Trajectory Generation

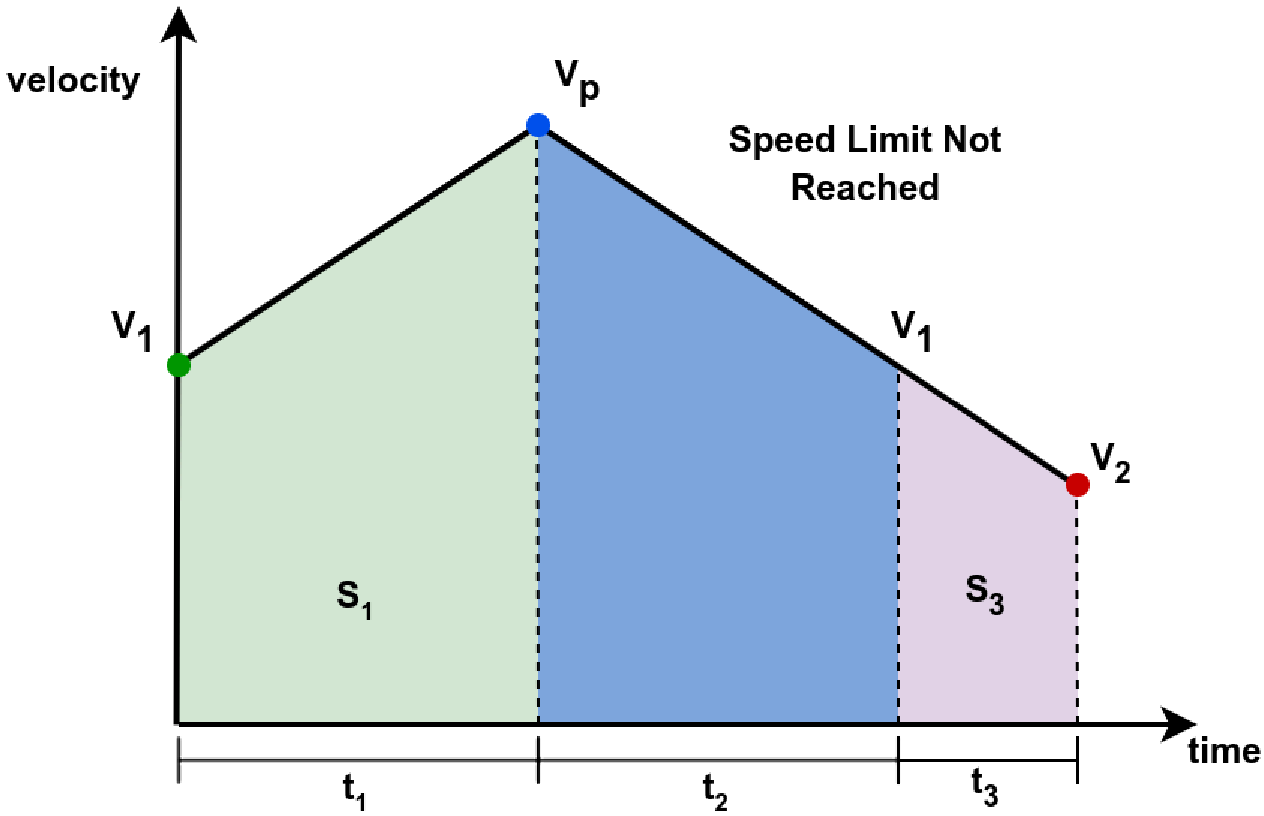
Trajectory Generation

Consider a robot moving along a straight line trajectory. At the beginning of the trajectory, the robot will have some starting velocity which may be zero, a target final distance and a target final speed which may also be zero. There are likely to be constraints on the maximum permitted speed and it is easiest to set a fixed value for acceleration and deceleration during the move.

To execute the trajectory, you will need to calculate the robot's velocity at every point along the path. The aim is to ensure that the target end conditions are met as closely as possible. There are many ways to achieve this. Consider first a calculated profile which will be triangular unless the maximum permitted speed limits velocity. In general, this is a trapezoidal profile. the profile needs to have a fixed point at which acceleration stops and when braking begins. It will also need to tell when the profile is complete. For many purposes, the most important ending condition os that the target position be met as accurately as possible so that is the most likely candidate for the terminating condition.

Trapezoidal Profile

Ignoring edge cases for now, assume the final velocity is smaller than the starting velocity and that both are non-zero. The straight line motion trajectory has a profile that looks like this.



We have a starting velocity,v_1, and an end velocity, v_2.

Assuming the speed does not reach a limit, it will peak at v_p after a time, t_1.

The profile velocity, after reaching its peak will pass back through v_1 again before reaching v_2. During that final phase it is easy to calculate the distance travelled, S_3.

$$S_3 = \frac{|\,(v_2^2 - v_1^2)\,|}{2a} \tag{1}$$

And the time needed for that phase, t_3 is just

$$t_3 = \frac{(v_2 - v_1)}{a} \tag{2}$$

For the profile generation, we need to calculate the time at which acceleration stops and deceleration begins. This is t_1.

Note that, if $v_1 < v_2$ then the time until the change from acceleration to deceleration is $t_1 + t_3$.

During that acceleration period, the distance travelled is S_1 (shaded in green) which can be calculated as:

$$S_1 = v_1 t_1 + \frac{1}{2} a t_1^2 \tag{3}$$

The period during which the speed reduces once more to v_1 is symmetrical about t_1 and so covers the same distance. Thus, the distance moved during the green and blue sections together is just $2S_1$ or

$$2S_1 = 2v_1t_1 + at_1^2 \tag{4}$$

Since we know the total distance, S, to be covered by the profile, we can write (4) in terms of things we know

$$2S_1 = S - S_3 = 2v_1t_1 + at_1^2 \tag{5}$$

Rearrange to get the quadratic equation in t_1:

$$at_1^2 + 2v_1t_1 + S_3 - S = 0 \tag{6}$$

Substitute for S_3 from (1):

$$at_1^2 + 2v_1t_1 + \frac{(v_1^2 - v_2^2)}{2a} - S = 0 \tag{7}$$

This can be solved in the normal way for quadratics by equating terms. Thus, if

$$Ax^2 + Bx + C = 0$$

has solutions:

$$x = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

then we have for t_1:

$$t_1 = \frac{-2v_1 \pm \sqrt{4v_1^2 - 4a\left(\frac{(v_1^2 - v_2^2)}{2a} - S\right)}}{2a}$$

$$t_1 = \frac{-2v_1 \pm \sqrt{4v_1^2 - 2v_1^2 + 2v_2^2 + 4aS}}{2a}$$

$$t_1 = \frac{-2v_1 + \sqrt{2v_1^2 + 2v_2^2 + 4aS}}{2a} \tag{8}$$

In a quadratic solution, the section in the square root, $\sqrt{2v_1^2 + 2v_2^2 + 4aS}$, is called the *determinant* of the solution. In general, the determinant can be added or subtracted to give either of the two solutions a typical quadratic has. In this equation, (8), the determinant is the part that calculates the time taken to pass through the symmetrical, triangular postion of the trajectory. It will always be positive because negative time would be tricky to explain away.

(NOTE: a different derivation would probably show more clearly the relationship between the various components of the solution. Perhaps later...)

Notice that the value of the determinant in (8) is added to the first term, which is $-2v_1$. This is a result of choosing $v_1 > v_2$. For the opposite condition, we still add the determinant but the first term is $-2v_2$.

The actual code simply uses the initial velocity for the first phase and the terminal velocity for the second phase and the numbers all work out.

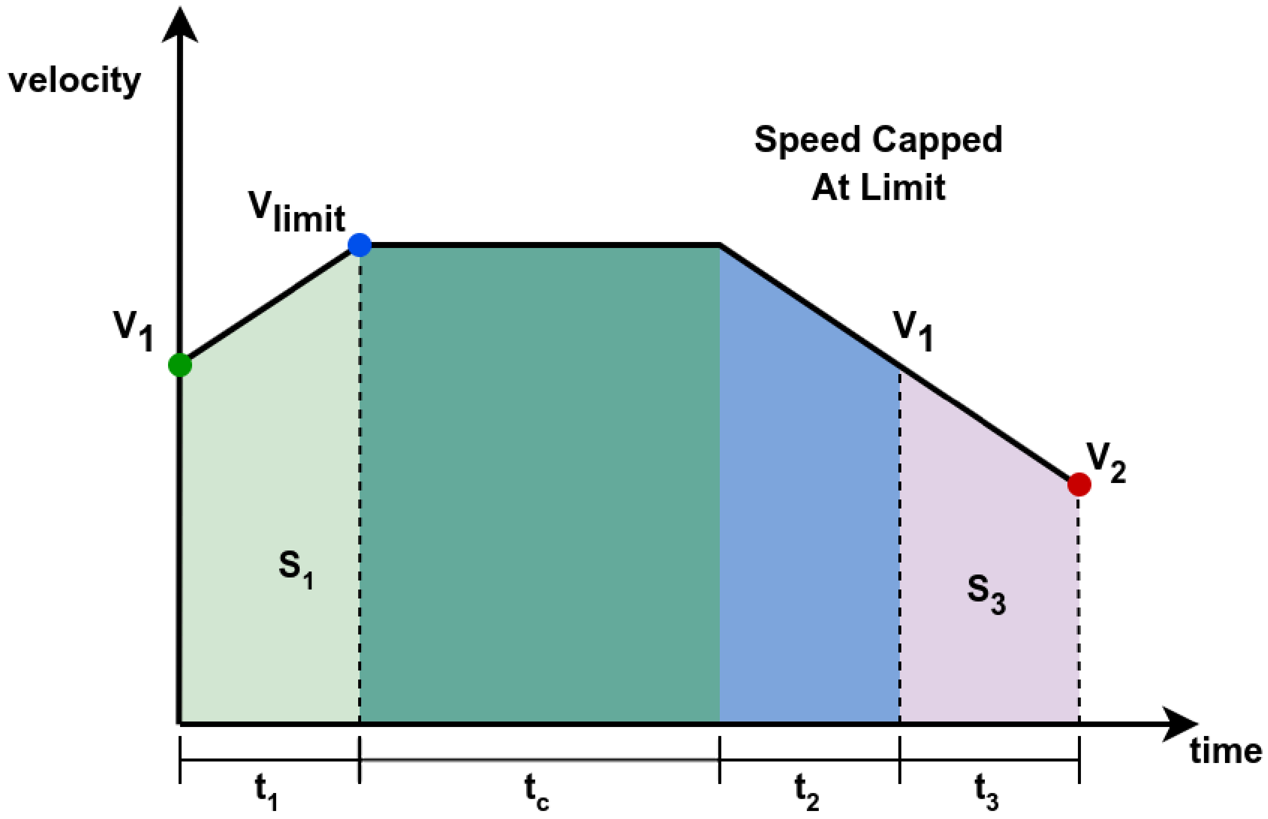
(TODO: a better explanation is needed for that)

Ensuring accuracy

In the code we can only perform calculations to update the velocity at fixed time intervals of dt so once the etimate of t_1 and t_3 have been made, the value used for the acceleration, a, is recalculated to guarantee no rounding errors.

Speed Limiting

If you were wondering what happened to t_2, consider the case where a limit is placed on the maximum achievable speed.



Now we must re-calculate the times to account for the coasting period where velocity is constant. This is done in the code and the derivation is left as an exercise for the reader.

Edge Cases

The most obvious edge cases are where the start and finish velocities are the same or where they are zero. These all come out in the wash with the calculations shown.

The most significant case that must be addressed is where the start or finish velocities are bigger than the velocity limit. As it stands, the code will fail under thos circumstances as one or other of the accelerating phases will have a negative time.

(TODO: Fix those cases)

Clothoid Arcs

The same technique could be used for turns where the forward motion of the robot is constant throuighout a changing angular velocity profile.

Normally a smooth moving turn would have $v_1 = v_2$ resulting in a clothoidal curve that starts and ends with straight-line motion and a smooth transition. For most cases there would be an angular velocity limit that will give a circular arc for the centre portion. Calculating the 2D path of such a trajectory is not so trivial and is best left for another occasion. Feel free though to experiment with trajectories that attempt to join curves with differing curvatures.

In Place (Pivot) Turns

If the robot has no forward motion, these profiles can be used for a turn-in-place that is accurate and repeatable.