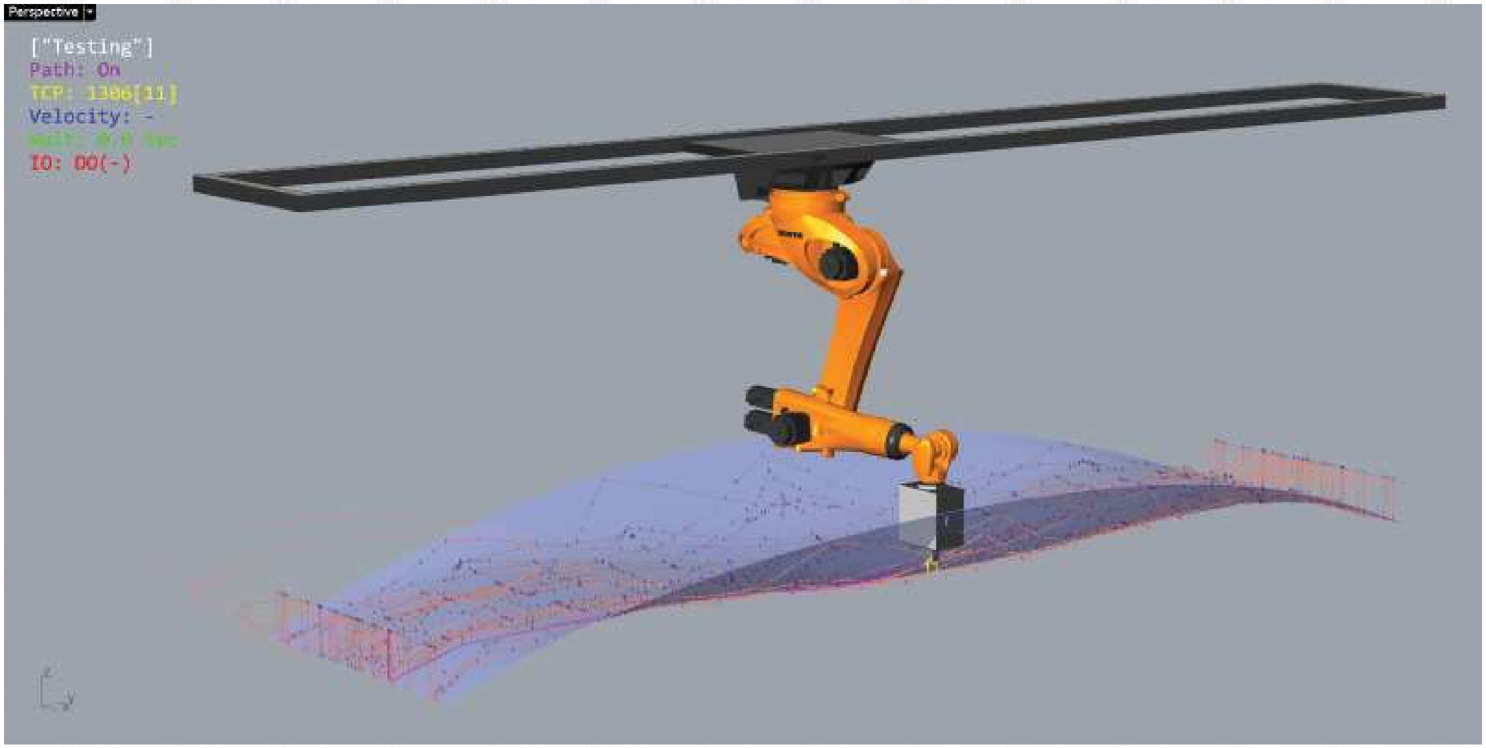
Visualization of Robotic Fiber Trajectory



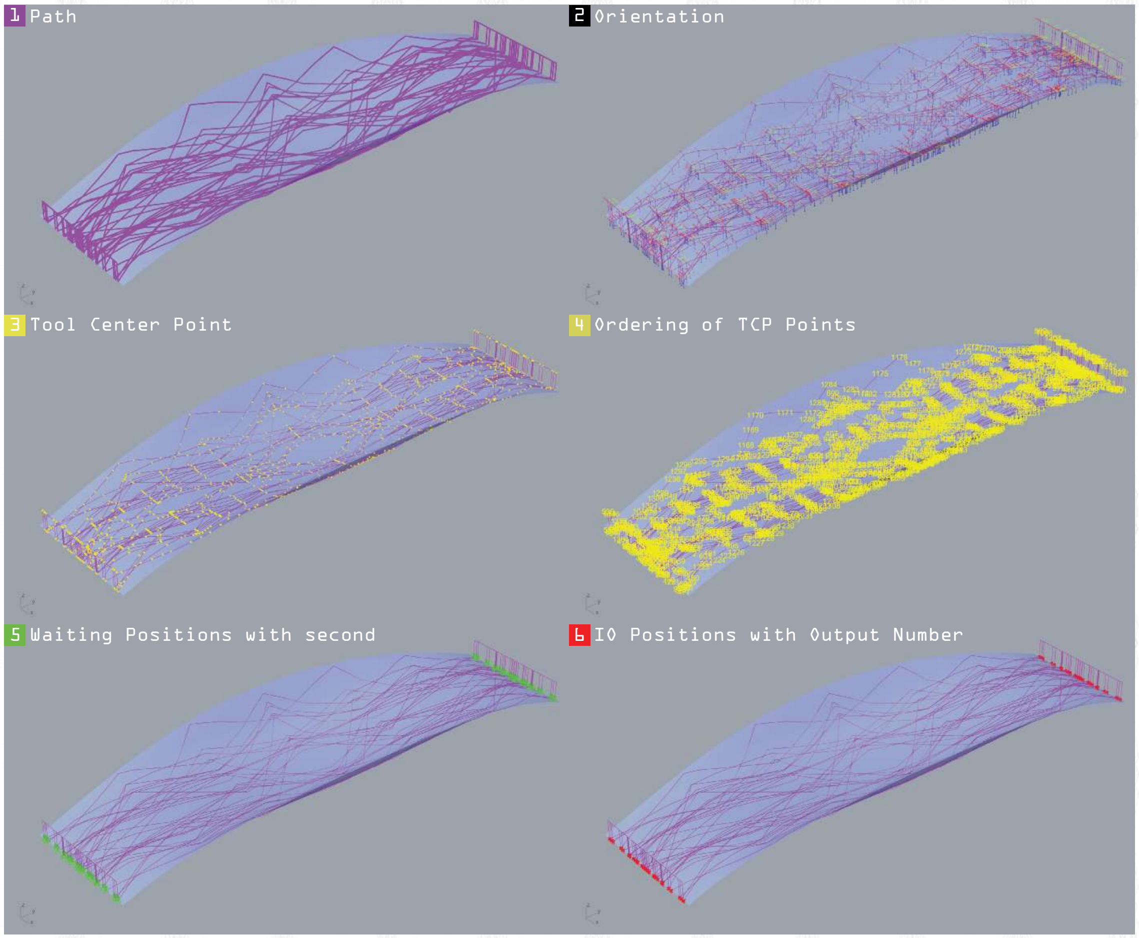
In the Fabrication process, the KUKA robot's movement path is visualized through analyzed the KRL program. The information of Task Name, Path, Count of TCP, Velocity, Waiting Position, and IO Output Number can be indicated in the HUD system, shown in the viewport's top-left. Users can intuitively understand their current status.

KRL Code Anaylysis

```
&ACCESS RVP
&REL 1
&PARAM TEMPLATE = C:\KRC\Roboter\Template\vorgabe
&PARAM EDITMASK = *
DEF tmp ( )


;FOLD INI
;FOLD BASISTECH INI
GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
INTERRUPT ON 3
BAS (#INITMOV,0 )
;ENDFOLD (BASISTECH INI)
;ENDFOLD (INI)

;FOLD STARTPOSITION - BASE IS 0, TOOL IS 6, SPEED IS 15%, POSITION IS A1 -90,A2 -140,A3 110,A4 5,A5 -60,A6 -5,E1 0,E2 0,E3 0,E4 0
$BWDSTART = FALSE
PDAT_ACT = {VEL 15,ACC 100,APO_DIST 50}
FDAT_ACT = {TOOL_NO 6,BASE_NO 0,IPO_FRAME #BASE}
BAS (#PTP_PARAMS,15)
PTP  {A1 -90,A2 -140,A3 110,A4 5,A5 -60,A6 -5,E1 0,E2 0,E3 0,E4 0}
;ENDFOLD

;FOLD LIN SPEED IS 0.25 m/sec, INTERPOLATION SETTINGS IN FOLD
$VEL.CP=0.25
$ADVANCE=3
;ENDFOLD

;FOLD COMMANDS IN FOLD. SELECT EDIT/FOLDS/OPEN ALL FOLDS TO DISPLAY
PTP {E6POS: X 379.403, Y -4029.587, Z -2547.221, A 0, B 90, C 0, E1 0, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 379.403, Y -4029.587, Z -2847.221, A 0, B 90, C 0, E1 307.692, E2 0, E3 0, E4 0, S 'B 010'}
WAIT SEC 2.5
$OUT[1]=TRUE
PTP {E6POS: X 496.95, Y -3587.989, Z -2664.518, A 0, B 90, C 0, E1 615.385, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 609.907, Y -3163.636, Z -2620.487, A 0, B 90, C 0, E1 923.077, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 734.492, Y -2695.602, Z -2578.111, A 0, B 90, C 0, E1 1230.769, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 672.082, Y -2264.559, Z -2570.343, A 0, B 90, C 0, E1 1538.462, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 603.92, Y -1793.781, Z -2618.261, A 0, B 90, C 0, E1 1846.154, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 541.915, Y -1365.534, Z -2658.015, A 0, B 90, C 0, E1 2153.846, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 655.23, Y -1090.227, Z -2657.23, A 0, B 90, C 0, E1 2461.538, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 768.573, Y -814.852, Z -2637.684, A 0, B 90, C 0, E1 2769.231, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 879.17, Y -546.146, Z -2567.226, A 0, B 90, C 0, E1 3076.923, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 955.373, Y -361.006, Z -2518.68, A 0, B 90, C 0, E1 3384.615, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 1020.247, Y -203.389, Z -2453.198, A 0, B 90, C 0, E1 3692.308, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 1091.783, Y -29.587, Z -2369.938, A 0, B 90, C 0, E1 4000, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 1020.247, Y 144.215, Z -2453.198, A 0, B 90, C 0, E1 4307.692, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 955.373, Y 301.832, Z -2518.68, A 0, B 90, C 0, E1 4615.385, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 879.17, Y 486.972, Z -2567.226, A 0, B 90, C 0, E1 4923.077, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 768.573, Y 755.678, Z -2637.684, A 0, B 90, C 0, E1 5230.769, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 655.23, Y 1031.053, Z -2657.23, A 0, B 90, C 0, E1 5538.462, E2 0, E3 0, E4 0, S 'B 010'}
PTP {E6POS: X 541.915, Y 1306.361, Z -2658.015, A 0, B 90, C 0, E1 5846.154, E2 0, E3 0, E4 0, S 'B 010'}
```

The process is started from analyzed the KRL, which is derived from the SRC file used to drive the KUKA Robot. The KRL Commands of ovement such as PTP and Linear are extracted and reconstructed into Tool Center Points Path, used for redraw in the viewport display in customized color and thickness.

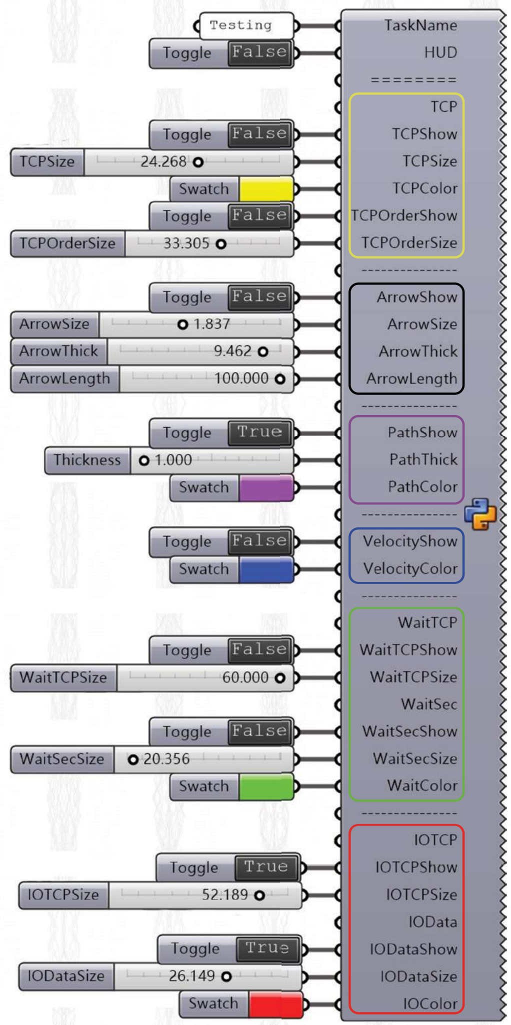
Visualization Item



1 Path
2 Orientation
3 Tool Center Point
4 Ordering of TCP Points
5 Waiting Positions with second
6 IO Positions with Output Number

The geometries' customized visualization functions are the Path, TCP Points, Ordering of TCP Points, TCP Orientations, Waiting Positions with second, IO Positions with Output Number. Each of them has the parameters to adjust the color, size, thickness, and a switch of whether to display.

Python Component in Grasshopper



Tool Center Point
Ordering of TCP Points

Orientation

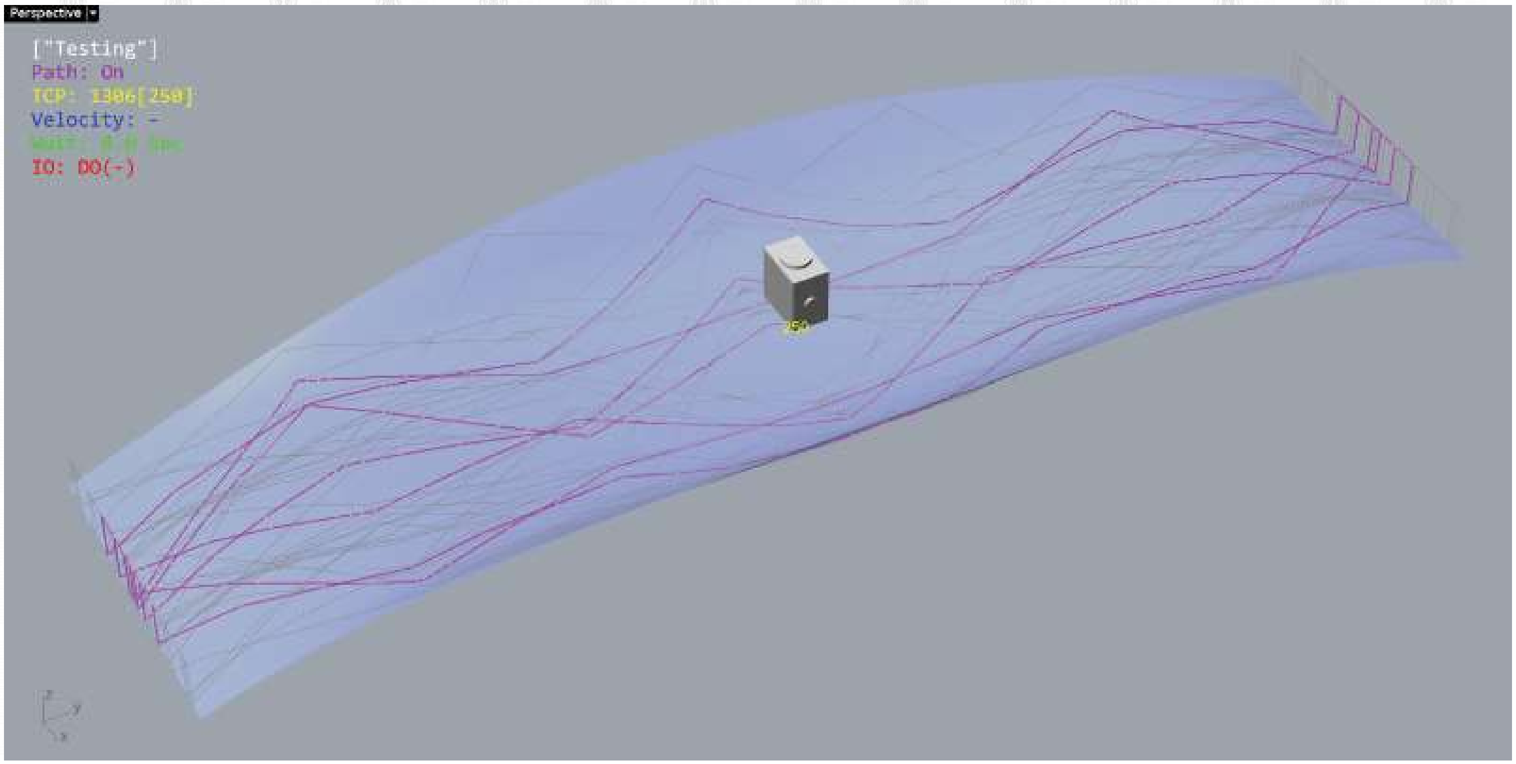
Path

Velocity

Waiting Positions with second

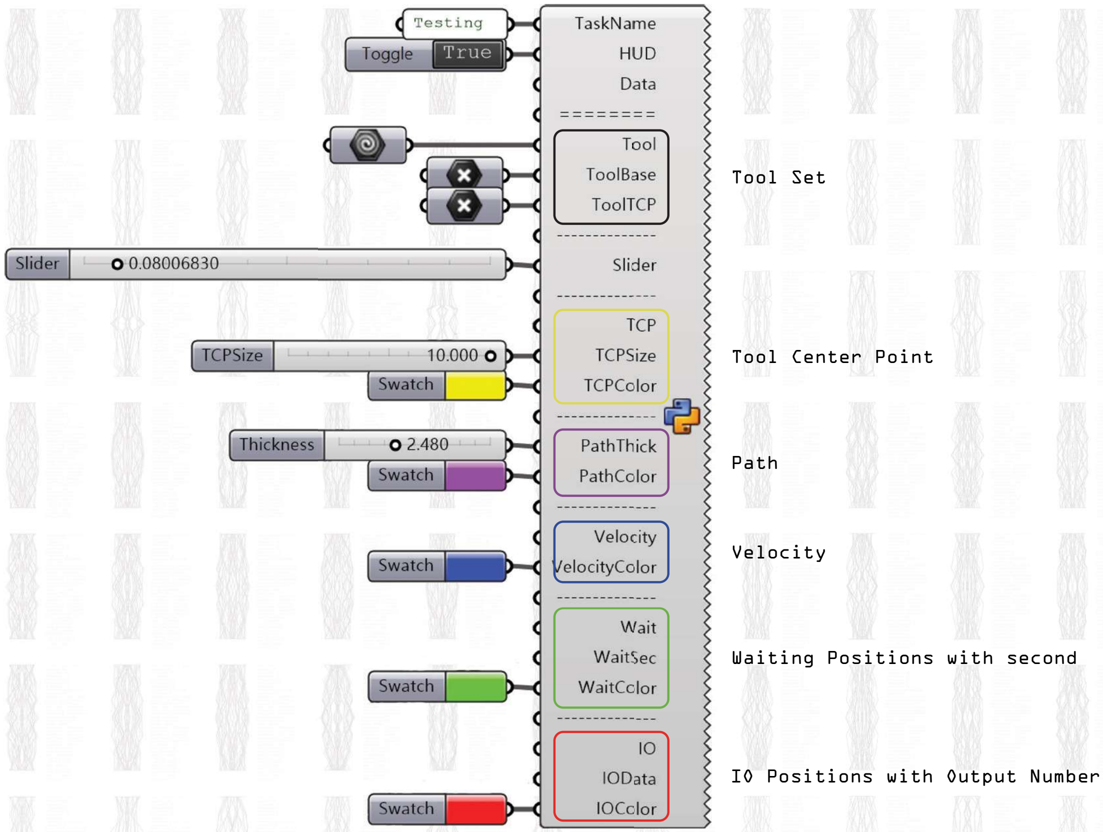
IO Positions with Output Number

Isolate the Current States

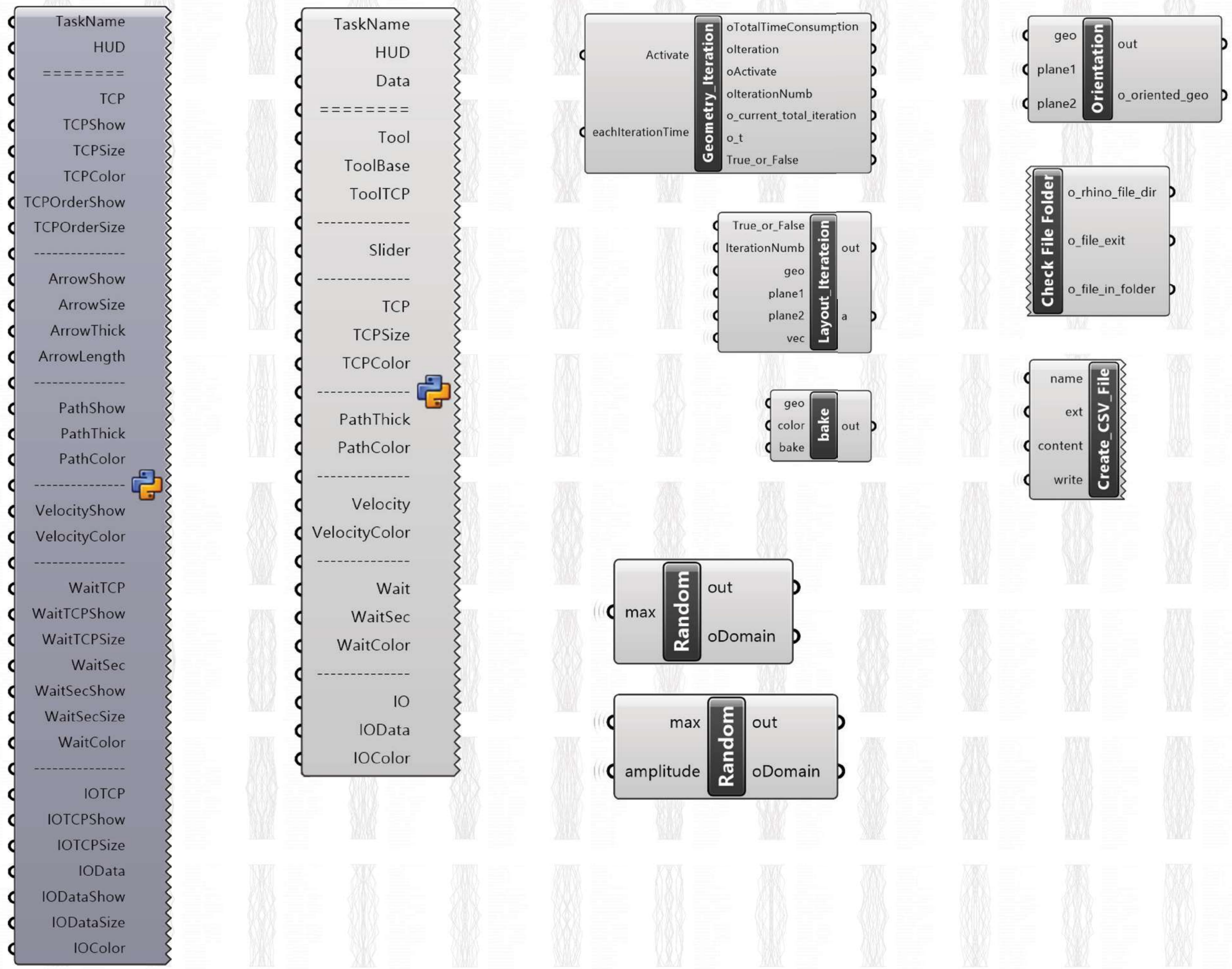


This component is to isolate the current states. The HUD displays the count of the TCP number, current line. The input slider can change the progress rate from 0.00 to 1.00. And the tool model, which is used as the end effector, can also be simulated in the different orientations of TCP Targets.

Python Component in Grasshopper



COMPONENT DEVELOPMENT



Summary

I have covered my vital interests and motivations in this project from Design, Structural Simulation, Machine Learning Optimization, and Fabrication process.

I have defined the goal to deal with each part of the overall computational design process to improve automation between each cycle. Using the coding ability, I tried to improve each workflow to make the process more user-friendly and develop a cutting-edge tool to optimize the design itself.

Machine Learning's application to the structural design and fabrication phase has revealed the significant potential in reducing prototyping times in the fabrication phase and the Design-Simulation process. I believe there are still many fields, especially in architecture and structural engineering, in which can explore further the possibility of applying the Machine Learning Application and deploy them.

Grasshopper, famous for its visual programming, promotes programming efficiency and is used in many fields. Meanwhile, intuitively use, and visual process inspection is also an essential part of fabrication in robotic programming. And there are many kinds of plug-ins for robotic programming in Grasshopper. My motivation is to integrate their advantages, find the lack of functions, and develop a more intuitive interface and better processes.