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

Subject: Re: Different ways to model tractive effort
Posted by Joerg von Lingen on Tue, 26 Feb 2019 09:53:43 GMT

Hi,

in our old tools we a mechanism to describe such curves per (speed) interval just marking the type and possible coefficents.

1) Polynom $F=A+B*v+C*v^2+D*v^3$ - by choosing the coefficents A, B, C and D one can describe a wide range of different
curves from just konstant to more complex ones

2) Hyperbolic $F=A/v$ - hyperbolic curve is used for intervals with constant power (A) diveded by speed

3) Quadratic $F=(A*v1)/v^2$ - quadratic curve is used for intervals (start at v1) with field weakening from power (A)

This principles I had in mind when I presented in Nov. 2003 a possible representation in MathML - refer attachment.

Regards,
Jörg von Lingen, Rollingstock coordinator

Laura Isenhoefer wrote on 26.02.2019 09:40:
> 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

> This equasion allows to precisely define the tractive effort
> 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 unambiguously 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
>

File Attachments

1) railML-MathML.pptx, downloaded 26 times

Subject: Re: Different ways to model tractive effort
Posted by Thomas Nygreen on Tue, 05 Mar 2019 13:20:27 GMT
View Forum Message <> Reply to Message

Dear all,

The current railML2 valueTable could support any of the segmented functions listed by Laura and Jörg, if we for each row apply the formula
F = Sum (y_z * v^z) for all z
where each value for z is given by columnHeader@zValue.

If no column header is found and only one column is given, we would assume z = 0, meaning that F = y. This allows programs to keep listing the tractive effort for small speed steps.

This approach would support any polynomial function, such as constant (only z=0), linear (0 and 1), quadratic (0, 1, 2) and cubic (0, 1, 2, 3), the simple hyperbolic (-1, 0) and quadratic hyperbolic (-2) listed by Laura and Jörg, and other simple rational functions where there is no shift of the x variable.
