Subject: Different ways to model tractive effort
Posted by Laura Isenhoefer on Tue, 26 Feb 2019 08:40:49 GMT
View Forum Message <> Reply to Message

Hi,

Jernbanedirektoratet has the need to define the tractive effort of an engine in a more flexible way than it is possible right now. Our aim is to cater for all the needs that our different tools have and ideally allow for a lossless transfer from one railML-file to each of the tools.

In general, our tools seem to use 3 different approaches:

1) Discrete value table: same as railML-value table. Each pair of speed and tractive effort get one entry, values between the given value pairs need to be interpolated (linear). Accuracy is user-defined.

2) Hyperbolic curves: the curve of the tractive effort curve is defined by a hyperbola. All you need to know are the coordinates of the start and the end point of the hyperbola and with the the equasion F=P/v (+c) you will be able to interpolate every point on the hyperbola. Additionally to the given value pairs there's the need to specify if those points should be connected linear or hyperbolic, which can currently not be done in railML. (But could probably be done easily with a simple extension).

3) Quadratic curves: The tractive effort can also be given by the following equasion:

for both the linear part as well as the curve, by giving b0, b1 and b2 (for different intervals). This could e.g. be implemented by using different z-values in the railML-value table to define the bi for the different speed-intervals.

As mentioned above, we would love to find a solution that allows all 3 possibilities, so that we are able to enter the tractive effort into all of our tools we use.
Mathml does not seem to be the solution here, since it does not seem to be able to unambiguosly define those equasions or tables.

One of our suggestions would be to have a table with 6 columns, so that each reading system can pick the values it needs:
(speed | tractive effort | linear/hyperbolic? | b0 | b1 | b2)

We're happy to hear other suggestions. The solution could first be a Norwegian extension and later be implemented into railML2.5.

Best regards,
Laura