FARO HD Animation Engine Equations of Motion

The FARO HD Animation Engine is a Kinematics system that applies Newton's Laws of Motion.

Equations of Motion

Newton's Laws of Linear Motion under Uniform Acceleration

$s == linear\ distance\ or\ position\ along\ path$

$s_o == initial\ position\ or\ distance$

$v == linear\ velocity$

$v_o == initial\ linear\ velocity$

$a == constant\ linear\ acceleration$

$t == time$

$s = s_o + v_o t + \frac{1}{2}at^2$ (Eq. 1)

$s = s_o + \frac{1}{2}(v + v_o)t$ (Eq. 2)

$v = v_o + at$ (Eq. 3)

$v^2 = v_o^2 + 2a(s - s_o)$ (Eq. 4)

Or

$v^2 = v_o^2 + 2ad$ (Eq. 4.1)

Where *d* is the total distance travelled

The user specifies the path that the model will follow specifying specific points where the path will always go through and the known velocities and rotations are known.

Each path segment, between two corresponding vertices or nodes, has the following kinematic properties that the user must specify:

Initial Velocity v_o

Final Velocity v

Uniform Acceleration a

Path distance, determined by the user physically drawing the path, Initial Velocity and Final Velocity are the primary components of the system.

The user must always specify the initial velocity on the path segment PLUS either the final velocity or the acceleration.

The distance along the path segment is determined geometrically by the path that the user drew.

So if the user specifies the Final Velocity, the Acceleration can be determined using (Eq. 4.1)

$$v^2 = v_o^2 + 2 * a * d$$

Solve for Acceleration

$$a = \frac{v^2 - v_o^2}{2 * d}$$

If the user specifies Acceleration instead, then the Final Velocity is then calculated using (Eq. 4.1) as well

$$v^2 = v_o^2 + 2 * a * d$$

With the restriction that the Final Velocity must be >= 0.

Total time along the segment is then calculated using (Eq. 2)

$$s = s_o + \frac{1}{2}(v + v_o)t$$

Solve for t, and substitute $d = (s - s_o)$

$$t = 2 * \frac{d}{v_o + v}$$

We know now all the required kinematic values.

When the animation system is running in real-time, time becomes the controlling factor.

The ‘VCR’ controller passes the current time into the animation engine.

This then determines the position along the path, and “instantaneous” velocity on the path at that point.

To determine the Position (s) along the path at a specific time, t, we simply use (Eq. 1)

$$s = s_o + v_o t + \frac{1}{2}at^2$$

To calculate the velocity at time t, we use the distance travelled at time t calculated in the previous step and input it into (Eq. 4.1)

$$v^2 = v_o^2 + 2ad$$

Rotation along the path segment is calculated as follows:

Each path segment has an initial and final value for each of the 3 available rotations (yaw, pitch and roll)

Each rotation variable is calculated in the same manner, but independently of the other.

First the distance, d, along the path is first determined as in the above steps.

Then the rotation is calculated as a percentage of distance travelled along the path times the total amount of rotation.

$\theta_o == Initial\ Rotation$

$\theta_f == Final\ Rotation$

$\Delta\theta = Total\ Rotation$

$\Delta d = Total\ Distance\ Along\ Path$

$d = distance\ travelled\ along\ path$

$\Delta\theta = \theta_f - \theta_o$

$$\mu = \frac{d}{\Delta d}$$

$\theta = \theta_o + \mu * \Delta\theta$