

2023 年第八届“数维杯”大学生 数学建模挑战赛论文

题 目 基于最优速度序列的节能列车控制优化方法研究

摘 要

迅速发展的城市轨道交通缓解了日益增长的客流压力，但与此同时也带来了能耗增长和其它负面效应。降低高铁列车运行能耗对于提升高速铁路竞争力、实现可持续运营具有重要意义。本文重点对高速列车节能操纵优化问题进行了研究。针对问题 1，采用 Matlab 仿真软件按照要求构建地铁列车牵引能耗计算模型的方法解决；针对问题 2，采用搜索最优速度序列的方法解决；针对问题 3，采用实时更新最优速度序列的方法解决。

对于问题 1 我们首先建立了地铁列车牵引能耗计算模型。首先利用问题所给出的参数进行预处理，其次计算了列车在不同运动状态下的受力情况，并借助范数逼近算法和 Matlab 软件完成了对列车运行过程状态变化的仿真。

对于问题 2 我们完善了问题 1 中的地铁列车牵引能耗计算模型，考虑了更多的影响因素。结合问题 2 给出的数据，通过寻找最优速度序列实现在保证列车正常运行的前提下，降低列车在站间的能量消耗。

对于问题 3 我们用实时更新的最优速度序列，在尽可能降低能量消耗的情况下，解决了突发情况下运行时间增长的问题。

本次研究暂未考虑列车辅助设备和载客量对能源消耗的影响，未来可在已提出的模型上加入上述影响因素。

关键词 节能；最优速度序列；数值积分；最优化模型

目 录

一、 问题描述.....	1
1.1 引言	1
1.2 要解决的具体问题	1
二、 问题分析.....	3
2.1 问题 1 的分析	5
2.2 问题 2 的分析	6
2.3 问题 3 的分析.....	7
三、 模型假设.....	7
四、 定义与符号说明.....	7
五、 模型的建立与求解.....	9
5.1 问题 1 的模型建立与求解	9
5.1.1 列车以最短时间到达站台 B 模型的建立.....	9
5.1.2 列车以最短时间到达站台 B 模型的求解	11
5.1.3 模型合理性分析	14
5.2 问题 2 的模型建立与求解	14
5.2.1 节能列车控制优化模型的建立	14
5.2.2 节能列车控制优化模型的求解	19
5.2.3 结果	22
5.3 问题 3 的模型建立与求解	22
5.3.1 实时运行状态规划模型的建立	22
5.3.2 实时运行状态规划模型的求解	23
六、 模型的评价及优化.....	25
6.1 误差分析	25
6.1.1 针对于问题 1 的误差分析	25
6.1.2 针对于问题 2 的误差分析	26
6.1.3 针对于问题 3 的误差分析.....	27
6.2 模型的优点（建模方法创新、求解特色等）	27
6.3 模型的缺点	27

6.4 模型的推广	27
参考文献.....	28
附录.....	29

一、 问题描述

1.1 引言

自上世纪 50 年代以来，大力发展城市轨道交通就成为了我国的重要政策。在“十二五”规划期间我国已有 22 座城市新开通运营城市轨道交通，建设城市轨道交通共计 95 条线路，运营总里程达到 3171 公里^[1]。截至 2020 年末“十三五”结束时，根据中国城市轨道交通协会发布的统计显示，中国内地 45 个城市已开通城市轨道交通运营线路长度达到 7978 公里。在“十四五”规划（2021 年—2025 年）的期间，我国城市轨道交通运营里程计划新增 5000 公里，全国总运营里程将达到 1.3 万公里，其中，北京和上海将有望分别形成 1000 公里以上的庞大轨道交通线网^[2]。随着我国城市轨道交通的快速发展，轨道线路运营里程和客运量不断增长的同时，运营总能耗也在不断增长。据统计，2020 年，全国城轨交通总电能耗达到 172.4 亿千瓦时，同比增长 12.9%。其中，牵引能耗达到 84 亿千瓦时，同比增长 6.3%^[3]。另外，随着新建城轨线路的增加，新冠疫情的好转，显而易见，未来几十年我国城市轨道交通总体的运营能耗也将持续攀升。因此，降低牵引能耗成本减轻城市电网负荷、实现可持续运营、践行“双碳”政策具有重要意义。

1.2 要解决的具体问题

在列车运行过程中，列车与外界会产生各种摩擦，进而消耗列车牵引的能量。列车运行过程中，被考虑的因素较多，如列车与轨道的摩擦、列车受到的空气阻力、列车势能的变化、列车运行过程中的位置限速等。在同一段旅途中，列车使用不同的驾驶策略通常会产生不同的能量和时间的消耗。

单列车在两个站台之间的运行过程如图 1-1 所示。

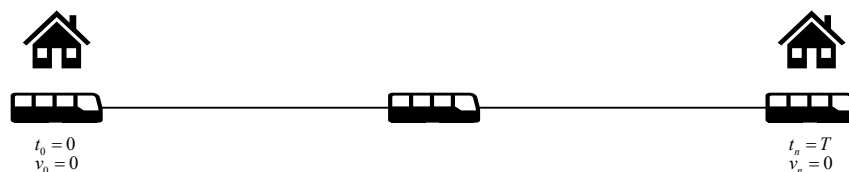


图 1-1 单列车运行过程

假设一辆列车在水平轨道上运行，从站台 A 运行至站台 B，其间距为 5144.7m，运行的速度上限为 100km/h，列车质量为 176.3t，列车旋转部件惯性的旋转质量因数 $\rho=1.08$ ，列车电机的最大牵引力为 310kN，机械制动部件的最大制动力为 760kN。列车

受到的阻力满足 Davis 阻力方程:

$$f = 2.0895 + 0.0098v + 0.006v^2 \quad (1.1)$$

其中, f 为阻力, 单位为 kN, v 为速度, 单位为 m/s。

根据以上条件, 通过建模方法编写程序获得列车运行过程的速度-距离曲线、牵引制动力-距离曲线、时间-距离曲线与能量消耗-距离曲线, 并记录程序运行时间。得到最短时间需要获取列车以最短时间到达站台 B、在最短运行时间上分别增加 10s、20s、50s、150s、300s 到达站台 B 总共六组曲线。

在列车运行的实际情况中, 需要考虑更多的因素。列车运行的旅途中不同的路段的限速不同, 路况不同, 电机的动态特性也更加复杂, 此外, 储能装置在列车节能领域有着重要的应用, 列车制动时, 会将一定比例的能量储存至储能装置中, 以待后续使用, 如图 1-2 所示。题目附件中给出从 XEQ 站至 SMKXY 站的路况数据与包含电机牵引/制动动态特性的列车相关参数数据。考虑路况信息以及电机的复杂动态过程。若列车计划运行时间为 T , 设计优化方案得到可行的速度轨迹, 使得运行过程的能耗越低越好。重新获取列车以最短时间到达站台 B、在最短运行时间上分别增加 10s、20s、50s、150s、300s 到达站台 B 总共六组曲线。

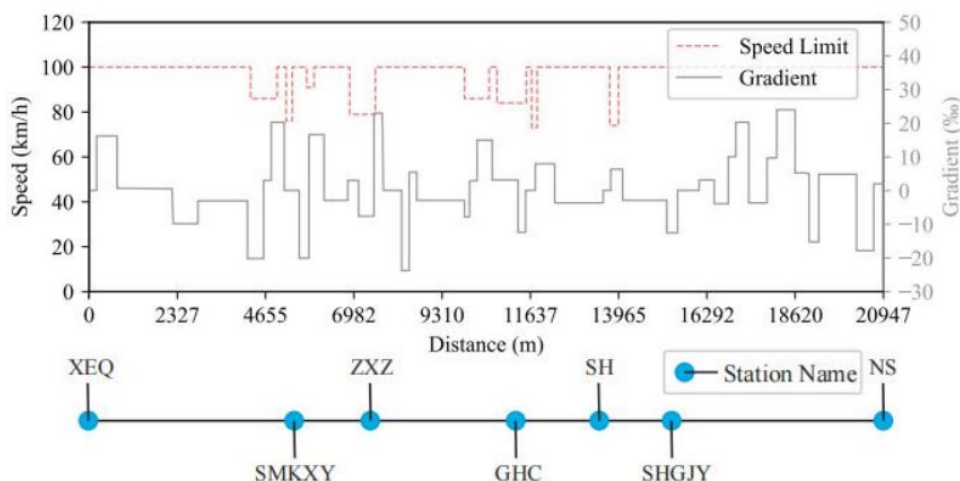


图 1-2 复杂路况的运行旅途

考虑列车在运行过程中可能会出现的情况导致列车需要提前或延时到达站台。列车的运行速度轨迹需要根据新的到站时间而发生变化。假设列车从起点出发, 原计划于 320s 后到达终点, 列车运行至 2000m 位置时, 由于前方突发事故, 需要延迟 60s 到达终点。请你设计优化方案在保持列车节能运行下, 能够快速地(越快越好)得到调整后的优化速度轨迹。作出列车运行过程的速度-距离曲线、牵引制动力-距离曲线、时间-

距离曲线与能量消耗-距离曲线。

二、 问题分析

轨道列车的装载功率主要包括牵引系统和附属设施的能量，在本题中忽略附属设施提供的能量，只考虑列车自身的牵引能能量。

轨道列车在不同运行状态下的牵引能量特性如图 2-1 所示。具体来说：

- (1) 当列车处于加速状态时，电能从牵引网络中吸收。开始加速，达到要求的速度。
- (2) 列车处于巡航状态时，吸收电能克服阻力。列车可以匀速行驶。
- (3) 当列车处于滑行状态时，既不吸收也不产生电能。这个阶段行驶速度会慢慢降低。
- (4) 当列车处于制动状态时，可产生再生能量，驱动电机发电回馈牵引网。

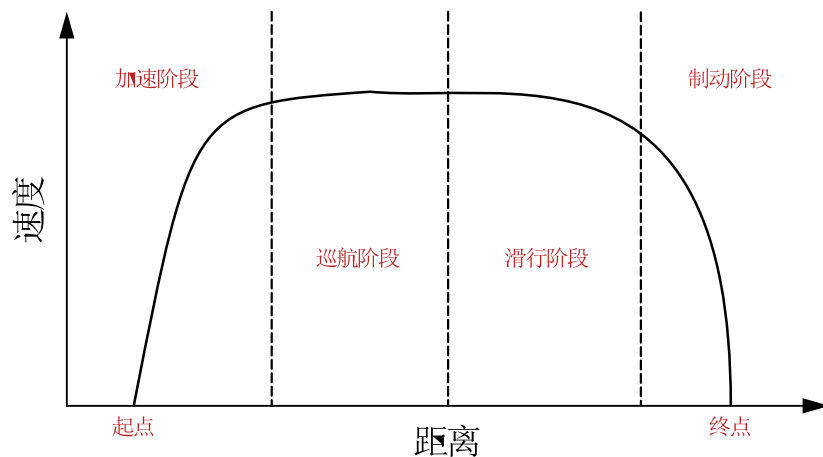


图 2-1 四种工况下站间速度曲线

当列车处于加速工况时，由牵引力、制动力和阻力决定的行驶速度与列车装载功率的关系为：

$$P_{R,i,t} = F_{q,i,t} v_{i,t} / \eta + P_{i,aux} \quad (2.1)$$

列车处于巡航工况时，由牵引力、制动力和阻力决定的行驶速度与列车装载功率的关系为：

$$P_{R,i,t} = (f_{i,t} + w_{i,t}) v_{i,t} \eta + P_{i,aux} \quad (2.2)$$

列车处于制动工况时，由牵引力、制动力和阻力决定的行驶速度与列车装载功率的关系为：

$$P_{R,i,t} = -F_{b,i,t}v_{i,t}\eta + P_{i,aux} \quad (2.3)$$

列车牵引力由电动机产生，与列车同向，其特性曲线为：

$$F_{q,i,t} = u_f f(v_{i,t}, E_i) \quad (2.4)$$

其中 $u_f \in [0,1]$ 。

列车的制动力是由电机产生的与列车方向相反的制动力，其特性曲线为：

$$F_{b,i,t} = u_b b(v_{i,t}, E_i) \quad (2.5)$$

其中 $u_b \in [0,1]$ 。

列车运行过程中的阻力由基本阻力和附加阻力组成。附加阻力包括斜坡阻力、弯道阻力和隧道阻力等，本赛题只考虑基本阻力和斜坡阻力。基本阻力和与列车行驶速度的函数关系由式（1.1）给出。下面给出斜坡阻力计算过程。

附件一中给出路况坡度变化表，我们作出 XEQ 站到 SMKXY 站的路况变化图如图 2-2 所示。

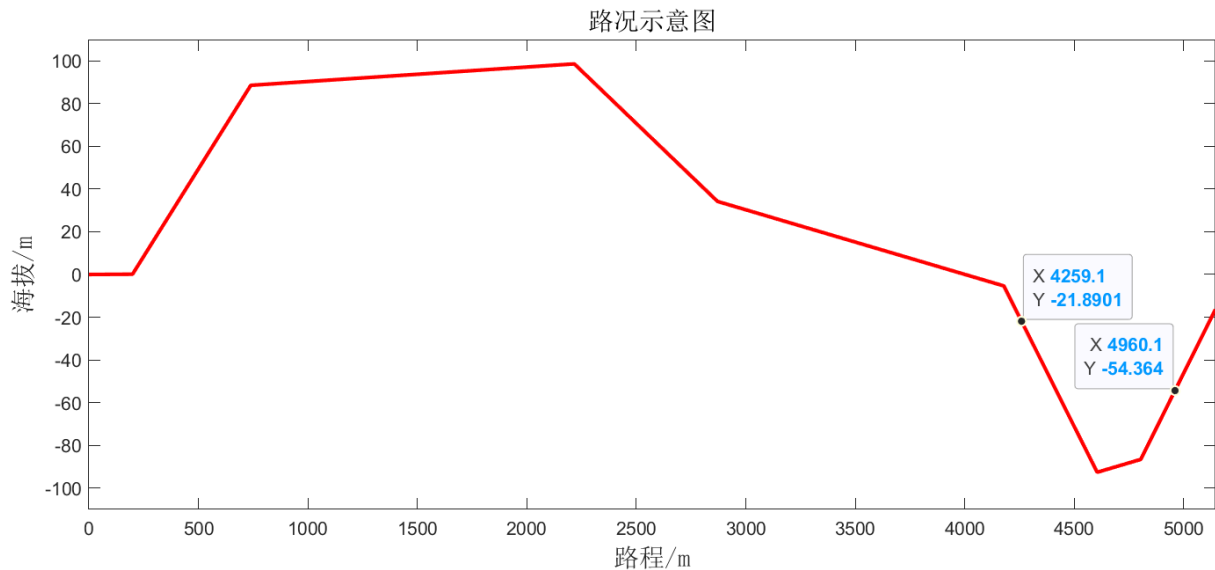


图 2-2 XEQ 站到 SMKXY 站的路况变化图

其中，设 XEQ 站的高度为参考平面（海拔为 0m）。路程中 4259.1—4960.1m 区间限速 86km/h，其余区间限速 100km/h。

经过简单受力分析，计算在不同坡度下的斜坡阻力，作出斜坡阻力-路程图如图 2-3 所示。

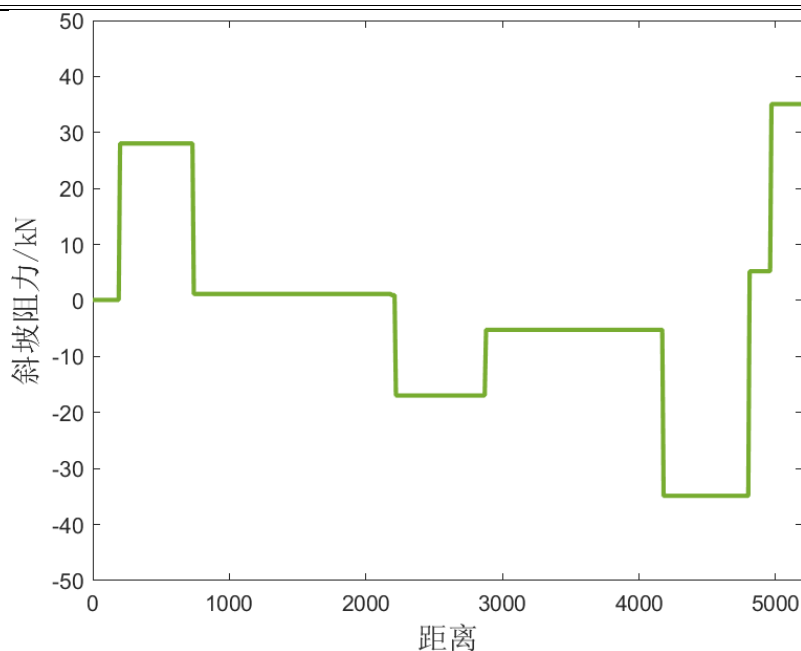


图 2-3 斜坡阻力-路程曲线

列车在站间运行时，在确保按行车计划准时到站的前提下，考虑乘客舒适度，转弯等因素，可以根据需要来控制 and 调整速度，如图 2-4 所示。

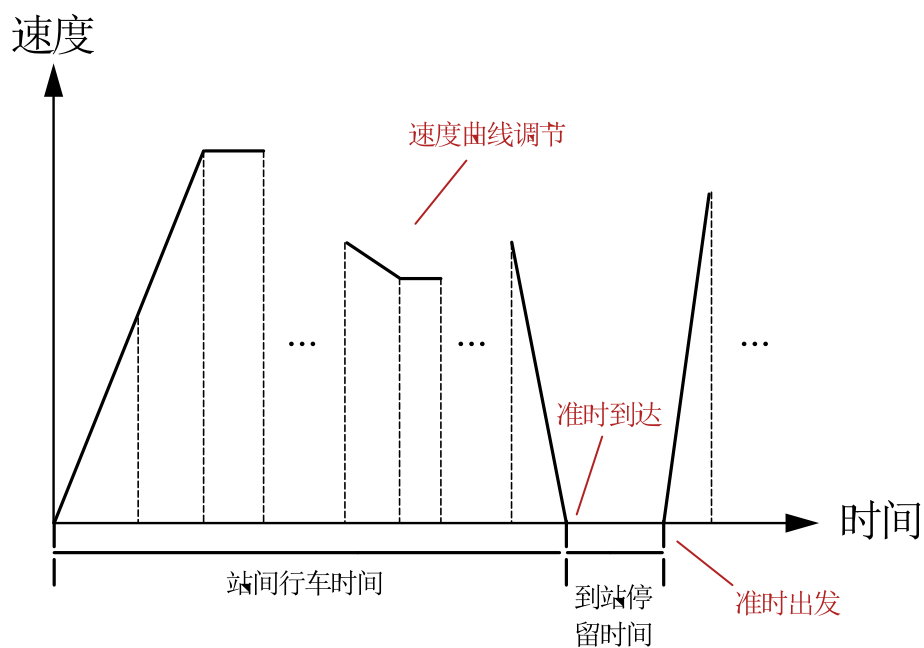


图 2-4 可控列车的速度-时间曲线

2.1 问题 1 的分析

问题 1 要求对列车在站间的运行过程进行建模仿真，模拟列车在不同的运行条件下的运行状态和性能，包括列车的速度、运行时间、制动力、牵引力、能耗等。同时要求能够根据不同的运行时间要求，快速求解出满足要求的高速列车运行操作策略优化方

法。此类仿真具有如下多方面的意义，如优化列车的设计、提高列车运行效率、提高列车安全性、降低列车运行成本。

问题 1 只要求对列车运行状况进行简单地建模仿真，通常可以利用 matlab 或者 python 程序仿真出相关过程。同时也可以利用 Simulink 或其他工具搭建列车运行的仿真模型，可以参考 MATLAB 提供的 Powertrain Blockset 和 Vehicle Dynamics Blockset 工具箱，或者使用专业软件如 CarMaker、CarSim 等。

问题一的难点在于如何模拟不同运行时间下列车的运行状态结果。在列车运动过程中，其运动状态复杂多变，需要通过特定的策略与准则来使列车的运动过程满足运行时间的约束。

基于以上的分析，我们可以首先对文档中给出的相关参数要求，建立相应的列车运行状态的数学模型，同时确立待优化的目标函数以及相应的约束条件。随后根据实际问题的特点，选择合适的优化方法对问题进行求解，并将结果进行分析、比较。

2.2 问题 2 的分析

问题 2 在问题 1 的基础上加入了路况信息和电机的复杂动态过程，使得问题约束条件更加复杂，也更加贴近生活实际，更具有实用性。问题 2 属于典型的混合整数非线性规划问题，通常对于这类问题，需要先根据所知条件，确立目标函数和约束函数。然后利用多种优化算法对问题进行求解，如遗传算法、分支定界算法等。同时问题 2 也加入了对于能耗的考虑，通过评估不同列车设计和运行方案的成本效益，从而降低列车的运行成本。

问题 2 中提供了两个附件。其中附件一为车站间不同路段的坡度分布情况和限速情况，从数据中可以看出，车站间总行程可以分为九个部分，以及两段不同的限速区域。附件二则为再生制动的牵引电机的运行范围特征，从中可以获取到列车牵引力和制动力在不同运行范围（速度范围）的变化情况以及牵引过程中的电机转化效率和制动过程中的再生制动效率。上述附件提供的数据，需要加入到列车运动状态变化的分析中去。

问题 2 相比问题 1 增加了需要考虑的因素，包括限速、坡度情况、电机的动态特性、储能等。同时也增加了在保证列车准时到站的前提下，减少整个运行中能耗的优化目标。这使得在分析、建模、求解的过程中，需要考虑更为复杂的模型，以及更多的约束条件和使用的求解算法。

鉴于以上的分析，我们将首先在问题 1 的基础上，对模型增加更多的约束条件，同

时改变目标函数，转化目标的可行域，利用空间遍历搜索的方法求解。最后对求解结果进行比较、分析。

2.3 问题 3 的分析

问题 3 设立了一个新的场景，要求面对路程中的突发情况，能够在保持列车节能运行的条件之下，快速的调整原先的运行方案，使其满足最新的运行时间要求。在现实生活中，突发情况是时有发生，设计一种快速的鲁棒性高的列车运行状态规划算法，可以评估列车在不同的运行条件下的安全性能，从而提高列车的安全性。

问题 3 与问题 2 类似，都属于是典型的混合整数非线性规划问题。只需要在问题 2 的基础上，增加一些初始限定条件即可，其建模、求解过程与问题 2 类似。首先列车发生事故前，即前 2000m，采用问题 2 提出的节能运行操作算法。在事故发生后，根据当前数据，重新运用问题 2 中的算法，确立新的节能运行操作策略。

三、模型假设

1. 问题 1 中列车未受到附加阻力的作用，问题 2 和问题 3 列车受到的附加阻力只有坡道附加阻力，忽略曲线附加阻力和隧道附加阻力的影响。
2. 将列车当作一个粒子，不考虑列车长度的影响。
3. 不考虑天气气候以及信号对列车运行的影响。
4. 不考虑列车运行的舒适性。
5. 忽略列车辅助设施的负载功率。
6. 列车运行时的能量转化效率题目中未给出，假设为 1。
7. 忽略列车运行状态与储能装置输出和牵引系统负载之间的相互作用。
8. 列车的牵引力和制动力稳定不会随意波动。

四、定义与符号说明

符号定义	符号说明
F	合力
F_q	牵引力
F_b	制动力

f	基本阻力
v	速度
P_{aux}	列车辅助设施的负载功率
a	加速度
m	列车质量
e	回转质量系数
v_{max}	最大限速
s	巡航路程
L	总路程长度
d_{ki}	第 i 段的距离
W	能量
P	功率
t_i	第 i 个距离点的到达时间
Δd	采样间隔距离
v_i	第 i 个距离点的速度
$F_{tra,i}$	第 i 个距离点的牵引力
f_g	坡度阻力
W_i	第 i 个小区间的能量消耗
y_1	电机牵引效率因子
y_2	电机再生制动效率因子
v_{tra}	牵引的恒转矩和恒功率区切换速度点
v_{bra}	制动的恒转矩和恒功率区切换速度点
F_{tra}	最大牵引力
$F_{bra,i}$	第 i 个距离点的制动力
F_{bra}	最大制动力
Δt	相邻距离点之间的时间差
a_i	第 i 个距离点的加速度

五、模型的建立与求解

5.1 问题 1 的模型建立与求解

5.1.1 列车以最短时间到达站台 B 模型的建立

1. 目标函数一：速度-距离曲线

列车在车站之间行驶通常包括牵引、巡航、惰行和制动四种工况以达到节能的目的，分别对应列车加速、匀速运行、减速和制动减速四个过程。为使列车能以最短时间到达站台 B，本题中将不考虑惰行情况。列车行驶过程中会受到牵引力、基本阻力、坡道阻力、曲线阻力、隧道阻力、制动力等力的作用。因问题 1 中没有明确说明列车行驶路况，因此我们将按照直线情况进行建模，在本题中忽略坡道阻力、曲线阻力、隧道阻力等力的作用，只存在牵引力、基本阻力和制动力。那么可以得到列车所受的合力 F 为：

$$F = F_q - F_b - f \quad (5.1)$$

其中 F_q 为牵引力， F_b 为制动力， f 为列车受到的基本阻力，阻力方程为：

$$f = 2.0895 + 0.0098v + 0.006v^2 \quad (5.2)$$

其中，公式中的速度单位是 m/s, 阻力单位为 kN。

根据站间列车行驶状态分为三个子区间，牵引、巡航和制动。利用数值积分的思想，将每个子区间的速度离散为一组目标速度集 $\{v_1, v_2, v_3 \cdots v_{q-1}, v_q\}$ ， v_q 列车最大限速，设在子区间 k 的长度为 d ，加速阶段的行驶距离为 d_{k1} ，恒速阶段的行驶距离为 d_{k2} ，减速阶段的行驶距离为 d_{k3} ，总距离 $L = d_{k1} + d_{k2} + d_{k3}$ 。我们可以分为三种情况计算：

(1) 当 $v_{k-1} < v_k$ 且 $v_{k-1} < v_k$ 时，列车牵引加速至 v_k ，在牵引加速阶段有：

$$d_{k1} = \int_{v_{k-1}}^{v_k} \frac{mv}{F_q - f} dv \quad (5.3)$$

$$t_{k1} = \int_{v_{k-1}}^{v_k} \frac{m}{F_q - f} dv \quad (5.4)$$

(2) 当速度达到 v_k 后，恒速运行，时间为：

$$t_{k2} = d_{k2} / v_k \quad (5.5)$$

(3) 当 $v_{k-1} > v_k$ ，列车制动减速至 v_k ，在制动减速阶段有：

$$d_{k3} = \int_{v_{k-1}}^{v_k} \frac{mv}{|-F_b - f|} dv \quad (5.6)$$

$$t_{k3} = \int_{v_{k-1}}^{v_k} \frac{m}{|-F_b - f|} dv \quad (5.7)$$

式中， m 为列车质量。

设列车以最短时间到达站台 B 的时间为 T ，要在最短运行时间上分别增加 10s、20s、50s、150s、300s，我们可以采用两种方法：

方法一，在巡航和制动过程之间增加惰行过程，即列车行驶中不存在牵引力与制动力，只存在基本阻力使列车自然减速。

方法二，牵引过程中速度未达到规定的最大值就进入巡航状态，即列车不以最大速度进行匀速行驶。在本题中，我们选择第二种方法。

2. 目标函数二：牵引制动力-距离曲线

列车行驶过程中有牵引、惰行、制动三种状态，因此我们也可以分为三种情况进行计算：

(1) 牵引状态下，列车受到牵引力和基本阻力的作用，可得：

$$F = F_q - f \quad (5.8)$$

(2) 惰行状态下，列车保持匀速运动，因此牵引力与基本阻力大小相等，可得：

$$F = F_q - f = 0 \quad (5.9)$$

(4) 制动状态下，列车受到制动力和基本阻力的作用，可得：

$$F = -F_b - f \quad (5.10)$$

3. 目标函数三：时间-距离曲线

由式 (5.4)、(5.5)、(5.7) 分析可得列车行驶过程中时间与距离的关系。

4. 目标函数四：能量消耗-距离曲线

当列车处于牵引、巡航和制动工况时，由牵引力、制动力和阻力确定的行驶速度与列车负载功率之间的关系表示为

$$P = \begin{cases} F_q v / \eta + P_{aux} & \text{牵引} \\ f v / \eta + P_{aux} & \text{巡航} \\ -F_b v / \eta + P_{aux} & \text{制动} \end{cases} \quad (5.11)$$

式中， η 表示能量转换效率，题中未给出值，因此建模中默认为 1， f 为列车运行过程中的阻力，由基本阻力和附加阻力组成，在本题中即基本阻力。 P_{aux} 表示列车辅助设施的负载功率。列车的牵引力由电机在与列车相同的方向上产生，其特征曲线公式为：

$$F_q = y_1 f_q \quad (5.12)$$

其中 $y_1 \in [0,1]$ ，表示电机牵引效率因子， f_q 表示列车产生的牵引力， F_q 表示实际产生牵引力。列车的制动力由电机在与列车相反的方向上产生，其特征曲线公式为：

$$F_b = y_2 f_b \quad (5.13)$$

其中 $y_2 \in [0,1]$ ，表示再生制动效率因子， f_b 表示列车产生的制动力， F_b 表示实际产生制动力。

当列车处于牵引模式时，牵引电机将切换到电动模式，在电动模式下，列车从牵引网中吸收电能并将其转换为动能。当列车处于再生制动条件下时，牵引电机的电气模式将转变为发电模式，将列车减速产生的动能转换为电能，反馈给直流牵引网，储存能量。

(1) 当列车处于加速状态时，电能从牵引网络吸收，开始加速以达到所需的速度。

(2) 当列车处于巡航状态时，它吸收电力以克服电阻，火车可以以恒定的速度行驶。

(3) 当列车处于惰行状态时，既不吸收也不产生电能，在此阶段，行驶速度将缓慢降低。

(4) 当列车处于制动状态时，可以产生再生能量，该再生能量驱动电动机产生可以反馈到牵引网络中的电力。

5.1.2 列车以最短时间到达站台 B 模型的求解

为满足题目要求以最短时间到达站台 B，在不考虑节省能耗的情况下，牵引力和制动力都为最大值，并且不考虑惰行状态。

1. 目标函数一单独求解：速度-距离曲线

计算时，任一时刻的加速度为：

$$a = F / m(1 + e) \quad (5.14)$$

式中， e 为回转质量系数 0.08， m 为列车质量 176.3t， F 为合力由 (5-1) 可得。

利用列车开始时和列车到达时速度都为 0，最大限速 $v_{\max} = 100\text{km/h}$ ，总路程 L 为 5144.7m，牵引力最大值为 310kN，制动力最大值为 760kN，我们可以求得列车的巡航路程 s 为：

$$s = L - d_{k1} - d_{k3} \quad (v = 100\text{km/h}) \quad (5.15)$$

牵引距离以及制动距离和速度的关系由 (5-3) 和 (5-6) 计算可得。选取多个采样点在 MATLAB 中进行仿真试验得到图 5-1，获取了速度-距离曲线。

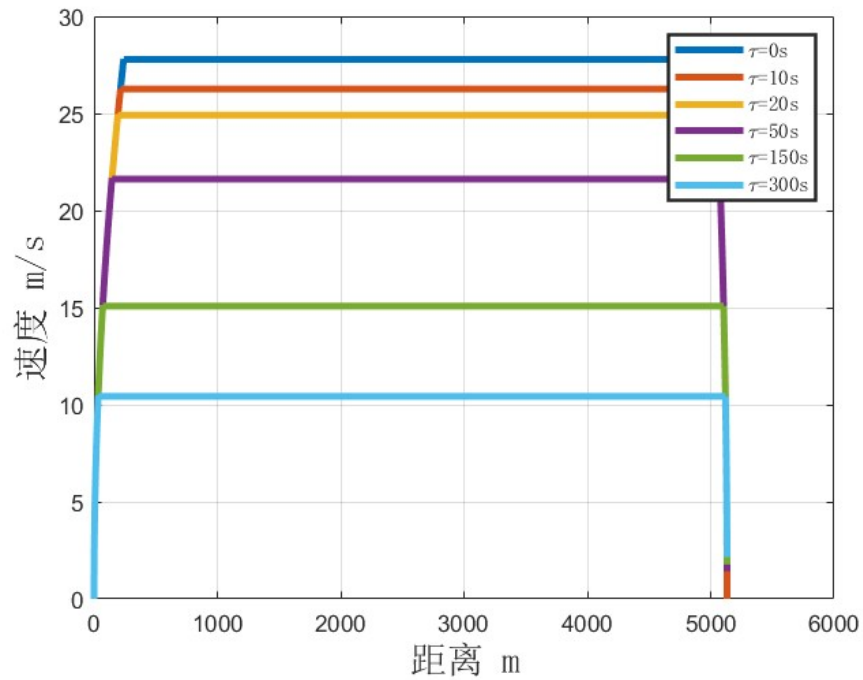


图 5-1 速度-距离曲线

改变行驶的最大速度可以延迟列车的到达时间。

2. 目标函数二单独求解：牵引制动力-距离曲线

列车行驶过程中，牵引和巡航的过程会受到牵引力的作用，而制动过程受到制动力的作用。我们得到图 5-2 和图 5-3，表明牵引力和距离的关系以及制动力和距离的关系。

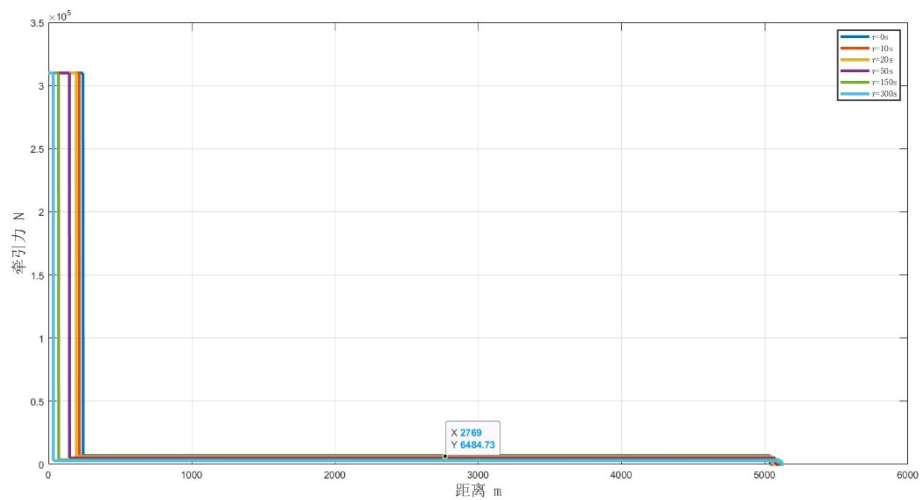


图 5-2 牵引力-距离曲线

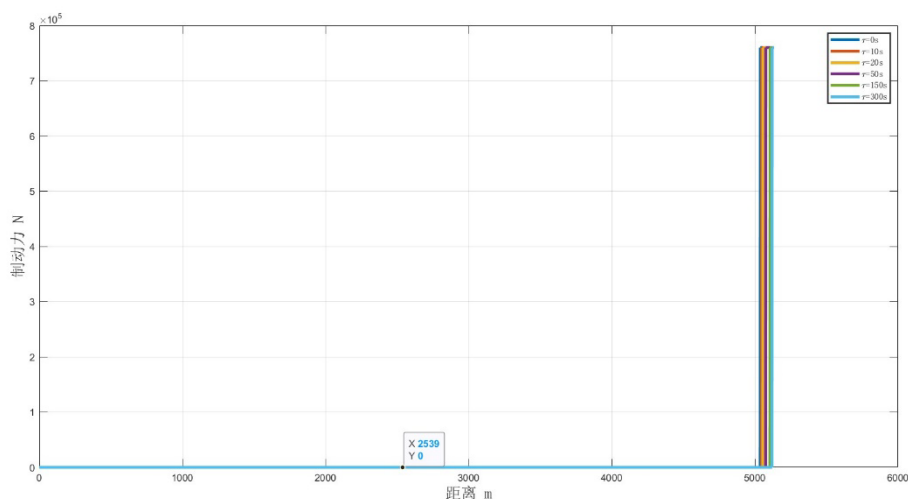


图 5-3 制动力-距离曲线

3. 目标函数三单独求解：时间-距离曲线

与速度-距离曲线类似，我们可以求得列车巡航时间 t 为

$$t = \frac{L - d_{k1} - d_{k3}}{v_{\max}} \quad (5.16)$$

牵引消耗时间以及制动消耗时间和速度的关系由（5-4）和（5-7）计算可得。再有速度和距离的关系可以联立，最终得到图 5-4。

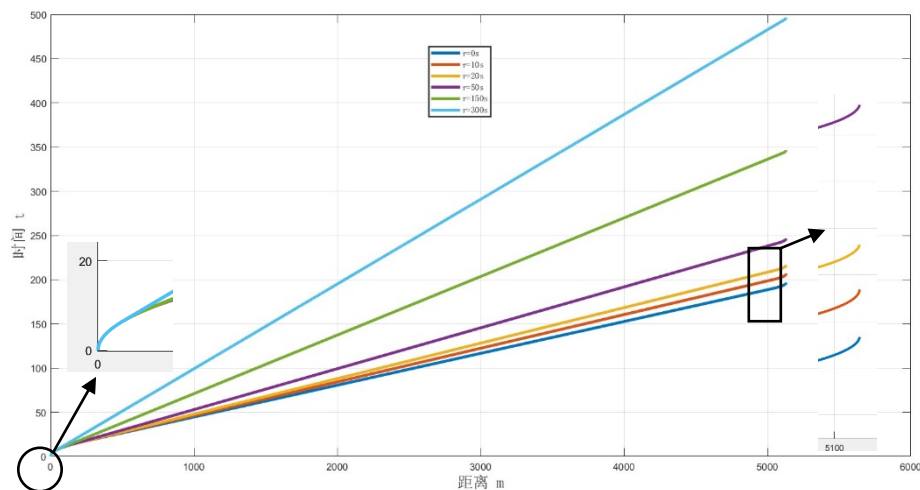


图 5-4 时间-距离曲线

4. 目标函数四单独求解：能量消耗-距离曲线

由（5-11）可以得到功率 P 在列车不同运行状态时的公式，再由功率和能量的关系 $W = Pt$ ，我们可以求得列车运行的能耗。牵引消耗能量，制动可以产生能量。能量消耗与距离的关系如图 5-5 所示。

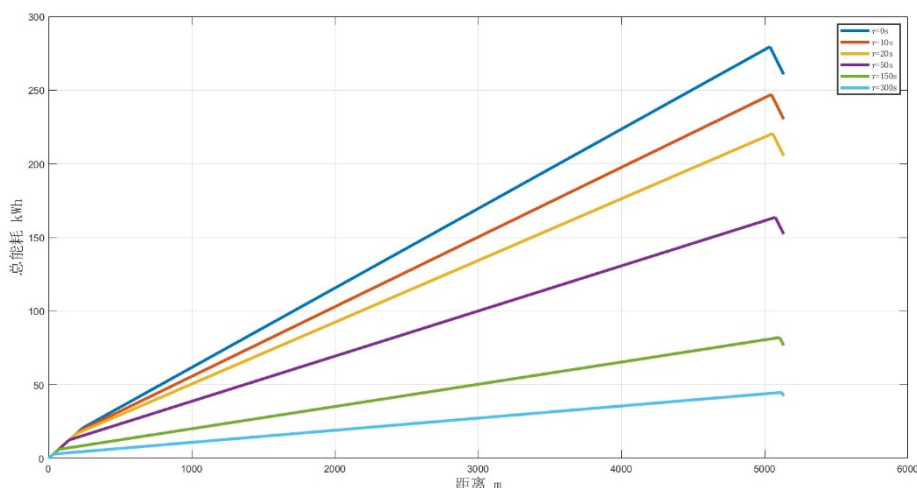


图 5-5 能量消耗-距离曲线

5.1.3 模型合理性分析

由上述模型分析可知，若牵引力没有达到最大值，那么列车速度增加幅度会小于牵引力最大时的增加幅度，当以小于 100km/h 的速度进行匀速运动时，耗时增加。当牵引力最大的情况下，使列车的速度达到 100km/h，然后进行制动减速，列车将会以最短时间到达站台 B。此外，由文献[6]可知，当限速值低于列车最佳巡航速度时，列车以最大允许速度行驶是一种最为节能的操纵方式。由此可以证明我们所采用的方法是最优，可以让时间达到最短的方法。

5.2 问题 2 的模型建立与求解

5.2.1 节能列车控制优化模型的建立

问题 2 需要在保证准时到达的前提下，减少能源消耗。而现实生活中，动车组列车在运行期间主要采用最大牵引、匀速、惰行和最大制动的操纵模式以达到节能目的，本题中高数列车节能运行策略优化正是基于以上方式。

本题中需要考虑路况中坡度的影响，附件一中数据表示的不同路况示意图和坡度阻力示意图 5-6 所示。

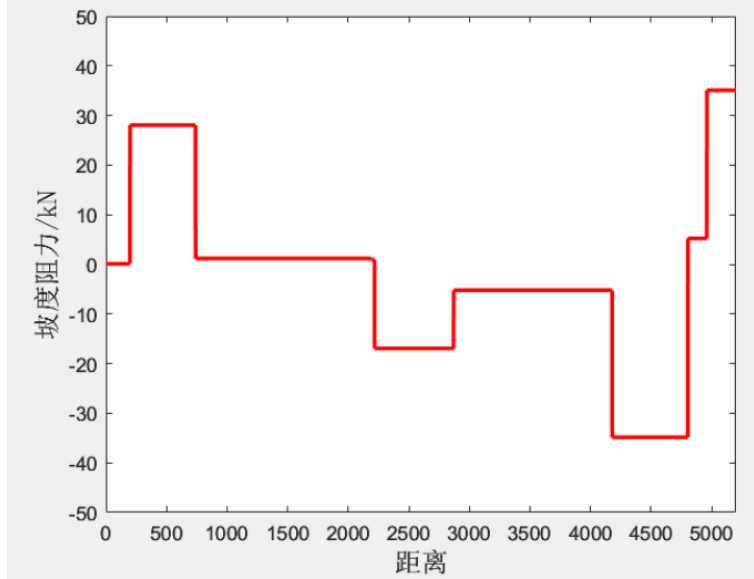


图 5-6 坡度阻力-距离曲线图

数据中坡度千分坡度与坡度阻力的换算公式如下，其中为坡度角度，为千分坡度，为列车质量，为重力加速度，为坡度阻力。因此在计算列车所受合力的情况下，应该加入坡度阻力的影响。

$$\theta = \arctan(\alpha \times 10^{-3}) \quad (5.17)$$

$$f_g = mg \sin \theta \quad (5.18)$$

其中 θ 为坡度角度， f_g 为坡度阻力。电机的动态特性也是本题中需要考虑的部分。

交流电机运行范围一般能够分为两个阶段：

恒转矩区：反电动势小于最大电压，电机运行只受到最大电流限制，可以输出最大转矩，最大输出功率与转速成正比上升；

弱磁区(恒功率区)：反电动势逼近最大电压，采用弱磁控制维持电压平衡，同时受到最大电压和最大电流的限制；

在牵引过程的两个阶段可以表示为如下形式：

$$f^{tra} = \begin{cases} f_1^{tra} & 0 \leq v \leq v_1^{tra} \\ f_1^{tra} \cdot \frac{v_1^{tra}}{v} & v_1^{tra} \leq v \end{cases} \quad (5.19)$$

其中 f^{tra} 为牵引力，恒转矩区 $0 \leq v \leq v_1^{tra}$ 的电机牵引力 $f_1^{tra} = 310 \text{ kN}$ ，再生电机制动力为 260 kN ，牵引与制动的恒转矩与恒功率区切换速度点分别为 $v_{fra} = 10 \text{ m/s}$ 以及 $v_{bra} = 17 \text{ m/s}$ 。

此外，电机牵引制动过程具有转换效率，本赛题考虑静态效率因子，即电机牵引效

率因子 η_{tra} 为 0.9，电机再生制动效率因子 η_{bra} 为 0.6。因此在计算电机输入功率和再生制动能时应该满足如式（5.20）和（5.21）：

$$P_{in} = P_{out} / \eta_{tra} \quad (5.20)$$

$$E_{rec} = E_{kin} / \eta_{bra} \quad (5.21)$$

其中 P_{in} 是电机输入的电功率， P_{out} 是电机输出的电功率， E_{rec} 是回收的电能， E_{kin} 制动前的车辆的动能。

对于本问题，根据站内各线路坡度条件将总路径划分为多个子区间，其中坡度值一致的连续线路为一个子区间如图 5-7 和图 5-8 所示。

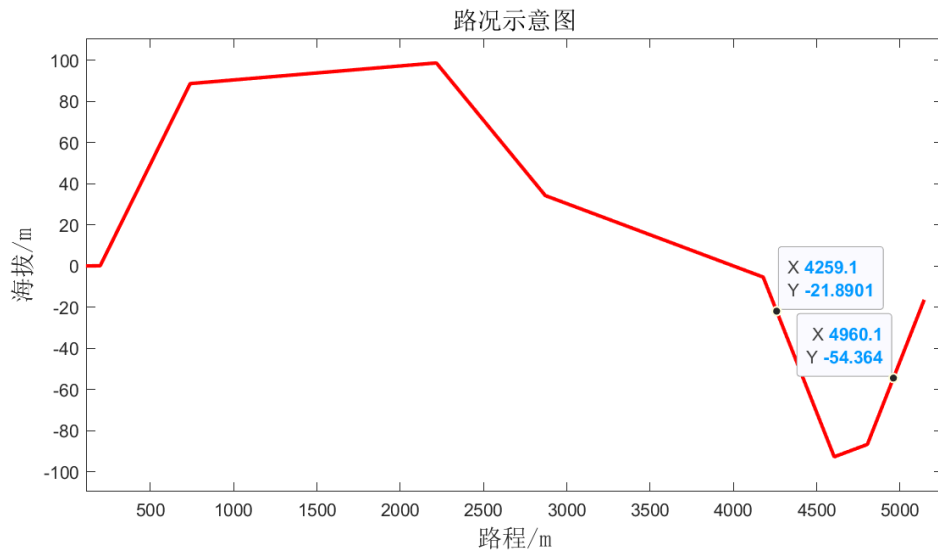


图 5-7 路况示意图

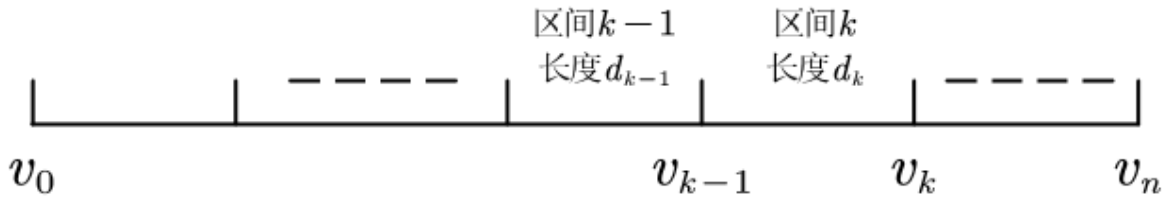


图 5-8 区间划分示意图

结合附件一和图可知，整个路段可以分为 9 个子区间，以及 10 个离散的位于子区间端点处的目标速度集合，即 $\{V_1, V_2, \dots, V_{10}\}$ 。现假设子区间 k 的长度为 d_k ，并且在子区间中有加速、减速、恒速三个运动状态，其中加速、减速阶段的行驶距离为 d_{k1} ，恒速阶段的形式距离为 d_{k2} ，同时为了统计整个路段中的运行状态，取 $\Delta d = 1m$ 为采样区间。

设立 t_i 、 $F_{tra,i}$ 、 $F_{bra,i}$ 、 x_i 、 W_i 、 v_i , $i=1,2,3,\dots,N$ 为第 i 个采样点中各状态值，不难得出 $i=0$ 时， $F_{tra,i} = F_{tmax}$ ，其余状态值为 0。在每一个采样区间中的运动状态也只有加速、

减速、恒速三种，在不同运动状态中采样点状态值跟新规则如下：

(1) 匀速运动

此时列车受力均衡，第 i 个采样点运动状态更新情况能够表示为：

$$t_i = t_{i-1} + \Delta d / v_{i-1} \quad (5.22)$$

$$v_i = v_{i-1} \quad (5.23)$$

$$F_{tra,i} = \begin{cases} -[f + f_g(v_i)] & f + f_g(v_i) > 0 \\ 0 & f + f_g(v_i) \leq 0 \end{cases} \quad (5.24)$$

$$W_i = \begin{cases} F_{bra,i} v_i \Delta d / y_2 & f + f_g(v_i) > 0 \\ y_1 F_{tra,i} v_i \Delta d & f + f_g(v_i) \leq 0 \end{cases} \quad (5.25)$$

式中 t_i 第 i 个距离点的到达时间， Δd 为采样间隔距离， v_i 为第 i 个距离点的速度， $F_{tra,i}$ 为第 i 个距离点的牵引力， $F_{bra,i}$ 为第 i 个距离点的制动力。注意到由于坡度的引入，会导致总阻力方向可能与运动方向相反或者一致，运动方向一致时，此时应该施加制动力，而当总阻力方向与运动方向相反时，则应该施加牵引力。

(2) 加速运动

此时对列车施加最大的牵引力，制动力设置为 0，使列车加速运动，该过程中第 i 个采样点的运动状态更新情况能够表示为：

$$a_i = \begin{cases} (F_{tra} v_{tra} / v_{i-1} - f - f_g(v_{i-1}) / m(e+1)) & v_{i-1} \geq v_{tra} \\ (F_{tra} - f - f_g(v_{i-1}) / m(e+1)) & v_{i-1} < v_{tra} \end{cases} \quad (5.26)$$

$$v_i = \sqrt{v_{i-1}^2 + 2a_i \Delta d} \quad (5.27)$$

$$F_{tra,i} = \begin{cases} F_{tra} v_{tra} / v_{i-1} & v_{i-1} \geq v_{tra} \\ F_{tra} & v_{i-1} < v_{tra} \end{cases} \quad (5.28)$$

$$F_{bra,i} = 0 \quad (5.29)$$

$$t_i = t_{i-1} + (v_i - v_{i-1}) / a_i \quad (5.30)$$

$$W_i = \int_0^{\Delta t} F_{tra,i} v_i dt \quad (5.31)$$

式中 W_i 为第 i 个小区间的能量消耗， Δt 为邻距离点之间的时间差。注意到由于采样间隔远小于总路程，本模型中假定在采样间隔中的加速运动服从匀加速运动。因此通过上一点的受力状态，即可求解出响应的加速度，进而更新其它运行状态值。同时从上式也可以看出，在分析受力过程中，考虑了电机的复杂动态过程。在计算能量消耗的过程

中，考虑了电机牵引效率因子的影响。

(3) 减速运动

此时应该对列车施加最大的制动力，牵引力设置为 0，使得列车做减速运动，该过程第 i 个采样点的运动状态更新情况能够表示为：

$$a_i = \begin{cases} (F_{bra} v_{bra} / v_{i-1} + f + f_g(v_{i-1}) / m(e+1)) & v_{i-1} \geq v_{tra} \\ (F_{bra} + f + f_g(v_{i-1}) / m(e+1)) & v_{i-1} < v_{tra} \end{cases} \quad (5.32)$$

$$v_i = \sqrt{v_{i-1}^2 - 2a_i \Delta d} \quad (5.33)$$

$$F_{tra,i} = 0 \quad (5.34)$$

$$F_{bra,i} = \begin{cases} F_{bra} v_{tra} / v_{i-1} & v_{i-1} \geq v_{bra} \\ F_{bra} & v_{i-1} < v_{bra} \end{cases} \quad (5.35)$$

$$t_i = t_{i-1} + (v_{i-1} - v_i) / a_i \quad (5.36)$$

$$W_i = \int_0^{\Delta t} F_{bra,i} v_i dt \quad (5.37)$$

同样，由于采样间隔远小于总路程，本模型中假定在采样间隔中的减速运动服从匀减速运动。从而能够求出响应的运行状态值。

对于每一个子区间中各个采样区间具体运行操作策略规则如下

(1) 当 $V_{k-1} < V_k$ ，并且位于区间 k 时，列车需要先牵引加速至 v_k ，当速度达到目标 v_k 后，保持恒速运动，对于总路程中的采样点的具体操作如下：

如果 $abs(v_{k-1} - V_k) < \delta$ ，其中 abs 是取绝对值运算， δ 是一个很小的值。当差值小于 δ 时，可以近似认为两个值相等，此时在采样间隔内应该做匀速运动。

如果 $v_{k-1} - V_k < \delta$ ，当前速度小于目标值 V_k ，因此在采样间隔中做加速运动。

如果 $v_{k-1} - V_k > \delta$ ，当前速度大于目标值 V_k ，因此在采样间隔中做减速运动。

本问题要求尽可能的减少能源消耗，因此运动的总体过程应该尽可能地符合加速-巡航-滑行-制动四个变化规律，如图 5-9 所示。

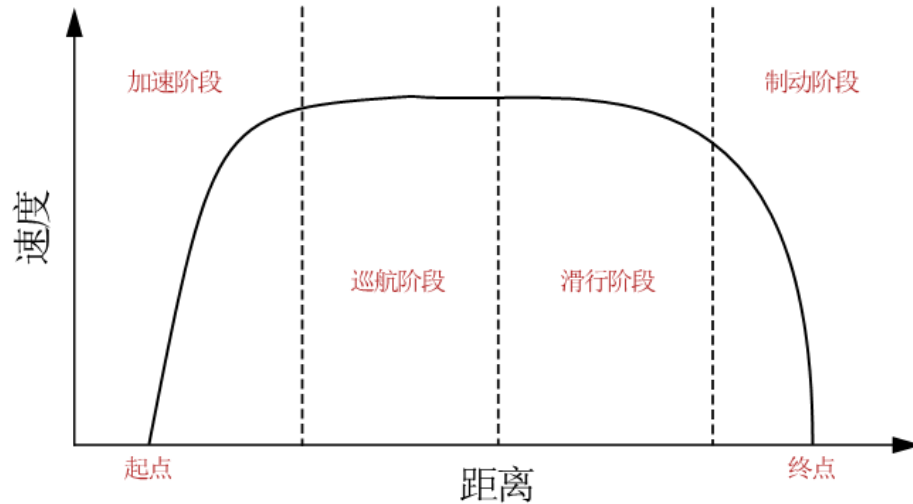


图 5-9 列车运行速度区间示意图

同时我们可以很容易的写出高速列车节能运行操纵策略优化的目标函数：

$$C = \min_{\{V_1, V_2, \dots\}} \sum_{n=1}^N W_n \quad (5.38)$$

值得注意的是，我们将目标函数中的可行域，由等参数组成的复杂的多维可行域空间转化为了离散速度序列所在的空间，通过在速度序列所在空间进行搜索，能够有效的降低算法的复杂度，以及求解效率。

5.2.2 节能列车控制优化模型的求解

按照问题 2 的要求，本部分在考虑了文档中坡度、电机的复杂动态过程等因素下，尽可能地减少能量损耗，通过对离散速度序列在一定的可行域范围内进行搜索，求解出了满足特定运行时间并且能量损耗较低的序列值。本次模型搭建、算法实现全部采用 MATLAB 实现，编程代码详见附件。下面对仿真结果进行说明介绍。

1. 速度-距离曲线

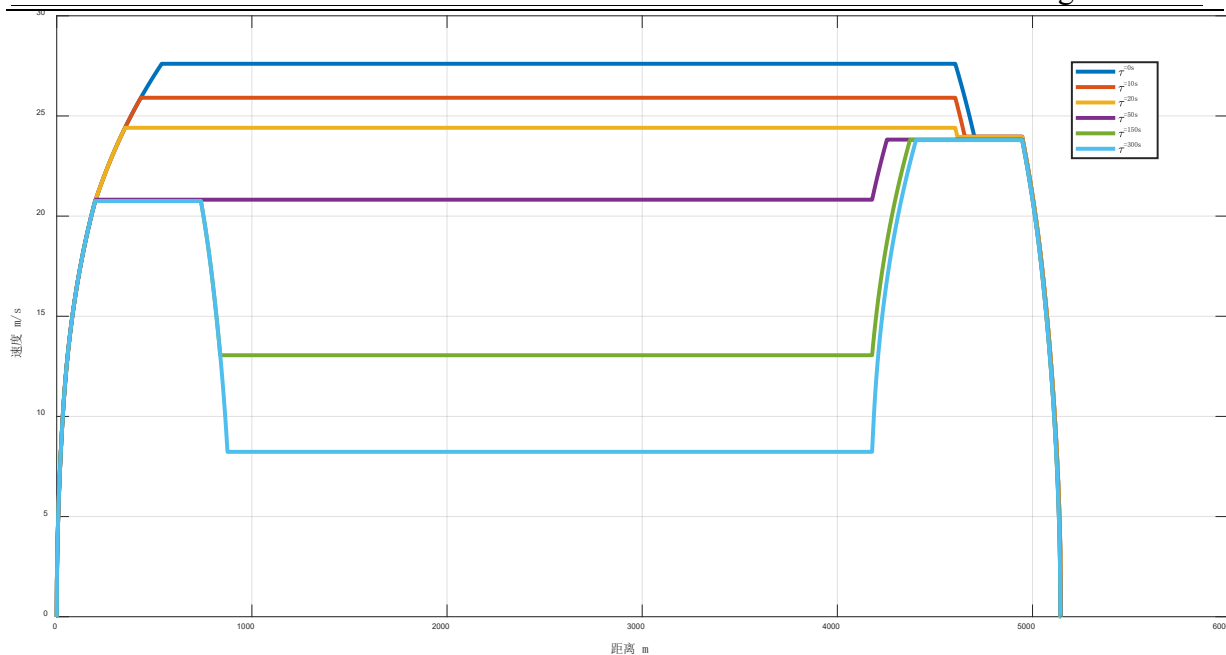


图 5-10 速度-距离曲线

图 5-10 是列车在不同运行时间下速度-距离仿真图，可以看出对于最短时间到达的曲线，采用了加速-恒速-减速的运行模式，恒速时保持当前路段的最大限速值。对于运行时间的增加，通过改变离散速度序列的取值，通过空间搜索的方法，得到能耗较低的列车节能运行操作策略。

2. 速度-牵引制动力曲线

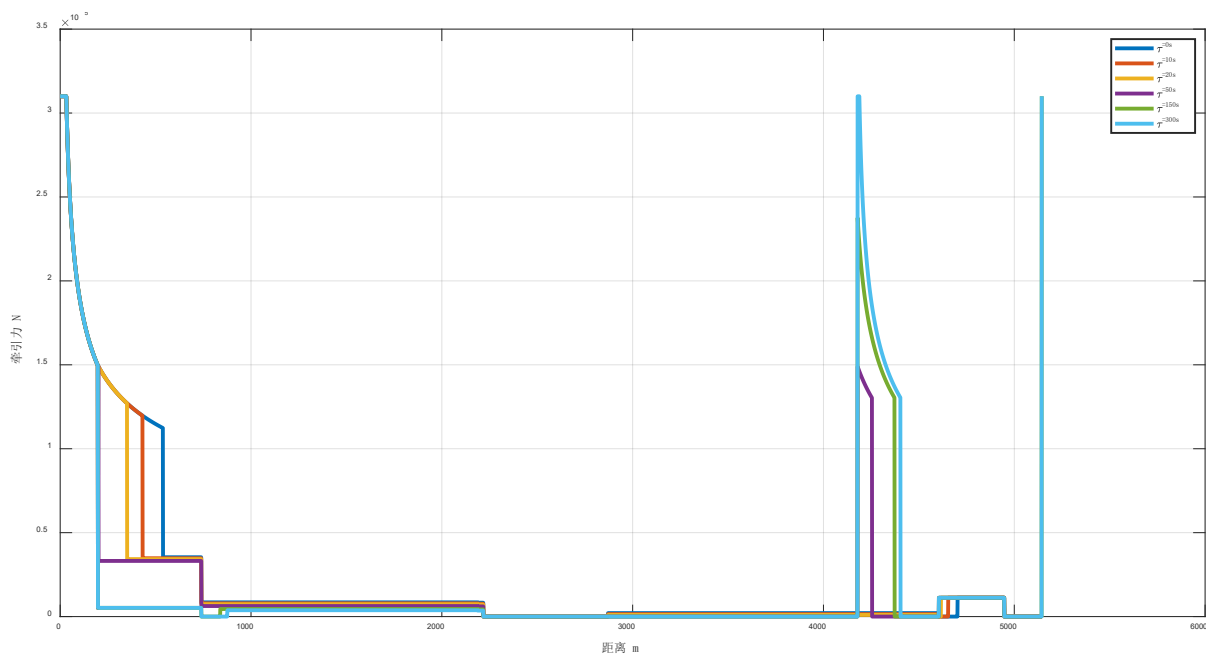


图 5-11 速度-牵引制动力曲线

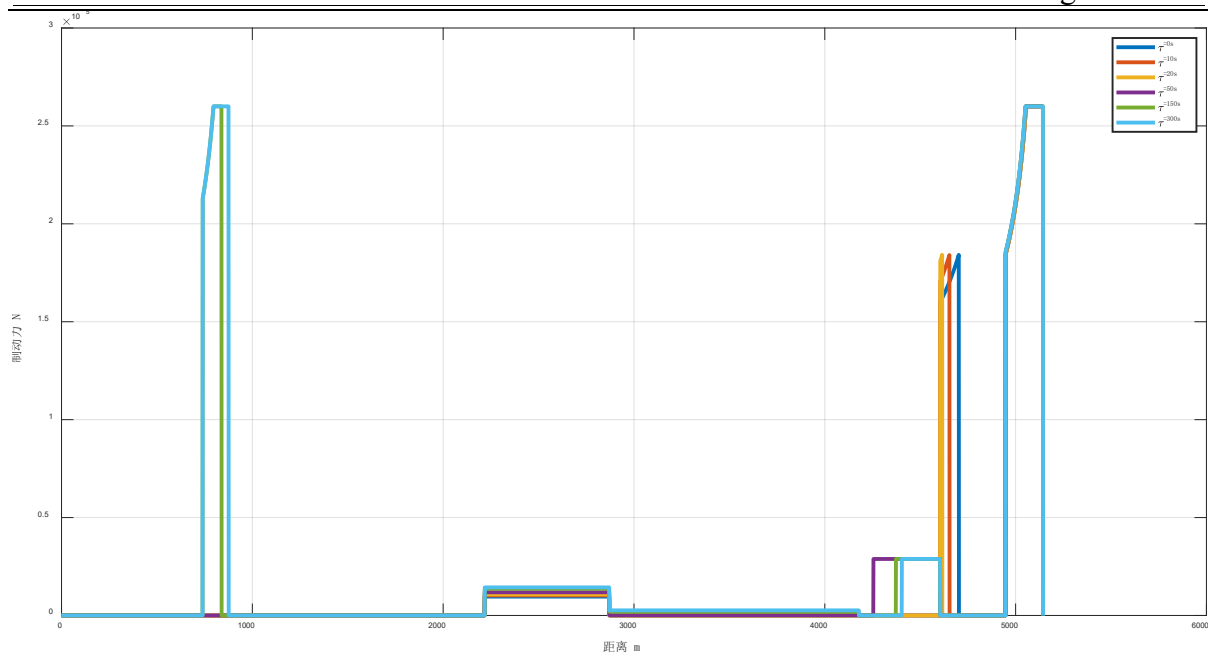


图 5-12 速度-牵引制动力曲线

图 5-11 和图 5-12 是列车在不同运行时间下牵引制动力-距离仿真图，总体运动趋势是，快速加速到最大限速值，保持匀速运动，降低到新的最大限速制，快速制动到终点。

3. 时间-距离曲线

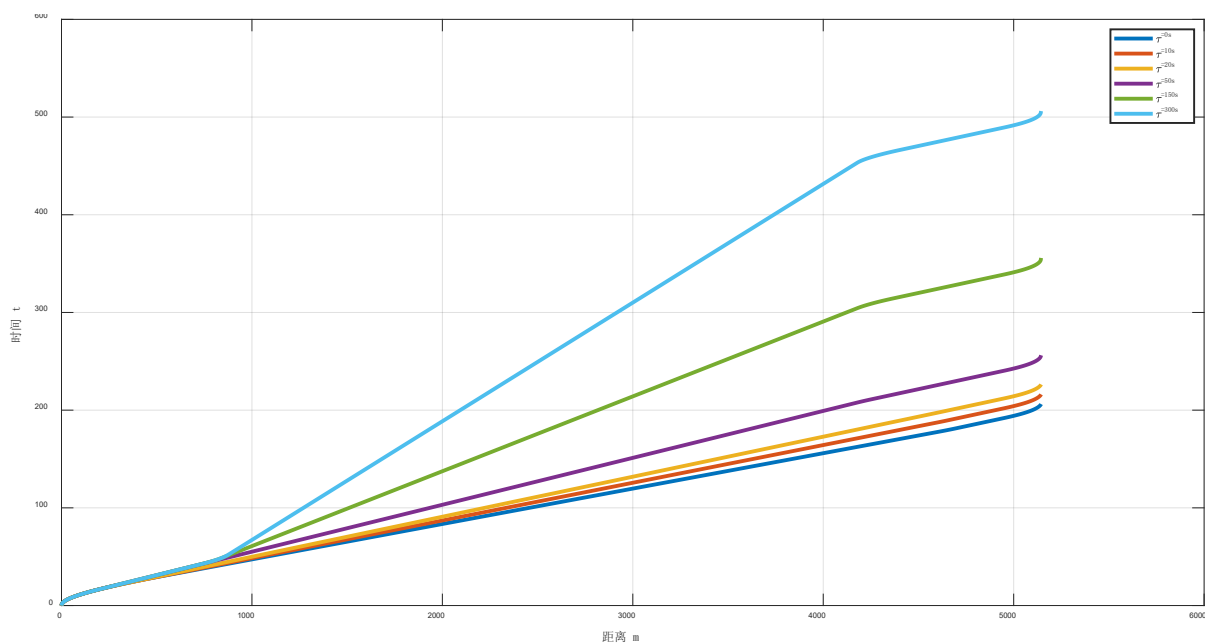


图 5-13 运行时间-距离曲线

图 5-13 是列车的运行时间-距离仿真图。从图中可以看出，时间随距离的变化趋势符合速度的变化趋势，同时最终到达时间点满足问题中列出的 6 种运行时间。其中最短

到达站台 B 的时间为 206s, 另外五组分别为在最短运行时间上增加 10s、20s、50s、150s、300s。

4. 能量消耗-距离曲线

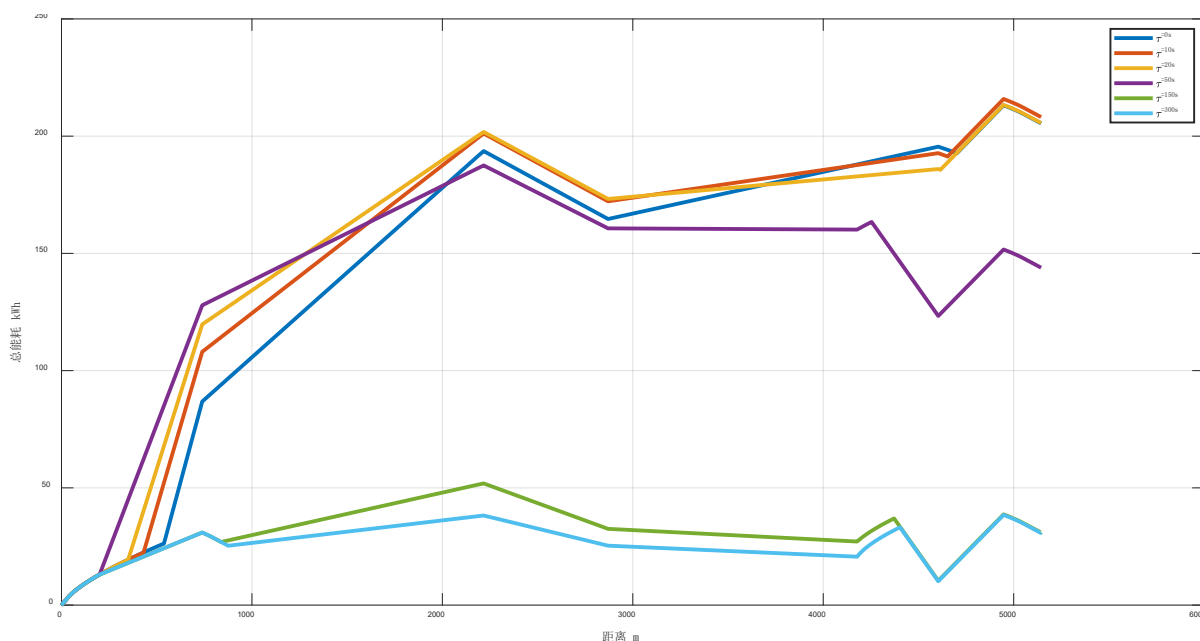


图 5-14 能量消耗-距离曲线

图 5-14 是列车的能量消耗-距离曲线。其中斜率为负数的线段，代表了在再生制动过程中电机将部分动能转化为了电能。从图中可以看出，再生制动技术能够有效的提高列车的能量利用效率，延长电池的续航里程。

5.2.3 结果

由上述实验结果可知，本文提出的模型以及求解算法，能够在降低能源消耗和准时到站的前提下，有效、合理的解决规划高速列车运行操作策略的问题。能够在现实生活中起到优化列车设计、提高列车运行效率、提高列车安全性、降低列车运行成本多种用处。

5.3 问题 3 的模型建立与求解

5.3.1 实时运行状态规划模型的建立

本题采用的模型及求解方法与问题 2 基本一致，总路程的前 2000m，采用问题 2 所提模型，搜索得出最优离散目标速度序列对应的运行操作策略。事故发生后，改变最优离散目标速度序列的后半段值，以使得运行时间增加 60s，同时也降低了能量消耗。

5.3.2 实时运行状态规划模型的求解

1. 速度-距离曲线

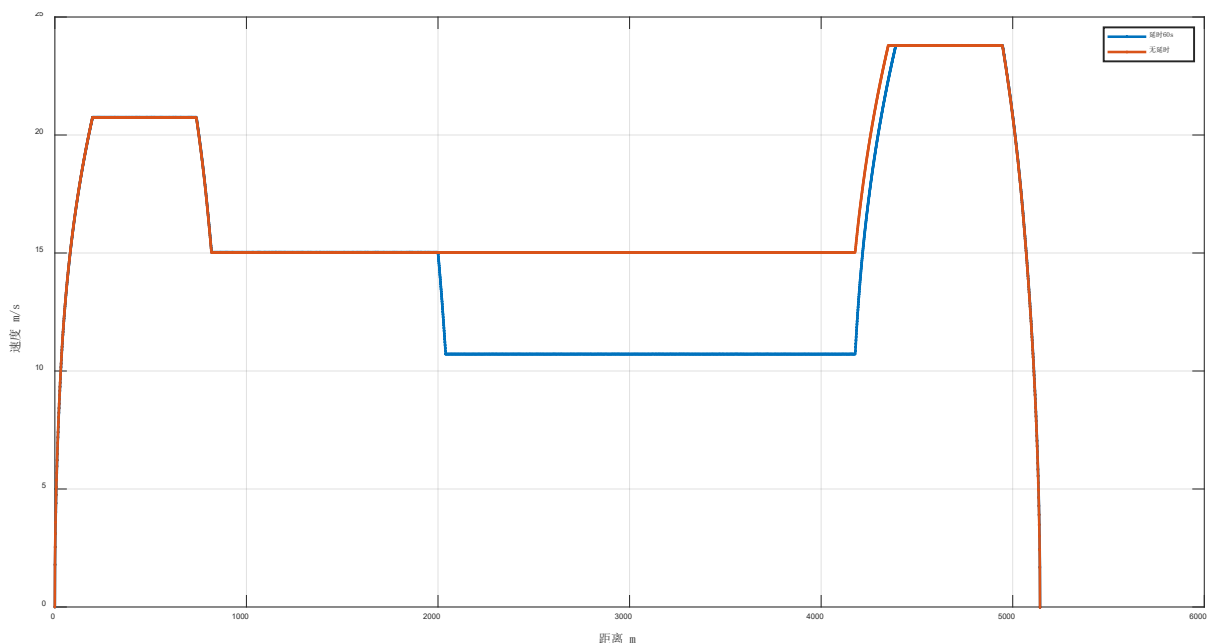


图 5-15 速度-距离曲线

图 5-15 是列车的速度-距离仿真图。黄线是在没有运行延时且运行时间为 320s 的列车运行情况，蓝线则是在 2000 米处遭遇突发事故后，采取新的最优离散目标速度序列应对时间延迟的运行情况。

2. 牵引制动力-距离曲线

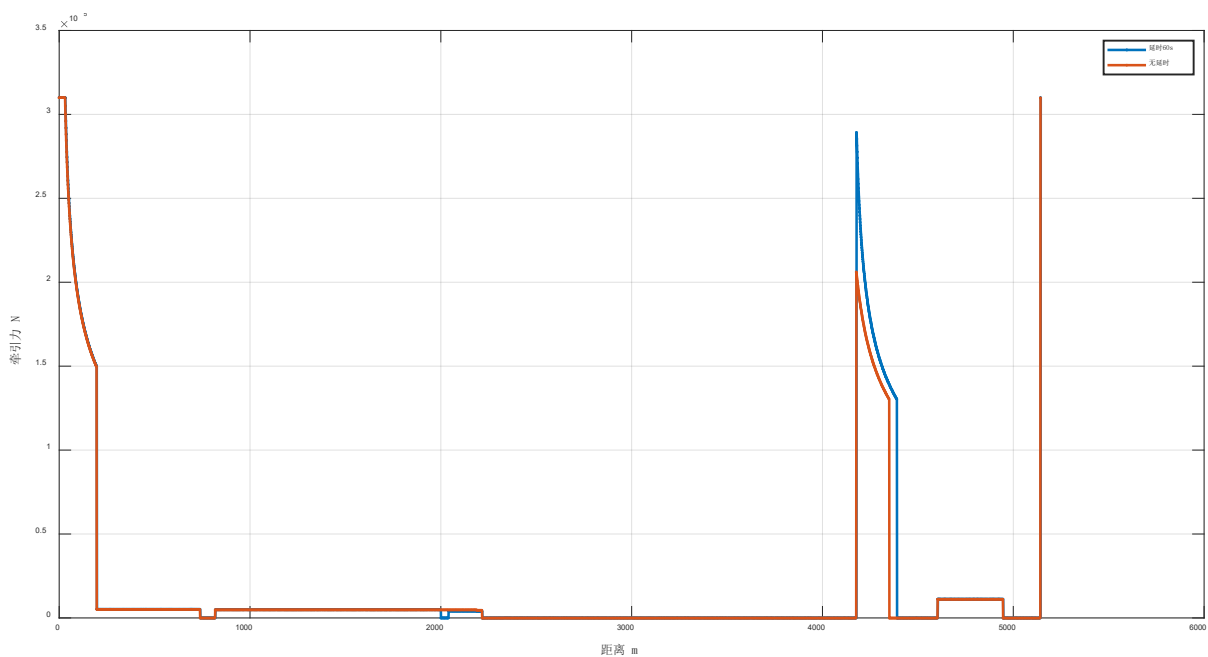


图 5-16 牵引制动力-距离曲线

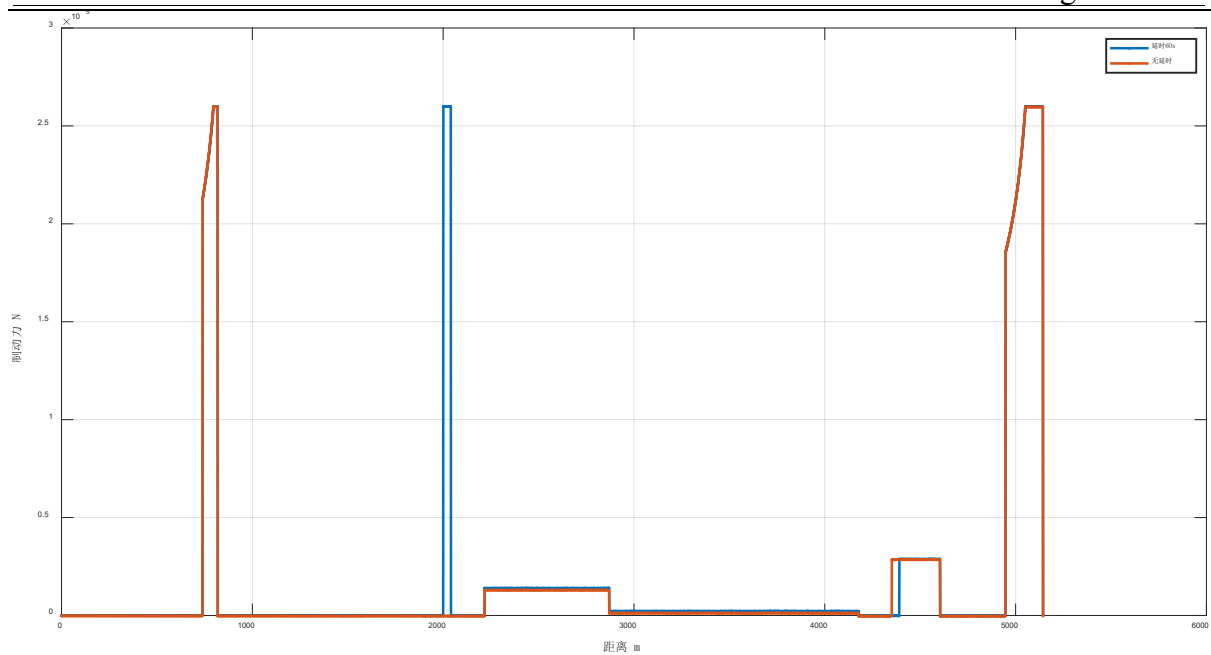


图 5-17 牵引制动力-距离曲线

图 5-16 和图 5-17 是列车的牵引制动力-距离仿真图。

3. 时间-距离曲线

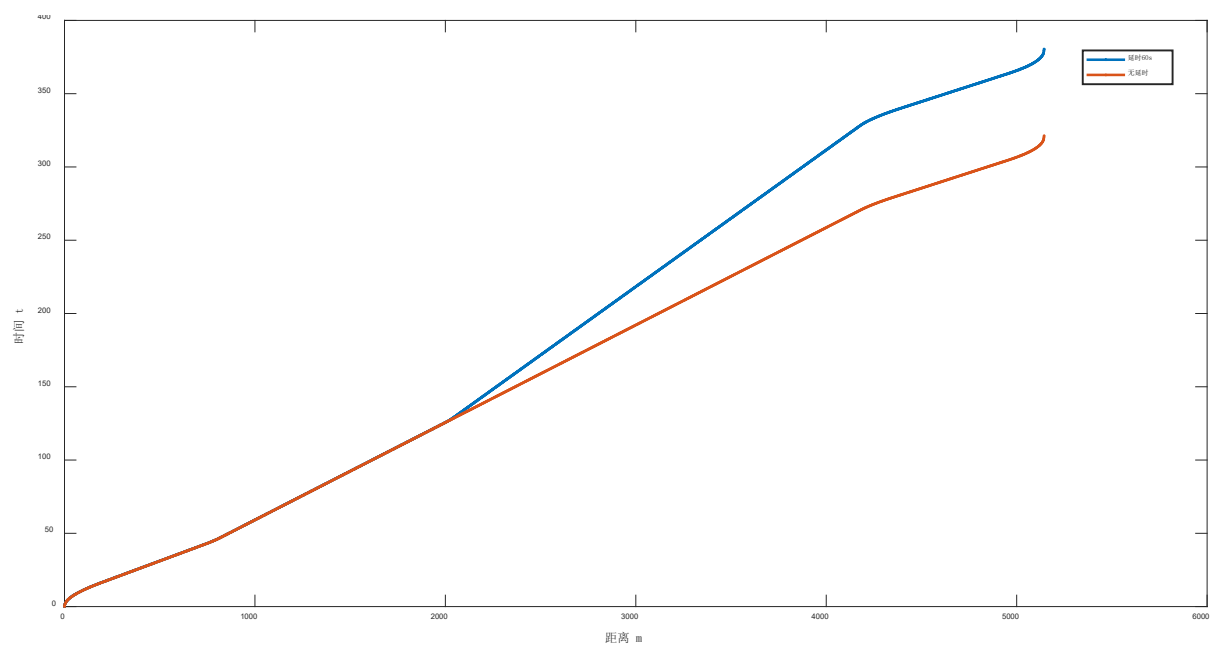


图 5-18 时间-距离曲线

图 5-18 是列车的速度-距离仿真图。

4. 能量消耗-距离曲线

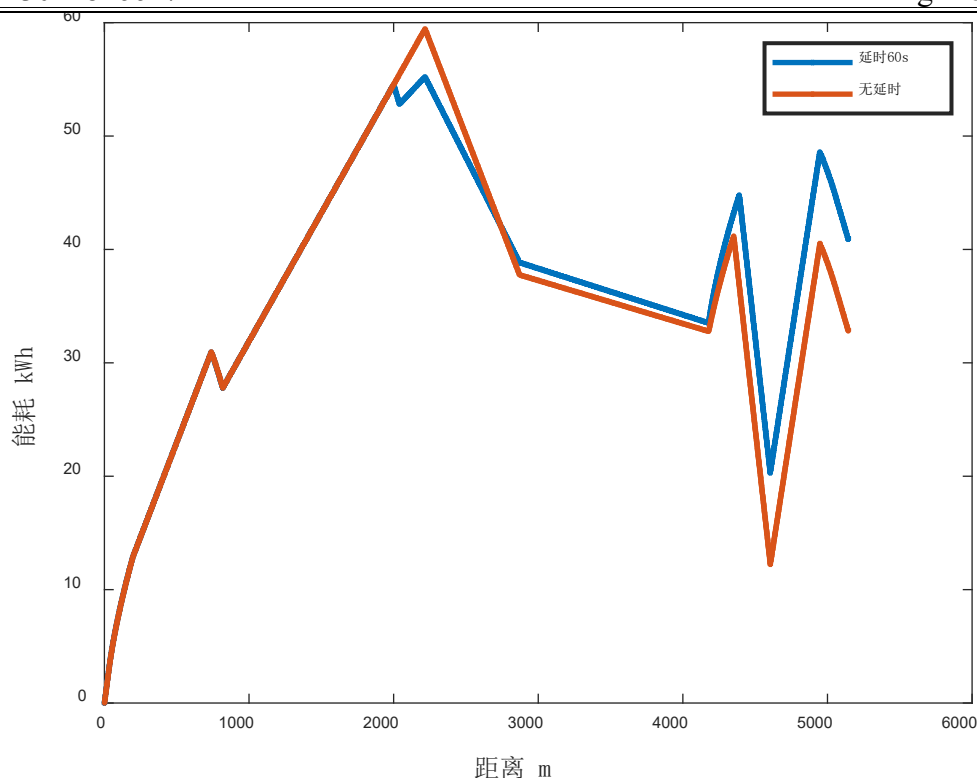


图 5-19 能量消耗-距离曲线

图 5-19 是列车的能量消耗-距离仿真图。

六、模型的评价及优化

6.1 误差分析

6.1.1 针对于问题 1 的误差分析

求解问题一时，因题目未明确说明路况，我们忽略了附加阻力带来的影响，问题二中重新考虑了附加阻力，因此这里不算是误差的影响因素。我们利用数值积分建立模型，由表 6-1 可知，列车到达终点的理论延时和实际用时的误差并不明显。

表 6-1 到达终点的理论延时、实际用时的关系

理论延时/s	实际用时/s
0	196.678
10	206.204
20	215.241
50	246.069

150	344.839
300	495.013

6.1.2 针对于问题 2 的误差分析

求解问题二时，我们在问题一的基础上改进了模型，将坡度附加阻力、牵引力和制动力变化纳入了考虑。采用数值积分方法建立模型，采取多个采样点，由表 6-2 可知，列车到达终点的理论延时和实际用时的误差并不明显。

表 6-2 到达终点的理论延时、实际用时的关系

理论延时/s	实际用时/s
0	204.847
10	216.132
20	225.161
50	254.885
150	354.657
300	506.137

6.1.3 针对于问题 3 的误差分析

当列车运行时遇到紧急突发状况，需要改变运行状态时，我们给出了列车的运行方案。与列车正常运行相比，给出的方案可以达到良好的延时到达的目的，结果较为精确。

6.2 模型的优点（建模方法创新、求解特色等）

问题一的模型分析了牵引力、基本阻力以及制动力对列车运行的影响，得到了列车到达车站 B 的最短时间，以及延时到达车站 B 的方法。

问题二的模型额外考虑了坡道附加阻力以及牵引力和制动力最大值随速度变化的影响，求得列车到达车站 B 的最短时间，并且优化方案，使得运行过程的能耗降低。

问题三的模型考虑了列车中途遇到紧急情况时需要做出相应反应，改变运行方案。

6.3 模型的缺点

问题解决过程中采用了数值积分的思想，采样点的个数将会影响结果的准确性。

模型中未考虑列车舒适度等因素，不能直接应用到实际生活中。

6.4 模型的推广

该模型为城市轨道交通列车运行模型，建立的模型及算法结构较简单，实用性强，对于城市轨道交通列车运行方案有较好的参考价值。在探索利用模型算法来解决城市轨道交通列车运行问题方面，对计算出合理可行的运行方案以及节约能耗具有非常重要的现实意义。

参考文献

- [1] 张 犇.我国城市轨道交通发展趋势及金融需求分析[J].工程经济,2021,31(10):61-64.DOI:10.19298/j.cnki.1672-2442.202110061.
- [2] 刘卡丁.发展轨道交通助推低碳经济[J].轨道交通,2010: 12-12.
- [3] 中国城市轨道交通协会.城市轨道交通 2020 年度统计和分析报告[M].中国城市轨道交通协会信息, 2021, 3: 1-66.
- [4] 尚梦影.城轨列车节能运行协同智能优化与控制[D].北京交通大学,2022.DOI:10.26944/d.cnki.gbfju.2022.000360.
- [5] HOWLETT P G.An Optimal Strategy for the Control of a Train[J].Journal of the Australian Mathematical Society,1990,31(4):454-471.
- [6] PUDNEY P, HOWLETT P G. Optimal Driving Strategies for a Train Journey with Speed Limits[J]. Anziam Journal,1994,36(1):38-49.
- [7] H. Yang et al., "Coordinated demand response of rail transit load and energy storage system considering driving comfort," in CSEE Journal of Power and Energy Systems, vol. 6, no. 4, pp. 749-759, Dec. 2020, doi: 10.17775/CSEEJPES.2020.02590.

附录

```

function [force] = calculateForce(v)
%UNTITLED7 此处显示有关此函数的摘要
% 此处显示详细说明
force = (2.0895 + 0.0098.*v + 0.006 * v.^2) * 1e3;
end

function [force] = calculateForce1(v)
%UNTITLED7 此处显示有关此函数的摘要
% 此处显示详细说明
force = (2.0895 + 0.0098.*v + 0.0065 * v.^2) * 1e3;
end

% 千分制坡度转换为阻力
function [force] = gradient2force(gradient, m)
gradient = gradient * 1e-3;
theta = atan(gradient);
force = m * 9.8 * sin(theta);
end

function loadData = importfile(workbookFile, sheetName, dataLines)
%IMPORTFILE 导入电子表格中的数据
% UNTITLED = IMPORTFILE(FILE) 读取名为 FILE 的 Microsoft Excel
% 电子表格文件的第一张工作表中的数据。 以表形式返回数据。
%
% UNTITLED = IMPORTFILE(FILE, SHEET) 从指定的工作表中读取。
%
% UNTITLED = IMPORTFILE(FILE, SHEET,
% DATALINES)按指定的行间隔读取指定工作表中的数据。对于不连续的行间隔, 请将 DATALINES 指
% 定为正整数标量或 N×2
% 正整数标量数组。
%
% 示例:
% Untitled = importfile("D:\UESTC\Race\2023 数维杯\代码\附件一.xls", "Sheet1", [2,
21]);
%
% 另请参阅 READTABLE。
%
% 由 MATLAB 于 2023-05-13 22:51:20 自动生成

%% 输入处理

% 如果未指定工作表, 则将读取第一张工作表
if nargin == 1 || isempty(sheetName)
    sheetName = 1;
end

% 如果未指定行的起点和终点, 则会定义默认值。
if nargin <= 2
    dataLines = [2, 21];
end

```



```

%% 设置导入选项并导入数据
opts = spreadsheetImportOptions("NumVariables", 4);

% 指定工作表和范围
opts.Sheet = sheetName;
opts.DataRange = "A" + dataLines(1, 1) + ":D" + dataLines(1, 2);

% 指定列名称和类型
opts.VariableNames = ["Distancem", "Gradient", "Var3", "Vlimitkmh"];
opts.SelectedVariableNames = ["Distancem", "Gradient", "Vlimitkmh"];
opts.VariableTypes = ["double", "double", "char", "double"];

% 指定变量属性
opts = setvaropts(opts, "Var3", "WhitespaceRule", "preserve");
opts = setvaropts(opts, "Var3", "EmptyFieldRule", "auto");

% 导入数据
loadData = readtable(workbookFile, opts, "UseExcel", false);

for idx = 2:size(dataLines, 1)
    opts.DataRange = "A" + dataLines(idx, 1) + ":D" + dataLines(idx, 2);
    tb = readtable(workbookFile, opts, "UseExcel", false);
    loadData = [loadData; tb]; %#ok<AGROW>
end

end

```

solution 1

```

clc;
clear;
close;

%% 参数设计
d = 5144.7; % 距离
m = 176.3e3; % 列车质量, kg
e = 1.08; % 旋转质量因素
TFmax = 310e3; % 最大牵引力;
BFmax = 760e3; % 最大制动力;

%% 状态变量
duration = [0 10 20 50 150 300];
deltaD = 1;
x = 0 : deltaD : d;
len = length(x);
v = zeros(6, len);
F1 = zeros(6, len); % 牵引力
F2 = zeros(6, len); % 制动力
t = zeros(6, len);
W = zeros(6, len - 1); % 每小段一段路程的能量消耗
tempV = zeros(1, len);
tempF1 = zeros(1, len); % 牵引力
tempF2 = zeros(1, len); % 制动力
tempT = zeros(1, len);
tempW = zeros(1, len - 1);
% 初始化状态
tempF1(1) = TFmax;

```

%% 计算从最大速度处最快制动时间, 以及最快达到最大速度的时间

```
vmax = 10.0 : 0.01 : 100.0;
vmax = vmax * 1e3 / 3600;
lenV = length(vmax);
tBrake = zeros(1, lenV);
tTraction = zeros(1, lenV);
dBrake = zeros(1, lenV);
time = zeros(1, lenV);
dTraction = zeros(1, lenV);
for i = 1 : length(vmax)
    t1 = @(x, F) m ./ (F + (2.0895 + 0.0098.*x + 0.006 .* x.^2).* 1e3) .* e ;
    d1 = @(x, F) m .* x ./ (F + (2.0895 + 0.0098.*x + 0.006 .* x.^2).* 1e3) .* e ;
    tBrake(i) = integral(@(x) t1(x, BFmax), 0, vmax(i));
    dBrake(i) = integral(@(x) d1(x, BFmax), 0, vmax(i));
    t2 = @(x, F) m ./ (F - (2.0895 + 0.0098.*x + 0.006 .* x.^2).* 1e3) .* e;
    d2 = @(x, F) m .* x ./ (F - (2.0895 + 0.0098.*x + 0.006 .* x.^2) .* 1e3) .* e;
    tTraction(i) = integral(@(x) t2(x, TFmax), 0, vmax(i));
    dTraction(i) = integral(@(x) d2(x, TFmax), 0, vmax(i));
    t3 = (d - dBrake(i) - dTraction(i)) / vmax(i);
    time(i) = tBrake(i) + tTraction(i) + t3;
end
figure
plot(vmax, time);
grid on;
```

%% 计算运动状态

```
index = [9001, 8455, 7970, 6783, 4431, 2755];
velocity = [vmax(index(1)), vmax(index(2)), vmax(index(3)), vmax(index(4)),
vmax(index(5)), vmax(index(6))];
for durationIndex = 1 : length(duration)
    curDuration = duration(durationIndex); % 延迟时间
    curVmax = velocity(durationIndex);
    curTBrake = dBrake(index(durationIndex)); % 当前时延对应的制动距离
    % 由制动时间算出加速度
    tempA = curVmax / curTBrake;
    curDBrake = curVmax^2 / (2 * tempA);
    for k = 2 : len
        remainD = d - x(k); % 剩余路程
        if remainD < curTBrake
            % 此时应该开始制动, 更新各个点的状态值
            totleF = (BFmax + (calculateForce(tempV(k - 1)))) ./ e;
            tempA = totleF / m;
            tempV(k) = sqrt(tempV(k-1)^2 - 2 * tempA * deltaD);
            tempF1(k) = 0;
            tempF2(k) = BFmax;
            tempT(k) = tempT(k - 1) + (tempV(k-1) - tempV(k)) / tempA;
            calW = @(t) (tempV(k-1) - tempA .* t).*(m * tempA - calculateForce(tempV(k-1)
- tempA .* t));
            tempW(k - 1) = -integral(calW, 0, tempT(k) - tempT(k - 1));
        elseif tempV(k-1) < curVmax
            % 此时处于加速阶段
            totleF = (TFmax - (calculateForce(tempV(k-1)))) ./ e;
            tempA = totleF / m;
            tempV(k) = sqrt(tempV(k-1)^2 + 2 * tempA * deltaD);
            tempF1(k) = TFmax;
            tempF2(k) = 0;
            tempT(k) = tempT(k - 1) + (tempV(k) - tempV(k-1)) / tempA;
```

```

        calW = @(t) (tempV(k-1) + tempA .* t).*(m * tempA * e +
calculateForce(tempV(k-1) + tempA .* t));
        tempW(k - 1) = integral(calW, 0, tempT(k) - tempT(k - 1));
    else
        % 此时处于匀速运动阶段
        tempV(k) = curVmax;
        tempF1(k) = calculateForce(tempV(k));
        tempF2(k) = 0;
        tempT(k) = tempT(k-1) + deltaD / tempV(k);
        tempW(k - 1) = tempF1(k) * tempV(k) * deltaD;
    end
end
v(durationIndex,:) = tempV;
F1(durationIndex,:) = tempF1; % 牵引力
F2(durationIndex,:) = tempF2; % 制动力
t(durationIndex,:) = tempT;
W(durationIndex,:) = tempW; % 每小段一段路程的能量消耗
end

%% 画图
lineWidth = 3;
MarkerSize = 1;
close;
figure(1)
for durationIndex = 1 : 6
h(durationIndex) = plot(x, v(durationIndex,:), '-', 'Linewidth', lineWidth,
'MarkerSize', MarkerSize); hold on;
set(h(durationIndex), 'MarkerFaceColor', get(h(durationIndex), 'color'));
end
grid on;
xlabel('距离 m', 'FontName', '宋体', 'FontSize', 16);
ylabel('速度 m/s', 'FontName', '宋体', 'FontSize', 16);
legend({'\tau=0s', '\tau=10s', '\tau=20s', '\tau=50s', '\tau=150s', '\tau=300s'}, 'Font
Name', '宋体', 'FontSize', 9, 'Linewidth', 1.5);

figure(2)
for durationIndex = 1 : 6
h(durationIndex) = plot(x, F1(durationIndex,:), '-', 'Linewidth', lineWidth,
'MarkerSize', MarkerSize); hold on;
set(h(durationIndex), 'MarkerFaceColor', get(h(durationIndex), 'color'));
end
grid on;
xlabel('距离 m', 'FontName', '宋体', 'FontSize', 16);
ylabel('牵引力 N', 'FontName', '宋体', 'FontSize', 16);
legend({'\tau=0s', '\tau=10s', '\tau=20s', '\tau=50s', '\tau=150s', '\tau=300s'}, 'Font
Name', '宋体', 'FontSize', 9, 'Linewidth', 1.5);

figure(3)
for durationIndex = 1 : 6
h(durationIndex) = plot(x, F2(durationIndex,:), '-', 'Linewidth', lineWidth,
'MarkerSize', MarkerSize); hold on;
set(h(durationIndex), 'MarkerFaceColor', get(h(durationIndex), 'color'));
end
grid on;
xlabel('距离 m', 'FontName', '宋体', 'FontSize', 16);
ylabel('制动力 N', 'FontName', '宋体', 'FontSize', 16);

```

```

legend({ '\tau=0s', '\tau=10s', '\tau=20s', '\tau=50s', '\tau=150s', '\tau=300s'}, 'Font
Name', '宋体', 'FontSize', 9, 'LineWidth', 1.5);

figure(4)
for durationIndex = 1 : 6
h(durationIndex) = plot(x, t(durationIndex,:), '-', 'LineWidth', lineWidth,
'MarkerSize', MarkerSize); hold on;
set(h(durationIndex), 'MarkerFaceColor', get(h(durationIndex), 'color'));
end
grid on;
xlabel('距离 m', 'FontName', '宋体', 'FontSize', 16);
ylabel('时间 t', 'FontName', '宋体', 'FontSize', 16);
legend({ '\tau=0s', '\tau=10s', '\tau=20s', '\tau=50s', '\tau=150s', '\tau=300s'}, 'Font
Name', '宋体', 'FontSize', 9, 'LineWidth', 1.5);

figure(5)
for durationIndex = 1 : 6
h(durationIndex) = plot(x, [0 , cumsum(W(durationIndex,:)) / (3.6 * 1e6)], '-',
'LineWidth', lineWidth, 'MarkerSize', MarkerSize); hold on;
set(h(durationIndex), 'MarkerFaceColor', get(h(durationIndex), 'color'));
end
grid on;
xlabel('距离 m', 'FontName', '宋体', 'FontSize', 16);
ylabel('总能耗 kWh', 'FontName', '宋体', 'FontSize', 16);
legend({ '\tau=0s', '\tau=10s', '\tau=20s', '\tau=50s', '\tau=150s', '\tau=300s'}, 'Font
Name', '宋体', 'FontSize', 9, 'LineWidth', 1.5);

```

solution 2

```

clc;
clear;
close;

%% 参数设计
d = 5144.7; % 距离
m = 176.3e3; % 列车质量, kg
e = 1.08; % 旋转质量因素
TFmax = 310e3; % 最大牵引力;
BFmax = 260e3; % 最大制动力;
lambda1 = 0.9; % 电机牵引效率因子
lambda2 = 0.6; % 电机制动效率因子
vTraThresh = 10; % 牵引恒转矩和恒功率区切换点
vBraThresh = 17; % 制动恒转矩和恒功率区切换点
% 引入数据
data = importfile("D:\UESTC\Race\2023 数维杯\代码\附件一.xls", "Sheet1", [2, 21]);
distance = table2array(data(:,1));
gradient = table2array(data(:,2)); % 坡度
gradient = gradient2force(gradient, m);
vLimit = table2array(data(:,3))/3.6;
vLimit1 = vLimit(1);
vLimit2 = vLimit(13);

%% 状态变量
duration = [0 10 20 50 150 300];
deltaD = 1;

```

```

x = 0 : deltaD : d;
len = length(x);
v = zeros(6, len);
F1 = zeros(6, len); % 牵引力
F2 = zeros(6, len); % 制动力
t = zeros(6, len);
W = zeros(6, len - 1); % 每小段一段路程的能量消耗
tempV = zeros(1, len);
tempF1 = zeros(1, len); % 牵引力
tempF2 = zeros(1, len); % 制动力
tempT = zeros(1, len);
tempW = zeros(1, len - 1);
% 初始化状态
tempF1(1) = TFmax;

% 初始化每个点的坡度阻力和最大速度
f = zeros(1, len);
limit = zeros(1, len);
for i = 1 :len
    curX = x(i);
    index = find(distance > curX, 1);
    f(i) = gradient(index - 1);
    limit(i) = vLimit(index - 1);
end
%% 计算牵引力拉满作为参考
% 速度阶段为 0-10 加速;
curGradient = gradient(1);
t1 = @(x, F) m ./ (F - (calculateForce1(x)) - curGradient) .* e;
d1 = @(x, F) m .* x ./ (F - (calculateForce1(x)) - curGradient) .* e;
t1Tra = integral(@(x) t1(x, TFmax), 0, vTraThresh);
d1Tra = integral(@(x) d1(x, TFmax), 0, vTraThresh);
% 到下一个坡度转化点前
tempV1 = 20.82;
t2 = @(x, F) m ./ (F * vTraThresh ./ x - (calculateForce1(x)) - curGradient) .* e;
d2 = @(x, F) m .* x ./ (F * vTraThresh ./ x - (calculateForce1(x)) - curGradient) .*
e ;
t2Tra = integral(@(x) t2(x, TFmax), vTraThresh, tempV1 );
d2Tra = integral(@(x) d2(x, TFmax), vTraThresh, tempV1 );
totalT1 = t1Tra + t2Tra;
totalD1 = d1Tra + d2Tra;
% 坡度转化直至达到 27
curGradient = gradient(3);
t3Tra = integral(@(x) t2(x, TFmax), tempV1, vLimit1);
d3Tra = integral(@(x) d2(x, TFmax), tempV1, vLimit1);
totalT2 = totalT1 + t3Tra;
totalD2 = totalD1 + d3Tra;

%% 计算减速阶段的过程, 反推
% 速度在 0-15 时
curGradient = gradient(20);
t1 = @(x, F) m ./ (F + (calculateForce1(x)) + curGradient) .* e;
d1 = @(x, F) m .* x ./ (F + (calculateForce1(x)) + curGradient) .* e;
t1Bra = integral(@(x) t1(x, BFmax), 0, vBraThresh);
d1Bra = integral(@(x) d1(x, BFmax), 0, vBraThresh);
% 到达下一个速度转化点, 使距离变成 184
tempV2 = 23.25;
t2 = @(x, F) m ./ (F * vBraThresh ./ x + (calculateForce1(x)) + curGradient) .* e;

```

```

d2 = @(x, F) m .* x ./ (F * vBraThresh ./ x + (calculateForce1(x)) + curGradient) .*
e ;
t2Bra = integral(@(x) t2(x, BFmax), vBraThresh, tempV2);
d2Bra = integral(@(x) d2(x, BFmax), vBraThresh, tempV2);
totalT3 = t1Bra + t2Bra;
totalD3 = d1Bra + d2Bra;
% 到达下一个坡度转化点, 341
t3Bra = integral(@(x) t2(x, BFmax), tempV2, vLimit2);
d3Bra = integral(@(x) d2(x, BFmax), tempV2, vLimit2);
totalT4 = totalT3 + t3Bra;
totalD4 = totalD3 + d3Bra;

%% 计算最短时间
seq = [198, 181, 166, 130, 51, 3];
tempIndex = 6;
vBoundary = 8 : 0.1 : vLimit1;
lenBoundary = length(vBoundary);
tau = zeros(1, lenBoundary);
for k = 1 : lenBoundary
    vLimit1 = vBoundary(k);
    % 最短时间的速度序列
    dataV = [0, tempV1, vLimit1, ...
            vLimit1, vLimit1, vLimit1, ...
            vLimit1, vLimit2, vLimit2, ...
            0];
    % 坡度转则点
    dataX = [0, 198.967, 739.018, ...
            2188.63, 2217.05, 2970.8, ...
            4178.29, 4604.66, 4803.63, ...
            5144.7];

    for i = 2 : len
        curX = x(i);
        index = find(dataX > curX, 1); % 确定 x 所在的区间段
        diff = tempV(i - 1) - dataV(index);
        thresh = 0.1;
        if (dataV(index) > dataV(index - 1))
            % 此时为加速阶段
            if (abs(diff) < thresh)
                % 此时加速已经完成, 需要匀速运动
                tempV(i) = tempV(i - 1);
                tempData = (calculateForce1(tempV(i)) + f(i));
                if (tempData >= 0)
                    tempF1(i) = tempData;
                    tempF2(i) = 0;
                    tempT(i) = tempT(i - 1) + deltaD / tempV(i);
                    tempW(i - 1) = tempF1(i) / lambda1 * tempV(i) * deltaD;
                else
                    tempF2(i) = -tempData;
                    tempF1(i) = 0;
                    tempT(i) = tempT(i - 1) + deltaD / tempV(i);
                    tempW(i - 1) = -tempF2(i) * tempV(i) * deltaD * lambda2;
                end
            elseif (diff > 0)
                % 此时匀减速
                if (tempV(i - 1) >= vBraThresh)
                    % 此时需要减速, 且要考虑电机特性

```

```

        totalF = (BFmax * vBraThresh ./ tempV(i - 1) +
calculateForce1(tempV(i - 1)) + f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
        tempF1(i) = 0;
        tempF2(i) = BFmax * vBraThresh ./ tempV(i - 1);
        tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
        calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));
        tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) * lambda2;
    else
        % 此时继续减速, 但不需要考虑电机特性了
        totalF = (BFmax + calculateForce1(tempV(i - 1)) + f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
        tempF1(i) = 0;
        tempF2(i) = BFmax;
        tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
        calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));
        tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) * lambda2;
    end
else
    if (tempV(i - 1) >= vTraThresh)
        % 此时需要加速运动, 在一小段区域内可以简化为匀加速运动
        totalF = (TFmax * vTraThresh ./ tempV(i - 1) -
calculateForce1(tempV(i - 1)) - f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
        tempF1(i) = TFmax * vTraThresh ./ tempV(i - 1);
        tempF2(i) = 0;
        tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
        calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
        tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) / lambda1;
    else
        % 此时不受电机影响进行加速
        totalF = (TFmax - calculateForce1(tempV(i - 1)) - f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
        tempF1(i) = TFmax;
        tempF2(i) = 0;
        tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
        calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
        tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) / lambda1;
    end
end
elseif (dataV(index) == dataV(index - 1))
    % 此时为匀速运动
    if (abs(diff) < thresh)
        % 此时匀速运动, 但是要考虑是制动还是牵引
        tempV(i) = tempV(i - 1);
        tempData = (calculateForce1(tempV(i)) + f(i));
        if (tempData >= 0)
            tempF1(i) = tempData;
            tempF2(i) = 0;
            tempT(i) = tempT(i - 1) + deltaD / tempV(i);

```

```

        tempW(i - 1) = tempF1(i) / lambda1 * tempV(i) * deltaD;
    else
        tempF2(i) = -tempData;
        tempF1(i) = 0;
        tempT(i) = tempT(i - 1) + deltaD / tempV(i);
        tempW(i - 1) = -tempF2(i) * tempV(i) * deltaD * lambda2;
    end
elseif diff < 0
    if (tempV(i - 1) >= vTraThresh)
        % 此时匀加速，并考虑电机影响
        totalF = (TFmax * vTraThresh ./ tempV(i - 1) -
calculateForce1(tempV(i - 1)) - f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
        tempF1(i) = TFmax * vTraThresh ./ tempV(i - 1);
        tempF2(i) = 0;
        tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
        calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
        tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) / lambda1;
    else
        % 此时不受电机影响进行加速
        totalF = (TFmax - calculateForce1(tempV(i - 1)) - f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
        tempF1(i) = TFmax;
        tempF2(i) = 0;
        tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
        calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
        tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) / lambda1;
    end
else
    % 此时应该匀减速
    if (tempV(i - 1) >= vBraThresh)
        % 此时需要减速，且要考虑电机特性
        totalF = (BFmax * vBraThresh ./ tempV(i - 1) +
calculateForce1(tempV(i - 1)) + f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
        tempF1(i) = 0;
        tempF2(i) = BFmax * vBraThresh ./ tempV(i - 1);
        tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
        calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));
        tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) * lambda2;
    else
        % 此时继续减速，但不需要考虑电机特性了
        totalF = (BFmax + calculateForce1(tempV(i - 1)) + f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
        tempF1(i) = 0;
        tempF2(i) = BFmax;
        tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
        calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));
        tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) * lambda2;
    end
end

```



```

end
else
    % 此时为减速阶段
    leftX = d - curX;
    if (leftX <= totalD4)
        if (abs(diff) < thresh)
            % 此时减速完成, 开始匀速运动
            tempV(i) = tempV(i-1);
            tempData = (calculateForce1(tempV(i)) + f(i));
            if (tempData >= 0)
                tempF1(i) = tempData;
                tempF2(i) = 0;
                tempT(i) = tempT(i - 1) + deltaD / tempV(i);
                tempW(i - 1) = tempF1(i) / lambda1 * tempV(i) * deltaD;
            else
                tempF2(i) = -tempData;
                tempF1(i) = 0;
                tempT(i) = tempT(i - 1) + deltaD / tempV(i);
                tempW(i - 1) = -tempF2(i) * tempV(i) * deltaD * lambda2;
            end
        elseif (diff > 0)
            if tempV(i - 1) > vBraThresh
                % 此时需要减速, 且要考虑电机特性
                totalF = (BFmax * vBraThresh ./ tempV(i - 1) +
calculateForce1(tempV(i - 1)) + f(i - 1));
                tempA = totalF / (m * e);
                tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
                tempF1(i) = 0;
                tempF2(i) = BFmax * vBraThresh ./ tempV(i - 1);
                tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
                calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));
                tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) *
lambda2;
            else
                % 此时继续减速, 但不需要考虑电机特性了
                totalF = (BFmax + calculateForce1(tempV(i - 1)) + f(i - 1));
                tempA = totalF / (m * e);
                tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
                tempF1(i) = 0;
                tempF2(i) = BFmax;
                tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
                calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));
                tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) *
lambda2;
            end
        else
            % 此时匀加速
            if (diff < 0)
                % 此时匀加速, 并考虑电机影响
                totalF = (TFmax * vTraThresh ./ tempV(i - 1) -
calculateForce1(tempV(i - 1)) - f(i - 1));
                tempA = totalF / (m * e);
                tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
                tempF1(i) = TFmax * vTraThresh ./ tempV(i - 1);
                tempF2(i) = 0;
                tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
            end
        end
    end
end

```

```

        calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
        tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) /
lambda1;
    else
        % 此时不受电机影响进行加速
        totalF = (TFmax - calculateForce1(tempV(i - 1)) - f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
        tempF1(i) = TFmax;
        tempF2(i) = 0;
        tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
        calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
        tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) /
lambda1;
    end
end
else
    % 保持上一个状态
    tempV(i) = tempV(i - 1);
    tempF1(i) = tempF1(i - 1);
    tempF2(i) = tempF2(i - 1);
    tempT(i) = tempT(i - 1) + deltaD / tempV(i);
    tempW(i - 1) = tempW(i - 2);
end
end
end
tempV(end) = 0;
tau(k) = tempT(end);
if ismember(k, seq)
    v(tempIndex,:) = tempV;
    F1(tempIndex,:) = tempF1; % 牵引力
    F2(tempIndex,:) = tempF2; % 制动力
    t(tempIndex,:) = tempT;
    W(tempIndex,:) = tempW;
    tempIndex = tempIndex - 1;
end
end
end

%% 画图
lineWidth = 3;
MarkerSize = 1;
close;

figure(7)
h = plot(vBoundary, tau, '-*', 'Linewidth', lineWidth, 'MarkerSize', MarkerSize);
set(h, 'MarkerFaceColor', get(h, 'color'));
grid on;
xlabel('最大速度 m/s', 'FontName', '宋体', 'FontSize', 16);
ylabel('花费时间 s', 'FontName', '宋体', 'FontSize', 16);

figure(1)
for durationIndex = 1 : 6
    h(durationIndex) = plot(x, v(durationIndex,:), '-', 'Linewidth', lineWidth,
'MarkerSize', MarkerSize); hold on;
set(h(durationIndex), 'MarkerFaceColor', get(h(durationIndex), 'color'));

```

```

end
grid on;
xlabel('距离 m','FontName','宋体','FontSize',16);
ylabel('速度 m/s','FontName','宋体','FontSize',16);
legend({'\tau=0s','\tau=10s','\tau=20s','\tau=50s','\tau=150s','\tau=300s'},'FontName','宋体','FontSize',9,'LineWidth',1.5);

figure(2)
for durationIndex = 1 : 6
h(durationIndex) = plot(x, F1(durationIndex,:), '-', 'LineWidth', lineWidth, 'MarkerSize', MarkerSize); hold on;
set(h(durationIndex), 'MarkerFaceColor', get(h(durationIndex), 'color'));
end
grid on;
xlabel('距离 m','FontName','宋体','FontSize',16);
ylabel('牵引力 N','FontName','宋体','FontSize',16);
legend({'\tau=0s','\tau=10s','\tau=20s','\tau=50s','\tau=150s','\tau=300s'},'FontName','宋体','FontSize',9,'LineWidth',1.5);

figure(3)
for durationIndex = 1 : 6
h(durationIndex) = plot(x, F2(durationIndex,:), '-', 'LineWidth', lineWidth, 'MarkerSize', MarkerSize); hold on;
set(h(durationIndex), 'MarkerFaceColor', get(h(durationIndex), 'color'));
end
grid on;
xlabel('距离 m','FontName','宋体','FontSize',16);
ylabel('制动力 N','FontName','宋体','FontSize',16);
legend({'\tau=0s','\tau=10s','\tau=20s','\tau=50s','\tau=150s','\tau=300s'},'FontName','宋体','FontSize',9,'LineWidth',1.5);

figure(4)
for durationIndex = 1 : 6
h(durationIndex) = plot(x, t(durationIndex,:), '-', 'LineWidth', lineWidth, 'MarkerSize', MarkerSize); hold on;
set(h(durationIndex), 'MarkerFaceColor', get(h(durationIndex), 'color'));
end
grid on;
xlabel('距离 m','FontName','宋体','FontSize',16);
ylabel('时间 t','FontName','宋体','FontSize',16);
legend({'\tau=0s','\tau=10s','\tau=20s','\tau=50s','\tau=150s','\tau=300s'},'FontName','宋体','FontSize',9,'LineWidth',1.5);

figure(5)
for durationIndex = 1 : 6
h(durationIndex) = plot(x, [0, cumsum(W(durationIndex,:)) / (3.6 * 1e6)], '-', 'LineWidth', lineWidth, 'MarkerSize', MarkerSize); hold on;
set(h(durationIndex), 'MarkerFaceColor', get(h(durationIndex), 'color'));
end
grid on;
xlabel('距离 m','FontName','宋体','FontSize',16);
ylabel('总能耗 kWh','FontName','宋体','FontSize',16);
legend({'\tau=0s','\tau=10s','\tau=20s','\tau=50s','\tau=150s','\tau=300s'},'FontName','宋体','FontSize',9,'LineWidth',1.5);

```

solution 3

```

clc;
clear;
close;

%% 参数设计
d = 5144.7; % 距离
m = 176.3e3; % 列车质量, kg
e = 1.08; % 旋转质量因素
TFmax = 310e3; % 最大牵引力;
BFmax = 260e3; % 最大制动力;
lambda1 = 0.9; % 电机牵引效率因子
lambda2 = 0.6; % 电机制动效率因子
vTraThresh = 10; % 牵引恒转矩和恒功率区切换点
vBraThresh = 17; % 制动恒转矩和恒功率区切换点
% 引入数据
data = importfile("D:\UESTC\Race\2023 数维杯\代码\附件一.xls", "Sheet1", [2, 21]);
distance = table2array(data(:,1));
gradient = table2array(data(:,2)); % 坡度
gradient = gradient2force(gradient, m);
vLimit = table2array(data(:,3))/3.6;
vLimit1 = vLimit(1);
vLimit2 = vLimit(13);

%% 状态变量
duration = [0 10 20 50 150 300];
deltaD = 1;
x = 0 : deltaD : d;
len = length(x);
v = zeros(2, len);
F1 = zeros(2, len); % 牵引力
F2 = zeros(2, len); % 制动力
t = zeros(2, len);
W = zeros(2, len - 1); % 每小段一段路程的能量消耗
tempV = zeros(1, len);
tempF1 = zeros(1, len); % 牵引力
tempF2 = zeros(1, len); % 制动力
tempT = zeros(1, len);
tempW = zeros(1, len - 1);
% 初始化状态
tempF1(1) = TFmax;

% 初始化每个点的坡度阻力和最大速度
f = zeros(1, len);
limit = zeros(1, len);
for i = 1 :len
    curX = x(i);
    index = find(distance > curX, 1);
    f(i) = gradient(index - 1);
    limit(i) = vLimit(index - 1);
end
%% 计算牵引力拉满作为参考
% 速度阶段为 0-10 加速;
curGradient = gradient(1);
t1 = @(x, F) m ./ (F - (calculateForce1(x)) - curGradient) .* e;

```

```

d1 = @(x, F) m .* x ./ (F - (calculateForce1(x)) - curGradient) .* e;
t1Tra = integral(@(x) t1(x, TFmax), 0, vTraThresh);
d1Tra = integral(@(x) d1(x, TFmax), 0, vTraThresh);
% 到下一个坡度转化点前
tempV1 = 20.82;
t2 = @(x, F) m ./ (F * vTraThresh ./ x - (calculateForce1(x)) - curGradient) .* e;
d2 = @(x, F) m .* x ./ (F * vTraThresh ./ x - (calculateForce1(x)) - curGradient) .*
e ;
t2Tra = integral(@(x) t2(x, TFmax), vTraThresh, tempV1 );
d2Tra = integral(@(x) d2(x, TFmax), vTraThresh, tempV1 );
totalT1 = t1Tra + t2Tra;
totalD1 = d1Tra + d2Tra;
% 坡度转化直至达到 27
curGradient = gradient(3);
t3Tra = integral(@(x) t2(x, TFmax), tempV1, vLimit1);
d3Tra = integral(@(x) d2(x, TFmax), tempV1, vLimit1);
totalT2 = totalT1 + t3Tra;
totalD2 = totalD1 + d3Tra;

%% 计算减速阶段的过程，反推
% 速度在 0-15 时
curGradient = gradient(20);
t1 = @(x, F) m ./ (F + (calculateForce1(x)) + curGradient) .* e;
d1 = @(x, F) m .* x ./ (F + (calculateForce1(x)) + curGradient) .* e;
t1Bra = integral(@(x) t1(x, BFmax), 0, vBraThresh);
d1Bra = integral(@(x) d1(x, BFmax), 0, vBraThresh);
% 到达下一个速度转化点，使距离变成 184
tempV2 = 23.25;
t2 = @(x, F) m ./ (F * vBraThresh ./ x + (calculateForce1(x)) + curGradient) .* e;
d2 = @(x, F) m .* x ./ (F * vBraThresh ./ x + (calculateForce1(x)) + curGradient) .*
e ;
t2Bra = integral(@(x) t2(x, BFmax), vBraThresh, tempV2);
d2Bra = integral(@(x) d2(x, BFmax), vBraThresh, tempV2);
totalT3 = t1Bra + t2Bra;
totalD3 = d1Bra + d2Bra;
% 到达下一个坡度转化点，341
t3Bra = integral(@(x) t2(x, BFmax), tempV2, vLimit2);
d3Bra = integral(@(x) d2(x, BFmax), tempV2, vLimit2);
totalT4 = totalT3 + t3Bra;
totalD4 = totalD3 + d3Bra;

%% 计算最短时间
seq = 28;
tempIndex = 1;
time1 = 125.67; % 2000m 时的时间
velocity = 15; % 2000m 时的速度
vLimit = [15, 11.85];

vBoundary = 8 : 0.1 : vLimit(1);
lenBoundary = length(vBoundary);
tau = zeros(1, lenBoundary);
for k = 1 : lenBoundary
    % 最短时间的速度序列
    vL = vLimit(1);
    dataV = [0, tempV1, vL, ...
            vL, vL, vL, ...
            vL, vLimit2, vLimit2, ...
            0];

```

```

% 坡度转则点
dataX = [0, 198.967, 739.018, ...
        2188.63, 2217.05, 2970.8, ...
        4178.29, 4604.66, 4803.63, ...
        5144.7];
for i = 2 : len
    if (i > 2001)
        vL = vBoundary(k);
        dataV = [0, tempV1, vL, ...
                vL, vL, vL, ...
                vL, vLimit2, vLimit2, ...
                0];

    end
    curX = x(i);
    index = find(dataX > curX, 1); % 确定 x 所在的区间段
    diff = tempV(i - 1) - dataV(index);
    thresh = 0.1;
    if (dataV(index) > dataV(index - 1))
        % 此时为加速阶段
        if (abs(diff) < thresh)
            % 此时加速已经完成, 需要匀速运动
            tempV(i) = tempV(i - 1);
            tempData = (calculateForce1(tempV(i)) + f(i));
            if (tempData >= 0)
                tempF1(i) = tempData;
                tempF2(i) = 0;
                tempT(i) = tempT(i - 1) + deltaD / tempV(i);
                tempW(i - 1) = tempF1(i) / lambda1 * tempV(i) * deltaD;
            else
                tempF2(i) = -tempData;
                tempF1(i) = 0;
                tempT(i) = tempT(i - 1) + deltaD / tempV(i);
                tempW(i - 1) = -tempF2(i) * tempV(i) * deltaD * lambda2;
            end
        elseif (diff > 0)
            % 此时匀减速
            if (tempV(i - 1) >= vBraThresh)
                % 此时需要减速, 且要考虑电机特性
                totalF = (BFmax * vBraThresh ./ tempV(i - 1) +
                calculateForce1(tempV(i - 1)) + f(i - 1));
                tempA = totalF / (m * e);
                tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
                tempF1(i) = 0;
                tempF2(i) = BFmax * vBraThresh ./ tempV(i - 1);
                tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
                calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
                calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));
                tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) * lambda2;
            else
                % 此时继续减速, 但不需要考虑电机特性了
                totalF = (BFmax + calculateForce1(tempV(i - 1)) + f(i - 1));
                tempA = totalF / (m * e);
                tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
                tempF1(i) = 0;
                tempF2(i) = BFmax;
                tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
                calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
                calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));

```

```

        tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) * lambda2;
    end
    else
        if (tempV(i - 1) >= vTraThresh)
            % 此时需要加速运动，在一小段区域内可以简化为匀加速运动
            totalF = (TFmax * vTraThresh ./ tempV(i - 1) -
calculateForce1(tempV(i - 1)) - f(i - 1));
            tempA = totalF / (m * e);
            tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
            tempF1(i) = TFmax * vTraThresh ./ tempV(i - 1);
            tempF2(i) = 0;
            tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
            calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
            tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) / lambda1;
        else
            % 此时不受电机影响进行加速
            totalF = (TFmax - calculateForce1(tempV(i - 1)) - f(i - 1));
            tempA = totalF / (m * e);
            tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
            tempF1(i) = TFmax;
            tempF2(i) = 0;
            tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
            calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
            tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) / lambda1;
        end
    end
    elseif (dataV(index) == dataV(index - 1))
        % 此时为匀速运动
        if (abs(diff) < thresh)
            % 此时匀速运动，但是要考虑是制动还是牵引
            tempV(i) = tempV(i - 1);
            tempData = (calculateForce1(tempV(i)) + f(i));
            if (tempData >= 0)
                tempF1(i) = tempData;
                tempF2(i) = 0;
                tempT(i) = tempT(i - 1) + deltaD / tempV(i);
                tempW(i - 1) = tempF1(i) / lambda1 * tempV(i) * deltaD;
            else
                tempF2(i) = -tempData;
                tempF1(i) = 0;
                tempT(i) = tempT(i - 1) + deltaD / tempV(i);
                tempW(i - 1) = -tempF2(i) * tempV(i) * deltaD * lambda2;
            end
        end
        elseif diff < 0
            if (tempV(i - 1) >= vTraThresh)
                % 此时匀加速，并考虑电机影响
                totalF = (TFmax * vTraThresh ./ tempV(i - 1) -
calculateForce1(tempV(i - 1)) - f(i - 1));
                tempA = totalF / (m * e);
                tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
                tempF1(i) = TFmax * vTraThresh ./ tempV(i - 1);
                tempF2(i) = 0;
                tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
                calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
                tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) / lambda1;
            end
        end
    end
end

```

```

else
    % 此时不受电机影响进行加速
    totalF = (TFmax - calculateForce1(tempV(i - 1)) - f(i - 1));
    tempA = totalF / (m * e);
    tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
    tempF1(i) = TFmax;
    tempF2(i) = 0;
    tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
    calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
    tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) / lambda1;
end
else
    % 此时应该匀减速
    if (tempV(i - 1) >= vBraThresh)
        % 此时需要减速, 且要考虑电机特性
        totalF = (BFmax * vBraThresh ./ tempV(i - 1) +
calculateForce1(tempV(i - 1)) + f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
        tempF1(i) = 0;
        tempF2(i) = BFmax * vBraThresh ./ tempV(i - 1);
        tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
        calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));
        tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) * lambda2;
    else
        % 此时继续减速, 但不需要考虑电机特性了
        totalF = (BFmax + calculateForce1(tempV(i - 1)) + f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
        tempF1(i) = 0;
        tempF2(i) = BFmax;
        tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
        calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));
        tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) * lambda2;
    end
end
else
    % 此时为减速阶段
    leftX = d - curX;
    if (leftX <= totalD4)
        if (abs(diff) < thresh)
            % 此时减速完成, 开始匀速运动
            tempV(i) = tempV(i-1);
            tempData = (calculateForce1(tempV(i)) + f(i));
            if (tempData >= 0)
                tempF1(i) = tempData;
                tempF2(i) = 0;
                tempT(i) = tempT(i - 1) + deltaD / tempV(i);
                tempW(i - 1) = tempF1(i) / lambda1 * tempV(i) * deltaD;
            else
                tempF2(i) = -tempData;
                tempF1(i) = 0;
                tempT(i) = tempT(i - 1) + deltaD / tempV(i);
                tempW(i - 1) = -tempF2(i) * tempV(i) * deltaD * lambda2;
            end
        end
    end
end

```



```

elseif (diff > 0)
    if tempV(i - 1) > vBraThresh
        % 此时需要减速，且要考虑电机特性
        totalF = (BFmax * vBraThresh ./ tempV(i - 1) +
calculateForce1(tempV(i - 1)) + f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
        tempF1(i) = 0;
        tempF2(i) = BFmax * vBraThresh ./ tempV(i - 1);
        tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
        calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));
        tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) *
lambda2;
    else
        % 此时继续减速，但不需要考虑电机特性了
        totalF = (BFmax + calculateForce1(tempV(i - 1)) + f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
        tempF1(i) = 0;
        tempF2(i) = BFmax;
        tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
        calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));
        tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) *
lambda2;
    end
else
    % 此时匀加速
    if (diff < 0)
        % 此时匀加速，并考虑电机影响
        totalF = (TFmax * vTraThresh ./ tempV(i - 1) -
calculateForce1(tempV(i - 1)) - f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
        tempF1(i) = TFmax * vTraThresh ./ tempV(i - 1);
        tempF2(i) = 0;
        tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
        calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
        tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) /
lambda1;
    else
        % 此时不受电机影响进行加速
        totalF = (TFmax - calculateForce1(tempV(i - 1)) - f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
        tempF1(i) = TFmax;
        tempF2(i) = 0;
        tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
        calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
        tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) /
lambda1;
    end
end
else
    % 保持上一个状态

```

```

        tempV(i) = tempV(i - 1);
        tempF1(i) = tempF1(i - 1);
        tempF2(i) = tempF2(i - 1);
        tempT(i) = tempT(i - 1) + deltaD / tempV(i);
        tempW(i - 1) = tempW(i - 2);
    end

    end

    end
    tempV(end) = 0;
    tau(k) = tempT(end);
    if ismember(k, seq)
        v(tempIndex,:) = tempV;
        F1(tempIndex,:) = tempF1; % 牵引力
        F2(tempIndex,:) = tempF2; % 制动力
        t(tempIndex,:) = tempT;
        W(tempIndex,:) = tempW;
        tempIndex = tempIndex - 1;
    end
end

vL = vLimit(1);
% 最短时间的速度序列
dataV = [0, tempV1, vL, ...
        vL, vL, vL, ...
        vL, vLimit2, vLimit2, ...
        0];
for i = 2 : len
    curX = x(i);
    index = find(dataX > curX, 1); % 确定 x 所在的区间段
    diff = tempV(i - 1) - dataV(index);
    thresh = 0.1;
    if (dataV(index) > dataV(index - 1))
        % 此时为加速阶段
        if (abs(diff) < thresh)
            % 此时加速已经完成, 需要匀速运动
            tempV(i) = tempV(i - 1);
            tempData = (calculateForce1(tempV(i)) + f(i));
            if (tempData >= 0)
                tempF1(i) = tempData;
                tempF2(i) = 0;
                tempT(i) = tempT(i - 1) + deltaD / tempV(i);
                tempW(i - 1) = tempF1(i) / lambda1 * tempV(i) * deltaD;
            else
                tempF2(i) = -tempData;
                tempF1(i) = 0;
                tempT(i) = tempT(i - 1) + deltaD / tempV(i);
                tempW(i - 1) = -tempF2(i) * tempV(i) * deltaD * lambda2;
            end
        elseif (diff > 0)
            % 此时匀减速
            if (tempV(i - 1) >= vBraThresh)
                % 此时需要减速, 且要考虑电机特性
                totalF = (BFmax * vBraThresh ./ tempV(i - 1) + calculateForce1(tempV(i - 1)) + f(i - 1));
                tempA = totalF / (m * e);
                tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
                tempF1(i) = 0;
            end
        end
    end
end

```

```

tempF2(i) = BFmax * vBraThresh ./ tempV(i - 1);
tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));
tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) * lambda2;
else
    % 此时继续减速, 但不需要考虑电机特性了
    totalF = (BFmax + calculateForce1(tempV(i - 1)) + f(i - 1));
    tempA = totalF / (m * e);
    tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
    tempF1(i) = 0;
    tempF2(i) = BFmax;
    tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
    calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));
    tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) * lambda2;
end
elseif (tempV(i - 1) >= vTraThresh)
    % 此时需要加速运动, 在一小段区域内可以简化为匀加速运动
    totalF = (TFmax * vTraThresh ./ tempV(i - 1) - calculateForce1(tempV(i
- 1)) - f(i - 1));
    tempA = totalF / (m * e);
    tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
    tempF1(i) = TFmax * vTraThresh ./ tempV(i - 1);
    tempF2(i) = 0;
    tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
    calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
    tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) / lambda1;
else
    % 此时不受电机影响进行加速
    totalF = (TFmax - calculateForce1(tempV(i - 1)) - f(i - 1));
    tempA = totalF / (m * e);
    tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
    tempF1(i) = TFmax;
    tempF2(i) = 0;
    tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
    calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
    tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) / lambda1;
end
elseif (dataV(index) == dataV(index - 1))
    % 此时为匀速运动
    if (abs(diff) < thresh)
        % 此时匀速运动, 但是要考虑是制动还是牵引
        tempV(i) = tempV(i - 1);
        tempData = (calculateForce1(tempV(i)) + f(i));
        if (tempData >= 0)
            tempF1(i) = tempData;
            tempF2(i) = 0;
            tempT(i) = tempT(i - 1) + deltaD / tempV(i);
            tempW(i - 1) = tempF1(i) / lambda1 * tempV(i) * deltaD;
        else
            tempF2(i) = -tempData;
            tempF1(i) = 0;
            tempT(i) = tempT(i - 1) + deltaD / tempV(i);

```

```

        tempW(i - 1) = -tempF2(i) * tempV(i) * deltaD * lambda2;
    end
elseif diff < 0
    if (tempV(i - 1) >= vTraThresh)
        % 此时匀加速, 并考虑电机影响
        totalF = (TFmax * vTraThresh ./ tempV(i - 1) - calculateForce1(tempV(i
- 1)) - f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
        tempF1(i) = TFmax * vTraThresh ./ tempV(i - 1);
        tempF2(i) = 0;
        tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
        calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
        tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) / lambda1;
    else
        % 此时不受电机影响进行加速
        totalF = (TFmax - calculateForce1(tempV(i - 1)) - f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
        tempF1(i) = TFmax;
        tempF2(i) = 0;
        tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
        calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
        tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) / lambda1;
    end
else
    % 此时应该匀减速
    if (tempV(i - 1) >= vBraThresh)
        % 此时需要减速, 且要考虑电机特性
        totalF = (BFmax * vBraThresh ./ tempV(i - 1) + calculateForce1(tempV(i
- 1)) + f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
        tempF1(i) = 0;
        tempF2(i) = BFmax * vBraThresh ./ tempV(i - 1);
        tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
        calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));
        tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) * lambda2;
    else
        % 此时继续减速, 但不需要考虑电机特性了
        totalF = (BFmax + calculateForce1(tempV(i - 1)) + f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
        tempF1(i) = 0;
        tempF2(i) = BFmax;
        tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
        calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));
        tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) * lambda2;
    end
end
else
    % 此时为减速阶段
    leftX = d - curX;
    if (leftX <= totalD4)

```

```

if (abs(diff) < thresh)
    % 此时减速完成, 开始匀速运动
    tempV(i) = tempV(i-1);
    tempData = (calculateForce1(tempV(i)) + f(i));
    if (tempData >= 0)
        tempF1(i) = tempData;
        tempF2(i) = 0;
        tempT(i) = tempT(i - 1) + deltaD / tempV(i);
        tempW(i - 1) = tempF1(i) / lambda1 * tempV(i) * deltaD;
    else
        tempF2(i) = -tempData;
        tempF1(i) = 0;
        tempT(i) = tempT(i - 1) + deltaD / tempV(i);
        tempW(i - 1) = -tempF2(i) * tempV(i) * deltaD * lambda2;
    end
elseif (diff > 0)
    if tempV(i - 1) > vBraThresh
        % 此时需要减速, 且要考虑电机特性
        totalF = (BFmax * vBraThresh ./ tempV(i - 1) +
calculateForce1(tempV(i - 1)) + f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
        tempF1(i) = 0;
        tempF2(i) = BFmax * vBraThresh ./ tempV(i - 1);
        tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
        calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));
        tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) * lambda2;
    else
        % 此时继续减速, 但不需要考虑电机特性了
        totalF = (BFmax + calculateForce1(tempV(i - 1)) + f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 - 2 * tempA * deltaD);
        tempF1(i) = 0;
        tempF2(i) = BFmax;
        tempT(i) = tempT(i - 1) + (tempV(i - 1) - tempV(i)) / tempA;
        calW = @(t) (tempV(i-1) - tempA .* t).*(m * tempA * e -
calculateForce1(tempV(i-1) - tempA .* t) - f(i - 1));
        tempW(i - 1) = -integral(calW, 0, tempT(i) - tempT(i - 1)) * lambda2;
    end
else
    % 此时匀加速
    if (diff < 0)
        % 此时匀加速, 并考虑电机影响
        totalF = (TFmax * vTraThresh ./ tempV(i - 1) -
calculateForce1(tempV(i - 1)) - f(i - 1));
        tempA = totalF / (m * e);
        tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
        tempF1(i) = TFmax * vTraThresh ./ tempV(i - 1);
        tempF2(i) = 0;
        tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
        calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
        tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) / lambda1;
    else
        % 此时不受电机影响进行加速
        totalF = (TFmax - calculateForce1(tempV(i - 1)) - f(i - 1));
        tempA = totalF / (m * e);

```

```

        tempV(i) = sqrt(tempV(i - 1).^2 + 2 * tempA * deltaD);
        tempF1(i) = TFmax;
        tempF2(i) = 0;
        tempT(i) = tempT(i - 1) + (tempV(i) - tempV(i - 1)) / tempA;
        calW = @(t) (tempV(i-1) + tempA .* t).*(m * tempA * e +
calculateForce1(tempV(i-1) + tempA .* t) + f(i - 1));
        tempW(i - 1) = integral(calW, 0, tempT(i) - tempT(i - 1)) / lambda1;
    end
end
else
    % 保持上一个状态
    tempV(i) = tempV(i - 1);
    tempF1(i) = tempF1(i - 1);
    tempF2(i) = tempF2(i - 1);
    tempT(i) = tempT(i - 1) + deltaD / tempV(i);
    tempW(i - 1) = tempW(i - 2);
end

end
end
tempV(end) = 0;
v(2,:) = tempV;
F1(2,:) = tempF1;
F2(2, :) = tempF2;
t(2, :) = tempT;
W(2, :) = tempW;
%% 画图
lineWidth = 2;
MarkerSize = 1;
close;

figure(1)
h1 = plot(x, v(1, :), '-*', 'LineWidth', lineWidth, 'MarkerSize', MarkerSize);
set(h1, 'MarkerFaceColor', get(h1, 'color')); hold on;
h2 = plot(x, v(2, :), '-+', 'LineWidth', lineWidth, 'MarkerSize', MarkerSize);
set(h2, 'MarkerFaceColor', get(h2, 'color'));
grid on;
xlabel('距离 m', 'FontName', '宋体', 'FontSize', 16);
ylabel('速度 m/s', 'FontName', '宋体', 'FontSize', 16); hold on;
legend({'延时 60s', '无延时'}, 'FontName', '宋体', 'FontSize', 9, 'LineWidth', 1.5);

figure(2)
h1 = plot(x, F1(1, :), '-*', 'LineWidth', lineWidth, 'MarkerSize', MarkerSize);
set(h1, 'MarkerFaceColor', get(h1, 'color')); hold on;
h2 = plot(x, F1(2, :), '-+', 'LineWidth', lineWidth, 'MarkerSize', MarkerSize);
set(h2, 'MarkerFaceColor', get(h2, 'color'));
grid on;
xlabel('距离 m', 'FontName', '宋体', 'FontSize', 16);
ylabel('牵引力 N', 'FontName', '宋体', 'FontSize', 16); hold on;
legend({'延时 60s', '无延时'}, 'FontName', '宋体', 'FontSize', 9, 'LineWidth', 1.5);

figure(3)
h1 = plot(x, F2(1, :), '-*', 'LineWidth', lineWidth, 'MarkerSize', MarkerSize);
set(h1, 'MarkerFaceColor', get(h1, 'color')); hold on;
h2 = plot(x, F2(2, :), '-+', 'LineWidth', lineWidth, 'MarkerSize', MarkerSize);
set(h2, 'MarkerFaceColor', get(h2, 'color'));
grid on;
xlabel('距离 m', 'FontName', '宋体', 'FontSize', 16);

```

```
ylabel('制动力 N','FontName','宋体','FontSize',16); hold on;
legend({'延时 60s','无延时'},'FontName','宋体','FontSize',9,'LineWidth',1.5);

figure(4)
h1 = plot(x, t(1, :), '-*', 'LineWidth', lineWidth, 'MarkerSize', MarkerSize);
set(h1,'MarkerFaceColor',get(h1,'color'));hold on;
h2 = plot(x, t(2, :), '-+', 'LineWidth', lineWidth, 'MarkerSize', MarkerSize);
set(h2,'MarkerFaceColor',get(h2,'color'));
xlabel('距离 m','FontName','宋体','FontSize',16);
ylabel('时间 t','FontName','宋体','FontSize',16); hold on;
legend({'延时 60s','无延时'},'FontName','宋体','FontSize',9,'LineWidth',1.5);

figure(5)
h1 = plot(x, [0, cumsum(W(1,:)) / (3.6 * 1e6)], '-*', 'LineWidth', lineWidth,
'MarkerSize', MarkerSize);
set(h1,'MarkerFaceColor',get(h1,'color'));hold on;
h2 = plot(x, [0, cumsum(W(2,:)) / (3.6 * 1e6)], '-+', 'LineWidth', lineWidth,
'MarkerSize', MarkerSize);
set(h2,'MarkerFaceColor',get(h2,'color'));
xlabel('距离 m','FontName','宋体','FontSize',16);
ylabel('能耗 kWh','FontName','宋体','FontSize',16); hold on;
legend({'延时 60s','无延时'},'FontName','宋体','FontSize',9,'LineWidth',1.5);
```