YASKAWA

Application Note: AN.MWIEC.03

How to set acceleration while maintaining move duration for S-Curve Motion

Applicable Product: MotionWorks IEC and MPiec Controller

YASKAWA

Product: MotionWorks IEC	Doc#: AN.MWIEC.03
Title: How to set acceleration while maintaining move duration for S-Curve Motion	

Application Overview

This Document details the steps for calculating the acceleration value when S-Curve is applied so that the resulting move duration is the same. To keep the move duration the same after enabling S-Curve, the acceleration must be increased. This document shows how to calculate the increased acceleration.

Application Highlights

Industry: MotionWorks IEC users
Results: Smooth Motion with S-Curve without increasing the time of the move

Product Used

Component	Product and Model Number
Software	MotionWorks IEC
Controller	Any MPiec Controller

Application Solution and Benefits:

In motion applications, S-Curve is a helpful function to smooth velocity during acceleration and deceleration.

Figure 1 (Comparison of move time with different S-Curve and accelearation) shows motion trace for three following scenarios:

Green Line: Original Velocity Trace with Old Acceleration with no added S-Curve.
Blue Line: New Velocity Trace with New Acceleration with no added S-Curve.
Red Line: New Velocity Trace with New Acceleration with added S-Curve.

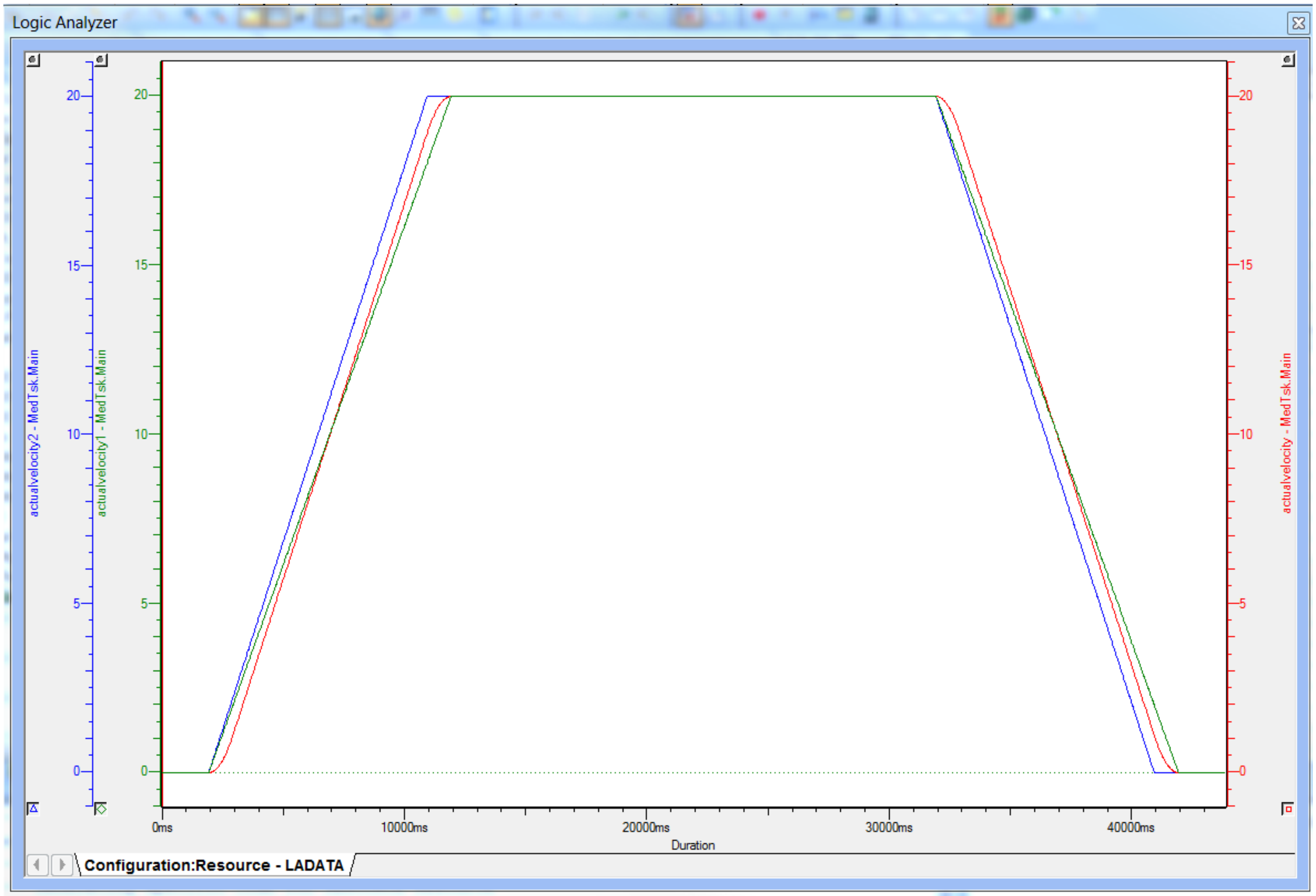

Figure 1: Comparison of move time with different S-Curve and accelearation

Here are highlights when observing Green Line and Red Line,:

- The areas under these two lines seperately are the same, which means the traveling distance is the same.
- Red Lines and Green Lines merge together at the end of the traces which means two motion are completed at the same time.

By observing Blue Line and Red Line, here are highlights:

- Blue Line is the trace without S-Curve function and Red Line is the one based on it but with S-Curve Applied.
- Red Line takes 1s longer to complete motion since Parameter 1301 Moving Average Filter 1 Time Constant is set to 1.

Implementation Method of Core Operation

1. Set Parameter 1300 & 1301

To enable S-Curve functionality, Parameter1300 Moving Average Filter 1 Enable should set to 'True'. The default value of this parameter is False.
Parameter1301 Moving Average Filter 1 Time Constant sets the time constant for the filter. The time constant determines time span over which the filter is active.

Parameter	Parameters	Current Value	Units	Min	Max	Default Value
1300	Moving Average Filter 1 Enable	True				False
1301	Moving Average Filter 1 Time Constant	1	s	0	5	0.1

2. How to calculate new acceleration with S-Curve function

To achieve the result in Figure1, the following steps are used to calculate and set parameters.
Precondition:
Required Constant Velocity = 20 m/s
Old Acceleration = Old Deceleration = 2 m/s^2
Required Travel Distance = 600m

S-Curve Setting:
Enable Pn1300
Prm1301 = 1 second

Calculate following Parameters:

- Old Acceleration Time
  Old Acc Time = Constant Velocity/Old Acceleration
  In the example, Old Acc Time = 10s

- Old Constant Velocity Time
  Old Constant Velocity Time = Total Travel Time - 2 * Old Acc Time
  In the example, Old Constant Velocity Time = 40 - 2*10 = 20s

- Total Travel Time
  Total Travel Time = Old Acc Time + Old Constant Velocity Time + New Acc Time
  = 10+20+10 = 40s

- New Acceleration Time & New Deceleration Time
  New Acceleration Time = Old Acc Time - Prm1301
  New Deceleration Time = Od Dec Time – Prm1301
  In the example, New Acceleration Time = 10 -1 = 9s
  New Deceleration Time = 10 -1 =9s

- New Constant Velocity Time
  New Constant Velocity Time = Old Constant Velocity Time + Prm1301

In the example, New Constant Velocity Time = 20+1 =21s

Bring all the available parameter into the following two equations and find out New Acceleration and New Deceleration:

New Acceleration = Constant Velocity / New Acc Time
In the example,
New Acceleration = 20/9 = 2.22 m/s^2

New Deceleration = Constant Velocity / New Dec Time
In the example,
New Acceleration = 20/9 = 2.22 m/s^2

This calculation method is the same when acceleration and deceleration are different.

Please see the following example which is demonstrating how to use structure text to compile above calculation.

```
            (*Calculate Original Trajectory Related Values*)
600.0000000 Distance := LREAL#600.0; (*m*)
 20.0000000 Velocity := LREAL#20.0; (*m/s*)
  2.0000000 OriginalAcc := LREAL#2.0;(*m/s^2*)
  2.0000000 OriginalDec := LREAL#2.0;(*m/s^2*)
 10.0000000 AccTime := Velocity/OriginalAcc;(*Second*)
 10.0000000 DecTime := Velocity/OriginalDec;(*Second*)
 20.0000000 ConstantRunTime := (Distance - LREAL#0.5*AccTime*Velocity -LREAL#0.5*DecTime*Velocity)/Velocity;
            (*Area of Trajectory is total distance, which equals Acc and Dec Distance plus constant velocity running distance.
            Thus ConstantRunTime is constant velocity running distance divides velocity*)
 40.0000000 TravelTime := AccTime + ConstantRunTime + DecTime;
            (*Original TravelTime without S-Curve*)

            (*Calucalte New Trajectory with New Acceleration and Deceleration, Set Pn1301 = 1 second*)
  9.0000000 NewAccTime := AccTime - LREAL#1.0;(*Second*)
  9.0000000 NewDecTime := DecTime - LREAL#1.0;(*Second*)
 21.0000000 NewConstantRunTime := ConstantRunTime + LREAL#1.0; (*Second*)
            (*New TravelTime without S-Curve, this is the trajectory which runs 1 second faster to be sufficent for S-curve*)
 39.0000000 NewTravelTime := NewAcctime + NewConstantRunTime + NewDecTime;(*Second*)

            (*Calculate new acceleration with finalizaed new acceleration time *)
  2.2222222 NewAcc := Velocity / NewAccTime;
  2.2222222 NewDec := Velocity/NewDecTime;
```