

# 曲线坡度折减超限问题及其计算机检测程序实现

刘卫青

中交(西安)铁道设计研究院有限公司 陕西 西安 710065

**摘要:** 本文研究了在线路设计工作中,尤其是双线设计中存在的实际坡度超限的问题。由实际工作出发,引出使用计算机程序辅助解决问题的方法,并提出了基于python的曲线坡度折减核算程序整体设计思路,提出了算法内核的实现方法和根据实际情况的优化办法。本文的研究可为今后解决类似的问题提供思路及参考,为设计工作的智能化数字化作出贡献,提高生产效率。

**关键词:** 铁路;线路设计;双线;计算机;算法

## 1 引言

随着铁路路网的铺开及新建铁路工程的减少,目前,在铁路线路设计中经常会遇到对既有铁路增加运量、进行改扩建的设计需求。在这些既有铁路改扩建情形中,又以增加第二线等复线改扩建形式对铁路运能提高的效果是直截了当的。增建第二线的设计,理应尽可能保证两线并行等高,减少既有线的改造工程量。平面线路需经济、平顺,尽量减少或不进行换侧,曲线宜设计为同心圆。纵断面理应将轨面设计为等高,并满足标准规定的限制坡度要求<sup>[1]</sup>。

在实际开展设计工作中发现,当增建的二线线路按并行等高来设计时,由于曲线地段内业断链的存在,可能导致按《线规》<sup>[2]</sup>折减坡度设计的路段出现新建二线纵坡实际坡度超限而设计坡度检算通过的情况,这种情况尤其在长曲线路段比较严重。

因此,面对线路设计工作中产生的此类实际纵坡超限问题,需要寻找到一个行之有效的检验核查办法,力求在设计阶段就解决潜在问题,使线路纵断面曲线平稳顺滑,确保列车按设计速度行驶。

## 2 铁路曲线坡度折减原理

### 2.1 曲线坡度折减

当列车行驶到曲线地段时,受到的总阻力包括了坡道引起的阻力和曲线引起的阻力。为使得列车通过速度不低于计算速度,需要降低坡道引起的阻力,即降低经过曲线的坡的坡度,从而使得牵引力和阻力可达到平衡,列车保持计算速度通行。

#### (1) 坡道附加阻力

列车在坡道运行时,重力由于坡道的原因在轨道方向和垂直于轨道的方向可以分解为两个分力。如图1。

由于坡道的倾角 $\beta$ 一般都很小,故而可以使得 $\sin\beta \approx \tan\beta$ ,得出 $F_{附加阻力} \approx G \tan\beta$  (kN)。i为坡度(%),则

$F_{附加阻力} \approx G \frac{i}{1000}$  (kN) =  $Gi$  (N)。引入单位坡道附加阻力的概念,由总附加阻力除以列车总质量,得单位坡道附加阻力 $\omega_i = gi$  (N/t)。

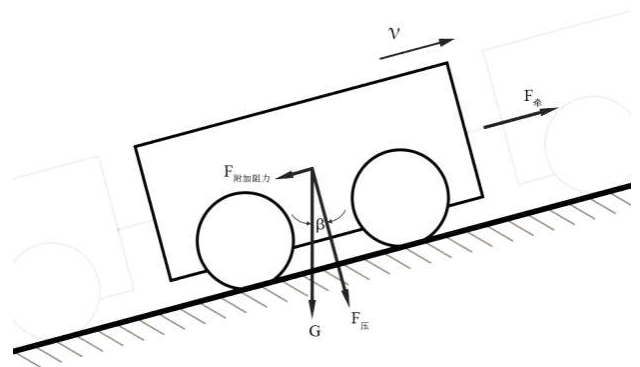


图1 坡道引起附加阻力示意图

#### (2) 曲线附加阻力

列车行驶到曲线上后,由于车辆的车架或转向架为近似刚体,其车轮轮缘会与曲线钢轨的轨头产生轮轨相互作用力,并且车架或转向架之间也因此会产生一定的摩擦阻力。因而会额外对车体运动产生新的附加阻力,这一附加阻力称为曲线附加阻力。由于曲线附加阻力的影响因素复杂,一般采用与曲线半径R相关联的经验公式计算,即单位曲线附加阻力为 $\omega_r = \frac{600}{R}g$  (N/t)。因曲线长度 $L_y$  (m)和转角 $\alpha$  (°)有 $R = \frac{180L_y}{\pi\alpha}$  (m)的关系,单位曲线附加阻力也可以表示为 $\omega_r = \frac{10.5\alpha}{L_y}g$  (N/t)。

#### 2.2 附加阻力换算坡度

由于单位坡道附加阻力为 $\omega_i = gi$  (N/t),因此可以类比地把曲线附加阻力表示为 $\omega_r = g \frac{600}{R} = gi_r$  (N/t),其中 $i_r$ 即为曲线附加阻力换算坡度,也可称之为曲线当量坡

度，其物理含义是将列车在半径为 $R$ 的曲线上行驶所产生的附加阻力等价变换为列车在坡度为 $i$ 的坡段上行驶所产生的附加阻力。等价变换后，要使得列车按不低于计算速度从某一经过曲线的坡段通过，只需将直线地段的限坡减去曲线附加阻力等价的当量坡度即得该经过曲线的坡段的设计坡度。

### 3 曲线坡度折减计算机程序实现

传统设计工作中去核查坡度超限问题主要靠人工来逐个核实，在数字化进程进展到如今的阶段，我们可以借助计算机作为辅助工具来帮助检查复线线路中的实际纵坡超限问题。

#### 3.1 程序设计流程

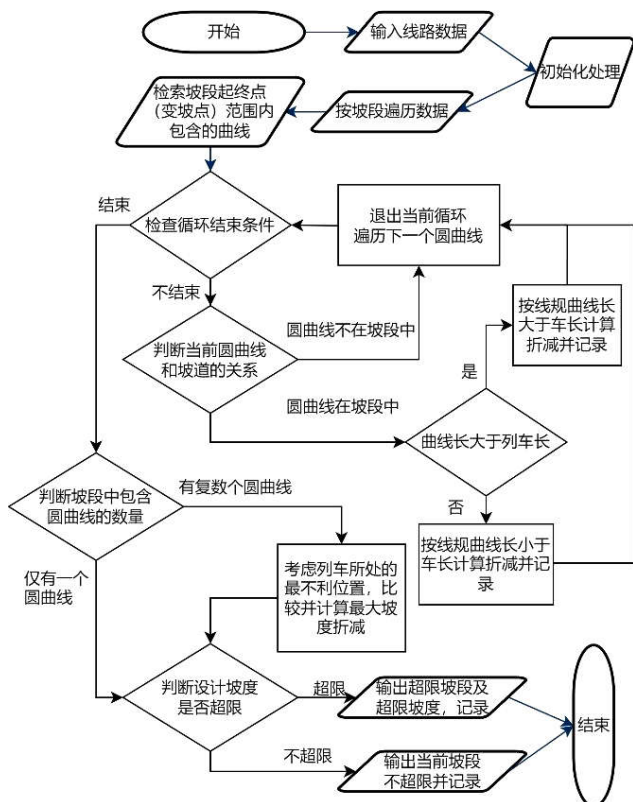


图2 坡度折减程序设计流程图

#### 3.2 基于python的坡度折减核算程序实现

输入输出：输入使用pandas库读取坡度表和曲线，输出到屏幕上和文本文件中以待查看。

初始化处理：根据输入设置列车长，初始化指针为0，初始化各计数器、数组。

循环：采用for循环遍历坡度表，for循环内嵌套while循环用来寻找当前坡度前后的曲线。

#### 3.3 算法实现

##### (1) 算法依据

本程序算法依据主要为最新的《铁路线路设计规范》

(TB 10098-2017)中的6.4.7节最大坡度折减规定<sup>[2]</sup>。

```

4 import math
5 import pandas as pd
6 from itertools import islice
7
8 # 输入数据
9 sspds=load('输入坡度表.xlsx') # 输入坡度表
10 sspds=load('输入曲线表.xlsx') # 输入曲线表
11 sspds=sspds[['id', 'start', 'end', 'slope']]
12 # 初始化
13 train_length=100 # 列车长度
14 # 初始化指针
15 start_index=0
16 # 初始化计数器
17 count=0
18 # 初始化数组
19 slope_array=[]
20 # 初始化文件
21 file=open('输出结果.txt', 'w')
22 # 遍历坡度表
23 for i in range(len(sspds)):
24     # 获取当前坡度段信息
25     start=sspds['start'][i]
26     end=sspds['end'][i]
27     slope=sspds['slope'][i]
28     # 获取当前曲线段信息
29     start_curve=sspds['start_curve'][i]
30     end_curve=sspds['end_curve'][i]
31     # 判断当前曲线是否在坡度段内
32     if start_curve < start && end_curve > end:
33         # 计算曲线长度
34         curve_length=end_curve-start_curve
35         # 判断曲线长度是否大于列车长度
36         if curve_length > train_length:
37             # 计算折减坡度
38             reduction_slope=...
39             # 记录折减坡度
40             file.write('坡度段: %s, 曲线: %s, 折减坡度: %s\n' % (start, end, reduction_slope))
41         else:
42             # 计算折减坡度
43             reduction_slope=...
44             # 记录折减坡度
45             file.write('坡度段: %s, 曲线: %s, 折减坡度: %s\n' % (start, end, reduction_slope))
46     else:
47         # 曲线不在坡度段内，跳过
48         continue
49 # 输出结果
50 file.close()
51 print('计算完成')
52 
```

图3 程序实例代码

##### (2) 实现方法

程序设计中的难点主要在于如何依据判断条件来选择对应的合适计算公式。此外，在多个曲线存在于同一坡度的情况下，确定列车经过时的最不利工况也是一个需要研究的问题。

本程序设计中，采用条件判断的方式来对读取的坡段和曲线的相对位置关系进行判定。坡段和圆曲线的位置关系可能出现如下两种大的情况：坡段范围内仅有一处圆曲线、坡段范围内有复数圆曲线。在坡段范围内有复数圆曲线的情况中，又可以分为个别曲线长大于列车长、所有曲线长均小于列车长等情况，需要按列车经过时的最不利工况来进行判断需选取哪些作为计算折减所用曲线。具体位置关系如下图所示。

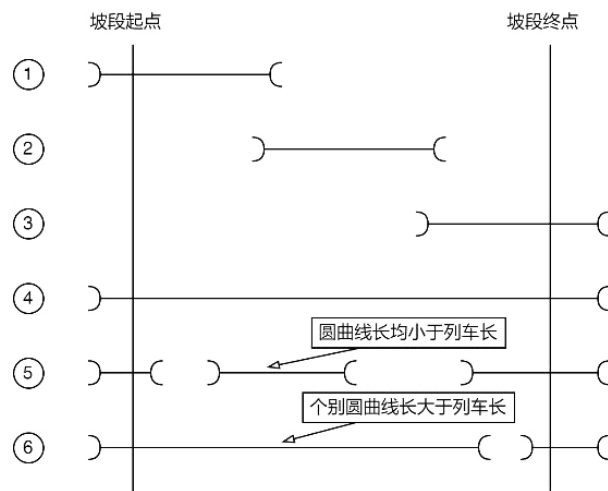


图4 坡段与圆曲线的位置关系图

具体搜索坡段上存在的圆曲线并计算对应工况坡度折减的算法实现可分为如下几个步骤。首先，将读取的曲线表按里程顺序排列，在当前坡段小里程端起，检索满足曲线起点在坡段起点之前、终点在坡段起点之后的曲线；寻找到第一个曲线后，判断曲线长是否大于列车长，并根据判断结果选用不同公式。其中对曲线长小于列车长的折减公式，需要进一步比较列车长和坡段长，取两者之间长度小的作为计算时采用的长度 $l$ 。需要注意的是，在取得计算坡长时，坡段长的长度应该按实际坡长进行计算，即需加上复线产生的内业断链后再进行比较。

当存在多个曲线时，则需要对列车经过时的最不利工况位置进行判断。其本质上是在挑选长度一定的情况下哪个路径的曲线转角之和最大。在理论情况下，坡段内有 $n$ 个曲线时，可以将问题等价于图论中的寻找最佳路径问题加以解决，而实际中由于坡段长度不可能无限伸长，坡段内的曲线数目有限，所以可以针对具体问题具体分析，采用较为简便的方法进行计算。例如以某铁路增建二线工程为例，其中一个坡段内的曲线至多不超过5个，且由于列车长度有限，同一时间列车最多经过不超过3个曲线，因此可以采用贪心策略，先选取曲线中转角最大的那一个，然后再判断与两侧哪一个曲线相加可以得到最大值，并以此作为计算依据，从而简化算法。

(3) 结果检验

```
21
坡起点里程: 745190.0 坡终点里程: 745800.296 坡长值 600.0
坡度折减后: 5.954
按实际坡长折减后(长链不算): 5.954
原坡度为 -6.0 .....核算不过!!!!!!!!!!!!!!!!!!!!!!

22
坡起点里程: 745800.296 坡终点里程: 746420.296 坡长值 620.0
坡度折减后: 5.736
按实际坡长折减后(长链不算): 5.736
原坡度为 -5.6 .....核算通过
```

图5 程序运行结果

如图所示，程序运行结果经人工核对，正确运行、

结果可信。

4 结论与展望

本文研究了复线设计中可能存在的实际坡度超限问题，并提出了使用计算机软件辅助进行查验核实的方法，提出了具体的解决思路及程序实现办法，给出了相对应的算法。

本文取得的研究成果如下：

(1) 编写了铁路曲线坡度折减计算及核验的计算机程序。

(2) 解决了坡度超限和实际坡度超限人工核查工作量大的问题。

(3) 在算法实现中给出了依据具体情况优化方法。

计算机程序乃至现如今的人工智能大模型都可谓是人们用于提高工作效率，确保工作质量的好帮手，在今后的工作中，可以对更多情况善用计算机程序进行辅助，从而在提升设计效率的同时确保设计质量，为轨道交通事业的建设添砖加瓦。

参考文献

[1]铁道部第一勘测设计院.铁路工程设计技术手册:线路[M].中国铁道出版社,1994.  
[2]国家铁路局.铁路线路设计规范:[S].中国铁道出版社,2017: