

**T.C.
BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MİMARLIK ANABİLİM DALI
YÜKSEK LİSANS TEZİ**

**PARAMETRİK TASARIM ARAÇLARI
KULLANILARAK OLUŞTURULAN VE ROBOTİK
ÜÇ BOYUTLU YAZICILAR İLE
ÜRETİLEBİLECEK BARINMA BİRİMİ TASARIMI**

Ahmet KORKMAZ

Danışman: Prof. Dr. Sevim ATEŞ CAN

BURDUR, 2024

ETİK KURALLARA UYGUNLUK BEYANI

Mehmet Akif Ersoy Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Yüksek Lisans Tezi / Doktora Tezi olarak sunduğum **“Parametrik Tasarım Araçları Kullanılarak Oluşturulan ve Robotik Üç Boyutlu Yazıcılar ile Üretilebilecek Barınma Birimi Tasarımı”** başlıklı bu tezin;

- Kendi çalışmam olduğunu,
- Sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi,
- Bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi,
- Kullandığım verilerde değişiklik yapmadığımı,
- Tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını,
- Bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı,
- Bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı, bildirir, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

22/04/2024

(İmza)

Ahmet KORKMAZ

ÖNSÖZ

Bu araştırma için beni yönlendiren, karşılaştığım zorlukları bilgi ve tecrübesi ile aşmamda yardımcı olan değerli Danışman Hocam **Prof. Dr. Sevim Ateş CAN**'a teşekkürlerimi sunarım.

Eğitim hayatımın her aşamasında beni her anlamda destekleyen aileme sonsuz sevgi ve saygılarımı sunarım.

Mayıs, 2024

Ahmet KORKMAZ



İÇİNDEKİLER

Sayfa

ÖNSÖZ	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	iii
SİMGELER VE KISALTMALAR DİZİNİ.....	vii
ÖZET	viii
ABSTRACT.....	ix
1. GİRİŞ	1
1.1. Problemin Tanımı.....	1
1.2. Tezin Amacı	2
2. GENEL BİLGİLER.....	3
2.1. Parametrik Tasarım	3
2.1.1. Parametrik Tasarımın Ortaya Çıkışı ve Gelişimi.....	5
2.1.2. Parametrik Tasarım Araçlarının Ortaya Çıkışı ve Gelişimi	11
2.1.3. Bir Model Verisi Üzerinden Geleneksel Tasarım Araçları ve Parametrik Tasarım Araçlarının Karşılaştırılması.....	18
2.1.4. Bir Mimarlık Akımı Olarak Parametrik Tasarım (Parametrisizm).....	28
3. 3. BOYUTLU BASKI (EKLEMELİ İMALAT) YAPABİLEN ROBOT KOLLAR ..	32
3.1. Dijital Fabrikasyon ve Eklemeli İmalat.....	32
3.2. 3 Boyutlu Yazıcıların Ortaya Çıkışı ve Gelişimi	41
3.3. 3 Boyutlu Yazıcı Tipleri.....	41
3.3.1. Kartezyen Tip Yazıcılar.....	41
3.3.2. Delta Tip Yazıcılar	42
3.3.3. Polar Tip Yazıcılar.....	42
3.3.4. Robotik (Scara) Yazıcılar	43
3.3.5. Non Planar- Düzlemsel Olmayan Yazıcılar	44
3.4. Yapı Ölçeğinde Eklemeli İmalat	44
3.4.1. Kontur Dolgu (Contour Crafting).....	45
3.4.2. D-Shape (D-Şekil)	46
3.4.2. Beton Yazdırma (Concrete Printing)	47
4. MATERYAL VE YÖNTEM.....	48
4.1. Materyal.....	48
4.1.1. Le Corbusier	48
4.1.2. Modülör	49
4.1.3. Le Cabanon.....	51
4.1.4. Bilgisayar Programları.....	53
4.2. Yöntem	53
5. ARAŞTIRMA BULGULARI VE TARTIŞMA	55
5.1. Barınma Biriminin Plan Oluşumu.....	55
5.2. Barınma Biriminin Kabuk Yapısı.....	57
5.3. Barınma Biriminin Parametrik Algoritması	59
6. SONUÇ VE ÖNERİLER	99
KAYNAKLAR.....	101

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 2.1. Astronomik hizalamalara göre yapılmış Roma yapısı olan Heidentor	6
Şekil 2.2. İki boyutlu algoritmik desen, İran'ın Şiraz şehrinde bir kubbe yüzeyinde (solda) ve iki yay arasındaki üç boyutlu geçiş(sağda).....	7
Şekil 2.3. Gaudi'nin Colonia Guell Kilisesi'ni tasarlamak için kullandığı analog parametrik mekanizma (sol) ve kilisenin nihai tasarımını gösteren bir model (sağ)	8
Şekil 2.4. Luigi Moretti'nin "Parametrik Mimarlık" araştırması modeli, 1960 Milan Trianeli.....	9
Şekil 2.5. Frei Otto'nun Sabun Köpüğü Deneyi	10
Şekil 2.6. Sketchpad ışıklı kalem kullanımı	12
Şekil 2.7. Sketchpad'deki on yedi atomik kısıtlamadan altısının diyagramı	13
Şekil 2.8. ArchiCAD algoritma ekseninde masa modelleme 1. Adım	18
Şekil 2.9. ArchiCAD algoritma ekseninde masa modelleme 2. Adım	18
Şekil 2.10. ArchiCAD algoritma ekseninde masa modelleme 3. Adım	19
Şekil 2.11. ArchiCAD algoritma ekseninde masa modelleme 4. Adım	19
Şekil 2.12. ArchiCAD algoritma ekseninde masa modelleme 5. Adım	20
Şekil 2.13. ArchiCAD algoritma ekseninde masa modelleme 6. Adım	20
Şekil 2.14. ArchiCAD algoritma ekseninde masa modelleme 7. Adım	21
Şekil 2.15. ArchiCAD algoritma ekseninde masa modelleme 8. Adım	21
Şekil 2.16. ArchiCAD algoritma ekseninde masa modelleme 9. Adım	22
Şekil 2.17. ArchiCAD algoritma ekseninde masa modelleme 10. Adım	22
Şekil 2.18. ArchiCAD algoritma ekseninde masa modelleme 11. Adım	23
Şekil 2.19. ArchiCAD algoritma ekseninde masa modelleme 12. Adım	23
Şekil 2.20. ArchiCAD algoritma ekseninde masa modelleme 13. Adım	24
Şekil 2.21. ArchiCAD algoritma ekseninde masa modelleme 14. Adım	24
Şekil 2.22. ArchiCAD algoritma ekseninde modellenen masa.....	25
Şekil 2.23. Masa'nın Grasshopper ile oluşturulan algoritması	25
Şekil 2.24. Masa'nın Grasshopper'daki parametreleri	26
Şekil 2.25. Masa'nın Grasshopper'daki değiştirilmiş parametreleri	26
Şekil 2.26. ArchiCAD parametrik masa objesinin parametre düzenleme arayüzü.....	27

Şekil 3.1. Basile Bouchon'un tezgahı, (yarı) otomatik bir makinenin ilk endüstriyel uygulaması.....	33
Şekil 3.2. John Wilkinson'un delme makinesi.....	33
Şekil 3.3. 80 sütunlu IBM'in geliştirdiği delikli kart EBCDIC	34
Şekil 3.4. Parsons'un patentini aldığı takım tezgahını konumlandırmak için motor kontrollü aparat.....	35
Şekil 3.5. CNC tezgahı ile üretilen alüminyum kül tablası	36
Şekil 3.6. İlk endüstriyel robot kol prototipi olan Unimate	36
Şekil 3.7. 6 Eksenli Rancho Kolu	37
Şekil 3.8. 3 eksenli CNC tezgahındaki eksenler	39
Şekil 3.9. 6 eksenli bir robot koldaki eksenler.....	39
Şekil 3.10. Çıkarımlı, biçimlendirmeli ve eklemeli imalat.....	40
Şekil 3.11. Kartezyen Tip Üç Boyutlu Yazıcı Şeması.....	42
Şekil 3.12. Delta Tip Üç Boyutlu Yazıcı Şeması.....	42
Şekil 3.13. Polar Üç Boyutlu Yazıcı Şeması	43
Şekil 3.14. Robotik (Scara) Yazıcı Şeması.....	43
Şekil 3.15. Kontur Dolgu sistemi.....	45
Şekil 3.16. İkinci nesil D-Shape makinesi	46
Şekil 3.17. Beton yazdırma şeması: 0. Sistem komutu; 1. Robot kontrolörü; 2. Yazdırma denetleyicisi; 3. Robotik kol; 4. Baskı Kafası; 5. Hızlandırıcı madde; 6. Peristaltik pompa hızlandırıcı madde; 7. Ön karışım için peristaltik pompa; 8. Ön karışım karıştırıcısı; 9. 3D baskılı nesne	47
Şekil 4.1. Modülör'un kuruluşu	49
Şekil 4.2. Modülör ve Altın Oran	50
Şekil 4.3. E.1027 ve Le Cabanon.....	52
Şekil 4.4. Le Cabanon plan oluşum dizisi.....	52
Şekil 5.1. Le Cabanon tefrişli plan.....	55
Şekil 5.2. Barınma birimi tefrişli plan ve ölçüleri	56
Şekil 5.3. Barınma birimi mobilya ilişkisi	57
Şekil 5.4 Birimin iç kabuğunu oluşturan eğrilerin referans kesitleri	58
Şekil 5.5 S-02 ve S-08 kesitlerinde kabuk eğrisi tefrişat ilişkisi	58
Şekil 5.6 Birimin kabuk yapısının referans eğrileri ile ilişkisi	59
Şekil 5.7 ArchiCAD'den gelen verilerin Grasshopper'a Aktarımı Algoritması.....	60
Şekil 5.8 Kapsayan eğrilerin başlangıç ve bitiş noktaları	60

Şekil 5.9. ArchiCAD’de Oluşturulan Eğrilerin Başlangıç ve Bitiş Noktaları	61
Şekil 5.10. Çatı boşluğunun oluşturulması, döndürülmesi ve bölümlendirilmesi algoritması	63
Şekil 5.11. Yapı kabuğunun ve duvar strüktürlerinin yüzeylerinin 3 boyutlu baskılama ile elde edilmesi için geliştirilen algoritma	64
Şekil 5.12. ArchiCAD’deki objelerin Grasshopper’a aktarımı.....	65
Şekil 5.13. Tefrişat öğelerinin eğri için referans noktalarının belirlenmesi	65
Şekil 5.14. Tüm objelerin Grasshopper’a aktarımı ve referans noktalarının belirlenmesi.....	66
Şekil 5.15. Çevreleyen Eğrilerin Oluşturulması	67
Şekil 5.16 Alt küme verilerinin bileşenlere atanması	68
Şekil 5.17. Çatı boşluğu için kesim düzlemi oluşturma	69
Şekil 5.18. Çevreleyen eğriler, offsetleri ve alt-üst trimleri kümesi.....	70
Şekil 5.19. Cluster işlevi çalıştırıldıktan sonra algoritmanın bileşene dönüşümü	70
Şekil 5.20. Çevreleyen eğriler, offsetleri ve alt-üst trimleri çıktısı.....	71
Şekil 5.21. Trim noktaları parametrelerine göre alt ve üst eğrilerin bölümlendirilmesi.	72
Şekil 5.22. Trim noktaları parametrelerine göre tüm alt ve üst eğrilerin bölümlendirilmesi	73
Şekil 5.23. Tekrar eden elemanların listeden çıkarılması	74
Şekil 5.24. Tüm tekrar eden elemanların listeden çıkarılması.....	74
Şekil 5.25. Strüktür için dalga akışı oluşturma	75
Şekil 5.26. Tüm strüktür için dalga akışı oluşturma	75
Şekil 5.27. Alt dalga offset	76
Şekil 5.28. Tüm eğriler için alt dalga offset.....	77
Şekil 5.29. Strüktür eğrisi bölümlendirme	78
Şekil 5.30. Strüktür eğrisi bölümlendirme tüm genişletmeler	78
Şekil 5.31. Eğriler ile Yüzey Oluşturma.....	79
Şekil 5.32. Eğriler ile yüzey oluşturma cluster algoritması	80
Şekil 5.33. Eğriler ile yüzey oluşturma tüm yüzeylerin oluşumu.....	80
Şekil 5.34. U-V yönünü ayarlama.....	82
Şekil 5.35. Tüm genişletme değerleri için U-V yönünü ayarlama	82
Şekil 5.36. Yüzeyleri 3D baskı pathwayi için bölümlendirme	83
Şekil 5.37. Tüm yüzeyleri 3D baskı pathwayi için bölümlendirme	84
Şekil 5.38. Pathwayleri sıralama.....	85

Şekil 5.39. Tüm pathwayleri sıralama	85
Şekil 5.40. Modül giriş kapısı verisinin archicad'den getirilip modül üretim toolpath eğrilerinden çıkarılacak olan box'un elde edilmesi.....	86
Şekil 5.41. Modül üretim toolpath eğrilerinden kapı kısmının çıkarılması	87
Şekil 5.42. Modül üretim takım yolu eğrilerinden giriş kapısı kısmının çıkarılıp organize edilmesi	88
Şekil 5.43. Kapı ve pencere boşluklarının takım yolu eğrilerinden çıkarılması.....	89
Şekil 5.44. Kapı ve pencere boşlukları için üretim takım yollarının yeniden organizasyonu.....	90
Şekil 5.45. Barınma biriminin kabuk yapısını oluşturan takım yolları.....	90
Şekil 5.46. İç duvar-1'i kapsayan yüzeylerin bulunması.....	91
Şekil 5.47. İç Duvar-1 Çizgisine Göre Duvar Kalınlığının Bulunması ve Baskı Ucu Çap Değerine Göre Offsetlenip Üretim Eğrilerinin Bulunması	92
Şekil 5.48. Üretim Eğrilerinin "z" Ekseninde Kapsayan Yüzeylere Yansıtılıp, "İç Duvar-1 Z-Loft Yüzeyleri"nin Elde Edilmesi	93
Şekil 5.49. "İç Duvar-1 Z-Loft Yüzeyleri"nin "x" Ekseninde Kapsayan Yüzeylere Yansıtılıp, "İç Duvar-1 X-Loft Yüzeyleri"nin Elde Edilmesi.....	94
Şekil 5.50. İç Duvar-1 Z-Loft Yüzeyleri" ve "İç Duvar-1 X-Loft Yüzeyleri"nin Birleştirilip "İç Duvar-1 Yüzeyleri"nin Elde Edilmesi	94
Şekil 5.51. İç duvar-1'in baskı ucu çap değerine göre bölümlendirilmesi.....	95
Şekil 5.52. İç duvarlar için oluşturulan algoritma kümesi	96
Şekil 5.53. İç duvarların bölümlendirilmiş takım yolu eğrileri	96
Şekil 5.54. Tüm model algoritması ve iç dış ilişkisi.....	97
Şekil 5.55. Barınma biriminin perspektif görüntüleri.....	97
Şekil 5.56. Tüm algoritma	98

SİMGELER VE KISALTMALAR DİZİNİ

BIM	: Building Information Modelling
CAD	: Computer Aided Design
CNC	: Computer Numeric Control
NC	: Numeric Control



ÖZET

Yüksek Lisans Tezi

**Parametrik Tasarım Araçları Kullanılarak Oluşturulan ve Robotik Üç Boyutlu
Yazıcılar ile Üretilebilecek Barınma Birimi Tasarımı**

Ahmet Korkmaz

**Burdur Mehmet Akif Ersoy Üniversitesi
Fen Bilimleri Enstitüsü
Mimarlık Anabilim Dalı**

Danışman: Prof. Dr. Sevim Ateş CAN

Ocak, 2024

Bu tez kapsamında, barınma biriminin kavramsal altyapısını oluşturmak için parametrik tasarım kavramı etimolojik olarak ele alınmış ve süreç boyunca kuramının nasıl şekillendiği incelenmiştir. Bunun yanında gelişimini sürdürmekte olan üç boyutlu yazıcıların ve yapı ölçeğinde üç boyutlu baskılama yapabilen endüstriyel robot kollarının tarihsel gelişimi araştırılarak bu yöntemle en küçük yapı ölçeğinde ürün üretme metotlarının çözümlenmesi amaçlanmıştır. Bu amaca binaen, barınma biriminin tasarımı için en az alanda en fazla verimliliği hedefleyen Le Corbusier'nin "Le Cabanon" adlı barınma birimi, tasarımın geliştirilmesinde kaynak/referans olarak kullanılmıştır. Bir parametrik tasarım aracı olan Grasshopper ve bir yapı bilgi modellemesi aracı olan ArchiCAD birlikte çalıştırılarak formunu fonksiyonundan alan bir model geliştirilmiş ve bu modelin düzlemsel olmayan katmanlı imalat yöntemi ile üretimi için görsel kodlama algoritması geliştirilmiştir.

Anahtar Kelimeler: algoritma, mimari tasarım, parametrik tasarım, robotik kol, üç boyutlu yazıcı

ABSTRACT

M. Sc. Thesis

Housing Unit Design Created Using Parametric Design Tools and Produced with Robotic Three-Dimensional Printers

Ahmet Korkmaz

**Burdur Mehmet Akif Ersoy University
Graduate School of Natural and Applied Sciences
Department of Architecture**

Supervisor: Prof. Dr. Sevim Ateş CAN

January, 2024

Within the scope of this thesis, in order to create the conceptual infrastructure of the model proposal, the concept of parametric design has been discussed etymologically and how its meaning has been shaped throughout the historical process has been examined. In addition, it is aimed to analyze the methods of producing products at the smallest building scale by researching the historical development of three-dimensional printers that continue to develop and industrial robot arms that can perform three-dimensional printing at the building scale. For this purpose, Le Corbusier's shelter unit called "Le Cabanon", which aims to maximize efficiency in the least amount of space for the design of the shelter unit, was used as a source reference in the development of the design. Grasshopper, a parametric design tool, and ArchiCAD, a building information modeling tool, were run together to develop a model that takes its form from its function, and a visual coding algorithm was developed to produce this model with the non-planar additive manufacturing method.

Keywords: algorithm, architectural design, parametric design, robotic arm, three-dimensional printer

1. GİRİŞ

İnsanoğlu var oluşundan beri işlerini ve görevlerini gerçekleştirmek için belirli araçlar kullanma ve geliştirme motivasyonunda olmuştur. Bu motivasyonun kaynağı genellikle zaman verimliliğini ve üretim kalitesini arttırmaktır. Yunan filozof Aristoteles'in M.Ö. IV. yy'da ifade ettiği, her araç kendi işini gerçekleştirebilseydi, bireylere ihtiyaç duyulmadan birçok işin gerçekleştirilebileceği düşüncesi (Barutçuoğlu, 2001) insanoğlunun geliştirdiği araçlara da yeni bir misyon yükleyip otomasyon fikrinin temellerini oluşturmuştur. Bu kapsamda, Basile Bouchon tarafından 18. yüzyılda önceden tanımlanmış bir dokumayı gerçekleştirmek amacıyla otomatik dokuma tezgâhı için delikli kartlar geliştirilmiştir. Bu delikli kartların çalışma prensibi numerik kontrol sistemlerinin geliştirilmesini sağlamış ve bu sistemler de günümüz dünyasını şekillendiren en önemli olgulardan biri olan veri biliminin temellerini oluşturmuştur. Geliştirilen delikli kartlar öncelikle numerik kontrol sistemlerine dönüştürülmüş ve otomatik tezgahlar geliştirilmiştir. Otomatik tezgahların gelişi bilgisayar donanımları dahil birçok endüstriyel ürünün üretimini kolaylaştırmıştır. Bu durum bilgisayar teknolojilerinin yaygınlaşmasını ve bilgisayarlı numerik kontrol sisteminin ortaya çıkmasını sağlamıştır. Bilgisayarlı numerik kontrol sistemleri, dijital ortamda daha kolay ve optimize edilebilir şekilde otomatik tezgahların eksen hareketlerinin önceden tanımlamasını ve hata kontrol mekanizmalarının ortaya çıkmasını sağlamıştır. Otomatik tezgahların farklı eksenlerde hareket eden motorlarını organize kontrol etme altyapısı benzer dönemlerde robot kolların çalışma prensibine de uygulanmış ve robot kollar ortaya çıkmıştır. Robot kollar ve getirdikleri otomasyon ortaya çıktığı dönemde devrim yaratmış ve günümüzün modern üretim altyapısının kaynağı olmuştur. Veri biliminin temellerini oluşturup mekanik üretim altyapısına daha sonra da seri üretim olarak tanımlanan günümüz gelişmiş endüstriyel üretim metoduna aktarılan tüm bu süreçler birbiriyle etkileşim halinde ve birbirlerini doğrudan besleyen süreçlerdir.

1.1. Problemin Tanımı

Yukarıda bahsedilen, otomasyon düşüncesine bağlı gelişen teknolojiler ve bunlara bağlı oluşan üretim yöntemleri çıkarımlı ve biçimlendirmeli olarak gelişmiş ve daha sonra otomatik tezgahlara ve robot kollara malzeme ekstrüze edebilecek başlıklar geliştirilerek üç boyutlu baskılama teknolojisi ortaya çıkmıştır.

Robot kolların üretimde devrim yarattığı ve birçok alanda kendine yaygın bir kullanım edindiği gözlemlenmiş fakat bu durumun mimarlıkta yapı üretim otomasyonunun sağlanması noktasında eksik kaldığı gözlemlenmiştir.

Üç boyutlu baskılama yöntemiyle yapı ölçeğinde üretiminin yakın dönemde yaygınlaştığı ve yapılan çalışmaların arttığı fakat üretim tekniklerinde genellikle yapıların duvarlarının bu yöntemle üretilip üst örtüsünün harici yöntemlerle çözüldüğü saptanmıştır. Bu durum üç boyutlu baskılama teknolojisinin genellikle 3 ekseninde hareket eden düzlemsel olan baskılamayla kullanımının kısıtlılıklarından kaynaklanmaktadır. Fakat robot kolların sahip olduğu fazla eksen sayısı düzlemsel olmayan baskılamayı mümkün kılmakta ve bu da robot kolun hareket kapasitesi gibi kısıtlamalar dahilinde düzlemsel olmayan üç boyutlu baskılama sayesinde üst örtünün de bu üretim yöntemiyle üretilmesini sağlayabilmektedir.

1.2. Tezin Amacı

Tezin amacı kısaca robot kollar ile düzlemsel olmayan baskılama yöntemiyle inşa edilebilecek bir barınma birimi için, yapı bilgi modellemesi ve hesaplamalı tasarım yazılımları eş zamanlı çalıştırılarak bir algoritma geliştirmektir. Bu kapsamda ArchiCAD ve Grasshopper yazılımları bir arada çalıştırılmıştır. Grasshopper'ın getirdiği karmaşık geometrilerin kontrolü sayesinde de bu birimin üretiminin dijital ortamda simüle edilmesi için robot kolun izleyeceği takım yolları elde edilmektedir.

Tezin diğer amacı otomatik tezgahların, robot kolların ve bilgisayar teknolojilerinin gelişimlerinin birbirini destekleyen unsurlar olduklarını saptamaktır. Bu doğrultuda bu teknolojilerin gelişimleri en temelden ele alınmış ve birbirleriyle olan ilişkileri tarihsel süreçte bir arada incelenmiştir.

Bu bağlamda çalışmanın hipotezi oluşturulurken günümüzde parametrik tasarım araçları olarak adlandırılan ve görsel kodlama ile algoritmalar üretilmesine dayanan olgulara karşılık gelen parametrik tasarım kavramı derinlemesine incelenmiş ve geliştirilen modelin kavramsal altyapısının parametrik tasarım düşüncesiyle örtüşür olması hedeflenmiştir.

2. GENEL BİLGİLER

2.1. Parametrik Tasarım

Parametrik tasarım kavramını tanımlayabilmek için öncelikle kavramı oluşturan ve “parametre” kelimesinden türetilen “parametrik” kelimesinin kökenini irdelemek gereklidir.

"Parametre" kelimesi, Eski Yunanca kökenli "pará" (bir şeyin yanında olma, yanyana durma, benzer ve gibi olma, aslı veya gerçeği olmama bildiren edat ve fiil öneki) ile "métrēō" (ölçmek) fiilinin birleşmesiyle oluşmuştur. Bu kelime, Fransızca'da "bir başka değere bağımlı olarak değişen ölçü" anlamındaki "paramètre" sözcüğünden alıntıdır. Fransızca kelime ise Eski Yunanca "paramétrēō" παραμέτρεω "bir şeyi bir şeyle ölçmek, kıyaslamak" fiilinden türetilmiştir (Etimolojitürkçe, 2023).

Türk Dil Kurumu Sözlüğü'ne göre Türkçe'ye Fransızca'dan geçen “parametre” terimi “değişken”, “parametrik” terimi ise “değişkenli” yani içerisinde bir veya birden çok parametre bulunduran durum anlamına gelmektedir (Türk Dil Kurumu Sözlükleri, 2023).

Parametre teriminin Türkçe literatürde geçtiği ilk kaynak TDK, Türkçe Sözlük, 1. Baskı (1945)'dir (Etimolojitürkçe, 2023).

Parametre kavramı farklı bilim dallarında aynı anlam paydasında bütünleşecek şekilde tanımlanmakta ve kullanılmaktadır. Parametrenin bazı kullanımları şunlardır: Matematikte bir parametre, bir fonksiyon tanımında yer alan ve fonksiyonun şeklini veya davranışını belirleyen bir değişkendir. Örneğin a , b ve c sabitlerinin atandığı, $f(x)=ax^2+bx+c$ fonksiyonunda, parametre(değişken) “ x ”in aldığı değerdir. Ve bu parametreye farklı sayılar atandığında fonksiyonu oluşturan sabitlerin oluşturduğu işlemler bütünlüğüne bağlı olarak denklemin sonucu değişecektir.

Bilgisayar programlamasında bir parametre, bir fonksiyon veya alt program çağrıldığında ona geçirilen bir değerdir. Örneğin, `print(message)` fonksiyonunda, parametre “message”dir ve ekrana ne yazılacağını belirtir, “print” ise ekrana yazılacağını belirten komuttur.

Mühendislikte bir parametre, bir sistemi veya bir bileşeni karakterize eden bir fiziksel büyüklüktür. Örneğin, bir devrede, parametreler direnç, kapasite, gerilim, akım gibi unsurlardır.

Farklı bilim dallarına ek olarak Mimarlıkta parametre kavramını tanımlamak gerektiğinde, mimari tasarım süreci ele alınmalıdır. Mimari tasarım sürecinde mimar, bir

binayı tasarlamaya başlamadan önce ve tasarım süreci boyunca tasarımı şekillendirecek değişkenlere sahiptir. Bu değişkenlere yer/bölge, konsept, fonksiyon, maliyet, kullanıcı profili vb. girmektedir. Mimari tasarım süreci matematikte bir formül gibi ele alınırsa parametrelere/değişkenlere bağlı olarak farklı tasarımlar/sonuçlar elde edilir.

Parametrik Tasarım kavramını tanımlayabilmek için bu kavramla doğrudan ilişkili olan algoritma kavramının da irdelenmesi gereklidir.

Algoritma terimi, kelime kökeni olarak 9. yüzyılda yaşamış İranlı matematikçi El-Harizmi'den (Al-Khwarizmi) gelmektedir. El-Harizmi, cebir alanında yaptığı algoritmik çalışmalarını "Hisab el-cebir ve el-mukabala" adlı eserinde kitaplaştırarak matematik dünyasına önemli katkılarda bulunmuştur (Tedeschi ve Wirz, 2014).

Bu eser, dünyanın ilk cebir kitabı olarak kabul edilir ve aynı zamanda algoritma verilerini içeren ilk örnektir. Latince çevirisi Avrupa'da büyük ilgi görmüş ve alimin adını doğru bir şekilde telaffuz edemeyen Avrupalılar tarafından "algorism" terimi, "Arap sayıları kullanarak aritmetik problemler çözme kuralları" anlamında kullanılmıştır. Bu terim daha sonra "algoritma"ya dönüşmüş ve genel anlamda kullanılmaya başlanmıştır (Arndt, 1983).

Algoritma ortaya çıktığı matematik biliminde tanımlandığı üzere en basit haliyle bir süreç bütünü olarak değerlendirilmektedir. Bu sürecin başında “girdi”ler vardır ve bu girdiler algoritma boyunca tanımlanmış işlemlere tabi tutularak “çıkı”ları oluştururlar. Algoritmalar genelde şu temel özelliklere sahiptir:

Tanımlı Adımlar: Algoritma, belirli ve sıralı adımlar içerir. Her adım, belirli bir görevi gerçekleştirmek için açık bir şekilde tanımlanmıştır.

Sistematik: Algoritma, adımların belli bir düzen içinde ve belirli bir mantık çerçevesinde sıralandığı bir sistematığe sahiptir.

Sonlu: Algoritma sonlanmalıdır. Sonsuz döngülere veya işlemlere izin vermez.

Giriş ve Çıkış: Her algoritma bir giriş alır, bu girişi işler ve bir çıkış üretir.

Bu tanıma bağlı olarak ikinci derece bilinmeyenli bir denklemin($ax^2+bx+c=0$) köklerini bulmak için gerekli işlemleri algoritma ekseninde oluşturmak gerekirse elde edeceğimiz 3 adımdan oluşan algoritma:

“1- Katsayıları a, b ve c'yi al.

2- Diskriminantı hesapla: $\Delta = b^2 - 4ac$

Δ 'nın değerine göre durumları ele al:

a. $\Delta > 0$ ise:

İki gerçek kök vardır.

Kökler şu şekildedir: $x_1 = (-b + \sqrt{\Delta}) / (2a)$ ve $x_2 = (-b - \sqrt{\Delta}) / (2a)$

b. $\Delta = 0$ ise:

İki kök birbirine eşittir (çakışık kök).

Kökler şu şekildedir: $x_1 = x_2 = -b / (2a)$

c. $\Delta < 0$ ise:

Gerçek kök yoktur (kompleks kök).

Kökler şu şekildedir: $x_1 = (-b + i\sqrt{|\Delta|}) / (2a)$ ve $x_2 = (-b - i\sqrt{|\Delta|}) / (2a)$

3- x_1 ve x_2 değerlerini yaz.” şeklindedir.

Bu algoritmada; girdiler a,b ve c değerleridir. Çıktılar ise x_1 ve x_2 değerleridir.

Görüldüğü üzere algoritmada tanımlı işlem grubu değişken değildir fakat a,b ve c değerleri değişken/parametre olarak kabul edilebilir. Bu parametreler farklı değerler aldığı anda girdiler ve çıktılar değişmektedir.

Tüm bu veriler ışığında kavramlar en basit tabirleriyle:

Parametre/değişken: Değiştirilebilen nicelik

Parametrik: İçerisinde birden fazla parametre/değişken bulunan

Algoritma: Parametrelerin işlem gördüğü süreç bütünü

Parametrik Tasarım: Parametrelere bağlı gelişen tasarım süreci şeklinde tanımlanabilir.

Parametrik Tasarım geniş perspektifte ele alındığında bir algoritma olarak konumlanmaktadır.

2.1.1. Parametrik Tasarımın Ortaya Çıkışı ve Gelişimi

Parametrik Tasarım kavramı etimolojik olarak incelendiğinde kavram ve kavramı oluşturan olgular Parametrik Tasarım ana başlığı altında somutlaştırılmıştır.

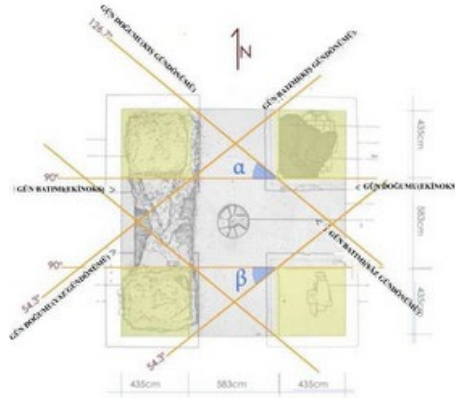
Parametrik Tasarım, mimarlar için yeni bir şey değildir ve tasarım elemanlarını etkileyen iç ve dış değişkenler arasındaki ilişkileri parametrelere çevirerek tasarım problemlerini çözmek için kullanılan, algoritmik düşünceyle ilgili bir kavramdır. Mimari tasarım bir noktada parametrik tasarımın da karşılığıdır.

Bilgisayar destekli parametrik modelleme yazılımları tasarımcılar arasında popülerlik kazandığından beri; Parametrik Tasarım terimi, Hesaplamalı Parametrikçilik ile karıştırılmış ve bu terimle sınırlanmıştır. Aslında, Parametrik Tasarımın mimarlık tarihinde derin kökleri vardır.

2.1.1.1. 19. yüzyıl öncesi

Bilindiği üzere antik medeniyetler, yapıları inşa ederken astronomik gözlem referanslarını ve astronomik hesaplamaları kullanmışlardır. Bu yaklaşım, matematik, geometri, astronomi ve coğrafyanın antik çağlarda aynı disiplin içinde bulunduğu dönemlere kadar uzanmaktadır. Dolayısıyla, büyük ve o dönem için karmaşık yapı tasarımları genellikle astronomik ve astrolojik hesaplamalardan türetilmiş bir geometri türü tarafından yönlendirilmiştir. Örneğin, Persler ve Romalılar ilk kubbe inşa girişimlerini, astroloji ve gökyüzüne referansla gerçekleştirmiştir. Romalılar, kubbeler ile gökyüzü arasında bir benzerlik kurmuşlardır. Roma'daki Pantheon, mimari geometrinin tanrıların ve göksel düzenin alanını nasıl temsil ettiğine dair bir örnektir (Assasi, 2019).

Persler, chartaqui adı verilen yapıları inşa etmişlerdir. Bu yapılar, dört sütunun desteklediği bir kubbenin, kare bir temel izi üzerine ve haç planına sahip olmasıyla öne çıkar. Ayrıca, bu yapılar yerel ufuktaki ekinoks ve gündönümü noktalarına yönlendirilmiştir. Bu durum, mimari geometrinin gökyüzü kubbesinin düzenine referans verdiğini gösterir. Romalılar da bazı yapılarında benzer astronomik parametreleri kullanmışlardır (Şekil 2.1) (Assasi, 2019).



Şekil 2.1. Astronomik hizalamalara göre yapılmış Roma yapısı olan Heidontor (Assasi, 2019)

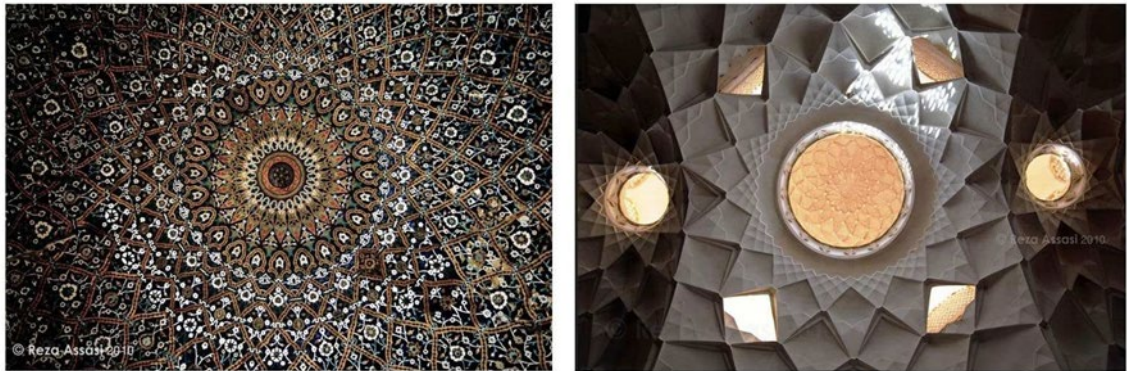
İslam medeniyetlerinin Roma ve Sasani İmparatorlukları'nı ele geçirmesinden sonra, bu yaklaşım İslam mimarisine adapte edilmiştir. Mimarlık geometrisi ve dekoratif desenler genellikle saf geometrinin geometrik soyutlamaları biçiminde ilahi olanı temsil etmek için kullanılmıştır. Genellikle çiçek motifleriyle karıştırılmış ve bu motiflerin dünyevi olmayan ilahi düzeni temsil ettiği varsayılmıştır. İslam geleneği, insan ve hayvan figürlerinin herhangi bir temsilini yasakladığı için soyut geometrinin kullanımı daha da yaygın hale gelmiştir. Ancak Ürdün'deki Qasr Amrah'daki Emevi sarayındaki bir

kubbenin üzerinde göksel cisimlerin ve astrolojik figürlerin resimlerine dair bir kayıt bulunmaktadır (Baker, 1980).

Bu durum İslam mimarisinde de bir kubbenin yapısında ve göksel cisimlerin arasında bir bağlantı olduğunu göstermektedir.

Soyut geometrik formların, şekillerin ve desenlerin kullanımı, İslam öncesi dönemde Yunanlılar, Romalılar ve Persler arasında zaten var olan bir kavram olduğu bilinmektedir. Daha önce bahsedildiği gibi, İslam geleneği herhangi bir formda insan veya hayvan temsilini yasaklamıştır. Bu kısıtlama sayesinde; matematikçiler, astronomlar ve bilim adamları sanat ve mimaride matematiksel desenlerin geniş kullanımına ve algoritmik yöntemlerin geliştirilmesine önemli katkılarda bulunabilmişlerdir. Bu dönemde zanaatkârlar tekstil, halı, seramik, çömlek ve dekoratif resimler için soyut geometrik formlardan oluşan desenleri üretmişlerdir.

Amaç, düz yüzeylere yerleştirilmiş bu desenleri kubbeler ve kemerler gibi eğimli yüzeylere yansıtmaktır (Şekil 2.2). Bunun için karmaşık algoritmik yöntemler gerekmektedir. Düz geometriyi bir düzlemden bir küreye çevirme saplantısının kökeni, antik çağlarda felsefi ve kozmolojik gelişmelerden sonraki dönemlere dayanmaktadır. Antik çağlarda, küresel geometriyi bir düz yüzeye yansıtmak için yapılan bu girişimler sırasında edinilen bilgilere ulaşmak için İslam dönemi bilim insanları orijinal Yunanca eserlerin Arapça tercümelerinden faydalanmıştır (Sidoli ve Berggren, 2007).



Şekil 2.2. İki boyutlu algoritmik desen, İran'ın Şiraz şehrinde bir kubbe yüzeyinde (solda) ve iki yay arasındaki üç boyutlu geçiş, İran'ın Kaşan şehrinde (sağda) (Assasi, 2019)

2.1.1.2. 19. Yüzyıl

Parametrik kavramının analitik geometri ve 3 boyutlu temsil alanında kullanımları 19. yüzyıl boyunca gelişimini sürdürmüştür. Kavramın ilk kullanımı James Dana'nın 1837'deki "Kristallerin Figürlerinin Çizimi Üzerine" adlı makalesine kadar uzanır.

Makalede Dana, parametreler, deęişkenler ve oranlarla bağlanmış bir dil kullanarak bir dizi kristal çizmek için genel adımları ve varyasyonlar için hükümleri açıklamaktadır. Dana bu çalışmasında parametrik terimini kullanmamıştır, ancak geliştirdiğı sistematik yıllar sonra Antoni Gaudi, Frei Otto ve Luigi Moretti gibi mimarlar tarafından form bulma amaçlı olarak kullanılacaktır (Davis, 2013a).

Diğer örnekler arasında John Leslie'nin 1821'de yayımlanan "Eğri Doğrularının Geometrik Analizi ve Geometrisi"(Leslie, 1821) ile Samuel Earnshaw'ın 1839'da yayımlanan "Aydınlık Eterin Oluşumunu Düzenleyen Moleküler Kuvvetlerin Doğası Üzerine" (Earnshaw, 1839) adlı çalışmaları yer almaktadır.

19. yüzyılın sonlarına doğru Avrupa'da, mühendisler ve mimarlar arasında parametrik tasarım konusunda uygulamaya yönelik yaklaşımlar ortaya çıktığı gözlenmektedir. Örneğin, Antonio Gaudi, karmaşık yapı formlarını tasarlamak için analog sistemleri kullanmıştır. Colonia Guell Kilisesi tasarımında, bağlantılı kemerler ve tonozlu tavanlar tasarlamak için kullanılan bir analog modelde, kuş tüyü ipliklerden oluşturulmuş ters bir analog model kullanmıştır. Gaudi, ölü yüklerin etkili bir şekilde dağıtılmasını sağlayan karmaşık bir sistem tasarlamak için tersine çevrilmiş bir yapı modelinde zincir eğrisi metodunu kullanmıştır. Yükleri manipüle ederken mimari formu inceleyebilmek için de modelin altına bir ayna yerleştirmiştir (Şekil 2.3) (Davis, 2013b).



Şekil 2.3. Gaudi'nin Colonia Guell Kilisesi'ni tasarlamak için kullandığı analog parametrik mekanizma (sol) ve kilisenin nihai tasarımını gösteren bir model (sağ) (Assasi, 2019)

Gaudi zincir eğrisi yöntemini daha sonraki tasarımları için geliştirerek kullanmaya devam etmiş ve tamamlanmasına ömrünün yetmediğı meşhur tasarımı La Sagrada Familia'da da kullanmıştır. Çok sayıda parametreye/değişkene sahip kompleks bir

durumun sabit bir yöntem ile kontrol edilmesi parametrik tasarım anlayışı noktasında önemli bir örnektir.

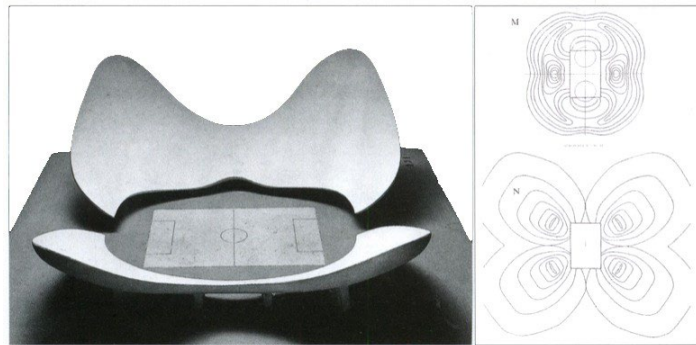
Gaudi'nin yönteminde; yükler parametrelerin, gergi tabanlı model sistemi de algoritmanın karşılığı olmaktadır. Bu algoritmanın çıktıları ise zincir eğrilerinin parametrelere bağlı anlık değişen şeklidir. Bu yönüyle Gaudi'nin gergi tabanlı model sistemi parametrik tasarımın gelişimi noktasında kritik önem taşımaktadır. Zincir eğrisi, X eksenini etrafında döndürüldüğünde katenoidi veya minimum yüzeyi veren matematiksel bir eğridir (Burry ve Burry, 2010).

Davis'in aksine Gaudi'nin daha önce geometriyi tanımlamak için parametrik denklemleri kullanan çeşitli bilim adamları ve matematikçilerden doğrudan etkilenip etkilenmediğini bilmek imkansızdır. Gaudi'nin Sagrada Familia'sının yönetici mimarlığını yapan Mark Burry, "Gaudi'nin kendisi tarafından onu sınırları zorlamaya iten motivasyonlar, teoriler ve uygulamalar hakkında yazılmış neredeyse hiçbir şey olmadığını" ifade etmiştir (M. Burry, 2007).

2.1.1.3. Modern zamanlar

Parametrik Tasarım kavramı "Parametrik Mimarlık" başlığı altında ilk kez İtalyan mimar Luigi Moretti tarafından 1939 yılında yazdığı 'Architettura Parametrica' kitabında tanımlanmıştır (Bucci ve Mulazzani, 2000).

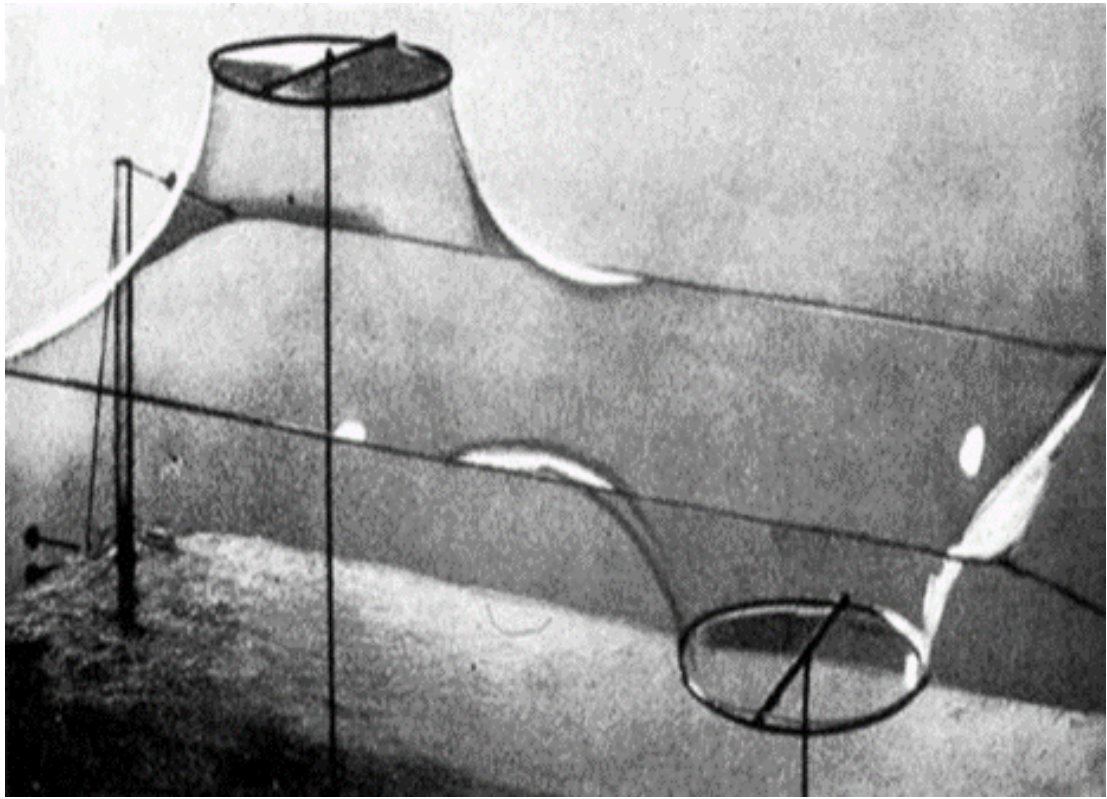
1960'ta 12. Milan Trienal Sergisi için geliştirdiği stadyum modeli, Luigi Moretti'nin parametrik kavramını ilk defa somut bir şekilde uyguladığı örnekler arasında yer almaktadır. Bu çalışma, günümüz standartlarına göre oldukça ilkel kabul edilebilecek bilgisayar konfigürasyonları kullanılarak gerçekleştirilmiştir. Moretti'nin belirlediği parametreler arasında izleyicilerin görüş açısı ve ekonomik uygulanabilirlik önemli yer tutmaktadır (Yıldırım, 2019).



Şekil 2.4. 1960 Milan Trianeli Luigi Moretti'nin "Parametrik Mimarlık" araştırması modeli (Davis, 2013b)

Moretti'nin 1960 yılındaki bu çalışmasından 1 yıl sonra Frei Otto ipler ve sabun köpükleri kullanarak tıpkı Gaudi gibi analog bir sistem geliştirmiştir. Frei Otto mimarlık literatüründe "form-finding" (form-bulma) terimini ilk kullanan kişidir; fiziksel modellerin tasarım motoru ve form-bulma aracı olarak kullanımına katkısı nedeniyle, Otto mimarlıkta parametrik tasarımın öncülerinden biri olarak kabul edilmektedir (Zexin ve Mei, 2017).

Otto, hafif germe ve membran yapılar konusunda uzmanlaşmıştır ve formlar, bir kapalı şekli örtmek için minimum yüzeyi bulmak için sıvının yüzey gerilimini kullanan bir dizi sabun köpüğü deneyinden türetilmiştir (Şekil 2.5) (Lopes vd., 2014).



Şekil 2.5. Frei Otto'nun sabun köpüğü deneyi (Zexin ve Mei, 2017)

Antoni Gaudí ve Frei Otto'nun deneylerinde kullandıkları hesaplama yöntemi, "Analog Hesaplama" veya "Malzeme Hesaplama" olarak bilinir. Hem Frei Otto'nun ekibi tarafından kullanılan "optimize edilmiş yol sistemleri" hem de Gaudí'nin Colonia Guell Kilisesi için kullandığı modelleme tekniği, formun hesaplanması (veya form bulma) için bir malzeme sistemini deneyimlemiştir (Leach, 2009).

Gaudi, Moretti ve Otto'nun tasarımları parametrik mimarlık ekseninde fonksiyona dayalı form bulma girişimleri olarak tanımlanabilir. Bu durum parametrik tasarım anlayışının 1895 yılında Amerikan mimar Louis Sullivan tarafından ortaya atılan "form

follow function” (form fonksiyonu takip eder) sloganıyla uygunluk içerisinde olduğunu göstermektedir.

2.1.1.4. Bilgisayarın sürece dahil oluşu

Hesaplamanın sayısallaştırılması, Gaudi ve Otto'nun analog parametrik modelleriyle mümkün olmayan hesaplamaları kolaylaştırmıştır. Gaudi ve Otto'nun seçilmiş parametrik denklemlerin hesaplanmasını hızlandırmak için fizik yasalarını kullanmaları gibi, Ivan Sutherland da herhangi bir parametrik denklemin hesaplanmasını hızlandırmak için bilgisayarları kullanmıştır (Sutherland, 1963).

Sutherland, yazılarında parametrik terimini hiç kullanmamıştır, ancak atomik kısıtlamalar bir parametrik denklemin tüm temel özelliklerine sahiptir: her kısıtlama, bir dizi bağımsız parametrenin açık bir fonksiyonu olarak ifade edilen bir dizi sonuca sahiptir.

Gaudi ve Otto'nun modellerinden farklı olarak, bu parametrik denklemler fiziksel yasalara bağlı değildir, sadece paralel, ortogonal ve çakışma gibi ilişkileri hesaplayabilmektedir.

Sketchpad ilk parametrik tasarım aracı olması yönüyle parametrik tasarımın gelişiminde çok büyük öneme sahiptir. Sketchpad'ın 1963 yılında ortaya çıkışı ile birlikte parametrik tasarımın gelişimi, parametrik tasarım araçlarındaki geliştirmeler üzerinden ilerlemiştir. Bu gelişmeleri tetikleyen en önemli unsurlar, aynı dönemde etkisini arttıran postmodernizm ve dekonstrüktivizm akımlarının tasarımcıları yeni formlar bulmaya itmesi ve bunun yanı sıra 1970'lerde başlayan bilgisayar teknolojilerindeki hızlı gelişmelerdir.

2.1.2. Parametrik Tasarım Araçlarının Ortaya Çıkışı ve Gelişimi

Tez kapsamında parametrik tasarımın gelişimi Sketchpad ile başlayan parametrik tasarım araçlarının ortaya çıkışı ve parametrik tasarım noktasında getirdikleri yenilikler üzerinden irdelenecektir.

Sketchpad, MIT Lincoln Labs'ın TX-2 bilgisayarında geliştirilen ilk bilgisayar destekli çizim aracıdır ve insan-bilgisayar etkileşimini test etmek amacıyla tasarlanmıştır, bu nedenle "Robot Ressam" olarak da adlandırılmıştır. TX-2, dönemin en büyük makinelerinden biri olan 306 kilobayt çekirdek belleği ile donatılmıştır. O dönemin ileri seviye özelliklerine sahip olan TX-2, 320 KB ana belleğe, 8 MB manyetik bant depolama aygıtına, 7 inç 1024×1024 monitöre, hafif kalem ve düğme kutusuna sahiptir. Programlar

genellikle makro montaj dilinde yazılmakta, delinmiş kağıt bantlara kaydedilmekte ve bilgisayarın kağıt bant okuyucusuna aktarılmaktadır (Sutherland, 1964).

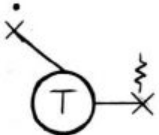
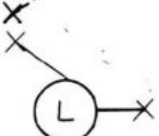
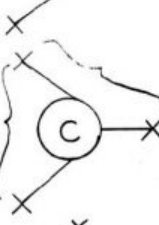
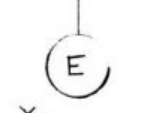
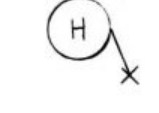
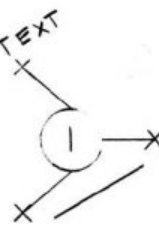
Önceki bilgisayar uygulamalarının genellikle toplu iş odaklı olmalarının aksine, Sketchpad etkileşimli bir yaklaşım sunmaktadır. Sketchpad kullanıcıları, "ışıklı kalem" (Şekil 2.6) aracılığıyla doğrudan bilgisayar ekranında çizim yapabilmektedir. Işıklı kalem, çizimin bölümlerini ekranda konumlandırmak ve değiştirmek için kullanılmaktaydı. Işıklı kalem ve giriş düğmeleri aracılığıyla kullanıcılar, artı işareti ile ekranda doğrudan çizim yapabilmekteydiler. Program, noktaları, çizgi parçalarını ve yayları temel öğeler olarak destekliyordu; ancak bu öğeler, kopyalanabilen veya örneklenebilen ana çizimlere kaydedilebiliyordu. Basılabilir düğmeler aracılığıyla, kullanıcılar değişiklikleri kontrol etmek için "sil", "taşı" gibi işlemleri gerçekleştirebiliyordu (Inspirationtuts, 2021).



Şekil 2.6. Sketchpad ışıklı kalem kullanımı (Inspirationtuts, 2021)

Sketchpad, 17 farklı türde kısıtlama içermektedir (Şekil 2.7); bu kısıtlamalar dikey, yatay, çakışan, paralel, hizalanmış, eşit boyut vb. gibi doğal veya "atomik" olarak adlandırılmaktadır (Davis, 2013b).

Sketchpad, kısıtlamaların görsel olarak ekranda temsil edilmesine olanak tanıyan simgeler kullanarak her türlü kısıtlamayı ifade etme imkânı sağlamaktadır. Kısıtlama sistemi sayesinde, bir şekli serbestçe çizmek ve ardından geometrik ve topolojik ilişkiler ekleyerek istenilen şekle dönüştürmek mümkündür. Bu kısıtlamalar düzenlenebilmekte ve daha karmaşık ilişkiler oluşturmak için kullanılabilir (Sutherland, 1964).

	43 T	point instance (point)	Point bears same relation to instance that (point) bears to its picture. GENERATED AUTOMATICALLY WITH INSTANCES
	33 L	p thing p thing p thing	Three things are collinear. Note: no distinction made about ordering of variables. GENERATED AUTOMATICALLY WHEN POINTS ARE CREATED ON LINES
	22 C	p thing p thing p thing	Distance from first to second is equal to distance from first to third. (First is circle center.) GENERATED AUTOMATICALLY WHEN POINTS ARE CREATED ON CIRCLES
	24 E	4 thing	Thing is erect or on its side. ↑ → ↓ ←
	27 H	p thing p thing	First thing is directly above or below, or directly beside second thing. (Horizontal or vertical line.) GENERATED AUTOMATICALLY FOR ANY LINE BY HORV BUTTON
	30 I	4 thing p thing p thing	4 thing is "parallel" to line between p things. Parallel to horizontal line means upright. (To set angle of text.)

Şekil 2.7. Sketchpad'deki on yedi atomik kısıtlamadan altısının diyagramı (Sutherland, 1963)

Sketchpad bu yönüyle, parametrik denklemleri keşfetmenin yeni bir yolunu sunmuştur. Gaudí ve Otto'nun modellerinde olduğu gibi, tasarımcılar parametreleri değiştirerek ve Sketchpad'ın geometriyi otomatik olarak yeniden hesaplamasını ve yeniden çizmesini sağlayarak varyasyonları keşfedebilmektedir. Tasarımcıların atomik kısıtlamaları değiştirmekte özgür olması geometrinin yeniden hesaplanmasına ve yeniden çizilmesine olanak sağlamaktadır. Bu nedenle, tasarımcının Sketchpad üzerindeki kontrolü, çoğu parametrik modelleme yazılımında olduğu gibi, yalnızca modelin parametreleri aracılığıyla değil, aynı zamanda modelin temel ilişkileri aracılığıyla da yapılabilmektedir.

Bundan dolayı Sketchpad tarihteki ilk bilgisayar destekli çizim aracı olmasının yanı sıra aynı zamanda tarihteki ilk parametrik tasarım aracıdır.

1960'lar ve 1970'ler bilgisayar teknolojilerinin gelişiminin ivme kazandığı dönemlerdi. Fakat bu süreç içerisinde halen bilgisayarlara mobilizasyon sağlanamamıştır.

Örneğin, Itek'in "Electronic Drafting Machine" (Elektronik Çizim Makinesi) gibi ilk ticari sistemler, 1962'de piyasaya sürüldüklerinde cihaz başına 3,5 milyon ABD dolarına mal oluyordu ve bu, yalnızca seçkin otomotiv ve havacılık şirketleri tarafından karşılanabilir bir maliyetti (Weisberg, 2008).

Bu yüzden bilgisayar destekli çizim araçları bu süreçte mimaride kullanıma tam olarak adapte olamamıştır. Bilgisayarların normal kullanıcılara ulaşabilecek kadar ucuzlaması, zamanla teknolojinin gelişmesi ve üretim maliyetlerinin düşmesiyle gerçekleşmiştir. 1980'ler, kişisel bilgisayarların popülerleştiği ve ev kullanıcıları için daha erişilebilir hale geldiği dönemlerdir. Bu dönemde, bilgisayar fiyatları giderek düşmüş ve kişisel bilgisayarlar evlerde daha yaygın bir araç haline gelmiştir. İlk kişisel bilgisayarlar genellikle masaüstü bilgisayarlar olarak bilinmektedir ve bugünkü bilgisayarlarla kıyaslandığında daha sınırlı özelliklere sahiptir (Davis, 2013b).

Bu dönemde Ağustos 1982'de AutoCAD piyasaya sürülmüş ve sonradan hakim olacağı bilgisayar destekli tasarım endüstrisinde hızla yükselmiştir (Weisberg, 2008). AutoCAD'de tasarımcının kalem yerine fare ve klavye kullanarak ekranda iki boyutlu çizgiler çizmesini sağlayan komutlarla birlikte değiştirilen eğriler ve kendi kendini kopyalayan geometriler geride kalmıştır.

On sekiz sürüm sonra yani Sketchpad'den 43 yıl sonra, AutoCAD 2010'da parametrik işlevsellik tanıtılmış ve basın bülteninde "çığır açan yeni bir yetenek" olarak lanse edilmiştir (Davis, 2013b).

Bu durum, parametrik tasarım kavramının hesaplamalı parametrikçilik ile girdiği kargaşayı ve uygulama üzerindeki anlamının bir tartışma konusu olduğunu göstermektedir.

AutoCAD 2010'un "çığır açan" yeni parametrik modelleme özellikleri aslında onlarca yıl önceki yazılımlarda mevcuttur. 1985 yılında eski matematik profesörü Samuel Geisbergten tarafından kurulan Parametric Technology Corporation 1988'de ticari olarak başarılı ilk parametrik yazılım olan Pro/ENGINEER'ı piyasaya sürmüştür. Sketchpad'de olduğu gibi, kullanıcılar çeşitli parametrik denklemler kullanarak Pro/ENGINEER'da bir geometrinin parçalarını bir araya getirebilmekteydiler. Sketchpad'den farklı olarak, geometri iki boyutlu değil üç boyutluydu ve değişiklikler birçok farklı kullanıcı tarafından oluşturulan birçok farklı çizim üzerinde yayılabilmekteydi (Davis, 2013b).

1993 yılında Industry Week ile yaptığı bir röportajda Geisberg, Pro/ENGINEER'ın orijinal motivasyonlarını kısa ve öz bir şekilde ifade ederek, parametrik tasarım yaklaşımını aktarmıştır:

“Amaç, mühendisi çeşitli tasarımları kolayca düşünmeye teşvik edecek kadar esnek bir sistem oluşturmaktır. Ve tasarımda değişiklik yapmanın maliyeti mümkün olduğunca sıfıra yakın olmalıdır. Fakat, günümüzdeki geleneksel CAD/CAM yazılımları, düşük maliyetli değişiklikleri mantıklı olmayan bir şekilde tasarım-mühendislik sürecinin yalnızca son noktasında sınırlıyor” (Teresko, 1993).

Geisberg iki noktaya değinmektedir. Birincisi, parametrik modellemenin tasarımcıların "çeşitli tasarımları" keşfetmelerini sağlaması gerektiğidir (Teresko, 1993). Bu, Pro/ENGINEER'da hem parametrelerin manipülasyonu hem de modelin temel ilişkilerinin manipülasyonu yoluyla mümkün olmaktadır. İkinci noktası, parametrik modellerin tasarım sürecinde, süreç tamamlansa bile çıktı verisinin dinamik bir şekilde güncellenmesini sağlayacak seçimler yapılmasına izin vermesidir.

1993 yılına gelindiğinde Dassault Systèmes, Pro/ENGINEER'in parametrik özelliklerini CATIA v4'e dahil etmiştir (Weisberg, 2008).

Gehry Partners'ın 2004 yılında piyasaya sürdüğü Digital Project yazılımı, CATIAv5'i temel alarak mimarlara, özellikle de karmaşık geometrilerle çalışan mimarlara yönelik özel araçlar sunan bir parametrik modelleme yazılımıdır. Digital Project'in büyük bir bölümü, mimarların geometrilerini tanımlayan parametreleri ve denklemleri revize etmelerine olanak tanıyan CATIAv5'in parametrik motoruna dayanmaktadır. Gehry Technology, bu çalışma temelinde 2001 yılında kurulmuştur ve Digital Project, mimarların rasyonelleştirmeye çalıştıkları karmaşık geometrileri daha etkili bir şekilde ele almalarına yardımcı olmaktadır (Davis, 2013b).

O dönemde karmaşık geometriler üretebilen sadece birkaç mimarlık firması, Dijital Project'i kullanmaya yönelik karmaşık projeler üretiyordu. Diğer büyük bir kısmı ise bilgisayarları, çizim setlerini hazırlamak ve koordine etmek amacıyla kullanmak yerine, AutoCAD ve benzeri yazılımlara bağlı kalmıştır. Bu dönemde, bazı mimarlar AutoCAD ve benzeri yazılımlara bağlı kalmayı tercih ederken, diğerleri Revit ve ArchiCAD gibi uzman yapı bilgi modelleme yazılımlarını benimsemeyi seçmiştir (Davis, 2013b).

Revit Technology Corporation, Parametric Technology Corporation geliştiricileri tarafından kurulmuş ve "mimarlar ve bina tasarım profesyonelleri için ilk parametrik bina modelleyicisini" oluşturmayı amaçlamıştır (Revit Technology Corporation, 2000).

Revit'in yazarları, parametriği, tasarımcının belirli koşullar için ayarlayabileceği parametrik denklemlere dayalı bir nesne olarak tanımlamıştır. İlerleyen sürümlerinde, tasarımcının çatının eğimini nasıl ayarlayabileceğini ve Revit'in "sırayla tüm planları,

yükseklikleri, kesitleri, programları, boyutları ve diğer unsurları nasıl 'revitleyeceğini' (anında revize edeceğini) açıklamışlardır (Revit Technology Corporation, 2001).

Revit ve benzer yazılımlar, otomatik revizyonlar için parametrik denklemleri kullanırken, tam anlamıyla parametrik modelleme yazılımları olan Pro/ENGINEER, CATIA ve Sketchpad'ın aksine, Revit'in parametrik ilişkileri arayüzün arkasına gizlenmiştir. Odak noktası, parametrik modelleri oluşturmak yerine kullanmaktır.

Revit, AutoDesk tarafından satın alındıktan sonra, parametrik modelleme etrafındaki retorik sona erdi ve tasarım markalarını belirtmek için Yapı Bilgi Modellemesi (BIM) adını icat ettiler (Weisberg, 2008).

Bunu yaparken, parametrik modelin kendisinin yönetiminin aksine, bilginin (parametrelerin) yönetimini vurgulayarak BIM'i parametrik modellemeden ayırmışlardır. Bu nedenle, mimarlık firmalarının çoğunluğu Digital Project veya Pro/ENGINEER gibi tam olarak parametrik yazılımları kullanmasalar da binalarını modellemek için belirli bir kapasitede parametrik denklemler kullanmışlardır.

Parametrik modelleme, yazılım paketlerinin komut dosyası arayüzleri aracılığıyla projelere entegre edilmiştir. Komut dosyası arabirimleri, tasarımcıların yazılımın bölümlerini otomatikleştirmek için kod yazmalarına olanak tanır.

Dino, komut dosyalarının doğası gereği parametrik olduğunu savunmakta ve "parametrik sistemlerin esas olarak algoritmik prensiplere dayandığını" belirtmektedir. Çünkü "bir algoritma, girdi olarak bir değer veya bir dizi değer alır, bu girdiyi dönüştüren bir dizi hesaplama adımını yürütür ve nihayetinde bir değer veya bir değer kümesi üretir." Bu nedenle, çoğu yazılım paketinde bulunan komut dosyası arabirimleri, doğuştan parametrik modeller oluşturmak için uygun bir yapıya sahiptir (Dino, 2012).

Metinsel komut dosyası arayüzleri, AutoCAD'in ilk günlerinden bu yana önemli ölçüde gelişim kaydedememiştir. Bunun yerine, bir tür komut dosyası arayüzü olan görsel kodlamanın ortaya çıktığı gözlemlenmiştir. Görsel kodlama, programları metin olarak değil, diyagramlar olarak temsil etmeyi içerir.

1990'lardan itibaren iki dikkate değer öncü şunlardır: müzisyenler arasında popüler olan MAX/MSP ve görsel efekt sanatçıları arasında popüler olan Sage (ismi daha sonra Houdini olarak değiştirilmiştir).

Mimarlar, Robert Aish'in Bentley Systems için çalıştığı dönemde Generative Components'i 2003 yılında seçili mimarlık firmalarında ilk aşamalarını test etmeye başladığında ilk görsel programlama dilini elde etmişlerdir. Robert McNeel & Associates, Generative Components'i lisanslamak için başarısız bir girişimde bulunduktan sonra

geliştirici David Rutten'e kendi versiyonunu yapma görevini verdi Rutten'in 2007'de yayınladığı Explicit History, daha sonra görsel programlama ara yüzüne Grasshopper adını verdi. Hem Grasshopper hem de Generative Components, genellikle geometrinin üretilmesiyle sonuçlanan kullanıcı tanımlı işlevler aracılığıyla parametrelerden akışı haritalayan grafiklere dayanmaktadır. Parametrelerin veya modelin ilişkilerdeki değişiklikler, geometriyi otomatik olarak yeniden çizmek için açık işlevler aracılığıyla değişikliklerin uygulanmasını sağlar. Bu nedenle, bunlar parametrik bir model oluşturma başka bir yoludur (Davis, 2013b).

Son yıllarda parametrik modelleme, Gaudi, Otto, Sutherland ve bazı mühendisler tarafından kullanılan matematiksel bir yöntem olmaktan çıkıp, mimarlık pratiğinin düzenli bir parçası haline gelmiştir. Matematikte parametrik, bir dizi bağımsız parametrenin açık bir fonksiyonu olarak ifade edilen bir dizi niceliği ifade ederken, mimaride parametrik terimine genellikle modelin olanaklarını keşfetmek için faydacı bir ihtiyaç eşlik eder. Bu keşif hem model parametrelerinin değiştirilmesi hem de model ilişkilerinin değiştirilmesi yoluyla kolaylaştırılmaktadır.

Parametrik modelleme artık CATIA ve Pro/ENGINEER gibi açıkça parametrik araçların münhasır alanı değildir. Bunun yerine, parametrik denklemler birçok BIM aracını arka planda yönlendirir, metinsel komut dosyası dillerinde tezahür ederler ve grafik tabanlı görsel komut dosyası arayüzleri tarafından ortaya çıkarılırlar. Parametrik modelleme, çoğu çağdaş mimari projede bir şekilde mevcuttur. Parametrik modellemenin uygulanmasındaki bu hızlı genişleme, anlaşılır bir şekilde anlamı üzerinde bazı karışıklıklara yol açmıştır.

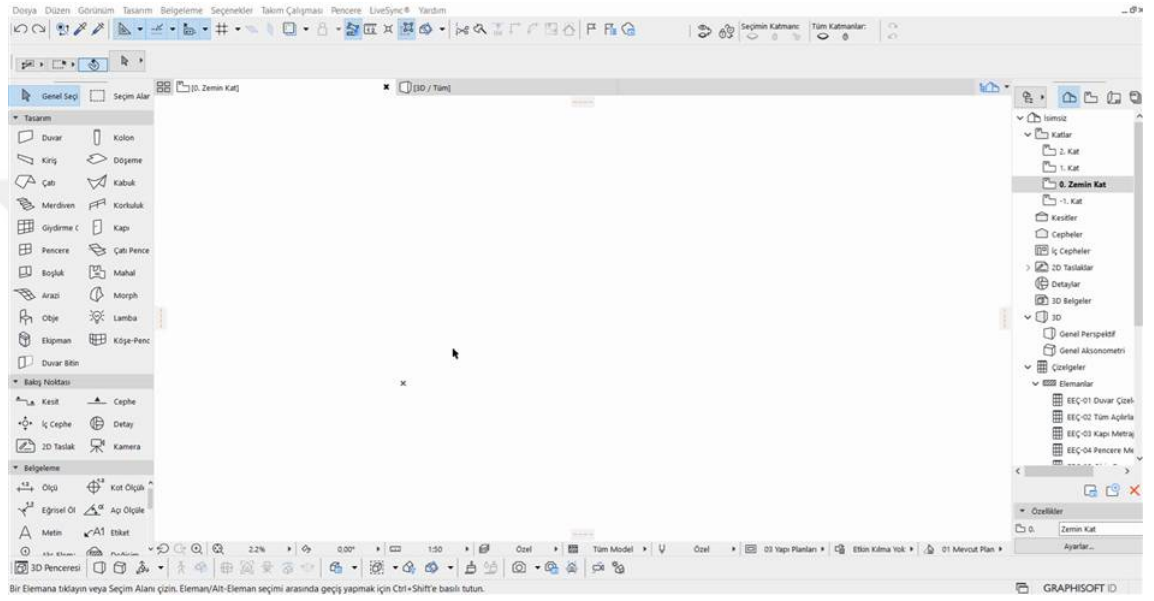
Günümüzde mimarlıkta yaygın olarak kullanılan parametrik modelleme eklentileri arasında Rhinoceros için Rhinoscript ve Grasshopper, 3DSMax için MaxScript ve Para3D, Autodesk Revit için Dynamo öne çıkar. MaxScript, yazısal kodlama arayüzüne ve diline sahiptir, Dynamo ve Grasshopper ise görsel kodlama dilini kullanır. Görsel kodlama dili, tasarımcılar gibi görsel algısı yüksek bireyler tarafından daha kolay kurulabilir, ancak gerçekleştirmek için daha fazla CPU (işlemci) ve GPU (grafik işlemci) gücü gerektirir. Bu nedenle, MaxScript kullanılarak daha düşük donanımlı bilgisayarlarda bile karmaşık modellemeler gerçekleştirilebilir (Eltaweel ve Su, 2017).

Bu tezde, yukarıda bahsedilen eklentiler arasında en yaygın olarak bilinen ve birçok farklı alanda kullanılan Grasshopper ve yapı bilgi modelleme yazılımı olan ArchiCAD kullanılmıştır.

2.1.3. Bir Model Verisi Üzerinden Geleneksel Tasarım Araçları ve Parametrik Tasarım Araçlarının Karşılaştırılması

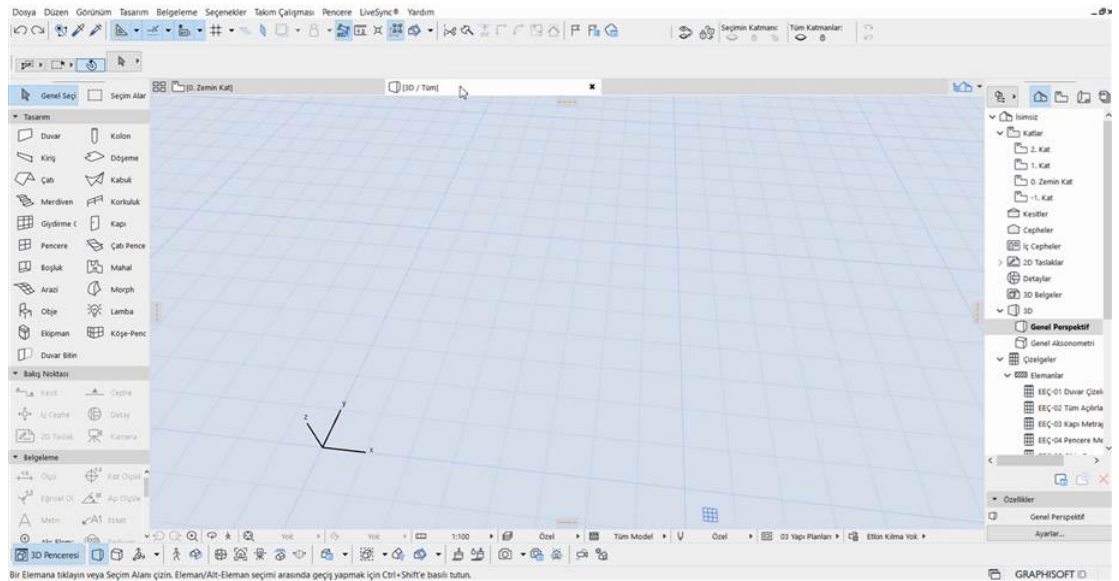
Tez kapsamında parametrik modellemenin fayda yönünü vurgulamak için basit bir masa modeli, bir algoritma oluşturma ekseninde önce bir BIM yazılımı olan ArchiCAD'deki morph aracı kullanılarak geleneksel yöntemler ile modellenecek daha sonra aynı masa, Grasshopper ile parametrik olarak modellenecektir.

1. Adım: ArchiCAD'de yeni bir proje dosyası aç (Şekil 2.8).



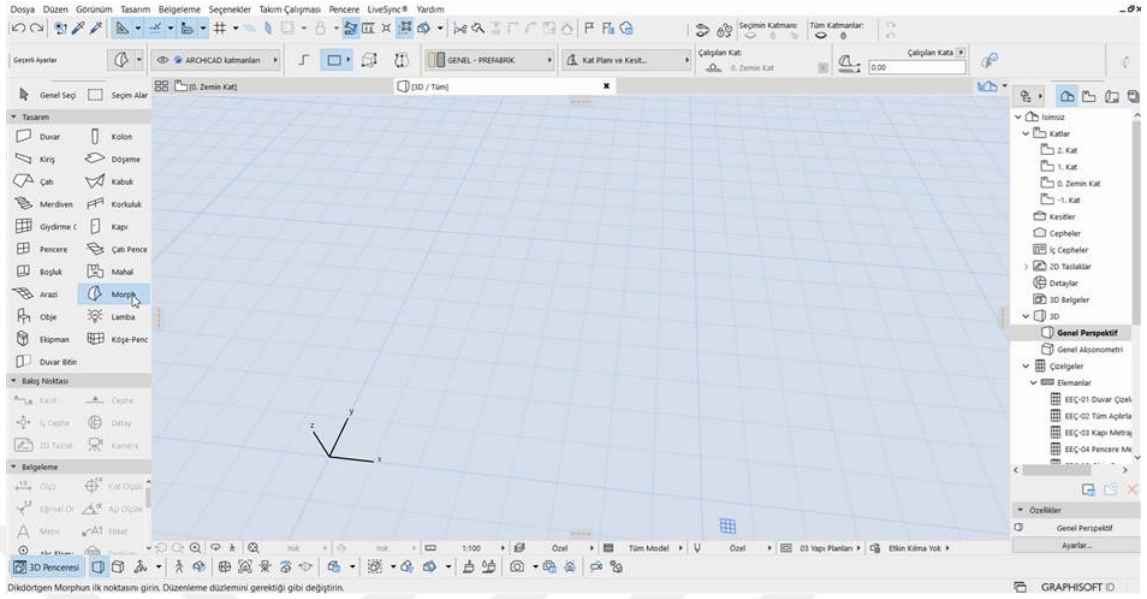
Şekil 2.8. ArchiCAD algoritma ekseninde masa modelleme 1. Adım

2. Adım: Fare imlecini 3D sekmesine sürükleyin ve farenin sol tuşuna basın (Şekil 2.9).



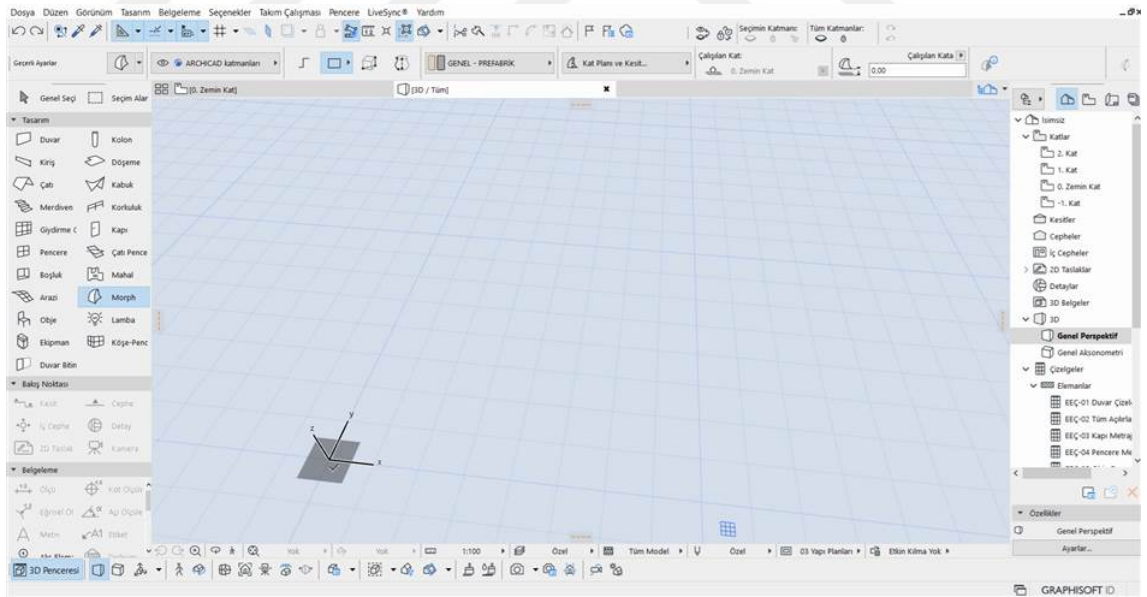
Şekil 2.9. ArchiCAD algoritma ekseninde masa modelleme 2. Adım

3. Adım: Fare imlecini Morph aracına sürükleyin ve farenin sol tuşuna bas (Şekil 2.10).



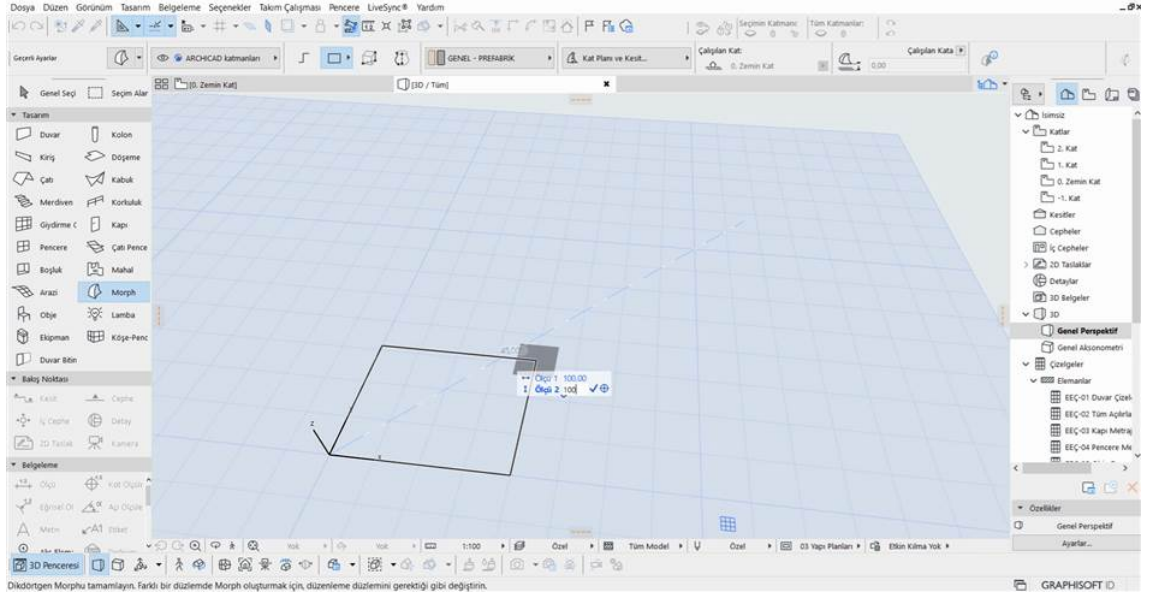
Şekil 2.10. ArchiCAD algoritma ekseninde masa modelleme 3. Adım

4. Adım: Fare imlecini 3D ekranında x, y ve z koordinatlarının 0 olduğu noktaya sürükleyin ve farenin sol tuşuna basarak morph aracını çalıştırın (Şekil 2.11).



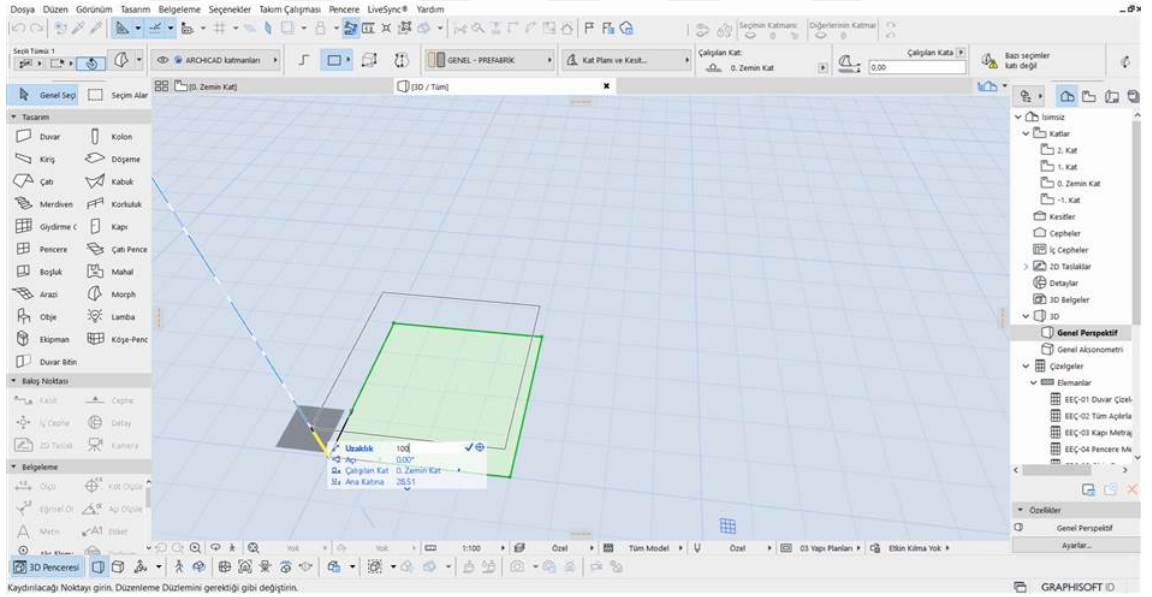
Şekil 2.11. ArchiCAD algoritma ekseninde masa modelleme 4. Adım

5. Adım: Morph aracını çalıştırdıktan sonra fare imlecini XY eksenini doğrultusunda hareket ettir, klavyede 100x100 değerlerini yaz ve ENTER tuşuna bas (Şekil 2.12).



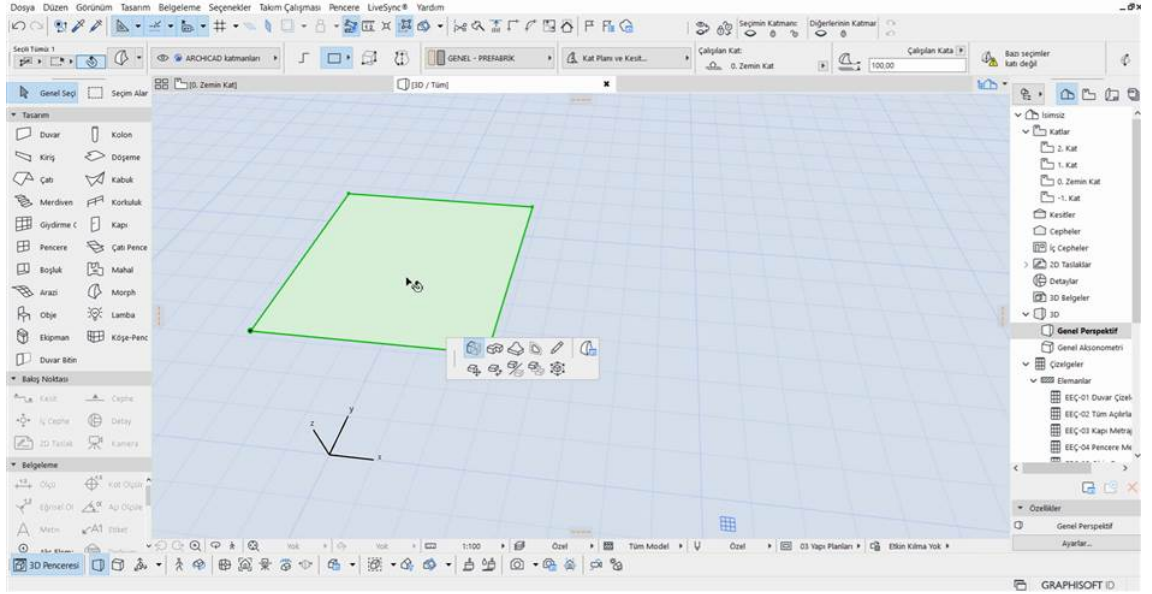
Şekil 2.12. ArchiCAD algoritma ekseninde masa modelleme 5. Adım

6. Adım: Fare imlecini 5. adımda oluşturulan yüzeyin köşe noktasına getir, farenin sol tuşuna bas, fare imlecini z eksenini doğrultusunda pozitif yönde hareket ettir, klavyede 100 değerini yaz ve ENTER tuşuna bas (Şekil 2.13).



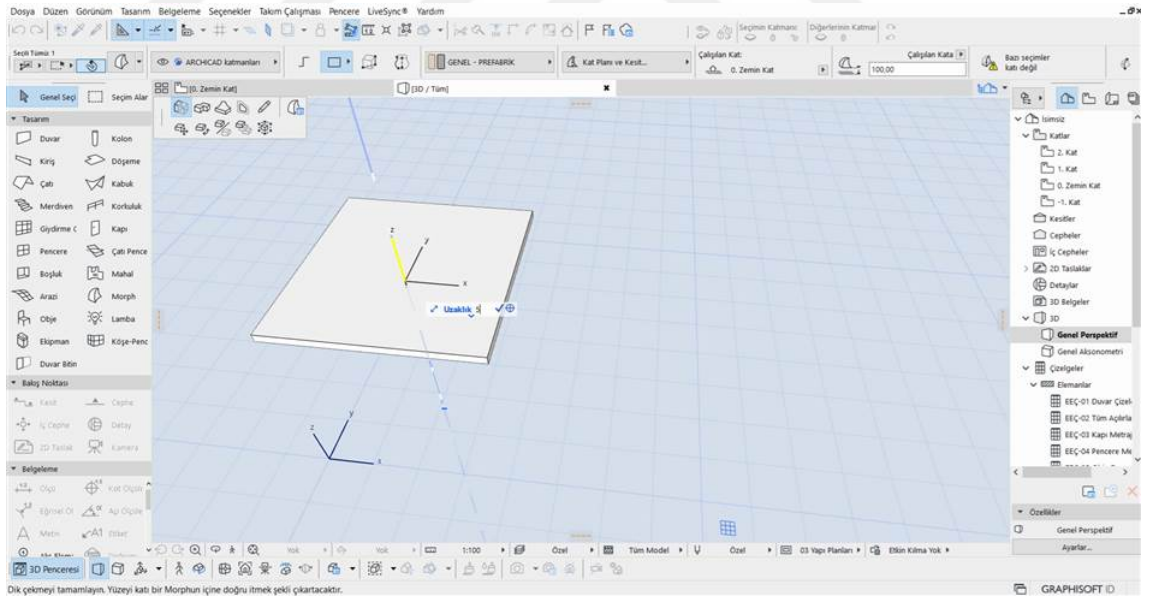
Şekil 2.13. ArchiCAD algoritma ekseninde masa modelleme 6. Adım

7. Adım: Fare imlecini 100 cm “yukarı” ötelenen yüzeyin herhangi bir noktasına sürükleyin, farenin sol tuşuna bas ve it/çek komutunu aktifleştir (Şekil 2.14).



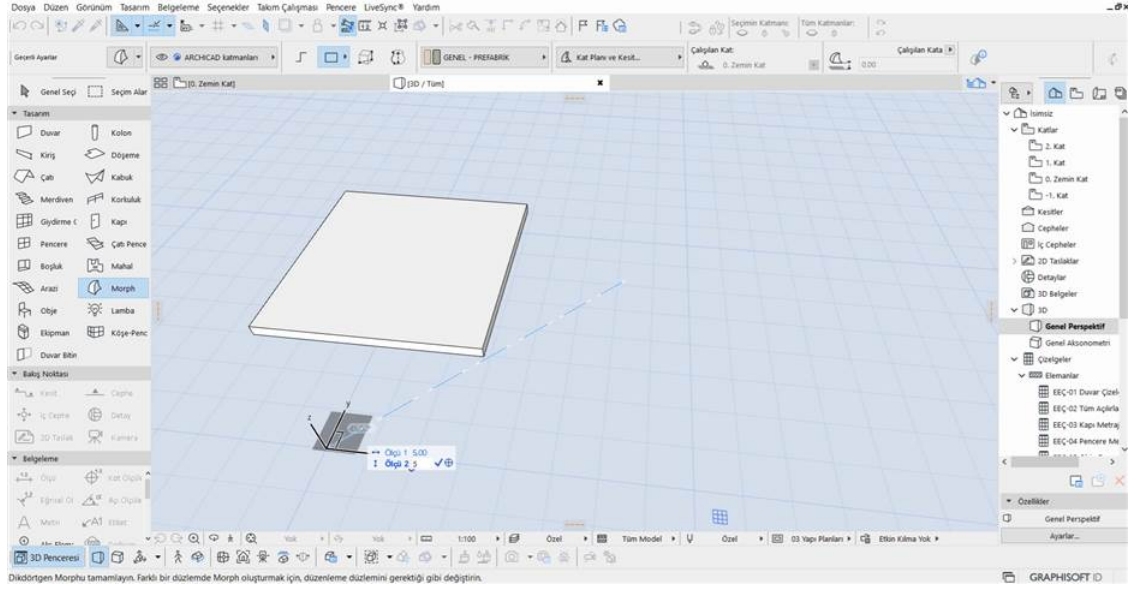
Şekil 2.14. ArchiCAD algoritma ekseninde masa modelleme 7. Adım

8. Adım: 7. Adımda aktifleştirilen it/çek komutu çalışırken fare imlecini z eksenı doğrultusunda negatif yönde hareket ettir, klavyede 5 değeri yaz ve ENTER'a bas (Şekil 2.15).



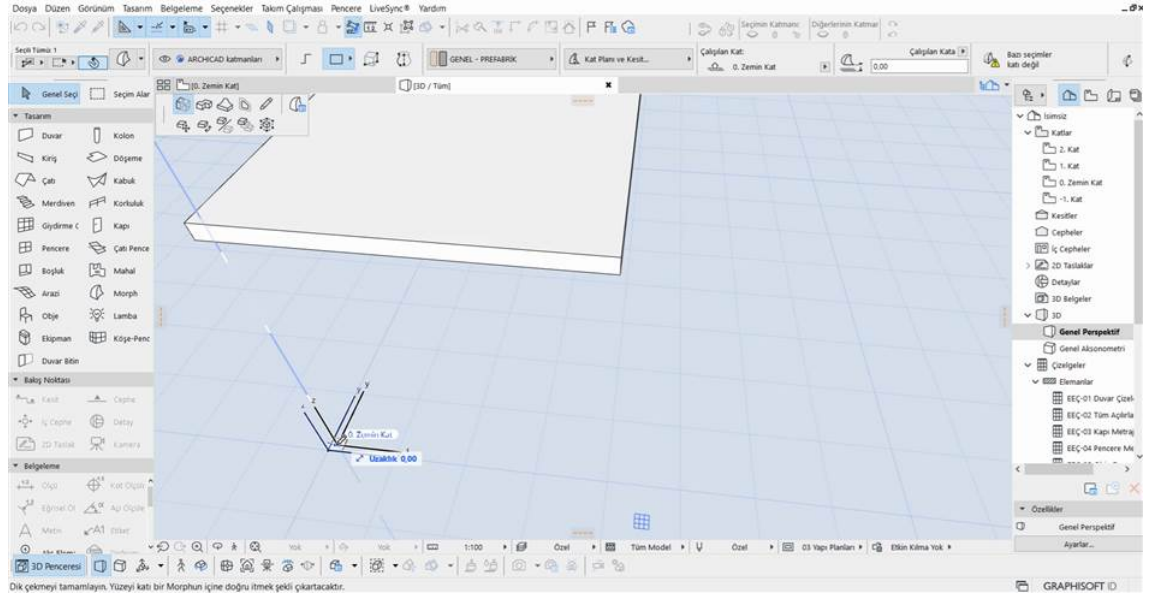
Şekil 2.15. ArchiCAD algoritma ekseninde masa modelleme 8. Adım

9. Adım: 8. adımda oluşturulan masanın tablasına masanın ayaklarını eklemek için morph aracını seç, fare imlecini 3D ekranında x, y ve z koordinatlarının 0 olduğu noktaya getir, morph aracını farenin sol tuşuna basarak aktifleştir, fare imlecini XY doğrultusunda hareket ettir, klavyede 5x5 değerlerini yaz ve ENTER tuşuna bas (Şekil 2.16).



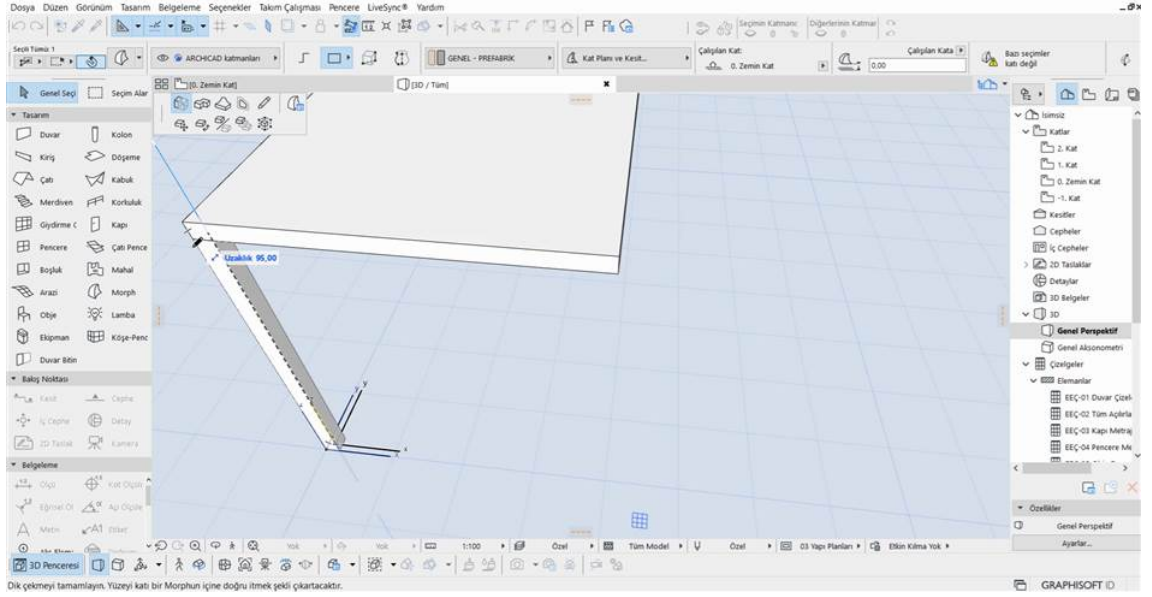
Şekil 2.16. ArchiCAD algoritma ekseninde masa modelleme 9. Adım

10. Adım: 9. adımda oluşturulan ve masanın ayağının enkesiti olan yüzeyin herhangi bir noktasına fare imlecini hareket ettir, fare imleci yüzey üzerindeyken farenin sol tuşuna bas ve it/çek komutunu aktifleştir (Şekil 2.17).



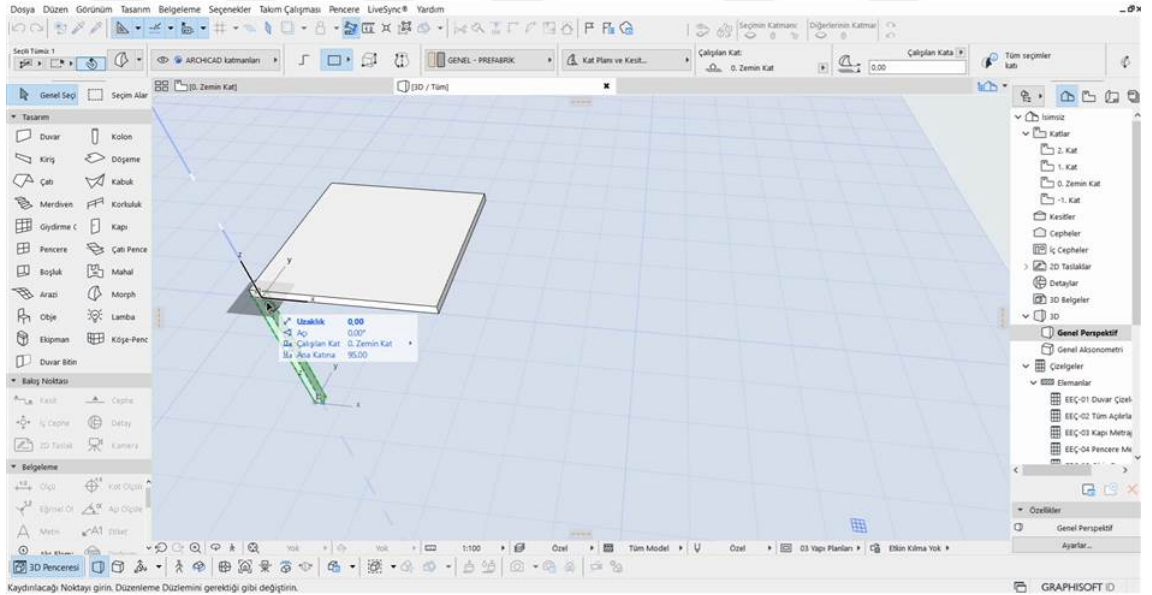
Şekil 2.17. ArchiCAD algoritma ekseninde masa modelleme 10. Adım

11. Adım: 10. adımda aktifleştirilen it/çek komutu çalışırken fare imlecini Z eksenini doğrultusunda pozitif yönde hareket ettir, masa tablasının sol alt noktasına sürükleyin ve fare imleci sol alt noktada siyaha dönüştüğünde farenin sol tuşuna bas (Şekil 2.18).



Şekil 2.18. ArchiCAD algoritma ekseninde masa modelleme 11. Adım

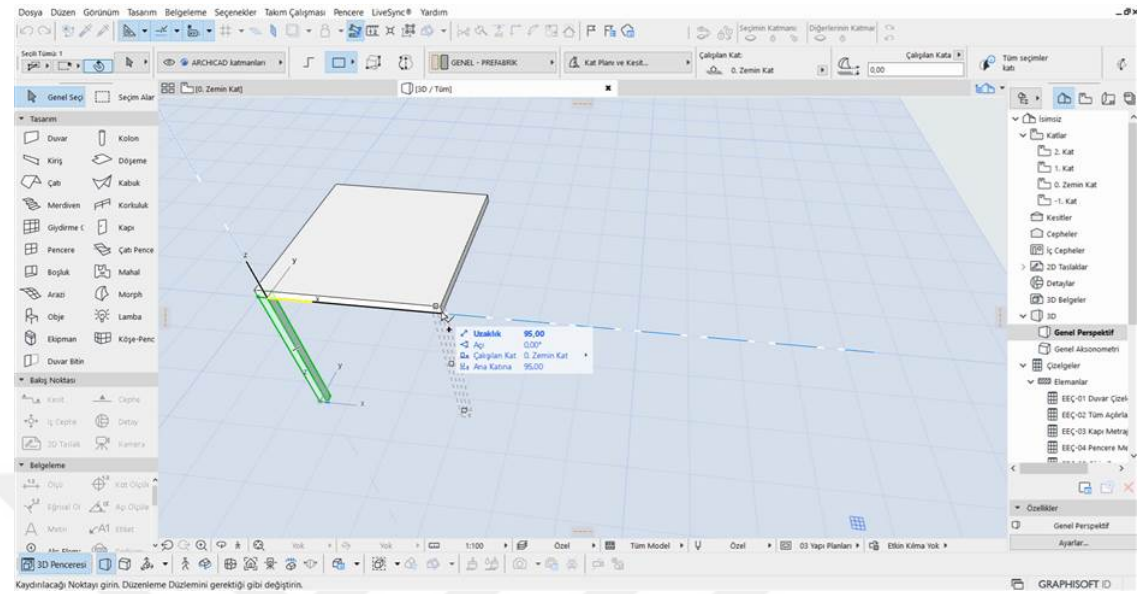
12. Adım: 11. Adımda oluşturulan masanın bir köşesindeki tek bir ayağını masanın 4 köşesine de kopyalamak için fare imlecini masa ayağının herhangi bir noktasına getir, masa ayağını seç ve klavyede Ctrl ve D tuşlarına aynı anda bas. Bu objeleri hareket ettirmeyi sağlayan ArchiCAD komutudur (Şekil 2.19).



Şekil 2.19. ArchiCAD algoritma ekseninde masa modelleme 12. Adım

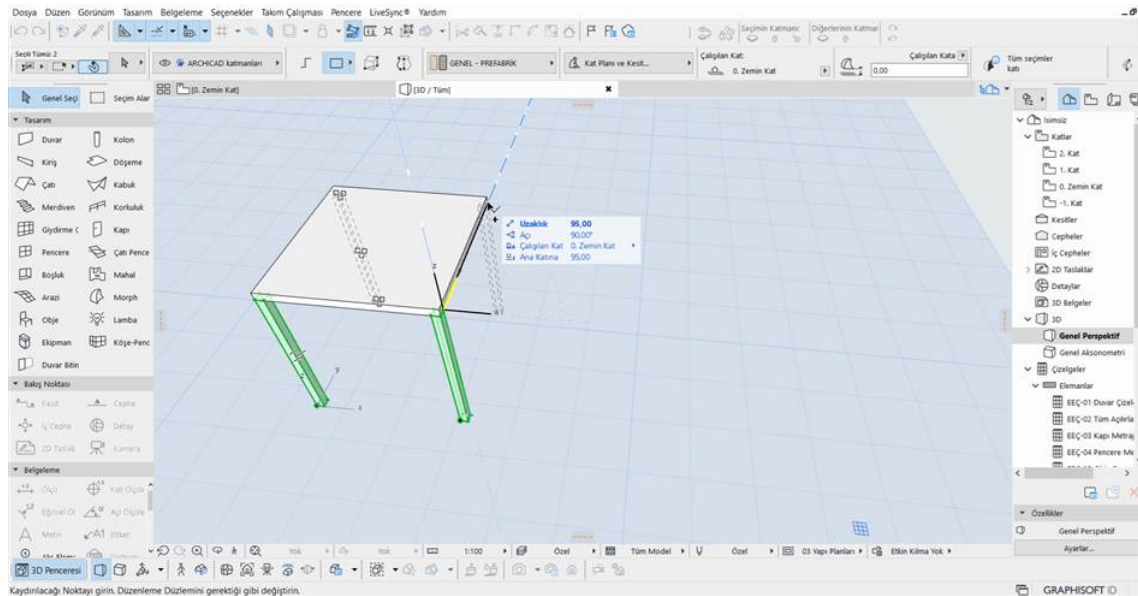
13. Adım: 12. Adımda aktifleştirilen taşıma komutu çalışırken fare imlecini x eksenini doğrultusunda yatay yönde hareket ettir, klavyede ctrl tuşuna bas, fare imlecini masa tablasının sağ alt noktasına sürükleyin ve farenin sol tuşuna bas. (Ctrl ArchiCAD’de

taşıma komutu çalışırken elemanları kopyalamayı sağlayan bir kısayol komutudur.) (Şekil 2.20).



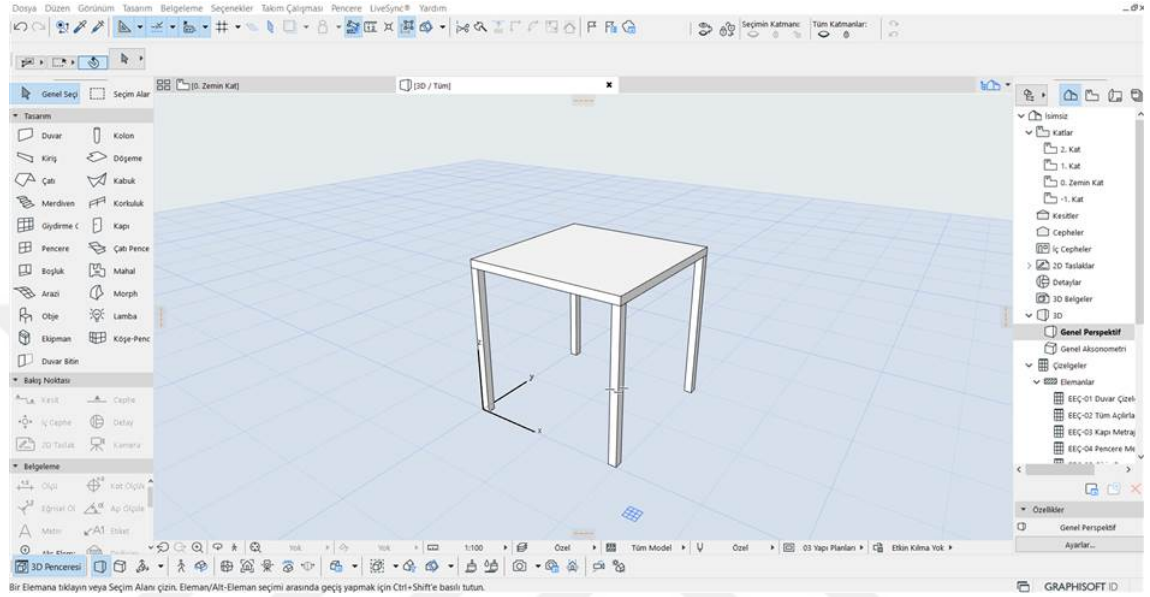
Şekil 2.20. ArchiCAD algoritma ekseninde masa modelleme 13. Adım

14. Adım: 13. adımda elde edilen masanın ayağını kopyalama adımından sonra masanın 4 ayağını da modellemek için 2 ayak seçiliyken (yeşil renkte kaplanması seçili olduğunu gösterir) klavyede Ctrl ve D tuşlarına aynı anda bas, fare imlecini masa tablasının sağ üst noktasına sürükley, taşıma komutu aktifken klavyede Ctrl tuşuna bas ve farenin sol tuşuna bas (Şekil 2.21).



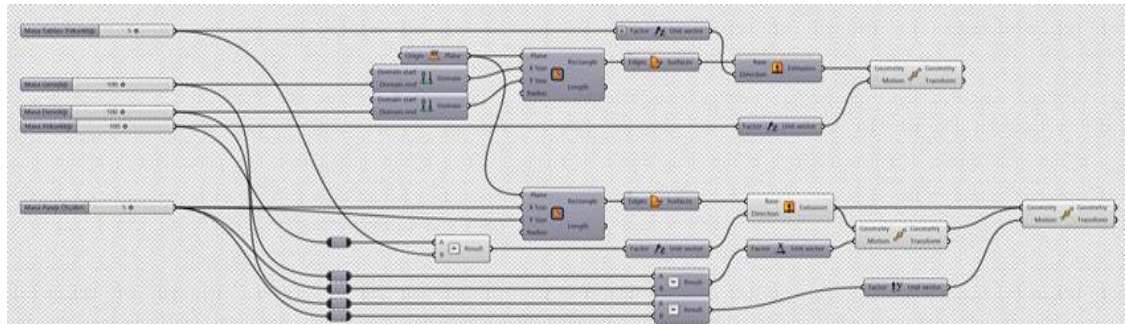
Şekil 2.21. ArchiCAD algoritma ekseninde masa modelleme 14. Adım

14 Adımdan oluşan bu algoritma sonucunda 100x100x100 ölçülerinde bir masa bir BIM aracı olan ArchiCAD yazılımında geleneksel modelleme yöntemleriyle modellenmiştir. Görüldüğü üzere geleneksel modelleme yöntemleriyle bir objeyi modellemek için işlenen süreç bir algoritma boyutunda tanımlanabilmektedir (Şekil 2.22).



Şekil 2.22. ArchiCAD algoritma ekseninde modellenen masa

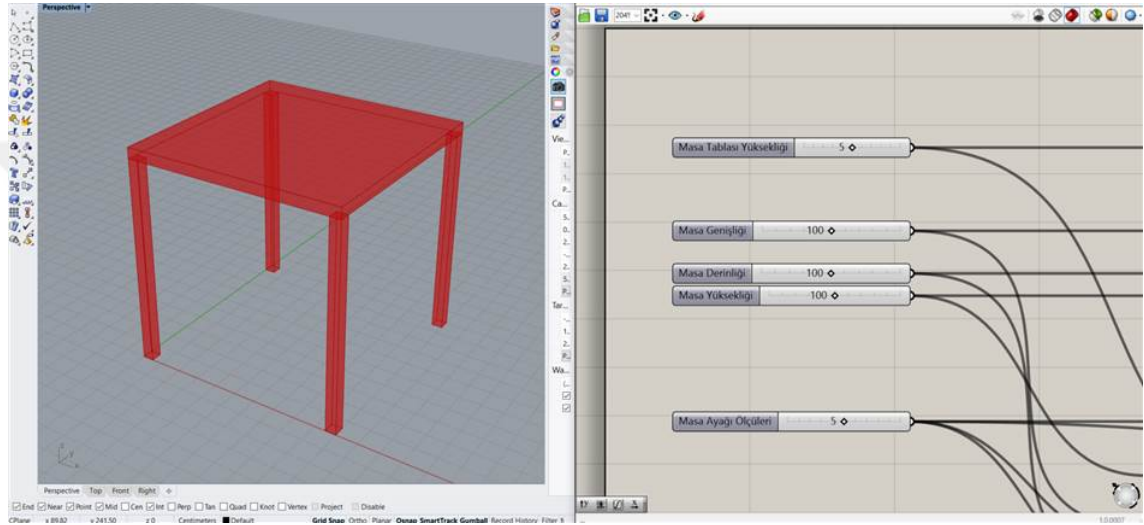
Aynı masayı Grasshopperda parametrik modellemek için aynı yollar izlendiğinde oluşturulan algoritma Şekil 2.23'te gösterilmektedir.



Şekil 2.23. Masa'nın Grasshopper ile oluşturulan algoritması

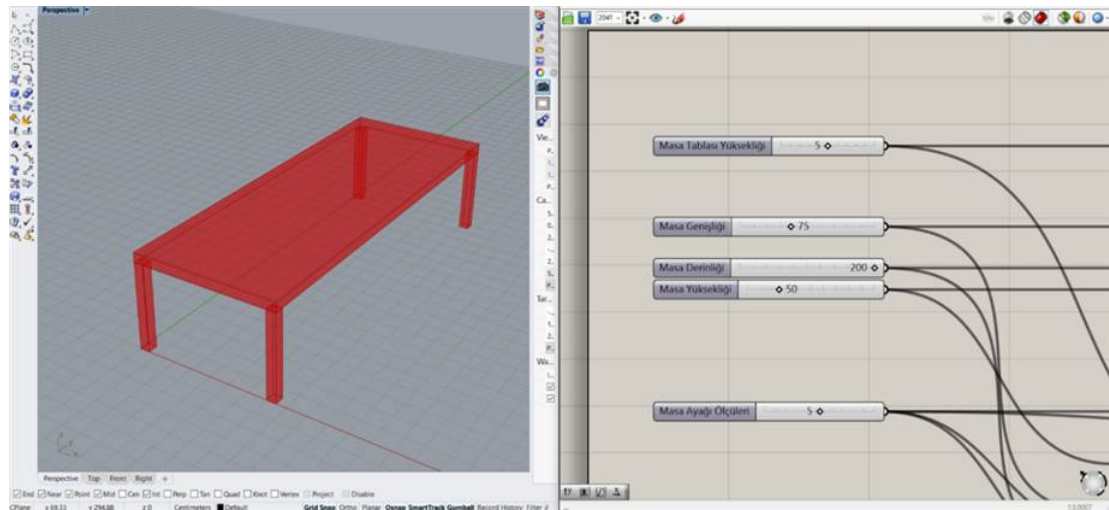
Grasshopperda parametrik olarak modellenen bir model üzerinde geriye dönük herhangi bir işlem yapılmak istendiğinde süreci başa almaya gerek kalmadan sadece parametreleri değiştirerek çıktıyı dinamik bir şekilde güncellemek mümkündür.

Algoritmanın giriş parametrelerine bakıldığında masa tablası yüksekliği, masa genişliği, masa derinliği, masa yüksekliği ve masa ayağı ölçüleri gibi değerleri görünmektedir (Şekil 2.23). Bu değerler algoritmanın tüm adımları geriye yönelik olarak ilişkilerini koruduğu için masa üzerinde derinlik ve genişlik gibi değerler değiştirilmek istendiğinde algoritmanın çıktısı dinamik olarak güncellenmektedir.



Şekil 2.24. Masa'nın Grasshopper'daki parametreleri

Örneğin masa genişliği parametresine 75, masa derinliği parametresine 200 ve masa yüksekliği parametresine 50 değeri verildiğinde dinamik bir şekilde yeniden oluşturulmuş 75x200x50 ölçülerinde bir masanın model verisi elde edilir (Şekil 2.23).



Şekil 2.25. Masa'nın Grasshopper'daki değiştirilmiş parametreleri

2.1.4. Bir Mimarlık Akımı Olarak Parametrik Tasarım (Parametrisizm)

Tez kapsamında parametrik tasarım kavramı önce etimolojik olarak incelenmiş daha sonra anlam bütünlüğüne göre mimarlıkta karşılığı irdelenmiştir. Kavramın mimarlıktaki derin temellere dayanan karşılığından bir tasarım sistematığına dönüşen süreci incelenmiştir.

Süreçlerin incelenmesi sonucunda kavramın temellerinin 9. yüzyılda matematikte keşfinden ve 19. yüzyılda mimarlıkta kullanımına, daha sonrasında ise 20. yüzyılın sonlarına doğru bilgisayarın sürece dahil olması ile birlikte anlamının nasıl geliştiği, değiştiği ve bazı noktalarda bir karmaşa içerisine girdiği gözlemlenmiştir. Günümüze gelindiğinde ise parametrik tasarım kavramı, “Parametrisizm” olarak adlandırılan bir mimarlık akımında, geçirdiği dönüşümlere göre aldığı son hali ile somutlaştırılmıştır.

Parametrisizm, mimarlıkta modernizmden sonraki en önemli akım olarak kabul edilmektedir. Bu akımın temel fikirleri, 2008 yılında Venedik Bienali’nde Zaha Hadid Mimarlık’ın ortaklarından Patrik Schumacher tarafından bir manifestoda sunulmuştur. Schumacher manifestosunda, parametrisizmin özgün özelliklerini, tasarım ilkelerini, uygulama kurallarını ve geliştirme potansiyelini ortaya koymuştur.

Parametrisizm, Schumacher’e göre parametrik tasarım tekniklerine dayanan bir tasarım yaklaşımıdır. Parametrik tasarım, mimarının her ölçeğinde ve her alanında etkili bir biçimde kullanılmaktadır. Kentsel tasarımdan mobilya tasarımına kadar, parametrik tasarımın izleri görülebilmektedir (Schumacher, 2008).

Parametrisizm, fordizmin standartlaştırdığı üretim ve tüketim biçimlerinin yerini alan post-fordist toplumun karmaşıklığını ve heterojenliğini yansıtmaktadır. Dijital çağın getirdiği yenilikler, üretim ve tüketim süreçlerini değiştirmiş, toplum ve mimarlık üzerinde yeni etkiler yaratmıştır (Schumacher, 2008).

Parametrisizm, avangart mimarlığın bir araştırma programı olarak tanımlanmaktadır. Schumacher, avangart akımları, yeni bir teori geliştirmeye çalışan bilimsel araştırmalarla karşılaştırmaktadır. Bilimsel araştırmalarda olduğu gibi, avangart akımlarda da kavramsal çerçeve, amaçlar ve yöntemler yeniden tanımlanmaktadır. Schumacher (2008), bilimsel teorilerle mimari akımlar arasındaki benzerlikleri vurgulamak için Lakatos’un araştırma programı kavramını kullanmaktadır. Lakatos’a göre, Newton’un yerçekimi teorisi, Einstein’ın görecelik teorisi, kuantum teorisi, Marxizm vb. gibi bilimsel fikirler, araştırma programlarıdır. Bu fikirler, her zaman çözülemeyen ya da anlaşılamayan sorunlarla karşılaşmış ve çürütülmüş olarak doğmuş

ve ölmüşlerdir. Bu durum, mimari akımlar için de geçerlidir. Her akım, kendine has prensipler ve karakteristik özellikler belirlemektedir (Schumacher, 2008).

Parametrisizm, postmodernizm ve dekonstrüktivizm gibi modernizm sonrası akımlardan farklı olarak, modernizmin ardından gelen en önemli akım olarak nitelendirilmektedir. Postmodernizm ve dekonstrüktivizm, modernizmin eleştirisi ve yenilenmesi sürecine katkıda bulunan geçiş dönemleri olarak görülmektedir (Schumacher, 2008).

Parametrisizm, özel parametrik tekniklerle gerçekleştirilebilen bir akımdır. Günümüzde, MaxScript, RhinoScript gibi kompüstasyonel tasarım teknikleri ve parametrik modelleme teknikleri yaygın olarak kullanılmaktadır. Bu teknikler, günümüz avangart mimarlığının rekabetçi ortamında önemli bir avantaj sağlamaktadır (Schumacher, 2008).

Parametrisizm, Modernizm gibi devrimci bir stil olarak kabul edilmektedir. Bu yüzden Parametrisizm'in Modernizm'den farklılaşan bazı yönleri vardır. Modernizm mekân kavramına dayanmaktadır. Parametrisizm ise "alanlar"ı ayırt etmektedir. Alanlar, ışın dalgaları, katmanlı akışlar ve girdaplar gibi akışkan bir ortam içinde tasarlanmaktadır. Büyük bir sergi salonu veya fuar alanı gibi bir mekân düşünüldüğünde, bu mekânın derinliği, iç mekân donatıları ve insan sürüsü (swarm) ile birlikte, keskin dış çizgilerden uzak bir biçim oluşturmaktadır. Bu durumda platonik biçimlerden bahsetmek mümkün değildir (Schumacher, 2008).

Schumacher, parametrisizmin beş temel özelliğini tanımlamakta ve bunların parametrik tasarıma yeni bir vizyon kazandıracağına inanmaktadır. Bu özellikler şunlardır: alt sistemlerin eklemlenmesi, parametrik duyarlılık, parametrik biçimlendirme, parametrik vurgulama ve parametrik şehircilik (Schumacher, 2008).

Parametrik duyarlılık, kentsel ve mimari ortamların yaygın kullanıma göre kendilerini yeniden tanımlayabilecekleri şekilde tasarlanabileceğini belirtmektedir. Bu bağlamda tasarım, zaman kavramının da dahil olduğu 4 boyutlu olarak gerçekleştirilmektedir. Sistem parametrik olarak modellenerek, farklı zamanlardaki verilere göre tasarım değiştirilebilmektedir (Schumacher, 2008).

Parametrik biçimlendirme, farklı şekillerde algılanabilecek karmaşık biçimlerin parametrik olarak modellenebileceğini anlatmaktadır. Parametrik model, değişkenlerine bağlı olarak Gestalt ilkelerini iyi bir şekilde yansıtabilmektedir. Bununla birlikte, bu parametrelerin sayısal olarak değişmesi sonucunda, algılanan biçimin niteliği değişerek bir "Gestalt yıkımına" neden olabilmektedir. Bu nedenle, algının tam olarak

oluşturulabilmesi için, modellenen nesnenin yanında, ortamın parametreleri ve gözlemci parametreleri de parametrik model içinde bir bütün olarak kurgulanmalıdır.

Parametrik vurgulama, bir biçimi dışarıdan bazı biçimler ekleyerek çoğaltmak yerine, biçimi parametrik olarak yeniden kurgulayarak, dışarıdan müdahale gerektirmeden değiştirerek, parametrik vurgusunu artırmayı hedeflemektedir.

Parametrik şehircilik, kentsel kütlelerin sürü (swarm) düzeni ile tanımlandığını ileri sürmektedir. Parametrik şehircilik sayesinde, yapıların sistematik bir modüler düzen içinde birleştirilebileceği iddia edilmektedir. Böylece daha güçlü kentsel etkilerin elde edilmesi amaçlanmaktadır. Parametrik şehircilik, parametrik vurgulama, parametrik biçimlendirme ve parametrik duyarlılık ile birlikte düşünülebilmektedir (Schumacher, 2008).

Çağdaş mimarlıkta deformasyon, düzenin bozulması anlamına gelmemektedir. Yeni düşünceye göre, dinamizm kurgusu öne çıkmaktadır. Bu dinamizm kurgusu içinde, mekanların yanı sıra, mobilyalar ve endüstriyel tasarım ürünleri de önemli bir rol oynamaktadır. Mobilyalar, mekândan bağımsız objeler değil, mekanları oluşturan objelerdir. Tasarıma dahil edilen tüm unsurlar, bir bütün içinde tasarlanabilmektedir. İç mekân donatıları, bazen dinamik bir sürü mantığıyla, bazen de bir yüzey oluşturacak şekilde tasarlanarak, “alanların farklılaştırılması” durumu ayrıntılandırılabilir.

Alt sistemlerin eklenmesi ile, tekil sistemlerin çoğul sistemlere dönüştüğü ifade edilmektedir. Alt sistemler kodlar aracılığıyla birleştirilerek, bir sistemde yapılan değişikliğin, tüm sistemi etkilemesi sağlanmaktadır (Schumacher, 2008).

Parametrisizm Manifestosu, parametrik tasarımın ürünleri olan mimari tasarımların değerlendirilmesi için bir çerçeve sunmaktadır. Schumacher, parametrisizm düşüncesinin özelliklerini belirlemek için uygulanması ve kaçınılması gereken kurallar ortaya koymuştur. Schumacher, bu kuralları üç farklı zamanda ifade etmiştir. Bunlardan ilki, 2008 yılında Venedik Bienali’nde “Parametricism as Style- Parametrisizm Manifestosu” nu ilk kez duyurduğu zamandır. Daha sonra, 2009 yılında “Parametricism: A New Global Style for Architecture and Urban Design” başlıklı yazısında; son olarak da 2011 yılında yayınladığı “Autopoiesis of Architecture” adlı kitabında parametrisizm kurallarını ele almıştır. Schumacher, uygulanması gereken kurallara kabuller (dogmalar), kaçınılması gereken kurallara ise redler (tabular) demiştir. Bu kurallar sayesinde, parametrisizm düşüncesi sistematik bir şekilde tanımlanarak, tasarımların bu bağlamda nesnel olarak değerlendirilmesi sağlanmıştır. Schumacher, kabuller ve redler hakkında “spot ifadeler” kullanarak açıklama yapmış, ancak bunların anlamı üzerinde yorum yapmamıştır.

Schumacher, parametrisizm ilkelerini üç farklı yazıda ortaya koymuştur. İlk yazısı 2008 yılında “Parametricism as Style- Parametrisizm Manifestosu” adıyla yayınlanmıştır. Bu yazıda, parametrisizmin kabulleri (dogmaları) şöyle sıralanmıştır: kod kullanımı, spline (eğrisel çizgi) kullanımı, NURBS kullanımı, aynı birimin değişerek tekrarlanması, deforme etme, yerinden koparma, biçim değiştirme, melezleştirme ve eklemlenme (Schumacher, 2008).

Schumacher’ in ikinci yazısı 2009 yılında “Parametricism: A New Global Style for Architecture and Urban Design” başlığıyla yayınlanmıştır. Bu yazıda, parametrisizmin kabulleri (dogmaları) şunlardır: tüm formların parametrik olarak işlenmiş olması, çeşitli oranlarda ve dereceli olarak farklılaştırma, sistematik olarak kıvrırmak (bükme) ve ilişkilendirmek (Schumacher, 2009).

Schumacher’ in son yazısı ise 2011 yılında “Autopoiesis of Architecture” isimli kitabında yer almıştır. Bu yazıda, parametrisizmin kabulleri (dogmaları) şu şekilde ifade edilmiştir: tüm formların parametrik olarak işlenmiş olması, tüm sistemlerin belli kurallar çerçevesinde farklılaşması ve tüm sistemlerin birbiriyle ilişkili olması (Schumacher, 2011).

Schumacher “Parametricism as Style- Parametrisizm Manifestosu” nda redler (tabular) için; sıradan tipolojiler, platonik (katı) şekiller, net hatlar, aynı ögenin yinelenmesi, düz çizgiler, dik açılar ve köşeler, bağlantısızlık kriterlerini ortaya koymuştur. 2009 yılında yayımlanan “Parametricism: A New Global Style for Architecture and Urban Design” adlı makalesinde ise redler; kare, üçgen, daire gibi katı geometrik şekillerden, öğelerin basit yinelenmesinden, bağlantısız öge veya sistemlerin bitişik gelişmesinden uzak durmak şeklinde tanımlanmıştır. 2011 yılında çıkan “Autopoiesis of Architecture” adlı eserinde de redler (tabular) olarak; katı şekillerden, alt öğelerin basit yinelenmesinden, bağlantısız tekil öğelerden, bağlantısız alt öğelerden sakınmak gerektiği belirtilmiştir.

3. 3 BOYUTLU BASKI (EKLEMELİ İMALAT) YAPABİLEN ROBOT KOLLAR

3.1. Dijital Fabrikasyon ve Eklemeli İmalat

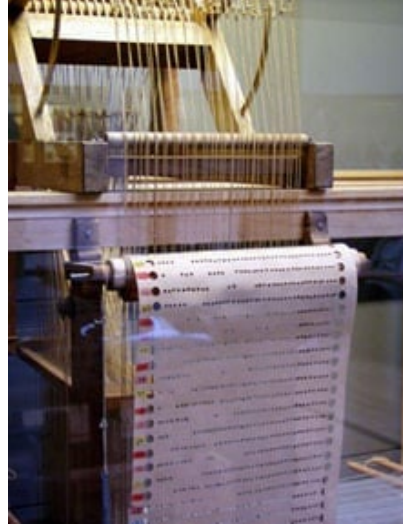
İmalat kelimesi, Latince kökenli "manus" (el) ve "factus" (yapma) kelimelerinden türetilmiştir, bu da kelimenin "el yapımı" anlamına geldiğini ifade eder (Groover, 2012). Tarihsel olarak, imalatın ilk evresi zanaatkarlıkla başlamıştır (Altın, 2012). Zanaatkarlar, el işçiliği kullanarak çeşitli ürünler üretmiş ve el aletleri yardımıyla farklı ürünlerin imalatını gerçekleştirmiştir. El işçiliği ile ortaya çıkan aletler, zanaatkarların imalatla etkin bir rol oynamasına katkıda bulunmuştur.

Zanaatkarlığın ardından, teknolojik gelişmelerle birlikte makineleşme süreci başlamış ve zanaatkarlığın zorlandığı hızlı ve aynı kalitede ürünleri üretme ihtiyacını karşılamak adına seri üretim sistemlerine geçiş yapılmıştır.

Seri üretim fikri, Yunan filozof Aristoteles'in M.Ö. IV. yy'de ifade ettiği, her araç kendi işini gerçekleştirebilseydi, bireylere ihtiyaç duyulmadan birçok işin gerçekleştirilebileceği düşüncesine kadar uzanmaktadır (Barutçuoğlu, 2001).

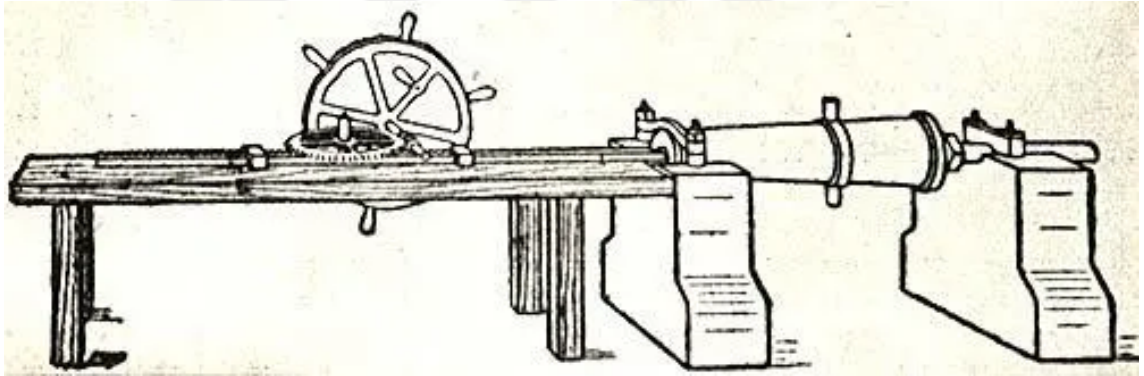
Seri üretimin tıpkı Aristoteles'in söylediği gibi insanlara ihtiyaç duymadan gerçekleşebilmesi için en büyük gelişme günümüzde CNC tezgâhı olarak adlandırılan ve genellikle x, y ve z koordinatlarında hareket edebilen motorların birleşmesi ile oluşan ve insan yönlendirmesine ihtiyaç duymadan, bilgisayarda üretilen verilerle çalışan mekanizmanın ortaya çıkışı ile olmuştur. CNC makinesinin ortaya çıkışı, endüstriyel robot kolların ortaya çıkışını getirmiştir. Tez kapsamında 3 boyutlu baskılama mantığının daha iyi açıklanması için sistemin temellerini oluşturan ve bağlı süreçlere sahip olan bu mekanizmaların tarihsel gelişimi ve bu gelişmelerin etkileri bu bölümde irdelenecektir.

CNC makinesinin bugünkü haline gelmesi için ortaya çıkan ilk gelişme 1725'te Fransız tekstil işçisi Basile Bouchon'un, bir dizi delikten kâğıt bantlara kodlanmış verileri kullanarak dokuma tezgahlarını kontrol etmenin bir yolunu icat etmesiyle gerçekleşmiştir (Şekil 3.1). Fakat ortaya çıktığı dönemde çok büyük bir gelişme olan bu yöntem, kırılgan bir yapıya sahip olup çalıştırmak için bir operatöre ihtiyaç duymaktadır (Mohammadi, 2019).



Şekil 3.1. Basile Bouchon'un tezgahı, (yarı) otomatik bir makinenin ilk endüstriyel uygulaması (Bales, 2023)

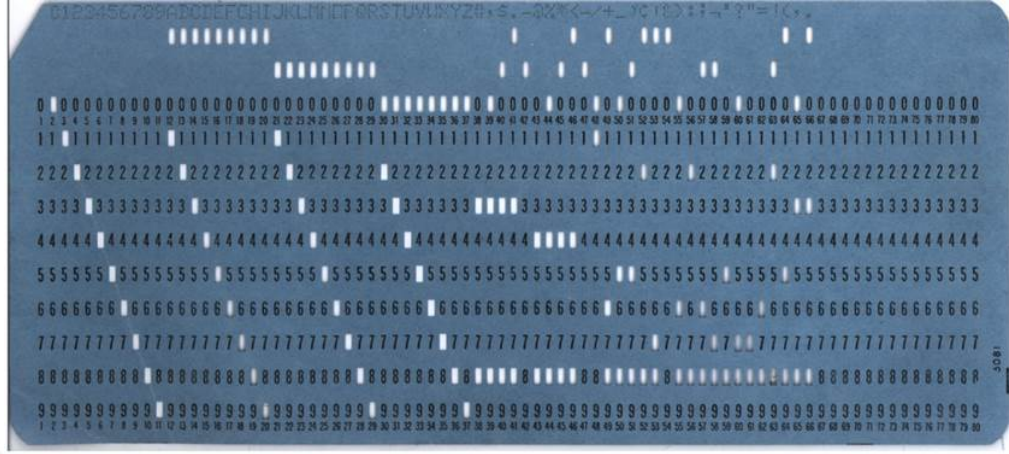
John Wilkinson'ın delme makinesi, 1775'te buhar motorları için silindirleri doğru bir şekilde delmek için bir çözüm olarak geliştirilmiştir (Şekil 3.1).



Şekil 3.2. John Wilkinson'un delme makinesi (Mohammadi, 2019)

1805'te Joseph Marie Jacquard, Basile Bouchon'un konseptini benimseyerek ve daha sağlam delikli kartları sırayla bağlayarak güçlendirmiş ve basitleştirmiştir. Böylece süreç otomatikleşmiştir. Bu delikli kartlar, modern bilgisayarların gelişmesinin temelini oluşturmaktadır. Delikli kartlar 1800'lerin ikinci yarısında gelişimini sürdürmüş ve farklı alanlarda kullanılmaya başlanmıştır. Telgraftan kendi kendine çalan piyanolara kadar birçok kullanımında gelişmeler gerçekleşmiştir. Bunlar mekanik kontrolle çalışırken, Herman Hollerith, elektromekanik delikli kart tabülatörünü geliştirerek sürecin dijital veriye aktarımında ilk adımlardan birini atmıştır (Şekil 3.3). Sisteminin patentini 1889'da Amerika Birleşik Devletleri Nüfus Sayım Bürosu için çalışırken almıştır. 1896'da Tabulating Machine Company'yi kurmuş ve 20. yüzyılın ikinci yarısında, delikli kartlar

ilk olarak bilgisayarlarda ve sayısal olarak kontrol edilen makinelerde veri girişı ve depolanması için kullanılmıřtır. Orijinal formatta beř sıra delik bulunurken, sonraki sřrřmlerde altı, yedi, sekiz veya daha fazla sıra bulunmaktadır (Mohammadi, 2019).



řekil 3.3. 80 sřtunlu IBM'in geliřtirdiđi delikli kart EBCDIC (Wikipedia, 2023)

1950'li yıllara kadar, sayısal olarak kontrol edilen makineler, genellikle manuel iřlemlerle řretilen delikli kartlardan elde edilen verilere dayanmaktadır ve NC (numerik kontrol) olarak adlandırılmaktadır. NC'den CNC'ye geçiř teknolojisinin evrimindeki kırılma noktası, bu kartların yerini, bilgisayarların geliřimiyle birlikte, aynı zamanda bilgisayar destekli tasarım (CAD) ve bilgisayar destekli řretim (CAM) programlarıyla entegre bir řekilde geliřtiđi zamanlarda olmuřtur.

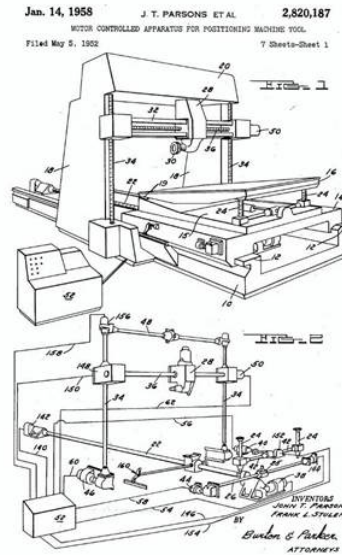
Westerlund, endřstriyel robotların iki farklı makineden kaynaklandıđını belirtmiřtir: Hidrolik montaj makineleri ve NC makineleri (Westerlund, 2000)

Bilgisayarın sřrece dahil olmasıyla birlikte freze ve torna tezgahlarındaki geliřme robot kol řretiminin de teknolojik ihtiyaını karřılařmıřtır. Charles Devol, 1954'te ilk endřstriyel robot olan Unimate'i geliřtirmiřtir (Robots.com, 2017).

1956 yılına gelindiğinde Massachusetts Teknoloji Enstitüsü (MIT) Servomekanizma Laboratuvarı'nda Bilgisayar Uygulamaları Grubu tarafından geliştirilen Otomatik Programlama Aracı (APT)'yi geliştirmiştir. APT sayısal kontrol sistemine sahip takım tezgahları için talimatlar oluşturmak amacıyla özel olarak tasarlanmış olan bir programlama dilidir (Robots.com, 2017).

1957 yılında Amerika Uçak Endüstrileri Birliği ve Hava Kuvvetleri'nin bir bölümü, APT ile çalışmayı standartlaştırmak ve ilk resmi CNC makinesini oluşturmak için MIT ile iş birliği yapmıştır. APT, geometriyi ve takım yollarını sayısal olarak kontrol edilen (NC) bir makineye iletmek için tek başına metin kullanmıştır (Turney, 2022).

1958 yılında Parsons, 1952 yılında geliştirmeye başladığı "Takım Tezgahını Konumlandırmak için Motor Kontrollü Aparat"(Şekil 3.4) makinesinin patentini almıştır (Mohammadi, 2019).



Şekil 3.4. Parsons'un patentini aldığı takım tezgahını konumlandırmak için motor kontrollü aparat (GooglePatents, 2023)

1959 yılında MIT ekibi, yeni CNC makine geliştirmelerini sergilemek için bir basın toplantısı düzenlemiştir. Bu toplantıda CNC ile frezelenmiş bir alüminyum kül tablası sergilenmiştir (Şekil 3.5). Böylece metal stoğu hareket ettiren vidaları döndürmek için bir makinist yerine bir bilgisayar programı kullanılması elle yapılabilecekten çok daha

karmaşık şekillere sahip bileşenlerin üretilmesi mümkün olmuştur. Bu gelişme döneminde çok fazla yankı bulmuş ve bilgisayar destekli üretim teknolojilerinde yeni bir dönem başlatmıştır (Gershenfeld, 2012).



MIT’de ki bu büyük gelişmenin ardından ilk endüstriyel robot kol prototipi olan Unimate (Şekil 3.6), 1961’de üretilmiş ve General Motors’un fabrikasında basınçlı döküm ve punta kaynağı için kurulmuştur (Wallén, 2008).



1963 yılında, Kaliforniya’daki Rancho Los Amigos Hastanesi’nde görev yapan araştırmacılar, engelli hastaların taşınmasına destek sağlamak amacıyla Rancho Kolu’nu (Şekil 3.7) geliştirdi. Bu, ilk bilgisayar kontrollü robotik kol olup, insan kolu gibi hareket etmesine olanak tanımak için altı eksenli olarak geliştirilmiştir (Nichols, 2019).



1969 yılında Unimation, Inc. General Motors (GM) tarafından büyük bir sipariş almıştır. Bu otomotiv üreticisi, Lordstown, Ohio'daki tesisini yeniden inşa ederek dünyanın en otomatik otomotiv fabrikasını kurma hedefine yönelmiştir. Fabrika faaliyete geçtiğinde, saatte 110 araba üretebilecek kapasiteye sahipti, bu da o dönemdeki diğer tesislerin üretim hızının iki katından fazlasını temsil etmekteydi. Bu artan üretkenlik, BMW, Volvo, Leyland, Fiat ve Mercedes-Benz gibi diğer otomobil üreticilerini benzer stratejileri benimsemeye teşvik etmiştir ve bu süreçte Unimation'ı Avrupa pazarında etkin bir konuma getirmiştir (Nichols, 2019).

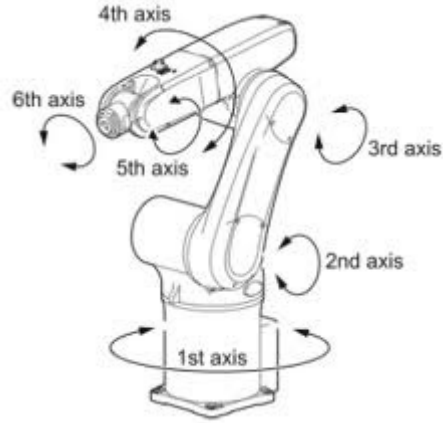
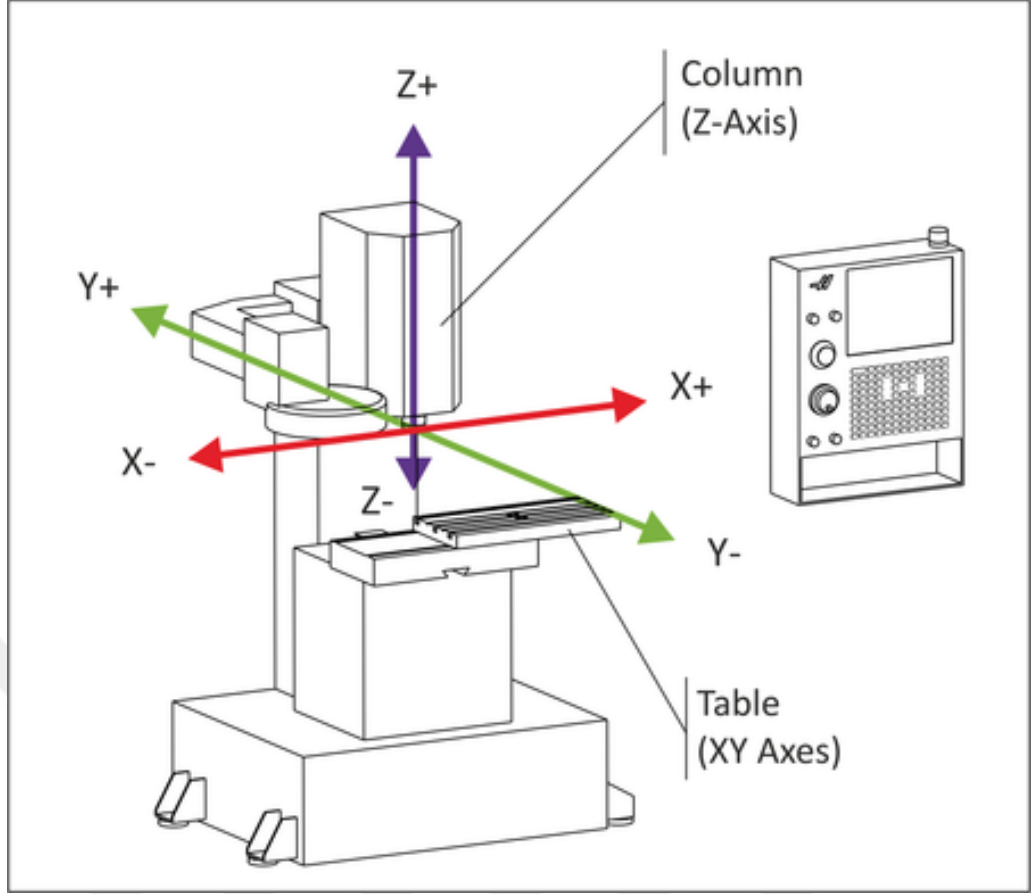
Eş zamanlı olarak, akademisyen Victor Scheinman'ın katkısıyla robotik, yeni bir boyuta ulaşmıştır. Stanford Üniversitesi'nin yapay zekâ (AI) laboratuvarında çalışan Scheinman, dünyanın ilk uygulanabilir çok eksenli robotik kolunu yaratarak robotik alanına geniş bir hareket serbestisi kazandırdı. Stanford Arm, hafif, elektrikle çalışan, antropomorfik ve çok programlanabilir özelliklere sahipti, bu da insan operatörlerin yerine çeşitli tehlikeli montaj hattı işlerini gerçekleştirebilmesini sağlıyordu. 1971'de ticari olarak piyasaya sürülen Stanford Arm, endüstriyel robotlar alanında devrim yarattı (Turney, 2022).

1960'lı ve 1970'li yıllarda, modern endüstriyel robot kolların küresel çapta gelişimi sürekli bir evrim göstermiştir. Dünya genelindeki şirketler arasındaki rekabet, endüstriyel robotlara yönelik artan talebi destekleyerek devam etmiştir. Bu durum, ek araştırma ve teknik gelişme çabalarını tetikleyerek, mikroişlemci gibi unsurların geliştirilmesi ve uygun maliyetli kontrol sistemlerinin oluşturulmasına katkı sağlamıştır.

Endüstriyel robotların gelişiminde verimli dönem olarak tanımlanabilecek bu dönemde diğer bir dönüm noktası 1973'te Alman şirketi Kuka'nın altı elektromanyetik aks kullanarak çalışan robotik bir kol olan Famulus'u piyasaya sürmesi ile gerçekleşmiştir. Bu kol, doğrudan her bir bağlantıya monte edilmiş motorlara sahip olması ile önceki kolları kullanan zincir veya tendon ihtiyacı, eklemlerin içine yerleştirilen elektrik motorları ile ortadan kaldırılmıştır (Jain, 2021).

Gargantua, Unimate ve Stanford Arm gibi konseptler, 90'lı yılların ortalarına kadar oldukça başarılı olmuş ve geniş kabul görmüştür. Ancak, 80'lerde endüstriyel lazerler gibi hızla gelişen teknolojiler, sensör teknolojisinin ve ilkel yapay görme sistemlerinin mümkün olmasını sağlamıştır. Bu dönemde endüstriyel robotların üretimdeki geleceği şekillenmeye başlamıştır. Bilgisayar kontrolü, masaüstü çağında standart hale gelmiş ve 1994'ün MRC (çoklu robot kontrolü) sistemi, bir robotu bir PC'den kontrol etme yeteneği sunarak önemli bir ilerleme sağlamıştır. Aynı dönemde mikroişlemci fiyatlarında keskin bir düşüş yaşanması, bilgisayar kontrollü robotları daha fazla endüstriyel alanda ve farklı sektörlerde kullanılabilir hale getirmiştir. Daha gelişmiş sensörler, uygun fiyatlı bilgi işleme gücü ve artan programlanabilirlik, endüstriyel robotları basit tekrarlayan görevlerin ötesine taşımış ve onlara insanların ilkel zeka olarak nitelendirdiği yetenekleri kazandırmıştır. Bu faktörler, robot teknolojisinin evriminde önemli bir dönüm noktasını temsil etmektedir (Turney, 2022).

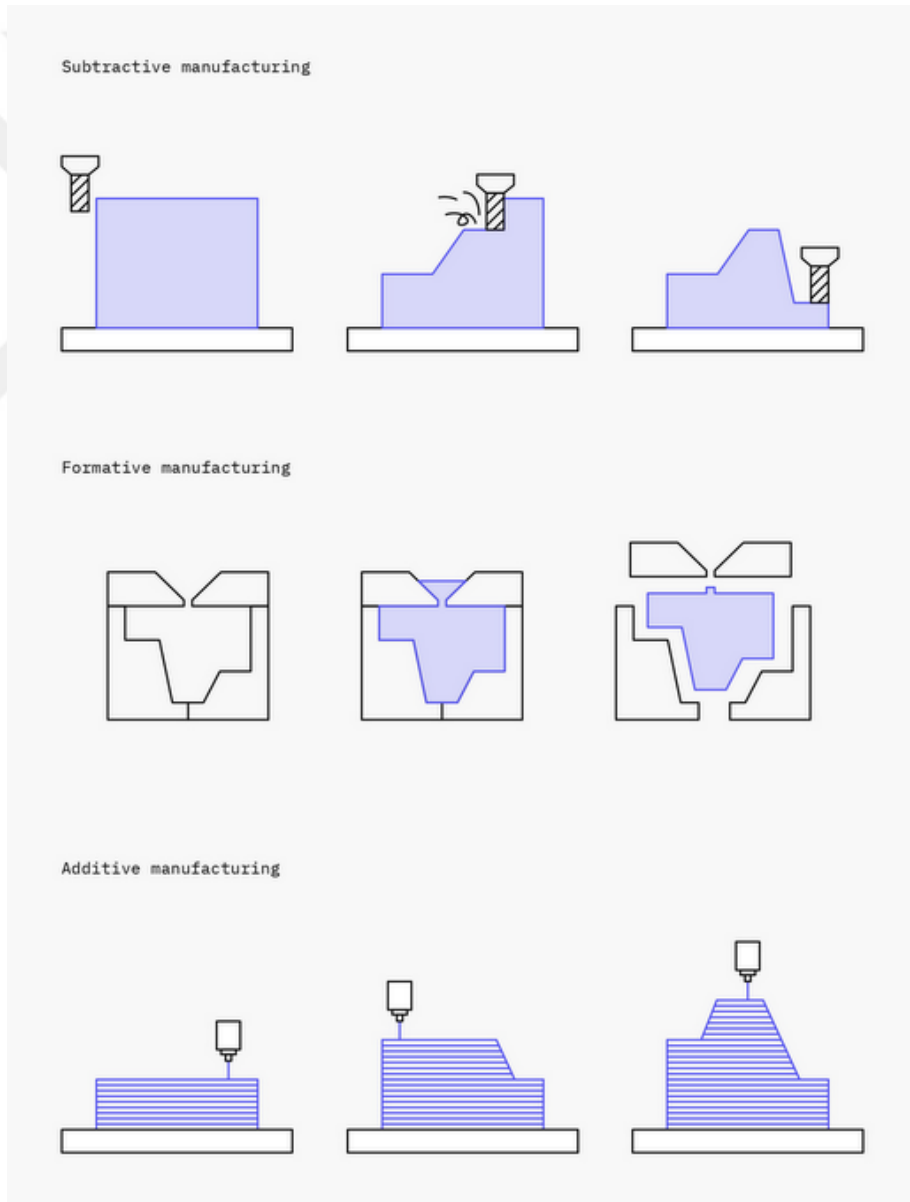
Tüm bu gelişimlerin kazanımları kısaca özetlenmek istendiğinde şu yargılara varılmaktadır. İnsanoğlu üretimlerini gerçekleştirebilmek için tarih boyunca insana ihtiyaç duymadan ve bir insanın yapabileceğinden daha verimli mekanizmalar üretme girişiminde bulunmuştur. Önce, günümüzdeki veri biliminin temellerini oluşturan ve bir mekanizmanın nasıl çalışması gerektiğini önceden kurgulayan delikli kartlar geliştirilmiş, sonraki süreçte bilgisayarlara bu sistem entegre edilmiştir. Sistemin amacı CNC tezgahlarındaki ve endüstriyel robotlardaki eksen hareketlerini Şekil 3.8 ve Şekil 3.9'da görüldüğü gibi daha öncesinde tanımlamaktır. Eksen hareketlerinin daha önce tanımlanabilmesi, otomasyonu sağlamıştır. Otomasyonun gelişimi üretim teknolojilerinin gelişimini sağlamıştır.



3 boyutlu baskılama teknolojisinin gerçekleşmesi için en önemli bileşenlerden olan bilgisayar ortamında üretilen veriye göre hareket eden bir sistem olan CNC'nin ve bu sistemin getirdiklerinden faydalanarak ortaya çıkan endüstriyel robot kolların ortaya çıkışı, gelişimi ve çok yönlü etkileri irdelenmiştir. MIT'in tanıttığı kül tablasından sonra

ve ilk bölümde bahsedilen bilgisayar destekli tasarım sistemlerinin gelişimiyle birlikte bu sistemler oldukça yaygınlaşmış ve döneminin geleneksel üretim yöntemini oluşturmuştur.

Bu geleneksel üretim yöntemlerinde, form genellikle malzeme bloğundan çıkarma veya biçimlendirme yöntemiyle oluşturulurken, katmanlı üretim veya eklemeli imalat form, malzemenin katmanlar halinde birleştirilmesiyle oluşturulur (Şekil 3.10). Bu birleştirme işlemi katmanlı üretim sürecinin temel aracı olan 3B yazıcılar aracılığı ile yapılır. Uluslararası Amerikan Test ve Malzeme Topluluğu (ASTM) tarafından, üç boyutlu model verilerinden nesneler oluşturmak amacıyla malzemeleri birleştirme işlemi olarak tanımlanan bu yöntem, genellikle üst üste katmanlar halinde birleştirme işlemine dayanır (ASTM, 2015).



Şekil 3.10. Çıkarımlı, biçimlendirmeli ve eklemeli imalat (Hubs, 2023)

Katmanlı üretim teknolojileri mimarlıkta yapı ölçeğinde üretim de dahil, birçok sektörde; havacılık, otomotiv, savunma, ürün geliştirme ve medikal alanlarında prototip üretiminden seri imalata kadar çeşitli süreçlerde kullanılmaktadır. Bu yöntem, dijital olarak oluşturulan nesnenin, özel yazılımlar aracılığıyla dijital olarak katmanlara ayrılması ve bu katmanların çeşitli hammaddelerin üst üste eklenmesiyle oluşturulma prensibine dayanır. Katmanlar halinde biriktirilen malzemenin sertleştirilme yöntemi, kullanılan malzemeye bağlı olarak eritme, kimyasal birleştirme veya bağlayıcılar kullanma gibi çeşitlilik gösterir.

3.2. 3 Boyutlu Yazıcıların Ortaya Çıkışı ve Gelişimi

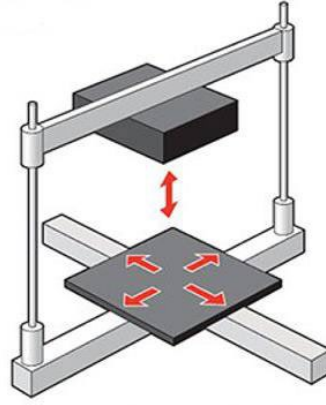
Üç boyutlu yazıcılar Japon bilim adamı Dr. Kodama tarafından 1980 yılında hızlı prototipleme tekniğiyle ortaya atılmıştır. Dr. Kodama, Stereolitografi (SLA) yöntemi olarak bilinen katman üzerine katman ekleme yaklaşımını kullanarak, imalat alanında bu adımı atan ilk bilim insanıdır. Aynı yıl Fransız bir araştırma grubu da Stereolitografi tekniği ile ilgilenmiş ancak patent başvurusu reddedilmiştir. Üç boyutlu yazıcılar için yapılan çalışmalara bu dönemde başlanmış olmasına rağmen, başarıya ulaşamamıştır (Bcn3d, 2020).

İlk kez 1984 yılında, üç boyutlu yazıcılar Charles (Chuck) Hull tarafından Stereolitografi (SLA) tekniği kullanılarak 3D System (3 Boyutlu Sistem) adlı bir şirketin öncülüğünde geliştirilmiştir (Çalışkan, 2015). 1986 yılında Stereolithography (SLA) adı ile patent alınmıştır. Üç boyutlu yazıcıların tarihsel gelişiminde, stereolitografi tekniğiyle katman üzerine katman ekleme yöntemi uygulanmıştır. Bu süreç, zaman içinde ortaya çıkan yeni yöntemler ve tekniklerin eklenmesiyle birlikte birçok alanda kullanılmaya devam etmiştir (Gibson vd., 2021).

3.3. 3 Boyutlu Yazıcı Tipleri

3.3.1. Kartezyen Tip Yazıcılar

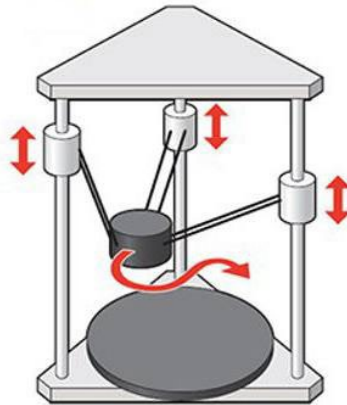
Kartezyen tip yazıcılar doğrusal koordinat sistemi kullanarak çalışmaktadır (Şekil 3.11). Yazıcı kafası formun yüksekliğine göre Z ekseninde hareket eder. Baskı platformu ise oluşturulmak istenen formun x ve y eksenindeki konumuna göre doğrusal hareket yapar. Bu yönüyle daha önce bahsedilen ilk CNC mekanizmalarının yansımasıdır. En yaygın kullanılan yazıcı tipidir (Schmitt vd., 2017).



Şekil 3.11. Kartezyen Tip Üç Boyutlu Yazıcı Şeması (Balat, 2019)

3.3.2. Delta Tip Yazıcılar

Delta tip yazıcılarda, kartezyen tip yazıcılardan farklı olarak, baskı platformu sabit bir konumda bulunmakta, yazıcı kafası ise her yöne serbestçe hareket etmektedir. Kartezyen koordinat sistemine alternatif olarak kullanılan bu yazıcı tipinde, yazıcı kafası Z eksenini boyunca doğrusal bir hareket gerçekleştirirken, X ve Y eksenlerinde sinüzoidal bir hareket izler. Üçgen yapıda bağlı olan kollar, yazıcı kafasını oluşturulmak istenen formun katman konumuna göre hassas bir şekilde konumlandırır. Delta yazıcılar, Kartezyen tip yazıcılara kıyasla, daha büyük üretim hacmi, hızlı baskı yapabilme ve sıcaklık kontrolünde daha iyi performans gibi avantajlara sahiptir (Schmitt vd., 2017) (Şekil 3.12).

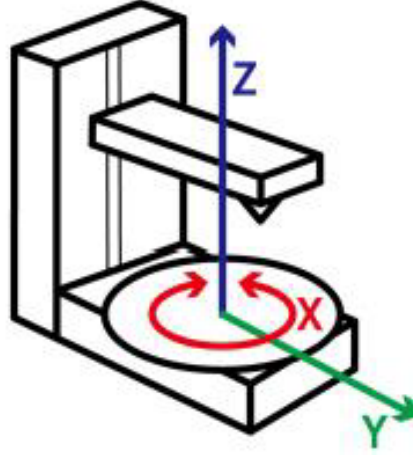


Şekil 3.12. Delta Tip Üç Boyutlu Yazıcı Şeması (Balat, 2019)

3.3.3. Polar Tip Yazıcılar

Polar yazıcılar, konumlandırma için kartezyen koordinat sistemi yerine polar koordinat sistemini benimserler. Polar koordinat kümeleri, kare yerine dairesel bir ızgaradaki noktaları tanımlar (Şekil 3.13). Yazıcı kafası her yöne doğrusal hareket

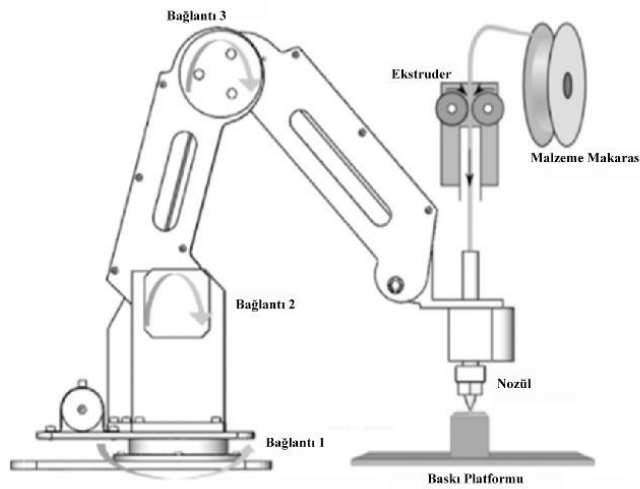
ettiğinde, platform dairesel hareket gerçekleştirir. Polar yazıcıların enerji verimliliği daha yüksektir ve sınırlı alanlarda bile daha büyük hacimli nesneleri başarılı bir şekilde üretebilirler.



Şekil 3.13. Polar Üç Boyutlu Yazıcı Şeması (Balat, 2019)

3.3.4. Robotik (Scara) Yazıcılar

Robotik yazıcılar, tek bir robot kolu tarafından kontrol edilen yazıcı kafasının doğrusal ve dairesel hareketini sağlamak üzere mafsallı bağlantılara sahiptir (Şekil 3.14). Sabit bir baskı platformu bulunmamaktadır. Yazıcı kafası, robotik kol sayesinde esnek bir hareket yeteneğine sahip olup, karmaşık yapıların üretimi daha kolay hale gelir (Balat, 2019). Bu tür yazıcılar, genellikle büyük ölçekli yapıların basımı gibi özel uygulamalarda tercih edilmektedir.



Şekil 3.14. Robotik (Scara) Yazıcı Şeması (Ulubeyli, 2022)

3.3.5. Non Planar- Düzlemsel Olmayan Yazıcılar

Düzlemsel olmayan üç boyutlu yazıcılarda hareket, yalnızca X, Y ve Z eksenlerinde değil, aynı zamanda dönme eksenlerinde de gerçekleştirilebilmektedir. Bu dönme hareketi, yazıcı kafası ya da yazdırma platformu aracılığıyla gerçekleştirilebilir. Yazıcı kafası ve platform arasında paralellik aranmaz; bunun yerine, oluşturulmak istenen formun belirli bir açıya göre konumlandırılması yapılır.

Düzlemsel olmayan yazıcılar, farklı yönlerde baskı yapabilme yeteneğinden dolayı yüzeyde biriktirme çizgisi (basamak etkisi) azaltılarak daha pürüzsüz yüzeyler elde etmeyi sağlar. Farklı açılarda biriktirme yapılabildiği için, destek baskılarına ihtiyaç duyulmadan üretim gerçekleştirilebilir (Fang vd., 2020).

2020 yılında Fang ve araştırma ekibi tarafından yapılan çalışmada, düzlemsel olmayan baskılama yöntemi ile üretilen nesnelerin düzlemsel baskılama ile oluşturulanlara göre 1.42 ile 6.35'e kadar daha dayanımı yüksek oldukları tespit edilmiştir (Fang vd., 2020).

Robotik yazıcılar da sahip oldukları fazla eksenlerinin getirdiği esnek hareket kabiliyetinden dolayı düzlemsel olmayan yazıcı grubuna girmektedir. Tez kapsamında geliştirilen modelin üretimi için robotik yazıcılar tercih edilmiştir.

3.4. Yapı Ölçeğinde Eklemeli İmalat

Yapı ölçeğinde eklemeli imalat süreci, beton karışımının önceden hazırlanması, yapı sahasına özelliklerini kaybetmeden taşınabilmesi ve uygulamanın kolaylaştırılması amacıyla geliştirilen beton pompalarının kullanımıyla başlamıştır. Beton pompalarının kullanımı, yapı sektöründe hazır betonun daha yaygın bir şekilde kullanılmasına olanak tanımıştır.

1997'de ilk kez Pegna tarafından ortaya atılan, beton malzeme ile üç boyutlu yazdırma fikri, Güney California Üniversitesi'nin 2003'te geliştirdiği Kontur Dolgu (Contour Crafting) teknolojisi ile hayata geçirilmiş ve 2010 yılında Loughborough Üniversitesi tarafından ilk beton yazıcı geliştirilmiştir. Farklı teknikler denense de, yapısal ihtiyaçlara odaklanan ve günümüzde kullanılan en yaygın teknikler Kontur Dolgu (Contour Crafting), D-Şekil (D-Shape) ve Beton Yazdırma (Concrete Printing)'tir (Çalışkan, 2015).

3.4.1. Kontur Dolgu (Contour Crafting)

Dr. Behrokh Khoshnevis'in NASA çalışmaları çerçevesinde Güney Kaliforniya Üniversitesi'nde yaptığı araştırmalar sonucunda geliştirilen teknoloji, mimari yapıların hızlı bir şekilde 3D yazdırılmasına olanak tanımaktadır (Leach, 2014). Uzay yerleşimine dair farklı araştırmalar yapan bilim insanları, literatürde "kontur üretimi" olarak adlandırılan bu üretim teknolojisi ile uzayda doğal malzemelerin kullanımını ve yapı üretimini sadece üretim sisteminin taşınması ile mümkün kılmayı amaçlamaktadır.

Kerpiç ve beton gibi yapı malzemeleriyle bina üretimi, üç boyutlu yazdırma teknolojisi kapsamında "kontur üretimi" (contour crafting) başlığı altında incelenmektedir. Halen gelişim süreci devam eden bu üretim teknolojisi, tasarlanan Bilgisayar Destekli Tasarım (CAD) modelinin g-code (g-kodu) formatında kaydedilen katmanlarında malzemenin üst üste biriktirilmesine dayanmaktadır (Lim, vd., 2009).

Kontur dolgu tekniğinde, duvarın dış hatları ekstrüzyon başlıkları ile katmanlar halinde basılır ve iç dolgu, dökme başlık ile oluşturulan duvarın dış hatlarına yerleştirilir. Ekstrüzyon başlığına monte edilen üst ve yan malalar, katman çizgilerini düzeltir ve merdiven etkisini azaltır (Şekil 3.15).



Şekil 3.15. Kontur Dolgu sistemi (Eplus3D, 2023)

Günümüzde kontur üretimi konusunda faaliyet gösteren otuzdan fazla araştırma grubundan bahsedilmektedir (Buswell vd., 2018). Hollandalı tasarım firması Dus Mimarlık'ın 2014 yılında dev bir yazıcı ile Amsterdam'da kanal evleri inşa etme kararı, bu alandaki çalışmalara yönelik bir örnektir (Hager vd., 2016).

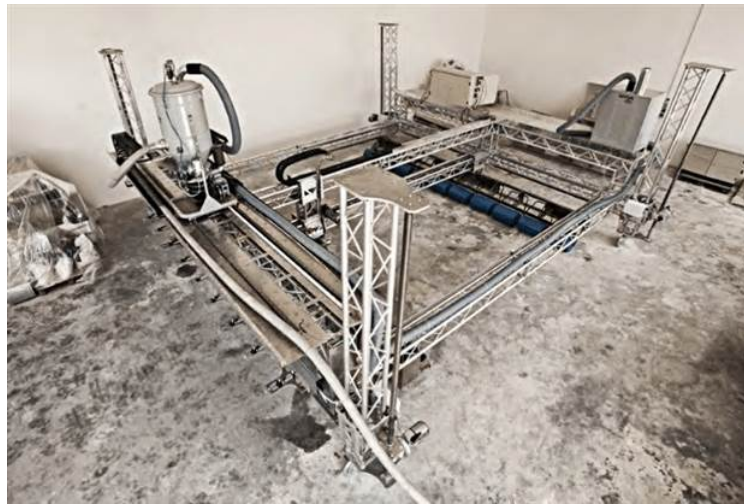
Kontur üretimi yönteminin geliştirilmesi sürecinde, sistem farklı açılardan çeşitli gelişmelere şahit olmuş, çeşitli üniversiteler ve firmalar tarafından üç boyutlu yazdırılabilir malzemeler, kompozit karışımlar ve çeşitli teknolojik makineler ve nozul modelleri geliştirilmiştir. İtalya'da 23 Temmuz 2016 tarihinde gerçekleştirilen bir çalışmada, 12 metre yüksekliğinde bir yazıcı ile kerpiç bina üretimi bir atölye etkinliği olarak sunulmuştur. İtalyan üretici tarafından geliştirilen pirinç ev, üç boyutlu yazıcılarla üretilen kerpiç yapılar için bir örnek olarak sunulabilir (Çalışkan, 2015).

Kontur üretimi sisteminde öncelikli olan sürdürülebilirlik, düşük maliyet ve doğal iklimlendirme prensipleri, tamamen geri dönüştürülebilir doğal malzeme ve atıklardan üretilen yapı malzemesinin kullanımını mümkün kılınmaktadır.

3.4.2. D-Shape (D-Şekil)

D-Şekil, magnezyum esaslı bağlayıcılar kullanarak beton objeler üretebilen bir yazdırma teknolojisidir (Ma vd., 2018). İtalyan bilim adamı Enrico Dini tarafından geliştirilen bu teknoloji, doğal yapı malzemeleri kullanarak mimari ölçekte üretimlere olanak tanımakta ve döneminin en büyük yazıcısı özelliğini taşımaktadır (Rael vd., 2011). 2006 yılında patente alınan sistem, kontur üretimi ve beton baskı sisteminden önemli ölçüde farklı bir üretim yöntemini benimsemektedir (Gardiner, 2011).

Kontur üretimi ve beton baskı sistemlerinde yığma tekniği kullanılırken, D-Form sisteminde selektif katman birleştirme yöntemi kullanılmaktadır. Bu sistem, bağlayıcı püskürtme alt yapısını kullanarak deniz suyu ve magnezyum esaslı bağlayıcı yardımı ile taş esaslı organik malzemelerden büyük ölçekte mimari bileşenlerin üretiminde kullanılmaktadır (Çalışkan, 2015) (Şekil 3.16).



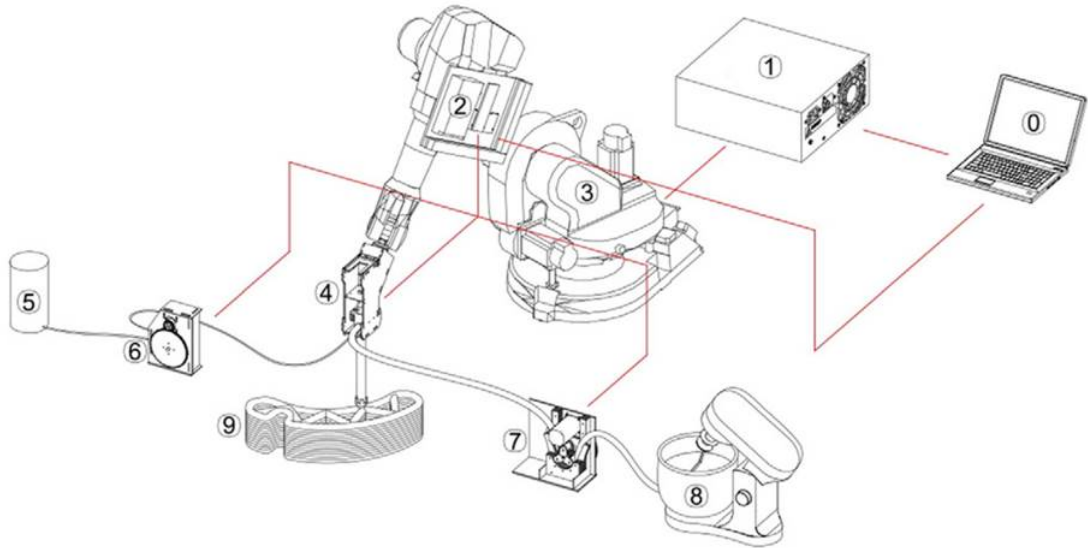
Şekil 3.16. İkinci nesil D-Shape makinesi (Gardiner, 2011)

3.4.2. Beton Yazdırma (Concrete Printing)

Beton Yazdırma (Concrete Printing), kontur üretimi sistemine benzemesine rağmen, daha küçük kapasiteli üretimler için uygun olan ve karmaşık geometrilerde daha iyi kontrol edilebilen bir sistem olarak bilinmektedir (Schipper vd., 2017).

Bu sistem, klasik üç boyutlu yazdırma sistemlerinden Eriyik Yığma Modelleme (Fused Deposition Modeling-FDM) yöntemine benzemekte ve yapı bileşenleri üretiminde kullanılmaktadır. İngiltere Loughborough Üniversitesi'nde yapılan bir araştırma sonucunda 2004 yılında icat edilen sistem, ilk üretimini 2006 yılında gerçekleştirmiş ve ilk dönemlerde "freeform construction" olarak adlandırılmıştır; ancak 2006 sonrasında beton baskı (concrete printing) adını almıştır (Gardiner, 2011).

Kontur üretimi sistemine oldukça benzer olan bu üretim yöntemi, nozul tasarımı ve üretim kapasitesi açısından farklılık gösterir. Daha detaylı üretimlere olanak tanıyan sistem, macun kıvamına getirilmiş harç ve alçı gibi malzemelerden bileşen üretiminde kullanılır. Kullanılan nozulun hassas çalışma kabiliyeti sayesinde, daha karmaşık geometrilerde beton formları oluşturulabilir (Şekil 3.17).



Şekil 3.17. Beton yazdırma şeması: 0. Sistem komutu; 1. Robot kontrolörü; 2. Yazdırma denetleyicisi; 3. Robotik kol; 4. Baskı Kafası; 5. Hızlandırıcı madde; 6. Peristaltik pompa hızlandırıcı madde; 7. Ön karışım için peristaltik pompa; 8. Ön karışım karıştırıcısı; 9. 3D baskılı nesne (Gosselin vd., 2016)

Tezin bu bölümünde bir makineye yapacağı bir işi önceden tarif etme ile başlayan sürecin bilgisayar teknolojilerinin üretim teknolojileri ile nasıl bir bütünlük içerisinde geliştiği gözlemlenmiştir.

4. MATERYAL VE YÖNTEM

4.1. Materyal

Bu çalışmada üretilmek istenen barınma birimi tasarımı, tek bir robot kolla üretilebilmenin doğası gereği en düşük alanda maksimum verimde mekân geliştirmek üzere kurgulanmıştır. Bunun için Le Corbusier'in kendi geliştirdiği Modülör'u kullanarak tasarladığı ilk standardize edilmiş mekân olarak tanımlanan Le Cabanon kaynak referans olarak belirlenmiştir.

Barınma biriminin dijital ortamda tasarımı ve üretimi için yapı bilgi modelleme yazılımı olan ArchiCAD ve hesaplamalı tasarım aracı olan Grasshopper kullanılmıştır.

4.1.1. Le Corbusier

1887 yılında İsviçre doğumlu olan Fransız vatandaşı Le Corbusier (gerçek adı Charles Edouard Jeanneret), akademik dogmalara meydan okuyarak modern mimarinin öncülerinden biri olmuştur. Le Corbusier, Hellenistik dönem mimarisinin pragmatik yönünden etkilenmiş ve bu etkiyi sanayiye yönlendirerek mimarinin geleceğini şekillendirmiştir (Berkin, 2021).

Pürizm akımının ilkelerini benimseyen Le Corbusier, 1919-1925 yılları arasında ressam Amédée Ozenfant ile birlikte "L'Esprit Nouveau" adlı dergiyi çıkarmıştır (Kortan, 1983). Pürizm, nesneleri alışılmadık şekilde yansıtan bir akımdır ve Le Corbusier, bu akımı takip ederek mimarlık camiasında yeni bir hareketin temellerini atmıştır.

1910 yılında Peter Behrens'in atölyesinde deneyim kazanan Le Corbusier, modern mimariye yönelik fikirlerini bu süreçte olgunlaştırmıştır. Daha sonra kendi çizgisini oluşturan Le Corbusier, 1923 yılında yayınladığı "mimarlıkta beş öge" ilkeleri ile mimari tasarımlarının temeli atmıştır (Berkin, 2021).

Sanayideki hızlı gelişim, Le Corbusier'nin mimarlık düşüncelerini derinden etkilemiş ve teknoloji transferi ve endüstri anlayışını yapıtlarına yansıtmıştır. Le Corbusier'e göre, endüstri, yeni çağa uygun gereçlerle mimarları donatarak mimarlığın evrimine öncülük etmiştir (Berkin, 2021).

Le Corbusier'in "zoning" (bölgeleme) yaklaşımı, karmaşık yapılar için kolaylık sağlayan bir temsil biçimidir (Turan, 2019). Mimar, diyagramı kullanarak planlamayı merkeze alır ve bu planlamayı insan bedeni üzerinden ölçülerle kurgular.

Le Corbusier, insanı tasarımın merkezine yerleştirirken, konutları yaşanabilir makineler haline getirme amacında idi. Ancak, bu yaklaşım birçok eleştiriye maruz kalmıştır, çünkü mekanik ve insan doğasına ters olarak nitelenmiştir (Berkin, 2021).

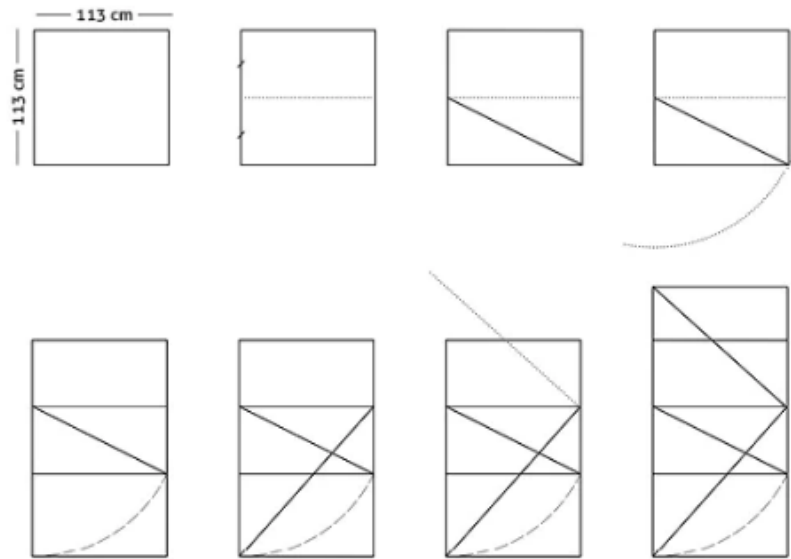
Dom-ino yapı sistemi, Le Corbusier'nin asgari mülkiyeti sağlama amacını yansıtan bir tasarım olarak öne çıkmıştır. Bu sistem, mimara esneklik ve özgürlük sağlayarak modern mimarinin temel ilkelerinden biri olan "açık plan"ı hayata geçirmiştir (Berkin, 2021).

Le Corbusier'in "Modulor" ve "Le Cabanon" gibi yapıları, farklı yaşam biçimlerine öncülük etmiş ve mimari tasarımda iz bırakmıştır (Akbulut ve Akay, 2013).

4.1.2. Modulor

Modulor, Le Corbusier ve ofis çalışanı Hanning tarafından 1943 yılında AFNOR kurumunun standartlaştırma talebine yanıt olarak geliştirilmiştir. Modulor ismi, Fransızca "module" (birim) ve "or" (altın) kelimelerinin birleşiminden türetilmiştir (Berkin, 2021). Bu matematiksel dizge, altın oranın insan boyutlarına uygun bir şekilde yerleştirildiği bir ölçü sistemini temsil etmektedir (Sözen ve Tanyeli, 2010).

Modulor ölçü sistemi, kare çizimi ile başlayıp altın dikdörtgene dönüşen bir şemayla işler. Çift kareye ulaşılırken, elini kaldırmış bir insan figürü bu şemaya yerleştirilir. Le Corbusier'nin belirlediği ölçüler, mobilya ve mekân tasarımında kullanılmaktadır (Şekil 4.1).



Şekil 4.1. Modulor'un kuruluşu (Berkin, 2021)

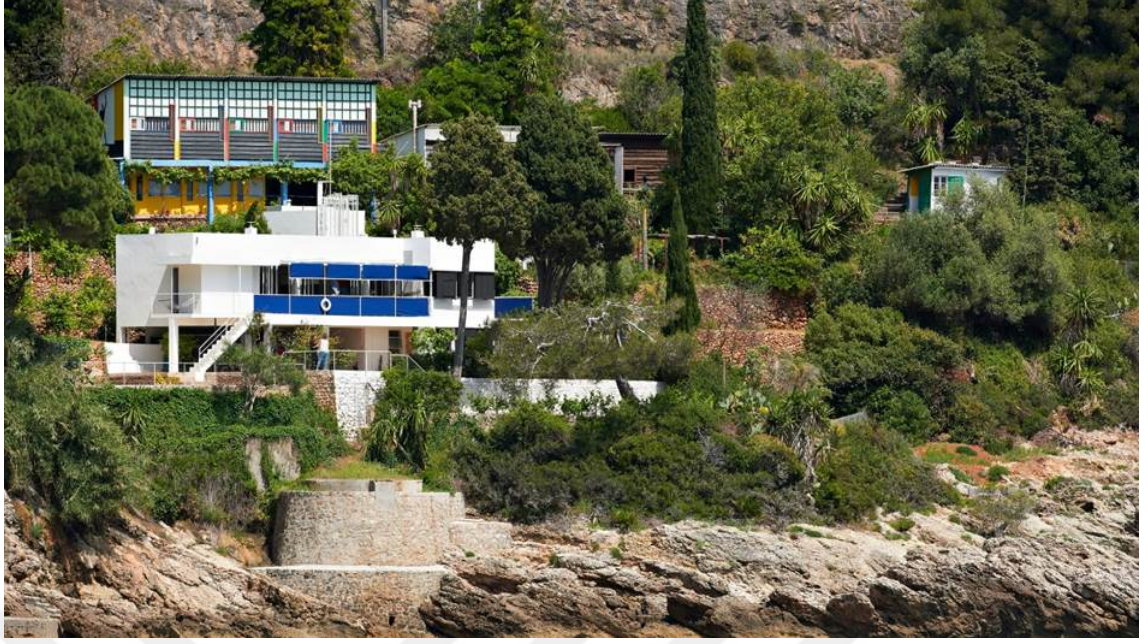
Le Corbusier, Modulor'u oluştururken, mimarlara estetikten ödün vermeden pratik kullanıma uygun bir şekilde mobilya ve mekân tasarımı yapabilmeleri için altın orana dayalı bir matematiksel düzen sağlamayı amaçlamıştır. Bu enstrüman hem planlama hem de cephe tasarımında kullanılmıştır. Le Corbusier'nin mimarlık serüveninde, mimarlığı endüstriyel bir ürün gibi ele alarak yapı tasarım sürecini diğer endüstriyel ürünlerden farksız gören bir bakış açısı benimsediği görülmektedir. Modulor, insan-makine-mekân ve mobilya ara yüzünü matematiksel bir düzen içinde birleştirmeyi amaçlayan bir araçtır. Le Corbusier, Modulor'unu oranlandırılmış bir ızgara olarak tanımlamış ve bu sistemi, mimarlıkta estetikten ödün vermeden pratik kullanıma yönelik bir referans olarak mimarlara bırakmıştır (Berkin, 2021).

4.1.3. Le Cabanon

Le Corbusier, deniz taşıtları ve uçaklardaki insan-makine-mekan uyumunu inceledikten sonra mimarlığın bu bağlamda evrim geçirmesi gerektiğine inanmıştır. Yapılan seyahatlerde küçük alanlarda deneyim kazanarak, kabin kavramının mimarının en küçük yaşanabilir mekânı olabileceği fikrine varmıştır. Kurnalı ve Koca'ya göre, kabin kelimesi temel olarak "küçük, özel bölme" ve "gemilerde, uçaklarda, uzay gemilerinde yer alan küçük bölme" anlamına gelir (Kurnalı ve Koca, 2018).

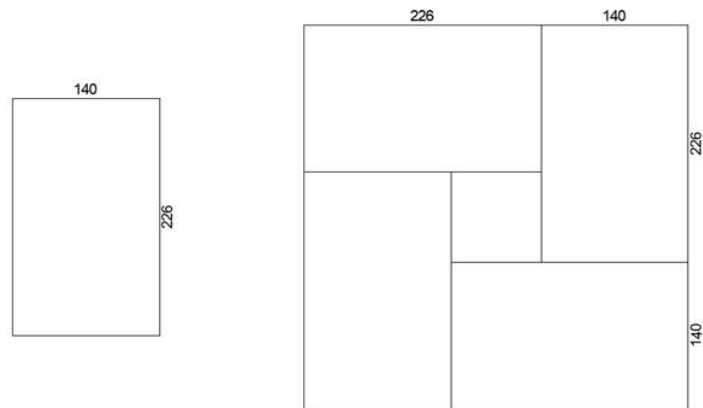
Le Corbusier, Roquebrune-Cap-Martin kıyılarında yaptığı gözlemlerle kabin yaşamının temellerini atmıştır. Bu bölge, samimi ve minimal yerleşim birimlerini birbirine bağlayan dar sokaklı bir kent dokusuna sahiptir. Eileen Gray'in bu bölgeyi keşfetmesi ve modern mimarının öğretilerini kullanarak tasarladığı E.1027 Evi, Le Corbusier'nin yanında yer almaktadır. Eileen Gray, bu evi "minimum house" olarak tanımlamış ve Le Corbusier'nin hemen yanına inşa ettiği 3,66×3,66 metrelik kabin-ev yapısıyla benzer bir konsepti gerçekleştirmiştir (Berkin, 2021).

Le Corbusier ve Eileen Gray'in eserleri neredeyse birbirine benzer nitelikte olup, ikisi de mimaride özgür düşüncüyü ön planda tutmuştur. E.1027 Evi'ni ziyaretinden sonra Le Corbusier, kendi geliştirdiği 5 mimari ilkenin bu evde hayata geçtiğini fark ederek, bu durum karşısında şaşırılmış ve etkilenmiştir. Le Corbusier, bu etkiyi hissettikten sonra, hemen evin 20 metre yukarısında Le Cabanon adını verdiği bir kulübe inşa etmiştir (Şekil 4.3). Her iki yapı da Roquebrune Cap de Martin'i mimari bir deneyim alanına dönüştürmüştür (Berkin, 2021).



Şekil 4.3. E.1027 ve Le Cabanon (Hemonet, 2023)

Eileen Gray, E.1027 Evi'nde bireysel kullanıma yönelik yaşam köşelerini, sosyalleşmeye olanak tanıyan meydanlara dönüştürmüştür. Altın oran ve Fibonacci spiriline dayalı olarak tasarladığı bu ev, Le Corbusier'nin Modulor'uyla benzer geometrik yapıları paylaşmaktadır. Le Corbusier, Modulor'u bu evi ziyaretinden sonra kendi projelerinde uygulamıştır. Le Cabanon, Le Corbusier için bir dinlenme ve yaratıcılığını tetikleme üssü olmuştur. Le Cabanon, Le Corbusier için modern mimarinin varlığını devam ettirebilmesi adına bir mekân tipolojisi olmuştur. Le Cabanon, Modulor'un ilk konut uygulaması olarak hayata geçirilmiş standart bir mekan olarak öne çıkar (Berkin, 2021).



Şekil 4.4. Le Cabanon plan oluşum dizisi

4.1.4. Bilgisayar Programları

4.1.4.1. ArchiCAD

Graphisoft, 1982 yılında başladığı çalışmalar sonucunda 1984 yılında piyasaya sürdüğü ArchiCAD yazılımı ile özellikle BIM (Building Information Modeling) alanında önemli bir gelişim kaydetmiştir. BIM yazılımları içerisinde sektörde önemli bir konumda bulunan ARCHICAD, son yıllarda çeşitli yazılımlarla gerçekleştirdiği iş birlikleri sayesinde potansiyelini daha da artırmıştır. Ayrıca, farklı disiplinlerle entegre bir şekilde çalışabilmesi, zaman ve tasarım etkinliği açısından olumlu gelişmeler olarak değerlendirilmektedir. Özellikle Grasshopper ile eş zamanlı çalışabilme özelliği sayesinde, BIM ve parametrik tasarımı entegre ederek karmaşık yapıların tasarlanmasına olanak sağlamıştır.

4.1.4.2. Grasshopper

Grasshopper, 2007 yılında David Rutten tarafından piyasaya sürülen ve Rhinoceros yazılımının bir eklentisi olarak çalışan bir plugindir. Bu eklenti, parametrik tasarımı mümkün kılar ve Rhinoceros içerisinde çeşitli algoritmik verilerin girilip kompleks tasarım formlarına dönüştürülmesine olanak tanır. İlk olarak Explicit History adıyla ortaya çıkan program, resmi adını 2007 yılında almıştır. Grasshopper yazılımı, Rhinoceros ile birlikte çalışarak parametrik tasarım imkanlarını ArchiCAD'e "Grasshopper Live Connection" adı altında aktarabilme yeteneği sunmuştur. Bu iş birliği sayesinde, Grasshopper ArchiCAD ile entegre olarak çalışabilir ve parametrik tasarım özelliklerini kullanıcıya sunar. Tez çalışmasında, bu özelliğin kullanımıyla ilgili bir model çalışması geliştirilmiştir.

4.2. Yöntem

Tez kapsamında yapılacak olan çalışmanın kavramsal altyapısını oluşturmak ve tanımlamak için ilk aşamada parametrik tasarım hakkında literatür araştırması yapılmıştır. Literatür araştırması için kavram hakkında yayınlanmış makale, tez ve kitap bölümleri incelenmiştir.

Barınma birimi tasarımının eklemeli imalat yapabilen robot kollar ile yapılması hedeflendiği için literatür araştırmasının ikinci kısmında dijital fabrikasyon ve eklemeli imalat konuları hakkında literatür araştırması yapılmıştır.

Barınma biriminin tasarımı robot kolların hareket kapasitesinin ve tek bir robot kolla üretimin gerçekleştirilebilme hedeflerinden dolayı en az alanda en fazla mekân verimliliği sağlamak üzere kurgulanmak istenmiştir. Bunun için yapılan araştırmalar sonucunda bu bağlamdaki en iyi örneklerden biri olan Le Corbusier'in Le Cabanon kulübesi tasarlanacak barınma biriminin ana materyali olarak belirlenmiş ve tezin materyal kısmında bu yapıya ait gerekli bilgiler verilmiştir.

Dijital ortamda üretilmek istenen barınma biriminin plan şemasının nasıl geliştirildiği ve barınma birimine dair detaylar; bunların yanı sıra hesaplamalı tasarım aracı olan Grasshopper ile üretilen algoritma, bulgular ve tartışma kısmında gösterilmiştir.



5. ARAŞTIRMA BULGULARI VE TARTIŞMA

5.1. Barınma Biriminin Plan Oluşumu

Le Cabanon'un planı, Le Corbusier tarafından dört ana bölgeye ayrılmıştır. Her bölge, 140×226 cm'lik altın dikdörtgen zemin modüllerinden oluşur (Şekil 4.4). İki bölge uyku ve dinlenme için ayrılırken diğer alanlar çalışma ve giyinme amacına hizmet etmektedir. Bu planlama, Le Corbusier'nin Modulor ölçülerinden türetilmiştir. Mobilyalar da aynı ölçülerde tasarlanmış olup, Le Cabanon'un içinde minimal ve işlevsel bir düzen yaratılmıştır. Le Cabanon'un mobilyaları, Le Corbusier'nin geliştirdiği Modulor ölçülerine uygun olarak tasarlanmıştır. Le Corbusier, mobilya tasarımlarını mümkün olan en sade ve kütesel şekilde gerçekleştirmiştir. Mobilyalar genellikle duvar kenarlarında yer almaktadır. Her zemin modülü (140×226), belirli bir bölgeye hizmet eden bir mobilya ve geçiş alanını (minimum 70 cm) içermektedir. Örneğin, yatak genişliği 70 cm olarak belirlenmiş ve bu zemin modülünden geçiş, 70 cm olarak hesaplanmıştır: 70+70=140 cm (Berkin, 2021) (Şekil 5.1).

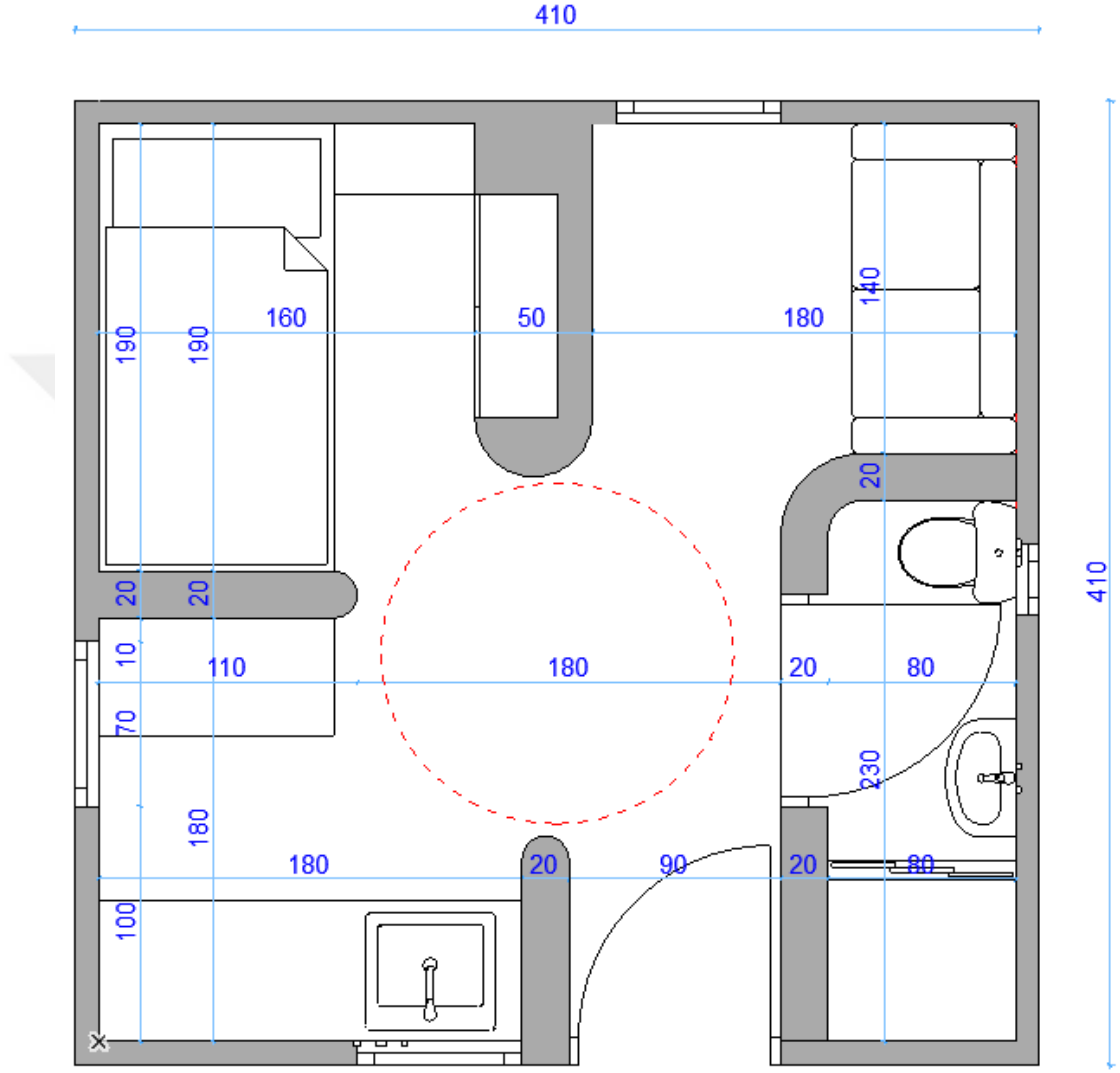


Le Cabanon; tefrişli plan.

- | | |
|--------------------------------|------------------------------|
| 1. Antre | 9. Masa |
| 2. Vestiyer | 10. Tabure (27x33 h = 43 cm) |
| 3. Yaşam alanına geçiş | 11. Depolama |
| 4. Pencere (15x105 cm) | 12. Konsol |
| 5. WC | 13. Pencere (33x70 cm) |
| 6. Gardrop (53x165 h = 140 cm) | 14. Lavabo |
| 7. Yatak (70x189 cm) | 15. Pencere (70x70 cm) |
| 8. Ek modül | 16. Pencere (15x105 cm) |

Şekil 5.1. Le Cabanon tefrişli plan (Berkin, 2021)

Tez kapsamında Le Cabanon incelendikten sonra geliştirilen plan Şekil 5.2’de gösterilmektedir.

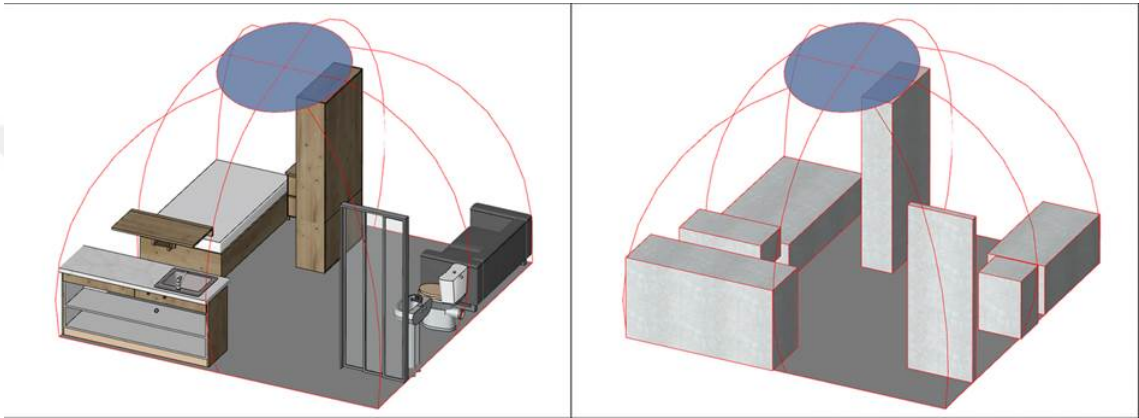


Şekil 5.2. Barınma birimi tefrişli plan ve ölçüleri

Tez kapsamında tasarlanan barınma biriminin plan şeması Le Cabanon’dan esinlenilerek bir kişinin yaşayabilmesi için minimum ihtiyaçlar doğrultusunda geliştirilmiştir. Birimin ölçülerindeki kısıtlılıkları oluşturan temel faktör, üretimin tek bir robot kol ile gerçekleştirilmesi üzerinedir. Bu kısıtlılık standartize edilmiş ilk mekân olarak nitelenen Le Cabanon ile benzerlik göstermektedir.

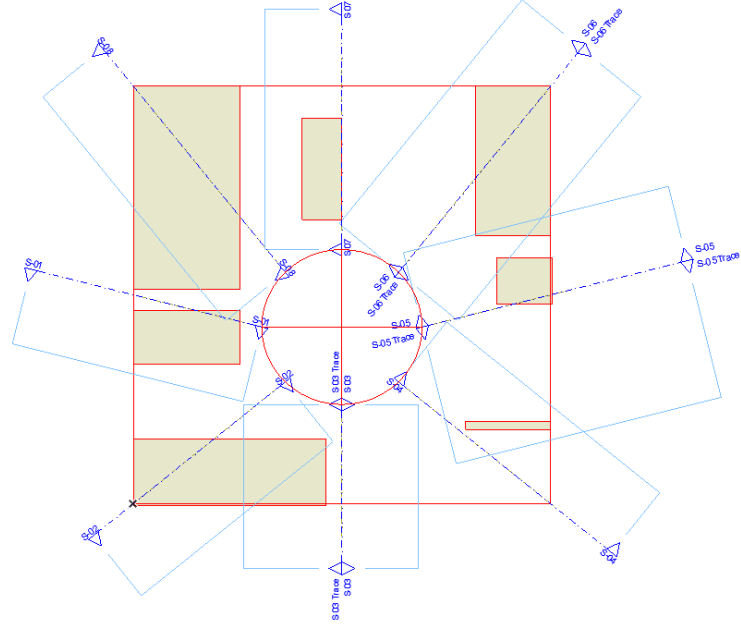
5.2. Barınma Biriminin Kabuk Yapısı

Schumacher'e göre mobilyalar, mekandan bağımsız objeler değil, mekanları oluşturan objelerdir (Schumacher, 2008). Bu kapsamda tasarlanan barınma biriminin kabuk yapısı mobilyaları kapsayacak şekilde üretilmiştir. Parametrik Tasarım araçlarının getirisi sayesinde de mobilyalardaki ölçüler ve mobilyaların yerleri belirli sınırlar dahilinde değiştirilse bile kabuk yapısı kendisini yeniden üretebilmektedir. Şekil 5.3'te barınma birimindeki tefriş elemanları önce gerçek modelleri ile daha sonra onları kapsayan kutular ile gösterilmiştir.



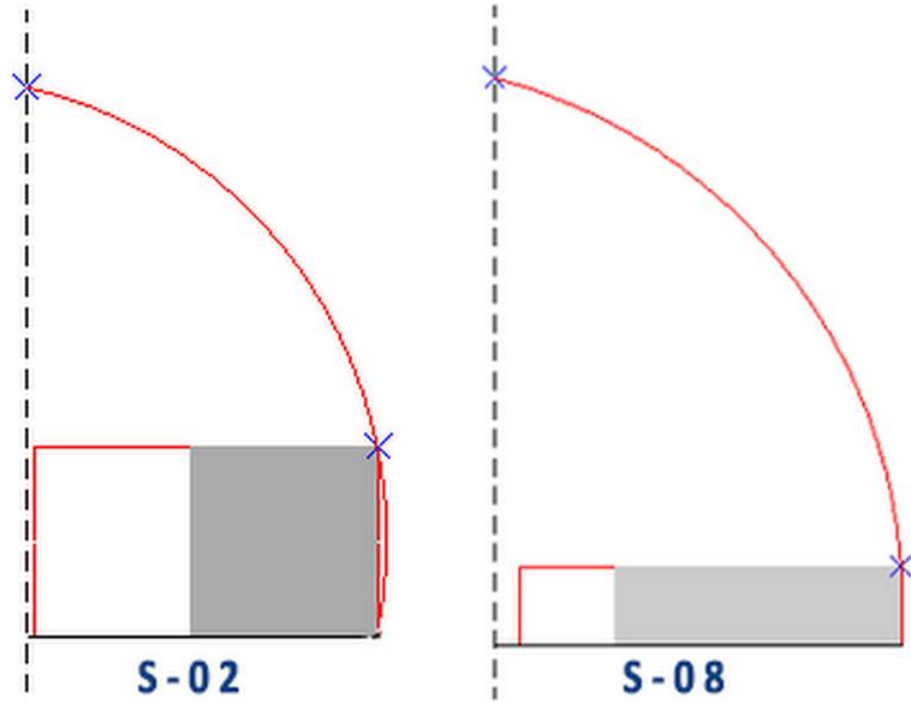
Şekil 5.3. Barınma birimi mobilya ilişkisi

Barınma biriminin iç kabuğu, her kenar ve köşedeki ilişkili tefrişatı kapsayacak şekilde 8 farklı birbirinin aynı olmayan kırmızı renkteki eğriler ile referanslanmıştır. Şekil 5.4'te bu eğrilerin ve bu eğrilere referanslanan kesitlerin plan düzleminde görünümü gösterilmiştir.



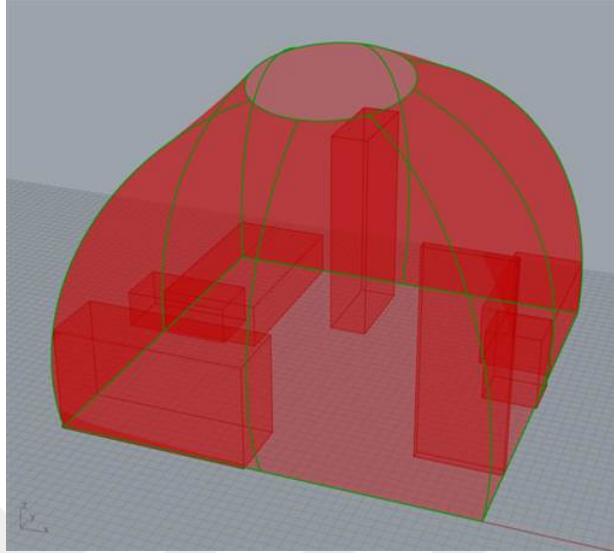
Şekil 5.4. Birimin iç kabuğunu oluşturan eğrilerin referans kesitleri.

Eğrilerin tefriş elemanlarını nasıl kapsadığı Şekil 5.5'te S-02 ve S-08 kesitleri üzerinden gösterilmiştir. Yapının iç kabuğunu oluşturan eğriler S-02'de mutfak tezgahını, S-08'de yatağı kapsamaktadır.



Şekil 5.5. S-02 ve S-08 kesitlerinde kabuk eğrisi tefrişat ilişkisi

Tefrişat parametresine göre şekillenen eğriler ile oluşan kabuk yapısını oluşturan ve mekânın sınırlarını belirleyen sekiz farklı eğri arasındaki sekiz farklı yüzeyin birleştirilmesi ile elde edilen bütüncül yüzey Şekil 5.6’da gösterilmiştir.

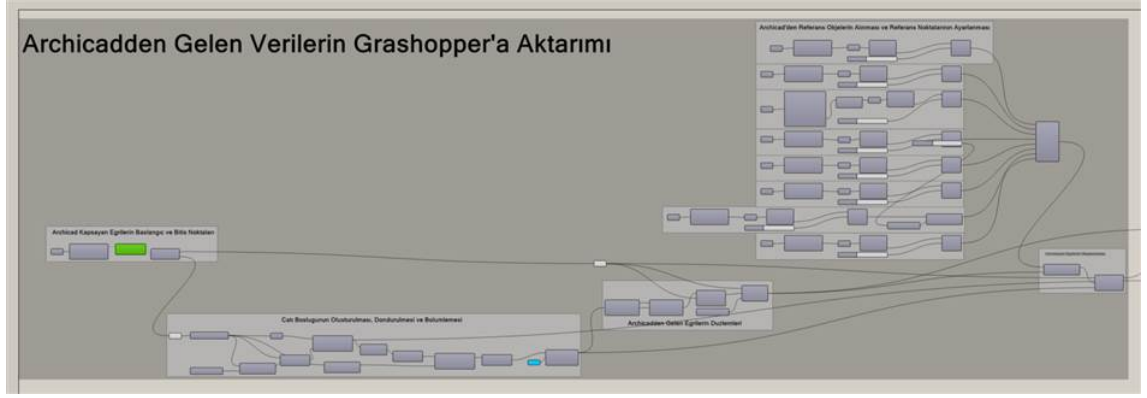


Şekil 5.6. Birimin kabuk yapısının referans eğrileri ile ilişkisi

Görülmektedir ki barınma birimi tasarlanırken parametrik tasarım ilkelerinden faydalanılmıştır. Sonraki bölümde barınma biriminin tasarımındaki parametrik tasarım anlayışının günümüzde şekillendiği hali olan dinamik model güncellemesini sağlayan algoritmalar ile nasıl ilişkilendirildiği aktarılacaktır.

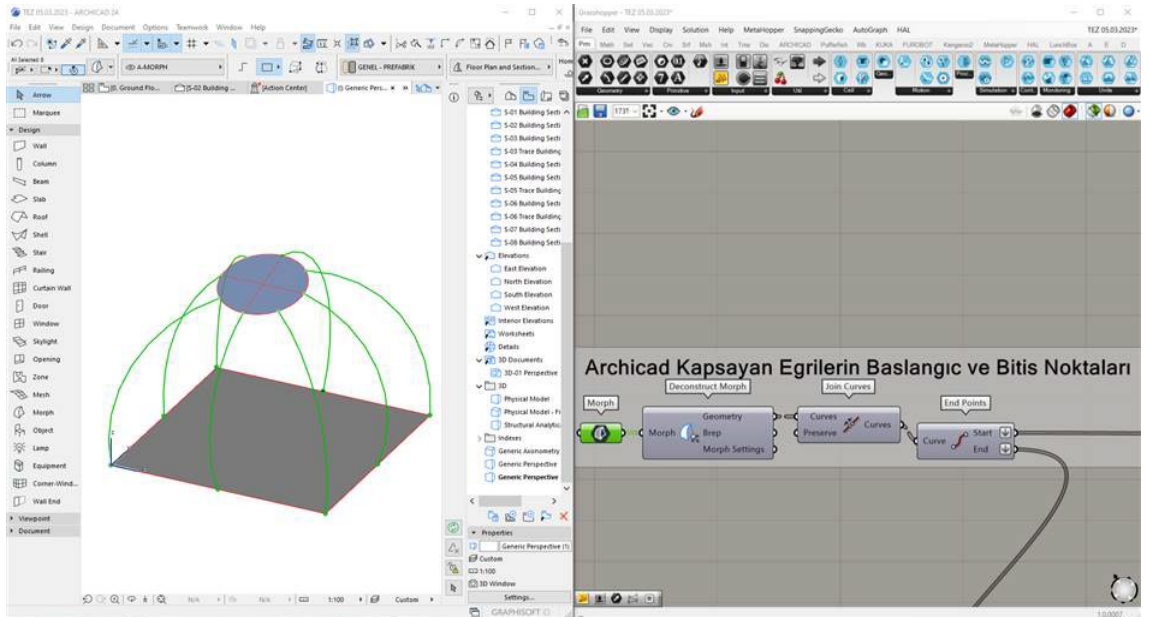
5.3. Barınma Biriminin Parametrik Algoritması

ArchiCAD’de plan şeması oluşturulan ve buna bağlı olarak tefrişlerin yerleştirildiği dosyadan gelen verilerin Grasshopper’a aktarımı için oluşturulan algoritma Şekil 5.7’de gösterilmiştir. Bu algoritma, genel kümede “ArchiCAD’den Gelen Verilerin Grasshopper’a Aktarımı” olarak isimlendirilmiş ve koyu gri renkte belirtilmiştir. Algoritmayı oluşturan alt kümeler ise açık gri renkte belirtilmiştir. Bu algoritmanın Grasshopper’da hangi bileşenler ile oluşturulduğu ve bu bileşenlerin işlevleri bu bölümde adım adım açıklanmaktadır.



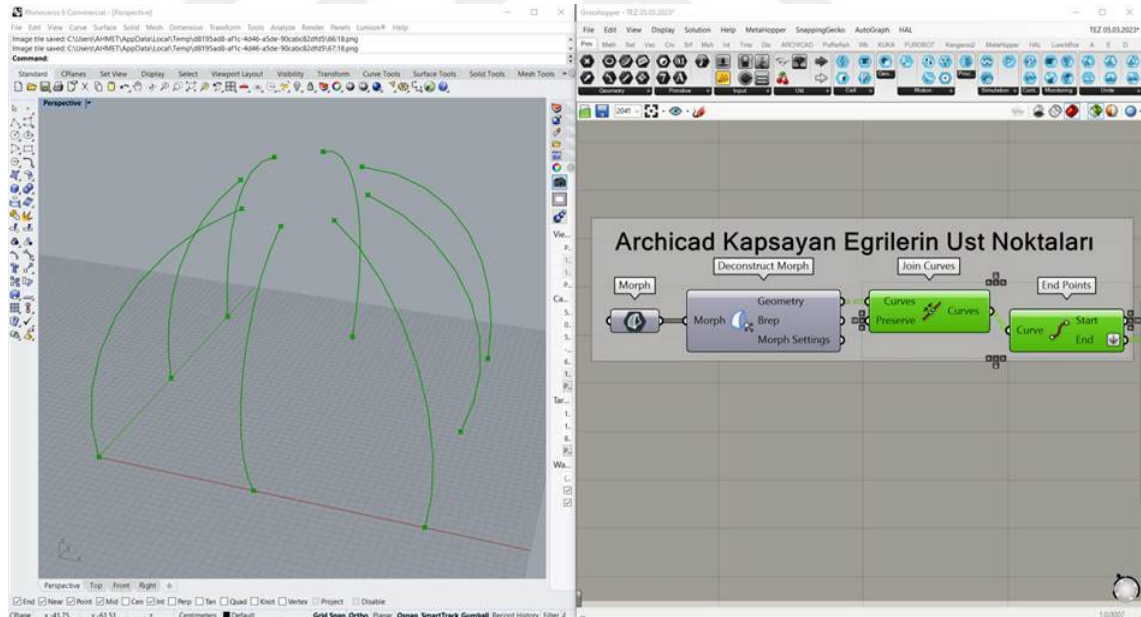
Şekil 5.7. ArchiCAD'den gelen verilerin Grasshopper'a Aktarımı Algoritması

İlk adımda oluşturulan algoritma alt kümesi “ArchiCAD Kapsayan Eğrilerin Başlangıç ve Bitiş Noktaları” olarak adlandırılmıştır. Bu alt kümede ArchiCAD’de oluşturulan ve barınma biriminin mobilyalarını kapsayan eğriler Grasshopper’a aktarılmıştır. Bu Grasshopper yazılımı ile ArchiCAD’in ortaklaşa çalışmasını sağlayan eklentinin Grasshopper’a yüklenmesi ile sağlanmıştır. Yükleme yapıldıktan sonra Grasshopper arayüzüne ArchiCAD bölümü eklenmiştir. Bu bölümde bulunan ve ArchiCAD parametrelerini Grasshopper’a aktaran “Morph” bileşenine ArchiCAD’de tüm eğriler seçilip atanmıştır. Şekil 5.8’de Grasshopper’da “Morph” bileşeni seçili iken ArchiCAD üç boyutlu görüntüleme arayüzünde bu eğrilerin yeşil renkte belirdiği gözlemlenmektedir.



Şekil 5.8. Kapsayan eğrilerin başlangıç ve bitiş noktaları

Bu adımda amaç ArchiCAD’de oluşturulan ve tefrişat elemanlarını kapsayan eğrilerin başlangıç ve bitiş noktalarının değerlerine ulaşmaktır. Bu yüzden bu algoritma alt kümesi “ArchiCAD’de Oluşturulan Eğrilerin Başlangıç ve Bitiş Noktaları” olarak adlandırılmıştır. Bunu sağlamak için ArchiCAD’de “Morph” komutu ile oluşturulan eğriler Grasshopper’daki “Morph” bileşenine atanmıştır. ArchiCAD’deki verilerin Grasshopper’a aktarılması için bu bileşen “Deconstruct Morph” bileşenine girdi olarak bağlanmıştır. ArchiCAD poligonal tabanlı bir modelleme verisi ürettiği için bu eğriler Grasshopper’a aktarıldığında parçalı olarak ulaşmıştır. Bu yüzden Grasshopper’da “Deconstruct Morph” bileşeninin çıktılarından olan “Geometry” verileri Grasshopper’ın kendi bileşenlerinden olan ve aynı alt kümedeki eğrilerin birleştirilmesini sağlayan “Join Curves” bileşenine atanmıştır. Bu bileşenin çıktısı olan ve artık Grasshopper verisi olan eğrilerin üst noktalarını saptamak için “End Points” bileşenine girdi olarak bağlanmıştır. Şekil 5.9’da seçili olan bileşenler yeşil renkte belirlemekte(sağ), Rhino arayüzünde de üç boyutlu model görünümü(sol) gözlemlenmektedir. Bu adımda iç kabuk yapısını oluşturan eğrilerin başlangıç ve bitiş noktaları uzay geometrisinde elde edilmiştir.

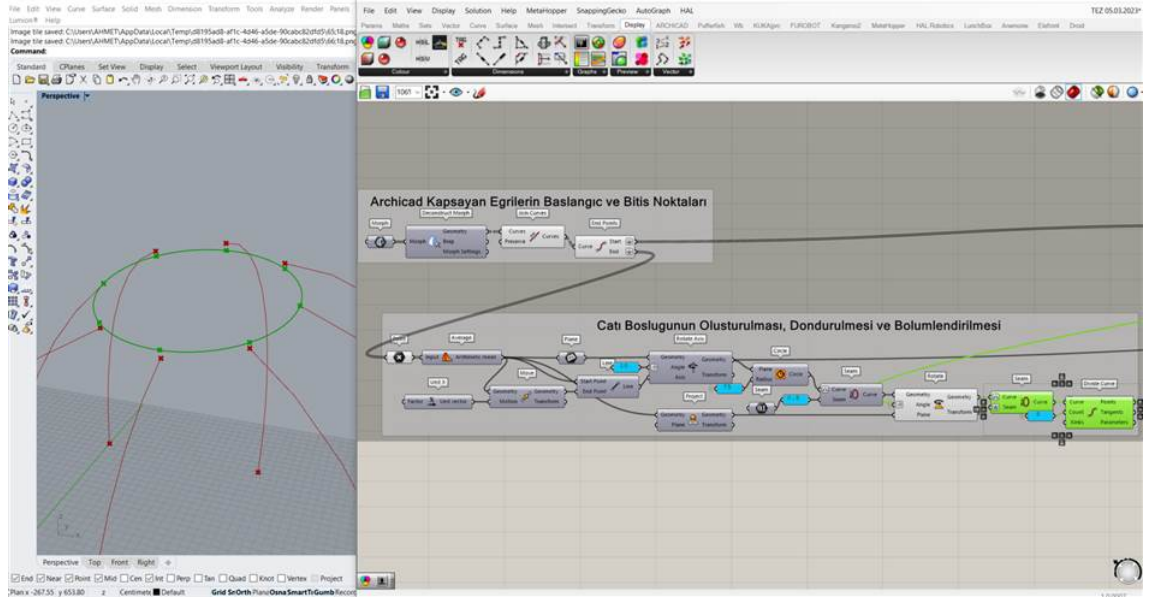


Şekil 5.9. ArchiCAD’de Oluşturulan Eğrilerin Başlangıç ve Bitiş Noktaları

İç kabuk yapısını oluşturan referans eğrilerinin başlangıç ve bitiş noktalarının verisi elde edildikten sonraki adımda amaç çatı boşluğunun verilerinin Grasshopper’a aktarılmasıdır. Bu yüzden bu algoritma alt kümesi “Çatı Boşluğunun Oluşturulması, Döndürülmesi ve Bölümlendirilmesi” olarak adlandırılmıştır.

Şekil 5.10'da görüleceği üzere bu kümenin ilk girdi verisi bir önceki kümeden gelen ve kapsayan eğrilerin bitiş noktası olan çatı boşluğunun referans noktaları olmuştur. Noktaları oluşturan bu veriler Grasshopper'ın bileşenlerinden olan "Point" bileşenine atanmıştır. "Average" bileşeninin girdisine bu noktalar bağlanarak tüm noktaların uzay geometrisinde ortalama noktası bulunmuştur. Bu nokta çatı boşluğu için oluşturulacak geometrinin merkez referansını bize vermektedir. Bu merkez referansına bağlı dairesel düzlemin döndürülmesi için nokta ile ilişkili bir doğru parçası oluşturulmalıdır. Bunun için bu noktayı "Move" bileşeni ve bu bileşenin "Motion" girdisine "Unit X" bileşeni bağlanarak x ekseninde 1 birim hareket etmesi sağlanmıştır. "Average" bileşeninin çıktısı olan ortalama nokta ve "Move" bileşeninin çıktısı olan hareket ettirilmiş nokta "Line" bileşeninin girdilerine bağlanarak çatı boşluğunun döndürülmesi için gerekli aksı sağlayan doğru parçası elde edilmiştir. Bu aksa bağlı döndürme işleminin gerçekleştirilebilmesi için gerekli olan "Rotate Axis" bileşenine ortalama noktanın çıktı verisinin "Plane" bileşenine bağlanması ile elde edilen çıktı verisi "Geometry" girdisine, açı verisi 10 derece olarak "Angle" girdisine ve elde edilen doğru parçası ise "Axis" girdisine bağlanmıştır. Bu işlem sonucunda çatı boşluğunun yerleşeceği düzlem parametrik olarak elde edilmiştir. Bu düzlem verisi "Circle" bileşeninin "Plane" girdisine ve 75 değeri de "Radius" girdisine bağlanarak çatı boşluğu için gerekli daire şekli elde edilmiştir. Bu dairenin bölümlendirilmesinin uygun noktadan başlaması için "Circle" bileşeninin çıktısı "Seam" bileşeninin "Curve" girdisine bağlanmış ve bu eğri tekrar parametrize edilip "Seam" girdisine 0.5 değeri bağlanarak bu işlem tamamlanmıştır. Bu işlemden sonra dairenin düğüm noktası uygun olarak ayarlanmış olup "Seam" bileşeninin çıktısı "Divide Curve" bileşeninin "Curve" girdisine, 8 değeri de "Count" girdisine bağlanmıştır.

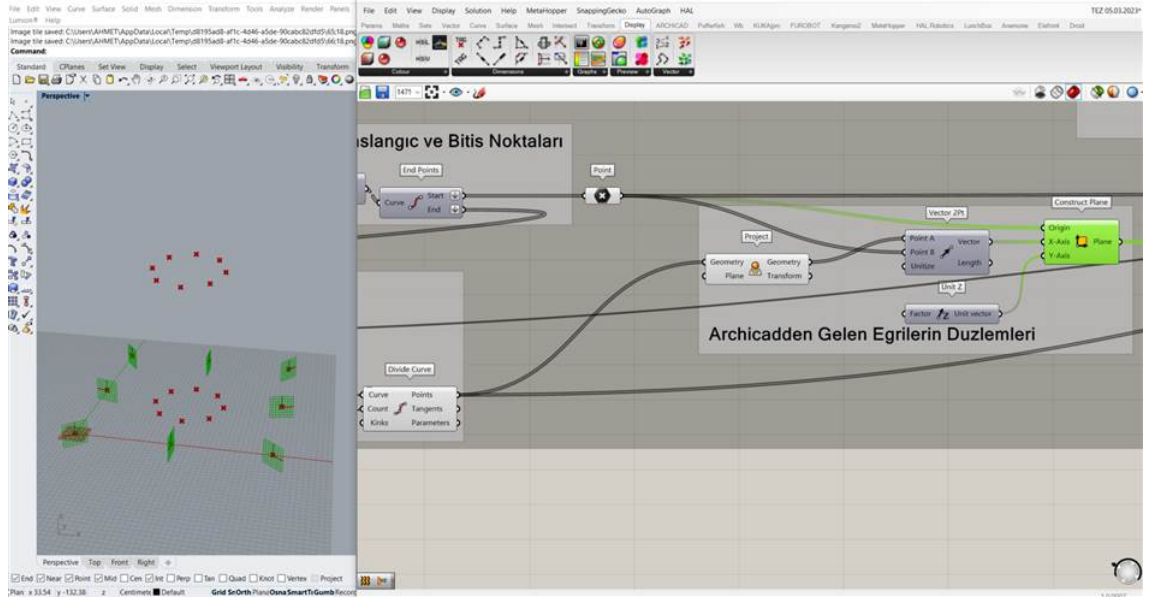
Tüm bu algoritma alt kümesi boyunca yapılan işlemler sonucunda ArchiCAD'de ki eğrilerin üst noktaları Grasshopper'a parametrik olarak aktarılmıştır. Şekil 5.10'da soldaki Rhino üç boyutlu görüntüleme arayüzünde kırmızı renkte olanlar ArchiCAD'den gelen eğriler ve bitiş noktaları, yeşil renkte olanlar ise çatı boşluğu için elde edilen daire ve bunun uygun bölümlendirilmiş noktalarıdır.



Şekil 5.10. Çatı boşluğunun oluşturulması, döndürülmesi ve bölümlendirilmesi algoritması

ArchiCAD’de oluşturulan eğrilerin başlangıç ve çatı boşluğuna uzanan noktaları Grasshopper’a aktarıldıktan sonra bu adımda hedef bu eğrilerin Grasshopper’da tekrar oluşturulabilmesi için eğrilerin yerleşeceği düzlemlerin elde edilmesidir. Bu yüzden bu algoritma alt kümesi “ArchiCAD’den Gelen Eğrilerin Düzlemleri” olarak adlandırılmıştır.

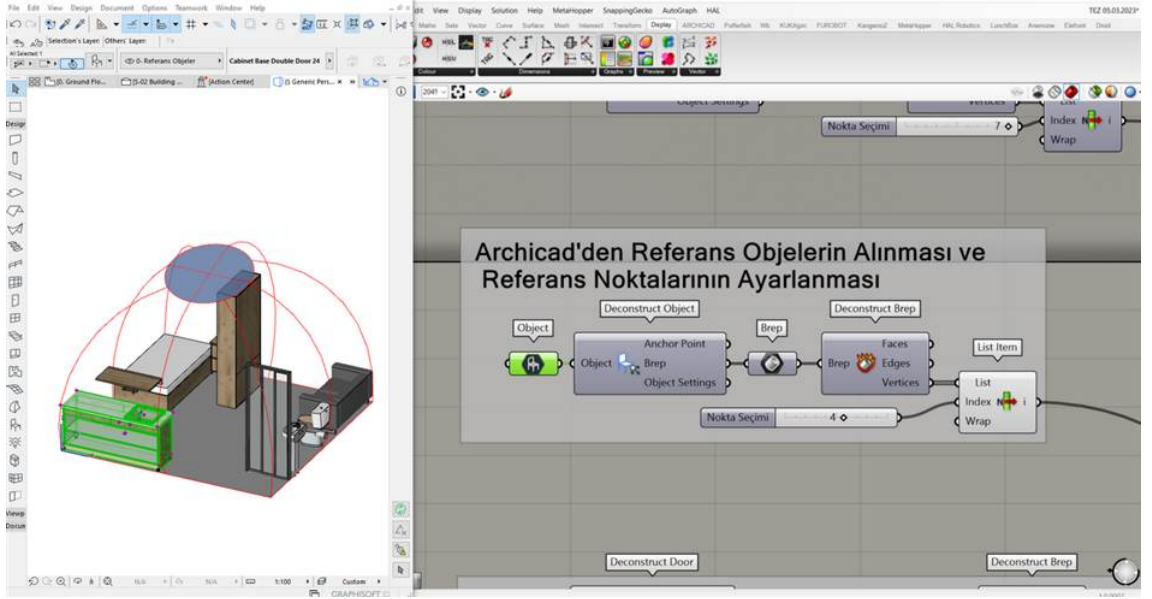
Bu adımda Şekil 5.11’de görüleceği üzere başlangıçta çatı boşluğunun düzenlenmiş veri seti noktalarının çıktı verisi “Project” bileşenine bağlanıp çıktı verisi olarak bu noktaların xy düzlemindeki yansımaları elde edilmiştir. Bu çıktı değeri “Vector 2Pt” bileşeninin girdilerinden “Point B” ye atanmıştır. Sonrasında ilk algoritma alt kümesinde elde edilen eğrilerin başlangıç noktalarının çıktı verisi “Vector 2Pt” bileşeninin “Point A” girdisine bağlanarak oluşturulmak istenen düzlemin x aksı elde edilmiştir. “Vector 2Pt” bileşeninin çıktı verisi olan ve oluşturulmak istenen düzlemin x aksının verisi olan “Vector” çıktı verisi “Construct Plane” bileşeninin “X-Axis” girdisine bağlanmıştır. Bu algoritma alt kümesinde yapılan işlemler sonucunda elde edilen veri, barınma birimini çevreleyen eğrilerin Grasshopper’a aktarılması için yerleşecekleri düzlemlerin verisini vermektedir Şekil 5.11’de bu düzlemler Rhino üç boyutlu görüntüleme ekranında yeşil renkte gözlenebilmektedir.



Şekil 5.11. Yapı kabuğunun ve duvar strüktürlerinin yüzeylerinin 3 boyutlu baskılama ile elde edilmesi için geliştirilen algoritma

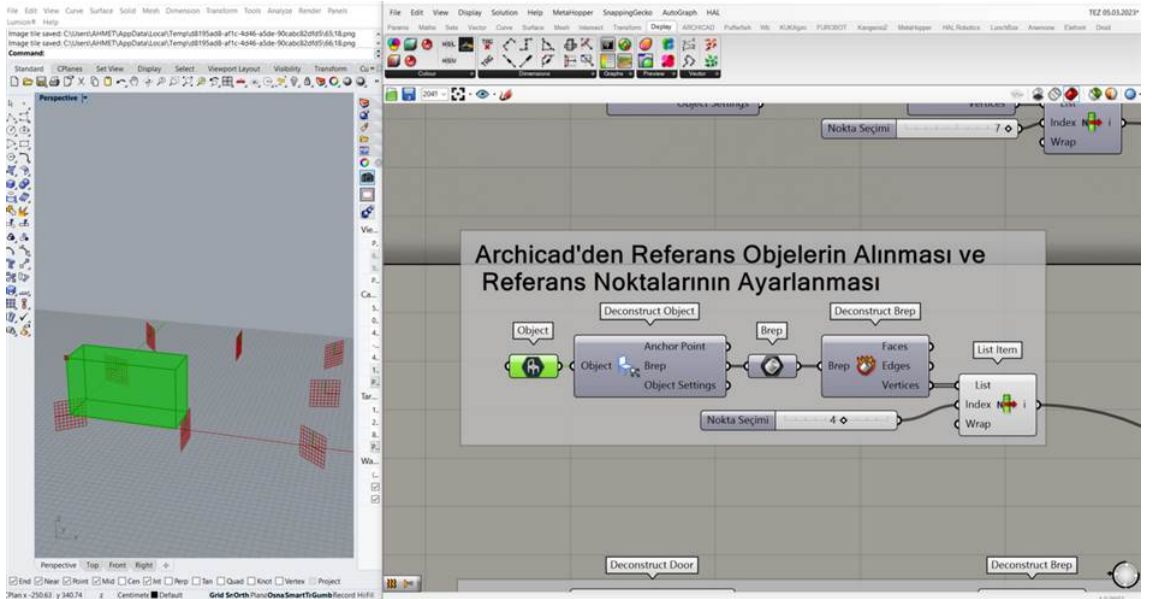
Barınma biriminin kabuk yapısı için çevreleyen eğrilerin düzlemleri elde edildikten sonra bu eğrilerin oluşumunun kurgusu için gerekli tefrişat elemanlarının Grasshopper'a aktarılması gerekmektedir. Bu adımda hedef referans objelerin Grasshopper'a aktartılıp eğrilerin kapsayacağı referans noktaları elde etmektir. Bu yüzden bu algoritma alt kümesi "ArchiCAD'den Referans Objelerin Alınması ve Referans Noktaların Ayarlanması" olarak adlandırılmıştır.

Şekil 5.11'de görüleceği üzere "Object" bileşenine ArchiCAD'de eklenen mutfak tezgâhı atanarak bu objenin verisinin Grasshopper'a aktarımı gerçekleştirilmiştir. "Deconstruct Object" bileşeni tanımlanan objenin verisinin "Brep" geometrisine dönüştürülmesini sağlamış, "Brep" geometrisinin çıktısı da "Deconstruct Brep" bileşenine girdi verisi olarak bağlanarak geometrinin "Vertices" çıktısında nokta verileri alınmıştır.



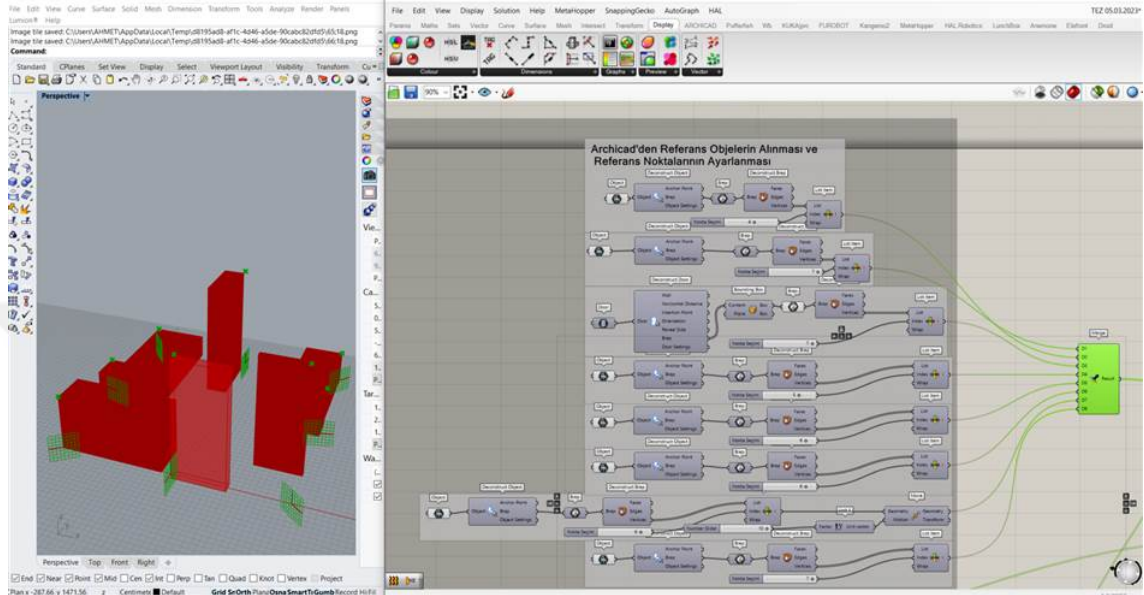
Şekil 5.12. ArchiCAD'deki objelerin Grasshopper'a aktarımı

Bu algoritma alt kümesinde yapılan işlemlerin devamında “Vertices” çıktısından alınan veriler “List Item” bileşenine girdi olarak bağlanıp çevreleyecek eğri ile ilişkili köşe “List Item” bileşeninin “Index” girdisine nokta seçimi olarak isimlendirilmiş “Number Slider” bileşeni ile gerekli nokta değeri bağlanarak elde edilmiştir. Şekil 5.13’de mutfak tezgahının Grasshopper’a aktarıldığında oluşan dikdörtgen prizma geometrisi, çevreleyecek eğri ile ilişkilendirilecek nokta verisi ve eğrilerin yerleşeceği düzlemler sol tarafta bulunan üç boyutlu görüntüleme arayüzünde gözlenmektedir.



Şekil 5.13. Tefrişat öğelerinin eğri için referans noktalarının belirlenmesi

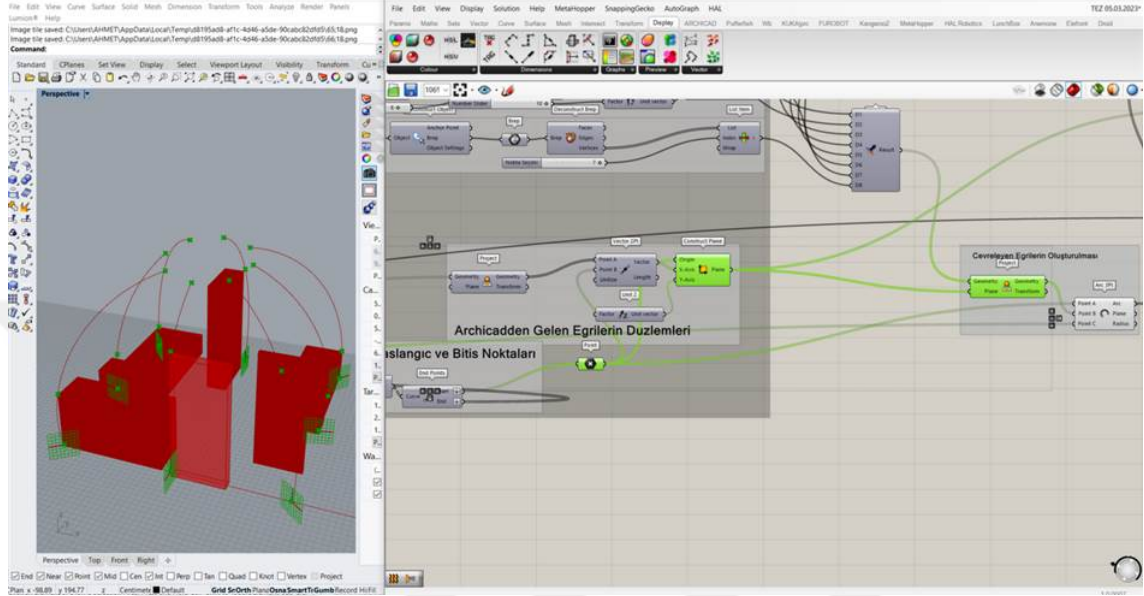
Aynı işlem referans eğrilerini belirlemek için barınma birimindeki ilgili objelerin her birinde tekrarlanmıştır. Tüm bu alt küme algoritmalarının çıktı verileri “Merge” bileşenine bağlanmıştır. Şekil 5.14’te objeleri kapsayan kutular kırmızı renkte, kapsayacak eğrileri belirleyecek olan noktalar ve bu eğrilerin yerleşeceği düzlemler yeşil renkte belirtilmektedir.



Şekil 5.14. Tüm objelerin Grasshopper’a aktarımı ve referans noktalarının belirlenmesi

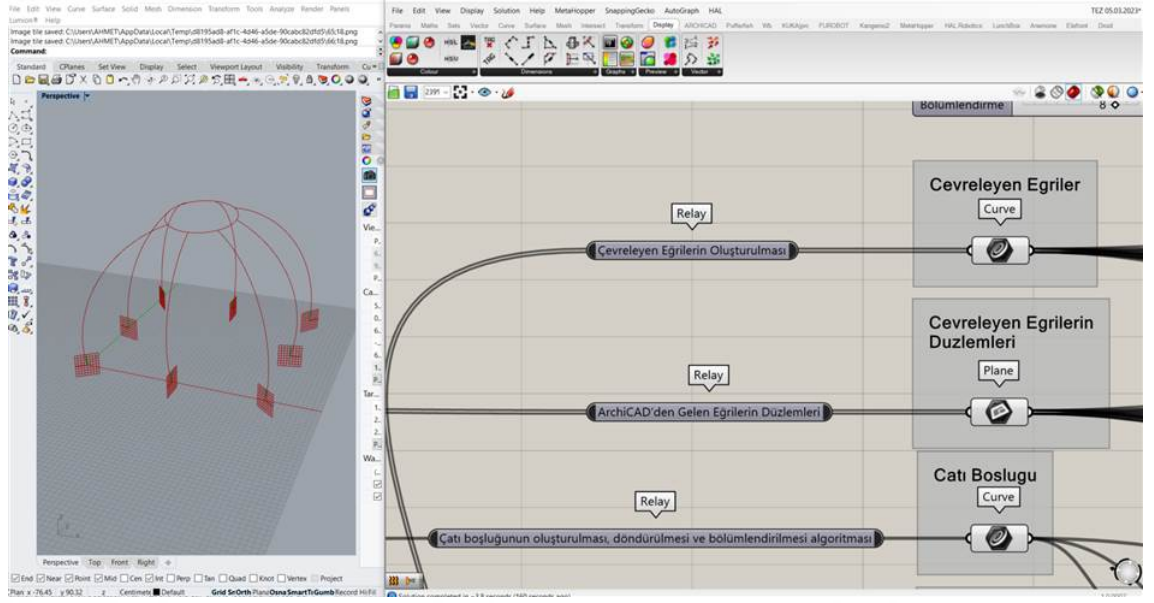
ArchiCAD’de oluşturulan eğrilerin başlangıç ve bitiş noktaları bunların yanı sıra ArchiCAD’de eklenen objelerin referans noktaları Grasshopper’a eklendikten sonra bu adımda amaç elde edilen verilerle Grasshopper’da parametrik olarak şekillenebilecek objeleri kapsayan yayları oluşturmaktır. Bu yüzden bu algoritma alt kümesi “Çevreleyen Eğrilerin Oluşturulması” olarak isimlendirilmiştir. Bu adımda öncelikle objelerin referans noktalarını eğrilerin düzlemlerine yerleştirmek için bir önceki adımdaki “Merge” bileşeninin çıktısı olan ve nokta kümesini oluşturan veriyi “Project” bileşeninin “Geometry” girdisine, eğrilerin yerleşecekleri düzlemleri de “Plane” girdisine bağlayarak referans noktaların gerekli düzlemlere yansıtılması gerçekleştirilmiştir. Bu bileşenin çıktı “Geometry” verisi “Arc 3Pt” bileşenini “Point B” girdisine, “ArchiCAD’de Oluşturulan Eğrilerin Başlangıç ve Bitiş Noktaları” alt kümesindeki başlangıç değerlerini veren “Start” çıktı verisi “Point A” bileşenine ve “Çatı Boşluğunun Oluşturulması, Döndürülmesi ve Bölümlendirilmesi” alt kümesinin “Points” çıktı verisi “Point C” bileşenine bağlanarak barınma biriminin iç kabuk yapısının formunu tefrişat öğelerini kapsayarak oluşturan eğriler Grasshopper’da parametrik olarak elde edilmiştir. Tefrişat

girdisi en başta ArchiCAD'den geldiği için ArchiCAD üzerinde yapılacak objenin konumunu boyutlarını değiştirme gibi değişiklikler doğrudan Grasshopper'a aktarılıp yapının formu üzerinde dinamik model değişikliğini sağlayacaktır. Şekil 5.15'te eğrileri oluşturan başlangıç, orta, bitiş noktaları ve eğrilerin yerleşeceği düzlemler yeşil renkte; objeler ve bu noktalara bağlı oluşan eğriler kırmızı renkte gözlenmektedir.



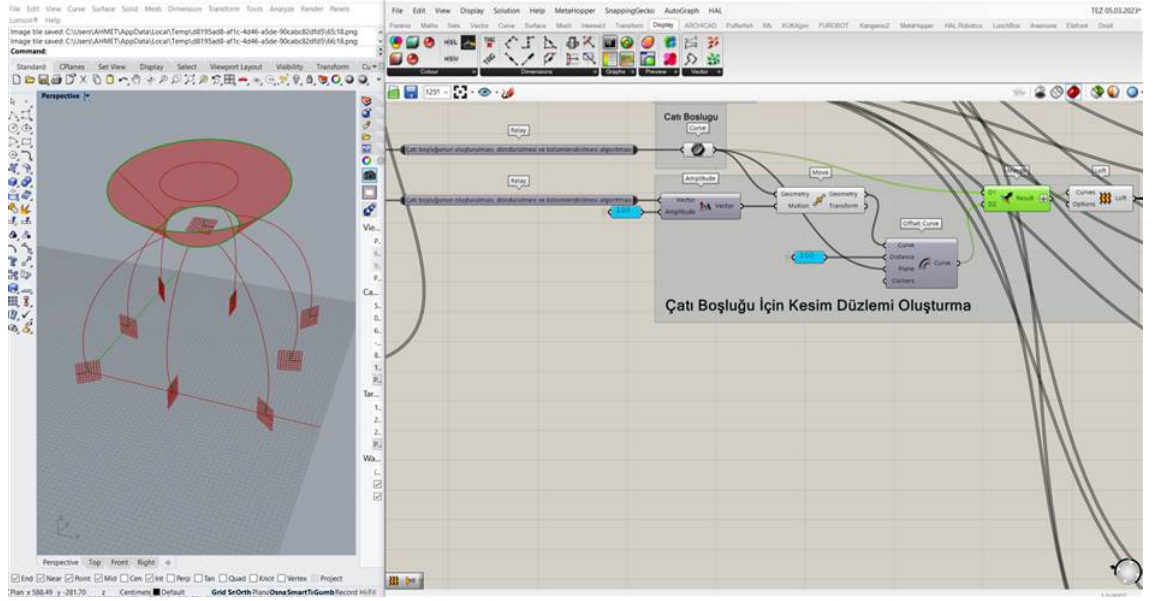
Şekil 5.15. Çevreleyen Eğrilerin Oluşturulması

“ArchiCAD'den Gelen Verilerin Grasshopper'a Aktarımı” üst algoritma kümesindeki işlemler tamamlandıktan sonra görsel kodlama arayüzünde düzeni sağlamak için Şekil 5.16'da görüleceği üzere gerekli algoritma alt kümelerinin çıktı verileri gerekli bileşenlere bağlanmıştır. Bu işlem tıpkı mimari çizim paftalarında nizamı sağlamak adına yapılan düzenlemeler olarak düşünülmelidir.



Şekil 5.16. Alt küme verilerinin bileşenlere atanması

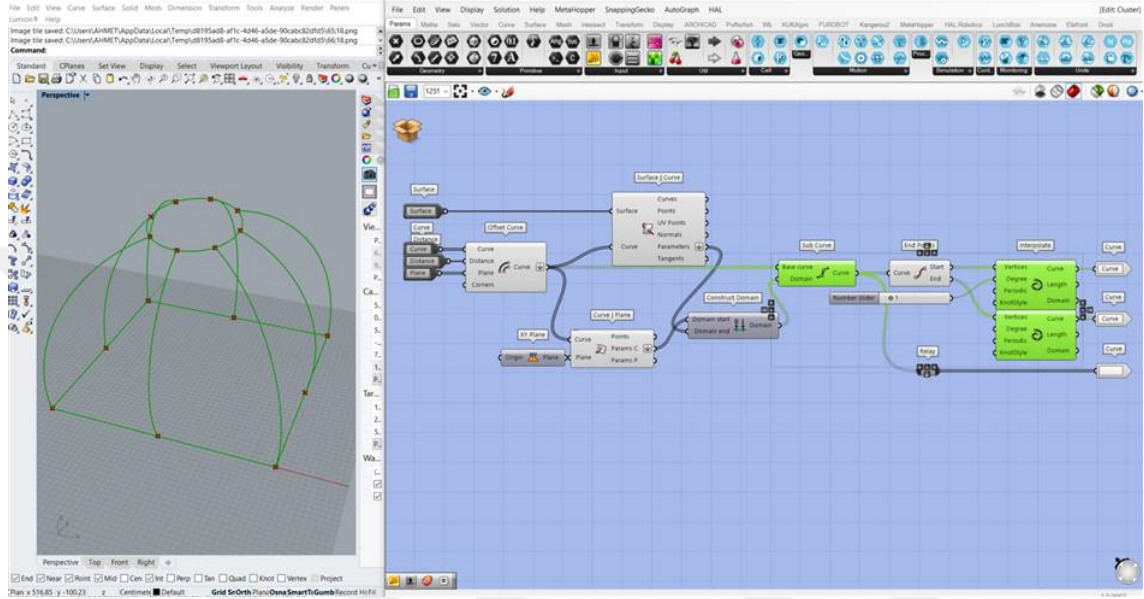
Bu adımda amaç iç kabuk yapısının geometrisinden, baskı için oluşturulacak takım yollarının genişletilmesini nizami bir şekilde yapmak için üst çatı örtüsünde parametrik modellenecek bir kesim düzlemi oluşturmaktır. Bu yüzden bu algoritma alt kümesi “Çatı Boşluğu İçin Kesim Düzlemi Oluşturma” şeklinde isimlendirilmiştir. Şekil 5.17’de görüldüğü üzere “Çatı Boşluğu” eğrisi öncelikle “Çatı Boşluğunun Oluşturulması, Döndürülmesi ve Bölümlendirilmesi” algoritma alt kümesinde elde edilen daire şeklinin çıktı verisi “Amplitude” bileşeninin “Vector” girdisine ve “Amplitude” girdisine ise 100 değeri girilerek gerekli referans vektörü elde edilmiştir. Bu referans vektörü dairenin kendi düzlemine dik hareket etmesini sağlayacaktır. Bu yüzden “Amplitude” verisinin çıktısı “Move” bileşeninin “Motion” girdisine, “Çatı Boşluğu” eğrisi ise “Geometry” girdisine bağlanarak dairenin kendi düzlemine dik olarak taşınması sağlanmıştır. Bu işlem sonrası elde edilen eğrinin çıktı verisi “Offset Curve” bileşeninin “Curve” girdisine bağlanmış, aynı bileşenin “Distance” girdisine 100 değeri ve “Plane” girdisine “Çatı Boşluğu” eğrisi bağlanmıştır. “Merge” bileşenine “Çatı Boşluğu” eğrisi ve “Offset Curve” bileşeninin çıktısı sırasıyla bağlanmış ve bu bileşenin çıktı verisi kesim düzlemi oluşturmak için yüzey oluşturmayı sağlayacak olan “Loft” bileşenine bağlanmıştır. Şekil 5.17’de çatı boşluğu eğrisini ve ötelenip genişletilmiş eğriyi yeşil renkte, kesim yüzeyi oluşturulan huni benzeri geometiyi kırmızı renkte gözlemlemekteyiz.



Şekil 5.17. Çatı boşluğu için kesim düzlemi oluşturma

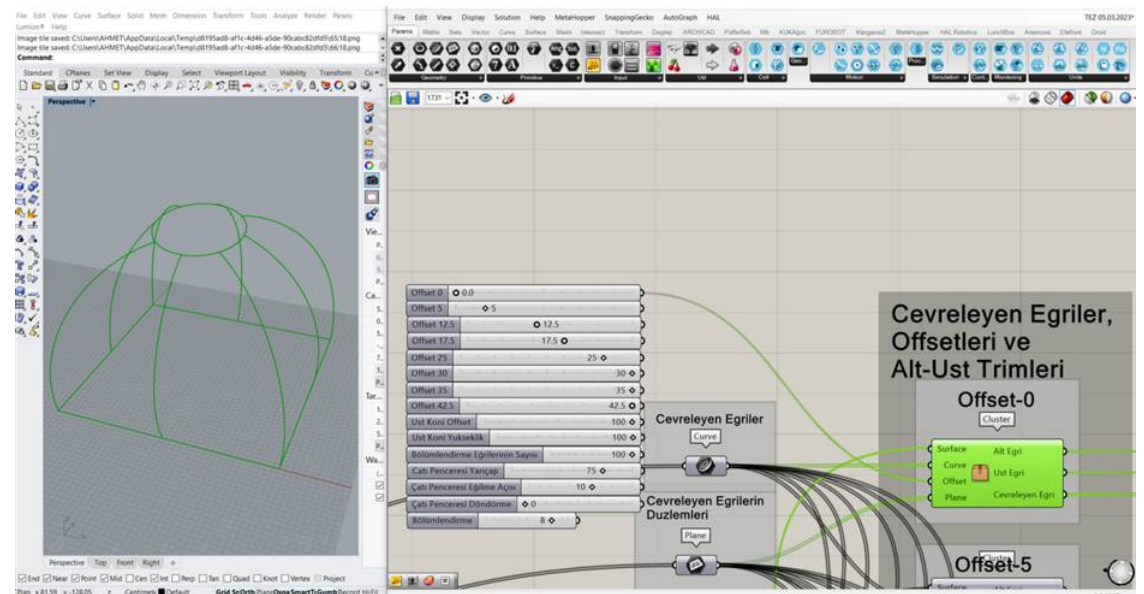
İç kabuk yapısının genişletilerek kabuğun strüktürünü ve dış kabuğunu oluşturmak için gerekli kesim düzlemi elde edildikten sonra bu adımda amaç gerekli genişletilme algoritmasını parametrik olarak elde etmektir. Bu yüzden bu adım “Çevreleyen Eğriler, Offsetleri ve Alt-Üst Trimleri” olarak adlandırılmıştır. “Offset Curve” bileşeninin “Curve” girdisine çevreleyen eğriler, “Distance” girdisine ilk genişletme parametresi olan 0 değeri “Plane” girdisine ise eğrilerin yerleşeceği düzlemler bağlanmıştır. Sonrasında nizami bir şekilde eğrileri kesmek için bir önceki adımda “Loft” bileşeninin çıktı verisi olan “Surface” verisi “Surface-Curve” bileşeninin “Surface” girdisine, bu adımdaki “Offset Curve” bileşeninin “Curve” çıktı verisi “Surface-Curve” bileşeninin “Curve” girdisine bağlanarak genişletilen eğrilerin yüzey ile kesiştikleri noktaların parametreleri elde edilmiştir. Çevreleyen eğrilerin çıktı verisi “Curve-Plane” bileşeninin “Curve” girdisine “XY Plane” bileşeni de “Plane” girdisine bağlanarak bu eğrilerin XY düzlemi ile kesişim noktalarının eğri üzerindeki parametresi elde edilmiştir. Bu parametreler elde edildikten sonra “Construct Domain” bileşeninin “Domain Start” ve “Domain End” girdilerine bu parametreler bağlanarak eğrilerin kesim parametrelerinin verisi elde edilmiştir. “Sub Curve” bileşeninin “Base Curve” girdisine “Offset Curve” bileşeninin çıktı verisi ve “Construct Domain” bileşeninin “Domain” çıktı verisi bağlanarak iç kabuk yapısını oluşturan eğrilerin nizami kesimleri elde edilmiştir. “Sub Curve” bileşeninin çıktı verisi “End Points” bileşenini girdisine bağlanarak bu eğrilerin başlangıç ve bitiş noktaları elde edilmiştir. “End Points” bileşeninin “Start” ve “End” çıktı verisi “Interpolate” bileşeninin “Vertices” bileşenine bağlanarak çevreleyen eğrileri

kapsayan zemindeki ve çatı boşluğundaki sınırlar elde edilmiştir. Bu algoritma alt kümesi Grasshopper’da “Cluster” işlevi çalıştırılarak gruplandırılmış genel algoritmada tek bir bileşen olarak görünmesi sağlanmıştır. Bunun sebebi aynı işlemin diğer genişletme değerlerinde devam ettirilecek olup görsel kodlama arayüzünde düzeni sağlamaktır. Şekil 5.18’de bu algoritma alt kümesinin çıktı verileri sol taraftaki Rhino üç boyutlu görüntüleme arayüzünde yeşil renkte görülmektedir.



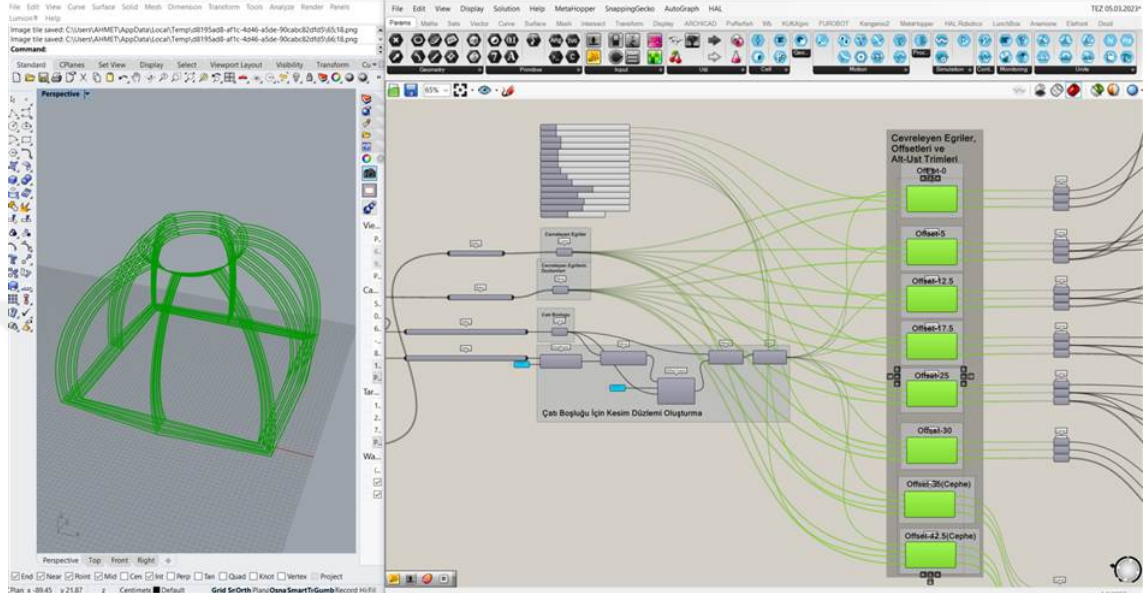
Şekil 5.18. Çevreleyen eğriler, offsetleri ve alt-üst trimleri kümesi

Şekil 5.19’da bir önceki adımda oluşturulan algoritmanın “Cluster” işlevi çalıştırıldıktan sonra bir Grasshopper bileşeni gibi görüldüğü gözlenmektedir.



Şekil 5.19. Cluster işlevi çalıştırıldıktan sonra algoritmanın bileşene dönüşümü

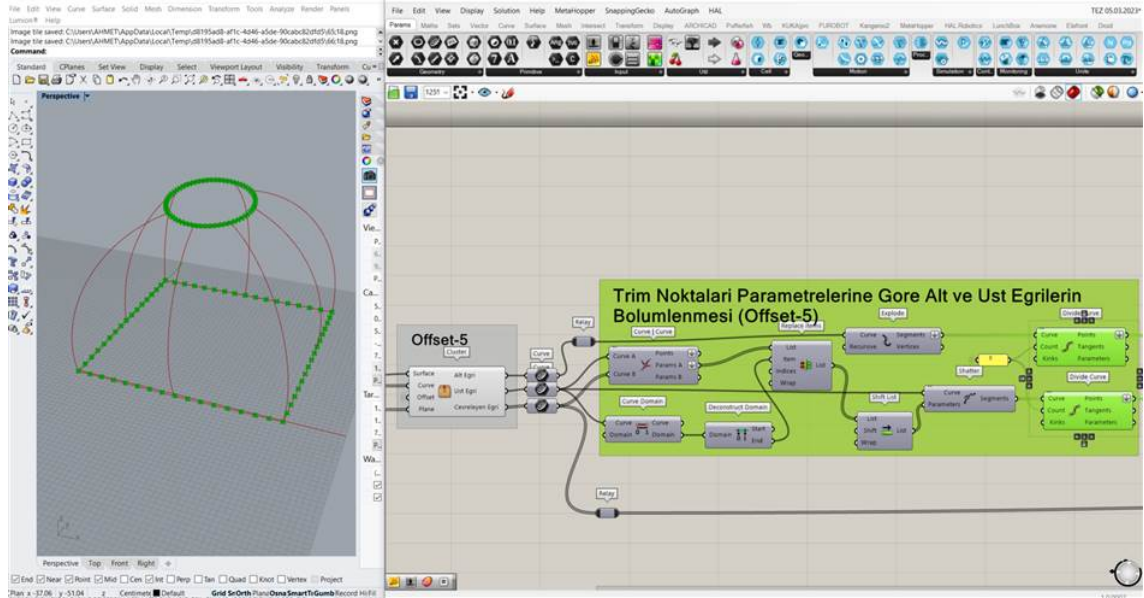
Şekil 5.20’de Şekil 5.19’da sol taraftaki “Number Slider” bileşenindeki parametrelerin dağıtıldığı ve aynı işlem tekrar edildiği için görsel kodlama arayüzünde “Cluster” işlevinin nizamı sağlamada getirdiği fayda ve yapının kabuğunun katmanlarının oluşturulması için iç kabuk eğrilerinden genişletilen eğrilerin oluşturduğu şekil Rhino üç boyutlu görüntüleme arayüzünde yeşil renkte gözlemlenmektedir.



Şekil 5.20. Çevreleyen eğriler, offsetleri ve alt-üst trimleri çıktısı

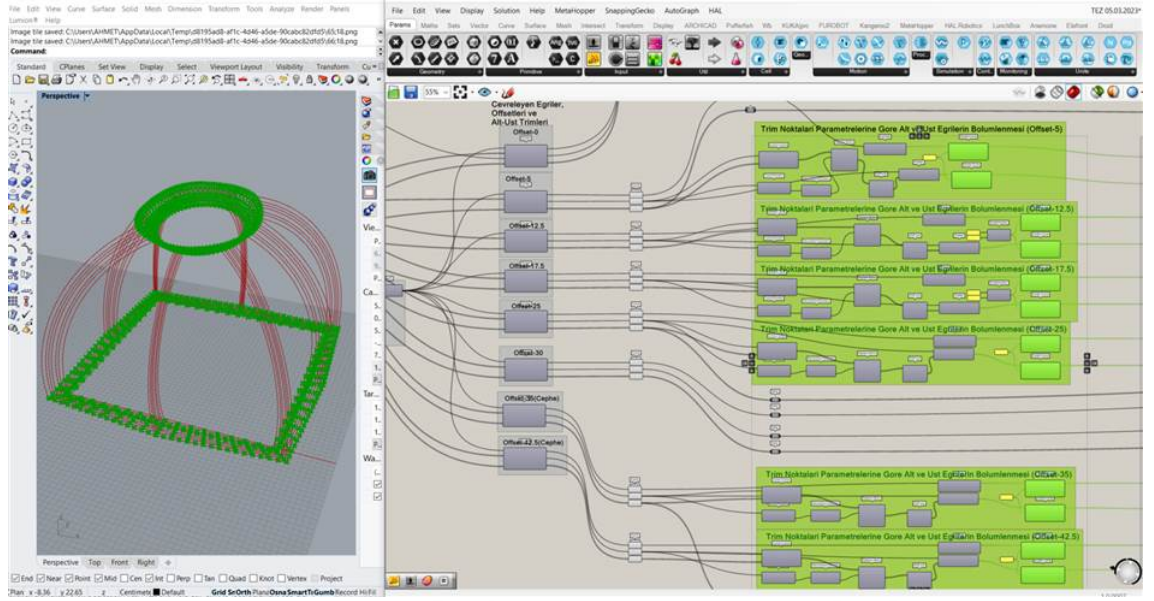
Barınma biriminin iç kabuk yapısını referans alan ve buna bağlı olarak genişletilen katmanları oluşturulduktan sonra bu adımda amaç bu katmanları birimin duvar strüktürlerinin algoritmalarını oluşturmak için hazır hale getirmektir. Bu sebeple bu algoritma alt kümesi “Trim Noktaları Parametrelerine Göre Alt ve Üst Eğrilerin Bölümlendirilmesi” olarak adlandırılmıştır. Önceki adımda “Cluster” işlevinin çıktıları olan ve “Offset-5” kümesine dahil olan “Alt Eğri”, “Üst Eğri” ve “Çevreleyen Eğri” görsel kodlama arayüzünde nizamı sağlamak adına “Curve” bileşenine bağlanmıştır. “Curve-Curve” bileşenine “Üst Eğri” ve “Çevreleyen Eğri”nin girdileri bağlanarak ilgili eğrilerin kesişim noktalarının parametreleri elde edilmiştir. “Üst Eğri” çıktısı “Curve Domain” bileşeninin “Curve” girdisine bağlanmış ve bu bileşenin “Domain” çıktı verisi “Deconstruct Domain” bileşenine bağlanarak “Üst Eğri” çıktısının bitiş noktasının parametre verisi elde edilmiştir. “Deconstruct Domain” bileşeninin “End” çıktısı “Replace Items” bileşeninin “Item” girdisine, “Curve-Curve” bileşeninin “Params A” çıktısı ise “List” girdisine bağlanmıştır. “Replace Items” bileşeninin çıktı verileri kontrol edildiğinde parametre değerlerinin doğru olduğu fakat listenin bir sıra kayması gerektiği

gözlemlenmiştir. Bu yüzden “Replace Items” bileşeninin çıktı verisi “Shift List” bileşeninin “List” girdisine bağlanarak liste 1 sıra kaydırılmıştır. “Shatter” bileşeninin “Curve” girdisine “Üst Eğri” çıktısı ve “Parameters” girdisine “Shift List” bileşeninin “List” çıktısı bağlanarak “Üst Eğri” ve “Çevreleyen Eğri”nin kesişimleri uygun biçimde elde edilmiştir. “Alt Eğri” çıktısı “Explode” bileşeninin “Curve” girdisine bağlanarak eğrinin kesişim noktalarına göre parçalanan sekiz farklı eğri elde edilmiştir. “Explode” ve “Shatter” bileşenlerinin “Segments” çıktıları ayrı ayrı “Divide Curve” bileşeninin “Curve” girdilerine, “Count” girdisine ise bölümlendirme sayısı olan 8 rakamı bağlanarak birimin strüktürünün oluşması için gerekli olan noktalar “Alt Eğri” ve “Üst Eğri”de elde edilmiştir. Şekil 5.21’de bu algoritma alt kümesinin girdi verileri olan eğriler kırmızı renkte, çıktı verileri olan noktalar ise yeşil renkte Rhino üç boyutlu görüntüleme arayüzünde gözlemlenmektedir.



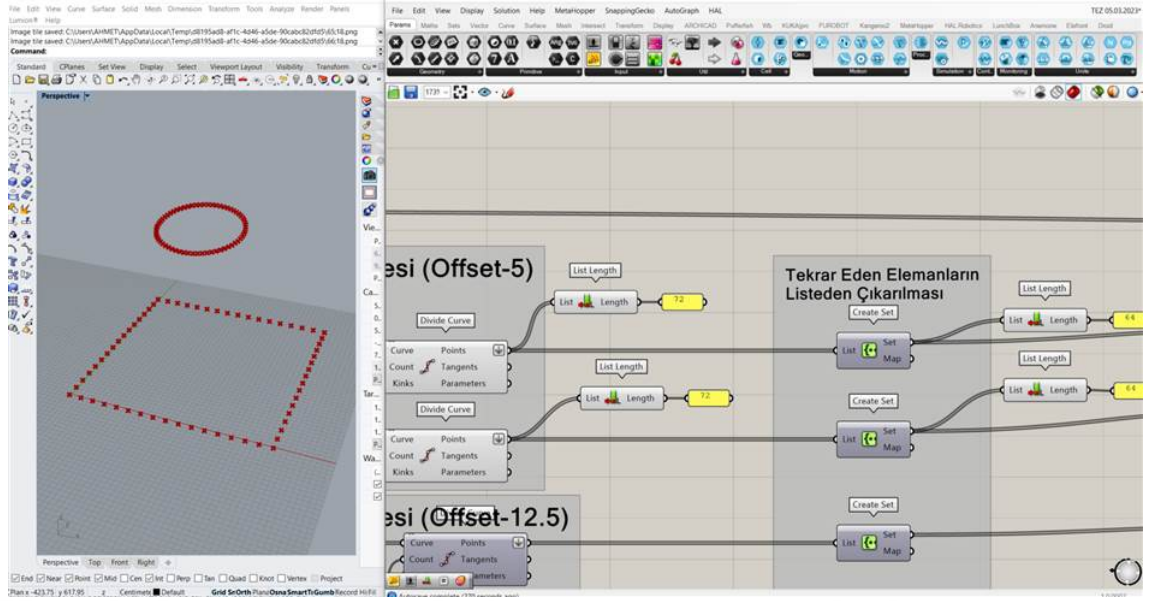
Şekil 5.21. Trim noktaları parametrelerine göre alt ve üst eğrilerin bölümlendirilmesi

Aynı işlem sadece strüktür kümelerinde 8 değil 16 farklı bölümlendirme ile gerekli tüm genişletme değerleri için tekrar edilmiştir. Bu sayede bu algoritma alt kümesi adımında iç kabuk, kabuk strüktür ve dış kabuk yapısı için gerekli noktalar elde edilmiştir. Şekil 5.22’de bu adımın girdi verileri kırmızı renkte çıktı verileri yeşil renkte Rhino üç boyutlu görüntüleme arayüzünde gözlenmektedir.



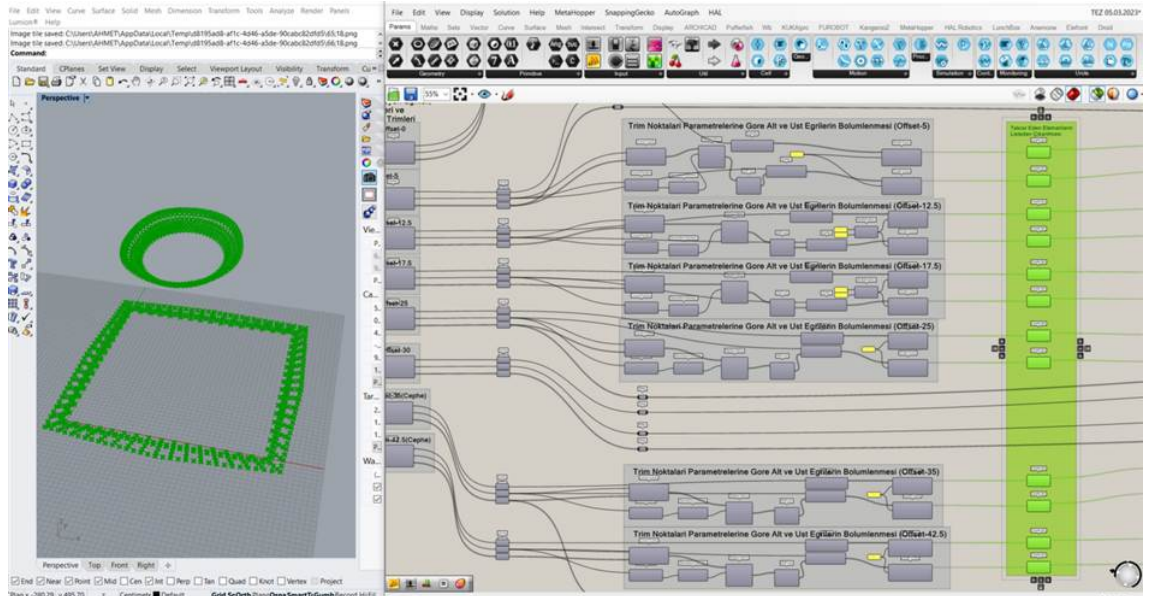
Şekil 5.22. Trim noktaları parametrelerine göre tüm alt ve üst eğrilerin bölümlendirilmesi

“Trim Noktaları Parametrelerine Göre Alt ve Üst Eğrilerin Bölümlendirilmesi” çıktı verileri kontrol edildiğinde 8 farklı çevreleyen eğriye bağlı 8 kesişim noktası ve bunların 8’e bölünmesi sonucu elde edilmesi gereken toplam 64 farklı noktaya ulaşmamız gerekirken listelerin başında ve sonunda tekrar eden nokta verilerinden dolayı bu alt kümenin çıktısı 72 adet nokta verisi verdiği gözlemlenmiştir. Bu yüzden bu algoritma alt kümesi “Tekrar Eden Elemanların Listeden Çıkarılması” olarak adlandırılmıştır. “List Length” bileşeni bir listede kaç adet veri olduğunu göstermektedir. Bu sorunu çözmek için önceki adımdaki alt kümenin çıktı verileri “Create Set” bileşeninin “List” girdisine bağlanarak tekrar eden elemanların liste sırasını bozmadan listeden çıkarılmasını sağlamıştır. Şekil 5.23’de görüleceği üzere girdi verilerindeki eleman sayısı 72 iken bu adımdaki çıktı verilerindeki eleman sayısı 64 olmuştur.



Şekil 5.23. Tekrar eden elemanların listeden çıkarılması

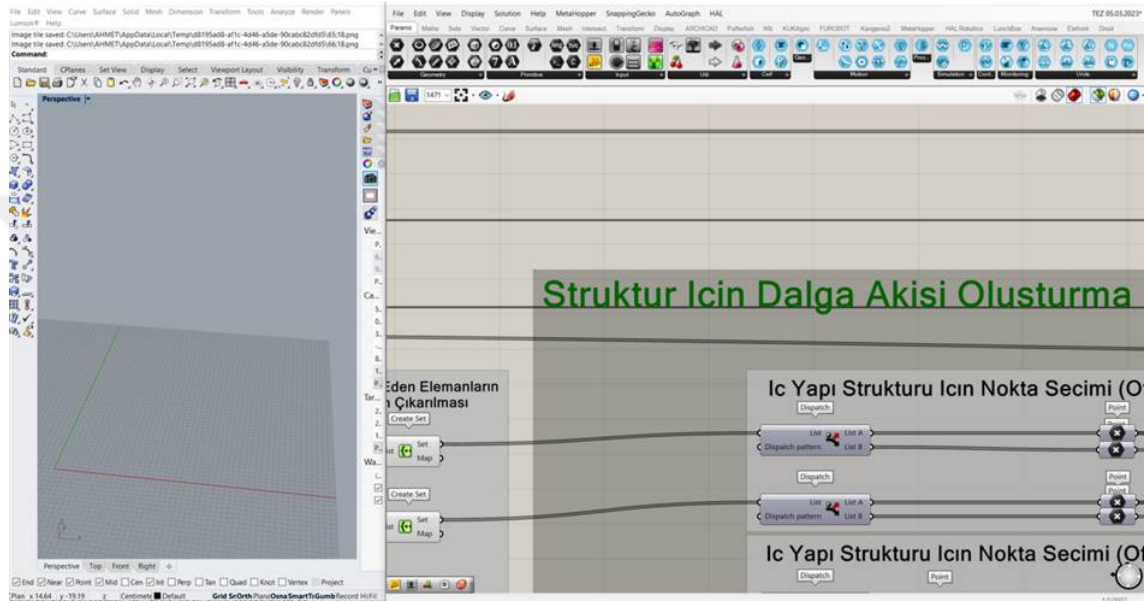
Aynı işlem bu algoritma alt kümesine tanımlı tüm genişletme değerleri için tekrar edilmiştir. Bu işlem sonucunda tekrar eden tüm nokta verileri liste sırası bozulmadan kaldırılmıştır. Şekil 5.23’de sağ tarafta bu algoritma alt kümesi son tarafta ise tüm elemanlar için oluşan çıktı verisi olan noktaların Rhino üç boyutlu görüntüleme arayüzündeki görüntüsü gözlenmektedir.



Şekil 5.24. Tüm tekrar eden elemanların listeden çıkarılması

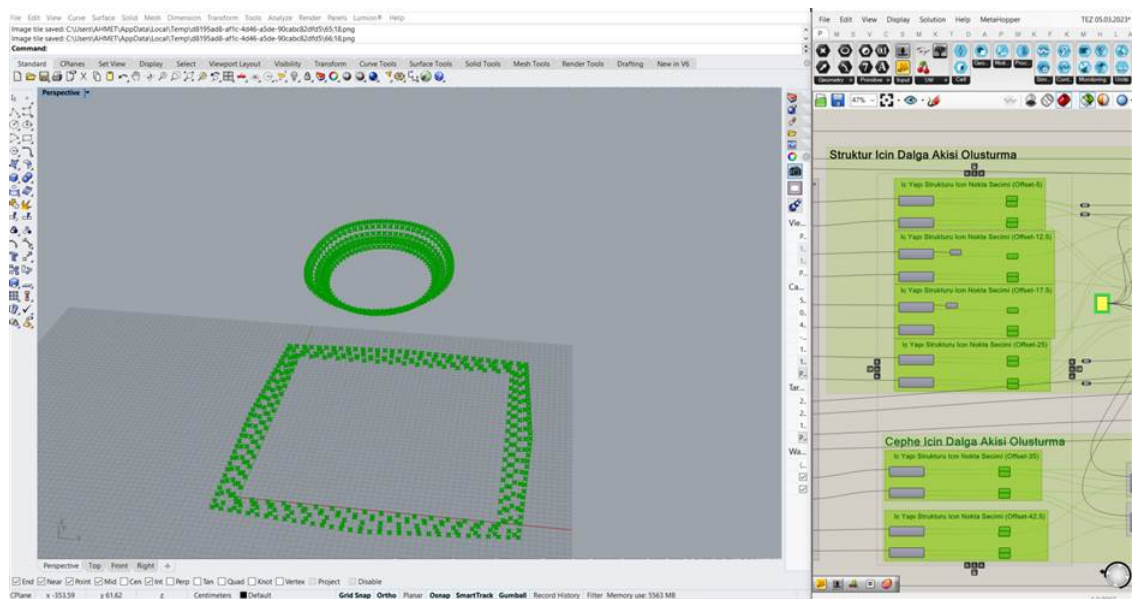
“Trim Noktaları Parametrelerine Göre Alt ve Üst Eğrilerin Bölünmesi” alt kümesinin çıktılarında iç kabuk ve kabuk strüktürünün oluşturulacağı veriler elde edildikten sonra sıradaki adım bu verileri eşleştirmek üzere kurgulanmıştır. Bu adımda

amaç veri setlerini organize ederek bir akış oluşturmaktır. Bu yüzden bu genel algoritma kümesi “Strüktür İçin Dalga Akışı Oluşturma” olarak adlandırılmıştır. Şekil 5.25’de görüleceği üzere bir önceki adımda “Create Set” bileşeninin çıktı verileri “Dispatch” bileşeninin “List” girdisine atanarak “Dispatch” bileşeni listeyi (True-False) kuralıyla ardışık elemanları iki farklı listeye bölmüştür. “Dispatch” bileşeninin çıktıları görsel kodlama arayüzünün düzenini sağlamak adına sırası ile iki farklı “Point” bileşenine bağlanmıştır.



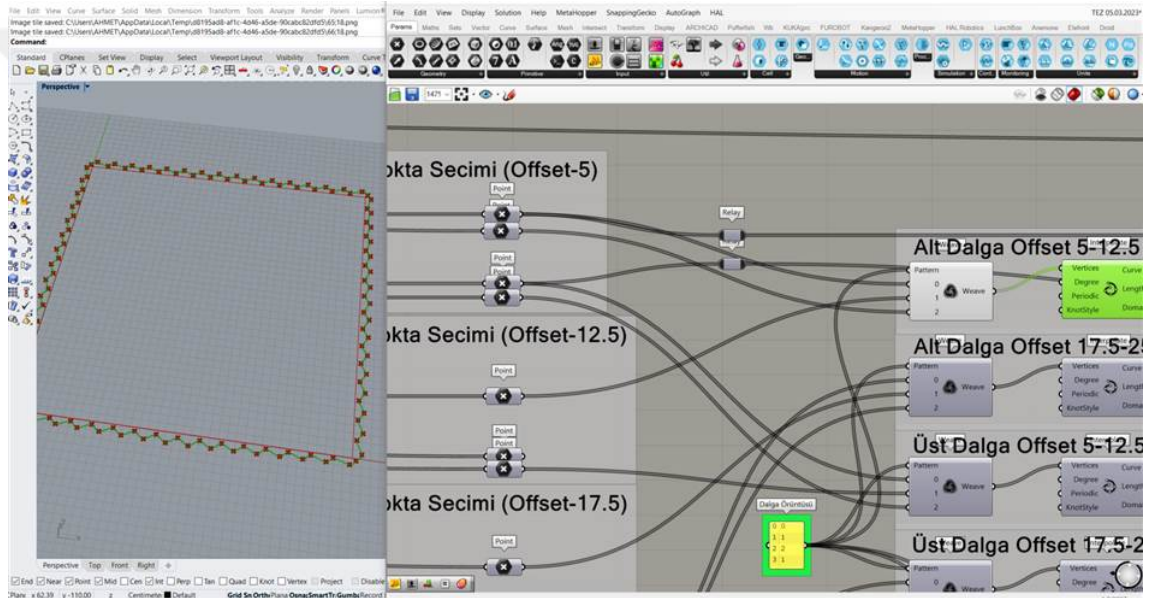
Şekil 5.25. Strüktür için dalga akışı oluşturma

Önceki adımdaki işlem tüm genişletme değerleri için tekrar edilmiştir (Şekil 5.26).



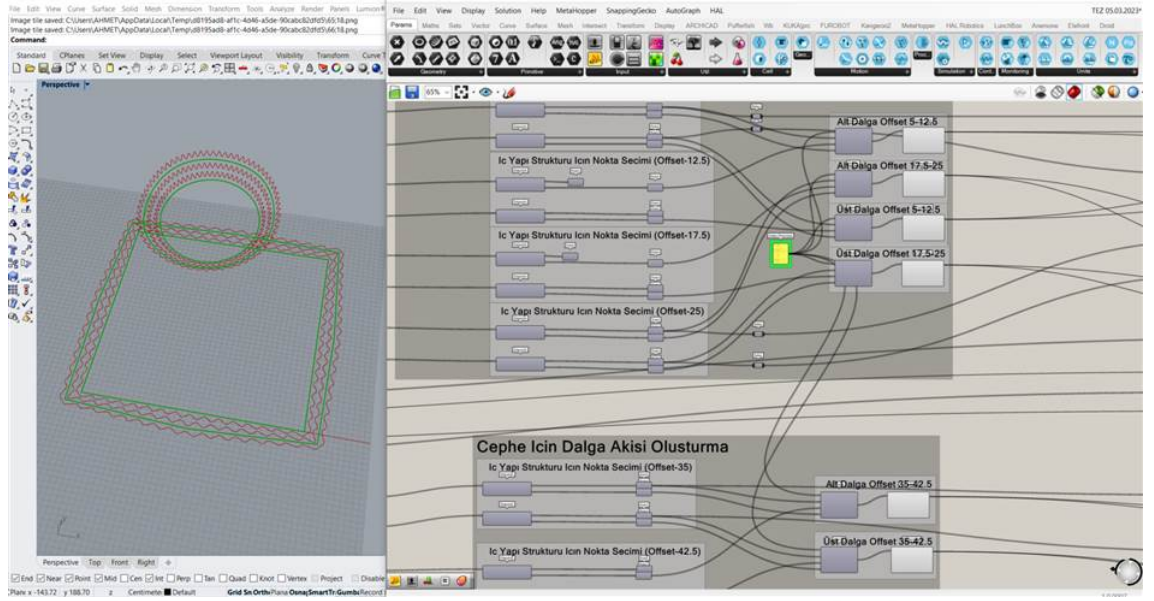
Şekil 5.26. Tüm strüktür için dalga akışı oluşturma

Önceki adımda nokta listelerinin (True-False) kuralı ile bölünmesinden sonra bu adımda amaç kabuk yapısının strüktürünü oluşturacak eğrileri elde etmektir. Bu yüzden bu algoritma alt kümesi “Alt Dalga Offset” olarak adlandırılmıştır. Önceki adımdaki “Dispatch” bileşeninin gerekli çıktıları “Weave” bileşeninin parametre girdilerine, istenilen örüntü de “Dalga Örüntüsü” bileşenine kural olarak belirlenerek “Pattern” girdisine atanmıştır. Şekil 5.27’de görüleceği üzere üç farklı nokta listesinin gerekli kurallar ile birleştirilerek parametrik olarak şekillenebilecek bir kabuk strüktürü başlangıç eğrisi oluşturulmuştur. “Pattern” bileşenini “Weave” çıktısı “Interpolate” bileşeninin “Vertices” girdisine bağlanarak hedeflenen strüktür eğrisi tek bir eğri şeklinde elde edilmiştir.



Şekil 5.27. Alt dalga offset

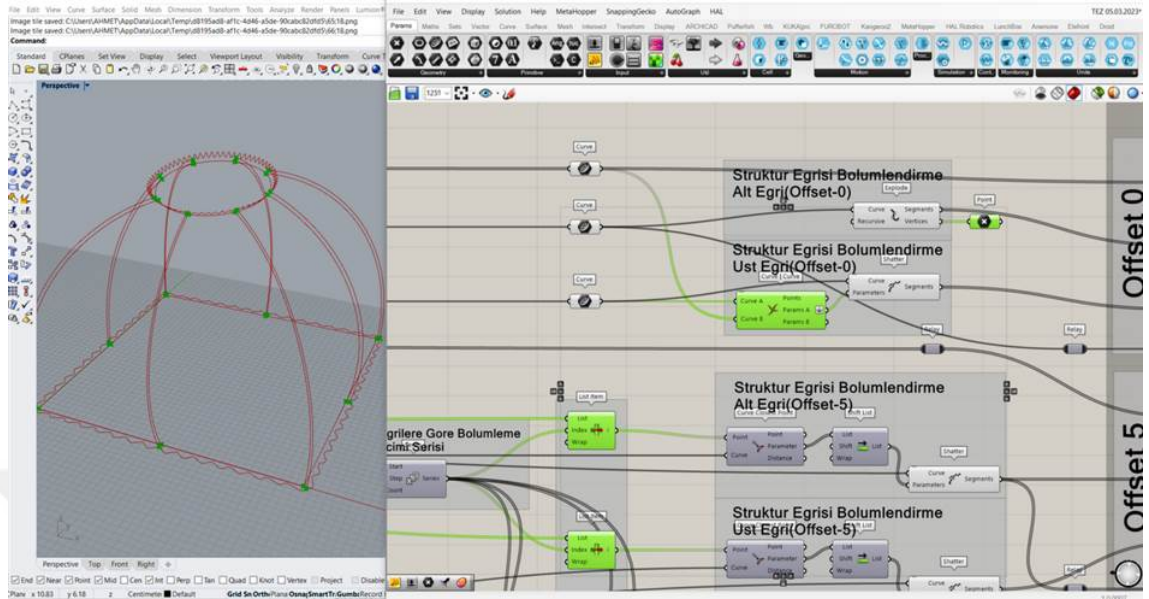
Bu işlem aynı kurallar dahilinde gerekli nokta veri setleri seçilerek tüm genişletme değerleri için tekrar edilmiştir. Şekil 5.28’de bu algoritma alt kümesi sonucunda elde edilen geometri Rhino üç boyutlu görüntüleme arayüzünde gözlenmektedir. Bu geometride yeşil renkte beliren eğriler yapı kabuğunun duvar olarak nitelendirilebilecek sınırlarını, kırmızı olanlar ise strüktürü oluşturacak yüzeyleri temsil etmektedir.



Şekil 5.28. Tüm eğriler için alt dalga offset

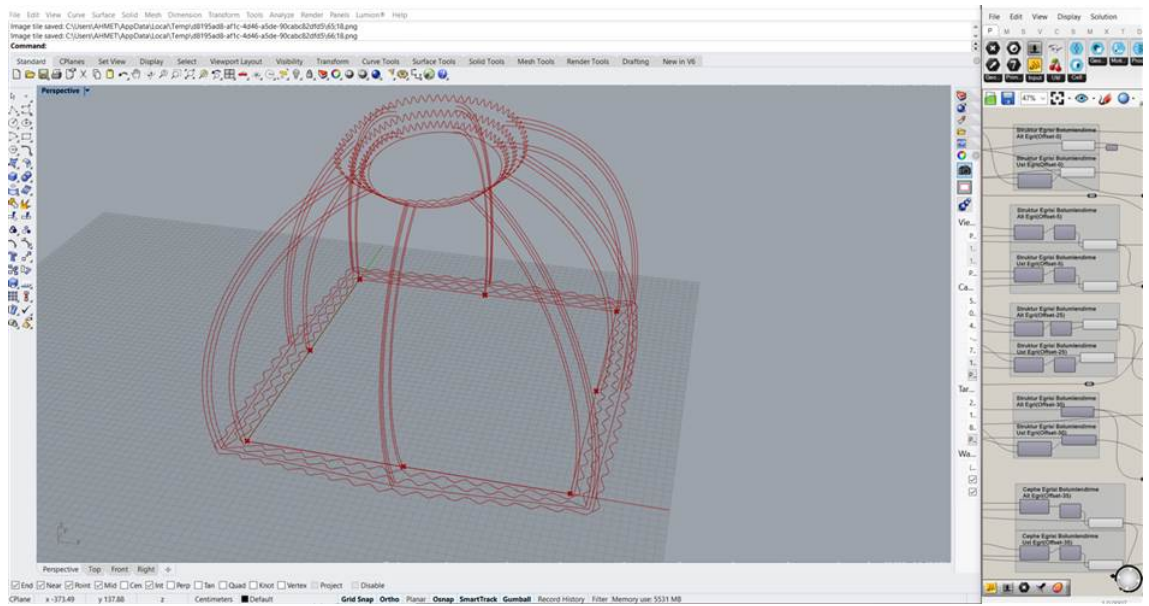
Zemin ve çatı kısmı için gerekli eğriler parametrik olarak oluşturulduktan sonra bu adımda amaç bu eğrileri kapsayacak yüzeyler oluşturmak için bu eğrileri bölümlenmektir. Bu yüzden bu algoritma alt kümesine “Strüktür Eğrisi Bölümlendirme” adı verilmiştir. Şekil 5.29’da görüldüğü gibi iç kabuk yapısını ve iç kabuk yapısına en yakın olan “Offset-5” eğrileri görünmektedir. İç kabuk yapısında bu eğrileri sırasıyla yüzeyler oluşturmak için “Alt Eğri” çıktısı “Explode” bileşeninin “Curve” girdisine bağlanmıştır. Bu işlem sonucu uygun sırayla gerekli parçalanmış eğriler elde edilmiştir. “Üst Eğri” çıktısını uygun bir şekilde bölümlendirmek için ise bu eğrinin çıktısı ve “Çevreleyen Eğri”nin çıktı verisi “Curve-Curve” bileşenine atanarak kesişim noktalarının parametreleri bu bileşenin “Params A” çıktısında elde edilmiştir. “Shatter” bileşeninin “Curve” girdisine “Üst Eğri”nin çıktısı, “Parameters” girdisine ise “Params A” çıktısı bağlanarak iç kabuk yapısının çatı ile ilişkilenecek kısmı da uygun bir şekilde bölümlenmiştir. Bu işlem aynı noktaya ulaşmak için dalga akışı oluşturan ve iç kabuğa en yakın katman olan “Offset-5” veri seti içinde farklı yöntemle aynı sonuca ulaşmak için yapılmıştır. Bu sonuca ulaşmak için yapılması gereken farklı işlem ise nokta seçimlerinin doğru yapılabilmesi için bir seri oluşturmaktır. Bunun için “Series” bileşeninin “Start”, “Step” ve “Count” girdilerine sırası ile 0, 4 ve 8 rakamları bağlanmıştır. Bu sayede (0, 4, 8, ..., 28) örüntüsü elde edilmiştir. Bu örüntü eğrilerin uygun noktalarda bölümlendirilmesi için eğrilerin nokta setlerinden seçim yapmayı sağlayacak olan “List Item” bileşenine bağlanmıştır. Bu işlemler sonucunda iç kabuk yapısını ve ona en yakın katmanındaki eğrilerin yüzey oluşumu için gerekli organizasyonunun algoritması oluşturulmuştur. Şekil 5.29’da

görülebileceği üzere yüzey oluşturmak için gerekli eğriler kırmızı renkte ve eğrilerin bölümlendirileceği noktalar yeşil renkte gözlenmektedir.



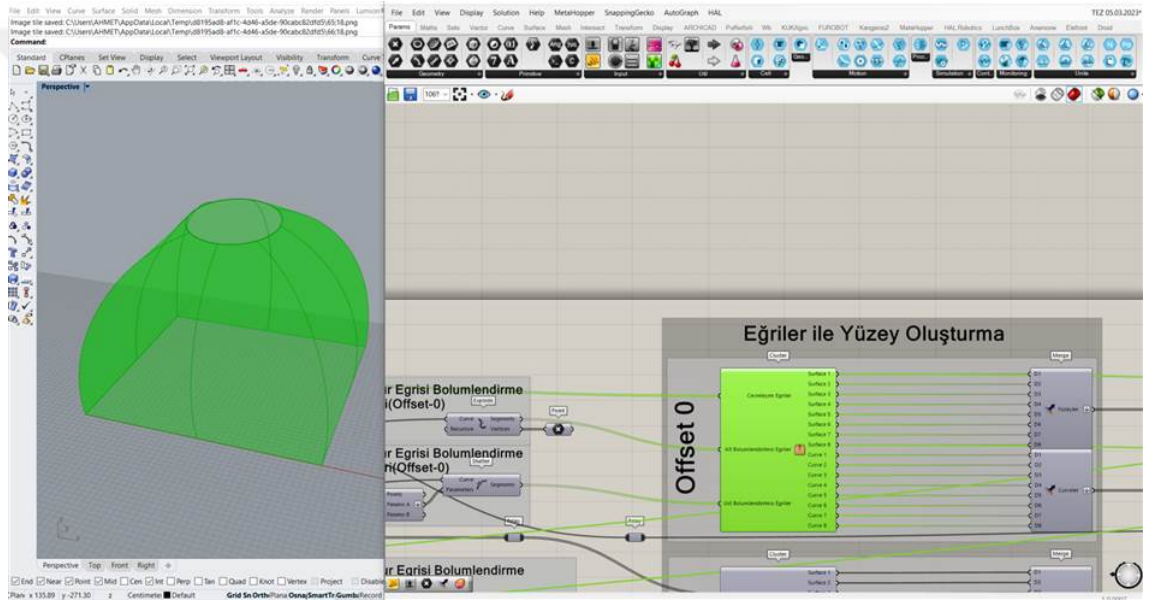
Şekil 5.29. Strüktür eğrisi bölümlendirme

“Strüktür Eğrisi Bölümlendirme” algoritma alt kümesinde yapılan işlemler barınma biriminin iç kabuk, strüktür ve dış kabuk yüzeylerini oluşturmak için tekrar edilerek organize edilmiştir. Bu işlemler sonucunda elde edilen çıktı verileri Şekil 5.30’da Rhino üç boyutlu görüntüleme arayüzünde gözlenmektedir.



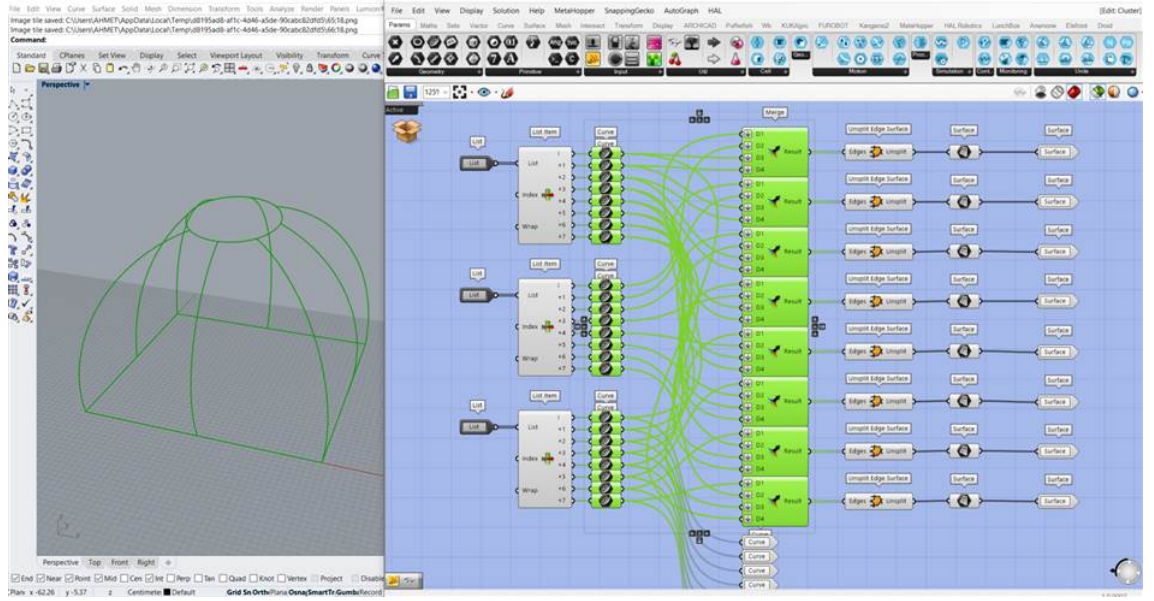
Şekil 5.30. Strüktür eğrisi bölümlendirme tüm genişletmeler

Önceki adımda geliştirilen algoritmanın çıktıları düzenlenmiş şekilde eğriler ile yüzeyler oluşturmayı mümkün kılmaktadır. Bu adımda amaç bu organize edilmiş eğriler ile kabuğun iç ve dış yapısını ve strüktürünü oluşturacak yüzeyleri elde etmektir. Bu yüzden bu algoritma kümesi “Eğriler ile Yüzey Oluşturma” olarak adlandırılmıştır. Şekil 5.31’de görüleceği üzere önceki adımın çıktıları olan “Alt Bölümlendirilmiş Eğriler” ve “Üst Bölümlendirilmiş Eğriler” Grasshopper’da “Cluster” işlevi çalıştırılarak elde edilen bileşende gerekli yerlere bağlanmıştır. Bir sonraki adımda görüleceği üzere “Cluster” işlevi görsel kodlama arayüzünde nizamı sağlamak adına oluşturulmuştur. Bu “Cluster”ın çıktısı verilerini her bir genişletme kümesine bağlı oluşan ayrı ayrı yüzeyler oluşturmaktadır.



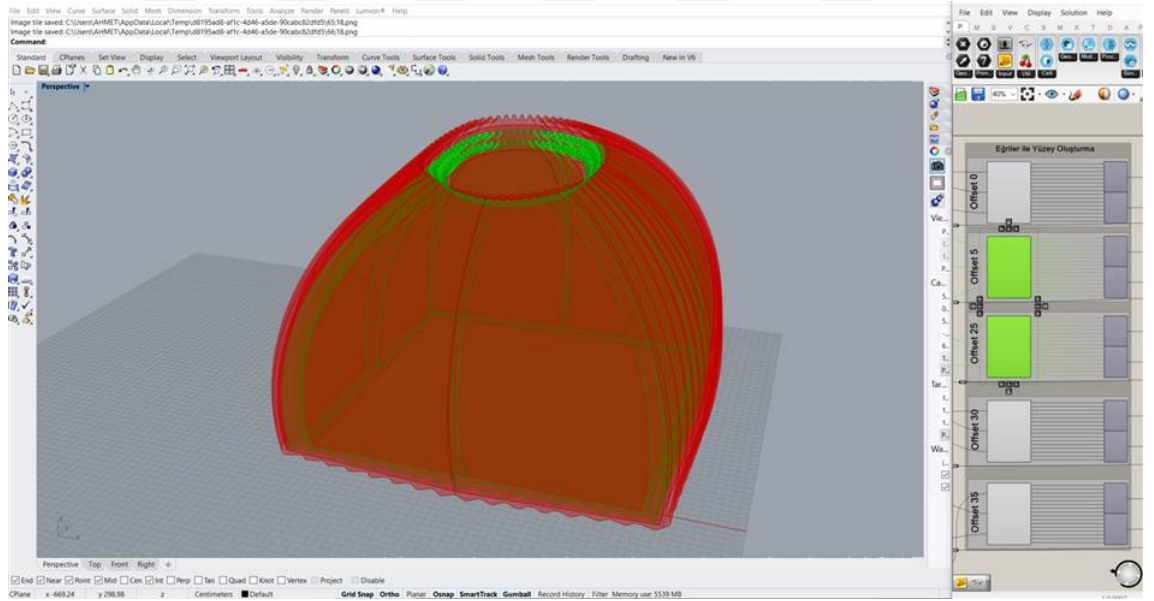
Şekil 5.31. Eğriler ile Yüzey Oluşturma

“Eğriler ile Yüzey Oluşturma” kümesinde yapılan işlemlerin temel bileşeni olan “Cluster”ın algoritması Şekil 5.32’de gözlenmektedir. Buna göre “Alt Eğri”, “Üst Eğri” ve “Çevreleyen Eğri”lerin çıktıları uygun sıra ile “Merge” bileşenlerine atanmış ve her bir “Merge” bileşeninin çıktısı “Unsplit Edge Surface” bileşenine atanarak sırası ile iç kabuk yapısını oluşturan sekiz farklı yüzey elde edilmiştir.



Şekil 5.32. Eğriler ile yüzey oluşturma cluster algoritması

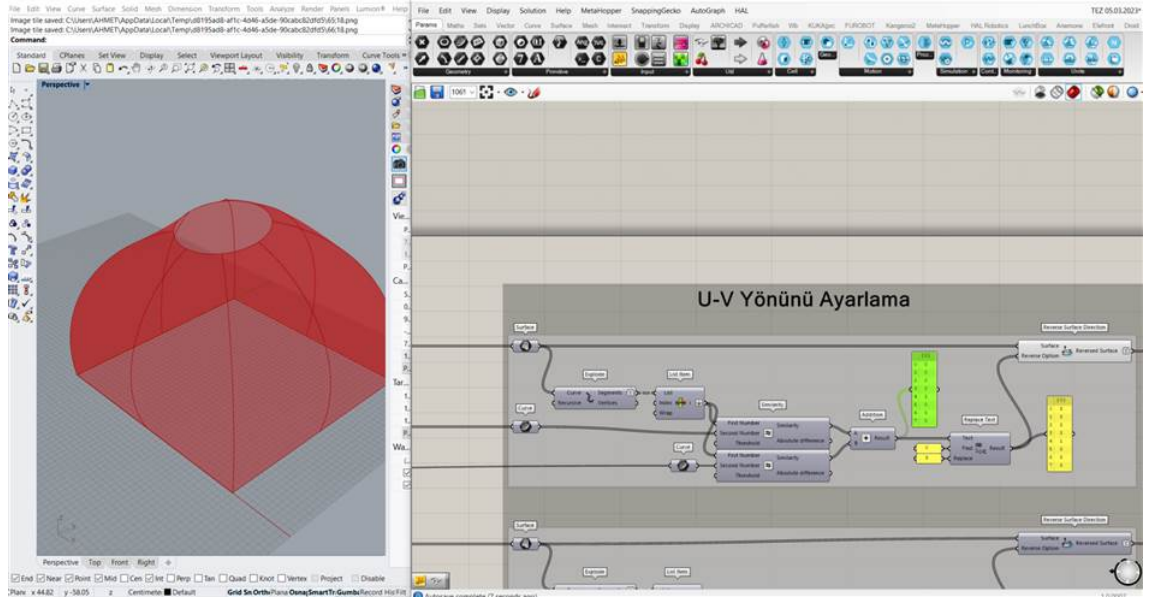
“Eğriler ile Yüzey Oluşturma” algoritma kümesi tüm genişletme değerlerinin çıktılarına tekrar edilmiş ve Şekil 5.33’te görüldüğü üzere barınma biriminin kabuk ve strüktür yapılarını oluşturan tüm yüzeyler parametrik olarak elde edilmiştir.



Şekil 5.33. Eğriler ile yüzey oluşturma tüm yüzeylerin oluşumu

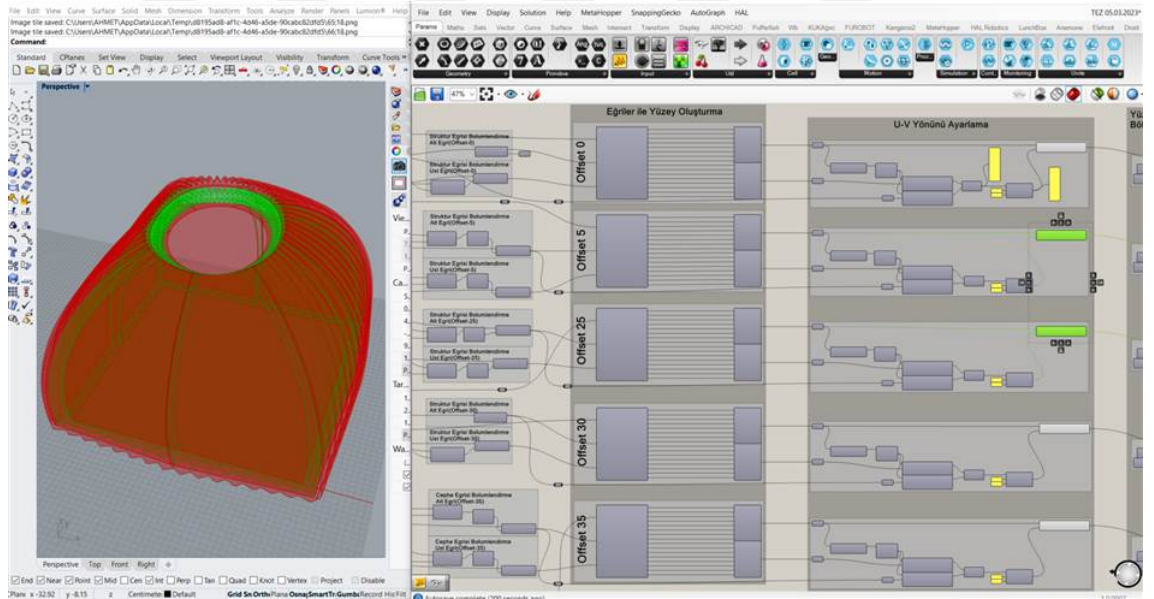
“Eğriler ile Yüzey Oluşturma” algoritma kümesinin çıktıları olan “Yüzeyler” ve “Curveler” bu adımda yüzeylerin aynı yönde olması adına yapılacak işlemler için referans parametleridir. Bu adımda amaç üç boyutlu baskılama için takım yolları oluşturmak için birbiri ile aynı geometrik standartlara sahip yüzeyler elde etmektir. Bu yüzden bu algoritma alt kümesi “U-V Yönünü Ayarlama” olarak adlandırılmıştır. Önceki adımdaki

“Yüzeyler” ve “Curveler” bileşenlerinin çıktıları görsel kodlama arayüzündeki nizamı sağlamak adına sırası ile “Surface” ve “Curve” bileşenlerine bağlanmıştır. “Surface” bileşeninin çıktısı “Explode” bileşeninin “Curve” girdisine bağlanarak yüzeyi kapsayan eğriler kendi parametrelerine göre bölümlendirilmiştir. “Explode” bileşeninin “Segments” çıktısı “List Item” bileşeninin “List” girdisine bağlanarak bileşen içerisindeki gerekli eğriler elde edilmiştir. “List Item” bileşeninin “0” çıktısı iki farklı “Similarity” bileşeninin “First Number” girdilerine aynı bileşenin “Second Number” girdilerine ise sırası ile “Alt Eğri” ve “Üst Eğri” bileşenleri bağlanmıştır. Normalde eğriyi tanımlayan bileşenler “Similarity” bileşeninin girdisine bağlandığında eğrilerin uzunluk verisini tanımlamaktadır. Burada amaç sekiz farklı yüzeydeki her bir yüzeyi oluşturan dört farklı eğrinin aynı sıra ile sıralanıp sıralanmadığını test etmektir. Eğer değerler aynı ise “Similarity” bileşeni “Similarity” çıktı verisinde “True” değerini ve bunun rakamsal karşılığı olan “1” değerini eğer farklı ise “False” değerini ve bunun rakamsal karşılığı olan “0” değerini vermektedir. Tüm bu algoritma “Reverse Surface Direction” bileşeninin “Reverse Options” girdisinde rakamlara bağlı işlevini ayarlamak adına yapılmıştır. Bu bileşende “Reverse Options” girdisine 0 değeri gelirse bileşen çalışmamakta, 1 değeri gelirse sadece yüzeyin sadece U yönünü döndürmekte, 2 değeri gelirse yüzeyin sadece V yönünü döndürmekte ve 3 değeri gelirse yüzeyin UV yani tümel yönünü döndürmektedir. İki farklı “Similarity” bileşeninde yapılan işlemlerin çıktıları “Addition” bileşenine bağlanarak “True” – “False” düzleminde bir toplama işlemi yapılmıştır. Bu işleme göre kenar değerleri birbiri ile uyumlu olmayan yüzeyler “False” yani 0 değerinin alacaklarından dolayı işlem sonucu 0 olarak elde edilmiştir. “Addition” bileşenini çıktı verilerini “Reverse Surface Direction” bileşenine uyarlamak için “Replace Text” bileşeninde “Text” girdisine aynı bileşende “Find” girdisine “0” ve “Replace” girdisine “3” değerleri girerek “0” olan değerler “3” değerine dönüştürülüp tüm yüzeylerin aynı yönde olması sağlanmıştır. Şekil 5.34’te bu algorima kümesinde elde edilen yüzey verisi rhino üç boyutlu görüntüleme arayüzünde gözlenmektedir.



Şekil 5.34. U-V yönünü ayarlama

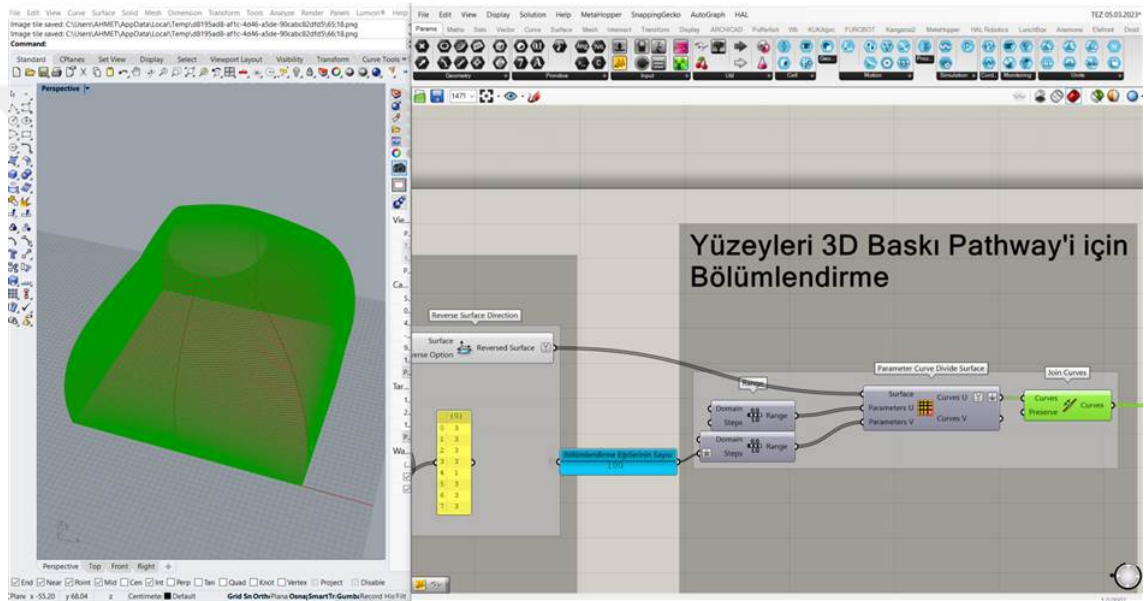
“U-V Yönünü Ayarlama” algoritma kümesindeki işlem her bir genişletme değerinde aynı şekilde tekrar edilmiştir. Şekil 5.35’te strüktür yapısını oluşturan yüzeyler yeşil renkte gözlenmektedir.



Şekil 5.35. Tüm genişletme değerleri için U-V yönünü ayarlama

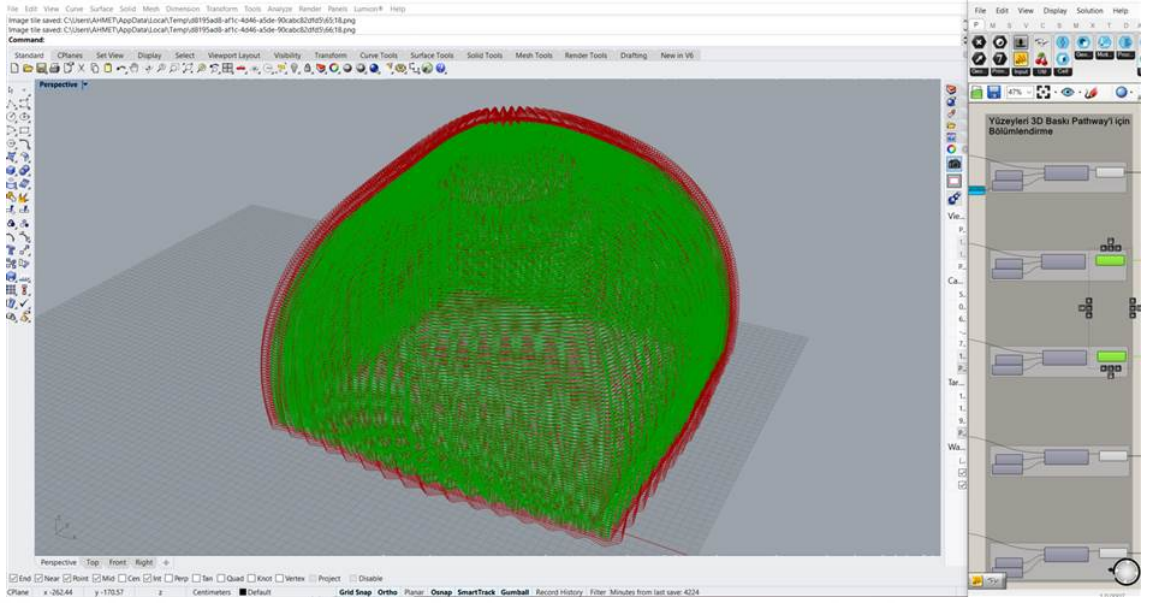
Bu adımda amaç “U-V Yönünü Ayarlama” algoritma kümesindeki “Reverse Surface Direction” bileşeninin çıktı verisinden gelen yüzeyleri uygun bir şekilde üç boyutlu baskılama takım yolları için bölümlendirmektir. Bu yüzden bu algoritma alt kümesi “Yüzeyleri 3D Baskı Pathway’i için Bölümlendirme” olarak adlandırılmıştır. “Reverse Surface Direction” bileşeninin çıktı verileri “Parameter Curve Divide Surface”

bileşeninin “Surface” girdisine bağlandıktan sonra “Parameters U” girdisine standart “Range” bileşeni bağlanarak 0-1 arası bir domain verisi elde edilmiş olup bu yatay yönde 1 adet bölümlendirme anlamına gelmektedir. “Parameters V” bileşenine ise parametreler bölümünde yer alan “Number Slider” bileşeninin 100 değeri “Range” bileşeninin “Steps” değerine bağlanarak dikey yönde 100 adet bölümlendirme parametre verisi elde edilmiştir. “Parameter Curve Divide Surface” bileşeninin “Curves U” çıktı verileri “Join Curves” bileşeninin “Curves” girdisine bağlanarak aynı veri ağacındaki birbiri ile ilişkili eğriler birleştirilmiştir. Şekil 5.36’da bu algoritma kümesinde yapılan işlemlerin çıktı verisi Rhino üç boyutlu görüntüleme arayüzünde yeşil renkte gözlenmektedir.



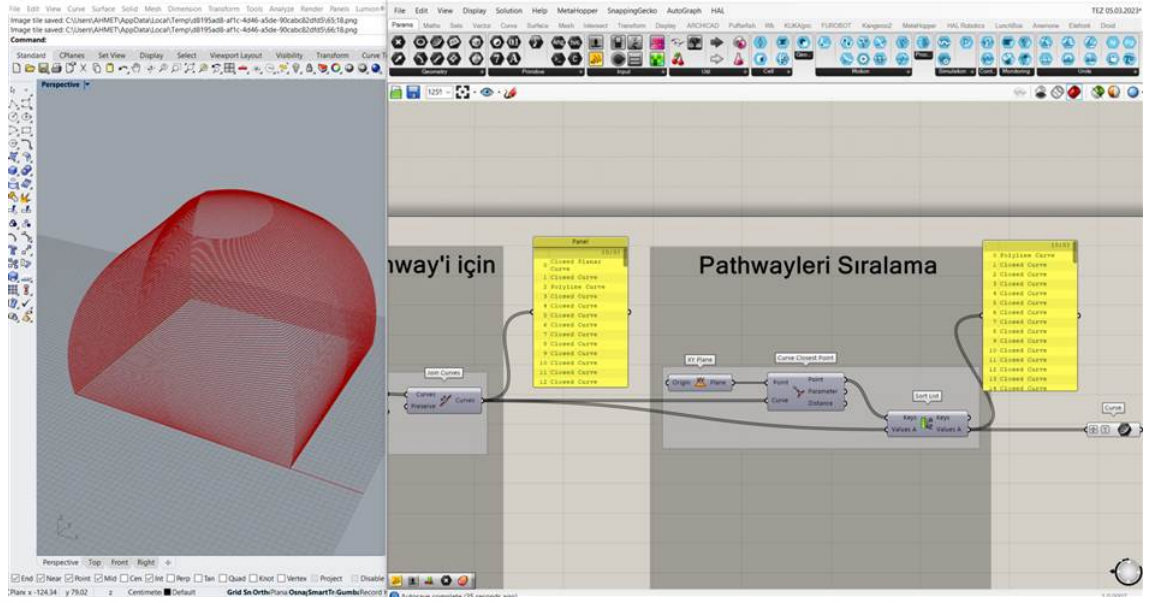
Şekil 5.36. Yüzeyleri 3D baskı pathwayi için bölümlendirme

“Yüzeyleri 3D Baskı Pathway’i için Bölümlendirme” algoritma kümesindeki işlemler her bir genişletme değeri için ayrı ayrı tekrar edilmiştir. Şekil 5.37’de Rhino üç boyutlu görüntüleme arayüzünde iç strüktür yapısını oluşturan eğriler yeşil renkte, barınma biriminin baskılanması için gerekli takım yolları ise kırmızı renkte gözlenmektedir.



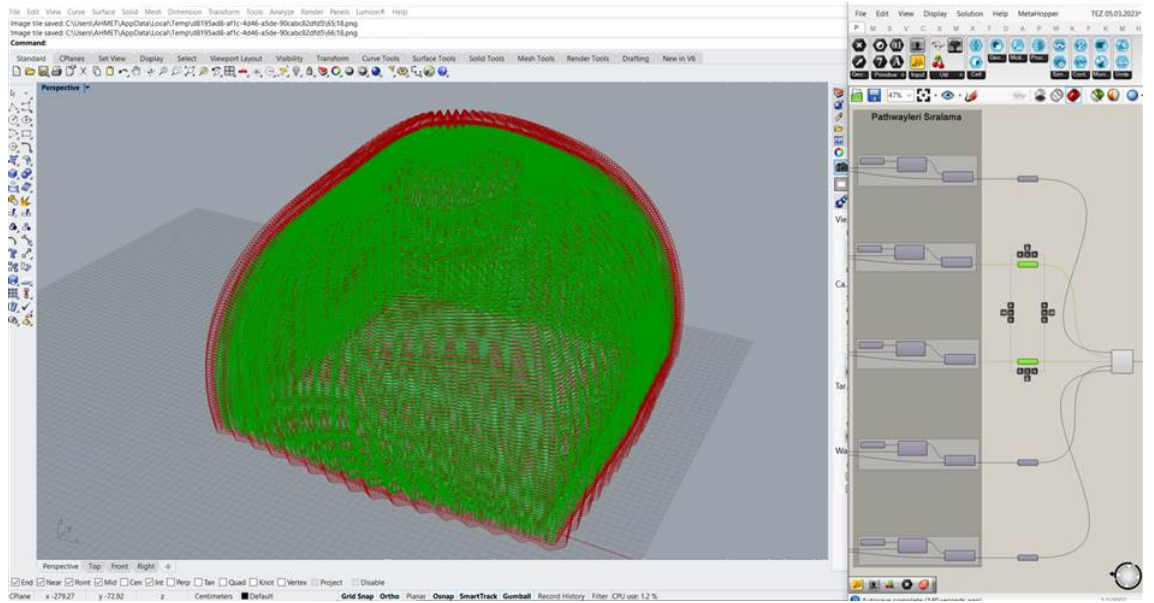
Şekil 5.37. Tüm yüzeyleri 3D baskı pathwayi için bölümlendirme

“Yüzeyleri 3D Baskı Pathway”i için Bölümlendirme” algoritma kümesindeki “Join Curves” bileşeninin çıktı verileri incelendiğinde bu eğrilerin sıralanmasında sapmalar olduğu gözlenmiştir. Bu yüzden bu adımda amaç bu eğrileri xy düzlemine yakınlıktan uzağa yani aşağıdan yukarı doğru tekrar sıralamaktır. Bu algoritma alt kümesi “Pathwayleri Sıralama” olarak adlandırılmıştır. “Curve Closest Point” bileşeninin “Point” girdisine “XY Plane” bileşeni “Curve girdisine ise “Join Curves” bileşeninin “Curves” çıktısı bağlanarak” her bir eğrinin xy düzlemine en yakın noktaları hesaplanmıştır. “Curve Closest Point” bileşeninin “Point” çıktısı “Sort List” bileşeninin “Keys” girdisine “Join Curves” bileşeninin “Curves” çıktısı ise “Values A” bileşenine bağlanarak “Sort List” bileşeninin xy düzlemine en yakın noktalara sahip olan eğrilerin yakından uzağa sıralaması gerçekleştirilmiştir. Şekil 5.38’de bu adımda gerçekleştirilen işlemler sonrası değişiklik “Panel” bileşenindeki eğri niteliklerinin farklılaşması ile gözlenmektedir.



Şekil 5.38. Pathwayleri sıralama

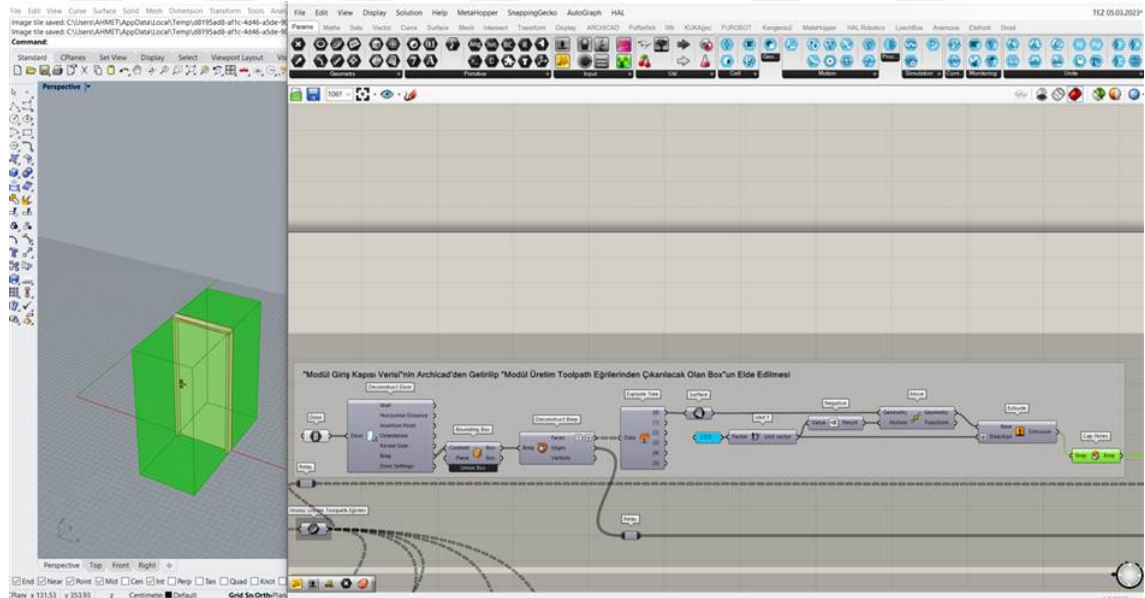
“Pathwayleri Sıralama” algoritma alt kümesindeki işlem tüm genişletme değerleri için ayrı ayrı uygulanmıştır. Bu uygulamadan sonra algoritma alt kümesinin çıktı verileri ayrı ayrı “Curve” bileşenine atanarak bu bileşenin çıktı verileri “Merge” bileşenine atanmış ve görsel kodlama arayüzündeki nizamı sağlamak için tek bir bileşende bir araya getirilmiştir. Şekil 5.39’da barınma biriminin üç boyutlu baskılama için oluşturulan tüm takım yollarının elde edildiği gözlenmektedir.



Şekil 5.39. Tüm pathwayleri sıralama

Barınma biriminin iç kabuk ve strüktür yapısının takım yolları elde edildikten sonra bu adımda amaç kapı ve pencere boşluklarının bu eğrilerden çıkarılıp takım yollarının

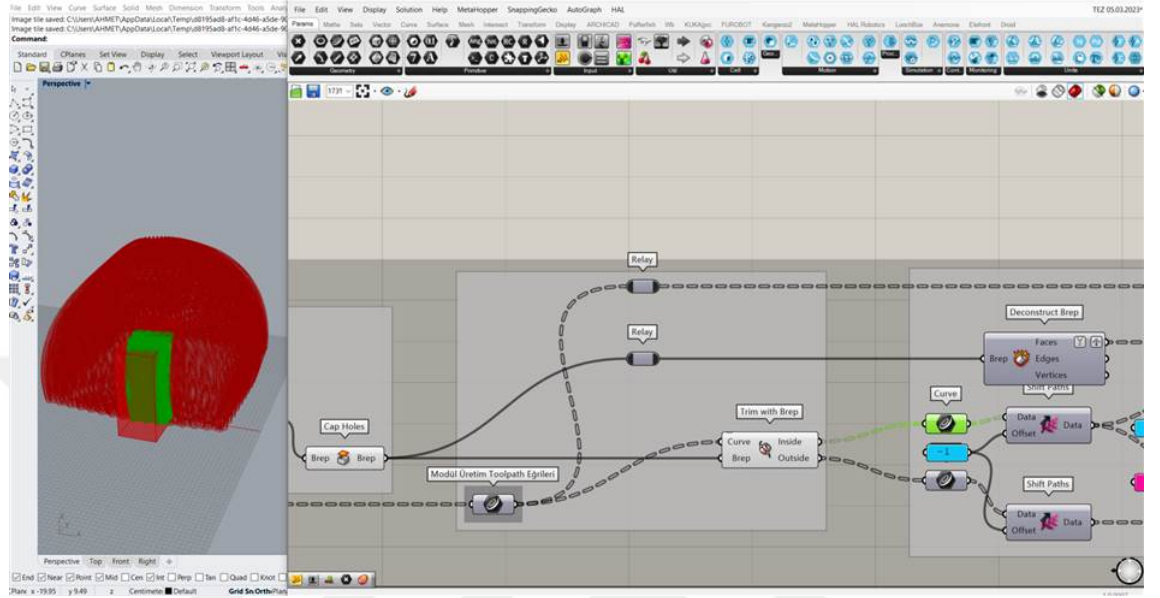
tekrar organize edilmesidir. Bunun için bu algoritma kümesinin ilk alt algoritma kümesi “Modül Giriş Kapısı Verisinin ArchiCAD’den getirilip Modül Üretim Toolpath Eğrilerinden Çıkarılacak Olan Box’un Elde Edilmesi” olarak adlandırılmıştır. Bunun için “Door” bileşenine ArchiCAD’de oluşturulan kapı atanıp bu bileşenini çıktısı “Deconstruct Door” bileşenine bağlanmıştır. “Deconstruct Door” bileşeninin “Brep” Çıktı verisi “Bounding Box” bileşeninin “Content” girdisine bağlanarak kapı elemanını çevreleyen küp elde edilmiştir. “Bounding Box” bileşeninin “Box” çıktı verisi “Deconstruct Brep” bileşenine bağlandıktan sonra “Faces” çıktı verisi “Graft” işlemi yapıp “Explode Tree” bileşeninden gerekli yüzeylerin çıktı verisi seçilmiştir. Seçilen yüzey “Surface” bileşenine görsel kodlama arayüzünde düzen sağlanması için bağlanmıştır. “Surface” bileşeninin çıktısı “Move” bileşeninin “Geometry” girdisine bağlanmıştır. “Move” bileşeninin “Motion” girdisine ise hem bu kapıyı hem de tüm takım yolu eğrilerini kesecek bir geometri elde etmek için y düzlemi yönünde 100cm hareket edecek algoritma oluşturulmuştur. “Move” bileşeninin çıktı verisi “Extrude” bileşeninin “Base” girdisine aynı bileşeninin “Direction” girdisine ise (x*2) işlemi tanımlanarak 200cm genişliğinde gerekli yüzeyden dikdörtgen prizma verisi elde edilmesi sağlanmıştır. Şekil 5.40’da bu geometri yeşil renkte gözlenmektedir.



Şekil 5.40. Modül giriş kapısı verisinin archicad’den getirilip modül üretim toolpath eğrilerinden çıkarılacak olan box’un elde edilmesi

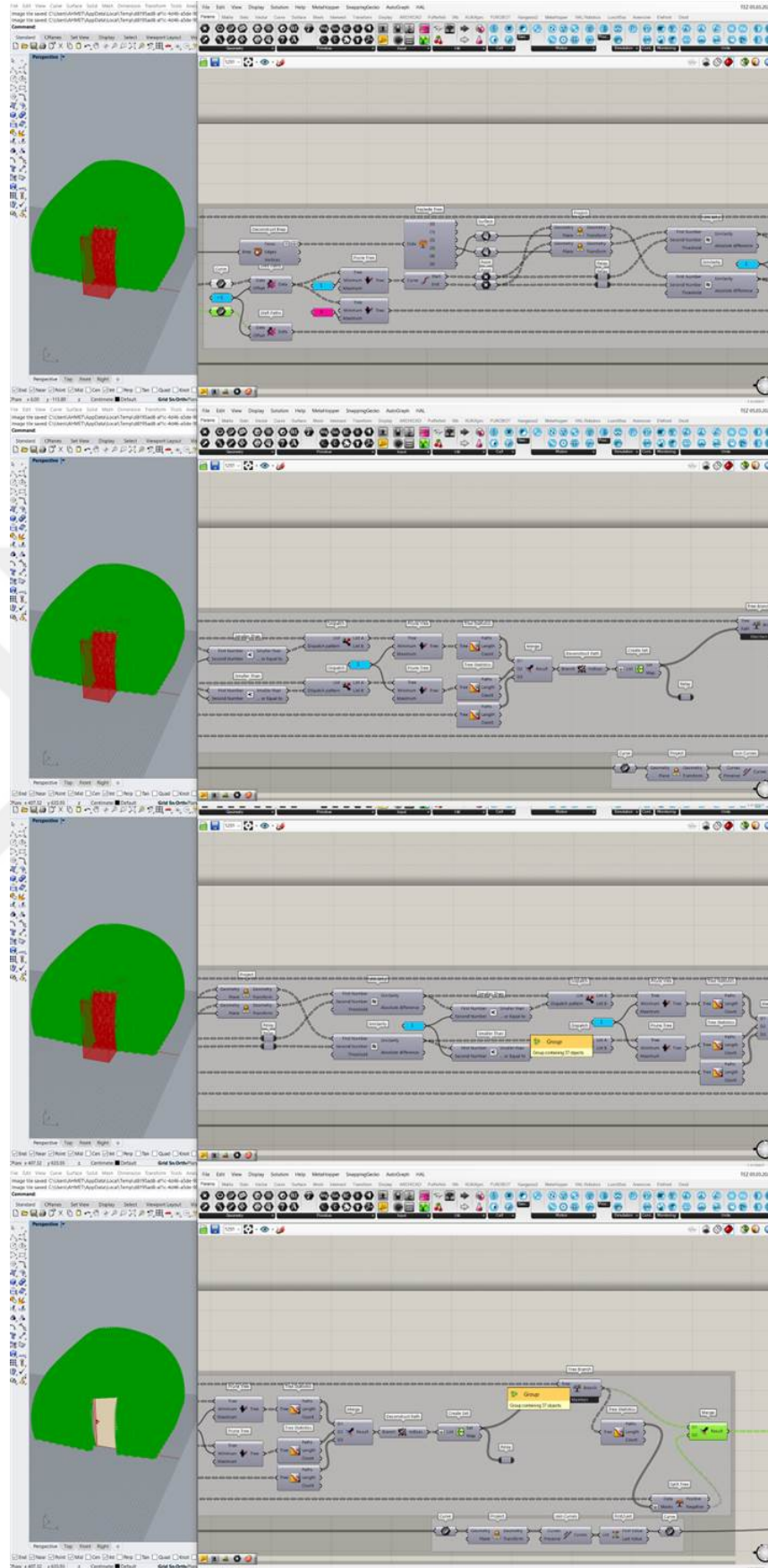
Sonraki adımda amaç kapıyı kapsayan dikdörtgen prizmasının modül üretim takım yolu eğrileri ile kesiştirilip kapıyı kapsayan eğrilerin geometriden çıkarılmasıdır. Bunu sağlamak için “Modül Üretim Toolpath Eğrileri” bileşeninin çıktı verisi “Trim with Brep”

bileşeninin “Curve” girdisine, kapıyı kapsayan dikdörtgen prizma olan “Cap Holes” bileşeninin çıktı verisi ise “Brep” bileşenine bağlanarak gerekli kesişim işlemi tamamlanmıştır. Şekil 5.41’de dikdörtgen prizma ile tüm toolpath eğrilerinin keşimi Rhino üç boyutlu önizleme ekranında gözlenmektedir.



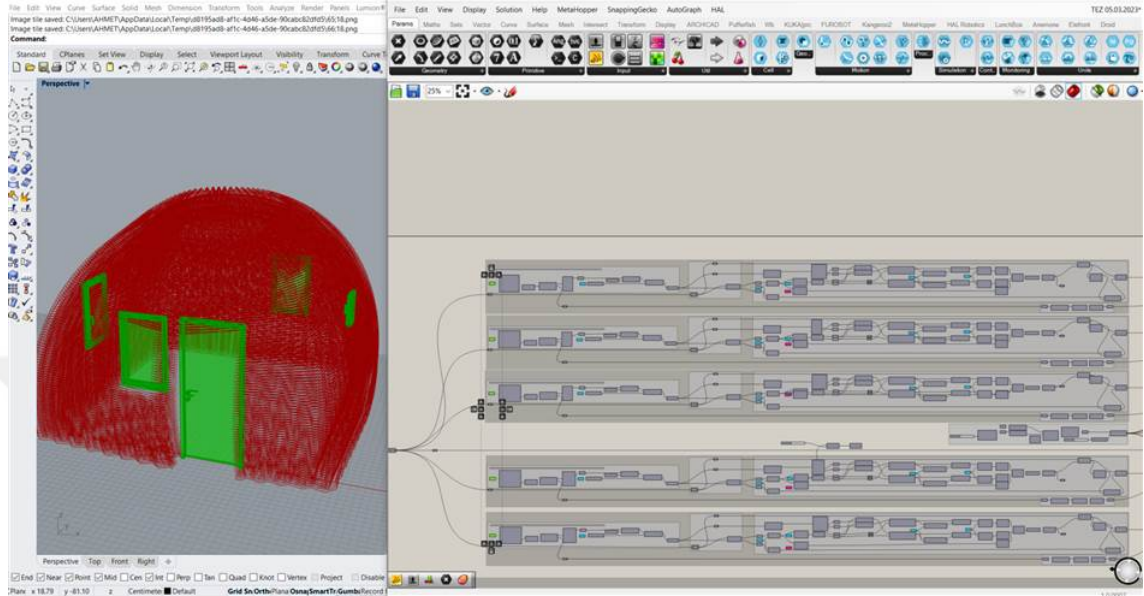
Şekil 5.41. Modül üretim toolpath eğrilerinden kapı kısmının çıkarılması

Bu algoritma alt kümesinin devamında yapılan işlemlerde ise amaç kapının kapsadığı takımyolu eğrileri çıkarıldıktan sonra üretimini gerçekleştirilmesi için bu eğrilerin düzenlenmesidir. Şekil 5.42’de algoritmanın devamı gözlenmektedir.



Şekil 5.42. Modül üretim takım yolu eğrilerinden giriş kapısı kısmının çıkarılıp organize edilmesi

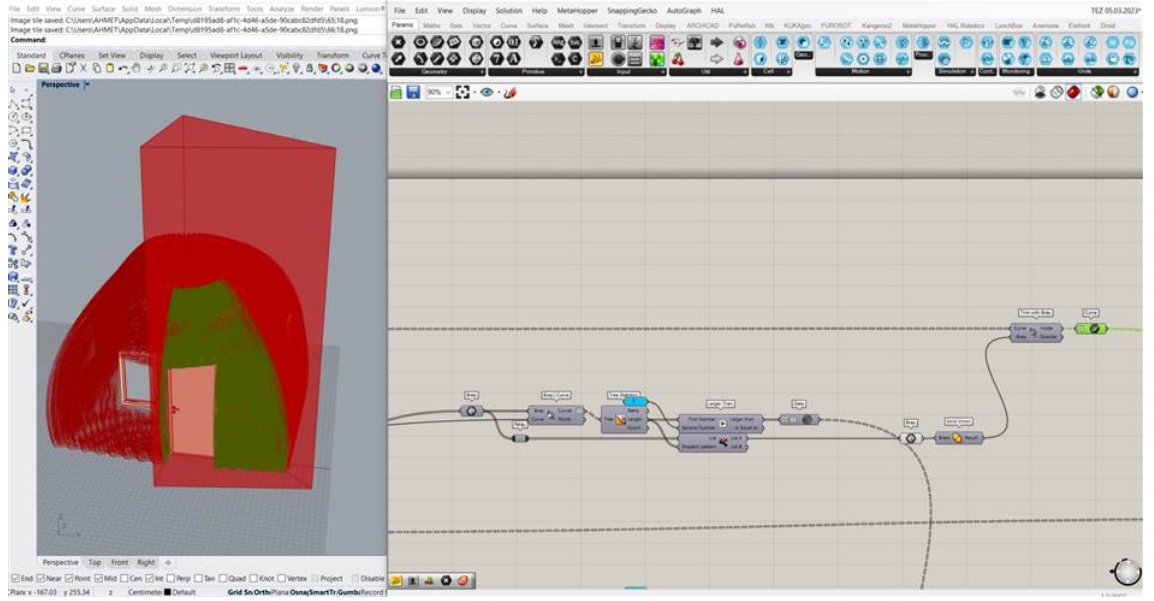
Bir önceki adımda yapılan işlemler aynı kurgu ile tüm kapı ve pencere boşlukları için tekrar edilmiştir. Şekil 5.43'te tüm kapı ve pencereler ve modül üretim takım yolu eğrilerin çıkarıldıktan sonraki görüntüsü Rhino üç boyutlu görüntüleme arayüzünde gözlenmektedir.



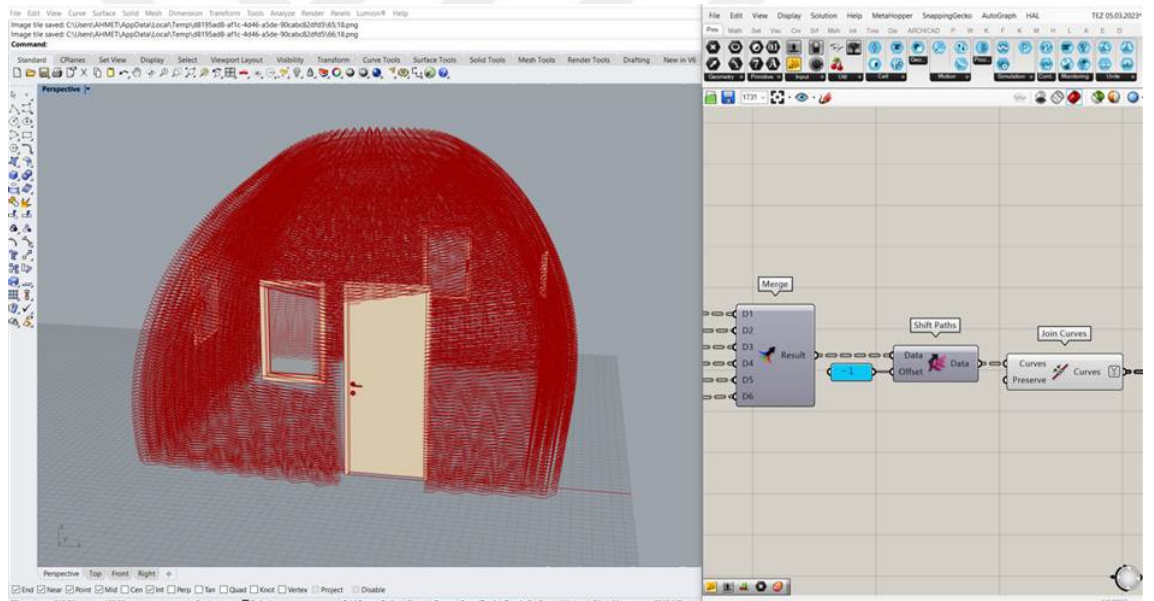
Şekil 5.43. Kapı ve pencere boşluklarının takım yolu eğrilerinden çıkarılması

Önceki adımda yapılan çıkarma işlemi sırasında takım yolu eğrilerinin verisinin organizasyonunun daha kompleks hale gelmemesi adına tüm takım yolu eğrileri her seferinde ayrı ayrı gerekli boşluk çıkarma işlemine maruz bırakılmıştır. Bu durum Şekil 5.44'te oluşturulan algoritma ile gerekli bölümlerin seçilip daha sonra ayrı ayrı birleştirilmesi ile organize edilmiştir.

Barınma biriminin kabuk yapısının takım yolları ve kapı pencereleri Şekil 5.45'te gözlenmektedir.



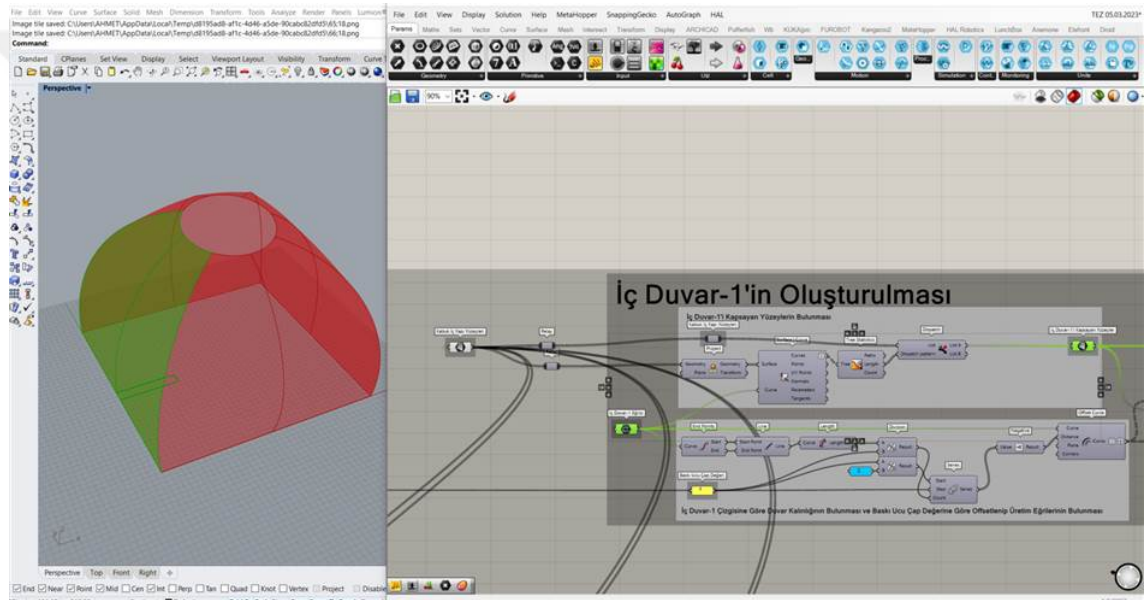
Şekil 5.44. Kapı ve pencere boşlukları için üretim takım yollarının yeniden organizasyonu



Şekil 5.45. Barınma biriminin kabuk yapısını oluşturan takım yolları

Barınma biriminin dış kabuk ve strüktür yapısının takım yolları elde edildikten sonraki adımda hedef iç duvarların oluşturulmasıdır. Bu yüzden bu algoritma alt kümesine “İç Duvar-1’in Oluşturulması” adı verilmiştir. İç duvarların parametrik olarak elde edilmesi ve kabukla olan ilişkilerinin parametrik modellenmesi için bu işlem bütününde seçime dayalı bir algoritma geliştirilmiştir. “Kabuk İç Yapı” yüzeyler bileşeninin çıktı verisi “Project” bileşeninin “Geometry” girdisine bağlanarak bu yüzeyler xy düzlemine yansıtılmıştır. “İç Duvar-1 Eğrisi” bileşeni ArchiCAD’de oluşturulan duvar kontur çizgisinin “2D Polyline” bileşenine atanması ile elde edilmiştir. Bu bileşenini

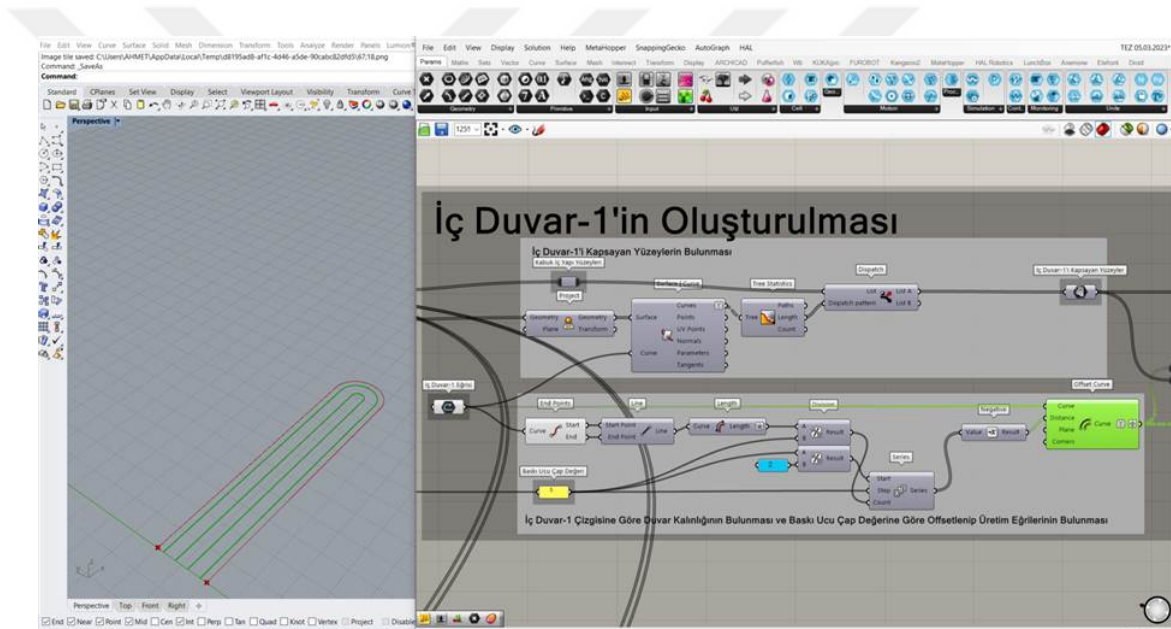
çıktısı “Surface-Curve” bileşeninin “Curve” girdisine “Project” bileşeninin “Geometry” çıktısı ise “Surface” girdisine bağlanarak “İç Duvar-1 Eğrisi”nin hangi yüzey ile kesişeceğinin hesaplanması hedeflenmiştir. Bu doğrultuda “Curves” çıktı verisi “Tree Statistics” bileşeninin “Tree” bileşenine bağlanarak farklı veri ağaçlarındaki kümelerde kesişim elemanlarının olup olmadığı test edilmiştir. Eğer kesişim elemanları var ise bu veri ağaçlarındaki liste uzunluğu en az 1 değerini alır. “Kabuk İç Yapı Yüzeyler” bileşeninin çıktısı “Dispatch” bileşenine, “Tree Statistics” bileşeninin “Length” çıktısı ise “Dispatch Pattern” girdisine bağlanarak bir seçim listesi oluşturulmuştur. Bu işlem sonucunda Şekil 5.46’da görüleceği üzere “İç Duvar-1’i Kapsayan Yüzeyler” parametrik olarak elde edilmiştir.



Şekil 5.46. İç duvar-1’i kapsayan yüzeylerin bulunması

“İç Duvar-1’in Oluşturulması” algoritma kümesinde yapılması gereken bir diğer işlem ise “Baskı Ucu Çap Değeri”ne göre iç duvar yapısının strüktür eğrilerinin elde edilmesidir. Bu yüzden bu algoritma alt kümesi “İç Duvar-1 Çizgisine Göre Duvar Kalınlığının Bulunması ve Baskı Ucu Çap Değerine Göre Offsetlenip Üretim Eğrilerinin Bulunması” olarak adlandırılmıştır. Bunu elde etmek için “İç Duvar-1 Eğrisi” bileşeninin çıktı verisi “End Point” bileşenine bağlanmış ve bu bileşenini “Start” ve “End” çıktıları yani “İç Duvar-1” eğrisinin başlangıç ve bitiş noktaları bulunmuştur. Çıktı verileri sırası ile “Line” bileşeninin “Start Point” ve “End Point” bileşenlerine bağlanarak başlangıç ve bitiş noktaları arasında bir çizgi oluşturulmuştur. “Line” bileşeninin çıktı verisi “Length” bileşenine bağlanarak başlangıç ve bitiş noktaları arasındaki mesafe hesaplanmıştır.

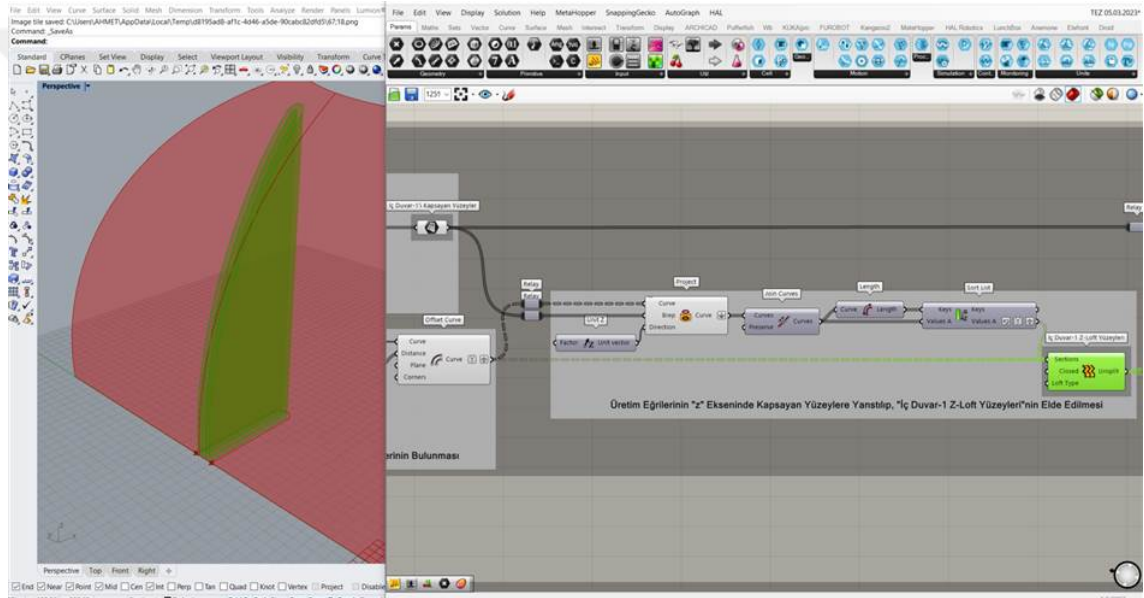
“Length” bileşeninin çıktı verisi “Division” bileşeninin “A” girdisine, “Baskı Ucu Çap Değeri” olan 5 parametresi ise “B” girdisine bağlanmıştır. “Baskı Ucu Çap Değeri” bir başka “Division” bileşenine bağlanarak bölünme girdisine sabit “2” değeri atanmıştır. İlk “Division” bileşeninin çıktı verisi “Series” bileşeninin “Count” girdisine, “Baskı Ucu Çap Değeri” “Step” girdisine ve ikinci “Division” bileşeninin çıktı verisi ise “Step” girdisine bağlanarak “İç Duvar-1 Eğrisinin” “Baskı Ucu Çap Değeri”ne bağlı parametrik olarak kaç kere ve hangi değerlerde offsetleneceği hesaplanmıştır. “Offset Curve” bileşeninin “Curve” girdisine “İç Duvar-1 Eğrisi” bileşeni ve “Distance” girdisine ise “Series” bileşeninin çıktı değerleri atanarak “İç Duvar-1 Eğrisi”nin hesaplanan değerlere göre iç offsetlenmesi gerçekleştirilmiştir. Şekil 5.47’de “İç Duvar-1 Eğrisi” kırmızı renkte iç offsetlenen eğriler ise yeşil renkte gözlenmektedir.



Şekil 5.47. İç Duvar-1 Çizgisine Göre Duvar Kalınlığının Bulunması ve Baskı Ucu Çap Değerine Göre Offsetlenip Üretim Eğrilerinin Bulunması

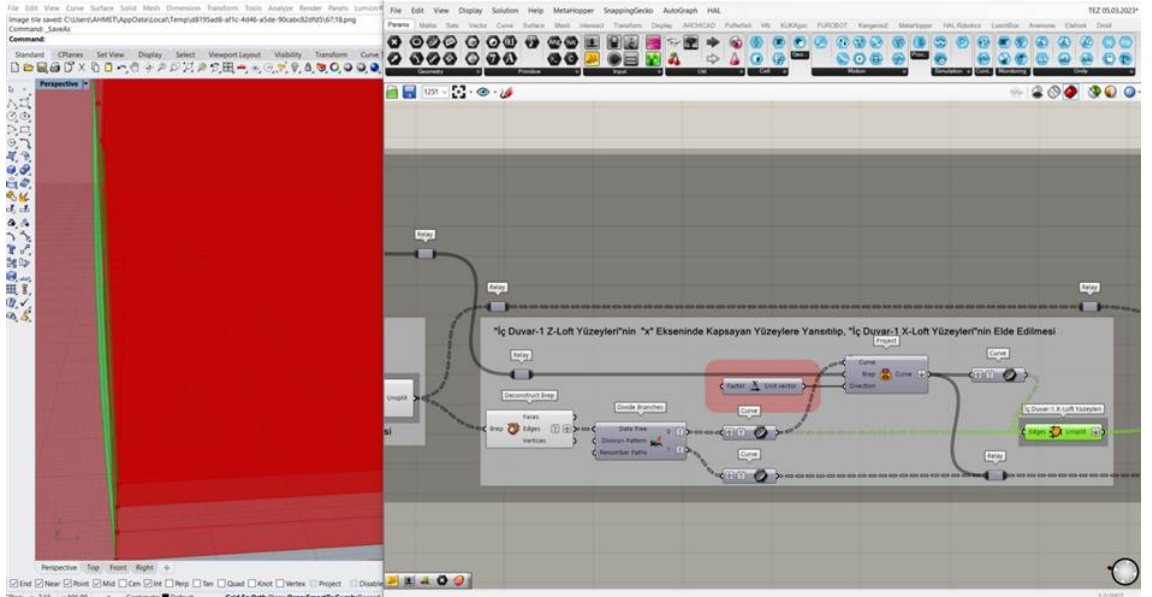
Bu adımda amaç “İç Duvar-1 Eğrilerinin” yapı iç kabuğuna erişecek şekilde oluşturulmasıdır. Bu yüzden bu algoritma alt kümesi “Üretim Eğrilerinin "z" Ekseninde Kapsayan Yüzeylere Yanstılıp, "İç Duvar-1 Z-Loft Yüzeyleri"nin Elde Edilmesi” olarak adlandırılmıştır. Bunu sağlamak için “Project” bileşeninin “Curve” girdisine “Offset Curve” bileşeninin çıktı verisi, “Brep” girdisine “İç Duvar-1'i Kapsayan Yüzeyler” bileşeni ve “Direction” girdisine ise “Unit Z” bileşeni bağlanmıştır. Bu işlem sonucunda “İç Duvar-1 Eğrilerinin” ilişkili yüzeye yansıtılması gerçekleştirilmiştir. Fakat adım sonunda elde edilen eğrilerin iki farklı yüzey boyunca bölündüğü gözlenmiştir. Bu sorunu çözmek için “Project” bileşeninin çıktı verisi “Joint Curves” bileşeninin “Curves”

girdisine bağlanarak eğriler birleştirilmiştir. Eğrilerin uygun bir şekilde sıralanması için dıştaki eğriler içteki eğrilere doğru uzundan kısaya gideceğinden ötürü “Join Curves” bileşeninin çıktı verisi “Length” bileşenine bağlanmış bu bileşeninin çıktı verisi “Sort List” bileşeninin “Keys” girdisine, “Join Curves” bileşeninin çıktı verisi ise “Values A” bileşenine atanarak eğrilerin uygun bir şekilde sıralanması elde edilmiştir. “Sort List” bileşeninin “Values A” çıktısı ve önceki adımda bulunan “Offset Curve” bileşeninin çıktısı “Unsplit Loft Surface” bileşeninin “Sections” girdisine bağlanarak “İç Duvar-1 Eğrisi”nin ilişkili yüzeyler ile parametrik modellenmesi tamamlanmıştır. Şekil 5.48’de Rhino üç boyutlu görüntüleme arayüzünde yeşil renkte “İç Duvar-1”in z ekseninde ilişkili duvarlar ile oluşan model gözlenmektedir.



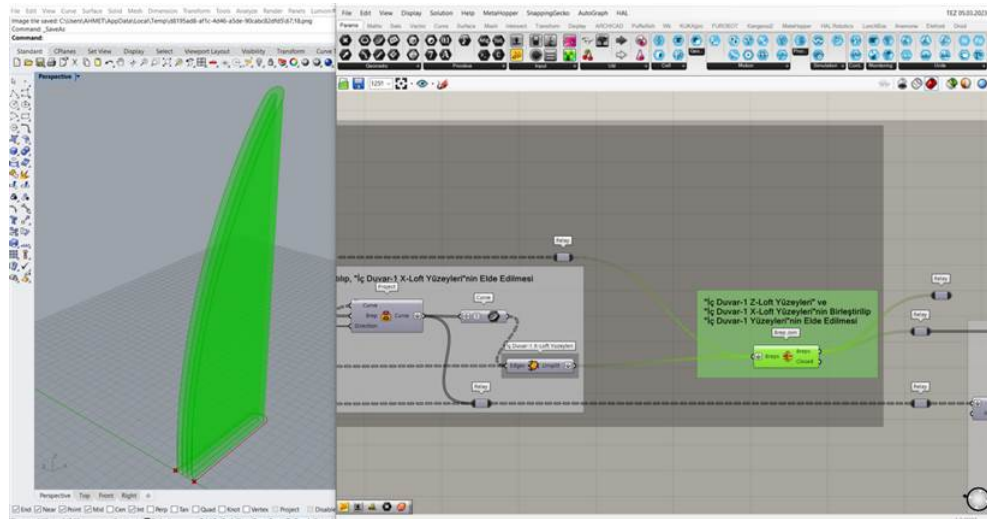
Şekil 5.48. Üretim Eğrilerinin "z" Ekseninde Kapsayan Yüzeyle Yanstılup, "İç Duvar-1 Z-Loft Yüzeyle"nin Elde Edilmesi

Önceki adımda “İç Duvar-1”in sadece z eksenı üzerinde yansıması parametrik olarak modellendiğı için Şekil 5.49’da görüleceğı üzere iç kabuk ile arasında tamamlanmayan bir boşluk kaldığı gözlenmiştir. Bunun için bu adımda amaç bu boşluğu ilgili eksende benzer algoritmayı kurarak sağlamaktır. Bu yüzden bu algoritma alt kümesi “İç Duvar-1 Z-Loft Yüzeyle”nin “x” Ekseninde Kapsayan Yüzeyle Yanstılup, "İç Duvar-1 X-Loft Yüzeyle"nin Elde Edilmesi” olarak adlandırılmıştır. Şekil 5.49’da Grasshopper arayüzünde algoritma ve Rhino üç boyutlu görüntüleme arayüzünde yeşil renkte algoritma alt kümesinin çıktısı gerekli boşluğu dolduran yüzeyler gözlenmektedir.



Şekil 5.49. "İç Duvar-1 Z-Loft Yüzeyleri"nin "x" Ekseninde Kapsayan Yüzeyle Yanıtlı, "İç Duvar-1 X-Loft Yüzeyleri"nin Elde Edilmesi

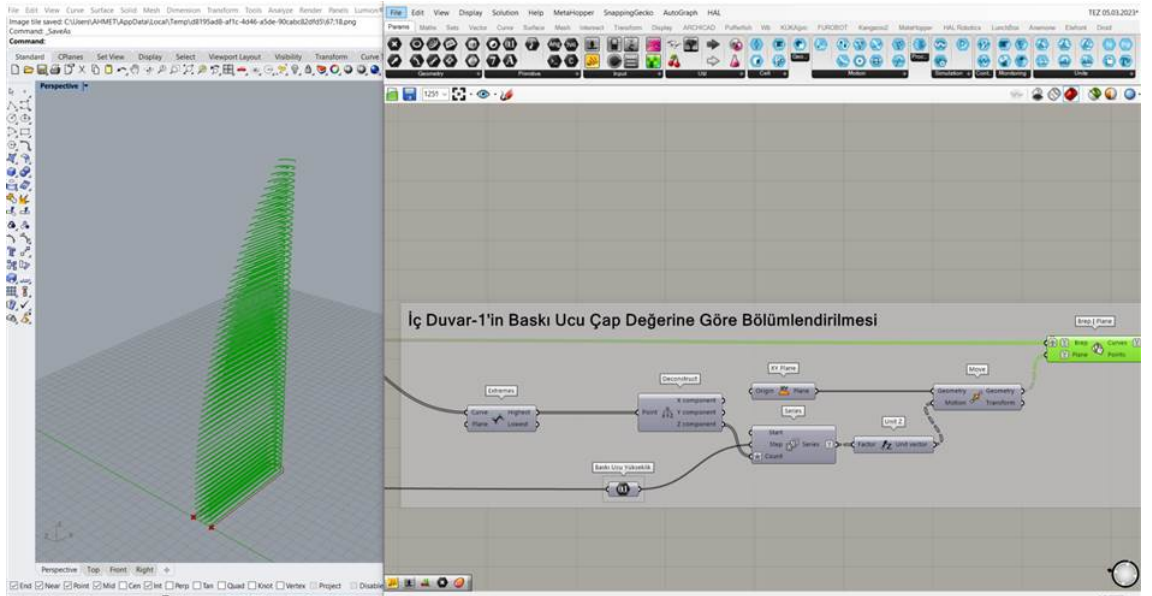
Bu adımda amaç son iki adımda oluşturulan yüzeylerin birleştirilmesidir. Bu yüzden bu algoritma alt kümesine “İç Duvar-1 Z-Loft Yüzeyleri” ve "İç Duvar-1 X-Loft Yüzeyleri"nin Birleştirilip "İç Duvar-1 Yüzeyleri"nin Elde Edilmesi” adı verilmiştir. Son iki algoritma alt kümesinin çıktı verileri “Brep Join” bileşeninin “Breps” girdilerine bağlanarak temas eden yüzeylerin brep geometrisinde birleştirilmesi sağlanmıştır. Şekil 5.50’de birleştirilen yüzeyler gözlenmektedir.



Şekil 5.50. "İç Duvar-1 Z-Loft Yüzeyleri" ve "İç Duvar-1 X-Loft Yüzeyleri"nin Birleştirilip "İç Duvar-1 Yüzeyleri"nin Elde Edilmesi

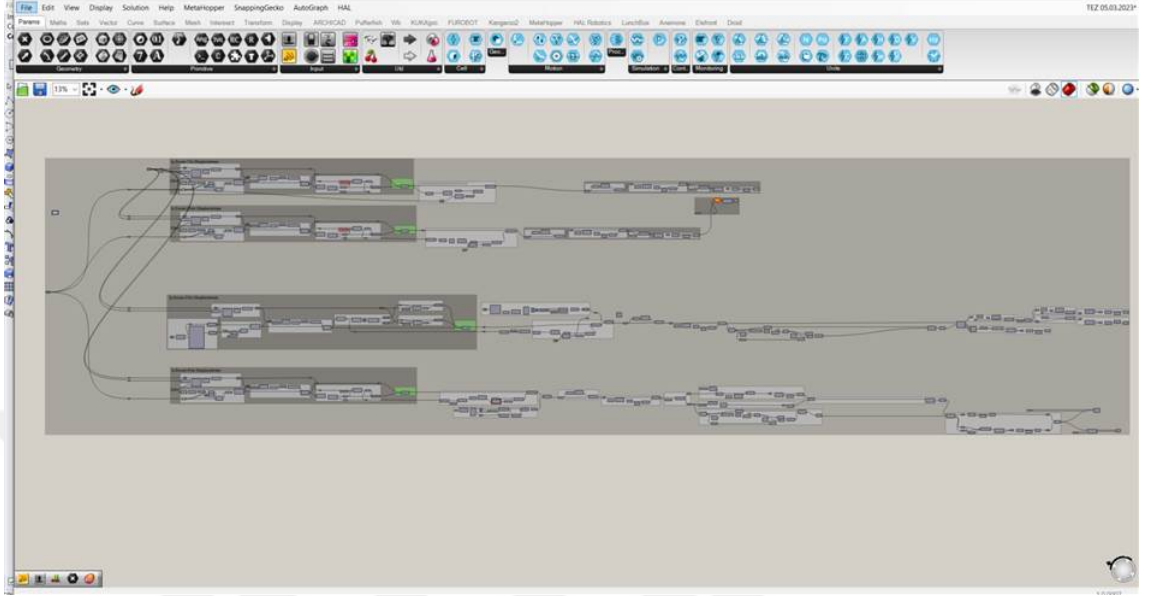
Bu adımda amaç bir önceki adımda tüm yüzeyleri parametrik olarak modellenmiş “İç Duvar-1”ın baskı ucu çap değerine ve yükseklik parametresine göre

bölümlendirilmesidir. Bu yüzden bu algoritma alt kümesi “İç Duvar-1’in Baskı Ucu Çap Değerine Göre Bölümlendirilmesi” olarak adlandırılmıştır. “Brep Join” bileşenini çıktı verisi “Extremes” bileşeninin “Curve” girdisine bağlanarak duvar yüzeylerindeki eğrilerin en yüksek ve en düşük nokta verileri elde edilmiştir. Bu bileşenin “Highest” çıktısı “Deconstruct” bileşenine bağlanarak “Deconstruct” bileşenin “Z Component” çıktısı ile en yüksek noktanın z koordinatı sayısal değer olarak elde edilmiştir. “Z Component” çıktısı “Series” bileşeninin “Count” girdisine ($x/5$ işlemi tanımlanmış), “Baskı Ucu Yükseklik” bileşeni “Step” girdisine bağlanarak yükseklik parametresine göre kaç adet kesim düzlemi oluşturulacağı hesaplanmıştır. “Series” bileşeninin “Series” çıktı verisi “Unit Z” bileşenine bağlanmış ve bu bileşenin çıktı verisi kesim düzlemi olan xy düzleminin kaç kez ve kaç farklı veri ağacında hareket edeceğini belirlemiştir. “Move” bileşeninin “Geometry” girdisine “XY Plane” bileşeni “Motion” girdisine ise “Unit Z” çıktısı bağlanarak gerekli xy düzlemleri elde edilmiştir. “Brep-Plane” bileşeninin “Breps” girdisine önceki algoritma alt kümesinin çıktı verisi duvar yüzeyleri, “Plane” girdisine ise bu adımda elde edilen düzlemler olan “Move” bileşeninin çıktı verisi bağlanarak duvar yüzeyleri gerekli düzlemler ile kesiştirilip gerekli eğriler elde edilmiştir. Şekil 5.51’de Rhino üç boyutlu görüntüleme arayüzünde “İç Duvar-1’in bölümlendirilmesi ile elde edilen eğriler yeşil renkte gözlenmektedir.



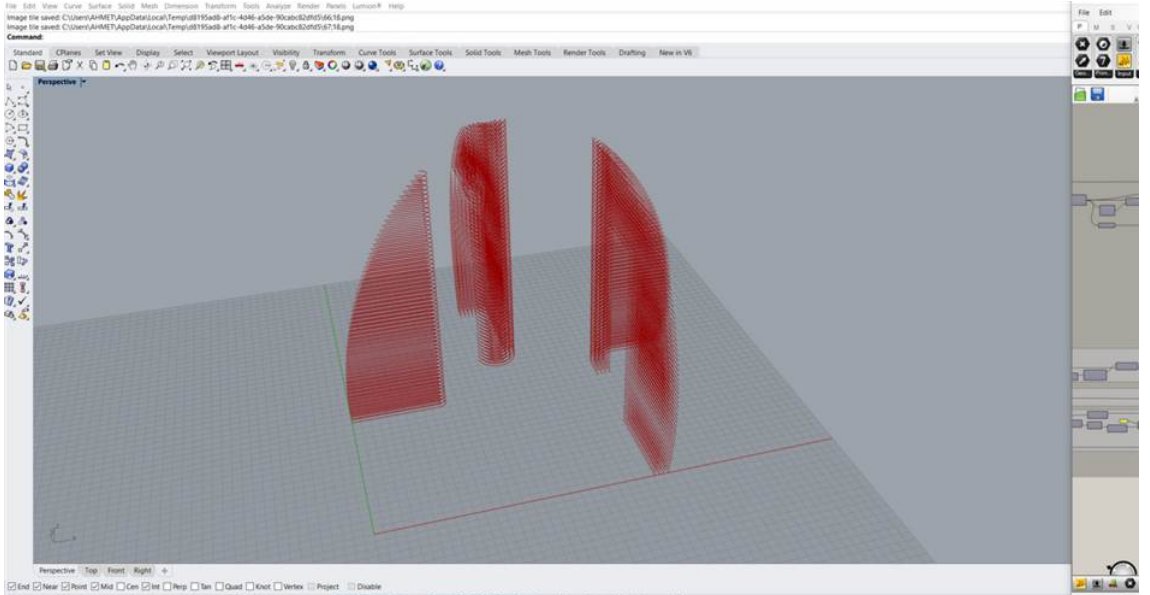
Şekil 5.51. İç duvar-1'in baskı ucu çap değerine göre bölümlendirilmesi

İç duvarların oluşturulması ve bölümlendirilmesi aynı kurgu ile tüm iç duvar eğrileri için tekrarlanmıştır. Şekil 5.52’de iç duvarların parametrik olarak modellenmesi için oluşturulan algoritma gözlenmektedir.



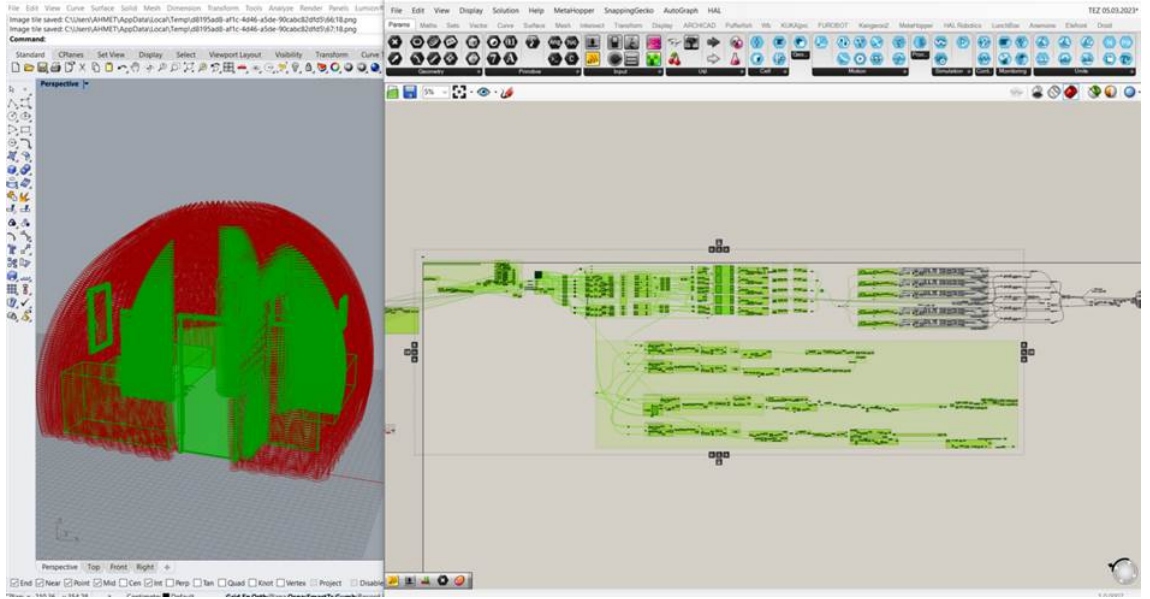
Şekil 5.52. İç duvarlar için oluşturulan algoritma kümesi

Şekil 5.53’te tüm iç duvarların bölümlendirilmesi sonucu elde edilen takım yolu eğrileri gözlenmektedir.



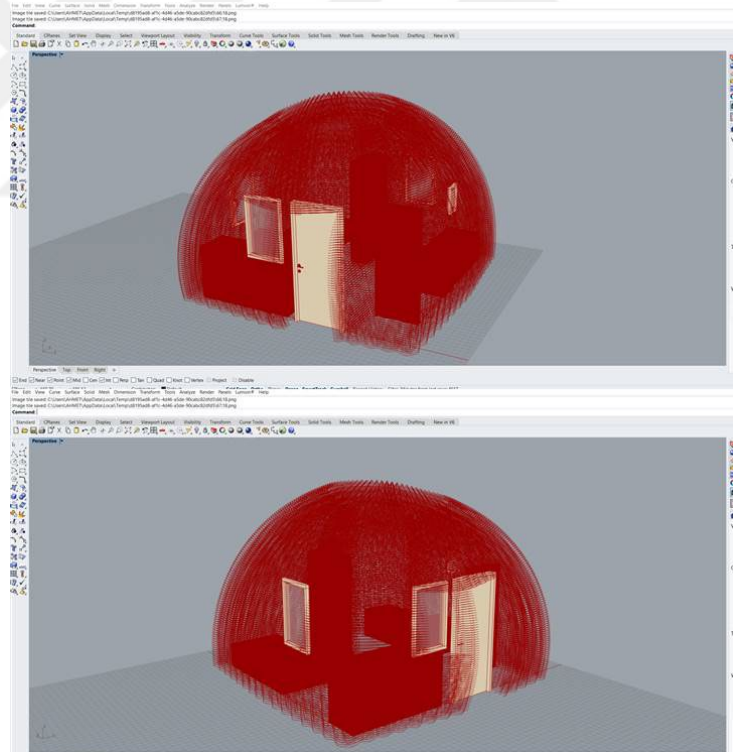
Şekil 5.53. İç duvarların bölümlendirilmiş takım yolu eğrileri

Şekil 5.54’te tüm model için geliştirilen algoritma ve Rhino üç boyutlu görüntüleme ara yüzünde üç boyutlu baskılama için üretim takım yolları gözlenmektedir.



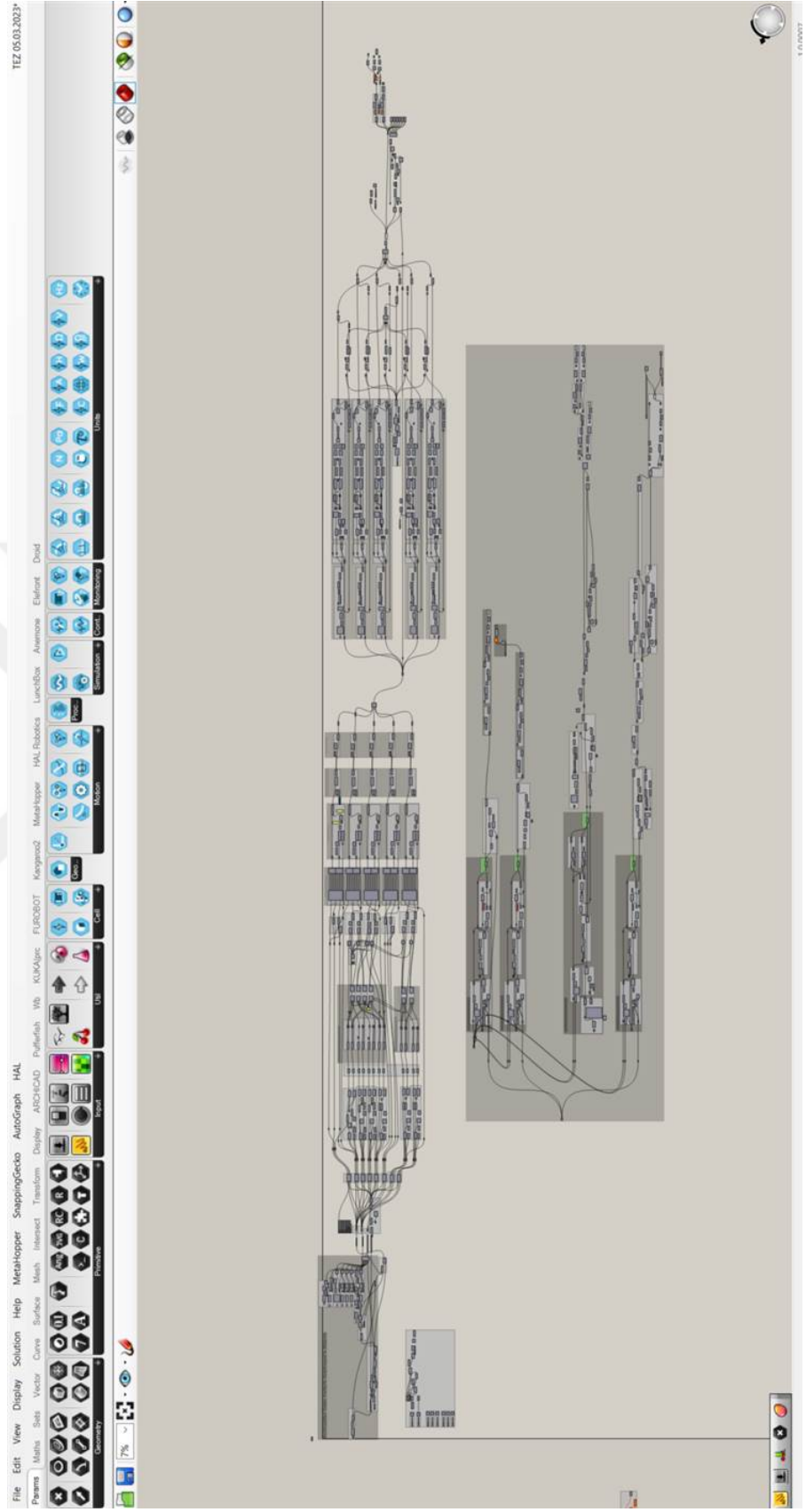
Şekil 5.54. Tüm model algoritması ve iç dış ilişkisi

Şekil 5.55'te oluşturulan modelin 2 farklı perspektiften görünümü gözlenmektedir.



Şekil 5.55. Barınma biriminin perspektif görüntüleri

Şekil 5.56'da barınma birimi için geliştirilen tüm algoritma gözlenmektedir.



Şekil 5.56. Tüm algoritma

6. SONUÇ VE ÖNERİLER

Bu çalışmada, öncelikle parametrik tasarım kavramının tarihsel süreçte detaylı incelenmesi yapıp bu doğrultuda parametrik tasarım kavramının bilgisayar teknolojileri ile bağdaştığı nokta saptanmıştır. Sonrasında, otomasyon fikrinin doğuşundan yola çıkılarak bir alete bir işin önceden tanımlanması ve bunun gerçekleştirilmesi gayesiyle ortaya çıkan ve gelişen teknolojilerin günümüz bilgisayar sistemlerinin altyapısını oluşturduğu gözlemlenmiştir.

Bu kuramsal incelemeler özüksendikten sonra çalışmanın ana amacı olan parametrik tasarım araçları ile oluşturulacak tasarımın, parametrik düşünce ilkesine bağlı olması sağlanmıştır.

20. yüzyılda ortaya çıkan ve kuramında bu tezde üretilen parametrik düşünce kavramı ile örtüşen "biçim işlevi takip eder" anlayışı; 21. yüzyıl mimarlık ortamında, bilgisayar teknolojisinin devrimsel gelişiminin etkisiyle evrimini tamamlamıştır. Bu evrim sonucunda günümüze gelindiğinde, biçimin sadece fonksiyona dayalı oluşturulduğu değil aynı zamanda optimizasyona, verimliliğe ve parametrelere bağlı olarak geliştirilen tasarımlar ile biçim işlevi takip eder anlayışı uygulanmaktadır. Bu tez kapsamında oluşturulan algoritmanın kullanımı ile biçimin işlevi takip ettiği ve üretimde fazla malzeme kullanılmadan yekpare bir kabuk yapısı ile oluşturulan barınma biriminin parametrik modellemesi üretilmiştir.

Yapı ölçeğinde eklemeli imalat teknolojileri incelendiğinde ölçek bazında bazı sınırlılıklar olduğu gözlemlenmiş ve bu sınırlılıklara bağlı kalarak üretilen tasarımın en az alanda maksimum verimlilik ilkesine göre tasarlanması gerektiği belirlenmiştir. Bu bağlamda Le Corbusier'in Le Cabanon'u incelenmiş ve tez kapsamında üretilen tasarımda bu yapıdan faydalanılarak yeni bir model oluşturulmuştur.

Katmanlı imalat teknolojisinin kalıplara ihtiyaç duymaması, geleneksel yöntemlerin getirdiği sınırlamaları ortadan kaldırarak, yerinde üretim ve yapı tasarımında kişiselleştirmeyi mümkün kılmış ve form çeşitliliğini artırmıştır. Ancak, bu teknolojinin yapı ölçeğindeki kullanımında kısıtlar bulunmaktadır; özellikle kullanılabilir malzeme çeşitliliğinin az olması bir dezavantajdır. Yapı ölçeğinde katmanlı imalat teknolojisi ile üretilebilecek birimin tasarlanması için parametrik tasarım tekniklerinin sunduğu avantajlar, geleneksel modelleme yazılımlarından farklı olarak modelin optimizasyonun geliştirilmesine fayda sağlamıştır. Parametrik modelde yapılan değişikliklerin, parametrik tasarım araçlarının dinamik veri güncelleme yeteneği sayesinde sonuç ürünün doğrudan

test edilme imkânı, çevresel koşullara ve farklı gereksinimlere uygun, serbest formlu tasarımlar oluşturmaya mümkün kılmaktadır.

Katmanlı imalat örnekleri incelendiğinde ise farklı üretim tekniklerine dayanan baskılama teknolojilerinde yapı ölçeğinde üretilen tasarımların genellikle duvarlarının katmanlı üretimle gerçekleştirildiği, üst örtüsünün ise harici elemanlarla tamamlandığı gözlemlenmiştir. Bu noktada tezde üretilen tasarımda robot kolların faydalarından biri olan düzlemsel olmayan baskılama yapabilme sayesinde yapının kabuğu bir bütün olarak üretilebilmekte ve bu durum hem malzeme kullanımı hem de tek metotla yapının üretiminin gerçekleşmesini sağlayabilmektedir.

Bilgisayar teknolojilerinin tasarım süreçlerinde kullanımında, Sketchpad ile başlayan kalem ve kâğıdın/çizim ekranının yerini alan fare ve klavyenin sınırları aşarak, bilgisayarların çalışma prensibinden ve artan bir ivmeyle gelişen hesaplama yeteneklerinden faydalanılmalı ve tasarımlar bu şekilde üretilmelidir. Bu bağlamda, "tasarımcı" tanımının yakın gelecekte genişleyeceği ve dönüşeceği öngörülmektedir. Tasarımcılar, artık sadece nihai tasarımı değil, tasarım süreçlerini de tasarlamalı ve bu şekilde parametrelere dayalı sonsuz tasarımlar üretebilen algoritmalar geliştirmelidirler.

Bu çalışmada geliştirilen algoritmadan elde edilebilecek veriler sayesinde amorf bir geometrinin parametrik modellemenin getirileri sayesinde robot kol ile eklemeli imalat yöntemiyle üretiminin takım yollarının elde edilmesinde hesaplamalı tasarım araştırmacılarına fayda sağlaması hedeflenmiştir.

KAYNAKLAR

- Akbulut, M. R. ve Akay, Z. (2013). Jeanneret Harikalar Diyarında: Le Corbusier'nin Doğu Yolculuğu ve İstanbul. *MSGSÜ Sosyal Bilimler*, (8), 81–93. <https://dergipark.org.tr/tr/pub/msgsusbd/issue/46973/589473> adresinden erişildi.
- Altın, M. A. (2012, Ağustos). *Özgün iç mekan bileşenlerinin üretiminde bilgisayar destekli tasarım ve üretim. (Sanatta Yeterlik Tezi)*. Eskişehir Anadolu Üniversitesi Güzel Sanatlar Enstitüsü, Eskişehir. <https://acikbilim.yok.gov.tr/handle/20.500.12812/328310> adresinden erişildi.
- Arndt, A. B. (1983). Al-Khwarizmi. *Mathematics Teacher* (ss. 668–670).
- Assasi, R. (2019). Parametric Design, A Historical and Theoretical Overview. *In Proceedings of the International Conference on Emerging Technologies in Architectural Design (ICETAD2019)*. Toronto: Department of Architectural Science of Ryerson University.
- ASTM. (2015). F2792 Standard Terminology for Additive Manufacturing Technologies, (Withdrawn 2015). doi:<https://doi.org/10.1520/F2792-12A>
- Autodesk. (2023). 7-machine-axes. 28 Aralık 2023 tarihinde <https://www.autodesk.com/products/fusion-360/blog/fr/generative-design-in-fusion-360-addresses-manufacturing-skills-gap/7-machine-axes/> adresinden erişildi.
- Baker, P. (1980). The Frescoes of Amra. *Saudi Aramco World*, 31(4), 22-25.
- Balat, İ. (2019). 3D Yazıcı Çeşitleri ve Şekilleri Nelerdir? Hangisi Daha İyi? (Kartezyen, Delta, Polar ve Scara Farkı). 28 Aralık 2023 tarihinde <https://kerteriz.net/3d-yazici-cesitleri-ve-sekilleri-nelerdir/> adresinden erişildi.
- Bales, R. (2023). Basile Bouchon — Biography, History and Inventions. 27 Aralık 2023 tarihinde <https://history-computer.com/basile-bouchon-biography-history-and-inventions/> adresinden erişildi.
- Barutçuoğlu, E. (2001). *Robotların Tarihçesi*. Boğaziçi Üniversitesi.
- Bcn3d. (2020). When Was 3D Printing Invented? The History of 3D Printing. 27 Aralık 2023 tarihinde <https://www.bcn3d.com/the-history-of-3d-printing-when-was-3d-printing-invented/> adresinden erişildi.
- Berkin, G. (2021). *MODULOR ve Le Corbusier'in Kulübesi*. İstanbul: YEM Yayınevi.
- Burry, J., & Burry, M. (2010). *The new mathematics of architecture*. London: Thames & Hudson.
- Burry, M. (2007). “Setting the Scene.” In Gaudí Unseen: Completing the Sagrada Família, edited by Mark Burry. *Berlin: Jovis.*, 10–58.

- Buswell, R. A., Leal de Silva, W. R., Jones, S. Z. ve Dirrenberger, J. (2018). 3D printing using concrete extrusion: A roadmap for research. *Cement and Concrete Research*, 112, 37–49. doi:10.1016/J.CEMCONRES.2018.05.006
- Çalışkan, M. G. (2015). Üç Boyutlu Yazıcılar ve Gelecekte Yaratacağı Olası Fikri Haklar Çatışmaları. *FMR Dergisi*, 1, 62. <https://silo.tips/download/boyutlu-yazicilar-ve-gelecekte-yar-atacai-olasi-fkr-haklar-atimalari> adresinden erişildi.
- Computerhistory. (2024). Rancho Arm. <https://www.computerhistory.org/revolution/artificial-intelligence-robotics/13/293/1278> adresinden erişildi.
- Davis, D. (2013a). *Modelled on software engineering: flexible parametric models in the practice of architecture*. RMIT University.
- Davis, D. (2013b). A History of Parametric. <https://www.danieldavis.com/a-history-of-parametric/> adresinden erişildi.
- Dino, I. G. (2012). Creative design exploration by parametric generative systems in architecture. *Metu Journal of the Faculty of Architecture*, 29(1), 207–224. doi:10.4305/METU.JFA.2012.1.12
- Earls, A. (2011). Origins: A Break from the Grind. 27 Aralık 2023 tarihinde <https://todaysmachiningworld.com/magazine/origins-a-break-from-the-grind/> adresinden erişildi.
- Earnshaw, S. (1839). On the Nature of the Molecular Forces which Regulate the Constitution of the Luminiferous Ether. in *Transactions of the Cambridge Philosophical Society* 7, 97–122.
- Eltaweel, A. ve Su, Y. (2017). Parametric design and daylighting: A literature review. *Renewable and Sustainable Energy Reviews*, 73, 1086–1103. doi:10.1016/J.RSER.2017.02.011
- Eplus3D. (2023). Application of 3D Printing House. 30 Aralık 2023 tarihinde <https://www.eplus3d.com/application-of-3d-printing-house.html> adresinden erişildi.
- Etimolojiturkçe. (2023). Parametre. 28 Aralık 2023 tarihinde <https://www.etimolojiturkce.com/kelime/parametre> adresinden erişildi.
- Fang, G., Zhong, S., Zhong, Z., Zhang, T., Chen, X. ve Wang, C. C. L. (2020). Reinforced FDM: Multi-Axis Filament Alignment with Controlled Anisotropic Strength. doi:10.1145/3414685.3417834
- Ganesan, A. (2015). Design and Control of 3-DOF Articulated Robotic Arm using LabVIEW and NI-myRIO. *International Journal Of Innovative Research In Electrical, Electronics, Instrumentation And Control Engineering*, 3(3). Doi:10.17148/Ijireeice.2015.3342

- Gardiner, J. (2011). *Exploring the emerging design territory of construction 3D printing - project led architectural research*.
<https://researchrepository.rmit.edu.au/esploro/outputs/doctoral/Exploring-the-emerging-design-territory-of/9921861544001341> adresinden erişildi.
- Gershenfeld, N. (2012). How to Make Almost Anything The Digital Fabrication Revolution.
https://acervo-digital.espm.br/clipping/20121109/how_to_make_almost_anything-8.pdf adresinden erişildi.
- Gibson, I., Rosen, D., Stucker, B. ve Khorasani, M. (2021). Development of Additive Manufacturing Technology. *Additive Manufacturing Technologies*, 23–51. doi:10.1007/978-3-030-56127-7_2
- GooglePatents. (2023, 5 Mayıs). Motor controlled apparatus for positioning machine tool. 27 Aralık 2023 tarihinde 27 Aralık 2023 adresinden erişildi.
- Gosselin, C., Duballet, R., Roux, P., Gaudillière, N., Dirrenberger, J. ve Morel, P. (2016). Large-scale 3D printing of ultra-high performance concrete - a new processing route for architects and builders. *Materials and Design*, 100, 102–109. doi:10.1016/j.matdes.2016.03.097
- Groover, M. P. (2012). *Fundamentals of Modern Manufacturing*. JOHN WILEY & SONS, INC.
[https://www.fcusd.org/cms/lib/CA01001934/Centricity/Domain/4529/Fundamentals of Modern Manufacturing Materials Processes and Systems 4th Edition.pdf](https://www.fcusd.org/cms/lib/CA01001934/Centricity/Domain/4529/Fundamentals%20of%20Modern%20Manufacturing%20Materials%20Processes%20and%20Systems%204th%20Edition.pdf) adresinden erişildi.
- Hager, I., Golonka, A. ve Putanowicz, R. (2016). 3D Printing of Buildings and Building Components as the Future of Sustainable Construction? *Procedia Engineering*, 151, 292–299. doi:10.1016/J.PROENG.2016.07.357
- Hemonet, M. (2023). La villa E-1027 d'Eileen Gray, une icône de l'architecture moderne. 9 Ocak 2023 tarihinde <https://www.admagazine.fr/article/villa-e-1027-eileen-gray-cap-martin> adresinden erişildi.
- Hubs. (2023). What is 3D printing? 28 Aralık 2023 tarihinde <https://www.hubs.com/guides/3d-printing/> adresinden erişildi.
- Inspirationtuts. (2021). Sketchpad The First 3D Software in History.
<https://inspirationtuts.com/sketchpad-the-first-3d-software-in-history/> adresinden erişildi.
- Jain, S. (2021). Robotic Arms: A Brief. 27 Aralık 2023 tarihinde https://erc-bpgc.github.io/blog/blog/robotic_arms/ adresinden erişildi.
- Kortan, E. (1983). *Le Corbusier Gözüyle Türk Mimarlık ve Şehirciliği*. Ankara: ODTÜ.
- Kurnalı, M. ve Koca, D. (2018). Mimarlıkta Bir Mekan Üretim Aracı Olarak Kabin. *Ulakbilge*, 22(8), 297–319. doi:10.7816/ulakbilge-06-22-04

- Leach, N. (2009). Swarm Urbanism. *Architectural Design*, 79(4), 56–63. doi:10.1002/AD.918
- Leach, N. (2014). 3D Printing in Space. *Architectural Design*, 84(6), 108–113. doi:10.1002/AD.1840
- Leslie, J. (1821). Geometrical Analysis and Geometry of Curve Lines. *Edinburgh: W. & C., Second edi.*
- Lim, S., Le, T., Webster, J. ve Buswell, R. A. (2009). Fabricating construction components using layer manufacturing technology. In *Global Innovation in Construction Conference*, 512–520. https://www.researchgate.net/publication/312919469_Fabricating_construction_components_using_layer_manufacturing_technology adresinden erişildi.
- Lopes, J., Paio, A. ve Sousa, J. (2014). *Parametric Urban Models Based On Frei Otto's Generative Form-Finding Processes*, 595–604.
- Ma, G. W., Wang, L. ve Ju, Y. (2018). State-of-the-art of 3D printing technology of cementitious material—An emerging technique for construction. *Science China Technological Sciences*, 61(4), 475–495. doi:10.1007/S11431-016-9077-7/METRICS
- Mohammadi, G. (2019). History of CNC Machining. 27 Aralık 2023 tarihinde <https://medium.com/cnc-life/history-of-cnc-machining-part-1-2a4b290d994d> adresinden erişildi.
- Nichols, M. R. (2019). Why Was the Robotic Arm Invented? 27 Aralık 2023 tarihinde <https://interestingengineering.com/innovation/why-was-the-robotic-arm-invented> adresinden erişildi.
- Rael, R. ve San Fratello, V. (2011). Developing Concrete Polymer Building Components for 3D Printing. *ACADIA proceedings*, 152–157. doi:10.52842/CONF.ACADIA.2011.152
- Revit Technology Corporation. (2000). Revit Technology Corporation Launches Industry's First Parametric Building Modeler. *Press release, April 5. Waltham.*
- Revit Technology Corporation. (2001). Technical Overview. http://web.archive.org/web/20010206204349fw_/http://www.revit.com/cornerstone/content_technical.html adresinden erişildi.
- Robots.com. (2017). Industrial Robot History. 27 Aralık 2023 tarihinde <https://www.robots.com/articles/industrial-robot-history> adresinden erişildi.
- Schipper, R., Costanzi, C. B., Bos, F., Ahmed, Z. ve Wolfs, R. (2017). Double curved concrete printing: printing on non-planar surfaces. *SPOOL*, 4(2), 17–21. doi:10.7480/SPOOL.2017.2.1920

- Schmitt, B. M., Zirbes, C. F., Bonin, C., Lohmann, D., Lencina, D. C. ve Da Costa Sabino Netto, A. (2017). A comparative study of cartesian and delta 3d printers on producing PLA parts. *Materials Research*, 20, 883–886. doi:10.1590/1980-5373-MR-2016-1039
- Schumacher, P. (2008). Parametricism as Style – Parametricist Manifesto. 27 Aralık 2023 tarihinde [http://www.patrikschumacher.com/Texts/Parametricism as Style.htm](http://www.patrikschumacher.com/Texts/Parametricism%20as%20Style.htm) adresinden erişildi.
- Schumacher, P. (2009). Parametricism: A new global style for architecture and urban design. *Architectural Design*, 79(4), 14–23. doi:10.1002/AD.912
- Schumacher, P. (2011). The Autopoiesis of Architecture. A new framework for architecture | Request PDF. 27 Aralık 2023 tarihinde https://www.researchgate.net/publication/303990577_The_Autopoiesis_of_Architecture_A_new_framework_for_architecture adresinden erişildi.
- Sidoli, D. ve Berggren, J. L. (2007). The Arabic version of Ptolemy’s Planisphere or Flattening the Surface of the Sphere: Text, Translation, Commentary. *The Arabic version of Ptolemy’s Planisphere or Flattening the Surface of the Sphere: Text, Translation, Commentary* içinde (s. 139).
- Sözen, M. ve Tanyeli, U. (2010). *Sanat Sözlüğü*. Remzi Kitabevi.
- Sutherland, I. (1963). *Sketchpad: A Man-Machine Graphical Communication System*. Massachusetts Institute of Technology.
- Sutherland, I. (1964). Sketch pad a man-machine graphical communication system. *Proceedings - Design Automation Conference*. doi:10.1145/800265.810742
- Tedeschi, A. ve Wirz, F. (2014). *AAD - Algorithms-aided design: parametric strategies using grasshopper*.
- Teresko, J. (1993). Parametric Technology Corp.: Changing the way Products are Designed. *Industry Week*.
- Turan, İ. (2019). *Temsil Tektonik Gerilimleri: Mies Van Der Rohe*. Bilgi Üniversitesi.
- Türk Dil Kurumu Sözlükleri. (2023). Parametre. <https://sozluk.gov.tr/> adresinden erişildi.
- Turney, D. (2022). The History of Industrial Robots, From Single Taskmaster to Self-Teacher. 27 Aralık 2023 tarihinde <https://www.autodesk.com/design-make/videos/history-of-industrial-robots> adresinden erişildi.
- Ulubeyli, S. (2022). Lunar shelter construction issues: The state-of-the-art towards 3D printing technologies. *Acta Astronautica*, 195, 318–343. doi:10.1016/J.ACTAASTRO.2022.03.033

- Wallén, J. (2008). Technical report from Automatic Control at Linköpings universitet The history of the industrial robot.
- Weisberg, D. (2008). The Engineering Design Revolution: The People, Companies and Computer Systems that Changed Forever the Practice of Engineering. <https://web.archive.org/web/20230204220210/http://cadhistory.net/> adresinden erişildi.
- Westerlund, L. (2000). *The Extended Arm of Man – A History of the Industrial Robot*. Informationsförlaget, Stockholm, Sweden.
- Wikipedia. (2023). EBCDIC. 27 Aralık 2023 tarihinde <https://en.wikipedia.org/wiki/EBCDIC> adresinden erişildi.
- Yıldırım, E. (2019). *Yaya Davranışlarının Algoritmik Yaklaşım ile Analizi ve Tasarım Aktarılması*. Dokuz Eylül Üniversitesi.
- Zexin, S. ve Mei, H. (2017). Robotic Form-Finding and Construction Based on the Architectural Projection Logic. *IOP Conference Series: Materials Science and Engineering*, 216(1). doi:10.1088/1757-899X/216/1/012058