Positional Motion Profiling for FRC

Ryan Greenblatt

A Zebramotion Labs White Paper:

Brought to you by Team 900:

http://team900.org

Motion Profiling Overview

This tutorial is intended for use of the CAN Talon SRX with LabVIEW running on the roboRIO. There is an example github project at https://github.com/FRC900/Motion-Profiling-Example. Specifically, to test and execute motion profiles run the Motion Profile Tester VI on the roboRIO. More details about the project can be found in this document.

We initially decided to investigate motion profiling because we were impressed by the Motion Planning & Control in FRC conference presented by team 254 and team 971 at the 2015 championship. If you haven't already seen the presentation we would highly recommend you watch it. Motion profiles consist of a series of points with a time value, a velocity value, and a positional value. The points are generated to always be physically possible for the mechanism. Motion profiles allow for very consistent, smooth motion. Having a tank drivetrain turn to a specific angle is difficult to execute quickly, precisely, and repeatedly because of wheels slipping. The method of generating a motion profile used here allows for control over the maximum jerk, acceleration, and speed of the motor so that it will not overshoot. Also, the jerk and acceleration can be set so the wheels never skid. Effectively, motion profiles allow an action like turning a drive train to an angle to be consistent and as fast as possible.

This example of motion profiling is run using LabVIEW on the roboRIO controlling Talon SRXs through CAN. This setup allows for sending the trajectory points of the motion profile to the Talon which sequentially executes the points when activated. Many other setups can be used but this one has several intrinsic advantages.

Having the motor controller execute the points independently from the roboRIO allows for greater resolution and more reliable point execution. The loop timing on the Talon is far more reliable than a loop running on the roboRIO so the execution is more consistent. Up to 128 points (the durations of the points are variable) may be preloaded while the motor controller is being used for another purpose or disabled. This makes it possible to execute a motion profile immediately at the start of a match or at the press of a button.

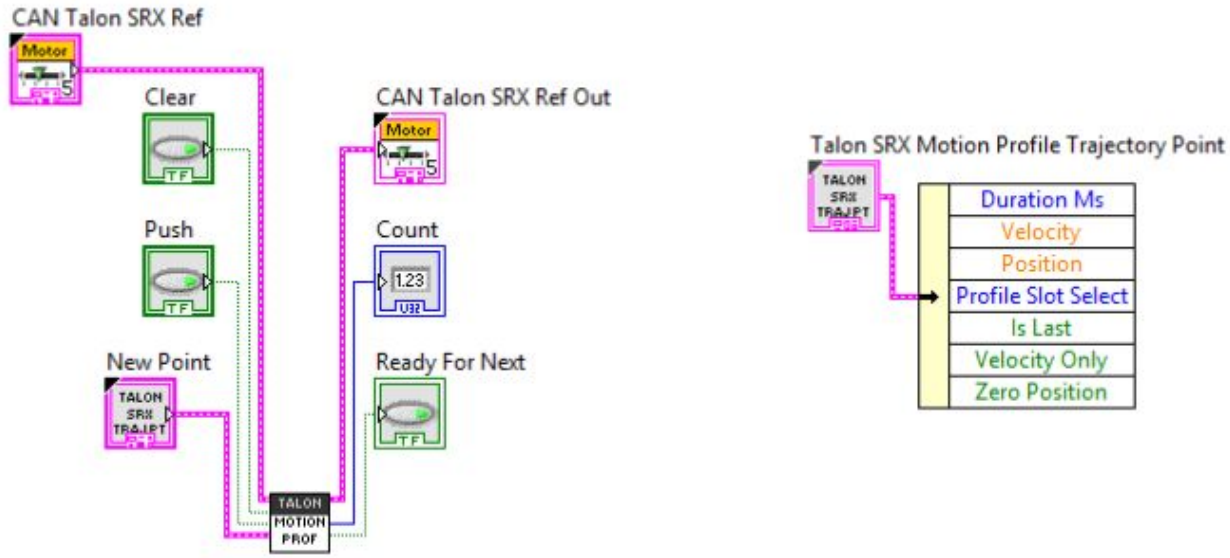
There are many limitations with this system as well. Some number of points must be preloaded to start the execution and make sure the buffer doesn't run out of points to execute. There is a limit to how fast the buffer can receive points so if

the profile can't be pre-generated and preloaded there will be a significant delay between when the profile generation is started and when it begins to run. This approach is less CPU intensive than most other methods of executing motion profiles. However, this approach can still be somewhat demanding on the CPU. If a high resolution profile is required the loop on the roboRIO that sends new points while the profile is running must be run quite quickly.

Motion Profiling API for the Talon SRX

For more detailed information about the Talon SRX API used for motion profiling please see the Talon SRX Motion Profile Reference Manual on the Cross the Road Electronics website.

The only additional VI needed in LabVIEW to use the Talon's motion profiling feature is WPI_CANTalonSRX_SendMotionProfile.vi which is found here: C:\Program Files (x86)\National Instruments\LabVIEW 2015\vi.lib\Rock Robotics\WPI\CAN\TalonSRX.



Inputs

- CAN Talon SRX Ref: The reference for the Talon
- Clear: If this input is true the entire buffer of points on the Talon will be cleared
- Push: If this input is true the "New Point" input will be added to the buffer of trajectory points on the Talon. A new point may be pushed regardless of what mode the Talon is in.

- New Point (cluster):
    - Elements (which are actively used when testing and executing motion profiles)
        - Velocity: The velocity that the motor should run at this point in either RPM or native units depending on the selected feedback reference of the motor. See the Talon SRX Software Reference Manual on the Cross the Road Electronics website for more information. The Talon will use feedforward to attempt to achieve this velocity. More detail about feedforward will be later in this document.
        - Position: The position where the motor should be at this point either in rotations or ticks depends on the selected feedback reference of the motor. The Talon will use the PID at the 0 parameter slot to attempt to achieve this position using the position output of an encoder to do this.
        - Zero position: All other points loaded to the buffer after this one will be treated as relative to this point. Often this is set to be true on the first point.
        - Velocity Only: If this is true the position input for this point will be ignored and only feedforward using the velocity input will be used. Please note that velocity-only mode is not an effective way to use a motion profile to try to get a motor to a specific velocity. The mode doesn't use any feedback at all so any environmental differences will cause a significant change in the velocity which is reached.
        - Is Last: This will cause the Talon to continue trying to get to the target position using positional PID even after the duration of the point has expired.
        - Duration MS: This will tell the Talon how long to run this point. 10 MS is a typical value but any value between 1 and 255 MS can be used. The loop which runs the motion profile on the Talon runs every millisecond.

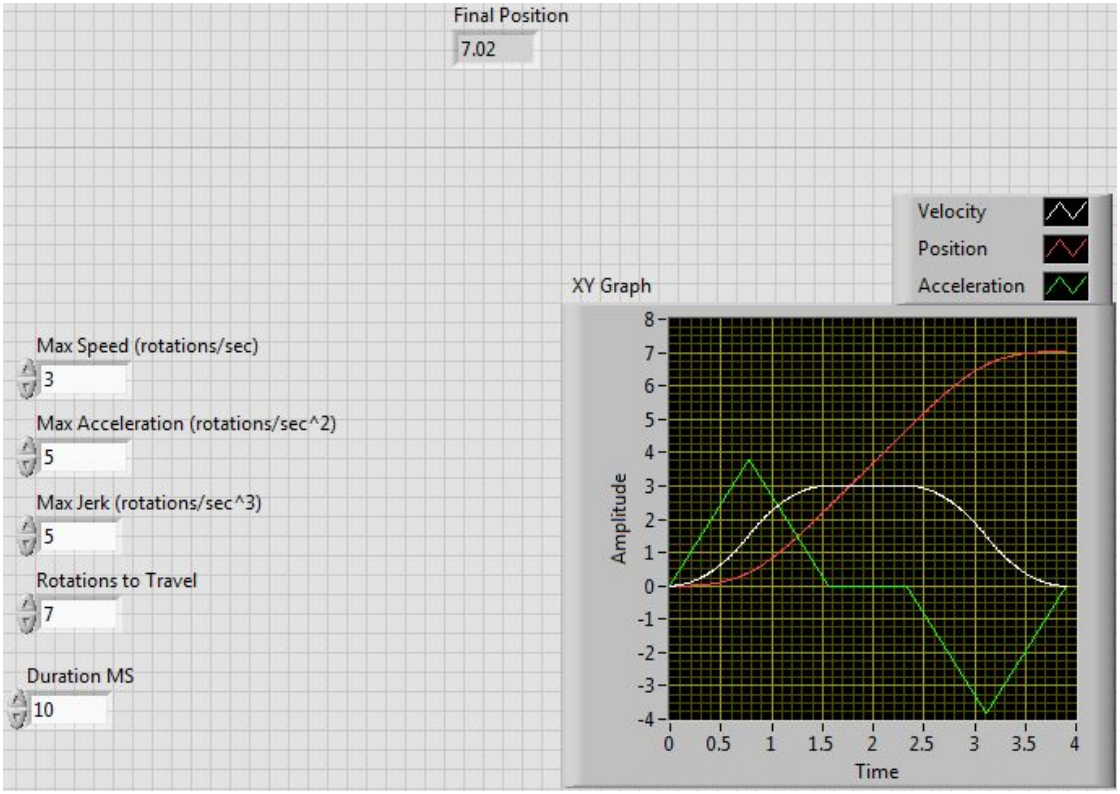
Outputs:

- CAN Talon SRX Ref Out: The reference for the Talon.
- Count: The number of points in the motion profile at the moment.
- Ready For Next: Whether or not the Talon is ready for the next point. Check that this is true before adding another point.

Execution

A loaded profile can be executed by putting the Talon into motion profile mode and setting the motor output to 1. This can either be set using the motor set output VI or Talon change mode VI. An output of 0 will disable the output of the motor controller and an output of 2 will cause the motor controller to hold the current point using positional PID. The Velocity set for the held point should be 0. After finishing executing the motion profile set the output back to 0 while in motion profile mode to make sure that the motion profile is finished correctly.

Point Generation

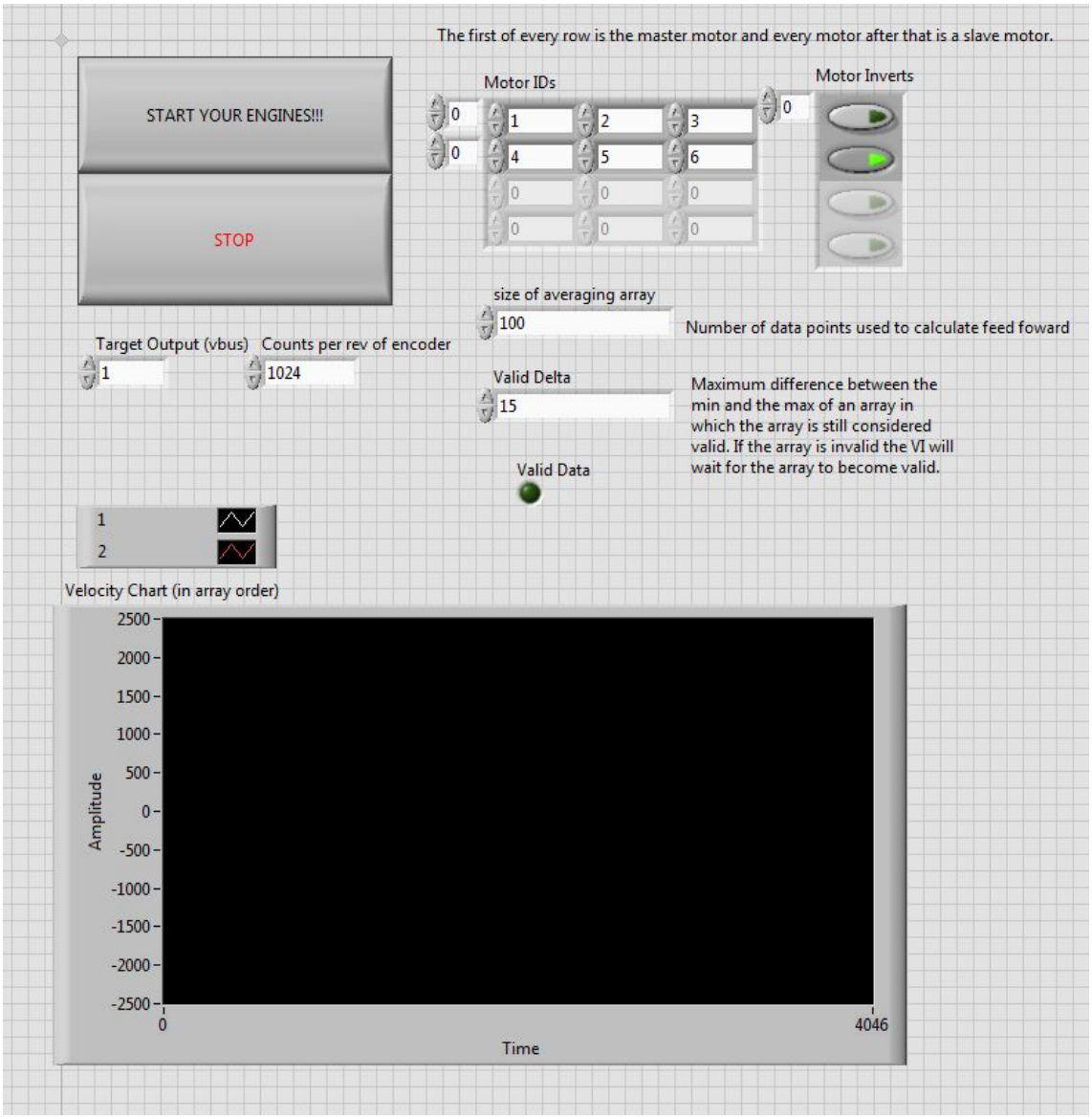


There are a multitude of different approaches to generating velocity and position points for a motion profile. The method used in the provided example project, which the Motion Profile Generator-Demo VI pictured above runs, was created by Paul Copioli with a few modifications. Specifically, a limitation was placed on maximum velocity and acceleration so that the maximum speed and acceleration are always

obtainable. If this isn't done and the maximum speed and acceleration inserted into the VI aren't obtainable then the profile either won't get to the target position or the profile won't be completed as quickly as is possible. This approach allows for control over the maximum speed, acceleration, and jerk. See this http://www.chiefdelphi.com/forums/showthread.php?s=821dac53446a6368545632e5cc108d9d&t=98358 chief delphi thread for more information about the "Copioli" method of motion profiling and see this http://www.chiefdelphi.com/forums/showthread.php?t=147293 chief delphi thread for more information about why maximum acceleration and speed must be limited. Please note that the math used in the code for limiting max acceleration is now slightly different than the math in the thread.

Setting Up and Running a Motion Profile

First, one should obtain an F value for the motor. This may either be done manually or using the Calculate Feedforward VI contained within the example project in the Calculate Feedforward folder. The VI should only be used if the motor can be allowed to spin quickly for an extended period of time. Open up the feedforward project contained within the folder and then the VI. Within the VI the target throttle should be set to 1 for the purposes of motion profiling, the size of the averaging array should be set to roughly 100, and the valid delta should be set to around 15. More information about what these numbers do is in the VI. If the feedforward value output by the VI is negative the motor should be inverted.

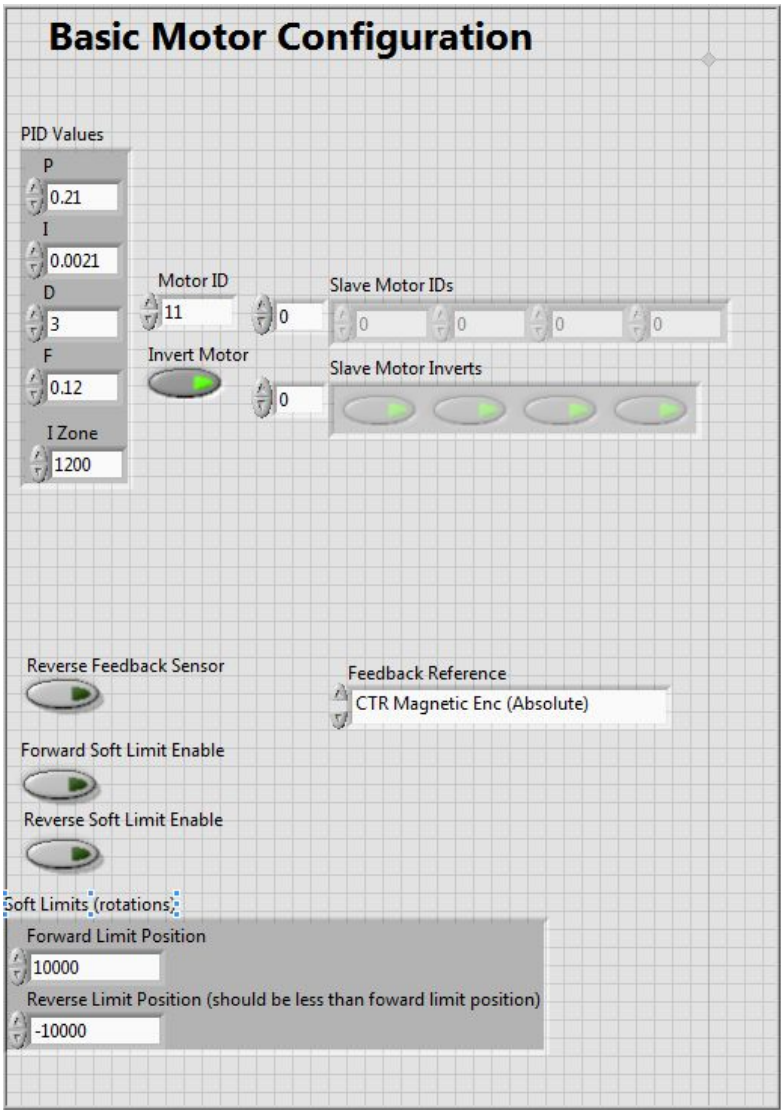


Now you will need to obtain positional PID values for the motor. Keep in mind that the PID shouldn't be very aggressive and it doesn't need to be particularly well tuned; the feedforward should do most of the work. For more information about effective positional PID tuning please see the Talon SRX Software Reference Manual on the Cross the Road Electronics site.

After that, open up the Motion Profiler Tester VI found within the Motion Profiling VIs folder. You will need to copy over your motor IDs, inverts, and PIDF values. Additionally, make sure you select the appropriate encoder type; it will affect some of the unit conversion math. If any encoder other than a Cross the Road Electronics magnetic encoder (analog or digital) is used, many of the units will no longer be valid,

but the VI will still work. (In the 2017 season units will likely be standardized. Section 21.25. FRC2016 LabVIEW: API Unit-Scaling Inconsistencies within the Talon SRX Software Reference Manual contains detail about current inconsistencies and the manual will contain details about units in the future.)

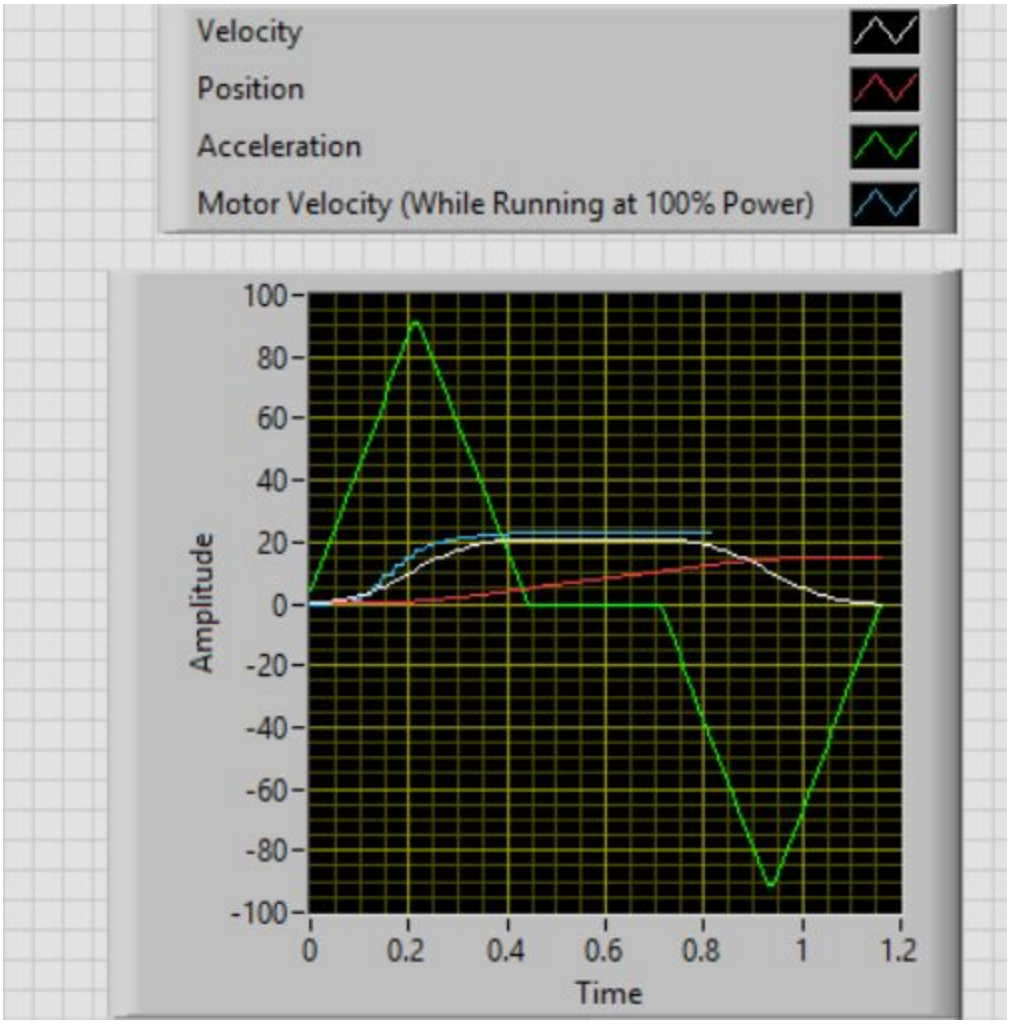
For example, maximum speed won't actually be in rotations/sec it will be in ticks/sec.



The first step is determining the correct maximum speed, acceleration, and jerk. Start off by running the motor at 100% voltage while the motor is under typical load. Please note that the motor will spin very quickly. If the motor can't be allowed to spin freely, another method must be used to determine the correct maximum speed,

acceleration, and jerk. Abort this operation when the motor has reached maximum speed.

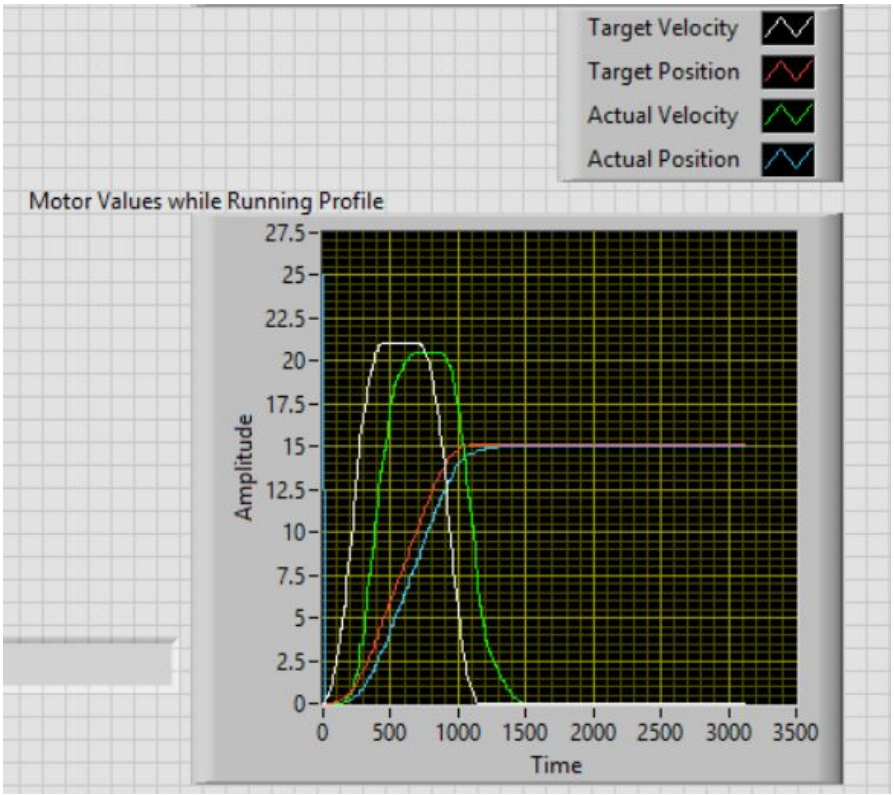
The VI should look like this at this point:



Next, use the graph of the motor velocity under 100% power to determine what max acceleration, speed, and jerk should be. After guessing or calculating the

maximums, generate a motion profile which has a large enough target position such that motor will reach maximum speed. Compare the velocity line from the generated motion profile with the velocity line while running at 100% speed. The velocity of the motion profile should always be below the velocity while running at 100% power. This could require adjustments of maximum speed, acceleration, and jerk. After you have obtained valid maximums go ahead and test your motion profile by generating it, loading it, and (as soon as it is done being loaded) running it. The maximums or PID values might need to be adjusted if the motor has difficulty keeping up or the applied throttle is higher than desired while running the profile.

An effective execution should look something like this:

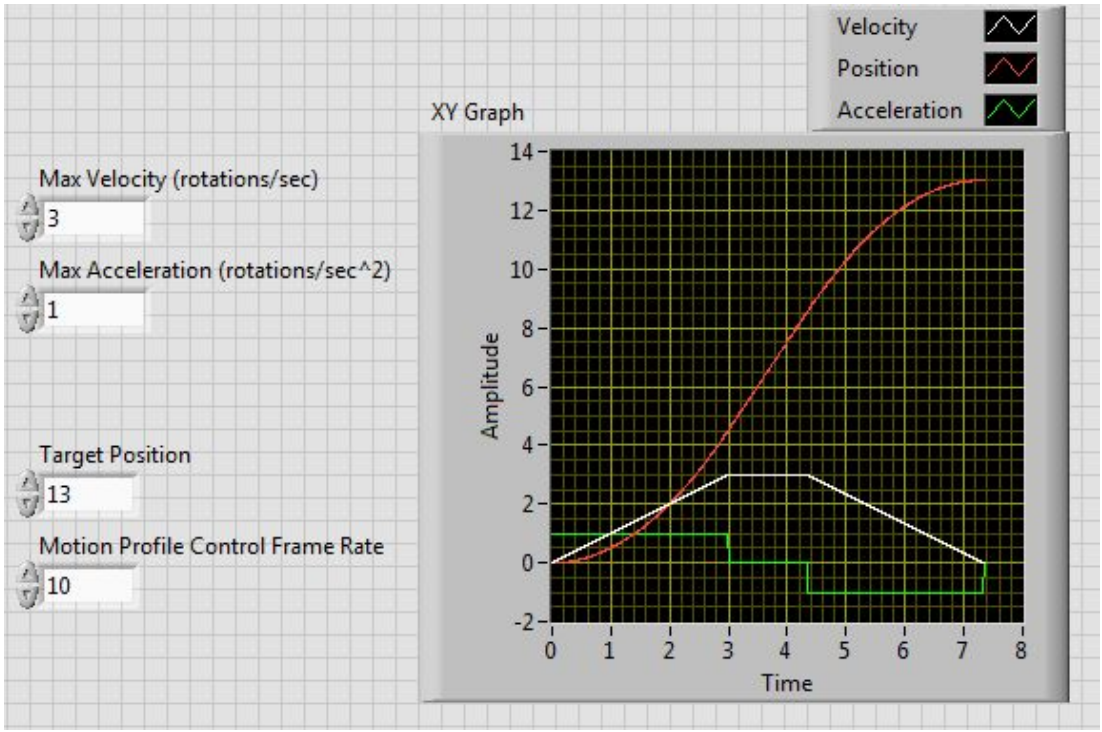


The actual velocity lags the target velocity slightly, because the sensor velocity output is averaged over a small duration.

Motion Magic

Motion magic is a new and unimplemented control mode for the Talon SRX. The mode will automatically generate and follow a motion profile to achieve a target location based on a specified cruising velocity, acceleration, and PIDF. This effectively means that the motor will follow a trapezoidal profile.

One example trapezoidal profile:



Because all of the points are generated on the Talon, the profile can be started almost instantly and CPU usage on the roboRIO is negligible. However, trapezoidal profiles can be jerkier than “S-curve” profiles. Following a trapezoidal profile perfectly would theoretically require the motor to experience infinite jerk at one or two points in time. Therefore, this control mode will be somewhat less accurate and consistent than executing a “S-curve” profile using the motion profile mode but far more accurate than using PID.