



宇阳的 板形模型 私房菜

热轧模型组

Table of Contents

总览	1.1
板形设定计算	1.2
包络线计算	1.2.1
分配计算	1.2.2
反馈计算	1.3
板形LPCE模块	2.1
板形LRG模块	2.2
板形CRLC模块	2.3
板形ALC模块	2.4
板形TARGT模块	2.5
SSU日志验算	2.6
SSU日志FAQ	2.7
CFG模型参数的梳理	2.8

宇阳的板形模型私房菜（板形模型消化）

cShapeSetupD::Main(..)

cShapeSetupD对象有一个status状态量，显示当前板形设定的状态（红灯或绿灯），以及判断是否合法的指示器ok。初始默认情况下status设为红灯，ok设为false。

```
this->status = cMdlparam::cs_red;  
this->ok      = false;
```

roll_change_count是判断是否换辊的计数器。

```
int    roll_change_count (0);
```

redrft_perm用于判断是否可以重新分配各机架厚度或者重新分配压下。初始情况下redrft_perm设为false。

后根据如下条件更新redrft_perm的值。ssu_load_enab一般为false。ssu_granted一般为true。s_CalId指的是FCD对象中的Calculation ID，最小为1，最大为2。

```
redrft_perm = ( true == pcSched->pcSetupD->pcSetup  
->ssu_load_enab ) &&  
  ( true == pcSched->pcFSSched->pcSSys->state.ssu_g  
ranted ) &&  
  ( 1 == pcSched->pcFCD->state.s_CalId );
```

redrft_perm的值与iter相同。

```
if ( redrft_perm )
{
    iter = 1;
}
else
{
    iter = 0;
}
```

在日志中的lst.iter与源码中此处iter并不完全一样。

短期自学习设定

短期自学习的设定受到换辊和钢种、规格跳档的影响。

跳档

针对钢种和规格的跳档，模型考虑了以下五种情况。

```
this->family_chg      = false;
this->narrow_to_wide_chg = false;
this->wide_to_narrow_chg = false;
this->prd_chg          = false;
this->lot_chg           = false;
```

钢种族跳档、宽度由窄变宽、宽度由宽变窄，这些都好理解。

而prd_chg指的是：钢种族跳档、宽度由窄变宽、宽度由宽变窄这三种情况至少有一种出现。

```

if ( (true == this->family_chg)      ||
      (true == this->narrow_to_wide_chg) ||
      (true == this->wide_to_narrow_chg) )
{
    this->prd_chg = true;
}

```

lot_chg指的是和前一块带钢相比，钢种族、厚度索引、宽度索引其中至少一者发生改变，则称为lot_chg。用于弯辊力干预自适应的计算。

```

if ( // family change
      ((pcSched->pcFSSched->pcSAMP->state.pr_family > 0)
      ) &&
      (abs(pcSched->pcPDI->state.family - pcSched->pcFSSched->pcSAMP->state.pr_family) > 0)) ||
      // gauge range table index change
      ((pcSched->pcFSSched->pcSAMP->state.pr_grt_idx > 0) &&
      (abs(pcSched->pcPDI->state.grt_idx - pcSched->pcFSSched->pcSAMP->state.pr_grt_idx) > 0)) ||
      // width range table index change
      ((pcSched->pcFSSched->pcSAMP->state.pr_wrt_idx > 0) &&
      (abs(pcSched->pcPDI->state.wrt_idx - pcSched->pcFSSched->pcSAMP->state.pr_wrt_idx) > 0)) )
{
    this->lot_chg = true;
}

```

短期自学习预设 CSAMP::Reset_Verniers(..)

根据是否跳档以及换辊，短期自学习在开始进行设定计算前，更新凸度和平直度的自学习。所谓的预设其实是在不同的条件下清零自学习。

- 当有机架换辊，则累积增加相应机架的换辊次数 num_rolls_chgd。
- 若超过两个机架换辊，则清零凸度自学习与平直度自学习（vrn和err），并初始化设定辊形自学习，设定last_pos_shft为0。
- 当出现lot_chg或开轧，清零弯辊力补偿以及bnd_ofs_counter。
- 当出现钢种族跳档，则凸度自学习（vrn和err）衰减后继承，衰减系数目前0.99。
- 如果出现窄到宽的跳档，同时周期内轧制块数超过30块，并且存在中浪趋势，则清零平直度自学习（vrn和err），若末机架工作辊辊形自学习也小于零，则清零末机架工作辊辊形自学习wr_crn_vrn。
- 若出现宽到窄的跳档，不做任何设定修改。

更新完短期自学习表之后，并且当前计算阶段处于course-2之后，则put短期自学习表到模型数据库（models database）。

cShapeSetupD::Init(..)

cShapeSetupD::Init(..)初始化了动态的SHAPESETUP对象以及其它相关的动态对象，比如：LPCE、LRG、UFD和TARGT。除此之外，这个函数还计算了执行机构的软极限，同时复制外部的数据给合

适的动态对象。

cShapeSetupD::Init(..)初始化之后，最初的哪两个状态布尔值更新为true。

```
this->ok      = true;  
this->status = cMdlparam::cs_green;
```

cShapeSetupD::Init(..)的实现在shapsetup_req.cxx文件中。

长短期自学习初始化

首先初始化凸度和平直度的目标tgt_profile和tgt_flatness。这两个目标一开始是PDI目标加上操作工的补偿。

模型用prf_vrn_sel_flag和flt_vrn_sel_flag这两个参数来标识长短期自学习的选择，默认以长期自学习为主。

自学习值包括凸度自学习prf_vrn_rm_tmp和prf_vrn_rs_tmp，以及平直度自学习flt_vrn_tmp，初始的凸度或平直度自学习为长期自学习。当这一块带钢和上一块带钢相比，出现钢种或规格跳档，则将长期自学习加上上一块增益后的短期自学习，作为新的自学习值（注意这个功能凸度自学习已经取消增益，仅长期自学习值加上短期自学习）。

cTargtD::Init(..)

在cTargtD::Init(..)中主要确定初始的凸度以及目标有效凸度的极限。prf_vrn是prf_vrn_rm_tmp和prf_vrn_rs_tmp的差，flt_vrn就是flt_vrn_tmp。

目标flt为pdi平直度目标加上平直度的操作工补偿。

初始目标凸度与平直度稍有差别。

```
prf_int = (pdi_prf + prf_op_off) * matl_exp_cof + prf_vrn;
```

prf_int为pdi凸度加上操作工补偿后的热态凸度，再加上凸度自学习量。也就是说，凸度自学习量是补偿热态下的凸度。

之后用凸度的容许偏差计算单位凸度的上下极限。

初始化的大循环

pcTargtD->Init(..)执行完之后，从首道次机架从前往后进行一系列的初始化工作，将近700行代码。

首先将sprp的相关调整系数初始化到相应的对象中（pcFSStdD），供后续板形计算使用，如ufd_mult和force_bnd_nom。

对非空道次计算出入口厚度对轧制力的偏导数或增益DForce_DEntthick、DForce_DExthick。之后初始化板形相关的动态对象，按先后顺序分别为UFD对象、CRLC对象和LRG对象。

注意在CRLC对象的初始化中，SPRP中的工作辊凸度补偿f_wr_crn_off_adj需要加到长期自学习工作辊凸度补偿psSLFG->wr_crn_off上。

f_wr_crn_off_adj的设定根据出口凸度分为三档。

```
float f_wr_crn_off_adj = 0.0F;
if ( pcTargtD->prf_del < 0.045F )
{
    f_wr_crn_off_adj = psSPRP->wr_crn_off_adj [ pass
Idx ] ;
}
else if ( pcTargtD->prf_del < 0.065F )
{
    f_wr_crn_off_adj = psSPRP->wr_crn_off_adj2 [ pas
sIdx ] ;
}
else
{
    f_wr_crn_off_adj = psSPRP->wr_crn_off_adj3 [ pas
sIdx ] ;
}
```

初始化分配和评估的横向带钢对象，利用pcFSPassD->pcAlcLPceD[iter]->Init(..)和pcFSPassD->pcEvILPceD[iter]->Init(..)进行初始化。

之后初始化和STD对象相关的参数。

- 窜辊和弯辊能否使用的标识shft_enab和bnd_enab。
- op弯辊力补偿op_bnd_off。
- 弯辊力pcFSStdDloc->force_bnd。
- 初始化窜辊位置。
- 计算窜辊的软极限。
- 计算弯辊的软极限。
- 插值计算中间坯凸度prf_pass0和中间坯单位凸度pu_prf_pass0。

- 利用长期平直度自学习值修正末道次F7的弯辊力极限
force_bnd_lim。

同样在STD初始化过程中有这几点需要注意。

第一，注意GSM_RB_OFS_BLEEDOFF这个宏在ssu_features.hxx中定义为（1）。

```
#define GSM_RB_OFS_BLEEDOFF          (1)          // [
-] GSM Operator RB offsets bleed off enabled
```

如果GSM_RB_OFS_BLEEDOFF为真或为1，需要用短期自学习的弯辊补偿bending_ofs修正op弯辊力补偿op_bnd_off。

第二，如果bnd_enab为true，并且轧辊辊形是平辊，则不用目标平直度自学习量flt_vrn修正（从中减去）初始的弯辊力force_bnd_nom；如果轧辊不是平辊，则用目标平直度自学习量flt_vrn修正（从中减去）初始的弯辊力force_bnd_nom。

```
if ( pcFSStdDloc == pcLstFSPassD->pcFSStdD[ iter ]
&& pcFSStdDloc->pcStdRollPrD->getProf(op_work) != rp
_parab )
{
//-----
-----
// If last stand and the roll is not parabolic it SH
OULD be CVC
//-----
-----
    pcFSStdDloc->force_bnd = pcFSStdDloc->pcFSStd->f
orce_bnd_nom
    - pcTargtD->flt_vrn;
```

```

}
else
{
    pcFSStdDloc->force_bnd = pcFSStdDloc->pcFSStd->f
    orce_bnd_nom;
}

```

force_bnd的值和从属的对象必须从日志中理清楚。如psSSys->force_bnd和pcFSStdDloc->force_bnd。

第三，设定了一个level_std标识，当换辊开轧后用于锁定窜辊，以便操作工进行调平。

```

if ( pcShapeSetup->num_coils_to_lvl >=
    pcFSStdDloc->pcStdRollPrD->getNBarRolled(rpos_top
, op_work) )
{
    level_std = true;
}

```

对于平辊，在轧制用于调平的带钢时，相应机架的窜辊位置为零位。

```

// these are parabolic rolls
// use SCF function references
if ( level_std )
    pcFSStdDloc->wr_shft = 0.0F;
else
    pcFSStdDloc->wr_shft = psSSys->targ_pos_shft[ pa
ssIdx ];

```

在确定窜辊软极限过程中，如果窜辊被操作工锁定了，那么窜弯辊的软极限就是当前窜辊值。

在确定窜辊软极限过程中，需要先根据窜辊的速度计算最大的窜辊位置变化量。如果是平辊的话直接选取SCF中的设定。

在中间坯凸度和单位凸度的计算中，如果有中间坯测量的凸度则用中间坯测量的凸度，如果没有则插值计算。中间坯或0道次的单位凸度和单位有效凸度包络线最大最小值均为中间坯单位凸度的计算值。同时pcTargtD->en_pu_prf等于pu_prf_pass0。代码中pcFSPassD为中间坯道次。

```
for ( int i = min1; i <= max1; i++ )
{
    pcFSPassD->pcVecPEnvD[ iter ]->pu_prf_env[ i ]
    =
        pu_prf_pass0;
    pcFSPassD->pcVecPEnvD[ iter ]->ef_pu_prf_env[ i ]
    =
        pu_prf_pass0;
}

//-----
//-----
// Initialize mill entry per unit profile
//-----
//-----
pcTargtD->en_pu_prf = pu_prf_pass0;
```

注意pcTargtD->flt_vrn和psSLFG->flt_vrn、psSAMP->flt_vrn之间的区别。

cShapeSetupD::References(..)

cShapeSetupD::References(..)计算了凸度与平直度控制目标下的相关设定值，必要时重新分配轧制力或压下。

首先从F1到F7更新动态LPCE对象和LRG对象。

进行包络线计算cPEnvD::Calculate(..)和分配计算cAlcD::Calculate(..)，之后从F1到F7对板形设定进行评估。

注意在每一道次评估前需要用op弯辊力补偿修正弯辊力。

```
force_bnd =  
    pcFSPassD->pcFSStdD[ iter ]->force_bnd +  
    pcFSPassD->pcFSStdD[ iter ]->op_bnd_off;
```

如果可以重新分配压下，或者平直度不满足条件，则需要重新进行板形分配计算和再一次的评估。

最后计算传递给一级的增益，以及用pcTargtD->Eval_Delvry_Pass(..)评估末道次。

cPEnvD::Calculate(..)

这个函数中主要计算包络线。

F1到F7包络线初始化

包络线计算初始化的过程中，从F1到F7前机架到末机架，计算各个道次的ufd_pu_prf_env以及std_ex_strn_lim，并利用这二者计算出入口（上一道次）有效单位凸度的极限pcFSPassD->pcPrvAct->pcPEnvD->ef_pu_prf_lim。

首先根据弯窜辊的极限值确定弯窜辊包络线的最大最小值。注意，弯辊力极限的最大值对应弯辊力包络线的最小值，弯辊力极限的最小值对应弯辊力包络线的最大值。窜辊的极限最值和包络线的最值同样是相反对应的。

```
// 弯辊力包络线的赋值
pcFSPassD->pcPEnvD->force_bnd_env[ minl ] =
  pcFSPassD->pcFSStdD[ iter ]->force_bnd_lim[ maxl ];

pcFSPassD->pcPEnvD->force_bnd_env[ maxl ] =
  pcFSPassD->pcFSStdD[ iter ]->force_bnd_lim[ minl ];

// 窜辊位置包络线的赋值
pcFSPassD->pcPEnvD->pos_shft_env[ minl ] =
  pcFSPassD->pcFSStdD[ iter ]->wr_shft_lim[ maxl ];
```

```
pcFSPassD->pcPEnvD->pos_shft_env[ maxl ] =
pcFSPassD->pcFSStdD[ iter ]->wr_shft_lim[ minl ];
```

之后用窜辊位置的极限值，代入pcCRLCD->Crns(..)计算辊系凸度的极限值pce_wr_crn_lim[maxl/minl]。最大值对应最大值，最小值对应最小值。

单位轧制力的大小极限值，直接用单位轧制力赋值。可以见板形模型单位轧制力的验算内容。

```
for ( i = minl; i <= maxl; i++ )
{
    pcFSPassD->pcPEnvD->force_pu_wid_lim[ i ] =
        pcFSPassD->pcPEnvD->force_pu_wid;
}
```

用默认窜辊位置wr_shft_nom，代入pcCRLCD->Crns(..)计算辊系凸度pce_wr_crn和wr_br_crn。

计算pcFSPassD->pcPEnvD->dprf_dfrcw，这里的偏导数dprf_dfrcw是用来判断force_pu_wid_lim给force_pu_wid_env赋值的方向。

```
//-----
//-----
// Calculate the UFD roll roll gap profile derivative with respect
// rolling force per unit piece width.
//-----
//-----

pcFSPassD->pcPEnvD->dprf_dfrcw =
    pcFSPassD->pcFSStdD[ iter ]->pcUFDD->Dprf_Dfrcw (
```



```

        pcFSPassD->pcPEnvD->force_pu_wid,
        pcFSPassD->pcFSStdD[ iter ]->pcFSStd
->force_bnd_nom,
        pce_wr_crn,
        wr_br_crn
    )
;

//-----
// Initialize the rolling force per unit width piece
// envelope.
// Note: The UFD roll gap profile derivative is used
// for direction.
//-----
if ( 0.0 <= pcFSPassD->pcPEnvD->dprf_dfrcw )
{
    pcFSPassD->pcPEnvD->force_pu_wid_env[ minl ] =
        pcFSPassD->pcPEnvD->force_pu_wid_lim[ minl ];

    pcFSPassD->pcPEnvD->force_pu_wid_env[ maxl ] =
        pcFSPassD->pcPEnvD->force_pu_wid_lim[ maxl ];

}
else
{
    pcFSPassD->pcPEnvD->force_pu_wid_env[ minl ] =
        pcFSPassD->pcPEnvD->force_pu_wid_lim[ maxl ];

    pcFSPassD->pcPEnvD->force_pu_wid_env[ maxl ] =
        pcFSPassD->pcPEnvD->force_pu_wid_lim[ minl ];

}

```

辊系凸度的极限值赋值给辊系凸度的包络线最大最小值。注意这里，极限值的最小值对应包络线的最大值，极限值的最大值对应包络线的最小值。

```

//-----
// Initialize the piece to work roll
stack crown and work roll
// backup roll stack crown envelopes.
//-----
-----
pcFSPassD->pcPEnvD->pce_wr_crn_env[
minl ] =
pcFSPassD->pcPEnvD->pce_wr_crn_l
im[ maxl ];
pcFSPassD->pcPEnvD->wr_br_crn_env[ m
inl ] =
pcFSPassD->pcPEnvD->wr_br_crn_lim
[ maxl ];
pcFSPassD->pcPEnvD->pce_wr_crn_env[
maxl ] =
pcFSPassD->pcPEnvD->pce_wr_crn_l
im[ minl ];
pcFSPassD->pcPEnvD->wr_br_crn_env[ m
axl ] =
pcFSPassD->pcPEnvD->wr_br_crn_lim
[ minl ];

```

此时，我们已经有了单位轧制力包络线force_pu_wid_env、弯辊力的包络线force_bnd_env、辊系凸度的包络线pce_wr_crn_env和wr_br_crn_env，利用pcUFDD->Prf(..)计算出UFD单位凸度的包络线

ufd_pu_prf_env。

```
for ( i = minl; i <= maxl; i++ )
{
    //-----
    // Establish the minimum / maximum UFD roll gap per unit profile
    // envelope.
    //-----
    line_num = __LINE__;
    pcFSPassD->pcPEnvD->ufd_pu_prf_env[ i ] =
        pcFSPassD->pcFSStdD[ iter ]->
        pcUFDD->Prf(
            pcFSPassD->pcPEnvD->for
            ce_pu_wid_env [ i ],
            pcFSPassD->pcPEnvD->for
            ce_bnd_env    [ i ],
            pcFSPassD->pcPEnvD->pce
            _wr_crn_env   [ i ],
            pcFSPassD->pcPEnvD->wr_
            br_crn_env    [ i ] ) /
            pcFSPassD->pcFSStdD[ iter ]->
            pcExPceD->thick;
}
```

确定出口应变差极限std_ex_strn_lim。

```
for ( i = we; i <= cb; i++ )
{
    //-----
```

```

-----
// Retrieve the piece critical buckl
ing limits for the given pass.
//-----
-----
line_num = __LINE__;
pcFSPassD->pcPEnvD->std_ex_strn_lim[
i ] =
pcFSPassD->pcLPceD->Crit_Bckl_Lim
( i );
}

```

利用各个道次的ufd_pu_prf_env以及std_ex_strn_lim二者计算出进口（上一道次）有效单位凸度的极限pcFSPassD->pcPrvAct->pcPEnvD->ef_pu_prf_lim。如果当前道次的带钢影响系数接近0，则松弛入口（上一道次）有效单位凸度的极限为正负1。其中ufd单位凸度包络线最小值和边浪极限一起参与计算，ufd单位凸度巴洛熙最大值和中浪极限一起参与计算。

```

//-----
-----
// Calculate the effective per unit
profile limits.
//-----
-----
line_num = __LINE__;
pcFSPassD->pcPrvAct->pcPEnvD->ef_pu_
prf_lim[ minl ] =
pcFSPassD->pcFSStdD[ iter ]->pcL
RGD->Ef_En_PU_Prfl(
pcFSPassD->pcPEnvD->u
fd_pu_prf_env [ minl ],

```

```

                                                                    pcFSPassD->pcPEnvD->s
td_ex_strn_lim [ we ] );

        line_num = __LINE__;
        pcFSPassD->pcPrvAct->pcPEnvD->ef_pu_
prf_lim[ maxl ] =
        pcFSPassD->pcFSStdD[ iter ]->pcL
RGD->Ef_En_PU_Prfl(
                                                                    pcFSPassD->pcPEnvD->u
fd_pu_prf_env  [ maxl ],
                                                                    pcFSPassD->pcPEnvD->s
td_ex_strn_lim [ cb ] );

```

考虑到最后一道次即末道次的目标可能发生变化，因此最后一道次的有效单位凸度的极限也松弛为正负1。

第一道次入口的pcLPceD->ef_pu_prf赋值给第一道次入口的pcPEnvD->ef_pu_prf_env包络线。同时初始化包络线限制道次数。

```

        for ( i = minl; i <= maxl; i++ )
        {
            //-----
            // Initialize first pass effective entry
            per unit profile.
            //-----
            ( ( cFSPassD* )(pcFstFSPassD->previous_o
bj) )->pcPEnvD->ef_pu_prf_env[ i ] =
            ( ( cFSPassD* )(pcFstFSPassD->previo
us_obj) )->pcLPceD->ef_pu_prf;

            //-----

```

```

        // Initialize the limiting pass envelope.

        //-----

        pas_env_lim[ i ] =
            ( ( cFSPassD* )(pcFstFSPassD->previo
us_obj) )->pcPass->num;
    }

```

协调包络线

确定包络线最小值组份

move_prv用来指示道次是否前移。是否前移，说明上一机架的有效单位凸度包络线是否存在调整变化的空间。

利用pcLRGD->Ef_Ex_PU_Pr3(..)计算，将上一道次的有效单位凸度包络线下限和本道次的ufd有效凸度包络线代入，获得本道次出口的有效单位凸度包络线下限pcFSPassD->pcPEnvD->ef_pu_prf_env[minl]。

```

        pcFSPassD->pcPEnvD->ef_pu_prf_env[ minl ]
    =
        pcFSPassD->pcFSStdD[ iter ]->pcLRGD->
Ef_Ex_PU_Pr3 (
                                pcFSPassD->pcLPceD->s
trn_rlf_cof,
                                pcFSPassD->pcPrvAct->
pcPEnvD->ef_pu_prf_env[ minl ],
                                pcFSPassD->pcPEnvD->u
fd_pu_prf_env[ minl ] );

```

如果本道次出口的有效单位凸度包络线下限低于有效单位凸度极限的下限，则需要进行一系列重新计算。

重新计算中包括ufd_pu_prf、istd_ex_pu_prf、ef_en_pu_prf，并利用上一道次的ef_pu_prf_env来clamp获得入口有效单位凸度（包络线下限）临时值ef_en_pu_prf_buf。

```
ef_en_pu_prf_buf =  
    cMathUty::Clamp( ef_en_pu_prf,  
        pcFSPassD->pcPrvAct->pcPEnvD->ef_pu_prf_env[  
minl ],  
        pcFSPassD->pcPrvAct->pcPEnvD->ef_pu_prf_env[  
maxl ] );
```

之后更新move_prv的指示器。

```
move_prv[ minl ] =  
    ( ef_en_pu_prf_buf !=  
        pcFSPassD->pcPrvAct->pcP  
EnvD->ef_pu_prf_env[ minl ] ) &&  
    ( pcFSPassD->pcPrvAct->pcPEN  
vD->ef_pu_prf_env[ minl ] !=  
        pcFSPassD->pcPrvAct->pcP  
EnvD->ef_pu_prf_env[ maxl ] );
```

更新上一道次或入口有效单位凸度极限的最小值，注意是极限。

```
pcFSPassD->pcPrvAct->pcPEnvD->ef_pu_prf_lim[ minl ] =  
    ef_en_pu_prf_buf;
```



这时判断move_prv前移指示器的状态，如果不能前移，说明上一道次的有效单位凸度或入口有效单位凸度不存在可调整和变化的空间，则将入口有效包络线的下限赋值给临时量ef_en_pu_prf_buf。

```
if ( !move_prv[ minl ] )
{
    ef_en_pu_prf_buf =
        pcFSPassD->pcPrvAct->pcP
EnvD->ef_pu_prf_env[ minl ];
}
```

之后用新的ef_en_pu_prf_buf值和ef_ex_pu_prf值，更新ufd有效单位凸度ufd_pu_prf，涉及的函数是pcLRGD->UFD_PU_Pr3(..)。并利用新的ufd单位凸度ufd_pu_prf、弯辊力和窜辊包络线的下限，计算变化后的辊系凸度pce_wr_crn和wr_br_crn。

再用pcCRLCD->Shft_Pos(..)更新窜辊位置包络线的下限。之后再次重计算（re-calculate）辊系凸度。接着考虑弯辊力包络线下限force_bnd_env_min，重计算（re-calculate）辊系凸度。

```
//-----
//-----
// Re-calculate the following compos
ite roll stack crown quantities:
//      Piece to work roll stack crown
//      Work roll to backup roll stac
k crown
//-----
```



```

-----
        line_num = __LINE__;
        pcFSPassD->pcFSStdD[ iter ]->pcCRLCD
->Crns(
                                pcFSPassD->pcPEnvD->pos_sh
ft_env  [ minl ],                                pcFSPassD->pcPEnvD->angl_p
c_env   [ minl ],                                pcFSPassD->pcPEnvD->pce_wr
_crn_env[ minl ],                                pcFSPassD->pcPEnvD->wr_br_
crn_env [ minl ] );

```

在弯窜辊都修正辊系凸度后，确定合适的弯辊力force_bnd_des，设定force_bnd_clmp指示器。

```

        force_bnd_clmp =
            force_bnd_des != pcFSPassD->pcPE
nvD->force_bnd_env[ minl ];

```

最终更新ufd有效单位凸度包络线的下限ufd_pu_prf_env_min。

如果force_bnd_des不等于弯辊力包络线的下限值，那么还需要调整。重新更新ef_en_pu_prf_buf以及更新上一道次或入口有效单位凸度极限的最小值。

如果未前移，则进行如下计算，从1190到1599行。如果出口应变差std_ex_strn超出出口应变差的极限范围，则进行一系列修正，目前这段修正在模型中被禁止执行。这样设置的原因是避免单位轧制力包络线和辊系凸度做大规模的修改和变化影响生产稳定性，出点浪形问题也是可以接受的。

```

        // was profile reduced too much
        //if ( std_ex_strn < pcFSPassD->pcPE
nvD->std_ex_strn_lim[ cb ] )
        if ( 1 < 0 )
        {...}
        // strain too high (due to low entry
profile)
        //if ( std_ex_strn > pcFSPassD->pcPE
nvD->std_ex_strn_lim[ we ] &&
        if ( 0 > 1 &&
            pcFSPassD->pcFSStdD[ iter ]->pc
LRGD->pce_infl_cof >
            pcFSPassD->pcPEnvD->pcPEnv->
pce_infl_cof_mn )
        {...}
        // 1 < 0 和 0 > 1说明这两段调整永远不
会执行

```

最后再更新一次本道次的出口有效单位凸度包络线下限。

```

        pcFSPassD->pcPEnvD->ef_pu_prf_env[ min1 ]
=
        pcFSPassD->pcFSStdD[ iter ]->pcLRGD->
Ef_Ex_PU_Pr3(
            pcFSPassD->pcLPceD->strn_
r1f_cof,
            pcFSPassD->pcPrvAct->pcPE
nvD->ef_pu_prf_env[ min1 ],
            pcFSPassD->pcPEnvD->ufd_p
u_prf_env [ min1 ] );

```


cAlcD::Calculate(..)

凸度分配计算。

Delvry_Pass(..)之前

cAlcD::Calculate(..)开始时，首先赋值中间坯的“分配厚度”。中间坯的“分配厚度”实际为F1的入口厚度，即中间坯的实际厚度。接着计算F1到F7的单位轧制力、分配厚度，以及引用轧辊咬入相关的对象。

如果可以重新分配压下，那么还会计算轧制力的最大改变量。

之后计算总的单位凸度改变量pu_prf_change_sum。

```
pu_prf_change_sum +=  
  pcFSPassDtmp->pcEvllPceD[ iter ]->strn_rlf_cof  
  / (pcFSPassDtmp->pcFSStdD[ iter ]->pcLRGD->pce_infl  
  l_cof  
  * pcFSPassDtmp->pcEvllPceD[ iter ]->elas_modu);
```

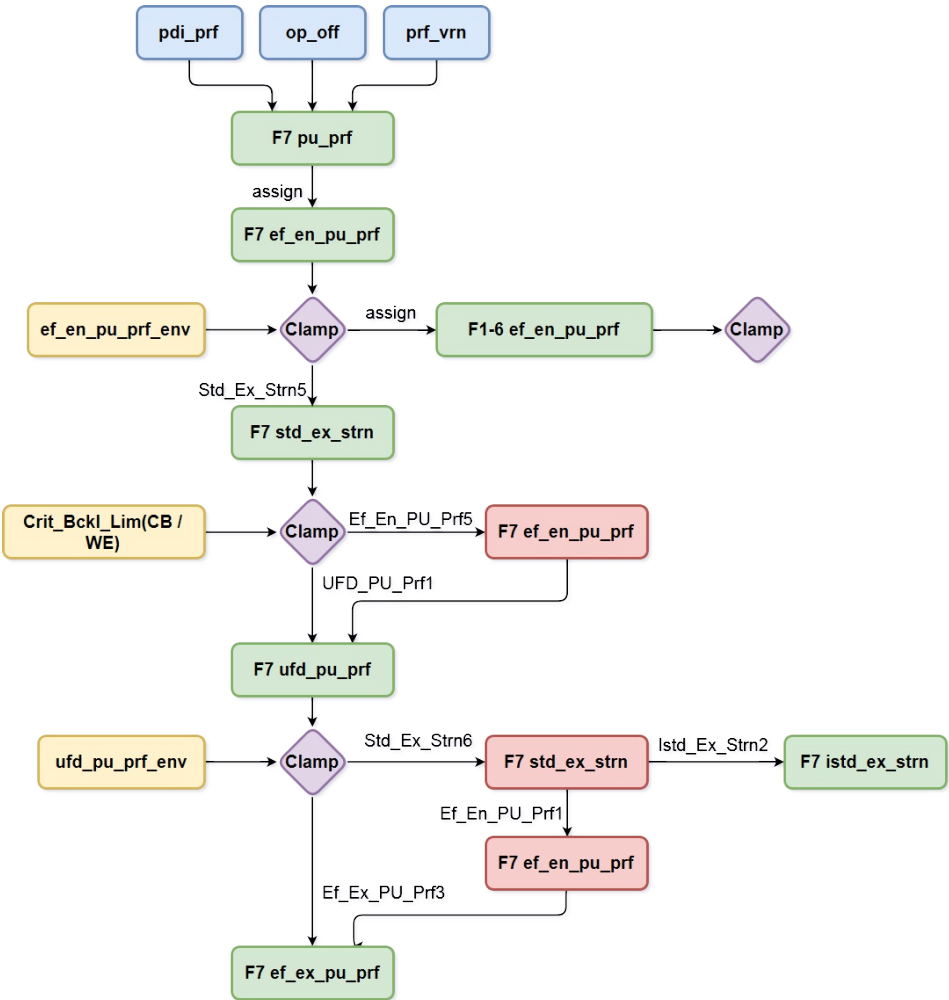
或者表示为：

$$pu_prf_change_sum = \sum_{i=1}^7 \left(\frac{strn_rlf_cof}{pce_infl_cof} \cdot Q_{elas_modu} \right)$$

Delvry_Pass(..)

Delvry_Pass(..)计算F7或最后一非空过道次的入口和出口有效单位凸度，以及出口istd应变差。

计算流程如下图所示。



基础量是加了操作工补偿和凸度自学习的单位凸度pu_prf。由pu_prf_env限幅。

首先将pu_prf直接赋值给最后一道次（F7）入口有效单位凸度ef_en_pu_prf。

之后将F7的ef_en_pu_prf直接赋值给F1到F6的入口有效单位凸度ef_en_pu_prf，并由各个道次的ef_pu_prf_env限幅。这是先假设所有机架单位凸度相同，理想状态下的情况。

回到F7，用Std_Ex_strn5计算末道次的机架出口应变差std_ex_strn，并用中浪和边浪的判别极限Crit_Bckl_Lim限幅。在判别极限中，中浪对应负值，边浪对应正值。限幅操作的意义是看std_ex_strn是否超出判别极限，若超出则一定会出现浪形，则当前std_ex_strn的值肯定不合适，需要重新计算，但是std_ex_strn的值依赖ef_en_pu_prf。因此，通过函数Ef_En_PU_Pr5，利用限幅后的std_ex_strn和目标pu_prf重计算F7的ef_en_pu_prf。

接着利用新的std_ex_strn计算F7的ufd_pu_prf_buf。同样在限幅时，若发现当前的ufd_pu_prf_buf不合适，需要重新计算。但是ufd_pu_prf的计算依赖于之前的std_ex_strn、ef_en_pu_prf计算结果，因此必须对这两个值重新计算。

最后，我们有重新计算的std_ex_strn、ef_en_pu_prf值。即可通过新的std_ex_strn值，计算出F7的出口istd应变差。利用新的ef_en_pu_prf和ufd_pu_prf_buf计算出F7的出口有效单位凸度ef_ex_pu_prf。

注意在Delvry_Pass(..)中的均载辊缝凸度只是作为中间计算结果存在，与后面分配阶段的ufd_pu_prf有所区别。

凸度分配计算的大循环

前期准备工作做完后进入凸度分配计算的大循环。

局部指针的引用

在每个循环体开始执行时，先用局部指针指向本次循环要用到的所有相关动态对象。

```
// create pointers to class objects that are part of
this pass
pcStdD      = pcFSPassD->pcFSStdD[ iter ];
pcCRLCD     = pcStdD->pcCRLCD;
pcAlcD      = pcFSPassD->pcAlcD;
pcUFDD      = pcStdD->pcUFDD;
pcLRGD      = pcStdD->pcLRGD;
pcLPceD     = pcFSPassD->pcLPceD;
pcPEnvD     = pcFSPassD->pcPEnvD;
pcEnPceD    = pcStdD->pcEnPceD;
pcExPceD    = pcStdD->pcExPceD;
// create a pointer to the previous active pass
pcPrvAct    = pcFSPassD->pcPrvAct;
```

目的是为提高性能。

更新综合辊缝凸度

凸度方面，模型首先更新综合辊缝凸度，保证带钢-工作辊凸度 pce_wr_crn 和工作辊-支承辊凸度 wr_br_crn 是当前状态下的最新值。

```
//-----
//-----
// Calculate the following composite roll stack crown
quantities:
//      Piece to work roll stack crown
//      Work roll to backup roll stack crown
//-----
```

```

-----
line_num = __LINE__;
pcCRLCD->Crns ( pcStdD->wr_shft,
                pcStdD->angl_pc,
                pce_wr_crn,
                wr_br_crn );

```

pcCRLCD->Crns(..)的计算详见CRLC模块说明。

空过的分配处理

分配从F7或末道次机架，从后往前倒者来。

若非末道次机架中，若本道次空过，则传递本道次的出口厚度给上游机架，也就是空过的机架前后带钢厚度不变。并且设定本道次均载辊缝凸度ufd_pu_prf为0。

```

        if( pcStdD->dummied )
        {
            //-----
            // 保存带钢的出口厚度给入口厚度。
            //-----
            ( ( cFSPassD* )pcFSPassD->previous_o
bj )->pcAlcD->thick = pcAlcD->thick;
            pcAlcD->ufd_pu_prf = 0.0;
        }

```

非空过部分的计算持续到start_over之前。

咬入计算与单位轧制力约束

之后进行带钢的咬入计算，咬入计算的输入量有入口宽度、出入口厚度、出入口张力、轧制速度，计算输出量有单位轧制力、前滑值和接触弧长度。

```
pcAlcD->pcRollbite->Calculate(      //@S014
    &rbStatus,
    // OUT status from calculations
    &force_pu_wid_buf,
    // OUT rolling force/width
    &fwd_slip,
    // OUT exit slip ratio [-]
    &arcon,
    // OUT length of arc [minor_length]
    ( ( cFSPassD* )pcFSPassD->previo
us_obj )->pcAlcD->thick, // IN  entry_thk
    pcAlcD->thick,
    // IN  exit_thk
    pcEnPceD->width,
    // IN  exit/entry width
    pcStdD->speed,
    // IN  roll peripheral speed
    pcEnPceD->tension,
    // IN Entry tension
    pcExPceD->tension )
    // IN Exit tension
```

为什么把咬入计算放在这里，是因为后面有重分配压下的打算，即redrft_perm为true时，需约束单位轧制力。

```
if( redrft_perm )
{
```

```

        pcAlcD->force_pu_wid = (float) force
        _pu_wid_buf;

        //-----
        // Restrict the rolling force per unit
        it piece width to within
        // the rolling force per unit piece
        width envelope.
        //-----
        line_num = __LINE__;
        cAlcD::Eval_Frc_PU_Wid( force_pu_wid
        _clp,
        pcAlcD->forc
        e_pu_wid,
        pcStdD->forc
        e_strip / pcStdD->pcEnPceD->width,
        pcPEnvD->for
        ce_pu_wid_env,
        pcAlcD->pcRo
        llbite->Precision() );
    }
    pcAlcD->flt_ok = true;

```

约束完单位轧制力后，设定一个标识浪形是否合格的指示器：
pcAlcD->flt_ok，其默认值为true。

均载辊缝单位凸度的计算

最重要的计算到来了，均载辊缝单位凸度的计算。

注意这里有两个均载辊缝单位凸度，一个是pcLPceD->ufd_pu_prf，另一个是pcAlcD->ufd_pu_prf，这两个ufd_pu_prf是相对的，因为要比较它们之间的偏差。

如果下游机架的带钢影响系数为0，则只计算实际的pcAlcD->ufd_pu_prf，不更新pcLPceD->ufd_pu_prf。如果下游机架的带钢影响系数不为0，则计算目标均载辊缝单位凸度pcAlcD->ufd_pu_prf，接着计算弯窜辊，最后用18项线性方程更新pcLPceD->ufd_pu_prf。

计算弯窜辊过程中可以选择执行机构的计算先后顺序，目前是先计算窜辊，再计算弯辊。执行机构的计算顺序保存在actr_prior中。cAlc::actrtyp_shift的条件则进行窜辊位置的计算，cAlc::actrtyp_bend的条件则进行弯辊力的计算。注意在优先级别actr_prior中cAlc::actrtyp_none，指的是无执行机构执行计算，表示预设位。

在窜辊计算中，首先根据目标均载辊缝单位凸度pcAlcD->ufd_pu_prf利用18项线性方程反算综合辊缝凸度：带钢-工作辊凸度pce_wr_crn和工作辊-支承辊凸度wr_br_crn。之后pce_wr_crn代入pcCRLCD->Shft_Pos(..)计算窜辊位置，最后利用pcCRLCD->Crns(..)和新计算的窜弯辊值更新综合辊缝凸度。

在窜辊计算中，用pcUFDD->Bnd_Frc(..)反算弯辊力，注意输出量force_bnd为force_bnd_des限幅后的结果。

设定一个表示目标均载辊缝单位凸度和实际均载辊缝单位凸度偏差的指示器。若偏差大于ufd_pu_prf_tol（目前为0.0001）则设定为true，表示均载辊缝单位凸度偏差超出了容许的范围，引出了后面有关alc_lim缩小偏差的一系列计算。

```
alc_lim = fabs( pcAlcD->ufd_pu_prf - pcLPceD->ufd_pu
```

```
_prf ) > pcAlcD->pcAlc->ufd_pu_prf_tol;
```

大循环中redrft_perm相关计算

重算单位轧制力以及更新相关动态参数。由于redrft_perm为false，这里的计算内容忽略。

alc_lim相关计算

首先计算个局部变量的ufd_pu_prf，用pcUFDD->Prf(..)计算，注意若轧制力不允许重新分配，那么这个局部的ufd_pu_prf和pcLPceD->ufd_pu_prf（目标）是一样的。也就是说，若轧制力重新分配，需要使用新的弯辊力和综合凸度去更新ufd_pu_prf。

接着利用pcLRGD->Ef_En_PU_Pr3(..)计算新的入口有效单位凸度ef_en_pu_prf_buf。此时旧的入口有效单位凸度为ef_en_pu_prf（old）。

上一道次的有效单位凸度包络线当然可以约束ef_en_pu_prf_buf。但是这种约束并不准确，因为厚度可能会变，因此约束标准应当有所放宽。所以在程序中用出口有效单位凸度ef_ex_pu_prf来约束。约束后新的入口有效单位凸度设为ef_en_pu_prf。

用新的ef_en_pu_prf求出std_ex_strn和ef_ex_pu_prf。至此，在这个阶段我们获得了可能合适的出入口有效单位凸度。但是别急，还需要判断出口的浪形，才能决定我们目前的分配凸度是否合适。

分配模型中，如果进入了alc_lim的计算，在重计算出入口有效单位凸度后必须进行浪形判别。F1和F6本道次的浪形判别，由本道次的应变差和下一道次应变差是否超死区极限决定。F7道次的浪

形判别，仅由本道次F7的应变差是否超限决定。

当浪形判别不通过，或者说flt_ok为假时，可以稍微放宽一点标准。如果非末道次机架的下道次应变差不超死区极限，那么也算本道次浪形判别通过。

```
if ( pcFSPassD != pcLstActFSPassD )
{
    pcAlcD->flt_ok =
        ( std_ex_strn_dn <= bckl_lim_dn[ we ] ) &&
        ( std_ex_strn_dn >= bckl_lim_dn[ cb ] );
}
```

到现在这一步，如果浪形判别还不能通过，那么需要重新设定目标单位凸度。先重新设定目标有效单位凸度，用pcTargD->Pass_Mill_Targ(..)计算获得，变量为ef_pu_prf_alt。再利用F7的istd_ex_strn反推std_ex_strn，结合ef_pu_prf_alt计算出目标单位凸度pu_prf。注意在alc模块中，大循环的这个位置是alc局部pu_prf变量第一次介入的地方，局部pu_prf变量预设为0。之后对pu_prf进行限幅，这样新的目标单位凸度就诞生了。

接着是一个难点问题。

```
pcCritFSPassD = pcFSPassD;
```

将更新了目标单位凸度的道次地址赋值给pcCritFSPassD指针。这个pcCritFSPassD最开始是指向F7道次的。pcCritFSPassD指针设定的意义在于：在迭代计算的过程中，浪形判别不合格的相应道次必须比之前更新过目标单位凸度的道次低。

在alc_lim计算过程的最后，若目标单位凸度发生改变，则设定start_over指示器为true，以进行后续start_over的流程。

非空过道次的更新

在alc_lim计算之后，更新给定条件下的ufd_pu_prf、ef_pu_prf、strn、prf，注意这些值都属于lpce对象。

start_over流程

如果目标单位凸度发生改变，则更新目标单位凸度的迭代次数（累积加一）。之后从F7重新开始大循环的计算，从Delivery_Pass(..)起步重算出入口有效单位凸度，并设定F7的有效单位凸度为出口有效凸度。

如果目标单位凸度没有发生改变，说明本道次的浪形是符合判别条件的，不需要更改目标；或者目标均载辊缝凸度达到了实际的均载辊缝凸度。进一步说，有两种情况会进入start_over为假的流程，一种是未进入alc_lim计算的状态，另一种是进入了alc_lim的计算，但是浪形判别合格的状态。则当前道次对象pcFSPassD可以前移一个道次。出入口有效单位凸度现在敲定是合适的。

之后是：对不均匀变形道次ef_en_pu_prf修正的过程。在此阶段，如下一段代码需要注意，在理解上可能会出错。

```
//-----  
  
// Increment pointer to previous dynamic PASS object.  
  
//-----
```

```

pcFSPassD = ( cFSPassD* )pcFSPassD->previous_obj;

//-----
// Save the effective entry per unit profile of the
previous
// pass into the effective exit per unit profile for
this pass.
//-----
ef_ex_pu_prf = ef_en_pu_prf;

//-----
// Determine the upstream effective per unit profile
to aim
// towards using the extreme downstream pass where t
he piece
// influence coefficient is zero.
//-----
ef_en_pu_prf = cMathUty::Clamp ( ef_ex_pu_prf,
    pcPceIZFSPassD->pcPEnvD->ef_pu_prf_env[ min1 ],
    pcPceIZFSPassD->pcPEnvD->ef_pu_prf_env[ max1 ] );

```

这段代码执行之后，ef_ex_pu_prf这个变量的意义已经发生改变，不再代表本道次的出口有效单位凸度。因为前面我们的pcFSPassD道次对象已经前移一个道次，因此程序的设计者为了简练，直接使用局部变量ef_en_pu_prf代表其它含义。

如代码所示，ef_ex_pu_prf保存的是原ef_en_pu_prf的值，而新的ef_en_pu_prf是受到pcPceIZFSPassD道次ef_pu_prf_env包络线限幅之后的值。

没经过浪形判别或经过浪形判别但没有改变目标单位凸度的过程参数ef_en_pu_prf，必须从出现不均匀变形的机架开始，用每个机架的单位凸度最大改变量约束和修正本道次的ef_en_pu_prf。

ef_en_pu_prf修正过程是个循环，从pcPceIZFSPassD道次的下一道次开始，且当前道次的上一道次在不均匀延伸的机架中，直至末道次。

ef_pu_prf_chg[cb/we]并不能直接作为真正的有效单位凸度最大改变量，或真正有效凸度改变约束条件。在不均匀变形的机架中，它需要本道次的ef_pu_prf_env和上一道次的ef_pu_prf_env介入，来获得一个更窄的变化区间ef_pu_prf_dlt[minl/maxl]，用所有存在不均匀变形机架的这个区间来修正本道次的ef_en_pu_prf。

```
//-----  
----  
// Calculate the delta effective per unit profile change  
// from stand entry to interstand exit.  
//-----  
----  
  
ef_pu_prf_dlt[ minl ] =  
    cMathUty::Max( pcBufFSPassD->pcFSStdD[ iter]->pc  
LRGD->ef_pu_prf_chg[ cb ],  
        cMathUty::Max( ef_ex_pu_prf,  
                        pcBufFSPassD->pcPEnvD->ef_pu  
_prf_env[ minl ] ) -  
        cMathUty::Min( ef_en_pu_prf,
```



```

((cFSPassD*)pcBufFSPassD->previous_obj)->pcPEnvD->ef_pu_prf_env[max1]));

ef_pu_prf_dlt[ max1 ] =
    cMathUty::Min( pcBufFSPassD->pcFSStdD[ iter]->pc
    LRGD->ef_pu_prf_chg[ we ],
        cMathUty::Min( ef_ex_pu_prf,
                        pcBufFSPassD->pcPEnvD->ef_pu
                        _prf_env[ max1 ] ) -
        cMathUty::Max( ef_en_pu_prf,
                        ((cFSPassD*)pcBufFSPassD->pr
                        evious_obj)->pcPEnvD->ef_pu_prf_env[min1]));

```

ef_pu_prf_dlt的计算，说白了就是用本道次机架的有效凸度减前一道次机架的有效凸度，只不过将本道次机架和前道次机架的包络线和ef_ex_pu_prf、ef_en_pu_prf联系起来，用于收窄死区。

ef_pu_prf_chg依据于理论计算，第二项的差值依据于本道次和上道次的包络线，若中浪则取最大的，若边浪则取最小的。

先求有效凸度改变的总量。从出现不均匀变形的机架

（pcPceIZFSPassD道次的下一道次）开始，如果pcPceIZFSPassD道次包络线限幅后的ef_en_pu_prf小于原值，说明pcPceIZFSPassD道次的包络线区间整体小于原ef_en_pu_prf，那么ef_pu_prf_sum累加ef_pu_prf_dlt[max1]。反之，若pcPceIZFSPassD道次包络线限幅后的ef_en_pu_prf大于原值，说明pcPceIZFSPassD道次的包络线区间整体小于原ef_en_pu_prf，那么ef_pu_prf_sum累加ef_pu_prf_dlt[min1]。

注意pcFSPassD是本道次的前一道次，而在修正ef_en_pu_prf的循环当中，pcBufFSPassD指向本道次的前一道次时，循环结束。

这时ef_en_pu_prf变量的含义又发生了改变，变回了字面意思，即本道次的入口有效单位凸度。修正就是从原始初设定的入口有效单位凸度加上（或减去）与累加有效凸度改变总量成比例的一部分ef_pu_prf_dlt，作为新的入口有效单位凸度存在。

```
// 以边浪情形为例
ef_en_pu_prf =
    ef_ex_pu_prf - ef_pu_prf_dlt[ maxl ] *
    ( ef_ex_pu_prf - ef_en_pu_prf ) /
    ef_pu_prf_sum;
```

修正结束后，考虑ef_en_pu_prf和ef_ex_pu_prf偏差太大的情况，则计算ef_en_pu_prf_dft作为最终的ef_en_pu_prf。

```
float ef_en_pu_prf_dft = ef_ex_pu_prf + (pcTargtD->e
n_pu_prf - pcTargtD->pu_prf)
    * pcFSPassD->pcAlcD->pu_prf_change;
```

最后用pcTargtD->Eval_Ef_En_PU_Prfl(..) 评估一下ef_en_pu_prf和ef_ex_pu_prf。

后续其它计算

若轧制力不重新分配，则预设所有机架force_ssu为0。

cShapeFeedbackD::Main(..)

注意设定弯辊力不是单力。

```
sup_bend = (float) num_chocks * pcSched->pcFSSched->  
pcSPassRef[ps-1]->state.force_bnd_org;
```

cShapeFeedbackD::Pass(..)

cShapeFeedbackD::Pass(..)主要分为以下三个部分：

- cShapeFeedbackD::Init(..)
- cShapeFeedbackD::Evaluate(..)
- cShapeFeedbackD::OprBnd(..)

cShapeFeedbackD::Init(..)

cShapeFeedbackD::Init(..)函数位于shapefeedback_req.cxx文件中，并于cShapeFeedbackD::Pass(..)函数中执行。

inhb_t_w_calc为指示计算辊系热胀和磨损计算的布尔值，默认为false。

用pcRollbite->Calculate_DForce_DEnthick(..)和pcRollbite->Calculate_DForce_DExthick(..)计算轧制力对出入口厚度的偏导数。

初始化cUFDD实例的动态参数，包括ufd_modifier。

inhib_t_w_calc 由于为false，因此初步设定 pce_wr_t_w_crn 和 wr_br_t_w_crn 为0.0。

f_wr_crn_off_adj从SPRP表中读取数据，之后用cCRLCD::Init(..)初始化CRLCD动态对象。注意有这么一个标识点wr_crn_off_sel_flag，用来指示初始化时psSLFG->wr_crn_off是否需要加上f_wr_crn_off_adj，一般这个值wr_crn_off_sel_flag默认为true。

```
if ( pcTargetD->pcTarget->wr_crn_off_sel_flag )
{
    pcFSPassD->pcFSStdD[ seg ]->pcCRLCD->Init(
        inhib_t_w_calc,
        pce_wr_t_w_crn,
        wr_br_t_w_crn,
        psSAMP->wr_crn_vrn[ pass_idx ],
        //@@@2ND-2(MAC014) begin
        //( psSLFG->wr_crn_off[ pass_idx ]
        + psSPRP->wr_crn_off_adj[ pass_idx ]),
        ( psSLFG->wr_crn_off[ pass_idx ] +
        f_wr_crn_off_adj ),
        //@@@2ND-2(MAC014) end
        pcFSPassD->pcFSStdD[ seg ]->pcEnPc
        eD->width,
        pcFSPassD->pcFSPass->wr_crn_vrn_i_
        gn,
        pcFSPassD->pcFSPass->wr_crn_off_i_
        gn,
        pcFSPassD->pcFSPass->wr_crn_cor_i_
        gn,
        pcFSPassD->pcFSStdD[ seg ]->pcStdR
        ollPrD
        //psPDI->grt_idx
    );
}
```

```

else
{
//@2ND(LC060) end
    pcFSPassD->pcFSStdD[ seg ]->pcCRLCD->Init(
        inhb_t_w_calc,
        pce_wr_t_w_crn,
        wr_br_t_w_crn,
        psSAMP->wr_crn_vrn[ pass_idx ],
        psSLFG->wr_crn_off[ pass_idx ],
        pcFSPassD->pcFSStdD[ seg ]->pcEnPc
eD->width,
        pcFSPassD->pcFSPass->wr_crn_vrn_i_
gn,
        pcFSPassD->pcFSPass->wr_crn_off_i_
gn,
        pcFSPassD->pcFSPass->wr_crn_cor_i_
gn,
        pcFSPassD->pcFSStdD[ seg ]->pcStdR
ollPrD
        //psPDI->grt_idx
    );

```

之后用cLRGD::Init(..)初始化LRG动态对象。

cShapeFeedbackD::Evaluate(..)

评估。

cShapeFeedbackD::OprBnd(..)

cShapeFeedbackD::OprBnd(..)是对操作工弯辊力调整的自适应，自学习值从。内部主要调用的函数为cCRLCD::Opr_Bnd_Frc_Adpt(..)。

操作工对工作辊弯辊力的补偿wr_bnd_off先转化成等效的辊系凸度修正值stk_bnd_err。将 pcCRLCD->wr_cr_vrn赋值给stk_vrn_bnd。以stk_bnd_err和stk_vrn_bnd为实参，代入cCRLCD::Opr_Bnd_Frc_Adpt(..)计算。自学习计算采用积分控制，积分系数为wr_crn_cor_i_gn。wr_crn_cor_i_gn的配置在配置文件cfg_fpass.txt当中按机架划分。

cShapeFeedbackD::Mill(..)

cShapeFeedbackD::Mill(..)函数的作用是根据实际测量的带钢凸度和平直度，计算辊系凸度自学习量。

逆序循环

从profile/flatness sensor中获得凸度和平直度的测量值meas_prf和meas_flt。

meas_flt直接作为出口应变差std_ex_strn，如果检测有误，则直接用std_ex_strn的计算值。

用出口检测到的meas_prf和std_ex_strn，代入pcLRGD->Ef_Ex_PU_Prfl(..)，重新计算ef_ex_pu_prf。

利用以上现有参数，结合pcSFBObsD[seg]->RepMill(..)重计算有效入口单位凸度ef_en_pu_prf。

LPCE模块重难点问题

LPCE模块可以说是SSU中最简单的模块了。

横向辊缝模型

横向辊缝模型的初始化中，主要做了三件事。

- 计算带钢影响系数
- 计算有效单位凸度的最大改变量（ w_e / c_b ）
- 计算凸度改变削弱因子

带钢影响系数

带钢影响系数用宽厚比插值获得，这个参数的用途是用来表示入口单位凸度对出口单位凸度的敏感程度。取值在0和1之间。

如果这个参数为0，说明带钢在该道次的变形只有均匀的横向压下，不会导致带钢宽度方向的延伸不均。同样，如果这个参数为1，说明带钢在该道次的变形导致完全的延伸不均，带钢在横向没有金属流动。

有效单位凸度的最大改变量

这个参数虽然依据于理论计算，但是我们有了mult调节参数后，这个参数实际就成了我们调整真正的有效单位凸度改变量的有效手段。

在板形设定凸度分配过程中，对于UFD目标值和计算值没偏差的有效单位凸度，以及UFD目标值和计算值存在偏差但通过了浪形判别的有效单位凸度，在修正过程中我们可以调整mult的值来调

整有效入口单位凸度的修正量。

凸度改变削弱因子

凸度改变削弱因子计算的输入量，除了厚度和宽度，还需要考虑单位轧制压力、单位轧制压力对出口厚度的偏导数、流变应力、弹性模量和接触弧长度。

```
Prf_Chng_Attn_Fac( pce_wid,  
                    en_pce_thck,  
                    ex_pce_thck,  
                    frc_pu_wid,  
                    dfrcw_dexthck,  
                    pce_flw_strs,  
                    pcLPceD->elas_modu,  
                    arc_of_contact );
```

流变应力

流变应力的单位是MPa。流变应力的维基解释如下： Flow stress is defined as the instantaneous value of stress required to continue plastically deforming the material – to keep the metal flowing. It is the middle value between yield strength and ultimate strength of the metal as a function of strain, which can be expressed:[1] $Y_f = K \epsilon^n$ [1] Y_f = Flow stress, MPa ϵ = True strain K = Strength Coefficient, MPa n = Strain hardening exponent Hence, Flow stress can also be defined as the stress required to sustain plastic deformation at a particular strain.

The flow stress is a function of plastic strain.

The following properties have an effect on flow stress: chemical composition, purity, crystal structure, phase constitution, exit microstructure, grain size, and heat treatment.

The flow stress is an important parameter in the fatigue failure of ductile materials. Fatigue failure is caused by crack propagation in materials under a varying load, typically a cyclically varying load. The rate of crack propagation is inversely proportional to the flow stress of the material.

CRLC模块重难点问题

cCRLCD::Init(..)

在初始化过程中，传入的宽度参数是各个机架入口的宽度，不是出口宽度，这个主要用来求半宽以及有载CVC辊形等效凸度。

cCRLCD::Shft_Pos(..)

CVC标签

CVC标签的作用是用来选择合适的插值向量。插值向量的配置在cfg_fcrlc.txt中。

在cCRLCD::Shft_Pos(..)中，需要注意CVC标签的问题。

```
switch ( rprof )
{
    case rp_cvc1 :
    case rp_cvc2 :
    case rp_cvc3 :
    case rp_cvc4 :
        {
            ...
        }
}
```

不管你rprof选择什么样的辊形，不管是CVC1还是CVC2还是CVC3，最终都要执行case rp_cvc4之后对应的语句。因为前三个case后面没有break。

但是，现在的模型当中，在窜辊计算时，不再用插值的方式计算窜辊位置或窜辊的等效凸度，而是通过和带钢入口宽度以及a1、a2、a3系数相关的计算获得窜辊的等效凸度。因此CVC的标签不管是CVC1还是CVC4，目前已经没有作用。

dlt是否需要除以2

在开始迭代之前，wr_grn_cr_req需要加上一个pce_wr_cr_dlt变化量，这个变化量在1580产线是除以2的，而在2250产线不除以2。这样设计有什么区别？

```
wr_grn_cr_req = wr_grn_cr + pce_wr_cr_dlt / 2
// wr_grn_cr_req = wr_grn_cr + pce_wr_cr_dlt
```

通过模拟发现，不进行减半操作，原窜辊位置和新窜辊位置之间的关系，更为平缓。

ALC模块重难点问题

分配过程与分配模块内的重点和难点问题的说明。

pcPceIZFSPassD

pcPceIZFSPassD是板形模型中用来标识带钢影响系数为0的道次的指针。

```
pcPceIZFSPassD = ( cFSPassD* )pcFstFSPassD->previous_obj;  
  
const cFSPassD* pcFSPassD = pcFstFSPassD;  
  
while ( pcFSPassD != NULL )  
{  
    // ...  
    if ( pcFSPassD->pcFSStdD[ iter ]->pcLRGD->pce_in  
f1_cof <= 0.0 )  
    {  
        pcPceIZFSPassD = pcFSPassD;  
    }  
    // ...  
    pcFSPassD = ( cFSPassD* )pcFSPassD->next_obj;  
}
```

首先预设pcPceIZFSPassD为pass0。之后一个循环，从F1到F7找到带钢影响系数为非正数的道次，用pcPceIZFSPassD指向这个道次。

一般说来，带钢影响系数从F1到F7是逐渐增大的，意味着从F1到F7不均匀变形是逐渐增大逐渐严重的过程（宽厚比增加）。所以设置pcPceIZFSPassD的意义在于：F1到F7若上游机架存在均匀变形，则找到均匀变形的上游机架和不均匀变形的下游机架之间的分界点。那么在 `start_over == true` 的流程中，`ef_en_pu_prf` 还要受到pcPceIZFSPassD道次的包络线约束。

凸度最终修正的说明

为化简情形，设当前板形满足以下条件：

- 各个机架或道次的带钢影响系数均不为0。
- 各个机架的包络线区间富余量足够大，能够包含死区极限。

那么整个凸度分配的

`cTargtD::Ef_PU_Prfl_Aim(..)`

`ef_pu_prf_buf = pu_prf - istd_ex_strn;`这一步的意义是什么

SSU日志验算

日志验算和说明。

开头

Int H w和Fin H w

--- Profile ---这一栏当中的“H w”指的是加了凸度自学习量的目标凸度。Int H w为初始的prf，Int H w为最终的prf。Int H w的计算过程如下所示。

```
prf_int = (pdi_prf + prf_op_off) * matl_exp_cof + prf_vrn;
```

热膨胀系数可以忽略不计，那么Int H w就是PDI的凸度、操作工补偿的凸度、凸度自学习的和。

```

|----- Profile -----
PDI [mm]: 0.0200 Int H w/ [mm]: 0.0602
Vrn [mm]: 0.0400 Fin H w/ [mm]: 0.0602
Vrn RM [mm]: 0.0150 Del C wov [mm]: 0.0200
Vrn RS [mm]: 0.0000 PU [mm/mm]: 0.0238
Tol LO [mm]: 0.0000 Ef PU [mm/mm]: 0.0000
Tol HI [mm]: 0.0000 Achv [-]: T
Op Ofs [mm]: 0.0000 Dev Lim [mm]: 0.0100
Raw Vrn RM [mm]: 0.0250
New Vrn RM [mm]: 0.0250

```

在板形模型目标初始化阶段，prf_vrn会被赋值为prf_vrn_rm和prf_vrn_rs的差（prf_vrn_rs常年为零）。从实际数据来看，prf_vrn_rm和prf_vrn_rs的差，与Vrn还是存在差距的，说明这里的Vrn在实际计算中还会出现变化。（可能存在板形问题？）

中间坯凸度插值计算

中间坯的凸度，在模型计算中为插值计算。

辊系凸度

工作辊辊系凸度pce_wr_cr的验算

工作辊辊系凸度pce_wr_cr在日志中的标识为Pce WR Gap，如下图

所示。

Std #						Composite Roll Stack Crowns						
	Pce WR Gap [mm]	WR BR Gap [mm]	Pce WR Wear [mm]	BR Wear [mm]	WR BR Wear [mm]	Pce WR Thrmm [mm]	WR BR Thrmm [mm]	WR Grnd [mm]	BR Grnd [mm]	WR Eqv [mm]	WR Vern [mm]	WR Offs [mm]
1	-0.4531	-0.0802	0.0000	-0.0000	0.0000	0.0000	0.0000	-0.1534	0.2398	0.0000	-0.0200	-0.2796
2	-0.3415	-0.0014	0.0000	-0.0000	0.0000	0.0000	0.0000	-0.2020	0.2398	0.0000	-0.0200	-0.1195
3	-0.2348	0.0740	0.0000	-0.0000	0.0000	0.0000	0.0000	-0.3256	0.2398	0.0000	-0.0200	0.1108
4	-0.1446	0.1377	0.0000	-0.0000	0.0000	0.0000	0.0000	-0.3053	0.2398	0.0000	-0.0200	0.1806
5	-0.0143	0.2297	0.0000	-0.0000	0.0000	0.0000	0.0000	-0.1752	0.2398	0.0000	-0.0200	0.1809
6	-0.3262	-0.2304	0.0000	-0.0000	0.0000	0.0000	0.0000	-0.3600	0.0000	0.0000	-0.0300	0.0638
7	-0.3309	-0.2337	0.0000	-0.0000	0.0000	0.0000	0.0000	-0.3600	0.0000	0.0000	-0.0300	0.0591

pce_wr_cr由五部分组成，分别是

- 带钢-工作辊磨损pce_wr_w_cr
- 带钢-工作辊热胀pce_wr_t_cr
- 工作辊辊形等效凸度wr_grn_cr
- 工作辊凸度自学习wr_cr_vrn
- 工作辊凸度补偿wr_cr_off

把这五部分加起来就是pce_wr_cr的值。

工作辊凸度补偿的验算

用长期遗传的工作辊凸度补偿cSLFG_wr_crn_off（对应日志中的SLFG WR Offs）和SPRP中的工作辊凸度补偿系数cSPRP_wr_crn_off_adj相加，即可获得工作辊凸度补偿wr_cr_off。

WR Vern [mm]	WR Offs [mm]	Dlt Pce WR Gap [mm]	Dlt WR BR Gap [mm]	SLFG WR Offs	SPRP WR Offs
-0.0200	-0.2796	0.0000	0.0000	0.1204	-0.4000
-0.0200	-0.1195	0.0000	0.0000	0.1305	-0.2500
-0.0200	0.1108	0.0000	0.0000	0.1108	0.0000
-0.0200	0.1806	0.0000	0.0000	0.0806	0.1000
-0.0200	0.1809	0.0000	0.0000	0.0809	0.1000
-0.0300	0.0638	0.0000	0.0000	0.0638	0.0000
-0.0300	0.0591	0.0000	0.0000	0.0591	0.0000

中间坯有效单位凸度

中间坯不存在应变差的概念，中间坯的单位凸度就是中间坯的有效单位凸度。

--- Transfer Bar ---中有每卷带钢的中间坯凸度Prof，由模型插值计算获得；以及中间坯的厚度，R2出口测量获得。可以通过凸度求厚度的商，作为中间坯的有效单位凸度，其值与--- Allocation Results (1st Iter.) ---这一栏中的零道次pass0的EF PU Prf对应。中间坯凸度值与--- Allocation Results (1st Iter.) ---这一栏中的零道次pass0的Prf值对应。

Allocation Results (1st Iter.)					
#	Des Bnd Frc [kN]	Ef PU Prf [mm/mm]	Prf [mm]	UFD PU Prf [mm/mm]	Std Ex Strn [mm/mm]
0		0.0063	0.2865		
1	815	0.0214	0.4122	0.0239	0.00017
2	890	0.0234	0.2100	0.0268	0.00014
3	890	0.0236	0.1417	0.0272	0.00016
4	890	0.0238	0.1026	0.0287	0.00018
5	840	0.0238	0.0814	0.0305	0.00021
6	315	0.0239	0.0686	0.0297	0.00018
7	320	0.0239	0.0602	0.0212	-0.00009

单位宽度轧制力的验算

单位宽度轧制力，直接拿轧制力除以带钢精轧宽度，由于受到限幅，一般单位轧制力的值是真正被限幅后的值。注意这里的精轧宽度指的是中间坯在F1入口的宽度，不是订单宽度也不是目标宽度。

所以，单位轧制力包络线的最大值、单位轧制力包络线的最小值，以及-- Allocation Requirements (1st Iter.) --这一栏中的单位轧制力分配结果一般三者在各自机架内都相等。

带钢弹性模量的计算误差

手动验算的时候发现elas_mod的计算结果总是与SSU日志中的记录结果存在一定的偏差，偏差精度在100以内。

产生这样偏差的原因是因为插值。弹性模量的插值通过每个机架的温度完成。插值表如下表所示。

avg_pce_tmp_interp_vec	elas_modu_interp_vec
600	138269
650	128069
700	117905
750	107751
800	97589
850	87415
900	77232
950	67054
1000	56909
1050	46829
1100	36863
1500	27067

带钢弹性模量的插值向量从27067到138269MPa，与温度的插值相比已经不在一个数量级。温度一单位改变会造成弹性模量更大的变动，因此出现100MPa以内的波动很正常。

Adaptation Results

在Adaptation Results这一栏中，注意开头的几个参数。

-----from prf scan----- Adaptation Results											
Bnd				Pce Wr				WR			
CrnErr ok CrnVrn				GapErr ok CrnVrn ok CrnOff ok				Prf			
#	[mm]		[mm]	[mm]		[mm]		[mm]		[mm]	
1	0.0000	T	-0.0200	-0.3765	T	-0.0200	T	-0.4411	T	0.0000	F
2	0.0000	T	-0.0200	-0.2250	T	-0.0200	T	-0.2733	T	0.0000	F
3	0.0000	T	-0.0200	-0.1377	T	-0.0200	T	-0.0152	T	0.0000	F
4	0.0000	T	-0.0200	-0.0876	T	-0.0200	T	0.0897	T	0.0000	F
5	0.0099	T	-0.0200	-0.0809	T	-0.0200	T	-0.0092	T	0.0000	F
6	0.0000	T	-0.0300	-0.0611	T	-0.0300	T	-0.0068	T	0.0000	F
7	-0.0033	T	-0.0300	-0.1539	T	-0.0300	T	0.0144	T	0.0000	T

- Bnd Crn Err对应代码中的stk_bnd_err。
- Bnd Crn Vrn对应代码中的stk_vrn_bnd。（本质就是wr_cr_vrn）
- Pce WrR Gap Err对应代码中的stk_rep_err。
- WR Crn Vrn对应代码中的stk_crn_vrn。
- WR Crn Off对应代码中的stk_crn_off

这些参数均属于pcSFBObsD指向的对象。

SSU日志 FAQ

Profile一栏中Vrn RM和Vrn RS是什么？

Profile一栏中Vrn RM和Vrn RS一般情况下指的是长期自学习值。

pcTargetD->pcTarget->prf_vrn_sel_flag默认值为true，在cfg_target.txt文件中设定，若此值为false，则Profile一栏中Vrn RM和Vrn RS为短期自学习的psSAMP->prf_vrn_rm和psSAMP->prf_vrn_rs。

为什么弯辊力包络线最大值和最小值相反？

标签max和min指的是ufd有效单位凸度的最大值和最小值，最大的弯辊力会计算获得最小的有效单位凸度，最小的弯辊力会计算获得最大的有效单位凸度。为保持一致性，弯辊力包络线max与min对调。

wr_crn_vrn_z是什么？

算是凸度自学习的一个初始值。当换辊算不准时restore进行补偿。配置文件里面有，据说很好用。

CFG模型参数整理

C++代码相关的cfg参数整理。

板形的cfg参数可以分为两大类，一类是SSU模块的本身的配置参数，一类的是与FSU共享的配置参数。

与板形计算相关的SSU核心配置参数文件如下。

```
include = SSU$CONFIG:cfg_alc.txt;           ! create
and configure static ALC object
include = MDS$CONFIG:cfg_fcrlc.txt;         ! create
and configure CRLC object
include = MDS$CONFIG:cfg_flpce.txt;         ! create
and configure LPCE object
include = MDS$CONFIG:cfg_flrg.txt;          ! create
and configure LRG object
include = SSU$CONFIG:cfg_fspass.txt;        ! create
and configure FSPass objects
include = SSU$CONFIG:cfg_fsstd.txt;         ! create
and configure static FSSTD object
include = MDS$CONFIG:cfg_fufd.txt;          ! create
and configure UFD object
include = SSU$CONFIG:cfg_penv.txt;          ! create
and configure static PENV object
include = SSU$CONFIG:cfg_shapesetup.txt;    ! create
and configure static SHAPESETUP object
include = SSU$CONFIG:cfg_shapefeedback.txt; ! create
and configure static SHAPEFEEDBACK object
include = SSU$CONFIG:cfg_targt.txt;         ! create
and configure static TARGT object
include = SSU$CONFIG:cfg_sfbobs.txt;        ! create
```



```
and configure static Obs object
```

FSU相关的配置参数如下。

```
include = MDS$CONFIG:cfg_fmll.txt;           !create t  
he remainder of static mill object for FM  
include = MDS$CONFIG:cfg_ftmpgrad.txt;       !create a  
nd configure a tmpgradcfg object  
include = FSHARED$CONFIG:cfg_std.txt;         !create t  
he static stand objects  
include = MDS$CONFIG:cfg_fmtr.txt;           !create t  
he static motor objects  
include = MDS$CONFIG:cfg_frollbite.txt;       !create t  
he static roll bite objects  
include = GSM$CONFIG:cfg_sensor.txt;         !create t  
he static sensor objects  
include = GSM$CONFIG:cfg_fzone.txt;          !create a  
nd configure the zone objects  
include = FSHARED$CONFIG:cfg_map.txt;        !create t  
he static map objects
```

SSU/cfg_alc

```
class = cAlc;  
cAlc = alc;  
  
//单位轧制力调节系数  
frcw_adj_mod      = 0.9;           ! [-]  
rolling force adjustment modifier  
  
//分配计算过程中迭代的最大次数  
loop_count_lim    = 10;           ! [-]
```

```

maximum number of iterations

//单位轧制力最小值
    force_pu_wid_mn = 2.0;                ! [kN
/mm] minimum rolling force per unit piece width

//目标UFD均载辊缝单位凸度和实际UFD均载辊缝单位凸度的
偏差容许范围
    ufd_pu_prf_tol = 0.0001;              ! [mm
/mm] UFD roll gap per unit profile tolerance

//是否合理的指示器
    vld = true;                            ! [-]
validity indicator

//计算窜弯辊过程中，窜辊万滚的计算顺序，一般先计算窜
辊，再计算弯辊
    actr_prior = 3,                        ! [-]
mechanical actuator priority
                                actrtyp_shift,    !
roll CVC shifting system (SMS)
                                actrtyp_bend,      !
roll bending system
                                actrtyp_none;      !
force fall through
    end;
end;

```

SSU/cfg_fcrlc

```

class = cCRLC;
    cCRLC = crlc;

```

```

//窜辊计算迭代的次数
! [-] Newton-Raphson maximum number of iterations on roll shift position
iter_mx = 15;

//CVC插值CVC等效凸度的向量，从cvc1标签到cvc4标签
! [mm] CVC work roll ground crown vector as a f(CVC profile type, CVC roll shift position)
cvc_cr_mat = 44,
             -0.860, -0.634, -0.522, -0.409, -0.296, -0.183, -0.070, 0.043, 0.156, 0.268, 0.494,
             -0.700, -0.608, -0.494, -0.379, -0.265, -0.150, -0.035, 0.079, 0.194, 0.308, 0.400,
             -0.700, -0.608, -0.494, -0.379, -0.265, -0.150, -0.035, 0.079, 0.194, 0.308, 0.400,
             -0.700, -0.608, -0.494, -0.379, -0.265, -0.150, -0.035, 0.079, 0.194, 0.308, 0.400;

//CVC插值的位置向量
! [mm] CVC roll shift position vector
cvc_shft_vec = 11,
              -150.00, -100.00, -75.00, -50.00, -25.00, 0.00, 25.00, 50.00, 75.00, 100.00, 150.00;

//带钢-工作辊系凸度的最大最小偏差
! [mm] minimum / maximum piece to work roll stack crown error
pce_wr_cr_er = 2,
              -1.1999,
              1.1999;

```

```

//窜辊迭代计算的带钢-工作辊凸度偏差最大值或容许范围
! [mm] Newton-Raphson piece to work roll stack crown tolerance
pce_wr_cr_tol = 0.01;

//每次窜辊迭代计算的窜辊位置变化
! [mm] Newton-Raphson delta roll shift position
pos_shft_dlt = 2.00;

//最小和最大辊系凸度补偿
! [mm] minimum / maximum work roll stack crown offset
wr_cr_of = 2,
          -1.20,
          1.20;

//最小和最大辊系凸度自学习量
! [mm] minimum / maximum work roll stack crown vernier
wr_cr_vr = 2,
          -1.20,
          1.20;

//CVC辊缝系数
cvc_gap_coef = 4, 0.000, 0.0000, 0.0000, 0.0000;

!!@2ND-2(MAC005) begin
//轧辊热胀和磨损的系数
pce_wr_wear_mult = 1.00;
wr_br_wear_mult = 1.00;
pce_wr_thrm_mult = 1.00;
wr_br_thrm_mult = 1.00;

```

```

//初始cvc宽度
    cvc_width_nominal= 1275.0; !! Nominal cvc width for a1 calculation[mm]
//窜辊最大极限
    cvc_Sm          = 150.0; !! Shift position limitation
//支承辊是否是cvc的指示器
    br_roll_cvc      = true;
!!@2ND-2(MAC005) end
end;
end;

```

SSU/cfg_target

```

class = cTargt;
    cTargt = targt;
// 平直度自学习当中的平直度偏差极限范围
    flt_err_lim      = 2,                ! [kN]
flatness errorlimits
                                -250.,    !
minimum
                                250.;      !
maximum

// 平直度自学习衰减系数
    flt_vrn_bled     = 0.9;              ! [-]
target flatness vernierbleed-off

// 平直度自学习极限范围
    flt_vrn_lim      = 2,                ! [kN]
target flatnessvernier limits
                                -800.0,    !

```

```

minimum
                                800.0;                                !

maximum

    // 平直度自学习PI控制系数
    flt_vrn_i_gn    = 0.6;                                ! [-]
target flatness controlloop integral gain
    flt_vrn_p_gn    = 0.3;                                ! [-]
target flatness controlloop proportional gain

    // 凸度波动极限
    prf_dev_lim     = 0.010;                                ! [mm]
target profile deviationlimit

    // 凸度自学习偏差极限
    prf_err_lim     = 2,                                    ! [mm]
profile error limits

                                -0.100,                    !
minimum
                                0.100;                        !
maximum

    // 凸度最大范围
    prf_lim         = 2,                                    ! [mm]
absolute limits

                                0.000,                        !
minimum
                                0.250;                        !
maximum

    // 凸度精度要求
    prf_tol         = 2,                                    ! [mm]
target profiletolerances

                                -0.050,                    !
minimum
                                0.050;                        !

```

```

maximum

    // 凸度自学习衰减系数
    prf_vrn_bled      = 0.9;                                ! [-]
    target profile vernier (re-predicted -setup) bleed-
off

    // 凸度自学习极限
    prf_vrn_lim       = 2,                                  ! [mm]
    target profilevernier limits
                                -0.070,                    !
minimum
                                0.070;                      !
maximum

    // 凸度自学习rm的PI控制系数
    prf_vrn_rm_i_gn = 0.4;
    ! [-] target profile vernier (re-predicted -
measured) control loop integral gain
    prf_vrn_rm_p_gn = 0.2;
    ! [-] target profile vernier (re-predicted -
measured) control loop proportional gain

    // 凸度自学习rs的PI控制系数
    prf_vrn_rs_i_gn = 0.2;
    ! [-] target profile vernier (re-predicted -
setup) control loop integral gain

    // 平直度自学习，学习的临界点
    flt_err_thrshld  = 100;
    ! [kN] Flatness errorthreshold minimum for fl
atness feedback

    // 操作工如果调整错误，但仍然进行自学习的弯辊力临界点
    opr_mx_wrng_corr = 100;
    ! [kN] Maximum operator correction in WRONG d

```

irection and still do flatness feedback

```
// APC修正开始的机架
apc_start_std      = 1;
! [-] APC Correction startstand

// 是否允许计算出口应变差
en_ex_strn_calc = true;
! [-] Enable exit strain calculation

// 出口应变差对应的最大厚度
ex_strn_thk = 10.00;                                ! [mm]
Exit strain match forthickness less than or equal to
.

// 是否排除不锈钢钢种
exclude_stainless = false;                            ! [-]
Exclude stainless steel

!@(CC087) start
// 凸度自学习，长短期以哪个为主的标识，默认长期自学习
prf_vrn_sel_flag = true;                                ! [-]
Vernier selection flag(false=samp, true=slfg)

// 平直度自学习，长短期以哪个为主的标识，默认长期自学习

flt_vrn_sel_flag = true;                                ! [-]
Vernier selection flag(false=samp, true=slfg)
!@(CC087) end

!@2ND(LC060) start
// 工作辊凸度补偿，标识是否使用sprp数据，默认使用长期自学习和sprp
wr_crn_off_sel_flag = true;                                ! [-] W
ork roll offset selection flat (false= slfg, true =
slfg+sprp)
```



```
!@2ND(LC061) end  
  end;  
end;
```