任务名称：确定推力控制转速目标值的计算周期

编号：D\_7720

功能：确定推力控制转速目标值的计算周期

前置条件：1

输入：

输出：thrust\_calculate\_circle

公式：thrust\_calculate\_circle = 60;

**任务名称**：确定选定推力等级\_无指定推力等级

**编号**：R\_8479

**功能**：当发动机不处于慢车状态、慢车以上状态、高转速风车起动状态时，置选定推力等级为无指定推力等级

**前置条件**：1

**输入**：engine\_state

**输出**：thrust\_level

**公式**：

if((engine\_state != ES\_idle) && (engine\_state != ES\_above\_idle) && (engine\_state != ES\_Hwindmill\_start))

{

thrust\_level = level\_default;

}

**任务名称**：高压换算转速控制目标计算\_地面慢车

**编号**：R\_6268

**功能**：控制软件应根据地面慢车调节计划表(PS\_GI\_H1~9)，通过高度H、环境温差Dtamb和马赫数Ma计算N2R25Dem\_GI

**前置条件**：1

**输入：**H, Dtamb, Ma, PS\_GI\_H

**输出：**N2R25Dem\_GI

**公式：**

N2R25Dem\_GI = interpolation(PS\_GI\_H, H, Dtamb, Ma);

**任务名称**：高压换算转速控制目标计算\_空中慢车

**编号**：R\_6270

**功能**：控制软件应根据地面慢车调节计划表(PS\_FI\_H1~15)，通过高度H、环境温差Dtamb和马赫数Ma计算N2R25Dem\_FI

**前置条件**：1

**输入：**H, Dtamb, Ma, PS\_FI\_H

**输出：**N2R25Dem\_FI

**公式：**

N2R25Dem\_FI = interpolation(PS\_FI\_H, H, Dtamb, Ma);

**任务名称**：高压换算转速控制目标计算\_进近慢车

**编号**：R\_3879

**功能**：控制软件应根据进近慢车调节计划表(PS\_AI\_H1~15)，通过高度H、环境温差Dtamb和马赫数Ma计算N2R25Dem\_AI

**前置条件**：1

**输入：**H, Dtamb, Ma, PS\_AI\_H

**输出：**N2R25Dem\_AI

**公式：**

N2R25Dem\_AI = interpolation(PS\_AI\_H, H, Dtamb, Ma);

**任务名称**：高压换算转速控制目标计算\_反推慢车

**编号**：R\_6275

**功能**：控制软件应根据反推慢车调节计划表(PS\_RI\_H1~9)，通过高度H、环境温差Dtamb和马赫数Ma计算N2R25Dem\_RI

**前置条件**：1

**输入：**H, Dtamb, Ma, PS\_RI\_H

**输出：**N2R25Dem\_RI

**公式：**

N2R25Dem\_RI = interpolation(PS\_RI\_H, H, Dtamb, Ma);

**任务名称**：N2R25Dem\_BAK计算

**编号**：R\_6631

**功能**：控制软件应根据N1RDem和Ma通过高低压换算转速N2R25-N1R转换关系表N2R25N1Rbak（可调）计算出N2R25备份控制目标N2R25Dem\_BAK。

**前置条件**：1

**输入：**Ma, N1RDem, N2R25N1Rbak

**输出：**N2R25Dem\_BAK

**公式：**

N2R25Dem\_BAK= interpolation(N2R25N1Rbak, N1RDem, Ma);

**任务名称**：N2R25Dem选择\_地面慢车

**编号**：R\_9948

**功能**：当选定推力等级为地面慢车，控制软件应置N2R25Dem=N2R25Dem\_GI

**前置条件**：1

**输入：**thrust\_level, N2R25Dem\_GI

**输出：**N2R25Dem

**公式：**

if(thrust\_level == level\_GI)

{

N2R25Dem = N2R25Dem\_GI;

}

**任务名称**：N2R25Dem选择\_空中慢车

**编号**：R\_9949

**功能**：当选定推力等级为空中慢车，控制软件应置N2R25Dem=N2R25Dem\_FI

**前置条件**：1

**输入：**thrust\_level, N2R25Dem\_FI

**输出：**N2R25Dem

**公式：**

if(thrust\_level == level\_FI)

{

N2R25Dem = N2R25Dem\_FI;

}

**任务名称**：N2R25Dem选择\_进近慢车

**编号**：R\_9950

**功能**：当选定推力等级为进近慢车，控制软件应置N2R25Dem=N2R25Dem\_AI

**前置条件**：1

**输入：**thrust\_level, N2R25Dem\_AI

**输出：**N2R25Dem

**公式：**

if(thrust\_level == level\_AI)

{

N2R25Dem = N2R25Dem\_AI;

}

**任务名称**：N2R25Dem选择\_反推慢车

**编号**：R\_9951

**功能**：当选定推力等级为反推慢车，控制软件应置N2R25Dem=N2R25Dem\_RI

**前置条件**：1

**输入：**thrust\_level, N2R25Dem\_RI

**输出：**N2R25Dem

**公式：**

if(thrust\_level == level\_RI)

{

N2R25Dem = N2R25Dem\_RI;

}

**任务名称**：N2R25Dem选择\_补充1

**编号**：R\_9952

**功能**：当选定推力等级为除：地面慢车、空中慢车、进近慢车、反推慢车、无指定推力等级以外的推力等级，N2R25Dem=N2R25Dem\_BAK。

**前置条件**：1

**输入：**thrust\_level, N2R25Dem\_ BAK

**输出：**N2R25Dem

**公式：**

if(thrust\_level != level\_GI && thrust\_level != level\_FI && thrust\_level != level\_AI && thrust\_level != level\_RI && thrust\_level != level\_default)

{

N2R25Dem = N2R25Dem\_ BAK;

}

**任务名称**：N2R25Dem选择\_补充2

**编号**：D\_6354

**功能**：当选定推力等级为无指定推力等级，且发动机状态为正常地面起动状态，应用软件应置N2R25控制目标N2R25Dem=N2R25Dem\_GI。

**前置条件**：1

**输入：**thrust\_level, engine\_state, N2R25Dem\_GI

**输出：**N2R25Dem

**公式：**

if(thrust\_level == level\_default && engine\_state == ES\_ ground\_start)

{

N2R25Dem = N2R25Dem\_GI;

}

**任务名称**：N2R25Dem选择\_补充3

**编号**：D\_6355

**功能**：当选定推力等级为无指定推力等级，且发动机状态为稳态风车起动状态、快速风车起动状态、空中辅助起动状态，应用软件应置N2R25控制目标N2R25Dem=N2R25Dem\_FI。

**前置条件**：1

**输入：**thrust\_level, engine\_state, N2R25Dem\_FI

**输出：**N2R25Dem

**公式：**

if(thrust\_level == level\_default && (engine\_state == ES\_windmill\_start || engine\_state == ES\_Qwindmill\_start || engine\_state == ES\_assist\_start))

{

N2R25Dem = N2R25Dem\_FI;

}

**任务名称**：N2R25Dem选择\_补充4

**编号**：D\_6357

**功能**：当选定推力等级为无指定推力等级，且发动机状态为初始状态、地面待机状态、空中待机状态、冷运转、假开车/油封/启封、正常停车、紧急停车、地面起动中止、空中起动中止状态，应用软件应置N2R25控制目标N2R25Dem=0。

**前置条件**：1

**输入：**thrust\_level, engine\_state

**输出：**N2R25Dem

**公式：**

if(thrust\_level == level\_default && (engine\_state == ES\_origin || engine\_state == ES\_ground\_standby || engine\_state == ES\_flight\_standby || engine\_state == ES\_cold\_operation || engine\_state == ES\_DR\_OS\_Us || engine\_state == ES\_normal\_stop || engine\_state == ES\_emergency\_stop || engine\_state == ES\_ground\_start\_termination || engine\_state == ES\_flight\_start\_terminnation ))

{

N2R25Dem = 0;

}

**任务名称**：设置起飞锁定指令\_有效

**编号**：R\_6536

**功能**：当同时满足以下条件时，置起飞锁定指令有效：

a) 轮载信号为地面；

b) 飞机校准表速CAS>PD\_ToLockCASHigh（可调）；

c) PLA处于（71，76]。

**前置条件**：1

**输入：**airdis\_WOW, CAS, PLA, PD\_ToLockCASHigh

**输出：**to\_lock\_sig

**公式：**

if (airdis\_WOW ==1 && CAS > PD\_ToLockCASHigh && PLA >71 && PLA <= 76)

{

to\_lock\_sig = 1;

}

**任务名称**：设置起飞阶段指令\_有效

**编号**：R\_6547

**功能**：同时满足以下条件，置起飞阶段指令有效：

a) 轮载信号为地面且维持PD\_ToPhaseTime1（可调）以上；

b) 校准表速CAS<PD\_ToPhaseCASHigh（可调）;

c) PLA处于（71，76]。

**前置条件**：1

**输入**：WOW\_ground\_time, airdis\_WOW, CAS, PD\_ToPhaseCASHigh, PLA, PD\_ToPhaseTime1

**输出：**to\_phase\_sig

**公式：**

if (airdis\_WOW == 1 && WOW\_ground\_time > PD\_ToPhaseTime1&&CAS < PD\_ToPhaseCASHigh && PLA >71 && PLA <= 76)

{

to\_phase\_sig = 1;

}

**任务名称**：起飞锁定指令有效后锁定参数

**编号**：R\_6555

**功能**：起飞锁定指令有效后，应用软件应锁定环境温差Dtamb、灵活温度、NAI、WAI和ECS引气构型和推力等级需求为锁定时刻的前一时刻值。

**前置条件**：1

**输入：**to\_lock\_sig

**输出：**lock\_to\_params\_flag

**公式：**

if (to\_lock\_sig == 1)

{

lock\_to\_params\_flag = 1;

}

**任务名称**：起飞阶段指令与起飞锁定指令同时无效采用实时参数

**编号**：R\_6563

**功能**：起飞阶段指令与起飞锁定指令同时无效时，应采用实时的环境温差Dtamb、推力等级需求、灵活温度、NAI、WAI和ECS引气构型参与控制。

**前置条件**：1

**输入：**to\_lock\_sig, to\_phase\_sig

**输出：**lock\_to\_params\_flag

**公式：**

if (to\_lock\_sig == 0 && to\_phase\_sig == 0)

{

lock\_to\_params\_flag = 0;

}

**任务名称**：起飞锁定指令失效后禁止再次启动起飞锁定指令

**编号**：R\_8267

**功能**：当起飞锁定指令失效后，禁止再次启动起飞锁定指令直到飞机再次回到地面（即轮载信号由空中变为地面）后并且飞机校准表速CAS小于PD\_ToLockCASLow（可调）。

**前置条件**：1

**输入：**to\_lock\_sig, airdis\_WOW, CAS, PD\_ToLockCASLow

**输出：**to\_lock\_sig\_enable

**公式：**

if (to\_lock\_sig == 0)

{

to\_lock\_sig\_enable = 0;

}

if (airdis\_WOW == 1 && CAS < PD\_ToLockCASLow)

{

to\_lock\_sig\_enable = 1;

}

**任务名称**：起飞阶段指令失效后禁止再次启动起飞阶段指令

**编号**：R\_9267

**功能**：当起飞阶段指令失效后，禁止再次启动起飞阶段指令直到飞机再次回到地面（即轮载信号由空中变为地面）后并且飞机校准表速CAS小于PD\_ToLockCASLow（可调）。

**前置条件**：1

**输入：**to\_phase\_sig, airdis\_WOW, CAS, PD\_ToLockCASLow

**输出：**to\_phase\_sig\_enable

**公式：**

if (to\_phase\_sig == 0)

{

to\_phase\_sig\_enable = 0;

}

if (airdis\_WOW == 1 && CAS < PD\_ToLockCASLow)

{

to\_phase\_sig\_enable = 1;

}

**任务名称**：基本推力计划计算\_最大起飞（MTO）

**编号**：R\_6284

**功能**：应用软件应根据最大起飞调节计划(PS\_MTO\_H1~21)，通过高度H、马赫数Ma、环境温差Dtamb线性插值计算最大起飞N1R基准控制目标N1RDemBase\_MTO。

**前置条件**：1

**输入：**H, Dtamb, Ma, PS\_MTO\_H

**输出：**N1RDemBase\_MTO

**公式：**

N1RDemBase\_MTO = interpolation(PS\_MTO\_H, H, Dtamb, Ma);

**任务名称**：基本推力计划计算\_复飞（GA）

**编号**：R\_6297

**功能**：应用软件应根据复飞调节计划（PS\_GA\_H1~21），通过高度H、马赫数Ma、环境温差Dtamb线性插值计算复飞N1R基准控制目标N1RDemBase\_GA。

**前置条件**：1

**输入：**H, Dtamb, Ma, PS\_GA\_H

**输出：**N1RDemBase\_GA

**公式：**

N1RDemBase\_GA= interpolation(PS\_GA\_H, H, Dtamb, Ma);

**任务名称**：基本推力计划计算\_减推力起飞1（DTO1）

**编号**：R\_6300

**功能**：应用软件应根据减推力起飞1调节计划（PS\_DTO1\_H1~21），通过高度H、马赫数Ma、环境温差Dtamb线性插值计算减推力起飞1N1R基准控制目标N1RDemBase\_DTO1。

**前置条件**：1

**输入：**H, Dtamb, Ma, PS\_DTO1\_H

**输出：**N1RDemBase\_DTO1

**公式：**

N1RDemBase\_DTO1 = interpolation(PS\_DTO1\_H, H, Dtamb, Ma);

**任务名称**：基本推力计划计算\_减推力起飞2（DTO2）

**编号**：R\_6311

**功能**：应用软件应根据减推力起飞2调节计划（PS\_DTO2\_H1~21），通过高度H、马赫数Ma、环境温差Dtamb线性插值计算减推力起飞2N1R基准控制目标N1RDemBase\_DTO2。

**前置条件**：1

**输入：**H, Dtamb, Ma, PS\_DTO2\_H

**输出：**N1RDemBase\_DTO2

**公式：**

N1RDemBase\_DTO2 = interpolation(PS\_DTO2\_H, H, Dtamb, Ma);

**任务名称**：基本推力计划计算\_增推力起飞（Bump）

**编号**：R\_6313

**功能**：应用软件应根据增推力起飞调节计划（PS\_Bump\_H1~21），通过高度H、马赫数Ma、环境温差Dtamb线性插值计算增推力起飞N1R基准控制目标N1RDemBase\_Bump。

**前置条件**：1

**输入：**H, Dtamb, Ma, PS\_Bump\_H

**输出：**N1RDemBase\_Bump

**公式：**

N1RDemBase\_Bump = interpolation(PS\_Bump\_H, H, Dtamb, Ma);

**任务名称**：基本推力计划计算\_ 灵活起飞（FLEXTO）

**编号**：R\_6315

**功能**：应用软件应根据最大起飞调节计划，通过高度H、马赫数Ma、灵活温差DTFLEX插值计算得到灵活起飞N1R目标转速N1RDem\_FLEXTO，计算过程中用灵活温差DTFLEX代替环境温差Dtamb进行插值计算。

**前置条件**：1

**输入：**H, DTFLEX, Ma, PS\_MTO\_H

**输出：**N1RDem\_FLEXTO

**公式：**

N1RDem\_FLEXTO = interpolation(PS\_MTO\_H, H, DTFLEX, Ma);

**任务名称**：基本推力计划计算\_最大连续（MCT）

**编号**：R\_6317

**功能**：应用软件应根据最大连续调节计划（PS\_MCT\_H1~21），通过高度H、马赫数Ma、环境温差Dtamb线性插值计算最大连续N1R基准控制目标N1RDemBase\_MCT。

**前置条件**：1

**输入：**H, Dtamb, Ma, PS\_MCT\_H

**输出：**N1RDemBase\_MCT

**公式：**

N1RDemBase\_MCT = interpolation(PS\_MCT\_H, H, Dtamb, Ma);

**任务名称**：基本推力计划计算\_最大爬升（MCL）

**编号**：R\_6319

**功能**：应用软件应根据最大爬升调节计划（PS\_MCL\_H1~18），通过高度H、马赫数Ma、环境温差Dtamb线性插值计算最大爬升N1R基准控制目标N1RDemBase\_MCL。

**前置条件**：1

**输入：**H, Dtamb, Ma, PS\_MCL\_H

**输出：**N1RDemBase\_MCL

**公式：**

N1RDemBase\_MCL = interpolation(PS\_MCL\_H, H, Dtamb, Ma);

**任务名称**：基本推力计划计算\_减推力爬升1

**编号**：R\_6321

**功能**：应用软件应根据减推力爬升1调节计划（PS\_DCL1\_H1~18），通过高度H、马赫数Ma、环境温差Dtamb线性插值计算减推力爬升1N1R基准控制目标N1RDemBase\_DCL1。

**前置条件**：1

**输入：**H, Dtamb, Ma, PS\_DCL1\_H

**输出：**N1RDemBase\_DCL1

**公式：**

N1RDemBase\_DCL1 = interpolation(PS\_DCL1\_H, H, Dtamb, Ma);

**任务名称**：基本推力计划计算\_减推力爬升2

**编号**：R\_6324

**功能**：应用软件应根据减推力爬升2调节计划(PS\_DCL2\_H1~18)，通过高度H、马赫数Ma、环境温差Dtamb线性插值计算减推力爬升2N1R基准控制目标N1RDemBase\_DCL2。

**前置条件**：1

**输入：**H, Dtamb, Ma, PS\_DCL2\_H

**输出：**N1RDemBase\_DCL2

**公式：**

N1RDemBase\_DCL2 = interpolation(PS\_DCL2\_H, H, Dtamb, Ma);

**任务名称**：基本推力计划计算\_最大巡航（MCR）

**编号**：R\_6326

**功能**：应用软件应根据最大巡航调节计划(PS\_MCR\_H1~15)，通过高度H、马赫数Ma、环境温差Dtamb线性插值计算最大巡航N1R基准控制目标N1RDemBase\_MCR。

**前置条件**：1

**输入：**H, Dtamb, Ma, PS\_MCR\_H

**输出：**N1RDemBase\_MCR

**公式：**

N1RDemBase\_MCR = interpolation(PS\_MCR\_H, H, Dtamb, Ma);

**任务名称**：计算最大起飞正常引气修正

**编号**：R\_10794

**功能**：应用软件应根据最大起飞MTO的ECS正常引气修正表（可调，PS\_ECSNormalMTO）通过高度H、马赫数Ma线性插值计算得到DN1RECSmto1

**前置条件**：1

**输入**：PS\_ECSNormalMTO, H, Ma

**输出**：DN1RECSmto1

**公式**：DN1RECSmto1 = interpolation(PS\_ECSNormalMTO, H, Ma);

**任务名称**：计算最大连续正常引气修正

**编号**：R\_10795

**功能**：应用软件应根据最大连续MCT的ECS正常引气修正表（可调，PS\_ECSNormalMCT）通过高度H、马赫数Ma线性线性插值计算得到DN1RECSmct1。

**前置条件**：1

**输入**：PS\_ECSNormalMCT, H, Ma

**输出**：DN1RECSmct1

**公式**：DN1RECSmct1 = interpolation(PS\_ECSNormalMCT, H, Ma);

**任务名称**：计算最大爬升正常引气修正

**编号**：R\_10796

**功能**：应用软件应根据最大爬升MCL的ECS正常引气修正表（可调，PS\_ECSNormalMCL）通过高度H、马赫数Ma线性线性插值计算得到DN1RECSmcl1。

**前置条件**：1

**输入**：PS\_ECSNormalMCL, H, Ma

**输出**：DN1RECSmcl1

**公式**：DN1RECSmcl1 = interpolation(PS\_ECSNormalMCL, H, Ma);

**任务名称**：计算最大巡航正常引气修正

**编号**：R\_10797

**功能**：应用软件应根据最大巡航MCR的ECS正常引气修正表（可调，PS\_ECSNormalMCR）通过高度H、马赫数Ma线性线性插值计算得到DN1RECSmcr1。

**前置条件**：1

**输入**：PS\_ECSNormalMCR, H, Ma

**输出**：DN1RECSmcr1

**公式**：DN1RECSmcr1 = interpolation(PS\_ECSNormalMCR, H, Ma);

**任务名称**：计算最大起飞异常低引气修正

**编号**：R\_10798

**功能**：应用软件应根据最大起飞MTO的ECS异常低引气修正表（可调，PS\_ECSAbnLowMTO）通过高度H、马赫数Ma线性线性插值计算得到DN1RECSmto2。

**前置条件**：1

**输入**：PS\_ECSAbnLowMTO, H, Ma

**输出**：DN1RECSmto2

**公式**：DN1RECSmto2 = interpolation(PS\_ECSAbnLowMTO, H, Ma);

**任务名称**：计算最大连续异常低引气修正

**编号**：R\_10799

**功能**：应用软件应根据最大连续MCT的ECS异常低引气修正表（可调，PS\_ECSAbnLowMCT）通过高度H、马赫数Ma线性线性插值计算得到DN1RECSmct2。

**前置条件**：1

**输入**：PS\_ECSAbnLowMCT, H, Ma

**输出**：DN1RECSmct2

**公式**：DN1RECSmct2 = interpolation(PS\_ECSAbnLowMCT, H, Ma);

**任务名称**：计算最大爬升异常低引气修正

**编号**：R\_10800

**功能**：应用软件应根据最大爬升MCL的ECS异常低引气修正表（可调，PS\_ECSAbnLowMCL）通过高度H、马赫数Ma线性线性插值计算得到DN1RECSmcl2。

**前置条件**：1

**输入**：PS\_ECSAbnLowMCL, H, Ma

**输出**：DN1RECSmcl2

**公式**：DN1RECSmcl2 = interpolation(PS\_ECSAbnLowMCL, H, Ma);

**任务名称**：计算最大巡航异常低引气修正

**编号**：R\_10801

**功能**：应用软件应根据最大巡航MCR的ECS异常低引气修正表（可调，PS\_ECSAbnLowMCR）通过高度H、马赫数Ma线性线性插值计算得到DN1RECSmcr2。

**前置条件**：1

**输入**：PS\_ECSAbnLowMCR, H, Ma

**输出**：DN1RECSmcr2

**公式**：DN1RECSmcr2 = interpolation(PS\_ECSAbnLowMCR, H, Ma);

**任务名称**：计算最大起飞异常高引气修正

**编号**：R\_10802

**功能**：应用软件应根据最大起飞MTO的ECS异常高引气修正表（可调，PS\_ECSAbnHighMTO）通过高度H、马赫数Ma线性线性插值计算得到DN1RECSmto3。

**前置条件**：1

**输入**：PS\_ECSAbnHighMTO, H, Ma

**输出**：DN1RECSmto3

**公式**：DN1RECSmto3 = interpolation(PS\_ECSAbnHighMTO, H, Ma);

**任务名称**：计算最大连续异常高引气修正

**编号**：R\_10803

**功能**：应用软件应根据最大连续MCT的ECS异常高引气修正表（可调，PS\_ECSAbnHighMCT）通过高度H、马赫数Ma线性线性插值计算得到DN1RECSmct3。

**前置条件**：1

**输入**：PS\_ECSAbnHighMCT, H, Ma

**输出**：DN1RECSmct3

**公式**：DN1RECSmct3 = interpolation(PS\_ECSAbnHighMCT, H, Ma);

**任务名称**：计算最大爬升异常高引气修正

**编号**：R\_10804

**功能**：应用软件应根据最大爬升MCL的ECS异常高引气修正表（可调，PS\_ECSAbnHighMCL）通过高度H、马赫数Ma线性线性插值计算得到DN1RECSmcl3。

**前置条件**：1

**输入**：PS\_ECSAbnHighMCL, H, Ma

**输出**：DN1RECSmcl3

**公式**：DN1RECSmcl3 = interpolation(PS\_ECSAbnHighMCL, H, Ma);

**任务名称**：计算最大巡航异常高引气修正

**编号**：R\_10805

**功能**：应用软件应根据最大巡航MCR的异常高引气修正表（PS\_ECSAbnHighMCR）通过高度H、马赫数Ma线性线性插值计算得到DN1RECSmcr3。

**前置条件**：1

**输入**：PS\_ECSAbnHighMCR, H, Ma

**输出**：DN1RECSmcr3

**公式**：DN1RECSmcr3 = interpolation(PS\_ECSAbnHighMCR, H, Ma);

**任务名称**：设置ECS参数（无引气）

**编号**：R\_10806

**功能**：当ECS引气构型ECSBleedCfg为无引气，DN1RECSmto=0，DN1RECSmct=0，DN1RECSmcl=0，DN1RECSmcr=0。

**前置条件**：1

**输入**：ECSBleedCfg

**输出**：DN1RECSmto, DN1RECSmct, DN1RECSmcl, DN1RECSmcr

**公式**：

if(ECSBleedCfg == no\_bleed)

{

DN1RECSmto=0;

DN1RECSmct=0;

DN1RECSmcl=0;

DN1RECSmcr=0;

}

**任务名称**：设置ECS参数（正常引气）

**编号**：R\_10807

**功能**：当ECS引气构型ECSBleedCfg为正常引气，DN1RECSmto=DN1RECSmto1，DN1RECSmct=DN1RECSmct1，DN1RECSmcl=DN1RECSmcl1，DN1RECSmcr=DN1RECSmcr1。

**前置条件**：1

**输入**：ECSBleedCfg, DN1RECSmto1, DN1RECSmct1, DN1RECSmcl1, DN1RECSmcr1

**输出**：DN1RECSmto, DN1RECSmct, DN1RECSmcl, DN1RECSmcr

**公式**：

if(ECSBleedCfg == norm\_bleed)

{

DN1RECSmto=DN1RECSmto1;

DN1RECSmct=DN1RECSmct1;

DN1RECSmcl=DN1RECSmcl1;

DN1RECSmcr=DN1RECSmcr1;

}

**任务名称**：设置ECS参数（异常低引气）

**编号**：R\_10808

**功能**：当ECS引气构型ECSBleedCfg为异常低引气，DN1RECSmto=DN1RECSmto2，DN1RECSmct=DN1RECSmct2，DN1RECSmcl=DN1RECSmcl2，DN1RECSmcr=DN1RECSmcr2。

**前置条件**：1

**输入**：ECSBleedCfg, DN1RECSmto2, DN1RECSmct2, DN1RECSmcl2, DN1RECSmcr2

**输出**：DN1RECSmto, DN1RECSmct, DN1RECSmcl, DN1RECSmcr

**公式**：

if(ECSBleedCfg == abn\_low\_bleed)

{

DN1RECSmto=DN1RECSmto2;

DN1RECSmct=DN1RECSmct2;

DN1RECSmcl=DN1RECSmcl2;

DN1RECSmcr=DN1RECSmcr2;

}

**任务名称**：设置ECS参数（异常高引气）

**编号**：R\_10809

**功能**：当ECS引气构型ECSBleedCfg为异常高引气，DN1RECSmto=DN1RECSmto3，DN1RECSmct=DN1RECSmct3，DN1RECSmcl=DN1RECSmcl3，DN1RECSmcr=DN1RECSmcr3

**前置条件**：1

**输入**：ECSBleedCfg, DN1RECSmto3, DN1RECSmct3, DN1RECSmcl3, DN1RECSmcr3

**输出**：DN1RECSmto, DN1RECSmct, DN1RECSmcl, DN1RECSmcr

**公式**：

if(ECSBleedCfg == abn\_high\_bleed)

{

DN1RECSmto=DN1RECSmto3;

DN1RECSmct=DN1RECSmct3;

DN1RECSmcl=DN1RECSmcl3;

DN1RECSmcr=DN1RECSmcr3;

}

**任务名称**：计算N1RDemWAI1\_MTO

**编号**：R\_10813

**功能**：应用软件应通过H、Ma、根据最大起飞WAI正常引气修正系数表（可调，PS\_WAINormalMTO）线性插值得到DDTwaimto1，通过H、Ma和Dtamb+DDTwaimto1根据最大起飞调节计划表计算得到N1RDemWAI1\_MTO

**前置条件**：1

**输入**：PS\_WAINormalMTO, H, Ma, Dtamb, PS\_MTO\_H, Temp

**输出**：N1RDemWAI1\_MTO

**公式**：

DDTwaimto1 = interpolation(PS\_WAINormalMTO, H, Ma);

Temp = Dtamb+DDTwaimto1;

N1RDemWAI1\_MTO = interpolation(PS\_MTO\_H, H, Ma,Temp);

**任务名称**：计算N1RDemWAI1\_MCT

**编号**：R\_10814

**功能**：应用软件应通过H、Ma、根据最大连续WAI正常引气修正系数表（可调，PS\_WAINormalMCT）线性插值得到DDTwaimct1，通过H、Ma和Dtamb+DDTwaimct1根据最大连续调节计划表计算得到N1RDemWAI1\_MCT。

**前置条件**：1

**输入**：PS\_WAINormalMCT, H, Ma, PS\_MCT\_H, Dtamb, Temp

**输出**：N1RDemWAI1\_MCT

**公式**：

DDTwaimct1 = interpolation(PS\_WAINormalMCT, H, Ma);

Temp = Dtamb+DDTwaimct1;

N1RDemWAI1\_MCT = interpolation(PS\_MCT\_H, H, Ma, Temp);

**任务名称**：计算N1RDemWAI2\_MTO

**编号**：R\_10815

**功能**：应用软件应通过H、Ma、根据最大起飞WAI异常高引气修正系数表（可调，PS\_WAIAbnHighMTO）线性插值得到DDTwaimto2，通过H、Ma和Dtamb+DDTwaimto2根据最大起飞调节计划表计算得到N1RDemWAI2\_MTO。

**前置条件**：1

**输入**：PS\_WAIAbnHighMTO, H, Ma, PS\_MTO\_H, Dtamb, Temp

**输出**：N1RDemWAI2\_MTO

**公式**：

DDTwaimto2 = interpolation(PS\_WAIAbnHighMTO, H, Ma);

Temp = Dtamb+ DDTwaimto2;

N1RDemWAI2\_MTO = interpolation(PS\_MTO\_H, H, Ma,Temp);

**任务名称**：计算N1RDemWAI2\_MCT

**编号**：R\_10816

**功能**：应用软件应通过H、Ma、根据最大连续WAI异常高引气修正系数表（可调，PS\_WAIAbnHighMCT）线性插值得到DDTwaimct2，通过H、Ma和Dtamb+DDTwaimct2根据最大连续调节计划表计算得到N1RDemWAI2\_MCT。

**前置条件**：1

**输入**：PS\_WAIAbnHighMCT, H, Ma, PS\_MCT\_H, Dtamb, Temp

**输出**：N1RDemWAI2\_MCT

**公式**：

DDTwaimct2 = interpolation(PS\_WAIAbnHighMCT, H, Ma);

Temp = Dtamb+DDTwaimct2;

N1RDemWAI2\_MCT = interpolation(PS\_MCT\_H, H, Ma,Temp);

**任务名称**：计算DN1RWAImcl1

**编号**：R\_10817

**功能**：应用软件应根据最大爬升MCL的WAI正常引气修正系数表（可调，PS\_WAINormalMCL）通过高度H、马赫数Ma线性插值计算得到DN1RWAImcl1。

**前置条件**：1

**输入**：PS\_WAINormalMCL, H, Ma

**输出**：DN1RWAImcl1

**公式**：DN1RWAImcl1 = interpolation(PS\_WAINormalMCL, H, Ma);

**任务名称**：计算DN1RWAImcr1

**编号**：R\_10818

**功能**：应用软件应根据最大巡航MCR的WAI正常引气修正系数表（可调，PS\_WAINormalMCR）通过高度H、马赫数Ma线性插值计算得到DN1RWAImcr1

**前置条件**：1

**输入**：PS\_WAINormalMCR, H, Ma

**输出**：DN1RWAImcr1

**公式**：DN1RWAImcr1 = interpolation(PS\_WAINormalMCR, H, Ma);

**任务名称**：计算DN1RWAImcl2

**编号**：R\_10819

**功能**：应用软件应根据最大爬升MCL的WAI异常高引气修正系数表（可调，PS\_WAIAbnHighMCL）通过高度H、马赫数Ma线性插值计算得到DN1RWAImcl2。

**前置条件**：1

**输入**：PS\_WAIAbnHighMCL, H, Ma

**输出**：DN1RWAImcl2

**公式**：DN1RWAImcl2 = interpolation(PS\_WAIAbnHighMCL, H, Ma);

**任务名称**：计算DN1RWAImcr2

**编号**：R\_10820

**功能**：应用软件应根据最大巡航MCR的WAI异常高引气修正系数表（可调，PS\_WAIAbnHighMCR）通过高度H、马赫数Ma线性插值计算得到DN1RWAImcr2。

**前置条件**：1

**输入**：PS\_WAIAbnHighMCR, H, Ma

**输出**：DN1RWAImcr2

**公式**：DN1RWAImcr2 = interpolation(PS\_WAIAbnHighMCR, H, Ma);

**任务名称**：设置WAI参数（无引气）

**编号**：R\_10821

**功能**：当WAI引气构型WAIBleedCfg为无引气，DN1RWAImto=0，DN1RWAImct=0，DN1RWAImcl=0，DN1RWAImcr=0。

**前置条件**：1

**输入**：WAIBleedCfg

**输出**：DN1RWAImto, DN1RWAImct, DN1RWAImcl, DN1RWAImcr

**公式**：

if(WAIBleedCfg == no\_bleed)

{

DN1RWAImto=0;

DN1RWAImct=0;

DN1RWAImcl=0;

DN1RWAImcr=0;

}

**任务名称**：设置WAI参数（正常引气）

**编号**：R\_10822

**功能**：当WAI引气构型WAIBleedCfg为正常引气，

a) DN1RWAImto=N1RDemWAI1\_MTO-N1RDemBase\_MTO;

b) DN1RWAImct=N1RDemWAI1\_MCT-N1RDemBase\_MCT;

c) DN1RWAImcl=DN1RWAImcl1;

d) DN1RWAImcr=DN1RWAImcr1;

**前置条件**：1

**输入**：WAIBleedCfg, N1RDemWAI1\_MTO, N1RDemBase\_MTO, N1RDemWAI1\_MCT, N1RDemBase\_MCT, DN1RWAImcl1, DN1RWAImcr1

**输出**：DN1RWAImto, DN1RWAImct, DN1RWAImcl, DN1RWAImcr

**公式**：

if(WAIBleedCfg == norm\_bleed)

{

DN1RWAImto=N1RDemWAI1\_MTO-N1RDemBase\_MTO;

DN1RWAImct=N1RDemWAI1\_MCT-N1RDemBase\_MCT;

DN1RWAImcl=DN1RWAImcl1;

DN1RWAImcr=DN1RWAImcr1;

}

**任务名称**：设置WAI参数（异常高引气）

**编号**：R\_10823

**功能**：当WAI引气构型WAIBleedCfg为异常高引气

a) DN1RWAImto=N1RDemWAI2\_MTO-N1RDemBase\_MTO;

b) DN1RWAImct=N1RDemWAI2\_MCT-N1RDemBase\_MCT;

c) DN1RWAImcl=DN1RWAImcl2;

d) DN1RWAImcr=DN1RWAImcr2;

**前置条件**：1

**输入**：WAIBleedCfg, N1RDemWAI2\_MTO, N1RDemBase\_MTO, N1RDemWAI2\_MCT, N1RDemBase\_MCT, DN1RWAImcl2, DN1RWAImcr2

**输出**：DN1RWAImto, DN1RWAImct, DN1RWAImcl, DN1RWAImcr

**公式**：

if(WAIBleedCfg == abn\_high\_bleed)

{

DN1RWAImto=N1RDemWAI2\_MTO-N1RDemBase\_MTO;

DN1RWAImct=N1RDemWAI2\_MCT-N1RDemBase\_MCT;

DN1RWAImcl=DN1RWAImcl2;

DN1RWAImcr=DN1RWAImcr2;

}

**任务名称**：计算N1RDemNAI1\_MTO

**编号**：R\_10824

**功能**：应用软件应通过H、Ma根据最大起飞NAI引气修正系数表（可调，PS\_NAIOpenMTO ）线性插值得到DDTnaimto1，通过H、Ma和Dtamb+ DDTnaimto1根据最大起飞调节计划表计算得到N1RDemNAI1\_MTO。

**前置条件**：1

**输入**：PS\_NAIOpenMTO, H, Ma, PS\_MTO\_H, Dtamb, Temp

**输出**：N1RDemNAI1\_MTO

**公式**：

DDTnaimto1 = interpolation(PS\_NAIOpenMTO, H, Ma);

Temp = Dtamb+ DDTnaimto1;

N1RDemNAI1\_MTO = interpolation(PS\_MTO\_H, H, Ma, Temp);

**任务名称**：计算N1RDemNAI1\_MCT

**编号**：R\_10825

**功能**：应用软件应通过H、Ma根据最大连续NAI引气修正系数表（可调，PS\_NAIOpenMCT）线性插值得到DDTnaimct1，通过H、Ma和Dtamb+ DDTnaimct1根据最大连续调节计划表计算得到N1RDemNAI1\_MCT。

**前置条件**：1

**输入**：PS\_NAIOpenMCT, H, Ma, PS\_MCT\_H, Dtamb, Temp

**输出**：N1RDemNAI1\_MCT

**公式**：

DDTnaimct1 = interpolation(PS\_NAIOpenMCT, H, Ma);

Temp = Dtamb+ DDTnaimct1;

N1RDemNAI1\_MCT = interpolation(PS\_MCT\_H, H, Ma,Temp);

**任务名称**：计算DN1RNAImcl1

**编号**：R\_10826

**功能**：应用软件应根据最大爬升MCL的NAI引气修正表（可调，PS\_NAIOpenMCL）通过高度H、马赫数Ma线性插值计算得到DN1RNAImcl1。

**前置条件**：1

**输入**：PS\_NAIOpenMCL, H, Ma

**输出**：DN1RNAImcl1

**公式**：DN1RNAImcl1 = interpolation(PS\_NAIOpenMCL, H, Ma);

**任务名称**：计算DN1RNAImcr1

**编号**：R\_10827

**功能**：应用软件应根据最大巡航MCR的NAI引气修正表（可调，PS\_NAIOpenMCR）通过高度H、马赫数Ma线性插值计算得到DN1RNAImcr1。

**前置条件**：1

**输入**：PS\_NAIOpenMCR, H, Ma

**输出**：DN1RNAImcr1

**公式**：DN1RNAImcr1 = interpolation(PS\_NAIOpenMCR, H, Ma);

**任务名称**：设置NAI参数（关闭）

**编号**：R\_10828

**功能**：当短舱防冰控制指令为关闭，DN1RNAImto=0，DN1RNAImct=0，DN1RNAImcl=0，DN1RNAImcr=0。

**前置条件**：1

**输入**：nai\_control

**输出**：DN1RNAImto, DN1RNAImct, DN1RNAImcl, DN1RNAImcr

**公式**：

if(nai\_control == 0)

{

DN1RNAImto=0;

DN1RNAImct=0;

DN1RNAImcl=0;

DN1RNAImcr=0;

}

**任务名称**：设置NAI参数（打开）

**编号**：R\_10829

**功能**：当短舱防冰控制指令为打开

DN1RNAImto=N1RDemNAI1\_MTO-N1RDemBase\_MTO；

DN1RNAImct=N1RDemNAI1\_MCT-N1RDemBase\_MCT；

DN1RNAImcl=DN1RNAImcl1；

DN1RNAImcr=DN1RNAImcr1

**前置条件**：1

**输入**：nai\_control, N1RDemNAI1\_MTO, N1RDemBase\_MTO, N1RDemNAI1\_MCT, N1RDemBase\_MCT, DN1RNAImcl1, DN1RNAImcr1

**输出**：DN1RNAImto, DN1RNAImct, DN1RNAImcl, DN1RNAImcr

**公式**：

if(nai\_control == 1)

{

DN1RNAImto=N1RDemNAI1\_MTO-N1RDemBase\_MTO;

DN1RNAImct=N1RDemNAI1\_MCT-N1RDemBase\_MCT;

DN1RNAImcl=DN1RNAImcl1;

DN1RNAImcr=DN1RNAImcr1;

}

**任务名称**：计算最大起飞引气修正量

**编号**：R\_10830

**功能**：应用软件应根据以下公式计算最大起飞引气修正量 DN1Rbleedmto: DN1Rbleedmto=DN1RECSmto+DN1RWAImto+DN1RNAImto。

**前置条件**：1

**输入**：DN1RECSmto, DN1RWAImto, DN1RNAImto

**输出**：DN1Rbleedmto

**公式**：DN1Rbleedmto=DN1RECSmto+DN1RWAImto+DN1RNAImto;

**任务名称**：计算最大连续引气修正量

**编号**：R\_10831

**功能**：应用软件应根据以下公式计算最大连续引气修正量 DN1Rbleedmct: DN1Rbleedmct=DN1RECSmct+DN1RWAImct+DN1RNAImct。

**前置条件**：1

**输入**：DN1RECSmct, DN1RWAImct, DN1RNAImct

**输出**：DN1Rbleedmct

**公式**：DN1Rbleedmct=DN1RECSmct+DN1RWAImct+DN1RNAImct;

**任务名称**：计算最大爬升引气修正量

**编号**：R\_10832

**功能**：应用软件应根据以下公式计算最大爬升引气修正量 DN1Rbleedmcl: DN1Rbleedmcl=DN1RECSmcl+DN1RWAImcl+DN1RNAImcl。

**前置条件**：1

**输入**：DN1RECSmcl, DN1RWAImcl, DN1RNAImcl

**输出**：DN1Rbleedmcl

**公式**：DN1Rbleedmcl=DN1RECSmcl+DN1RWAImcl+DN1RNAImcl;

**任务名称**：计算最大巡航引气修正量

**编号**：R\_10833

**功能**：应用软件应根据以下公式计算最大巡航引气修正量DN1Rbleedmcr: DN1Rbleedmcr=DN1RECSmcr+DN1RWAImcr+DN1RNAImcr。

**前置条件**：1

**输入**：DN1RECSmcr, DN1RWAImcr, DN1RNAImcr

**输出**：DN1Rbleedmcr

**公式**：DN1Rbleedmcr=DN1RECSmcr+DN1RWAImcr+DN1RNAImcr;

**任务名称**：设置减推力起飞DTO1、DTO2、灵活起飞FLEXTO、复飞GA、增推力起飞Bump的引气修正量

**编号**：R\_10834

**功能**：减推力起飞DTO1、DTO2、灵活起飞FLEXTO、复飞GA、增推力起飞Bump的引气修正量与最大起飞MTO相同，即DN1Rbleeddto1=DN1Rbleeddto2= DN1Rbleedflexto=DN1Rbleedga=DN1Rbleedbump=DN1Rbleedmto。

**前置条件**：1

**输入**：DN1Rbleedmto

**输出**：DN1Rbleeddto1, DN1Rbleeddto2, DN1Rbleedflexto, DN1Rbleedga, DN1Rbleedbump

**公式**：

DN1Rbleeddto1 = DN1Rbleedmto;

DN1Rbleeddto2 = DN1Rbleedmto;

DN1Rbleedflexto = DN1Rbleedmto;

DN1Rbleedga = DN1Rbleedmto;

DN1Rbleedbump = DN1Rbleedmto;

**任务名称**：设置减推力爬升DCL1、DCL2的引气修正量

**编号**：R\_10835

**功能**：减推力爬升DCL1、DCL2的引气修正量与最大爬升MCL相同，即DN1Rbleeddcl1=DN1Rbleeddcl2=DN1Rbleedmcl

**前置条件**：1

**输入**：DN1Rbleedmcl

**输出**：DN1Rbleeddcl1, DN1Rbleeddcl2

**公式**：

DN1Rbleeddcl1 = DN1Rbleedmcl;

DN1Rbleeddcl2 = DN1Rbleedmcl;

**任务名称**：推力配平判断

**编号**：R\_6526

**功能**：应用软件进行推力配平判断，置推力配平标志atrim=1或atrim=0

**前置条件**：1

**输入**：N1\_trim\_enable, N1Trim\_signal, airdis\_WOW, thrust\_level, dspeed\_state, dPLA, PD\_atrimpla

**输出**：atrim

**公式**：

if((N1\_trim\_enable == 1) && (N1Trim\_signal == 1) && (airdis\_WOW == 0) && ((thrust\_level == level\_MCR) || (thrust\_level == level\_MCT) || (thrust\_level == level\_MCL) || (thrust\_level == level\_CR) || (thrust\_level == level\_CT) || (thrust\_level == level\_CL) || (thrust\_level == level\_DCL1) || (thrust\_level == level\_DCL2)) && ((dspeed\_state == 0) || (((dspeed\_state == 1) || (dspeed\_state == 2)) && (dPLA < PD\_atrimpla))))

{

atrim=1;

}

else

{

atrim=0;

}

**任务名称**：计算推力配平修正量

**编号**：R\_6519

**功能**：推力配平修正量DN1RDemTrim计算公式应为：

DN1DemTrim = min(max(PD\_N1TrimLowCo, N1Trim), PD\_N1TrimUpCo) \* N1Design \* 0.01 \* atrim;

DN1RDemTrim=DN1DemTrim/sqrt(T12/288.15)。

**前置条件**：1

**输入**：PD\_N1TrimLowCo, N1Trim, PD\_N1TrimUpCo, N1Design, atrim, T12

**输出**：DN1RDemTrim

**公式**：

DN1DemTrim = min(max(PD\_N1TrimLowCo, N1Trim), PD\_N1TrimUpCo) \* N1Design \* 0.01 \* atrim;

DN1RDemTrim = DN1DemTrim / sqrt(T12 / 288.15);

**任务名称**：设置最大巡航、最大爬升、减推力爬升1、减推力爬升2、最大连续的推力配平修正量

**编号**：R\_10837

**功能**：设置最大巡航、最大爬升、减推力爬升1、减推力爬升2、最大连续的推力配平修正量

**前置条件**：1

**输入**：DN1RDemTrim

**输出**：DN1RTrimmcr, DN1RTrimmcl, DN1RTrimdcl1, DN1RTrimdcl2, DN1RTrimmct

**公式**：

DN1RTrimmcr = DN1RDemTrim;

DN1RTrimmcl = DN1RDemTrim;

DN1RTrimdcl1 = DN1RDemTrim;

DN1RTrimdcl2 = DN1RDemTrim;

DN1RTrimmct = DN1RDemTrim;

**任务名称**：设置最大起飞、减推力起飞1、减推力起飞2、复飞、灵活起飞、增推力起飞的推力配平修正量

**编号**：R\_10838

**功能**：设置最大起飞、减推力起飞1、减推力起飞2、复飞、灵活起飞、增推力起飞的推力配平修正量

**前置条件**：1

**输入**：

**输出**：DN1RTrimmto, DN1RTrimdto1, DN1RTrimdto2, DN1RTrimga, DN1RTrimflexto, DN1RTrimbump

**公式**：

DN1RTrimmto = 0;

DN1RTrimdto1 = 0;

DN1RTrimdto2 = 0;

DN1RTrimga = 0;

DN1RTrimflexto = 0;

DN1RTrimbump = 0;

**任务名称**：油门杆插值目标计算-最大起飞

**编号**：R\_10850

**功能**：应用软件应按照以下公式计算用于油门杆插值的最大起飞N1R控制目标N1RDem\_MTO。其中PD\_N1RDemTOLmt为可调。

**前置条件**：1

**输入**：N1RDem\_MAX, N1RDemBase\_MTO, DN1Rbleedmto, DN1RTrimmto, PD\_N1RDemTOLmt, N1RDesign

**输出**：N1RDem\_MTO

**公式**：N1RDem\_MTO=min(N1RDem\_MAX,max(N1RDemBase\_MTO+DN1Rbleedmto+DN1RTrimmto,PD\_N1RDemTOLmt\*N1RDesign));

**任务名称**：油门杆插值目标计算-复飞

**编号**：R\_10865

**功能**：应用软件应按照以下公式计算用于油门杆插值的复飞N1R控制目标N1RDem\_GA.

**前置条件**：1

**输入**：N1RDem\_MAX, N1RDemBase\_GA, DN1Rbleedga, DN1RTrimga, PD\_N1RDemTOLmt, N1RDesign

**输出**：N1RDem\_GA

**公式**：N1RDem\_GA=min(N1RDem\_MAX,max(N1RDemBase\_GA+DN1Rbleedga+DN1RTrimga,PD\_N1RDemTOLmt\*N1RDesign));

**任务名称**：油门杆插值目标计算-减推力起飞1

**编号**：R\_10871

**功能**：应用软件应按照以下公式计算用于油门杆插值的减推力起飞1 N1R控制目标N1RDem\_DTO1

**前置条件**：1

**输入**：N1RDem\_MAX, N1RDemBase\_DTO1, DN1Rbleeddto1, DN1RTrimdto1, PD\_N1RDemTOLmt, N1RDesign

**输出**：N1RDem\_DTO1

**公式**：N1RDem\_DTO1=min(N1RDem\_MAX,max(N1RDemBase\_DTO1+DN1Rbleeddto1+DN1RTrimdto1,PD\_N1RDemTOLmt\*N1RDesign));

**任务名称**：油门杆插值目标计算-减推力起飞2

**编号**：R\_10874

**功能**：应用软件应按照以下公式计算用于油门杆插值的减推力起飞2 N1R控制目标N1RDem\_DTO2

**前置条件**：1

**输入**：N1RDem\_MAX, N1RDemBase\_DTO2, DN1Rbleeddto2, DN1RTrimdto2, PD\_N1RDemTOLmt, N1RDesign

**输出**：N1RDem\_DTO2

**公式**：N1RDem\_DTO2=min(N1RDem\_MAX,max(N1RDemBase\_DTO2+DN1Rbleeddto2+DN1RTrimdto2,PD\_N1RDemTOLmt\* N1RDesign));

**任务名称**：油门杆插值目标计算-增推力起飞

**编号**：R\_10877

**功能**：应用软件应按照以下公式计算用于油门杆插值的增推力起飞N1R控制目标N1RDem\_Bump。

**前置条件**：1

**输入**：N1RDem\_MAX, N1RDemBase\_Bump, DN1Rbleedbump, DN1RTrimbump, PD\_N1RDemTOLmt, N1RDesign

**输出**：N1RDem\_Bump

**公式**：N1RDem\_Bump = min( N1RDem\_MAX ,max(N1RDemBase\_Bump+DN1Rbleedbump+DN1RTrimbump,PD\_N1RDemTOLmt\*N1RDesign));

**任务名称**：油门杆插值目标计算-灵活起飞

**编号**：R\_10880

**功能**：应用软件应按照以下公式计算用于油门杆插值的灵活起飞N1R控制目标N1RDem\_FLEXTO。其中PD\_N1RdemFlexLmt为可调。

**前置条件**：1

**输入**：N1RDem\_MAX, N1RDemBase\_FLEXTO, DN1Rbleedflexto, DN1RTrimflexto, PD\_N1RDemFlexLmt, N1RDesign

**输出**：N1RDem\_FLEXTO

**公式**：N1RDem\_FLEXTO = min(N1RDem\_MAX,max(N1RDemBase\_FLEXTO+DN1Rbleedflexto+DN1RTrimflexto,PD\_N1RDemFlexLmt\*N1RDesign));

**任务名称**：油门杆插值目标计算-最大连续

**编号**：R\_10884

**功能**：应用软件应按照以下公式计算用于油门杆插值的最大连续N1R控制目标N1RDem\_MCT。

**前置条件**：1

**输入**：N1RDemBase\_MCT, DN1Rbleedmct, DN1RTrimmct, N1RDem\_PLATO

**输出**：N1RDem\_MCT

**公式**：N1RDem\_MCT = min(N1RDemBase\_MCT +DN1Rbleedmct +DN1RTrimmct,N1RDem\_PLATO);

**任务名称**：油门杆插值目标计算-最大爬升

**编号**：R\_10887

**功能**：应用软件应按照以下公式计算用于油门杆插值的最大爬升N1R控制目标N1RDem\_MCL。

**前置条件**：1

**输入**：N1RDemBase\_MCL, DN1Rbleedmcl, DN1RTrimmcl, N1RDem\_MCT

**输出**：N1RDem\_MCL

**公式**：N1RDem\_MCL = min(N1RDemBase\_MCL +DN1Rbleedmcl+DN1RTrimmcl,N1RDem\_MCT);

**任务名称**：油门杆插值目标计算-减推力爬升1

**编号**：R\_10890

**功能**：应用软件应按照以下公式计算用于油门杆插值的减推力爬升1 N1R控制目标N1RDem\_DCL1。

**前置条件**：1

**输入**：N1RDemBase\_DCL1, DN1Rbleeddcl1, DN1RTrimdcl1, N1RDem\_MCT

**输出**：N1RDem\_DCL1

**公式**：N1RDem\_DCL1 = min(N1RDemBase\_DCL1 + DN1Rbleeddcl1 + DN1RTrimdcl1 ，N1RDem\_MCT );

**任务名称**：油门杆插值目标计算-减推力爬升2

**编号**：R\_10893

**功能**：应用软件应按照以下公式计算用于油门杆插值的减推力爬升2 N1R控制目标N1RDem\_DCL2。

**前置条件**：1

**输入**：N1RDemBase\_DCL2, DN1Rbleeddcl2, DN1RTrimdcl2, N1RDem\_MCT

**输出**：N1RDem\_DCL2

**公式**：N1RDem\_DCL2 = min(N1RDemBase\_DCL2 + DN1Rbleeddcl2 + DN1RTrimdcl2 ，N1RDem\_MCT );

**任务名称**：油门杆插值目标计算-最大巡航

**编号**：R\_10896

**功能**：应用软件应按照以下公式计算用于油门杆插值的最大巡航N1R控制目标N1RDem\_MCR。

**前置条件**：1

**输入**：N1RDemBase\_MCR, DN1Rbleedmcr, DN1RTrimmcr, N1RDem\_PLACL

**输出**：N1RDem\_MCR

**公式**：N1RDem\_MCR = min(N1RDemBase\_MCR+ DN1Rbleedmcr + DN1RTrimmcr ，N1RDem\_PLACL );

**任务名称**：油门杆插值目标计算-最大反推(MREV)

**编号**：R\_6279

**功能**：应用软件应根据最大反推调节计划(PS\_MREV\_H1~10)，通过高度H、马赫数Ma计算、环境温差Dtamb线性插值计算最大反推N1R控制目标N1RDem\_MREV。

**前置条件**：1

**输入**：PS\_MREV\_H, H, Ma, Dtamb

**输出**：N1RDem\_MREV

**公式**：N1RDem\_MREV=interpolation(PS\_MREV\_H, H, Ma, Dtamb);

**任务名称**：油门杆插值目标计算-最大推力增量计算

**编号**：R\_9500

**功能**：应用软件应根据最大推力增量调节计划表(PS\_MAX\_Ma1~3)，通过高度H、马赫数Ma计算、环境温差Dtamb线性插值计算最大推力N1R控制目标增量dN1RDem\_MAX。

**前置条件**：1

**输入**：PS\_MAX\_Ma, H, Ma, Dtamb

**输出**：dN1RDem\_MAX

**公式**：dN1RDem\_MAX = interpolation(PS\_MAX\_Ma, H, Ma, Dtamb);

**任务名称**：油门杆插值目标计算-最大推力计算

**编号**：R\_7656

**功能**：应用软件应根据公式计算最大推力N1R控制目标N1RDem\_MAX。

**前置条件**：1

**输入**：N1RDem\_MTO, dN1RDem\_MAX

**输出**：N1RDem\_MAX

**公式**：N1RDem\_MAX = N1RDem\_MTO + dN1RDem\_MAX;

**任务名称**：油门杆插值目标计算-N1RDem\_PLATO计算

**编号**：R\_8443

**功能**：应用软件应根据飞机信号按照表7825计算起飞域N1R控制目标N1RDem\_PLATO。

**前置条件**：1

**输入**：TL\_require, DTO1, DTO2, GA, Bump, FLEXTO, flex\_dT\_flag, N1RDem\_MTO, N1RDem\_DTO1, N1RDem\_DTO2, N1RDem\_GA, N1RDem\_FLEXTO, N1RDem\_Bump

**输出**：N1RDem\_PLATO

**公式**：

if(TL\_require!=DTO1&&TL\_require!=DTO2&&TL\_require!=GA&&TL\_require!=Bump&&(TL\_require!=FLEXTO||(TL\_require==FLEXTO&&flex\_dT\_flag==0)))

{

N1RDem\_PLATO=N1RDem\_MTO;

}

else

{

if(TL\_require==DTO1)

{

N1RDem\_PLATO=N1RDem\_DTO1;

}

else

{

if(TL\_require==DTO2)

{

N1RDem\_PLATO=N1RDem\_DTO2;

}

else  
 {

if(TL\_require==GA)

{

N1RDem\_PLATO=N1RDem\_GA;

}

else

{

if(TL\_require==FLEXTO&&flex\_dT\_flag==1)

{

N1RDem\_PLATO=N1RDem\_FLEXTO;

}

else

{

if(TL\_require==Bump)

{

N1RDem\_PLATO=N1RDem\_Bump;

}

}

}

}

}

}

**任务名称**：油门杆插值目标计算-N1RDem\_PLACL计算

**编号**：R\_8447

**功能**：应用软件应根据飞机信号按照表8217计算爬升域N1R控制目标N1RDem\_PLACL。

**前置条件**：1

**输入**：TL\_require, DCL1, DCL2, N1RDem\_MCL, N1RDem\_DCL1, N1RDem\_DCL2

**输出**：N1RDem\_PLACL

**公式**：

if(TL\_require!=DCL1&&TL\_require!=DCL2)

{

N1RDem\_PLACL=N1RDem\_MCL;

}

else

{

if(TL\_require==DCL1)

{

N1RDem\_PLACL=N1RDem\_DCL1;

}

else

{

if(TL\_require==DCL2)

{

N1RDem\_PLACL=N1RDem\_DCL2;

}

}

}

**任务名称**：油门杆插值目标计算-转换点转速N1Rtr计算-慢车域实时N2R25控制目标计算

**编号**：R\_9501

**功能**：应用软件应根据飞机信号按照表9953计算慢车域实时N2R25控制目标N2R25Dem\_PLAIdle（慢车转速）。

**前置条件**：1

**输入**：airdis\_WOW, PLA, N2R25Dem\_RI, N2R25Dem\_GI, approach\_air, N2R25Dem\_FI, N2R25Dem\_AI

**输出**：N2R25Dem\_PLAIdle

**公式**：

if(airdis\_WOW==1&&PLA<0)

{

N2R25Dem\_PLAIdle=N2R25Dem\_RI;

}

else

{

if(airdis\_WOW==1&&PLA>=0)

{

N2R25Dem\_PLAIdle=N2R25Dem\_GI;

}

else

{

if(airdis\_WOW==0&&approach\_air==1)

{

N2R25Dem\_PLAIdle=N2R25Dem\_FI;

}

else

{

if(airdis\_WOW==0&&approach\_air==0)

{

N2R25Dem\_PLAIdle=N2R25Dem\_AI;

}

}

}

}

**任务名称**：油门杆插值目标计算-转换点转速N1Rtr计算-高低压换算转速

**编号**：R\_6376

**功能**：应用软件应将N2R25Dem\_PLAIdle通过“高低压换算转速N2R25-N1R转换关系”表N2R25N1Rbak（可调）转换为N1RDem\_PLAIdle。

**前置条件**：1

**输入**：N2R25N1Rbak, N2R25Dem\_PLAIdle

**输出**：N1RDem\_PLAIdle

**公式**：N1RDem\_PLAIdle=interpolation(N2R25N1Rbak, N2R25Dem\_PLAIdle);

**任务名称**：油门杆插值目标计算-转换点转速N1Rtr计算-N1r-PLAIdle

**编号**：R\_10898

**功能**：转换点转速N1Rtr应为转换前一时刻慢车状态（距当前时刻最近的慢车状态时刻）的N1r与N1RDem\_PLAIdle两者的高选值。

**前置条件**：1

**输入**：N1r, N1RDem\_PLAIdle

**输出**：N1Rtr

**公式**：

if(N1r>N1RDem\_PLAIdle)

{

N1Rtr=N1r;

}

else

{  
 N1Rtr=N1RDem\_PLAIdle;

}

**任务名称**：油门杆插值目标计算-转换点转速N1Rtr计算-MCR

**编号**：R\_6388

**功能**：若N2R25Dem\_PLAIdle选择N2R25Dem\_GI、N2R25Dem\_FI或N2R25Dem\_AI，且N1Rtr>N1RDem\_MCR，则应用软件应置N1Rtr = N1RDem\_MCR。

**前置条件**：1

**输入**：N2R25Dem\_PLAIdle, N2R25Dem\_GI, N2R25Dem\_FI, N2R25Dem\_AI, N1RDem\_MCR, N1Rtr

**输出**：N1Rtr

**公式**：

if(N2R25Dem\_PLAIdle==N2R25Dem\_GI||N2R25Dem\_PLAIdle==N2R25Dem\_FI||N2R25Dem\_PLAIdle==N2R25Dem\_AI&&(N1Rtr>N1RDem\_MCR))

{

N1Rtr=N1RDem\_MCR;

}

**任务名称**：油门杆插值目标计算-转换点转速N1Rtr计算-MREV

**编号**：R\_7831

**功能**：若N2R25Dem\_PLAIdle选择N2R25Dem\_RI，且N1Rtr>N1RDem\_MREV，则将应用软件应置N1Rtr = N1RDem\_MREV。

**前置条件**：1

**输入**：N2R25Dem\_PLAIdle, N2R25Dem\_RI, N1Rtr, N1RDem\_MREV

**输出**：N1Rtr

**公式**：

if(N2R25Dem\_PLAIdle==N2R25Dem\_RI&&N1Rtr>N1RDem\_MREV)

{

N1Rtr=N1RDem\_MREV;

}

**任务名称**：油门杆插值目标计算-准最大推力

**编号**：R\_6345

**功能**：应用软件应根据PLA在N1RDem\_PLATO和N1RDem\_MAX之间进行一维线性插值得到准最大推力N1R控制目标N1RDem\_TO2MAX。

**前置条件**：1

**输入**：PS\_PLA\_N1RDem\_PLATO\_MAX, PLA, N1RDem\_PLATO, N1RDem\_MAX

**输出**：N1RDem\_TO2MAX

**公式**：N1RDem\_TO2MAX=interpolation(PS\_PLA\_N1RDem\_PLATO\_MAX, PLA, N1RDem\_PLATO, N1RDem\_MAX);

**任务名称**：油门杆插值目标计算-起飞（TO）

**编号**：R\_6333

**功能**：应用软件应根据PLA在N1RDem\_MCT和N1RDem\_PLATO之间进行一维线性插值得到起飞N1R控制目标N1RDem\_TO。

**前置条件**：1

**输入**：PS\_PLA\_N1RDem\_MCT\_PLATO, PLA, N1RDem\_MCT, N1RDem\_PLATO

**输出**：N1RDem\_TO

**公式**：N1RDem\_TO=interpolation(PS\_PLA\_N1RDem\_MCT\_PLATO, PLA, N1RDem\_MCT, N1RDem\_PLATO);

**任务名称**：油门杆插值目标计算-连续（CT）

**编号**：R\_6335

**功能**：应用软件应根据PLA在N1RDem\_PLACL和N1RDem\_MCT之间进行一维线性插值得到连续N1R控制目标N1RDem\_CT。

**前置条件**：1

**输入**：PS\_PLA\_N1RDem\_PLACL\_MCT, PLA, N1RDem\_PLACL, N1RDem\_MCT

**输出**：N1RDem\_CT

**公式**：N1RDem\_CT=interpolation(PS\_PLA\_N1RDem\_PLACL\_MCT, PLA, N1RDem\_PLACL, N1RDem\_MCT);

**任务名称**：油门杆插值目标计算-爬升（CL）

**编号**：R\_6337

**功能**：应用软件应根据PLA在N1RDem\_MCR和N1RDem\_PLACL之间进行一维线性插值得到爬升N1R控制目标N1RDem\_CL。

**前置条件**：1

**输入**：PS\_PLA\_N1RDem\_MCR\_PLACL, PLA, N1RDem\_MCR, N1RDem\_PLACL

**输出**：N1RDem\_CL

**公式**：N1RDem\_CL=interpolation(PS\_PLA\_N1RDem\_MCR\_PLACL, PLA, N1RDem\_MCR, N1RDem\_PLACL);

**任务名称**：油门杆插值目标计算-巡航（CR）

**编号**：R\_6339

**功能**：应用软件应根据PLA在转换点转速N1Rtr和最大巡航N1R控制目标N1RDem\_MCR之间进行一维线性插值得到巡航N1R控制目标N1RDem\_CR。

**前置条件**：1

**输入**：PS\_PLA\_N1Rtr\_N1RDem\_MCR, PLA, N1Rtr, N1RDem\_MCR

**输出**：N1RDem\_CR

**公式**：N1RDem\_CR=interpolation(PS\_PLA\_N1Rtr\_N1RDem\_MCR, PLA, N1Rtr, N1RDem\_MCR);

**任务名称**：油门杆插值目标计算-反推（REV）

**编号**：R\_6352

**功能**：应用软件应根据PLA在转换点转速N1Rtr和N1RDem\_MREV之间进行一维线性插值得到反推N1R控制目标值N1RDem\_REV。

**前置条件**：1

**输入**：PS\_PLA\_N1Rtr\_N1RDem\_MREV, PLA, N1Rtr, N1RDem\_MREV

**输出**：N1RDem\_REV

**公式**：N1RDem\_REV=interpolation(PS\_PLA\_N1Rtr\_N1RDem\_MREV, PLA, N1Rtr, N1RDem\_MREV);

**任务名称**：油门杆插值目标计算-N1RDem\_BAK计算

**编号**：R\_6347

**功能**：应用软件应根据N2R25Dem和Ma通过高低压换算转速N2R25-N1R转换关系表N2R25N1Rbak（可调）反算出N1R备份控制目标N1RDem\_BAK。

**前置条件**：1

**输入**：N2R25N1Rbak, N2R25Dem, Ma

**输出**：N1RDem\_BAK

**公式**：N1RDem\_BAK = interpolation(N2R25N1Rbak, N2R25Dem, Ma);

**任务名称**：N1R指示量N1RDemInd计算（最大反推）

**编号**：R\_9970

**功能**：当选定推力等级为最大反推时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_MREV, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_MREV)

{

N1RDemInd = N1RDem\_MREV;

}

**任务名称**：N1R指示量N1RDemInd计算（反推）

**编号**：R\_9976

**功能**：当选定推力等级为反推时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_REV, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_REV)

{

N1RDemInd = N1RDem\_REV;

}

**任务名称**：N1R指示量N1RDemInd计算（反推慢车、地面慢车、空中慢车、进近慢车）

**编号**：R\_9972

**功能**：当选定推力等级为反推慢车、地面慢车、空中慢车、进近慢车时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_BAK, thrust\_level

**输出**：N1RDemInd

**公式**：

if( thrust\_level == level\_RI || thrust\_level == level\_GI || thrust\_level == level\_FI || thrust\_level == level\_AI)

{

N1RDemInd = N1RDem\_BAK;

}

**任务名称**：N1R指示量N1RDemInd计算（巡航）

**编号**：R\_9974

**功能**：当选定推力等级为巡航时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_CR, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_CR)

{

N1RDemInd = N1RDem\_CR;

}

**任务名称**：N1R指示量N1RDemInd计算（最大巡航）

**编号**：R\_9969

**功能**：当选定推力等级为最大巡航时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_MCR, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_MCR)

{

N1RDemInd = N1RDem\_MCR;

}

**任务名称**：N1R指示量N1RDemInd计算（爬升）

**编号**：R\_9977

**功能**：当选定推力等级为爬升时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_CL, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_CL)

{

N1RDemInd = N1RDem\_CL;

}

**任务名称**：N1R指示量N1RDemInd计算（最大爬升）

**编号**：R\_9966

**功能**：当选定推力等级为最大爬升时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_MCL, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_MCL)

{

N1RDemInd = N1RDem\_MCL;

}

**任务名称**：N1R指示量N1RDemInd计算（减推力爬升1）

**编号**：R\_9967

**功能**：当选定推力等级为减推力爬升1时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_DCL1, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_DCL1)

{

N1RDemInd = N1RDem\_DCL1;

}

**任务名称**：N1R指示量N1RDemInd计算（减推力爬升2）

**编号**：R\_9968

**功能**：当选定推力等级为减推力爬升2时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_DCL2, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_DCL2)

{

N1RDemInd = N1RDem\_DCL2;

}

**任务名称**：N1R指示量N1RDemInd计算（连续）

**编号**：R\_9973

**功能**：当选定推力等级为连续时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_CT, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_CT)

{

N1RDemInd = N1RDem\_CT;

}

**任务名称**：N1R指示量N1RDemInd计算（最大连续）

**编号**：R\_9965

**功能**：当选定推力等级为最大连续时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_MCT, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_MCT)

{

N1RDemInd = N1RDem\_MCT;

}

**任务名称**：N1R指示量N1RDemInd计算（起飞）

**编号**：R\_9978

**功能**：当选定推力等级为起飞时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_TO, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_TO)

{

N1RDemInd = N1RDem\_TO;

}

**任务名称**：N1R指示量N1RDemInd计算（最大起飞）

**编号**：R\_9959

**功能**：当选定推力等级为最大起飞时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_MTO, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_MTO)

{

N1RDemInd = N1RDem\_MTO;

}

**任务名称**：N1R指示量N1RDemInd计算（减推力起飞1）

**编号**：R\_9961

**功能**：当选定推力等级为减推力起飞1时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_DTO1, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_DTO1)

{

N1RDemInd = N1RDem\_DTO1;

}

**任务名称**：N1R指示量N1RDemInd计算（减推力起飞2）

**编号**：R\_9962

**功能**：当选定推力等级为减推力起飞2时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_DTO2, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_DTO2)

{

N1RDemInd = N1RDem\_DTO2;

}

**任务名称**：N1R指示量N1RDemInd计算（复飞）

**编号**：R\_9960

**功能**：当选定推力等级为复飞时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_GA, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_GA)

{

N1RDemInd = N1RDem\_GA;

}

**任务名称**：N1R指示量N1RDemInd计算（增推力起飞）

**编号**：R\_9963

**功能**：当选定推力等级为增推力起飞时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_Bump, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_Bump)

{

N1RDemInd = N1RDem\_Bump;

}

**任务名称**：N1R指示量N1RDemInd计算（灵活起飞）

**编号**：R\_9964

**功能**：当选定推力等级为灵活起飞时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_FLEXTO, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_FLEXTO)

{

N1RDemInd = N1RDem\_FLEXTO;

}

**任务名称**：N1R指示量N1RDemInd计算（准最大推力）

**编号**：R\_9975

**功能**：当选定推力等级为准最大推力时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_TO2MAX, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_TO2MAX)

{

N1RDemInd = N1RDem\_TO2MAX;

}

**任务名称**：N1R指示量N1RDemInd计算（最大推力）

**编号**：R\_9971

**功能**：当选定推力等级为最大推力时，应用软件需要计算N1R指示量N1RDemInd用于飞机指示。

**前置条件**：1

**输入**：N1RDem\_MAX, thrust\_level

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_MAX)

{

N1RDemInd = N1RDem\_MAX;

}

**任务名称**：N1R指示量N1RDemInd计算（无指定推力等级补充1）

**编号**：D\_7842

**功能**：当选定推力等级为无指定推力等级，且发动机状态为正常地面起动状态、稳态风车起动、快速风车起动、空中辅助起动状态，应用软件应置N1RDemInd = N1RDem\_BAK。

**前置条件**：1

**输入**：N1RDem\_BAK, thrust\_level, engine\_state

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_default && (engine\_state == ES\_ground\_start || engine\_state == ES\_windmill\_start || engine\_state == ES\_Qwindmill\_start || engine\_state == ES\_assist\_start))

{

N1RDemInd = N1RDem\_BAK;

}

**任务名称**：N1R指示量N1RDemInd计算（无指定推力等级补充2）

**编号**：D\_7843

**功能**：当选定推力等级为无指定推力等级，且发动机状态为初始状态、地面待机状态、空中待机状态、冷运转、假开车/油封/启封、正常停车、紧急停车、地面起动中止、空中起动中止状态，应用软件应置N1RDemInd = 0。

**前置条件**：1

**输入**：thrust\_level, engine\_state

**输出**：N1RDemInd

**公式**：

if(thrust\_level == level\_default && (engine\_state == ES\_origin || engine\_state == ES\_ground\_standby || engine\_state == ES\_flight\_standby || engine\_state == ES\_cold\_operation || engine\_state == ES\_DR\_OS\_Us || engine\_state == ES\_normal\_stop || engine\_state == ES\_emergency\_stop || engine\_state == ES\_ground\_start\_termination || engine\_state == ES\_flight\_start\_terminnation))

{

N1RDemInd = 0;

}

**任务名称**：N1公差修正量计算

**编号**：R\_7661

**功能**：应用软件应根据N1公差修正等级PD\_N1RMod1~8选择对应的N1公差修正量DN1RMod。

**前置条件**：1

**输入**：DN1RMod

**输出**：

**公式**：

**任务名称**：N1R控制量N1RDem计算（推力等级为无指定推力等级、最大反推、反推、反推慢车、地面慢车、空中慢车、进近慢车、巡航、最大巡航、最大推力）

**编号**：R\_10901

**功能**：当选定推力等级为无指定推力等级、最大反推、反推、反推慢车、地面慢车、空中慢车、进近慢车、巡航、最大巡航、最大推力，N1RDem = N1RDemInd。

**前置条件**：1

**输入**：N1RDemInd, thrust\_level

**输出**：N1RDem

**公式**：

if(thrust\_level == level\_default || thrust\_level == level\_MREV || thrust\_level == level\_REV || thrust\_level == level\_RI || thrust\_level == level\_GI || thrust\_level == level\_FI || thrust\_level == level\_AI || thrust\_level == level\_CR || thrust\_level == level\_MCR || thrust\_level == level\_MAX)

{

N1RDem = N1RDemInd;

}

**任务名称**：N1R控制量N1RDem计算（推力等级为最大爬升、减推力爬升1、减推力爬升2、连续、最大连续、起飞、最大起飞、减推力起飞1、减推力起飞2、复飞、增推力起飞、灵活起飞）

**编号**：R\_10902

**功能**：当选定推力等级为最大爬升、减推力爬升1、减推力爬升2、连续、最大连续、起飞、最大起飞、减推力起飞1、减推力起飞2、复飞、增推力起飞、灵活起飞，N1RDem = N1RDemInd + DN1RMod。

**前置条件**：1

**输入**：N1RDemInd, thrust\_level, DN1RMod

**输出**：N1RDem

**公式**：

if(thrust\_level == level\_MCL || thrust\_level == level\_DCL1 || thrust\_level == level\_DCL2 || thrust\_level == level\_CT || thrust\_level == level\_MCT || thrust\_level == level\_TO || thrust\_level == level\_MTO || thrust\_level == level\_DTO1 || thrust\_level == level\_DTO2 || thrust\_level == level\_GA || thrust\_level == level\_Bump || thrust\_level == level\_FLEXTO)

{

N1RDem = N1RDemInd + DN1RMod;

}

**任务名称**：N1R控制量N1RDem计算（推力等级为爬升）

**编号**：R\_10903

**功能**：当选定推力等级为爬升，应用软件应根据PLA在N1RDem\_MCR和N1RDem\_PLACL+DN1RMod之间进行一维线性插值得到N1RDem。

**前置条件**：1

**输入**：PLA, thrust\_level, N1RDem\_MCR, N1RDem\_PLACL, DN1RMod, level\_CL, PS\_PLA\_N1RDem\_MCR\_N1RDem\_PLACL\_Plus\_DN1RMod, Temp

**输出**：N1RDem

**公式**：

if(thrust\_level == level\_CL)

{

Temp = N1RDem\_PLACL+DN1RMod;

N1RDem =interpolation(PS\_PLA\_N1RDem\_MCR\_N1RDem\_PLACL\_Plus\_DN1RMod, PLA, N1RDem\_MCR,Temp);

}

**任务名称**：N1R控制量N1RDem计算（推力等级为准最大推力）

**编号**：R\_10904

**功能**：当选定推力等级为准最大推力，应用软件应根据PLA在N1RDem\_PLATO+DN1RMod和N1RDem\_MAX之间进行一维线性插值得到N1RDem。

**前置条件**：1

**输入**：PLA, thrust\_level, N1RDem\_PLATO, N1RDem\_MAX, DN1RMod, level\_TO2MAX, PS\_PLA\_N1RDem\_PLATO\_Plus\_DN1RMod\_N1RDem\_MAX, Temp

**输出**：N1RDem

**公式**：

if(thrust\_level == level\_TO2MAX)

{

Temp = N1RDem\_PLATO+DN1RMod;

N1RDem =interpolation(PS\_PLA\_N1RDem\_PLATO\_Plus\_DN1RMod\_N1RDem\_ MAX, PLA, Temp, N1RDem\_MAX);

}

**任务名称**：高低压切换控制（无N2双通道传感器故障）

**编号**：R\_10254

**功能**：当无N2双通道传感器故障，控制模式应按以下逻辑设置：

a) 发动机状态为慢车状态，应用软件应设置控制模式为高压控制模式，置N2R25最终控制目标N2r25Dmd = N2R25Dem，置N1R最终控制目标N1rDmd = N1RDem。[C-10205]

b) 发动机状态为慢车以上状态，且处于转换逻辑一过程中，应用软件应按照转换逻辑一控制。[C-10206]

c) 发动机状态为慢车以上状态，且处于转换逻辑二过程中，应用软件应按照转换逻辑二控制。[C-10246]

d) 发动机状态为慢车以上状态，PLA不处于慢车域，且不处于转换逻辑一、二过程中，应用软件应设置控制模式为低压控制模式，置N1R最终控制目标N1rDmd = N1RDem，置N2R25最终控制目标N2r25Dmd = N2R25Dem。[C-10248]

e) 发动机状态为慢车以上状态，PLA处于慢车域，且不处于转换逻辑一、二过程中，应用软件应设置控制模式为高压控制模式，置N2R25最终控制目标N2r25Dmd = N2R25Dem，置N1R最终控制目标N1rDmd = N1RDem。[C-10349]

f) 发动机状态为高转速风车起动状态，选定推力等级为反推慢车、地面慢车、空中慢车、进近慢车以外的等级，应用软件应设置控制模式为低压控制模式，置N1R最终控制目标N1rDmd = N1RDem，置N2R25最终控制目标N2r25Dmd = N2R25Dem。[C-11939]

g) 发动机状态为高转速风车起动状态，选定推力等级为反推慢车、地面慢车、空中慢车、进近慢车，应用软件应设置控制模式为高压控制模式，置N2R25最终控制目标N2r25Dmd = N2R25Dem，置N1R最终控制目标N1rDmd = N1RDem。[C-11938]

h) 发动机状态为慢车状态和慢车以上、高转速风车起动状态以外的其它状态，应用软件应设置控制模式为高压控制模式，置N2R25最终控制目标N2r25Dmd = N2R25Dem，置N1R最终控制目标N1rDmd = N1RDem。[C-10471]

**前置条件**：N2\_dual\_channel\_sensor\_fault\_flag == 0

**输入**：engine\_state, PLA, thrust\_level, N2R25Dem, N1RDem, software\_control\_mode , PD\_IdleSwitchPlaThsld

**输出**：N2r25Dmd, N1rDmd, software\_control\_mode

**公式**：

if(engine\_state == ES\_idle)

{

software\_control\_mode = High\_Press\_Control\_Mode;

N2r25Dmd = N2R25Dem;

N1rDmd = N1RDem;

}

if(engine\_state == ES\_above\_idle && ((PLA<2 + PD\_IdleSwitchPlaThsld && PLA>2) && (last(PLA)<=2 && last(PLA)>=0)) || ((PLA>-6-PD\_IdleSwitchPlaThsld && PLA<-6) && (last(PLA)<0 && last(PLA)>-6)))

{  
 software\_control\_mode = Low\_Press\_Control\_Mode;

}

if(engine\_state == ES\_above\_idle && (PLA<0 && PLA >=-6 && last(PLA)>=-33 && last(PLA)<-6 || PLA<=2 && PLA >=0 && last(PLA)>2 && last(PLA)<=85))

{  
 software\_control\_mode = High\_Press\_Control\_Mode;

}

if(engine\_state == ES\_above\_idle && (PLA >= 2 + PD\_IdleSwitchPlaThsld || PLA <= 2 || last(PLA) > 2 || last(PLA) < 0) && (PLA <= -6 - PD\_IdleSwitchPlaThsld || PLA >= -6 || last(PLA) >= 0 || last(PLA) <= -6) && (PLA < -6 || PLA > 2))

{  
 software\_control\_mode = Low\_Press\_Control\_Mode;

N1rDmd = N1RDem;

N2r25Dmd = N2R25Dem;

}

if(engine\_state == ES\_above\_idle && ((PLA >= 2 + PD\_IdleSwitchPlaThsld || PLA <= 2 || last(PLA) > 2 || last(PLA) < 0) && (PLA <= -6 - PD\_IdleSwitchPlaThsld || PLA >= -6 || last(PLA) >= 0 || last(PLA) <= -6) && (PLA >= 0 || PLA < -6 || last(PLA) < -33 || last(PLA) >= -6) && (PLA > 2 || PLA < 0 || last(PLA) <= 2 || last(PLA) > 85) && PLA <= 2 && PLA >= -6))

{  
 software\_control\_mode = High\_Press\_Control\_Mode;

N1rDmd = N1RDem;

N2r25Dmd = N2R25Dem;

}

if(engine\_state == ES\_Hwindmill\_start && thrust\_level != level\_RI && thrust\_level != level\_GI && thrust\_level != level\_FI && thrust\_level != level\_AI)

{  
 software\_control\_mode = Low\_Press\_Control\_Mode;

N1rDmd = N1RDem;

N2r25Dmd = N2R25Dem;

}

if(engine\_state == ES\_Hwindmill\_start && thrust\_level == level\_RI && thrust\_level == level\_GI && thrust\_level == level\_FI && thrust\_level == level\_AI)

{  
 software\_control\_mode = High\_Press\_Control\_Mode;

N1rDmd = N1RDem;

N2r25Dmd = N2R25Dem;

}

if(engine\_state != ES\_Hwindmill\_start && engine\_state != ES\_above\_idle && engine\_state != ES\_idle)

{  
 software\_control\_mode = High\_Press\_Control\_Mode;

N1rDmd = N1RDem;

N2r25Dmd = N2R25Dem;

}

**任务名称**：高低压切换控制（N2双通道传感器故障）

**编号**：R\_10905

**功能**：当N2双通道传感器故障，应用软件应该设置控制模式为低压控制模式，置N2R25最终控制目标N2r25Dmd = N2R25Dem,并按如下公式计算N1R最终控制目标N1rDmd：

N1rDmd = min(N1r + N1RDesign \* PD\_DN1RCoN2Flt, max(N1RDem, N1r-N1RDesign\*PD\_DN1RCoN2Flt))[C-10907]其中PD\_DN1RCoN2Flt为N2双通道传感器故障对应的N1R目标值变化率限制。

**前置条件**： N2\_dual\_channel\_sensor\_fault\_flag == 1

**输入**： N1r, N2R25Dem, N1RDem, N1RDesign, PD\_DN1RCoN2Flt

**输出**：N2r25Dmd, N1rDmd, software\_control\_mode

**公式**：

software\_control\_mode = Low\_Press\_Control\_Mode;  
N2r25Dmd = N2R25Dem;

N1rDmd = min(N1r + N1RDesign \* PD\_DN1RCoN2Flt, max(N1RDem, N1r-N1RDesign\*PD\_DN1RCoN2Flt));

**任务名称**：慢车到慢车以上切换控制（进入转换逻辑一）

**编号**：R\_8107

**功能**：当满足以下任一条件时，应用软件应进入转换逻辑一：

a) 油门杆从[0,2]推至(2。,2。+PD\_IdleSwitchPlaThsld（可调）)；[C-10219]

b) 油门杆从[-6,0)推至(-6。-PD\_IdleSwitchPlaThsld（可调）,-6。)。[C-8451]

**前置条件**：1

**输入**： PLA, PD\_IdleSwitchPlaThsld

**输出**：software\_control\_mode

**公式**：

if( ((PLA<2 + PD\_IdleSwitchPlaThsld && PLA>2) && (last(PLA)<=2 && last(PLA)>=0)) || ((PLA>-6-PD\_IdleSwitchPlaThsld && PLA<-6) && (last(PLA)<0 && last(PLA)>-6)) ){

software\_control\_mode = Low\_Press\_Control\_Mode;

}

**任务名称**：慢车到慢车以上切换控制（处于转换逻辑一，控制模式设置）

**编号**：R\_10209

**功能**：处于转换逻辑一过程中，应用软件应设置控制模式为低压控制模式。

**前置条件**：1

**输入**：PLA, PD\_IdleSwitchPlaThsld

**输出**：software\_control\_mode

**公式**：

if(((PLA<2 + PD\_IdleSwitchPlaThsld && PLA>2) && (last(PLA)<=2 && last(PLA)>=0)) || ((PLA>-6-PD\_IdleSwitchPlaThsld && PLA<-6) && (last(PLA)<0 && last(PLA)>-6)) ){

software\_control\_mode = Low\_Press\_Control\_Mode;

}

**任务名称**：慢车到慢车以上切换控制（处于转换逻辑一，N1R最终控制目标设置）

**编号**：R\_10210

**功能**：处于转换逻辑一过程中，应用软件应置N1R最终控制目标N1rDmd = N1Rrd。

**前置条件**：1

**输入**：N1Rrd, PLA, PD\_IdleSwitchPlaThsld

**输出**：N1rDmd

**公式**：

if(((PLA<2 + PD\_IdleSwitchPlaThsld && PLA>2) && (last(PLA)<=2 && last(PLA)>=0)) || ((PLA>-6-PD\_IdleSwitchPlaThsld && PLA<-6) && (last(PLA)<0 && last(PLA)>-6)) ){

N1rDmd = N1Rrd;

}

**任务名称**：慢车到慢车以上切换控制（处于转换逻辑一，N2R25最终控制目标设置）

**编号**：R\_10473

**功能**：处于转换逻辑一过程中，应用软件应置N2R25最终控制目标N2r25Dmd = N2R25Dem。

**前置条件**：1

**输入**：N2R25Dem, PLA, PD\_IdleSwitchPlaThsld

**输出**：N2r25Dmd

**公式**：

if(((PLA<2 + PD\_IdleSwitchPlaThsld && PLA>2) && (last(PLA)<=2 && last(PLA)>=0)) || ((PLA>-6-PD\_IdleSwitchPlaThsld && PLA<-6) && (last(PLA)<0 && last(PLA)>-6)) ){

N2r25Dmd = N2R25Dem;

}

**任务名称**：慢车到慢车以上切换控制（PLA大于等于零，计算转换过程N1R控制目标）

**编号**：R\_10211

**功能**：若油门杆PLA大于等于零，应用软件应按以下逻辑计算转换过程N1R控制目标N1Rrd:

a) 按以下计算低压换算转速N1R\_1:根据转换点转速N1Rtr、最大巡航转速控制目标N1RDem\_MCR，以本周期的PLA线性插值计算得到；[C-8110]

b) 按以下计算本周期低压换算转速N1R\_2，N1R\_2 = N1RPre + N1RPre\*PD\_DeltN1RCo(可调)，其中N1RPre是上一周期低压换算转速值；[C-8108]

c) 按以下计算本周期控制目标N1Rrd: N1Rrd = min(N1R\_1,N1R\_2);[C-8111]

**前置条件**：1

**输入**：N1Rtr, N1RDem\_MCR, PLA, N1RPre, PD\_DeltN1RCo, PS\_PLA\_N1Rtr\_N1RDem\_MCR

**输出**：N1Rrd

**公式**：

if(PLA >= 0){

N1R\_1 = interpolation( PS\_PLA\_N1Rtr\_N1RDem\_MCR, N1Rtr, N1RDem\_MCR, PLA);  
 N1R\_2 = N1RPre + N1RPre\*PD\_DeltN1RCo;

N1Rrd = min(N1R\_1,N1R\_2);

}

**任务名称**：慢车到慢车以上切换控制（PLA小于零，计算转换过程N1R控制目标N1Rrd）

**编号**：R\_10217

**功能**：若油门杆PLA小于零，应用软件应按以下逻辑计算转换过程N1R控制目标N1Rrd:

a) 按以下计算低压换算转速N1R\_1：根据转换点转速N1Rtr、最大反推转速目标值N1RDem\_MREV,以本周期的PLA线性插值计算得到；[C-8452]

b) 按以下计算本周期低压换算转速N1R\_2，N1R\_2 = N1RPre + N1RPre\*PD\_DeltN1RCo(可调)，其中N1RPre是上一周期低压换算转速值；[C-8453]

c) 按以下计算本周期控制目标N1Rrd: N1Rrd = min(N1R\_1,N1R\_2);[C-8454]

**前置条件**：1

**输入**：N1Rtr, N1RDem\_MREV, PLA, N1RPre, PD\_DeltN1RCo

**输出**：N1Rrd

**公式**：

if(PLA < 0){

N1R\_1=interpolation(PS\_PLA\_N1Rtr\_N1RDem\_MREV, N1Rtr, N1RDem\_MREV, PLA);  
 N1R\_2=N1RPre + N1RPre\*PD\_DeltN1RCo;

N1Rrd=min(N1R\_1,N1R\_2);

}

**任务名称**：慢车到慢车以上切换控制（应用软件应退出转换逻辑一）

**编号**：R\_8109

**功能**：当满足以下任一条件时，应用软件应退出转换逻辑一：

a) abs(N1Rrd-N1Rtr)<PD\_N1RpIdleOut(可调)\*N1RDesign；[C-10212]

b) PLA>2。维持PD\_IdleOutTime(可调)；[C-10213]

c) PLA<-6。维持PD\_IdleOutTime(可调)；[C-8455]

d) 本周期高压换算转速N2R25p大于PD\_IdleSwitchN2R25pMax(可调)；[C-10214]

e) 本周期低压换算转速N1Rp大于PD\_IdleSwitchN1RpMax(可调)；[C-10215]

f) PLA超出(2。,2。+PD\_IdleSwitchPlaThsld（可调）)和(-6。-PD\_IdleSwitchPlaThsld（可调）,-6。)；[C-10216]

g) PLA重新进入慢车域[-6,2];[C-10345]

h) 进入转换逻辑一的时间持续PD\_IdleOutTime(可调)以上。[C-10346]

**前置条件**：1

**输入**：N1Rtr, N1Rrd, PD\_N1RpIdleOut, N1RDesign, PLA, PD\_IdleOutTime, N2R25p, N1Rp, PD\_IdleSwitchN2R25pMax, PD\_IdleSwitchN1RpMax, PD\_IdleSwitchPlaThsld, software\_control\_mode

**输出**：software\_control\_mode

**公式**：

if((abs(N1Rrd - N1Rtr) < PD\_N1RpIdleOut \* N1RDesign) || (duration(PD\_IdleOutTime , ms, PLA > 2)) || (duration(PD\_IdleOutTime, ms, PLA < -6)) || (N2R25p > PD\_IdleSwitchN2R25pMax) || (N1Rp > PD\_IdleSwitchN1RpMax) || (PLA <= -6 - PD\_IdleSwitchPlaThsld) || (PLA >= 2 + PD\_IdleSwitchPlaThsld) || ((PLA <= 2) && (PLA >= -6)) || (duration(PD\_IdleOutTime , ms, logic\_transition\_mode == Logic1\_Control\_Mode))) {

software\_control\_mode = last(software\_control\_mode);

}

**任务名称**：慢车以上到慢车切换控制（油门杆从(2,85]推至[0,2]时）

**编号**：R\_6424

**功能**：油门杆从(2,85]推至[0,2]时，若N2r25 转速进入N2R25Dem±N2R25Design\*PD\_DN2R25Switch（可调）范围内，应用软件应进入转换逻辑二

**前置条件**：1

**输入**：PLA, N2r25, N2R25Dem, N2R25Design, PD\_DN2R25Switch

**输出**：software\_control\_mode

**公式**：

if ((PLA <= 2 && PLA >= 0) && (last(PLA) > 2 && last(PLA) <= 85) && (N2r25 <= (N2R25Dem + N2R25Design \* PD\_DN2R25Switch)) && (N2r25 >= (N2R25Dem - N2R25Design \* PD\_DN2R25Switch))) {

software\_control\_mode = High\_Press\_Control\_Mode;

}

**任务名称**：慢车以上到慢车切换控制（油门杆从[-33,-6)推至[-6,0)时）

**编号**：R\_10222

**功能**：油门杆从[-33,-6)推至[-6,0)时，若N2r25 转速进入N2R25Dem±N2R25Design\*PD\_DN2R25Switch（可调）范围内，应用软件应进入转换逻辑二

**前置条件**：1

**输入**：PLA, N2r25, N2R25Dem, N2R25Design, PD\_DN2R25Switch

**输出**：software\_control\_mode

**公式**：

if ((PLA < 0) && (PLA >= -6) && (last(PLA) >= -33) && (last(PLA) < -6) && (N2r25 <= (N2R25Dem + N2R25Design \* PD\_DN2R25Switch)) && (N2r25 >= (N2R25Dem - N2R25Design \* PD\_DN2R25Switch))) {

software\_control\_mode = High\_Press\_Control\_Mode;

}

**任务名称**：慢车以上到慢车切换控制（处于转换逻辑二，控制模式设置）

**编号**：R\_10220

**功能**：处于转换逻辑二过程中，应用软件应设置控制模式为高压控制模式。

**前置条件**：1

**输入**：PLA

**输出**：software\_control\_mode

**公式**：

if((PLA<0 && PLA >=-6 && last(PLA)>=-33 && last(PLA)<-6 || PLA<=2 && PLA >=0 && last(PLA)>2 && last(PLA)<=85)){

software\_control\_mode = High\_Press\_Control\_Mode;

}

**任务名称**：慢车以上到慢车切换控制（处于转换逻辑二，N1R最终控制目标设置）

**编号**：R\_10474

**功能**：PLA处于转换逻辑二过程中，应用软件应置N1R最终控制目标N1rDmd = N1RDem。

**前置条件**：1

**输入**：N1RDem, PLA

**输出**：N1rDmd

**公式**：

if((PLA<0 && PLA >=-6 && last(PLA)>=-33 && last(PLA)<-6 || PLA<=2 && PLA >=0 && last(PLA)>2 && last(PLA)<=85)){

N1rDmd = N1RDem;

}

**任务名称**：慢车以上到慢车切换控制（处于转换逻辑二，N2R25最终控制目标设置）

**编号**：R\_10221

**功能**：处于转换逻辑二过程中，应用软件应置N2R25最终控制目标N2r25Dmd = N2R25rd。

**前置条件**：1

**输入**：N2R25rd, PLA

**输出**：N2r25Dmd

**公式**：

if((PLA<0 && PLA >=-6 && last(PLA)>=-33 && last(PLA)<-6 || PLA<=2 && PLA >=0 && last(PLA)>2 && last(PLA)<=85)){

N2r25Dmd = N2R25rd;

}

**任务名称**：慢车以上到慢车切换控制（计算转换过程N2R25控制目标N2R25rd）

**编号**：R\_10224

**功能**：应用软件应按以下逻辑计算转换过程N2R25控制目标N2R25rd:

a) 根据正常高压换算转速控制计算控制期望N2R25\_1 = N2R25Dem；［C-8101］

b) 若N2r25<=N2R25\_1，应用软件应置本周期控制期望 N2R25rd＝N2r25＋N2R25Design\*PD\_DeltN2R25Co（可调）；［C-6449］

c) 若N2r25>N2R25\_1，应用软件应置本周期控制期望N2R25rd＝N2r25-N2R25Design\*PD\_DeltN2R25Co（可调）。［C-10225］

**前置条件**：1

**输入**：N2R25Dem, N2r25, N2R25Design, PD\_DeltN2R25Co

**输出**：N2R25rd

**公式**：

N2R25\_1 = N2R25Dem;

if(N2r25<=N2R25\_1){

N2R25rd=N2r25+N2R25Design\*PD\_DeltN2R25Co;

}else {

N2R25rd=N2r25-N2R25Design\*PD\_DeltN2R25Co;

}

任务名称：慢车以上到慢车切换控制（应用软件应退出转换逻辑二）

编号：R\_6450

功能：当满足以下任一条件时，应用软件应退出转换逻辑二：

a) 根据正常高压换算转速控制计算控制期望N2R25\_1后，abs（N2R25rd-N2R25\_1）<N2R25Design\*PD\_N2R25pIdleIn1 （可调）[C-10226]

b)进入转换逻辑二并维持PD\_IdleInTime(可调)；[C-10227]

c)N2r25p< PD\_IdleSwitchN2R25pMin（可调）[C-10228]

d)N1rp< PD\_IdleSwitchN1RpMin（可调）[C-10229]

e) abs（N2r25-N2R25Dem）>N2R25Design\*PD\_N2R25pIdleIn3(可调) [C-10230]

f)上一周期PLA<=2°且本周期PLA>2° [C-8099]

g)上一周期PLA>=-6°且本周期PLA<-6° [C-8457]

前置条件：1

输入：N2R25rd, N2R25Dem, N2R25Design, PD\_N2R25pIdleIn, PD\_IdleInTime, N2r25p, PD\_IdleSwitchN2R25pMin, N1rp, PD\_IdleSwitchN1RpMin, N2r25, N2R25Dem, PLA, software\_control\_mode

输出：software\_control\_mode

公式：

N2R25\_1 = N2R25Dem;

if ((abs(N2R25rd - N2R25\_1) < N2R25Design \* PD\_N2R25pIdleIn1) || (duration(PD\_IdleInTime , ms, logic\_transition\_mode == Logic2\_Control\_Mode)) || (N2r25p < PD\_IdleSwitchN2R25pMin) || (N1rp < PD\_IdleSwitchN1RpMin) || (abs(N2r25 - N2R25Dem) > N2R25Design \* PD\_N2R25pIdleIn3) || (PLA > 2 && last(PLA) <= 2) || (last(PLA) >= -6 && PLA < -6)) {

software\_control\_mode = last(software\_control\_mode);

}