目录

[一．形式化需求文档内容 2](#_Toc480965065)

[4.2.2信号输入管理需求 2](#_Toc480965066)

[4.2.3 故障诊断与处理需求 22](#_Toc480965067)

[4.2.4 控制规律管理需求 34](#_Toc480965068)

[4.2.5 机内自检测要求 38](#_Toc480965069)

[二．相关解释 44](#_Toc480965070)

[三．其他 45](#_Toc480965071)

## 一．形式化需求文档内容

### 4.2.2信号输入管理需求

**任务名称**：信号初始需求

**编号**：4.2.2.1

**功能**：

**前置条件:** after(60,ms,定时任务开始)

**输入**：无

**输出**：信号有效值，信号故障标志

**公式**：

信号有效值= 初始值；

本通道传感器故障 = 0；

双通道传感器故障 = 0；

信号故障状态 = 0；

**任务名称**：极值诊断

**编号**：4.2.2.2.1

**功能**：对于连续性信号变量的通用极值诊断处理

**前置条件:**

**输入**：XX.ExtreSetTime, XX.ExtreClrTime, XX.ExtreMin, XX.ExtreMax, XX.MeanData

**输出**：XX.ExtreInv, XX.ExtreFlt

**公式**：

if (XX.ExtreMin<=XX.MeanData<=XX.ExtreMax)

{

XX.ExtreInv = 0;

}

else

{

XX.ExtreInv = 1;

}

if(duration(XX.ExtreSetTime,s,XX.ExtreInv = 1))

{

XX.ExtreFlt = 1;

}

if(duration(XX.ExtreSetTime,s,XX.ExtreInv = 0))

{

XX.ExtreFlt = 0;

}

//T为该模块处理周期

**任务名称**：斜率诊断

**编号**：4.2.2.2.2

**功能**：对于连续性信号变量的通用极值诊断处理

**前置条件:** after(60+T,ms,定时任务开始)&& duration(检测到“FADEC交流发电机检测信号”无效开始时间,100,ms,N2p>16.0% )

**输入**：XX.SlopeSetTime, XX.SlopeLimit, XX.PreMeanData, XX.MeanData

**输出**：XX.SlopeInv, XX.SlopeFlt

**公式**：

if (Abs.(XX.MeanData, XX.PreMeanData)>XX.SlopeLimit))

{

XX.SlopeInv = 1;

}

else

{

XX.ExtreInv = 0;

}

if(duration(XX. SlopeSetTime, s, XX.SlopeInv = 1))

{

XX.SlopeInv = 1;

}

if(duration(XX. SlopeSetTime, s, XX. SlopeInv = 0))

{

XX.SlopeInv = 0;

}

**任务名称**：惯性滤波

**编号**：4.2.2.2.3

**功能**：根据滤波系数和当前周期输入计算出惯性滤波之后的有效值

**前置条件:** after(60,ms,定时任务开始)

**输入**：XX.InerFilterCo, XX.VaildData

**输出**：XX.FiltData

**公式**：

if(XX.FirstFlag == 0)

{

XX.FiltData = XX.VaildData

}

else

{

XX.FiltData = XX.InerFilterCo \* XX.VaildData + (1 – XX.InerFilterCo ) \*XX.LastFiltData

}

**任务名称**：超前滞后校正

**编号**：4.2.2.2.4

**功能**：由于传感器的滞后性， 对部分信号需进行超前滞后校正

**前置条件:** after(60,ms,定时任务开始)

**输入**：XX.PreLeadData, XX.FiltData, XX.PreFiltData, T, XX.T1, XX.T2

**输出**：XX.LeadData

**公式**：

if(XX.FirstFlag == 0)

{

XX. LeadData = XX. FiltData

}

else

{

XX.LeadData = [(XX.T2- T/2) \* XX.PreLeadData +

(XX.T1 + T/2)\*XX.FiltData – (XX.T1- T/2)\*XX.PreFiltData]/ (XX.T2+T/2)

}

if(abs(XX.LeadData, XX.PreLeadData)>XX.StepLimit)

{

if(XX.LeadData-XX.PreLeadData > 0)

{

XX.LeadData = XX.PreLeadData + XX.StepLimit

}

Else

{

XX.LeadData = XX.PreLeadData - XX.StepLimit

}

}

4.2.2.3 连续量输入信号处理

本文约定，传感器信号的诊断包括BIT诊断，极值诊断，斜率诊断。三类故障诊断并行处理。

**任务名称**：AD采样电路状态诊断

**编号**：4.2.2.3.1

**功能**：

**前置条件:**

**输入**：ADRefP.SqlData, ADRefN.SqlData, ADRefP.Max, ADRefP.Min, ADRefN.Min, ADRefN.Max, ADRef.SetTime , ADRef.ClrTime

**输出**：ADRefSts，ADRef.Flt

**公式**：

if(ADRefP.Min <= ADRefP.SqlData<= ADRefP.Max

&& ADRefN.Min <= ADRefN.SqlData <= ADRefN.Max)

{

ADRef.Sts = 0 ;

}

else

{

ADRef.Sts = 1 ;

}

if( duration(ADRef.SetTime,s,ADRef.Sts = 1 ))

{

ADRef.Flt = 1;

}

if( duration(ADRef.SetTime,s,ADRef.Sts = 0 ))

{

ADRef.Flt = 0;

}

应保证PLA处理周期为PLA\_ProcTime

**任务名称**：油门杆(PLA)信号处理

**编号**：4.2.2.3.2

**功能**：PLA为两支单余度信号，A、B通道各一支。

**前置条件:** ADRef.Flt = 0;

**输入**：ID\_AI\_PLAsin(正弦采集端口)，ID\_AI\_PLAcos, ID\_AI\_PLAsinpluscos, PLA.PtnrDiagData（对方通道信号诊断后有效值）, PLA.PtnrFlt（对方通道故障标志）, CCDLFlt（CCDL通讯故障标志）, PLA.DemarCurve（标定曲线）, PLA.BITSetTime（故障确认时间）, PLA.BITClrTime（清除BIT故障确认时间）, PLA.ExtreSetTime（极值故障确认时间）, PLA.ExtrClrTime, PLA.SlopeSetTime（斜率故障确认时间）, PLA.FirstBITTime（开始BIT诊断时间）, PLA.FirstSlopeTime, PLA.FirstExtreTime, PLA.BITMin, PLA.BITMax, PLA.ExtreMin, PLA.ExtreMax, PLA.SlopeLimit（斜率诊断阈值）, PLA.WarpLimit（双通道超差阈值）, PLA.WarpLimit（双通道超差阈值）, PLA.InitVal（初始值）, PLA.InerFilterCo（惯性滤波系数）, PLA.StepLimit, PLA.SquareUL（平方和监控上限）, PLA.SquareLL（下限）

**输出**：PLA.sin, PLA.cos, PLA.sum, PLA.Radian, PLA.MeanData（标定后角度）, PLA.DiagData, PLA.VoteData（表决后有效值）, PLA.ValidData（重构后有效值）, PLA.FiltData（滤波后有效值）, PLA.BITInv（失效标志）, PLA.BITFlt（故障标志）, PLA.ExtreInv, PLA.ExtreFlt, PLA.SlopeInv, PLA.SlopeFlt, PLA.SlopeFlt, PLA.SelfFlt（本通道传感器故障）, PLA.SensorFlt（双通道传感器故障）, PLA.SignalFlt（信号故障标志）, PLA.OTFlt（双通道超差告警标志）

**公式**：

PLA.sin = get[ID\_AI\_PLAsin];

PLA.cos = get [ID\_AI\_PLAcos];

PLA.sum = get [ID\_AI\_PLAsum];

after(PLA.FirstBITTime,ms,定时任务开始)

{

调用4.2.5.1[PLA];

}

PLA.radian = atan(PLA.sin/PLA.cos);

after(PLA.FirstExtreTime, ms, 定时任务开始)

{

调用4.2.2.2.1[PLA]；

}

after(PLA.FirstSlopeTime,定时任务开始)

{

调用4.2.2.2.2[PLA]；

}

if(PLA.BITFlt == 1 || PLA.ExtreFlt = 1 || PLA.SlopeFlt = 1)

{

PLA.SelfFlt = 1;

}

else

{

PLA.SelfFlt = 0；

}

if(PLA.SelfFlt = 1 || PLA.BITInv = 1 || PLA.ExtreInv = 1 || PLA.SlopeInv = 1 )

{

if(XX.FirstFlag == 0 )

{

PLA.DiagData = PLA.DiagData-PLA\_ProcTime;

}

else

{

PLA.DiagData =PLA.InitVal;

}

}

else

{

PLA.DiagData = PLA.MeanData ;

}

调用4.2.2.5.2.1[PLA];

PLA.SignalFlt = PLA.SensorFlt;

PLA.ValidData = PLA.VoteData;

调用4.2.2.2.3[PLA]；

if(PLA.FiltData - PLA.LastFiltData > PLA.StepLimit)

{

PLA.FiltData = PLA.LastFiltData + PLA.StepLimit;

}

else if(PLA.LastFiltData - PLA.FiltData > PLA.StepLimit)

{

PLA.FiltData = PLA.LastFiltData - PLA.StepLimit;

}

else

{

PLA.FiltData = PLA.FiltData;

}

if(强制慢车策略字 == 有效)

{

PLA.FiltData = 1°;

}

应保证T25处理周期为T25\_ProcTime

**任务名称**：高压压气机进口总温(T25)信号处理

**编号**：4.2.2.3.3

**功能**：T25为双余度信号，两余度分别进入A、B通道。

**前置条件**：ADRef.Flt = 0;

**输入**：ID\_AI\_V\_T25, T25.PtnrDiagData, T25.PtnrFlt, CCDLFlt, T25.DemarCurve, T25.BITSetTime, , T25.BITClrTime, T25.ExtreSetTime, T25.ExtrClrTime, T25.SlopeSetTime, T25.FirstBITTime, T25.FirstSlopeTime, T25.FirstExtreTime, T25.BITMin, T25.BITMax, T25.ExtreMin, T25.ExtreMax, T25.SlopeLimit, T25.WarpLimit, T25.WarpLimit, T25.InitVal, T25.InerFilterCo， T25.T1（超前滞后校正时间常数T1）, T25.T2, T25.StepLimit

**输出**：T25.SqlData, T25.MeanData, T25.DiagData, T25.VoteData, T25.FiltData, T25.LeadData, T25.ValidData, T25.BITInv, T25.BITFlt, T25.ExtreInv, T25.ExtreFlt, T25.SlopeInv, T25.SlopeFlt, T25.SelfFlt, T25.SensorFlt, T25.SignalFlt, T25.OTFlt

**公式**：

T25.SqlData =get[ID\_AI\_V\_T25];

after(T25.FirstBITTime, ms, 定时任务开始)

{

调用4.2.5.1[T25];

}

T25.MeanData = f(T25.SqlData, T25.DemarCurve); //暂定，查标定曲线

after(T25.FirstExtreTime, ms, 定时任务开始)

{

调用4.2.2.2.1[T25]；

}

after(T25.FirstSlopeTime, ms, 定时任务开始)

{

调用4.2.2.2.2[T25]；

}

if(T25.BITFlt == 1 || T25.ExtreFlt = 1 || T25.SlopeFlt = 1)

{

T25.SelfFlt = 1;

}

else

{

T25.SelfFlt = 0；

}

if(T25.SelfFlt = 1 || T25.BITInv = 1 || T25.ExtreInv = 1 || T25.SlopeInv = 1 )

{

if(T25.FirstFlag == 0)

{

T25.DiagData = T25.DiagData-T25\_ProcTime ;

}

else

{

T25.DiagData= T25.InitVal;

}

}

else

{

T25.DiagData = T25.MeanData;

}

调用4.2.2.5.2.1[T25];

T25.SignalFlt = T25.SensorFlt;

T25.ValidData = T25.VoteData;

调用4.2.2.2.3[T25]；

调用4.2.2.2.4[T25];

if(T25.LeadData - T25.LastLeadData > T25.StepLimit)

{

T25.LeadData = T25.LastLeadData + T25.StepLimit;

}

else if(T25.LastLeadData - T25.LeadData > T25.StepLimit)

{

T25.LeadData = T25.LastLeadData - T25.StepLimit;

}

else

{

T25.LeadData = T25.LeadData;

}

4.2.2.4 开关量输入信号处理

开关量输入信号baokuo用于控制和监视的SWI01-SWI18、SWI22-SWI24共21路开关量输入信号以及用于开关量输出回路断线、在线及过流检测的回采信号，模拟量输出过流检测回采信号。其中需对部分开关量输入信号进行处理，处理的目的是通过消抖处理获取稳定的信号值。

对开关量的输入信号必须遵循以下处理流程：信号采样→信号消抖→信号表决。

S04为三余度，S14（B），15（A），16（B）,18(A)为单余度，其余均为双余度

采样周期为SWI01\_ProcTime

**任务名称**：发动机位置1(SWI01)

**编号**：4.2.2.4.1

**功能**：SWI01为双余度信号，分别进入A、B通道

**前置条件**：

**输入**：SWI01.PtnrFlt(对方通道开入处理电路故障标志), SWI01.PtnrFirmSts（对方通道信号消抖后状态）, SWI01.SelfFlt（本通道开入处理电路故障标志）, SWI01.FirmTime（消抖确认时间）, SWI01.VoteLor（表决策略）, SWI01.Default（默认值）, SWI01.InitSts（初始值）

**输出**：SWI01.OrgSts（采集状态）, SWI01.FirmSts（消抖确认后状态）, SWI01.ValidSts（表决后状态）, SWI01.DiffFlt（不一致告警标志）

**公式**：

SWI01.InitSts=0

Duration（SWI01.FirmTime,s, SWI01.FirmSts ==SWI01.InitSts）;

if(SWI01.FirstFlag == 1)

{

调用4.2.5.8[SWI01];

SWI01.FirstFlag = 0

}

SWI01.OrgSts = get[待定];

If(duration(SWI01.FirmTime,s,SWI01.OrgSts==1))

{

SWI01.FirmSts = 1;

}

else if(duration(SWI01.FirmTime,s,SWI01.OrgSts==0))

{

SWI01.FirmSts = 0;

}

调用4.2.2.5.2.3[SWI01];

**任务名称**：开关量输出电路BIT状态信号

**编号**：4.2.2.4.22

**功能**：开关量输出电路断线BIT(开关量无输出)、在线BIT(kaiguanl 输出)以及过流BIT由硬件进行检测，软件读取硬件检测结果，再结合 开关量输出状态实现对开关量输出电路的检测。

**前置条件**：

**输入**：采集接口见某某文档

**输出**：SWOBIT.OFLSts（硬件断线检测结果）、SWOBIT.OLSts（在线检测结果）、SWOBIT.SCSts（过流检测结果）

**公式**：

SWOBIT.OFLSts = get[设备号]；

SWOBIT.OLSts = get[设备号]；

SWOBIT.SCSts = get[设备号]；

**任务名称**：模拟量输出回路过流BIT状态信号

**编号**：4.2.2.4.23

**功能**：模拟量输出电路过流BIT由硬件进行检测，软件读取硬件检测结果，根据硬件检测结果指示模拟量输出回路的过流状态。

**前置条件**：

**输入**：采集接口见某某文档

**输出**：AOBIT.SCSts（过流检测结果）

**公式**：

AOBIT.SCSts = get[设备号]；

4.2.2.5 信号余度管理

本节需求的目的是针对某一信号，对各信号源有效值以及故障信号进行综合评估，以获取该信号的有效值和故障状态。

4.2.2.5.1 单余度信号管理

**任务名称**：连续信号（余度管理）

**编号**：4.2.2.5.1.1

**功能**：单余度连续信号指仅单通道采集的信号，核心机单余度连续信号仅有A通道采集的Tfuel、Nq以及Wf

**前置条件**：

**输入**：CCDLFlt, XX.PtnrFlt, XX.PtnrDiagData, XX.SelfFlt, XX.DiagData

**输出**：XX.VoteDATA, XX.SensorFlt, XX.OTFFlt

**公式**：

XX.OTFlt = null;

if(通道标记==本通道)

{

if(XX.SensorFlt == 0)

{

XX.VoteData = XX.DiagData;

XX.SensorFlt = 0;

}

else

{

if(XX.FirstFlag)

{

XX.VoteData = XX.DiagData;

}

else

{

XX.VoteData = 当前周期有效值;

}

XX.SensorFlt = 1;

}

}

else

{

if(XX.CCDLFlt == 0 && XX.PtnrFlt == 0)

{

XX.VoteData = XXPtnr.DiagData;

XX.SensorFlt = 0;

}

else

{

if(上电第一周期)

{

XX.VoteData = 初始值;

}

else

{

XX.VoteData = 当前周期有效值;

}

XX.SensorFlt = 1;

}

}

**任务名称**：离散信号（余度管理）

**编号**：4.2.2.5.1.2

**功能**：核心机单余度离散信号有仅A通道采集的SWI15、SWI18以及仅B通道采集的SWI14、SWI16

**前置条件**：

**输入**：CCDLFlt, XX.PtnrFlt, XX.PtnrFirmSts, XX.SelfFlt, XX.FirmSts, XX.Default

**输出**：XX.ValidSts, XX.DiffFlt

**公式**：

XX.DiffFlt = 0;

if(本通道采集)

{

if(XX.SelfFlt == 0)

{

XX.ValidSts = XX.FirmSts;

}

else

{

XX.ValidSts = XX.Default

}

}

else

{

if(XX.CCDLFlt == 0 && XX.PtnrFlt == 0)

{

XX.ValidSts = XX.PtnrFirmSts;

}

else

{

XX.ValidSts = XX.Default;

}

}

4.2.2.5.2 双余度信号管理

信号的双余度是指本通道采集信号，以及通过通道间数据链路（CCDL）接收自对方通道的信号。

**任务名称**：连续信号(非伺服回路反馈)

**编号**：4.2.2.5.2.1

**功能**：无

**前置条件**：

**输入**：CCDLFlt，XX.PtnrFlt, XX.PtnrDiagData, XX.SelfFlt, XX.DiagData, XX.FirstFlag, XX.WarpLimit, XX.WarpLorH

**输出**：XX.VoteData, XX.SensorFlt, XX.OTFlt

**公式**：

if(CCDLFlt == 0 && XX.SelfFlt == 0 && XX.PtnrFlt == 0

&& abs(XX.DiagData-XX.PtnrDiagData)<= XX.WarpLimit)

{

XX.VoteData = (XX.DiagData + XX.PtnrDiagData)/2;

XX.SensorFlt = NOFAULT;

XX.OTFlt = NOFAULT;

}

if(CCDLFlt == 0 && XX.SelfFlt == 0 && XX.PtnrFlt == 0

&& abs(XX.DiagData-XX.PtnrDiagData)> XX.WarpLimit )

{

XX.VoteData = Max(XX.DiagData,XX.PtnrDiagData) or Min(XX.DiagData,XX.PtnrDiagData);

XX.SensorFault = NOFAULT;

XX.OTFlt = FAULT;

}

if(CCDLFlt == 0 && XX.SelfFlt == 0 && XX.PtnrFlt == 1)

{

XX.VoteData = XX.DiagData;

XX.SensorFlt = NOFAULT;

XX.OTFlt = NOFAULT;

}

if(CCDLFlt == 1 && XX.SelfFlt == 0 )

{

XX.VoteData = XX.DiagData;

XX.SensorFlt = NOFAULT;

XX.OTFlt = NOFAULT;

}

if(CCDLFlt == 0 && XX.SelfFlt == 1 && XX.PtnrFlt == 0)

{

XX.VoteData = XX.PtnrDiagData;

XX.SensorFlt = NOFAULT;

XX.OTFlt = NOFAULT;

}

if(CCDLFlt == 0 && XX.SelfFlt == 1 && XX.PtnrFlt == 1 && XX.FirstFlag == 0)

{

XX.VoteData = XX.PreVoteData;

XX.SensorFlt = FAULT;

XX.OTFlt = NOFAULT;

}

if(CCDLFlt == 0 && XX.SelfFlt == 1 && XX.PtnrFlt == 1 && XX.FirstFlag == 1)

{

XX.VoteData = XX.InitVal;

XX.SensorFlt = FAULT;

XX.OTFlt = NOFAULT;

}

if(CCDLFlt == 1 && XX.SelfFlt == 1 && XX.FirstFlag == 0)

{

XX.VoteData = XX.PreVoteData;

XX.SensorFlt = FAULT;

XX.OTFlt = NOFAULT;

}

if(CCDLFlt == 1 && XX.SelfFlt == 1 && XX.FirstFlag == 1)

{

XX.VoteData = XX.InitVal;

XX.SensorFlt = FAULT;

XX.OTFlt = NOFAULT;

}

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 条件 | | | | | 结果 | | |
| SelfFlt | CCDLPtnrFlt | 超差 | FirstFlag | 表决策略 | VoteData | SensorFlt | OTFlt |
| 1 | F | F | F | F | F | Average | F | F |
| 2 | F | F | F | F | T | Average | F | F |
| 3 | F | F | F | T | F | Average | F | F |
| 4 | F | F | F | T | T | Average | F | F |
| 5 | F | F | T | F | F | HigheData | F | T |
| 6 | F | F | T | F | T | LowData | F | T |
| 7 | F | F | T | T | F | HigheData | F | T |
| 8 | F | F | T | T | T | LowData | F | T |
| 9 | F | T | F | F | F | DiagData | F | F |
| 10 | F | T | F | F | T | DiagData | F | F |
| 11 | F | T | F | T | F | DiagData | F | F |
| 12 | F | T | F | T | T | DiagData | F | F |
| 13 | F | T | T | F | F | DiagData | F | F |
| 14 | F | T | T | F | T | DiagData | F | F |
| 15 | F | T | T | T | F | DiagData | F | F |
| 16 | F | T | T | T | T | DiagData | F | F |
| 17 | T | F | F | F | F | PtnrDiagData | F | F |
| 18 | T | F | F | F | T | PtnrDiagData | F | F |
| 19 | T | F | F | T | F | PtnrDiagData | F | F |
| 20 | T | F | F | T | T | PtnrDiagData | F | F |
| 21 | T | F | T | F | F | PtnrDiagData | F | F |
| 22 | T | F | T | F | T | PtnrDiagData | F | F |
| 23 | T | F | T | T | F | PtnrDiagData | F | F |
| 24 | T | F | T | T | T | PtnrDiagData | F | F |
| 25 | T | T | F | F | F | PreVoteData | T | F |
| 26 | T | T | F | F | T | PreVoteData | T | F |
| 27 | T | T | F | T | F | InitVal | T | F |
| 28 | T | T | F | T | T | InitVal | T | F |
| 29 | T | T | T | F | F | PreVoteData | T | F |
| 30 | T | T | T | F | T | PreVoteData | T | F |
| 31 | T | T | T | T | F | InitVal | T | F |
| 32 | T | T | T | T | T | InitVal | T | F |
| 说明：   1. 将CCDL和PtnrFlt两个变量合成为一个变量CCDLPtnrFlt，CCDLPtnrFlt = CCDL or PtnrFlt， 两者是或地关系，有一个故障就判定故障； 2. 由于每一种情况下都存在高低选的情况，所以输入变量定位5种，因此共存在32行； 3. 需求有9条，按前后顺序标为1到9，每一条需求所对应的情况如表格第二列所示； 4. F:若是SelfFlt则指本通道传感器无故障，若是CCDLPtnrFlt则指CCDL通讯故障&&对方通道传感器无故障，若是双通道超差则无超差，若是FirstFlag则不是上电第一周期，若是WarpLorH则是高选，若是SensorFlt则是传感器无故障，若是OTFlt则是无超差告警； 5. T:与F互斥 6. 输出定义：PreVoteData是指前一拍表决值， InitVal是指初始值，   Average = (Signal.DiagData + Signal.PtnrDiagData) / 2,  HighData = MAX (Signal.DiagData, Signal.PtnrDiagData),  LowData = MIN(Signal.DiagData, Signal.PtnrDiagData), | | | | | | | | |

**任务名称**：连续信号(伺服回路反馈)

**编号**：4.2.2.5.2.2

**功能**：无

**前置条件**：

**输入**：ChanSts(本通道主控状态), CCDLFlt，XX.PtnrFlt, XX.PtnrDiagData, XX.SelfFlt, XX.DiagData, XX.WarpLimit, XX.WarpLorH

**输出**：XX.VoteData, XX.SensorFlt, XX.OTFlt

**公式**：

if( ChanSts == 1 && XX.SelfFlt == 0)

{

XX.VoteData = XX.DiagData;

XX.SensorFlt = NOFAULT;

XX.OTFlt = NOFAULT;

}

if( ChanSts == 1 && XX.SelfFlt == 1 && CCDLFlt == 0 &&　XX.PtnrFlt == 0)

{

XX.VoteData = XX.PtnrDiagData;

XX.SensorFlt = NOFAULT;

XX.OTFlt = NOFAULT;

}

if( ChanSts == 1 && XX.SelfFlt == 1

&& (CCDLFlt == 1 ||　XX.PtnrFlt == 1)

&& XX.FirstFlag == 0

)

{

XX.VoteData = XX.PreVoteData;

XX.SensorFlt = FAULT;

XX.OTFlt = NOFAULT;

}

if( ChanSts == 1 && XX.SelfFlt == 1

&& (CCDLFlt == 1 ||　XX.PtnrFlt == 1)

&& XX.FirstFlag == 1)

{

XX.VoteData = XX.InitVal;

XX.SensorFlt = FAULT;

}

if( ChanSts == 0 && XX.SelfFlt == 1 && CCDLFlt == 0 &&　XX.PtnrFlt == 0)

{

XX.VoteData = XX.PtnrDiagData;

XX.SensorFlt = NOFAULT;

}

if( ChanSts == 0 && (CCDLFlt == 1 || XX.PtnrFlt ==1)&&　XX.SelfFlt == 0)

{

XX.VoteData = XX.DiagData;

XX.SensorFlt = NOFAULT;

}

if( ChanSts == 0 && (CCDLFlt == 1 || XX.PtnrFlt ==1)&&XX.SelfFlt == 1 && XX.FirstFlag == 0)

{

XX.VoteData = XX.PreDiagData;

XX.SensorFlt = FAULT;

}

if( ChanSts == 0 && (CCDLFlt == 1 || XX.PtnrFlt ==1)&&XX.SelfFlt == 1 && XX.FirstFlag == 1)

{

XX.VoteData = XX.InitVal;

XX.SensorFlt = FAULT;

}

if(XX.SelfFlt == 0

&& XX.PtnrFlt ==0

&& CCDLFlt == 0

&& abs(XX.DiagData-XX.PtnrDiagData)<= XX.WarpLimit　)

{

XX.OTFlt = NOFAULT;

}

if(XX.SelfFlt == 0

&& XX.PtnrFlt ==0

&& CCDLFlt == 0

&& abs(XX.DiagData-XX.PtnrDiagData)> XX.WarpLimit　)

{

XX.OTFlt = FAULT;

}

if(XX.SelfFlt==1 || XX.PtnrFlt==1 || CCDLFlt==1 )

{

XX.OTFlt = NOFAULT;

}

（附16行真值表，此处略）

**任务名称**：离散信号

**编号**：4.2.2.5.2.

**功能**：无

**前置条件**：

**输入**：CCDLFlt，XX.PtnrFlt, XX.PtnrFirmSts, XX.SelfFlt, XX.FirmSts, XX.Default

**输出**：XX.ValidSts, XX.DiffFlt

**公式**：

（处理流程与上述类似，为真值表条件判断，此处略）

4.2.2.5.3 三余度信号管理

**任务名称**：连续信号

**编号**：4.2.2.5.3.1.

**功能**：无

**前置条件**：

**输入**：CCDLFlt，XX.PtnrFlt, XX.PtnrDiagData, XX.SelfFlt, XX.DiagData, XX.PlaneFlt, XX.PlaneData, XX.WarpLimit, XX.WarpLorH

**输出**：XX.VoteData, XX.SensorFlt, XX.OTFlt

**公式**：

（处理流程与上述类似，为真值表条件判断，此处略）

**任务名称**：离散信号

**编号**：4.2.2.5.3.2

**功能**：无

**前置条件**：

**输入**：CCDLFlt，XX.PtnrFlt, XX.PtnrFirmSts, XX.SelfFlt, XX.FirmSts, XX.PlaneFlt, XX.PlaneSts, XX.Default

**输出**：XX.ValidSts, XX.DiffFlt

**公式**：

（处理流程与上述类似，为真值表条件判断，此处略）

4.2.2.5.4 四余度管理

**任务名称**：T44信号处理

**编号**：4.2.2.5.4.1.

**功能**：无

**前置条件**：

**输入**：CCDLFlt，XX.PtnrFlt, XX.PtnrDiagData, XX.SelfFlt, XX.DiagData, XX.PlaneFlt, XX.PlaneData, XX.WarpLimit, XX.WarpLorH

**输出**：XX.VoteData, XX.SensorFlt, XX.OTFlt

**公式**：

（处理流程与上述类似，为真值表条件判断，此处略）

**任务名称**：Tcj信号处理

**编号**：4.2.2.5.4.1.

**功能**：无

**前置条件**：

**输入**：CCDLFlt，XX.PtnrFlt, XX.PtnrDiagData, XX.SelfFlt, XX.DiagData, XX.PlaneFlt, XX.PlaneData, XX.WarpLimit, XX.WarpLorH

**输出**：XX.VoteData, XX.SensorFlt, XX.OTFlt

**公式**：

（处理流程与上述类似，为真值表条件判断，此处略）

4.2.2.6 信号重构

4.2.2.6.1 控制用信号重构

**任务名称**：N2信号重构

**编号**：4.2.2.6.1.1.

**功能**：无

**前置条件**：

**输入**：N2.SensorFlt, Nac.VoteData, Nac.SensorFlt, NacVsN2(Nac和N2测量位置传动比), N2\_1, Nac\_1

**输出**：N2.ValidData, N2.SignalFlt, Nac.SignalFlt

**公式**：

if(N2.SensorFlt == FAULT && Nac.SensorFlt == NOFAULT)

{

if(故障瞬时

&& (N2\_1 – NAC\_1/NacVsN2) < ModelLimit \* N2Design

&& 此条件仅在故障瞬时判定一次 )

{

N2.ValidData = Nac.VoteData;

N2.SignalFlt = NOFAULT;

}

}

if(N2.SensorFlt == FAULT && Nac.SensorFlt == FAULT)

{

应保持前值;//个人理解应为 N2.ValidData = N2\_1;

N2. SignalFlt = FAULT;

}

if( N2传感器故障瞬时

&& Nac.SensorFlt == NOFAULT

&& (N2\_1 – NAC\_1/NacVsN2)>= ModelLimit \* N2Design)

{

应保持前值;//个人理解应为 N2.ValidData = N2\_1;

N2. SignalFlt = FAULT;

}

if( N2传感器故障瞬时

&& (N2\_1 – NAC\_1/NacVsN2)>= ModelLimit \* N2Design)

{

Nac\_SensorFlt= FAULT;

}

**任务名称**：P3b信号重构

**编号**：4.2.2.6.1.2.

**功能**：无

**前置条件:**

**输入**：P3b.SensorFlt, Ps3.VoteData, Ps3.SensorFlt

**输出**：P3b.ValidData, P3b.SignalFlt

**公式**：

if( P3b.SensorFlt == Fault && Ps3.SensorFlt == NoFault)

{

P3b.ValidData = Ps3.VoteData;

P3b.SignalFlt = NOFAULT;

}

if( P3b.SensorFlt == Fault && Ps3.SensorFlt == Fault)

{

保持前值;

P3b.SignalFlt = FAULT;

}

4.2.2.6.2 监控用信号重构

**任务名称**：N2信号重构

**编号**：4.2.2.6.2.1

**功能**：无

**前置条件:**

**输入**：T3.SensorFlt, T3.VoteData, T25.SensorFlt, T25.VoteData, Ps3.VoteData, Ps3.SensorFlt, P25.VoteData, P25.SensorFlt

**输出**：N2.MntData

**公式**：

if(T3.SensorFlt == NOFAULT && T25.SensorFlt == NOFAULT)

{

查插值表“T3和T25信号合成N2信号插值表”；

}

if(T3.SensorFlt == FAULT

&& Ps3.SensorFlt == NOFAULT

&& T25.SensorFlt == NOFAULT

&& P25.SensorFlt == NOFAULT)

{

查插值表“Ps3和P25信号合成N2信号插值表”；

}

If（合成条件均不满足）

{

N2.MntData = -1 ;

}

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T3/T25=1.9 | T25(K) | 200 | 220 | 240 | 260 | 280 | 300 | 320 | 340 | 360 | 380 | 400 |
| N2 (rpm) | \*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* |
| T3/T25=2.3 | T25(K) | 200 | 220 | 240 | 260 | 280 | 300 | 320 | 340 | 360 | 380 | 400 |
| N2 (rpm) | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* |

表1 T3和T25信号合成N2信号插值表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ps3/P25=1 | T25(K) | 200 | 220 | 240 | 260 | 280 | 300 | 320 | 340 | 360 | 380 | 400 |
| N2 (rpm) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Ps3/P25=2 | T25(K) | 200 | 220 | 240 | 260 | 280 | 300 | 320 | 340 | 360 | 380 | 400 |
| N2 (rpm) | \*\*\*\* | \*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* |

表2 T3和T25信号合成N2信号插值表

**任务名称**：T25信号重构

**编号**：4.2.2.6.2.2

**功能**：无

**前置条件:**

**输入**：T3.SensorFlt, T3.VoteData, N2.FilterData, N2.SignalFlt

**输出**：T25.MntData

**公式**：

信号重构部分类似，包含判断条件及插值表

**任务名称**：T3信号重构

**编号**：4.2.2.6.2.3

**功能**：无

**前置条件:**

**输入**：N2.FilterData, N2.SignalFlt, T25.SensorFlt, T25.VoteData, Ps3.SensorFlt, Ps3.VoteData, P25.SensorFlt, P25.VoteData,

**输出**：T3.MntData

**公式**：

信号重构部分类似，包含判断条件及插值表

**任务名称**：P25信号重构

**编号**：4.2.2.6.2.4

**功能**：无

**前置条件:**

**输入**：N2.FilterData, N2.SignalFlt, T25.SensorFlt, T25.VoteData, Ps3.SensorFlt, Ps3.VoteData,

**输出**：P25.MntData

**公式**：

信号重构部分类似，包含判断条件及插值表

**任务名称**：Ps3信号重构

**编号**：4.2.2.6.2.5

**功能**：无

**输入**：N2.FilterData, N2.SignalFlt, T25.SensorFlt, T25.VoteData, T3.SensorFlt, T3.VoteData, P25.SensorFlt, P25.VoteData,

**输出**：Ps3.MntData

**公式**：

信号重构部分类似，包含判断条件及插值表

4.2.2.7 参数计算处理

**任务名称**：N2R25计算

**编号**：4.2.2.7.1

**功能**：无

**前置条件:**

**输入**：N2.FilterData, T25.LeadData

**输出**：N2r25

**公式**：

­­

**任务名称**：转速加速度计算

**编号**：4.2.2.7.2

**功能**：无

**前置条件:**

**输入**：N2.FilterData, N2dot.InerFilterCo

**输出**：N2r25

**公式**：

­­

// Ts为N2采样时间0.02s

调用4.2.2.2.4[N2dot(k)]进行滤波;

**任务名称**：其它参数计算

**编号**：4.2.2.7.3

**功能**：无

**前置条件:**

**输入**：

**输出**：N2Rel,N2RelT25,P0VsP0Std, P25EST, Ps3VsP0Std, SqrB1t25VsT10Calc, SqrB2t25VsT10Calc, SqrB3t25VsT10Calc

**公式**：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 |  | 信号名称 | 意义 | 换算公式 | 说明 |
| 1 |  | N2Rel | 高压转子相对转速 | N2Rel= N2/N2Design |  |
| 2 |  | N2RelT25 |  |  |  |
| 3 |  | P0VsP0Std |  |  |  |
| 4 |  | P25EST |  |  |  |
| 5 |  | Ps3VsP0Std |  |  |  |
| 6 |  | SqrB1t25VsT10Calc |  |  |  |
| 7 |  | SqrB2t25VsT10Calc |  |  |  |
| 8 |  | SqrB3t25VsT10Calc |  |  |  |

### 4.2.3 故障诊断与处理需求

故障诊断与处理模块主要完成控制系统故障诊断、通道健康状态判定、故障处理对策建立、告警等功能。

**模块名称**：需求故障诊断初始化

**编号**：4.2.3.1

**功能**：定时任务开始后，所有控制系统故障字无故障

**前置条件:**

**输入**：

**输出**：所有系统故障字

**公式**：

if(定时任务开始)

所有控制系统故障字=0

4.2.3.1.1 控制回路故障诊断

4.2.3.1.1.1 回路故障诊断算法概述

目前进行故障诊断的控制回路有：FMV控制回路、FSV控制回路、VSV控制回路、TBV4R控制回路、TBV4L控制回路。

**模块名称**：FMV控制回路故障

**编号**：4.2.3.1.1.2

**功能**：无

**前置条件:**

**输入**：N2p，Lfmv，LfmvDmd，发动机状态字，N2故障字，本通道健康评分，对方通道健康评分

**输出**：FMV回路故障字

**公式**：

If(not(本通道==备份状态&&发动机状态字==停车状态))

If(Duration(0.5，s，(N2p>FmvDiag\_N2Thresold(25%))&&(|Lfmv-LfmvDmd\*G(s)|>FmvDiag\_LvdtThresold(7°))&&FMV回路G(s)==1))

FMV回路故障字=故障

If(本通道==备用状态&&对方通道FMV回故障字==故障&&(本通道健康评分>对方通道健康评分))

FMV回路故障字=故障

If(FMV回路故障字==故障)

FMV可清除=false

**模块名称**：VSV控制回路故障

**编号**：4.2.3.1.1.3

**功能**：无

**前置条件:**

**输入**：N2p，Lvsv，LvsvDmd，发动机状态字，N2故障字，本通道健康评分，对方通道健康评分

**输出**：VSV回路故障字

**公式**：

If(not(本通道==备份状态&&发动机状态字==停车状态))

If(Duration(0.5，s，(N2p>VsvDiag\_N2Thresold(25%))&&(|Lvsv-LvsvDmd\*G(s)|>VsvDiag\_LvdtThresold(6.5mm))&&VSV回路G(s)==1))

VSV回路故障字=故障

If(本通道==备用状态&&对方通道VSV回路故障字==故障&&(本通道健康评分>对方通道健康评分))

VSV回路故障字=故障

If(VSV回路故障字==故障)

VSV可清除=false

**模块名称**：FSV控制回路故障

**编号**：4.2.3.1.1.4

**功能**：无

**前置条件:**

**输入**：N2p，Lfsv，LfsvDmd，发动机状态字，N2故障字，本通道健康评分，对方通道健康评分

**输出**：FSV回路故障字

**公式**：

If(not(本通道==备份状态&&发动机状态字==停车状态))

If(Duration(0.5，s，(N2p>FsvDiag\_N2Thresold(25%))&&(|Lfsv-LfsvDmd\*G(s)|>FsvDiag\_LvdtThresold(3mm))&&FSV回路G(s)==1))

FSV回路故障字=故障

If(本通道==备用状态&&对方通道FSV回路故障字==故障&&(本通道健康评分>对方通道健康评分))

FSV回路故障字=故障

If(FSV回路故障字==故障)

FSV可清除=false

**模块名称**：FST控制回路故障

**编号**：4.2.3.1.1.5

**功能**：无

**前置条件:**

**输入**：N2p，Lfst，LfstDmd，发动机状态字，N2故障字，本通道健康评分，对方通道健康评分

**输出**：FST回路故障字

**公式**：

If(not(本通道==备份状态&&发动机状态字==停车状态))

If(Duration(0.5，s，(N2p>FstDiag\_N2Thresold(25%))&&(|Lfst-LfstDmd\*G(s)|>FstDiag\_LvdtThresold(待定))&&FST回路G(s)==1))

FST回路故障字=故障

If(本通道==备用状态&&对方通道FST回路故障字==故障&&(本通道健康评分>对方通道健康评分))

FST回路故障字=故障

If(FST回路故障字==故障)

FST可清除=false

**模块名称**TBV4R控制回路故障

**编号**：4.2.3.1.1.6

**功能**：无

**前置条件:**

**输入**：N2p，Ltbv4R，Ltbv4RDmd，发动机状态字，N2故障字，本通道健康评分，对方通道健康评分

**输出**：TBV4R回路故障字

**公式**：

If(not(本通道==备份状态&&发动机状态字==停车状态))

If(Duration(0.5，s，(N2p>Tbv4RDiag\_N2Thresold(25%))&&(|Ltbv4R-Ltbv4RDmd\*G(s)|>Tbv4RDiag\_LvdtThresold(7.1mm))&& TBV4R回路G(s)==1))

TBV4R回路故障字=故障

If(本通道==备用状态&&对方通道TBV4R回路故障字==故障&&(本通道健康评分>对方通道健康评分))

TBV4R回路故障字=故障

If(TBV4R回路故障字==故障)

TBV4R可清除=false

**模块名称**TBV4L控制回路故障

**编号**：4.2.3.1.1.7

**功能**：无

**前置条件:**

**输入**：N2p，Ltbv4L，Ltbv4LDmd，发动机状态字，N2故障字，本通道健康评分，对方通道健康评分

**输出**：TBV4L回路故障字

**公式**：

If(not(本通道==备份状态&&发动机状态字==停车状态))

If(Duration(0.5，s，(N2p>Tbv4LDiag\_N2Thresold(25%))&&(|Ltbv4L-Ltbv4LDmd\*G(s)|>Tbv4LDiag\_LvdtThresold(7.1mm))&& TBV4L回路G(s)==1))

TBV4L回路故障字=故障

If(本通道==备用状态&&对方通道TBV4L回路故障字==故障&&(本通道健康评分>对方通道健康评分))

TBV4L回路故障字=故障

If(TBV4R回路故障字==故障)

TBV4L可清除=false

**模块名称**：高温涡轮出口总温（T44）超温故障

**编号**：4.2.3.1.2.1

**功能**：无

**前置条件:**

**输入**：T44高压涡轮出口总温，超温故障注入开关，发动机状态字

**输出**：高压涡轮出口总温超温故障字

**公式**：

If(ES== ES\_GI&&ES==ES\_ABOVEIDLE)

If(ES== ES\_GI&&ES==ES\_ABOVEIDLE&&T44>=EgtExcdDiag\_ExcdValue)

高压涡轮出口总温超温故障字=故障

If(超温故障注入开关==连通)

高压涡轮出口总温超温故障字=故障

If(T44<EgtExcdDiag\_ClearFaultValue&&超温故障注入开关==断开)

高压涡轮出口总温超温故障字=故障

**模块名称**：高压压气机出口静压（Ps3）超压故障

**编号**：4.2.3.1.2.2

**功能**：无

**前置条件:**

**输入**：Ps3，发动机状态字

**输出**：高压压气机出口静压超压故障字

**公式**：

If(ES== ES\_GI&&ES==ES\_ABOVEIDLE)

If(高压压气机出口静压Ps3>=Ps3ExcdDiag\_ExcdValue(待定))

高压压气机出口静压超压故障字=故障

If(Ps3<Ps3ExcdDiag\_ClearFaultValue(待定))

高压涡轮出口总温超温故障字=无故障

**模块名称**：发动机超转故障

**编号**：4.2.3.1.2.3

**功能**：无

**前置条件:**

**输入**：N2，超转故障注入开关，发动机状态字

**输出**：发动机超转故障字

**公式**：

If(ES== ES\_GI&&ES==ES\_ABOVEIDLE)

If(N2>=N2ExcdDiag\_ExcdValue(19337rpm))

发动机超转故障字=故障

If(超转故障注入开关==接通)

发动机超转故障字=故障

If(N2< N2ExcdDiag\_ClearFaultValue(待定)&& 超转故障注入开关==断开)

发动机超转故障字=无故障

4.2.3.1.3 发动机启动故障诊断

**模块名称**： T44启动超温故障

**编号**：4.2.3.1.3.1

**功能**：无

**前置条件:**

**输入**：T44，发动机状态字

**输出**：T44启动超温故障字

**公式**：

If(ES== ES\_GRDSTART &&Duration(0.5,s,T44>TExcdEgtThresold))

T44启动超温故障字=故障

If(ES== ES\_ STAND)

T44启动超温故障字=无故障

**模块名称**： 启动失速故障

**编号**：4.2.3.1.3.2

**功能**：无

**前置条件:**

**输入**：T44，Ps3，N2dot，发动机状态字

**输出**：启动失速故障字

**公式**：

If(ES== ES\_GRDSTART && Duration(0.5,s,(T44>StallEgtThresold(待定)) && (N2dot<N2dotEgtThresold(待定))))

启动失速故障字=故障

If(ES== ES\_GRDSTART && Duration(15,s,(Ps3<T44StallEgtThresold(待定)) && (N2dot<N2dotPs3Thresold(待定))))

启动失速故障字=故障

If(ES== ES\_ STAND)

启动失速故障字=无故障

**模块名称**： 启动悬挂故障

**编号**：4.2.3.1.3.3

**功能**：无

**前置条件:**

**输入**：N2dot，发动机状态字，点火成功标志

**输出**：启动悬挂故障字

**公式**：

If(ES== ES\_GRDSTART && 点火成功标志==true && Duration(5,s, N2dot<N2dotEgtThresold(0.1rpm/s)))

启动悬挂故障字=故障

If(ES== ES\_ STAND)

启动悬挂故障字=无故障

**模块名称**：点火失败故障

**编号**：4.2.3.1.3.4

**功能**：（注：点火成功判据见8706）

**输入**：开始供油标志，发动机状态字，点火成功标志

**输出**：点火失败故障字

**公式**：

If(ES== ES\_GRDSTART &&(not Duration(15,s, 开始供油标志==true))&&点火成功标志=无效)

点火失败故障字=故障

If (ES== ES\_ STAND)

点火失败故障字=无故障

**模块名称**：滑油压力低故障

**编号**：4.2.3.1.3.5

**功能**：

**前置条件:**

**输入**：N2p，Poil，发动机状态字

**输出**：滑油压力低故障字

**公式**：

If(ES== ES\_GRDSTART &&Duration(100,ms, N2p>OilN2Thresold(待定)&&Poil< OilPoilThresold(待定)))

滑油压力低故障字=故障

If (ES== ES\_ STAND)

滑油压力低故障字=无故障

**模块名称**：起动机未正常脱开故障

**编号**：4.2.3.1.3.6

**功能**：

**输入**：N2p，发动机状态字，Nq，N2

**输出**：起动机未正常脱开故障字

**公式**：

If ((Duration (60, s, ES==ES\_GRDSTART) || (N2p>IsolateN2Thresold)) && Nq/N2>IsolateNqVsN2(2.4))

滑油压力低故障字=故障

If (ES== ES\_ STAND)

滑油压力低故障字=无故障

**模块名称**：主燃油滤堵塞故障

**编号**：4.2.3.1.3.7

**功能**：

**前置条件:**

**输入**：主燃油滤堵塞信号

**输出**：主燃油滤堵塞故障字

**公式**：

If (主燃油滤堵塞信号==1)

主燃油滤堵塞故障字=故障

if(主燃油滤堵塞信号==0)

主燃油滤堵塞故障字=无故障

**编号**：4.2.3.1.3.8

**功能**：

**前置条件:**

**输入**：喘振告警开关信号，喘振故障注入开关

**输出**：喘振故障字

**公式**：

If (喘振告警开关信号==1)

喘振故障字=故障

if(喘振故障注入开关==1)

喘振故障字=故障

if(喘振告警开关信号==0 && 喘振故障注入开关==0)

喘振故障字=无故障

4.2.3.1.4 电源故障诊断

**编号**：4.2.3.1.4.1

**功能**：

**前置条件:**

**输入**：N2，FADEC交流发电机供电检测信号，FADEC交流发电机过压检测信号，FADEC交流发电机欠压检测信号，28VDC供电检测，28VDC电源切断到位检测信号，过压检测信号，欠压检测信号，交流电接入指令信号(SWO10)，直流电切断指令信号(SWO11)

**输出**：“交流电接入失败”故障字，“交流电意外接入”故障字，“交流电过压”故障字，“交流电欠压”故障字，“直流电供电无效”故障字，“直流电意外切断”故障字，“直流电切断失败”故障字，“EEC供电过压”故障字，“EEC供电欠压”故障字

**公式**：

供电电源故障可清除=false

If (Duration(100,ms,交流电接入指令信号=null && FADEC交流发电机供电检测信号==有效))

“交流电意外接入”故障字=故障

if (Duration(100,ms,交流电接入指令信号!=null&& FADEC交流发电机供电检测信号==无效))

“交流电接入失败”故障字=故障

if(Duration(启动时间，停车时间-起动时间，“交流电接入失败”故障次数>3))

故障状态=锁定

if(Duration(100,ms,FADEC交流发电机过压检测信号==无效))

“交流电过压”故障字=故障

if(Duration(100,ms,FADEC交流发电机欠压检测信号==无效))

“交流电欠压”故障字=故障

if(Duration(100,ms,28VDC供电检测==无效))

“直流电供电无效”故障字=故障

If (Duration(100,ms, 直流电切断指令信号=null && 28VDC电源切断到位检测信号==有效))

“直流电意外切断”故障字=故障

If (Duration(100,ms, 直流电切断指令信号!=null && 28VDC电源切断到位检测信号==无效))

“直流电切断失败”故障字=故障

if(Duration(100,ms,过压检测==无效))

“EEC供电过压”故障字=故障

if(Duration(100,ms,欠压检测==无效))

“EEC供电欠压”故障字=故障

**编号**：4.2.3.2

**功能**：通道健康状态判断

**前置条件:**

**输入**：发动机状态，CPU故障，内部RAM故障，外部RAM故障，参数Flash故障，代码Flash故障，AD故障，看门狗故障，Ifmv故障，FMV故障，Ifsv故障，FSV回路故障，Ivsv故障，VSV回路故障，主燃油分级活门控制电磁阀故障，Itbv4L故障，TBV4L回路故障，Itbv4R故障，TBV4R回路故障，点火器1故障，点火器2故障，启动空气阀电磁铁驱动信号故障

**输出**：通道健康状态权值

**公式**：

If (ES==ES\_ ORIGIN)//初始状态特殊情况

{

本通道健康评分=0

通道健康状态权值1=0

通道健康状态权值2=0

通道健康状态权值3=0

通道健康状态权值4=0

通道健康状态权值5=0

通道健康状态权值6=0

通道健康状态权值7=0

通道健康状态权值8=0

通道健康状态权值9=0

通道健康状态权值10=0

通道健康状态权值11=0

}

if((CPU故障==故障||内部RAM故障==故障||外部RAM故障==故障||参数Flash故障==故障||代码Flash故障==故障||AD故障==故障||看门狗故障==故障)&&(通道标志==本通道)

通道健康状态权值1=512

if((Ifmv故障==输出故障||FMV故障==故障)&&(通道标志==本通道))

通道健康状态权值2=256

if((Ifsv故障==输出故障||FSV回路故障==故障)&&(通道标志==本通道))

通道健康状态权值3=128

if((Ivsv故障==输出故障||VSV回路故障==故障)&&(通道标志==本通道))

通道健康状态权值4=64

if(主燃油分级活门控制电磁阀故障==故障)&&通道标志==本通道)

通道健康状态权值5=32

if((Itbv4L故障==输出故障||TBV4L回路故障==故障)&&(通道标志==本通道))

通道健康状态权值6=8

if((Itbv4R故障==输出故障||TBV4R回路故障==故障)&&(通道标志==本通道))

通道健康状态权值7=8

if((ES==ES\_ STAND||ES==ES\_ COOLRUN||ES=ES\_ FUELSEAL)&& 点火器1故障==故障&&点火器2故障==故障&&通道标志==本通道)

通道健康状态权值8=2

if((ES==ES\_ STAND||ES==ES\_ COOLRUN||ES=ES\_ FUELSEAL)&& 启动空气阀电磁铁驱动信号故障==故障&&通道标志==本通道)

通道健康状态权值9=1

if((ES==ES\_ GI ||ES==ES\_ ABOVEIDLE ||ES=ES\_ STARTABORT||ES=ES\_ NORMALSTOP ||ES=ES\_ URGENSTOP)&& 点火器1故障==故障&&点火器2故障==故障&&通道标志==本通道)

通道健康状态权值10=0

if((ES==ES\_ GI ||ES==ES\_ ABOVEIDLE ||ES=ES\_ STARTABORT||ES=ES\_ NORMALSTOP ||ES=ES\_ URGENSTOP)&& 启动空气阀电磁铁驱动信号故障==故障&&通道标志==本通道)

通道健康状态权值11=0

（周期任务：20ms）

If(ES!=ES\_ ORIGIN)

{

If(通道健康状态权值6=通道健康状态权值7 && 通道健康状态权值6=8)

本通道健康评分=通道健康状态权值1+通道健康状态权值2+通道健康状态权值3+通道健康状态权值4+通道健康状态权值5+通道健康状态权值6+通道健康状态权值7+通道健康状态权值9+通道健康状态权值10+通道健康状态权值11

Else

本通道健康评分=通道健康状态权值1+通道健康状态权值2+通道健康状态权值3+通道健康状态权值4+通道健康状态权值5+通道健康状态权值6+通道健康状态权值7+通道健康状态权值8+通道健康状态权值9+通道健康状态权值10+通道健康状态权值11

}

**编号**：4.2.3.3

**功能**：故障处理对策

**前置条件:**

**输入**：

**输出**：停车策略字，禁止启动策略字，强制慢车策略字，发动机降级使用策略字，取消加减速策略字，VSV安全策略字，TBV4L安全策略字，TBV4R安全策略字，FSV安全策略字，FST安全策略字，T25重构T3策略字，Ps3重构P3b策略字，采用故障前策略字，Ivsv零电流输出策略字，Itbv4l零电流输出策略字，Itbv4r零电流输出策略字，Ifsv零电流输出策略字，Ifmv零电流输出策略字，Ifst零电流输出策略字，通道切换策略字，取消T3限制策略字，取消T44限制策略字，取消最大N2限制策略字，取消N2r25限制策略字，冷运转中断策略字

**公式**：

停车策略字=0

禁止启动策略字=0

强制慢车策略字=0

发动机降级使用策略字=0

取消加减速策略字=0

VSV安全策略字=0

TBV4L安全策略字=0

TBV4R安全策略字=0

FSV安全策略字=0

FST安全策略字=0

T25重构T3策略字=0

Ps3重构P3b策略字=0

采用故障前策略字=0

Ivsv零电流输出策略字=0

Itbv4l零电流输出策略字=0

Itbv4r零电流输出策略字=0

Ifsv零电流输出策略字=0

Ifmv零电流输出策略字=0

Ifst零电流输出策略字=0

通道切换策略字=0

取消T3限制策略字=0

取消T44限制策略字=0

取消最大N2限制策略字=0

取消N2r25限制策略字=0

冷运转中断策略字=0

**编号**：4.2.3.3.1

**功能**：停车策略

**前置条件:**

**输入**：发动机状态字，N2信号故障字，T44超温故障字，火警信号，Lfmv传感器故障字，Ifmv输出故障字，FMV回路故障字，喘振故障字，Ps3传感器故障字，超转故障字，Lfsv传感器故障字，Ifsv输出故障字，FSV回路故障字，Lfst传感器故障字，Ifst输出故障字，FST回路故障字

**输出**：停车策略对策字

**公式**：

If (ES== ES\_GI&&ES==ES\_ABOVEIDLE&& N2信号故障字==故障)

停车策略对策字=1

If (ES== ES\_GI && T44超温故障字==故障)

停车策略对策字=1

If (ES== ES\_GI&&ES==ES\_ABOVEIDLE&& 火警信号==故障)

停车策略对策字=1

If (ES== ES\_GI&&ES==ES\_ABOVEIDLE&&双通道Lfmv传感器故障字==故障)

停车策略对策字=1

If (ES== ES\_GI&&ES==ES\_ABOVEIDLE&& 双通道Ifmv传感器故障字==故障)

停车策略对策字=1

If (ES== ES\_GI&&ES==ES\_ABOVEIDLE&&双通道FMV回路故障字==故障)

停车策略对策字=1

If (ES== ES\_GI && 喘振故障字==故障)

停车策略对策字=1

If (ES== ES\_GI&&ES==ES\_ABOVEIDLE&& 单通道Ifmv输出故障字==故障)

停车策略对策字=1

If (ES== ES\_GI&&ES==ES\_ABOVEIDLE&&双通道Ps3传感器故障字==故障)

停车策略对策字=1

If (ES== ES\_GI &&超转故障字==故障)

停车策略对策字=1

If (ES== ES\_GI&&ES==ES\_ABOVEIDLE&&双通道Lfsv传感器故障字==故障)

停车策略对策字=1

If (ES== ES\_GI&&ES==ES\_ABOVEIDLE&&双通道Ifsv输出故障字==故障)

停车策略对策字=1

If (ES== ES\_GI&&ES==ES\_ABOVEIDLE&&双通道FSV回路故障字==故障)

停车策略对策字=1

If (ES== ES\_GI&&ES==ES\_ABOVEIDLE&& 单通道Ifsv输出故障字==故障&&另一通道FSV回路故障字==故障)

停车策略对策字=1

If (ES== ES\_GI&&ES==ES\_ABOVEIDLE&&双通道Lfst传感器故障字==故障)

停车策略对策字=1

If (ES== ES\_GI&&ES==ES\_ABOVEIDLE&&双通道Ifst输出故障字==故障)

停车策略对策字=1

If (ES== ES\_GI&&ES==ES\_ABOVEIDLE&&双通道FST回路故障字==故障)

停车策略对策字=1

If (ES== ES\_GI&&ES==ES\_ABOVEIDLE&& 单通道Ifst输出故障字==故障&&另一通道FST回路故障字==故障)

停车策略对策字=1

If (ES== ES\_ STAND)

停车策略对策字=0

**编号**：4.2.3.3.15

**功能**：Itbv4L零电流输出策略字

**前置条件:**

**输入**：TBV4L回路故障，ltbv4L输出故障，ltbv4L传感器故障

**输出**：Itbv4L零电流输出策略字

**公式**：

If (ltbv4L输出故障==故障)

Itbv4L零电流输出策略字=启用

If (Ltbv4L输出故障==故障)

Itbv4L零电流输出策略字=启用

If (TBV4L回路故障==故障)

Itbv4L零电流输出策略字=启用

If (Selfltbv4L输出故障==故障&&PtnrTBV4L回路故障==故障)

Itbv4L零电流输出策略字=启用

Itbv4L零电流输出策略字可清除标志=false

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 变量名 | 变量符号 | 类型 | 类型解释 | 变量注释 |
| TBV4L回路故障 |  | Bool | 1：故障，0：无故障 |  |
| ltbv4L输出故障 | ltbv4LOutputFlt | Bool | 1：故障，0：无故障 |  |
| ltbv4L传感器故障 | ltbv4LSensorFlt | Bool | 1：故障，0：无故障 |  |
| Selfltbv4L输出故障 | Selfltbv4LOutputFlt | Bool | 1：故障，0：无故障 | 本通道ltbv4L输出故障 |
| PtnrTBV4L回路故障 |  | Bool | 1：故障，0：无故障 | 另一通道TBV4L回路故障 |
| Itbv4L零电流输出策略字可清除标志 |  | Bool | 1：可清除，0：不可清除 |  |
| Itbv4L零电流输出策略字 |  | Bool | 1：启用，0：不启用 |  |

//主控信息：主控状态，备份状态

周期任务：每20ms比较一次本通道和对方通道健康评分

**编号：**4.2.3.3.20

**功能**：通道切换策略

**前置条件:**

**输入**：本通道主控次数，对方通道主控次数，CCDL故障字，N2P，本通道健康状态权值，对方通道健康状态权值，本通道发动机状态字，对方通道发动机状态字，本通道主控信息，对方通道主控信息，A通道主控信息

**输出**：通道切换策略字

**公式**：

//应记录通道主控次数，并且存放于非易失性存储器中

If(ES==ES\_STAND && N2转速>=80% && N2.FirstFlag==1)

主控通道的主控次数=主控通道的主控次数+1

FirstFlag=0

if(本通道主控信息==主控状态&&Duration(40,ms, 对方通道健康评分<本通道健康评分))

通道切换策略字=启用

if(ES==ES\_STAND && Last\_ES!= ES\_STAND&&本通道健康状态权值==对方通道健康状态权值&& A通道主控信息==主控状态)

通道切换策略字=启用

if(ES==ES\_ GRDSTART &&本通道主控信息==主控状态&&本通道健康状态权值==对方通道健康状态权值&&N2P>=30%&&启动建立标志位==1)

通道切换策略字=启用，启动建立标志位=0

if(ES==ES\_ GRDSTART &&本通道主控信息==备份状态&&本通道健康状态权值==对方通道健康状态权值&&N2P>=40%&&启动建立标志位==1)

通道切换策略字=启用，启动建立标志位=0

if(ES==ES\_ FUELSEAL &&本通道主控信息==主控状态&&本通道健康状态权值==对方通道健康状态权值&&N2P>=18%&&油封建立标志位==1)

通道切换策略字=启用，油封建立标志位=0

if(ES==ES\_ FUELSEAL &&本通道主控信息==备份状态&&本通道健康状态权值==对方通道健康状态权值&&N2P>=22%&&油封建立标志位==1)

通道切换策略字=启用，油封建立标志位=0

if(ES==ES\_ COOLRUN &&主控通道健康状态未发生变化)

通道切换策略字=禁用

/\*

检测到单通道Ifmv输出故障，另一通道FMV回路故障，应给出启动终止策略

if(（selfIfmv输出故障==故障&&PtnrFMV回路故障）||（PtnrIfmv输出故障==故障&&selfFMV回路故障）)

启动终止策略==1

\*/

**编号**：4.2.3.4.1.10

**功能**：高压关断活门位置告警

**前置条件:**

**输入**：发动机状态字，发动机关断活门位置开关（SWI15），N2r25

**输出**：高压关断活门位置告警字

**公式**：

if(!During(100,ms, SWI15==连接)) // SWI15是否状态是连接和断开

高压关断活门位置告警字=1

if(ES==ES\_NORMALSTOP&&During(1.2,s,SWI15==断开))

高压关断活门位置告警字=1

if(高压关断活门位置告警字==1)

高压关断活门位置告警字可清除标志=false

**编号**：4.2.3.4.1.11

**功能**：停车电磁阀告警

**前置条件:**

**输入**：发动机状态字，N2r25

**输出**：停车电磁阀告警字

**公式**：

/\*进入正常停车状态时刻记录当前时刻N2r25,1s后记录相应时刻N2r25转速，若两者之差小于18\*/

if(abs(N2r251- N2r250)<1&&ES=正常停车)// N2r250当前时刻N2r25

停车电磁阀告警字=1

if(停车电磁阀告警字==1)

停车电磁阀告警字可清除标志=false

### 4.2.4 控制规律管理需求

**编号**：4.2.4.5.1

**功能**：燃油控制

**前置条件:**

**输入**：发动机状态字，N2p，N2r25p，Ps3，P0

**输出**：WfDem，开始供油标志

**公式**：

if(ES=ES\_ GRDSTART)

{

if(N2>=N2StartWfP)

{

WfDem=max(fst\_wf,fig);

开始供油标志=1;

}

if(Duration(3,s, N2>N2StartWfTimerP))

{

WfDem=max(fst\_wf,fig);

开始供油标志=1;

}

if(!Duration(3,s, N2>N2StartWfTimerP)&& N2<N2StartWfP)

{

WfDem=WfGrdStartLow(kg/h)

}

}

fst=find[表8690，N2r25p，P25]；

fst\_wf=fst\*(Ps3/P0Std)\*(T25/T10)\*0.5；

if(ES==ES\_ STAND)

开始供油标志=0;

表8690

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P25\N2r25p | 0.22 | 0.26 | 0.27 | 0.30 | 0.35 | 0.40 | 0.43 | 0.46 | 0.50 |
| 101.325kpa | 0.0000 |  |  |  |  |  |  |  |  |
| 57.182kpa | 0.0000 |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 101.325 |  |  |  |  |  |  |  |  |  |
| 57.182 |  |  |  |  |  |  |  |  |  |

4.2.4.5.1.2 慢车及慢车以上状态燃油控制

**编号**：4.2.4.5.1.2.1

**功能**：N2回路燃油控制

**前置条件:**

**输入**：N2r25Dem，N2，T25，T10

**输出**：WfN2

**公式**：

N2MinLmt=11716rpm

N2r25MinLmt=12551rpm

N2r25MaxLmt=16631rpm

N2MaxLmt=18958rpm

/\*N2Dem表示N2给定值，N2r25Dem表示N2r25控制目标值，N2MAXLmt表示N2最大限制，N2MinLmt表示N2最小限制，N2r25MaxLmt表示N2R25最大限制，N2r25MinLmt表示N2R25最小限制\*/

WfN2=(Kp+Ki\*Ts/2)\*(N2Cmd-N2)-(Kp-Ki\*Ts/2)\*StateN2Pre+WfPre

StateN2=(N2Cmd-N2)+(WfPre-WfN2Pre)/(Kp+Ki\*Ts/2)

/\*StateN2Pre表示StateN2上一周期值，WfPre, 表示Wf上一周期值，WfN2Pre表示WfN2上一周期值\*/

Kp=K1\*Kp0()

Ki=K2\*Ki0()

表9432 Kp()随()转速变化的规律表

表9436 Kp修正系数K1随T25和P25变化规律表

表9434 Ki修正系数K2随T25和P25变化规律表

**编号**：4.2.4.5.1.2.2

**功能**：Ps3最小限制回路燃油计算

**前置条件:**

**输入**：ps3

**输出**：WfPs3MinLmt

**公式**：

Ts=0.02s

if(初始标记位==1)

{

StatePs3MinPre=PsMinLmt-Ps3

WfPs3MinPre=WPre

}

WfPs3Min=(Kp+Ki\*Ts/2)\*(Ps3Min-Ps3)-(Kp-Ki\*Ts/2)\*StatePsMinPre+WfPre

StatePs3Min=(Ps3MinLmt-Ps3)+(WfPre-WfPs3MinPre)/(Kp+Ki\*Ts/2)

//WfDem是燃油选择值 Kp:比例系数 Ki：积分洗漱

Kp=K1\*Kp()

Ki=K2\*Ki()

if(ES==ES\_ GI||ES==ES\_ ABOVEIDLE)

Ps3MinLmt=180KPa

表9439 Kp()随()转速变化的规律表

表9443 Kp修正系数K1随T25和P25变化规律表

表9461 Ki修正系数K2随T25和P25变化规律表

**编号**：4.2.4.5.1.2.3

**功能**：Ps3最大限制回路燃油计算

**前置条件:**

**输入**：ps3

**输出**：WfPs3Max

**公式**：

Ts=0.02s

if(初始标记位==1)

{

StatePs3MinPre=PsMinLmt-Ps3(StatePs3MinPre初值)

WfPs3MinPre=WPre(StatePs3MinPre初值)

}

WfPs3Max=(Kp+Ki\*Ts/2)\*(Ps3MaxLmt-Ps3)-(Kp-Ki\*Ts/2)\*StatePs3MaxPre+WfPre

StatePs3Max=(Ps3MaxLmt-Ps3)+(WfPre-WfPs3MaxPre)/(Kp+Ki\*Ts/2)

//Kp:比例系数 Ki：积分系数

Kp=K1\*Kp()

Ki=K2\*Ki()

if(ES==ES\_ GI||ES==ES\_ ABOVEIDLE)

Ps3MaxLmt=3550KPa

表9439 Kp()随()转速变化的规律表

表9443 Kp修正系数K1随T25和P25变化规律表

表9461 Ki修正系数K2随T25和P25变化规律表

**编号**：4.2.4.5.1.2.4

**功能**：T3最大限制回路燃油计算

**前置条件:**

**输入**：T3

**输出**：WfT3Max

**公式**：

Ts=0.02s

if(初始标记位==1)

{

StateT3MinPre=TMinLmt-T3(StateT3MinPre初值)

WfT3MinPre=WPre(StateT3MinPre初值)

}

WfT3Max=(Kp+Ki\*Ts/2)\*(T3MaxLmt-T3)-(Kp-Ki\*Ts/2)\*StateT3MaxPre+WfPre

StateT3Max=(T3MaxLmt-T3)+(WfPre-WfT3MaxPre)/(Kp+Ki\*Ts/2)

//Kp:比例系数 Ki：积分系数

Kp=K1\*Kp()

Ki=K2\*Ki()

if(ES==ES\_ GI||ES==ES\_ ABOVEIDLE)

T3MaxLmt=920K

表9450 Kp()随()转速变化的规律表

表9448 Kp修正系数KP随T25和P25变化规律表

表9446 Ki修正系数Ki随T25和P25变化规律表

**编号**：4.2.4.5.1.2.5

**功能**：加速状态燃油计算

**前置条件:**

**输入**：加速状态标记，N2r25p，P0，T25，Ps3，P25，N2dot

**输出**：WfAcc

**公式**：

if(初始标记位==1)

{

dWfN2dotAccPre=0；

WfN2dotAccPre=0；

}

Ts=0.02s

Wf\_P\_r=find[表8404，N2r25p，P0]；//压比限制换算燃油

Wf\_T\_r=find[表8405，N2r25p，T25]；//温度限制换算燃油

, a=0.70

N2dotAcc\_r=find[表8406，N2r25p，P0]；//换算加速率控制期望

N2dotAccDem= N2dotAcc\_r\*P25

N2dotAccCmd=min(N2dotAccDem, N2dotAccelLmt)

dWfN2dotAcc=Ki\*Ts\*( N2dotAccCmd-N2dot)+ dWfN2dotAccPre+(WfPre-WfN2dotAccPre)

WfNdotAcc= dWfN2dotAcc + dWfN2dotAccSche

=find（表8409，N2r25p）；

, a=0.70

Ki\_std=find[表8411，N2r25p]

Ki\_cor=find[表8413，P25,T25]

Ki=Ki\_std\*Ki\_cor

WfAcc=Wf\_Acc\_Sch

### 4.2.5 机内自检测要求

**编号**：4.2.5.7

**功能**：LVDT激励电路BIT诊断

**前置条件:** after(StbTime,ms,定时任务开始)

**输入**：发动机状态字，发动机关断活门位置开关（SWI15），N2r25

**输出**：高压关断活门位置告警字

**公式**：

if(!During(100,ms, SWI15==连接)) // SWI15是否状态是连接和断开

高压关断活门位置告警字=1

if(ES==ES\_NORMALSTOP&&During(1.2,s,SWI15==断开))

高压关断活门位置告警字=1

if(高压关断活门位置告警字==1)

高压关断活门位置告警字可清除标志=false

**任务名称**：开关量输入信号处理电路BIT诊断

**编号**：4.2.5.8

**功能**：开关量输入信号处理电路自检仅检测处理电路的状态，无法检测外部开关的状态。

**前置条件:**

**输入**：JRJ35.IO,JRJ35.I.SWI

**输出**：SWI01.BITFlt-SWI24.BITFlt

**公式**：

if(信号为单余度 && 本通道不采集的开关量)

{

XX.BITFlt = NOFAULT;

}

else

{

if(after(5,ms,开关量输入处理电路电源==off))

{

采集开关量输入状态;

if(XX.Data == 0)

{

XX.BITFlt = FAULT;

}

if(XX.Data == 1)

{

XX.BITFlt = NOFAULT;

}

}

}

4.2.5.9 开关量输出电路BIT诊断

开关量输出信号处理电路BIT诊断提供断线检测、在线检测和过流检测三种，主控和备份通道均可检测。支持这些BIT功能的开关量输出信号有：SWO01- SWO15

SWO03检测周期 = SWO03\_ProcTime

**任务名称**：起动空气阀电磁铁驱动信号（SWO3）BIT诊断

**编号**：4.2.5.9.1

**功能**：

**前置条件:** after(60,ms,定时任务开始)

**输入**：SWO03BIT.OFLSts（硬件断线检测结果）、SWO03BIT.OLSts（在线检测结果）、SWO03BIT.SCSts（过流检测结果）, SWO03.Sts, ChanSts, SWO03.BITSetTime, SWO03.BITClrTime, 通道健康状态，SWO03.PtnrFlt

**输出**：SWI03.OFLFlt, SWO03.OLFlt, SWO03.SCFlt, SWO03.SelfBITFlt

**公式**：

获取SWO03BIT.OFLSts, SWO03BIT.OLSTS, SWO03BIT.SCSts

if(SWO03 == UNOPEN || ChanSts == 0)

{

if(duration(SWO03.BITSetTime,s,SWO03BIT.OFLSts == 1))

{

SWO03.OFLFlt = 1 ;

}

if(duration(SWO03.BitClrTime,s,SWO03BIT.OFLSts == 0))

{

SWO03.OFLFlt = 0 ;

}

}

else

{

if(duration(SWO03.BITSetTime,s,SWO03BIT.OLSts == 0))

{

SWO03.OLFlt = 1 ;

}

if(duration(SWO03.BitClrTime,s,SWO03BIT.OLSts == 1))

{

SWO03.OLFlt = 0 ;

}

if(duration(SWO03.BITSetTime,s,SWO03BIT.SCSts == 1))

{

SWO03.SCFlt = 1 ;

SWO03.SCFltFlag = 1; // 锁定不可清除。

}

}

if( SWO03.OFLFlt == 1 || SWO03.OLFlt == 1 || SWO03.SCFlt == 1)

{

SWO03.SelfBITFlt = 1;

}

else

{

SWO03.SelfBITFlt = 0;

}

SWO06检测周期为SWO06\_ProcTime

**任务名称**：备份电磁阀（SWO06）BIT诊断

**编号**：4.2.5.9.2

**功能**：

**前置条件:** after(60,ms,定时任务开始)

**输入**：SWO06BIT.OFLSts、SWO06BIT.OLSts、SWO06BIT.SCSts, SWO06.Sts, ChanSts, SWO06.BITSetTime, SWO06.BITClrTime

**输出**：SWI06.OFLFlt, SWO06.OLFlt, SWO06.SCFlt, SWO06.SelfBITFlt

**公式**：

处理流程与4.2.5.9.1类似，此处略

SWO07检测周期为SWO07\_ProcTime

**任务名称**：主燃级分级活门控制电磁阀（SWO07）BIT诊断

**编号**：4.2.5.9.3

**功能**：

**前置条件:** after(60,ms,定时任务开始)

**输入**：SWO07BIT.OFLSts、SWO07BIT.OLSts、SWO07BIT.SCSts, SWO07.Sts, ChanSts, SWO07.BITSetTime, SWO07.BITClrTime

**输出**：SWI07.OFLFlt, SWO07.OLFlt, SWO07.SCFlt, SWO07.SelfBITFlt

**公式**：

处理流程与4.2.5.9.1类似，此处略

保证SWO08检测周期为SWO08\_ProcTime

**任务名称**：点火器1驱动信号（SWO08）BIT诊断

**编号**：4.2.5.9.4

**功能**：

**前置条件:** after(60,ms,定时任务开始)

**输入**：SWO08BIT.OFLSts、SWO08BIT.OLSts、SWO08BIT.SCSts, SWO08.Sts, ChanSts, SWO08.BITSetTime, SWO08.BITClrTime

**输出**：SWI08.OFLFlt, SWO08.OLFlt, SWO08.SCFlt, SWO08.SelfBITFlt

**公式**：

处理流程与4.2.5.9.1类似，此处略

保证SWO09检测周期为SWO09\_ProcTime

**任务名称**：主燃级分级活门控制电磁阀（SWO09）BIT诊断

**编号**：4.2.5.9.5

**功能**：

**前置条件:** after(60,ms,定时任务开始)

**输入**：SWO09BIT.OFLSts、SWO09BIT.OLSts、SWO09BIT.SCSts, SWO09.Sts, ChanSts, SWO09.BITSetTime, SWO09.BITClrTime

**输出**：SWI09.OFLFlt, SWO09.OLFlt, SWO09.SCFlt, SWO09.SelfBITFlt

**公式**：

处理流程与4.2.5.9.1类似，此处略

4.2.5.10 模拟量输出电路BIT诊断

模拟量输出包括Ifmv、Ivsv、Itbv4L、Itbv4R、Ifst、Ifsv、Ibak、电液伺服阀总闸，相应的在线监测回采信号见4.2.4.3.30至4.2.4.3.37

保证Ifmv检测周期为Ifmv \_ProcTime

**任务名称**：Ifmv输出电路BIT诊断

**编号**：4.2.5.10.1

**功能**：

**前置条件:** after(60,ms,定时任务开始)

**输入**：IfmvTE.SqlData, Ifmv.BITMin, Ifmv.BITMAX, Ifmv.CASCSts, Ifmv.CBSCSts, Ifmv.BITSetTime

**输出**：Ifmv.BreakBITFlt, Ifmv.SCFlt, Ifmv.SelfBITFlt

**公式**：

If(duration(Ifmv .BITSetTime,s,Ifmv.SqlData < Ifmv.BITMin || IfmvTE.SqlData > Ifmv.BITMax))

{

Ifmv.BreakBITFlt = 1;

}

If(duration(Ifmv .BITSetTime,s,Ifmv.BITMin <=IfmvTE.SqlData <= Ifmv.BITMax))

{

Ifmv.BreakBITFlt = 0;

}

If(duration(Ifmv.BITSetTime,s,Ifmv.CASCSts == 1 || Ifmv.CBSCSts == 1))

{

Ifmv.SCFlt =1;

Ifmv.SCFltFlag = 1；

}

if(Ifmv.BreakBITFlt == 1 || Ifmv.SCFlt == 1)

{

Ifmv.SelfBITFlt = 1;

}

else

{

Ifmv.SelfBITFlt = 0;

}

Ifmv检测周期为Ifmv \_ProcTime

**任务名称**：Ifmv输出电路BIT诊断

**编号**：4.2.5.10.2 - 4.2.5.10.8

**功能**： 模拟量输出电路BIT诊断，处理流程均与4.2.5.10.1相似，此处略

**前置条件:** after(60,ms,定时任务开始)

**输入**：XX.SqlData, XX.BITMin, XX.BITMAX, XX.CASCSts, XX.CBSCSts, XX.BITSetTime

**输出**：XX.BreakBITFlt, XX.SCFlt, XX.SelfBITFlt

**任务名称**：燃油分配控制

**编号**：4.2.4.5.2

**功能**：

**前置条件:**

**输入**：Wf, Ps3, T3, EL, N2r25p, T25重构T3策略, 起动中止条件确认时刻， 正常停车条件确认时刻， 紧急停车条件确认时刻

**输出**：FSV控制位置期望指令LfsvDem, FST控制位置期望指令LfstDem

**公式：**

if(（T3>=700K && Ps3>=1.10MPa）||（T3>=657K && Ps3>=3.45MPa）)

{

主燃级分配活门=1；

;

;

查找9100，9102，9107，9109，9111获得Main+P1；

}

if（T3<657K ||( 657K <=T3<700K && Ps3<3.45MPa) || (T3>=700k && Ps3<1.10MPa)）

{

主燃级分配活门=0；

分配比与分配活门位置关系以一维插值表的形式给定，具体数据待定；

}

if(T25重构T3策略==1)

{

T3=1.8\*T25;

}

If(正常停车条件确认时刻)

{

Stay（NstopCfmTimerThsld，s，前一时刻的LfsvDem）

{

关闭主燃级分配活门=0；

}

}

If(紧急停车条件确认时刻)

{

Stay（EstopCfmTimerThsld，s，前一时刻的LfsvDem）

{

关闭主燃级分配活门；

}

}

If（大功率切油再起动状态==0）

{

喷嘴加富比=1.7;

}

If（大功率切油再启动状态==1）

{

喷嘴加富比=4;

}

If(ES == ES\_ GRDSTART)

{

喷嘴加富比=1;

}

Lfmv伺服回路控制周期应为5ms

**编号**：4.2.4.7.1

**功能**：Lfmv伺服回路控制

**前置条件:**

**输入**：WfDem，Lfmv，故障对策字，发动机状态字

**输出**：FMV控制电流指令Ifmv

**公式**：

LfmvDem=WfDem以及燃油标定曲线（由实验整定）

//k为第k个采样周期

,

if(ES=ES\_ STAND)

Ifmv=Fmv\_AdjServoZeroCurrent

if(FMV安全位置策略==1)

Ifmv=Fmv\_AdjServosafeCurrent（暂不实施）

if(Ifmv零电流策略==1)

Ifmv=Fmv\_AdjServoZeroCurrent

SWO10的输出周期为20ms

**编号**：4.2.3.6.3

**功能**：SWO10N2（1924rpm）接入信号

**前置条件:**

**输入**：N2p，N2，FADEC交流发电机电源过压检测信号，FADEC交流发电机电源欠压检测信号，交流电接入失败锁定标记，“交流电过压”故障，“交流电欠压”故障

**输出**：SWO10

**公式**：

Output(500,ms, SWO10)

if((N2p>=12.8%||N2>=1924rpm)&& FADEC交流发电机电源过压检测信号==0&& FADEC交流发电机电源欠压检测信号==0&&交流电接入失败锁定标记==0)

Output(SWO10N2（1924rpm）接入信号)

if(28VDC电源切断刀位检测信号==1)

Output(SWO10)

if(输出SWO10&&(N2<1924rpm||“交流电过压”故障==1||“交流电欠压”故障==1||交流电接入失败锁定标记==1))

停止输出SWO10

Output(本通道)

Output(其他通道)

//信号双通道应同时输出

## 二．相关解释

**关键词注释：**

get[端口号A]：获取端口号A的值

duration(Time1,单位,condition1):当前时刻开始持续Time1个单位内保持条件condition1成立

duration(T0, Time2,单位, ,condition2): T0时刻开始持续Time2个单位内保持条件condition2成立

find[表1, 参数1, 参数2, …]：根据参数1、参数2等查表1获取变量值

after(Time3,单位,condition1)：condition1成立后Time3个单位时间

Output(var1) :变量var1输出

Output(time，unit，var2) ：var2持续输出time个单位

**新增内容：**

模块增加前置条件部分，模块在被调用时须满足前置条件后才可执行模块内容

模块增加通用模块调用，如：调用4.2.5.1[PLA]，表示调用4.2.5.1通用模块，PLA为通用模块所需参数

使用‘//或/\*\*/’来对需求进行注释描述

**数据字典示例：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 变量名 | 变量符号 | 类型 | 类型解释 | 变量注释 |
| TBV4L回路故障 |  | Bool | 1：故障，0：无故障 |  |
| ltbv4L输出故障 | ltbv4LOutputFlt | Bool | 1：故障，0：无故障 |  |
| ltbv4L传感器故障 | ltbv4LSensorFlt | Bool | 1：故障，0：无故障 |  |
| Selfltbv4L输出故障 | Selfltbv4LOutputFlt | Bool | 1：故障，0：无故障 | 本通道ltbv4L输出故障 |
| PtnrTBV4L回路故障 |  | Bool | 1：故障，0：无故障 | 另一通道TBV4L回路故障 |
| Itbv4L零电流输出策略字可清除标志 |  | Bool | 1：可清除，0：不可清除 |  |
| Itbv4L零电流输出策略字 |  | Bool | 1：启用，0：不启用 |  |

## 三．其他

以下两个框图内容为4.2.2部分转化信号输入管理需求的处理流程总结

|  |
| --- |
| 连续量输入信号通用处理流程：   1. 指定端口号获取BIT值 2. BIT诊断 3. 计算MeanData（此步骤不同的信号量计算方法有所不同） 4. 极值诊断 5. 斜率诊断 6. XX.SelfFlt赋值 7. 判定是否当前周期失效，诊断开始后的第一个周期为赋初值 8. 信号表决 9. XX.SignalFlt，XX.ValidData 赋值 10. 惯性滤波 11. 超前滞后校正 12. 限制步长变化。   性能要求：应保证XX处理周期为XX\_ProcTime。  约束条件：XX信号处理应在AD采样正常的条件下进行；  应在定时任务开始XX.FirstBITTime后对XX进行BIT故障诊断。  应在定时任务开始XX.FirstExtreTime后对XX进行极值故障诊断。  应在定时任务开始XX.FirstSlopeTime后对XX进行斜率故障诊断。 |

|  |
| --- |
| 连续量变化特殊情况单独处理：  Thptc的冷端信号源发生变化时，不进行斜率故障诊断。  Tk，Tp，Tpss未进行斜率故障诊断及极值故障诊断  Toil1，Toil2, Wf不进行斜率故障诊断。  P0，P25，Ps3, P3b, 未进行BIT诊断。  Tbak1, Tbak2不进行极值和斜率故障诊断。  Ltbv4, Ltbv4R, 不进行极值诊断  N2, Nac, Nq, Wf的BIT诊断仅在上电进行一次  IfmvTE, IvsvTE, Itbv4LTE, Itbv4RTE, IfstTE, IfsvTE, IbakTE, ISCMP, ISCMN, 处理流程仅为：获取SqlData，根据公式计算MeanData，性能要求为保证处理周期为XX\_ProcTime。 此类需求为模拟量输出回采信号。  LvdtPower1, LvdtPower2功能处理仅为直接获取XX.SqlData值。 |