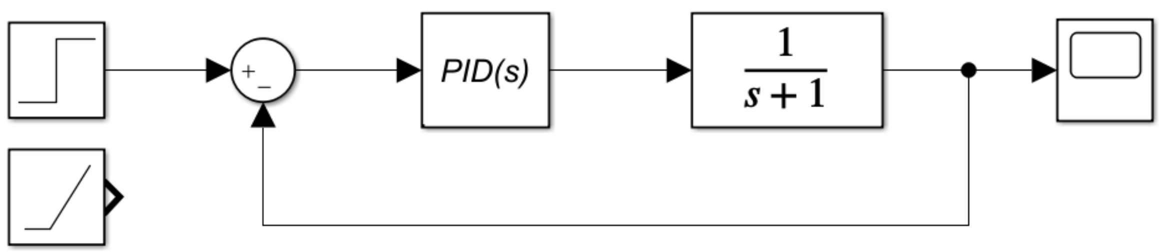
Time Optimal and Minimum Jerk Profiles for Motion Control– A Tutorial

[The material in this tutorial is based on standard curriculum of K. N. Toosi University of Technology, Faculty of Electrical Engineering. For more information, please write to Mohammad Kia: M_Kia@hotmail.com]

What are we talking about?

In all concepts and sections of robotics, mechatronics and control systems engineering, when we want to design a controller for a system, no matter what the system is, we study the step, ramp or Dirac delta impulse responses. Considering the controller is perfect and the system will follow the step input, there will be two challenging questions to be answered here:

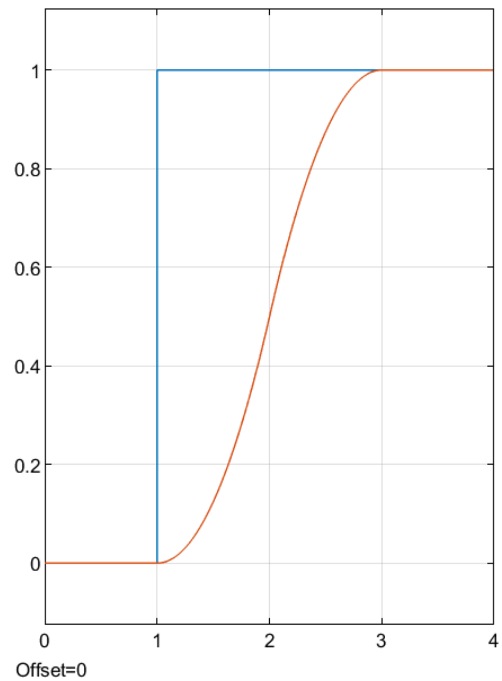


1) **Do we really need the system to behave like a step or ramp function?!** Consider the altitude control system of an airliner (a type of aircraft for transporting passengers and air cargo) during takeoff. You are sitting relaxed with seatbelts fastened, then you will be shot into the air at the height of 3000ft like a catapult! That is not what you want.
2) **Should the controller bandwidths and gains be tuned such that they could follow a step input?!** That is, when you know that you do not want the system to be following an impulse, step or even a ramp input, should you tune the controller for those behaviors or you could perform the tuning in a more optimum way?

The answer to such questions has been found in years of implementing mechatronics systems, especially when the target plant is to be interfaced and interacted with a human being. And the answer is ***absolutely not!*** Scientific studies in bio-mechatronics and cybernetics shows that different motional behavior of the human body such as walking and running, reaching and grasping, turning and rotating, and so on, follow some specific patterns proportional to the jerk of the current movement which is what we are going to discuss here. Moreover, when we want to detail every step of the current movement, it is much easier to change the input of the system instead of interfering with the controller itself. Furthermore, take the PID controller for instance, if we input a step function into it, it will push a step pattern and its derivative, an infinite impulse, into the target plant. This will cause a pump and glitch in the mechanical system and if the mechanical system is something sensitive like an aircraft, it will lead to a total disaster. Last but not least, when we do not want to have an ultimate fast response, such as step, we can tune the controller in a more aggressive way with higher bandwidths and gains, and this will give us a perfect trajectory tracking and more energy and time saving results in the end.

What should we achieve?

Up to now we have undrestood that instead of a step function we should input the system with somethig like this figure.



Now the main question comes in to part: ***what should the characteristics of this perfectly balanced curve be?*** The answer lies within the concept of ***jerk***. Neville Hogan (1984) noted that smoothness can be quantified as a function of jerk, which is the time derivative of acceleration. Hence, jerk is the third time derivative of location (i.e., position). If the location of a system is specified by variable x(t) , then the jerk of that system is:

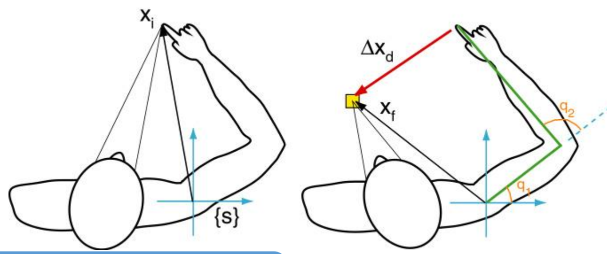
$$\dddot{x}(t) = \frac{d^3x(t)}{dt^3}$$

Now, as a general information which is essential to know when you are working on motion control systems, here we have provided a perfect table about derivatives and integrals of position and their names in physics and dynamics.

Derivative	Terminology	SI Unit	Meaning
-5	Absounce	$m.s^5$	Time integral of abserk
-4	Abserk	$m.s^4$	Time integral of abseleration
-3	Abseleration	$m.s^3$	Time integral of absity
-2	Absity	$m.s^2$	Time integral of absement
-1	Absement (Absition)	$m.s^1$	Time integral of position
0	Position (Displacement)	m	Position
1	Velocity	$m.s^{-1}$	Rate-of-change of position
2	Acceleration	$m.s^{-2}$	Rate of change of velocity
3	Jerk	$m.s^{-3}$	Rate of change of acceleration
4	Jounce (Snap)	$m.s^{-4}$	Rate of change of jerk
5	Crackle	$m.s^{-5}$	Rate of change of jounce
6	Pop	$m.s^{-6}$	Rate of change of crackle
7	Lock	$m.s^{-7}$	Rate of change of pop
8	Drop	$m.s^{-8}$	Rate of change of lock

The curved profile above in motion control systems is called ***trajectory***. It is believed that for controlling systems which have interaction with human body involvement, such as automotive cruise control, elevator and escalator movements, active joint prostheses like elbow and knee prostheses, airliners, surgery robots and so on, the absolute value of jerk and how it changes defines the smoothness of the motion. Three renowned and most commonly used type of these profiles are **Time Optimal S-Curve**, **Time Optimal ST-Curve**, and **Minimum Jerk Trajectory**.

Now consider a point-to-point hand movement control system of the body. The relation of position, speed, acceleration and jerk in both x and y coordinates to time is the trajectory. The overall flowchart for a control system will be:



Define a set of motion constraints for a controlled mechanical system

Receive a command to transition the controlled mechanical system to a new position or a new velocity

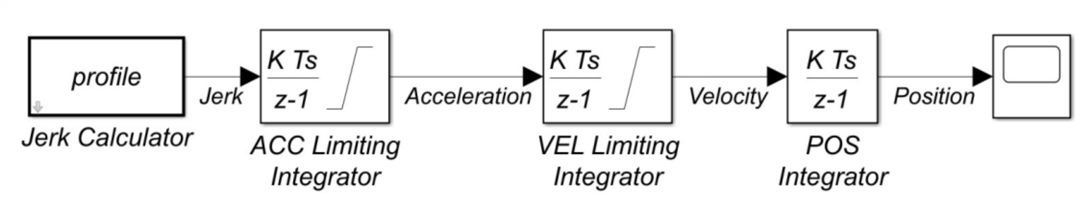
Calculate a motion profile for traversing to the new position or the new velocity, the motion profile including a continuous jerk reference

Instruct the controlled mechanical system to traverse from its current position or velocity to the new position or velocity according to the motion profile

For every desired profile we have a set of motion constraints and a reference position or velocity defined:

- P is the position step,
- J is the maximum acceleration jerk,
- I is the maximum deceleration jerk,
- A is the maximum acceleration,
- D is the maximum deceleration, and
- V is the maximum velocity.

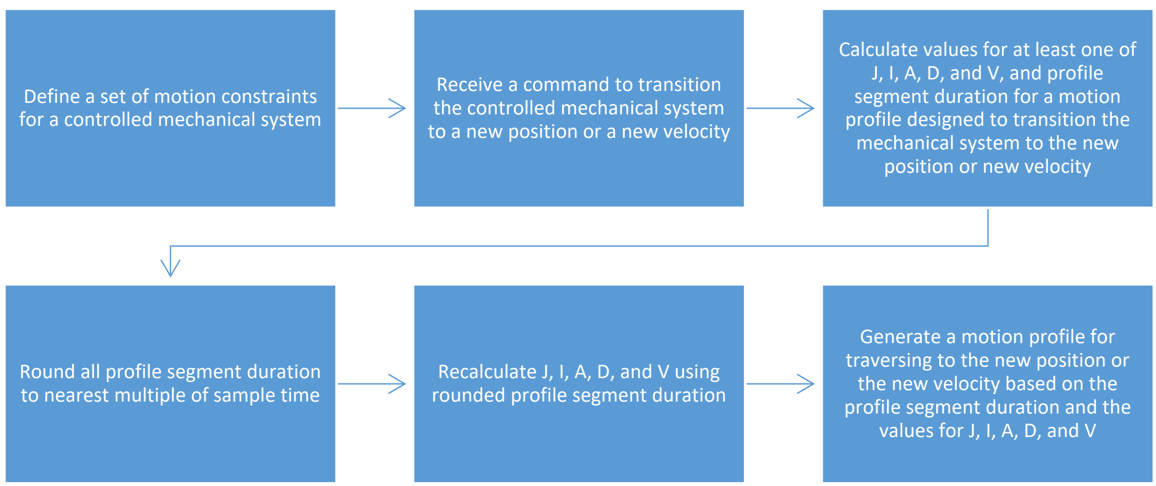
Considering the sets of constraints, it can be calculated whether the system can or cannot transfer to the desired state from its initial state. Know that if the profile generation is performed online, the initial state is always the current state. The conditions to check this possibility is presented in the following sections. If the equations have an answer, the whole profile can be generated by merely integrating the needed jerk in every time instance. The integrators apply the maximum acceleration and velocity limits. So all we need to do is to precisely calculate the jerk at each sample time.



Time Optimal Profiles - Time Based Approach

Systems and methods are provided for generating a constraint-based, time-optimal motion profile for controlling the trajectory of a point-to-point move in a motion control system. A profile generator can calculate an S-Curve or ST-curve motion profile that includes a jerk reference that varies continuously over time for at least one of the motion profile segments, thereby producing a smooth, time-optimal trajectory. The profile generator can create the motion profile to conform to a set of motion constraints provided by the user. The profile generator also supports calculation of time-optimal motion profiles having segments that align to the sample time of the motion control system. In some embodiments, the profile generator can efficiently generate the motion profile by performing reference calculations only for those segments that will be used in the final motion profile for a given point-to-point move.

So the overall flowchart for a control system is:



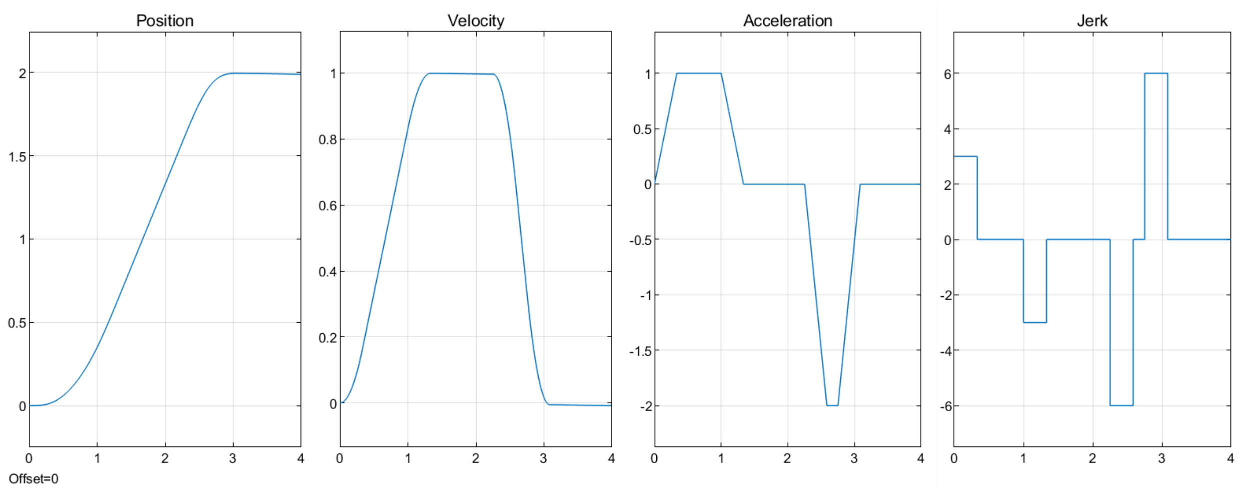
A for a time based S-Curve or ST-Curve profile we have seven performance segments:

Number	Segment Name	Description
1	Acceleration Increase	Acceleration profile increases from zero to maximum acceleration
2	Acceleration Hold	Acceleration profile stays constant as the maximum acceleration
3	Acceleration Decrease	Acceleration profile decreases from maximum acceleration to zero
4	Velocity Hold	Velocity profile stays constant
5	Deceleration Increase	Deceleration profile absolute value increases from zero to maximum deceleration

6	Deceleration Hold	Deceleration profile absolute value stays constant as the maximum deceleration
7	Deceleration Decrease	Deceleration profile absolute value decreases from maximum deceleration to zero

S-Curve Profile Formula

A typical S-Curve profile is something like the figure below. From left to right we can see the positon, velocity, acceleration and jerk values in relation to the spent time. We can clearly see the 7 profile segments described above in here.



The jerk formula as a function of time is achieved by this function (it means give J for t_1 seconds, then give zero for t_2 seconds, after that give –J for t_1 seconds again and so on):

$$\ddot{\theta}(t) = \begin{cases} J & 0 < t < t_1 \\ 0 & 0 < t < t_2 \\ -J & 0 < t < t_1 \\ 0 & 0 < t < t_3 \\ -I & 0 < t < t_4 \\ 0 & 0 < t < t_5 \\ I & 0 < t < t_4 \end{cases} \quad where \quad \begin{cases} t_1 = \dfrac{A}{J} \\ t_2 = \dfrac{V}{A} - \dfrac{A}{J} \\ t_3 = \dfrac{P}{V} - \dfrac{A}{2J} - \dfrac{V}{2A} - \dfrac{D}{2I} - \dfrac{V}{2D} \\ t_4 = \dfrac{D}{I} \\ t_5 = \dfrac{V}{D} - \dfrac{D}{I} \end{cases}$$

Given that t_1, t_2, t_3, t_4 and t_5 should all be greater than or equal to zero the following set of inequalities can be established:

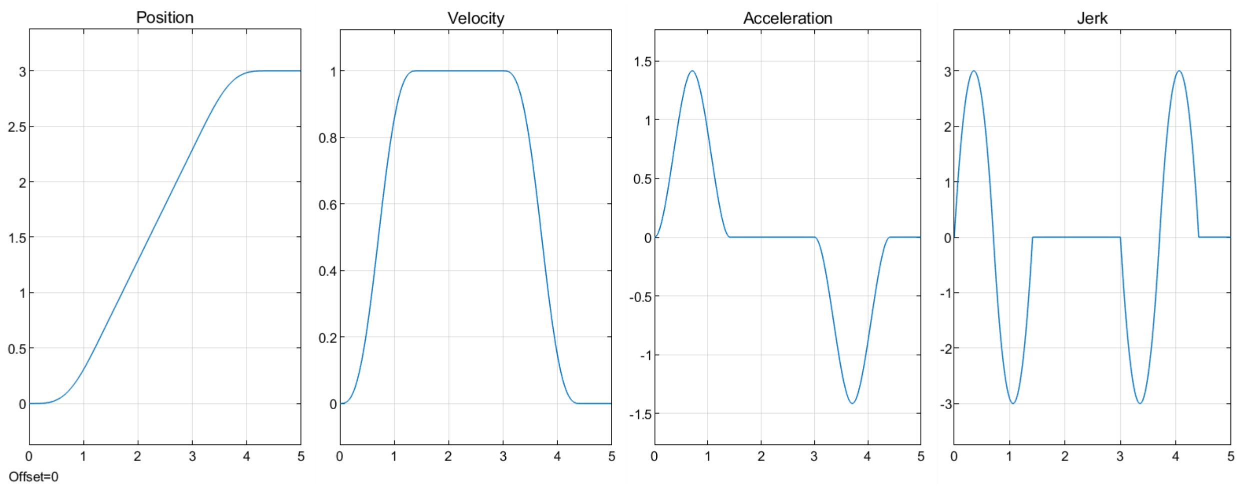
$$V > \frac{A^2}{J} \quad , \quad V > \frac{D^2}{I} \quad , \quad \frac{P}{V} > \frac{A}{2J} + \frac{V}{2A} + \frac{D}{2I} + \frac{V}{2D}$$

For back calculation and precise computation of the maximum accelerations and velocity we can use:

$$
\begin{cases}
A = P\dfrac{2}{(t_1+t_2)(t_5+2t_4+2t_3+2t_1+t_2)} \\
D = P\dfrac{2}{(t_4+t_5)(t_5+2t_4+2t_3+2t_1+t_2)} \\
V = P\dfrac{2}{t_5+2t_4+2t_3+2t_1+t_2}
\end{cases}
$$

ST-Curve Profile Formula

A typical ST-Curve profile is something like the figure below. We can see the 7 profile segments and positon, velocity, acceleration and jerk values just like before. We can see that the main difference between S-Curve and ST-Curve is the jerk.



The jerk formula as a function of time is achieved by the below function (it means that for each segment generate a parabola:

$$
\ddot{\theta}(t) = \begin{cases}
K_1 t(t_1 - t) & 0 < t < t_1 \\
0 & 0 < t < t_2 \\
-K_1 t(t_1 - t) & 0 < t < t_1 \\
0 & 0 < t < t_3 \\
-K_2 t(t_4 - t) & 0 < t < t_4 \\
0 & 0 < t < t_5 \\
K_2 t(t_4 - t) & 0 < t < t_4
\end{cases}
\quad where \quad
\begin{cases}
t_1 = \dfrac{3A}{2J} \\
t_2 = \dfrac{V}{A} - \dfrac{3A}{2J} \\
t_3 = \dfrac{P}{V} - \dfrac{3A}{4J} - \dfrac{V}{2A} - \dfrac{3D}{4I} - \dfrac{V}{2D} \\
t_4 = \dfrac{3D}{I2} \\
t_5 = \dfrac{V}{D} - \dfrac{3D}{2I}
\end{cases}
$$

$$
and \begin{cases}
K_1 = P\dfrac{12}{t_1^3(t_1+t_2)(t_5+2t_4+2t_3+2t_1+t_2)} \\
K_2 = P\dfrac{12}{t_4^3(t_4+t_5)(t_5+2t_4+2t_3+2t_1+t_2)}
\end{cases}
$$

Given that t_1, t_2, t_3, t_4 and t_5 should all be greater than or equal to zero the following set of inequalities can be established:

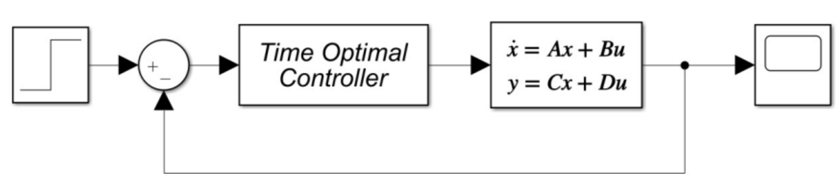
$$V > \frac{3A^2}{2J} \quad , \quad V > \frac{3D^2}{2I} \quad , \quad \frac{P}{V} > \frac{3A}{4J} + \frac{V}{2A} + \frac{3D}{4I} + \frac{V}{2D}$$

For back calculation and precise computation of the maximum accelerations and velocity we can use:

$$\begin{cases} J = P\dfrac{3}{t_1(t_1+t_2)(t_5+2t_4+2t_3+2t_1+t_2)} \\ I = P\dfrac{3}{t_4(t_4+t_5)(t_5+2t_4+2t_3+2t_1+t_2)} \\ A = P\dfrac{2}{(t_1+t_2)(t_5+2t_4+2t_3+2t_1+t_2)} \\ D = P\dfrac{2}{(t_4+t_5)(t_5+2t_4+2t_3+2t_1+t_2)} \\ V = P\dfrac{2}{t_5+2t_4+2t_3+2t_1+t_2} \end{cases}$$

S-Curve Profiles – A Closed From Solution

Up to now we have shown how to generate the desired profile limited by a set of constraints. These profiles are perfect when you have a fully causal system where the current time of the system is available, and so you can design the geometry of more complex profiles like the mentioned ST-Curve. But there may be situations that you do not have the current system time or you have to start the profile from ***any initial condition!*** In such cases it is not very easy to just calculate the solution from its geometry and we need a complete controller to just calculate the profile for us. Previously we have stated that if we could calculate the needed jerk at any sample time, we can integrate it to achieve the target velocity or position. Consider the following diagram:



Where the plant is a double or triple integrator system. In optimal control theory, it can be proved that such system can have a controller which transfers the system from any initial condition to X = $[0]_{n \times 1}$ final state in the minimum admissible time.

S-Curve Velocity Profile Generator – Double Integral Plant

It is well known in optimal control, the time optimal control law (TOC) for a double integral plant can be written as:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u, \qquad u = -J \times sign\left(x_1 - ref + \frac{x_2|x_2|}{2J}\right), \qquad where\!: |u| \le J$$

Prof. Jingqing Han, a control theorist and educator who inspired generations of students and colleagues in China, proved that the TOC law of the double integrator system in digital form can exactly be written like this (x_1 is velocity and x_2 is acceleration): $\left(u = fhan(x_1, x_2, r, h)\right)$

$$fhan(v_1, v_2, r, h) \rightarrow \begin{cases} d = rh^2, \quad a_0 = hv_2, \quad y = v_1 + a_0 \\ a_1 = \sqrt{d(d + 8|y|)} \\ a_2 = a_0 + \dfrac{a_1 - d}{2} \times sign(y) \\ s_y = \dfrac{sign(y + d) - sign(y - d)}{2} \\ a = s_y(a_0 + y - a_2) + a_2 \\ s_a = \dfrac{sign(a + d) - sign(a - d)}{2} \\ f = -rs_a\left(\dfrac{a}{d} - sign(a)\right) - r \times sign(a) \end{cases}, \quad where \begin{cases} r = J \ Maximum\ jerk \\ h = T_s \ Sampling\ time \\ v_1 = x_1 - ref \\ v_2 = x_2 \end{cases}$$

S-Curve Position Profile Generator – Triple Integral Plant

After a long time of research and practice in motion control, we had to also solve the TOC law for the triple integral plant as well, and the result is (x_1 is position and x_2 is velocity and x_3 is acceleration):

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u,$$

$$u = \begin{cases} -J \times sign(xcal), & |x_c| > eps \\ -J \times sign\left(x_2 + \dfrac{x_3|x_3|}{2J}\right), & |x_c| \le eps \end{cases}, \begin{cases} y_1 = ref - x_1 + \dfrac{x_3^3}{3J^2} + \dfrac{x_3|x_2|}{J} \\ y_3 = sign(x_2)(|x_2| + \dfrac{x_3^2}{2J}) \\ x_c = y_1 + y_3\sqrt{\dfrac{|y_{3|}}{J}} \end{cases}$$

With these time optimal control laws we can easily write a program to generate a motion profile from any initial condition to the desired state.

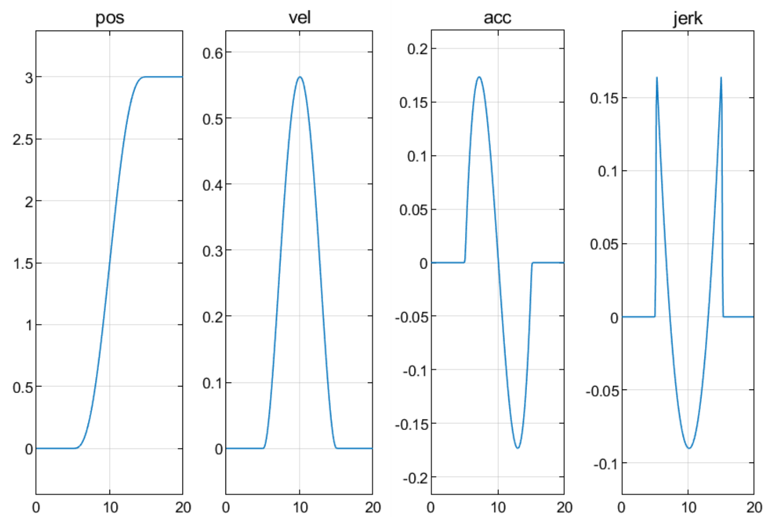
Minimum Jerk Trajectory

Among all the motion profiles, the minimum jerk trajectory is the easiest of them all to generate. It is proved in mathematics that for a trajectory to be in minimum jerk, its sixth derivative must be equal to zero. Hence from integrating and substituting the initial and final states, the whole profile model can be achieved. The function $x(t)$ represents the minimum jerk trajectory in one dimension. Hogan noted that, in general, if something you wanted to move something from location $x = x_i$ to $x = x_f$ in $t = d$ seconds, the minimum jerk trajectory would be:

$$x(t) = x_i + (x_f - x_i)\left(10\left(\frac{t}{d}\right)^3 - 15\left(\frac{t}{d}\right)^4 + 6\left(\frac{t}{d}\right)^5\right)$$

And the minimum jerk trajectory in two dimensions is:

$$\begin{cases} x(t) = x_i + (x_f - x_i)\left(10\left(\frac{t}{d}\right)^3 - 15\left(\frac{t}{d}\right)^4 + 6\left(\frac{t}{d}\right)^5\right) \\ y(t) = y_i + (y_f - y_i)\left(10\left(\frac{t}{d}\right)^3 - 15\left(\frac{t}{d}\right)^4 + 6\left(\frac{t}{d}\right)^5\right) \end{cases}$$

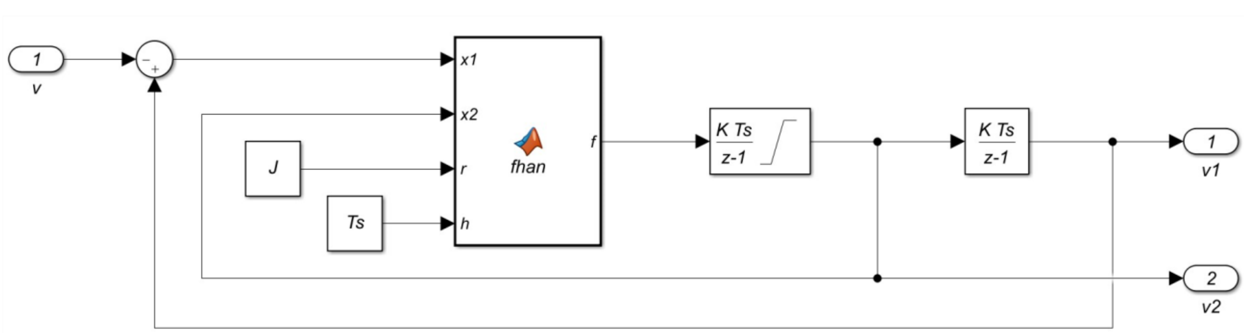


Examples

The usage of S-Curve and ST-Curve profiles is mostly in the whole human body movement such as elevators, cruise control systems, etc. The minimum Jerk trajectories are mostly used for human motor control like functions such as elbow and knee prostheses and robotic arms.

S-Curve Velocity Profile Generator – Closed Form

In MATLAB you only need to create a model like this:

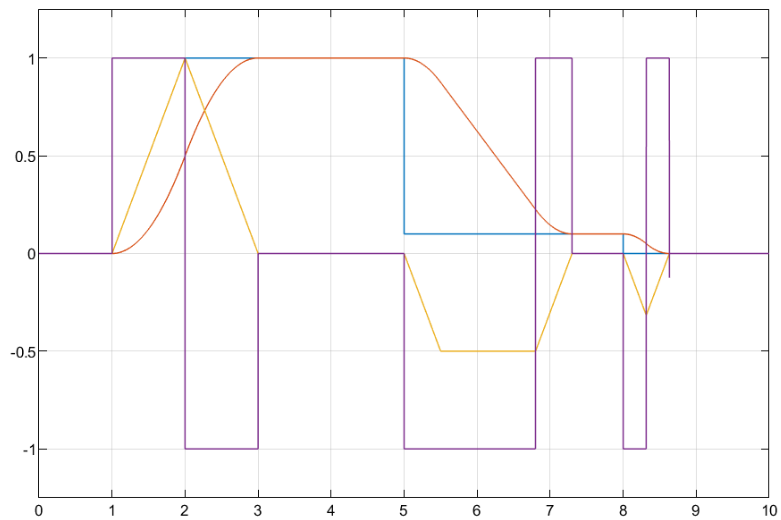


And write the following code in the MATLAB function block:

```
function f = fhan(x1, x2, r, h)

    d = r * h^2;
    a0 = h * x2;
    y = x1 + a0;
    a1 = sqrt(d * (d + 8 * abs(y)));
```

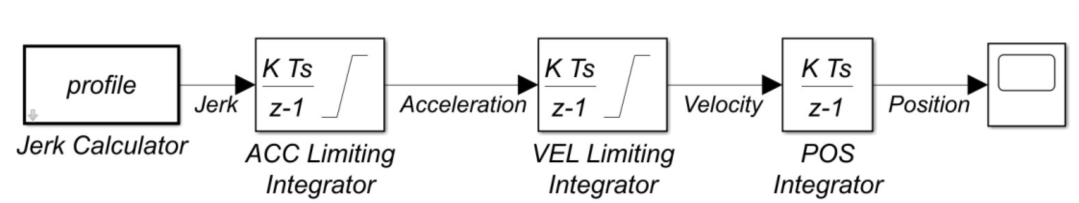
```
    a2 = a0 + sign(y) * (a1 - d) / 2;
    sy = (sign(y + d) - sign(y - d)) / 2;
    a = (a0 + y - a2) * sy + a2;
    sa = (sign(a + d) - sign(a - d)) / 2;
    f = -r * (a / d - sign(a)) * sa - r * sign(a);

end
```

Then set the variables Ts = 0.001, J = 1, A = 1 and D = 0.5 ("A" and "D" are the acceleration integrator output limits) and input a stair function with [0, 1, 0.1, 0] values at [0, 1, 5, 8] seconds. The result is:



S-Curve Position Profile Generator – Time Based Mode

Write the presented code in this section in a MATLAB function block and pass it through three integrators like this:

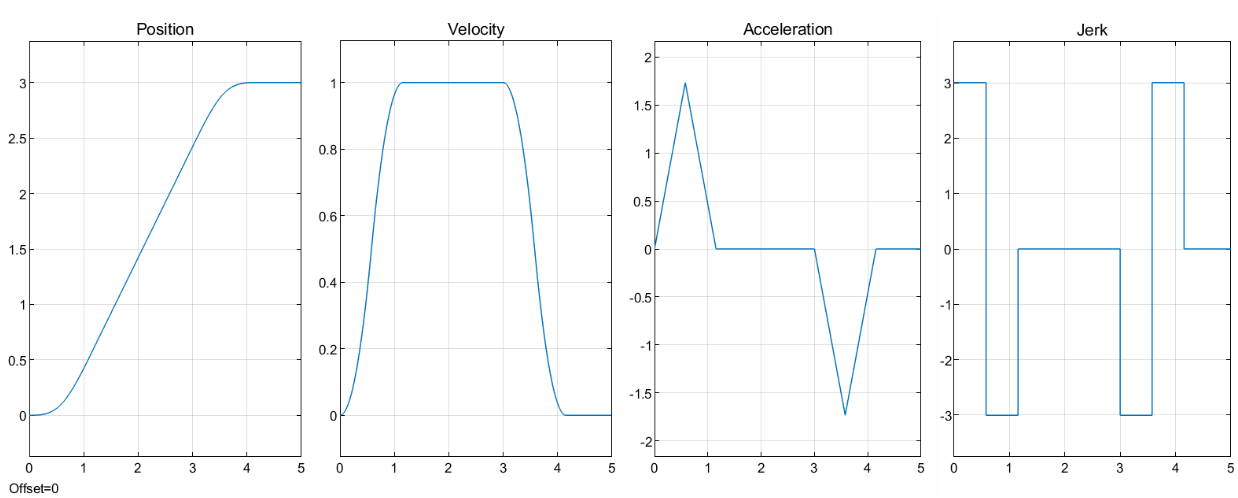


Set variables P = 3, V = 1, A = 2, J = 3, T_s = 0.001. The following code solves the time based mode in the simplest way where A = D and J = I.

```
function r = profile(P, V, A, J, Ts)

persistent t;
if isempty(t)
    t = 0;
else
    t = t + Ts;
end

if P/V < (A/J + V/A)
    b = A/J;
    V = A * (-b + sqrt(b^2 + 4*P/A)) / 2;
end
```

```
if V*J < A^2
    A = sqrt(J*V);
end

t1 = A/J;
t2 = V/A - A/J;
t3 = P/V - V/A - A/J;
t4 = t1;
t5 = t2;

t = round(t / Ts) * Ts;
t1 = round(t1 / Ts) * Ts;
t2 = round(t2 / Ts) * Ts;
t3 = round(t3 / Ts) * Ts;
t4 = round(t4 / Ts) * Ts;
t5 = round(t5 / Ts) * Ts;

K = 2*P/(t1*(t1+t2)*(t5+2*t4+2*t3+2*t1+t2));

if 0
elseif t < (t1)
    r = +K;
elseif t < (t1 + t2)
    r = 0;
elseif t < (2*t1 + t2)
    r = -K;
elseif t < (2*t1 + t2 + t3)
    r = 0;
elseif t < (2*t1 + t2 + t3 + t4)
    r = -K;
elseif t < (2*t1 + t2 + t3 + t4 + t5)
    r = 0;
elseif t < (2*t1 + t2 + t3 + 2*t4 + t5)
    r = K;
else
    r = 0;
end

end
```

The result will be:



Minimum Jerk Trajectory Generator

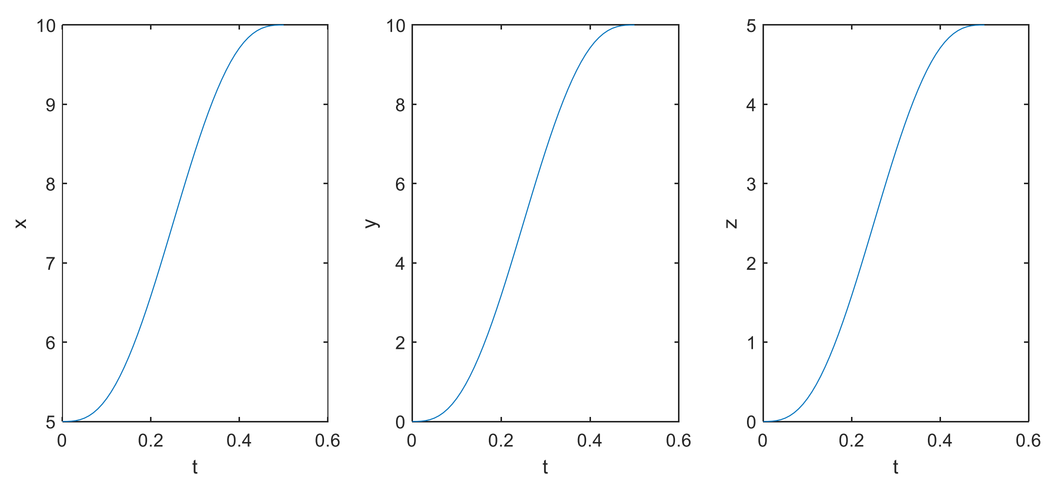
The following function generates a minimum jerk trajectory from xi to xf:

```
function output = min_jerk(xi, xf, t)
% Generate a minimum jerk trajectory from xi to xf.
% xi: starting position, 1x3 matrix
% xf: final position, 1x3 matrix
% t: the time vector, Nx1 matrix
% output: the generated trajectory, Nx3 matrix

d = t(end);
N = length(t);
a = repmat((xf - xi), N, 1);
b = repmat((10 * (t/d).^3 - 15 * (t/d).^4 + 6 * (t/d).^5)',1,3) ;
output = repmat(xi, N, 1) + a .* b;

end
```

And the following code is an example for how to use it:

```
i = 1;
t = 0:0.01:0.5;

out = min_jerk([5 0 0], [10 10 5], t);

subplot(3,1,1);
plot(t, out(:,1));
xlabel('t');
ylabel('x');

subplot(3,1,2);
plot(t, out(:,2));
xlabel('t');
ylabel('y');

subplot(3,1,3);
plot(t, out(:,3));
xlabel('t');
ylabel('z');
```

The result will be like this:



Summary

Please Note:

1) The presented solutions are the result of very complex mathematics and mentioning them in a tutorial is out of discussion. For more information please refer to the references.
2) There are many other types of trajectories in control systems for different purposes, but here we only study those which are perfect for ***human related motion control systems***. For other types of trajectories you could search for example minimum-snap trajectory which is widely used for differentially flat systems, such as quadrotor.
3) What presented here is the result of years of investigation, largely performed experimentally in computer simulations with the scientific spirit of daring imaginations, painstaking observations, careful generalization and abstraction, and truthful verifications of principles in real-world applications. It may help the newcomers to trajectory concept greatly if one abandons the initial "*How can this be right?*" attitude and, instead, run a few simulations and practical tests of the proposed solutions and observe the results. Perhaps the facts, or data, are more convincing than mere articulation of ideas.
4) The benefit of the proposed solutions is that they present the current acceleration and jerk in addition to the desired velocity or position, therefore these additiona values can also be used as feedforward

References:

[1] "From PID to ADRC", Jingqing Han

[2] "On Discrete Time Optimal Control: A Closed-Form Solution", Zhiqiang Gao

[3] "Motion Profile Generator", Gang Tian, LINESTREAM TECHNOLOGIES

[4] "A Minimum-Jerk Trajectory", R. Shadmehr and S. P. Wisey, Shadmehrlab Laboratory for Computational Motor Control