



Muottirakenteen parametrinen mallinnus ja tulostus FGF-prosessilla

Maxim Narbrough – Projektipäällikkö, 3DTY-hanke

FMT-päivät 2025, Nivala



Euroopan unionin
osarahoittama



LSAM tulostus CEAD AM-Flexbot Robottisolulla

Robottisolu

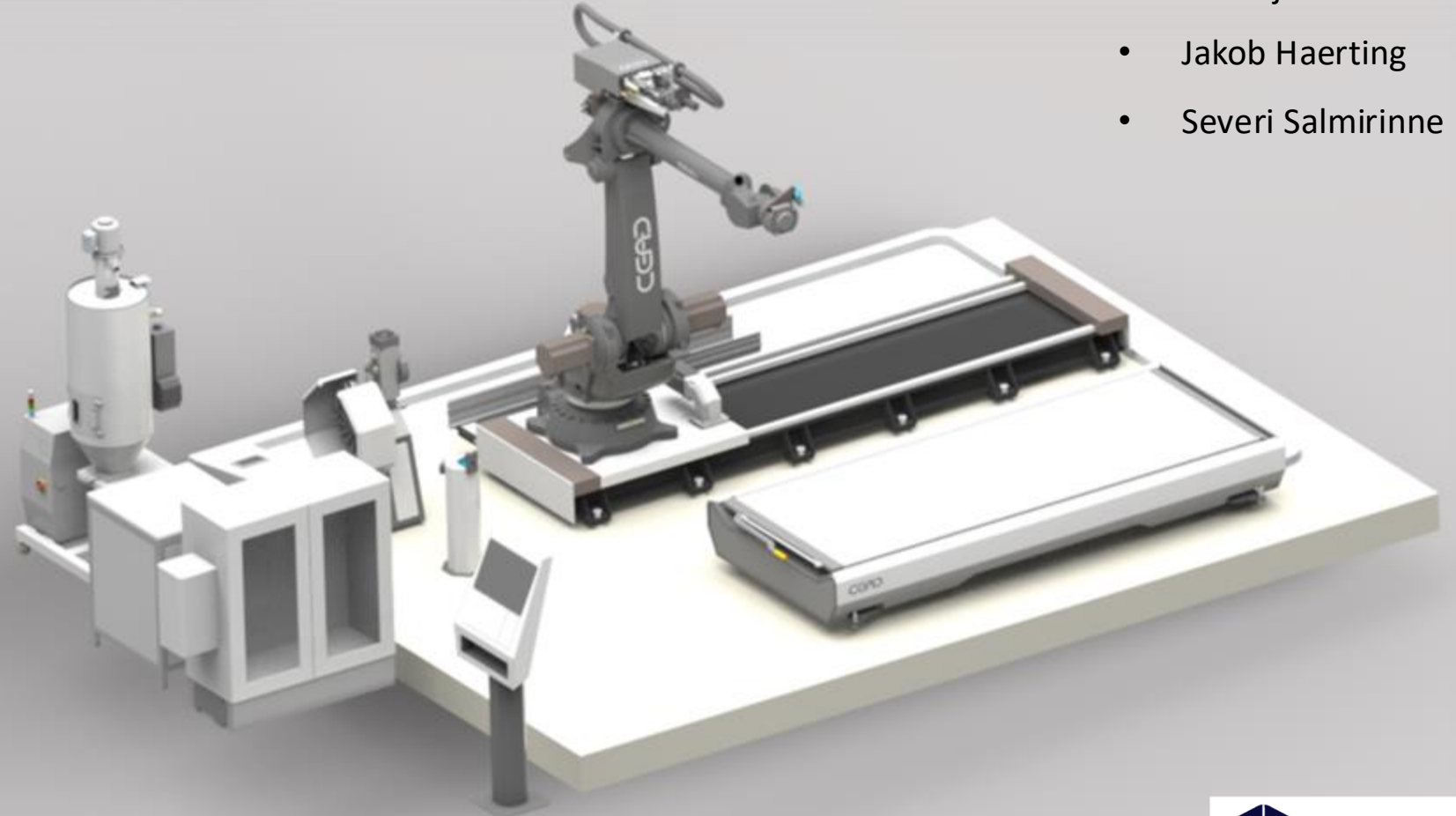
- COMAU NJ-165 3.0 Robotti
- CEAD E25 tulostuspää
- HITECO 9.5kW jyrsinpää
- VANSICHEN 2.5m kisko
- Lämmitettävä peti 3 x 1.2m
- Siemens Sinumerik 840D

Ohjelmistot

- Siemens NX
- Rhino & Grasshopper

Asiantuntijat

- Jakob Haerting
- Severi Salmirinne



Muotin valmistelun haasteet

- Ajoneuvon komponentin haastava geometria johon Siemens NX ei tarjonnut sopivaa tulostusstrategiaa.
- Adaxis Adaone tarjosi parempia tulostusmenetelmiä, mutta edelleen haasteellinen.
- Testattiin tulostusstrategioiden luonti algoritmiavusteisella ohjelmoinnilla (Grasshopper/Rhino3D).
- Aibuild ohjelmisto myös testikäytössä.
- Haasteet: monimutkainen geometria, ylitykset, tiukat mutkat, jyrkät muodon muutokset, muotojen tarkkuus, kompensointi

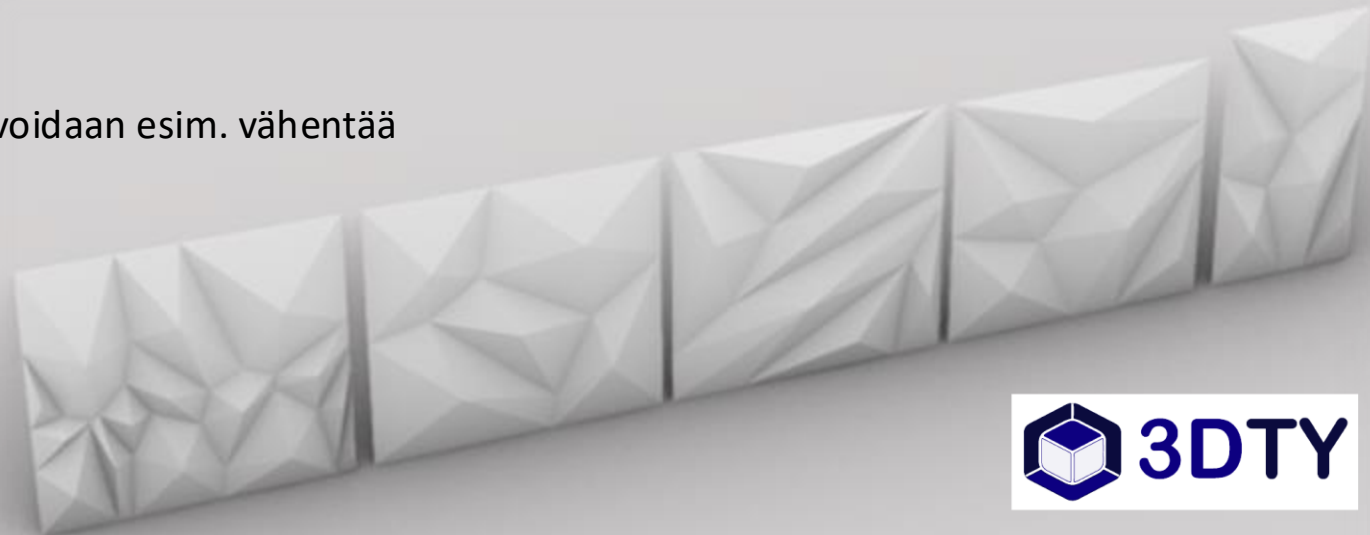


Parametrisuus apuna muottien suunnittelussa

Parametrinen mallinnus : suunnittelumenetelmä, jossa geometriaa ohjataan muuttujien ja sääntöjen avulla

Algoritmiavusteinen mallinnus : suunnittelumenetelmä, jossa geometria luodaan sääntöihin ja laskentaan perustuvien algoritmien avulla.

- Mahdollistaa **geometrian ominaisuuksien, kuten koon, muodon ja paksuuden, helpon säätämisen** ilman tarvetta piirtää mallia uudelleen.
- Tukee **nopeaa muunnelmien kokeilua**, mikä nopeuttaa tuotekehitystä ja testausta.
- **Tehostaa vaihtoehtojen arvioinnin ja simuloinnin** tulostettavuuden, rakenteellisen kestävyuden tai muotoilun kannalta.
- Sallii **dataohjattua suunnittelua**, minkä avulla voidaan esim. vähentää hukkamateriaalia ja tulostusajat.



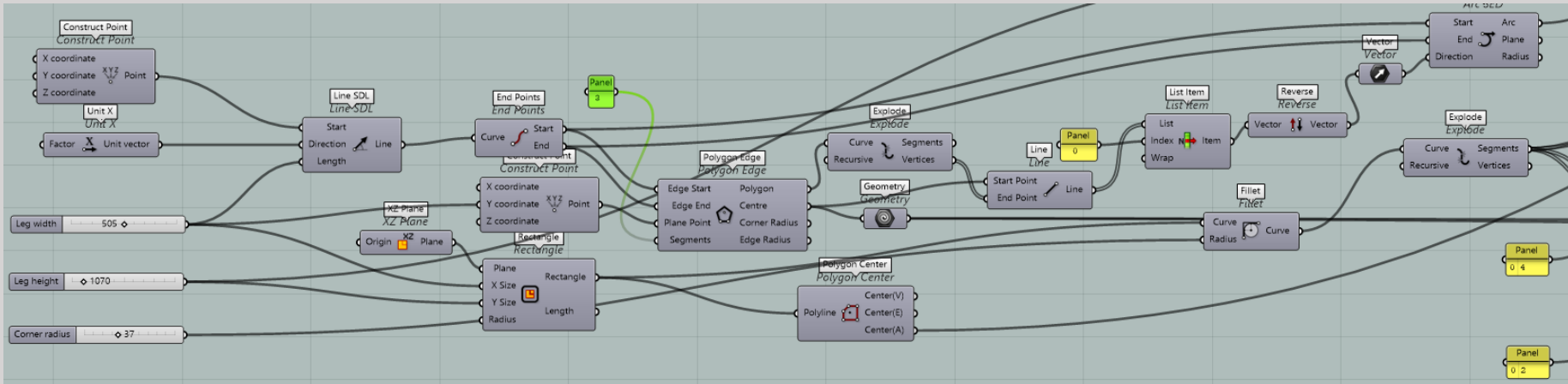
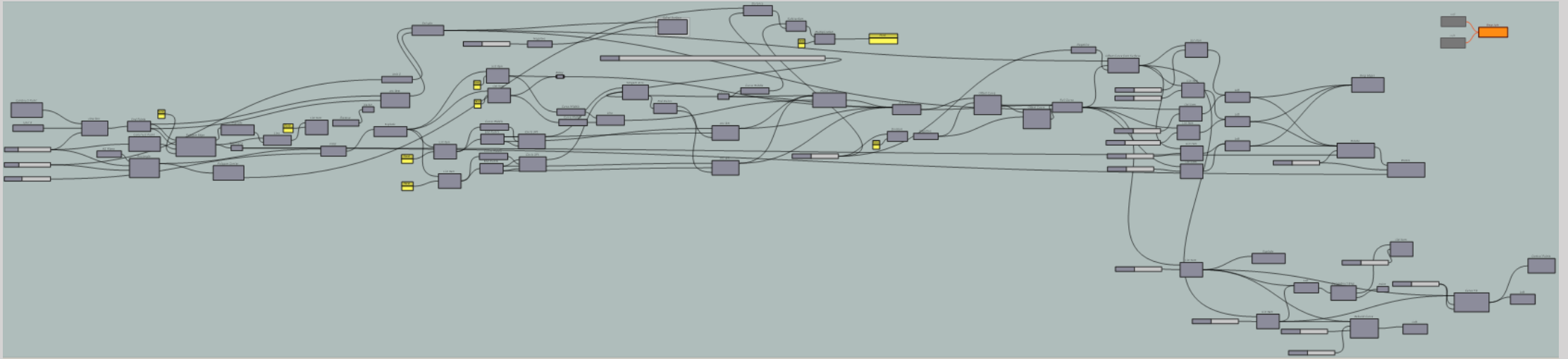
Huonekalukonseptin suunnittelu ja työvaiheet algoritmeiksi

Työn kulku

- Konseptin 3D-pintamallinnus
- 3D-mallin kääntäminen parametriseksi malliksi
- Tukimuottien generointi
- Tulostuksen viipalointi



Työkalujen visuaalinen ohjelmointi



Selkeytetty ohjauspaneeli tulostusratojen luontiin

Control panel

Print geometry

Slicing method
Slicing method Isocurves

Slicing degree
-45

Slicing surfaces

Print parameters

interpolation count when slicing with surfaces 173

Layer height 5.00

Layer width 6.0

Overall flow multiplier 1.5

Layer time 300

C deg 0

Print path tolerance 0.05

Tilt degree limit 15

Decimal amount 4

Feedrate multiplier

Domain start 0.8

Domain end 1.1

Seam placement

True for point, false for curve

True for point, false for curve **False**

Guide point location 0.200

Drop a guide curve here

Swap contour order
Swap contour order **True**

45 degree EM value filter
45 degree EM value filter **False**

Enable tilt limit
Enable tilt limit **False**

use most common layer height
use most common layer height **True**

Slicing by degrees?
Slicing by degrees? **False**

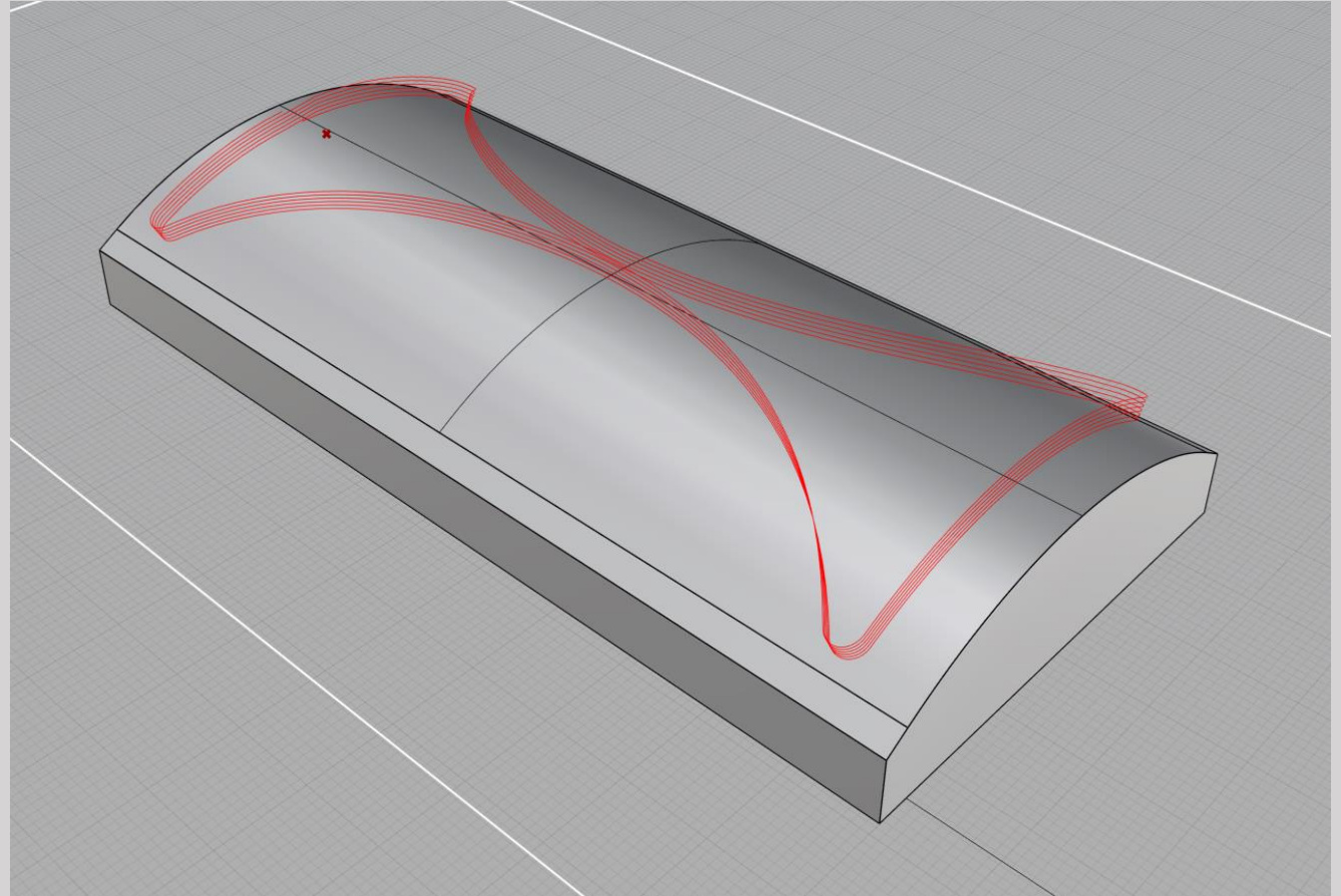
Use toolpath compensation
Use toolpath compensation **False**

Flip toolpath compensation
Flip toolpath compensation **True**

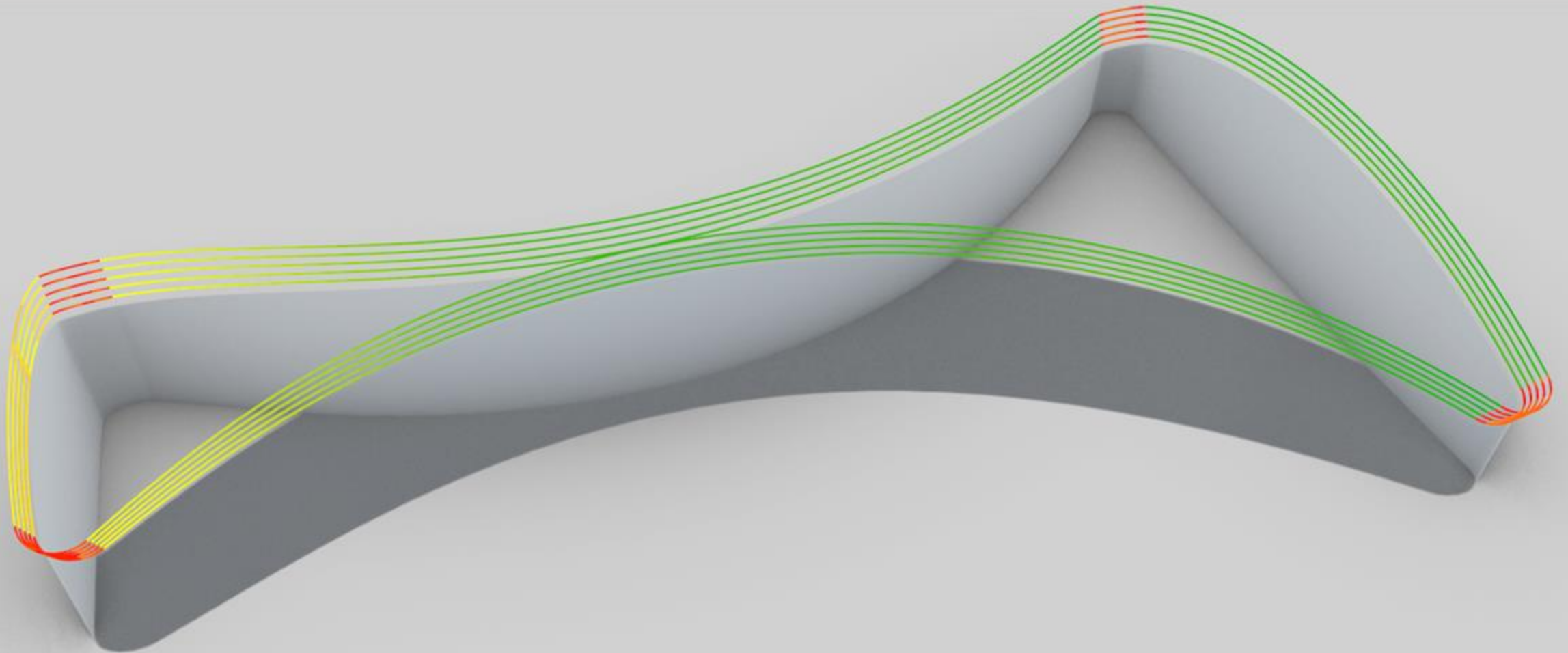
Z-hop
Z-hop **True**

Use tilt compensation
Use tilt compensation **False**

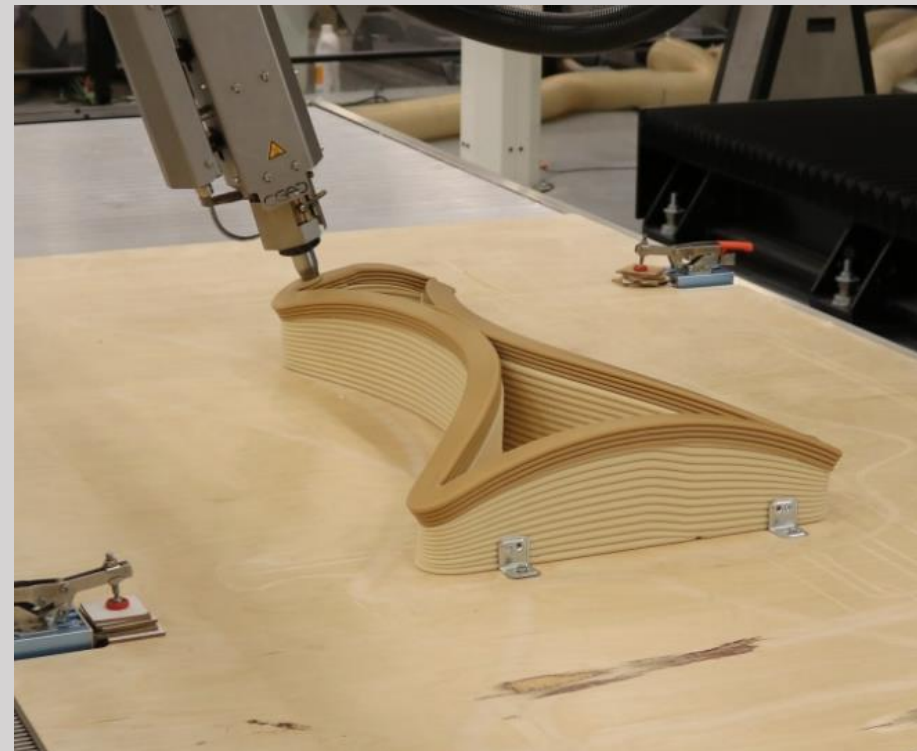
Send data
Send data **True**



Muotin tulostusstrategian optimointi



Optimoidut muotit ja kappaleet



- Muotin tulostusaika noin 2 tuntia - alkuperäinen yli 5 tuntia
- Muotin koneistusaika 15-20 minuuttia - alkuperäinen yli 2.5 tuntia
- Kappaleen tulostus noin 20 minuuttia



Kalusteet Formnext messuilla 2024



**Euroopan unionin
osarahoittama**



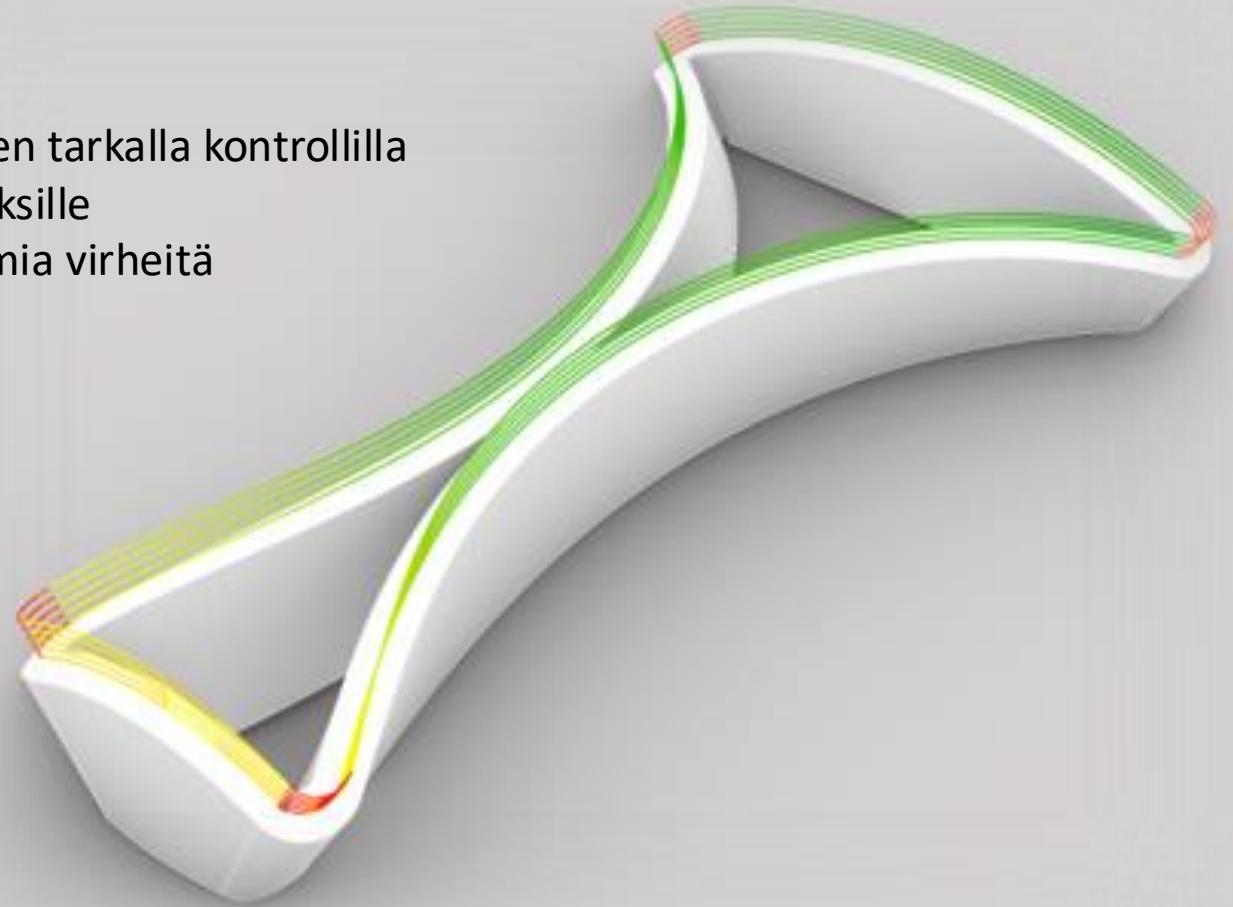
Algoritmiavusteisten työkalujen hyödyt ja haasteet

Hyödyt

- Manuaaliset työvaiheet voidaan automatisoida
- Tiettyjen geometrioiden haasteiden ratkaiseminen tarkalla kontrollilla
- Olevia työkaluja voidaan muokata uusille sovelluksille
- Voi vähentää manuaalisen mallintamisen tuottamia virheitä
- Ulkoisen datan ja tekoälyn hyödyntäminen
- Tukirakenteiden luomisen automatisointi

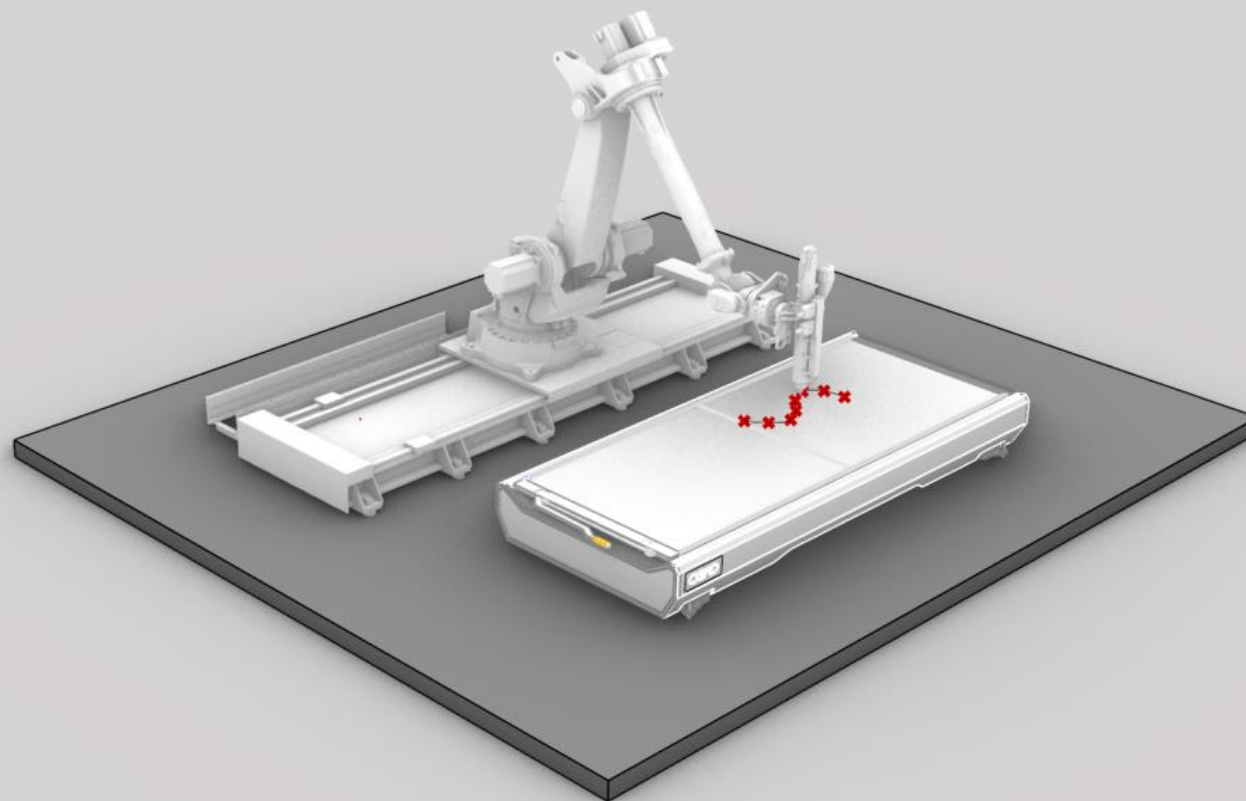
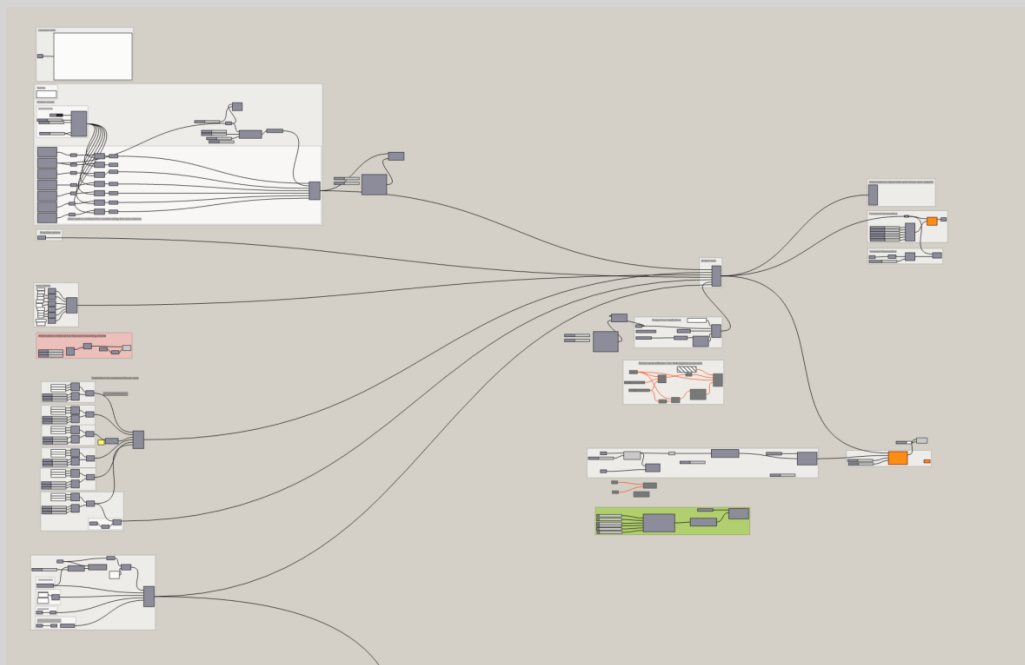
Haasteet

- Ohjelmointi voi olla hidasta ja monimutkainen
- Mallin tarkka kontrollointi haastavampaa
- Vaatii syvällistä perehtymistä tulostuslogiikkaan
- Ei sovi kaikille



Muiden robottien ohjelmointi Grasshopperilla

- Mahdollisuus luoda ja simuloida eri robottisoluja
- ABB, KUKA ja yleisimmät valmistajat tuettu lisäosilla (KUKA PRC, Robots, Robot components)
- Kustannustehokas ratkaisu



Tulevaisuuden tuulia - koodia ja koneoppiminen

- Analysointiin ja simulointiin hyödyllisen koodin integrointi
- Koneoppimisellä voidaan ajaa automatisoituja sekä reaaliaikaisia optimointimenetelmiä

ChatGPT

Voitko luoda minulle python tiedoston jonka voin liittää grasshopperin python komponenttiin. Haluaisin parametrisesti ohjattavan kukka vaasin jonka voi 3D-tulostaa pelletti tulostimella

Muisti päivitetty

Tässä on Python-koodi, joka luo parametrisesti ohjattavan kukkavaasin Grasshopperissa. Se hyödyntää Rhino.Geometry-kirjastoa ja generoi vaasin parametrien perusteella.

Parametric Vase

Tämä koodi generoi parametrisesti ohjattavan vaasin Grasshopperissa. Voit säätää korkeutta, ylä- ja alaosan sädettä, aaltojen määrää ja amplitudia. Liitä tämä koodi Grasshopperin Python-komponenttiin ja anna parametrit liukusäätimillä! Haluatko lisätä esimerkiksi seinämän paksuuden metriksi?

Kysy mitä tahansa

ChatGPT voi tehdä virheitä. Suosittelemme tarkastamaan tärkeät tiedot.

```

1 import Rhino.Geometry as rg
2 import math
3
4 # Parametrit (liitetään Grasshopperin liukusäätimiin)
5 height = H # Vaasin korkeus
6 radius_bottom = Rb # Alaosan säde
7 radius_mid = Rm # Keskikohdan säde
8 radius_top = Rt # Yläosan säde
9 waves = W # Aaltojen määrä
10 amplitude = A # Aaltojen korkeus
11 texture_intensity = T # Tekstuurin voimakkuus
12 resolution = 60 # Profiilien määrä
13 sections = []
14
15 # Luo profiilikäyrät vaasia varten ilman siniä ja kosinia
16 for i in range(resolution + 1):
17     t = i / float(resolution)
18     z = t * height
19
20 # Kolmivaiheinen interpolointi säteelle
21 if t < 0.5:
22     smooth_t = t * 2
23     r = (1 - smooth_t) * radius_bottom + smooth_t * radius_mid
24 else:
25     smooth_t = (t - 0.5) * 2
26     r = (1 - smooth_t) * radius_mid + smooth_t * radius_top
27
28 # Aaltojen hienosäätö ilman trigonometriaa
29 wave_factor = 1 + amplitude * (t % 0.2) # Pieni jaksollinen häiriö
30 r *= wave_factor
31
32 curve_points = []
33 for j in range(0, 360, 5): # Tiheämpi jakoväli
34     angle = math.radians(j)
35     texture = texture_intensity * ((j % 20) / 20.0) # Lineaarinen muutos tekstuuriin
36     x = (r + texture) * math.cos(angle)
37     y = (r + texture) * math.sin(angle)
38     curve_points.append(rg.Point3d(x, y, z))
39 curve_points.append(curve_points[0]) # Sulje käyrä
40 sections.append(rg.NurbsCurve.Create(True, 3, curve_points))
41
42 # Luo vaasi loftilla
43 loft = rg.Brep.CreateFromLoft(sections, rg.Point3d.Unset, rg.Point3d.Unset, rg.LoftType

```

Script Editor

```

1 import Rhino.Geometry as rg
2 import math
3
4 # Parametrit (liitetään Grasshopperin liukusäätimiin)
5 height = H # Vaasin korkeus
6 radius_bottom = Rb # Alaosan säde
7 radius_mid = Rm # Keskikohdan säde
8 radius_top = Rt # Yläosan säde
9 waves = W # Aaltojen määrä
10 amplitude = A # Aaltojen korkeus
11 texture_intensity = T # Tekstuurin voimakkuus
12 resolution = 60 # Profiilien määrä
13 sections = []
14
15 # Luo profiilikäyrät vaasia varten parannetulla muotoilulla
16 for i in range(resolution + 1):
17     t = i / float(resolution)
18     z = t * height
19
20 # Kolmivaiheinen interpolointi säteelle
21 if t < 0.5:
22     smooth_t = t * 2

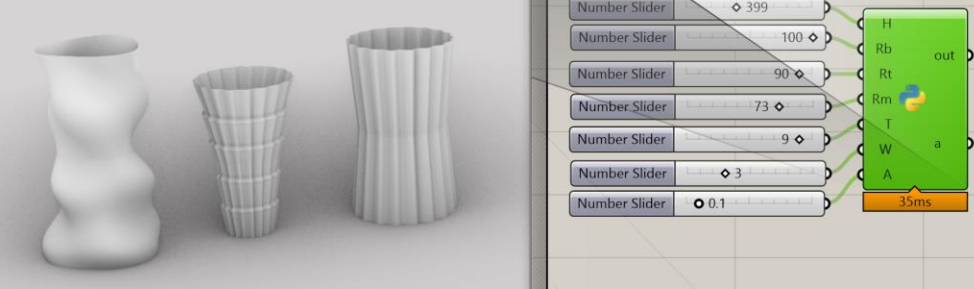
```

Python 3.9.10

Grasshopper - unnamed

Geometry Primitive Input

206%





Kiitos!

Maxim Narbrough
maxim.narbrough@redu.fi
www.redu.fi/3dty



Euroopan unionin
osarahoittama

