**Tunnel Intersection Measurement**

Diagram

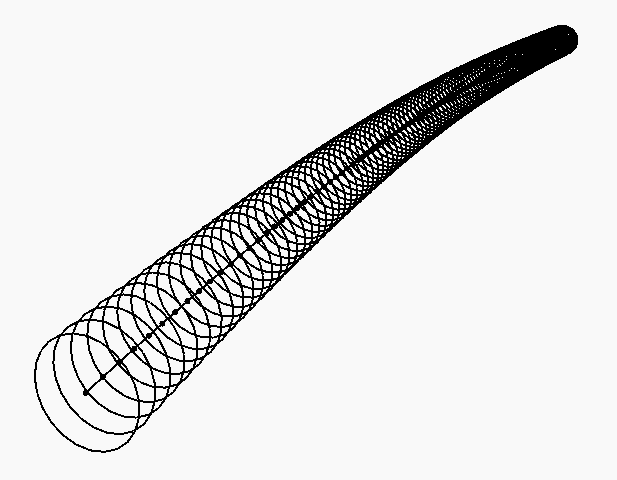
Description automatically generated

This workflow is devided into 11 parts, flows from left to right, parts 1 to 5 contains different operations, parts 6 to 11 contains simelier operations for each of the left and right sides of the tunnel

**Part 1**

Diagram

Description automatically generated

This part of the workflow creates surfaces along the path of the tunnel, at the designated intervals, this will allow us to determine which points we want to measure later in the workflow

Graphical user interface, text, application

Description automatically generated

This node measures the length of a curve, its used here to measure the length of the sweep path, to be able to set the maximum point distance we need for intersection

Graphical user interface, application

Description automatically generated

This node creates a point inside dynamo, and takes 2 inputs, the curve which will be the sweep path, and the segment length, which will be the series of numbers at the desired intervals

Graphical user interface, application

Description automatically generated

This node creates an imaginary vector inside Dynamo, which will be used alongside the points to create circles

Table

Description automatically generated

This node creates circles using 3 inputs, center points, radius and normal (which is the vector) from the previous node

Graphical user interface, application

Description automatically generated

This node creates a surface using a closed curve, it used the circles from the previous node, we will need surfaces to be able to intersect it with the curved and rectangular tunnels boundaries

**Part 2**

Graphical user interface

Description automatically generated

This part creates a 3D surface sweep using the path and rectangular profile from Civil 3D, this surface will be intersected at the predetermined intervals from part 1

Shape

Description automatically generated

Graphical user interface, text, application

Description automatically generated

This node reads the Revit geometry, that is a DWG import from Civil 3D, this import contains 2 items, the sweep path, and the rectangular profile

Graphical user interface, text, application

Description automatically generated

This node extracts the data from DWG geometry to Dynamo geometry, this is essential to extract elements from Revit (including the DWG imports)

Graphical user interface, application

Description automatically generated

This node selects items from a bigger list, it uses the index of the items needed, its used in the workflow to separate the sweep path and the rectangular profile

Graphical user interface, application

Description automatically generated

This node creates a surface using a profile and a path, in our case using the rectangular profile and the sweep path from the DWG file

Graphical user interface, text, application

Description automatically generated

This node explodes the geometry into separate parts, this will allow us to get the four sides of the sweep rectangle separately, and get the upper sweep only

Graphical user interface, application

Description automatically generated

This node makes nested lists inside the main list, the lengths will be 4, since the rectangle has 4 sides, this node is used because we have a lot of smaller swept rectangles

Graphical user interface, application

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

This node selects items from a bigger list, this time it’s used to select the 3rd item from each of the nested lists, this 3rd item in the nested list is the upper surface, this node will repeat the process for all the small sweeps and selects all the top surfaces

Graphical user interface, text, application

Description automatically generated

This node creates a continues Polysurface by joining the individual adjacent surfaces, its used here to get the top part of our rectangular tunnel as a single item

**Part 3**

Diagram

Description automatically generated

This part of the workflow creates a sweep from the DWG arcs, which resembles the edges of the tunnel, the surface resulting from the sweep will be intersected at the predetermined intervals from part 1

Shape, arrow

Description automatically generated

Table

Description automatically generated with low confidence

This node reads the Revit geometry, that is a DWG import from Civil 3D, this import contains many arcs, each one of them will be considered a separate item

Graphical user interface, text, application

Description automatically generated

This node extracts the data from DWG geometry to Dynamo geometry, same function as the once from part 2

Graphical user interface

Description automatically generated with low confidence

This node reads Revit geometry, but this time it reads a line that’s drawn in Revit manually from the starting point of the tunnel to the ending point, this curve serves 2 functions:

1 - it is essential because Dynamo understands starting and ending point by the actual starting click and ending click, so we are making sure that we are measuring starting from the correct side

2 - we need it to use it to intersect the DWG arcs to sort them in the correct sequence, because Dynamo organize items in a list by the modeling sequence, so this makes sure we are sorting them from one side to the other

Graphical user interface, text, application

Description automatically generated

This node extracts the data from DWG geometry to Dynamo geometry, this time we are taking a native Revit curve, it follows the same rules as DWG imports

Graphical user interface, application

Description automatically generated

This node extrudes a curve using 3 inputs, a curve, direction, and distance, we’re using it to extrude the manually drawn curve inside Dynamo upwards, this will create a surface that intersects with the arcs

A picture containing text, saw, tool

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

This node intersects geometries with each other, each one of the 2 inputs may take a single or multiple elements, in this part of the workflow, its used to intersect the extruded Revit curve with the DWG arcs

Graphical user interface, application

Description automatically generated

This node reads the UV coordinates of points on a surface, the surface used is the extruded Revit curve, and the points are the points of intersection from the previous node

Graphical user interface, application

Description automatically generated

This node extracts the V value only from the previous UV, we’ll use this value to be able to be able to sort the arcs indexes

Graphical user interface, text, application

Description automatically generated

This node sorts a list of numbers from lowest to highest, we’ll use it to sort the V value of the arc’s intersections with the extruded Revit curve, the outcome will be an unorganized list of numbers, which states the order of their modeling inside Civil 3D

Table

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

This node gets an item at an index, in this case, the list is the unorganized list of arcs, and the index is the previous node outcome, the output will be a reorganized list of arcs starting from west to east, having the arcs in the correct order is essential for the next node to work

Graphical user interface, application

Description automatically generated with medium confidence

This node creates a 3D surface using a list of cross sections, in our case, we’re using the reorganized arcs from the previous node to create the loft

**Part 4**

Diagram

Description automatically generated

This part intersects the polysurface created in part 2 with the patched circles created in part 1

Shape

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated with medium confidence

This node intersects geometries with each other, this time intersecting the top part of the rectangular sweep which we recreated as a polysurface in part 2 of the workflow with the surfaces that’s create from the last node in part 1 (surface by patch)

Graphical user interface, application

Description automatically generated

This node joins lines that are overlapping together, it ensures that we don’t have any lines overlapping which would cause issues in the script later on

**Part 5**

Diagram

Description automatically generated

In this part, we intersect the patched circles from part 1 with the tunnel from part 3 of the workflow, and create new arcs at the designated intervals

A close-up of a pen

Description automatically generated with low confidence

Graphical user interface

Description automatically generated with medium confidence

This node intersects geometries with each other, now it intersects the patched circles from part 1 with the tunnel surface from part 3, and create the arcs perfectly aligned with the lines (top of rectangles) created in part 4

Graphical user interface, text, application

Description automatically generated

This node takes the start point of a curve, in our case, each arc intersection is created of many small segments, so the node is used to neatly rebuild the intersected arcs by taking the start point of each one of them and will be used in the next node

Graphical user interface, application

Description automatically generated with medium confidence

This node creates a nurbs curve using points, we’ll use the points from the previous node to create the nurbs curve, using this node will create accurate and neat arcs instead of messy multi segmented arcs

**Parts 6 & 7**

Graphical user interface, diagram

Description automatically generated

Graphical user interface, diagram

Description automatically generated

In these parts, we deal with left and right side of the tunnel separately, using almost identical nodes, these parts measure the distance between the corner of the rectangle and the edge of the arc at the predetermined distance from part 1, then decides if it’s clashing or not

A picture containing saw, wire, line

Description automatically generated

Graphical user interface, application

Description automatically generated

This takes the end point of the top segments of the rectangle, which is in this case the left side, this node is found in part 6 only, the right-side end point is taken in part 7 using the “curve start point” node

Graphical user interface, text, application

Description automatically generated

This node takes the closest point to a geometry on another geometry, in this case it’s used to find the closest point on the rebuilt arcs to the topmost left side point of each rectangle

Graphical user interface, application

Description automatically generated

This node measures the distance between two geometries, in our case, its used to measure the distances between the topmost left points on the rectangles with the closest point to it on the rebuilt arcs

Graphical user interface, application

Description automatically generated

This node rounds a number to the desired digits, in our case it’s used to round the measurements from the previous node to 1 decimal place, measuring in millimeters

Graphical user interface, application

Description automatically generated

This node creates an imaginary vector inside Dynamo, this vector will be starting from the topmost left side of the rectangular tunnel and heading to the closest point on the rebuilt arcs

Table

Description automatically generated

This node creates a line using 3 inputs, a start point, direction, and length, in the part of the workflow, it uses the topmost left point on the rectangle and the vector by two points node heading from the rectangle towards the arcs, if the rectangle is enclosed inside the arc, the line should be heading outwards, if there’s and issue and the rectangle is getting out of the arc, then the line is heading inwards, this makes it as a visual tool inside the workflow to check if there’s an issue, this line will also be used to check the intersection with a translated surface from the polysurface from part 2

Graphical user interface

Description automatically generated with medium confidence

This node moves geometry using a direction and distance, in our case, it’s used to move the polysurface from part 2 of the workflow downwards 0.7 meters, this moved polysurface will be used to check the intersection in the next node

Graphical user interface, application

Description automatically generated

This node checks if a geometry is intersecting with another geometry or not, in out case, it’s used to check the lines created from the “line by start point, direction and length” node are intersecting with the translated surface from the previous node, if it’s intersecting, then the rectangle is getting out of the arc, if it’s not, then the rectangle is inside the arc

Table

Description automatically generated

This node filters results based on a Boolean (a true or false value), it takes 3 inputs, a test result of a Boolean item or list, an item or list when the test result is true, and an item or list when the test result is false, in our case, the test input is taken from the previous node, the true list is the positive value of the distances outputted from the “Math round” node, and the false list is the negative value of the distances outputted from the “Math round” node, if the value is negative then the rectangle is inside the arc, if the value is positive then the rectangle is clashing with the arc

**Parts 8,9,10 & 11**

Graphical user interface, application, Teams

Description automatically generated

Diagram

Description automatically generated

Graphical user interface, application

Description automatically generated

Diagram

Description automatically generated

These are the last parts of the workflow, they take the measurements from parts 6 & 7, then display them, the maximum and minimum values within them, also export the results to an external CSV file

Graphical user interface, application

Description automatically generated

This node reads the value of the minimum item in a list

Graphical user interface, text, application

Description automatically generated

This node reads the value of the maximum item in a list

Graphical user interface, text, application, chat or text message

Description automatically generated

This node reads a path of a file on the computer, in our case, it’s used to read the location of the CSV file we’ll write that data to

Graphical user interface

Description automatically generated with medium confidence

This node writes the data results from the workflow to an external CSV file