CERN - European Organization for Nuclear Research

Draft Version r5

How To Create GEAR XML Files For LP1

Martin Killenberg*

* *CERN, Switzerland*

November 22, 2012

Abstract

An introduction how to apply the GEAR XML parameters for the LP1 detector geometry.

1 The GEAR Parameters

In GEAR, originally the concept of modules was not known. There was one pad plane per end plate, and the coordinate system of the pad plane automatically was the global coordinate system. The LP1 required the placement of seven modules on the end plate, each with it's own pad plane. The TPCModule was introduced to allow this. Each module contains a pad plane and allows the displacement and rotation of the pad plane's coordinate system on the end plate. In fact the TPCModule is an abstract wrapper object which provides just this: A coordinate transformation between the global coordinate system of the end plate and the local coordinate system of the pad plane in the module. It is not a description of the physical module frame!

For each module there is a pad plane with a local coordinate system. For modules with circular pad rows (like most modules for the LP1) the local coordinate origin is in the centre of the circles. Obviously only concentric pad rows can be described with this. The local coordinate system is in polar coordinates (r, φ). But also rectangular pads in straight rows are possible (RectangularPadRowLayout), which would have Cartesian local coordinates (x, y).

The global coordinate system can also be either Cartesian or polar. Global polar coordinates are only useful for backward compatibility: If there is only one pad plane and the local and the global coordinate systems are identical.[1)] In this case the modular TPC behaves exactly like the old, non-modular version.

For the LP1 the only reasonable choice is global Cartesian coordinates.

1.1 TPCParameters

For the large prototype, the TPC's `coordinateType` should be set to `cartesian`, as global polar coordinates with shifted local polar pad planes do not make sense.

For the prototype reconstruction the `maxDriftLength` parameter is not used in the reconstruction, as $z = 0$ is at the readout, not at the cathode like for ILD-like collider TPC with two end caps. In the existing examples it is set to 600.0 mm.

The `driftVelocity` parameter of GEAR should never be used. This value comes from conditions data.

1.2 TPCModule

When placing a module, the position of the local origin in the global coordinate system can be given, as well as a rotation of the local $\varphi = 0$ reference line around the local origin. For a local polar coordinate system in Cartesian global coordinates this is visualised in figure 1.

The `offset` parameter has two options: `x_r` and `y_phi`. They represent the position of the local origin in global coordinates. As GEAR allows Cartesian or polar global coordinates, the pair `(x_r, y_phi)` can be either interpreted as x and y in case of global Cartesian coordinates (LP1 end plate), or as r and φ in case of global polar coordinates (usually not useful). So for the LP1 case the relevant parameters are:

[1] If all the modules are concentric to the same origin and there is no (mis)alignment, global polar coordinates might also be OK. Although also here Cartesian might be preferable.

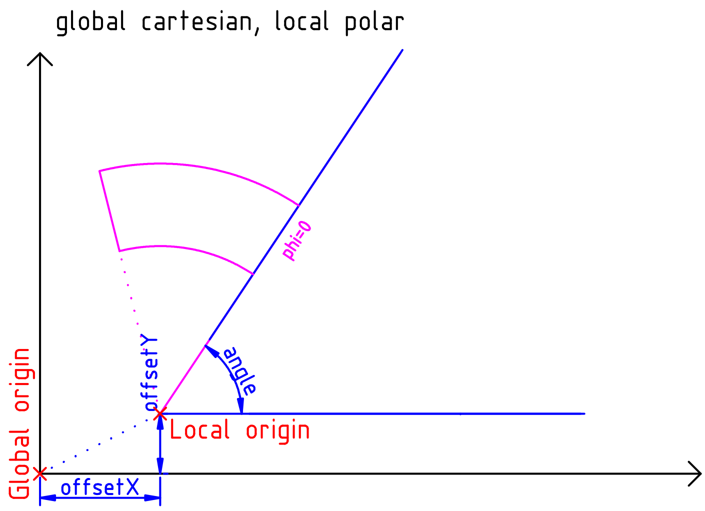

Figure 1: Offset and angle of a local polar pad plane in global Cartesian coordinates. In this example the local $\varphi = 0$ is the right edge of the module, so all local angles are positive. This is not necessarily the case.

`x_r`	The x value of the offset in mm
`y_phi`	The y value of the offset in mm
`angle`	The rotation angle in radian

1.3 VersatileDiskRowLayout

The VersatileDiskRowLayout describes keystone shaped pads, arranged in concentric rows. The layout is created row wise, starting from the inner radius, which is the lower edge of the first pad row. The pad geometry is always described by `rowHeight` and `padPitch`, which describe the effective pad dimensions. `padWidth` and `padHeight` are the dimensions of the metal, without the gaps. They are kept only for backwards compatibility and are irrelevant for the reconstruction. The pads are aligned so the right edge of the pad pitch of the first pad (padNumber 0 in the row) is at the local $\varphi = 0$ line. They can be rotated by an offset.

All values are given in mm.[2)] `padPitch`, `offset` (and `padWidth`) are measured in the middle of the pad row, along the circumference of the corresponding circle. All offsets are measured with respect to $\varphi = 0$, which is the zero of the local coordinate φ. The pads are arranged in positive φ direction, starting at the "right" edge of the first pad (pitch).

[2] In the XML syntax and the constructor the padPitch, offset and padWidth are set in mm. Note that the getPadPitch() and getPadWidth() functions of the C++-class are derived from PadRowLayout2D and answer in radian, as it is a local polar coordinate system.

The parameters are visualised in figure 2 and summarised in table 1. For visualisation purposes the $\varphi = 0$ line (local coordinate system) has been rotated. In GEAR, this rotation with respect to the horizontal axis is a parameter of the module (not shown in the picture because it is not a parameter of the VersatileDiskRowLayout).

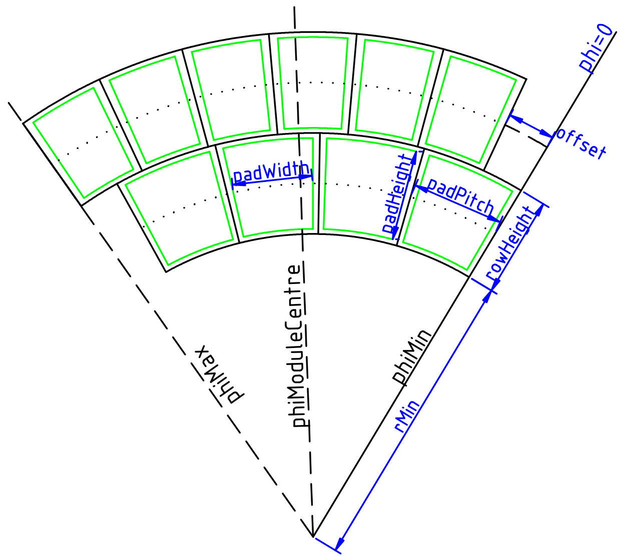

Figure 2: Visualisation of the XML parameters (blue) of the VersatileDiskRowLayout. In this example `phiMin` coincides with `phi=0` because the first row does not have an offset, and the second offset is positive. This is not necessarily the case. `phiMin` reported by the plain extent is $\min([offset_i/r_i])$, `phiMax` is $\max([(offset_i + nPads_i \cdot padPitch_i)/r_i])$. The central angle `phiModuleCentre = (phiMax+ phiMin)/2` is only shown for illustration. It is never used or calculated.
For visualisation purposes the $\varphi = 0$ line (local coordinate system) has been rotated. In GEAR, this rotation with respect to the horizontal axis is a parameter of the module (not shown in the picture because it is not a parameter of the VersatileDiskRowLayout).

For the complete XML syntax please refer to the Doxygen documentation of GEAR (trunk version, currently being update).

1.4 FixedPadAngleDiskLayout

The FixedPadAngleDiskLayout is a much simpler design. All pads have the same angle $\varphi_{\text{pad}} = (\varphi_{\text{max}} - \varphi_{\text{min}})/nPadsInRow$, and all rows have the same height $(r_{\text{max}} - r_{\text{min}})/nRow$. In this abstract geometry there is no gap between the pads, so `padPitch = padWidth =` φ_{pad}, and `padHeight=rowHeight`. The required parameters are visualised in figure 3 and summarised in

Parameters for the layout itself	
`rMin`	The radius of the inner edge of the first pad row in mm.
Parameters for the rows	
`nPad`	Number of pads in the row.
`rowHeight`	Height of the row in mm.
`padPitch`	Pitch of the pad in mm, measured at the radius at the pad centre.
`padWidth` (optional)	Width of the metal of the pad at the central radius in mm. Has to be $<$ `padPitch`. If not given it is `padPitch`.
`padHeight` (optional)	Height of the metal of the pad in mm. Has to be $<$ `rowHeight` If not given it is `rowHeight`.
`offset` (optional)	Offset in mm with respect to the alignment axis $\varphi = 0$. Measured at the central radius of the pad.
`repeat` (optional)	Insert ”repeat” identical rows instead of only one.

Table 1: XML parameters of the VersatileDiskRowLayout.

table 2.

`rMin`	Inner radius of the inner pad row in mm
`rMax`	Outer radius of the outer pad row in mm
`nRow`	Number of rows
`phiMin`	Minimal angle in radian, measured from the $\varphi = 0$ reference line
`phiMax`	Maximal angle in radian, measured from the $\varphi = 0$ reference line
`nPadsInRow`	Number of pads per row

Table 2: XML parameters of the VersatileDiskRowLayout.

2 The LP1 Parameters

The parameters are taken from the technical drawings provided by Dan Petersen. [1] The bounding box is the reference for the construction, and has only lines pointing to the local origin, and circles concentric around this origin. The whole construction of the module row should be concentric around this origin.

The GEAR geometry is an abstract geometry. It's local origin is always the centre of the pad row circles. As only concentric geometry can be described with GEAR, not all elements in the physical module can be described (like the edges of the module which are shifted with respect to the edge of the bounding box). As the GEAR geometry is abstract, neither the bounding box nor the physical module are directly relevant for it. Only the positioning of the pads with the local origin, which is the centre of the circles describing the pad rows, is needed (plus the relative positions of the modules).

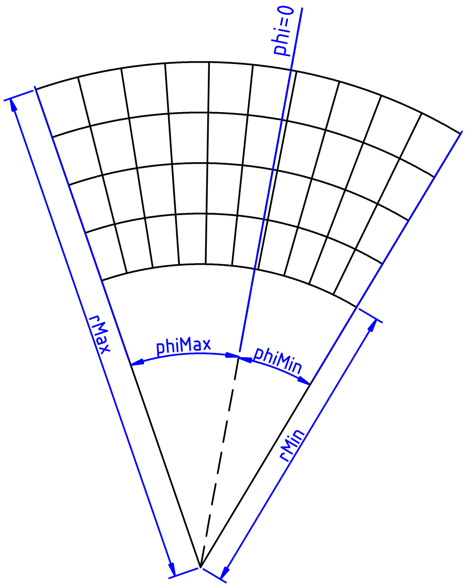

Figure 3: Visualisation of the XML parameters of the FixedPadAngleDiskLayout. In this example `phi=0` is an angle inside the module, so `phiMin` will be negative and `phiMax` will be positive. Note that the reference line might also be outside the pad plane, resulting in both angles being either positive or negative.

The generic choice of the local GEAR origin would the origin of the bounding box. Like this the pad rows are concentric with the module positions on the end plate, as it should be. Throughout the following this is always the case, unless mentioned otherwise.

2.1 The bounding box (page 11)

The bounding box is a segment of a circular ring. The only relevant value is the angle of the bounding box, which is 8.3870°. Inner and outer radius are only relevant in the sense that the rows must not exceed the radial range, which is always the case if they fit onto the physical module.

2.2 Placement on the end plate (page 12)

Note: The drawing on page 12 is to be read rotated by 180°, so the central angle becomes the global $\varphi = 0$. According to the drawing the view is from the outside.

2.2.1 Position of the module rows

The bounding boxes / modules are placed on the end plate in three rows. The modules in one row have the same local origin and are rotated by different angles. The origins of the three rows

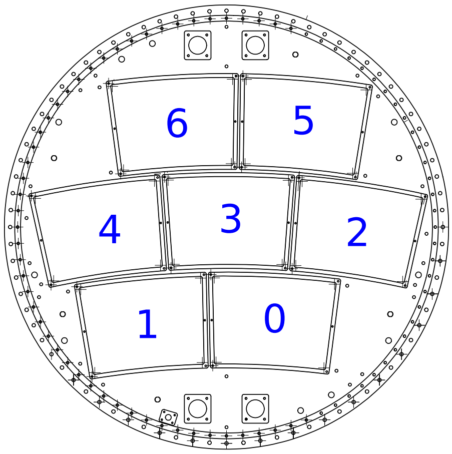

Figure 4: The proposed numbering convention for modules in the LP1.

are displaced in x.

The current versions of the GEAR files use the local origin of the middle row as the global origin. This gives the following offsets (the inner row is the one towards smaller local radii):

inner / lower row	-172.5 mm
middle row	0 mm
outer / upper row	+171.615 mm

An alternative solution would be to have the global origin in the middle of the end plate / TPC cylinder, resulting in these offsets for the rows:

inner / lower row	-1676.115 mm
middle row	-1503.615 mm
outer / upper row	-1332.000 mm

2.2.2 Rotation of the modules

GEAR does not provide any scheme for numbering the modules on the end plate. For the LP1 the recommended numbering is to count from 0 starting in the inner / lower row, counting up with increasing angle. The proposed module numbers on the end plate are shown in figure 4. The according angles of the central axis of the bounding box are listed in table 3.

3 Examples for GEAR modules in LP1

The following examples are meant to help getting a feeling what the parameters mean and how a correct description can be made. They probably have nothing to do with the real LP1 pad planes. In principle the pad geometry described by GEAR has nothing to do with neither the bounding box nor the physical module (see figure 5), although it likely is the case.

ModuleID	**row**	**position**	**angle**
0	inner / lower	right	-3.00°
1	inner / lower	left	5.39°
2	middle	right	-8.50°
3	middle	centre	-0.11°
4	middle	left	8.27°
5	outer / upper	right	-5.00°
6	outer / upper	left	3.39°

Table 3: The module angles, taken from the technical drawings.

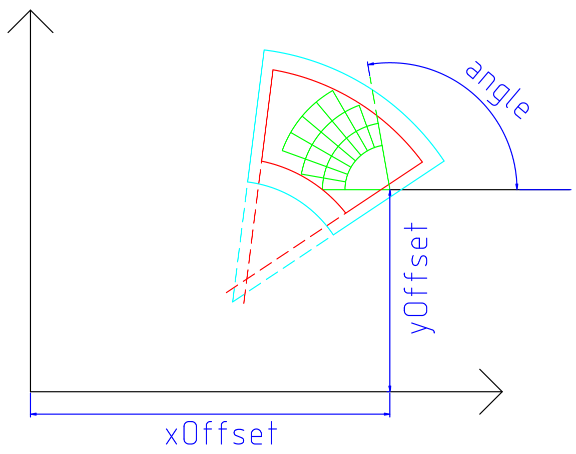

Figure 5: Someone got it wrong: Although completely contained in the physical module (red), the pad plane (green) has nothing to do with the module or the bounding box (cyan) geometry. It still can be described by GEAR, which just needs the local origin in the (arbitrary) global coordinate system, and the rotation angle of the local $\varphi = 0$ line.

3.1 FixedPadAngleDiskLayout

The description of a polar pad plane usually is ambiguous, because the choice of the $\varphi = 0$ line is arbitrary. Just for demonstration we give three different possibilities to describe the placement of a module 0 with an active area angle of $\varphi_{\text{active}} = 8.27°$. All three descriptions are correct, only resulting in different local coordinates. As the lcio file stores global coordinates they should give identical results. In the following examples the changing code is blue, the part which is common to all three examples stays black. All tree examples are visualised in figure 6.

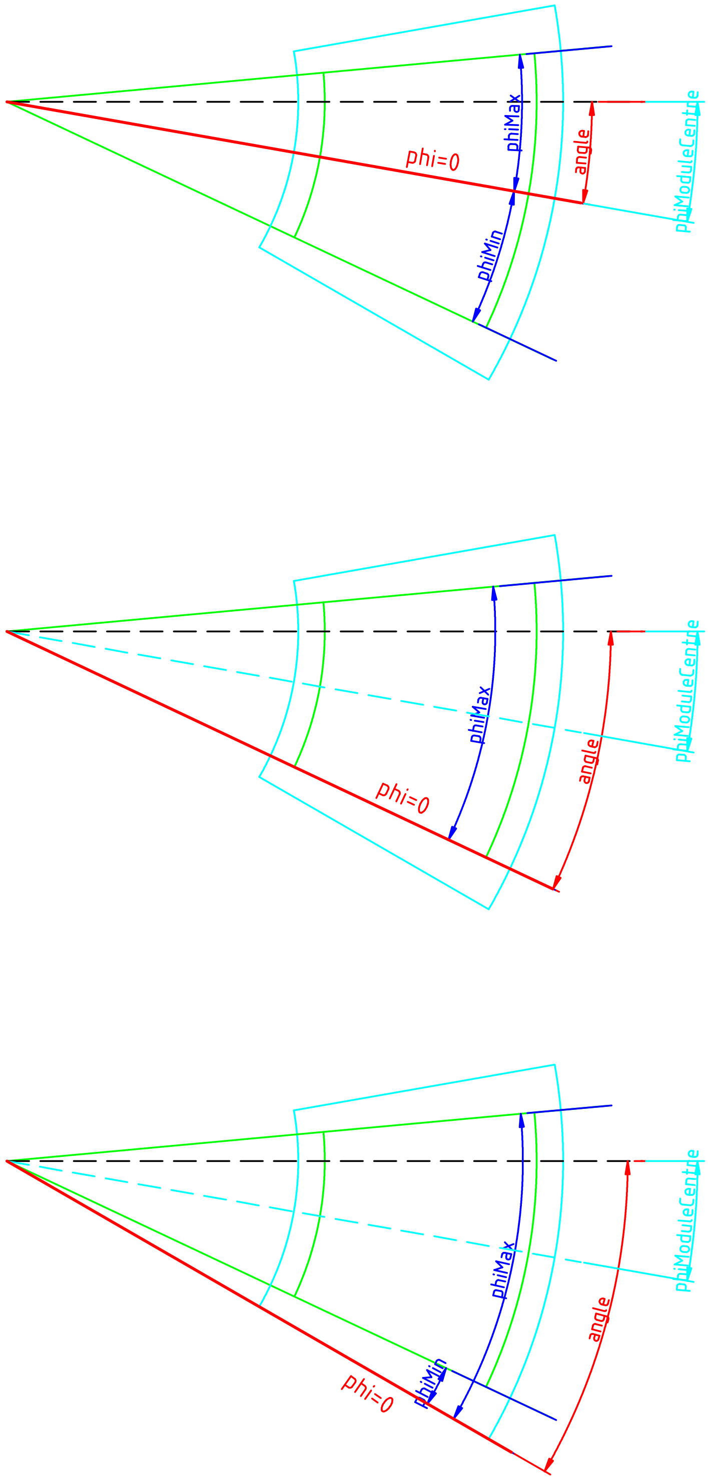

Figure 6: The visualisation of tree different choices for the $\varphi = 0$ line for a `FixedPadAngleDiskLayout` (not to scale with the example values).

3.1.1 Phi=0 as the central angle

As $\varphi = 0$ is the central line of the module, which is also the value given in the construction drawing, the calculation is easy: The rotation angle of the module just stays $-3.00°$ (see table 3), phiMin and phiMax are $\pm\phi_{\text{active}}/2$. We just have to convert them to radian. I just do this via cut and paste from the calculator. The angles certainly do not have 10 digits precision in reality.

```
<module>

  <!-- phiMin / phiMax = +- 8.27 deg /2 /180 * pi-->
  < PadRowLayout2D type =" FixedPadAngleDiskLayout"
    rMin ="1450" rMax ="1570"
    nRow ="14"
    phiMin ="-0.07216936457"
    phiMax =" 0.07216936457"
    nPadsInRow ="100" >
  </ PadRowLayout2D >

  <!-- angle = 3 deg /180 * pi -->
  < angle value ="0.0523598775598" / >

  <!-- Offset for the lower module row in mm -->
  < offset x_r ="-172.5" y_phi ="0." / >

</module>
```

3.1.2 Phi=0 as the right edge of the pad plane

When describing the same module position with $\varphi = 0$ at the edge of the module, the rotation angle must be adapted so that angle$= -3.00° - \varphi_{\text{active}}/2$.

phiMin becomes 0, as this is the edge of the pad plane where the reference angle is, and phiMax simply is the active module opening angle φ_{active}.

```
<module>
  <!-- phiMin=0; phiMax = +8.27 deg /180 * pi-->
  < PadRowLayout2D type =" FixedPadAngleDiskLayout"
    rMin ="1450" rMax ="1570"
    nRow ="14"
    phiMin ="0.0"
    phiMax ="0.14433872914"
    nPadsInRow ="100" >
  </ PadRowLayout2D >

  <!-- angle = (-3 deg - 8.27 deg /2) /180 * pi - -->
  < angle value ="-0.12452924213" / >
```

```
  <!-- Offset for the lower module row in mm -->
  < offset x_r ="-172.5" y_phi ="0." / >
</module>
```

3.1.3 Phi=0 as the right edge of the bounding box

For the FixedPadAngleDiskLayout this does not seem like a natural choice. For the VersatileDiskRowLayout, however it is useful because it is the line pointing to the origin which is closest to the right edge of the pad row, which is used for alignment. We just show it as a further example.

The $\varphi = 0$ line now has a rotation of $-\varphi_{\mathrm{BBox}}/2$ with respect to the angle from the drawing. The angles of the pad plane now have to be rotated to their correct position inside the bounding box, which are $\phi_{\min} = (\varphi_{\mathrm{BBox}} - \varphi_{\mathrm{active}})/2$, $\phi_{\max} = \varphi_{min} + \varphi_{\mathrm{active}}$.

```
<module>
  <!-- phiMin= (8.3870 deg - 8.27 deg)/2 /180 * pi -->
  <!-- phiMax = 8.27 deg /180 * pi + phiMin -->
  < PadRowLayout2D type =" FixedPadAngleDiskLayout"
    rMin ="1450" rMax ="1570"
    nRow ="14"
    phiMin ="0.00102101761"
    phiMax ="8.27102101761"
    nPadsInRow ="100" >
  </ PadRowLayout2D >

  <!-- angle = (-3 deg - 8.3870 deg /2) /180 * pi - -->
  < angle value ="-0.125401906756" / >

  <!-- Offset for the lower module row in mm -->
  < offset x_r ="-172.5" y_phi ="0." / >
</module>
```

3.2 VersatileDiskRowLayout

For the VersatileDiskRowLayout we use the bounding box as the reference, as it is the reference for the module construction. The physical module itself cannot be used in this case. Its outer edge is parallel to the edge of the bounding box, and thus not pointing to the local origin (see figure 7). This cannot be described in GEAR. We chose the right edge of the bounding box to be $\varphi = 0$.

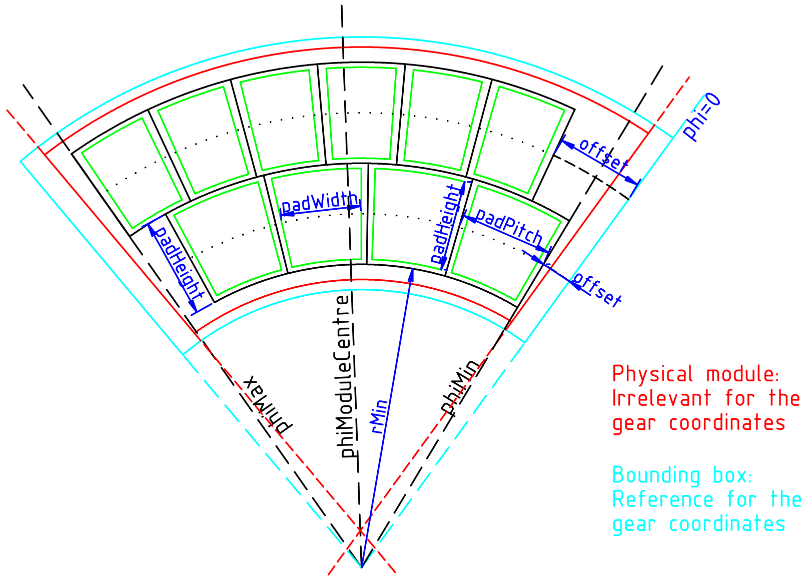

Figure 7: Example of a VersatileDiskRowLayout with phi=0 at the right edge of the bounding box (not to scale with the example values).

3.2.1 Example 1

Let's assume the pads are aligned with a distance of 1 mm with respect to the edge of the physical module,[3] which itself is 2 mm from the edge of the bounding box.[4] Every second row is staggered by half a pad pitch. So the offsets are 3 mm[5] (+ `padPitch/2` for staggering pads). Note that the pads do not form a line pointing to the origin (see figure 8 for visualisation).

We further assume that there is a constant number of pads in all rows and that the pads are arranged such that the gap on the left side of the module is also 3 mm to the bounding box. The pitch is varying accordingly from row to row and can be calculated as $\texttt{padPitch}_i = (\varphi_{\text{BBox}}/180° \cdot \pi \cdot r_i - 2 \cdot 3\text{ mm})$ / (`nPads+0.5`). r_i is the radius measured at the centre of row i. The +0.5 for the Number of pads is due to the fact that there should be the empty space of half a pad pitch either at the left or the right edge of the module. The following example is given for a row height of 6 mm, an inner radius of 1440 mm and 100 pads per row. The choice of 100 µm gap is not relevant. Pad height and pad width should never be used in the reconstruction.

[3] Numbers in blue are arbitrary choices in the example and have to be adapted.

[4] Values in red are taken from the technical drawings.

[5] To be correct the offset has to be measured along the circumference, while in this example the reference line is 3 mm parallel from $\varphi = 0$. Hence the correct offset is $r \cdot \arcsin(3\text{ mm}/r)$, which gives a bias of 2 nm for a radius of 1440 mm. For readability we approximate the offset with 3 mm.

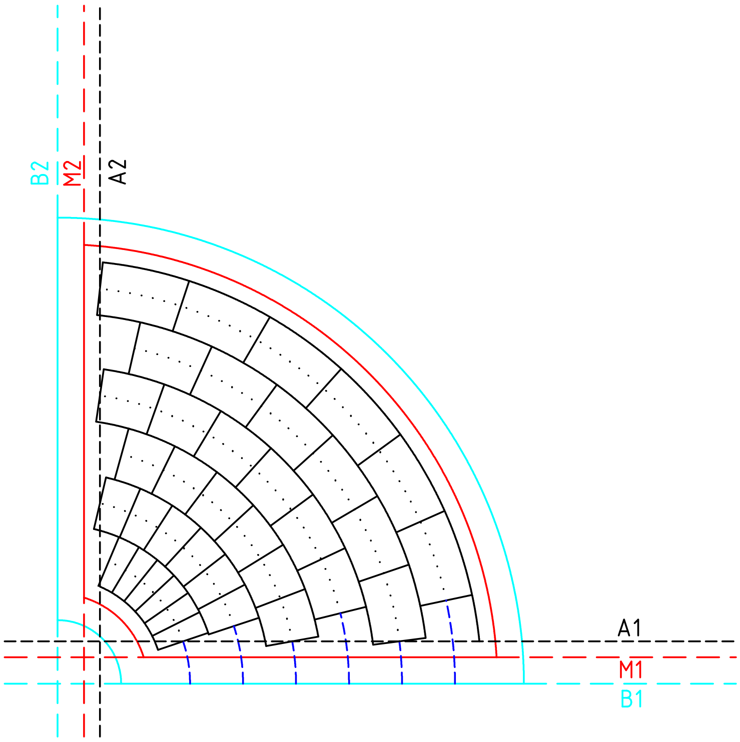

Figure 8: Example of a VersatileDiskRowLayout with offsets calculated as lines parallel to the edge of the bounding box (section 3.2.1). Note that all circles (pad rows) are concentric to the local origin, which is the intersection of the edges of the bounding box (B1 and B2). The circles do not intersect at a 90° angle with the edges of the module (M1 and M2) or the alignment lines A1 and A2. The intersection of M1 and M2 is irrelevant for the GEAR geometry, as is the intersection of A1 and A2. The row offsets (blue) are measured from the edge of the bounding box B1 to the alignment line A1 along the central circle for rows 0, 2 and 4, and have an additional half pad pitch for rows 1, 3 and 5.

```
<!-- module 0 -->
<module>
  <PadRowLayout2D type="VersatileDiskRowLayout" rMin="1440">

            <!-- row 0 -->
            <!-- pitch = 8.3870 deg /180 deg * pi * (1440 + 6/2 )  / 100 -->
            <row nPad="100" padPitch="2.1122744297835503" rowHeight="6"
                 padWidth="2.0122744297835503" padHeight="5.9"
                 offset="3.0" />
```

```
            <!-- row 1 -->
            <!-- pitch = 8.3870 deg /180 deg * pi * (1440 + 1 * 6 + 6/2 )  / 100 -->
            <!-- offset = 3.0 + pitch/2. -->
            <row nPad="100" padPitch="2.1210572756454362" rowHeight="6"
                 padWidth="2.0210572756454362" padHeight="5.9"
                 offset="4.0605286378227181"/>

            .....

  </ PadRowLayout2D >

  <!-- Note: Although the numbers come from the construction, -->
  <!-- the choice of the phi=0 line is still an example. -->
  <!-- angle = (-3 deg - 8.3870 deg /2) /180 * pi - -->
  < angle value ="-0.125401906756" / >

  <!-- Offset for the lower module row in mm -->
  < offset x_r ="-172.5" y_phi ="0." / >
</module>
```

3.2.2 Example 2

In the second example the pads are aligned to the central axis of the module, the staggered rows shifted by $\pm 1/4$ of a pitch. The pitch is constant at 1.2 mm, and the number of pads is adapted to fill the space up to 1mm distance to the module boundary. In this case the offsets will be alternating ± 0.3 mm, while the obvious choice of the $\varphi = 0$ line is the centre of the module. So `angle=`$3^\circ/180^\circ \cdot \pi$ for module 0. I leave the calculation of the numbers of pads per row to the reader.

3.3 VersatileDiskRowLayout with Shifted Origin

Until now all examples used the bounding box as a reference, which made the pad rows concentric with the module construction and the other modules in the row. This is not necessarily the case. As a last example we show the case that the centre of the pad rows is the intersection of the straight edges of the physical module (figure 9). The $\varphi = 0$ line now becomes the edge of the physical module instead of the bounding box.

The local origin of the GEAR coordinates has to be adapted. In this example it can be described by a shift along the central axis of the module by $2\ \mathrm{mm}/\sin(\varphi_{\mathrm{BBox}}/2) \approx 13.712$ mm. The rotation angle stays the same. For the rest of the example we stick to the values of section 3.2.1: The pads are aligned 1 mm from the edge of the module, being staggered and fill up the full angular range with a fixed number of 100 pads. The inner radius has been decreased by 13.712 mm so the middle pad row is at the same position as in the example with the bounding box as reference.

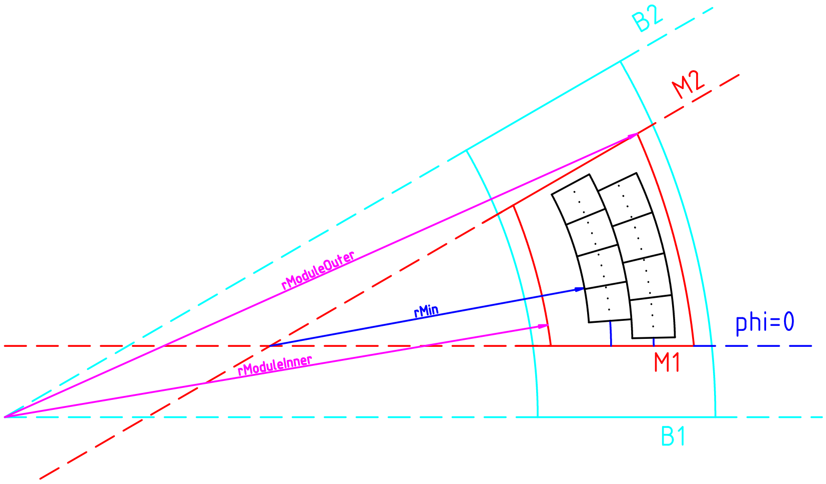

Figure 9: Example of a VersatileDiskRowLayout with the local origin of the pad plane at the intersection of the physical module edges M1 and M2. Note that the pad rows are not concentric with neither the module frame nor the other modules in the same row. This is due to the special shape of the physical module where M1 and M2 do not intersect at the centre of the inner and outer radius, and the “edges” of the frame do not have a right angle.

```
<!-- module 0 -->
<module>
  <PadRowLayout2D type="VersatileDiskRowLayout" rMin="1426.288">

            <!-- row 0 -->
            <!-- pitch = 8.3870 deg /180. deg * pi * (1440.-13.712 + 6./2. )  / 100. -->
            <row nPad="100" padPitch="2.092202699373854" rowHeight="6"
                 padWidth="1.992202699373854" padHeight="5.9"
                 offset="1.0" />

            <!-- row 1 -->
            <!-- pitch = 8.3870 deg /180 deg * pi *(1440-13.712 + 1 * 6 + 6/2 ) /100 -->
            <!-- offset = 1.0 + pitch/2. -->
            <row nPad="100" padPitch="2.1009855452357398" rowHeight="6"
                 padWidth="2.0009855452357398" padHeight="5.9"
                 offset="2.05049277261787"/>
```

```
            .....

  </ PadRowLayout2D >

  <!-- Note: Although the numbers come from the construction, -->
  <!-- the choice of the phi=0 line is still an example. -->
  <!-- angle = (-3 deg - 8.3870 deg /2) /180 * pi - -->
  < angle value ="-0.125401906756" / >

  <!-- Note: Although all the numbers come from the construction, -->
  <!-- the location of the local origin has been changed in this example. -->
  <!-- Bounding box offset for the lower module row in mm: -172.5 -->
  <!-- offset x_r = -172.5 mm + (2 mm / sin(8.3870 deg) * cos(-3. deg)) -->
  <!-- offset y_phi =  2 mm / sin(8.3870 deg) * sin(-3. deg) -->
  < offset x_r ="-186.19312152070296" y_phi ="-0.7176260903528622" / >
</module>
```

4 Summary

We gave some examples to show how the parameters of the abstract GEAR geometry can be calculated from the parameters of the LP1 TPC. The actual values of the pad planes have to be determined individually from the technical drawings.

It is important to realise that GEAR's TPCModule is not the description of a physical module, but the placement of a pad plane with its own local coordinates in a global coordinate system. As GEAR pad planes can only be contiguous at the moment, it can even be useful to place more than one TPCModule per physical module. The Desy GEM module with an upper and a lower half is an example, or several Timepix chips on one module frame, which allows alignment of the chips by tuning the offsets individually.

References

[1] Dan Petersen, Technical Drawings of the LP1 End Plate
http://www.lepp.cornell.edu/~dpp/linear_collider/images/LargePrototype/20080312-6080-102-Endplate-drawing.pdf

[2] GEAR: GEometry API for Reconstruction, http://ilcsoft.desy.de/portal/software_packages/gear/