



宇阳的 板形模型 私房菜

热轧模型组

Table of Contents

Introduction	1.1
GSM总览	1.2

SSU板形设定模型

SSU板形设定模型	2.1
板形模型初始化	2.1.1
包络线计算	2.1.2
凸度分配计算	2.1.3
板形评估	2.1.4
板形物理模型	2.2
偏导数参数或增益	2.3

凸度分配

凸度分配和浪形的关系	3.1
道次出口目标计算	3.1.1
辊系凸度计算	3.1.2
UFD均载辊缝凸度计算	3.1.3

重难点模块与问题

重难点模块与问题	4.1
LPCE横向带钢模型	4.1.1
LRG横向辊缝模型	4.1.2
ALC模块	4.1.3

CRLC模块	4.1.4
CVC辊形设计	4.1.5

板形自学习

SMLC板形自学习模型	5.1
平直度自学习	5.1.1

ROP

SCF

板形动态控制

板形动态控制总览	8.1
----------	-----

参数和日志

参数和日志总览	9.1
SSU参数日志验算	9.1.1
SSU参数日志FAQ	9.1.2
CFG模型参数梳理	9.1.3
CTool参数梳理	9.1.4

板形模型实战

实战!	10.1
目标凸度变化影响板形模型设定	10.1.1

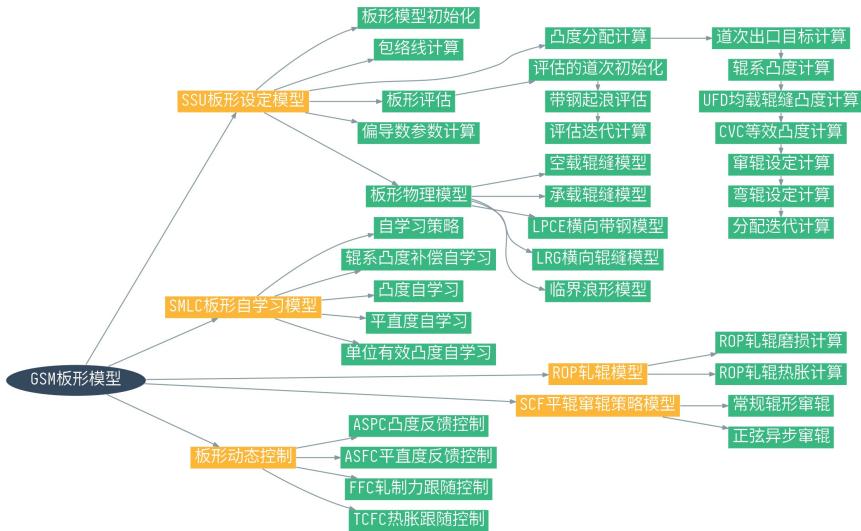
宇阳的板形模型私房菜（板形模型消化）

板形模型总览

GSM板形模型可以分为以下几个大方面：

- SSU板形设定模型
- SMLC板形自学习模型
- ROP轧辊模型
- SCF平辊窜辊策略模型
- 板形动态控制

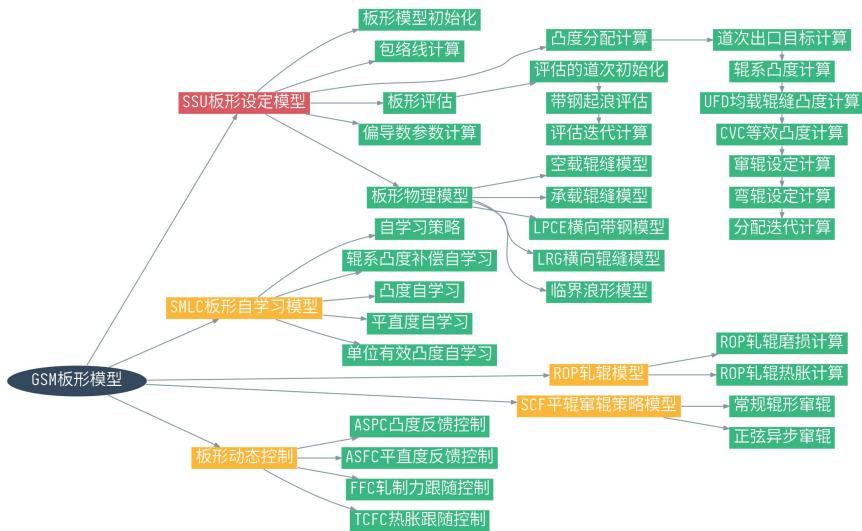
将GSM板形模型完全解剖，如下图所示。



本书从板形设定部分开始，对整个板形模型做详细的阐述。

板形设定模型总览

本章节主要介绍板形设定模型，本章节相对于整个GSM的位置如下图中红色节点所示。



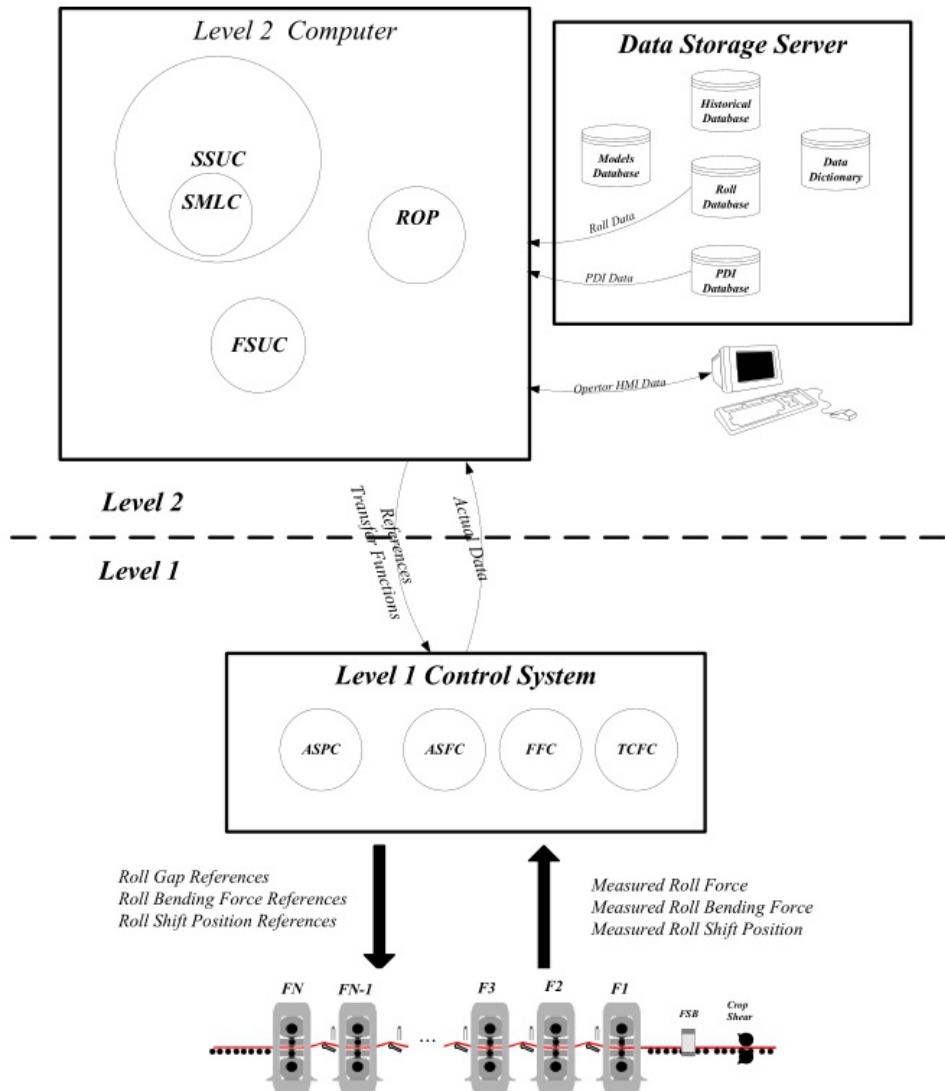
板形设定模型的功能

板形设定模型SSU的主要作用是计算并设定执行机构（主要是弯辊、窜辊）的参数，以获得符合质量要求的带钢凸度、带钢平直度。

SSU的输入参数来自FSU，包括各个机架的带钢出入口厚度、宽度、轧制力以及带钢温度等。SSU和FSU之间最重要的联系是各个机架的轧制力，一方面，FSU对轧制力的计算负责，另一方面各个机架的负荷分配直接影响到板形设定模型中的凸度分配。

板形模型的信息沟通

板形模型的参数来源于二级FSU的输出，同样也有来自L1一级基础自动化的参数和数据库的参数。板形模型的信息沟通如下图所示。



板形设定模型的步骤

板形设定模型的步骤可以分为如下几步：

- 根据执行机构的软硬极限以及浪形临界条件，计算板形控制的包络线。

- 根据轧机板形能力以及末道次机架出口的目标凸度，分配各个机架的出入口有效单位凸度。
- 计算窜辊系统和弯辊系统的设定值。
- 评估板形设定值以保证带钢浪形。

包络线计算

包络线计算实际反映了轧机的板形控制能力或板形控制极限。首先要确定弯窜辊的软硬极限，再进行凸度分配操作。

执行机构的软硬极限确定了UFD均载辊缝凸度的极限。如下面的公式所示，最小的弯辊力和最大的窜辊位置对应最大的UFD均载辊缝凸度；最大的弯辊力和最小的窜辊位置对应了最小的UFD均载辊缝凸度。

$$Cp_{ufd}|_{max} = func(F_b(min), Sp(max), \theta_{pc}(max))$$

$$Cp_{ufd}|_{min} = func(F_b(max), Sp(min), \theta_{pc}(min))$$

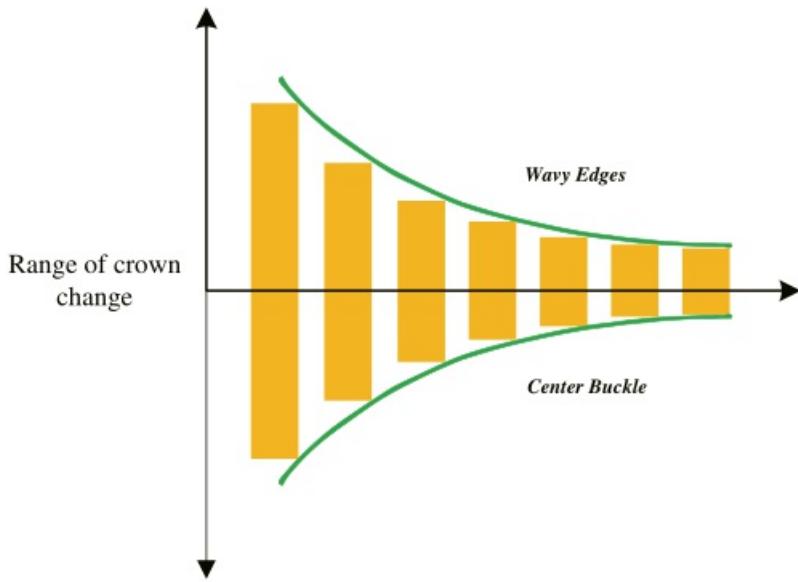
公式中， F_b 为弯辊力， Sp 为窜辊位置， θ_{pc} 为PC角角度。

根据最大最小UFD均载辊缝凸度可以确定最大最小的入口有效单位凸度。

$$Cp_{entry_effct}|_{max} = Cp_{ufd}|_{max} - \epsilon_{cb} \cdot \frac{K_{chg_attn}}{K_{pce_infl}}$$

$$Cp_{entry_effct}|_{min} = Cp_{ufd}|_{min} - \epsilon_{we} \cdot \frac{K_{chg_attn}}{K_{pce_infl}}$$

带钢的屈曲极限或临界浪形决定了每一个机架的凸度改变量。如下图所示，注意上限代表双边浪的极限，下限代表中浪的极限，两极限中间的区域表征良好板形，超出这个区间的范围表征浪形缺陷。



每个机架的最大的双边浪改变量 ΔCp_{we} 和最大的中浪改变量 ΔCp_{cb} 构成了各个机架的板形调整能力。

$$\Delta Cp_{we} = \epsilon_{we} \cdot \frac{1 - K_{pce_infl} \cdot (1 - (1 - f_{prf_rlf}) \cdot f_{strn_rlf})}{K_{pce_infl}}$$

$$\Delta Cp_{cb} = \epsilon_{cb} \cdot \frac{1 - K_{pce_infl} \cdot (1 - (1 - f_{prf_rlf}) \cdot f_{strn_rlf})}{K_{pce_infl}}$$

已知执行机构约束和带钢浪形约束的条件下，我们协调这二者以获得各个机架的带钢凸度包络线。

协调包络线过程中最重要的一个原则就是，上一个机架的出口单位凸度必须在当前机架的执行机构极限范围内，同时也要在下一个机架的凸度改变范围内。

板形模型初始化

板形模型初始化主要包含在cShapeSetupD::Main(..)函数中。

状态量的建立

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 = cMdtparam::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_granted ) &&
( 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1st.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;
}

```

短期自学习预设定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。所谓的预设定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TARGET。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 = cMdparam::cs_green;

```

cShapeSetupD::Init(..)的实现见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）。

cTargetD::Init(..)

在cTargetD::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

pcTargetD->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_DEnthick、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 [ passIdx ];
}
else if ( pcTargtD->prf_del < 0.065F )
{
    f_wr_crn_off_adj = psSPRP->wr_crn_off_adj2 [ passIdx ];
}
else
{
    f_wr_crn_off_adj = psSPRP->wr_crn_off_adj3 [ passIdx ];
}
```

初始化分配和评估的横向带钢对象，利用pcFSPassD->pcAlcLPceD[iter]->Init(..)和pcFSPassD->pcEvLPceD[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 SHOULD be CVC  
//-----  
    pcFSStdDloc->force_bnd = pcFSStdDloc->pcFSStd->force_bnd_nom  
    - pcTargtD->flt_vrn;  
}  
else  
{  
    pcFSStdDloc->force_bnd = pcFSStdDloc->pcFSStd->force_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 )
```

```

pcFSSStdDloc->wr_shft = 0.0F;
else
    pcFSSStdDloc->wr_shft = psSSys->targ_pos_shft[ passIdx ];

```

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 = minI; i <= maxI; 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，必要情况下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。

板形包络线计算

在cPEnvD::Calculate(..)这个函数中计算包络线。

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。

首先根据弯辊辊的极限值确定弯辊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pee_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_lim[ maxl ];
pcFSPassD->pcPEnvD->wr_br_crn_env[ minl ] =
    pcFSPassD->pcPEnvD->wr_br_crn_lim[ maxl ];
pcFSPassD->pcPEnvD->pce_wr_crn_env[ maxl ] =
    pcFSPassD->pcPEnvD->pce_wr_crn_lim[ minl ];

```

```

pcFSPassD->pcPEnvD->wr_br_crn_env[ maxl ] =
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 p
profile
    // envelope.
    //-----

    line_num = __LINE__;
    pcFSPassD->pcPEnvD->ufd_pu_prf_env[ i ] =
        pcFSPassD->pcFSStdD[ iter ]->pcUFDD->Prf(
            pcFSPassD->pcPEnvD->force_pu_wid_env [ i ],
            pcFSPassD->pcPEnvD->force_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 buckling limits for the given p
ass.
    //-----

    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二者计算出入口（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 ]->pclRGD->Ef_En_PU_Prf1(
        pcFSPassD->pcPEnvD->ufd_pu_prf_env [ minl ],
        pcFSPassD->pcPEnvD->std_ex_strn_lim [ we ] );

line_num = __LINE__;
pcFSPassD->pcPrvAct->pcPEnvD->ef_pu_prf_lim[ maxl ] =
    pcFSPassD->pcFSStdD[ iter ]->pclRGD->Ef_En_PU_Prf1(
        pcFSPassD->pcPEnvD->ufd_pu_prf_env [ maxl ],
        pcFSPassD->pcPEnvD->std_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_obj) )->pcPEnvD->ef_pu_prf_
env[ i ] =
    ( ( cFSPassD* )(pcFstFSPassD->previous_obj) )->pcLPceD->ef_pu_
prf;

    //-----
    // Initialize the limiting pass envelope.
    //-----
    pas_env_lim[ i ] =
    ( ( cFSPassD* )(pcFstFSPassD->previous_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_Prf3(..)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_Prf3 (
        pcFSPassD->pcLPceD->strn_rlf_cof,
        pcFSPassD->pcPrvAct->pcPEnvD->ef_pu_prf_env[ mi
    n1 ],
    pcFSPassD->pcPEnvD->ufd_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->pcPEnvD->ef_pu_prf_env[ minl ]
) &&
!=
( pcFSPassD->pcPrvAct->pcPEnvD->ef_pu_prf_env[ minl ]
pcFSPassD->pcPrvAct->pcPEnvD->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->pcPEnvD->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_Prf3(..)。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_min，重计算(re-calculate)辊系凸度。

```

//-----
-----  

// Re-calculate the following composite roll stack crown quant  

ities:  

//      Piece to work roll stack crown  

//      Work roll to backup roll stack crown  

//-----  

-----  

line_num = __LINE__;  

pcFSPassD->pcFSStdD[ iter ]->pcCRLCD->Crns(  

    pcFSPassD->pcPEnvD->pos_shft_env [ minl ],  

    pcFSPassD->pcPEnvD->angl_pc_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->pcPEnvD->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->pcPEnvD->std_ex_strn_lim[ cb ]
)
    if ( 1 < 0 )
{..}
// strain too high (due to low entry profile)
//if ( std_ex_strn > pcFSPassD->pcPEnvD->std_ex_strn_lim[ we ]
&&
    if ( 0 > 1 &&
        pcFSPassD->pcFSStdD[ iter ]->pcLRGD->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_Prf3(
    pcFSPassD->pcLPceD->strn_rlf_cof,
    pcFSPassD->pcPrvAct->pcPEnvD->ef_pu_prf_env[ min1 ],
    pcFSPassD->pcPEnvD->ufd_pu_prf_env           [ min1 ]
);
```

凸度分配计算

凸度分配计算主要在cAlcD::Calculate(..)函数中执行。

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->pcEv1LPceD[ iter ]->strn_rlf_cof  
/ (pcFSPassDtmp->pcFSStdD[ iter ]->pclRGD->pce_infl_cof  
* pcFSPassDtmp->pcEv1LPceD[ iter ]->elas_modu);
```

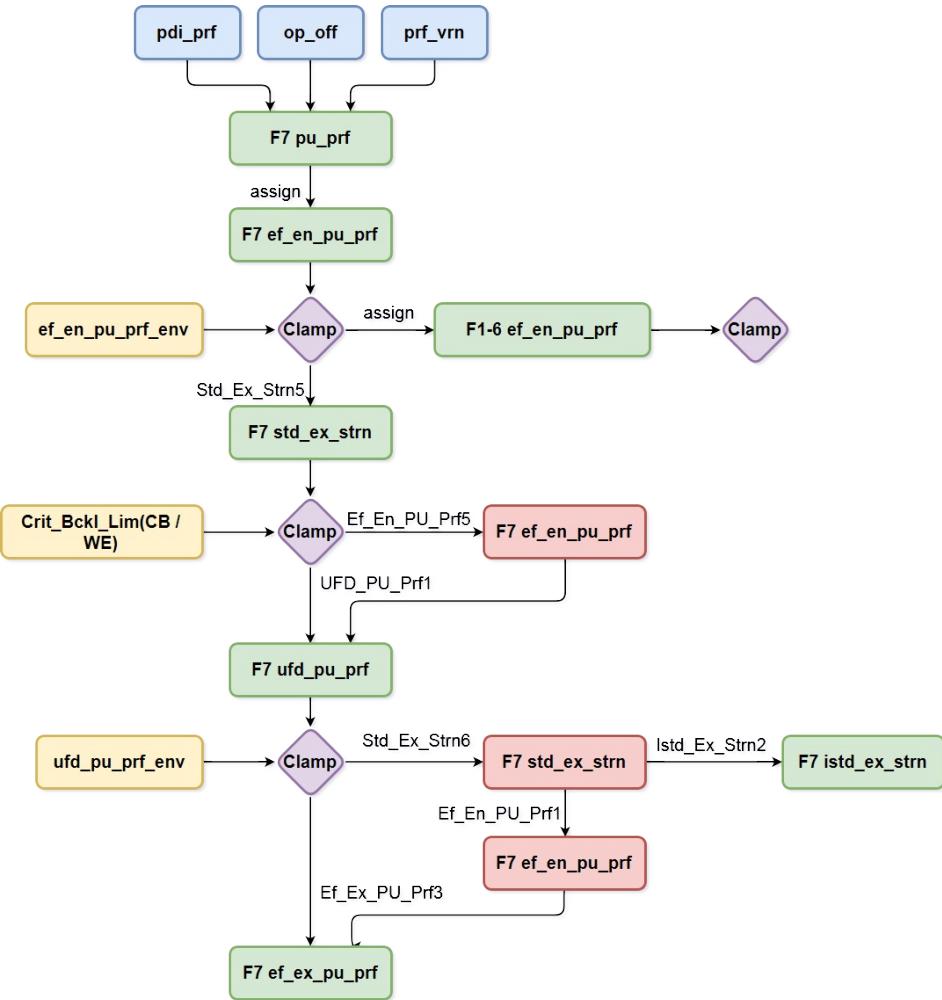
或者表示为：

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

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_Prf5，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_obj )->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 cal-
culations
    &force_pu_wid_buf,                // OUT rolling force/w-
idth
    [-]
    &fwd_slip,                        // OUT exit slip ratio
    &arcon,                           // OUT length of arc [
    minor_length]
    ( ( cFSPassD* )pcFSPassD->previous_obj )->pcAlcD->thick,
    // IN entry_thk
    pcAlcD->thick,                  // IN exit_thk
    pcEnPceD->width,                // IN exit/entry width
    speed
    pcStdD->speed,                  // IN roll peripheral
    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 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->force_pu_wid,
        pcStdD->force_strip / pcStdD->pcEnPceD
    ->width,
        pcPEnvD->force_pu_wid_env,
        pcAlcD->pcRollbite->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_Prf3(..)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pcTargtD->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Delivry_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 the piece
// influence coefficient is zero.
//-----
ef_en_pu_prf = cMathUty::Clamp( ef_ex_pu_prf,
    pcPceIZFSPassD->pcPEnvD->ef_pu_prf_env[ minl ],
    pcPceIZFSPassD->pcPEnvD->ef_pu_prf_env[ maxl ] );

```

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，或真正的有
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( pcBuffFSPassD->pcFSStdD[ iter ]->pcLRGD->ef_pu_prf_chg[ cb ],
        cMathUty::Max( ef_ex_pu_prf,
            pcBuffFSPassD->pcPEnvD->ef_pu_prf_env[ minl ] ) -
        cMathUty::Min( ef_en_pu_prf,
            ((cFSPassD*)pcBuffFSPassD->previous_obj)->pcPEnvD->ef_pu_prf_env[ maxl ] ) );
ef_pu_prf_dlt[ maxl ] =
    cMathUty::Min( pcBuffFSPassD->pcFSStdD[ iter ]->pcLRGD->ef_pu_prf_chg[ we ],
        cMathUty::Min( ef_ex_pu_prf,
            pcBuffFSPassD->pcPEnvD->ef_pu_prf_env[ maxl ] ) -
        cMathUty::Max( ef_en_pu_prf,
            ((cFSPassD*)pcBuffFSPassD->previous_obj)->pcPEnvD->ef_pu_prf_env[ minl ] ) );

```

```
((cFSPassD*)pcBufFSPassD->previous_obj)->pcPEnvD->ef_p  
u_prf_env[minl]));
```

ef_pu_prf_dlt的计算，讲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[maxl]。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大于原ef_en_pu_prf，那么ef_pu_prf_sum累加ef_pu_prf_dlt[minl]。

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[minl] *  
    (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->en_pu_prf - pcTargtD->pu_prf  
)  
* pcFSPassD->pcAlcD->pu_prf_change;
```

最后用pcTargtD->Eval_Ef_En_PU_Prf(..) 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。

板形评估

分配阶段结束后就进入评估阶段。

评估的手段主要是从前道次机架往后依次对各个机架的浪形情况进行评估。如果机架的出口应变差超出了死区范围，那么模型就试着利用弯辊去消除多余的应变差。如果弯辊不能对浪形情况进行修正，对凸度削弱因子进行调整，最终获得合适的板形。

板形评估的过程主要在cShapeSetupD::Evaluate(..)函数中执行。

评估准备工作

板形评估的作用是用来评估带钢的浪形是否满足要求。因此会设定flt_ok指示器。

```
pcLPceD->flt_ok = true;
```

之后更新横向带钢模型，获得最新的应变释放系数、带钢弹性模量以及最新的屈曲判别极限。

```
pcLPceD->Update( pcFSSstdD->pcExPceD->temp_avg,
                    pcFSSstdD->pcExPceD->width,
                    thick,
                    pcFSSstdD->pcExPceD->tension );
```

如果当前机架非空过，则计算当前机架轧制力相对于出口带钢厚度的偏导数，供后面横向辊缝模型参数的更新使用。

```
if ( !pcFSSstdD->pcRollbite->Calculate_DForce_DExthick( 0.01F ) )
{
    EMSG << "Evaluate: PID="
    << pcFSSstdD->pcExPceD->pcPce->prod_id
    << ", invalid ROLLBITE dforce_dexgag"
    << END_OF_MESSAGE;
}
```

横向辊缝模型参数的更新。

```
pcFSStdD->pcLRGD->Update( pcFSStdD->dummied,
                                pcFSStdD->pcExPceD->width,
                                pcFSStdD->pcEnPceD->thick,
                                thick,
                                force_pu_wid,
                                pcFSStdD->pcEnPceD->pcPce->family,
                                pcFSStdD->pcRollbite->DForce_DExthick(),
                                pcFSStdD->fs,
                                pcFSStdD->arcon,
                                pcLPceD );
```

之后利用cCRLCD::Crns(..)计算带钢-工作辊和工作辊-支承辊的辊系凸度。

```
pcFSStdD->pcCRLCD->Crns ( pcFSStdD->wr_shft,
                                pcFSStdD->angl_pc,
                                pcFSStdD->pcCRLCD->pce_wr_cr,
                                pcFSStdD->pcCRLCD->wr_br_cr);
```

单位凸度和浪形的计算

准备工作结束后，利用pcUFDD->Prf(..)计算pcLPceD->ufd_pu_prf。

```
pcLPceD->ufd_pu_prf =
pcFSStdD->pcUFDD->Prf ( force_pu_wid,
                            force_bnd,
                            pcFSStdD->pcCRLCD->pce_wr_cr,
                            pcFSStdD->pcCRLCD->wr_br_cr )/ thick;
```

用上一个机架的有效单位凸度和当前机架的UFD均载辊缝凸度，计算出口应变差。

```
pcLPceD->strn = pcFSStdD->pcLRGD->Std_Ex_Strn1( pcPrvLPceD->ef_pu_prf,
                                                        pcLPceD->ufd_pu_prf );
```

如果应变差超出了屈曲判别的极限，且弯辊力设定值还可以修改，则根据当前的应变差重新计算UFD均载辊缝单位凸度，并更新弯辊力的值，最后再次更新应变差。

这是如果应变差还是不满足屈曲判别的要求，那么修正带钢的凸度改变削弱因子。

```
pcFSStdD->pcLRGD->Cor_Prf_Chng_Attn_Fac_Sup (
    bckl_lim,
    pcLPceD->strn );
```

最后更新出口应变差和单位有效凸度，并根据前两者计算新的凸度prf。

板形物理模型

板形模型的精度，需要一些板形理论参数支撑。比如带钢影响系数、应变释放系数等，本文进行简要说明。

带钢影响系数

金属的横向流动对板形和浪形的影响都很大。模型在这里运用了“带钢影响系数”来表征这一影响因素。

带钢影响系数反映了入口比例凸度对出口比例凸度的影响，取值在0到1之间。如果带钢影响系数取值为0，说明带钢板形的变化不会造成带钢的延伸，如果带钢影响系数取值为1，说明带钢板形的变化完全影响了带钢的延伸，并且带钢在横向没有出现重新分配。

凸度削弱因子

板形的变化会对自身的板形有印制作用。

如果带钢的比例凸度增加，边部的带钢的压下率要明显高于中部带钢的压下率。因此，更高的轧制力导致了轧辊出现更大的压扁和偏斜，进而导致了边部的带钢更厚，从而对比例凸度造成抑制。在这里，这种现象和带钢的硬度紧密相关联。

同时，带钢的比例凸度增加，会导致带钢在宽度方向上出现更不均匀的纵向延伸，边部的带钢受更大的压应力。因此，在带钢边部分布的单位轧制力更大，更高的轧制力导致了轧辊出现更大的压扁和偏斜，进而导致了边部的带钢更厚，从而对比例凸度造成抑制。在这里，这种现象和带钢所受的张力紧密相关联。

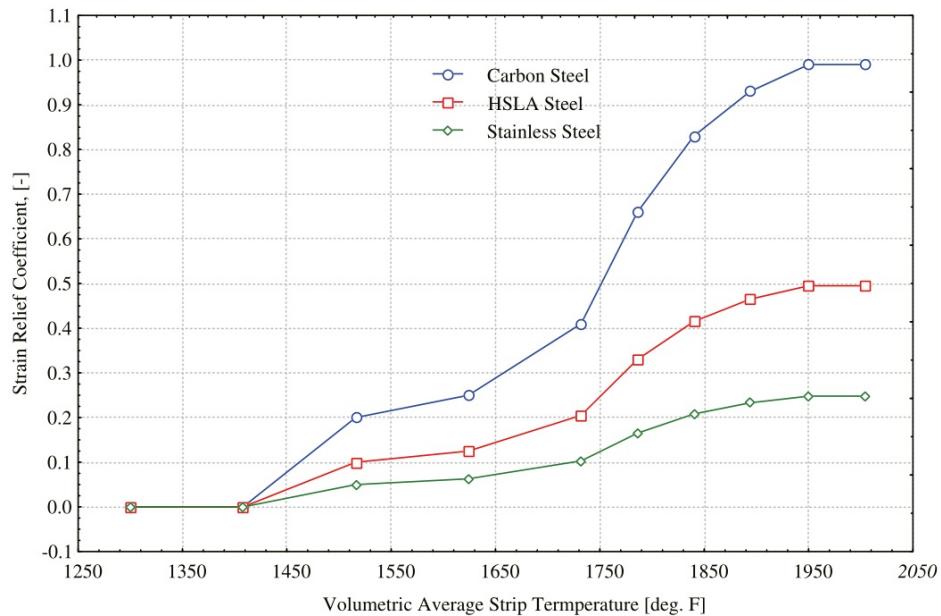
针对以上现象，板形理论用“凸度削弱因子”来表征这种特性。在板形设定模型的评估阶段，如果修改弯辊力也无法满足临界浪形的要求，则重新计算凸度削弱因子，以改变双边浪和中浪的最大调整量。

应变释放系数

应变恢复系数用来描述带钢的塑性恢复和机架间带钢的二次变形。应变恢复系数是个经验系数，它和带钢温度和化学成分有关，在板形设定模型当中，这个值通过带钢机架出口温度插值计算获得。

应变释放系数的取值在0和1之间。取值0说明应变不会释放，取值1说明应变完全释放。应变的释放与否和带钢的速度有很大关系。因为带钢的速度的背后，是轧制过程时间、带钢温度和化学成分对应变释放的影响。

如下图所示是不同的钢种（不同的化学成分）对带钢应变释放系数的影响。一般来说，不锈钢相对于碳钢需要更长的时间进行应变恢复。



上一机架出口的带钢板形，到下一机架的入口前发生应变恢复。上一机架出口的应变差一般不等于下一机架入口的应变差，二者之间需要应变释放系数进行修正，如下面公式所示。其中 f_{strn_rlf} 为应变释放系数。

$$\epsilon_{entry}^{i+1} = (1 - f_{strn_rlf}) \cdot \epsilon_{exit}^i$$

板形恢复系数

由于应变恢复，带钢的板形会出现一些微小的变化。因此我们用“板形恢复系数”来表征应变释放对板形变化的影响大小。因此各个机架的单位凸度改变量如下公式所示。

$$\Delta Cp^i = f_{prf_recv} \cdot \Delta \epsilon_{exit}^i = f_{prf_recv} \cdot f_{strn_rlf} \cdot \epsilon_{exit}^i$$

下一机架入口的单位凸度如下公式所示。

$$Cp_{entry}^{i+1} = Cp_{exit}^i - \Delta Cp^i = Cp_{exit}^i - f_{prf_recv} \cdot f_{strn_rlf} \cdot \epsilon_{exit}^i$$

偏导数参数或增益

在板形模型的中间计算结果当中会涉及到一些偏导数参数，这些参数反映了两个板形物理量之间的变化关系，或者说增益。

弯辊力增益

弯辊力主要的偏导数计算在cShapeSetupD::Xfer_Functions(..)函数中完成。这个函数中计算了以下几个参数对之间的增益。

- 弯辊力对轧制力的增益
- 弯辊力对窜辊位置的增益
- 弯辊力对UFD均载辊缝凸度的增益
- 弯辊力对机架出口应变差的增益
- 弯辊力对带钢-工作辊辊系凸度的增益
- 带钢-工作辊辊系凸度对工作辊-支承辊辊系凸度的增益

UFD凸度增益

UFD凸度增益分为以下四个方面。

- UFD凸度对弯辊力的增益
- UFD凸度对轧制力的增益
- UFD凸度对带钢-工作辊辊系凸度的增益
- UFD凸度对工作辊-支承辊辊系凸度的增益

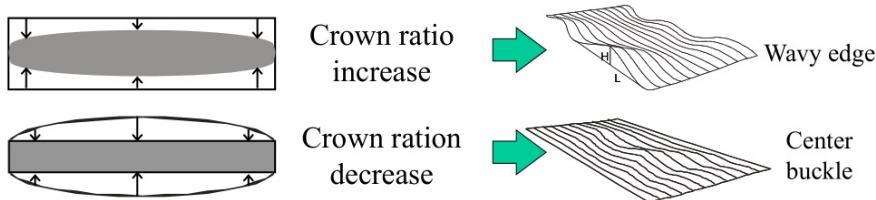
这四个增益以系数（mult）的形式参与和UFD模块有关的计算。

凸度分配和板形的关系

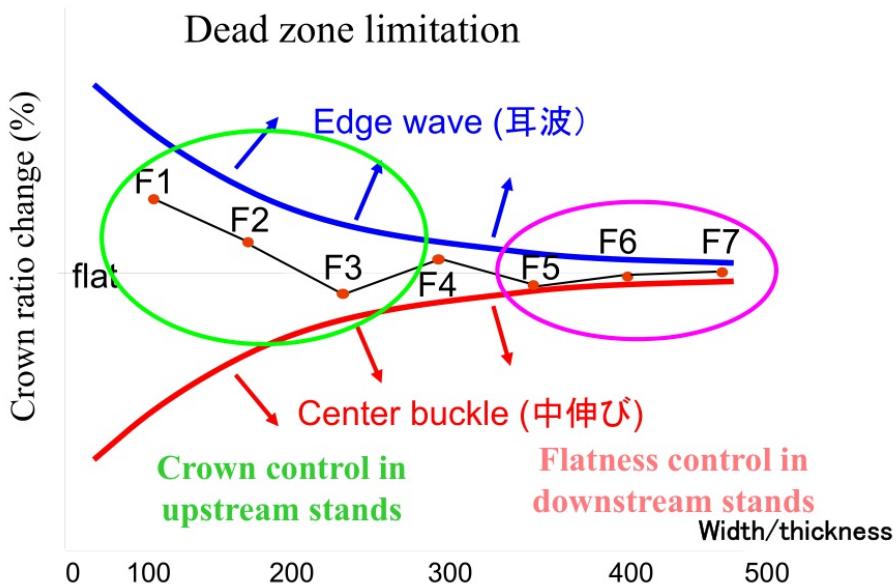
热轧七机架连轧，从中间坯到F7出口成品，精轧模型确定了各个机架的负荷分配，板形模型在精轧模型的基础上确定了各个机架的凸度分配。凸度分配的策略是板形设定模型的主旋律。

浪形和凸度的关系

凸度分配的好坏直接影响了带钢的浪形，因此带钢的浪形缺陷通过机架间的凸度分配，和带钢凸度之间建立关系。上游机架凸度小时辊缝大，而下游机架凸度大同时辊缝小，会导致边浪缺陷。上游机架凸度大同时辊缝大，而下游机架凸度小时辊缝小，会导致中浪缺陷。为了有效描述板形，我们引入了比例凸度的概念，比例凸度指的是当前机架的凸度和当前带钢厚度的比值。



因此，浪形和凸度之间的关系依靠“比例凸度的变化”进行衔接。热连轧前后机架之间比例凸度不同，比例凸度的变化如果在允许的范围内，则不会出现中浪和边浪，但是如果比例凸度的变化超过了允许的范围，则必然产生带钢的浪形。这里所允许的范围被称作死区。如下图所示为死区。从F1到F7，死区区间大小逐渐收窄，形成喇叭口。



在板形模型中，用应变差（differential）的概念来表示比例凸度变化在浪形死区上的分布情况。比如在某一机架出口，此处的应变差指的是出口的比例凸度和入口有效比例凸度的差值。什么是“入口有效比例凸度”？也就是入口的比例凸度减去入口的应变差。应变差的变化就是模型对“比例凸度变化”的具体描述。

浪形和凸度关系的数学证明

在某一机架，我们假设带钢宽度方向上中心部位的入口厚度为 H_c ，中心入口长度为 L_c ，则机架出口的中心部位厚度为 h_c ，中心部位长度为 l_c 。再假设边部入口厚度为 H_e ，边部入口长度为 L_e ，则机架出口的边部出口厚度为 h_e ，边部出口长度为 l_e 。

根据体积不变定律，在宽度不变的情况下，有如下的关系。

$$H_c \cdot L_c = h_c \cdot l_c$$

$$H_e \cdot L_e = h_e \cdot l_e$$

将以上两个式子上下相处再变换，我们获得：

$$\frac{l_e}{l_c} = \frac{H_e}{H_c} \cdot \frac{h_c}{h_e} \cdot \frac{L_e}{L_c}$$

将式子两边取对数，我们获得：

$$\ln\left(\frac{l_e}{l_c}\right) = \ln\left(\frac{H_e}{H_c} \cdot \frac{h_c}{h_e} \cdot \frac{L_e}{L_c}\right)$$

将上式右边拆分，可以获得：

$$\ln\left(\frac{l_e}{l_c}\right) = \ln\left(\frac{H_e}{H_c}\right) - \ln\left(\frac{h_e}{h_c}\right) + \ln\left(\frac{L_e}{L_c}\right)$$

这四个量都是无穷小量，因此可以利用等价无穷小，当 x 趋向于1时， $\ln x$ 的等价无穷小为 $x - 1$ 。将上式转化成如下公式。

$$\frac{l_e - l_c}{l_c} = \frac{h_c - h_e}{h_c} - \frac{H_c - H_e}{H_c} + \frac{L_e - L_c}{L_c}$$

在上式中， $\frac{l_e - l_c}{l_c}$ 为中间和边部延伸的变化率，可以当作出口应变差。 $\frac{h_c - h_e}{h_c}$ 中，

分子为出口带钢凸度，分母为出口带钢厚度， $\frac{h_c - h_e}{h_c}$ 可以表征为带钢的出口单位凸度。因此上式最终转化为：

$$\epsilon_{exit} = Cp_{exit} - (Cp_{entry} - \epsilon_{entry}) = Cp_{exit} - Cp_{entry_effct}$$

式中， ϵ 为长度方向上的应变差， Cp 为单位凸度或比例凸度， Cp_{entry_effct} 此机架入口的有效单位凸度。有效单位凸度的物理含义是，消除内部应力和金属横向流动的单位凸度。

板形出口目标计算

对于板形模型来说，它的目标不是工艺工程师眼中的PDI目标，板形模型的凸度目标还需综合考虑各种自学习量和补偿值。

操作工补偿op_off

目标初始化在cTargtD::Init(..)函数中执行。

PDI板形目标和板形模型角度的板形目度之间有各种自学习量和补偿量的偏差。

首先，PDI目标凸度和目标平直度需要加上操作工的补偿。

下面的代码是在cTargtD::Init(..)外部执行。

```
pcFstFSPassD->pcFSStdD[ iter ]->pcEnPceD->pcPce->tgt_profile =  
    psPDI->tgt_profile + psSSys->op_prf_off;  
  
pcFstFSPassD->pcFSStdD[ iter ]->pcEnPceD->pcPce->tgt_flatness =  
    psPDI->tgt_flatness + psSSys->op_flt_off;
```

下面的代码是在cTargtD::Init(..)内部执行，计算初始目标凸度。

```
//-----  
// Calculate the target flatness.  
//-----  
flt = ( pdi_flt + flt_op_off ) / Physcon.i_units;  
  
//-----  
// Calculate the initial target profile.  
//-----  
  
prf_int = ( pdi_prf + prf_op_off );
```

凸度自学习量

凸度自学习量的计算如下：

```
//-----
// Calculate the target profile vernier.
//-----
prf_vrn = prf_rm_vrn - prf_rs_vrn;
```

之后将凸度自学习量补到初始目标凸度上。

```
prf_pdi_tgt = prf_int * matl_exp_cof;
prf_int = prf_int * matl_exp_cof + prf_vrn;
```

最后计算目标单位凸度。目标单位凸度才是模型角度的板形目标值。

```
pu_prf_tgt = prf_int / pce_thck;
```

辊系凸度计算

辊系凸度是板形模型中比较重要的概念。辊系凸度其实就是将空载情况下轧辊的影响因素全部转化为凸度值，再将各个等效的凸度累加的结果。

辊系凸度分为带钢-工作辊辊系凸度和工作辊-支承辊辊系凸度。以带钢工作辊辊系凸度为例子。

带钢-工作辊辊系凸度 = 磨损 + 热胀 + CVC等效凸度 + 工作辊系自学习 + 工作辊补偿。

如下图所示，运算获得总辊系凸度时，各个影响因素之间是相加的关系。

Composite Roll Stack Crowns																						
Pce	WR	WR	BR	Pce	WR	BR	WR	BR	Pce	WR	WR	BR	WR	WR	Dlt	Pce	Dlt	WR	SLFG	SPRP		
Std	Gap	Gap	Wear	Wear	Wear	Thrm	Thrm	Grnd	Grnd	Eqv	Vern	Offs	WR_Gap	BR_Gap	WR	WR	Dlt	Pce	Dlt	WR	SLFG	SPRP
#	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	Offs	Offs	
1	-0.7641	-0.4791	-0.0665	-0.0665	0.1824	0.1824	-0.8600	0.0945	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0048	-0.0075	-0.0075	-0.0075		
2	-0.4295	-0.2192	-0.0093	-0.0093	0.1823	0.1823	-0.5224	0.0945	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0034	-0.0048	-0.0058	-0.0058	-0.0058		
3	-0.1554	0.0003	-0.0186	-0.0000	0.1816	0.1833	-0.2761	0.0945	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0035	0.0006	-0.0025	-0.0025	-0.0025		
4	-0.0256	0.1063	-0.0256	-0.0000	-0.0256	0.1686	0.1686	-0.1686	0.0945	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0038	0.0010	-0.0020	-0.0020	-0.0020		
5	0.1370	0.2398	-0.0354	-0.0000	-0.0354	0.2094	0.2094	-0.0369	0.0945	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002	0.0002	-0.0020	-0.0020	-0.0020		
6	0.0108	0.1425	-0.0355	-0.0000	-0.0355	0.2144	0.2144	-0.1681	0.0945	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002	0.0002	-0.0016	-0.0016	-0.0016		
7	-0.0125	0.1213	-0.0412	-0.0000	-0.0412	0.2058	0.2058	-0.1771	0.0945	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0030	0.0011	-0.0016	-0.0016	-0.0016		

Composite Roll Stack Crowns																						
Pce	WR	WR	BR	Pce	WR	BR	Pce	WR	WR	WR	WR	WR	WR	WR	Dlt	Pce	Dlt	WR	SLFG	SPRP		
Std	Gap	Gap	Wear	Wear	Wear	Thrm	Thrm	Grnd	Grnd	Eqv	Vern	Offs	WR_Gap	BR_Gap	WR	WR	Dlt	Pce	Dlt	WR	SLFG	SPRP
#	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	Offs	Offs	
1	-0.8299	-0.4503	-0.0181	-0.0000	-0.0258	0.2275	0.2275	-0.7970	0.2299	0.0000	-0.0200	-0.4223	0.0000	0.0000	0.0000	0.0777	-0.5000	-0.4000	-0.4000	-0.4000		
2	-0.4948	-0.0712	-0.0319	-0.0000	-0.0527	0.2873	0.2873	-0.4712	0.2299	0.0000	-0.0200	-0.2582	0.0000	0.0000	0.0000	0.0918	-0.3500	-0.2500	-0.2500	-0.2500		
3	-0.3875	0.0112	-0.0451	-0.0000	-0.0858	0.2294	0.2294	-0.2154	0.3656	0.2299	0.0000	-0.0200	-0.1062	0.0000	0.0000	0.0000	0.0938	-0.2000	0.0000	0.0000	0.0000	
4	-0.2845	0.0766	-0.0663	-0.0000	-0.1149	0.2525	0.2379	-0.3675	0.2299	0.0000	-0.0200	-0.0833	0.0000	0.0000	0.0000	0.0967	-0.1000	0.1000	0.1000	0.1000		
5	-0.3983	-0.0558	-0.0486	-0.0000	-0.0865	0.1752	0.1669	-0.3934	0.2299	0.0000	-0.0200	-0.1033	0.0000	0.0000	0.0000	0.0967	-0.2000	0.1000	0.1000	0.1000		
6	-0.2837	-0.1460	-0.0754	-0.0000	-0.0755	0.1637	0.1358	-0.3600	0.0000	0.0000	-0.0300	0.0979	0.0000	0.0000	0.0000	0.0979	0.0000	0.0000	0.0000	0.0000		
7	-0.1051	-0.0764	-0.0667	-0.0000	-0.0634	0.2016	0.1565	-0.3600	0.0000	0.0000	-0.0300	0.1500	0.0000	0.0000	0.0000	0.1500	0.0000	0.0000	0.0000	0.0000		

Composite Roll Stack Crowns																						
Pce	WR	WR	BR	Pce	WR	WR	BR	WR	Pce	WR	WR	BR	WR	WR	Dlt	Pce	Dlt	WR	SLFG	SPRP		
Std	Gap	Gap	Wear	Wear	Wear	Thrm	Thrm	Grnd	Grnd	Eqv	Vern	Offs	WR_Gap	BR_Gap	WR	WR	Dlt	Pce	Dlt	WR	SLFG	SPRP
#	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	Offs	Offs	
1	-0.8299	-0.4503	-0.0181	-0.0000	-0.0258	0.2275	0.2275	-0.7970	0.2299	0.0000	-0.0200	-0.4223	0.0000	0.0000	0.0000	0.0777	-0.5000	-0.4000	-0.4000	-0.4000		
2	-0.4948	-0.0712	-0.0319	-0.0000	-0.0527	0.2873	0.2873	-0.4712	0.2299	0.0000	-0.0200	-0.2582	0.0000	0.0000	0.0000	0.0918	-0.3500	-0.2500	-0.2500	-0.2500		
3	-0.3875	0.0112	-0.0451	-0.0000	-0.0858	0.2294	0.2294	-0.2154	0.3656	0.2299	0.0000	-0.0200	-0.1062	0.0000	0.0000	0.0000	0.0938	-0.2000	0.0000	0.0000	0.0000	
4	-0.2845	0.0766	-0.0663	-0.0000	-0.1149	0.2525	0.2379	-0.3675	0.2299	0.0000	-0.0200	-0.0833	0.0000	0.0000	0.0000	0.0967	-0.1000	0.1000	0.1000	0.1000		
5	-0.3983	-0.0558	-0.0486	-0.0000	-0.0865	0.1752	0.1669	-0.3934	0.2299	0.0000	-0.0200	-0.1033	0.0000	0.0000	0.0000	0.0967	-0.2000	0.1000	0.1000	0.1000		
6	-0.2837	-0.1460	-0.0754	-0.0000	-0.0755	0.1637	0.1358	-0.3600	0.0000	0.0000	-0.0300	0.0979	0.0000	0.0000	0.0000	0.0979	0.0000	0.0000	0.0000	0.0000		
7	-0.1051	-0.0764	-0.0667	-0.0000	-0.0634	0.2016	0.1565	-0.3600	0.0000	0.0000	-0.0300	0.1500	0.0000	0.0000	0.0000	0.1500	0.0000	0.0000	0.0000	0.0000		

UFD均载辊缝凸度计算

轧辊和带钢直接接触，因此轧辊提供的辊缝形状决定了最终的带钢板形。在热轧生产过程中，对轧辊辊缝形状有直接影响的因素是：

(1) 轧辊的初始辊形。 (2) 轧制力造成的轧辊弯曲。 (3) 弯辊力造成的轧辊弯曲。 (4) 因窜辊导致的轧辊辊形改变。 (5) 轧辊受热后的膨胀。 (6) 轧辊和轧件接触造成的轧辊磨损。

以上六大主要因素共同作用最终形成了轧辊的辊缝形状。

板形设定模型最大的价值在于获取足够高的板形精度。遗憾的是在高精度板形设定这一块没有什么高精度的算法支持，所以主要是通过线性拟合的方式计算辊缝凸度的基准值，再逐步进行修正。

如果把设定模型对板形的计算比作绘制一幅肖像画，那么第一步需要勾勒出人物大概的形状轮廓，再逐步添加细节。

所以在板形设定中有了UFD的概念。UFD的全称是Uniform force distribution，意思是在假设载荷均匀的条件下，轧制过程中的轧辊辊缝形状。

我们研究物理问题总是先设定各种假定条件，比如研究天文问题，先把地球比作一个质点，研究力学问题，先把小车比作一个方块。在板形模型当中，先假设轧辊的受力是均匀的。

因此板形设定模型对板形的设定可以分为两部。第一步是根据上文提出的六大影响因素先计算出UFD（均载辊缝的形状），然后结合边界条件对带钢形状精细调整。

板形模型和模块重难点问题

本章节对板形模型和相关模块当中的重难点问题进行说明。

LPCE横向带钢模型

LPCE模块可以说是SSU中最简单的模块了。

本模块主要用于计算以下三个参数。

- 带钢弹性模量elas_modu
- 应力释放系数strn_rlf_cof
- 屈曲判别极限Crit_Bckl_Lim

插值计算elas_modu和strn_rlf_cof

带钢弹性模量和应力释放系数主要通过插值计算的方式获得。

利用各个机架的出口温度，和带钢弹性模量和应力释放系数的插值表进行插值计算，获得各个机架的带钢弹性模量和应力释放系数。

一般插值表如下，很少对此插值表进行改动。

avg_pce_tmp_interp_vec	elas_modu_interp_vec	strn_rlf_cof_interp_vec
600	138269	0
650	128069	0
700	117905	0
750	107751	0.056
800	97589	0.091
850	87415	0.16
900	77232	0.303
950	67054	0.521
1000	56909	0.771
1050	46829	0.968
1100	36863	0.984
1500	27067	0.984

屈曲判别极限的计算

屈曲判别极限分为边浪的判别极限和中浪的判别极限。各个机架的屈曲判别极限和带钢的出口厚度、出口宽度、纵向平均张力、弹性模量有关，根据以下经验公式进行计算。

$$\epsilon_{we} = 80 \cdot \left(\frac{h_{ex}}{B_{ex}}\right)^2 + \sigma_{coef_we} \cdot \frac{\sigma_t}{E_{mod}}$$

$$\epsilon_{cb} = -40 \cdot \left(\frac{h_{ex}}{B_{ex}}\right)^2 + \sigma_{coef_cb} \cdot \frac{\sigma_t}{E_{mod}}$$

$$\sigma_{coef_we} = 1.5$$

$$\sigma_{coef_cb} = -3$$

屈曲判别极限的调节参数

CTOOL中GSM模块中有CenterBuckleTuning和CenterBuckleTuning两张表，这两张表可以用来对屈曲判别标准的中浪和双边浪极限值进行调整。

这两个表对于每个机架F1到F7分别有两个参数。一个是比例系数multiplier，作为乘数而存在，另一个是补偿值Offset，作为加数而存在。

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（we / cb）
- 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_Atn_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 (Flow Stress)

凸度改变削弱因子的影响因素中，流变应力是一个较难理解的概念。流变应力指的是材料继续进行塑性变形所需的瞬时应力值，简而言之，流变应力是驱动金属流动的力，流变应力的单位是MPa。流变应力的值介于屈服强度和抗拉强度之间，是关于塑性应变的函数，它可以表示如下，单位均为MPa。

$$Y_f = K\epsilon^n$$

其中 Y_f 为流变应力，单位是MPa。 ϵ 是真实应变。 K 为延伸系数，单位是MPa。 n 为加工硬化指数。因此，流变应力同样可以表示为在特定应变条件下保持塑性变形的应力。以下因素对流变应力有影响。

- 化学成分
- 纯净度
- 晶体结构
- 金相组成
- 出口组织
- 晶粒组织
- 热处理

流变应力是韧性材料疲劳失效的一个重要参数。在变化的载荷作用下，裂纹的传播扩展导致金属的疲劳失效，尤其是周期变化的载荷作用影响更甚。金属流变应力和裂纹的传播速度成反比例关系。

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_infl_cof <= 0.0 )
    {
        pcPceIZFSPassD = pcFSPassD;
    }
    // ...
    pcFSPassD = ( cFSPassD* )pcFSPassD->next_obj;
}
```

首先预设定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。

凸度最终分配的说明

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

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。

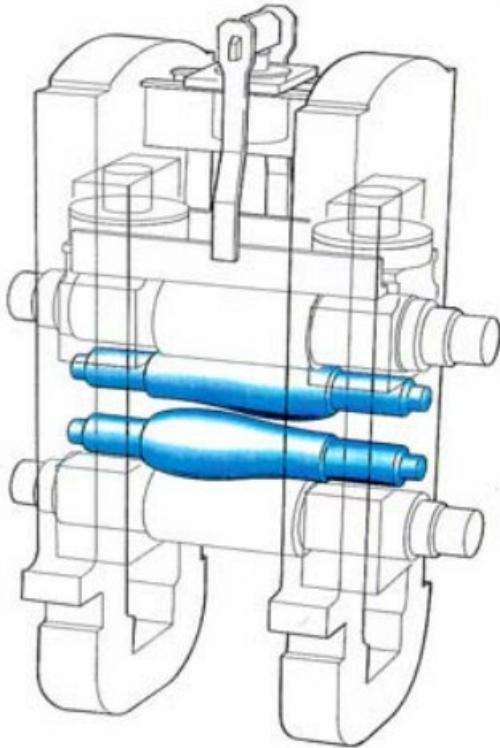
dlt是否需要除以2

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。

CVC辊形设计原理

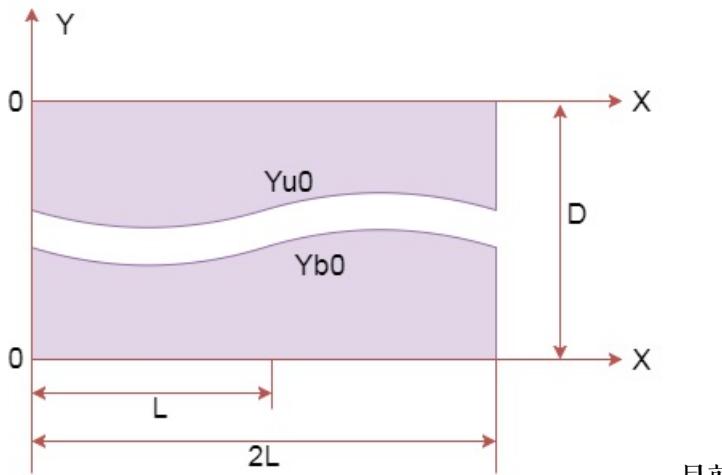


CVC的曲线函数一般是一个最高次项为奇数的多项式。如下式所示。其中 $y_u(x)$ 和 $y_b(x)$ 分别表示上辊和下辊的辊形曲线。 a_0 到 a_n 为多项式系数，或称为辊形系数。 n 为曲线的最高次幂， $2L$ 为辊身长度， s 为窜辊值。

$$y_u(x) = \sum_{i=0}^n a_i (x - s)^i$$

$$y_b(x) = \sum_{i=0}^n a_i (2L - x - s)^i$$

辊形曲线的建立



目前三次曲线用的比

较多，下面以三次曲线举例，上辊的辊形曲线如下。

$$y_{u0}(x) = a_0 + a_1 x^1 + a_2 x^2 + a_3 x^3$$

同时也可以确认下辊的辊形曲线。其中L为工作辊辊身长度的一半。

$$y_{b0}(x) = a_0 + a_1(2L - x)^1 + a_2(2L - x)^2 + a_3(2L - x)^3$$

若考虑窜辊s，则上下辊的辊形曲线需要稍做修改。考虑窜辊的上辊辊形曲线如下。

$$y_{u1}(x) = a_0 + a_1(x - s)^1 + a_2(x - s)^2 + a_3(x - s)^3$$

考虑窜辊的下辊辊形曲线如下。

$$y_{b1}(x) = a_0 + a_1(2L - x - s)^1 + a_2(2L - x - s)^2 + a_3(2L - x - s)^3$$

以上是围绕辊形实际形状几何条件建立的曲线方程。

轧辊等效凸度和窜辊的关系

辊形曲线已知，那么在辊身长度方向上的辊缝大小可以通过如下公式获得。其中D为两辊径之间的距离。

$$g(x) = D - y_{u1}(x) - y_{b1}(x)$$

$$g(x) = 2(2a_3(s-L) - a_2)(x-L)^2 + 2(a_3(s-L)^3 - a_2(s-L)^2 + a_1(s-L) - a_0$$

整个CVC轧辊形成的辊缝等效凸度可以用如下公式表示。

$$C_w = -(g(L) - g(0)) = g(0) - g(L)$$

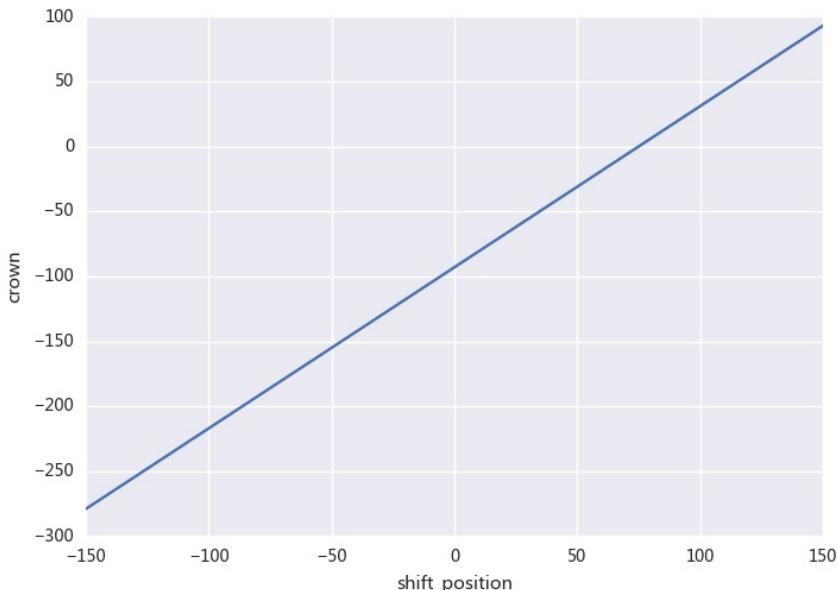
$$C_w = 6a_3L^2s - (6a_3L + 2a_2)L^2$$

可以看出辊缝凸度和窜辊之间满足线性关系 $y = mx + n$ 。其中：

$$m = 6a_3L^2$$

$$n = (6a_3L + 2a_2)L^2$$

并且 m 和 n 只与 a_2 、 a_3 相关。在实际生产过程中，通过调整窜辊位置就可以线性地调节凸度。



辊形参数的设计

在设计CVC辊形时，窜辊行程和设备有关，一般为 $[-150, +150]$ ，所以主要的设计手段是调整辊缝凸度的最大值 C_{max} 和最小值 C_{min} 。当辊缝凸度控制范围 $[C_{min}, C_{max}]$ ，窜辊行程范围 $[S_{min}, S_{max}]$ 已知时，将相应凸度极值和窜辊极值代入 C_w 的公式解二元一次方程，可以获得 a_2 、 a_3 的解。

$$a_2 = -\frac{C_{min}}{2L^2} + 3a_3(S_{min} - L)$$

$$a_3 = \frac{C_{min} - C_{max}}{6L^2(S_{min} - S_{max})}$$

由于 a_1 和板形无关，确定 a_1 的值有很多种设计思路，比如两端辊径差最小、中部辊径差最小、轴向力最小等。以轴向力最小的设计思路为例。 a_1 在轧辊设计的过程中考虑轧辊轴向力的边界条件。轧辊轴向力的计算如下：

$$F = \int_{y_{u1}(L-b)}^{y_{u1}(L+b)} pdy = p(y_{u1}(L+b) - y_{u1}(L-b))$$

$y_{u1}(L+b)$ 和 $y_{u1}(L-b)$ 结合对应公式代入后，取轴向力为0，则可以获得 a_1 的表达式。

$$a_1 = -2a_2L - 3a_3L^2 - a_3b^2$$

其中 b 为带钢宽度中心距离宽度边部的长度。将相应 a_2 、 a_3 的值代入求得 a_1 的值。当以轧辊轴线作为横坐标轴时， a_0 实际上就为轧辊的名义半径或中心半径 $D_0/2$ 。 a_1 中存在带钢宽度影响项，宽度值的选取采用经验公式，一般取辊身长度的65%。

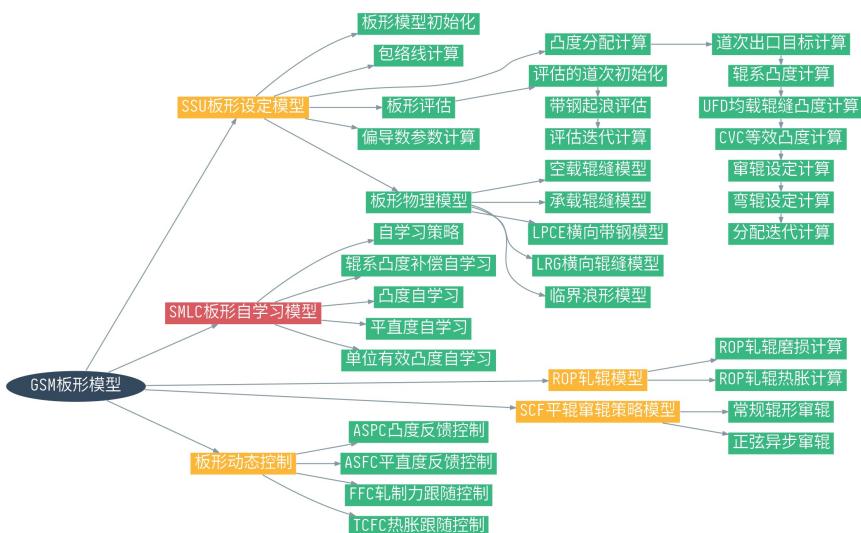
板形自学习总览

SMLC (Shape Model Learning Calculation) 模型的主要作用是，利用带钢FM出口实际测量的凸度和平直度，用实测值和计算值的偏差对下一块带钢的板形设定做调整和修正，提高板形模型的控制精度。

板形自学习可以分为三大块：

- 轧系自学习
- 凸度自学习
- 平直度自学习

板形自学习在整个板形模型中的定位如下图中红色节点所示。



平直度自学习

王宇阳整理

平直度自学习程序梳理

Piece Director将板形设定信息和检测数据、PDI等请求发送给板形反馈调度，由 `cSfbDispatch::Receive_Work` 接受板形设定参数并处理。之后触发 `cShapeFeedbackD::Main` 主函数的运行，主函数中调用运行 `cShapeFeedbackD::Flt_Adpt`，进而调用 `cTargtD::Flt_Adpt` 函数进行平直度自学的计算。

`cTargtD::Flt_Adpt`

平直度自学习的主函数是 `cTargtD` 类中的 `cTargtD::Flt_Adpt` 函数。此函数有6个输入量和两个输出量。函数的定义如下所示。

```
void cTargtD::Flt_Adpt(
    const bool ssu_ok,                                     // [-] SSU model valid
    const bool ssu_grnt,                                    // [-] SSU model grant
    const bool bnd_enab,                                   // [-] roll bending sys
    const float flt_mea,                                  //      indicator
    const float force_bnd_off,                           // [mm/mm_in/in_mm/mm]
    const float dbnd_dflt,                               // [mton/etton_kn] oper
    const float dbnd_dfltn,                             //      bending force of
    const float flt_err,                                // [mton/etton_kn] IN/O
    const float flt_vrn)                                // [mton/etton_kn] IN/O
```

```
    UT flatness vernier  
        ) const
```

其中 `ssu_ok` 和 `ssu_grnt` 表示SSU板形设定模型的运行状态，`bnd_enab` 表示弯辊的使用状态。`flt_mea` 是平坦度的测量值，`force_bnd_off` 为弯辊力的干预补偿量，`dbnd_dflt` 是弯辊对平坦度的偏导数。输出为平坦度的偏差 `flt_err` 和平坦度的自学习值 `flt_vrn`。

函数分为两个步骤，先计算平直度偏差的临时量，再计算自学习的值。平直度偏差的临时量用以下变量名表示。

```
float flt_err_buf ( 0.0 );
```

第一个步骤的代码如下所示，当模型运行状态不正常或者弯辊设定不能使用时，直接将平直度偏差的临时量设定为0。当模型状态正常，则计算平直度的偏差量。

```
if ( bnd_enab &&  
    ssu_grnt &&  
    ssu_ok )  
{  
    flt_err_buf = ( flt - flt_mea ) * Physcon.i_units * dbnd_dflt;  
}  
else if ( !ssu_grnt ||  
         !bnd_enab )  
{  
  
    flt_err_buf = 0.0;  
}
```

注意计算公式中：`flt` 为目标平坦度，`flt_mea` 为测量的平坦度。本处计算 `flt_err_buf` 为弯辊力的大小，即通过 `dbnd_dflt` 将平坦度偏差转化为弯辊力偏差。

第二个步骤计算平直度自学习值。在第二个步骤当中，平坦度自学习定义了一个变量，`flt_err_thrshld`，表示平坦度自学习的门槛，这个变量位于 `cfg_targt.txt`，默认值为100kN。当模型条件正常时，如果平直度偏差小于这个值，则只记录偏差，不进行自学习，如下代码所示。

```
if ( fabs(flt_err_buf) <= pcTargt->flt_err_thrshld )  
    flt_err = flt_err_buf;
```

当模型条件正常时，如果平直度偏差大于 `flt_err_thrshld`，满足 `flt_err_thrshld` 最小门槛条件，且操作工又没有调整错误趋势，则进行平直度自学习。如下代码所示。

```
if ( ( flt_err_buf > pcTargt->flt_err_thrshld &&
       force_bnd_off >= -pcTargt->opr_mx_wrng_corr ) ||
     ( flt_err_buf < -pcTargt->flt_err_thrshld &&
       force_bnd_off <= pcTargt->opr_mx_wrng_corr ) )
{
    ....
}
```

在这里，`opr_mx_wrng_corr` 表示操作工对弯辊力做修正的门槛条件，`opr_mx_wrng_corr` 同样定义于 `cfg_targt.txt`，默认值为 100kN。将 `force_bnd_off` 和 `opr_mx_wrng_corr` 一起纳入判断条件，说明当操作工调整过量导致浪形出现时，不触发平直度自学习的计算。

在计算平直度自学习前，先进行限幅检查。

```
flt_err_buf = cMathUty::
    Clamp( flt_err_buf,
           pcTargt->flt_err_lim[ min1 ],
           pcTargt->flt_err_lim[ max1 ] );
```

`flt_err_lim` 是平直度偏差的极限值，定义于 `cfg_targt.txt`，默认值为 -250kN 和 +250kN。

限幅检查之后，计算平直度自学习值 `flt_vrn`。

```
flt_vrn =
    flt_vrn + flt_err_buf *
    ( pcTargt->flt_vrn_i_gn / 2.0F + pcTargt->flt_vrn_p_gn) +
    flt_err * ( pcTargt->flt_vrn_i_gn / 2.0F - pcTargt->flt_vrn_p_gn);
```

`flt_vrn` 的计算采用 PI 方式，偏差的选取选用的本块 `flt_err_buf` 和上一块的 `flt_vrn`、`flt_err`，按 PI 进行分配计算。`flt_vrn_p_gn` 和 `flt_vrn_i_gn` 分别为比例参数和积分参数，在配置文件 `cfg_targt.txt` 中定义：

```
prf_vrn_rm_i_gn = 0.6;
prf_vrn_rm_p_gn = 0.3;
```

这样的配置参数，其实公式后面的 `flt_err` 是没有用上的，因为后面一项相减后结果为0。平坦度自学习结束以后，对自学习值进行限幅值比较及存储，供下一块PI控制使用。

最后，对平直度自学习值 `flt_vrn` 限幅并更新平直度偏差 `flt_err`。`flt_vrn_lim` 为平直度自学习值的极限，在`cfg_targt.txt`中定义，默认值为-800kN和+800kN。

```
flt_vrn = cMathUty::  
    Clamp( flt_vrn,  
        pcTargt->flt_vrn_lim[ minl ],  
        pcTargt->flt_vrn_lim[ maxl ] );  
  
flt_err = flt_err_buf;
```

在第二个步骤当中，如果模型条件和弯辊力使用不正常，则衰减平直度自学习值 `flt_vrn` 并记录平直度偏差 `flt_err`。

```
else if ( !ssu_grnt ||  
        !bnd_enab )  
{  
    //-----  
    // Roll bending system was not under model control, bleed-off  
    // flatness vernier.  
    //-----  
    flt_vrn = flt_vrn * pcTargt->flt_vrn_bled;  
    //-----  
    // Restrict the flatness vernier to within limits.  
    //-----  
    flt_vrn = cMathUty::  
        Clamp( flt_vrn,  
            pcTargt->flt_vrn_lim[ minl ],  
            pcTargt->flt_vrn_lim[ maxl ] );  
    //-----  
    // Update the flatness error.  
    //-----  
    flt_err = flt_err_buf;  
}
```

在这里，`flt_vrn_bled` 为平直度自学习的衰减系数，默认值为0.9。无论模型条件是否正常，最后都要对平直度自学习值限幅检查并记录和更新平直度偏差量。

平直度自学习配置参数

和平直度自学习相关的配置参数名称、位置、默认值和含义如下表所示。| 配置参数 | 所在配置文件位置 | 默认值 | 含义 |

opr_mx_wrng_corr	cfg_targt.txt	100kN	平直度偏差的门槛条件
flt_err_thrshld	cfg_targt.txt	-250kN和+250kN	平直度偏差的极限值
cfg_targt.txt	0.6	PI计算的积分参数	
cfg_targt.txt	0.3	PI计算的比例参数	
flt_vrn_bleed	cfg_targt.txt	0.9	平直度自学习的衰减系数
cfg_targt.txt	-800kN和+800kN	平直度自学习量的极限值	

平直度测量的计算

在 `cTargetD::Flt_Adpt` 中计算平直度的偏差需要用到测量平直度的值 `flt_mea`，那么平直度的测量值 `flt_mea` 在板形模型中是如何计算的呢？

`shapefeedback.cxx` 中可以找到答案，如下代码所示。

```
meas_flt     = pcFlatnessSensorD->strn;
meas_flt_vld = pcFlatnessSensorD->strn_ok;
```

测量平直度的值来自 `cFlatnessSensorD` 实例对象的 `strn` 属性。

`strn` 的计算实现在 `cFlatnessSensorD::Init` 初始化函数当中，如下代码所示。

```
strn =
    ( pcSFXFltFbk->state.fib_len[ 0 ] + pcSFXFltFbk->state.fib_len[ num_fib - 1
] ) *
    (float) pow( ( pcExPceD->width - 2.0F * prf_edg_dist ) /
        ( fabs( pcSFXFltFbk->state.fib_pos[ 0 ] ) +
        fabs( pcSFXFltFbk->state.fib_pos[ num_triangl - 1 ] ) ), 2.0F ) / Physcon.i_units;
```

整个 `strn` 可以分为两个部分相乘。一个部分是实际的逻辑计算，另一部分为系数，如下公式所示。

$$strn = \frac{L_{OS} - L_M + L_{DS} - L_M}{2 \times 10^3} \times \left(\frac{B_{crn}}{B_{flt}} \right)^2$$

其中 \$L\{OS\}\$、\$L\{M\}\$、\$L\{DS\}\$ 分别为操作侧、中部、传动侧的带钢纤维长度。\$B_{crn}\$ 是实际出口带钢去除边部距离（一般是40mm）的宽度，\$B_{flt}\$ 是操作侧纤维测量点和传动侧纤维测量点的距离。

当然，在 `cFlatnessSensorD::Init` 的实现当中，还考虑了缺乏操作侧或传动侧平直度数据时，`strn` 的计算方式。

注意这里的 `fib_len` 指的是边部测量的纤维长度相对于中部纤维长度的差值。

类似于 `fib_len` 这样和平直度检测有关的参数，如纤维长度、纤维位置等，在 `sfxltfbk.hxx` 的 `sSFXFltFbk` 结构中定义。如下代码所示。

```

typedef struct
{
    float      fib_len[ num_fib ];           // [i-units] strip relativ
e fiber                                         // length
                                                // [0] = Rho channe
ls 1 - 4                                         // [1] = Rho channe
ls 2 - 4                                         // [2] = Rho channe
ls 3 - 4                                         // [4] = Rho channe
ls 5 - 4                                         // [5] = Rho channe
ls 6 - 4                                         // [6] = Rho channe
ls 7 - 4                                         // [7] = Rho channe
    float      fib_len_sd[ num_fib ];          // [i-units] strip relativ
e fiber                                         // length standard devi
ation
                                                // [0] = Rho channe
ls 1 - 4                                         // [1] = Rho channe
ls 2 - 4                                         // [2] = Rho channe
ls 3 - 4                                         // [4] = Rho channe
ls 5 - 4                                         // [5] = Rho channe
ls 6 - 4                                         // [6] = Rho channe
ls 7 - 4                                         // [7] = Rho channe
    float      fib_pos[ num_triangl ];          // [mm_in] strip fiber pos
ition

```

```

ls 1 - 4                                // [0] = Rho channe
ls 2 - 4                                // [1] = Rho channe
ls 3 - 4                                // [2] = Rho channe
ls 5 - 4                                // [4] = Rho channe
ls 6 - 4                                // [5] = Rho channe
ls 7 - 4                                // [6] = Rho channe
    float      fib_pos_sd[ num_triangl ];   // [mm_in] strip fiber pos
    ition
                                         // standard deviation
                                         // [0] = Rho channe
ls 1 - 4                                // [1] = Rho channe
ls 2 - 4                                // [2] = Rho channe
ls 3 - 4                                // [4] = Rho channe
ls 5 - 4                                // [5] = Rho channe
ls 6 - 4                                // [6] = Rho channe
ls 7 - 4
    float      qtr_bckl;                  // [i-units] strip quarter
    buckle
    float      qtr_bckl_sd;              // [i-units] strip quarter
    buckle
    float      lvl[ num_lvl ];          // [i-units] strip level
                                         // [0] = Rho channels
1 - 5
ls 1 - 7                                // [0] = Rho channe
                                         // [1] = Rho channe
ls 2 - 6
    float      lvl_sd[ num_lvl ];        // [i-units] strip level s
    tandard
                                         // deviation
                                         // [0] = Rho channe
ls 1 - 7                                // [1] = Rho channe
ls 2 - 6
    bool      fib_len_ok[ num_fib ];     // [-] strip relative fiber
                                         // length validity indi
cator
                                         // [0] = Rho channe
ls 1 - 4                                // [1] = Rho channe
ls 2 - 4                                // [2] = Rho channe
ls 3 - 4

```

```

ls 5 - 4                                // [4] = Rho channe
ls 6 - 4                                // [5] = Rho channe
ls 7 - 4                                // [6] = Rho channe
ls 7 - 4
    bool      fib_pos_ok[ num_triangl ];   // [-] strip fiber position
                                                // validity indicator
                                                // [0] = Rho channe
ls 1 - 4                                // [1] = Rho channe
ls 2 - 4                                // [2] = Rho channe
ls 3 - 4                                // [4] = Rho channe
ls 5 - 4                                // [5] = Rho channe
ls 6 - 4                                // [6] = Rho channe
ls 7 - 4
    bool      qtr_bck1_ok;                // [-] quarter buckle vali
dity
    bool      lvl_ok[ num_lvl ];         // [-] strip level validity
                                                // indicator
                                                // [0] = Rho channe
ls 1 - 7                                // [1] = Rho channe
ls 2 - 6
} sSFXFltFbk;

```



`fib_len` 代表不同位置的纤维和中心纤维的相对长度差。`fib_pos` 代表纤维的横向位置（以宽度中心为零点）。`lvl` 代表对称纤维的两侧偏差。

需要注意的是，以上这些量的计算是在 `cSfbDispatch::Load_Feedback_Objects` 中完成的。`fib_len` 的计算分头部和中部。

```

pcFlatnessSensorD->pcSFXFltFbk->state.fib_pos[0] =
    0.5F*pcSched->pcFMD->state.f_Width - mill().data().flatEdgeDist;

pcFlatnessSensorD->pcSFXFltFbk->state.fib_pos[num_triangl - 1] =
    -pcFlatnessSensorD->pcSFXFltFbk->state.fib_pos[0];

pcFlatnessSensorD->pcSFXFltFbk->state.fib_len[0] =
    pcSched->pcFMD->state.f_Mid_Flatness + 0.5F* pcSched->pcFMD->s
state.f_Mid_Level;

pcFlatnessSensorD->pcSFXFltFbk->state.fib_len[num_fib - 1] =

```

```
pcSched->pcFMD->state.f_Mid_Flatness - 0.5F* pcSched->pcFMD->state.f_M  
id_Level;  
  
pcFlatnessSensorD->pcSFXFltFbk->state.lvl[0] = pcSched->pcFMD->state.f_Mid_Le  
vel;
```

以上代码涉及到的 `flatEdgeDist`，和 `profEdgeDist` 一起在 `cfg_mill.txt` 中设定，默认值二者均为40微米。

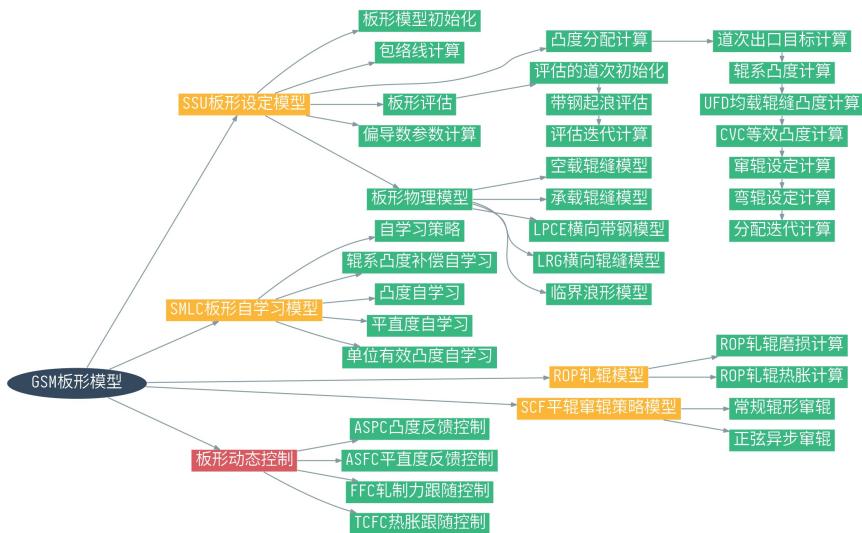
以带钢头部为例（带钢中段与其类似），实际上 `f_Head_Flatness` 为对称平直度，`f_Head_Level` 为非对称平直度。从如下 `write_ssu_log.cxx` 中的代码可以看出。

```
fprintf( cfPtr, " Hd_Sym: %4d", (int)pcSched->pcFMD->state.f_Head_Flatness);  
fprintf( cfPtr, " ASym: %4d", (int)pcSched->pcFMD->state.f_Head_Level);  
fprintf( cfPtr, " Vld: %s", pcSched->pcFMD->state.b_Head_Flatness_Valid ? "T":  
"F");
```

板形动态控制

板形动态控制是一级的功能，用于对身部的带钢板形进行增益控制。

板形动态控制在整个板形模型中的定位如下图红色节点所示。



板形动态控制可以分为四个部分。

1. 凸度动态控制ASPC
2. 平直度动态控制ASFC
3. 轧制力跟随控制FFC
4. 热胀跟随控制TCFC

凸度动态控制

ASPC的全称是Automatic Strip Profile Control，ASPC的功能是用来计算弯辊力相对实测凸度反馈值的增益，用于保持全长方向上的凸度。F7是凸度动态控制的基础执行机构。

平直度动态控制

ASFC的全称是Automatic Strip Flatness Control，ASFC的功能是用来计算弯辊力相对实测平直度反馈的增益，用于保持全长方向上的平直度。F7是平直度动态控制的唯一执行机构。

轧制力跟随控制

FFC的全称是Force Following Control，轧制力跟随控制功能是对弯辊力相对轧制力的变化进行动态调整，降低轧制力变化对轧辊辊缝的影响。弯辊力的变化通过过程增益函数进行计算。

热胀跟随控制

TCFC的全称是Thermal Crown Following Control，热胀跟随控制的作用是通过弯辊力的调整，降低热胀变化对轧辊辊缝的影响。辊系凸度变化在过程转换函数中被转换为等效的弯辊力变化值。

板形模型的参数和日志

板形模型的参数和日志，是工艺人员与模型沟通的手段和方式。工艺人员通过修改参数来调整模型的运行，模型生成日志反馈有用的运算信息。板形模型的参数是输入，板形模型是计算和处理，板形模型的日志是输出。

模型在生产每卷带钢每个计算周期形成日志文件，按照不同的模型模块进行划分。Log日志为日志系统的产物，日志中变化的参数直接从C++代码中用输出流语句输出。

板形模型的参数可以分为C-Tool参数和CFG配置文件两大类。

C-Tool

C-Tool直接读取和写入数据库数据。C-Tool数据的修改，在put后直接生效，不需要重启。

CFG配置文件

CFG配置文件为脚本式的参数，由TMEIC自己写的Parser进行解析。CFG配置文件一般在Model Browser中进行热修改。脚本中的静态修改仅在进程重启后生效。

需要注意的是重启进行后，Model Browser中的热修改全部失效。

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

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	WR	Dlt	Pce	Dlt	WR	SLFG	SPRP
Vern	Offs	WR_Gap	[mm]	BR_Gap	[mm]	WR	WR
	[mm]					Offs	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		UFD PU		Std Ex Strn [mm/mm]	
		Prf [mm/mm]	Prf [mm]	Prf [mm/mm]	Prf [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 [mm]	CrnErr ok [mm]	Pce [mm]	Wr [mm]	WR [mm]	CrnVrn ok [mm]	GapErr ok [mm]	CrnVrn ok [mm]	CrnOff ok [mm]	ErrRM ok [mm]	Prf [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。

pcTargtD->pcTargt->prf_vrn_sel_flag默认值为true，在cfg_targt.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_flrq.txt;           ! create and configure LRG object
include = SSU$CONFIG:cfg_fsppass.txt;         ! create and configure FSPPass object
include = SSU$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$CONFIG:cfg_shapeselect.txt;      ! create and configure static SHAPE SETUP object
include = SSU$CONFIG:cfg_shapefeedback.txt;    ! create and configure static SHAPEFEDBACK object
include = SSU$CONFIG:cfg_targt.txt;            ! create and configure static TARGT object
include = SSU$CONFIG:cfg_sfobobs.txt;          ! create and configure static Obs object
```

FSU相关的配置参数如下。

```
include = MDS$CONFIG:cfg_fmill.txt;           ! create the remainder of static mil object for FM
include = MDS$CONFIG:cfg_ftmpgrad.txt;         ! create and configure a tmpgradcfg object
include = FSHARED$CONFIG:cfg_std.txt;          ! create the static stand objects
include = MDS$CONFIG:cfg_fmtr.txt;             ! create the static motor objects
include = MDS$CONFIG:cfg_frollbite.txt;        ! create the static roll bite object
include = GSM$CONFIG:cfg_sensor.txt;           ! create the static sensor objects
include = GSM$CONFIG:cfg_fzone.txt;            ! create and configure the zone objects
include = FSHARED$CONFIG:cfg_map.txt;          ! create the 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,
(SMS)             actrtyp_bend,
actrtyp_none;          ! roll CVC shifting system
                      ! roll bending system
                      ! 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_coeff = 4, 0.000, 0.0000, 0.0000, 0.0000;

!!@ND-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 limitaion
    //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 = cTarget;
cTarget = target;
// 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 vernierbl
eed-off

// 平直度自学习极限范围
    flt_vrn_lim    = 2,                                ! [kN] target flatnessvernier l
imits
                -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 controlloo
p integral gain
    flt_vrn_p_gn   = 0.3;                            ! [-] target flatness controllo
op proportional gain

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

// 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 li-
mits
                    -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 int-
egral gain
prf_vrn_rm_p_gn = 0.2;
! [-] target profile vernier (re-predicted - measured) control loop pro-
portional 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 integr-
al gain

// 平直度自学习，学习的临界点
flt_err_thrshld = 100;
! [kN] Flatness errorthreshold minimum for flatness 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 direction and still doflatn-
ess 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 forthi-
nkness 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(fa-
lse=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
lse=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;       ! [-] Work roll offset selection
flat (false= slfg, true = slfg+sprp)
!@2ND(LC061) end
    end;
end;
```



C-Tool 板形GSM

和板形控制有关的C-Tool参数。

GSM调节参数总览

GSM的C-TOOL表中的参数如下表所示。

参数名	所在CTOOL的表（忽略前缀）	初始参数变量名
UFD调节值	UFD_Stand(Tuning)Multiplier.xlsb	psSPRP->ufd_mult
初始弯辊力	NormalBendForce.xlsb	psSPRP->bend_nom
窜辊机构最小软极限调节值	ShiftActuatorLimit.xlsb	psSPRP->min_accu_lmt
窜辊机构最大软极限调节值	ShiftActuatorLimit.xlsb	psSPRP->max_accu_lmt
弯辊机构最小软极限调节值	BendingActuatorLimit.xlsb	psSPRP->min_bend_lmt
弯辊机构最大软极限调节值	BendingActuatorLimit.xlsb	psSPRP->max_bend_lmt
边浪调节因子	WavinessTuning.xlsb	psSPRP->wav_mult
边浪补偿值	WavinessTuning.xlsb	psSPRP->wav_ofs
中浪调节因子	CenterBuckleTuning.xlsb	psSPRP->cb_mult
中浪补偿值	CenterBuckleTuning.xlsb	psSPRP->cb_ofs
综合凸度补偿系数	WorkCrownOffsetTuning.xlsb	psSLFG->wr_crn_off

UFD multiplier调节值

调节参数在表中的位置

UFD凸度multiplier调节值位于GSM的UFD_Stand(Tuning)Multiplier.xlsb当中。按钢种组分类，针对每个宽度区间和厚度区间、每个机架，有一个UFD凸度multiplier调节值，供工艺人员进行调节。

如下图所示。

Width index	Gauge index	F1	F2	F3	F4	F5	F6	F7
		ufd-mult	ufd-mult	ufd-mult	ufd-mult	ufd-mult	ufd-mult	ufd-mult
0	800<= w < 900	0 1.00<=h<1.15	1	1	1	1	1	1
	1 1.15<=h<1.30	1	1	1	1	1	1	1
	2 1.30<=h<1.50	1	1	1	1	1	1	1
	3 1.50<=h<1.70	1	1	1	1	1	1	1
	4 1.70<=h<1.85	1	1	1	1	1	1	1
	5 1.85<=h<2.05	1	1	1	1	1	1	1
	6 2.05<=h<2.25	1	1	1	1	1	1	1
	7 2.25<=h<2.50	1	1	1	1	1	1	1
	8 2.50<=h<2.75	1	1	1	1	1	1	1
	9 2.75<=h<3.00	1	1	1	1	1	1	1
	10 3.00<=h<3.40	1	1	1	1	1	1	1
	11 3.40<=h<4.00	1	1	1	1	1	1	1
	12 4.00<=h<5.00	1	1	1	1	1	1	1
	13 5.00<=h<6.00	1	1	1	1	1	1	1
	14 6.00<=h<7.50	1	1	1	1	1	1	1
	15 7.50<=h<9.00	1	1	1	1	1	1	1
	16 9.00<=h<10.50	1	1	1	1	1	1	1
	17 10.50<=h<11.50	1	1	1	1	1	1	1
	18 11.50<=h<12.70	1	1	1	1	1	1	1
	19 12.70<=h<14.00	1	1	1	1	1	1	1
	20 (spare)	1	1	1	1	1	1	1
	21 (spare)	1	1	1	1	1	1	1
	22 (spare)	1	1	1	1	1	1	1
	23 (spare)	1	1	1	1	1	1	1
	24 (spare)	1	1	1	1	1	1	1
	25 (spare)	1	1	1	1	1	1	1
	26 (spare)	1	1	1	1	1	1	1
	27 (spare)	1	1	1	1	1	1	1

UFD调节参数的作用

这个参数的作用是对UFD凸度的计算进行补偿和修正。同时影响到其它所有涉及到UFD凸度的工艺量计算，或者说影响到其它以UFD凸度为参数的函数，如单位宽度轧制力、弯辊力、带钢-工作辊凸度的计算等。

UFD调节参数参与的过程

初始化

首先在cShapeSetupD::Init()的初始化过程，从sSPRP结构中将此参数映射的变量ufd_mult赋值给pcFSSD对象中的变量ufd_mult（两者同名）。

在cShapeSetupD::Init()初始化过程中，有一个cUFDD::Init()函数，pcFSStdD对象的ufd_mult作为最后一个参数变量传入，对相关的cUFDD对象进行了初始化。pcFSStdD对象的ufd_mult被赋值给了cUFDD对象的ufd_modifier。ufd_modifier是cUFDD对象的一个属性。之后的各种计算，此参数均以ufd_modifier的变量形式进行参与。

UFD调节参数参与的函数总览

cUFDD::Init()初始化之后，ufd_modifier参与计算的函数如下表所示。

受参数影响的函数	函数作用
cUFDD::Prf	计算辊缝凸度
cUFDD::Bnd_Frc	计算弯辊力
cUFDD::Frc_PU_Wid	计算单位宽度轧制力
cUFDD::Pee_WR_Crn	单独计算带钢-工作辊凸度
cUFDD::Crns	计算综合轧辊凸度

cUFDD::Prf

cUFDD::Prf此函数是用来计算UFD辊缝凸度的函数，输入量为单位宽度轧制力、弯辊力、带钢-工作辊凸度、工作辊-支撑辊凸度。在函数中，ufd_modifier介入的地方如下所示。

```

float cUFDD::Prf(
    const float force_pu_wid, // [kn/mm_mton/mm_eton/in] roll
    ling // force per unit piece width
    th // const float force_bnd, // [kn_mton_eton] roll bending
    force // const float pce_wr_crn, // [mm_mm_in] piece to work roll
    11 // stack crown
    up // const float wr_br_crn // [mm_mm_in] work roll to back
    up // const float ufd_modifier // [-] UFD (family,grt,wrt)t
    uning modifier
        ) const // [mm_mm_in] UFD roll gap profile
    file

```

```

{ // Begin of PRF function

    return
        ( b_cof[ 0 ] * force_pu_wid +
        b_cof[ 1 ] * pow( force_pu_wid, float(1.5) ) +
        b_cof[ 2 ] * pce_wr_crn +
        b_cof[ 3 ] * wr_br_crn * force_pu_wid +
        b_cof[ 4 ] * wr_br_crn * pow( force_pu_wid, float(1.5) ) +
        b_cof[ 5 ] * force_bnd +
        b_cof[ 6 ] * force_bnd * force_pu_wid +
        b_cof[ 7 ] * force_bnd * pow( force_pu_wid, float(2.0) ) +
        b_cof[ 8 ] * wr_br_crn +
        b_cof[ 9 ] * force_pu_wid +
        b_cof[ 10 ] * force_bnd +
        b_cof[ 11 ] * force_pu_wid +
        b_cof[ 12 ] * pow( force_pu_wid, float(1.5) ) +
        b_cof[ 13 ] * force_bnd +
        b_cof[ 14 ] * pce_wr_crn +
        b_cof[ 15 ] * pce_wr_crn +
        b_cof[ 16 ] +
        b_cof[ 17 ] ) * ufd_modifier;

} // End of PRF function

```

ufd_modifier是作为一个乘数，乘到UFD辊缝凸度的返回值上。在包络线生成、分配计算、评估计算等阶段，均运用到了cUFDD::Prf函数。当ufd_modifier或ufd调整参数增大，cUFDD::Prf的计算结果增大；当ufd_modifier或ufd调整参数减小，cUFDD::Prf的计算结果减小。

cUFDD::Bnd_Frc

cUFDD::Bnd_Frc此函数用来计算弯辊力，输入量为UFD辊缝凸度、单位宽度轧制力、带钢-工作辊凸度、工作辊-支撑辊凸度、弯辊力软极限（弯辊力软极限的调节系数在这里介入），输出量为计算的最终弯辊力force_bnd和实际需要的弯辊力force_bnd_des。如下所示。

```

void cUFDD::Bnd_Frc(
    const float ufd_prf, // [mm_mm_in] UFD roll gap pro
file
    const float force_pu_wid, // [kn/mm_mton/mm_eton/in] rol
ling
    const float pce_wr_crn, // [mm_mm_in] piece to work ro
th
    ll // stack crown

```

```

        const float wr_br_crn,           // [mm_mm_in] work roll to bac
    kup
        const float force_bnd_lim[2],   // roll stack crown
        float& force_bnd,             // [mton_eton_kn] roll bending
        float& force_bnd_des         // soft limit (min,max)
    force                                         // [mton_eton_kn] roll bending
        ) const                                     // [mton_eton_kn] desired roll
                                                // bending force

{   // Begin of BND_FRC function

//-----
// Calculate the roll bending force.
//-----
force_bnd_des =
( ufd_prf / ufd_modifier -
b_cof[ 0 ] * force_pu_wid -
b_cof[ 1 ] * pow( force_pu_wid, float(1.5) ) -
b_cof[ 2 ] * pce_wr_crn -
b_cof[ 3 ] * wr_br_crn * force_pu_wid -
b_cof[ 4 ] * wr_br_crn * pow( force_pu_wid, float(1.5) ) -
b_cof[ 8 ] * wr_br_crn -
b_cof[ 9 ] * force_pu_wid -
b_cof[ 11 ] * force_pu_wid -
b_cof[ 12 ] * pow( force_pu_wid, float(1.5) ) -
b_cof[ 14 ] * pce_wr_crn -
b_cof[ 15 ] * pce_wr_crn -
b_cof[ 16 ] -
b_cof[ 17 ] ) /
( b_cof[ 5 ] +
b_cof[ 6 ] * force_pu_wid +
b_cof[ 7 ] * pow( force_pu_wid, float(2.0) ) +
b_cof[ 10 ] +
b_cof[ 13 ] );

//-----
// Restrict the roll bending force to within soft limits.
//-----
force_bnd = cMathUty::
    Clamp( force_bnd_des,
           force_bnd_lim[min1],
           force_bnd_lim[max1] );

}   // End of BND_FRC function

```

在函数中，首先计算实际需要的弯辊力，如下图所示。在这个过程中，ufd_modifier作为除数参与弯辊力计算，从UFD辊缝凸度中除去。

```
force_bnd_des =
```

```

( ufd_prf / ufd_modifier -
b_cof[ 0 ] * force_pu_wid -
b_cof[ 1 ] * pow( force_pu_wid, float(1.5) ) -
b_cof[ 2 ] * pce_wr_crn -
b_cof[ 3 ] * wr_br_crn * force_pu_wid -
b_cof[ 4 ] * wr_br_crn * pow( force_pu_wid, float(1.5) ) -
b_cof[ 8 ] * wr_br_crn -
b_cof[ 9 ] * force_pu_wid -
b_cof[ 11 ] * force_pu_wid -
b_cof[ 12 ] * pow( force_pu_wid, float(1.5) ) -
b_cof[ 14 ] * pce_wr_crn -
b_cof[ 15 ] * pce_wr_crn -
b_cof[ 16 ] -
b_cof[ 17 ] ) /
( b_cof[ 5 ] +
b_cof[ 6 ] * force_pu_wid +
b_cof[ 7 ] * pow( force_pu_wid, float(2.0) ) +
b_cof[ 10 ] +
b_cof[ 13 ] );

```

cUFDD::Frc_PU_Wid

cUFDD::Frc_PU_Wid此函数用于计算单位宽度轧制力，输入量为UFD辊缝凸度、弯辊力、综合轧辊凸度，输出量为单位宽度轧制力。ufd_modifier在这里参与的是计算过程中被求导函数的常数项的计算。cof_4计算中，ufd_modifier从辊缝凸度中除去，对辊缝凸度进行修正。

cUFDD::Pce_WR_Crn

```

float cUFDD::Pce_WR_Crn(
    const float ufd_prf, // [mm_mm_in] UFD roll gap pro
file
    const float force_pu_wid, // [kn/mm_mton/mm_eton/in] rol
ling
    // force per unit piece wid
th
    const float force_bnd, // [kn_mton_eton] roll bending
force
    const float wr_br_crn // [mm_mm_in] work roll to bac
kup
    // roll stack crown
    ) const // [mm_mm_in] piece to work ro
11
    // stack crown

{ // Begin of PCE_WR_CRN function

return

```

```

( ufd_prf / ufd_modifier -
b_cof[ 0 ] * force_pu_wid -
b_cof[ 1 ] * pow( force_pu_wid, float(1.5) ) -
b_cof[ 3 ] * wr_br_crn * force_pu_wid -
b_cof[ 4 ] * wr_br_crn * pow( force_pu_wid, float(1.5) ) -
b_cof[ 5 ] * force_bnd -
b_cof[ 6 ] * force_bnd * force_pu_wid -
b_cof[ 7 ] * force_bnd * pow( force_pu_wid, float(2.0) ) -
b_cof[ 8 ] * wr_br_crn -
b_cof[ 9 ] * force_pu_wid -
b_cof[ 10 ] * force_bnd -
b_cof[ 11 ] * force_pu_wid -
b_cof[ 12 ] * pow( force_pu_wid, float(1.5) ) -
b_cof[ 13 ] * force_bnd -
b_cof[ 16 ] -
b_cof[ 17 ] ) /
( b_cof[ 2 ] +
b_cof[ 14 ] +
b_cof[ 15 ] );

} // End of PCE_WR_CRN function

```

cUFDD::Pce_WR_Crn函数在包络线生成和分配计算阶段进行调用，而在分配阶段仅调用一次。

传入的UFD辊缝凸度对应的UFD单位凸度来自cLRGD->UFD_PU_Prf3，是利用入口有效凸度、出口有效凸度以及应变释放系数计算的结果，ufd_modifier是对这个UFD单位凸度对应的辊缝凸度进行修正。

cUFDD::Crns

cUFDD::Crns是用来同时计算带钢-工作辊凸度和工作辊-支撑辊凸度的。输入量为UFD辊缝凸度、单位宽度轧制力和弯辊力。

在这里ufd_modifier不直接参与cUFDD::Crns的补偿，但是在分配计算阶段，其输入量弯辊力是利用cUFDD::Bnd_Frc进行计算的结果。

初始弯辊力

初始弯辊力在表中的位置

初始弯辊力调节值位于GSM的NormalBendForce.xlsb当中。按钢种组分类，针对每个宽度区间和厚度区间、针对每个道次，有一个初始弯辊力调节值，供工艺人员进行调节。如下图所示。

A	B	C	D	E	F	G	H	I	J	K
1 Normal Bend Force				bend-nom [kN/roll]						
2 Last Get : 1/14/2017 10:43:19 AM										
3 Width index				Gauge index						
4				bend-nom [kN/roll]						
5	0	800<=w < 900		0	1.00<=h<1.15	890	890	890	890	840
6				1	1.15<=h<1.30	890	890	890	890	840
7				2	1.30<=h<1.50	890	890	890	890	840
8				3	1.50<=h<1.70	890	890	890	890	840
9				4	1.70<=h<1.85	890	890	890	890	840
10				5	1.85<=h<2.05	890	890	890	890	840
11				6	2.05<=h<2.25	890	890	890	890	840
12				7	2.25<=h<2.50	890	890	890	890	840
13				8	2.50<=h<2.75	890	890	890	890	840
14				9	2.75<=h<3.00	890	890	890	890	840
15				10	3.00<=h<3.40	890	890	890	890	840
16				11	3.40<=h<4.00	890	890	890	890	840
17				12	4.00<=h<5.00	890	890	890	890	840
18				13	5.00<=h<6.00	890	890	890	890	840
19				14	6.00<=h<7.50	890	890	890	890	840
20				15	7.50<=h<9.00	890	890	890	890	840
21				16	9.00<=h<10.50	890	890	890	890	840
22				17	10.50<=h<11.50	890	890	890	890	840
23				18	11.50<=h<12.70	890	890	890	890	840
24				19	12.70<=h<14.00	890	890	890	890	840
25				20	(spare)	890	890	890	890	840
26				21	(spare)	890	890	890	890	840
27				22	(spare)	890	890	890	890	840
28				23	(spare)	890	890	890	890	840
29				24	(spare)	890	890	890	890	840
30				25	(spare)	890	890	890	890	840
31				26	(spare)	890	890	890	890	840
32				27	(spare)	890	890	890	890	840
33	1	900 <= w < 1050		0	1.00<=h<1.15	890	890	890	890	840
34				1	1.15<=h<1.30	890	890	890	890	840
35				2	1.30<=h<1.50	890	890	890	890	840

初始弯辊力的初始化

首先在cShapeSetupD::Init()的初始化过程，从sSPRP结构中将此参数psSPRP->bend_nom[passIdx]按制定的道次赋值给pcFSStd对象中的变量pcFSSPassD->pcFSStdD[iter]->pcFSStd->force_bnd_nom。

在cfg_fsstd.txt中，前6道次机架的force_bnd_nom为1500kN，第七机架为1000kN。

初始弯辊力的作用

PENV

在包络线计算过程中，force_bnd_nom主要参与UFD辊缝凸度对单位轧制力偏导数的计算。作为第二个参数传入。

```
pcFSPassD->pcPEnvD->dprf_dfrcw =
    pcFSPassD->pcFSStdD[ iter ]->pcUFDD->Dprf_Dfrcw (
        pcFSPassD->pcPEnvD->force_pu_wid,
        pcFSPassD->pcFSStdD[ iter ]->pcFSStd->force_bn
        d_nom,
        pce_wr_crn,
        wr_br_crn
    );
};
```

分配计算

在分配计算阶段，force_bnd_nom参与SSU轧制力的计算。

窜辊机构软极限调节值

调节值位置

窜辊机构软极限调节值位于GSM的ShiftActuatorLimit.xlsb当中。按钢种组分类，针对每个宽度区间和厚度区间、针对每个道次，有一组最大和最小窜辊软极限调节值，供工艺人员进行调节。

窜辊机构软极限调节值的作用

在SSU初始设定阶段，窜辊机构软极限调节值主要是用来作为乘数，乘到窜辊硬极限上，用于修正和约束窜辊软极限。

窜辊调节值参与过程

首先利用窜辊速度和卷卷带钢之间的间隙时间计算窜辊步长的最大变化量。默认的窜辊步长最大变化量为100mm，与计算值比较并取最小值。如果磨损存在异常，则从新按磨损异常的规则确立最大窜辊步长。之后计算窜辊的位置极限，原窜辊位置加上最大最小窜辊步长变化量求窜辊位置的软极限。

弯辊机构软极限调节值

调节值位置

弯辊机构软极限调节值位于BendingActuatorLimit.xlsb中，针对每个钢种族，不同宽度与不同厚度、不同机架分别有最大最小两个调节值。

调节值作用

psSPRP->min_bend_lmt参与计算弯辊力软极限。

```
pcFSStdDloc->force_bnd_lim[ i ] = pcFSStdDloc->force_bnd_lim_org[ i ] =
pcFSStdDloc->pcFSStd->force_bnd_lim[ i ] *
psSPRP->min_bend_lmt[ pcFSStdDloc->pcFSStd->num-1 ];
```

CenterBuckleTuning和WavinessTuning

CTOOL中GSM模块中有CenterBuckleTuning和CenterBuckleTuning两张表，这两张表主要用来对屈曲判别标准的中浪和双边浪极限值进行调整。个人理解是用来调整在带钢约束条件下的死区极限范围。

这两个表对于每个机架F1到F7分别有两个参数。一个是比例系数multiplier，作为乘数而存在，另一个是补偿值Offset，作为加数而存在。

在模型代码中，用到参数表的地方主要在LPCE模块当中。

参数表中的中浪和双边浪调节系数作为函数的参数参与LPce对象的初始化过程cLPceD::Init()，保存在bckl_mul和bckl_off中。参数表中的数据

在cLPceD::Crit_Bckl_Lim中参与屈曲极限值的计算。首先建立屈曲极限的缓冲区并初始化为0值。中浪和双边浪的屈曲极限的原极限计算值是在cLPceD::Crit_Bckl()当中完成的，并存储于bckl_lim。如下图所示。需要用到带钢的宽度、厚度、带钢所受的机架张力和杨氏模量等进行计算。

之后在cLPceD::Crit_Bckl_Lim中用调节系数进行修正，multiplier作为乘数，offset作为加数。值得注意的是，补偿计算极限值有一个约束条件，那就是在补偿计算后，双边浪的极限值必须大于中浪的极限值，也就是说必须保证死区的存在，否则返回中浪极限值和双边浪极限值的均值。如下图中的if条件分支所示。

工作辊综合凸度补偿

综合凸度补偿的调节在WorkCrownOffsetTuning.xlsb当中，初始变量为psSLFG->wr_crn_off[pass_idx]，主要参与综合凸度的计算。

Width index	Gauge index	F1	F2	F3	F4	F5	F6	F7
		wr_crn_off_adj						
0 800<= w < 900	0 1.00<=h<1.15	-0.4	-0.25	0	0.1	0.1	0	0
	1 1.15<=h<1.30	-0.4	-0.25	0	0.1	0.1	0	0
	2 1.30<=h<1.50	-0.4	-0.25	0	0.1	0.1	0	0
	3 1.50<=h<1.70	-0.4	-0.25	0	0.1	0.1	0	0
	4 1.70<=h<1.85	-0.4	-0.25	0	0.1	0.1	0	0
	5 1.85<=h<2.05	-0.4	-0.25	0	0.1	0.1	0	0
	6 2.05<=h<2.25	-0.4	-0.25	0	0.1	0.1	0	0
	7 2.25<=h<2.50	-0.4	-0.25	0	0.1	0.1	0	0
	8 2.50<=h<2.75	-0.4	-0.25	0	0.1	0.1	0	0
	9 2.75<=h<3.00	-0.4	-0.25	0	0.1	0.1	0	0
	10 3.00<=h<3.40	-0.4	-0.25	0	0.1	0.1	0	0
	11 3.40<=h<4.00	-0.4	-0.25	0	0.1	0.1	0	0
	12 4.00<=h<5.00	-0.4	-0.25	0	0.1	0.1	0	0
	13 5.00<=h<6.00	-0.4	-0.25	0	0.1	0.1	0	0
	14 6.00<=h<7.50	-0.4	-0.25	0	0.1	0.1	0	0
	15 7.50<=h<9.00	-0.4	-0.25	0	0.1	0.1	0	0
	16 9.00<=h<10.50	-0.4	-0.25	0	0.1	0.1	0	0
	17 10.50<=h<11.50	-0.4	-0.25	0	0.1	0.1	0	0
	18 11.50<=h<12.70	-0.4	-0.25	0	0.1	0.1	0	0
	19 12.70<=h<14.00	-0.4	-0.25	0	0.1	0.1	0	0
	20 (spare)	0	0	0	0	0	0	0
	21 (spare)	0	0	0	0	0	0	0
	22 (spare)	0	0	0	0	0	0	0
	23 (spare)	0	0	0	0	0	0	0
	24 (spare)	0	0	0	0	0	0	0
	25 (spare)	0	0	0	0	0	0	0
	26 (spare)	0	0	0	0	0	0	0
	27 (spare)	0	0	0	0	0	0	0
1 900 <= w < 1050	0 1.00<=h<1.15	-0.4	-0.25	0	0.1	0.1	0	0
	1 1.15<=h<1.30	-0.4	-0.25	0	0.1	0.1	0	0

在ShapeSetup初始化过程中，综合凸度的补偿值作为输入参数传入cCRLCD::Init()进行初始化。

```

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 ],
//@@@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 ]->pcEnPceD->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 ]->pcStdRollPrD
    //psPDI->grt_idx
);

```

板形模型实战

模型的掌握和使用，万万不可脱离实战。

目标凸度变化影响板形模型设定

事件经过

2018年5月15日，15:15，丙（白）班，MRTLA42005钢种，卷号M18057558H，规格3.2*1172mm。带钢穿带过程中F1-F4机架跑偏操作侧严重，带钢头部至F6-7机架时瞬间跑偏传动侧，操作侧起浪，操作调整F6-7调平，随即带钢反向跑偏后起套堆钢。

从视频回放来看，存在两方面原因：第一，F1-F7机架跑偏严重；第二，F6-7出现反向跑偏。带钢跑偏严重主要是由于窜辊量变化较大引起。窜辊变化的原因主要为目标凸度剧变造成整个板形模型凸度分配异常，具体影响过程详见下文。

轧制力分析

轧制力方面如下。

机架	FBK/KN	MEAS/KN	偏差
F1	25292	22984	-9.13%
F2	21368	21824	2.13%
F3	19981	19268	-3.57%
F4	17007	17273	1.56%
F5	15672	15443	-1.46%
F6	12749	12483	-2.09%
F7	13318	13300	-0.14%

F1轧制力预报出现偏差，但是在10%以内，属于正常范围，其余预报正常；但是F7轧制力大于F6轧制力，属于异常情况，查询发现F7压下率锁在13%。按照轧制力分配原则，F5-F7机架轧制力应按照等比例分配，保证板形稳定，但是F7轧制力分配明显异常。

窜辊和板形所受影响

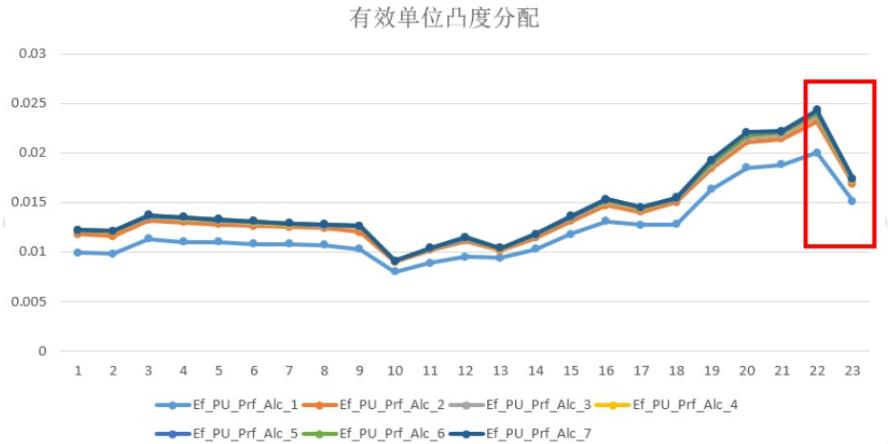
从开轧至堆钢的窜辊情况如下。

卷号	F1	F2	F3	F4	F5	F6	F7
M18057536H	36.6	10.2	-6.3	-2.3	-30.0	-0.0	0.0
M18057537C	28.7	0.3	-12.5	-7.3	-60.0	-0.0	0.0
M18057538C	13.2	-10.8	-29.3	-20.6	-90.0	0.0	24.3
M18057539C	12.4	-12.9	-30.8	-21.3	-120.0	0.0	55.5
M18057540C	14.4	-9.3	-29.0	-21.4	-133.1	0.0	83.9
M18057541C	15.2	-8.8	-29.8	-21.7	-131.1	24.3	108.1
M18057542C	14.9	-10.1	-30.3	-23.6	-133.3	55.5	126.9
M18057543C	14.3	-9.7	-31.5	-25.2	-133.4	83.9	139.3
M18057544C	31.8	-0.0	0.0	-0.0	-0.0	-0.0	-0.0
M18057545M	79.5	41.8	33.8	30.0	-7.4	126.9	140.0
M18057546M	63.6	23.4	19.8	38.6	-25.2	139.3	134.0
M18057547C	49.3	11.2	9.9	29.7	-38.1	140.0	94.7
M18057548C	81.3	32.4	22.3	9.8	-31.2	140.0	77.4
M18057549C	61.3	15.8	11.7	1.2	-38.4	134.0	56.2
M18057550C	41.1	-1.5	-2.3	-9.6	-53.8	94.7	32.3
M18057551C	25.5	-10.3	-5.7	-9.9	-48.5	77.4	6.7
M18057552C	50.4	13.0	2.4	-4.1	-37.2	56.2	-19.2
M18057553H	25.3	5.8	-9.6	-9.5	-69.2	32.3	-44.2
M18057554H	-4.5	-12.4	-22.9	-21.0	-92.7	6.7	-67.0
M18057555P	-25.7	-37.4	-35.1	1.4	-57.3	-19.2	-86.3
M18057556P	-32.8	-47.1	-41.6	-3.2	-50.5	-44.2	-101.4
M18057557H	-87.1	-83.4	-65.7	-48.8	-109.6	-66.9	-111.4
M18057558H]	13.1	-15.4	-22.7	10.0	-38.7	-86.3	-115.8

M18057557H到M18057558H窜辊变化剧烈。 M18057558H窜辊剧烈变化的原因如下：
M18057558H这卷带钢的CVC设定值达到窜辊设定软极限，如下图所示。

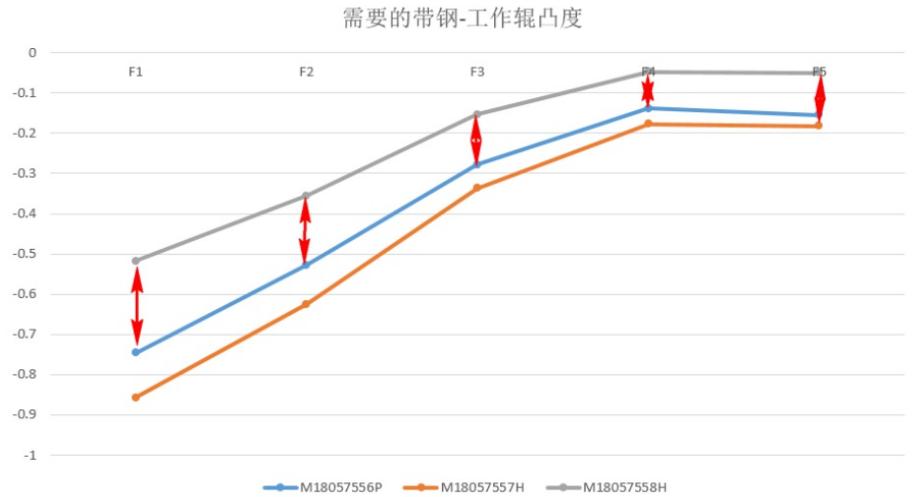
WR	Shft	
Min [mm]	Nom [mm]	Max [mm]
-140.60	0.00	13.06
-133.39	0.00	-15.39
-105.73	0.00	-22.73
-78.79	0.00	10.09
-139.58	0.00	-38.72
-86.34	0.00	-86.34
-115.80	0.00	-115.80

以F1为例，M18057557H带钢F1的CVC窜辊为-87.1mm，根据窜辊限幅的计算，F1机架的窜辊最大步长为100mm，因此下一卷带钢M18057558H的窜辊限幅的最大值为+13.1mm。但是，实际需要的窜辊变化量比窜辊限幅最大值还要大，因此在计算板形设定时，窜辊值被限制在软极限。造成实际窜辊变化量远超窜辊限幅最大值的原因，是因为凸度分配计算中，同规格同钢种的各个机架出入口有效单位凸度变化大。



有效单位凸度在分配计算中是从F7往前计算，F7出口依赖目标凸度作为输入，M18057557H目标凸度为 $50 \mu\text{m}$ ，M18057558H目标凸度为 $30 \mu\text{m}$ ，同钢种同规格， $50 \mu\text{m}$ 到 $30 \mu\text{m}$ 为一个天一个地。因此造成板形模型分配计算中，从F7到F1推算过程中，F7的初始计算存在较大偏差。M18057557H的F7出口有效单位凸度为 $0.024 \mu\text{m/mm}$ ，而M18057558H的F7出口有效单位凸度为 $0.0174 \mu\text{m/mm}$ ，相差较大。

有效单位凸度偏差较大，导致计算的所需的带钢-工作辊凸度偏差也较大，变化量能达到50%以上。



所需的带钢-工作辊凸度是窜辊迭代计算的直接输入量，M18057558H和M18057557H相比，所需的带钢-工作辊凸度减小1倍以上，所分配设定的窜辊值必然向正方向移动明显。以上解释了为什么目标凸度的变化影响了整个板形模型分配的计算。

解决措施

解决措施很简单，梳理各钢种PDI目标凸度，规范同钢种凸度目标值设定。