

SOLO Embedded Motion Profile Engine

St-Curve Position and Velocity Profile Generation

Time-based and Time-optimal

Revision	Date	Changes
V1.0.0	20/12/2022	- First Release
		-
		-
		-



Table of Contents

Introduction	4
What is the notion of Motion Profile Generation?	4
Architecture of SOLO's Embedded Motion Profile Generator:	9
Conventional PVT methods versus Embedded Profile Generation Engine of SOLO	11
Conventional PVT methods	11
Embedded Profile Generation Engine of SOLO	13
Embedded St-curve Position Profile Generation	14
Time-based versus Time-optimal Position Profile St-Curve	15
Time-Optimal Position Profile St-curve Constraints Validations	16
St-curve Position Profile Tuning	17
Embedded St-curve Velocity Profile Generation	19
Time-based versus Time-optimal Velocity Profile St-Curve	20
Time-Optimal Velocity Profile St-curve Constraints Validations	20
St-Curve Velocity Profile Tuning	21



Introduction

SOLO Motor Controllers products are now coming with an advanced Embedded Motion Profile Generation Engine that allows its users to simply create very smooth and low jerk Position or Speed trajectories using state of art “St-curve” profiles both in “Time-based” as well as “Time-optimal” manner, using such solutions the users can experience higher level of performance versus conventional methods known as PVT solutions that we will discuss more in detail later in this document.

What is the notion of Motion Profile Generation?

The answer to this question requires some basics to be met regarding the overall architecture of controlling a Motor specially for 3-phase motors like BLDC or PMSM ones while using FOC algorithms will be shaped similar to the diagram below:

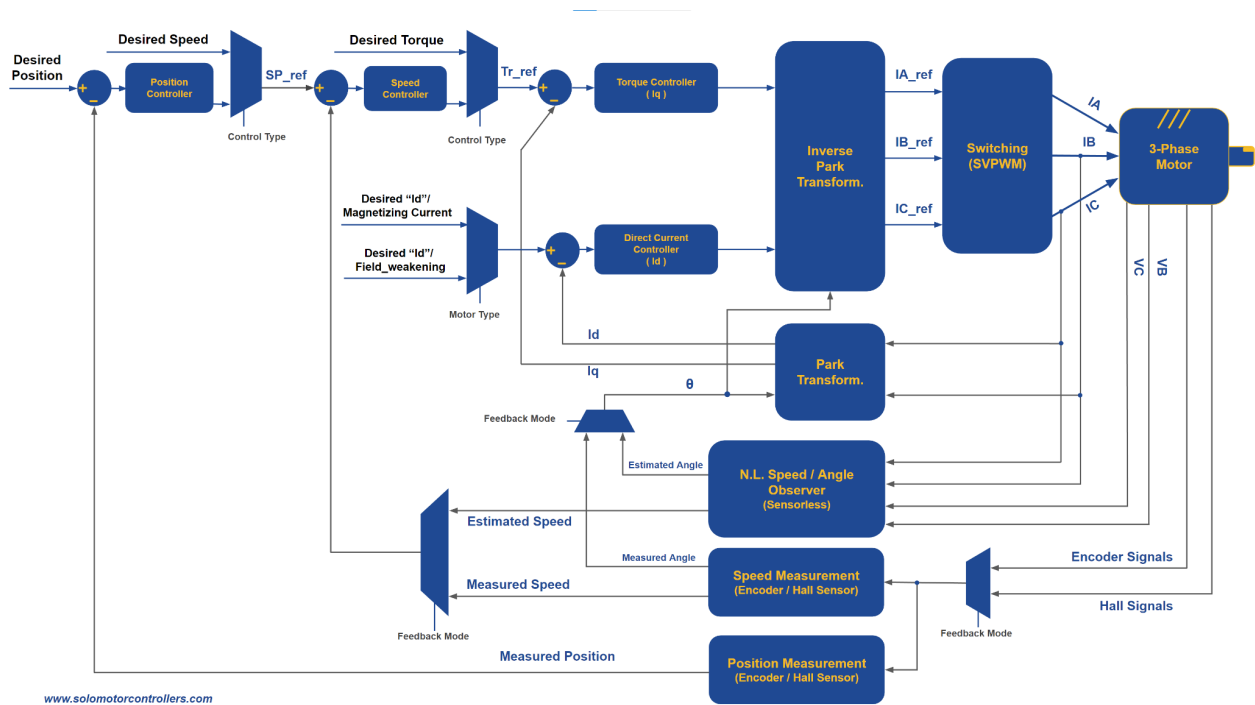


Figure 1 - Generic FOC control architecture for 3-phase Motors



Basically to control a 3-phase motor in FOC manner, there are at least 4 control loops involved and 3 of which are coming back to back in cascaded format, the Torque, Speed and Position loops.

This cascade control topology works in a way that the most exterior controller provides the reference for the interior controllers, for instance the Speed controller provides the Torque reference for the Torque controller and the Position controller provides the Speed reference for the Speed controller, thus the chain of these three controllers will allow you to control eventually the position of a 3-phase motor.

Of course if one just needs to control the speed of a 3-phase motor, they can discard the Position controller and provide the reference directly to the speed controller and the same applies when only the Torque controller is required, where both Position and Speed controllers are discarded and the torque reference is directly fed to the Torque controller.

Now in all of these configurations the notion of “reference” refers to what the user provides to the controller and it defines the final set-point for the controller as a goal to achieve, so basically the reference or the set-points are momentary changes in the controller input known as step inputs that are provided by the user, but there is no control on them as they can be any acceptable value, thus the controller has to follow the user input and this can create many challenges like oscillatory responses, controller saturation or very aggressive reaction of the controller and subsequently harsh motion of the Motor due to significant changes of the reference, to avoid all these we can behave the reference by either slowing it down or forcing it to have a nicer shape than just a simple Step input, this is where the role of Smooth Position or Speed Profile generation becomes bold.

By smoothening the reference or the set-points we will allow the controller to follow a smooth trajectory and this will affect the whole system, of course there are limitations like the dynamic responses or how fast you want the controller to achieve the set-points but in most applications the quality of the Motion has higher priority than the quickness of it, thus the smooth motion profiles are preferred compared to rough step inputs.

To understand this phenomena better, we have provided two examples of applying a direct step input to the speed controller of a PMSM versus a smooth st-curve generated velocity profile to achieve the same set-point of 1000 RPM as below:



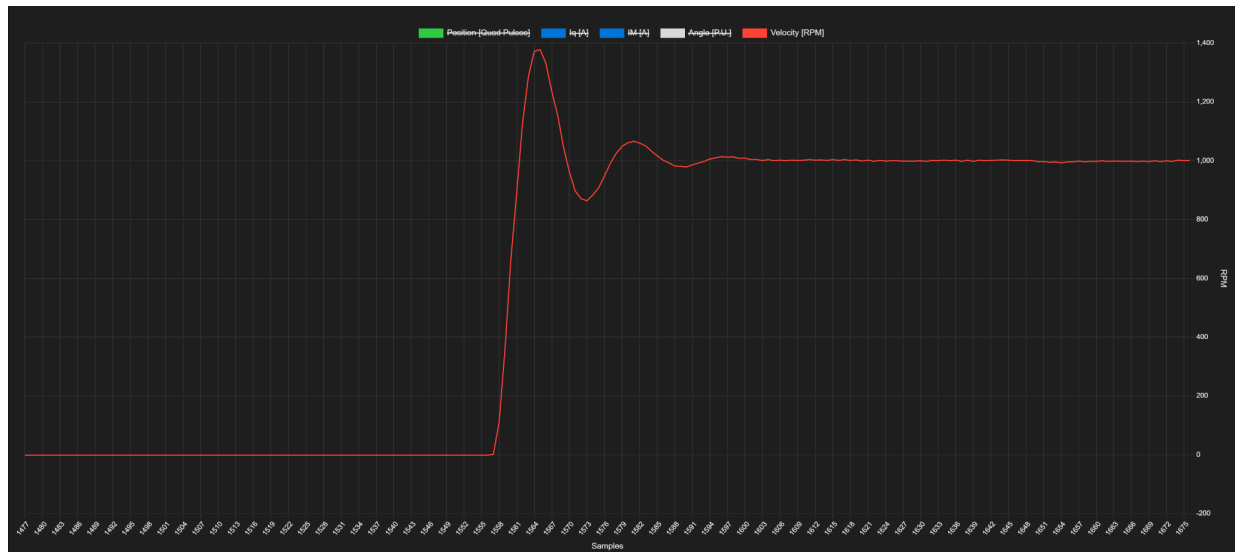


Figure 2- Speed feedback of a PMSM in reaction to a 1000RPM step set-point

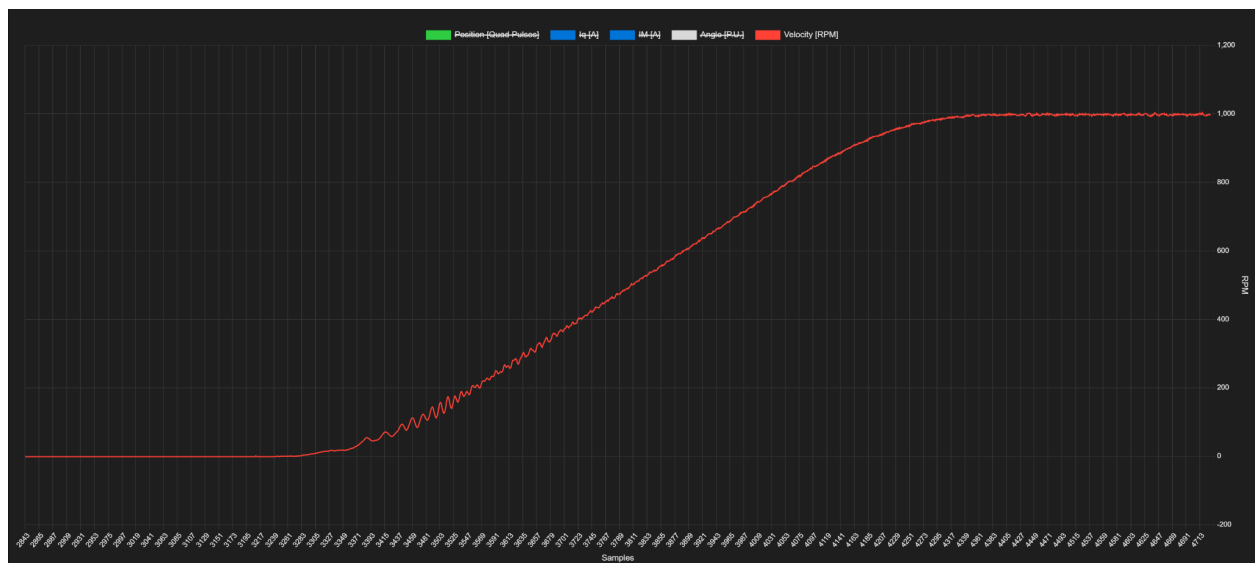


Figure 3- Speed feedback of a PMSM in reaction to a 1000RPM set-point generated by SOLO's Embedded Time-based St-curve Generator

As can be seen in Figure 2, the Motor reaches to the reference much faster compared to Figure 3 with smooth profile, but there is a considerable overshoot in the system response on the feedback measured that can come very costly to sensitive applications, thus the Motion generated in Figure 3, although being slow, can be much more valuable as it offers a extraordinary low jerk transition that is very important for many applications including robotics applications.

It is worth mentioning that in both situations the speed controller is an identical PI controller with exact same gains, and the only variation is the difference between the references to the controller.

The same effect can be observed with Position Controlling applications, for instance below you can see two different scenarios of controlling the position of a Brushless motor done using direct step input as well as the results using Time-Optimal St-curve profile generation with the exact same gains for PI controllers of Speed and Position loops.

It's evident that in case of step response the motion is jerky in the beginning due to over-shoots and oscillations caused by the speed controller, however in case of Time-optimal St-curve the motion is very smooth, providing a soft acceleration and take-off and then smoothly landing on the set-point, resulting in an extremely low jerk motion, as can be seen in Figure 5, beside the Position profile, the velocity profile is very smooth too.





Figure 4- Position and Speed feedback shapes on step response of a PMSM in reaction to a +80,000 pulses as position set-point

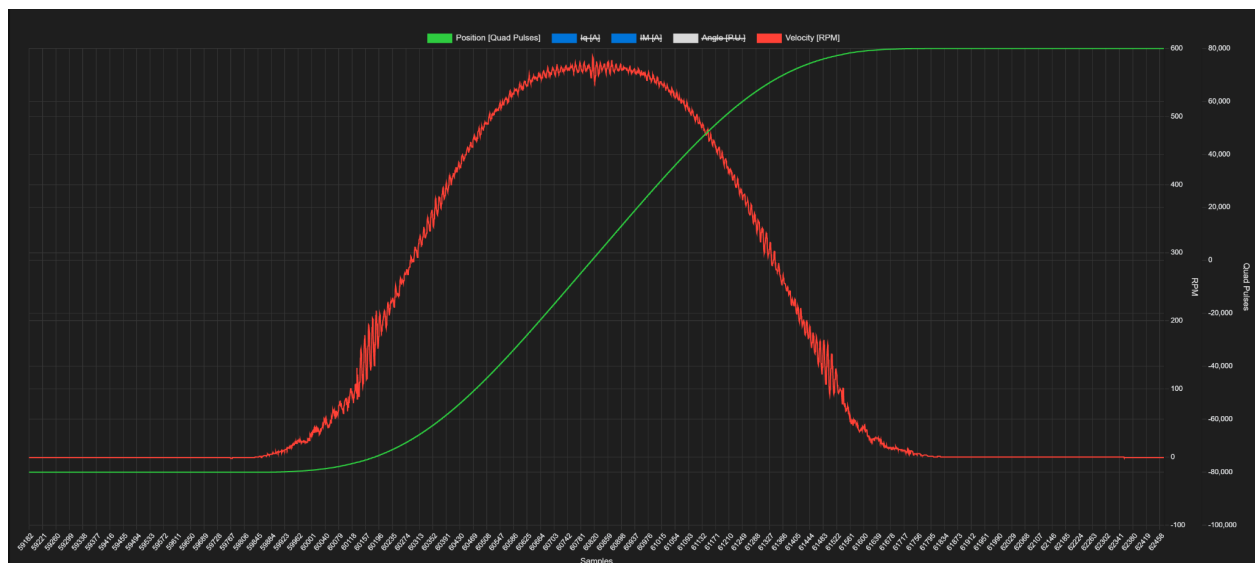


Figure5 - Position and Speed feedback of a PMSM in reaction to a +80,000 pulses set-point using SOLO's Time-optimal Embedded position profile generator

Architecture of SOLO's Embedded Motion Profile Generator:

The Motion profile generator of SOLO acts on the reference input of the controllers, and depending on what type of control mode SOLO is operating on (Speed or Position) and also the type of the Motor, the internal firmware architecture will follow one of the diagrams below:

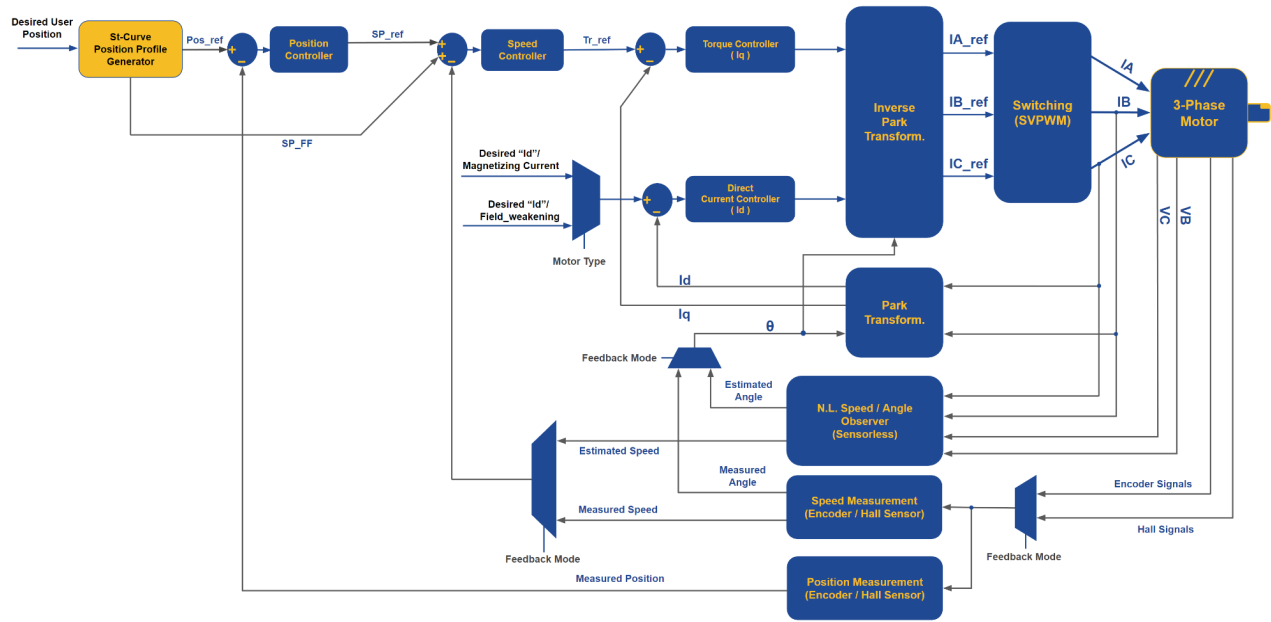


Figure 6: St-curve Position Profile Generator Architecture for 3-phase Motors

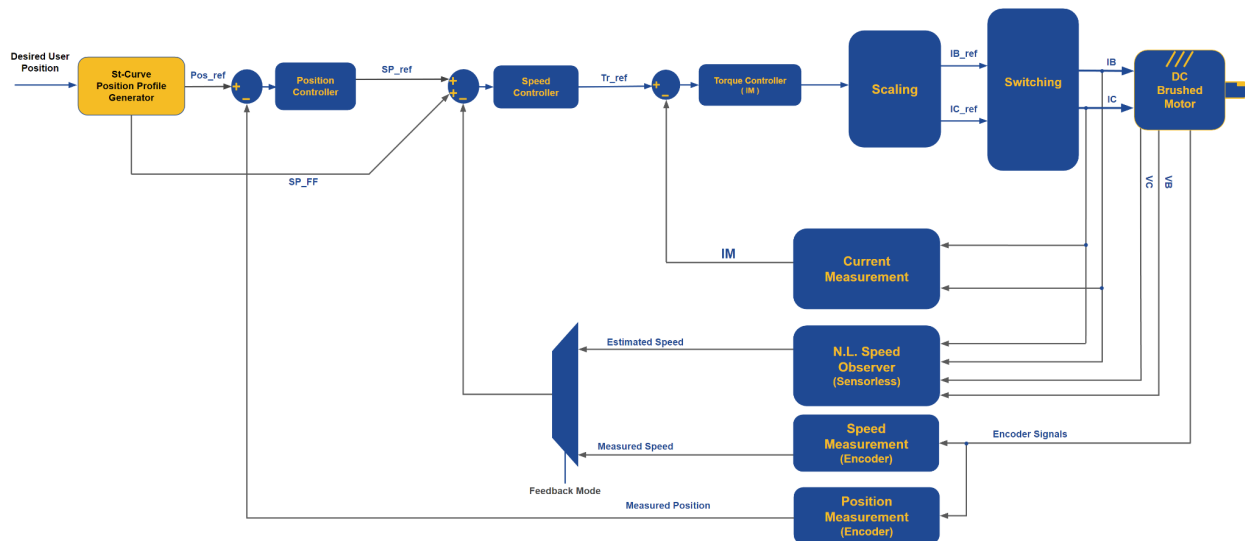


Figure 7: St-curve Position Profile Generator Architecture for DC brushed Motors



The St-curve position Profile Generator provides a Feed-Forward to the Speed reference, smoothing further the output of the Position controller to better follow the velocity profile with low jerk.

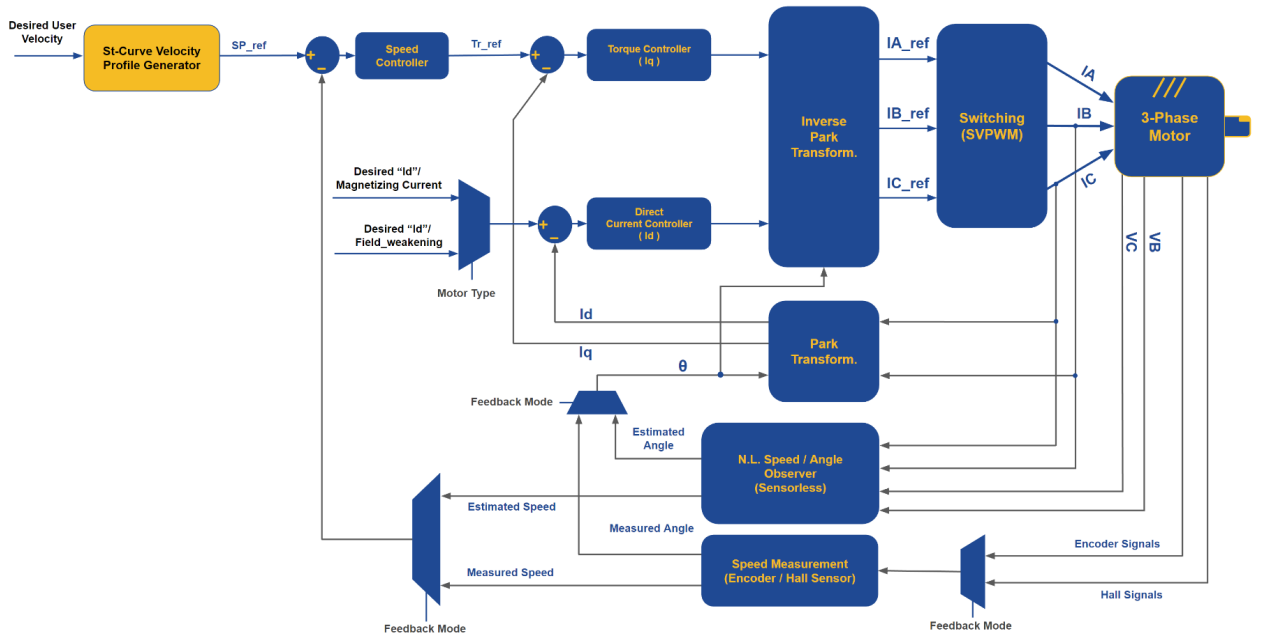


Figure 8: St-curve Velocity Profile Generator Architecture for 3-phase Motors

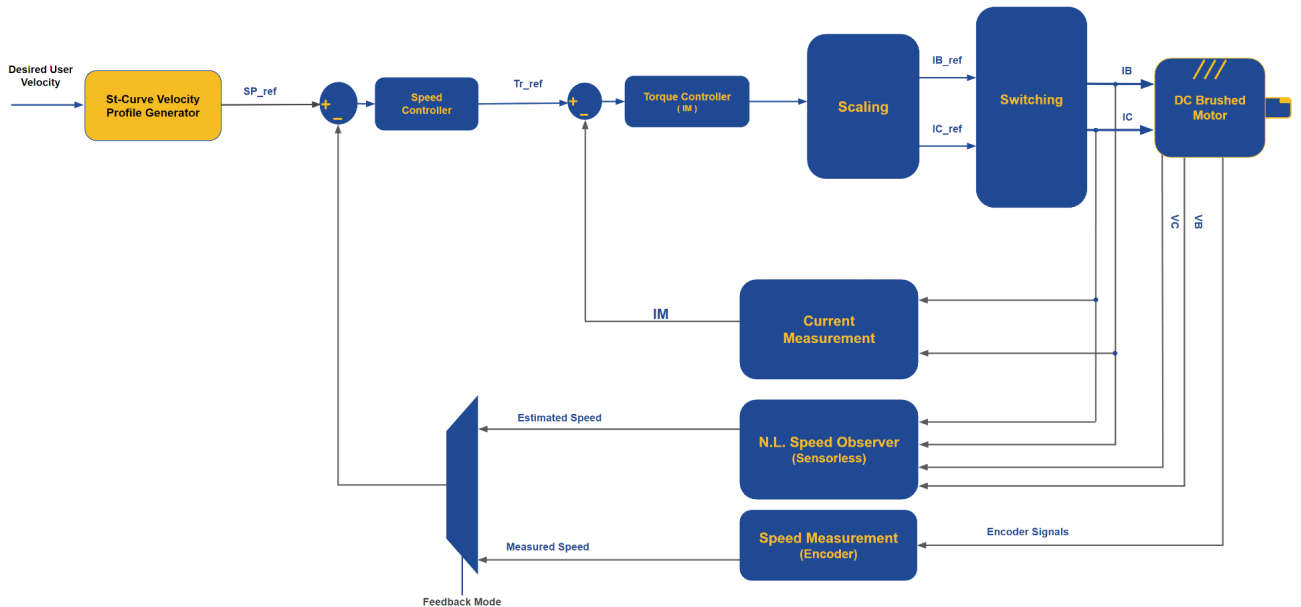


Figure 9: St-curve Velocity Profile Generator Architecture for 3-phase Motors



Conventional PVT methods versus Embedded Profile Generation Engine of SOLO

Conventional PVT methods

The conventional motion profile generation and controlling for servo-drives is known as “PVT” which stands for Position, Velocity and Time, in this method there are the following components working together as can be seen in Figure 10 below :

1. The User:

The user provides the final desired target position with some other constraints like timings or max values for inputs like velocity, acceleration, jerk and so on, and finally as a result of all of these inputs, the host can compute for the whole system the final target position, for instance the final position for the end-effector of a robot.

2. Host:

The Host, is usually a real-time system that has to compute and create the whole motion profile, this part of the system has to guarantee time determinism and it must work in a very stable fashion so that it can provide all the necessary points to the Servo Drive in a timely fashion without any interruption or considerable delay!

The Host has to locally compute the whole profile from the current position of the actuator to where it has to end up and has to fit a nice curve in between based on the algorithm that it uses (St-curve, S-curve, Trapezoidal, ...), we call this part PVT Encoding.

Once the Host computes and fits the curve, it has to send some of the computed points to the Servo drive with reduced sampling rate, as can be seen in Figure 11, the main reason for this is the fact that sending all the computed points will occupy a huge portion of the data line and it consumes a lot of resources at the Host side or in many cases if the sampling rate is too high, it's not feasible to send all these points in a timely fashion deterministically, so the Host has to reduce the samples for the Servo drive.

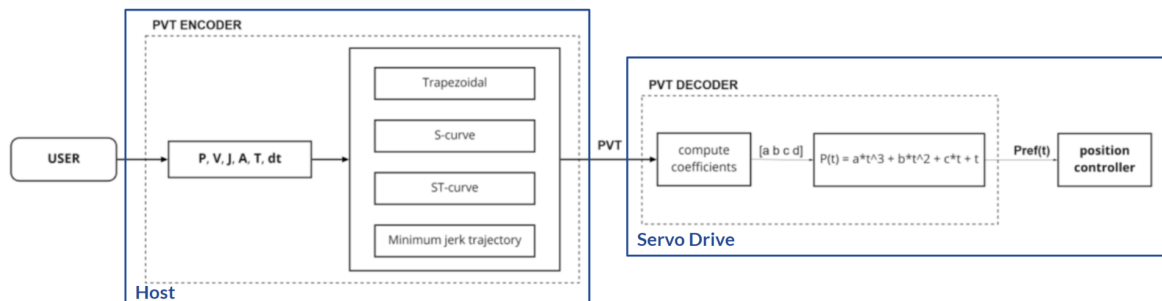


Figure 10 - Architecture of a conventional PVT method from the user to the servo drive

In Figure 11, the red dots are all the points that are sent from the Host to the Servo drive Through data lines like CANbus, EtherCAT, RS232, et..

The biggest challenge with such methods is making sure all the points from the Host to the Servo drive are sent at the right time and the Operating system on the Host can handle in real-time all the asynchronous messages between the servo drive and the host which will require intensive programming at both side with many challenges to handle.

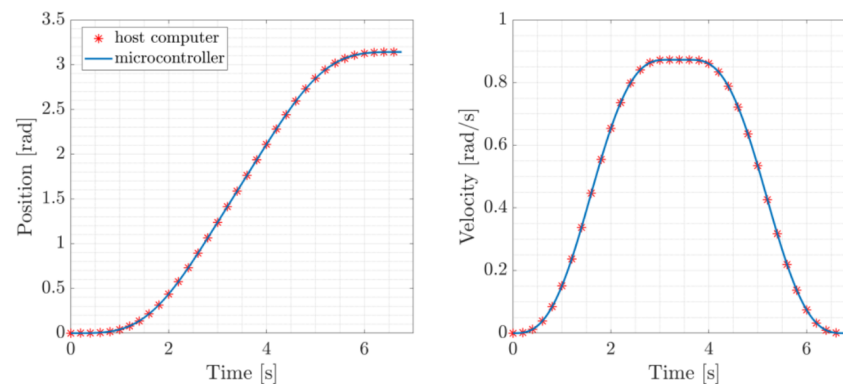


Figure 11 - Computed Host curve versus the points sent to the Servo-drive

3. Servo Drive:

The servo drive receives all the points one by one from the Host computer in real-time and interpolates between them through a third or fourth order equation, as for the fact that the servo drives' internal control loop usually runs up to 10 times faster than the PVT feeding rate from the Host side, so the servo drive has to interpolated at least 10 times between each two points sent from the Host till receiving the new points again.

The interpolation no matter how accurate, still won't be the same as the real points computed at the Host side and this might leave some errors and discrepancies between what has been computed on the Host and what has been achieved on the real motor.

Embedded Profile Generation Engine of SOLO

The embedded profile Generation Engine in SOLO, is like a PVT Encoder that is placed locally inside the Main Control Unit (DSP), in other words, the whole profile fitting, generation and execution is done locally inside SOLO's Firmware with the same execution rate of the controllers (Speed and Position) and this will have the following results:

- It Requires only the Target position from the user with no interpolation done between the points (all points generated locally at run-time)
- Much less occupation of the the Data lines (one target point instead of hundreds)
- No special programming skills required at the user side! (automatic profile generation)

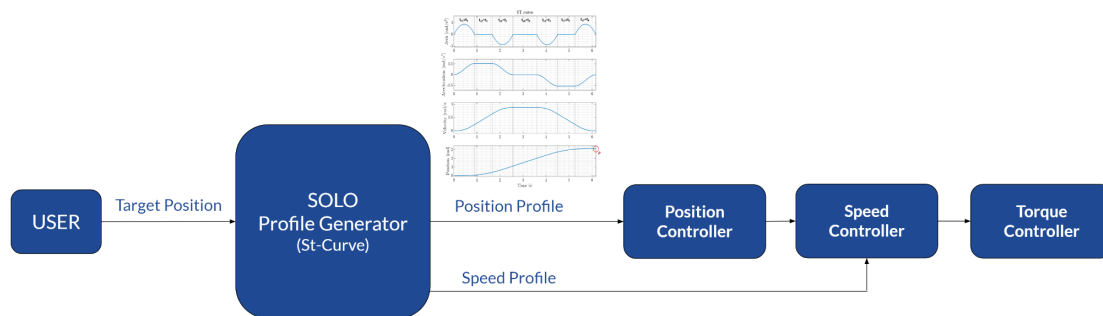


Figure 12 - SOLO Motion Profile Generation Engine diagram

Embedded St-curve Position Profile Generation

The St-curve is one the smoothest profiles in position controlling applications, not only it provides a very smooth transition from one position to the other (initial position to the target position) but also it offers smooth trajectories for velocity, acceleration and jerk profiles.

The jerk is actually the main point of focus as the eventual behavior of the jerk will define the quality of the motion in terms of having higher or lower vibrations and potential undesirable motions.

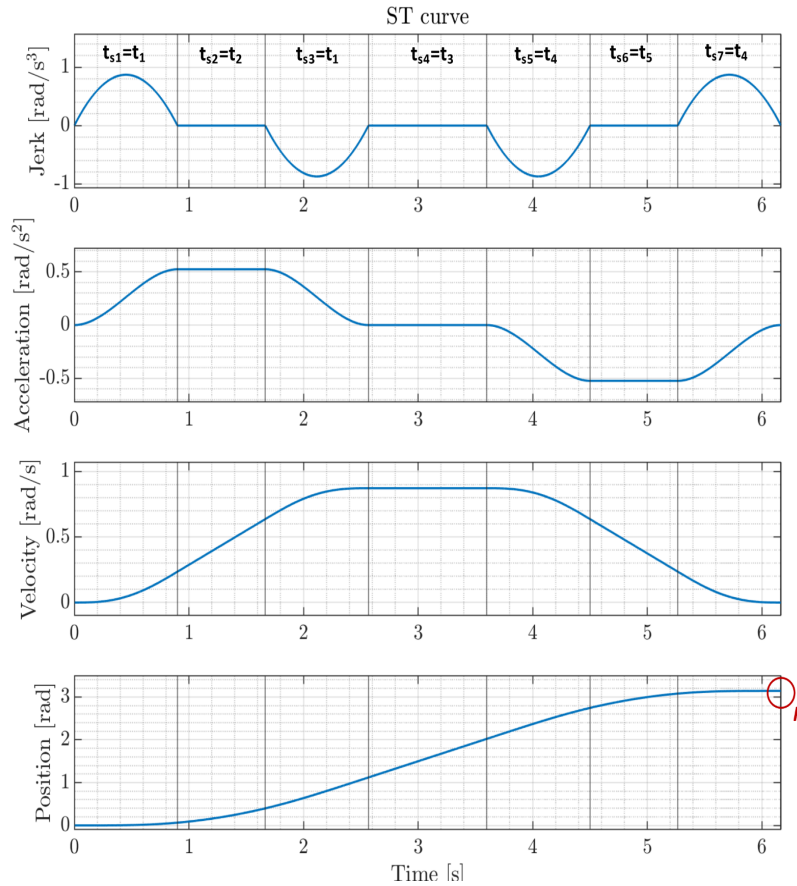


Figure13- St-Curve Position Profile combined out of 7 separate segments

To understand better the St-curve functionality, we consider a motion as the action of going from point A to point B, where point A stands for the initial position that can be anywhere in trajectory of the Motor, and point B is the target where we would like to achieve, the Point A in graphs shown in Figure 13 is considered as 0 and the Point B is shown with "P" that is the final or target position. The whole St-curve profile is divided into 7 different segments where the final trajectory is fitted as a smooth profile between point A to B within these 7 segments.

Each segment has a duration that is shown above with sign “ t_x ” that are having the following meaning:

- t_1 – duration of the segment with increasing and decreasing acceleration (parabolic jerk)
- t_2 – duration of the segment with the constant acceleration
- t_3 – duration of the segment with zero acceleration (constant velocity) phase
- t_4 – duration of the segment with decreasing and increasing deceleration (parabolic jerk)
- t_5 – duration of the segment with the constant deceleration

In SOLO’s Embedded engine the segments 1 and 3 are having identical duration, the same applies for segments 5 and 7.

Time-based versus Time-optimal Position Profile St-Curve

SOLO offers two different methods for computing the St-curve profile for a given motion as below:

- **Time-based St-curve:**

In Time-based St-curves, the user has to input the strict timings for each segment of the motion as an input to SOLO, these timings later will be used to create the St-curve profile locally.

- **Time-Optimal St-curve:**

Once using Time optimal St-curve, the user has to provide motion constraints like maximum velocities, maximum accelerations and maximum jerks for the whole trajectory and the timings of each segment is internally computed based on these constraints at run-time in the firmware of SOLO.

The Time-based St-curve has an intuitive notion as it works directly with timings, however the Time-optimal St-curve has some considerations that we will discuss in detail in the next section.



Time-Optimal Position Profile St-curve Constraints Validations

Maximum allowable jerk, acceleration, and velocity can be directly controlled, i.e. a time-optimal motion profile can be computed by providing maximum values of acceleration jerk J , deceleration jerk I , acceleration A , deceleration D , and velocity V . In this way, segments' duration is not explicitly provided, but they are computed at runtime in SOLO's firmware.

Given that t_1 , t_2 , t_3 , t_4 , and t_5 must be greater than or equal to zero, the following inequality constraints must be satisfied:

1. Velocity constraint

$$\frac{P}{V} > \frac{3A}{4J} + \frac{V}{2A} + \frac{4D}{4I} + \frac{V}{2D}$$

Equation 1

If desired maximum velocity does not satisfy this constraint, the maximum allowable velocity can be computed as:

$$V_{allowable} = \frac{-b + \sqrt{b^2 + 4aP}}{2a}, \text{ where } \{a = \frac{1}{2A} + \frac{1}{2D}, b = \frac{3A}{4J} + \frac{3D}{4I}\}$$

Equation 2

2. Acceleration constraint

$$V > \frac{3A^2}{2J}$$

Equation 3

If desired maximum acceleration does not satisfy this constraint, the maximum allowable acceleration can be computed as:

$$A_{allowable} = \sqrt{\frac{2}{3} * J * V_{allowable}}$$

Equation 4

3. Deceleration constraint

$$V > \frac{3D^2}{2I}$$

Equation 5



If desired maximum deceleration does not satisfy this constraint, the maximum allowable deceleration can be computed as:

$$D_{allowable} = \sqrt{\frac{2}{3} * I * V_{allowable}}$$

Equation 6

The user has to make sure, once in Time-optimal mode, they check the constraints with these given equations to make sure the computed allowable values are matching with their reference inputs to avoid having undesired motion.

St-curve Position Profile Tuning

Finding the optimum St-curve form for a given application is balancing the time spent transitioning from zero acceleration to maximum acceleration (segments 1 and 3) as a percentage of the time spent at maximum acceleration (segment 2). On the one hand, if we spend nearly all our time transitioning (segment 1 and 3) and very little time at maximum acceleration (segment 2), the motion will be extremely smooth, but the total time of the motion will be increased. This is because, during segment 1, the St-curve profile travels only a short distance, therefore, the more significant amount of time spent here will increase the overall transfer time. On the other hand, if we spend only a small fraction of time in segments 1 and 3, then the time of the total motion will be minimized, however, the profile will not be as smooth and may cause unwanted vibration in the load. However, there is a caveat because even relatively brief segments 1 and 3 can dramatically reduce load-induced oscillations.

In fact, there is often a 'magic' value for a given mechanism for the St-curve phase ratio, while too little transition time can inject too much vibrational energy, surprisingly, too much transition time can have a similar effect. There is often a specific value for jerk which 'sticks the landing' and significantly or entirely reduces settling oscillation at the end of the trajectory. Finding that value is the goal of the St-curve tuning process. Practically speaking, unless the weight of the load never changes or the weight of the load is exactly known, you will probably tune the St-curve shape for a point-to-point transfer of an average load. But even an average-optimized value can make a big difference in effective point to point transfer times. Two different examples of tuning the Position Profile St-curves are presented in Figures 14 and 15 below.



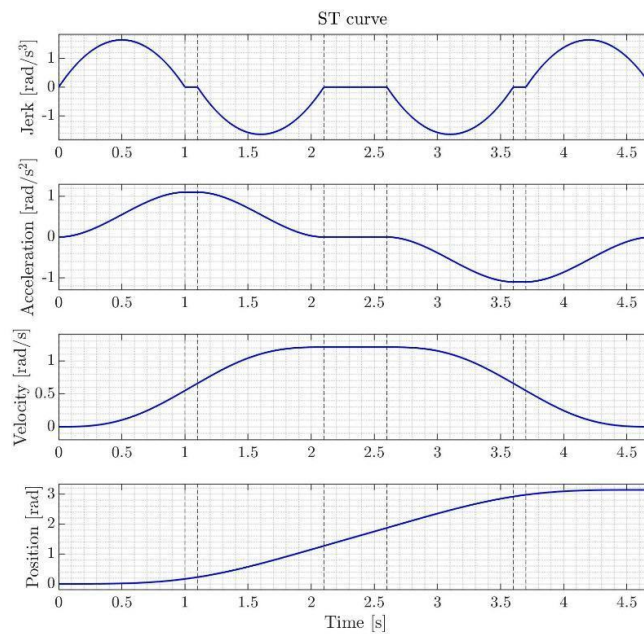


Figure 14: "Smooth" St-curve motion profile; $t_1=1s$, $t_2=0.1s$, $t_3=0.5s$, $t_4=1s$, $t_5=0.1s$

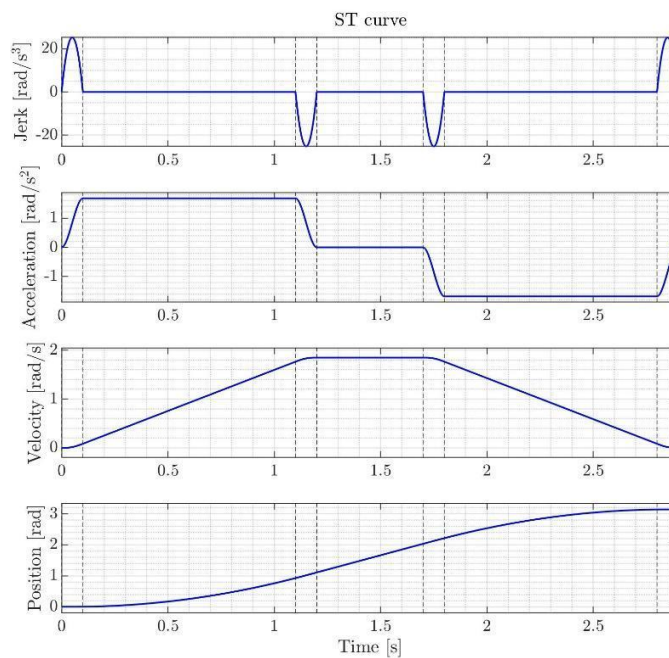


Figure 15: "Fast" St-curve motion profile; $t_1=0.1s$, $t_2=1s$, $t_3=0.5s$, $t_4=0.1s$, $t_5=1s$

Embedded St-curve Velocity Profile Generation

Besides position controlling, electrical motors are most commonly used in applications where speed controlling is essential. There are well-known techniques for controlling a motor speed around a reference point. However, there are numerous applications, like high-precision machining or human and goods transportation, where it is not only essential to reach the desired speed but also to track the desired speed profile. Considering the numerous advantages of the St-curve for motion control.

By observing Figure 13 of the St-curve position profile, we can see that the velocity profile reaches a constant value at the end of the third segment. Accordingly, a speed reference profile is created as a subset of the corresponding position profile as can be seen in Figure 16 below.

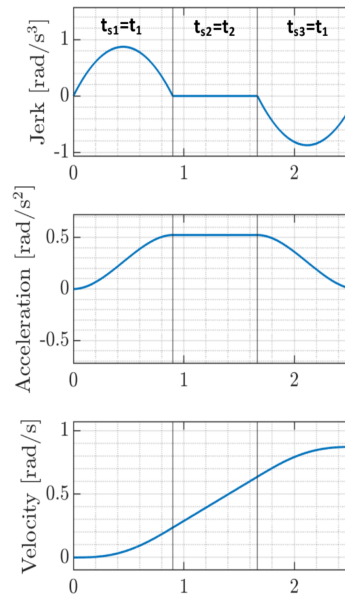


Figure16- St-Curve Velocity Profile combined out of 3 separated segments

St-curve can be considered an advanced version of the S-curve since it provides a continuous jerk profile. It also resolves the problems with the trapezoidal profile by adding two intermediate segments between transitions, having three motion phases in total. By defining a maximum acceleration jerk value, the acceleration does not switch instantly but changes smoothly until it reaches the maximum value, following a parabolic jerk profile. With the limited and controlled amount of jerk, St-curve provides smoother motion. However, the St-curve profile is incomparably powerful because it injects significantly less unwanted vibrational energy into the system, which substantially improves the overall performance.

The segments duration of the velocity profile in St-curve are:

- t_1 – duration of the segment with increasing and decreasing acceleration (parabolic jerk)
- t_2 – duration of the segment with the constant acceleration

Time-based versus Time-optimal Velocity Profile St-Curve

SOLO offers two different methods for computing the St-curve profile for a given motion as below:

- **Time-based St-curve:**

In Time-based St-curves, the user has to input the strict timings for each segment of the motion as an input to SOLO, these timings later will be used to create the St-curve profile locally.

- **Time-Optimal St-curve:**

Once using Time optimal St-curve, the user has to provide motion constraints like maximum accelerations and maximum jerks along the velocity trajectory and the timings of each segment is internally computed based on these constraints at run-time between each two target points.

The Time-based St-curve has an intuitive notion as it works directly with timings, however the Time-optimal St-curve has some considerations that we will discuss in detail in the next section.

Time-Optimal Velocity Profile St-curve Constraints Validations

In Time-Optimal velocity profiles, the Maximum allowable jerk and acceleration can be directly controlled, i.e. a time-optimal motion profile can be computed by providing maximum values of jerk J and acceleration A . In this way, segments' duration is not explicitly provided, but they are computed at runtime in SOLO's firmware.

Given that t_1 and t_2 must be greater than or equal to zero, the acceleration inequality constraint must be satisfied:

$$V > \frac{3A^2}{2J}$$

Equation 7

If desired maximum acceleration does not satisfy this constraint, the maximum allowable acceleration can be computed as:

$$A_{allowable} = \sqrt{\frac{2}{3}JV}$$

Equation 8



The user has to make sure, once in Time-optimal mode, they check the constraints with these given equations to make sure the computed allowable values are matching with their reference inputs to avoid having undesired motion.

St-Curve Velocity Profile Tuning

Finding the optimum St-curve form for a given application is a matter of balancing the time spent transitioning from zero acceleration to maximum acceleration (segments 1 and 3) as a percentage of the time spent at maximum acceleration (segment 2).

On the one hand, if we spend nearly all our time transitioning (segment 1 and 3) and very little time at maximum acceleration (segment 2), the motion will be extremely smooth, but the total time of the move will increase. This is because, during segment 1, the St-curve profile travels only a short distance, therefore, the greater amount of time spent here will increase the overall transfer time.

On the other hand, if we spend only a small fraction of time in segments 1 and 3, then the time of the total move will be minimized, however, the profile will not be as smooth and may cause unwanted vibration in the load. However, there is a caveat because even relatively brief segments 1 and 3 can dramatically reduce load-induced oscillations.

Practically speaking, unless the weight of the load never changes or the weight of the load is exactly known, you will probably tune the St-curve shape for a point-to-point transfer of an average load, two different examples of tuning the velocity St-curves are presented in Figures 17 and 18 below.

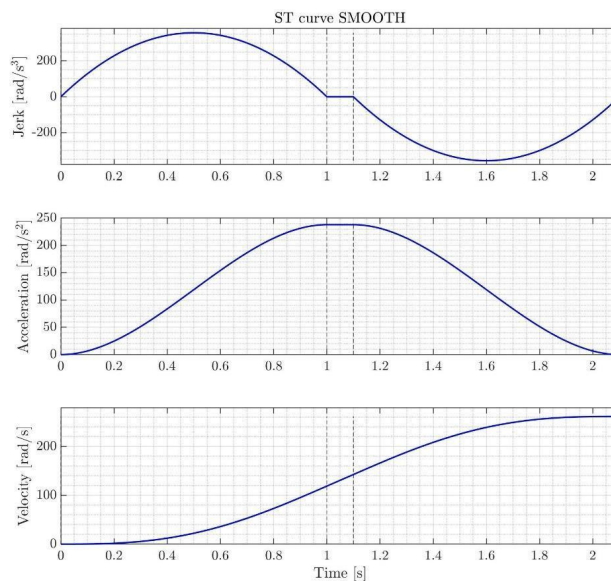


Figure 17: “Smooth” velocity St-curve motion profile; $t_1, t_3=1s$, $t_2=0.1s$



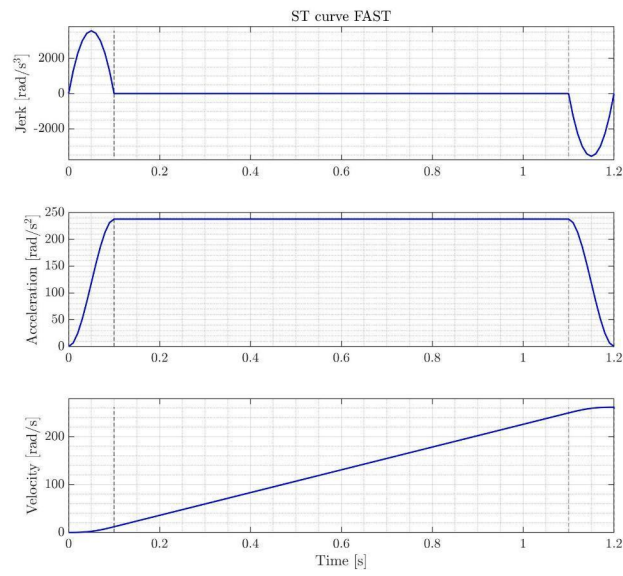


Figure 18: “Fast” velocity St-curve motion profile; $t_1, t_3=0.1$ s, $t_2=1$ s