OBJECT ROTATOR for Microsoft Train Simulator

Manual for Version 2a

Freeware
by Michael Vone

Object Rotator is a tool for advanced users of MSTS to rotate objects accurately.
It is used to modify an object's QDirection and/or Position in world tile files (*.w).
It requires Microsoft Excel (it will not function with an Excel viewer).

Major applications of this tool are the precise alignment of bridges and platforms along tracks or roads, the accurate orientation of objects, the precise lining up of rows of objects, along a straight or curved line, and the accurate aiming of straight tracks or roads between distant points.

This tool was designed for simple objects (such as bridges, buildings, and track and road sections); it has not been tested with complex or interactive track objects (such as forests, signals, level crossings, mileposts and gantries).

Object Rotator requires familiarity with editing files (using WordPad). It is also helpful to mentally "see" rotations in 3D. Object Rotator is typically run at the same time as the MSTS Route Editor.

The reason for offering this tool is that rotating objects is generally complicated, and especially difficult in the MSTS Route Editor (RE): the RE tools for rotations are not precise, making it hard to align objects accurately.

Note that precise rotations remain labor-intensive, even with this tool.

CREDITS: I am very grateful for the critical help of the beta-testers: Bill Burnett, Chris Cyko, Richard Garber, and Jim Ward.

VERSION HISTORY:

1.01: Corrects wrong results for 3rd and later objects for left curve in Curved track and road; corrects description of functions on first sheet.

2: Adds sections 10 (Aim with curve), 15 (Shift), 16 (Parallel deviation), 17 (Island split), 18 (Next track section), 19 (Rollercoaster curves), 20 (S countercurve), 21 (S joint), 22 (Curved joint), and in sections 25 and 26 discussions of QDirection and heading/slope/bank angles; section 27 discusses making rollercoasters.

2a: Minor reformatting of text.

TABLE OF CONTENTS

1. FUNCTIONS OF THIS TOOL

This tool provides the following main options, which you select by clicking on the corresponding tab at the bottom of the Object Rotator Excel spreadsheet. They are arranged from easier to more complex. It is therefore recommended to learn using these functions in the same order in which they appear. Each of these functions is described in more detail in sections 3 to 22 of this manual.

- **Set parallel**: lets you turn an object so it is parallel to another object, such as a bridge parallel to track;
- **Set antiparallel**: lets you place an object parallel to another, but pointing in the opposite direction; for example, place a bridge or platform parallel to track or road, but with bridge pier at other end, or platform edge on opposite side
- **Stand vertical**: lets you turn an object so it stands vertically, without changing its heading (compass direction); if an object leans sideways, this function will make it vertical
- **Stand perpendicular**: lets you turn an object so it stands perpendicular to another, without changing its heading (compass direction); for example, makes an object lean sideways so it stands perpendicular to sloping terrain or track/road
- **Lay across**: lets you turn an object so it lies across another, such as a bridge oriented 90º to a track; use this function to correctly place certain "cross-wise" objects like fences;
- **Line up**: lets you place a series of identical objects in a line, such as a long bridge or long platform, or an underground tunnel;
- **Aim straight**: lets you orient straight track or road to link distant points, such as to a distant invisible point or through a mountain tunnel;
- **Aim with curve**: lets you connect existing track to another point with an initial curve followed by straight track;
- **Curved track and road**: lets you place track and road in a constant curve in difficult situations, such as underground or with very high slope;
- **Curved row**: lets you place a series of objects along a curve, such as a multi-section bridge or platform along a curved track;
- **Turn about OBJECT axes**: lets you turn an object by changing its internal bank, slope or heading angles: adds or subtracts a bank, slope or heading change to an existing orientation;
- **Turn about world axes**: reorient an object to achieve a particular bank, slope or heading; this is similar to the case with object axes, but using less intuitive world axes, while additionally offering several preset orientations;
- **Make a parallel deviation**: place an S-curve to make straight track deviate behind a platform or building;
- **Split dual track around an island**: place S-curves to split straight dual track symmetrically around a platform;
- **Shift objects without rotation**: move an object (without rotation) through a particular distance along a particular direction;
- **Attach the next track section**: join another (single-) track section to an existing one;
- **Smoothly continue steep dynamic track for rollercoasters**: join another dynamic track section to an existing one, with smooth slope and "bank";
- **Place an S countercurve**: placing a second "countercurve" to reverse the turn of the first curve;
- **Make an S joint**: fill a gap with two countercurving dynamic track sections;
- **Make a curved joint**: fill a gap with a curved and a straight dynamic track sections;
- **Data storage**: this sheet lets you save data (like QDirection or Position data) for future use, if you save the Object Rotator after use.

Section 2 tells you the general procedure for using Object Rotator.

In addition, sections 23 to 27 discuss the following topics:

- Procedure for copying results from one worksheet cell to another;
- Measuring the dimensions of objects;
- How do QDirection and rotations work?
- Relations between QDirection and heading/slope/bank;
- Making rollercoasters.

2. GENERAL PROCEDURE FOR USING OBJECT ROTATOR

Typically, you use this tool as follows (variations will occur depending on precisely what you want to do):

- load your route in the RE (Route Editor);
- place some objects near where you want them to be (keeping their default orientation);
- save the route;
- open Object Rotator in Excel and go to the sheet that you need;
- follow the instructions on that sheet (or in this manual, especially when you are learning);
- using WordPad, open the route's world tile file containing those new objects;
- copy data from that world tile file to the Object Rotator;
- use Object Rotator to generate new orientation and/or position data for those objects;
- copy those new data to the same world tile file;
- save the world tile file (in Unicode format);
- reload the route in the RE (without saving);
- move the new objects to their desired location (without rotating them);
- save the route.

More detailed instructions are given on the individual sheets of Object Rotator.

If you are not familiar with using Excel spreadsheets, you should understand the following: whenever you insert a value in a cell, Excel considers that you have changed the spreadsheet, so Excel will ask you whether to save the changed spreadsheet or not when you try to close the spreadsheet; if you choose to save, your new insertions will be saved; otherwise, the old values will remain unchanged; you should choose what suits you best.

This tool is protected: you can only insert the requested numbers, and copy the results. You can quickly tab from one "unlocked" cell to the next. If you accidentally change the content of a cell, try to Undo the change; if that fails, delete the whole Excel file from your hard disk, and start with a new copy from the source zip file.

3. SETTING OBJECTS PARALLEL

USAGE: - to learn, follow the detailed instructions below;
- OR follow the brief instructions on the Set parallel sheet of Object Rotator.

It is relatively easy to make two objects be precisely aligned parallel to each other. Examples where this is useful include: placing a bridge or platforms along a straight track, placing a bridge along a road, or placing a road along a track.

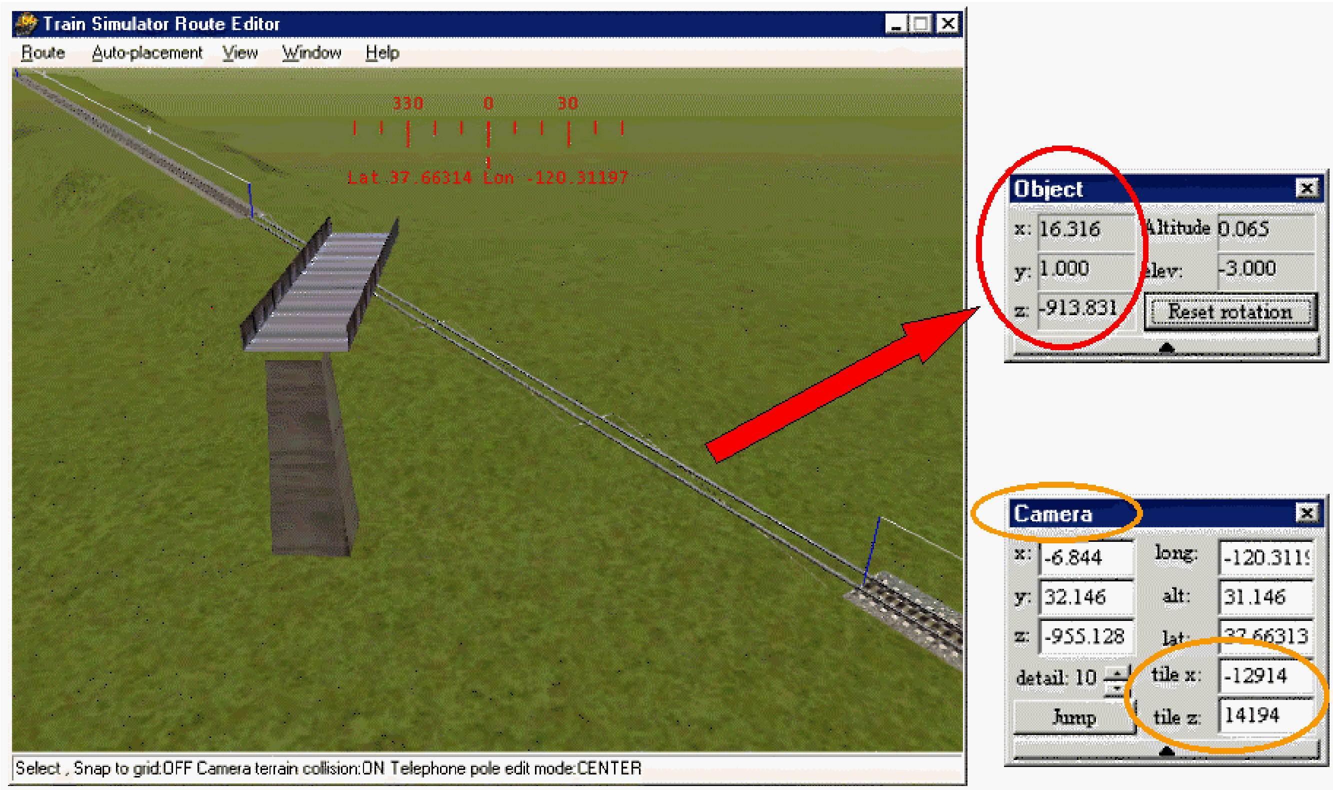
NOTE: If you want to place a row of parallel objects (such as a row of bridge or platform sections along a track or road), use the **Line up sheet of Object Rotator**.

To set objects parallel requires finding and copying data within the MSTS world tile file that contains the two objects: so the primary task here is to find the correct tile file and, within it, the entries for the two objects. (This task will also be central to all the other functions provided by the Object Rotator, so it is useful to learn this task in this relatively simple case. Unlike the other functions of Object Rotator, however, setting objects parallel does not require running Excel.)

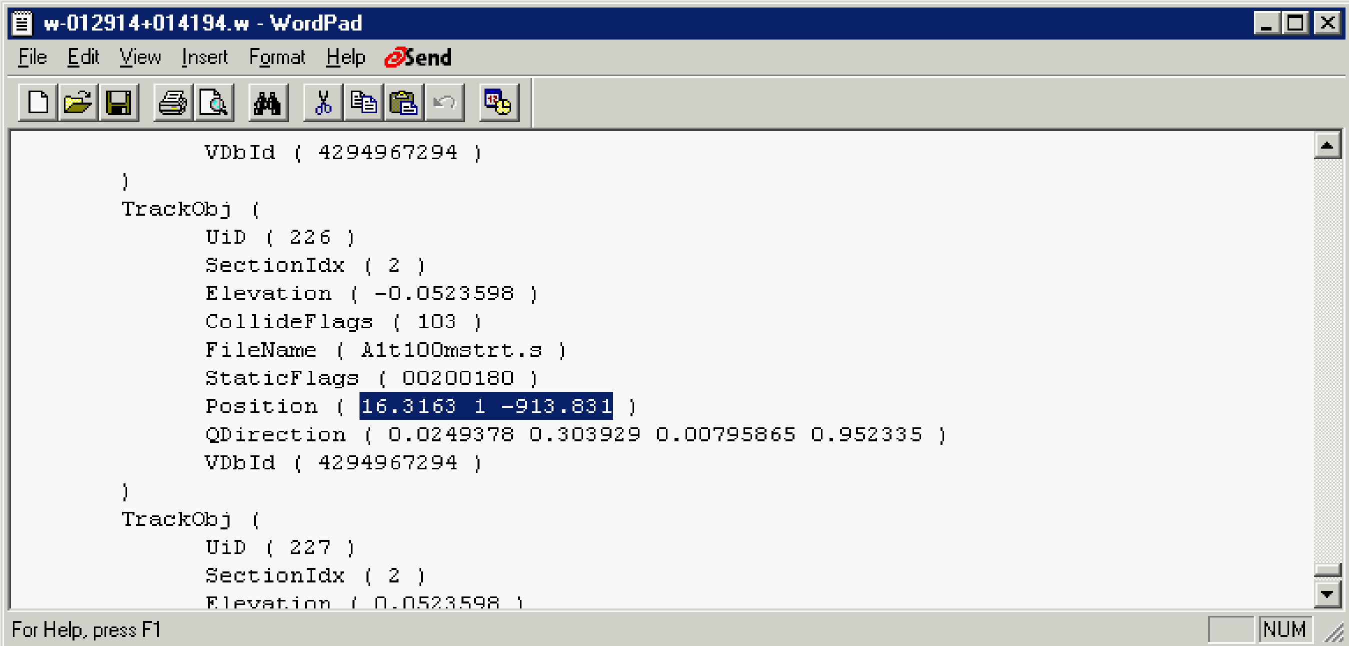
We illustrate here the process with a particular case: let's decide to place a railway bridge parallel to a straight track, under the track. We will allow the track to be sloped and oriented in an arbitrary direction, and want the bridge to have exactly the same slope and orientation. The way to do this is the following (the step numbers do NOT correspond to those in the Object Rotator sheet, because more detailed steps are given here):

1) place your straight track in your route, with its desired slope and orientation (we will use A1t100mstrt.s for illustration), as shown in figure 3.1;
2) place a section of bridge nearby, without worrying about its slope, orientation or exact placement (we will use Jp2bluebrg.s for illustration); note that the bridge points to the north by default, as illustrated;
3) make sure the bridge section is in the same tile as the track section under which it should fit: to do that, press F7 and check that both the bridge section and the track section are within the same tile bordered by blue lines;
4) save the route;
5) check that the camera is within the same tile as bridge and track;
6) write down the two tile numbers of this tile: they are the two numbers named "tile x" and "tile z" in the Camera window of the Route Editor, as marked in orange in the figure below (when you become more used to this process, you will be able to skip this writing and go directly to the relevant file);
7) select the track section under which you want the bridge section to be placed, as shown in the figure below;
8) write down the position of this track section: it is given as "x", "y", "z" in the Object window of the Route Editor, as marked in red in the figure below (when you become more used to this process, you will be able to skip this writing and go directly to the relevant file to find this object there);

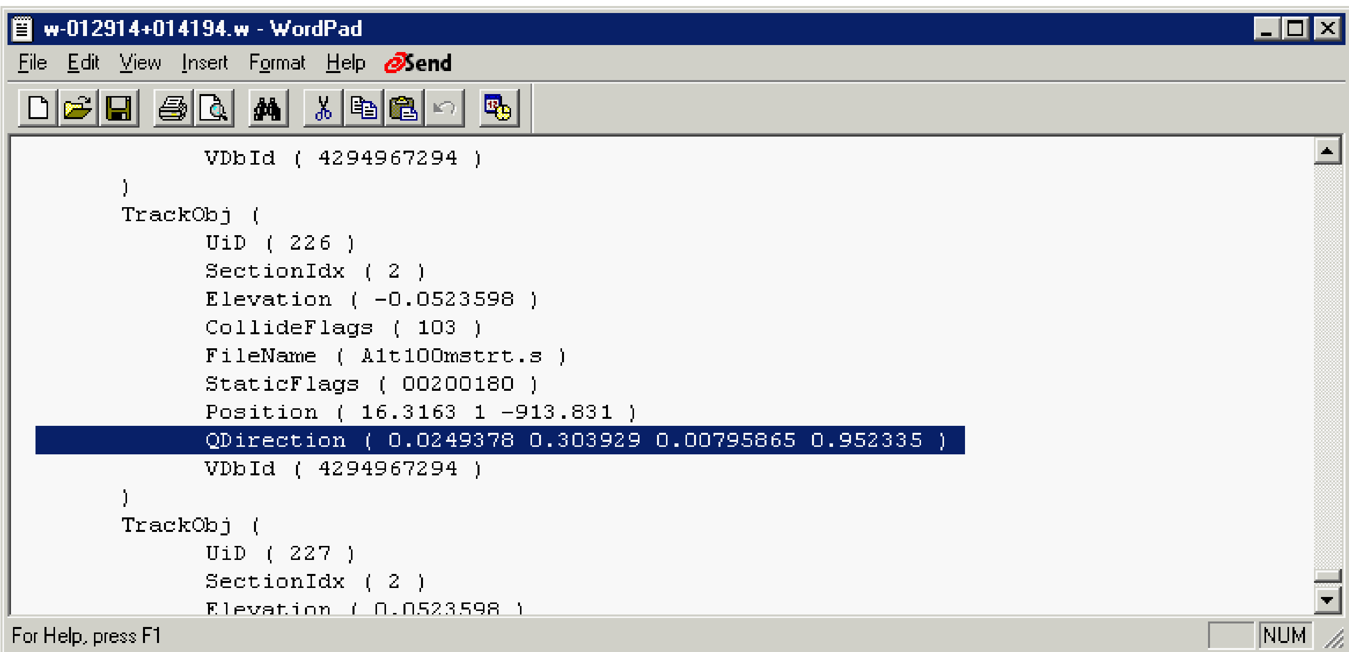
FIGURE 3.1. A BRIDGE PLACED NEAR SLOPING TRACK:
the bridge is JP2bluebrg.s, placed in its default orientation near straight track sloped 3°;
shown in orange is how to read off the tile numbers, and in red how to read off the track Position



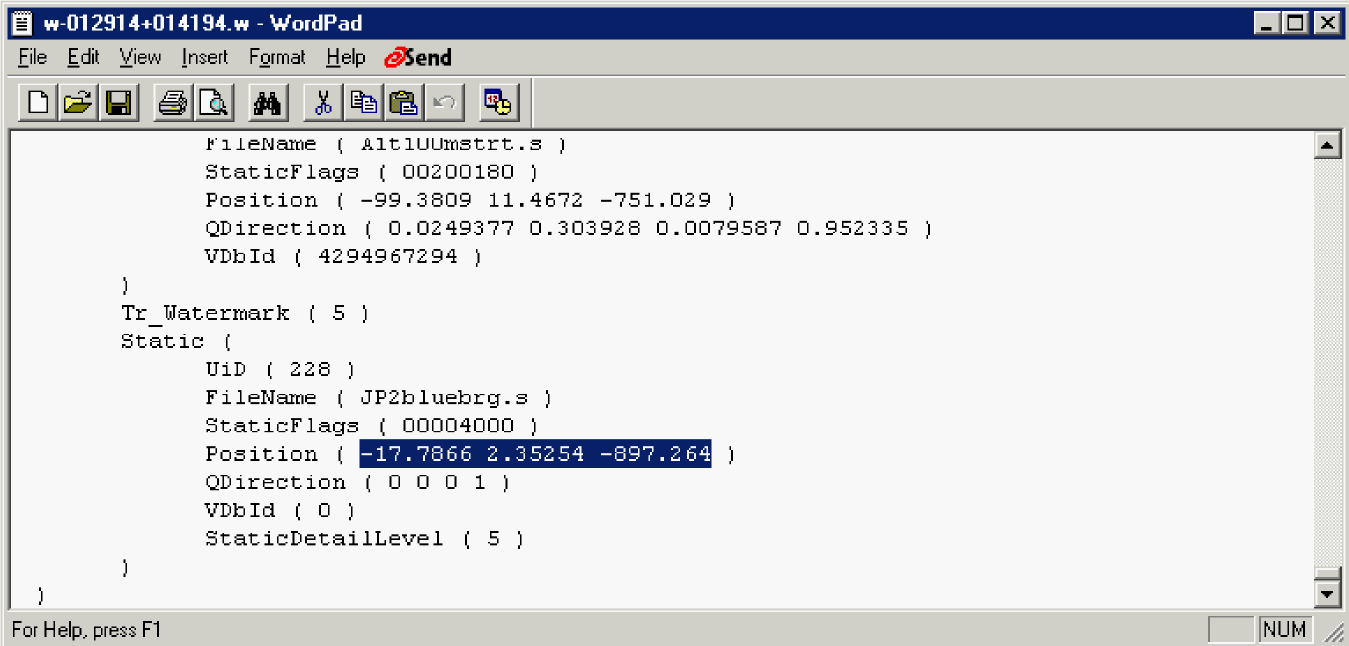
9) go to your route's WORLD folder and find the *.w file that contains the two tile numbers that you noted: in this example, the file is called w-012914+014194.w;
10) open the *.w file with a Unicode-capable text editor (like WordPad);
11) search the *.w file for the line containing the x value of the track section, as shown in the next illustration (note that the number of digits after the decimal points can vary!);
12) check that the object's "FileName", and its y and z values match those of the track section that you selected;
13) if they do not match, continue searching in the *.w file until you find the desired track section;



14) now select the line containing the track section's QDirection, highlighted in the next figure: copy the entire line with Ctrl-C;

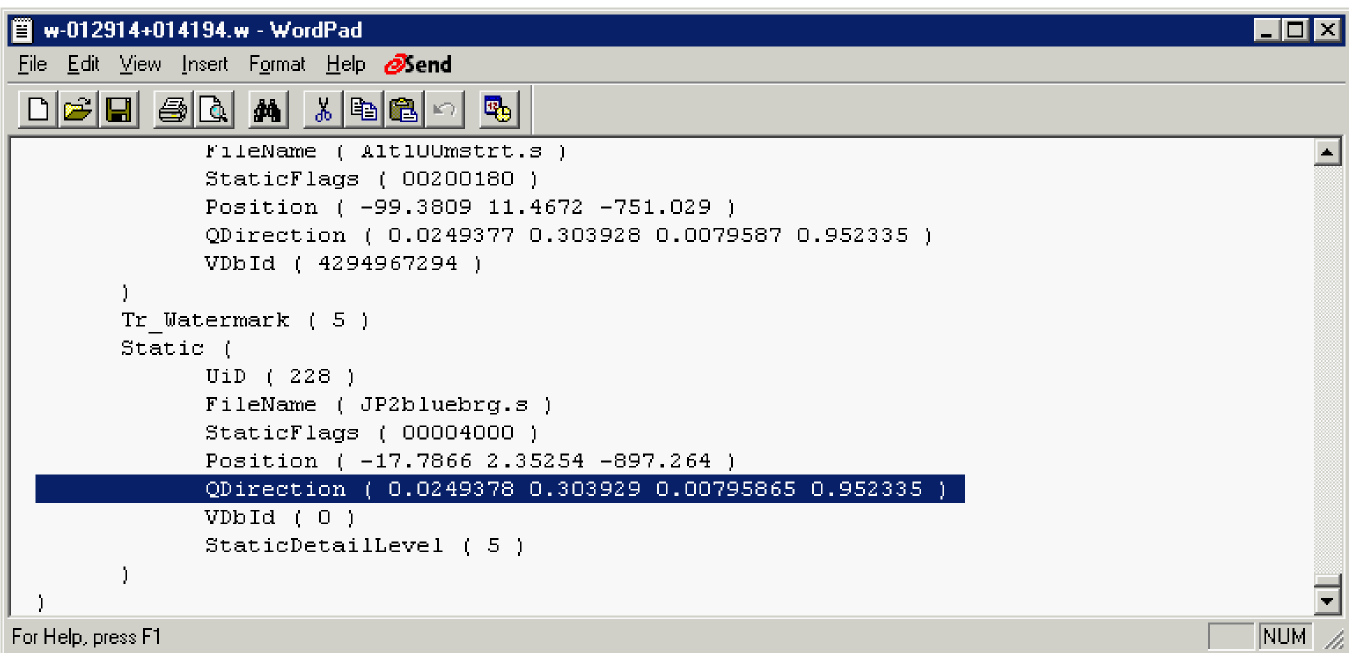


15) go back to the Route Editor;
16) select the bridge section that you placed earlier;
17) note the position of this bridge section: it is again given as "x", "y", "z" in the Object window of the Route Editor (not illustrated, because very similar to the case shown above for the track section);
18) return to editing the same *.w file;
19) search the *.w file for the line containing the x value of the bridge section, as shown in the next illustration;
20) check that the object's "FileName", and its y and z values match those of the bridge section that you selected;
21) if they do not match, continue searching in the *.w file until you find the desired bridge section;




```
                FileName ( A1t100mstrt.s )
                StaticFlags ( 00200180 )
                Position ( -99.3809 11.4672 -751.029 )
                QDirection ( 0.0249377 0.303928 0.0079587 0.952335 )
                VDbId ( 4294967294 )
        )
        Tr_Watermark ( 5 )
        Static (
                UiD ( 228 )
                FileName ( JP2bluebrg.s )
                StaticFlags ( 00004000 )
                Position ( -17.7866 2.35254 -897.264 )
                QDirection ( 0 0 0 1 )
                VDbId ( 0 )
                StaticDetailLevel ( 5 )
        )
)
```

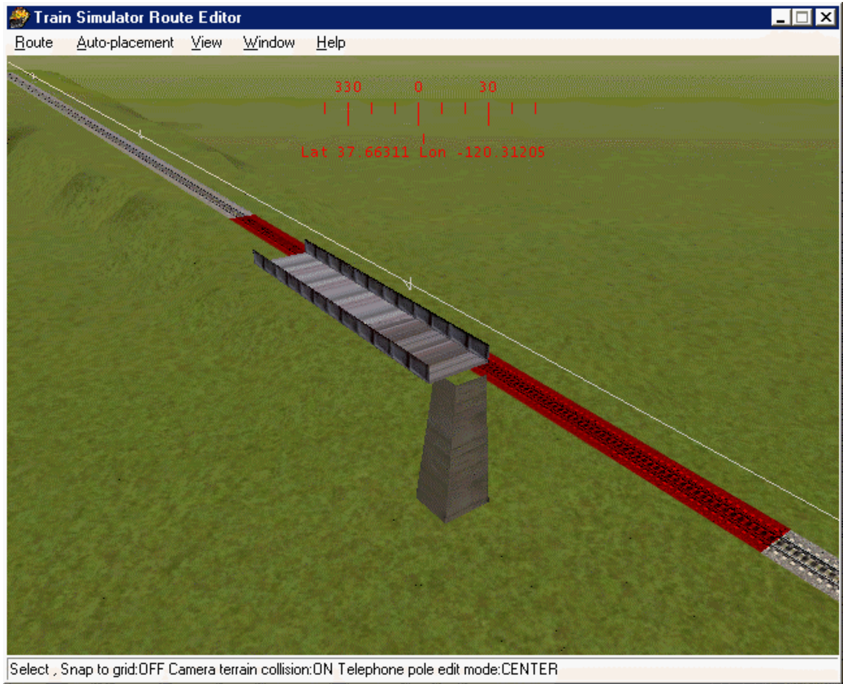
22) now overwrite the line containing this bridge's QDirection with the line that you copied before: paste using Ctrl-V;
23) the result should look like this (with the QDirection data from the track section now placed in the QDirection data of the bridge section):




```
                FileName ( A1t100mstrt.s )
                StaticFlags ( 00200180 )
                Position ( -99.3809 11.4672 -751.029 )
                QDirection ( 0.0249377 0.303928 0.0079587 0.952335 )
                VDbId ( 4294967294 )
        )
        Tr_Watermark ( 5 )
        Static (
                UiD ( 228 )
                FileName ( JP2bluebrg.s )
                StaticFlags ( 00004000 )
                Position ( -17.7866 2.35254 -897.264 )
                QDirection ( 0.0249378 0.303929 0.00795865 0.952335 )
                VDbId ( 0 )
                StaticDetailLevel ( 5 )
        )
)
```

24) save the *.w file;
25) return to the Route Editor;
26) deselect any objects;
27) reload the route in the Route Editor, BUT: do NOT save world placement changes or terrain changes when asked! (if you do save these, you will lose your reorientation of the bridge section); IMPORTANT: if the object that you set parallel is itself a track or road section, you must reload the route with the Advanced option "Rebuild Track Database"!
28) if you now return to the same camera viewpoint, you should see something like the following figure 3.2, in which the bridge section is precisely aligned with the track, but not yet placed under the track;

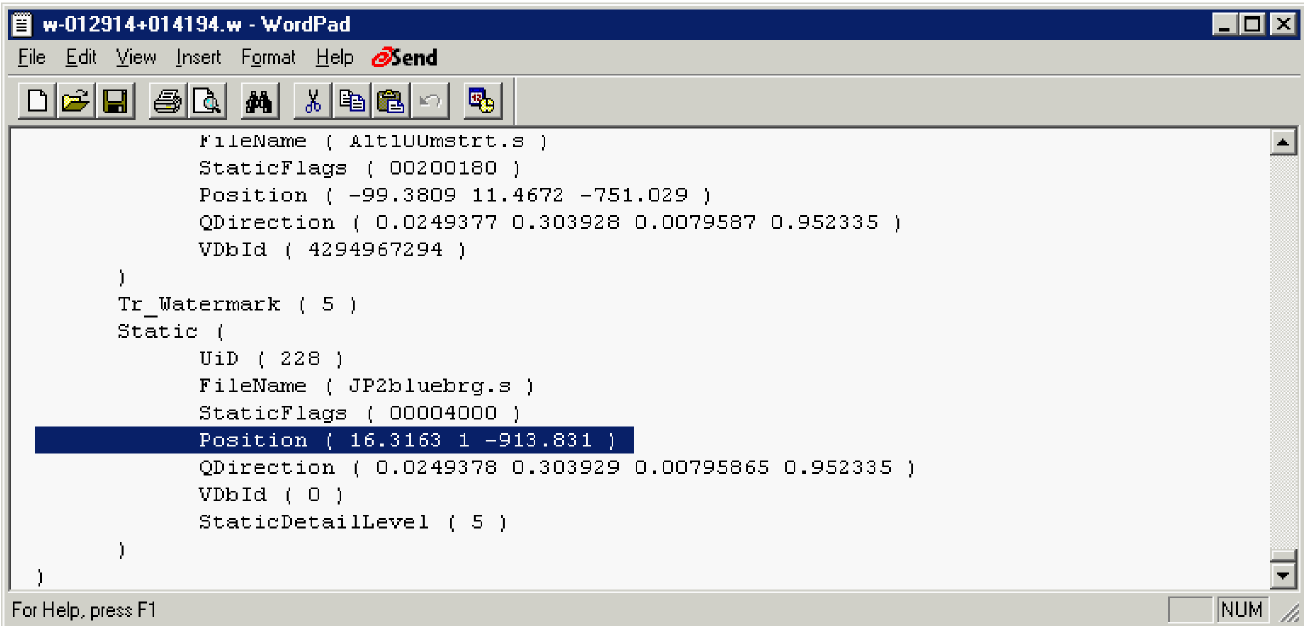
FIGURE 3.2. A BRIDGE SET PARALLEL TO SLOPING TRACK:
the bridge is JP2bluebrg.s, set parallel to straight track sloped 3°; note the stylized arrowhead in the white line over the track sections, pointing to top left, and indicating the direction of the track sections



29) next you should move the bridge section (without any rotations!) until it fits under the track where you want it: press F3, and move the bridge section as usual - this can be done visually with good precision;
30) save the route as usual.

If you need better than visual precision in placing the bridge section (so it is exactly centered on the tracks), you may try to copy the track's position over the bridge's position, as detailed next. As you will see, this will not always do exactly what you want, however!

31) in the same *.w file, copy the line containing the track section's Position over the line containing the bridge section's Position, as illustrated next;

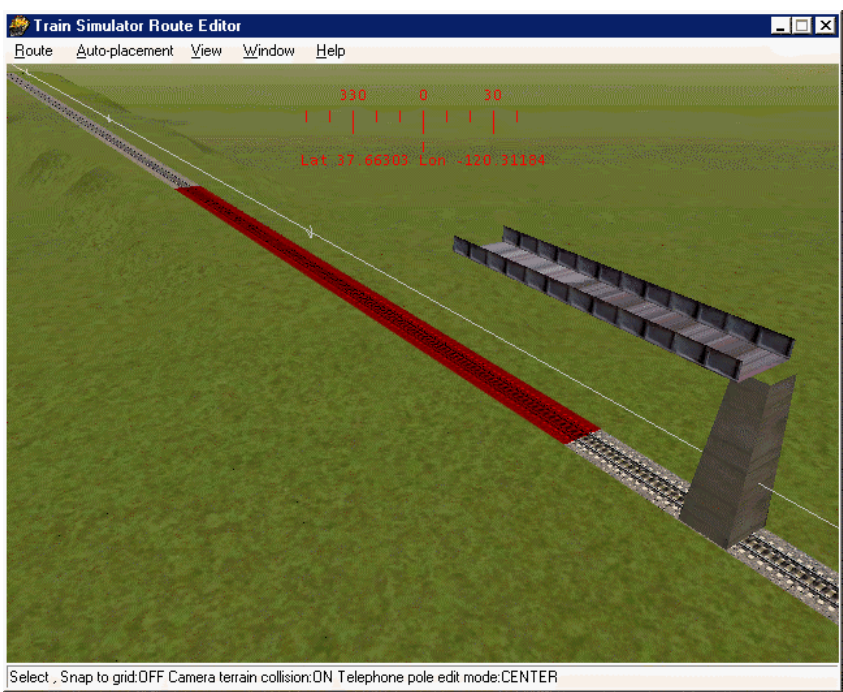


```
w-012914+014194.w - WordPad
File  Edit  View  Insert  Format  Help  Send
            FileName ( Alt1UUmstrt.s )
            StaticFlags ( 00200180 )
            Position ( -99.3809 11.4672 -751.029 )
            QDirection ( 0.0249377 0.303928 0.0079587 0.952335 )
            VDbId ( 4294967294 )
      )
      Tr_Watermark ( 5 )
      Static (
            UiD ( 228 )
            FileName ( JP2bluebrg.s )
            StaticFlags ( 00004000 )
            Position ( 16.3163 1 -913.831 )
            QDirection ( 0.0249378 0.303929 0.00795865 0.952335 )
            VDbId ( 0 )
            StaticDetailLevel ( 5 )
      )
)
For Help, press F1                                    NUM
```

32) save the *.w file;
33) return to the Route Editor;
34) deselect any objects;
35) reload the route in the Route Editor (do NOT save world placement changes or terrain changes when asked);
IMPORTANT: if the object that you set parallel is itself a track or road section, you must reload the route with the Advanced option "Rebuild Track Database"!

The result will be as shown in figure 3.3. What has happened is that the bridge's pivot point (its center of rotation in MSTS) has been placed at the same position as the track's pivot point; however, the bridge's pivot point is at its bottom center, while the track's pivot point is at its end (at the right end of the red-highlighted track section).

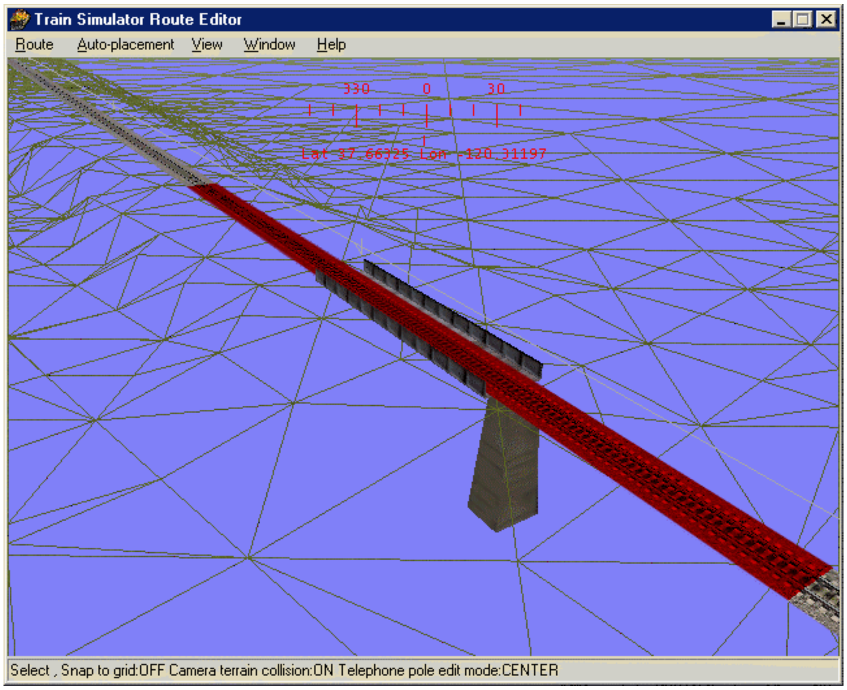
FIGURE 3.3. A BRIDGE SET PARALLEL AND CO-LOCATED TO SLOPING TRACK:
the bridge has been given the Position coordinates of the red-highlighted track section;
the bridge's pivot point is located at the track section's starting point



So, in this case, you still need to push the bridge down to match the track altitude. Furthermore, you may still want to move the bridge along the track, depending on other scenery. This is most easily done visually! So it is usually easier (and satisfactory) to skip the step of copying the track's position over the bridge's position, and make a "manual" shift instead.

That produces the result in the figure 3.4, using wire mode to better show the bridge, which is now mostly underground. (Remember to change the bridge properties to Terrain object. You may also raise the bridge a bit so it hides the track ballast.)

FIGURE 3.4. A BRIDGE SET PARALLEL AND MOVED BELOW SLOPING TRACK:
the bridge has been moved manually under the track section



Other objects, like platforms, may need to be shifted sideways anyway, so a "manual" placement is usually best.

4. SETTING OBJECTS ANTIPARALLEL

USAGE: **- learn from the detailed instructions below;**
- AND follow the instructions on the <u>Set antiparallel sheet</u> of Object Rotator.

There are cases where you want to orient an object in the opposite direction to a track or road. For example, in the above case of a bridge section under a track, you may want the bridge section to point down-slope instead of up-slope along the track. Also, if you want to place platforms along both sides of a track, you may need to orient the platform on one side of the track in the opposite direction compared to the platform on the other side, so the track side of the platform faces the tracks in both cases.

I call this opposite direction antiparallel, as opposed to parallel.

You can do this in one of two ways: by editing a *.w file with WordPad or by using the Object Rotator software. We will discuss both options further below.

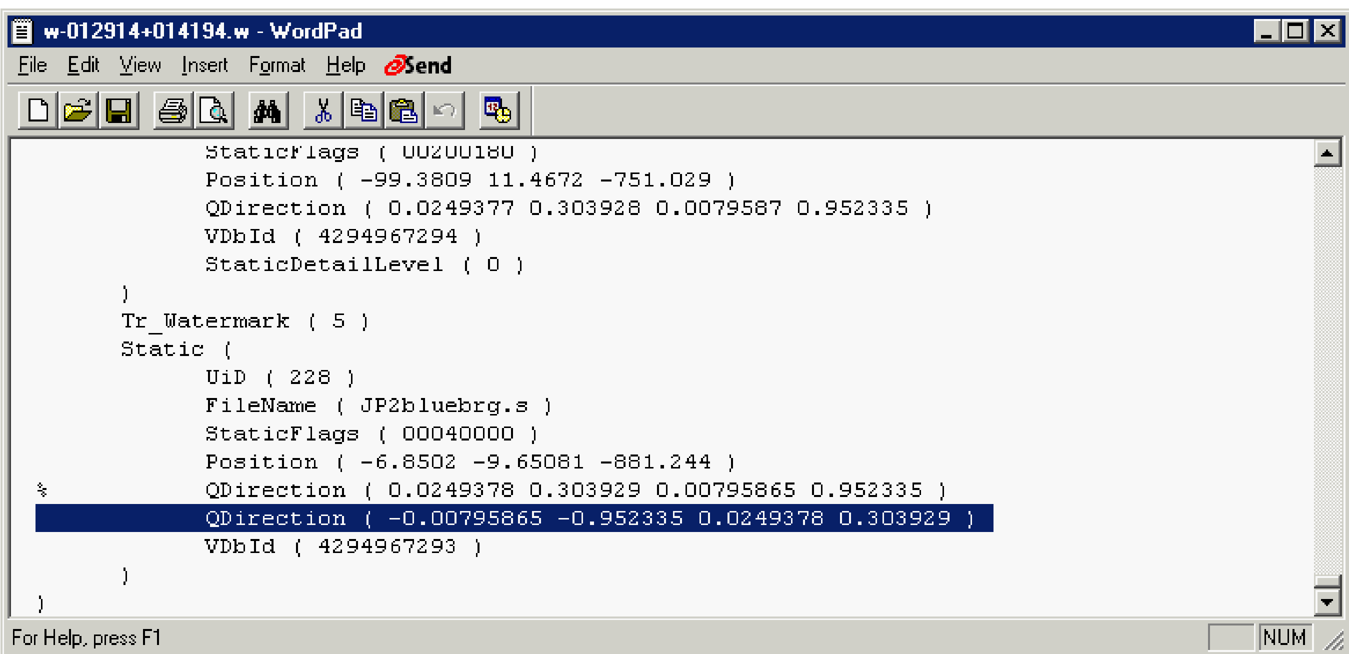
To reorient an object to make it antiparallel requires changing the individual data within the object's QDirection line in the appropriate *.w file. The general procedure for reorienting an object "nose to tail" so that its slope and bank are reversed, is to change its QDirection data from

```
QDirection ( a b c d )
```

to

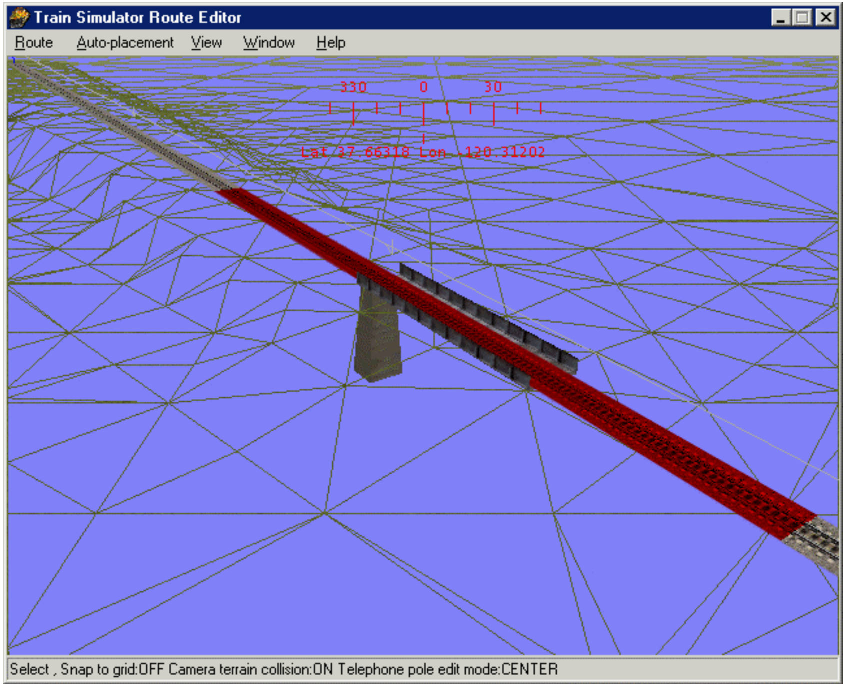
```
QDirection ( -c -d a b )
```

For example, using the parallel bridge placed in the last section, it can be set antiparallel as shown next (where the old line has been commented out with % to show how it looked before setting antiparallel).



The result will be similar to the figure 4.1: the "nose" of the bridge now points down-slope along the track, instead of up-slope. Note that it may be necessary to readjust the bridge altitude after this rotation.

FIGURE 4.1. A BRIDGE SET ANTIPARALLEL AND MOVED BELOW SLOPING TRACK:
the bridge has been moved manually under the track section



How do you know when objects are parallel vs. antiparallel? All objects have a well-defined "nose" direction. All objects orient their "nose" to the north by default when placed in RE (with the exception of track and road sections that connect to existing tracks or roads). For track and road sections, the "nose" direction is made visible in RE by stylized arrowheads that are part of the overhead white wires: in the above pictures you may barely see some of those arrowheads pointing up-slope. When the "noses" of objects point in the same direction, they are parallel; when their noses point in opposite directions, they are antiparallel.

Setting antiparallel by editing

Usually, the quickest way to set antiparallel an object is to directly edit the QDirection data in its *.w file. Assuming that you have set parallel the object as described in the previous section, do the following:

1) find the QDirection line of the track or object in its *.w file (as described above under Setting Objects Parallel);
2) make the change shown above by moving the last two numbers of QDirection in front of the first two numbers, thus getting

```
QDirection ( c d a b );
```

3) add negative signs before the first two numbers, getting

```
QDirection ( -c -d a b )
```

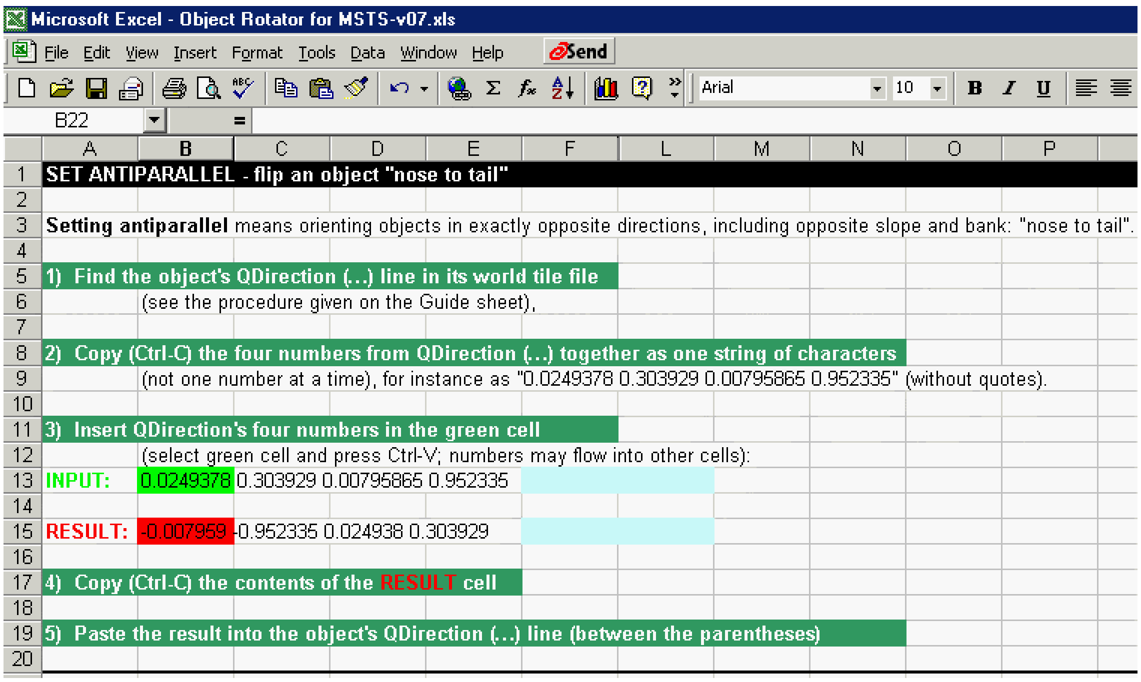
4) save the *.w file;
5) return to the Route Editor;
6) deselect any objects;
7) reload the route in the Route Editor (do NOT save world placement changes or terrain changes when asked!);
IMPORTANT: if the object that you set antiparallel is itself a track or road section, you must reload the route with the Advanced option "Rebuild Track Database"!
8) adjust the object's altitude (and perhaps position) if needed;
9) save the route as usual.

Setting antiparallel with Object Rotator

If you don't want to edit the QDirection numbers as described above, you can let Object Rotator do it for you. (But you will find that this approach is not quicker!)

For illustration, we here take the same case of a bridge section to be placed along a track section, but now antiparallel. The process will be very similar to that given above to set an object parallel, but will involve use of the software Object Rotator at one point. We assume that you have already placed your track and bridge sections in the route and saced the route. Proceed as follows:

1) find the QDirection line of the track or object in its *.w file (as described above under Setting Objects Parallel);
2) copy (using Ctrl-C) its QDirection data, BUT copy just the four numbers together as a SINGLE unit, for example copy EXACTLY the part between the quotes: `"0.0249378 0.303929 0.00795865 0.952335"`
3) start up **Object Rotator** (by starting Excel);
4) select the **Set antiparallel sheet** (using the tabs at the bottom): it is pictured below;
5) point at the green cell labeled INPUT, and press Ctrl-V to paste those four numbers together into that single green cell, as illustrated next;



6) move down the Excel sheet to the red cell labeled RESULT; it should show a set of four numbers (spilling out of the cell to the right);
7) click on that one red cell, and copy its contents with Ctrl-C (this copies all four numbers at the same time);
8) return to editing the *.w file;
9) find the lines for your bridge section (as described above under Setting Objects Parallel);
10) paste the result into the QDirection line of the bridge section, INSIDE the parentheses (leaving spaces where they were before; don't worry about character formatting);
11) save the *.w file;
12) return to the Route Editor;
13) deselect any objects;
14) reload the route in the Route Editor (do NOT save world placement changes or terrain changes when asked!); IMPORTANT: if the object that you set antiparallel is itself a track or road section, you must reload the route with the Advanced option "Rebuild Track Database"!
15) adjust the object's altitude (and perhaps position) if needed;
16) save the route as usual.

5. STANDING OBJECTS VERTICAL

USAGE: **- read overview below;**
- AND follow the instructions on the Stand vertical sheet of Object Rotator.

You may want to rotate a tilted object to make it stand vertical, perpendicular to a horizontal plane, without changing its compass direction.

For example, you may have placed a lamp, a house or any other object that normally stands vertical, but you may have changed its orientation, on purpose or accidentally, so it now is tilting unrealistically. Then you will want to remove the tilt by setting the object vertical.

Note that pressing O in the RE will also set the object vertical, but with its default compass direction (nose pointing north): so if the compass orientation is not important, pressing O in RE is much simpler than using Object Rotator.

And pressing N in the RE sets the object perpendicular to the terrain it stands on, while keeping its compass direction: this will result immediately in a vertical object if the terrain is perfectly horizontal, with the desired compass direction; if the terrain is not perfectly horizontal, you can press O in RE (see above), or use Object Rotator.

The procedure for doing this is very similar to that for setting objects antiparallel to each other, except that you now use the object's own QDirection data: these are then modified by the **Stand vertical sheet of Object Rotator**, so that you can copy the modified values back into the object's *.w file.

Just follow the instructions in the **Stand vertical sheet of Object Rotator**.

6. STANDING OBJECTS PERPENDICULAR

USAGE: **- read overview below;**
- AND follow the instructions on the Stand perpendicular sheet of Object Rotator.

You may want to rotate an object to make it stand perpendicular to another, without changing its compass direction.

For example, you may want to tilt an object (such as a bridge) so its stands exactly perpendicular to the plane of a track or road section, without losing its compass orientation. Note that this can give the object a bank (a sideways tilt) that may not be realistic for a bridge.

Pressing N in the RE sets the object perpendicular to the terrain it stands on, but that terrain will not normally be exactly parallel to a track or road, when slopes are involved.

The procedure for doing this is similar to that for standing objects vertical, except that you now need both the object's own QDirection data and the Qdirection data of the other object: the QDirection data of the object to be adjusted are then modified by the **Stand perpendicular sheet of Object Rotator**, so that you can copy the modified values back into the object's *.w file.

Follow the instructions in the **Stand perpendicular sheet of Object Rotator**.

7. LAYING OBJECTS ACROSS EACH OTHER

USAGE: - learn from the detailed instructions below;
- AND follow the instructions on the Lay across sheet of Object Rotator.

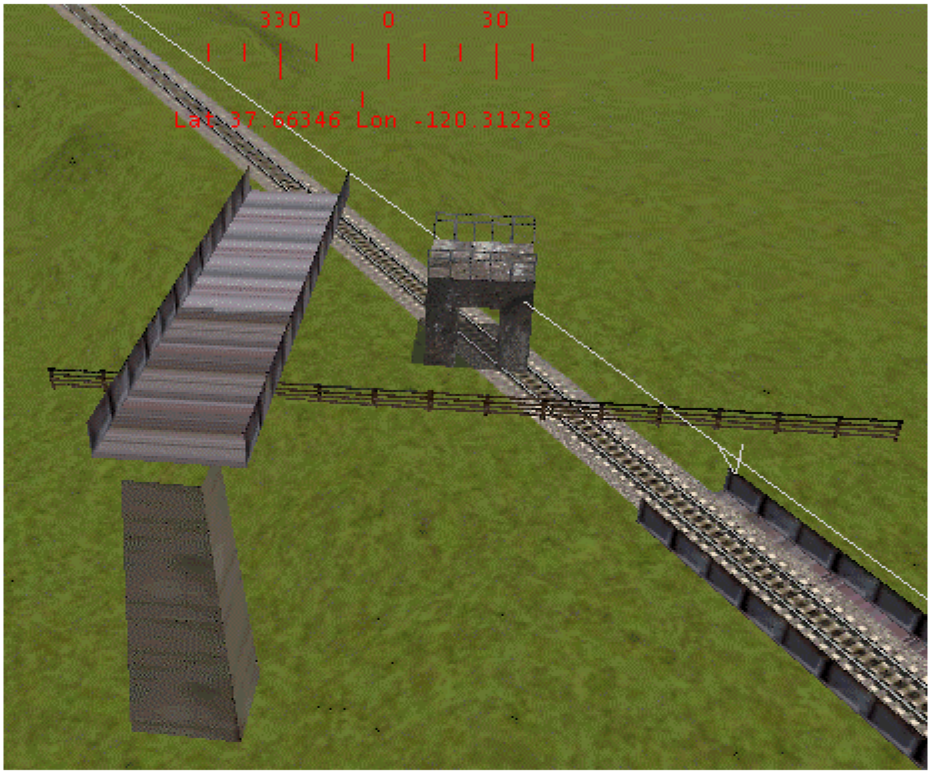
You may want to place one object exactly "cross-wise" across another, such as a road bridge (shown at left in figure 7.1), telephone poles or lamps exactly across a track. These should stand vertical, but others may need to be tilted to fit the slope of the track, if it is sloped.

Some MSTS objects have their "nose" oriented in unexpected directions: when placed in the Route Editor they are by default oriented "cross-wise", that is perpendicular to the northerly direction you would expect.

All of this is done with the **Lay across sheet of Object Rotator**.

Examples of cross-wise objects include fences and some bridges, as illustrated in figure 7.1: the two bridges and the fence are placed with their default orientations, with "nose" pointing north. The left bridge is the same as used in previous sections, and simply points north. However, the small bridge has its top and railings oriented east/west, and the fence has its length oriented east/west. To align these "crosswise" objects to sloping track (or road) requires turning them by 90º and usually also requires making them bank (tilt) sideways to fit the track or road slope.

FIGURE 7.1. BRIDGES AND A FENCE PLACED NEAR SLOPING TRACK:
the large bridge at left is JP2bluebrg.s, the small bridge is OE_Bridge09.s, the fence is BarFence50m.s;
they are placed in their default orientation near straight track sloped 3°; their "nose" points north;
(a parallel-oriented bridge section is shown at lower right)



Other objects that are oriented east/west by default can include gantries and telephone poles (don't use the MSTS telepoles, because they can cause serious trouble!). These objects should however remain vertical, even when the track or road is sloped, so they require a different treatment than the crosswise bridges and fences.

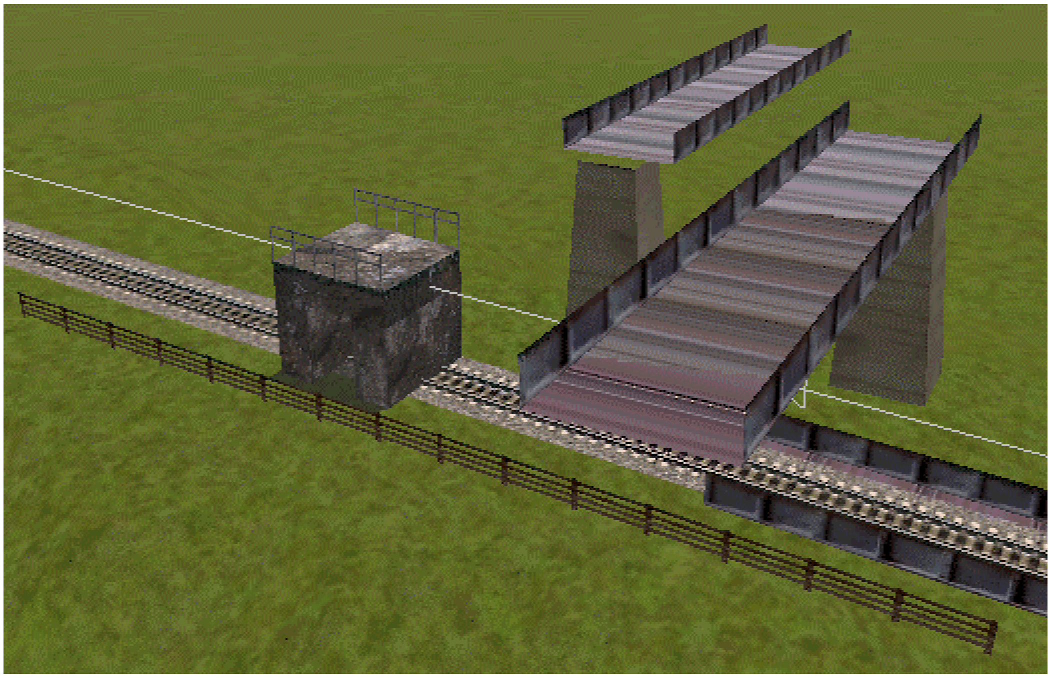
So there are 4 cases to consider: turned 90º to the left or right; and banked or not.

Simple editing of the *.w files cannot do this, even when there is no slope involved: more complex trigonometric calculations are needed, so it is necessary for this purpose to use the **Lay across sheet of the Object Rotator**. The way Object Rotator works to lay objects across each other is to use the track's (or road's) QDirection data to produce the 4 choices of directions: banked and rotated +90º, banked and rotated -90º, vertical and rotated +90º, and vertical and rotated -90º. You may then choose which result you need.

So you use the Lay across sheet of the Object Rotator in much the same way as you do the Set antiparallel sheet: get the QDirection of a track or road section (or other object) from its *.w file, copy that to the Lay across sheet, then select the result that you want and copy it back to the object to be laid across in its *.w file.

The results are illustrated in figure 7.2: the small bridge and fence now follow the slope of the track (the small bridge needs to be pushed below the track); the result is the same whether you choose to rotate them by +90º or -90º because these objects look the same either way. I have added a second copy of the large bridge, to show the effect of rotating it by +90º vs. -90º: these two large bridge sections are kept non-banked, so a road or track on top of those bridges could also stay non-banked. This would also be the correct orientation for vertical gantries and telephone poles.

FIGURE 7.2. BRIDGES AND A FENCE PLACED NEAR SLOPING TRACK:
the large bridge has been duplicated, oriented +90° and -90° relative to the track, but kept vertical;
the small bridge and fence are oriented +90° (or -90°) relative to the track, banked to fit the track slope;
their positions can be adjusted manually



8. LINING UP OBJECTS

USAGE: **- learn from the overview below;**
- AND follow the instructions on <u>Line up sheet</u> of Object Rotator.

Using precise orientations, you can line up a sequence of identical objects to make a perfect straight line. For example, you can line up a series of bridge sections to form a long bridge, or line up platform sections to form a long platform, or line up telephone poles along straight track. This requires great accuracy, because any error in the initial alignment will build up as you drive along the track. Depending on the situation, you may need to set the objects in the line parallel, antiparallel or across, tilting them to match sloping track or road, or keeping them vertical: all these options are offered by Object Rotator.

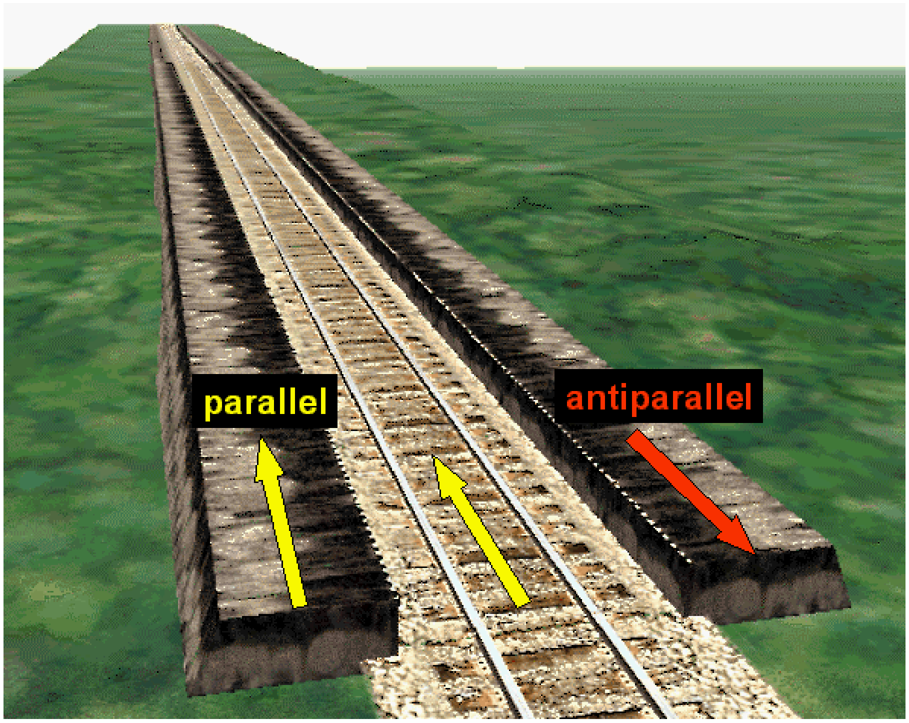
You may also connect track or road sections to each other in a line (this is normally done much faster in the Route Editor). This becomes especially useful when you want the slope of the track or road to be larger than the normal maximum (3° for tracks and 20° for roads), because it is not so easy to make them slope that strongly in the Route Editor. This approach is also convenient for laying tracks in a tunnel: it avoids actually moving track sections below ground, or making the terrain drop temporarily below the tunnel.

The **Line up sheet of Object Rotator** gives you the two ingredients that you need for this: the orientation and the positions of the individual objects. The positions are calculated so that the objects fit end-to-end (tail-to-nose).

Note that some overlap between non-connecting objects like bridges and platforms can be good! Bridge sections usually have piers at both ends: you may want to make piers of successive sections coincide, by using as length of a bridge section the distance between piers at opposite ends of that section.

Figure 8.1 shows two long platforms that are perfectly aligned next to sloping straight track. Although they are 250 m long, the platforms match the track as well at the far end as they do at the near end.

FIGURE 8.1. TWO VERY LONG PLATFORMS FITTING SLOPING STRAIGHT TRACK:
each "platform" has 5 sections of JP2Wall50m.s, sloping 3°,
placed parallel to the track on one side and antiparallel on the other side;
they extend to the horizon in this view (the graphics engine changes their distant appearance)



This case illustrates the value of being able to set objects antiparallel. The "platform" (I actually used a sunken wall in the figure, because MSTS has no convenient platform among its default objects), is "one-sided": it has a "platform edge" on one side, but not on the other side. So, if you select the parallel orientation, you will be able to fit the platform on one side of the track (at the left in the figure), but not on the other side. The antiparallel orientation automatically allows you to make your platform fit the other side of the track.

The procedure for lining up objects is the following (see the Line up sheet of the Object Rotator for more details):

1) determine the length of the objects if you want to join them tail-to-nose: this can be done either by trial and error (guess, try, guess again, try again, etc.), or by measuring the length as explained in the section Measuring the dimensions of objects (see further below);
2) place a number of copies of the object that you want to line up in the current tile, without worrying about their position or orientation (it is best to place more than you will need, since it is easy to delete the surplus later);
3) save the route;
4) as with setting parallel and antiparallel, select an object whose orientation will be used to orient the new objects;
5) find the QDirection of that selected object in its *.w file;
6) copy these QDirection data into Option 1 in part 1) of the Line up sheet of Object Rotator (leave Option 1 selected);
7) enter the length of the object or spacing between objects into part 2);
8) enter the tile numbers of the tile where you start the line of objects into part 3); if you do this, Object Rotator can tell you when objects should be placed in a different tile than the current one, since the line of objects could extend across tile boundaries;
9) enter the start position of the line of objects in part 4); you can get the start position from the *.w file, similarly to the QDirection data that you got before (leave Option 1 selected, unless you want to type in start coordinates by hand, in which case you should select Option 2);
10) now the Line up sheet gives you the results as Positions and QDirections for 12 successive objects in the line: select the Position and the appropriate QDirection data for your situation, and copy them to the objects in the respective *.w files;
11) you will be warned if you need to switch to another tile (the line of objects can easily spill over into another tile): in that case you will have to place new objects for the line on that other tile, save the route, open the *.w file for that tile, and continue copying the Positions and QDirections from Object Rotator to that *.w file;
12) if you need more than 12 objects in the line, repeat the process by copying the Position, Tile and QDirection data of the 12th object to parts 3) and 4) of the Line up sheet, and repeat the process for the next 12 objects.

9. AIMING STRAIGHT (MAINLY FOR TRACKS AND ROADS)

USAGE: **- learn from the overview below;**
- AND follow the instructions on the Aim straight sheet and then the Line up sheet of Object Rotator.

The **Aim straight sheet of Object Rotator** allows you to orient your track or road in the correct direction and/or slope to reach a distant point that you cannot see in the RE: this is particularly useful when the distant point is more than 1 km away, because it disappears in the RE haze or perhaps behind a hill or mountain. And this is especially convenient for building an underground tunnel: together with the Line up sheet, you can lay track underground without having to worry about reshaping the ground as you go. (Just make sure you first place enough track or road sections in the correct tiles, so they add up to cover at least the required distance.)

You can also aim a line of other objects, such as telephone poles, with this method.

NOTE: To connect existing track (of any orientation) to a given point, you should use the **Aim with curve sheet of Object Rotator**: that will first produce a curve to aim in the correct direction.

To aim, the Aim straight sheet needs to know the tile coordinates (not the geographical longitude/latitude) of both the start point and the end point. The Aim straight sheet tells you how to do that even when you only know the geographical longitude/latitude, by placing the camera so it has the required longitude/latitude.

The **Aim straight sheet of Object Rotator** then gives you the QDirection information needed to orient your track or road, including a constant slope if desired. Using the **Line up sheet of Object Rotator**, you then create a line of track or road sections (or other objects).

10. AIMING WITH CURVE (FOR TRACKS ONLY)

USAGE: **- learn from the overview below;**
- AND follow the instructions on the <u>Aim with curve sheet</u> and then the <u>Line up sheet</u> of Object Rotator.

The **Aim with curve sheet of Object Rotator** allows you to connect existing track to a given end point (distant or nearby). It uses an initial dynamical curve in order to aim subsequent straight tracks exactly toward the given point. The curve is held horizontal to avoid unrealistic slope changes, while the straight track may be sloped to reach the given end point at its own altitude.

The **Aim with curve sheet of Object Rotator** differs from the **Aim with curve sheet** in two major ways:
- the **Aim with curve sheet of Object Rotator** only works for tracks (because it requires using dynamic track);
- it allows starting from existing track of any orientation (the **Aim with curve sheet** requires making a joint separately).

NOTE: To aim directly (without an initial curve) at a distant point, or to use roads and other objects, you should use the **Aim straight sheet of Object Rotator**.

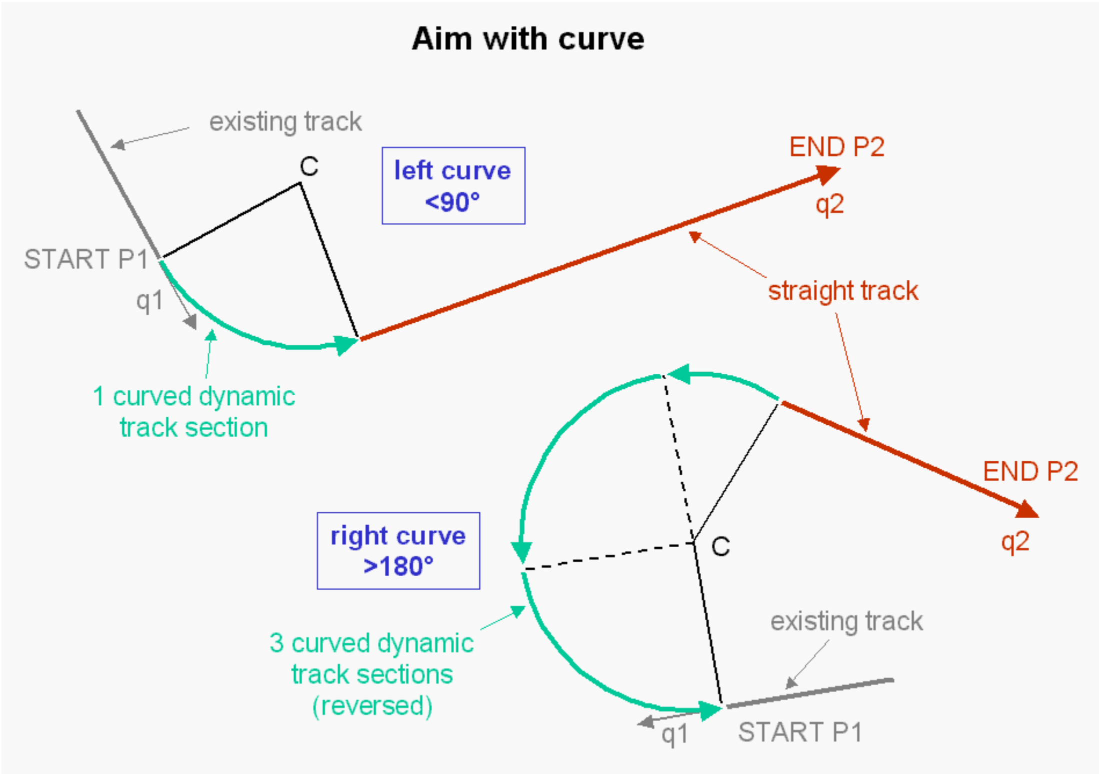
The Aim with curve sheet needs to know the tile coordinates (not the geographical longitude/latitude) of both the start point and the end point; it also needs to know the orientation of the existing track. The Aim with curve sheet tells you how to do that even when you only know the geographical longitude/latitude, by placing the camera so it has the required longitude/latitude.

The **Aim with curve sheet of Object Rotator** then gives you the curve and QDirection information needed to orient your track, including a constant slope if desired. Using the **Line up sheet of Object Rotator**, you then create a line of straight track sections to the given end point.

Figure 10.1 shows two examples, one with a left turn, the other with a right turn, joining a start point P1 to an end point P2. Whether the turn goes left or right is decided by the side on which the given end point is located relative to the existing track. If a turn angle larger than 90° is needed, the curve is broken up into separate 90° and shorter sections (these should NOT be combined into two-curve dynamic tracks sections, because that inserts shorts straight sections between the curves).

CAUTION: Because RE rounds off turn angles in dynamic tracks (to 3 decimal places), it is important to correct the final straight orientation (QDirection q2 in figure 10.1) in order to reach the end point precisely: instructions to do this are given in the **Aim with curve sheet**. This ensures a precision of better than 0.1 m over a distance of 1 km (better than 1:10000).

FIGURE 10.1. AIMING WITH CURVE:
at top left is an example with a left curve, turning less than 90°;
at bottom right is a case with a right curve, turning more than 180°



<u>**11. CURVED TRACK AND ROAD**</u>

USAGE: **- learn from the overview below;**
- AND follow the instructions on the <u>Curved track and road sheet</u> of Object Rotator.

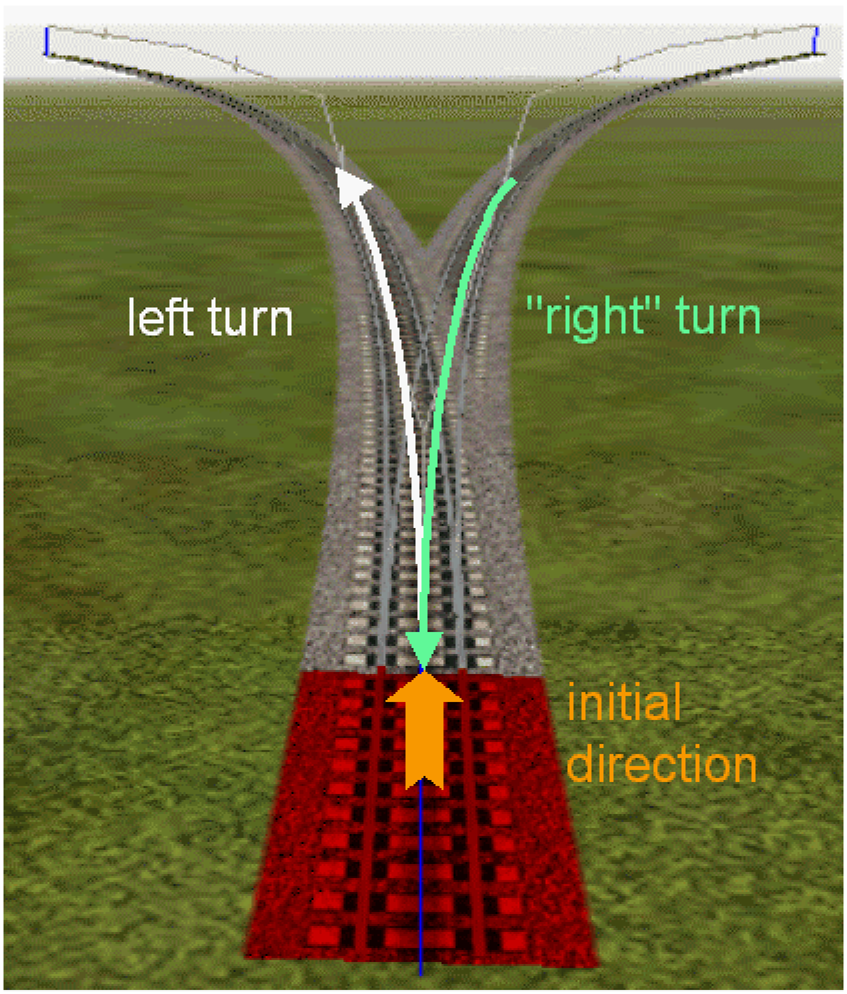
The **Curved track and road sheet of Object Rotator** allows you to place and connect track and road sections along a circle. The tracks may have a constant slope, in which case they will climb along a screw. Normally, this is more easily done by direct placement of track and road sections in RE. But there are some situations where using Object Rotator can be more convenient:

- if the track is underground (in a tunnel), then Object Rotator avoids changing the terrain;
- if the slope of the track or road exceeds the normal RE limits (3° for tracks and 20° for roads), then Object Rotator simplifies the job.

This method is not limited to tracks and roads, but can also be used to place other objects along a curve. However, the objects will be placed end-to-end, which may leave unrealistic gaps: this is further explained in the following section dealing with **Curved row**.

The **Curved track and road sheet of Object Rotator** will give you both a left-turning and a right-turning track, as shown in figure 11.1.

FIGURE 11.1. SLOPING TRACK CURVED LEFT AND RIGHT:
the curve starts after the red-highlighted track section (the terrain is kept flat in this view);
a first <u>left-curved</u> section defines the initial direction needed in Object Rotator;
a right curve is made in MSTS by placing "left-curved" sections, as shown in green



The procedure for placing connected curved track and road sections with the **Curved track and road sheet of Object Rotator** is as follows:

1) we assume that your curved track will continue existing track; if it does not, first place a short straight track section to which you will be able to connect your curve; this is illustrated as the red-highlighted track section in the next picture;
2) place a track or road section at the precise location where you want to start the turn (this section should be the first of the curved sections), by connecting the new section to the existing track or road; make sure this section has the desired slope, but <u>make it turn left</u>, even if you actually want to turn right;
3) place a sufficient number of other identical track or road sections on the route, without worrying about their exact location or direction (it does not matter if they are connected together);
4) in the case of new dynamic tracks, make sure they contain only one curved segment (no straight segment at all), and that they are all identically dimensioned as desired (you cannot adjust their dimensions after they are placed);
5) save the route;
6) determine the initial direction of the curve (shown in the next figure): find the first curved section that you placed in its *.w file, and copy its QDirection data into the Curve T&R sheet of Object Rotator, using option 1 in part 1) [you may alternatively start a fresh (unconnected) curved track or road with option 2 in part 1, following the instructions given there];
7) next you must enter the turn radius and angle of the track or road section in part 2); note that the angle must be positive (you don't need to specify whether the turn will go to the left or right, because results will be given for both cases); also, for dynamic track, the turn angle must be rounded to 3 digits after the decimal point (otherwise, the sections will not connect properly); if you only know the chord and degree of curvature, you can convert that information to turn radius and angle with the simple converter provided in part 2);
8) in part 3), enter the tile numbers (from the camera location), if you expect your track or road to spill over into a neighboring tile;
9) in part 4), enter the start position, from the Position of the first new track or road section (copy it from the *.w file).

Now the resulting Position and QDirection data are given as results for the next dozen track or road sections. Note that there are two sets of results, one set for left curves, and one set for right curves: pick the results that correspond to your choice. For right turns, the sections will automatically be reversed as shown in the next picture (similar to pressing T in RE, which makes "right" turns out of left turns), so you need not worry about this.

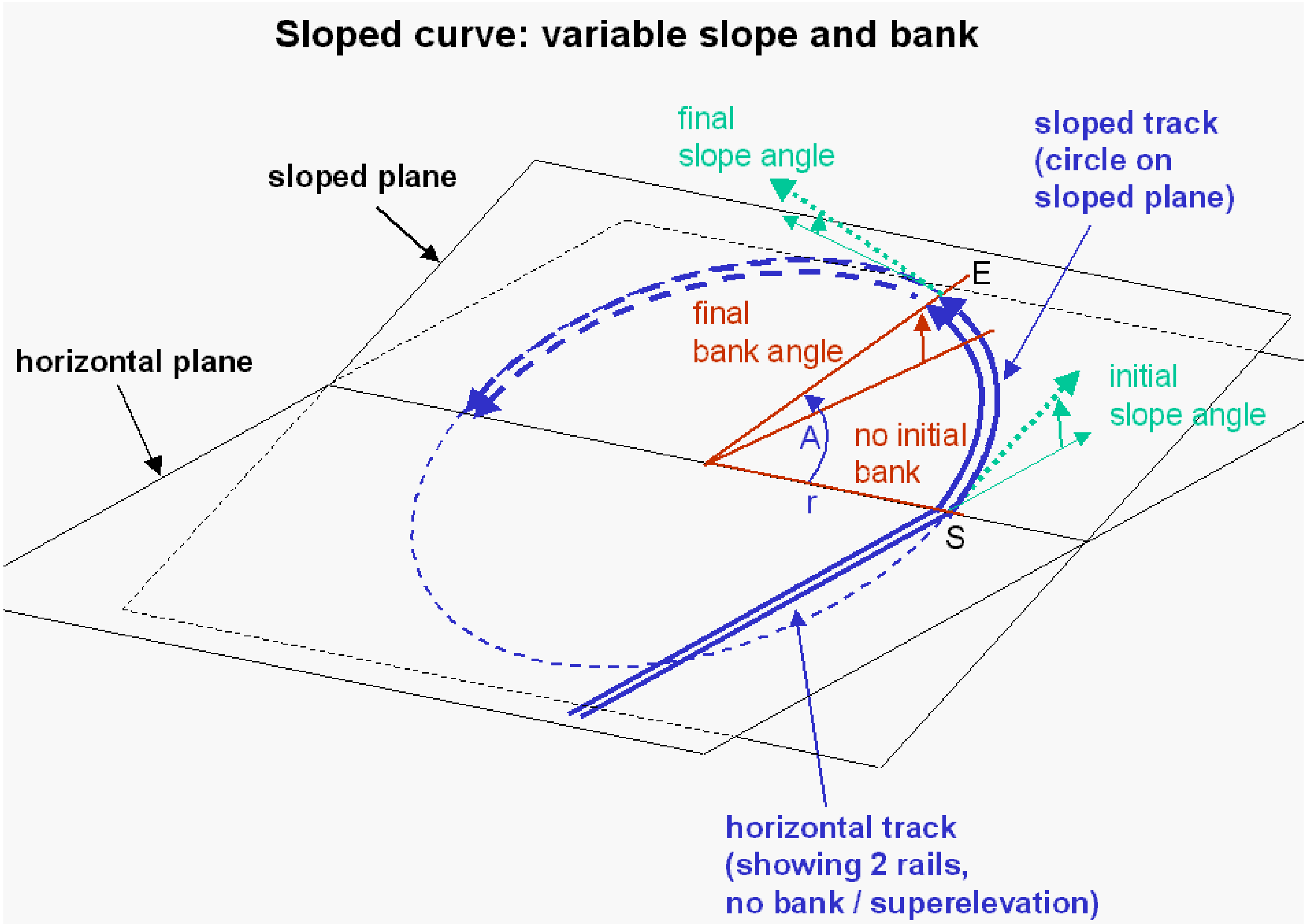
After copying the Position and QDirection data to the appropriate *.w file(s), make sure you reload the route with the option to **rebuild the track database** (also if you are only placing roads).

IMPORTANT NOTE ABOUT SLOPED CURVED TRACKS.

It is important to understand how sloped curved tracks are shaped in MSTS: they are shaped as an arc of a circle on a sloped plane, as illustrated in figure 11.2. If you imagine the curve continuing around a complete circle, it will close the circle at the curve's starting point. A sloped curve starts with an initial slope set in the RE. However, as it curves the slope diminishes until it reaches zero after a turn of 90°; the slope then changes sign and grows until a turn of 180° is completed, when it reaches the negative of the initial slope.

Another effect is important also: sloped regular (non-dynamic) curves also have a changing sideways "bank" (superelevation), as shown in figure 11.2. At its starting point the curve has zero bank by default (no superelevation); but as the track curves it starts to bank, reaching maximum bank after a 90° turn (at that point, where the slope itself has dropped to zero, the bank angle has grown to be equal to the initial slope); then the bank diminishes again, and reaches zero after a 180° turn. This bank effect is actually different with dynamic track: dynamic curves keep a zero bank (superelevation) everywhere along the curve. However, even when there is a bank (as on non-dynamic sloped curves), MSTS makes trains ride as if there were no bank: the track bank is ignored!

FIGURE 11.2. SLOPED CURVING TRACK:
a curve in MSTS is drawn as an arc of circle on a sloped plane,
so that the slope and "bank" (superelevation) of the curved track change from point to point



The importance of this comes into play when you attach another track section at the end of a sloped curving track section, as illustrated in figure 11.3. By default, the new track section will start with zero bank, and whatever slope you give it: suppose you give it the same initial slope as the preceding track section.

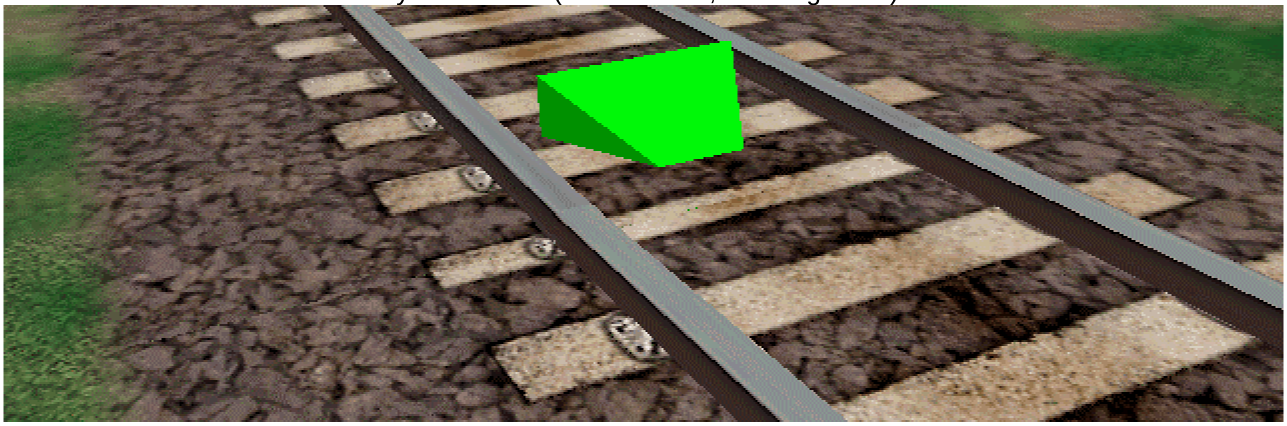
With dynamic sloped curves (top view of figure 11.3), the bank is always zero, so the track joint looks good. Nevertheless, there is a slight change in slope, but it is quite small for small turn angles (figure 11.3 uses 20° turns): in the illustrated case, it is in fact invisible to the eye from all viewing directions. However, for larger turn angles the mismatch grows: for a 90° turn, the slope mismatch would be 3° for 3°-sloped curve sections!

With non-dynamic sloped curve sections, the end point of the first section has a smaller slope and a non-zero bank at the joint, creating a double mismatch: there will be sudden changes in both slope and bank. This is illustrated in the middle panel of figure 11.3. Again, the slope change is invisible for small turn angles, but the bank change is clearly seen. The effect is greatly amplified if you use multiple-track sections, as shown in the bottom panel of figure 11.3: the effect is so large there that one of the tracks does not connect across the joint (as shown by the presence of the blue pole, which indicates that the route's database does not recognize a joint there). This bank mismatch is unavoidable with non-dynamic tracks.

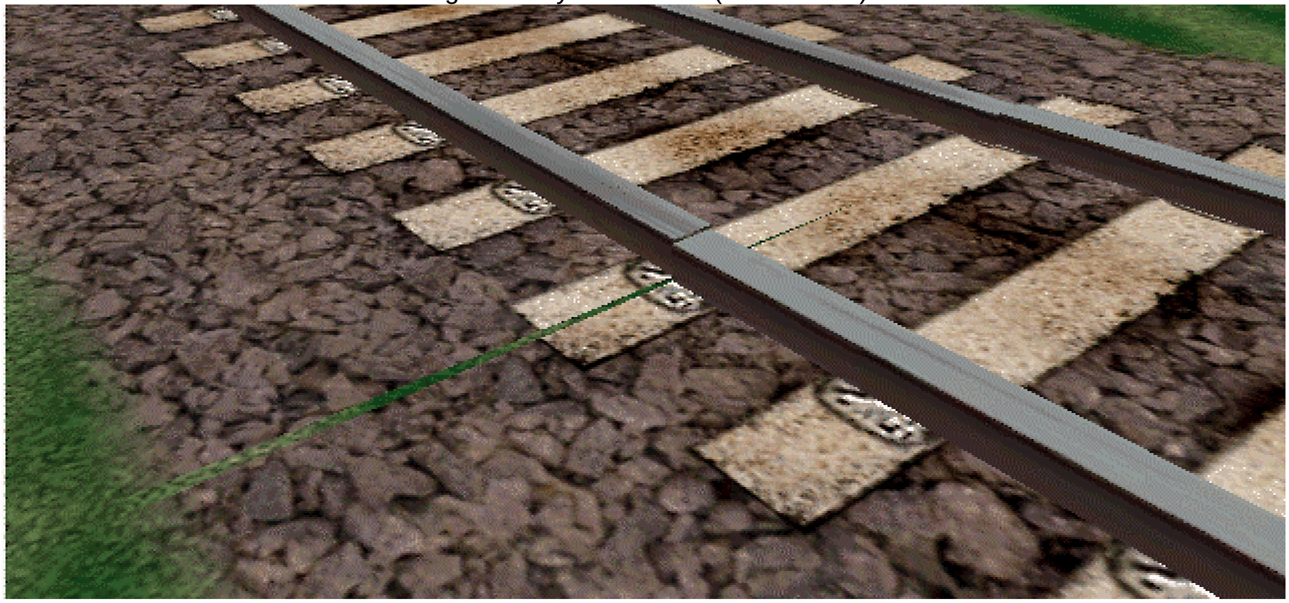
Such joint mismatches can cause trains to derail. To minimize the mismatch in both slope and bank, you should use short curved sections.

FIGURE 11.3. JOINTS IN SLOPED CURVING TRACK:
in each case shown the initial slope is 3°;

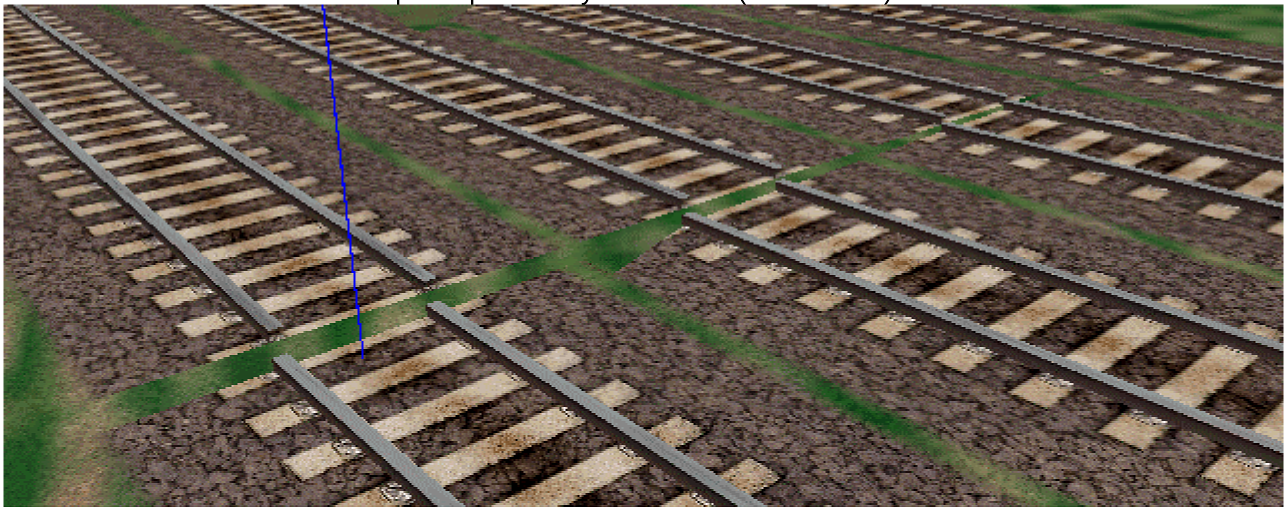
dynamic track (radius 500m, turn angle 20°):



single non-dynamic track (A1t500r20d):



quadruple non-dynamic track (A4t500r20d):



12. CURVED ROW (NON-TRACK AND NON-ROAD)

USAGE: **- learn from the overview below;**
- AND follow the instructions on the Curved row sheet of Object Rotator.

The **Curved row sheet of Object Rotator** allows you to place objects along a curved and sloping track or road (which you may have created with the **Curved track and road sheet**).

The **simplest case is that of track-side or road-side objects that are not joined**, such as telegraph poles, lampposts, or other objects that you want to place at regular intervals and at a constant distance away from the curved track or road (caution: the MSTS "telepole" is dangerous and not suitable for placement with Object Rotator). Just make sure you use a suitable radius (smaller or larger than the track or road radius if the objects are to be placed inside or outside the curve, respectively).

The positions generated by Object Rotator may need adjustment after being transferred to the objects in the Route Editor: typically, the objects will start lined up along the track or road centerline. To move them sideways without losing their orientations and relative positions, select all objects and move them all together (after pressing F3), while also adjusting their height if needed. You may also press H in irregular terrain, so each object adjusts its altitude to the local terrain (without changing its orientation).

Placing objects that should join up along a curve is more complex. This applies to bridge and platform sections, for example. Figure 12.1 illustrates the fitting of a long bridge to a curving sloped track. It is assumed that the slope remains constant: if it continued climbing around a circle, the track (and bridge) would form a screw (helix), mounting on top of itself.

FIGURE 12.1 A SLOPED BRIDGE FIT TO A SLOPED CURVING TRACK:
the bridge sections are JP2bluebrg.s, sloped 3°, placed on a circle of 173 m radius,
using a length of 29.62 m, a width of 6.05 m and an "offset" of 1.045 m;
it is matched to a curve made of A1tEndPnt10dRgt.s sloped 3°;
the white box outlines one bridge section, with its centerline shown in orange;
the blue track centerline is made to swing back and forth across the bridge centerline

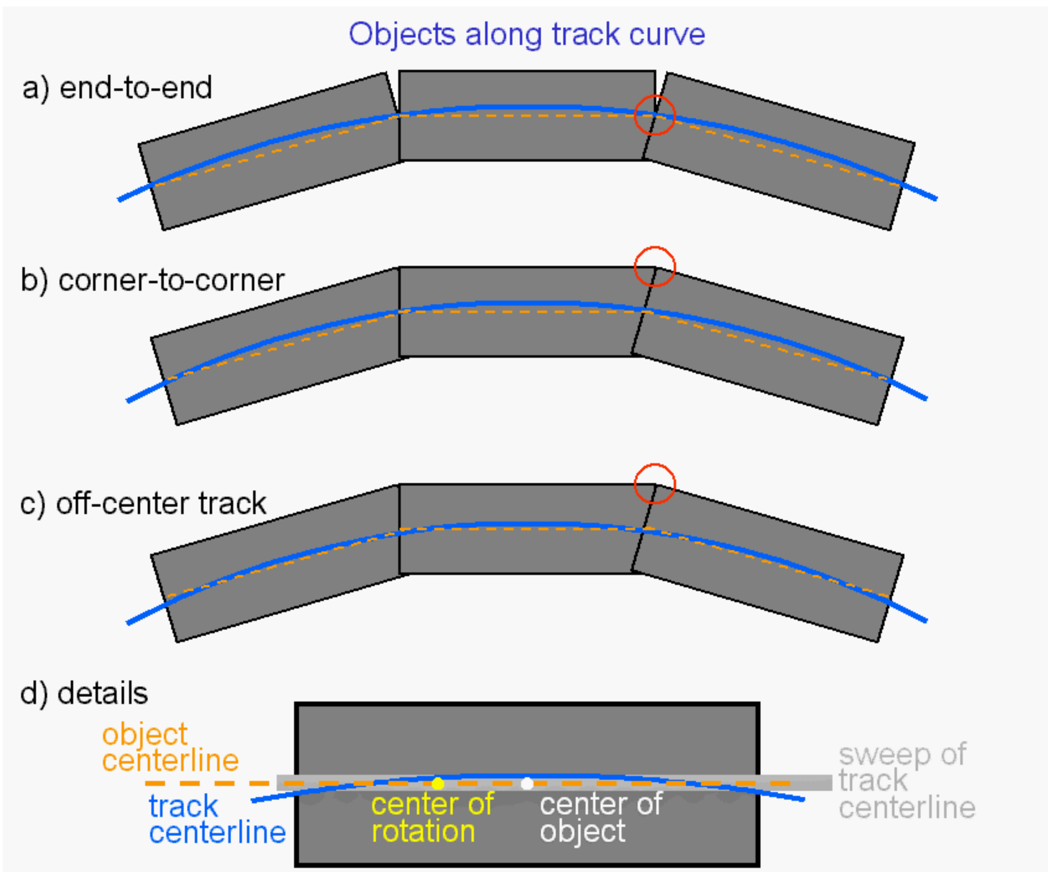
We discuss the procedures in more detail with bridge sections, but much the same applies to platforms, etc.

First, to fit a curve, two successive bridge sections must obviously be rotated by a certain angle between them, visible in the picture above. That angle of rotation depends on the **length of the sections**: therefore you need to provide that object length to Object Rotator. If you want to make bridge piers overlap, you can choose to give the pier-to-pier distance, instead of the overall length of a section. To measure the length of a section, see the section **Measuring the dimensions of objects**.

Second, when two end-to-end bridge sections are rotated with respect to each other to fit a curve, a V-shaped gap opens up between them: this is illustrated in part a) of the next graph, where the red circle shows where the sections are joined. To avoid this gap, the sections must overlap to a greater extent, by joining the corners of the sections, as shown in part b) of the next graph. The amount of overlap required depends in particular on the **width of the bridge section**. Object Rotator asks you to provide that width, and then automatically calculates the required overlap. So, you will also need to measure the width of an object.

Third, as can be seen in figure 12.1, a curving track on a straight bridge section must be allowed to swing to both sides of the bridge section's centerline, so as to make best use of the bridge. Thus, the track should enter a straight bridge section to one side of the section centerline (on the right side in the picture), and then swing over to the other side at mid-span (on the left side in the picture), before curving back to the original side to exit the section. This is also shown in part c) of figure 12.2, and detailed in part d). Object Rotator calculates this lateral offset automatically, in order to make the sweep of the track centerline extend an equal distance to both sides of the object centerline: so you don't have to worry about this. But you must be aware that if the bridge section is too long for a given radius of curvature, the track will swing outside the bridge edges, which is of course unrealistic. To avoid that, you should use shorter or wider bridge sections.

FIGURE 12.2 FITTING OBJECTS ALONG CURVING TRACK:

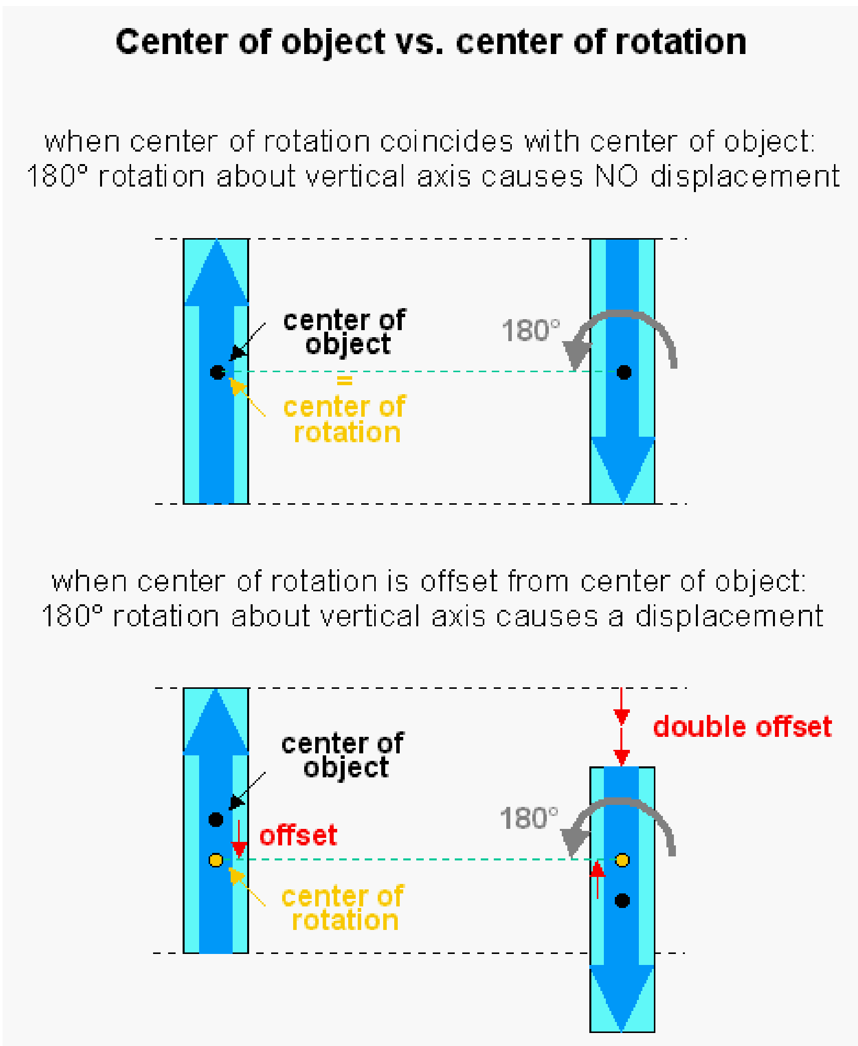


Fourth, the correct alignment of bridge sections depends on where exactly the section's vertical axis of rotation is located (it is labeled as center of rotation in figure 12.2; that is also the object's pivot point): without that information, sections cannot be aligned properly. **Many objects in MSTS have their axis of vertical rotation at their center: this is the easiest case, and is characterized by an "offset" of zero**. This is shown in the top half of the next graph. It is easy to recognize this situation: if rotating the object by 180° around its vertical axis simply exchanges the ends without displacement, the object has zero offset.

Unfortunately, **some objects in MSTS have their axis of vertical rotations "offset" from their center along the centerline towards the "tail" of the object: in that case you have to measure this offset**, as explained next. You recognize this situation by the fact that a 180° rotation around the vertical axis results in a displaced version of the object, as shown in the bottom half of figure 12.3.

FIGURE 12.3. HOW TO RECOGNIZE THAT AN OBJECT'S CENTER OF ROTATION IS OFFSET FROM ITS CENTER POINT:

A blue object is shown at top and bottom left looking vertically down; rotating it 180° about its vertical axis can produce either the result shown at top right or at bottom right; if the result is as shown at top right, the object's center of rotation coincides with its center point; but if the result is as shown at bottom right, the center of rotation is offset by an amount that is half the displacement of the object

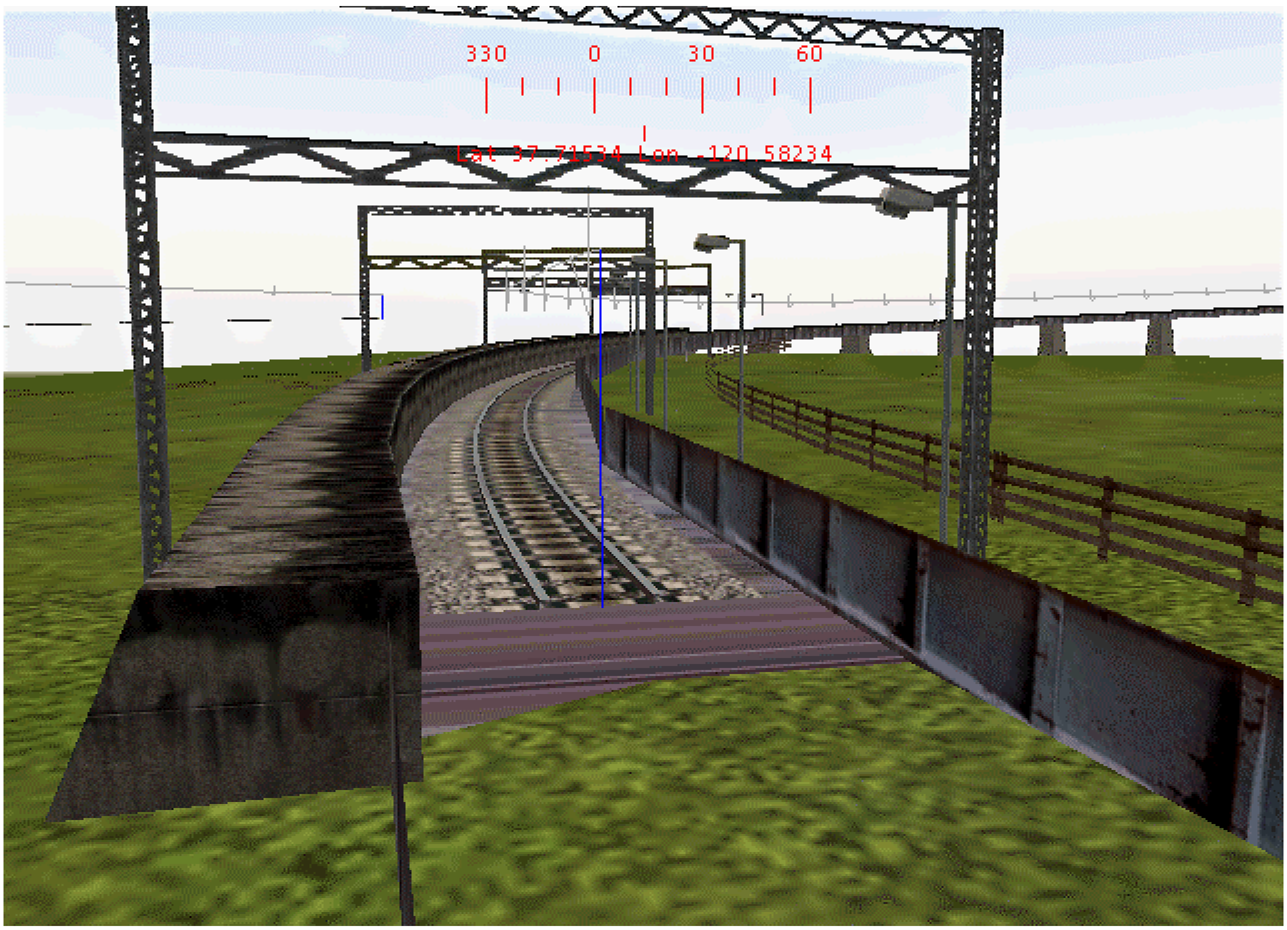


If the object has an offset center of rotations, you must measure it and enter its value in Object Rotator. To measure it, follow the instructions given in the section **Measuring the dimensions of objects**.

In addition to the four complications explained above, you still may want to have the options of setting parallelg, antiparallel, and across, while tilting the objects to fit a slope or while keeping them vertical. For setting parallel, you should place the objects starting from the other end of the row, so they can be set parallel (the reason for this is that the "offset" causes much complication, so that it is simpler to avoid it).

Figure 12.4 shows several such cases along a sloped curving track. The bridge is the same as illustrated before. The wall at left is oriented parallel to the track and has the same slope. The gantries are also oriented "parallel" to the track (because they are built cross-wise to their length), but they are kept vertical. The fence is also built cross-wise to its "nose", so it must be set across to fit alongside the track, while being banked to fit the track slope. Finally, the lamps also need to be set across (for the same reason), but they are kept vertical.

FIGURE 12.4. ROWS OF WALLS, GANTRIES, FENCES AND LAMPS
PLACED NEAR CURVED SLOPING TRACK:
this is the same curved sloping bridge shown above, with incomplete track;
the wall at left is made of JP2Wall10m.s, oriented "Parallel sloped";
the gantries are JP1gantry2.s, oriented "Parallel vertical", spaced 30 m apart;
the fence is made of BarFence10m.s, oriented "Across +90° banked";
the lamps are Za-Lampe2.s (by Joachim Zander), oriented "Across +90° vertical", spaced 10 m apart



The procedure for using the **Curved row sheet of Object Rotator** is the following:

1) determine the length, width and "offset" of the objects if you want to join them tail-to-nose: this can be done as explained in the section Measuring the dimensions of objects (see further below);
2) place a number of copies of the object that you want to place in the current tile, without worrying about their position or orientation (it is best to place more than you will need, since it is easy to delete the surplus later);
3) save the route;
4) select an object whose orientation will be used to orient the first new objects (for instance the nearest track section in the picture above);
5) find the QDirection of that selected object in its *.w file;
6) copy these QDirection data into Option 1 in part 1) of the Curved row sheet of Object Rotator (leave Option 1 selected);

7) enter the curve radius in part 2) of the Curved row sheet; if you use tracks of type A1tEndPnt10d... you can use an approximate radius of about 173m; with tracks of type A1tEndPnt5d..., use a radius of about 869 m;
8) in part 3), select whether the turn will go left or right;
9) enter the length of the object or spacing between objects into part 4);
10) enter the object width and rotation axis offset in part 5); these quantities were measured earlier; use values 0 if you don't need precise joining, or if you use thin objects like fences;
11) enter the tile numbers of the tile where you start the line of objects into part 6); if you do this, Object Rotator can tell you when objects should be placed in a different tile than the current one, since the line of objects could extend across tile boundaries;
12) enter the start position of the line of objects in part 7); you can get the start position from the *.w file, similarly to the QDirection data that you got before (leave Option 1 selected, unless you want to type in start coordinates by hand, in which case you should select Option 2);
13) now the Curved row sheet gives you the results as Positions and QDirections for 12 successive objects in the row: select the Position and the appropriate QDirection data for your situation, and copy them to the objects in the respective *.w files;
14) you will be warned if you need to switch to another tile (the line of objects can easily spill over into another tile): in that case you will have to place new objects for the line on that other tile, save the route, open the *.w file for that tile, and continue copying the Positions and QDirections from Object Rotator to that *.w file;
15) if you need more than 12 objects in the line, repeat the process by copying the Position, Tile and QDirection data of the 12th object to parts 1), 6) and 7) of the Curved row sheet, and repeat the process for the next 12 objects.

13. TURNING ABOUT OBJECT AXES

USAGE: **- learn from the overview below;**
- AND follow the instructions on the Turn about object axes sheet of Object Rotator.

The **Turn about object axes sheet of Object Rotator** allows you to turn an object about any of its 3 internal axes, no matter how they are oriented.

This method is convenient for an object that already has a non-default orientation: you can change its bank, slope or heading by any amount, without changing the other angles. For example, you may change the slope of an object, without changing its bank or heading. By successive application of such rotations, you can also turn the object about the other axes, if desired.

However, this method is not recommended for turning track and road sections that are already connected. To orient and attach track or road sections, use the **Line up** and **Curved track & road sheets** instead of this sheet.

An alternative method for rotating objects about 3 axes is that given in the **Turn about world axes sheet of Object Rotator**. That method rotates an object around the fixed world axes (east/west axis, north/south axis and vertical axis), but is much less intuitive when trying to rotate an object that already has a non-default orientation.

We need to define more precisely what we mean by bank, slope and heading. The analogy with how an airplane pilot sees the plane's orientation is helpful to understand this. In figure 13.1, an object is shown in the form of a solid arrow, which can also be viewed as an airplane. The default orientation is shown in the top half of the picture: the object (or airplane) points straight north (with zero heading), horizontally (with zero slope), and level (with zero bank). A general orientation is shown in the bottom half of the picture: now the object (airplane) points more to the west ("heading"), with a steep grade ("slope"), and with a very slight sideways lean ("bank"). So we define in general:

- **bank is a sideways lean**, like an airplane rolling right or left; in railroading bank is also called **superelevation**; in our case, a **positive bank leans down to the right**, as seen from tail to nose; a bank is the result of a **turn about the object's z-axis** (which points from its tail to its nose); a bank can range from -360 to +360°;

- **slope is an up or down tilt**, like an airplane's nose pointing up or down; in railroading slope is also called **grade**; in our case, a **positive slope moves the nose up** and the tail down; a slope is the result of a **turn about the object's x-axis** (which points from its left "wing" to its right "wing"); a slope can range from -90 to +90°;

- **heading is a horizontal twist**, like a plane's nose pointing right or left; it can also be called compass direction or bearing; in our case, a **positive heading turns the nose to the right** as seen from above, from north to east, then south and west; a heading is the result of a **turn about the object's y-axis** (which points "up" relative to the object's body); a heading can range from -360 to +360°.

So the purpose of the **Turn about object axes sheet of Object Rotator** is to change the bank, slope or heading of an object: you decide by how many degrees you want to change these quantities. For example, you may want to increase the bank by 5°, or the slope by 2°, or decrease the heading by 30°.

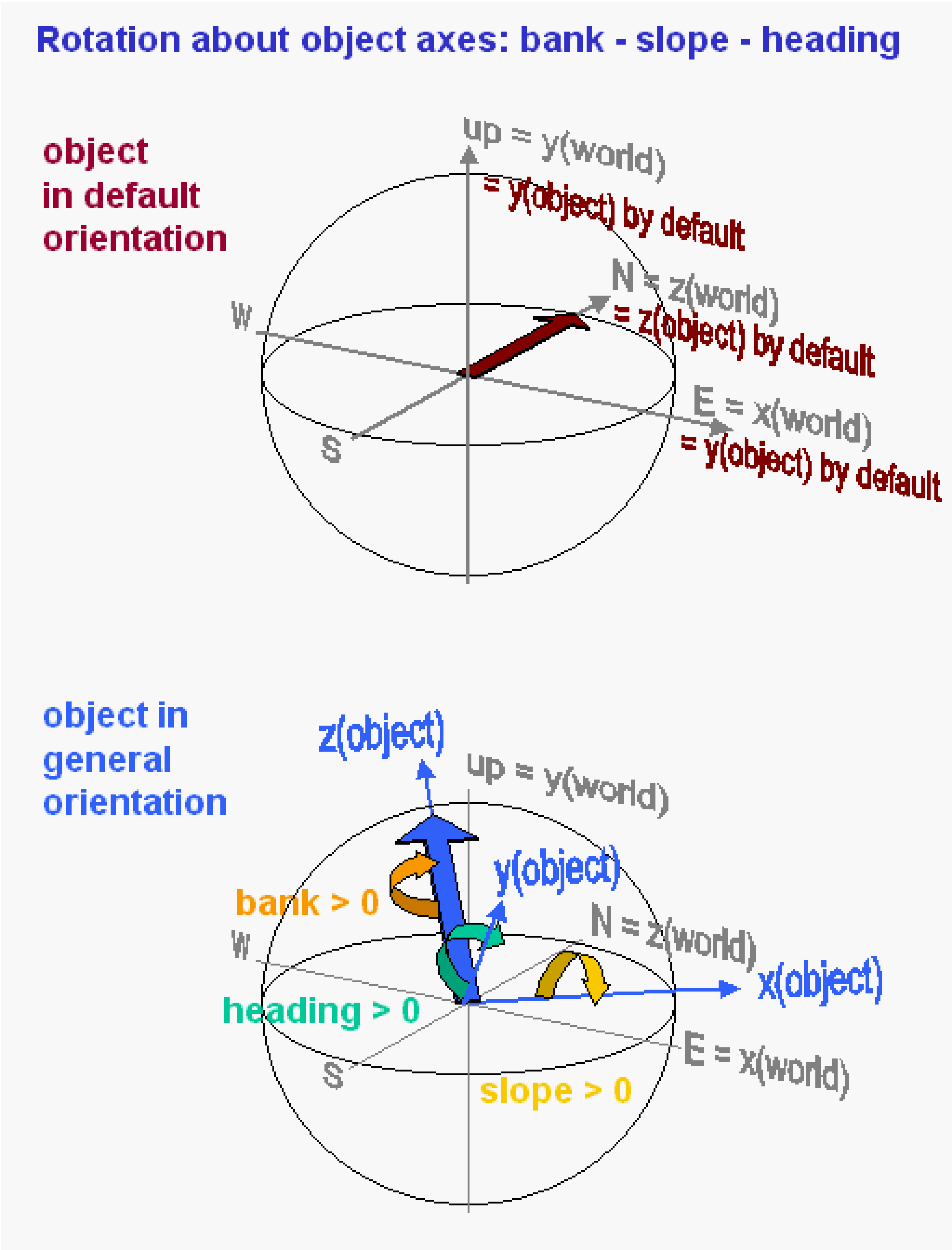
NOTE: You may only change one of these angles at a time. To change more than one of these (for example both the slope and the heading), you first change one of them (the slope), producing new QDirection date for the object; then you copy those new QDirection data back into the input of the **Turn about object axes sheet**, and specify the second change (for the heading): that produces yet another set of QDirection data, which are the final result. (You could still apply a change of bank, if you wish, as a third step, in the same manner.)

The procedure for using the **Turn about object axes sheet of Object Rotator** is the following:

1) copy the QDirection data of the object into part 2) of the sheet;
2) the sheet will show you the object's current bank, slope and heading;
3) in part 3) of the sheet, you must insert the change you want to make to ONE of the angles, as a positive or negative number that will be added to the current value of that angle; the other 2 angle changes should be zero;
4) the resulting new QDirection data will appear in part 3);

5) if you wish to make a change to another angle, copy these QDirection data to the input in part 2) of this sheet (use Paste Special - Values) and repeat the above steps;
6) copy the resulting QDirection data to the object's *.w file.

FIGURE 13.1. ROTATION ABOUT OBJECT AXES



14. TURNING ABOUT WORLD AXES

USAGE: **- learn from the overview below;**
- AND follow the instructions on the <u>Turn about world axes sheet</u> of Object Rotator.

The **Turn about <u>world</u> axes sheet of Object Rotator** allows you to turn an object around the fixed world axes (east-west, north-south, and vertical). This can be done either by changing the object's current bank, slope or heading, or by applying a further rotation about the world axes.

This sheet also automatically offers several preset alternative orientations, such as antiparallel, across, etc. Some of these are identical to those offered in other sheets, such as **Set antiparallel** and **Lay across**.

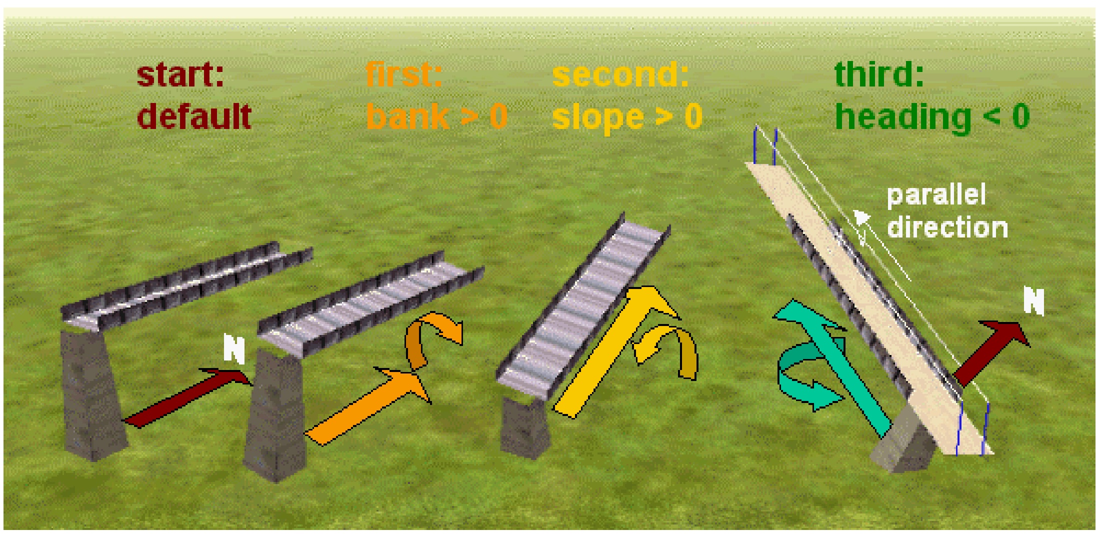
RECOMMENDATION: use the **Turn about <u>OBJECT</u> axes sheet** for most object rotations, because using fixed world axes is generally less intuitive than using mobile object axes for rotations.

To turn an object in this sheet, you must first enter its current orientation, given by its QDirection definition found in its world tile file. Then you must enter three rotation angles on this sheet (be careful with the order of rotations!). This sheet will then produce four numbers that you must overwrite into the QDirection definition of the object in its world tile file.

WARNING: You must be careful with the <u>order</u> in which you apply rotations, because you get different results if you apply rotations in a different sequence (rotating about object axes does not have this limitation). For example, a bank followed by a slope does not give the same result as the same slope followed by the same bank, when using fixed world axes of rotation.

The recommended approach is to apply first a bank, second a slope and third a heading, as illustrated in figure 14.1 for the case of an object placed with its default orientation (pointing north, horizontal and level).

FIGURE 14.1. A BRIDGE TURNED ABOUT FIXED WORLD COORDINATES
IN THE RECOMMENDED ORDER:
the bridge is JP2bluebrg.s, placed in its default orientation at left;
first a bank of +10° is applied (orange), second a slope of +20° (yellow), and third a heading of -45° (green);



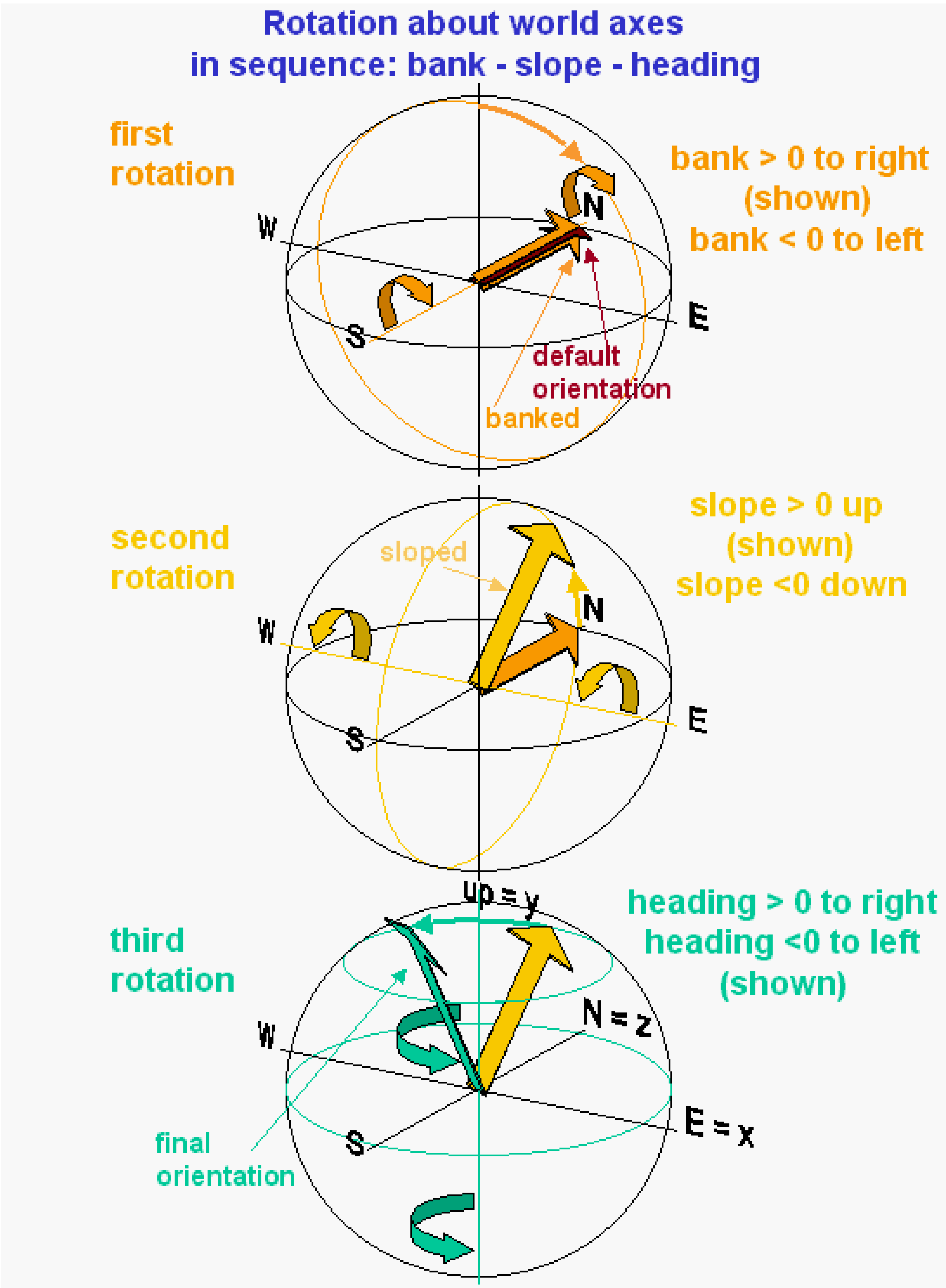
The definition of bank, slope and heading is slightly different with fixed world axes, compared to the case of object axes (also see figure 14.2):

- **<u>bank</u> is a sideways tilt about the horizontal north/south z-axis**: the east side goes down as the west side goes up, or vice versa; in our case, a **positive bank leans down to the east**; a bank can range from -360 to +360°;

- **slope is a vertical turn about the horizontal east/west x-axis**: the north end goes up as the south end goes down, or vice versa; in our case, a **positive slope moves the north end up** and the south end down; a slope can range from -90 to +90°;

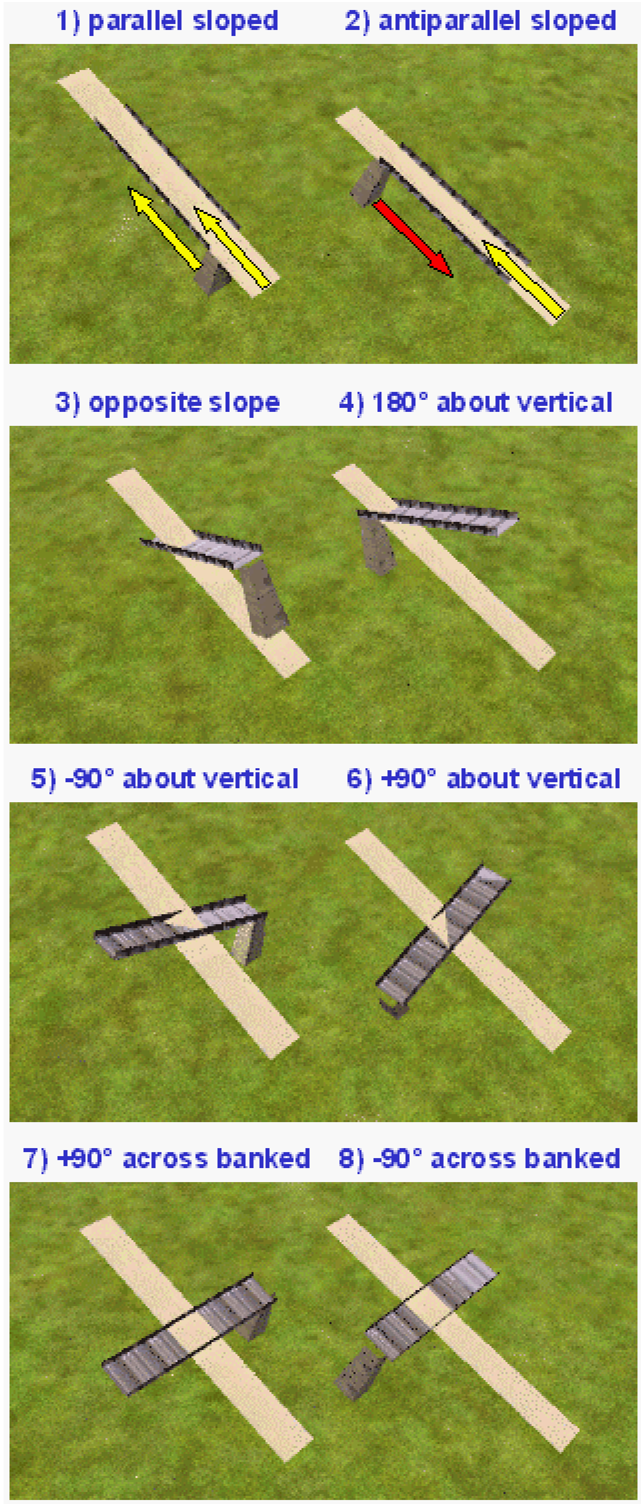
- **heading is a horizontal turn about the vertical y-axis**: the object's north end turns to the right as its south end goes left, or vice versa; in our case, a **positive heading turns the object to the right** as seen from straight above, from north to east, then to south and to west; a heading can range from -360 to +360°.

FIGURE 14.2. ROTATION ABOUT WORLD AXES



The **Turn about <u>world</u> axes sheet** also offers several alternative object orientations, related to the one you specify: these are illustrated in figure 14.3.

FIGURE 14.3. 8 DIFFERENT PRESET OBJECT ORIENTATIONS:
the bridge is JP2bluebrg.s is placed in 8 different orientations, preset in the Turn about world axes sheet; for reference, a fixed road section with the "parallel" orientation shows the effect of these rotations; in cases 1), 2), 7) and 8), the tilt of the road perfectly fits the bridge deck tilt



The procedure for using the **Turn about <u>world</u> axes sheet of Object Rotator** is the following:

1) copy the QDirection data of the object into part 1) of the sheet;
2) the sheet will show you the object's current bank, slope and heading;
3) in part 2) of the sheet, select whether you want to give new rotation angles, or apply a further rotation to the existing orientation;
4) in part 3) of the sheet, insert up to three desired rotations: for each of the three rotations you insert either a bank, a slope or a heading (you may not combine them in one rotation!);
5) the resulting new QDirection data will appear in part 3);
6) related orientations appear further down the sheet;
7) copy the desired QDirection data to the object's *.w file.

15. SHIFTING OBJECTS WITHOUT ROTATION

USAGE: - learn from the detailed instructions below;
- AND follow the instructions on the Shift sheet of Object Rotator.

The **Shift sheet of Object Rotator** allows you to move objects by a particular distance along a particular direction, as defined by the distance and direction between two existing objects.

For example, you could shift the red object in figure 15.1a to its green position, using the distance and direction defined by the black and gray objects. Or, you may have placed a sequence of track sections, and then discovered that you want to insert another one in between (figure 15.1b), or to change the slope of one of the track sections (figure 15.1c), so that the subsequent track sections need to be shifted (without changing their slopes): rather than replacing all those track sections one by one (readjusting the slope for each), you can use this sheet to adjust their positions in the corresponding world file(s). Note that each track section may be different, with different slopes.

The procedure for using the **Shift sheet of Object Rotator** depends somewhat on the situation, exemplified with the cases a, b and c of figure 15.1. Case a is relatively simple, because the object to be shifted is different from the two objects that define the shift distance and direction. Cases b and c are more complex, because the object to be shifted is also one of the objects used to define the shift; you will find that this process with Object Rotator is only worth using if you have to shift many track sections, especially if they have varying slopes; to shift only a few track sections, especially level track sections, it is easier to do it all within RE.

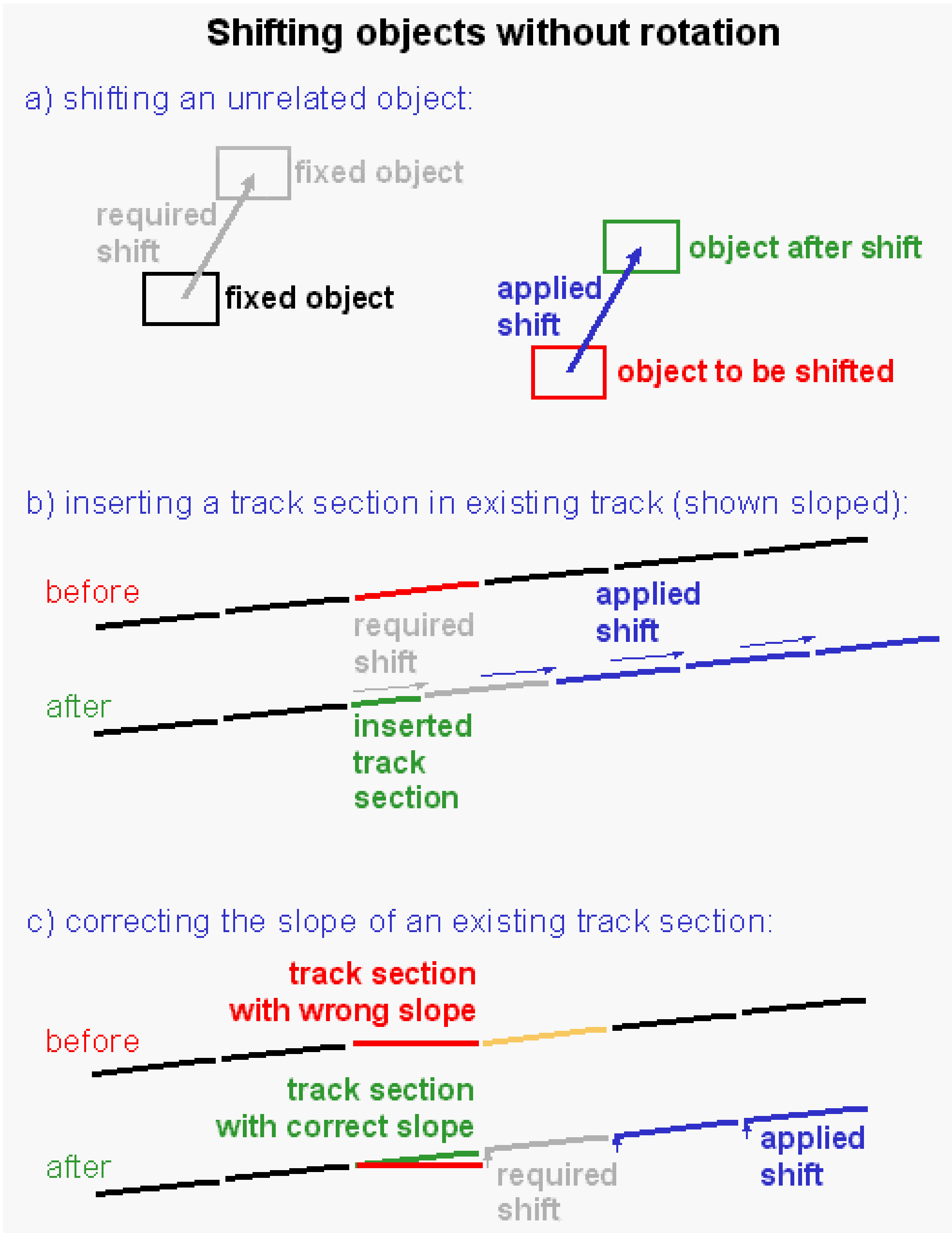
Case a (the red object in figure 15.1a is to be shifted to the green position, based on the relative positions of the black and gray objects):

1) copy the Position data of the black object from its *.w file into section 2) of the Shift sheet;
2) copy the Position data of the gray object from its *.w file into section 3) of the Shift sheet;
3) copy the Position data of the red object from its *.w file into section 4) of the Shift sheet;
4) copy the resulting Position data from section 5) of the Shift sheet to overwrite the existing Position data of the red object in its *.w file;
5) repeat steps 3 and 4 for other objects to be shifted by the same distance and direction;
6) repeat steps 1 to 5 for other objects to be shifted by another distance and direction;
7) save the *.w file;
8) reload the route (without saving it; no track database rebuild is needed after shifting non-track and non-road objects).

Case b (the green track section in figure 15.1b is to be inserted, shifting the red and other track sections):

1) copy the QDirection data (4 numbers) of the (existing) red track section to the Data storage sheet of Object Rotator;
2) copy the Position of the (existing) red track section to section 2) of the Shift sheet,
AND ALSO to section 4) of the Shift sheet;
3) in RE, move the red track section so it is disconnected from other tracks;
4) place the new green track section where the red section was originally, with the desired slope;
5) in RE, move the old red track section so it connects to the new green track section (it should snap in place in the gray position, but with zero slope);
6) save the route;
7) reopen the corresponding *.w file;
8) copy the previously stored QDirection data back to the gray track section data in its *.w file;
9) copy the Position data of the now gray track section to section 3) of the Shift sheet;
10) to shift other track sections by the same distance and direction, do this for each in turn: copy their Position data to section 4) of the Shift sheet, and then copy the resulting Position data from section 5) of the Shift sheet to overwrite their old Position data in the *.w file;
11) save the *.w file;
12) reload the route with track database rebuild, but without saving it.

FIGURE 15.1. SHIFTING OBJECTS WITHOUT ROTATIONS:
example a shows the shift of one object (red) based on the positions of two other objects (black and gray);
examples b and c involve sloping tracks;
each track section (which may have an individual curvature and length) is shown as one bar



Case c (the red track section in figure 15.1c is to be given a different slope, shifting the track sections on its right):

1) copy the QDirection data (4 numbers) of the yellow track section (immediately to the right of the red track section) from the corresponding *.w file to the Data storage sheet of Object Rotator;
2) copy the Position of the yellow track section to section 2) of the Shift sheet,
AND ALSO to section 4) of the Shift sheet;
3) in RE, adjust the slope of the red track section as desired, generating the green orientation;
4) in RE, move the yellow track section so it connects to the green section, producing the gray track section (it should snap in place, but with zero slope);
5) save the route;

6) reopen the corresponding *.w file;
7) copy the Position data of the now gray track section to section 3) of the Shift sheet;
8) copy the previously stored QDirection data back to the gray track section data in its *.w file;
9) copy the resulting Position data from section 5) of the Shift sheet to overwrite the existing Position data of the gray section in its *.w file;
10) to shift other track sections by the same distance and direction, do this for each in turn: copy their Position data to section 4) of the Shift sheet, and then copy the resulting Position data from section 5) of the Shift sheet to overwrite their old Position data in the *.w file;
11) save the *.w file;
12) reload the route with track database rebuild, but without saving it.

16. MAKING A PARALLEL DEVIATION

USAGE: **- learn from the brief instructions below;**
- AND follow the instructions on the Parallel deviation sheet of Object Rotator.

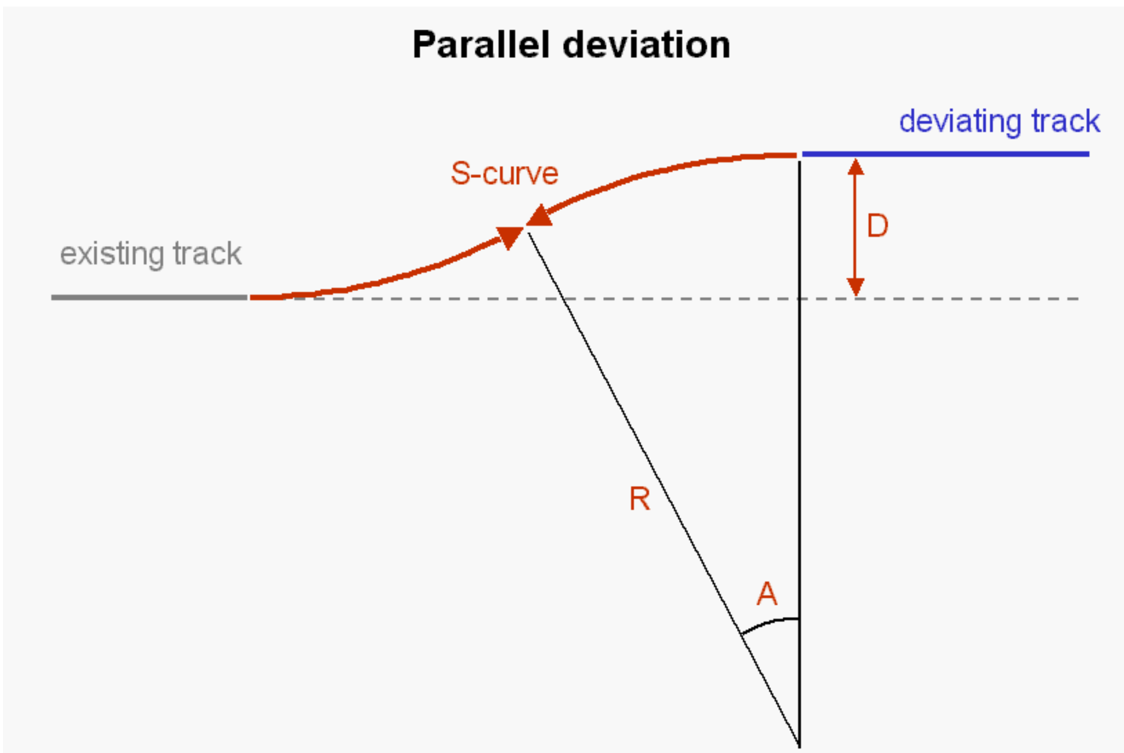
The **Parallel deviation sheet of Object Rotator** allows you to place an S-curve to shift straight track sideways by a desired amount, then continuing in the same direction parallel to the initial track, as shown in figure 16.1. This is useful to make the track pass behind a platform or building.

You select the turn radius and the result is the turn angle, which you use in a pair of dynamic track sections.

CAUTION: RE's rounding of the angle to 3 decimal places results in a slightly incorrect deviation D; it may be off by a few centimeters.

NOTE: To make a pair of tracks split equally (symmetrically) around a platform, use the **Island split sheet of Object Rotator**.

FIGURE 16.1. PLACING AN S-CURVE TO MAKE A DEVIATION:
for a given deviation D and a desired radius R, angle A is calculated for use in 2 dynamic curve sections



17. SPLITTING DUAL TRACKS AROUND AN ISLAND

USAGE: **- learn from the brief instructions below;**
- AND follow the instructions on the Island split section sheet of Object Rotator.

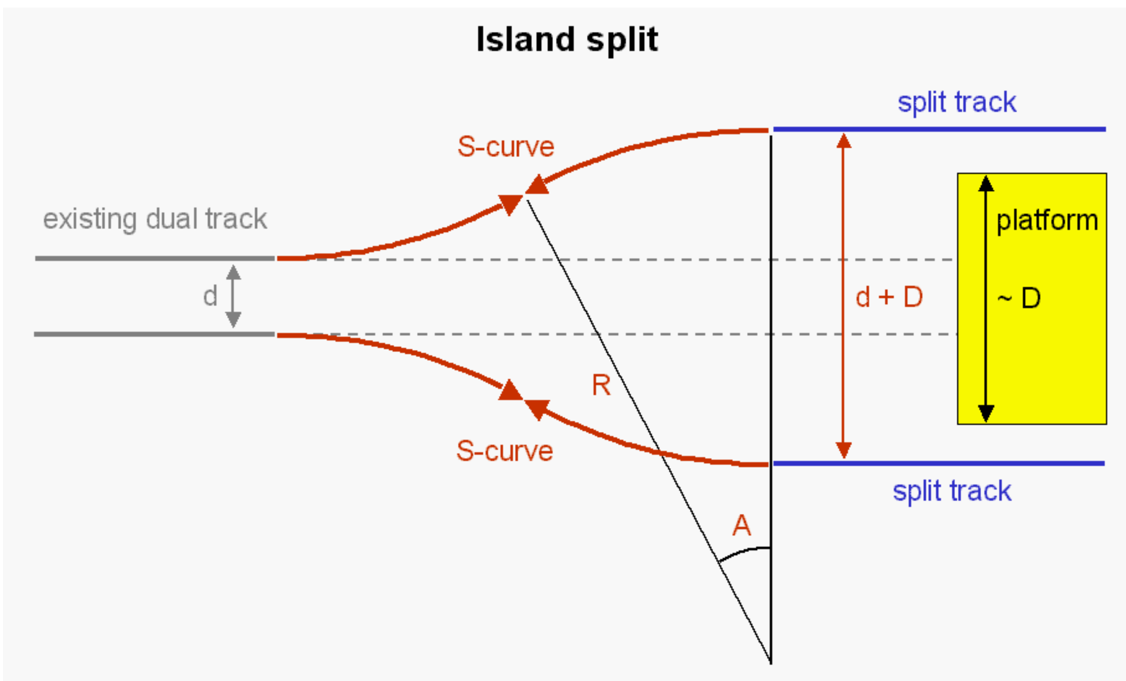
The **Island split sheet of Object Rotator** allows you to place S-curves that deviate two parallel tracks left and right by the same amount (symmetrically), then continuing in the same direction parallel to the initial tracks, as shown in figure 17.1. This is useful to make dual tracks pass around a platform.

You select the turn radius and the result is the turn angle, which you use in pairs of dynamic track sections.

CAUTION: RE's rounding of the angle to 3 decimal places results in a slightly incorrect added spacing D; it may be off by a few centimeters.

NOTE: To make a single track deviate sideways, use the **Parallel deviation sheet of Object Rotator**.

FIGURE 17.1. SPLITTING DUAL TRACKS AROUND AN ISLAND (PLATFORM):
the additional spacing D between tracks is about equal to the platform width;
for a given extra spacing D and a desired radius R, angle A is calculated for use in dynamic curve sections



18. ATTACHING THE NEXT TRACK SECTION

USAGE: **- learn from the brief instructions below;**
- AND follow the instructions on the Next track section sheet of Object Rotator.

The **Next track section sheet of Object Rotator** allows you to join another (single-) track section to an existing one, as is usually done most easily within RE. However, this sheet automatically gives the new track section the same slope as the previous section. Use this sheet particularly for (regular or dynamic) tracks that have large slopes, or when placing tracks deep underground or high above ground.

NOTE: this sheet will not work for multi-track sections (such as A2t…, A3t…, etc.).

NOTE: do not use this sheet for compound track sections that contain both straight and curved segments, such as A1tEndPt… sections, or dynamic curved track sections with both curved and straight segments.

Nine cases are treated in the **Next track section sheet of Object Rotator**: the existing track section may be either a left curve, a straight section or a right curve; and the next track section also may be either a left curve, a straight section or a right curve. The 9 combinations are illustrated in section 26, in figures 26.1-3.

The general procedure involves:
- placing a new track section, without adjusting its slope, either attached to existing track or not;
- copying Position and QDirection data from an existing track section (from the corresponding *.w file);
- entering track information (length, or radius and turn angle), for both the existing track section and the new track section;
- copying resulting Position and QDirection data back to the *.w file.

19. SMOOTHLY CONTINUING STEEP DYNAMIC TRACK FOR ROLLERCOASTERS

USAGE: **- learn from the detailed instructions below;**
- AND follow the instructions on the Rollercoaster curves sheet of Object Rotator.

The **Rollercoaster curves sheet of Object Rotator** allows you to smoothly join steep dynamic track for rollercoasters.

The situation is illustrated in figure 27.3: there a dynamic track section starts at point P1 with QDirection q1. The objective is to attach another dynamic track section at its end, at point P2 with a QDirection q2 such that the joint at point P2 is perfectly smooth. The **Rollercoaster curves sheet of Object Rotator** therefore calculates P2 and q2 for a given P1 and q1, together with a given internal shape of the first dynamic track section (given by the lengths, radii and turn angles of its 5 segment).

NOTE: If the first dynamic track section is reversed (as by pressing T in RE), then the situation is simpler. Then the second dynamic track section must be joined to the "start" P1 of this dynamic track section (instead of to its end, P2): so P2 = P1 and q2 = antiparallel (q1). In this case, the internal shape of the first dynamic track section (its segment lengths, radii and turn angles) is not needed.

The procedure for using the **Rollercoaster curves sheet of Object Rotator** is the following:

1) in RE place the new dynamic track section at the free end of the existing dynamic track section;
2) in RE, specify the internal shape of the new track section (its segment lengths, radii and turn angles);
3) save the route;
4) find the EXISTING dynamic track section in the corresponding *.w file;
5) copy its QDirection and Position data to the **Rollercoaster curves sheet of Object Rotator**;
6) specify whether the existing dynamic track section is reversed (to make it right-curving) or not: if reversed, enter R for right-curving, and skip directly to the results; otherwise enter L for left-curving and move to step 7);
7) if you selected L (left-curving) in 6), also copy the internal shape data of the existing dynamic track section to the **Rollercoaster curves sheet of Object Rotator** (see below how to find these internal shape data);
8) if you selected L (left-curving) in 6), copy the corresponding QDirection and Position data (given in the LEFT column) to the NEW dynamic track section in its *.w file; if you selected R (right-curving) in 6), copy the corresponding QDirection and Position data (given in the RIGHT column) to the NEW dynamic track section in its *.w file;
9) if you want to change the bank of the new dynamic track section, copy the resulting QDirection from the **Rollercoaster curves sheet of Object Rotator** to the **Turn about OBJECT axes sheet of Object Rotator**: there change the bank angle as desired, and then copy the resulting QDirection from that sheet to the *.w file;
10) reload the route with track database rebuild.

CAUTION: if you later want to change the internal shape of the new dynamic track section, do NOT do it in the *.w file; instead do it with the RE dynamic track window. That will restore the default QDirection (no slope, no bank), so you will have to repeat the entire process of finding its q2 and P2, starting with step 3) above.

To extract the internal shape of a dynamic track section, look at its data in the corresponding *.w file. It should look something like this:

```
Dyntrack (
      UiD ( 144 )
      TrackSections (
            TrackSection (
                  SectionCurve ( 0 ) 454 20 0 [1st straight: length]
            )
            TrackSection (
                  SectionCurve ( 1 ) 456 -1.1 50 [1st curve: turn angle and radius]
            )
            TrackSection (
                  SectionCurve ( 0 ) 458 10 0 [2nd straight: length]
            )
            TrackSection (
```

```
                SectionCurve ( 1 ) 460 -0.8 70 [2nd curve: turn angle and radius]
            )
            TrackSection (
                SectionCurve ( 0 ) 462 30 0 [3rd straight: length]
            )
        )
        SectionIdx ( 287 )
        Elevation ( 0 )
        CollideFlags ( 71 )
        StaticFlags ( 00100000 )
        Position ( 846.805 105 -394.903 )
        QDirection ( 0.000000 0.000000 -0.500000 0.866025 )
        VDbId ( 4294967294 )
        StaticDetailLevel ( 0 )
    )
```

The 5 segments are defined in the 5 `SectionCurve` lines, in the order from start to end of the dynamic track section. The red numbers highlighted in red are those you need, as annotated above. The units used are radians and meters, as in the RE dynamic track window, and as needed in the **Rollercoaster curves sheet of Object Rotator**.

20. PLACING AN S COUNTERCURVE

USAGE: **- learn from the detailed instructions below;**
- AND follow the instructions on the S countercurve sheet of Object Rotator.

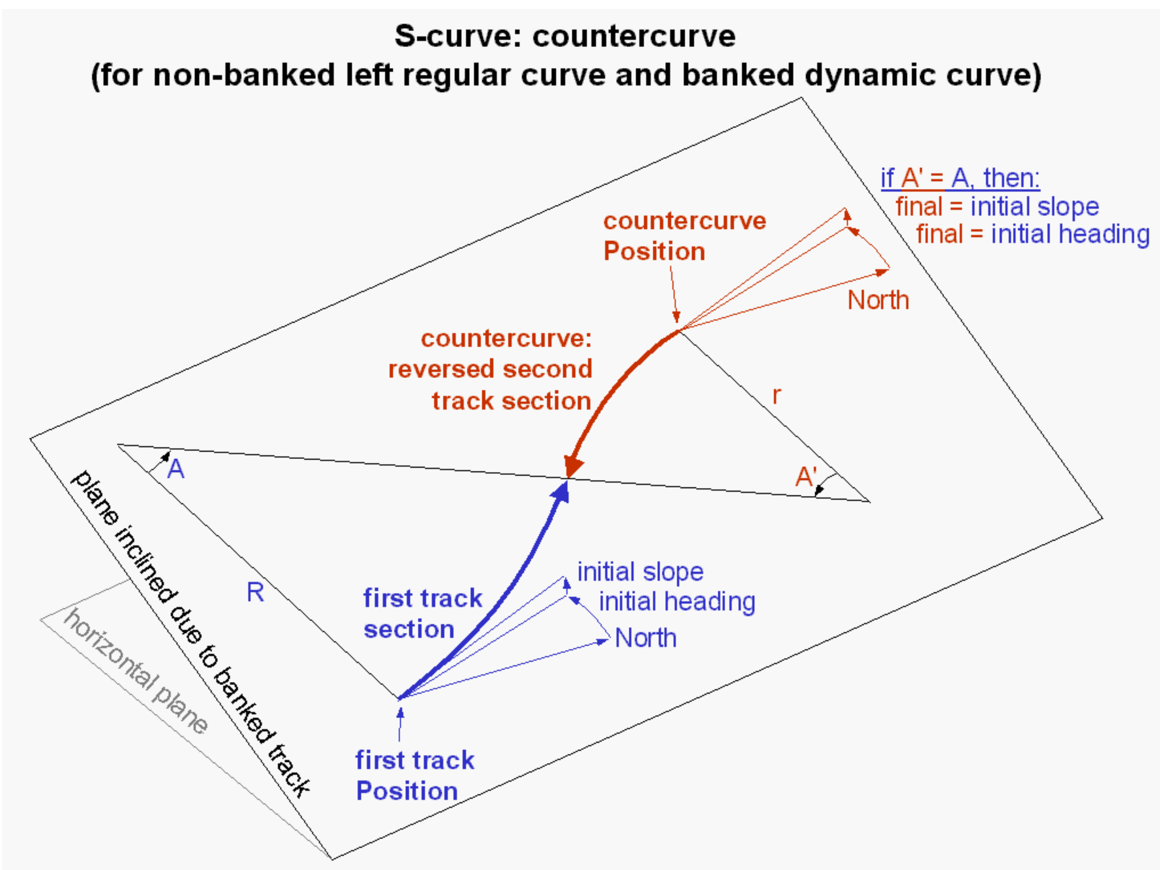
The **S countercurve sheet of Object Rotator** allows you to complete the second half of an S curved track, particularly for (regular or dynamic) tracks that have large slopes, or (dynamic) tracks that are banked. The same turn angle is used for the second curve.

Figure 20.1 shows the goal: once the blue curved track has been placed, the Position and QDirection of the red curved section (the "countercurve") are to be determined, so that the final heading and slope (and bank) will equal the initial heading and slope (and bank). Note that the radius (r) of the red curve may be different from the radius (R) of the blue curve: the final heading and slope (and bank) will still equal the initial heading and slope (and bank).

Special case: right-curved regular tracks. This case does not require Object Rotator as long as the slope does not exceed the 3° limit. Then, normal placement and reversal (by pressing T) in RE is sufficient. If the slope exceeds 3°, the situation is more complex and not treated here.

Special case: right-curved dynamic tracks. This case is handled just like the left-curved case described below; however, the right curves are obtained by using bank angles above 90° or below -90°. One drawback of this approach is that the trackbed texture may be drawn shadowed in MSTS (because the track is then effectively upside-down, even though the rails are correctly drawn above the trackbed).

FIGURE 20.1. PLACING AN S COUNTERCURVE:
after the blue curved track has been placed, the red curved section must be determined



For left-curved tracks (as shown in figure 20.1), there are two cases:
- regular track sections may have any slope (including more than 3°), but no bank (bank = 0°);
- dynamic curved track sections may have any slope (including more than 3°) and any bank (including above 90° and below -90°).

IMPORTANT: dynamic curved track sections must contain only one curve and no straight segments.

The general procedure is to first place the blue track section in RE, and set its slope and/or bank (possibly with the **Turn about world axes sheet**). Then the characteristics of that blue curve (Position, QDirection, radius, turn angle and tile numbers) must be entered, together with the radius of the red track section, into the **S-countercurve sheet**, which will calculate the Position and QDirection of the red track section.

The procedure for using the **S countercurve sheet of Object Rotator** is the following:

1) place the first (blue) curve in RE, with its desired heading and turn angle (and slope if less than 3°);
2) place the second (red) curve in RE, attached to the second curve (it will turn left), with its desired radius, but with the same turn angle as the first curve (leave its slope at 0°);
3) save the route;
4) if the second track section is regular (non-dynamic) and its slope should be larger than 3°, adjust its QDirection by using the **Turn about world axes sheet**;
5) if the second track section is dynamic and its slope should be larger than 3° and/or its bank should be different from 0°, adjust its QDirection by using the **Turn about world axes sheet**;
6) in the **S countercurve sheet**, enter the radius and turn angle of the first curve;
7) enter the first curve's tile numbers;
8) enter the second curve's radius;
9) copy the resulting Position and QDirection data to the countercurve's (second curve's) data in its *.w file.

21. MAKING AN S JOINT

USAGE: **- learn from the detailed instructions below;**
- AND follow the instructions on the <u>S joint sheet</u> of Object Rotator.

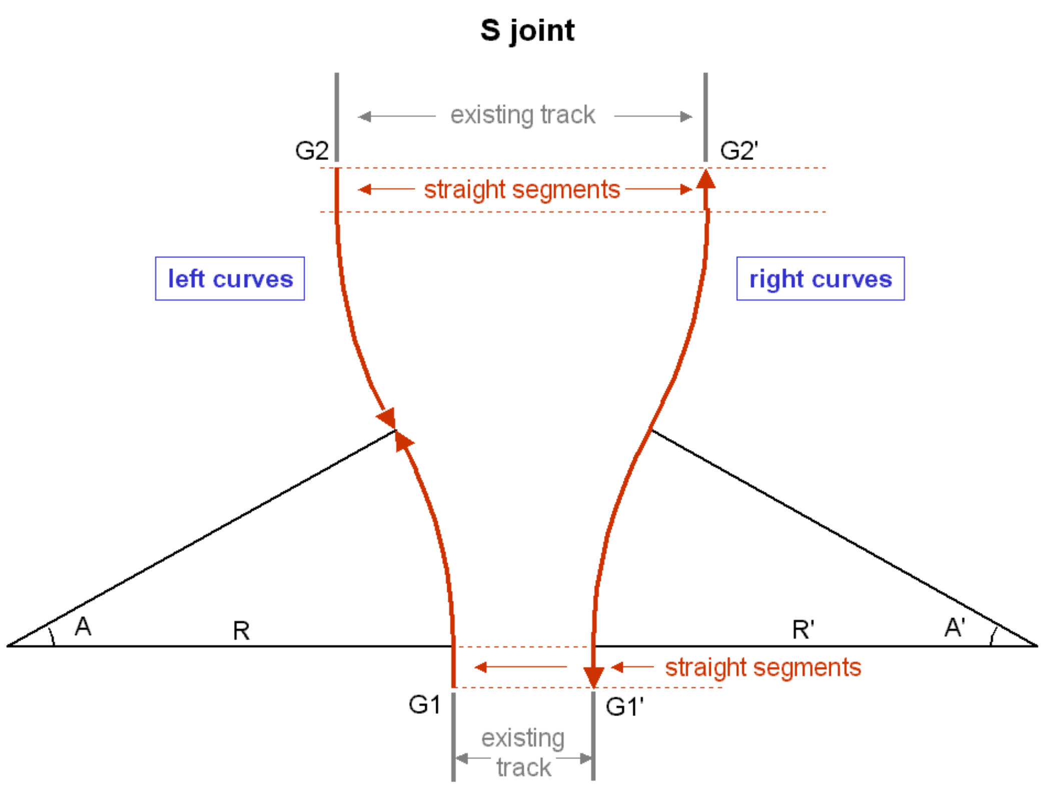
The **S joint sheet of Object Rotator** allows you to fill a gap in a track with two countercurving dynamic track sections.

Two cases need to be distinguished, as shown in figure 21.1: a "left" S joint made of left curves (as seen from the two ends of the gap); and a "right" S joint made of right curves (as seen from the two ends of the gap). The second case requires reversing the curves.

The results of the **S joint sheet of Object Rotator** are calculated for the case of parallel track on the same level (same altitude). However, if the two tracks are not quite on the same level and/or not quite parallel, small manual adjustments can be made to make good joints. Estimated slopes will be given.

In any case, small manual adjustment may be needed, since the dynamic tracks cannot be refined to high precision (due to the MSTS limitation to 2 or 3 digits after the decimal point in their length, radius and turn angle).

FIGURE 21.1. MAKING AN S JOINT:
left case: "left-curving" dynamic tracks will be needed;
right case: "right-curving" (reversed) dynamic tracks will be needed



The procedure for using the **S joint sheet of Object Rotator** is the following:

1) connect a first dynamic track section to one end of the gap (such as point G1 or G1' in figure 21.1), keeping its default settings (don't reverse it at G1'!);
2) connect a second dynamic track section at the other end of the gap (point G2 or G2' in figure 21.1), keeping its default settings (don't reverse it at G2'!);
3) save the route;
4) from the corresponding *.w file, copy the QDirection and Position numbers of the FIRST dynamic track section into section 4) of the **S joint sheet of Object Rotator**;

5) from the corresponding *.w file, copy the QDirection and Position numbers of the SECOND dynamic track section into section 5) of the **S joint sheet of Object Rotator**;
6) several tests are now made to find whether the tracks are parallel and level: if not, you may still be able to use the results, but you will probably need to make manual adjustments at the end; also, the gap length is compared with the gap width to see whether curves of less than 90° will fit: if not, you may have to lengthen the gap;
7) enter the desired turn radius (it is limited by the gap length);
8) a test will tell you if the turn radius is small enough to fit in the gap; also, a further test makes sure the gap is not too long to accept the straight segments (see figure 21.1) that will be needed to fill the non-curved parts of the gap (the straight segments will be part of the dynamic track sections, and are limited to 200m): if this test fails, increase the turn radius, or shorten the gap by placing straight track sections within it;
9) for a "left" S joint, in RE, adjust the two dynamic tracks sections the same way: use the resulting values to set the length of their first straight segment, and the radius and angle of their first curved segment;
10) for a "right" S joint, first change their Position and QDirection data in the corresponding *.w file (to reverse them to make right turns), using the results given; then, in RE, adjust the two dynamic tracks sections the same way: use the resulting values to set the radius and angle of their first curved segment, the length of their SECOND straight segment, finally setting the length of the FIRST straight segment to zero;
11) reload the route, with track database rebuild;
12) if necessary, make fine adjustments to the dynamic track values to improve the track-track joints;
13) save the route.

22. MAKING A CURVED JOINT

USAGE: **- learn from the detailed instructions below;**
- AND follow the instructions on the <u>Curved joint sheet</u> of Object Rotator.

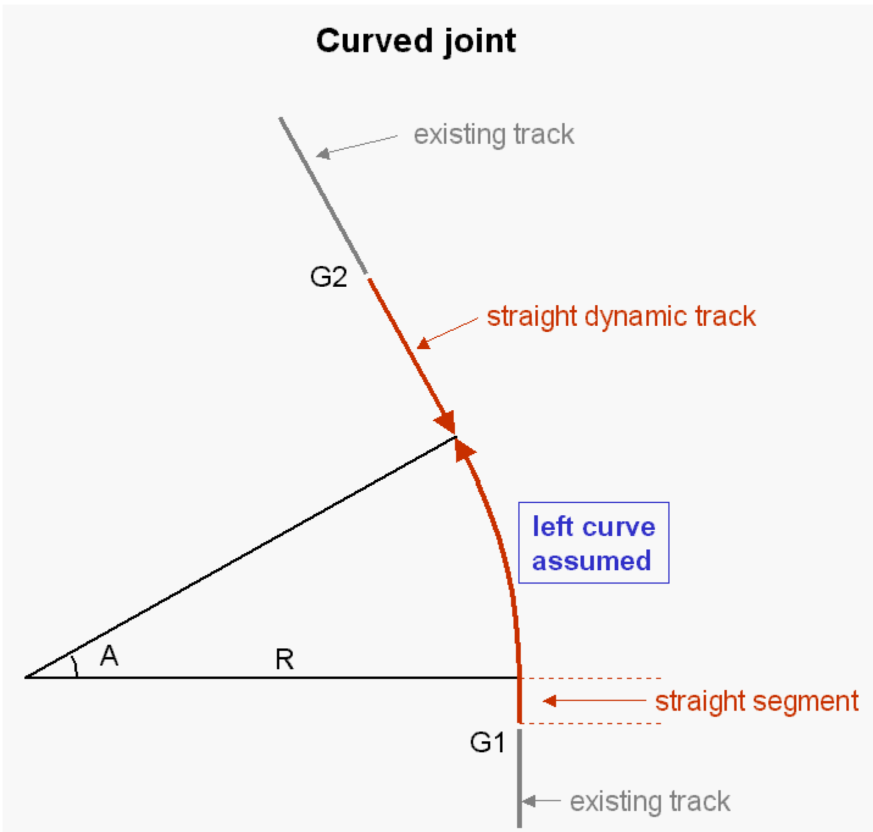
The **Curved joint sheet of Object Rotator** allows you to fill a gap in a track with one curving and one straight dynamic track sections, as shown in figure 22.1.

The case where two (opposite) curves are needed to fill the gap is discussed further below.

The results of the **Curved joint sheet of Object Rotator** are calculated for the case of tracks on the same level (same altitude). However, if the two tracks are not quite on the same level, small manual adjustments can be made to make good joints. Estimated slopes will be given.

In any case, small manual adjustment may be needed, since the dynamic tracks cannot be refined to high precision (due to the MSTS limitation to 2 or 3 digits after the decimal point in their length, radius and turn angle).

FIGURE 22.1. MAKING A CURVED JOINT WITH A SINGLE CURVE:
the gap end G1 is chosen such that a LEFT curve will lead to gap end G2



The procedure for using the **Curved joint sheet of Object Rotator** is the following:

1) go to that end of the gap which will give a LEFT curve toward the other end, as shown in figure 22.1: so go to point G1 (not G2, which would require a RIGHT turn);
2) connect a first dynamic track section to the "left-curve" end of the gap (point G1 in figure 22.1), keeping its default settings;
3) connect a second dynamic track section at the other end of the gap (point G2 in figure 22.1), keeping its default settings;
4) save the route;
5) from the corresponding *.w file, copy the QDirection and Position numbers of the FIRST dynamic track section into section 4) of the **Curved joint sheet of Object Rotator**;

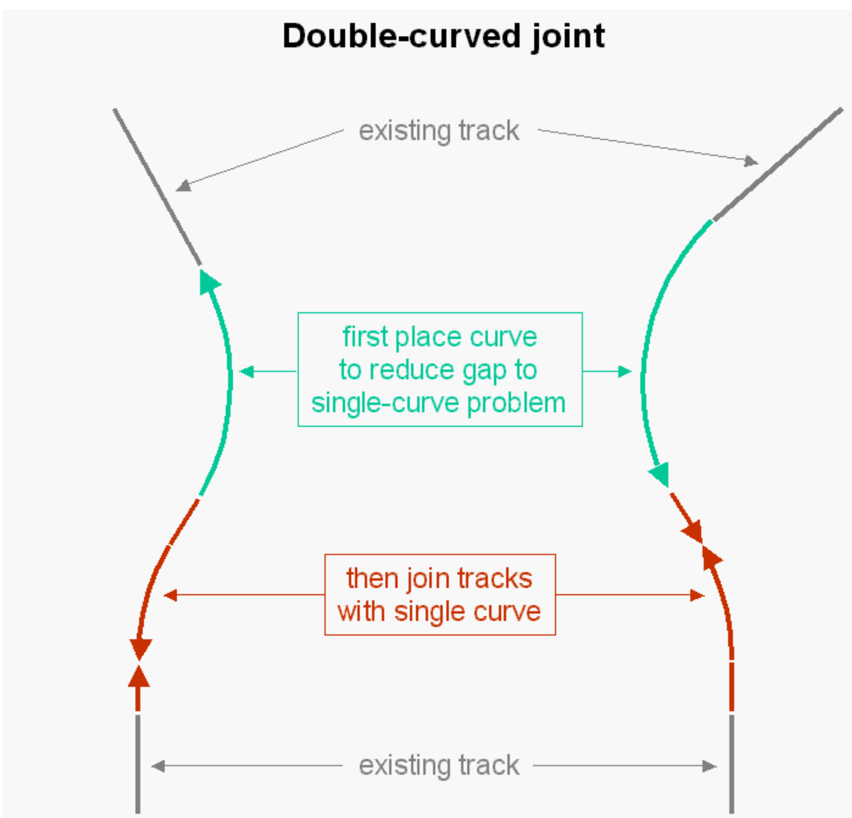
6) from the corresponding *.w file, copy the QDirection and Position numbers of the SECOND dynamic track section into section 5) of the **Curved joint sheet of Object Rotator**;
7) several tests are now made to find whether the tracks are parallel (bad!), and/or at the same level: if not at the same level, you may still be able to use the results, but you will probably need to make manual adjustments at the end; also, it is checked that the intersection of the two existing tracks (each being extended straight through the gap) actually lies within the gap: if not, you may have to lengthen the gap;
8) enter the desired turn radius (it is limited by the gap length);
9) a test will tell you if the turn radius is small enough to fit in the gap;
10) in RE, adjust the FIRST dynamic track section (which will curve left from point G1): use the resulting values to set the length of its first straight segment, and the radius and angle of its first curved segment;
11) in RE, adjust the SECOND dynamic track section (which will go straight from point G2): use the resulting value to set the length of its first straight segment;
12) reload the route, with track database rebuild;
13) if necessary, make fine adjustments to the dynamic track values to improve the track-track joints;
14) save the route.

Cases where two (opposite) curves are needed.

Figure 22.2 shows two cases where a single curve will not fit in the gap: two opposite (left and right) curves are needed because one of the existing tracks (at top in figure 22.2) "overshoots" the gap. This case is easily reduced to the case where a single curve can be used: first place a curve of reasonable radius (shown green) so it corrects the "overshoot" and brings the existing track within reach of a single curve (shown red). Then fill the remaining gap as discussed above, using the **Curved joint sheet of Object Rotator**.

Alternatively, if conditions allow, you could shorten the overshooting existing track so it does not overshoot, thereby increasing the length of the gap, until a single curve will fit in this larger gap.

FIGURE 22.2. MAKING A CURVED JOINT WITH A DOUBLE CURVE:



23. PROCEDURE FOR COPYING RESULTS FROM ONE WORKSHEET CELL TO ANOTHER

You can NOT simply use Ctrl-C (copy) and Ctrl-V (paste) to copy a result from one spreadsheet cell to another (because that copies not the value in the cell, but any formula and formatting in the cell).

Instead, you must use the **Paste Special** option, as follows:

- select the cell containing the result to be copied (left-click in the cell);
- press Ctrl-C;
- select the destination cell;
- right click;
- select "Paste Special…";
- select Paste Values;
- press OK.

24. MEASURING THE DIMENSIONS OF OBJECTS

Object Rotator may need the length, width and "offset" of an object, to properly place the object. All of this information is encoded within the object's *.s shape file, but it is not simple to extract it from there.

So I suggest using a dynamic track section as a measuring tape to measure these dimensions. This way you can get a precision of a couple of centimeters in measuring lengths, which is sufficient for our purposes.

The process is described here step-by-step and illustrated in figure 24.1:

- place the object in question (for example a bridge section or a platform section) on flat terrain;

- leave the object in its default orientation, pointing to the north, as shown at the left in figure 24.1 (a "cross-wise" object, which is wider than it is long, will have its long dimension pointing east/west: you may need to rotate it 90° so it points north, if that is the way you want to align it with track or road);

- if needed, adjust the altitude of the object so its important part (like the bridge deck or the platform surface that needs to be joined) is close to the ground;

- place a dynamic track section a bit to the south and a bit off to the side of the object, as shown at the left in figure 24.1;

- lengthen the dynamic track section so it reaches the near end of the object (as shown in figure 24.1 in the leftmost diagram of the bridge section); read off the length of the dynamic track section: this is called L1 in the picture;

- lengthen the dynamic track section further so it reaches the far end of the object (as shown in figure 24.1 in the second diagram of the bridge section); read off the length of the dynamic track section: this is called L2 in the picture;

- now rotate the object by 180° about its vertical axis (press F4 and use the arrow keys; you don't need perfection here), as shown in figure 24.1 in the third diagram of the bridge section;

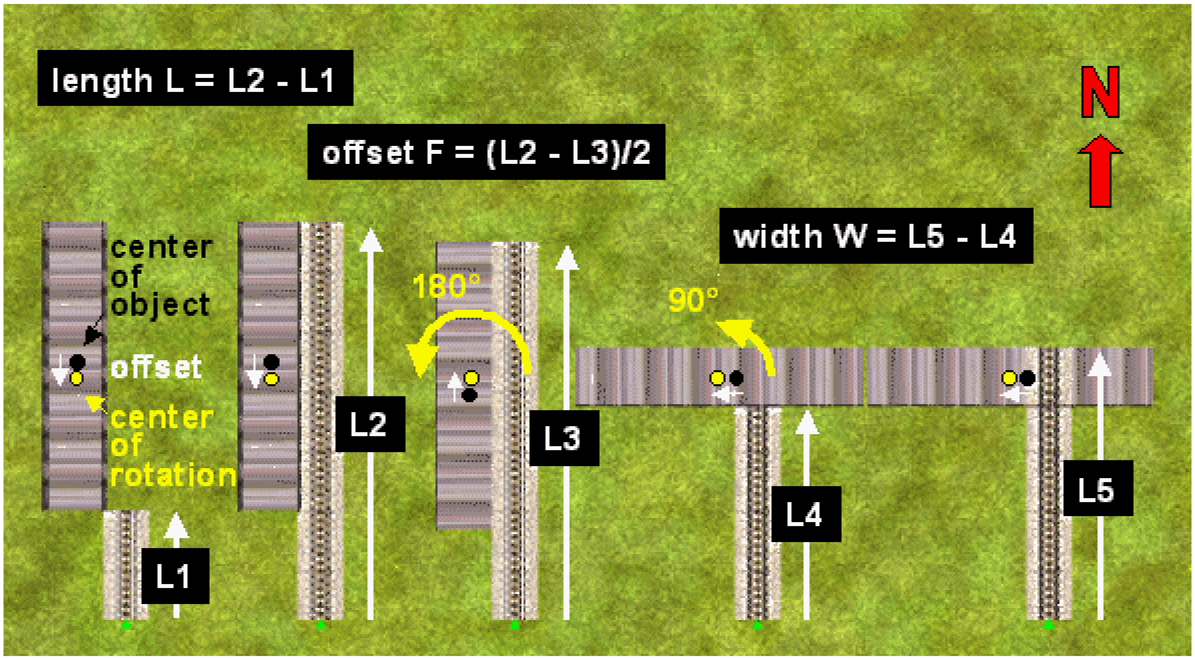
- adjust the length of the dynamic track section further so it reaches the far end of the object (as shown in figure 24.1 in the third diagram of the bridge section); read off the length of the dynamic track section: this is called L3 in the picture;

- next rotate the object another 90° about its vertical axis, as shown in figure 24.1 in the fourth diagram of the bridge section;

- by adjusting the dynamic track section again to fit the near and far edges of the object, you can measure L4 and L5, as shown in figure 24.1 in the fourth diagram of the bridge section;

- finally you can calculate the length L, offset F and width W, using the formulas given in figure 24.1. The results are given in meters, since the length of tracks is measured in meters.

FIGURE 24.1. USING DYNAMIC TRACK TO MEASURE THE DIMENSIONS OF AN OBJECT:
the view looks vertically down; the bridge placed at left is JP2bluebrg.s with its default orientation (to the north); the dynamic track section is placed so it can span the length of the object, giving L1 and L2; the object is then rotated 180° (approximately): measuring L2 gives the offset F through L2 and L3; rotating the object another 90° (approximately) allows measuring the width through L4 and L5; these measurements for JP2bluebrg.s give a length L = L2 - L1 = 29.62 m, an "offset" F = (L2 - L3) / 2 = 1.045 m, and a width W = L5 - L4 = 6.05 m



25. HOW DO QDIRECTION AND ROTATIONS WORK?

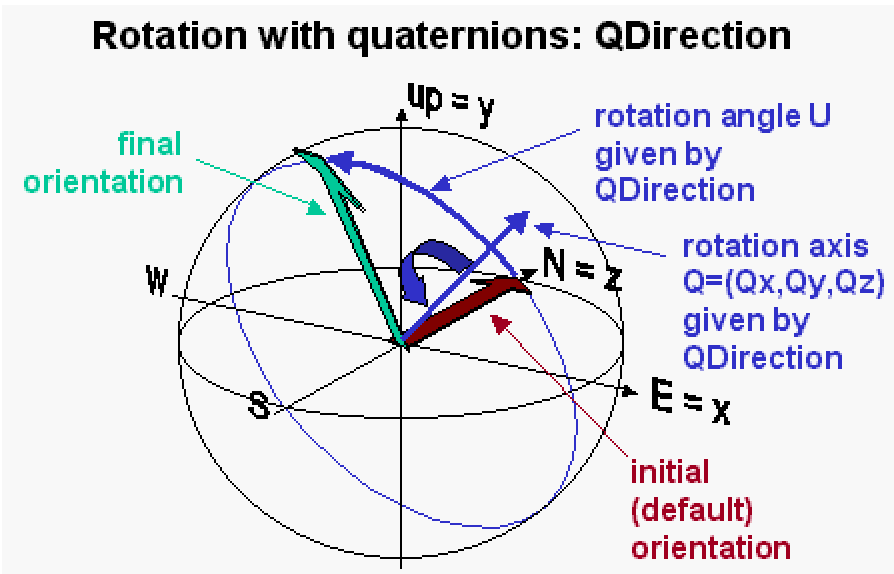
The four numbers appearing in QDirection (a b c d) are examples of what mathematicians call **quaternions**. Quaternions are often used to rotate objects in computer visualization and animation, as well as to orient telescopes and spacecraft, among many other applications that require efficient rotations.

A quaternion defines an **axis of rotation** and an **angle of rotation** about that axis (in addition, an object can be moved along that same axis without rotation).

This is illustrated in figure 25.1. In MSTS, by default, an object is placed pointing north, without slope or bank, as shown as a brown arrow. To rotate the object (the brown arrow) from its default orientation to its desired final orientation (the green arrow), it is possible to find an axis and angle of rotation that reorient the object in one single rotating motion: this axis is shown as the blue arrow labeled Q = (Qx,Qy,Qz), where Qx, Qy and Qz are the x-, y- and z-components of the vector Q, and the angle is shown as U.

[We will assume that Q has unit length, so |Q| = sqrt (Qx.Qx + Qy.Qy + Qz.Qz) = 1; we will use commas in 3D vectors like Q, but spaces in 4D vectors like QDirection q; dots mean multiplication.]

FIGURE 25.1. ROTATION WITH QUATERNIONS



Now q = QDirection (a b c d) follows directly from Q and U:

a = Qx.sin (U/2)
b = Qy.sin (U/2)
c = Qz.sin (U/2)
d = cos (U/2)

[Note that q also has unit length: |q| = sqrt (a.a + b.b + c.c + d.d) = 1.]

IMPORTANT: If you change all signs in q, the result is the same: (a b c d) = (-a -b -c -d).

CAUTION: MSTS uses an unconventional order in QDirection; more frequently, one finds in the literature the order q = (d b a c).

In practice, however, it is not at all obvious how to choose that axis Q and angle U: there is no simple geometrical construction that tells us which axis and angle to choose to reach a desired final orientation from the default orientation (the axis is not simply perpendicular to the initial and final directions of the object). So we will not normally use the rotation vector Q and the rotation angle U. Rather we will think in terms of heading (compass direction), slope (grade or "elevation") and bank (sideways tilt or superelevation).

Nonetheless, there are a few special cases where this rotation axis and angle are simple to find. For example, suppose you place a track section in default orientation (pointing north, without slope or bank), and wish to rotate it in the horizontal plane to a new "heading" H, such as east, for which H = 90° (H = 0° to the north). Then it is clear that the axis of rotation Q is the vertical (y = up), so that Qx = Qz = 0, and Qy = 1; and the rotation angle U then needs to be 90°, resulting in sin (U/2) = cos (U/2) = 0.707107; therefore:

a = 0
b = 0.707107
c = 0
d = 0.707107

or q = (a b c d) = (0 0.707107 0 0.707107). Likewise, choosing a heading of H = 180° (south) results in U = 180° and q = (a b c d) = (0 1 0 0). And the default orientation, with H = U = 0°, gives q = (a b c d) = (0 0 0 1); thus, after placing an object in its default north orientation in the Route Editor, you will find it defined in the corresponding tile file (*.w in the route's TILES folder) with QDirection (0 0 0 1).

A useful relation gives the "antiparallel" orientation of an object. If q (a b c d), then qa = (-c -d a b) gives the antiparallel orientation. Here, "antiparallel" is meant as in section 4: "nose to tail" so that the slope and bank are reversed, which means, for example, that an object that fits inclined terrain will be flipped nose to tail in such a way that it still fits that same inclined terrain. In particular, the north direction (0 0 0 1) becomes the south direction (0 -1 0 0) = (0 1 0 0), and the east direction (0 0.707107 0 0.707107) becomes the west direction (0 -0.707107 0 0.707107).

CAUTION: It is important not to confuse the axis of rotation Q with the internal axis of the object in its rotated orientation, normally the vector z = (0,0,1) when the object has the default orientation!

Combining two rotations into one rotation

You may want to further rotate an object that already was rotated: for example, you may have changed the slope of an object, and now want to also change its heading. So you want to apply a second rotation after a first rotation was already applied, and end up with a single QDirection that represents the combined result of both rotations (because MSTS defines an object's orientation by a single QDirection relative to the default object orientation).

This means multiplying another QDirection (defining the new rotation) into the first QDirection (which defined the original orientation of the object). Suppose an object is first oriented using QDirection q = (a b c d), and it must be reoriented by a further rotation QDirection q' = (a' b' c' d'). Then the new (combined) orientation of the object becomes q" = (a" b" c" d") = q'.q, such that:

a" = a'.d - b'.c + c'.b + d'.a
b" = a'.c + b'.d - c'.a + d'.b
c" = - a'.b + b'.a +c'.d + d'.c
d" = - a'.a - b'.b - c'.c + d'.d

IMPORTANT: it is essential to respect the order q'.q, a'.d, etc., where q is the FIRST rotation and q' the SECOND rotation! (We use the right-to-left matrix product order when writing q'.q) The reversed product q.q' gives a very different result. Note that the notation q'.q represents a "quaternion multiplication", defined by the 4 formulas given here.

The new QDirection q" represents a single rotation from default orientation to new orientation, combining the two earlier rotations q and q': q" defines a single new rotation axis and a single new rotation angle, which give the same result as applying the earlier two rotations one after the other.

These formulas may seem complicated, but this is where the use of QDirection (namely quaternions) is so effective computationally: these formulas, which only require multiplications and additions, can be computed much faster than formulas involving trigonometric functions of various angles.

Rotating a 3D vector

Since QDirection rotates an object, it may be used to find where a particular point of that object ends up after rotation. Such a point is given by a 3D vector P, which is rotated to point P'. We assume that the object's center is at (0,0,0) (if the center is at a different position T, the end result should be shifted by the vector T). We then ask to which position P' = (x',y',z') the point P = (x,y,z) is rotated by QDirection q = (a b c d). The answer is given by:

x' = (a.a - b.b - c.c + d.d).x + (2.a.b + 2.c.d).y + (2.a.c - 2.b.d).z
y' = (2.a.b - 2.c.d).x + (- a.a + b.b - c.c + d.d).y + (2.a.d + 2.b.c).z
z' = (2.a.c + 2.b.d).x + (- 2.a.d + 2.b.c).y + (- a.a - b.b + c.c + d.d).z

[The general formula that produces this result is p' = q.p.q^{-1}, using quaternion multiplications, where p' = (0 x' y' z'), p = (0 x y z) and q^{-1} = (-a -b -c d); q^{-1} is the "inverse rotation", though angle -U.]

In particular, the point (x,y,z) = (1,0,0) will be rotated to the point (x1,y1,z1) with:

x1 = a.a - b.b - c.c + d.d
y1 = 2.a.b - 2.c.d
z1 = 2.a.c + 2.b.d

Likewise, (x,y,z) = (0,1,0) will be rotated to the point (x2,y2,z2) with:

x2 = 2.a.b + 2.c.d
y2 = - a.a + b.b - c.c + d.d
z2 = - 2.a.d + 2.b.c

And, (x,y,z) = (0,0,1) will be rotated to the point (x3,y3,z3) with:

x3 = 2.a.c - 2.b.d
y3 = 2.a.d + 2.b.c
z3 = - a.a - b.b + c.c + d.d

These last 3 results can be useful when rotating points that can be expressed conveniently as a simple combination of the vectors (1,0,0), (0,1,0) and (0,0,1). An example would be the end point of a track section, discussed next.

Finding the end point of a straight or curved track section

NOTE: The results given in this section apply only to perfect curved sections, not to compounded curves+straights like A1tEndPnt10d or dynamic track sections that include more than a single curved segment.

NOTE: The results given in this section also apply to road sections.

Take a straight track section with orientation given by q = (a b c d), and with a length of 100 meter. Then its end point, in the default orientation, is at (0,0,100) or 100.(0,0,1). Since the point (0,0,1) rotates to point (x3,y3,z3), we know immediately that, after rotation, the end point will move to position 100.(x3,y3,z3) = (100.x3, 100.y3, 100.z3), where x3, y3 and z3 are given above in terms of the known a, b, c and d.

More generally, if the length of a straight track section is d, then its end point will be turned from d.(0,0,1) to d.(x3,y3,z3).

A slightly more complex case is the end point of a single-width curved track section, illustrated at the bottom of figure 25.2. We again assume that the start of the track section, labeled S, is at position (0,0,0). By default then, the start of this track section will initially point north along (0,0,1), while curving left through an angle A with a radius r.

The track's end point is located at E: its x-coordinate relative to S is the distance from point S to point B, namely -r.(1 - cosA) for a left turn (it would have the opposite sign for a right turn); and its z-coordinate relative to S is the distance from point B to point E, namely r.sinA. So we can write the position E relative to S as a combination of x and z components, using the unit-length vectors (1,0,0) and (0,0,1): the vector from S to E becomes SE = - r.(1 - cosA).(1,0,0) + r.sinA.(0,0,1). Or we can write the 3D components of the vector from S to E:

SEx = - r.(1 - cosA)
SEy = 0
SEz = r.sinA

Now what happens when we rotate this track section using QDirection (a b c d)? The position E changes in a complicated way to position E' (not drawn). However, the vectors (1,0,0) and (0,0,1) change in a relatively simple way: (1,0,0) turns to (x,y,z) and (0,0,1) turns to (x1,y1,z1), as given earlier.

So SE turns from

SE = - r.(1 - cosA).(1,0,0) + r.sinA (0,0,1)

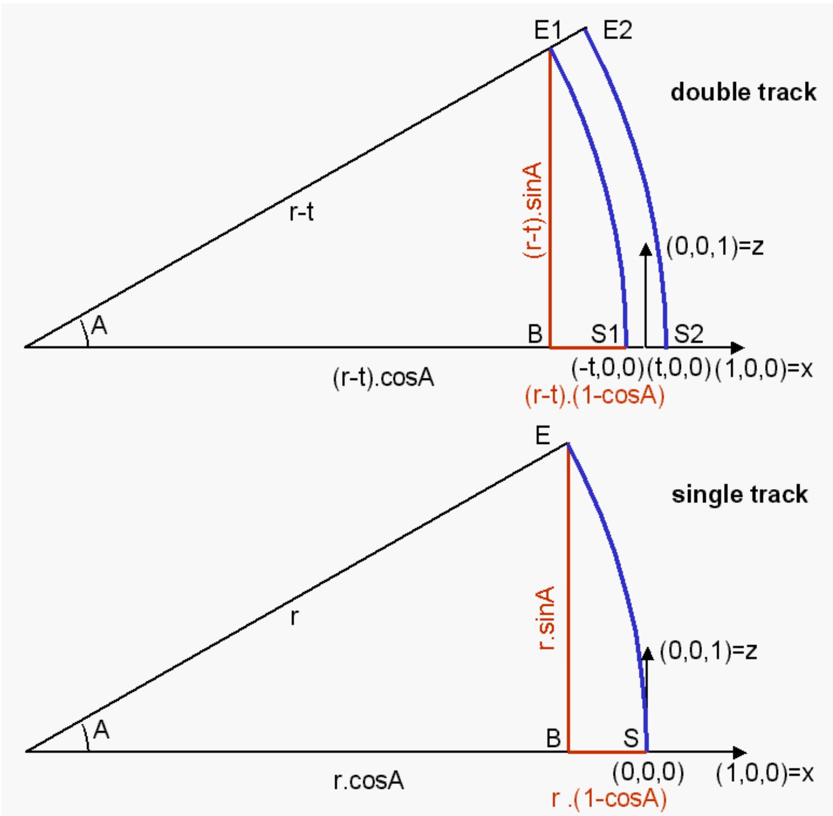
to

SE' = - r.(1 - cosA).(x1,y1,z1) + r.sinA.(x3,y3,z3).

And the x-, y- and z-components of SE' will therefore be:

SE'x = - r.(1 - cosA).x1 + r.sinA.x3
SE'y = - r.(1 - cosA).y1 + r.sinA.y3
SE'z = - r.(1 - cosA).z1 + r.sinA.z3

FIGURE 25.2. GEOMETRY OF CURVED SINGLE AND DOUBLE TRACKS



Finally, you must shift the result from the "start" position (0,0,0) to the actual position T = (Tx,Ty,Tz) by adding (Tx,Ty,Tz) to the above result:

SE'x + Tx

SE'y + Ty
SE'z + Tz

Now we can address the case of a double track section, shown at top in figure 25.2. We leave the original "start" position S at (0,0,0). Now the two track beginnings S1 and S2 are at distances t to the west and east of that point: S1 = (-t,0,0), S2 = (t,0,0). Also the 2 tracks have different radii: r - t for the inner track, r + t for the outer track; t = 2.4925 m in MSTS.

We apply the same logic as before to each of these curved track sections, except for the fact that their beginning is slightly displaced and their radii differ.

So, before rotation, the end point of track 1 will be at:

S1E1x = - t - (r - t).(1-cosA)
S1E1y = 0
S1E1z = (r - t).sinA

Rotation through QDirection (a b c d) will bring S1E1 to:

S1E1'x = [- t - (r - t).(1 - cosA)].x1 + (r - t).sinA.x3
S1E1'y = - (r - t).(1 - cosA).y1 + (r - t).sinA.y3
S1E1'z = - (r - t).(1 - cosA).z1 + (r - t) sinA.z3

You similarly get S2E2 and S2E2' by replacing -t by +t in these results:

S2E2'x = [t - (r + t).(1 - cosA)].x1 + (r + t).sinA.x3
S2E2'y = - (r + t).(1 - cosA).y1 + (r + t).sinA.y3
S2E2'z = - (r + t).(1 - cosA).z1 + (r + t).sinA.z3

Again, you still have to shift S1E1' and S2E2' by (Tx,Ty,Tz), if the track did not start at (0,0,0) but at position T.

26. SOME FORMULAS FOR SPECIFIC TRACK ARRANGEMENTS

ATTACHING THE NEXT TRACK SECTION

We deal here with the relatively simple process of adding a track section to existing track: this is the usual mode of extending tracks in RE and it is usually done most easily in RE (without using Object Rotator). However, you may need to know the position and orientation (QDirection) of the next track section that gives a smooth joint when the track is sloping: that orientation is not available from RE. Here we provide general formulas that can be used for such situations, including the case of strong slopes and even banking tracks: these are in particular useful to make complex rollercoaster curves.

Our focus is on providing formulas that determine the Position and QDirection of the next track (as programmed into Object Rotator's **Next track section** sheet).

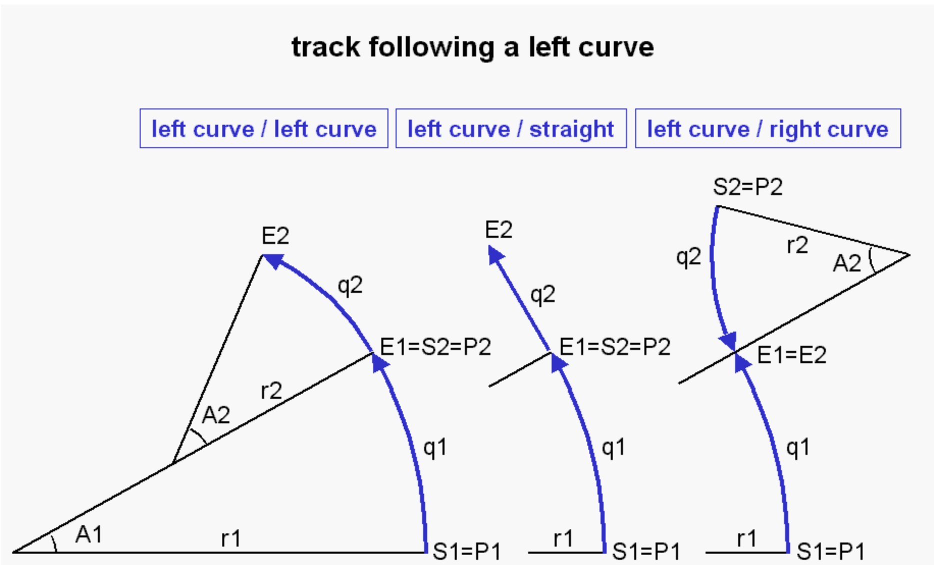
We will consider 9 cases, illustrated in figures 26.1, 26.2 and 26.3:
- starting from an existing left curve, attach another left curve (left/left), or a straight (left/straight), or a right curve (left/right) - see figure 26.1;
- starting from an existing straight, attach a left curve (straight /left), or another straight (straight /straight), or a right curve (straight /right) - see figure 26.2;
- starting from an existing right curve, attach a left curve (right /left), or a straight (right /straight), or another right curve (right /right) - see figure 26.3.

Starting from an existing left curve

In figure 26.1 the existing left curve is shown three times at bottom, connecting with three different cases of "next" (new) track sections. The existing track section has a Position P1 and a QDirection q1. P1 lies at the beginning of the track, P1 = S1, while q1 gives the orientation of the track at that position P1 = S1. We can calculate the position of the end E1 of this track section with formulas given in section 25, knowing its radius r1 and turn angle A1.

The next track section starts at S2 and ends at E2, so its Position will be P2 = S2. It will have QDirection q2. If the next track section is a left curve or a straight, then P2 = S2 = E1; also q2 will be the same regardless whether the next track is a left curve or straight. But if the next track section is a right curve, then P2 = S2 does not coincide with E2, because the track will have to be reversed to make it turn right instead of left. In that case, it will be necessary to use its radius r2 and turn angle A2 to calculate where P2 = S2 is; and q2 will need to include the reversal.

FIGURE 26.1. ATTACHING TRACK TO AN EXISTING LEFT CURVE



To calculate q2 for a left-curved or straight next track, we apply a horizontal rotation (around the vertical y-axis) to q1, by the angle A1: q2 = q1.qA1, where qA1 = (0 sin(A1/2) 0 cos(A1/2)). It is important to use the order q1.qA1, so that the slope of q1 is maintained. P2 is obtained by adding S1E1 to P1, where S1E1 was calculated in section 25: it is the result of rotation, so it is actually the result S1E1' (with the prime!) of section 25, obtained from the present q1.

For a next right curve, q2 becomes

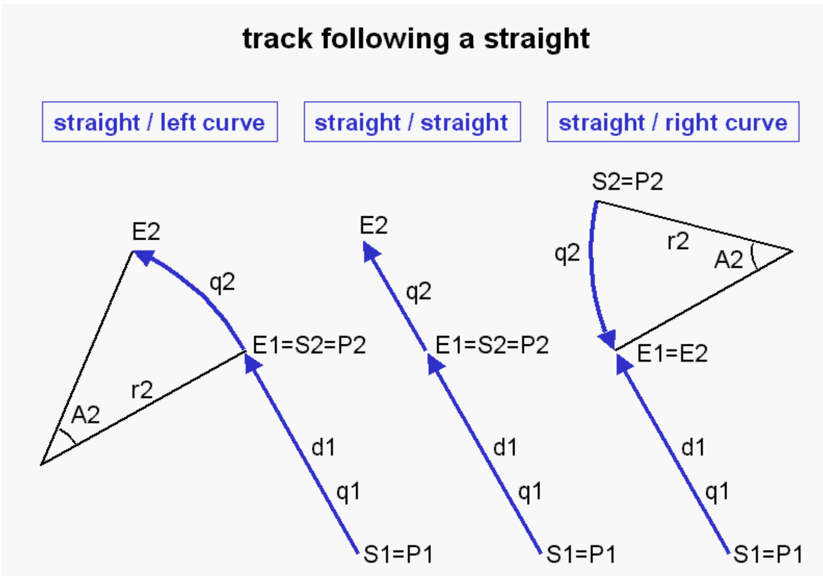
q2 = antiparallel (q1.qA1.q-A2)

where qA1 is given above, q-A2 = (0 sin(-A2/2) 0 cos(-A2/2)) (using the negative of A2), and antiparallel (a b c d) = (-c-d a b) (this reverses the track). P2 is now obtained as P2 = P1 + S1E1 - S2E2, with S1E1 as before, and S2E2 determined by radius r2 and turn angle A2.

Starting from an existing straight

In figure 26.2 the existing track is again shown at bottom three times, now straight with length d1, and again connecting with three different cases of "next" (new) track sections.

FIGURE 26.2. ATTACHING TRACK TO AN EXISTING STRAIGHT



This situation is slightly simpler than with an existing left curve. For the straight/left and straight/straight combinations, q2 = q1; and P2 = P1 + S1E1, with S1E1 = d1.(x3,y3,z3), where (x3,y3,z3) is derived from q1, as shown in section 25.

For the straight/right combination, q2 is

q2 = antiparallel (q1.q-A2)

where q-A2 was defined above. And P2 = P1 + S1E1 - S2E2, with S1E1 and S2E2 as given before.

Starting from an existing right curve

Figure 26.3 shows this situation: it is even simpler because the point P1 = S1 (the start point of the existing track) is already at the joint position.

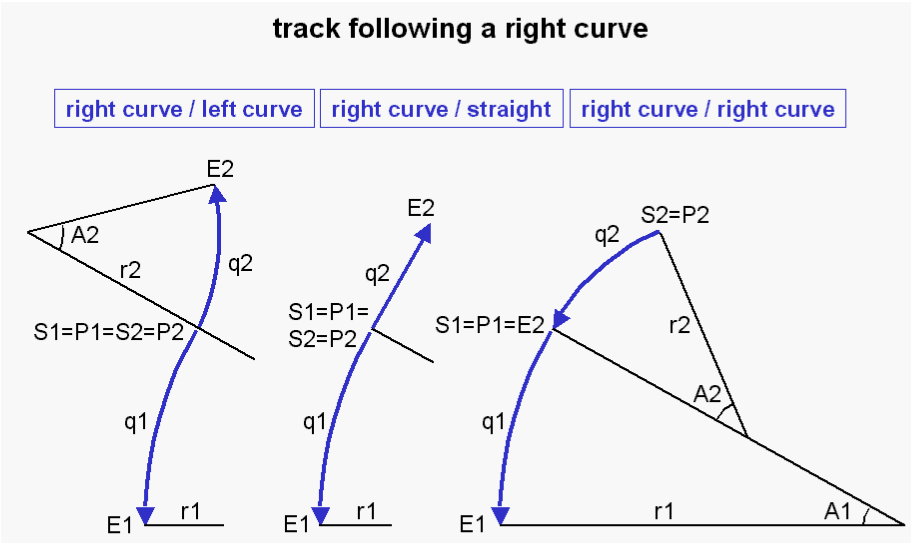
So, P2 = P1 for left and straight next tracks. But q2 = antiparallel (q1), because the existing track section must be reversed to get the next track's direction.

For the right curve (right/right combination), we have

q2 = antiparallel [antiparallel(q1).q-A2]

with q-A2 as defined before (we again must first reverse q1 to get the correct direction). And P2 = P1 - S2E2, with S2E2 given above.

FIGURE 26.3. ATTACHING TRACK TO AN EXISTING RIGHT CURVE



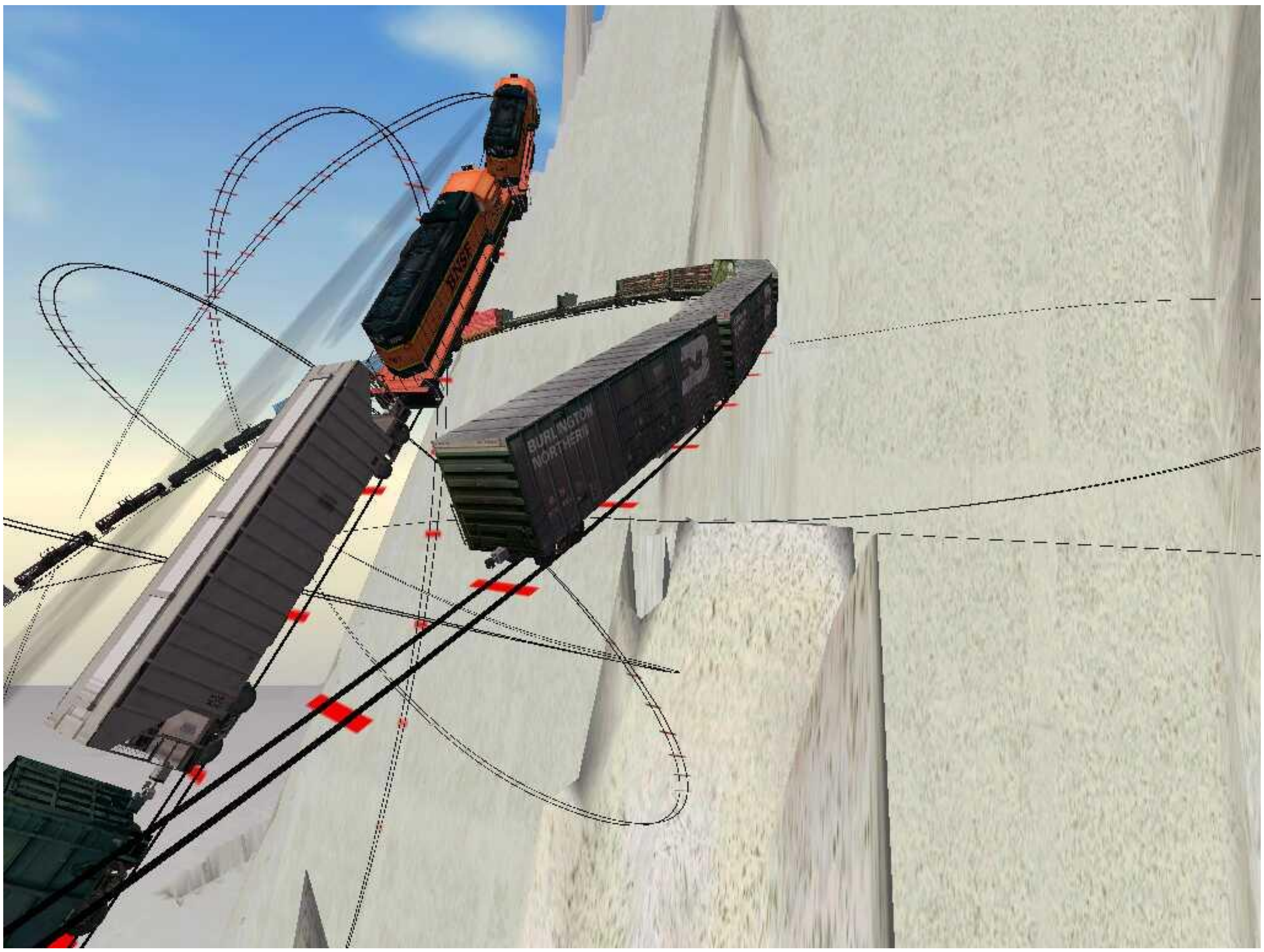
27. MAKING ROLLERCOASTERS

To some extent, MSTS allows rollercoasters to be made: this section discusses methods to do so.

An example using **dynamic track sections** is shown in figure 27.1. Some limitations with dynamic track sections are:

- rails are not "banked" (superelevated);
- sloped and/or banked dynamic tracks have a reduced gauge, which varies smoothly;
- trains will always ride upright (never banked and never upside down).

FIGURE 27.1. A ROLLERCOASTER MADE WITH DYNAMIC TRACKS:
a view from my route called MegaCoaster,
using a custom-made track texture



It is also possible, but much more labor-intensive, to make steep rollercoaster tracks with **regular track sections**, as shown in figure 27.2. One advantage is that rails can be banked (superelevated), but the trains still ride upright. Another advantage is that the track gauge is not reduced with regular track sections. On the other hand, you need to place very many regular track sections to cover the same length as one dynamic track section, which is very time-consuming. Also, regular tracks will show slope and bank mismatches, which can be avoided with dynamic tracks: the track in figure 27.2 is "bumpy" compared to the very smooth curves seen in figure 27.1.

For these reasons, dynamic tracks are preferable to regular tracks for making rollercoasters, and we will only discuss the case of dynamic tracks from here on.

FIGURE 27.2. A ROLLERCOASTER LOOP MADE WITH REGULAR TRACKS (DETAIL AT RIGHT):
the views look down along a steep acceleration slope;
the vertical curves are made with 10m straight sections using 6° slope changes;
the 360° vertical loop thus requires about 360 / 6 = 60 such straight sections
(2 dynamic track sections would suffice in their place);
curves (inserted at bottom of loop) are needed to prevent the loop from rejoining itself;
where the loop passes through the vertical (shown at lower right),
the tracks and trains flip around to remain upright

The shape of dynamic tracks

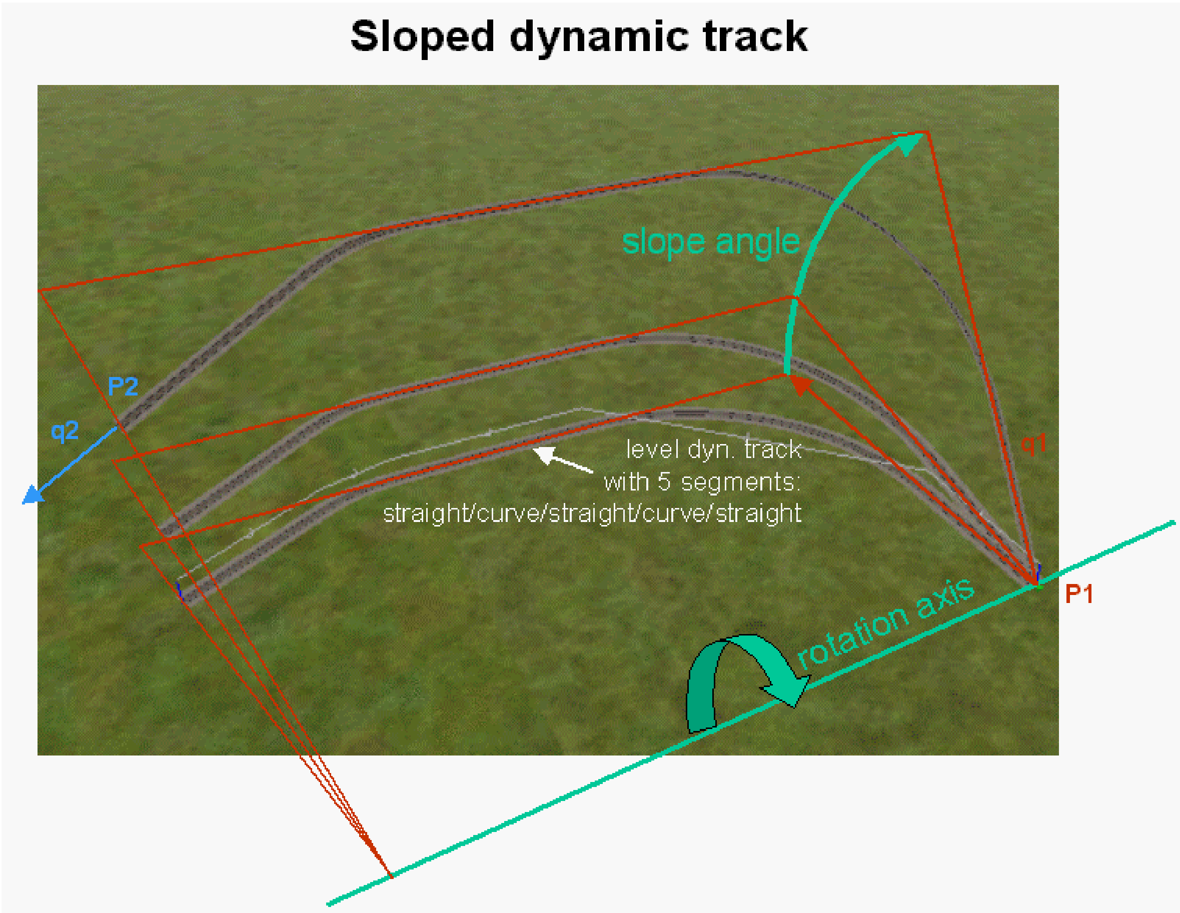
To make smooth steep tracks using dynamic track sections, it is important to understand their geometry. Similar to a regular curved track section (shown in figure 11.2), a dynamic track section is drawn in one plane: but while a regular curved track section contains only one curved segment, a dynamic track section can contain 1 or 2 curved segments and 1, 2 or 3 straight segments. The bottom track in Figure 27.3 is one single horizontal dynamic track section with 5 segments: straight/curve/straight/curve/straight, starting from point P1 (see the straight white lines, one over each of the 5 segments). Note that by default all curves turn left.

A most important property of a dynamic track section is that its plane rotates rigidly as its slope and/or its bank are changed. Figure 27.3 shows the effect of **sloping** a level dynamic track section: the entire set of 5 segments rotates rigidly around the drawn rotation axis, which lies PERPENDICULAR to its first segment. Each shown sloped track section lies in a different sloped plane, as suggested by the red lines.

There is no benefit in using slopes of more than 90°: if you try, the track will be drawn upright (not upside down), and the trains will still ride upright as well (not upside down).

You may use positive as well as negative slope angles (from near +90° to near -90°): positive angles are shown in figure 27.3.

FIGURE 27.3. SLOPED DYNAMIC TRACKS:
the slope of a dynamic track section rotates its (red) plane rigidly;
the rotation axis goes through the start of the track section and lies PERPENDICULAR to its first segment
shown are slope angles of 0° (bottom), 15° and 60° (top);
in this example, the first curved segment turns left by 90°

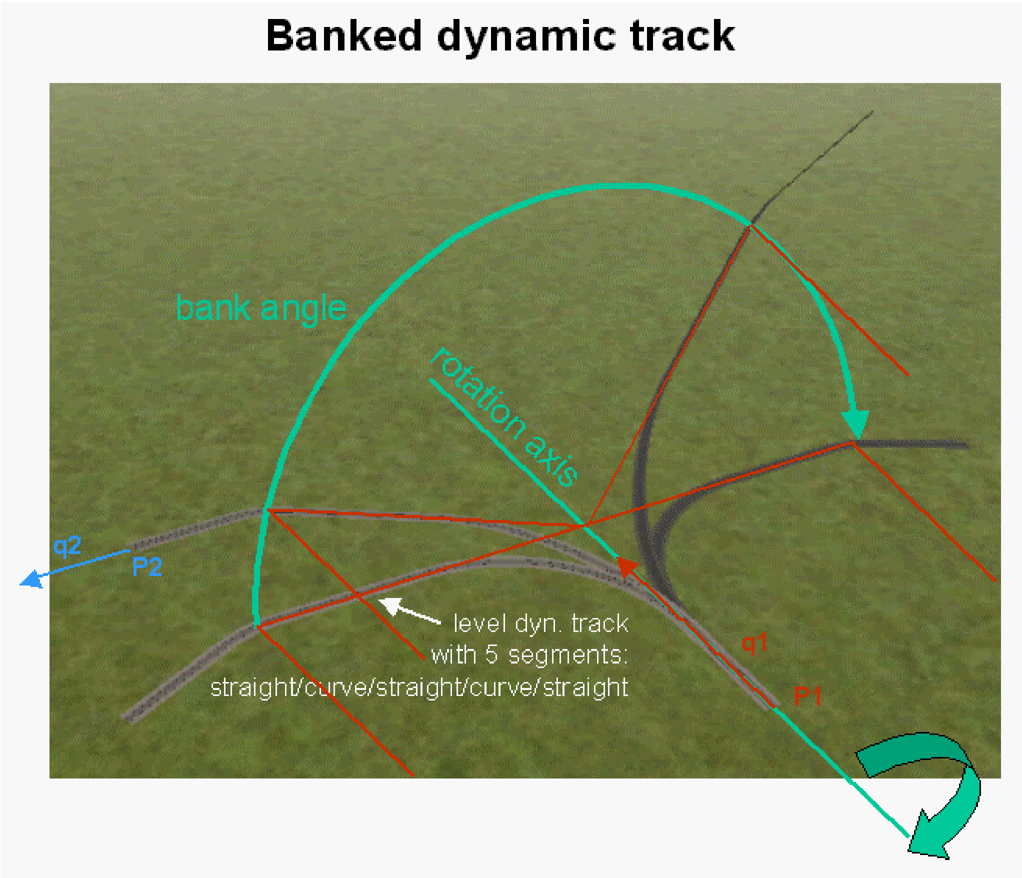


Unlike regular track (figure 11.2), the rails and railbed of a dynamic track are drawn non-banked all along the sloped track in MSTS: there is no superelevation. So the two rails are at the same level at each point along the track. This is good, because trains also ride non-banked: so the train's wheels will always be level with the rails. Strictly speaking, it is only the centerline between the two rails that remains in that inclined plane: the rails themselves twist to stay level everywhere.

However, sloped dynamic track is drawn, rather unrealistically, with a reduced gauge! The gauge (the distance between the rails) decreases as the track slope increases. In fact, when dynamic track becomes nearly vertical, its gauge becomes almost zero. Fortunately, the gauge changes very smoothly with the slope, so that this behavior is not too disturbing; also, trains don't suffer from a reduced gauge.

Banking is similar to sloping, see figure 27.4: the entire set of 5 segments again rotates rigidly, but now around an axis that is PARALLEL to its first straight segment. You may use positive as well as negative bank angles (from +180° to -180°): positive angles are shown in figure 27.4. Again, it is the centerline between the rails that stays in one plane, while the railbed and rails twist so as to stay level everywhere. And the gauge (distance between the rails) also decreases with increasing bank angle until the gauge is zero for vertical track.

FIGURE 27.4. BANKED DYNAMIC TRACKS:
the bank of a dynamic track section rotates its (red) plane rigidly;
the rotation axis goes through the start of the track section and lies PARALLEL to its first segment;
shown are bank angles of 0° (bottom left), 20°, 120° and 180° (right);
this example uses the same dynamic track section as figure 27.3



With a bank angle over 90°, the track goes upside down, as shown at right in figure 27.4: the trackbed therefore appears shaded, but fortunately the rails are still drawn above the trackbed. A bank angle of 180° tips the track over, so that it lies level and upside down, forming right-hand curves! I often use this property of dynamic tracks to **create right-hand curves by strong banking**, rather than the more usual reversal (by pressing T in RE). Fortunately, trains run perfectly on upside down tracks, without themselves riding upside down: the trains stay upright!

A very useful property of banking is this, because it gives much freedom of track design: **banking does not change the joint between dynamic track sections**. Imagine a dynamic track section ending at point P1 in figure 27.4, having the same direction there as the red arrow (horizontal in this case). By default, placing the section shown in figure 27.4 will create a perfect joint, since it will also be horizontal, with horizontal left turns. If you now bank this new section to any degree (such as shown in figure 27.4), the match at P1 will not change at all: the railbed and rails will remain perfectly level and smooth across the joint, no matter how much you bank the new section. More generally, you get a perfectly smooth match between dynamic track sections if they have both the same heading and the same slope at the joint, regardless of their respective degrees of bank.

As figure 27.4 shows, pure banking (without sloping) in fact also causes the track to have a slope, just as sloping causes banking. This gives you the freedom to **use both sloping and banking, separately or together, to shape dynamic track into rollercoasters**.

To summarize these **important properties of dynamic track sections**:
- a dynamic track section is sloped and banked rigidly in a plane;
- right curves ("upside down left curves") can be obtained by banking by more than 90°;
- with dynamic tracks, rails are not "banked" (superelevated);
- the joint between dynamic track sections is perfectly smooth as long as both sections have the same heading and slope at the joint, regardless of their respective bank angles;
- sloped and/or banked dynamic tracks have a reduced gauge, which varies smoothly;
- trains will always ride upright (never banked and never upside down).

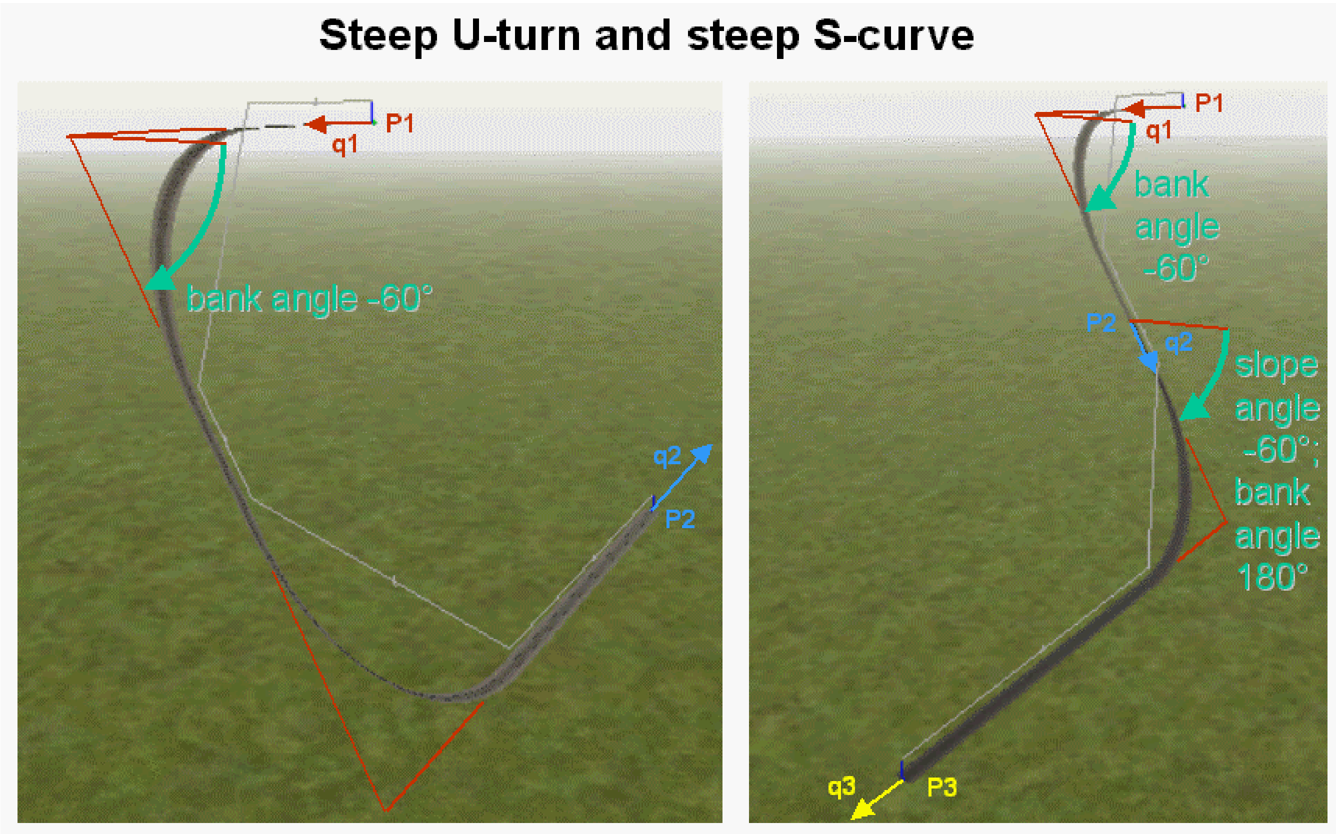
Simple steep curves made with dynamic track sections

You can already achieve much variety in a rollercoaster with relatively simple dynamic track sections. Figure 27.5 shows two basic shapes that can be used with many variations: a steep U-turn and a steep S-curve. Another simple shape is the S countercurve, discussed further below. (My MegaCoaster route, shown in figure 27.1, was built entirely with only these track shapes.)

The main advantage of these simple track shapes is that they can start from horizontal track and lead again to horizontal track with the same or opposite direction: that way you can keep good control over slopes and directions; also the default placement of the next track section will automatically provide a perfectly smooth horizontal joint, and banking the next track section will not damage that perfect joint.

The **steep U-turn** is made with a single dynamic track section that contains two 90° curves: as shown in figure 27.5, it may also include any lengths of straight segments: one before the first curve, one between the two curves, and one after the second curve. Notice how this dynamic track section is planar, shown by the red corner lines.

FIGURE 27.5. SIMPLE STEEP CURVES MADE WITH DYNAMIC TRACKS:
at left, a steep U-turn is made with a single dynamic track section, banked;
at right, a steep S-curve is made with two dynamic track sections, the first banked and the second sloped;
90° curves are used



This dynamic track section (which starts at position P1 with orientation given by QDirection q1) is banked around the red arrow to produce the "slope" of its middle straight segment: in fact, the bank angle (-60° in the figure) causes a slope of equal angle (-60°) in that segment; a bank of +60° would make it slope up by +60°.

The end of the U-turn, at point P2, is again horizontal; it is oriented antiparallel to its start (QDirection q2 points opposite q1). Note that the two curves may have different radii.

The closer the bank angle comes to -90° or +90°, the steeper the middle segment becomes (and the narrower its gauge becomes). This causes trains to twist rapidly and can break their couplers. To reduce this effect, avoid very steep slopes (banks) or increase the turn radius of the curves.

The **steep S curve** in figure 27.5 is made with 2 separate dynamic track sections, so that the second can turn in the other direction (right instead of left). The first section contains only one curve of 90° (and 2 optional straight sections): its bank of -60° causes a track slope of -60° at its end (point P2).

To continue this track at point P2 requires a slope of -60°, so the second dynamic track section starts at position P2 with a slope of -60° (which is included in its QDirection q2). However, this second section would by default turn left (producing a U-turn), whereas it should turn right to produce the S: this is obtained by also banking it by 180°, since that twists it around the blue arrow into a right turn that ends up horizontally at point P3.

The track direction at point P3 (given by QDirection q3) is horizontal, and parallel to that at point P1 (q3 = q1). Note that the two curves may have different radii.

Another relatively simple track layout is the **S countercurve**, illustrated in figure 20.1. The S countercurve is similar to the S curve of figure 27.5, but it does not require 90° turns, giving more flexibility. In section 20, the S countercurve is made by actually reversing a dynamic track section (as if by pressing T in RE), rather than by banking it upside down, as is done for the steep S curve here.

The procedure to create steeply sloped or banked dynamic track sections is given in section 19: first place a dynamic track section (connected to existing track) in RE, specifying its segment lengths, radii and turn angles, and then save the route. Next copy its QDirection data (called q1 in the figures above) into the **Turn about <u>object</u> axes sheet of Object Rotator** and give it the desired bank or slope: the resulting QDirection data can then be copied to the corresponding *.w file, before reloading the route in RE. In the case of an S countercurve, do the same thing, but use the **S countercurve sheet of Object Rotator** (see also section 20).

More complex track shapes

The relatively simple track shapes discussed above (especially the steep U-turn and the steep S curve) have some limitations. For example, they do not allow you to make a track that keeps sloping up or down over long distances (as in a helix or screw), since they bring it back to horizontal every now and then.

But you can see from figures 27.3 and 27.4 that the end orientation of a dynamic track section (as shown by the blue arrows labeled q2) can have any heading and slope, allowing much more freedom of design. To make use of this freedom requires that you know this end orientation q2, so that you can give the next track section that same orientation q2 and create a perfectly smooth joint. This is the purpose of the **Rollercoaster curves sheet of Object Rotator** (see also section 19): it calculates the end orientation q2 and end position P2 of a dynamic track section defined by its start orientation q1 and start position P1, together with its segment lengths, radii and turn angles.

It is important to realize that the orientation q2 produced this way still allows further freedom in shaping the next dynamic track section: its bank angle can still be freely changed without damaging the perfect joint at point P2.

General guidelines for designing a successful rollercoaster in MSTS

The shape of rollercoaster tracks, especially its ups and downs, very much determines how a train will ride over it. The following suggestions can help for the case of MSTS:

- arrange for the altitude of successive peaks to generally decrease: if your train used no power, that would be essential, but even when using power, you must allow for friction and resistance in tight turns; so the optimum result will depend on which train is used, and the distance between peaks! the final train speed at the end of the rollercoaster will also depend very much on the train type and length, so plan accordingly;

- do not expect your rollercoaster to return to its start position (closed circuits don't work in MSTS anyway): it will have lost too much altitude to return to the start altitude, except through a very long and slow climb;

- start your rollercoaster far above sea level, to avoid dipping below sea level (it is harder to shape terrain below sea level than above it, since the Y key press in RE does not function well below sea level, especially with twisted dynamic tracks!); count at the very least about 100 m total altitude change for a single near-vertical loop (often double that or more); my MegaCoaster starts at an altitude of about 2000 m and drops to near sea level, giving speeds over 400 km/h (250 mph);

- avoid too sharp curve radii (stay above about 50 m), especially with strong bank angles: they produce large twists between cars and can easily cause couplers to break;

- test the route at each construction stage in MSTS with trains, especially after reaching the next peak, to make sure your trains survive your route;

- if you set a high speed limit for the route (such as 500 km/h, or anything else well above the maximum speed that a train will reach), the route can be run equally with the MSTS derail option switched on or off;

- MSTS allows changing the strength of gravity, by using the line GravityScale (1.0) as the last statement in the route's *.trk file; however, I have found that this has little effect on the train's motion and speed; I would expect less acceleration and lower speeds in low gravity, as on the Moon, and more acceleration and higher speeds with stronger gravity, as on Jupiter; the statement GravityScale (1.0) corresponds to Earth's gravity (1G). Change the value 1.0 to 0.1653 for the Moon or 2.643 for Jupiter, the planet with the strongest surface gravity;

- trains will NOT lift off from the track when speeding over a hump (the way a race car or motorcycle would jump off the ground over a hump): trains are "glued" to the track, except when derailing.