On the Possibility of Using Whip-Action to Accelerate a Space Launch Vehicle

Gerry Flanagan
Alna Space Program

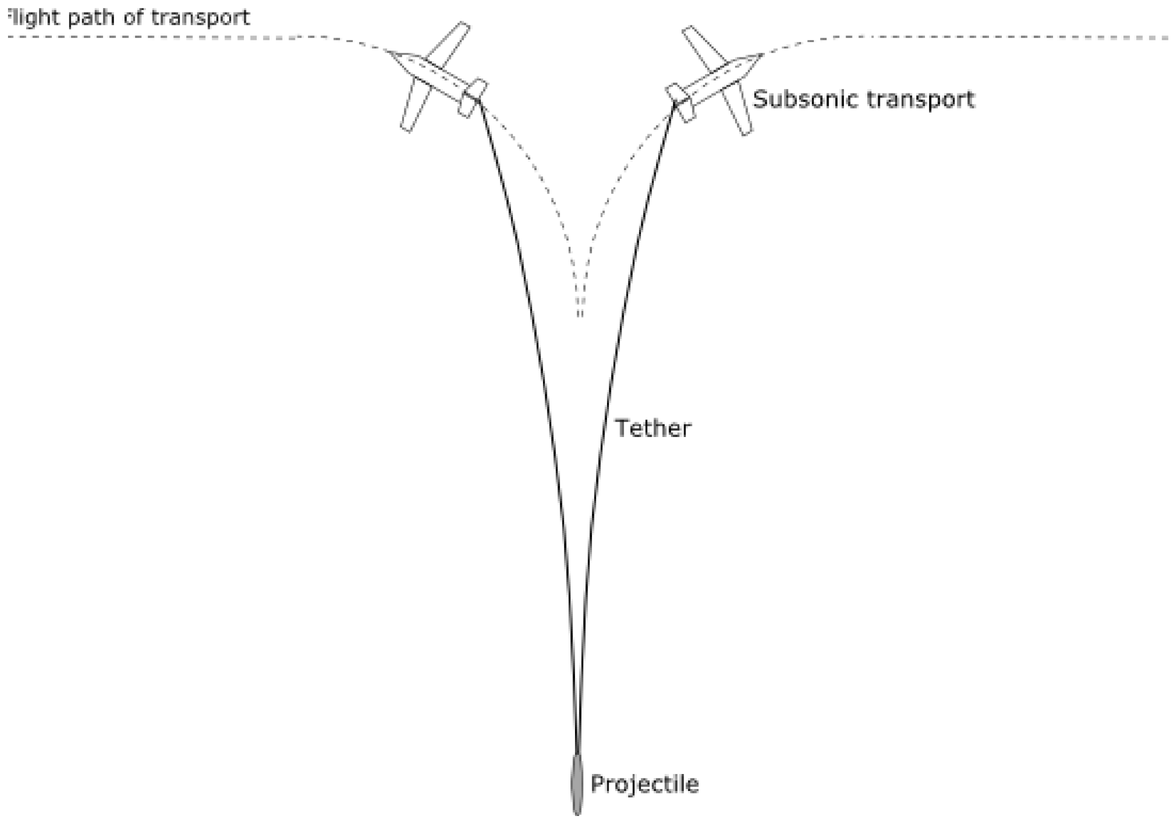
July 7, 2012, Revised July 16, 2012

Introduction

This study uses a dynamic model of a chain to determine if the whipping action of a tether can be used to accelerate a vehicle to a meaningful fraction of orbital velocity. A possible scenario is shown in Figure 1. The arrangement uses two subsonic transports that are much more massive than the projectile. A pair of long tethers connect the transports to the projectile. Initially, the transports fly parallel to each other. At the beginning of the sequence, the transports begin to turn away from each other and go along paths that are in opposite directions. The connecting tether is pulled taut, accelerating the projectile to a high velocity. At the maximum projectile, the projectile is released from the tethers, and the tethers separate. The overall acceleration effect is similar to the tip of a whip. The results will show that high accelerations are involved, so the method would only apply to unmanned payloads.

The simulation uses a dynamic chain model written as a *Mathematica* package. A description of the model and it's use is contained in the notebook chain_dynamic _model _guide.nb.

In[1]:=



Out[1]=

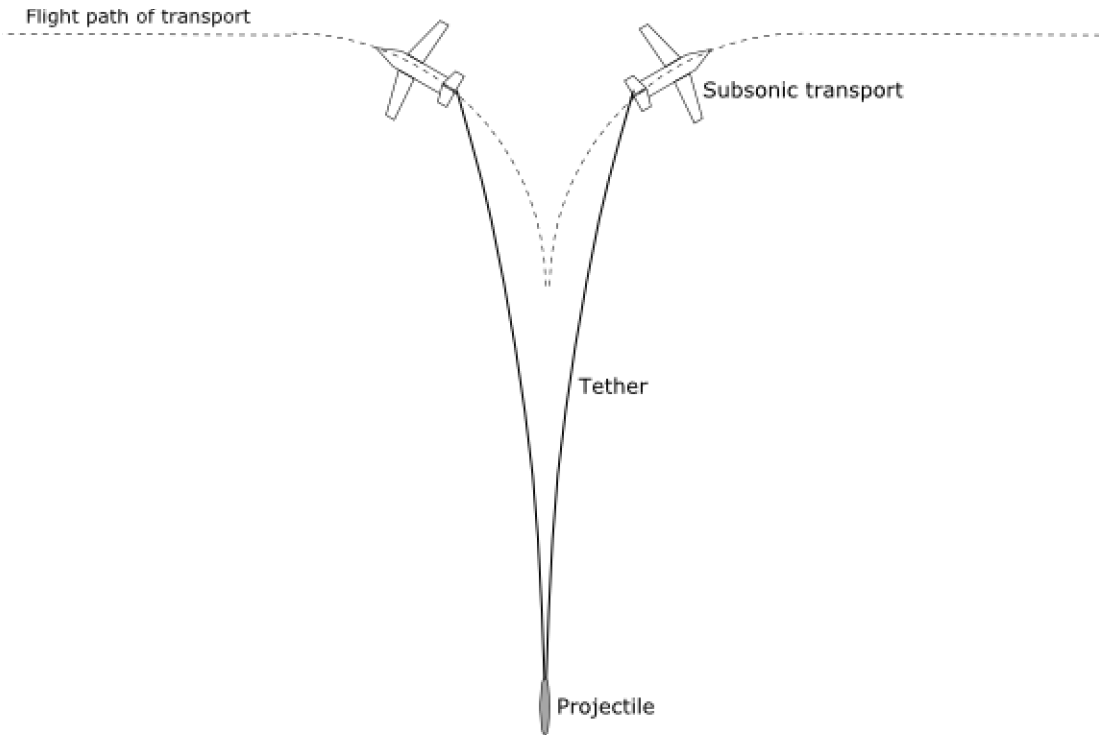
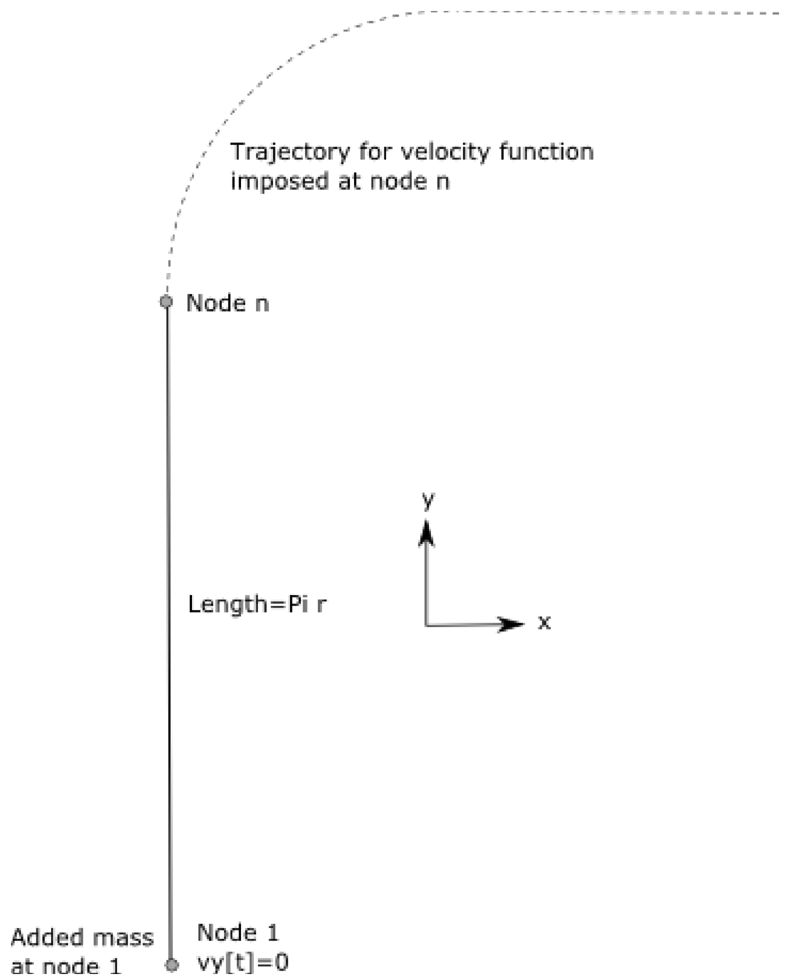


Figure 1. Elements of Projectile Accelerator

A simplified model is used to evaluate the proposed scenario. A single tether is modeled, with symmetry imposed by preventing any y-direction motion of the projectile. The driving motion of the transports is simulated by specifying a velocity history on the end-node of the tether. The velocity history initially follows a uniform arc, followed by a straight path in the x-direction after a 90 degree turn. The magnitude of the velocity is constant, implying that the transports are massive enough, or have sufficient power to not slow down during the maneuver. The simulation does not model the projectile release; we simply look for the point of maximum velocity of the projectile.

In[2]:=



Out[2]=

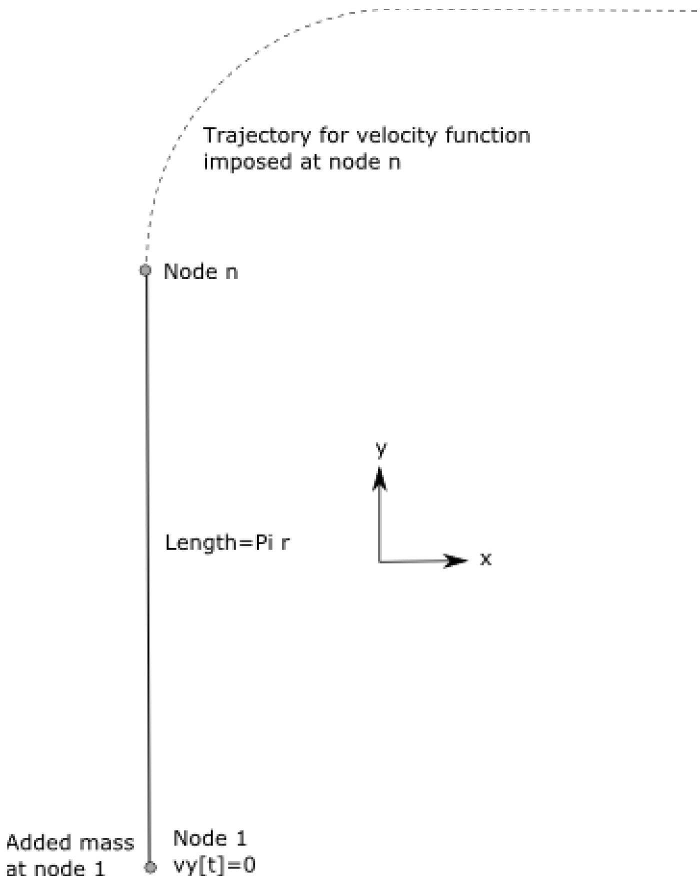


Figure 2 Model Idealization

Example Cases

Initialize the modeling program and override the animation display rate.

In[3]:=
```
<< spaceLaunch`cableDynamicModel`
SetOptions[ListAnimate, AnimationRate → 2];
```

Case 1. Lightweight Projectile

Assume that the initial velocity of the system is 300 m/sec (high subsonic speed)

In[5]:=
```
v0 = 300;
```

Also assume that the transport can make a 2g turn. The turn radius is then

In[6]:=
```
r = v0^2 / (2 × 9.81)
```

Out[6]=
```
4587.16
```

The tether cross sectional area is 10^{-3}m, equivalent to a diameter of 3.5 cm. The initial tension in the tether is zero. Aerodynamic drag is ignored.

In[7]:=
```
A = 10^-3;
tension = 0;
```

The tether material is graphite fiber with a Young's modulus of 400 GPa and a density of 1750 kg/m^3. The stiffness and linear density are therefore

```
In[9]:= EA = 400 × 10^9 * A;
        rho = 1750 * A;
```

The initial path of the tether is simply a straight line in the y-direction, and the initial velocity is uniformly equal to v0. The tether length is assumed to be Pi r. These functions define the initial path and velocity of the tether. s is a parameter that has the range 0≤s≤1.

```
In[11]:= path = Function[{s}, {0, Pi r s}];
         vel = Function[{s}, {0, v0}];
```

Create a model with 50 modes. The projectile is modeled as a point mass of 500 kg at node 1. Because of the symmetry imposed on the model, the total projectile mass is 1000 kg.

```
In[13]:= model = pathSubdivide[path, vel, tension, EA, rho, 50, addPointMass → {{1, 500}}]

Out[13]= --Chain model data --
```

ω is the rotational velocity of the transports as they make their turn.

```
In[14]:= ω = v0 / r;
```

Define the velocity functions for the transports. The time to go around the quarter circle is Pi r/(2 v0).

```
In[15]:= vx = Function[{t}, Piecewise[{{v0 Sin[ω t], t < Pi r / v0 / 2}, {v0, True}}]];
         vy = Function[{t}, Piecewise[{{v0 Cos[ω t], t < Pi r / v0 / 2}, {0, True}}]];
```

The simulation time is at least the tether length (Pi r) divided by the velocity v0. After some trial runs, an additional factor of 1.4 was added to insure the simulation goes to the point of maximum velocity.

```
In[17]:= maxTime = 1.4 Pi r / v0 // N

Out[17]= 67.2512
```

For convenience, define a function that returns zero for all s.

```
In[18]:= zero = Function[{s}, 0];
```

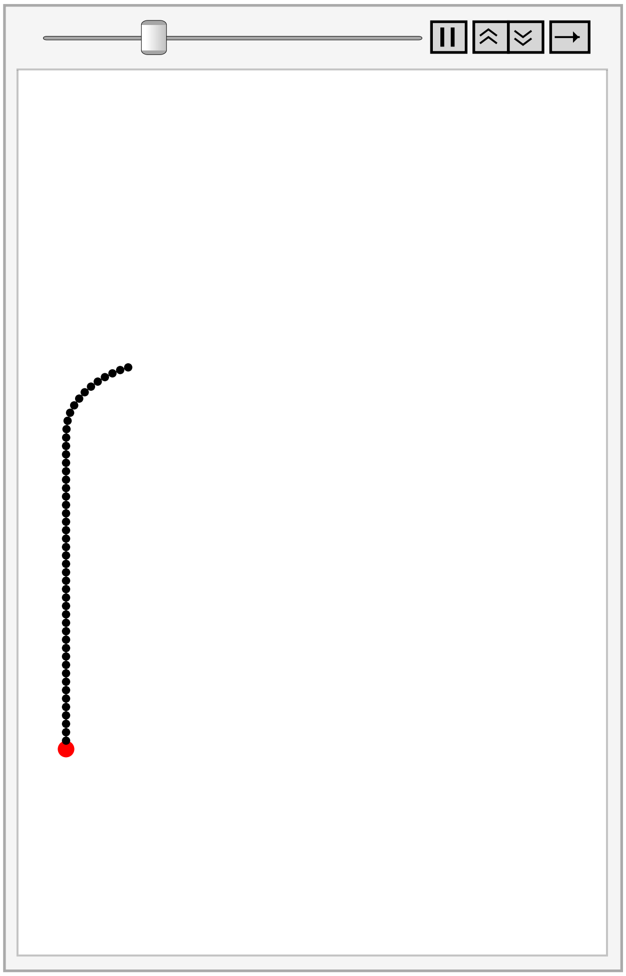
Run the simulation. Trial runs indicated that the maximum number of steps had to be increased from the default value.

```
In[19]:= h = chainHistory[model, .01, maxTime, {{{50, 1}, vx}, {{50, 2}, vy}, {{1, 1}, zero}},
           maxNumberSteps → 3000];
```

Show the time history of the tether trajectory. The red disk represents the payload.

In[20]:=
```
plotChain[h, PlotRange → Automatic, diskSizeRatio → .005, highlightNode1 → True]
```

Out[20]=



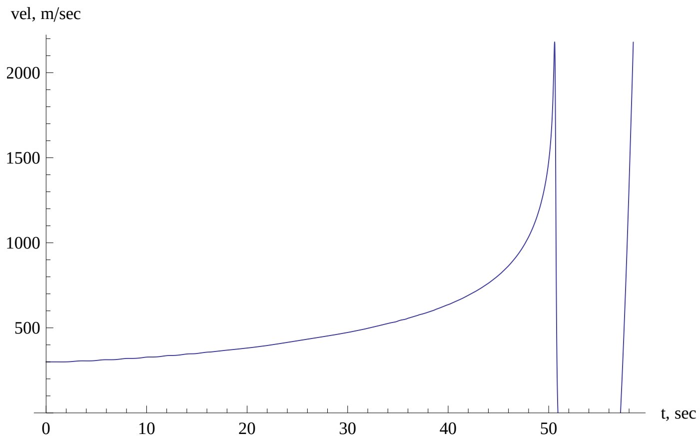
Velocity is the derivative of the trajectory.

In[21]:=
```
vh = h';
```

Plot the velocity of the projectile (node 1) over time. Only the results up to the first peak are meaningful because the projectile would be released at the peak.

In[22]:=
```
Plot[vh[t][[1, 2]], {t, 0, maxTime}, PlotRange → {All, {0, Automatic}},
 AxesLabel → {"t, sec", "vel, m/sec"}]
```

Out[22]=



Find the time at which the first peak occurs. The FindMaximum function requires that we create a new input function that is the velocity of the projectile, extracted from the total list of results. From looking at the plot, we know that the first peak occurs before 52 sec. FindMaximum will issue warning messages because the algorithm has problems achieving the default accuracy and precision with an input function with such a sharp peak.

In[23]:=
```
vm[t_?NumberQ] := vh[t][[1, 2]]
vmax = FindMaximum[{vm[t], t < 52}, {t, 30}]
```

Out[24]=
```
{2185.4, {t → 50.593}}
```

Define timeMaxVel as the time of maximum velocity.

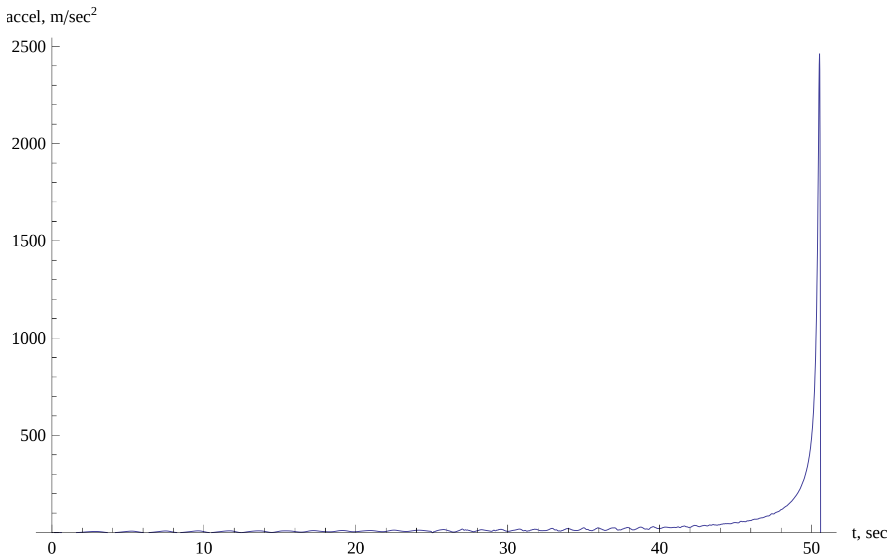
In[35]:=
```
timeMaxVel = t /. vmax[[2]]
```

Out[35]=
```
50.593
```

Compute the acceleration of the nodes. Plot the acceleration of the projectile.

In[26]:=
```
ah = vh';
Plot[ah[t][[1, 2]], {t, 0, timeMaxVel}, PlotRange → {All, {0, All}},
 AxesLabel → {"t, sec", "accel, m/sec²"}]
```

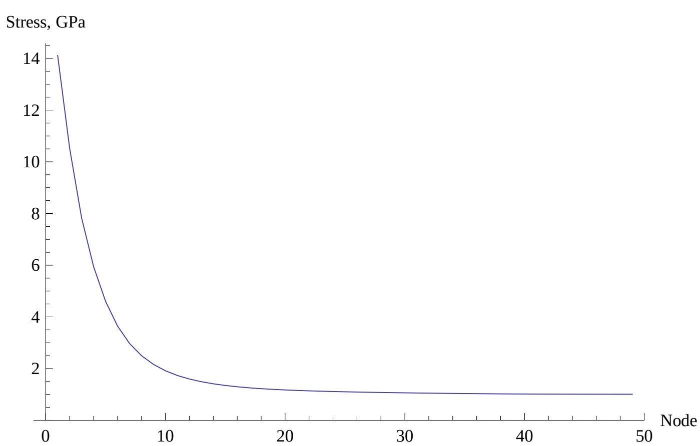
Out[27]=



Plot the tether stress distribution (in GPa) at timeMaxVel. The material strength of graphite fiber is on the order of 3 GPa. The result shows that a tether using current materials cannot survive. The horizontal axis is node number, which is proportional to the length.

In[28]:=
```
ListPlot[segmentTension[h, model][timeMaxVel] / 10^9 / A, PlotRange → {All, {0, All}},
 Joined → True, AxesLabel → {"Node", "Stress, GPa"}]
```

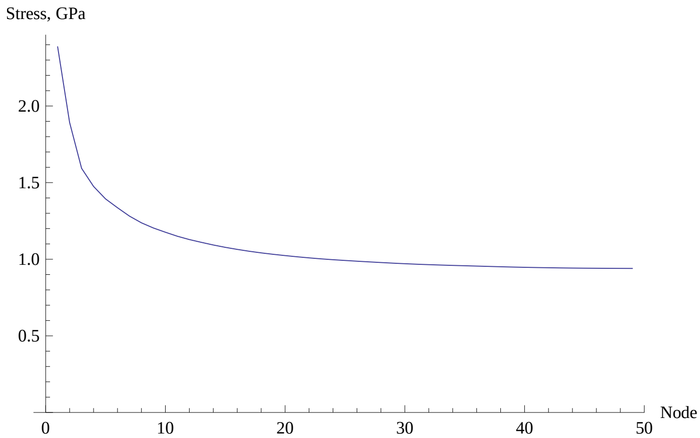
Out[28]=



To make the concept work, we can determine a time before timeMaxVel where the tether stress might be achievable. The projectile separation and tether release would occur before timeMaxVel, hopefully preventing failure. For example, 0.997 timeMaxVel gives the following stress distribution.

In[41]:=
```
ListPlot[segmentTension[h, model][.997 timeMaxVel] / 10^9 / A,
 PlotRange → {All, {0, All}}, Joined → True, AxesLabel → {"Node", "Stress, GPa"}]
```

Out[41]=



At the 0.997 timeMaxVel, the projectile velocity.

In[42]:=
```
vm[.997 timeMaxVel]
```

Out[42]= 1879.47

And the maximum acceleration in g's would be

In[45]:=
```
ah[0.997 timeMaxVel][[1, 2]] / 9.81
```

Out[45]= 189.647

We are assuming that releasing the projectile and separating the tethers will prevent the stress from continuing to climb. However, it is quite possible that the stress waves in the tether will still exceed the tensile strength. To evaluate, one should create a new model using the tether path, velocity, and stress state at separation as the input to a new model in which the mass are symmetry condition are removed. This follow-on model has not been done yet.

As a final note, the mass of a single tether is

In[32]:=
```
rho * Pi r
```

Out[32]= 25 219.2

This mass should be manageable with a 747 class transport.

Case 2: Heavy Projectile

There are many input parameters that could be varied in the model to find a better match between high release velocity and acceptable stress. In the previous case, the relationship between tether area and projectile mass was arbitrary. As a quick comparison, we will increase the projectile by a factor of ten.

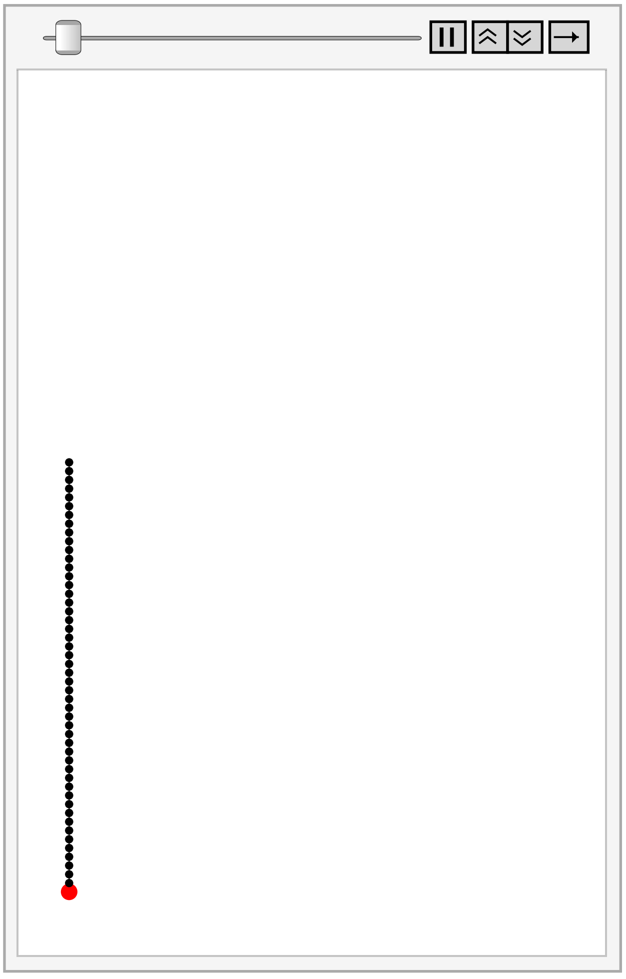
In[46]:=
```
model = pathSubdivide[path, vel, tension, EA, rho, 50, addPointMass → {{1, 5000}}]
```

Out[46]= --Chain model data --

In[47]:=
```
h = chainHistory[model, .01, maxTime, {{{50, 1}, vx}, {{50, 2}, vy}, {{1, 1}, zero}},
   maxNumberSteps → 3000];
```

In[48]:=
```
plotChain[h, PlotRange → Automatic, diskSizeRatio → .005, highlightNode1 → True]
```

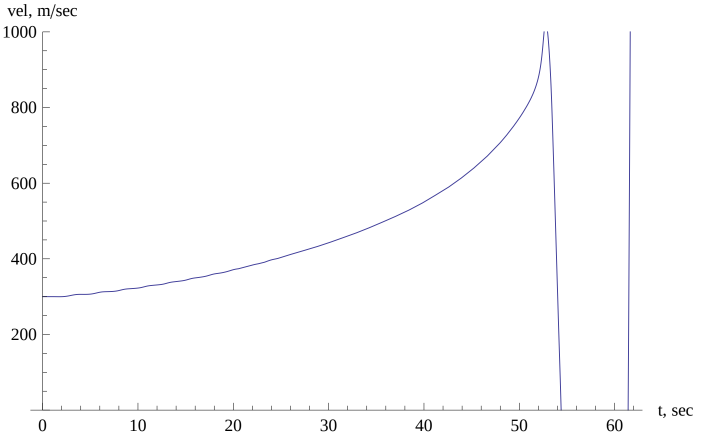
Out[48]=



Velocity of projectile over time.

In[49]:=
```
vh = h';
Plot[vh[t][[1, 2]], {t, 0, maxTime}, PlotRange → {All, {0, 1000}},
 AxesLabel → {"t, sec", "vel, m/sec"}]
```

Out[50]=



Compute the time at which the maximum velocity occurs. The output from FindMaximum also shows the peak velocity.

```
In[51]:= vm[t_?NumberQ] := vh[t][[1, 2]];
max = FindMaximum[{vm[t], t < 55}, {t, 35}, AccuracyGoal → 2]

Out[52]= {1024.86, {t → 52.7686}}
```

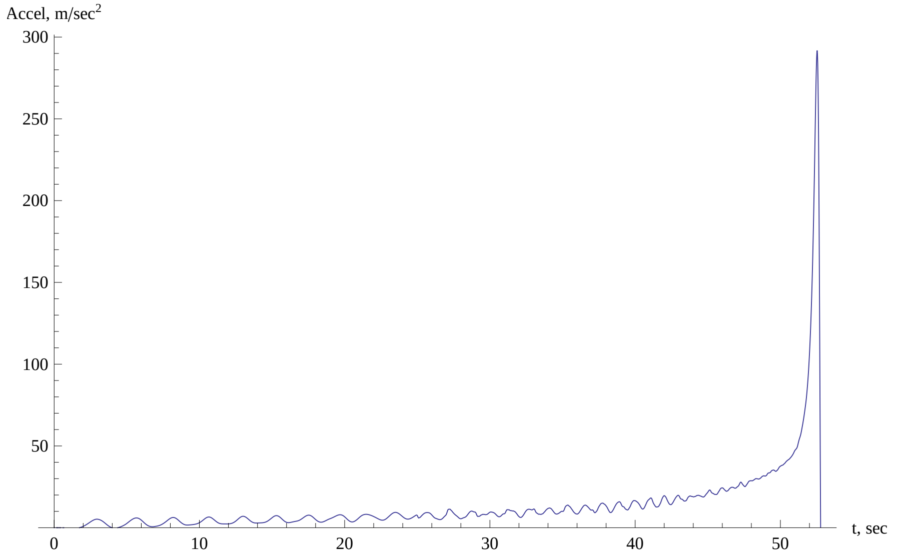
```
In[53]:= timeMaxVel = t /. max[[2]]

Out[53]= 52.7686
```

Acceleration of projectile over time.

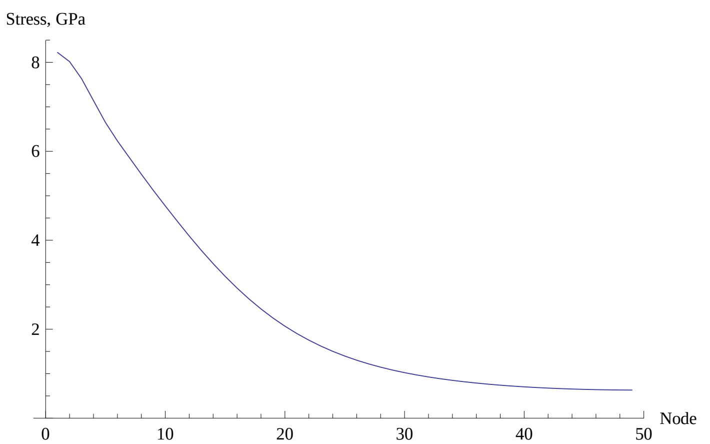
```
In[54]:= ah = vh';
Plot[ah[t][[1, 2]], {t, 0, timeMaxVel}, PlotRange → {All, {0, All}},
 AxesLabel → {"t, sec", "Accel, m/sec^2"}]
```

Out[55]=




```
In[56]:= ListPlot[segmentTension[h, model][timeMaxVel] / 10^9 / A, PlotRange → {All, {0, All}},
 Joined → True, AxesLabel → {"Node", "Stress, GPa"}]
```

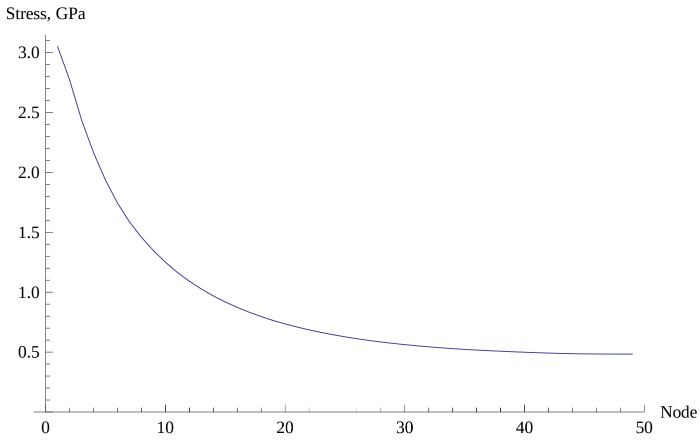
Out[56]=



For this case, the stress would be limited to the acceptable range by releasing the projectile at 0.994 timeMaxVel. Show the stress distribution at that time.

```
In[64]:= ListPlot[segmentTension[h, model][.994 timeMaxVel] / 10^9 / A,
          PlotRange → {All, {0, All}}, Joined → True, AxesLabel → {"Node", "Stress, GPa"}]
```

Out[64]=



The velocity at the release time.

```
In[65]:= vm[.994 timeMaxVel]

Out[65]= 957.925
```

Using these parameters, we get 50% of the peak velocity with 10 times the payload. That may represent a more effective use of all the equipment and resources involved.

Conclusion

The analysis shows that it is possible to magnify the speed of subsonic transport by a factor of 3-6 using the whip action of a long tether. We cannot evaluate whether this degree of magnification would justify the cost and complexity of the proposed system. In this study, there was little attempt to optimize the input parameter, so better performance ratios may well be possible. One possible variation is to use a tapered tether, as in a traditional whip. The taper could lead to higher tip velocities, although intuitively it would seem to exacerbate the tensile stress problem.

The big missing element in this simulation is air drag. The drag of the tether would increase dramatically as portions of the tether start to move supersonically. This factor alone could kill any prospects for the system.