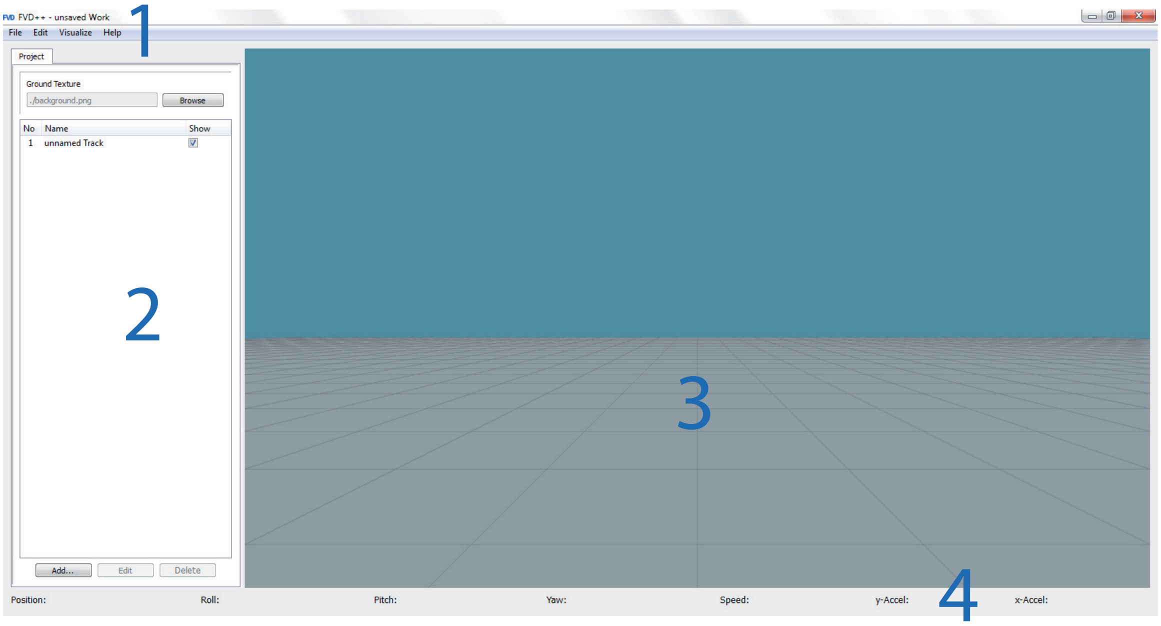
FVD++ Documentation

Table of Contents

1 User Interface and Configuration

1.1 Basic Layout

After having started the program, the following window will appear, which is splitted up into 4 different areas:



On the top (1), the main menu allows basic operations and settings concerning the whole program (see **Main Menu**).

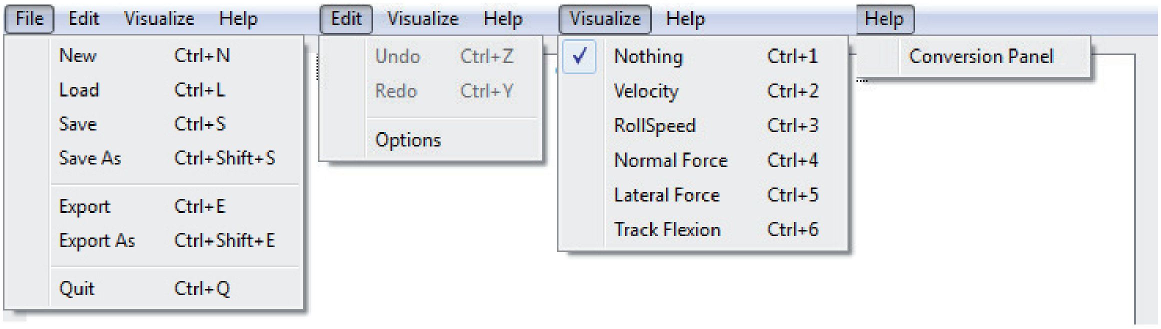
The sidebar (2) shows various information depending on which part of the project is being worked on. Right now, only project-wide settings can be seen. FVD++ uses tabulators (short: tabs) to organise different areas within the program and to show their relation to each other (see **Project Tab**).

The viewport (3) will show a 3D-rendered model of the coaster. In version 0.5 beta orthogonal views aren't implemented yet, which will likely be done in later versions. More on moving the camera and the POV mode see **Viewport and Camera Controls**.

Lastly, the statistics panel (4) which shows all kinds of different values which refer to the end point of the section being currently selected in the sections list or the current position if the POV mode is active (see **Statistics Panel**).

1.2 Main Menu

At the very top of the program window there's the main menu which consists of the following sub-menus:



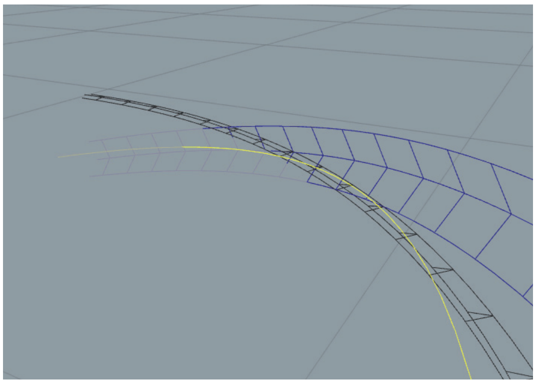
In *File*, the project files (extension: *.fvd) can be loaded, saved and transfered to NoLimits by using the export window (see **Exporting**). *Export As* will do that on every click, whereas *Export* will just save the nlelem-file with the exact same settings which were used in the last exporting action without asking.

The *Edit* panel allows to undo and redo all actions. Clicking on *Options* will open up a new window to define the color settings, the measure and the OpenGL version in case of problems with the program or when working on a Mac (see **Options Window**).

The visualisations have the effect of coloring the track dependently on the attribute being chosen to be kept in check. Having it set to *Nothing* will result in track models without any visualisations, so each of them has the defined colors (see **Track Options**).

The *Conversion* Panel in *Help* stores heartline values and performs measure conversions (see **Conversions Panel**).

1.3 Viewport and Camera Controls



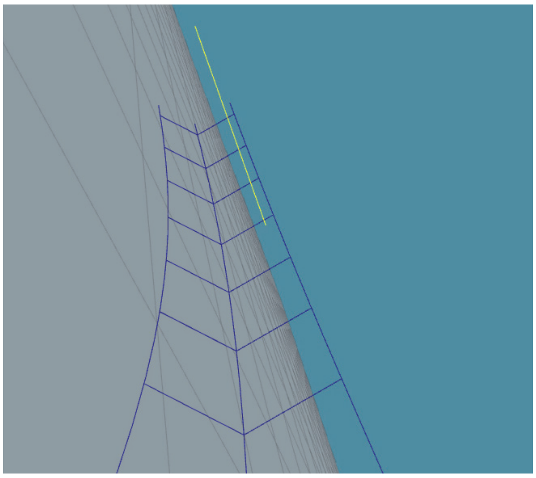
Right next to the section list there is the viewport which shows a simplified model of the coaster inside a virtual room. Unlike in NoLimits, the track can exist beneath the terrain level. The grid consists of squares of 10 by 10 meters.

As being configurable in the track options, the track has different colors, even if the visualisiations are turned off. As demonstrated in the screenshot, any track existing in negative heights is set to be displayed differently than normal. The part of the track which is effected by the currently selected transition is displayed green by default (configurable in the **Track Options**).

All of the following controls demand a right-click inside the viewport so that the mouse pointer disappears. Now the camera can be turned into any direction by moving the mouse. Flying forwards/backwards is done by pressing W or S, paning to the side by pressing A or D.

Entering or leaving the POV mode (camera follows the track) is done by pressing the space bar (again provided the mouse pointer isn't displayed).

The speed the camera has while riding the coaster in the POV mode is the calculated speed of the coaster which allows in-program checking of the smoothness and pacing of the track. For some section types, a constant speed can be set up which then will also be used in the POV mode. Using a constant speed for the whole track is possible as well.



In each one of the two viewing modes (normal, POV), the speed can be either halved by pressing Ctrl or doubled by pressing Shift while moving.

1.4 Statistics Panel

This panel shows different values refering to the endpoint of the currently selected section or the current position in the POV mode.

X: 18.56m Y: 9.39m Z: 1.18m Roll: -51.41° (-50.00°/s) Pitch: 39.64° (+36.46°/s) Yaw: -29.04° (-100.39°/s)

The first half gives information about the position and the direction of the end point. After each angle there's an additional number in parantheses which is the current change rate of that angle (see **Roll Speed**).

Speed: 17.01m/s y-Accel: 3.00g x-Accel: 0.00g

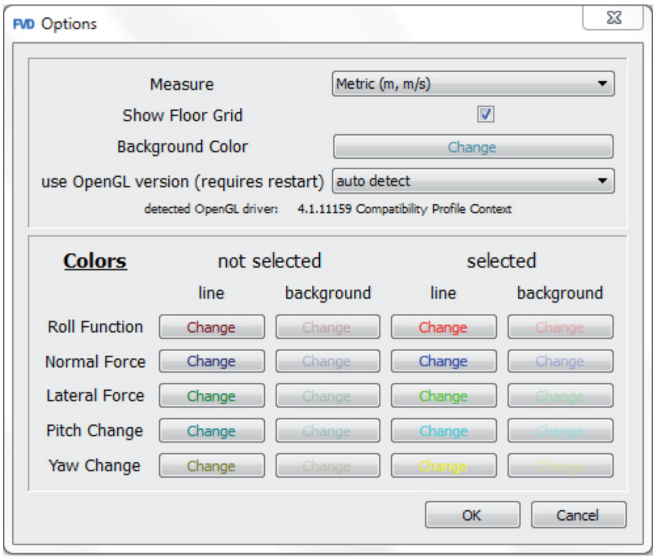
The second half shows the speed and the forces in both vertical and horizontal direction.

Speed: nanm/s y-Accel: nang x-Accel: nang

If any of these values show “nan” (not a number) followed by the unit, the speed to complete the selected section might has already reached zero before the end. As a consequence, any following track can’t be rendered and the exported track might be unusable.

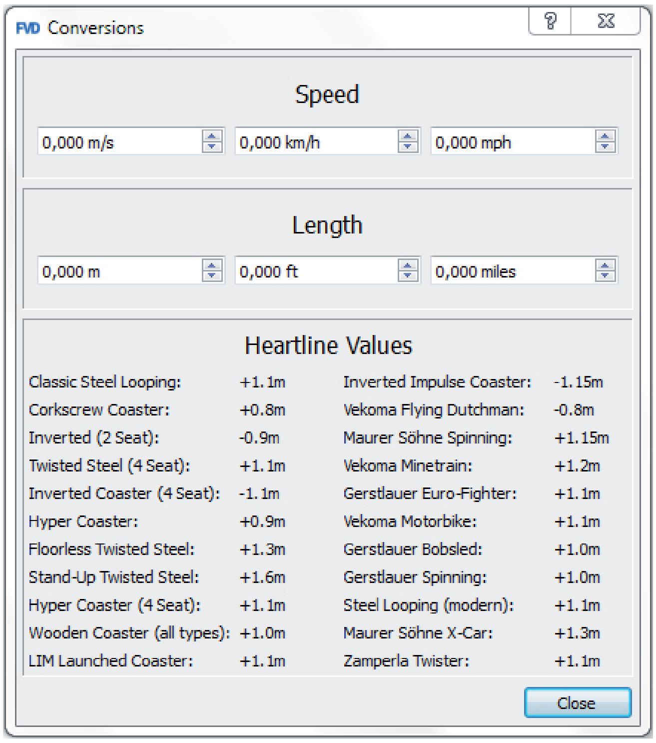
1.5 Options Window

Accessible in the main menu (*Edit*), the *Options* window allows further configuration of the measure, the OpenGL version (for running FVD++ on a Mac) and the background color. The floor grid can be deactivated here. The second half allows further configuration of the graph line and background colors in both selected and not selected state.



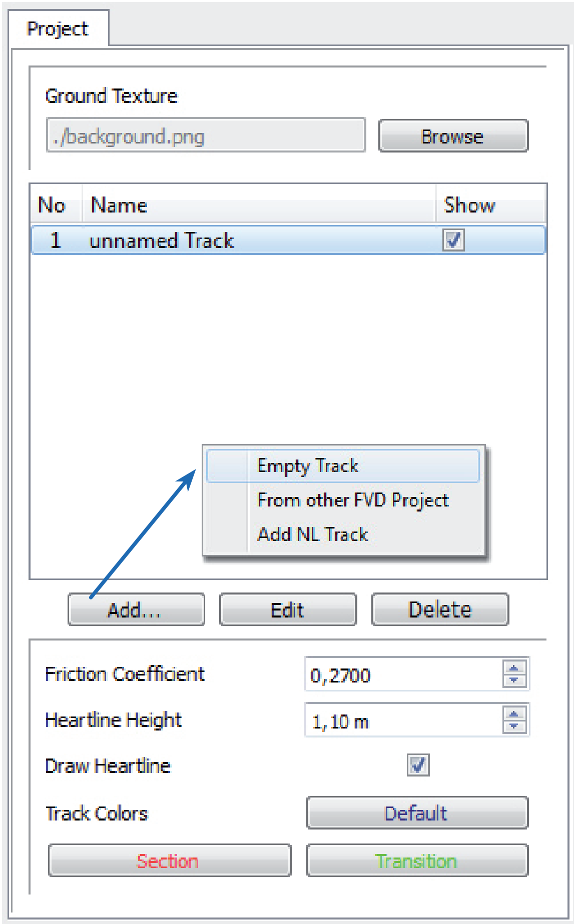
1.6 Conversions Panel

This window allows speed and length conversions and stores the heartline values of every NoLimits coaster type. The heartline value for a certain track can be set in the **Track Options**. The window can be opened by using the sub-menu *Help* in the main menu.



2 Project Tab

2.1 Ground Texture



Here, a png file can be selected which will be showed as the ground texture inside the viewport with the grid lines still existent. Only png files being in the subdirectory of FVD++ can be chosen. The picture will be stretched to fit in the building area which spans 440 by 440 meters.

2.2 Track List

FVD++ supports multiple tracks in one project file (*.fvd), whereas each individual track can be hidden by deactivating the *Show* clickbox and can be renamed by double-clicking on the name itself. Below the list, several buttons allow the organisation of the tracks: *Edit* will open up the next tab (see **Track Tab**). *Delete* will remove the track from the project.

The button *Add...* allows a few possible actions:

- Insert an empty track in the project.
- Insert a FVD++ track from another project file.
- Insert a NoLimits track as a visual guide.

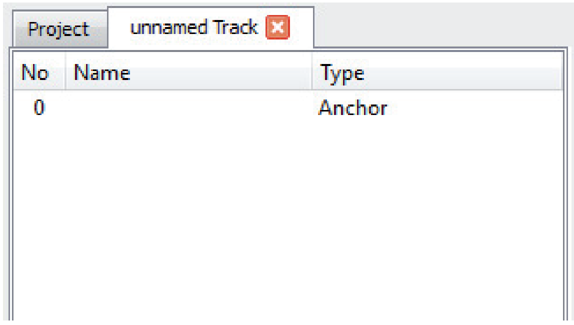
2.3 Track Options

The track options below the track list offer several settings concerning the currently selected track. *Friction Coefficient* controls how strong the friction will slow down the train on its course. To convert friction coefficients from newton to FVD++, multiplying by 9.81 is a good start for further experimentation. The *Heartline Height* is the position of the rotational point of the track (see **Conversions Panel** for common values) and *Draw Heartline* allows to hide it in the viewport. *Track Color* specifies how the track itself gets displayed in the viewport in the normal state or as a selected section or transition.

3 Track Tab

By clicking on *Edit* in the track list the track tab of the selected track will open up:

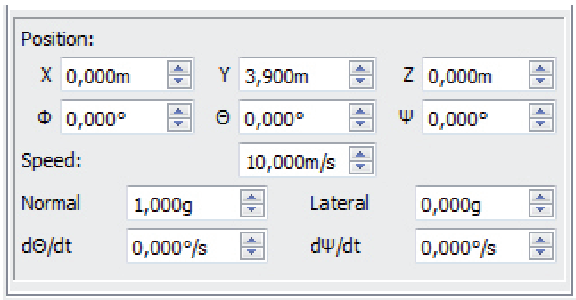
3.1 Sections List



The track tab contains another list, called the sections list, which divides the track into different parts of different characteristics, called sections. Again, new items can be added and deleted by using the buttons below or by right clicking in the list. Each section can be renamed except the *Anchor*. In the following, the different section types will be explained in detail:

3.2 Anchor

The *Anchor* is the fixed first item in the sections list, therefore can't be deleted. It's used to specify important settings of the starting point.



By clicking on the *Anchor* list item, the list of possible options appears below. The first group is about the *Position* and direction of the starting point. *X*, *Y* and *Z* stand for the cartesian coordinates, whereas *Y* stands for the height component. *Φ* (Phi) stands for the roll angle, *Θ* (Theta) is the pitch angle and *Ψ* (Psi) the yaw angle.

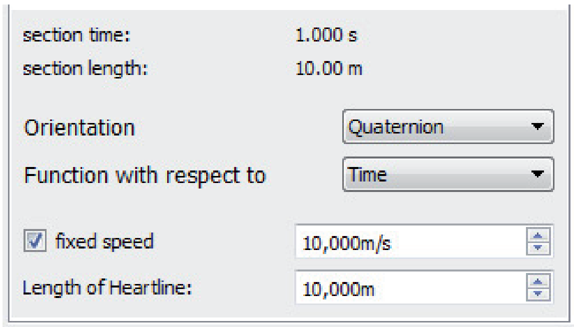
Speed specifies the starting velocity and *Normal* / *Lateral* the starting forces in normal and lateral direction (see **Normal Force / Lateral Force**).

The last row of possible settings concerns angular change rates: *dΘ/dt* specifies the starting pitch change, *dΨ/dt* the yaw change (see **Pitch Change / Yaw Change**).

FVD++ uses the center point between the track rails as a reference for any coordinates concerning position, in contrast to pure heartline-based coordinates of newton, for example. Any transition graph is calculated for the heartline path.

3.3 Straight Sections

These are the simplest section types in FVD++, representing a straight piece of track which can have custom banking.



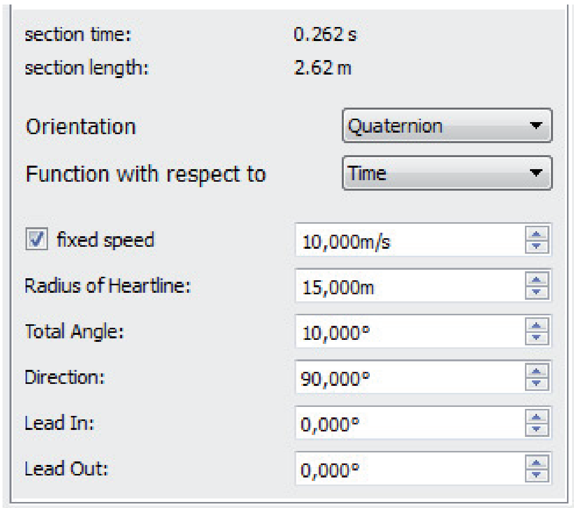
Section time and *section length* are being calculated dynamically when changes to the section are being made. The latter refers to the track length, not the length of the heartline path.

Below, two drop down menus are placed, which are explained in the chapter **Advanced Features**.

The numberfield *fixed speed* sets, if activated, a constant velocity which will be used for the whole section. *Length of Heartline* defines, how long the heartline path of this section has to be.

3.4 Curved Sections

Basically, these sections represent a part of a circle with further configurations, which are the following:



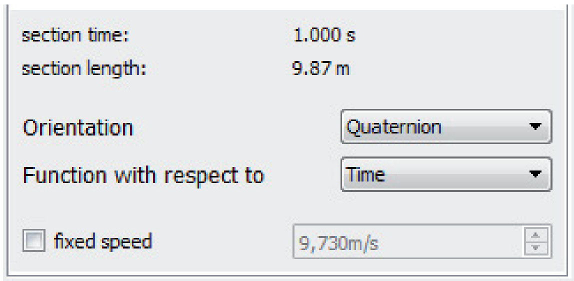
The *Total Angle* defines the angle the track has to bend, including the setting in which *Direction* the bend goes. The *Direction* is pre-set at 90 degrees which results in a track bending to the left. At 0 degrees, the curve would point upwards, downwards at 180 degrees and so on.

Lead in and *Lead out* define how much of the *Total Angle* will be used in order to have a smooth blend from straight to bended track with the specified *Radius of Heartline*. The sum of the lead in and lead out can't exceed the *Total Angle*.

3.5 Force and Geometric Sections

These two are by far the most sophisticated section types FVD++ has to offer. They are quite similiar in use and, most importantly, transition graphs have to be used in order to shape the track using these section types.

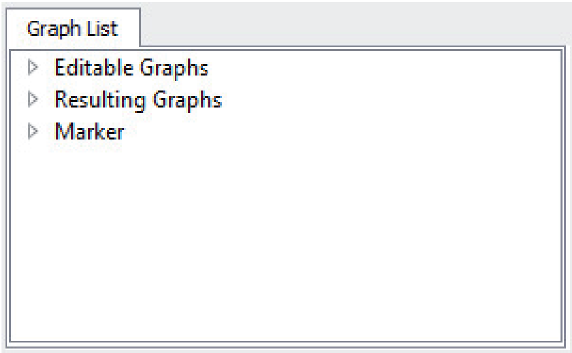
In the track options panel, these two sections types have no specific settings apart from the already discussed ones. To manipulate them and the banking of the section types mentioned before, see the next chapter called **Transition Graphs**.



4 Transition Graphs

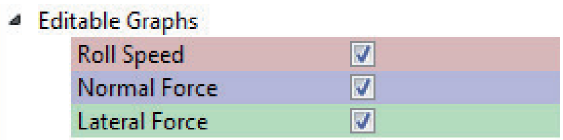
Transition graphs represent custom mathematical functions that are linked to certain characteristics that form the shape and banking of the track. They are the core feature of FVD programs in general.

4.1 Graph List



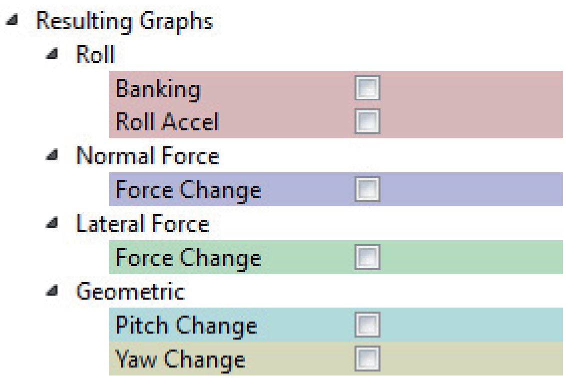
As severly different types of transition graphs are available, they are organised in the *Graph List* below the track tab to define which of them should be displayed in the **Graph Panel** at once (clickbox after each item). The list is splitted up into *Editable Graphs*, *Resulting Graphs* and *Markers*.

4.1.1 Editable Graphs



This set shows the graphs that can only be directly manipulated by the user. *Roll Speed* is the roll speed rate in the track, *Normal Force* the force on the riders in vertical direction and *Lateral Force* to the side. Each of these graph types will be further explained in their own chapters.

4.1.2 Resulting Graphs

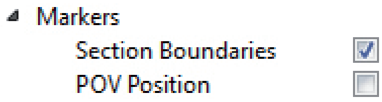


In contrast to the editable graphs, this list shows graph types which are exclusively generated by the program as they represent mathematical derivates or integrals of the editable graphs. The *Roll* section offers the absolute roll value an the rolling acceleration (*Roll Accel*). Furthermore, the change rates of *Normal* and *Lateral Force* graphs can be activated, same for *Pitch* and *Yaw Change* in *Geometric*.

Which graph types are available in the *Graph List* is dependent on the type of section being currently selected. In the *Anchor*, all types of graphs of the whole track can be reviewed without any manipulation functionality.

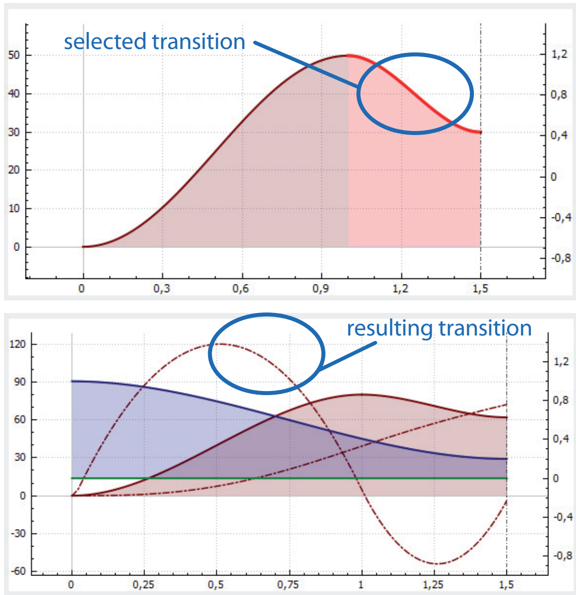
4.1.3 Markers

An activated marker will be shown in the **Graph Panel** as a vertical line to indicate either the *Section Boundaries* or the current *POV Position* (updated dynamically).



4.2 Graph Panel

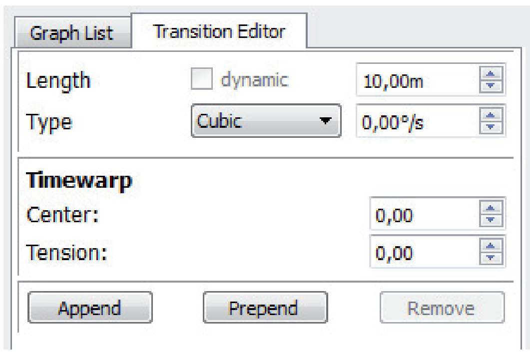
Together with the *Transition Options*, the graph panel is the main workplace in the program. Here, all activated graphs can be seen together in a coordinate system. Editable graphs will be rendered as regular lines, most of the resulting ones as dotted lines which can't be selected. The x-axis is the riding time or the position on the heartline path (see **Function Argument**), in both cases measured from the start of each section. There are y-axes on both sides with the left hand one representing angles, angle change rates and accelerations and the axis on the right side for forces and force change rates.



A transition graph can be selected by clicking on its line, turning it into a lighter color (colors are customizable). Using the scrollwheel, zooming in and out of the graph panel is possible. Paning to the left/right by clicking somewhere in the panel and dragging.

4.3 Transition Editor

Whenever a transition graph gets selected, a new tab next to the *Graph List* appears, called the *Transition Editor*. The numberfields on the bottom, called *Timewarp* settings, will be explained in the **Advanced Features**. The basic settings on the top are the following:



Length specifies how long the transition will last in the section or if it will adapt the length of other transitions, if *dynamic* is checked. More on that in **Dynamic Transitions**.

The numberfield after the dropdown list *Type* sets the main value for the transition. In other words, how much effect this transition will have on the characteristic it stands for.

4.3.1 Transition Types

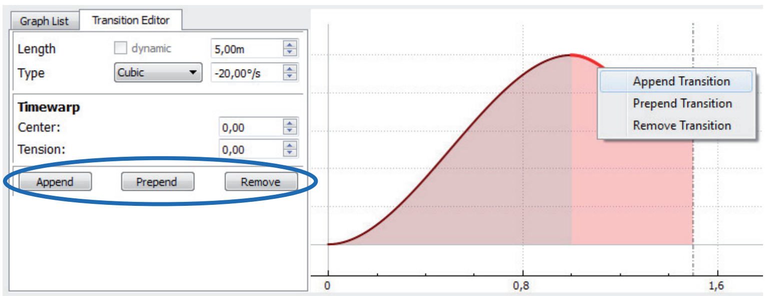
Transition types allow to select from a list of 6 different types and 12 different variants how the specified change in value should be performed mathematically. If the selected type of the first dropdown list contains more than one variant, a second list appears below the first one, showing the variants of that transition type.

In the **Transition Types**, a tabular representation can be found.

4.4 Working with Transitions

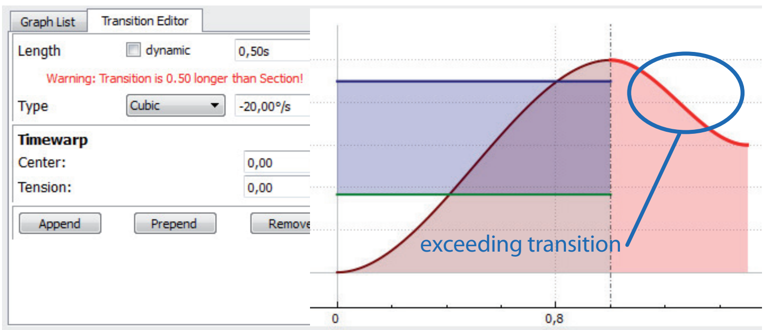
4.4.1 Appending, Prepending, Removing

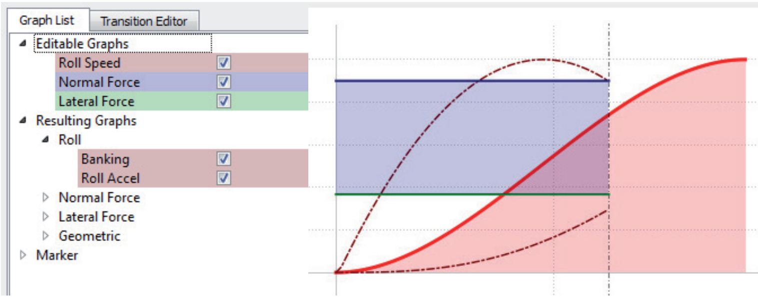
By using the three buttons in the *Transition Editor* or the context menu (right click), a selected transition which is not the only one in the section can be deleted. On each selected transition, a new transition of the same type can be appended or prepended.



4.4.2 Exceeding Transitions

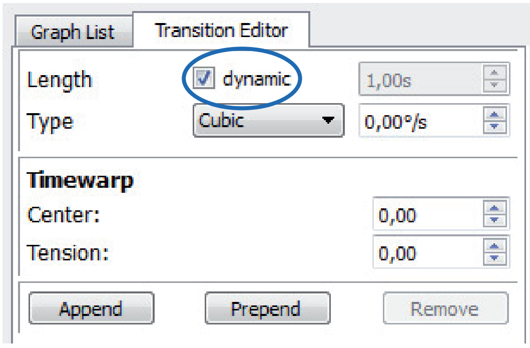
Whenever one or more transitions are longer than the shortest transition in the section, a warning appears in the *Transition Editor*. The parts of the exceeding transition won't have an effect on the section, therefore the next section uses the values of all transitions on the point the shortest of them ends.





The same rule applies to situations where the exceeding transition gets cut off in between. As visualised by two *Resulting Graphs*, the roll speed transition effectively ends on a point where a change of roll value, roll speed and roll acceleration still takes place. This effect could be the origin of unwanted bumps between two sections or could be used for special trickery. The transition variants *Quadratic blend in* and *blend out* (see **Transition Types**) could be used for this purpose, too.

4.4.3 Dynamic Transitions



The checkbox *dynamic* can be activated for every transition, setting it to adapt the lenght of the shortest transition which is not dynamic. The following rules have to be kept in mind when using this feature:

- In every section, at least one graph type will remain not dynamic.
- If the feature is used on a transition being followed by another transition, the latter will be shifted towards the end.
- If the feature is used on a transtion which is after the first exceeding transition, the new dynamic transition will effectively dissappear from the graph panel and can only be recovered by undoing the action or lenghtening the other transitions.

4.5 Graph Types

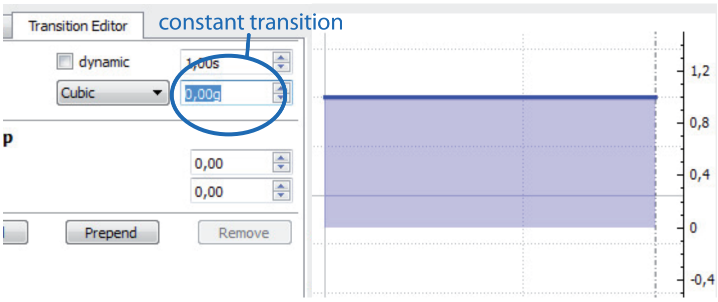
As mentioned above, transitions allow a precise mathematical description of the track that is to be shaped. As the track itself has different attributes, the transitions are grouped in graph types, where each type has his own specific color representation and look. In the chapter **Graph List**, the division into *Editable* and *Resulting Graphs* was already introduced and first the *Editable Graphs* will be explained in detail:

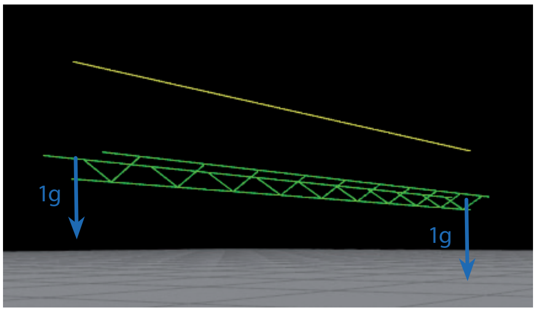
4.5.1 Normal Force

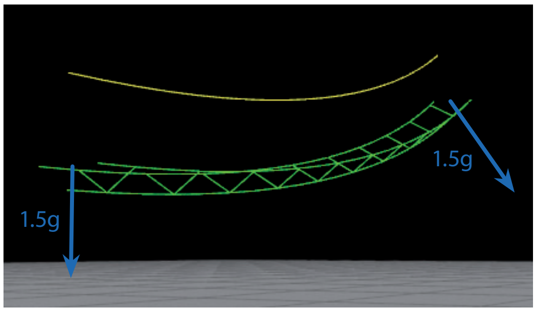
As FVD++ is a force vector design tool, one possible way to form the track is based on forces the user has specified in the transition graphs including real-time adjustments on them. The *Normal Force* graph type can be only edited in *Force Sections*.

After having added a new *Force Section* in the sections list, the group *Editable Graphs* in the *Graph List* can be expanded and other graph types other than the *Normal Force* can be deactivated, if necessary.

Now the transitions showed in blue color can be altered only concerning the vertical forces on the track. In this case, the track will sustain a constant vertical force throughout as the transiton value is set to 0 g change.

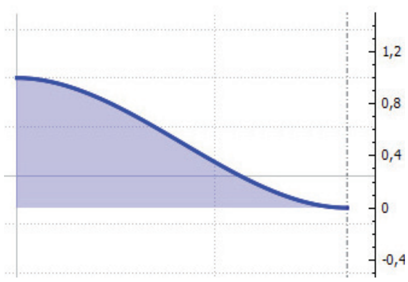


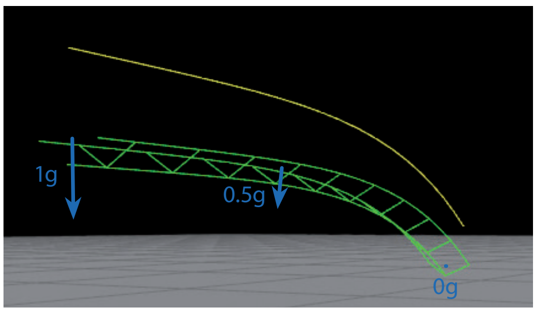




In the left picture, the *Anchor* was set to the default 1 g, on the right, the starting normal force was set to 1.5 g. In both cases, the force at the start of the section remains the same at the end.

The next picture shows a graph that changes the vertical / normal force by -1g over the section. As a result, the track will bend downwards to generate 0g with the specific speed the virtual train has on the end of the section.

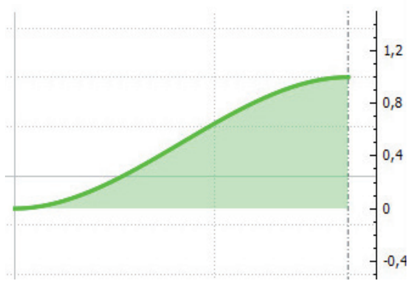


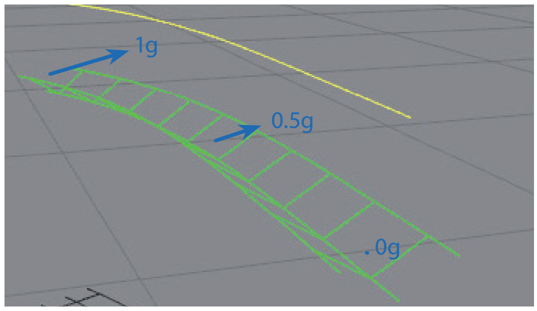


4.5.2 Lateral Force

Lateral Force graphs are following the very same principles like the *Normal Force* ones, but they will control how the rider will be pushed to the side while riding the track.

In the *Graph Panel*, they are displayed in green.



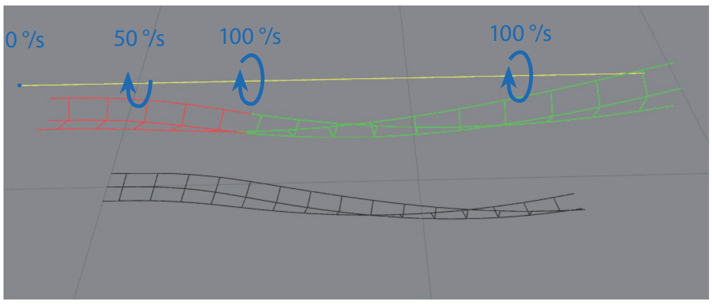


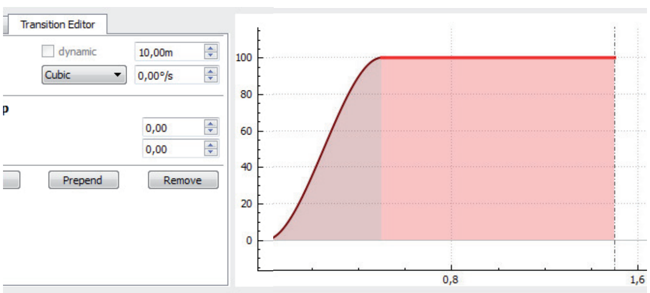
4.5.3 Roll Speed

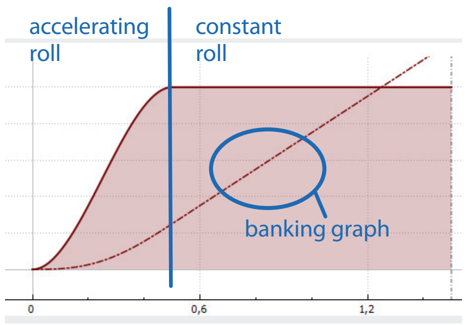
Whereas both of the two discussed force graphs are related to the absolute values of vertical / lateral force, *Roll Speed* introduces a completely different concept. It describes the rate at which the rider spins around the heartline.

In the simplest case, the *Roll Speed* of 0 will be taken from the previous section and the *Roll Speed* transitions remain constant. Then, the banking will stay zero over the whole section, as no change rate is defined.

The picture shows the entrance of a heartline roll. At the beginning, the *Roll Speed* increases by 100 °/s and stays constant at this rate for the rest of the element. To end the rolling motion, another transition bringing the roll rate back to 0 °/s needs to be appended.





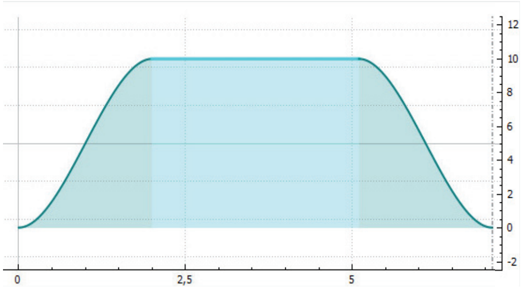


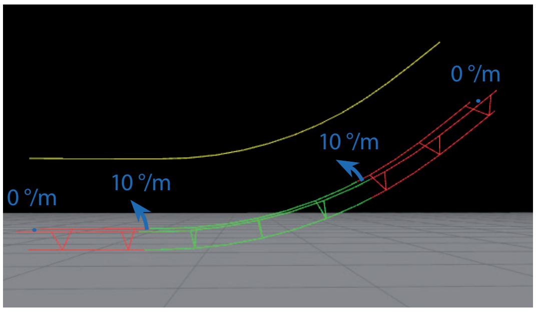
Following the same pinciple as *Roll Speed*, namly angular change rates, the two last graph types aren't typically used as widely but they make the core part of *Geometric Sections*:

4.5.4 Pitch Change

Pitch Change, visualised through cyan graphs, describes how fast the pitch angle relative to the ground plane (see **Orientation**) will change. In some cases, this graph type behaves quite similar to *Normal Force* and it offers more configuration than a simple *Curved Section*. The example on the right shows the use for a lifthill pull up.

Here, the *Function Argument* was set to *Distance*. More on that in the corresponding chapter.

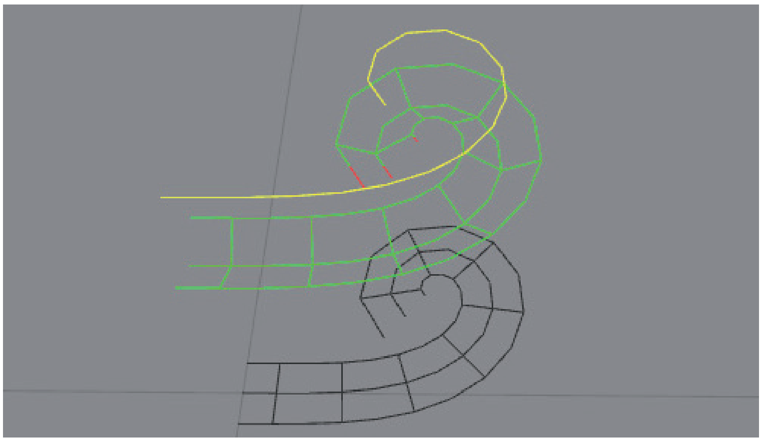




4.5.5 Yaw Change

Yaw Change is basically the same as *Pitch Change*, the only difference is in which direction the effects of the graph type take place, in this case to the side of the track (again relative to the ground's coordinate system). The graphs are displayed in yellow.

The last example shows the *Yaw Change* increasing from 0 °/m to 80 °/m over the section. As the result, the track needs to bend more and more to produce the specified angular change rates:

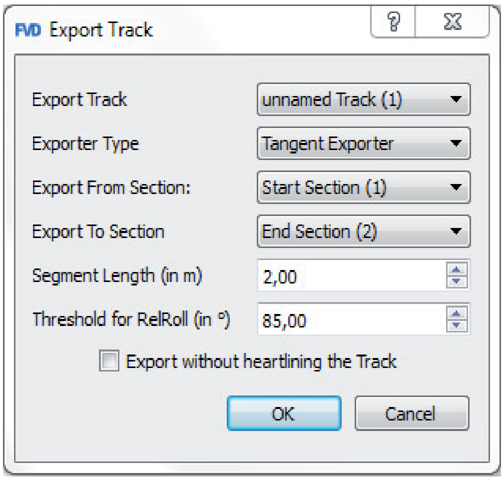


5 Exporting

5.1 General Information

When the track should be imported into NoLimits, it needs to be exported out of FVD++ in order to create an *.nlelem file which can be inserted in the NoLimits Editor. After clicking on *Export* (Ctrl+E) in the *File* menu, a new window will appear to specify the desired exporting configuration:

In this window, different exporting settings can be defined and by clicking on *OK*, a file manager window will open to select the directory the generated NoLimits element will be saved to. After having done the first exporting action, FVD++ will keep the settings so each time *Export* gets clicked or Ctrl+E pressed, the export will be done exactly the same way without asking about the settings. To specifiy a new exporting configuration, a click on *Export As* (Ctrl+Shift+E) will open the *Export* window again.



5.2 Export settings

5.2.1 Basic settings

Export Track allows to specify which track should be used for the exporting action.

Export From Section and *Export To Section* defines the start and ending points, allowing to select only a part of the coaster to be exported. Next to the section names their numbers is displayed as well.

The *Segment Length* will give the exporter engine the length each exported NoLimits segment has to have. Values can be set from 0.2 m to 10 m but the lowest value in use should not be smaller than about 0.5m because otherwise it might create an unsmooth NoLimits track (this depends on the final geometry).

5.2.2 Advanced Settings

The *Exporter Type* can be choosen between the *Tangent Exporter* (preset) and the *Spline Exporter*. The latter tries to create a bezier curve that follows the different sampling points calculated by FVD++ which might cause problems when exporting different sections individually. The endings of each exported section won't match as each of them only had to follow its own curve. For instance, exporting a *Curved Section* will result in a NoLimits element which has a effectively smaller *Total Angle*. In contrast to this, the *Tangent Exporter* will keep the tangents of each exported vertex point exactly how they were indended to be, therefore each angle will be perfectly matched up with the track model inside FVD++.

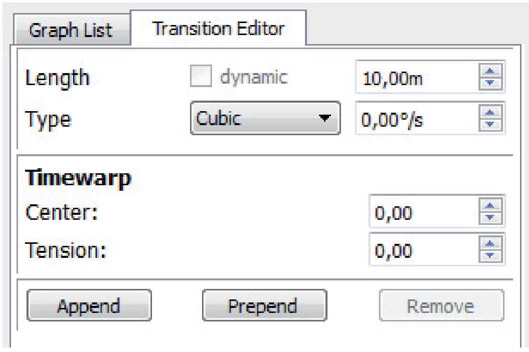
Threshold for RelRoll defines at which pitch angle of the track – from 75 to 90° – the vertices inside NoLimits will be set to relative rolling instead of normal rolling.

The clickbox *Export without heartlining the Track* will export a bezier curve that represents the heartline path itself (including banking) instead of the actual track which rolls around the heartline.

6 Advanced Features

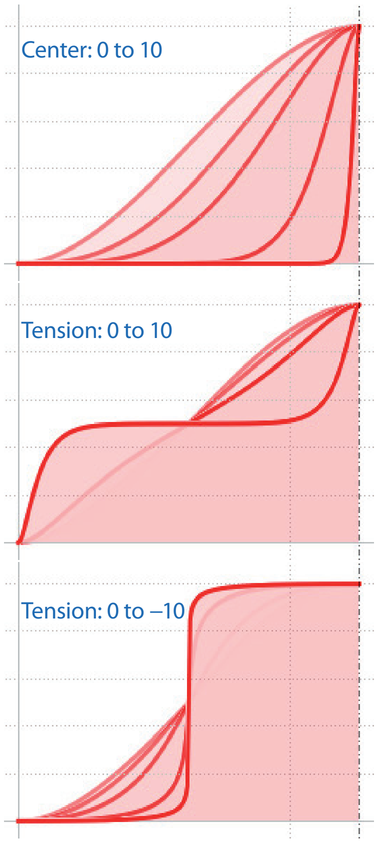
6.1 Timewarping

The timewarping panel can be found in the *Transition Editor* below the transition settings. Two number fields called *Center* and *Tension* are available.



Timewarping will provide more control over the selected transition which helps to improve the shaping, getting the track into the desired position or using less transitions for certain tasks.

Positive values entered in *Center* will make the transition's center shift to the right, negative values will shift to the left. Extreme values (>5) will flatten out the opposite side.



Picture on the right: values 0, 1, 2, 5 and 10 on a *Cubic* transition.

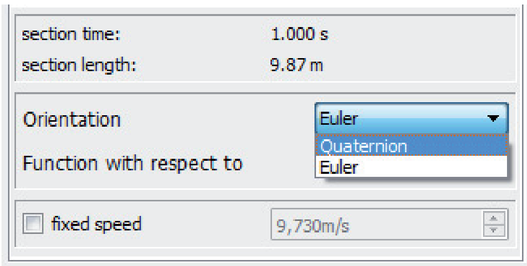
Tension values ranging from 0.01 to about 1.5 will make the transition more gradual, higher values (up to 10) will make the middle of the transition be more horizontal. Negative values up to 10 will make the transition more abrupt in the middle part and more flatten out on the endings.

The second picture shows the positive Tension values 0, 1, 2, 5 and 10 and the third one the negative values.

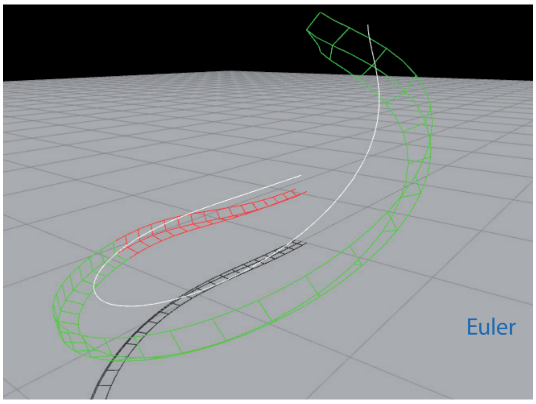
In most cases, values ranging from –2 to 2 can be used without any problems, whereas more extreme values might cause unwanted bumps in the track, but there are exceptions.

6.2 Orientation

One of the most crucial settings can be found in the section settings, the *Orientation*. In a nutshell, this setting – switchable between *Quaternion* and *Euler* – controls in which way the program creates the final track banking based on the *Roll Speed* graph. Taking the example of a basic helix will reveal the underlying principles.

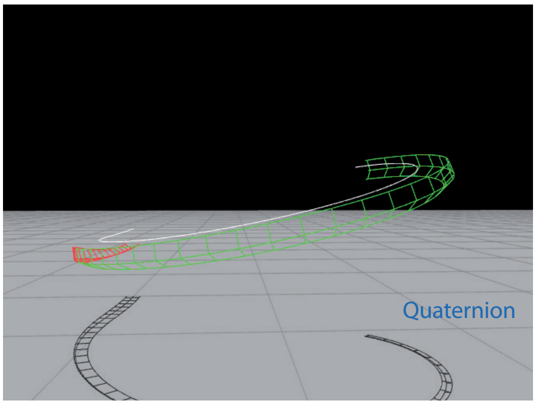


The first attempt of building the helix is by using the *Euler Orientation*. In this case, the roll values of the track are always measured relative to the ground plane and as the *Roll Speed* rate stays 0°/s for the rest of the helix, the roll of the beginning of the element has to stay constant. As a consequence of this fact, the train slowing down on its ascend and the constant 3 g *Normal Force*, the track starts to screw itself upwards with a rapidly decreasing radius. In the *Euler Orientation*, this is the path a train has to follow in order to have constantly 65° roll relative to the ground plane. A possible solution to this would be to avoid slower speeds or manual *Roll Speed* corrections..



Another issue with *Euler* is a vertical (pitch=90°) piece of track as it can't be calculated because the roll angles can't be defined properly in this case.

By using the *Quaternion* orientation the picture is fairly different. This time, the helix doesn't form a spiral. Instead, it stays perfectly in a slanted plane which is defined by the constant roll value of 65°. *Quaternion* does not use the global coordinate system as a reference for their roll values, it defines the roll based on the coordinate system of the track itself, which is tilted by 65° at the beginning of the helix part. But the roll reference is not fixed to this point, it will change whenever the roll of the track has to be altered due to the transition graphs. A consequence of this principle is that the (Euler-based) roll of the helix will get



higher in the ascending part, being 85° at the end of the example element. This is simply due to the slanted plane the roll is based on; in the perspective of the said plane, the roll stays perfectly constant.

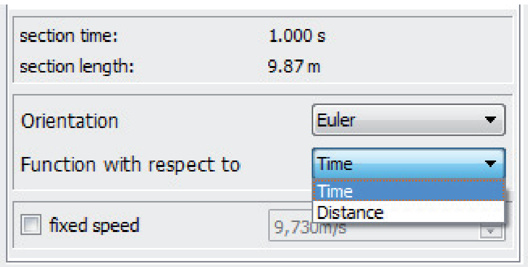
In conclusion, the main difference between the two orientations is the reference for the roll values which lies either on the ground plane (*Euler*) or on the track itself (*Quaternion*).

As a side note: the statistics panel uses *Euler* for the display of the roll value, whereas the *Roll Speed* is always calculated using *Quaternion Orientation*.

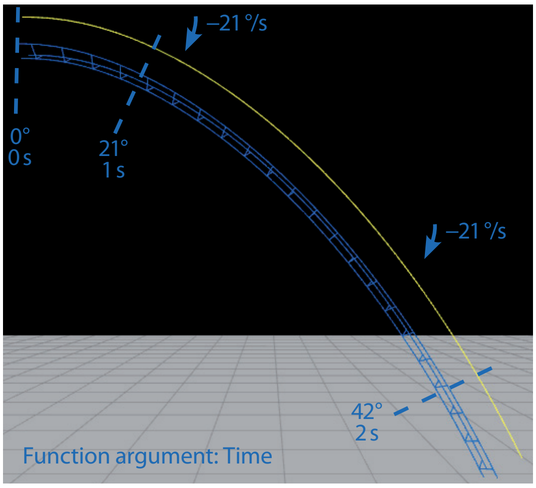
6.3 Function Argument

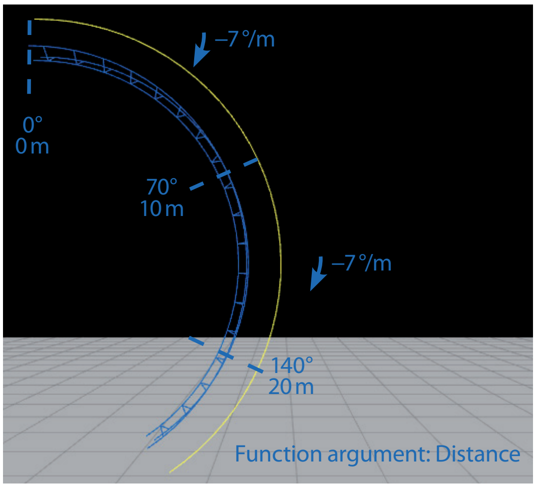
This additional setting has already been used on some points in this document and will now be explained in full:

In FVD++, each section can be set to have their transition graphs based on *Time* or *Distance*. This setting, namely *Function with respect to* can be found in the section settings below the **Orientation**. It defines on which physical dimension the program's calculations will be based on in the current section, in other words what the x-axis in the graph panel is supposed to represent.



This drop uses *Time* as the function argument. Over the whole element, the *Pitch Change* stays at constantly –21 °/s. As showed in the picture on the right, the track gets more and more streched (decreasing track flexion) when further going down. This effect is due to the acceleration of the train, therefore one second of ride time needs a longer piece of track which has to bend by 21° every second. The result is the track becoming more flatten out as the speed gets higher. This effect creates bumps at slower speeds as the track flexion will have change even more faster.





Using the *Pitch Change* based on *Distance*, the result resembles a perfect circle instead. In this case, the rising speed has no effect on the track curvature at all. With 3 m/s starting velocity, –21 °/s will convert into –7 °/m *Pitch Change*.

Often a good strategy to avoid unwanted effects is to use both techniques together. For a first drop, the best way would be to start with a distance-based apex bend using a fixed speed, followed by a distance-based bend with calculated speed which is followed again by a time-based bend and after that, a *Force Section* to create the valley.

The same principles apply to all other graph types as well.

7 Principles

This chapter provides further information on some of the principles FVD++ is build on and how to use them in the designing process. This chapter is not of the same structure as in the previous chapters and uses a different style. It should be regarded as a collection of noteworthy explainings, being seperated from the main text.

7.1 Graph types

7.1.1 Grouping by Accessibility

All of the graph types can also be divided into four different groups, distinguishable by the aspect of how the graph types are usable in different section types:

1 Force: *Normal Force* and *Lateral Force* are editable in only one section type, the *Force Section*.

2 Geometric: *Pitch Change* and *Yaw Change* are editable only in *Geometric Sections*.

3 Roll Speed: Can be edited in every section type.

4 Resulting: Can only be seen as a resulting graph, never editable.

Now, for example, when working in a *Force Section*, the graphs of the force group plus *Roll Speed* are editable and all the other groups can be found in the *Resulting Graphs*. In other words: Each section type has its own rules of which graph types are editable and which not and this grouping method allows to understand the principle behind that.

7.1.2 Graphical Representation

As mentioned in the chapter **Graph List**, the *Resulting Graphs* have a different graphical representation than the *Editable* ones. Generally speaking this is true but, in fact, not the whole answer.

If a graph type of the force or geometric group (see 7.1.1 above) is placed as a *Resulting Graph*, it will still be rendered as a solid line, even as being not editable. The reason for this is that these graph types are used in other section types as editable graphs. Therefore they have to be distinguished from graphs of the resulting group which are displayed only for reviewing purposes.

7.2 Linking

Now onto one of the major differences between graphs of the force group and the geometric group: the linking. It specifies on which coordinate system the transition values are based on.

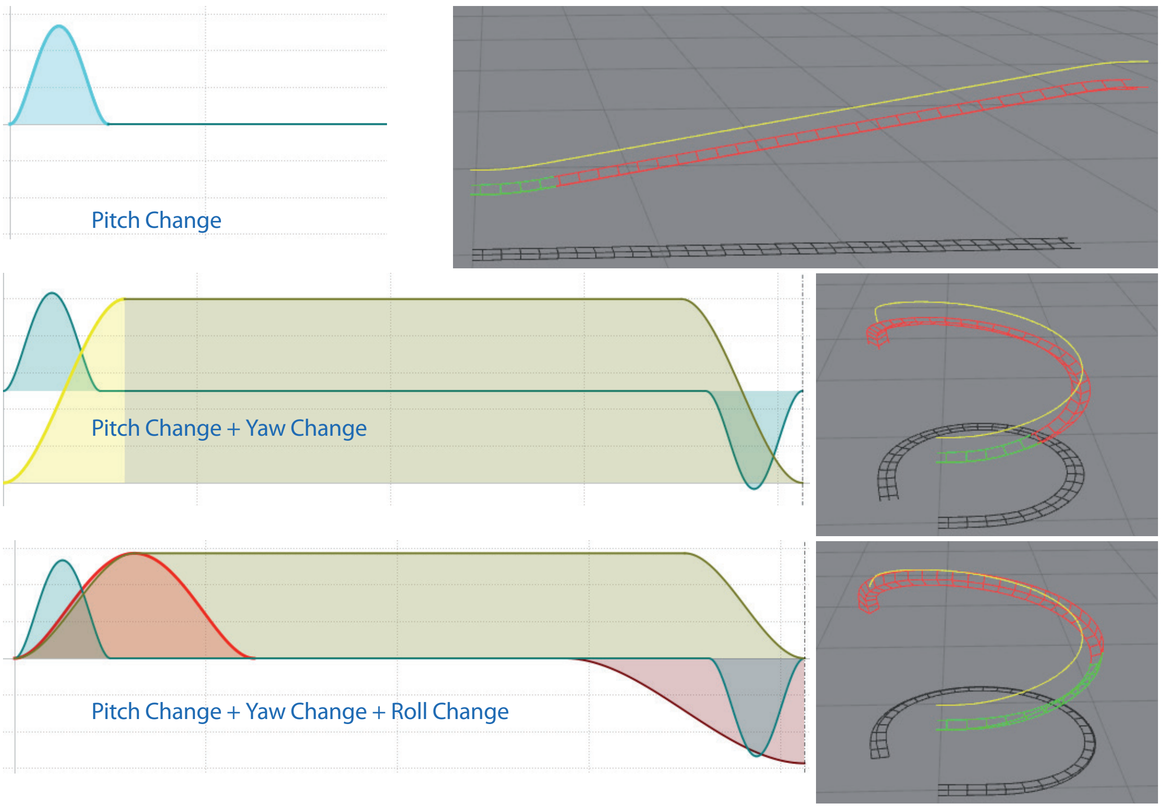
Compare with the chapter **Orientation** which essentially describes the linking of roll values.

7.2.1 Global

Using the global, on-ground linking, a specified change of a value is seen by the program in relation to the global coordinate system. This linking type is used on *Geometric* sections, for example.

The following picture shows a helix that was built as a *Geometric* section. The *Pitch Change* was used to have a constant track slope during the helix including smooth lead in and lead outs. The *Yaw Change* introduces the circular shape of the helix when viewed from above. Lastly, minor *Roll Speed* variation reduces unwanted lateral forces.

The following pictures show how a helix using a *Geometric* section can be seen as a compilation of the different graphs with the seperated effects of *Pitch Change*, *Yaw Change* and *Roll Speed*.



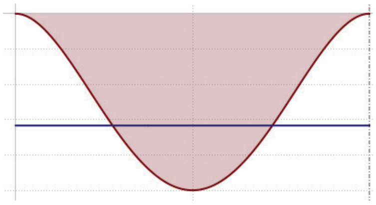
The example element shows clearly how the different graph types are based on the global coordinate system as each of their effects can be seen individually.

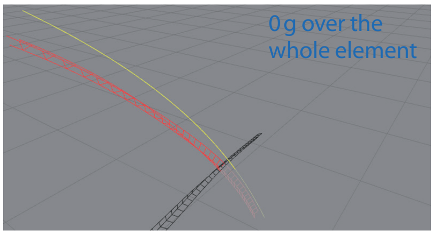
7.2.2 Local

Local is the exact opposite linking type as global linking. Here, the change of a value is seen in relation to the local coodinate system which follows the track dependently on the actual position on the track and the current roll angle.

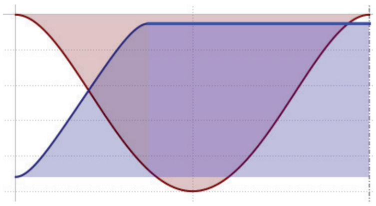
As in *Geometric* sections the yaw change refers to the track's yaw angle change rate, the *Lateral Force* graph in *Force* sections refers to the bend of the track as seen by the track itself, not from the ground perspective. *Lateral* forces are experienced by the riders, not from the outside.

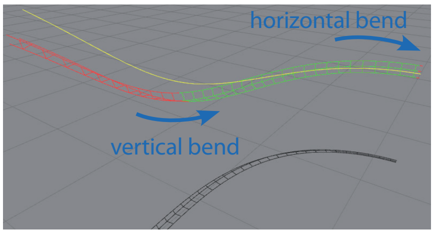
The example element is the transition from an airtime hill (0 g vert. and 0° roll) to a flat turn (3.5 g vert. and 80° roll). The first pair of pictures is set without any *Normal Force Change*, so the vertical force stays at 0 g, and the needed *Roll Speed* to achieve 80° roll a the end of the element.





The next pictures show the same element with the needed change to 3.5 g *Normal Force*. As *Normal Force* (and *Lateral Force*, too) use local linking, the applied *Normal Force* change bends the track not only up, but also to the side at the end of the element (see the shadow). The reason for this is that the *Normal Force* always effects the track in the direction vertical to the track (so called normal direction). So the resulting curvature of the track is dependent on the roll and lateral forces as well.





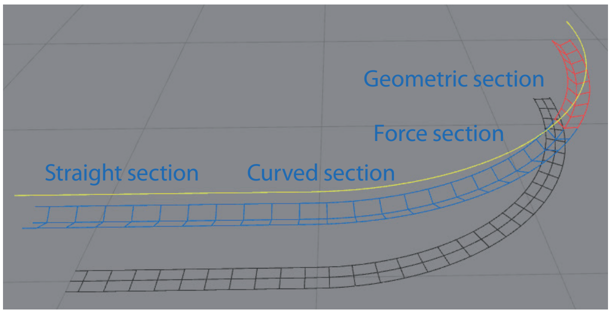
See **Coordinate Systems** for a graphical comparison between the two coordinate systems.

7.3 Continuity

This is the very last chapter of the explanational part of the FVD++ Documentation. It's about a useful feature which can be called continuity (of physical properties).

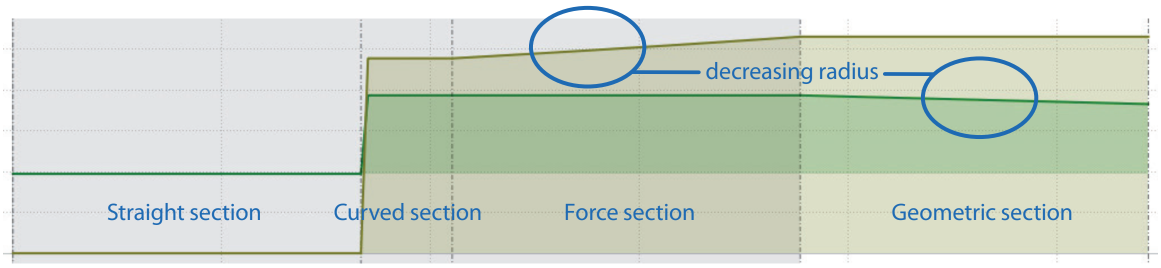
The example track is quite simple but reveals some stunning facts once properly examined. It consists of all available section types of FVD++ in a row, with each one using the default settings.

Whereas the boundary between the *Straight* and *Curved* sections is properly defined through the section settings, the second boundary after the *Curved* section is more interesting:



Here, the following *Force* section remains the same curvature as the radius of the *Curved* section before. The same happens with the *Geometric* section at the end which is curved as well.

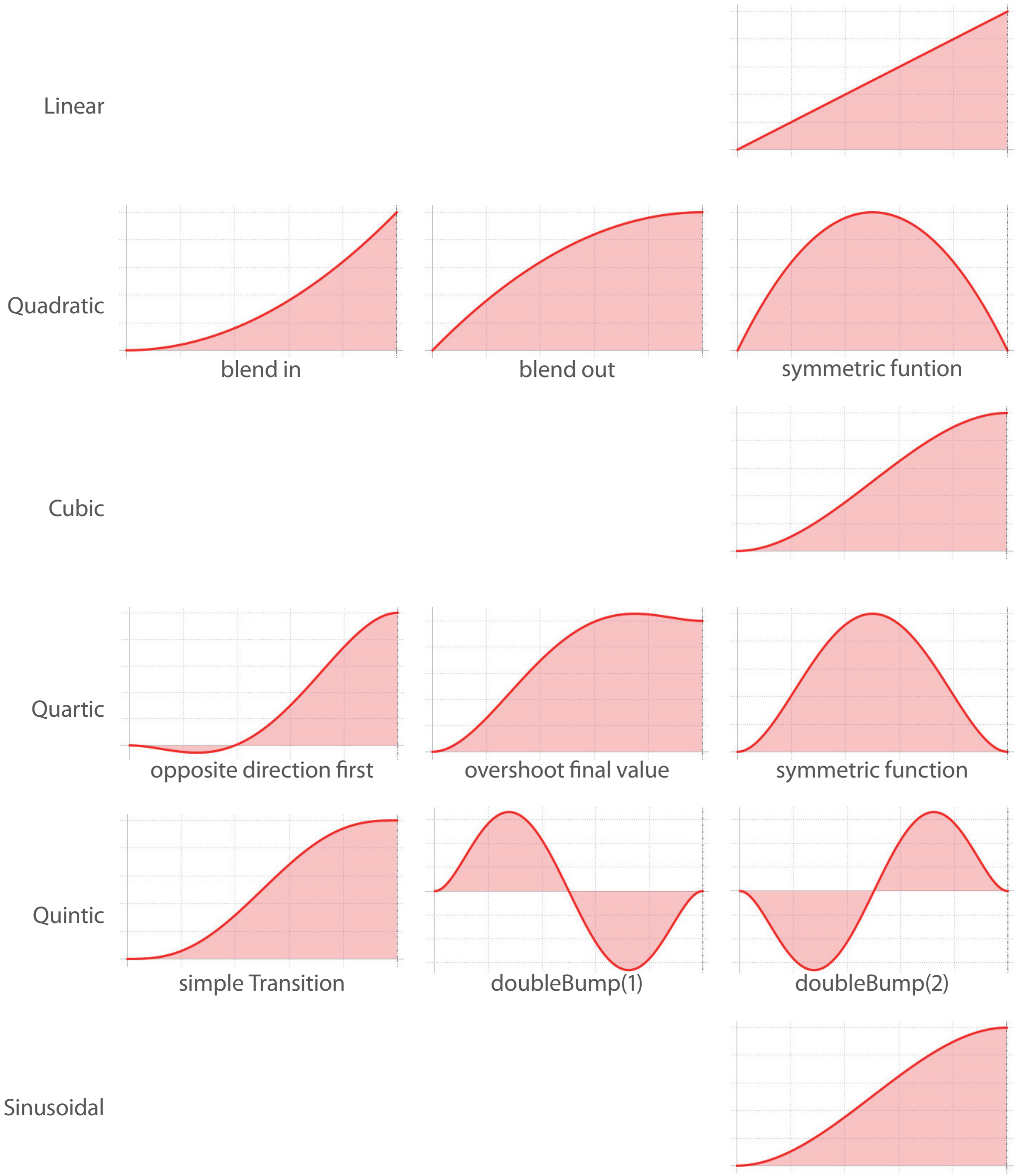
The continuity principle of FVD++ lies in the fact that every new section tries to remain the track geometry at the end of the last section as close as possible. Often the section isn't of the same type so the continuity has to be accomplished only by the graph types this particular section has to work with. In the example, the *Force* section introduces a certain constant *Lateral Force* to remain the horizontal radius of the *Curved* section. This *Lateral Force* is converted into a starting *Yaw Change* value in the *Geometric* section.



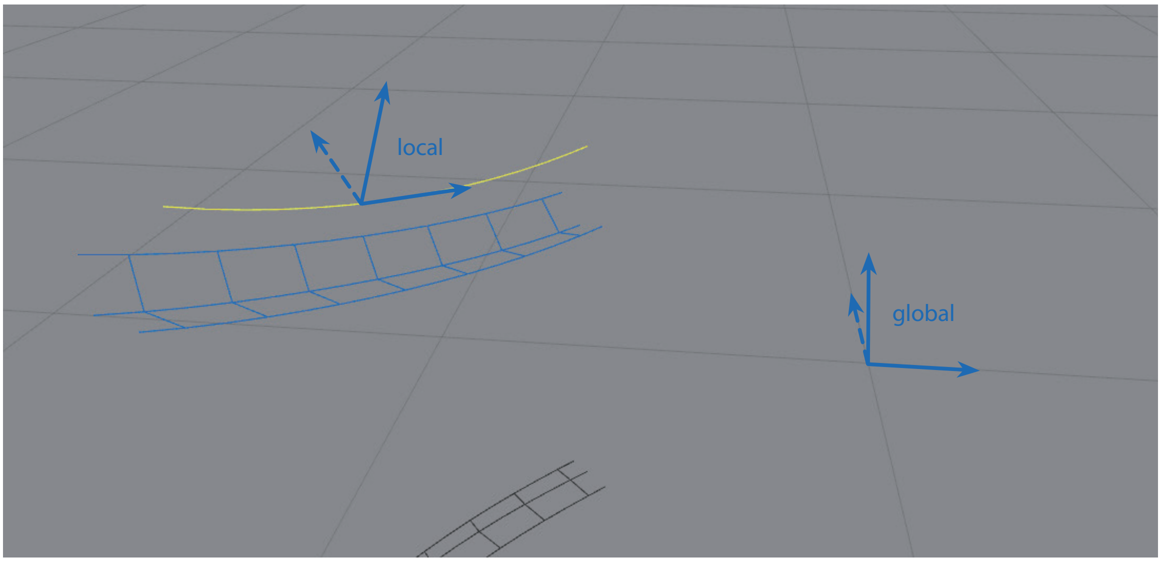
In this example element, the *Friction Coefficient* has been raised to 1 ro reveal another effect: When the *Geometric* section is selected (picture above), the *Yaw Change* in the previous *Force* section is rising slightly, therefore the track's radius gets smaller. This happens because of speed losses due to friction. As the *Force* section remains a constant *Lateral Force* but with a decreasing velocity, the resulting track radius gets smaller. In return, the *Lateral Force* (green graph) in the *Geometric section* decreases because of the same reasons.

8 Appendix

8.1 Transition Types



8.2 Coordinate Systems



8.3 Greek Symbols Cheat Sheet

roll angle:	Φ (Phi)
pitch angle:	Θ (Theta)
yaw angle:	Ψ (Psi)

written by Lucas van den Bosch (privat@lucasbosch.de)

FVD++ development by Stephan Alt (alt.stephan@web.de)

Further helping resources:

FVD++ Beta: Discussion Thread (v0.5):

http://forum.nolimits-exchange.com/comments.php?DiscussionID=2646

Lenny's Engineering Thread (about the early development):

http://forum.nolimits-exchange.com/comments.php?DiscussionID=2546

FVD++ Help and Tutorial Thread:

http://forum.nolimits-exchange.com/comments.php?DiscussionID=2712

YouTube Channel FVD++ Tutorials:

http://www.youtube.com/user/fvdtutorials