Q & A Summary - Switch Geometry

February 14, 2017

This document is a result of a Q & A meeting that took place on February 3rd 2017 between Claus Feyling along with two employees of RailComplete AS and Vasco Kolmorgen & Christian Rahmig representing railML. The main discussion was centred around the definition of switches, and how these can be accurately modeled. The wiki page on modeling switches (`https://wiki.railML.org/index.php?title=Connection_between_tracks`) seems to be somewhat outdated. To stimulate discussion in an attempt to reach a common and concise framework for switch modeling we here present a overview of possible switches, along with a table summarizing the different properties and the equivalent railML representation. The document will not include crossings or slip switches, but these will be discussed in a later document.

Switch Geometry

The first concept to define is *switch geometry*. This term simply states the physical shape of the switch, and is completely separated from any topological information. Geometry is a property of a switch representing the physical shape that is built and marketed by the switch producer, not yet knowing where to use the switch. The switch geometry [1] may be defined in any standard format used by switch factories, usually supplied as a paper drawing or electronic drawing with material definitions and detailed dimensions shown. An ordinary two-way switch comes in two distinct geometries; left and right, also known as left-handed and right-handed switches:

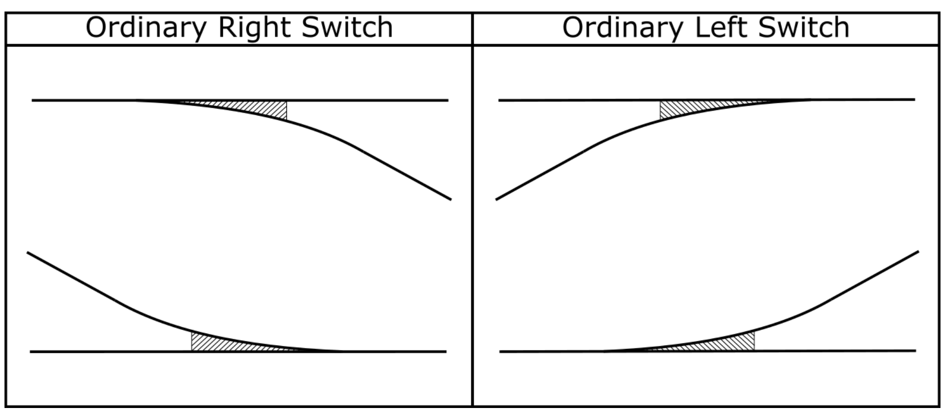

Figure 1: Ordinary Switch Geometries

while curved two-way switches come in three categories; Inside Curved Left, Inside Curved Right and Outside Curved. Note that the orientation or alignment of the tracks does not affect the switch geometry.

[1] Switch geometry can also be supplied in LandXML(r), or in railML whenever a switch library is needed, e.g. for automated recognition of standard switches based on actual track alignments.

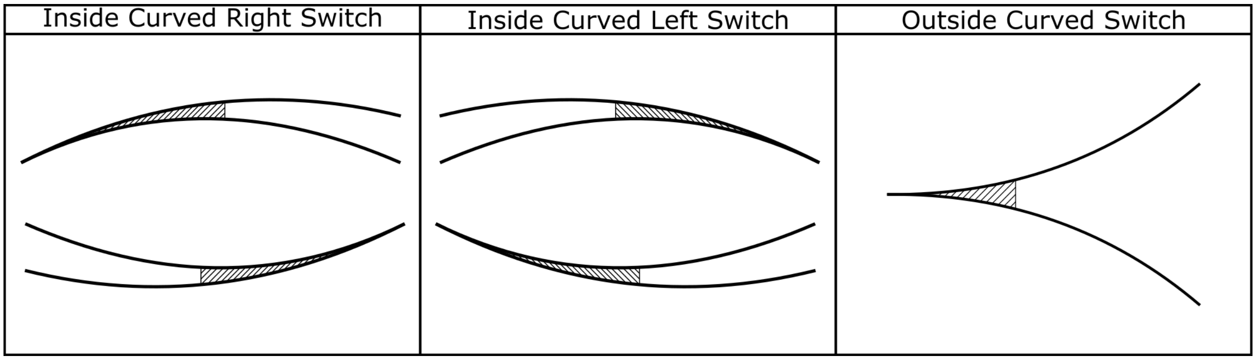

Figure 2: Curved Switch Geometries

Branching

Two-way switches are further characterized by their *branching direction.* "Branching" expresses a topological property of a switch when planning its use together with the mathematically defined alignments (track geometries). When planning the switch's position and its usage in a concrete railway project, the planner "slides" the switch along a chosen track, called the 'continuing track'. The other track branches off from the continuing track in a topological construction named a 'connection'. This track is called 'the branching track'. A branching track's alignment either starts or ends in a connection, and has no common geometry with the continuing track (or any other track, for that sake). Note that 3-way switches have two branching tracks. A branch track may of course play the role as a continuing track for another switch, but not exactly where the first switch's connection is located. For two-way switches, each switch geometry can branch either left or right, resulting in a total number of ten distinct possibilities. Ordinary two-way switches give the following possibilities:

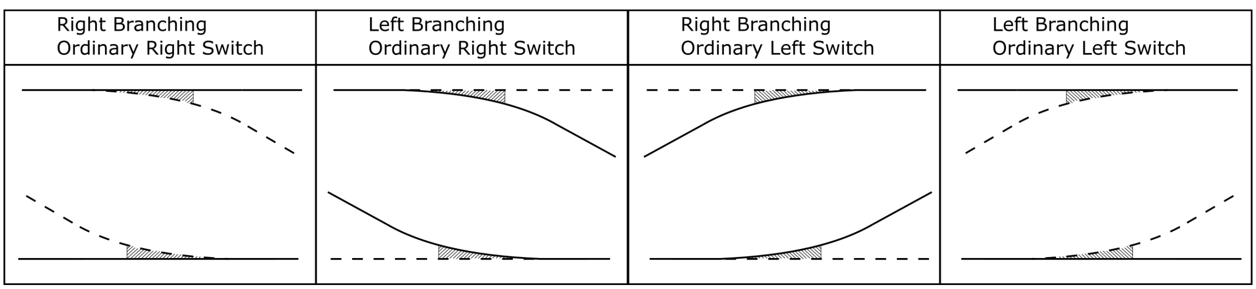

Figure 3: Ordinary Branching Switches. The dashed lines represent the branching track, while the solid lines represent the continuing track

Incoming and Outgoing

The final property to be assigned is whether the switch is an *incoming switch* or an *outgoing switch.* This results in two additional distinctions per branch, and a total of twenty distinct two-way switch situations. For an Ordinary Right Switch, the four possibilities are:

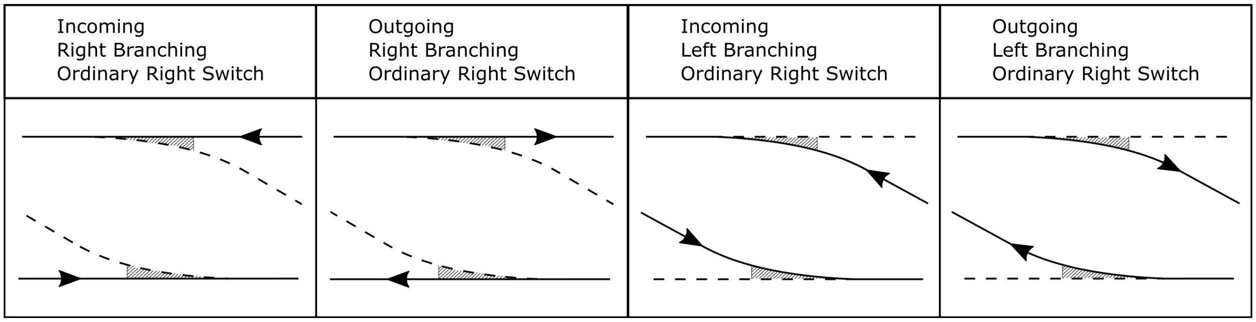

Figure 4: The four distinctions for an Ordinary Right Switch. The arrow represents the direction of increasing mileage

A .pdf file with names and graphic representation for all the twenty distinctions is given as an attachement to this document.

Three-way switches

Three-way switches come in three distinct geometries: left-handed, right-handed and symmetrical. The geometry is determined by the branching of the switch, when looking ”into” the switch.

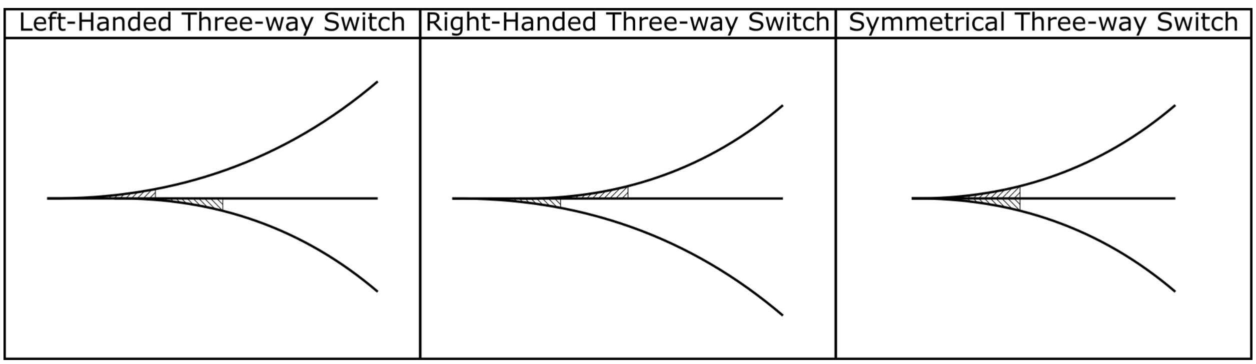

Figure 5: The three geometries for a three-way switch

The switches are further defined by their continuing track, (left, right or straight) giving three possibilities for each three-way geometry. Finally, each configuration again comes as both incoming and outgoing, resulting in eighteen distinct three-way switch situations. Again, see the attachment for a full overview.

Defining the properties in railML

The following table summarizes how the defining properties of the switches are represented in railML in its current state (railML 2.3). Please note that this is not a complete list of the properties of a switch, but rather a list of the properties relevant to the current topic.

Property	**railML implementation**	**Possible Values**
	Under <switch>:	

Switch Type	*type*	`ordinarySwitch`
		`insideCurvedSwitch`
		`outsideCurvedSwitch`
		`threewaySwitch`
		`other`

Continuing Track	*trackContinueCourse*	`straight`
		`left`
		`right`
		`other`

	Under <connection>:	

Connection Course	*course*	`straight`
		`left`
		`right`
		`other`

Incoming/Outgoing	*orientation*	`incoming`
		`outgoing`
		`rightAngled`
		`unknown`
		`other`

Comments

- In the current railML schema, there is no clear way of defining switch geometry. Given the type, a property with values {`Left, Right, Symmetric`} should satisfy the necessary information needed for specification of switch geometry.
- A switch should be a defined as belonging to one and only one track, that has to coincide with the *trackContinueCourse* property. The switch can be positioned at any point along the track:

$$\text{track.pos}_{\text{start}} \leq \text{switch.pos} \leq \text{track.pos}_{\text{end}},$$

  where $\text{track.pos}_{\text{start}}$ denotes the point where $\text{track.pos} = 0$, and $\text{track.pos}_{\text{end}}$ denotes the length-of-alignment for the given track.
- Incoming and outgoing switches are strictly defined by direction of increasing mileage.
- To improve integration with OpenRail, orientation could be reworked to have values {`up, down, both, unkown`}. (This may cause confusion with British terminology, e.g. "The train is travelling up towards London")

Postulates

- Any branching tracks have to either start or end at the switch. The connection in a switch is a topological feature that railML employs to model the switch, and contains information about which switch-leg it coincides with. A two-way switch will always have one connection, while a three-way switch will have two. An important restriction is that a branching course can not be equal to the continued course of the track.

$$\text{switch.continuedCourse} \neq \text{connection.course}$$

- In all two-way switches the position of the connection has to be identical to the position of the switch

$$\text{connection.pos} = \text{switch.pos}$$

- In a left-handed or right-handed three-way switch, the position of the first connection has to be identical to the position of the switch. The position of the second connection has to be equal to the position of the switch plus the distance between first and the second tongue:

$$\text{connection1.pos} = \text{switch.pos}$$

$$\text{connection2.pos} = \text{switch.pos} + \delta,$$

  where δ denotes the distance between the first and second tongues.

- In a left-handed or right-handed three-way switch, the continuing track has to continue past the position of the switch by at least δ to define a second connection.

Restrictions

- All tracks connected to a track through switches must branch off from the continuing track. This means that for a left-handed three-way switch, the continuing track can never be the track branching right, and vice versa for a right-handed three-way switch.

Additional Switch Properties

This section will cover some properties that are not currently modeled by railML. We present these properties as possible candidates for future implementation.

Constituent Parts of a Switch

In order to be useful, a switch must have moving parts which will direct a train to the left, straight or right leg when running in the opposing direction. The part of the rails carrying out this function is known as the "tongues", or "switch blades". The tongues may be welded to the rest of the switch, speaking of "feather tongues", or they may be mechanically hinged, known as "hinged tongues" - the latter often being used for very short or for very stiff tongues.

The tip of tongue should ideally coincide with the mathematically defined location where the branch track bifurcates from the continuing track, but this would lead to hopelessly thin metal parts, or to mechanical problems due to metallic expansion and retraction with seasonal temperature variations.

The sleepers in a switch may be placed at an angle to both the left and the right leg, or perpendicularly to one of the tracks, usually the straightest one. The fixation of rails to sleepers must take place where it is possible, in between guards rails, diamonds, noses and other parts of a switch. The placement and type of sleeper may vary throughout a switch, at least the length of each sleeper will usually vary. The rails may be fixed flat to a flat sleeper (i.e. rail inclination 1:oo), or they may be inclined (usually 1:20), and there may be transition zones where the rail inclination is gradually changed from 1:20 to 1:oo again.

The tongues may move on so-called frictionless slide plates, or they may be lifted slightly up when moving and roll on tongue roller bearings.

The switch assembly may use any of several techniques for holding the rails in place, providing both holding force, mechanical damping and electrical insulation.

railML does not model tongues, sleepers, rail inclincation or rail fixation.

Isolation Joints - Switch Rail Numbering System

Many rail administrations use the rails as part of a train detection device known as track circuit. Somewhere in the vicinity of a switch there must be isolated joints in two or more rails, in order to separate track occupation in one leg from the other leg. One system for specifying the exact location of isolated joints in a switch is to measure the distance from the stock rail joint (at the start of curvature for the most curved leg) along the track's centre line to location, and then denominate the rails as "1" on the far left side to "4" on the far right side (or "6" in the case of a three-way switch), when seen in the opposing direction.

Please note that the middle rails (the left leg's right rail and the right leg's left rail) change number identity after the switch nose, i.e. after the location where those rails meet.

railML does not model such isolated joints as part of the switch, but as elements of the individual tracks.

Tongues and Normal Position

A switch will usually have a designated "normal position". The normal position denotes the course that a train will take when running in the opposing direction and the switch tongues are somehow held in this normal position.

The normal position is defined by the signaling engineer, and does not necessarily reflect the actual running course taken by most trains. However, the engineer will usually end up with defining the normal position to coincide with the running course for the main intended traffic over that switch. As a special case, a switch may be symmetrically arranged between two equally used tracks on a double-track line or in a narrow spaced station, in order to provide flank protection for both tracks. Such switches, called "double protecting switches" ("Zwieschutzweichen"), will have to move to its flank protecting position depending on which track that now has a passing train. I.e. it is as much needed in its right as in its left position.

The term "normal" relates to relay drawings for a railway implementation. Relays are still necessary, even with electronic interlockings. A relay will always have a "normal" position. Depending on the relay construction, this normal relay position may be closing an electrical circuit or opening it. It is customary to present all the electrical wiring in a wiring diagram assuming that every relay is open or closed in a way that corresponds to the tongues being held in their designated normal position. The normal position for a relay may be its electrically activated state or its electrically passive state, depending what is deemed most fail-safe by the construction engineer.

Tongues are prone to wear and tear and will often be changed several times over a switch's lifetime. It should therefore be useful to model tongues as a maintenance object separate from the rest of the switch, such as isolated rail joints are - but railML does not model tongues.

It is actually the tongues that are in or away from the normal position. Since railML does not model tongues directly, we will instead speak of a switch as being in its "normal position" or in its "deviating position". A three-way switch has two distinct deviating positions.

railML models normal position as a property of a switch.

Nomenclature and Common Definitions

This closing section of the document presents some definitions of more in-depth properties of switches. This can hopefully be adopted as a common reference for railML when discussing future changes or new implementations.

Further Concepts in Geometry and Topology

The concepts of "left" and "right" are related to the running direction through a switch. When speaking of a switch's inherent geometry in ordinary language, we assume the perspective of a driver encountering the switch such that it gives him a choice of two or three further courses. If the switch's tongues (the moveable switch parts) force the train to take the leftmost course available, then the train will be said to follow the "left leg" of the switch. The other leg of an ordinary switch is called the "right leg". Three-way switches also have a "straight leg" between the left leg and the right leg.

It is customary to design switches such that the geometry can be built in one piece at the factory, maybe broken down into smaller parts that are transported individually to the mounting site and assembled during a short line possession. In railML, we do not model these individual parts, but only the switch as a whole.

It is often seen that two switches are so close to each other that they actually have no ordinary sleepers between them. In railML we model these as two separate switches, but they might have non-standard geometries.

From time to time there is a need to supply a rail crossing with four switches between two parallel main tracks or station tracks. The crossing being very close to all four switches leads to one huge steel-and-concrete construction being designed as one piece at the factory, and assembled later on-site. In railML, we model this construction as one crossing and four individual switches, maybe with non-standard switch geometries.

In railML, we do not restrict our modeling to "flat" switches only, i.e. switches where all geometry belongs to a plane in Euclidian space. railML models switch geometry with individual alignments for the continuing track and the branching track(s) (see definitions below), and these may differ both in vertical profile and in superelevation (cant), at least in the section following the switch nose. A fairly common example is a curved crossover, i.e. a track connecting two main tracks running side by side through a sharp curve. In this case, all tracks are 'rolled', i.e. they share the same superelevation at every position seen perpendicularly to the main tracks. In practice, such switches will actually be built somewhat "concave" due to the occurrence of simultaneous radius and superelevation, and they can not be built without some "twisting and bending" if standard switches are used. These switches are highly non-standard and generally not recommended to use.

It is customary that the mathematical alignment for the straight leg of a three-way switch is actually a straight line in Euclidian space, but this is actually no requirement in railML.

When a switch is encountered such that it gives you a choice of courses, we say that the switch is encountered in the "opposing direction". In the opposing direction, a train will pass the stock rail joint before the end of the point.

If a switch is encountered such that you are forced into the same course, the we say that the switch is "trailed", or encountered in the "trailing direction".

Some switches are mechanically built to withstand numerous trailing incidents, others are not. We then speak of "trailable switches" or "non-trailable" switches. If the switch is equipped with actuating devices which move tongues, also known as "point machines", these will correspondingly be called "trailable point machines" or "non-trailable point machines".

Most modern switches are equipped with point machines. These will usually be located either on the left side - as seen in the opposing direction - or on the right side. We then speak of a "left mounted point machine" or a "right mounted point machine", no matter what the increasing mileage direction is or what the actual passing trains running direction is.

Equipment mounted on the same side as the point machine (the motor) is said to belong to the "motor side". Equipment belonging to the opposite side belong to the "opposite motor side".

Milage and Absolute Position

Railway positioning systems usually define an ”absolute position”, being a numerical name for a certain spot in a given track on the railway network, usually measured perpendicularly to one nearby track, from a well-defined starting point.

Stationary equipment - everything that is not rolling stock - is said to be ”located on the left side” of a given track whenever you can reach its location having the starting point for mileage counting behind your back and the equipment in question on your left side when you pass it, as you drive along that track in the increasing mileage direction. ”Right side” is similarly defined.

When train drivers speak of ”left” and ”right”, they must also specify the location (station) they are departing from and the location (station) they are arriving at in order to have a unique identification of side-of-track. An alternative to specifying the departure and arrival locations is to speak of ”up” and ”down” directions, where ”up”, for a track, is always the running direction where mileage count increases, and the ”down” direction is the running direction where the mileage count decreases.

The trains meant to operate on a given track may have a preferred running direction. In that case, we speak of the ”main direction” being ”up” if the normal usage for that track is running in the direction of increasing mileage, and ”down” if the normal usage for that track is running in the direction of decreasing mileage. Double track lines usually have one track with ”up” as main direction, and the other one as the ”down” direction.

To complicate matters somewhat, the British-speaking community will speak of ”going up to London” in ordinary language. This is contrary to railML definitions and should therefore be discouraged, as long as London is located at the lower end of the mileage counting system for that network.

Radius and Divergence

The radius for a given leg in a switch denotes the constant radius that the leg's mathematical alignment follows. An ordinary switch has one straight leg and one curved leg, but a double curvature switch will have two curved legs.

At a connection, the alignments for the two tracks that connect must share tangents in Euclidian space.

Curvature in railML is expressed as the curve radius, in meters, for a curved track, or as ”0” for an infinite curvature, i.e. a straight leg.

The ”divergence” of a switch relates to the direction of the tangents for the two (or three) legs where the curvature (or the switch itself) ends, away from the connection. The angle between two neighbouring legs' tangents can be expressed as the number of length units you move along the straightest leg's tangent for each unit extra separation between the legs, as measured perpendicular to the straightest leg's tangent line. I.e., the divergence is atan(1/N) where 1:N is the divergence. The number N is actually referred to as the switch's ”number” in some countries.

The geometrical point where the two track tangents meet is called the ”theoretical crossing”. In ordinary switches, the theoretical crossing is located on the straight leg's center line. In doubly curved switches, the theoretical crossing is located outside the switch geometry. Left- and right-handed three-way switches have two theoretical crossings, whereas symmetrical three-way switches have only one theoretical crossing.

Length

A switch geometry always includes at least one curved leg. The running distance along the fictitious center line for this leg's alignment, from start of curvature (the connection) to a location named the ”end” is called the ”length” of the switch. However, the track may be continued with a straight continuation or with the same curvature from the end of the switch, or it may finish a clothoid or another type of transition curve. For each such continuation, a different type of switch must be ordered if its divergence is so tight that the two legs can not be built as separate tracks on site. railML does not model switches at this level of detail. The legs of a switch will have to follow the individual track's alignments anyway, which is expressed in LandXML(r) and in railML as curves and lines belonging to an alignment.

Switch manufacturers will have a complete list of lengths describing the various parts of a switch containing tongues, curvature, long sleepers, short sleepers etc. These lengths are not modeled in railML.

In fact, the switch length property in railML is rarely, if ever, needed. It was the result of a legacy description of a switch as a being part of the rails, instead of being a point object with certain constraints relating to the legs having to coincide with the actual alignments it is built for. Modern railML (from 2.x) sees a switch as a point object (with a second, related, connection in the case of a three-way switch).