

浅谈提升系统中 plc 给定速度 s 曲线的改进方法

摘要：本文指出了目前在用提升系统中的 plc 速度 s 曲线存在多处不平滑点的问题，同时提出了相应的解决方案实现拐点的平滑过渡。文章还指出了采用 plc 实现 s 曲线的逻辑关系。

关键字：提升系统，恒加速度，加加速，可变加速度，速度差，拐点速度。

引言

提升系统是井下矿山必不可少的大型运输设备，它主要负责井下与地面之间的物资、人员的运输，以及将井下矿石提运至地面的工作。其容量依据实际生产情况从几十千瓦至几千千瓦不等。通俗一点讲，可以认为它是一个大型的电梯。但与电梯的工艺又有不同之处，通常电梯的手动与自动情况下的速度均已设定好，运行过程中其速度不再受人为了的控制；而提升机的速度虽然同样也是在设计时均设定好，但当其运行在手动情况下时，其给定速度 v_{REF} 是依据控制台的速度手柄状态进行设定，也就是说此时的给定速度 v_{REF} 是一个变量。

目前我矿提升机采用以下几种速度给定方式，速度给定方式依据提升机运行模式而不同。在手动运行时，手柄负责给定速度，由提升方式选择开关设定相应的速度最大值。速度手柄的输出信号从-12v 至+12v 线性变化， v_{REF} 与该电压信号成正比关系。如表一所示：

手柄位置 v_{REF} 提升 方式	全 速	检 修	验 绳
+12v	10.6m/s	3m/s	0.3m/s
-12v	-10.6m/s	-3m/s	-0.3m/s
(0v,+12v)	(0,10.6)m/s	(0,3)m/s	(0,0.3)m/s
(-12v,0v)	(-10.6,0)m/s	(-3,0)m/s	(-0.3,0)m/s

表

一：

手动给定速度表

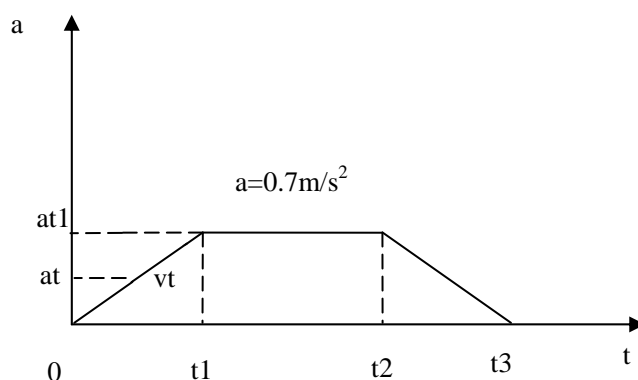
在自动模式下，手柄不参与速度的给定，由逻辑控制程序给定 $v_{REF}=10.6m/s$ 。

在 plc 逻辑程序中，s 曲线将接收到的外部给定速度信号按 $v=f(t)$ 的函数表达式从 0 递增至 v_{REF} ，在减速期间将实际给定速度按 $v=f(t)$ 的函数表达式递减至 v_{REF} 。逻辑程序中的实际给定速度实时的送入传动系统以供其用于控制。所以 s 曲线是否是平滑直接影响到系统是否运行平稳以及相关机械设备的使用寿命。目

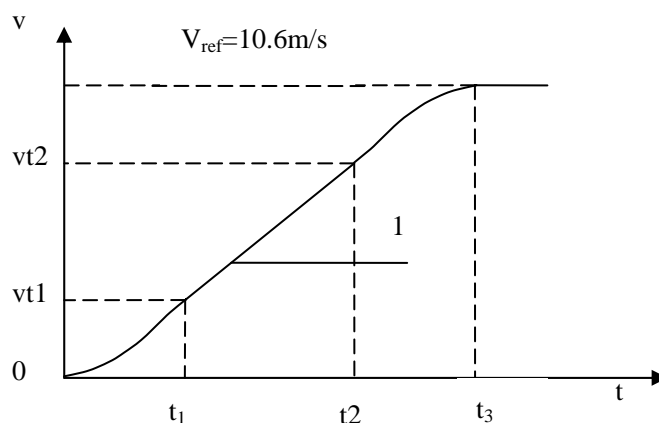
前，我矿在用提升系统具备控制系统 s 曲线和传动系统 s 曲线，传动系统 s 曲线始终处于内包络，所以 plc 的 s 曲线的缺陷在日常运行中并不会表现出来。

在用 s 曲线的分析：

目前提升系统中的 s 曲线采用恒定的加速度 $a=0.7\text{m/s}^2$ ，恒定的加加速 $\Delta a=0.35\text{m/s}^3$ ；曲线认定外部的最大给定速度始终是 10.6m/s ，也就是说无论手柄和提升方式开关给定速度是多少，该曲线自身都会升至 10.6m/s ，在曲线上升段，逻辑程序比较给定速度 v_{REF} 与 $v_{t1}=f(t_1)$ ，较小值有效，当然在下降段也是类似。正因如此所以本文仅针对上升段进行分析。具体如图所示：



图一：在用 s 曲线 a-t 图



图二：在用 s 曲线 v-t 图

由图可知各段曲线的加速度和速度的时间函数表达式：

0—— t_1 ：

$$a_t = \Delta a t \quad (1)$$

$$v_t = \int_0^t a dt = \frac{1}{2} a t^2 \quad (2)$$

t_1 —— t_2 ：

$$a_t = 0.7 \quad (3)$$

$$v_t = v_{t1} + a t \quad (4)$$

t_2 —— t_3 ：

$$a_t = 0.7 - \Delta a t \quad (5)$$

$$v_t = v_{t2} + a t - \frac{1}{2} \Delta a t^2 \quad (6)$$

计算 t_1 , t_2 , t_3 几个时刻的拐点速度及加速度：

$$a_{t1} = a_{t2} = a = 0.7$$

$$v_{t1} = \frac{1}{2} a t^2 = \frac{a^2}{2 \Delta a} = 0.7 \text{ m/s}$$

$$v_{t2} = v_{t3} - a t + \frac{1}{2} \Delta a t^2$$

$$\text{又 } t = \frac{a}{\Delta a}$$

所以

$$v_{t2} = v_{t3} - \frac{a^2}{2 \Delta a} = 9.9 \text{ m/s}$$

$$v_{t3} = 10.6 \text{ m/s}$$

由以上计算可知，如果系统处于自动运行或者在手动模式下全速运行且手柄完全推到位，那么 s 曲线可以完整的运行完毕，曲线平滑经过拐点；若系统处于手动模式下且并非全速运行时，给定速度如图二中的 1 所示。分析如下：

1、当 s 曲线与 1 相交后，逻辑程序不再使用 s 曲线的计算速度作为给定速度，而直接采用 v_{REF} 作为给定速度。此时给定速度便出现了不平滑的拐点。同时， s 曲线还是继续保持上升，也就是说它与给定速度不再相关。

2、当系统运行在速度 $v_t=v_{REF}$ 一段时间后，s 曲线自身的给定速度达到了 10.6m/s，如果此时手柄给定速度增大，逻辑系统仍然比较手柄给定速度与 s 曲线给定速度，最小值有效，那么手柄给定速度将有效，逻辑程序将该值作为最终给定速度。毫无疑问，此时给定速度将直接从 v_t 上升至当前的外部给定 v_{REF} 。显然，给定速度出现了跃变。

3、如上所说，提升系统工艺要求决定了其外部给定速度可能在 0—— t_3 期间的任一时刻。所以当处于手动情况时，该 s 曲线几乎完全不适用。虽然处于外包络状态，但此时其保护作用也不太明显。

因此，为了使 s 曲线趋于理想，现提出以下改进方法及具体实现逻辑关系。

s 曲线改进方案

一、s 曲线实时采集外部给定速度，并将该值作为曲线给定速度的最大值。由 s 曲线的逻辑程序根据运行模式、提升方式以及手柄的实际输入追踪实时的外部给定速度。这样就使得 s 曲线接收到的外部给定速度根据实际生产、操作情况在 0——10.6m/s 之间变化。利用实时的外部给定速度取代恒定的最大给定速度，可以解决在用 s 曲线给定速度与外部给定速度不相干的问题。当 s 曲线的给定速度达到了外部给定速度后，s 曲线也就达到了自身允许的最大速度，其速度不会继续上升。因此在该方案中，外部给定速度钳制了 s 曲线的实际给定速度，实时的外部给定速度是 s 曲线的最大输出速度，见图四曲线。

二、采用系统原有的加加速 $0.35m/s^2$ 。

三、采用可变加速度，s 曲线的逻辑程序根据外部给定速度的大小或者外部给定速度与曲线当前给定速度差值选择相应的加速度。只要变化的加速度不大于设计值，那么不会给系统造成任何危害。由图二可知要使得 s 曲线完全平滑，那么上升段曲线必须至少经过 0—— t_1 及 t_2 —— t_3 两段抛物线而达到给定值。

由前面的计算结果可知两段抛物线的和 $\frac{a^2}{\Delta a}$ ，所以只要 $\frac{a^2}{\Delta a}$ 小于等于 v_{REF} ，那么实际给定速度将按照 s 曲线的形状平滑的达到 v_{REF} 。

当系统的加速度为 0.7 时，

$$\frac{a^2}{\Delta a}=1.4$$

也就是说， $v_{REF}>1.4$ 时，可以采用设计加速度 0.7，曲线给定速度按 s 形状

达到外部给定速度值。

当 $v_{REF} \leq 1.4$ 时，令

$$v_{REF} = \frac{a^2}{\Delta a}$$

由上式可得 $a = \sqrt{v_{REF} \Delta a} \leq 0.7$

所以当 v_{REF} 大于 1.4 时，则采用设计加速度 $a=0.7$ ；如果 v_{REF} 小于等于 1.4 时，则采用加速度 $a = \sqrt{v_{REF} \Delta a}$ 。由变加速度的计算公式可以知道，只要采用的是变加速，

它不再具有中间的匀加速段， s 曲线的上升部分仅由两端抛物线构成。

这样选择加速度的原因主要是基于当外部给定速度满足使用自定加速度时，表明外部给定速度已经很小，如果再使曲线具备中间的匀加速段，必然造成系统的速度上升时间延长，降低生产效率。如果采用设计加速度，那么上升段包含了匀加速段。一旦在 $0 \sim t_1$ 期间确定了加速度，那么在此之后的匀加速段将不允许再次改变加速度。比如说，在 $0 \sim t_1$ 期间 s 曲线检测到外部给定速度大于 1.4m/s，此时采用的加速度为 0.7。而在之后的匀加速段，由于生产的实际需要，要求操作人员使用 1m/s 的速度开车，那么势必适当回拉手柄，外部给定速度将小于 1.4m/s，按前面提及的加速度选择逻辑，曲线就应该采用加速度 $\sqrt{v_{REF} \Delta a}$ 。

若 s 曲线改变加速度，那么 $v \sim t$ 图中的匀加速段从斜率为 0.7 的直线变为斜率为 $\sqrt{v_{REF} \Delta a}$ 的直线，两条直线之间将出现明显的转折点，给定速度出现不平滑改变。所以一旦确定了加速度，那么在之后的匀加速段便使用该加速度。

采用利用 s 曲线自定加速的功能，可以消除在用 s 曲线中的非平滑拐点出现的问题，可保证曲线给定速度平滑的从 0 上升至任何一个给定值，参见图四中的曲线 2。

四、系统运行于匀速段时，如果手柄给定速度增加，那么允许 s 曲线从匀速段进行二次加速。在此时的上升段，加速度的选择根据如下表达式确定：

令 s 曲线匀速段的速度为 v_t ，当前的给定速度为 v_{REF2} ，

$$\Delta v = v_{REF2} - v_t$$

当 $\Delta v > 1.4$ 时，则采用设计加速度 $a=0.7$ ；

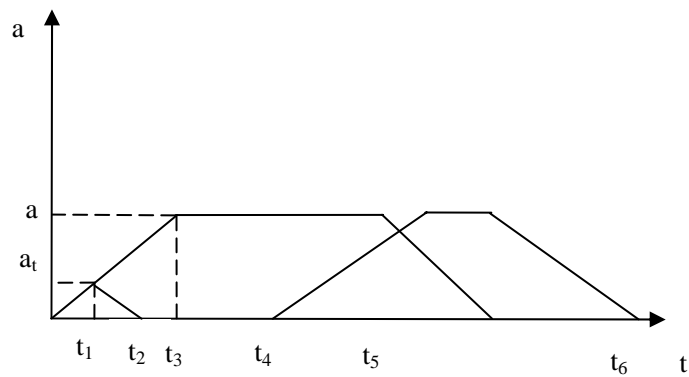
当 $\Delta v \leq 1.4$ 时，令

$$\Delta v = \frac{a^2}{\Delta a}$$

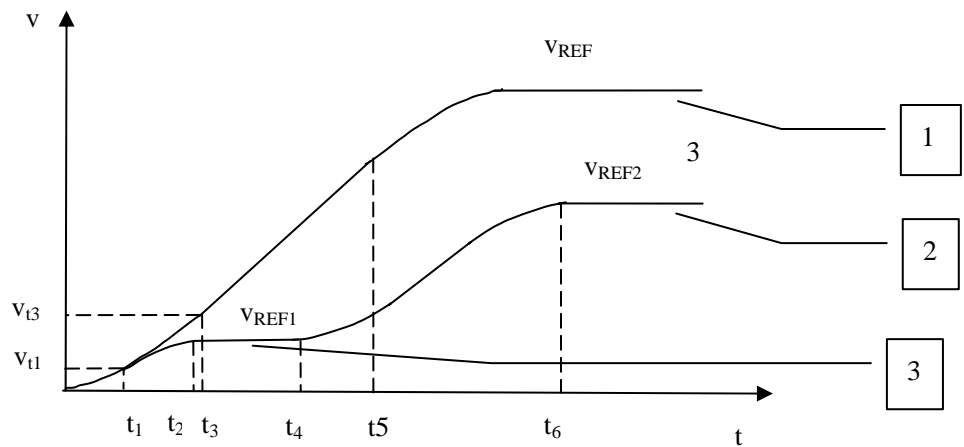
得 $a = \sqrt{\Delta v \Delta a}$

当 $\Delta v \leq 1.4$ 时，则采用加速度 $a = \sqrt{\Delta v \Delta a}$ 。

利用速度差判断 s 曲线二次加速的加速度可以消除原有 s 曲线中出现的跃变情况，不仅使曲线变得平滑，而且系统可以多次的从某一匀速段平滑过渡到另一匀速段，见图四中的曲线 3。



图三:改进后的 a—t 图



图四:改进后的 v—t 图

利用 plc 程序实现 s 曲线的方法

利用 plc 程序实现 s 曲线总的来说采用顺序控制比较方便,这里讲的顺序控制就是将 s 曲线分为第一段抛物线、匀加速段、第二段抛物线、匀速段,在上升期间曲线顺序完成以上曲线段,当其中一条曲线使能时,其余曲线段都不工作。下面介绍顺序控制的实现方法。

一、准备工作:

由于 plc 程序的扫描时间并不是固定的,所以为了确保给定速度的平稳性,曲线应该在定时中断程序里编译,现假定中断程序的时间为 0.01s 即 10ms。

当 s 曲线接收到允许运行信号后,首先进行与 s 曲线相关存储器的清零工作,

以免历史运行数据影响本次运行，但清零条件应串入上升沿脉冲有效，防止速度给定过程中重复清零寄存器，引起意外设备故障。然后进行加速度、加加速度的预置以及实时外部给定速度的读入。

二、第一段抛物线的实现：

当准备工作完成后，就应该使能第一段抛物线，同时使上升段的匀加速段、第二段抛物线等曲线段失效。在该段曲线给定速度与加速度的表达式如下：

$$a_i = \Delta at$$

$$v_i = \frac{1}{2} \Delta at^2$$

$$t = 0.01n \quad (n \text{ 为程序扫描次数})$$

计算抛物线结束点即第一个拐点的速度。由于当系统运行于手动模式时，外部给定速度很有可能变化不定，所以程序在计算出的拐点加速度也应该随着外部给定速度的变化而变化：

当 $v_{REF} > 1.4$ 时，

$$a_{i1} = a$$

$$v_{i1} = \frac{a^2}{2\Delta a} \quad (a = 0.7)$$

当 $v_{REF} \leq 1.4$ 时，

$$a_{i1} = \sqrt{v_{REF} \Delta a}$$

$$v_{i1} = \frac{v_{REF}}{2}$$

当实际给定的加速度或者速度等于拐点值时，取消第一段抛物线的使能，使能匀加速段或者第二段抛物线，如果拐点加速度为 0.7，那么使能匀加速段，如果拐点加速度小于 0.7，那么使能第二段抛物线。

三、匀加速段的实现：

从改进方案中可知，匀加速段仅用于加速度为 0.7 的状态，当加速度小于该值时，s 曲线不具备匀加速段。由于匀加速段可能是本次运行中第一次出现（如图四中曲线 1 的匀加速段），也可能第 n 次出现（如图四中曲线 2 的匀加速段），所以当匀加速段使能时，必须应用上升沿有效指令锁存 s 曲线当前的实际给定速度，以供该部分程序使用。该段曲线给定速度的计算式如下：

$$v_i = v_{i1} + at$$

$$t = 0.01n \quad (n \text{ 为程序扫描次数})$$

计算匀加速段的结束点，由于系统处于手动运行模式时，外部给定速度在匀加速段也很有可能发生变化，所以该段曲线的结束点应该基于外部给定来计算，表达式如下：

$$v_{i5} = v_{REF} - \frac{a^2}{2\Delta a}$$

若外部给定速度在匀加速段减少，使得

$$v_t + \frac{a^2}{2\Delta a} > v_{REF}$$

那么，认为实际给定速度达到了拐点值。

当实际给定的加速度或者速度等于拐点值时，取消匀速段使能，同时使能第二段抛物线。

四、第二段抛物线的实现：

当该段曲线使能时，由于该段抛物线可能直接从第一段抛物线过渡而来，也可能是经过了匀加速段而来，此时的加速度也就不确定，所以应采用上升沿有效指令存储当前的曲线给定加速度与速度。同时，应锁存当前的外部给定速度 v_{REF_t}

以备后用。根据锁存的加速度与 0.7 的关系确定下面的曲线给定加速度和速度。当拐点加速度为 0.7 时，

$$v_t = v_{t5} + at - \frac{1}{2} \Delta at^2$$

$$a = 0.7 - \Delta at$$

$$t = 0.01n \quad (n \text{ 为程序扫描次数})$$

当拐点加速度小于 0.7 时，

$$v_t = v_{t1} + a_{t1}t - \frac{1}{2} \Delta at^2$$

$$a_t = a_{t1} - \Delta at$$

$$t = 0.01n \quad (n \text{ 为程序扫描次数})$$

计算第二段抛物线结束点，由于在该段抛物线执行期间，外部给定速度是完全可能发生变化的，从以上加速度与速度的表达式可以看出，在该段曲线给定过程中是不响应外部给定速度的变化的，所以其结束点的计算就应该按以下表达式进行：

$$a_t = 0$$

或者

$$\text{若 } v_t < v_{REF} < v_{REF_t}$$

则结束点为

$$v_t = v_{REF}$$

$$\text{若 } v_t < v_{REF_t} < v_{REF}$$

则结束点为

$$v_t = v_{REF_t}$$

当给定速度或者加速度达到拐点值时，取消该段抛物线使能，使能匀速段。

五、匀速段的实现：

在该段曲线使能时，由于加速度已经为 0，所以只需采用上升沿有效指令锁存当前的曲线给定速度。由于第二段抛物线是不响应外部给定速度的增加的，所以还应锁存当前的外部给定速度，如果曲线给定速度和外部给定速度相同，那么使用当前锁存的外部给定速度作为匀速段的曲线给定速度。比较锁存的外部给定速度与实时的外部给定值，当外部给定值大于锁存值时，则取消本曲线段的使能，使能二次加速。

六、系统二次加速的实现：

二次加速的逻辑思路和上述类似，只是在第一段抛物线要锁存匀速段的给定速度与加速度，在曲线给定速度表达式中增加了匀速段的速度值，其他没有大的变化。通常在做 plc 程序时，应考虑减少系统扫描时间，尽量提高控制系统响应速度。所以最好将以上逻辑表达式做适当的修改，减少逻辑指令的应用。

结束语

本文没有对减速段进行介绍主要是 s 曲线的上升段和减速段十分类似，值得一提的是减速段的使能可能在上升段或者匀速段的任何一个时刻出现，所以当 $v_t > v_{REF}$ 时，就必须使能减速段。

通过对在用 s 曲线的改进，不仅能实时的响应外部给定值，而且使 plc 输出的给定速度更加的平滑。虽然 plc 的 s 曲线在整个系统中处于外包络状态，但较原有的 s 曲线具有更好的保护作用。