throttle) = 1.00 \\ \quad CorrLambda = Klambda \\ \text{ELSE} \\ \quad CorrLambda = CartoObjectif, f(RPM, Throttle) \end{array}}$$

where: ***KLambda,*** closed loop gain for richness target of 1 (between 0 and 2).

EE.Lbd.cx_rich = 1. Function without correction and without closed loop.

$$\boxed{CorrLambda = 1}$$

EE.Lbd.cx_rich = 2. Function with richness target map but without closed loop.

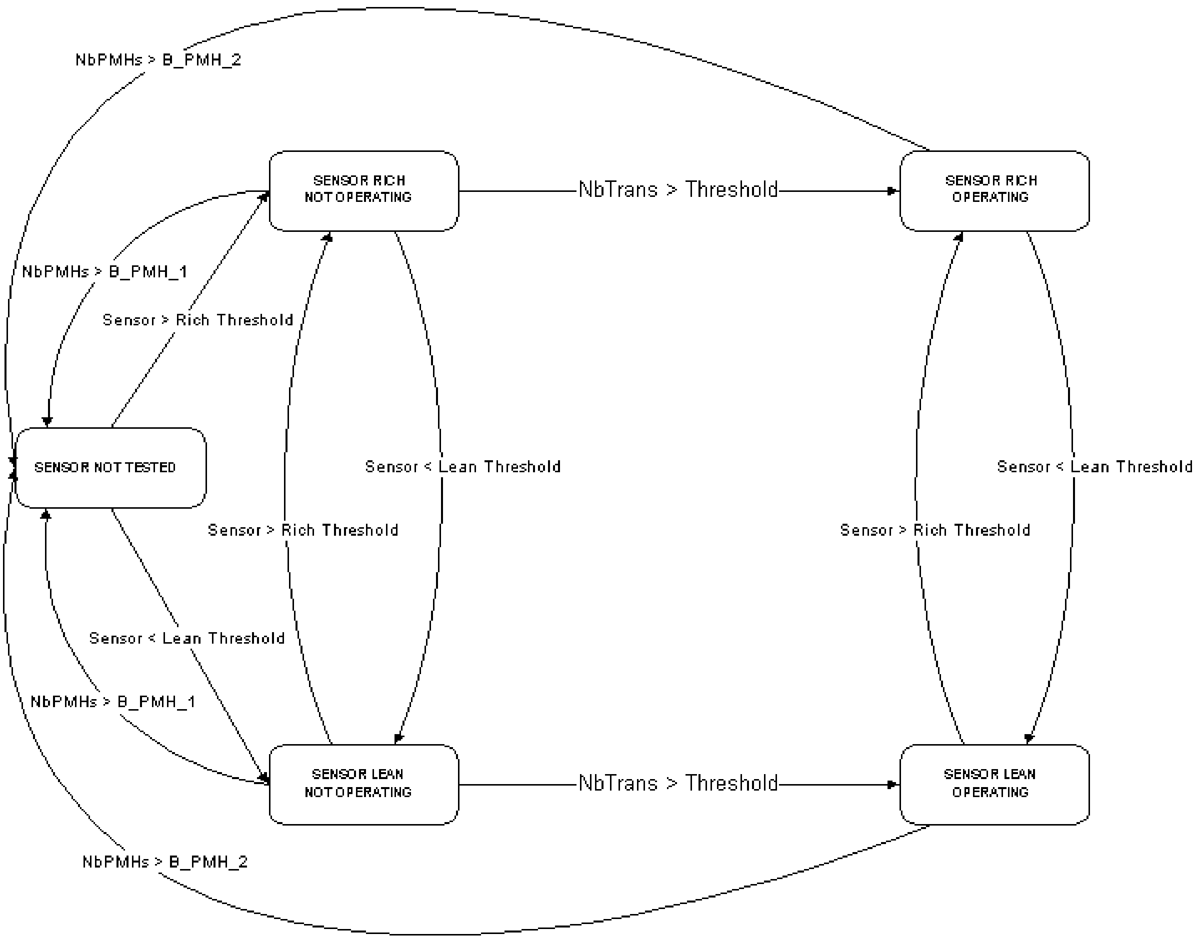
$$\boxed{CorrLambda = CartoObjectif, f(RPM, Throttle) * PotKobj}$$

Where:

- **EE.Ti.tab_Kobj :** Richness Objective f(RPM,Thr)
  Richness objective gain on the quantity of fuel injected from 0,5 to 1,5 as a function of engine speed and throttle position.
- ***PotKobj,*** regulation by «VISON» to adjust the target map between 0.5 and 1.5

12.3.4 Validation of the Oxygen Sensor

The validation of the oxygen sensor is verified using the following diagram:



12.3.4.1 Tests and operating conditions of the sensor

In the 3 states on the left of the above diagram, the regulation does not work.
But the sensor is still declared operative if it switches alternatively more than NO2_Trans times over the rich and below lean thresholds without staying in any of the two “non operational” states for more than B_PMH_1 TDCs.

Rich and lean thresholds:

Minimum number of transitions between rich and lean (before accepting the sensor as operative) :

Time-out for “non operative” states (if it stays in any of the two states over this specified number of TDCs).

Used maps

- **EE.Lbd.SO2_PAUVRE** :Lambda Sensor Lean Thresh
  Threshold in mVolts of the lambda sensor input below which the mixture is said to be lean.

- **EE.Lbd.SO2_MID** :Lambda Transition Rich/Lean
  Threshold in mVolts of the lambda input voltage determining the switch point between rich and lean.

- **EE.Lbd.SO2_RICHE** :Lambda Sensor Rich Thresh
  Threshold in mVolts of the lambda sensor input above which the mixture is said to be rich.

- **EE.Lbd.S_Teau_SO2** :Regulation TH2O Thresh
  Threshold of water temperature in °C above which the lambda regulation is enabled.

- **EE.Lbd.NO2_TRANS** :Sensor Valid Number of Trans
  Number of transitions from rich to lean and lean to rich before the lambda sensor signal is declared valid.

- **EE.Lbd.B_PMH_1** :Sensor Not Working Fault Time
  Number of TDCs before declaring the lambda sensor in default in the non operational state.

12.3.4.2 Sensor is operative

The sensor is operative only in the two states on the right of the diagram, in which it does not stay for more than B_PMH_2 TDCs without switching. The information Rich / Lean for the regulation is determined from a switch threshold.

Switch threshold :

Time-out for “operative” states (to be still considered valid, the sensors must not stay in the operatives states more than this number of TDCs).

- **EE.Lbd.B_PMH_2** :Sensor Working Fault Time
  Number of TDCs before declaring the lambda sensor in default in the operational state.

12.3.5 Closed Loop Correction Term Calculation

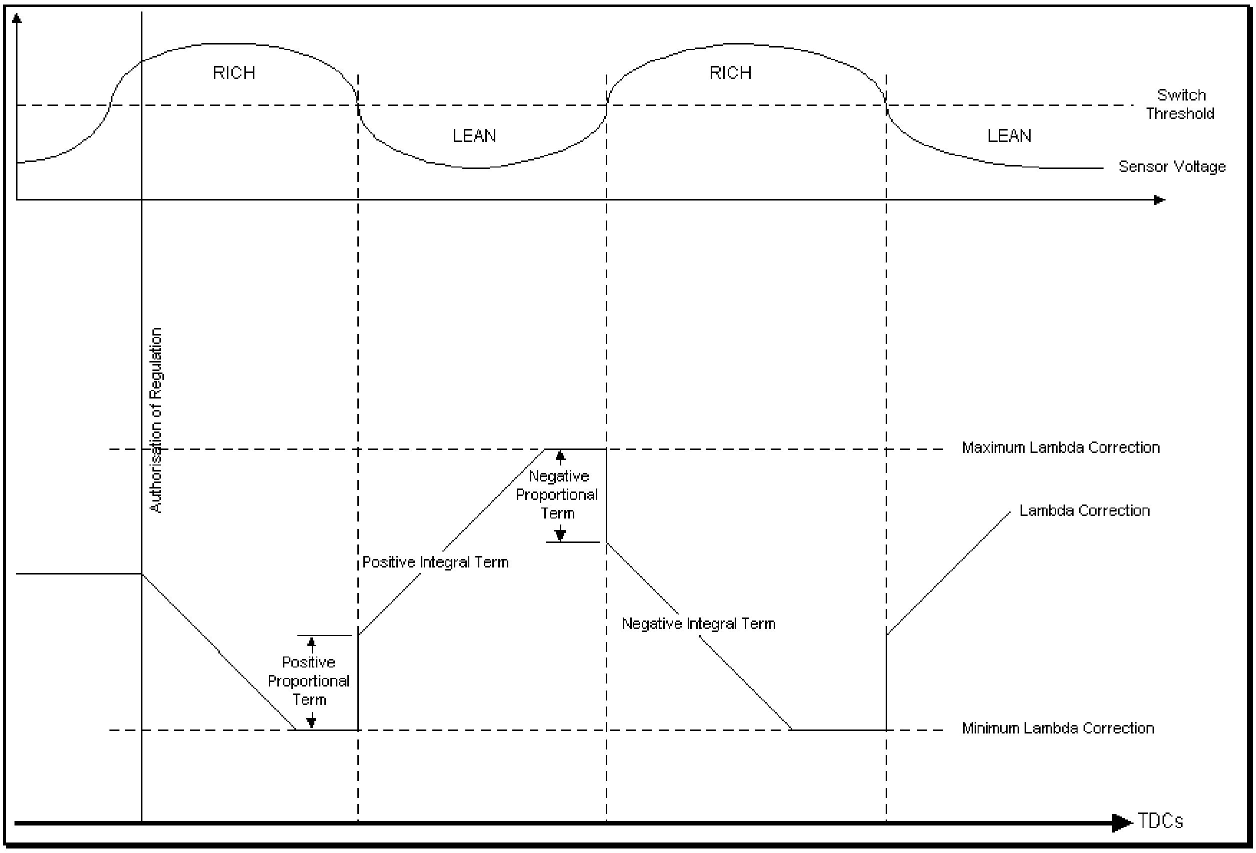
When the mode of operation selected is “0” (closed loop regulation; see) and the richness target is equal to 1, then if the following conditions are met, the correction calculation takes place:

No deceleration injection cut.
Water temperature is greater than a threshold :

Outside these conditions the correction term is always:

$$\boxed{KLambda = 1}$$

The sensor regulation function uses a «pseudo-proportional» term and an integral term as shown in the following diagram:



Regulation activation

- If EE.Lbd.cx_rich <> 0 or
  FD.Ti.Kobj or
  Teau <= EE.Lbd.S_Teau_SO2 or
  Decceleration injection cut or
  Decceleration injection cut

  Then Regulation is desactivated

Regulation calculation when regulation activated :

LEAN -> RICH transition

Klambda = Klambda (previous) + **EE.Lbd.PRAL_R_P**

Klambda = Klambda (previous) + **EE.Lbd.IRAL_R_P**

RICH State
Klambda = Klambda (previous) + **EE.Lbd.IRAL_R_P**

RICH->LEAN transition
Klambda = Klambda (previous) - **EE.Lbd.PRAL_R_N**
Klambda = Klambda (previous) - **EE.Lbd.IRAL_R_N**

LEAN State
Klambda = Klambda (previous) - **EE.Lbd.IRAL_R_N**

12.3.5.1 Pseudo-proportional terms

- **EE.Lbd.PRAL_R_P**: Positive Proportional Corr
  Pseudo-proportional term in % added to the lambda correction on the change in state from rich to lean.

- **EE.Lbd.PRAL_R_N**: Negative Proportional Corr
  Pseudo-proportional term in % subtracted from the lambda correction on the change in state from lean to rich.

12.3.5.2 Integral terms applied every TDC

- **EE.Lbd.IRAL_R_P**: Positive Intregral Corr
  Integral term (in % / TDC) added at each TDC to the lambda correction when the state remains lean.

- **EE.Lbd.IRAL_R_N**: Negative Intregral Corr
  Integral term (in % / TDC) subtracted at each TDC from the lambda correction when the state remains rich.

12.3.5.3 Minimum and maximum limits for Klambda

- **EE.Lbd.KO2MIN** :Min Lambda Correction Clamp
  Minimum limit in % of the lambda correction.

- **EE.Lbd.KO2MAX** :Max Lambda Correction Clamp
  Maximum limit in % of the lambda correction.