How to Combine S-curve Moves with CML

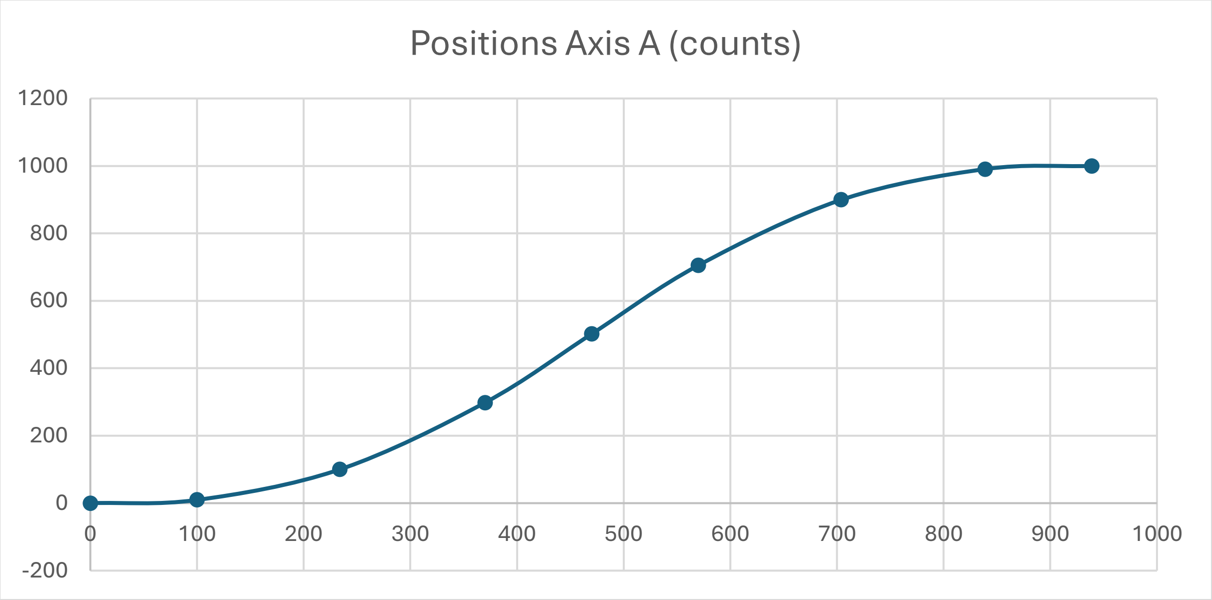
First, review the C++ example code on the Copley Controls GitHub page.

```
double velocity = 10000; // firmware units = 0.1 counts/sec
double accel = 10000;    //                = 10 counts/sec^2
double decel = 10000;    //                = 10 counts/sec^2
double jerk = 100000;    //                = 100 counts/sec^3
```

PVT table for first move on Axis A:

PVT Point #	Time Absolute (ms)	Positions Axis A (counts)
1	0	0
2	100	9.6225
3	234	100.139
4	370	298.009
5	470	501.765
6	570	705.043
7	704	899.495
8	839	990.566
9	939	1000

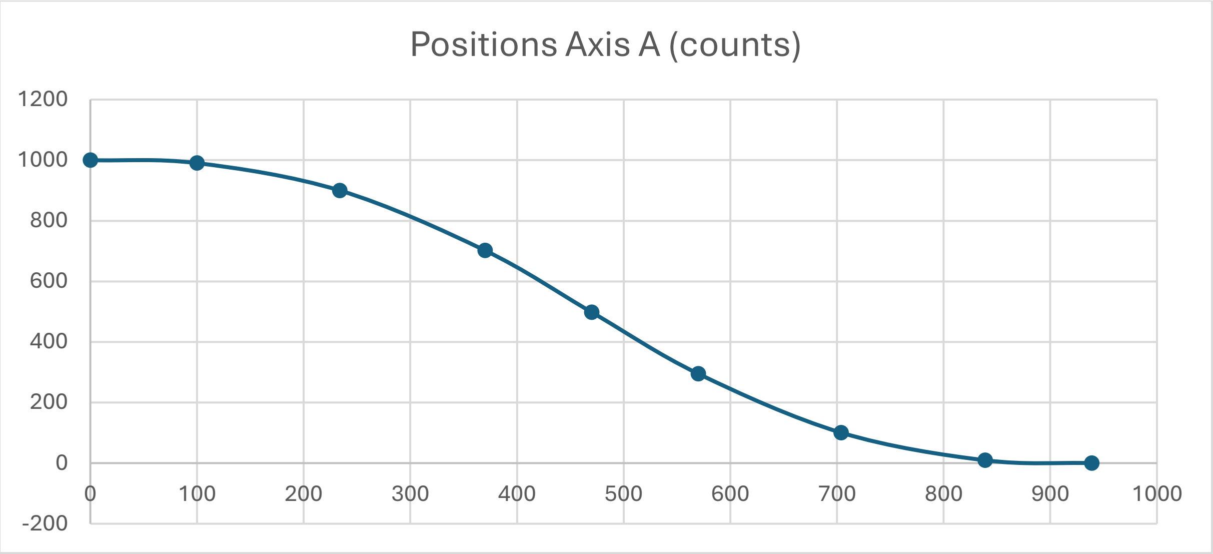
First move on Axis A:



PVT table for second move on Axis A:

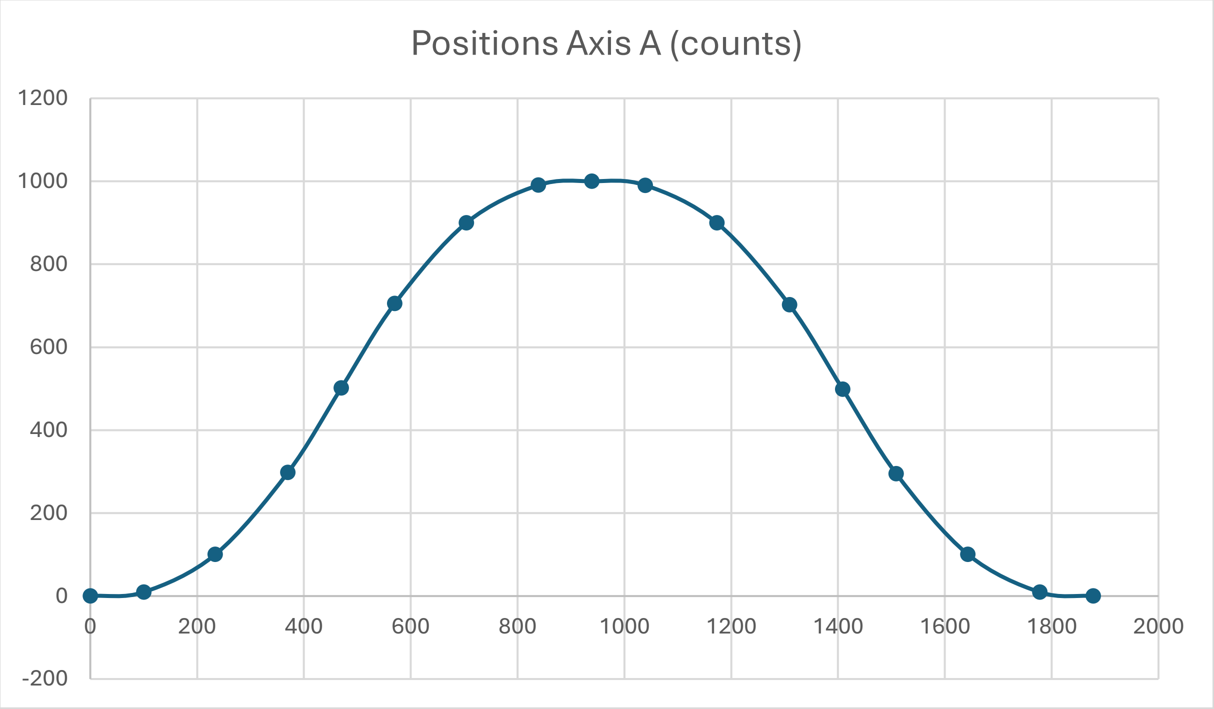
PVT Point #	Time Absolute (ms)	Positions Axis A (counts)
1	0	1000
2	100	990.377
3	234	899.861
4	370	701.991
5	470	498.235
6	570	294.957
7	704	100.505
8	839	9.43432
9	939	0

Second move on Axis A:

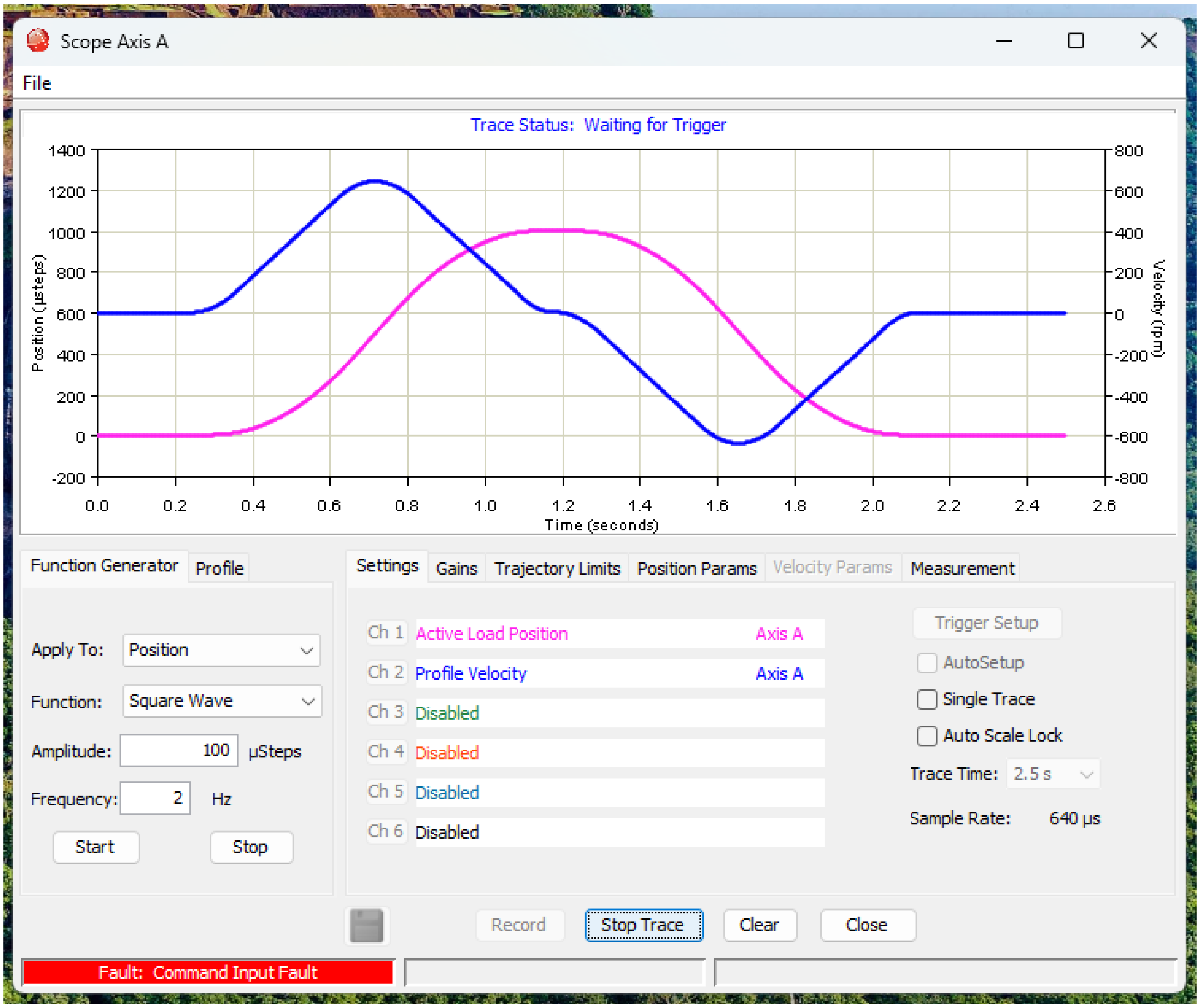


When we combine these two moves, we can erase the last point of the first move because it is the same as the starting point for the second move.

PVT Point #	Time Absolute (ms)	Positions Axis A (counts)
1	0	0
2	100	9.6225
3	234	100.139
4	370	298.009
5	470	501.765
6	570	705.043
7	704	899.495
8	839	990.566
9	939	1000
10	1039	990.377
11	1173	899.861
12	1309	701.991
13	1409	498.235
14	1509	294.957
15	1643	100.505
16	1778	9.43432
17	1878	0



In a real system, here is the performance:



Trace Status: Waiting for Trigger

Apply To: Position

Function: Square Wave

Amplitude: 100 µSteps

Frequency: 2 Hz

Ch 1 Active Load Position Axis A

Ch 2 Profile Velocity Axis A

Ch 3 Disabled

Ch 4 Disabled

Ch 5 Disabled

Ch 6 Disabled

□ AutoSetup

□ Single Trace

□ Auto Scale Lock

Trace Time: 2.5 s

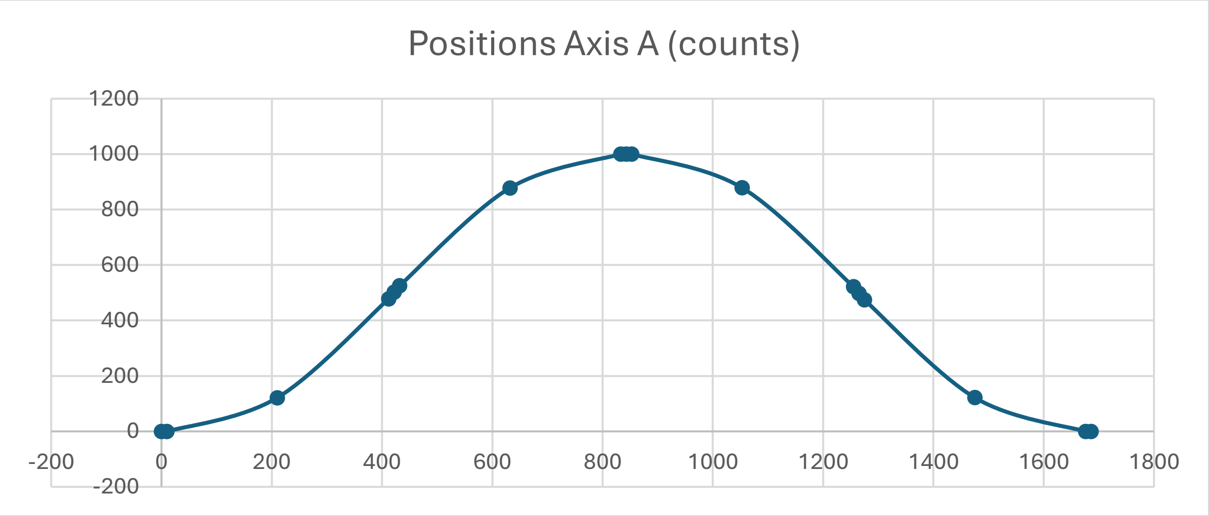
Sample Rate: 640 µs

Fault: Command Input Fault

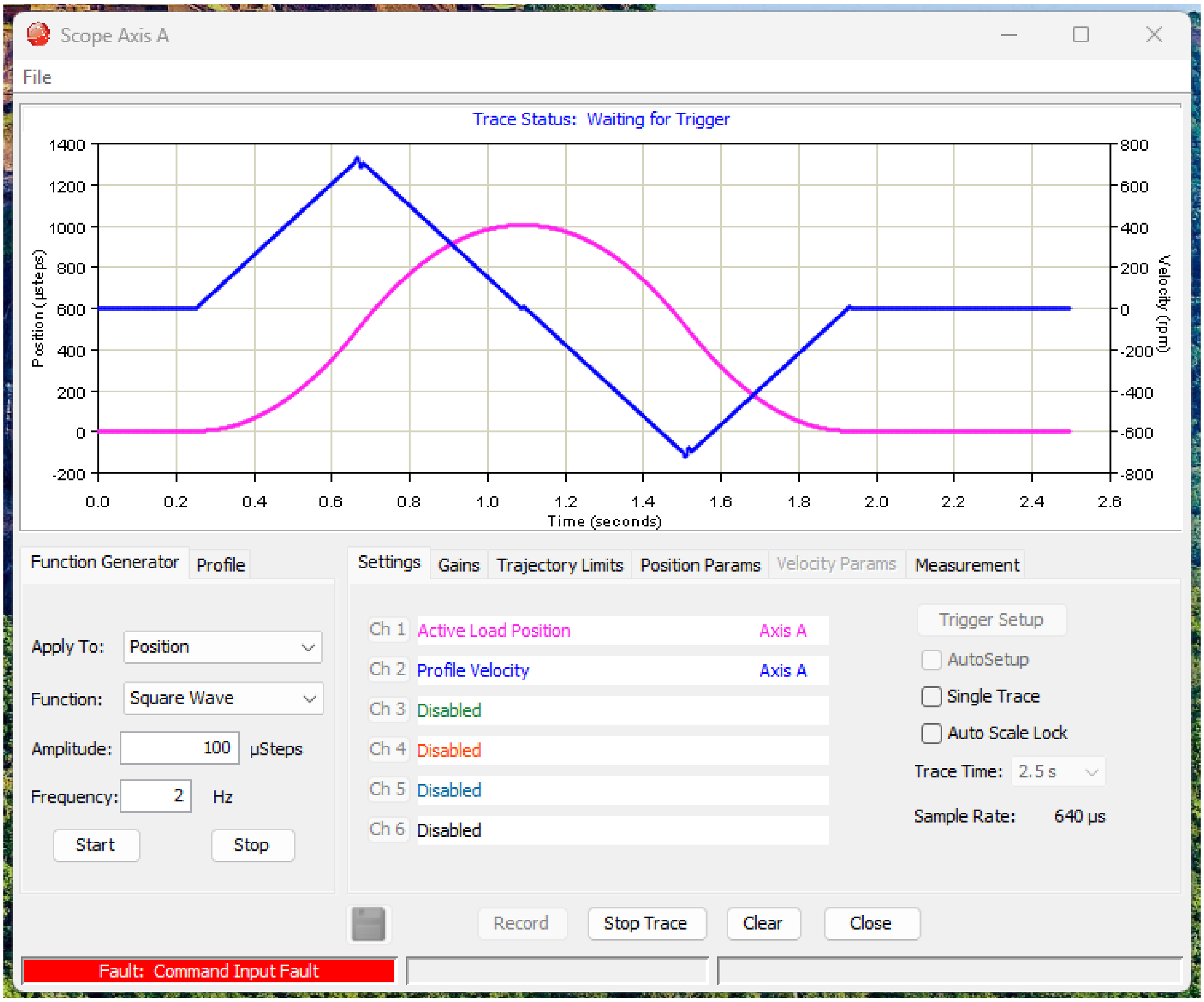
Too much settling time between the two moves? Increase your jerk value in CML.

```
    double velocity = 10000; // firmware units = 0.1 counts/sec
    double accel = 10000;    //                = 10 counts/sec^2
    double decel = 10000;    //                = 10 counts/sec^2
    double jerk = 1000000;   //                = 100 counts/sec^3
```

PVT Point #	Time Absolute (ms)	Positions Axis A (counts)
1	0	0
2	10	0.096225
3	210	121.34
4	412	478.211
5	422	501.877
6	432	525.498
7	632	878.165
8	833	999.92
9	843	1000
10	853	999.904
11	1053	878.66
12	1255	521.789
13	1265	498.123
14	1275	474.502
15	1475	121.835
16	1676	0.0803922
17	1686	0



Here is what this profile looks like in a real system.



Apply To: Position

Function: Square Wave

Amplitude: 100 µSteps

Frequency: 2 Hz

Ch 1 Active Load Position Axis A

Ch 2 Profile Velocity Axis A

Ch 3 Disabled

Ch 4 Disabled

Ch 5 Disabled

Ch 6 Disabled

□ AutoSetup

□ Single Trace

□ Auto Scale Lock

Trace Time: 2.5 s

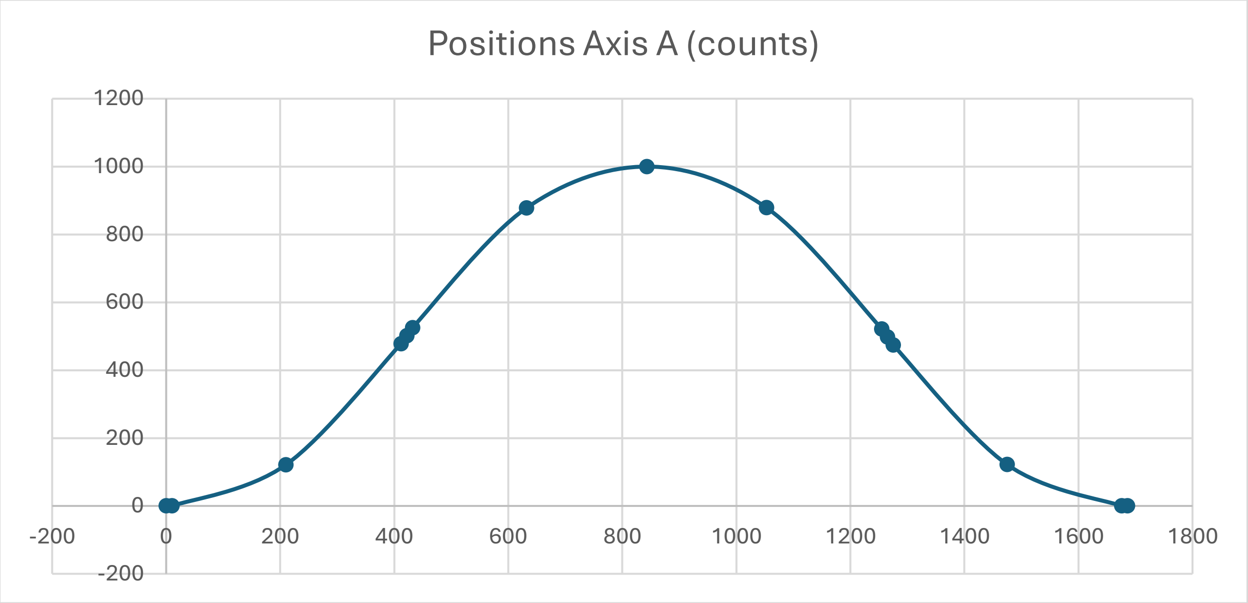
Sample Rate: 640 µs

Fault: Command Input Fault

Remove PVT points #8 and #10 adjacent to the apex to further smooth the transition between the two moves.

New Table:

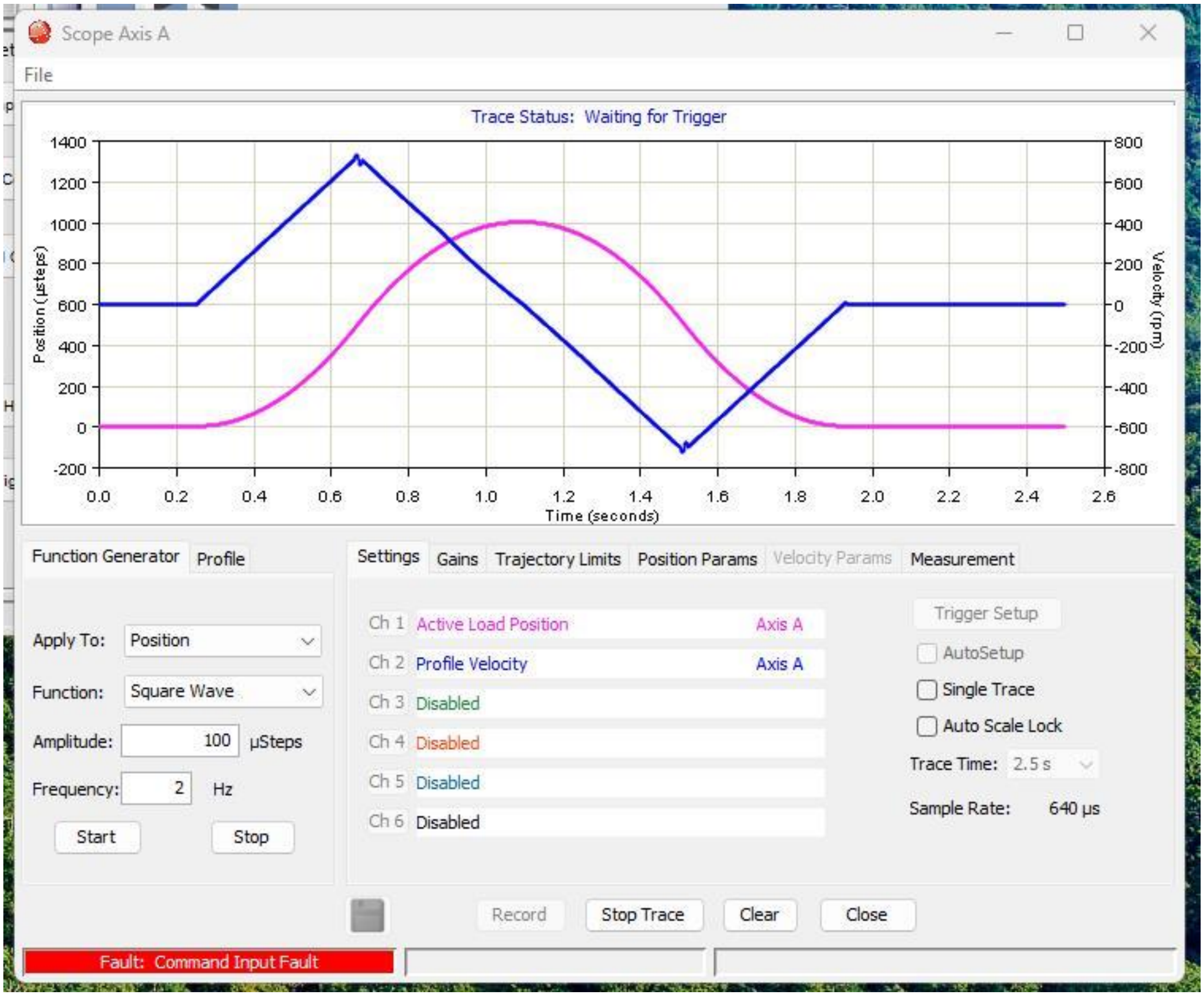
PVT Point #	Time Absolute (ms)	Positions Axis A (counts)
1	0	0
2	10	0.096225
3	210	121.34
4	412	478.211
5	422	501.877
6	432	525.498
7	632	878.165
8	843	1000
9	1053	878.66
10	1255	521.789
11	1265	498.123
12	1275	474.502
13	1475	121.835
14	1676	0.0803922
15	1686	0



Now we can manually define these points in a vector in C++ and add them to the PvtObj in CML.

```
    vector<vector<double>> manuallyEditedPositions =
    {
        {0.0},
        {0.096225},
        {121.34},
        {478.211},
        {501.877},
        {525.498},
        {878.165},
        {1000.0},
        {878.66},
        {521.789},
        {498.123},
        {474.502},
        {121.835},
        {0.0803922},
        {0.0}
    };

    vector<uint8> manuallyEditedTimes = { 10, 200, 202, 10, 10, 200, 211, 210, 202,
10, 10, 200, 201, 10, 0 };

    LoadPointsIntoPvtObj(manuallyEditedPositions, manuallyEditedTimes, pvtObj);

    err = link.SendTrajectory(pvtObj);
    showerr(err, "starting the move");

    err = link.WaitMoveDone(-1);
    showerr(err, "waiting for move to finish");
```

This is an example of manual trajectory editing. The resulting profile velocity looks like more of a straight line during the transition between the two moves.



Trace Status: Waiting for Trigger

Apply To: Position

Function: Square Wave

Amplitude: 100 µSteps

Frequency: 2 Hz

Ch 1 Active Load Position Axis A

Ch 2 Profile Velocity Axis A

Ch 3 Disabled

Ch 4 Disabled

Ch 5 Disabled

Ch 6 Disabled

☐ AutoSetup

☐ Single Trace

☐ Auto Scale Lock

Trace Time: 2.5 s

Sample Rate: 640 µs

Fault: Command Input Fault