



## ENDÜSTRİ ÜRÜNLERİ TASARIMI EĞİTİMİNDE HESAPLAMALI TASARIM YAKLAŞIMI

COMPUTATIONAL DESIGN APPROACH IN INDUSTRIAL PRODUCT DESIGN EDUCATION

Erkut ERYAYAR\*

Sanat-Tasarım Dergisi 2017, Sayı: 8 ISSN: 1309-2235 ss.15-19 DOI: 10.17490/Sanat.2018.16

### Öz

Dijital teknolojiler, tüm bilim ve sanat alanlarında olduğu gibi Endüstri Ürünleri Tasarımı Meslek Disiplini eğitiminde de yoğun bir şekilde kullanılmaktadır. Eğitim kurumları, ders içeriklerine dijital tasarım ve üretim teknolojilerini içeren dersler ekleyerek öğrencilerini profesyonel hayata hazırlamaktadırlar. Tasarlama yöntemlerinde, sadece teknik birer çizim aracı olarak kullanılan bilgisayarlar, özellikle içinde bulunduğumuz zaman diliminde hızlı bir biçimde gelişen donanım ve yazılımların yardımı ile mimarlık, mühendislik, grafik ve endüstriyel tasarım gibi meslek disiplinlerinde, güçlü birer tasarım enstrümanı olarak yerini almıştır. Bilgisayar destekli tasarım ve üretim (CAD/CAM) teknolojilerinde sıklıkla kullanılmaya başlanan parametrik ve hesaplmalı tasarım modelleme sistemleri, yaratıcı süreci hızlandırarak, tasarımcılara alternatif form üretme konusunda sonsuz olasılık sunmaktadır. Bu çalışmanın amacı hesaplmalı tasarım yaklaşımının, tasarım eğitimi içindeki yerini ve önemini incelemektir.

**Anahtar Sözcükler:** Hesaplmalı tasarım, Eğitimde Hesaplmalı Tasarım, Endüstri Ürünleri Tasarımı Eğitimi.

### Abstract

Digital technologies are used extensively in the Industrial Product Design Professional Discipline as well as in all fields of science and arts. Educational institutions prepare their students for professional life by adding courses related to digital design and production techniques. In design methods, computer, which was only used as a technical drawing tool, has become a powerful design instrument in professional disciplines such as architecture, engineering, graphic and industrial design with the help of today's rapidly improving hardware and software. Parametric design modeling systems, which began to be frequently used in computer aided design and manufacturing (CAD / CAM) technologies, offer infinite possibilities for creating alternative forms to designers by accelerating the creative process. The aim of this study is to examine the place and importance of computational design approach in design education.

**Keywords:** Computational Design, Computational Design in Education, Industrial Product Design Education.

### Giriş

Ürün tasarımı süreci, tasarımcının pek çok alternatif üretmesi ve geliştirmesi gereken bir süreçtir. Bu sürecin başlangıcında pek çok tasarımcı

kağıt ve kalem kullanır. Tasarım eğitimi veren pek çok okul, eğitim sürecinin ilk yıllarında Bilgisayar Destekli Tasarım araçlarının kullanılmasının yerine, kağıt ve kalem kullanımını teşvik ederek öğrencilerin fikirlerini kısa zamanda sunmaları pratiklerini geliştirmeye çalışır. Eğitim kurumlarında genel olarak kabul görmüş ve desteklenen bu uygulamanın amacı, öğrencilere geliştirecekleri tasarım fikirlerini ve alternatiflerini oluşturmak için gerekli düşünce pratiğini ve hızını kazandırmaktır. Bu süreci içselleştiren tasarımcı adayı öğrenci, eğitim sürecinin ilerleyen zamanlarında tasarlama aracı olarak bilgisayar destekli tasarım araçlarını kullanmaya başlar. Hesaplmalı tasarım yaklaşımı, günümüzde hızla gelişen ve Parametrik modelleme imkanları sunan yazılım ve donanımlar sayesinde Mühendislik ve Mimarlık gibi pek çok tasarım disiplininde olduğu gibi Endüstri Ürünleri Tasarımı alanında da gerek eğitim sürecine gerekse tasarım sürecine yardımcı olmak için uygulama ve araştırmada standart haline gelmektedir.

### Hesaplama (Computation) Kavramı

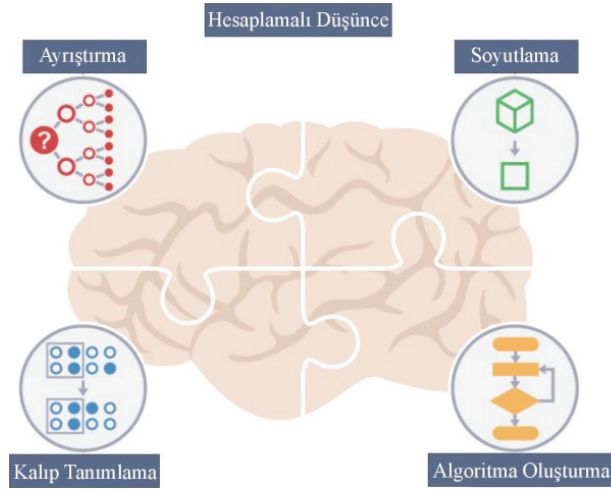
Dilimize "Hesaplama" olarak tercüme edebileceğimiz Computation kelimesinin etimolojik kökenine bakıldığında Latince computare'den geldiğini görülmektedir. Kökündeki 'com' beraber, ile (with) anlamına gelirken 'putare' ise açıklığa kavuşturmak (clear up), yerli yerine oturtmak (settle), hesaba katmak/hesaplaşmak (reckon) anlamlarına gelir; yani computare hepsini beraber açıklığa kavuşturmak, yerli yerine oturtmak ve hesaba katmak anlamlarına gelmektedir. Computation, kelime olarak 15.yy'da ilk kez kullanılmıştır (Çinici, 2012). Bu bağlamda köken olarak kelimenin, bilgisayar teknolojilerinin gelişmesiyle kullanıma girmediği, daha çok bir düşünce sistematiğini ve kavramı tanımlamak amacı ile türetildiği anlaşılmaktadır.

### Hesaplmalı Düşünce (Computational Thinking) / Hesaplmalı Tasarım (Computational Design)

Hesaplmalı düşünce yaklaşımının bilgisayar bilimi içerisinde uzun bir geçmişi vardır. 1950'lerde ve 1960'larda "algoritmik düşünce" olarak bilinen bu düşünce sistematiği, birtakım girdilerin bir çıktıya dönüşümü olarak problemleri formüle etmek, dönüşümleri gerçekleştirmek üzere algoritmalar aramak için zihinsel bir yönelim anlamına gelmektedir. Günümüzde bu terim, birçok soyutlama seviyesiyle düşünmeyi, algoritmalar geliştirmek için matematiğin kullanımını ve bir çözümün farklı boyutlardaki sorunların ne kadar iyi ölçeklendiğini incelemek üzere genişletilmiştir (Denning, 2011). Hesaplmalı düşünce, mantık çerçevesinde eldeki verileri analiz etme ve bu doğrultuda belirlenen algoritmaları kullanarak açık

uçlu problemlere çözümler üretmeyi içeren bir problem çözme sürecidir. Her ne kadar ilk kullanım alanı, bilgisayar uygulamalarının geliştirilmesi olsa da, günümüzde beşeri bilimler, matematik ve tasarımın pek çok alanı da dahil olmak üzere tüm disiplinlerde problem çözmeyi desteklemek için kullanılmaya başlanmıştır.

Hesaplamalı düşünce, çevremizden edindiğimiz verileri sınıflandırmada, kurgulamada ve sistematize etmekte yapısal bir rol üstlenmektedir. Günümüzde, hesaplamalı teori ile çeşitli sistemlerin tanımlanmasında sabit denklemlerden çok, 'süreç'in kurgulanması öne çıkmaktadır. Böylece doğadaki süreç ya da oluşumlar hesaplamalı teori ile kurgulanıp tanımlandığında; sistem statik /sabit sayısal matematiksel denklemler yerine, çok boyutlu dinamik/hareketli/devinimsel ilişkileriyle ve bağlantılarıyla ele alınabilmektedir (Erdoğan & Sorguç, 2011).



Şekil 1. Hesaplamalı Düşünce Yöntemi (<https://www.bbc.co.uk/>) (Telif hakkı yazara aittir)

Hesaplamalı düşünme yönteminde dört temel teknik vardır. Bunlar, ayrıştırma, kalıp tanımlama, soyutlama ve algoritma oluşturmadır. Ayrıştırma adımı, karmaşık bir sorunu veya sistemi daha küçük, daha yönetilebilir parçalara bölme sürecidir. Kalıp tanımlama adımı ile tanımlanan problemler arasında ve içinde benzerlikler aranır. Soyutlama ile yalnızca önemli bilgilere odaklanarak alakasız ayrıntıları yok sayılmaya çalışılır. Elde edilen bulgular doğrultusunda algoritmalar oluşturularak soruna adım adım bir çözüm geliştirme veya sorunu çözmek için bir yol haritası çizilir. Endüstri ürünleri tasarımı sürecinde Hesaplamalı düşünme yöntemi ile oluşturulan algoritma, parametrik tasarlama yöntemi içinde ele alınır ve hayata geçirilir.

## Parametrik Tasarım

Parametre kelimesi Antik Yunanca kökenli olup, *Parametron*; "para" ikinci, yardımcı ve "metron" ölçmek kelimelerinin birleşiminden oluşmaktadır. Dilimizde parametre kelimesi matematiksel bir terim olarak kullanılmakta ve değişken anlamına gelmektedir (Parametre, t.y.). Bilgisayar teknolojilerinde ise bir dizi komutun, sisteme girilen çeşitli veriler üzerinden işlem yapmasıyla ilgili bir terimdir.

Parametrik ya da hesaplamalı tasarım, tasarım amacı ve tasarım yanıtı arasındaki ilişkiyi birlikte tanımlayan, kodlayan ve netleştiren parametrelerin ve kuralların ifadesine olanak tanıyan algoritmik düşünceye dayalı bir süreçtir. Gelişen dijital teknolojiler sayesinde süreç, tasarımcıların etkili bir tasarıma ulaşabilmek için, insanın tek başına yaratabileceğinin ötesinde, yeni seçenekler üretmelerini sağlayabilecek kapasiteye gelmiştir. Parametrik tasarım kavramı, bilgisayar destekli tasarım terminolojisinde, bir formu unsurlar arasındaki ilişkileri kullanarak değişken değerler ile ifade etmek olarak tanımlanabilir. Parametrik tasarım sürecinde tasarımı etkileyecek verilerin parametreler olarak belirlenmesi ve organizasyonu esastır. Hangi verinin diğeriyle nasıl ilişkili olduğunun sayısal ve geometrik olarak tanımlandığı ve sınırlamaların belirtildiği bir tasarım stratejisi kurulur. Bu tür bir ilişkisel model bir kez kurulduğunda parametre değerleri değiştirilerek tüm olası durumları araştırılabilir ya da türetilir (Akipek, 2007).

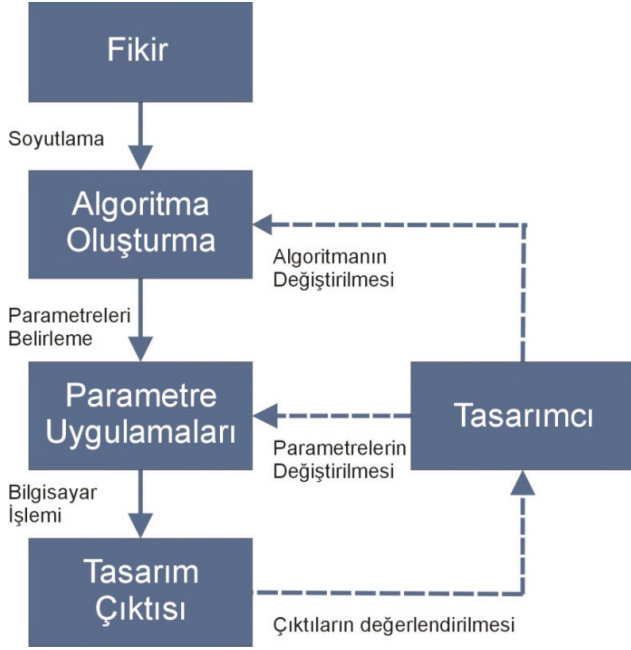
Oluşturulan parametrik yapı (Parametric Structure), bir modelleme sisteminin nesnelerin oluşturulması, düzenlenmesinde kullanılan değişken ve kısıtlarla sınırlı kalmayıp, hem matematiksel hem de topolojik ilişkileri ifade eden genel bir tanımlamadır. Sistem çalışma mantığı nedeniyle ilişkisel yapı (Relational Structure) olarak da adlandırılmaktadır (Uzun, 2001).

Parametrik Tasarlama süreci başından sonuna algoritmik bir süreçtir. "Algoritma, matematikte ve bilgisayar biliminde bir işi yapmak için tanımlanan, bir başlangıç durumundan başladığında, açıkça belirlenmiş bir son durumda sonlanan, sonlu işlemler kümesidir. Yani belli bir problemi çözmek veya belirli bir amaca ulaşmak için çizilen yola algoritma denir. Genellikle programlamada kullanılır ve tüm programlama dillerinin temeli algoritmaya dayanır. Aynı zamanda algoritma tek bir problemi çözecek davranışın, temel işleri yapan komutların veya deyimlerin adım adım ortaya konulmasıdır ve bu adımların sıralamasına dikkat edilmelidir" (Algoritma, t.y.). Bir başka deyişle tasarımcı, kendi oluşturduğu algoritmik tasarlama süreci içerisinde zamanda ileri ve geri yolculuk yapabilmektedir. Parametrik tabanlı tasarlama metodları günümüzde ürün tasarımcıları, mimarlar, grafikerler ve heykeltıraşlar gibi pek çok farklı alanda çalışan tasarımcılar tarafından sıklıkla kullanılmaktadır. Parametrik tabanlı bilgisayar destekli tasarım programları, kullanıcıya temel olarak kullanabileceği bir form yaratmasını ve bu formu istediği gibi düzenlemesini sağlar. Parametrik tabanlı programların çalışma mantığı, unsur (feature), hafıza (history) ile bunların birbirleri ile ilişkileri (relations) temelli olduğu için, tasarımcı tasarlama sürecinin en başından sonuna kadar sürecin her aşamasına istediği zaman müdahale edebilir ve değiştirebilir.

Parametrik tabanlı bilgisayar destekli tasarım programları arasında gerek tasarımcılar, gerekse konu ile ilgilenen sanatçılar tarafından en sık kullanılan yazılımlardan biri Rhinoceros 3D ile bu yazılım ile beraber çalışan Grasshopper eklentisidir.

Parametrik tasarım sürecinde, tasarımcı çoğu zaman değişkenlerin birbirleri ile ilişkilerini ve değerlerini değiştirerek yeni formlar arar. Tasarımcı bu süreçte bazen bütün parametreleri değiştirebileceği gibi, bazen istediklerini sabit tutarak sonuca ulaşmaya çalışır. Parametrik modelleme programları, kullanıcıya temel olarak kullanabileceği bir form yaratmasını ve bu formu istediği gibi düzenlemesini sağlar. Parametrik tabanlı

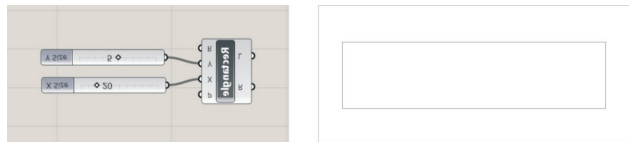
programların çalışma mantığı, tasarımcının ortaya koyduğu soyut fikirler doğrultusunda oluşturulan algoritmanın içinde yer alan parametreler yardımı ile somut bir biçimin ortaya konmasıdır. Tasarımcı, tasarlama sürecinin en başından sonuna kadar sürecin her aşamasına istediği zaman müdahale edebilir ve değiştirebilir.



Şekil 2. Hesaplmalı Tasarım Süreci (Telif hakkı yazara aittir)

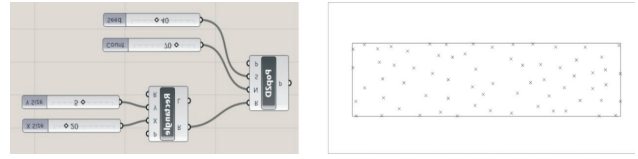
Çalışmada Endüstri Ürünleri Tasarımının hesaplmalı düşünce pratiği ile ilişkisi, biçimi araştırma ya da geliştirme sürecinde, geometrinin rolü üzerinden başlayan bir yaklaşım doğrultusunda ele alınacaktır. Bu bağlamda yapılan çalışmada, Rhinoceros 3D programı ve bu program altında çalışan Grasshopper eklentisi ile oluşturulan parametrik algoritma sistemi kullanılarak, tasarım sürecinde kullanılacak form alternatiflerinin üretim araştırması gerçekleştirilecektir.

Bu bağlamda hazırlanan çalışmada iki örnek tasarım uygulaması yapılmıştır. İlk tasarım, parametrik olarak hazırlanan bir yüzey deseninin üç boyut kazandırılarak bir ürüne döndürülmesi çalışmasıdır. Formu oluştururken Grasshopper 3D eklentisi altında çalışan "Voronoi" hücre oluşturma algoritması temel alınacak ve bu algoritma ile beraber çalışan alt parametreler tanımlanacaktır.



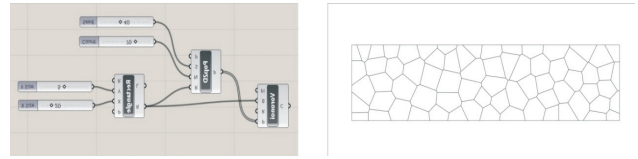
Şekil 3. Rectangle komutu uygulaması (Telif hakkı yazara aittir)

Yapılan çalışmanın birinci adımında (Bkz. Şekil 3), Rectangle komutunu oluşturularak X ve Y eksenleri değerlerini belirliyoruz.



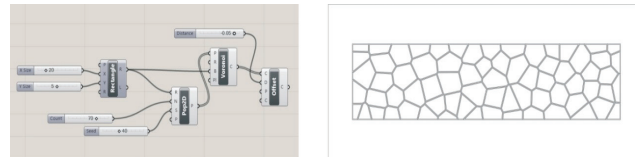
Şekil 4. Populate 2D komutu uygulaması (Telif hakkı yazara aittir)

İkinci adımda (Bkz. Şekil 4) oluşturduğumuz alanı Populate 2D komutu ile ilişkilendirerek noktalar ile dolduruyoruz. Bunun için Populate 2D kutucuğu girdi bölgesinde yer alan Region (R) bağlacına oluşturduğumuz dörtgeni bağlıyoruz. Kaç adet nokta olması istiyorsak Populate 2D kutucuğu girdi bölgesinde yer alan Number bağlacına oluşturduğumuz Count (sayı) çubuğunu bağlıyoruz. Seed kutucuğuna rastgele ekleme faktörünü girerek noktalarımızı alan içine dağıtıyoruz.



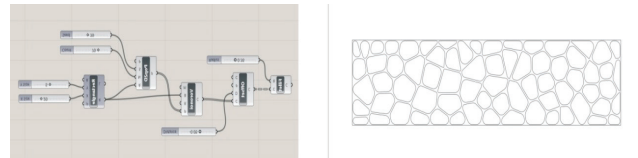
Şekil 5. Voronoi komutu uygulaması (Telif hakkı yazara aittir)

Üçüncü adımda (Bkz. Şekil 5) rastgele bir mantık çerçevesinde oluşturduğumuz noktaları temel olarak Voronoi hücrelerimizi oluşturuyoruz. Bunun için Voronoi hücre kutucuğu girdi bölgesinde yer alan Points (P) bağlacına Populate 2D kutucuğunun çıktısı ile hücrelerin sınırının belirlenebilmesi için Boundry (B) bağlacına, Rectangle kutucuğunun aynı adlı çıktısını bağlıyoruz. Bu komut, bize oluşturduğumuz alan içerisinde parametrelerini istediğimiz zaman değiştirebileceğimiz, ama aynı zamanda programın yapay zekası tarafından rastgele yerleştirilen noktaları temel alan hücreleri oluşturuyor.



Şekil 6. Offset komutu uygulaması (Telif hakkı yazara aittir)

Dördüncü adımda (Bkz. Şekil 6) oluşturduğumuz Voronoi hücrelerin sınır çizgilerini Offset komutu ile her iki yana öteleyerek çoğaltıyoruz. Bunun için Voronoi hücre komutunun çıktı kısmında yer alan Cell (C) bağlacını, Offset komutu girdi kısmında yer alan Curve (C) bağlacı ile ilişkilendirerek Distance (D) bağlacına değer giriyoruz.



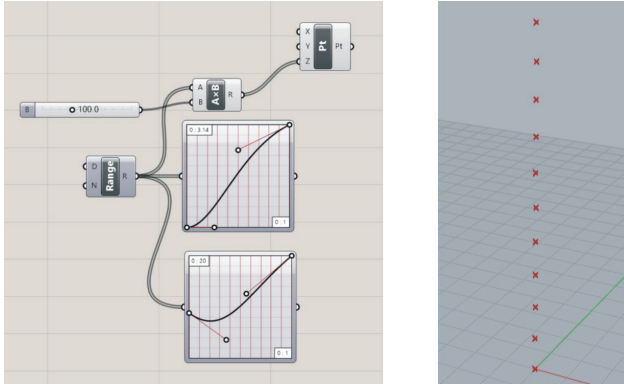
Şekil 7. Fillet komutu uygulaması (Telif hakkı yazara aittir)

Beşinci ve son adımda (Bkz. Şekil 7) oluşturduğumuz sınır çizgileri ötelenmiş Voronoi hücrelerin sınır çizgilerini fillet komutu ile yuvarlayarak yüzey desenimizi sonlandırıyoruz. Görüldüğü üzere önceden planlanan algoritmik sistem doğrultusunda adım adım oluşturulan komutların birbirleri ile ilişkilendirilmesi ve tasarımı belirleyen parametrelerin tanımlanması sonucunda elde edilen biçimin, istenen aşamasında parametrelerinin değiştirilebilmesi sayesinde varyasyonlarının oluşturulması mümkündür. Bu aşamadan sonra temel üç boyutlu modelleme ve modifikasyon yöntemleri ile tasarımımızı tamamlıyoruz (Bkz. Şekil 8).



Şekil 8. Elde edilen iki boyutlu yüzey deseninin üç boyut uygulaması (Telif hakkı yazara aittir)

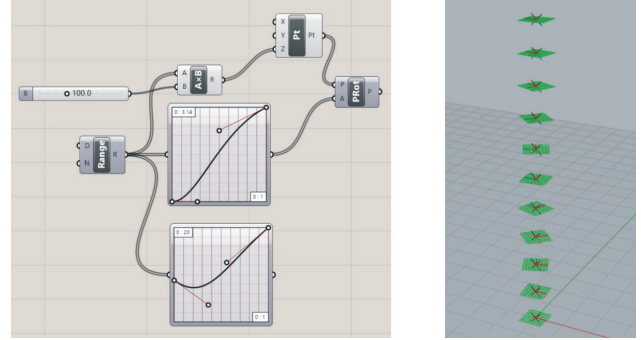
Çalışma kapsamında uygulanacak ikinci tasarım, başlangıcından son aşamasına kadar üç boyutlu olarak kurgulanmış bir vazo formu çalışmasıdır. Tasarım uygulamasında, tanımlanan kesitler arasında yüzey oluşturan "Loft" komutu temel alınmış, parametreler bu doğrultuda tanımlanarak tasarım algoritması oluşturulmuştur.



Şekil 9. Point ve Graph mapper komutlarının uygulaması (Telif hakkı yazara aittir)

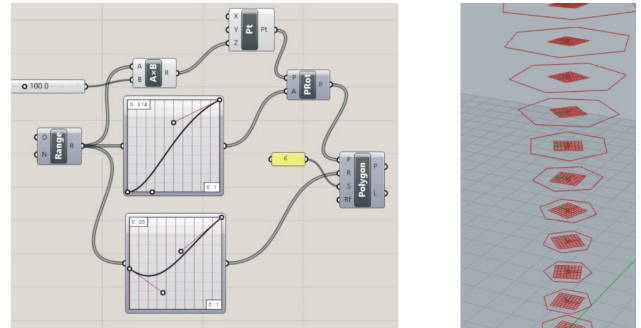
Birinci adımda (Bkz. Şekil 9) Vazonun temel biçimini oluşturacak kesit çizgilerinin yer alacağı düzlemlerin yerleşmesi için noktalarımızı oluşturuyoruz. Amacımız, oluşturacağımız düzlemleri döndürerek burgu formunda bir vazo oluşturmak olduğu için algoritmanın aşamalarını bu yönde kurguluyoruz. Bu aşamada vazonun yüksekliğini belirleyecek parametrenin tanımlanması ile başlıyoruz. Yükseklik için belirlenen değeri işleyebilmek için istediğimiz değeri, iki adet veriyi aynı anda çıktı olarak sunan çarpan

(AxB) komutunun (B) girdi sekmesi ile ilişkilendiriyoruz. Komutun diğer girdi sekmesine (A) ise oluşturacağımız düzlemlerin dönme açılarını ve kesitlerin oranlarını tanımlayabileceğimiz iki adet Graph mapper komutunu bağlıyoruz. Gördüğümüz üzere algoritmik komutlar, girdi ve çıktı sayıları ve özellikleri sayesinde farklı pek çok veriyi alıp kendi içinde ilişkilendirebilme yeteneğine sahiptir. Bu sayede "Range" komutunun çıktısını kullanarak iki adet Graph mapper komutunu AxB çarpan komutu ile ilişkilendirerek tek bir girdide, hem düzlemlerin dönme açılarını hem de düzlemlere daha sonraki aşamalarda yerleştireceğimiz kesitlerin boyutlarını belirleyebilmek için bir ilişki oluşturuyoruz. Graph mapper komutu yapısı gereği işlediği verileri kullanıcıya bir grafik eğri ile gösteren bir komuttur. Kullanıcı grafik eğriyi üzerinde yer alan kulakçıklar sayesinde değiştirerek sonucu manipüle edebilmektedir. Bu aşamada her ne kadar sonucu göremesek dahi, oluşturduğumuz ve birbirleri ile ilişkilendirdiğimiz komutlar, düzlemlerin yer alacağı noktalara fiziksel özelliklerini tanımlamış durumdadır. Elde ettiğimiz veriyi Point (Pt) komutunun "z" düzlem bilgisi girdisi ile ilişkilendirerek yer düzlemine dik açı ile oluşan istenilen uzunluğa eşit aralıklı olarak yerleşen noktalarımızı oluşturuyoruz.



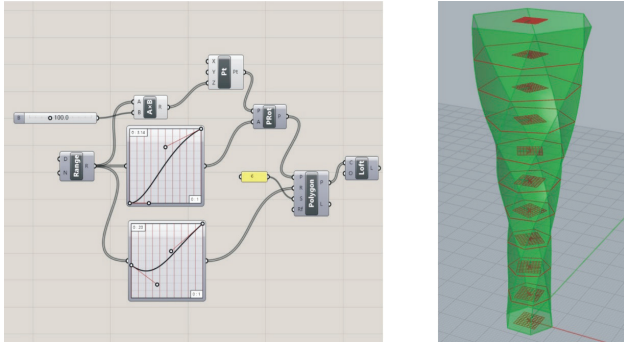
Şekil 10. Rotated Plane komutu uygulaması (Telif hakkı yazara aittir)

İkinci adımda (Bkz. Şekil 10), oluşturduğumuz noktalarda yer alacak düzlemlerimizi oluşturacağız. Tasarımın başında kurgumuzu yaparken düzlemlerin dönebilmesini hedeflediğimiz için PRot (Rotated Plane) komutunu kullanıyoruz. Komutun Plane (P) girdisine Point komutunun çıktısını, Angle (A) girdisine ise oluşturduğumuz Graph mapper komutunu bağlayarak ilişkilendiriyoruz.



Şekil 11. Polygon komutu uygulaması (Telif hakkı yazara aittir)

Üçüncü adımda (Bkz. Şekil 11) oluşturduğumuz düzlemlerde yer alacak kesitlerin biçimini tanımlıyoruz. Yaptığımız örnekte kesit, çokgen olarak tercih edilmiştir. Bunun için Polygon komutunu oluşturarak gerekli parametreler ile ilişkilendiriyoruz. Polygon komutunda yer alan ve kesitin yer alması gereken pozisyonu tanımlayan Plane (P) girdisi ile PRot (Rotated Plane) komutunun çıktısını ilişkilendiriyoruz. Bu sayede tanımlanan bütün düzlemlere birer adet çokgen yerleştiriyoruz. Komutun girdi kısmında yer alan Radius (R) girdisi ile ikinci Graph mapper komutunu bağlayarak her bir düzlemde yer alan çokgenlerin açılarının ölçülerini grafik olarak değiştirebiliyoruz. Komutun Segment (S) girdisine çokgenin kenar sayısı bilgisini girerek kesitimizi tamamlıyoruz. İsteğe bağlı olarak Radius Fillet (Rf) girdisi ile köşeleri yuvarlayarak biçimimizi yumuşatıyoruz.



Şekil 12. Loft komutu uygulaması (Telif hakkı yazara aittir)

Dördüncü ve son adımda (Bkz. Şekil 12) oluşturduğumuz farklı düzlemlerde yer alan kesitleri Loft komutu ile ilişkilendirerek üç boyutlu formumuzu tamamlıyoruz. Bunun için Polygon komutunun aynı isimli çıktısı ile (P) Loft komutunun Curves (C) girdisini birbirine bağlayarak ilişkilendiriyoruz. Aynı komut algoritması içinde yer alan, yükseklik, düzlem açıları ve çokgen kenar sayıları ile çaplarının parametreleri değiştirilerek uygulanan tasarım örnekleri Şekil 13'te yer almaktadır. Tasarım algoritması ve ilişkilerin doğru bir yapıda kurgulanmasından sonra tasarım alternatiflerinin oluşturulması son derece kolaylaşmaktadır.



Şekil 13. Parametre değişiklikleri ile elde edilen tasarım alternatifleri (Telif hakkı yazara aittir)

## Sonuç

Tasarımı tanımlayan geometrinin ardındaki hesaba dayalı matematiksel mantığın kavranması, özellikle ürün geliştirme sürecinde tasarımcıların en sık kullandığı düşünsel pratiklerden biridir. Bu bağlamda hesaplamalı tasarım teknikleri ile tasarlanan ürünün özellikle karmaşık biçimli kurgusunun ardındaki temel geometrinin algılanması ve biçimin tanımlanması, eğitim sürecinde son derece önemli bir yere sahiptir. Eğitim kurumları, gerek yazılım gerekse donanım bakımından gelişmeleri yakından takip etmeli ve eğitim programları içerisine hesaplamalı tasarım yaklaşımını sadece bilgisayar destekli tasarım eğitimi içerisinde değil, tasarlama sürecinin düşünsel yaratım aşamasında da dahil ederek, öğrencileri profesyonel hayata hazırlamalıdır.

## \*Yrd. Doç. Erkut ERYAYAR

E-posta: erkut.eryayar@marmara.edu.tr

Marmara Üniversitesi Güzel Sanatlar Fakültesi, Endüstri Ürünleri Tasarımı Bölümü

Küçükçamlica, 34718, Acıbadem, Kadıköy/İstanbul

## Kaynaklar

- *Algoritma*. (t.y.). Wikipedia içinde. Erişim adresi: <http://tr.wikipedia.org/wiki/Algoritma>, (Erişim tarihi: 25.06.2013)
- Akipek, F. (2007). Bilgisayar Destekli Tasarım ve Üretim Teknolojilerinin Mimarlıktaki Kullanımları. *Megaron YTÜ Mim. Fak. E-Dergisi*, Cilt 2, Sayı 4.
- Çinici, Ş.Y. (2012). "Computation" Çevirisi ve Anlaması Kolay Olmayan Dil, *Düşünce ve Mimarlık, Dosya Dergisi*, 29. Sayı TMMOB mimarlar odası Ankara şubesi, Kasım 2012, ss. 12-19.
- Denning, P. (2009). Beyond Computational Thinking. *Communications of the ACM*, Vol. 52 No. 6, June 2009, ss. 28-30.
- Erdoğan, E. ve Gönenç Sorguç, A. (2011). "Hesaplamalı Modeller Aracılığıyla Mimari ve Doğal Biçim Türetim İlkelerini İlişkilendirmek", *ODTÜ Mimarlık Fakültesi Dergisi*, 28, ss. 269-281
- *Parametre*. (t.y.). Türk Dil Kurumu güncel Türkçe sözlük içinde. Erişim adresi: <http://www.tdk.gov.tr> (Erişim tarihi: 25.06.2017)
- Uzun, A. (2001). Günümüz Ürün Geliştirme Sürecinde Bilgisayar Tabanlı Üçboyutlu Modelleme. *Yayımlanmamış Y. Lisans Tezi*, M.Ü.G.S.E., İstanbul.
- Yazar, T. ve Uysal, S. (2016). *Grasshopper ile Parametrik Modelleme*. İstanbul: Pusula 20 Teknoloji ve Yayıncılık
- <https://www.bbc.co.uk/education/guides/zp92mp3/revision/1> (Erişim tarihi: 27.12.2017)

The first part of the document discusses the importance of maintaining accurate records of all transactions. This includes not only sales and purchases but also any other financial activities that may occur. It is essential to ensure that all entries are properly documented and supported by appropriate evidence.

In addition, the document emphasizes the need for transparency and accountability in financial reporting. This means providing clear and concise information to all stakeholders, including management, investors, and regulatory authorities. It is important to disclose any potential risks or uncertainties that may affect the financial performance of the organization.

Furthermore, the document highlights the significance of internal controls and risk management. These measures are crucial for ensuring the integrity and reliability of financial data. By implementing robust internal controls, organizations can minimize the risk of errors and fraud, thereby enhancing the overall quality of their financial reporting.

Finally, the document stresses the importance of staying up-to-date with the latest accounting standards and regulations. This requires a commitment to continuous learning and professional development. By staying informed about changes in the regulatory environment, organizations can ensure that their financial reporting practices remain compliant and up-to-date.