

T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

İKİ BOYUTLU KALMAN FİLTRESİNİN
YERSEL LAZER TARAMA VERİSİNE
UYGULANMASI

Emre PINARCI

DANIŞMANI:

Yrd. Doç. Dr. M. Halis SAKA

YÜKSEK LİSANS TEZİ

JEODEZİ VE FOTOGRAMETRİ MÜHENDİSLİĞİ
ANABİLİM DALI

GEBZE

2007

 GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ	YÜKSEK LİSANS JÜRİ ONAY FORMU
--	---

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
 tarih ve sayılı kararıyla oluşturulan jüri tarafından
/...../..... tarihinde tez savunma sınavı yapılan
 'ın tez çalışması Anabilim Dalında Yüksek
 Lisans tezi olarak kabul edilmiştir.

JÜRİ

ÜYE
 (TEZ DANIŞMANI) :

ÜYE :

ÜYE :

ONAY

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
 tarih ve/..... sayılı kararı.

İMZA/MÜHÜR

ÖZET

TEZİN BAŞLIĞI: İKİ BOYUTLU KALMAN FİLTRESİNİN YERSEL LAZER TARAMA VERİSİNE UYGULANMASI

YAZAR ADI: Emre PINARCI

Bu çalışmada, günümüzde giderek yaygınlaşmaya başlayan yersel lazer tarayıcılardan elde edilen veride, taranan asıl cismin önünde bulunan diğer cisimlerden kaynaklanan boşlukların, iki boyutlu Kalman filtresi kullanarak doldurulması amaçlanmıştır.

Matlab ortamında kodlanan iki boyutlu Kalman filtresinin sonuçlarını daha iyi yorumlayabilmek için, gürültülü ve gürültüsüz yapay veri setleri üzerinde bir takım denemeler yapıldı. Ayrıca, veri setlerine bir de lineer enterpolasyon yöntemi uygulanarak, iki yöntemin sonuçları arasında karşılaştırma yapıldı. Burada, iki boyutlu Kalman filtresinin, gürültülü verideki boşluklar için, lineer enterpolasyon yönteminden daha iyi sonuçlar verdiği gözlenmiştir.

Gerçek uygulama olarak, Leica HDS3000 lazer tarayıcı ile taranmış olan Süleymaniye Camii kubbelerinden birisine ilişkin nokta bulutu verisindeki boşluklar, iki boyutlu Kalman filtresiyle dolduruldu.

Sonuç olarak, böyle bir uygulama için, Kalman filtresinin enterpolasyon yöntemlerine bir alternatif olarak kullanılabileceği anlaşılmıştır. Ayrıca, ölçme anında obje dışındaki anlık engellemelerin tespit edilmesinde ve ölçmelere yardımcı olması bakımından da kullanılabilir.

Anahtar Kelimeler: İki Boyutlu Kalman Filtresi, Yersel Lazer Tarayıcılar, Nokta Bulutundaki Boşlukların Doldurulması, Enterpolasyon

SUMMARY

TITLE: APPLYING TWO DIMENSIONAL KALMAN FILTERING TO TERRESTRIAL LAZER SCANNER DATA

AUTHOR: Emre PINARCI

In this research, it is aimed to fill the gaps on terrestrial laser scanner data, which are caused by other objects located in front of the scanned object during scanning, by using two dimensional Kalman filtering.

The filter, which is coded in Matlab, was applied to noisy and normal artificial data sets to understand the result more clearly. Linear interpolation method is also applied to the same data sets and the results of these two methods were compared. It is seen that the filter produces better results, especially when the data effected by noise, compared to linear interpolation method.

As a real application, one of the domes of Süleymaniye mosque which was scanned via Leica HDS3000, was used and the gaps on this dome surface was filled by 2-D Kalman filter.

As a result, it is seen that, 2-D Kalman filtering can be used as an alternative for linear interpolation method, for such surfaces. Furthermore, 2-D Kalman filter can be used to detect instantaneous blocks during scanning process.

Key Words: Two Dimensional Kalman Filtering, Terrestrial Laser Scanners, Filling the Gaps on Point Clouds, Interpolation

TEŞEKKÜRLER

Öncelikle, bana bir öğrenci gibi değil de bir meslektaş gibi davranan, usanmadan bana yol gösteren ve beni araştırmaya teşvik eden, GYTE öğretim üyesi, Danışman hocam Sayın Yrd. Doç. Dr. M. Halis SAKA'ya teşekkürü bir borç bilirim.

Nokta bulutu verisinin sağlanmasında Sayın Prof. Dr. İbrahim Baz hocama ve tüm BİMTAŞ A.Ş. yetkililerine; Şefim Sayın S. Özgür Oral'a; taramaları beraber yaptığımız çalışma arkadaşım Sayın H. İhsan Altundaş'a ve tüm çalışma arkadaşlarıma teşekkürler.

Hayat boyu benden maddi ve manevi hiçbir desteği esirgemeyen Annem Zeliha Pınarcı ve Babam Yusuf Pınarcı'ya ne kadar teşekkür etsem azdır. Tahsilimin bu aşamaya gelmesinde emekleri büyüktür.

Ve son olarak, tanıştığımız günden beri hayatı sonsuz sevgi ve hoşgörüyle benimle paylaşan, bana her konuda destek olan, çok sevdiğim eşim Meryem Pınarcı'ya çok teşekkürler. Hayat seninle daha kolay.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	iv
SUMMARY	v
TEŞEKKÜRLER	vi
İÇİNDEKİLER	vii
SİMGELER ve KISALTMALAR	viii
ŞEKİLLER	ix
TABLolar	xii
1. GİRİŞ	1
2. YERSEL LAZER TARAYICILAR	4
2.1. Yersel Lazer Tarayıcıların Çalışma Prensibi	4
2.2. Yersel Lazer Tarayıcıların Sınıflandırılması	5
2.3. Uygulamada Kullanılan Lazer Tarayıcı	8
3. KALMAN FİLTRESİ	9
3.1. Kestirilen İşlem	9
3.2. Filtrenin Sayısal Kökenleri	10
3.3. Filtrenin Algoritması	12
4. İKİ BOYUTLU KALMAN FİLTRESİ	15
4.1. 2-B Kalman İşlemcisi	15
4.2. İki Adımlı Filtreleme Yöntemi	19
5. YAPAY VERİ ÜZERİNDE DENEMELER	20
6. UYGULAMA	49
7. SONUÇ ve ÖNERİLER	58
KAYNAKLAR	60
ÖZGEÇMİŞ	62
EKLER	63
EK-1. Leica HDS3000 Teknik Özellikleri	64
EK-2. Matlab Uygulama Kodları	65
EK-3. Matlab Küre Örnekleme Kodu	77
EK-4. Matlab Lineer Enterpolasyon Kodu	78
EK-5. Matlab Gürültü Ekleme Kodu	79

SİMGELER ve KISALTMALAR

SAM	: Sayısal Arazi Modeli
YLT	: Yersel Lazer Tarayıcı
$S^-(i, j)$	: Durum vektör kestiriminin tahmin matrisi
$P^-(i, j)$	: $S^-(i, j)$ ile ilişkilendirilmiş kovaryans matrisi
$K(i, j)$	: Kalman kazancı
$S^+(i, j)$	: Durum vektör kestiriminin güncellenmiş hali
$P^+(i, j)$	: $S^+(i, j)$ ile ilişkilendirilmiş kovaryans matrisi
Z_r	: Gerçek Z değeri
Z_k	: İki boyutlu Kalman filtresinin hesapladığı Z değeri
Z_i	: Lineer enterpolasyon yönteminin hesapladığı Z değeri
Z_n	: Gürültülü Z değeri
R	: Ölçünün kovaryansı
Q	: Modelin kovaryansı
Std sap	: Standart sapma

ŞEKİLLER

<u>Sekil</u>	<u>Sayfa</u>
2.1. Lazer tarayıcı ölçüleri	5
2.2. Uçuş zamanlı sistem	6
2.3. Faz farklı sistem	6
2.4. Leica HDS 3000	8
3.1. Kalman filtresi döngüsü	13
3.2. Tekrarlamalı Kalman filtresinin çalışma prensibi	14
5.1. Yapay veriden silinen 1 dilimlik düşey noktalar	21
5.2. Yapayveriden silinen 1 dilimlik düşey noktaların Z_r, Z_k, Z_i grafiği	22
5.3. Yapay veriden silinen 1 dilimlik düşey noktaların Z_r-Z_k, Z_r-Z_i grafiği	22
5.4. Yapay veriden silinen 3 dilimlik düşey noktalar	23
5.5. Yapay veriden silinen 3 dilimlik düşey noktaların Z_r, Z_k, Z_i grafiği	24
5.6. Yapay veriden silinen 3 dilimlik düşey noktaların Z_r-Z_k, Z_r-Z_i grafiği	24
5.7. Yapay veriden silinen 5 dilimlik düşey noktalar	25
5.8. Yapay veriden silinen 5 dilimlik düşey noktaların Z_r, Z_k, Z_i grafiği	26
5.9. Yapay veriden silinen 5 dilimlik düşey noktaların Z_r-Z_k, Z_r-Z_i grafiği	26
5.10. Yapay veriden silinen 1 dilimlik yatay noktalar	28
5.11. Yapay veriden silinen 1 dilimlik yatay noktaların Z_r, Z_k, Z_i grafiği	28
5.12. Yapay veriden silinen 1 dilimlik yatay noktaların $Z_r-Z_k,$ Z_r-Z_i grafiği	29
5.13. Yapay veriden silinen 3 dilimlik yatay noktalar	30
5.14. Yapay veriden silinen 3 dilimlik yatay noktaların Z_r, Z_k, Z_i grafiği	30

5.15. Yapay veriden silinen 3 dilimlik yatay noktaların Z_r - Z_k , Z_r - Z_i grafiği	31
5.16. Yapay veriden silinen 5 dilimlik yatay noktalar	32
5.17. Yapay veriden silinen 5 dilimlik yatay noktaların Z_r , Z_k , Z_i grafiği	32
5.18. Yapay veriden silinen 5 dilimlik yatay noktaların Z_r - Z_k , Z_r - Z_i grafiği	33
5.19. Yapay veriden silinen eğik noktalar	34
5.20. Yapay veriden silinen eğik noktaların Z_r , Z_k , Z_i grafiği	35
5.21. Yapay veriden silinen eğik noktaların Z_r - Z_k , Z_r - Z_i grafiği	35
5.22. Gürültülü yapay veriden silinen 1 dilimlik düşey noktaların Z_r , Z_k , Z_i , Z_n grafiği	37
5.23. Gürültülü yapay veriden silinen 1 dilimlik düşey noktaların Z_r - Z_k , Z_r - Z_i , Z_r - Z_n grafiği	37
5.24. Gürültülü yapay veriden silinen 3 dilimlik düşey noktaların Z_r , Z_k , Z_i , Z_n grafiği	38
5.25. Gürültülü yapay veriden silinen 3 dilimlik düşey noktaların Z_r - Z_k , Z_r - Z_i , Z_r - Z_n grafiği	39
5.26. Gürültülü yapay veriden silinen 5 dilimlik düşey noktaların Z_r , Z_k , Z_i , Z_n grafiği	40
5.27. Gürültülü yapay veriden silinen 5 dilimlik düşey noktaların Z_r - Z_k , Z_r - Z_i , Z_r - Z_n grafiği	40
5.28. Gürültülü yapay veriden silinen 1 dilimlik yatay noktaların Z_r , Z_k , Z_i , Z_n grafiği	42
5.29. Gürültülü yapay veriden silinen 1 dilimlik yatay noktaların Z_r - Z_k , Z_r - Z_i , Z_r - Z_n grafiği	42
5.30. Gürültülü yapay veriden silinen 3 dilimlik yatay noktaların Z_r , Z_k , Z_i , Z_n grafiği	43
5.31. Gürültülü yapay veriden silinen 3 dilimlik yatay noktaların Z_r - Z_k , Z_r - Z_i , Z_r - Z_n grafiği	44
5.32. Gürültülü yapay veriden silinen 5 dilimlik yatay noktaların Z_r , Z_k , Z_i , Z_n grafiği	45

5.33. Gürültülü yapay veriden silinen 5 dilimlik yatay noktaların Z_r-Z_k , Z_r-Z_i , Z_r-Z_n grafiği	45
5.34. Gürültülü yapay veriden silinen eğik noktaların Z_r , Z_k , Z_i , Z_n grafiği	47
5.35. Gürültülü yapay veriden silinen eğik noktaların Z_r-Z_k , Z_r-Z_i , Z_r-Z_n grafiği	47
6.1. Nokta Bulutu ascii dosya formatı (X,Y,Z) koordinatları	51
6.2. Kullanılan Kalman filtresinin işlem adımları	52
6.3. Orijinal veri üstten görünüm	53
6.4. Hesaplanan boşluklar üstten görünüm	54
6.5. Orijinal veri hesaplanan boşlukların çakıştırılmış görüntüsü (üstten görünüm)	54
6.6. Orijinal veri hesaplanan boşlukların çakıştırılmış görüntüsü (3-B görünüm)	55

TABLolar

<u>Tablo</u>	<u>Sayfa</u>
2.1. Yersel Lazer Tarayıcıların sınıflandırılması	7
5.1. Gürültüsüz verilerdeki düşey boşluklar için hesaplanan standart sapmalar	27
5.2. Gürültüsüz verilerdeki yatay boşluklar için hesaplanan standart sapmalar	33
5.3. Gürültüsüz veriler üzerinde yapılan çalışmaların özeti	36
5.4. Gürültülü verilerdeki düşey boşluklar için hesaplanan standart sapmalar	41
5.5. Gürültülü verilerdeki yatay boşluklar için hesaplanan standart sapmalar	46
5.6. Gürültülü veriler üzerinde yapılan çalışmaların özeti	48
6.1. Nokta matrisi	55
6.2. İlk adımda hesaplanan değerler	56
6.3. İkinci adımda hesaplanan değerler	56
6.4. İki adımın ortalaması	57
6.5. Gerçek ölçüler	57

1. GİRİŞ

Sürekli gelişen teknoloji sayesinde, yaşamımıza yeni araç ve yöntemler girmektedir. Bu süreç içerisinde, geleneksel araç ve yöntemleri bir kenara bırakıp, yeni teknolojiye adapte olmak yeni sorunları da beraberinde getirebilir. İnsanoğlu, her ne kadar kendini bildiği ve tanıdığı yerde, bildiği ve tanıdığı araçlarla ve insanlarla birlikte güvende hissetse de, içindeki merak dürtüsü nedeniyle, tarih boyunca sürekli yeni arayışlara itilmiştir.

Günümüzde, kendisine birçok yeni uygulama alanları kazanan, ölçme işlemine yeni bir boyut katan yersel lazer tarayıcılar, birçok araştırmanın konusu olmaktadır. Her ne kadar, lazer teknolojisi çok yeni bir teknoloji olmasa da, mühendislik ve mimari projelerde kullanım amaçlı olarak lazer tarayıcıların piyasaya çıkabilmeleri, gelişen bilgisayar donanımları ve yazılımları sayesinde olmuştur.

Lazer tarayıcıların sağladığı yüksek çözünürlüklü veri, yoğun bir veri seti olduğu için, bu verilerin depolanması ve işlenmesi, yüksek hesap gücü ve depolama alanına sahip bilgisayarları kullanmayı zorunlu kılmaktadır.

Lazer tarayıcılarla yapılan taramalar sırasında, tarayıcının önünden geçen insan ve taşıtlar ile taranan cismin önündeki diğer cisimler, asıl taranmak istenen yüzey üzerinde eksikliklere neden olmaktadır. Bu çalışmada boşlukların, benzer aynı amaçlı çalışmaların aksine, mevcut noktalardan herhangi bir üçgen model oluşturmadan, sadece noktaların kendilerini kullanarak, çok fazla işlem gücü gerektirmeyen iki boyutlu Kalman filtresi ile nasıl doldurulabileceği araştırılmıştır.

Kalman filtresi, dinamik bir sistemin durum vektörünü, eksik ve gürültülü ölçü setinden kestiren, tekrarlamalı bir filtredir. Filtre, adını, mucidi olan Rudolf E. Kalman'dan almaktadır. Kalman filtresi genel olarak, navigasyon sistemlerinde, radarlarda ve görüntü işlemede kullanılmaktadır. Filtre ayrıca, kontrol teorisi ve kontrol sistemlerinin önemli bir konusudur. Mesela, radar uygulamalarında amaç, bir hedefin izlenmesidir. Hedefe ait konum, hız ve ivme verileri, herhangi bir anlık zamanda, önemsenecek kadar büyük gürültü etkisiyle elde edilir. Kalman filtresi, hedefin konumunu, gürültünün etkisini kaldırarak en iyi şekilde kestirmeye çalışır.

Bu tezde, Wang (1998)'de tanıtılan iki boyutlu Kalman filtresi kullanılmıştır. Kullanılan bu iki boyutlu Kalman filtresinde, etki alanı koordinattır. Filtre veriye iki adımlı olarak uygulanmıştır. Wang 1998 yılında yayımladığı makalesinde, iki boyutlu Kalman filtresini sayısal arazi modellerine uygulamıştır. Sayısal arazi modelleri ve yersel lazer taramaları, karakteristik olarak aynı veri türleridir. Her ikisi de X,Y,Z kartezyen koordinatlarından oluşan nokta kümeleridir. Yersel lazer tarama verilerinde, nokta aralığı mm ve cm'ler mertebesinde olduğu için, lineer model kabulüne uygundur.

Tezde üzerinde çalışmak amacıyla, Bimtaş A.Ş.'nin, Tarihi Yarımada projesi kapsamındaki Süleymaniye Camii'nde yapılan iç kubbe taramaları kullanıldı. Taramalar sırasında avize kabloları ve avizelerin kubbe yüzeyi üzerinde oluşturduğu boşlukların, iki boyutlu Kalman filtresi ile doldurulması amaçlandı. Bu amaç doğrultusunda, Wang (1998)'deki algoritma esas alınarak, filtre Matlab ortamında kodlandı.

Matlab ortamında kodlanan iki boyutlu Kalman filtresi, gerçek veri üzerinde denenmeden önce, gerçek veri ile benzer özellikte yapay veri oluşturularak denendi. Bu aşamada amaç, yazılan kodda hata olup olmadığını izlemek, hata varsa düzeltmek ve veri üzerinden istenilen

noktaların silinmesiyle oluşan boşluklarda filtreyi çalıştırıp, kestirilen değerler ile silinen değerler arasında karşılaştırma yapabilmektedir.

Filtrenin, benzer yöntemler ile karşılaştırmasını yapabilmek ve karakteri hakkında daha iyi fikir sahibi olabilmek için, verilere ayrıca lineer enterpolasyon yöntemi uygulandı. Böylelikle, iki yöntemin sonuçları arasında karşılaştırma yapabilmek mümkün oldu.

Yersel lazer tarayıcılar ile yapılan taramalar, tarayıcı dahilinde bulunan ölçme sistemlerinden kaynaklanan bir takım gürültü etkisine maruz kalmaktadır. Bu tezde uygulanan iki boyutlu Kalman filtresinin karakteri hakkında daha doğru fikir sahibi olmak maksadıyla, yapay veriye normal dağılımlı gürültü eklenerek, her iki yöntemle tekrar işlendi ve sonuçlar karşılaştırıldı.

Tezin son aşamasında, Matlab ortamında kodlanan iki boyutlu Kalman filtresi, Süleymaniye Camii'nin iç kubbe taramalarından birine uygulandı. Elde edilen sonuçlar 6. bölümde sunuldu.

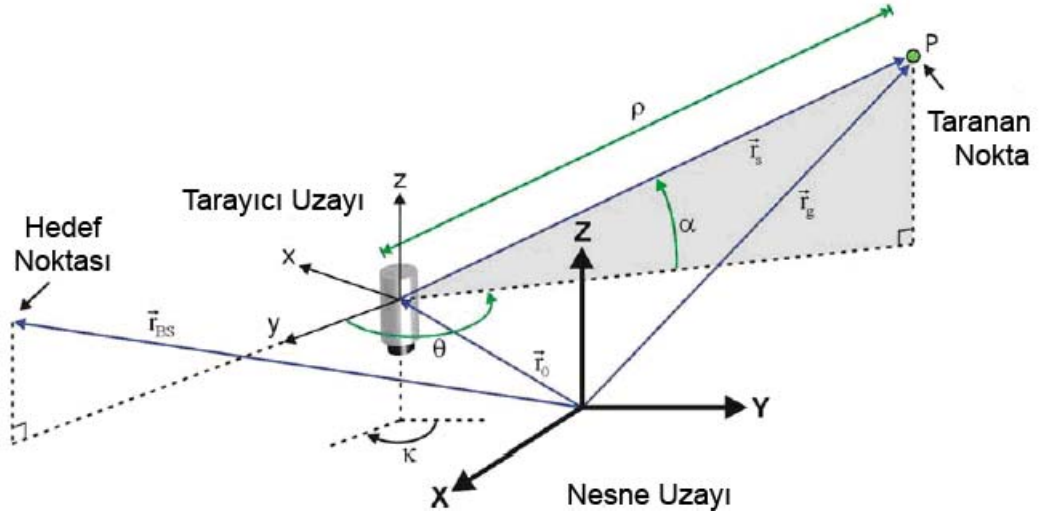
Yapay veriler üzerinde yapılan karşılaştırmalı denemeler sonucunda, gürültüsüz veriler için lineer enterpolasyon yönteminin daha doğru sonuç verdiği gözlemlenirken, gürültülü veriler için iki boyutlu Kalman filtresinin daha doğru sonuç verdiği gözlenmiştir. İki boyutlu Kalman filtresi, gürültünün neden olduğu olumsuz etkiden, lineer enterpolasyon yöntemine göre daha az etkilenmiştir. 2-B Kalman filtresi, yatay doğrultulu boşluklarda, düşeye nazaran daha doğru sonuç üretmiştir. Sonuç olarak, gürültülü kubbe yüzeyindeki boşlukların doldurulmasında, lineer enterpolasyon yöntemine alternatif olabilecek niteliktedir.

2. YERSEL LAZER TARAYICILAR

Yersel Lazer Tarayıcılar (YLT) artış eğilimi gösteren bir popüleriteye sahiptir. Günümüzde, endüstriyel uygulamalardan mühendislik uygulamalarına, tarihi eser dokümantasyonundan mimari uygulamalara kadar, benzer birçok uygulamalarda başarıyla kullanılmaktadırlar. Şüphesiz bu kadar geniş uygulama alanına girmiş olmasındaki en büyük etken, YLT'lerin ölçme aşamasında kazandırdığı avantajlardan olsa gerektir. Bunlardan en önemlisi, üç boyutlu, yüksek çözünürlüklü verileri, yüksek doğrulukta ve kısa zamanda elde etmemizi sağlamasıdır. Hâlihazırda piyasada farklı özelliklerde tarayıcılar üreten firmalar mevcuttur.

2.1 Yersel Lazer Tarayıcıların Çalışma Prensibi

Lazer tarayıcılar, yatay ve düşeyde belirli açısal aralıkta, bir dizi uzaklık ölçerek, bir görüntü sağlar. Bu görüntü aslında belirli aralıkta noktaların bir araya gelmesiyle oluşmakta ve bu yüzden “Nokta Bulutu” olarak adlandırılmaktadır. Tarayıcı bu noktaları dönen ayna veya dönen kafa veya bunların bir kombinasyonu ile lazer ışınını nesneye göndererek elde eder. Böylelikle tarayıcı, kendi dâhili koordinat sisteminde, yatay açı(θ), düşey açı (α) ve uzaklık (ρ) ölçerek, taranan noktanın (X,Y,Z) kartezyen koordinatını hesaplar (Lichti and Gordon, 2004). (X,Y,Z) kartezyen koordinatlarının dışında kayıtlı edilen bir diğer veri de nesneye çarpıp geri yansıyan enerjinin yoğunluğudur (intensity). Görüldüğü üzere, çalışma prensibi incelendiğinde, lazer tarayıcılar aslında birer aktif uzaktan algılama sistemleridir (Fangi et al, 2001).

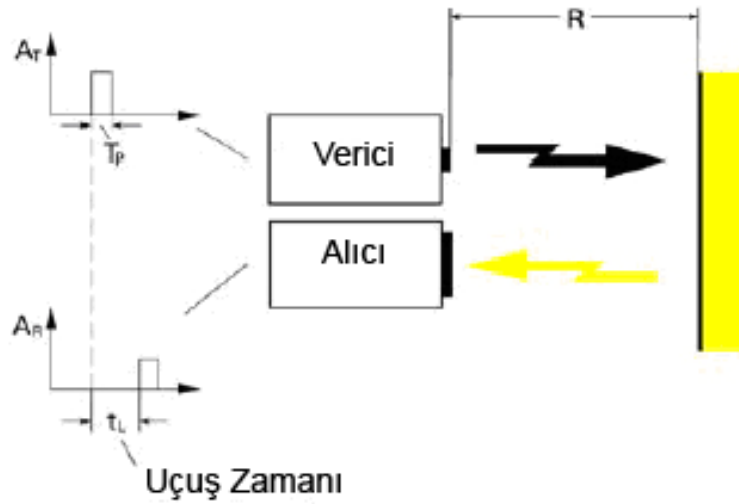


Şekil-2.1. Lazer tarayıcı ölçüleri(Lichti and Gordon, 2004)

2.2 Yersel Lazer Tarayıcıların Sınıflandırılması

Öncelikle, literatürde YLT'lerin sınıflandırılması için belirli bir yöntem bulunmamaktadır. Fakat belirli bir kriteri baz alarak sınıflandırma yapmak mümkündür. Zaten şuanda piyasada, her türlü projede kullanılabilecek bir tarayıcıdan söz etmek, pek mümkün değildir. Bazı tarayıcılar daha çok iç mekân kullanımına uygunken (menzili 100m'ye kadar olanlar), bazıları ise sundukları uzun menzil ile (1000m menzile kadar) dış mekân kullanımına daha uygundur. Bazı tarayıcılar da kısa mesafede yüksek doğruluğa sahip olup (10m'ye kadar mm. altında doğruluk), endüstriyel uygulamalarda kullanılmaktadırlar. Uzaklık ölçme prensiplerine bakıldığında YLT'leri üç gruba ayırmak mümkündür (Frölich and Metterleiter, 2004).

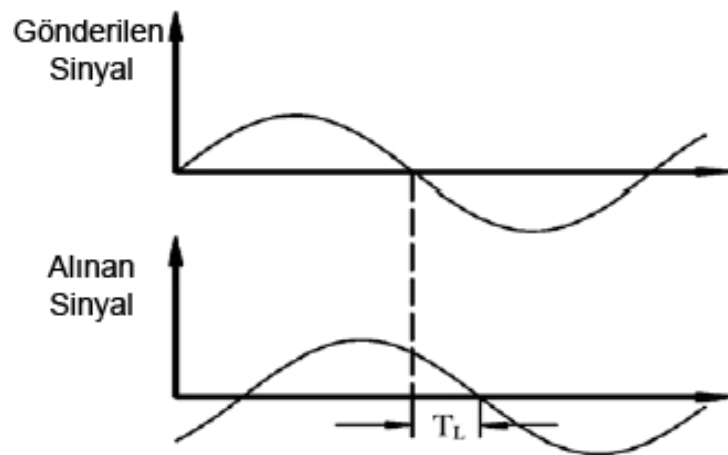
Günümüzde YLT'lerde kullanılan en yaygın mesafe, ölçme prensibi uçuş zamanıdır (time of flight) (Frölich and Metterleiter, 2004). Sistemin temel prensibi radarlara benzetilebilir. Sistem gönderilen sinyal ile alınan sinyal arasındaki zaman farkından mesafeyi hesaplamaktadır.



Şekil-2.2. Uçuş zamanlı sistem (Wehr, 2005)

Bu prensip üzerine kurulmuş sistemler ile uzun mesafe taramalarını (birkaç 100m) makul bir doğrulukla elde etmek mümkündür (Frölich and Metterleiter, 2004).

Uçuş zamanı prensibinin yanında, günümüzde kullanılan diğer yaygın sistemler faz farkı (phase difference) sistemlerdir. Sistem gönderilen sinyal ile geri alınan sinyal arasındaki faz farkından mesafeyi hesaplamaktadır.



Şekil-2.3. Faz farklı sistem (Wehr, 2005)

Bu sistemlerde menzil uzun olmamakla birlikte (100m) makul mesafelerde, yüksek doğruluk yakalamak mümkündür (birkaç mm) (Frölich and Metterleiter, 2004). Bu sistemlerin getirdiği en büyük avantaj sağladıkları yüksek tarama hızıdır.

Üçüncü prensip üzerine kurulu olan sistemler, optik üçgenlemeli (optical triangulation) sistemlerdir. Daha çok endüstriyel ve tersine mühendislik uygulamalarında kullanılırlar. Bu sistemler ile elde edilen doğruluk, mikrometre seviyesine kadar inmektedir (Frölich and Metterleiter, 2004).

Aşağıdaki tabloya bakıldığında, tarayıcılar uzunluk ölçme prensibine göre sınıflandırıldıklarında daha anlaşılabilir bir şekilde görülebilmektedir.

Tablo-2.1. YLT'lerin sınıflandırılması (Frölich and Metterleiter, 2004)

Ölçme Teknolojisi	Menzil (m)	Doğruluk (mm)	Üreticiler
Uçuş zamanlı	<100	<10	Callidus, Leica, Mensi, Optech, Riegl
	<1000	<20	Optech, Riegl
Faz farklı	<100	<10	IQSun, Leica, VisImage, Zoller+Fröhlich
Optik üçgenlemeli	<5	<1	Mensi, Minolta

2.3 Uygulamada Kullanılan Lazer Tarayıcı

Bu çalışmada kullanılan veri, Leica firmasının HDS 3000 model tarayıcısı ile elde edilmiştir. Tarayıcı uçuş zamanlı ölçme prensibine sahip olup, kullandığı iki pencere ile $360^{\circ} \times 270^{\circ}$ görüş alanına sahiptir. Bu görüş alanını, yatayda servo motor destekli kafa, düşeyde dönen ayna ile sağlamaktadır. Efektif optimum menzil 1m-100m olup, 3R sınıfı, görünür yeşil renkli lazer kullanmaktadır. Taranan noktanın konum doğruluğu 50m de 6mm dir. Tarama hızı 1800 nokta/saniye olup, maksimum tarama yoğunluğu 20000(nokta/satır) x 5000(nokta/sütun) dir. Cihaz dâhili 1mp kameraya sahip olup, $360^{\circ} \times 270^{\circ}$ görüş alanı için yaklaşık 64mp'lik bir görüntü sağlamaktadır. Tarayıcı tarafından sağlanan bu görüntü, taranacak alanın seçimi ve elde edilen nokta bulutunun renklendirilmesinde kullanılmaktadır. Tarayıcı, bir laptop veya PC'ye Ethernet arabirimi ile bağlanarak, Cyclone programı ile kumanda edilir. Tarayıcı üzerinde bellek bulunmayıp, taranan noktalar eşzamanlı olarak laptop veya PC'ye kayıt edilmektedir. Tarayıcı hakkındaki diğer bilgiler Ek-1 de sunulmuştur.



Şekil-2.4. Leica HDS 3000

3. KALMAN FİLTRESİ

Kalman filtresi adını Rudolph E. Kalman'dan almaktadır (Kalman, 1960). Kalman, 1960'da, ayırık verilerin lineer filtreleme probleminin çözümüne yönelik, tekrarlamalı çözümünü tanımladığı makalesini yayımlamıştır. Kalman filtresi, gürültülü ve eksik veri serisinden, dinamik bir sistemin durumunu kestiren, verimli bir tekrarlamalı filtredir (URL1).

Kalman filtresi aslında, farz edilen durumlar oluştuğunda, kestirilen hata kovaryanslarını minimize etmek için optimum tahmin edici-düzeltilici kestirimcilerin çalıştırıldığı matematiksel denklemler setidir. Tanıtıldığı günden bu yana, özellikle otonom ve destekli yön belirlemede, birçok araştırma ve uygulamanın konusu olmuştur. Filtreyi bu kadar geniş bir alanda uygulanabilir kılan özellikler, sayısal hesaplamadaki avantajları, rölatif basitliği ve sağlam doğasıdır. Filtrenin optimum sonuç vermesi için gerekli durumlar gerçekte tamamıyla var olmasa da, birçok uygulamada iyi sonuç vermektedir (Welch and Bishop, 2001).

3.1. Kestirilen İşlem

Kalman filtresi, lineer stokastik fark denklemleri tarafından idare edilen ayırık zaman kontrollü işlemin durumu $x \in \mathbb{R}^n$ i kestirmeye çalışır.

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \quad (3.1)$$

$z \in \mathbb{R}^m$ olan bir ölçü ile

$$z_k = Hx_k + v_k \quad (3.2)$$

Rasgele değişkenler w_k ve v_k , sırasıyla işlemin ve ölçünün gürültüsünü temsil ederler. Bunların birbirlerinden bağımsız ve normal dağılımlı olduğu kabul edilir.

$$p(w) \sim N(0, Q) \quad (3.3)$$

$$p(v) \sim N(0, R) \quad (3.4)$$

Uygulamada, işlem kovaryansı Q ve ölçü kovaryansı R matrisleri her zaman adımı ya da ölçüde değişebilir, fakat burada sabit kabul edilirler.

(3.1) deki $n \times n$ lik A matrisi, sürücü fonksiyonun ya da işlem gürültüsünün bulunmadığı durumlarda, $k-1$ zamanındaki önceki durumla, k zamanındaki şimdiki durum arasındaki bağlantıyı kurar. Not düşülmelidir ki, uygulamada A matrisi her işlem adımında değişebilir, fakat burada sabit kabul edilmiştir. $n \times l$ lik B matrisi, opsiyonel kontrol girdisi $u \in \mathcal{R}^l$ ile durum x arasındaki bağlantıyı sağlar. (3.2) deki ölçü denklemlerindeki $m \times n$ lik H matrisi, durum ile ölçü z_k arasındaki bağlantıyı sağlar. Uygulamada H , her zaman adımında ya da ölçüde değişebilir, fakat burada sabit kabul edilmiştir.

3.2. Filtrenin Sayısal Kökenleri

$\hat{x}_k^- \in \mathcal{R}^n$, k adımı öncesi verilen işlem bilgilerine göre, k adımındaki *a priori* durum kestirimi ve $\hat{x}_k \in \mathcal{R}^n$, verilen z_k ölçülerine göre, k adımındaki *a posteriori* durum kestirimi olarak tanımlanır. Sonrasında *a priori* ve *a posteriori* hataları şöyledir:

$$e_k^- \equiv x_k - \hat{x}_k^-$$

$$e_k \equiv x_k - \hat{x}_k$$

a priori kestirim hata kovaryansı:

$$P_k^- = E \left[e_k^- e_k^{-T} \right] \quad (3.5)$$

a posteriori kestirim hata kovaryansı:

$$P_k = E \left[e_k e_k^T \right] \quad (3.6)$$

Denklemleri Kalman filtresi için türetirken, asıl amaç *a posteriori* durum kestirimi $\hat{x}_k$ yı, *a priori* kestirim $\hat{x}_k^-$ ile gerçek ölçü z_k ve ölçü kestirimi $H\hat{x}_k^-$ arasındaki ağırlıklandırılmış farkın lineer kombinasyonundan hesaplayan bir denklem bulmaktır.

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) \quad (3.7)$$

(3.7) denklemindeki $(z_k - H\hat{x}_k^-)$ farkına ölçü düzeltmesi denir. Düzeltme, tahmin edilen ölçü $H\hat{x}_k^-$ ile gerçek ölçü z_k arasındaki uyuşumsuzluğu yansıtır. Sıfır düzeltmenin anlamı iki değer arasındaki mutlak uyumu gösterir.

(3.7) denklemindeki $n \times m$ lik K matrisi, *a posteriori* hata kovaryansı denklemleri (3.6) yı minimize eden, kazanç (gain) veya harmanlama faktörü (blending factor) olarak seçilir. Bu minimizasyon işleminin ayrıntıları (Maybeck, 1979) da görülebilir. (3.6)'yı minimize eden K matrisinin bir örneği:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (3.8)$$

(3.8) denkleminde bakıldığında, ölçü hata kovaryansı R sıfıra yaklaştıkça, kazanç K düzeltmeleri daha fazla ağırlıklandırır. Belirli bir biçimde:

$$\lim_{R_k \rightarrow 0} K_k = H^{-1}$$

Diğer taraftan, *a priori* kestirim hata kovaryansı P_k^- sıfıra yaklaştıkça, kazanç K düzeltmeleri daha az ağırlıklandırır. Belirli bir biçimde:

$$\lim_{P_k^- \rightarrow 0} K_k = 0$$

K nin yaptığı ağırlıklandırma işleminin açıklaması şudur: Ölçü hata kovaryansı R sıfıra yaklaştıkça, gerçek ölçü z_k daha da fazla güvenilirken, tahmin edilen ölçü $H\hat{x}_k^-$ daha da az güvenilirdir. Öte yandan, *a priori* kestirim hata kovaryansı P_k^- sıfıra yaklaştıkça gerçek ölçü z_k daha da az güvenilirken, tahmin edilen ölçü $H\hat{x}_k^-$ daha da fazla güvenilirdir.

3.3 Filtrenin Algoritması

Kalman filtresi, bir işlemi kestirirken, bir çeşit geri-besleme kontrol mekanizması kullanır. Filtre, bir zaman adımı için işlemin kestirim değerini hesaplar ve daha sonra gürültülü ölçüden geri-besleme alır. Bu yüzden Kalman filtresi denklemleri iki gruba ayrılır: *Zaman güncelleme* denklemleri ve *ölçü güncelleme* denklemleri. Zaman güncelleme denklemleri, bir sonraki zaman adımındaki *a priori* kestirimleri elde etmek için, geleceğin, şimdiki durumun ve hata kovaryans kestirimlerinin tasarlanmasından sorumludur. Ölçü güncelleme denklemleri geri-beslemeden sorumludurlar. Daha basit bir anlatımla, zaman güncelleme denklemleri geleceği tahmin eden denklemlerken, ölçü güncelleme denklemleri düzeltici denklemlerdir.

Doğrusu, son algoritma, sayısal problemleri çözmek için tahmin edici-düzeltici algoritmalara benzemektedir.



Şekil-3.1. Kalman filtresi döngüsü.

Kalman filtresi zaman güncelleme denklemleri:

$$\hat{x}_k^- = A\hat{x}_{k-1}Bu_k \quad (3.9)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (3.10)$$

Yukarıdaki denklemlerde, zaman güncelleme denklemlerinin durum ve kovaryans kestirimlerini $k-1$ adımından k adımına nasıl aktardığı görülmektedir (Welch and Bishop, 2001).

Kalman filtresi ölçü güncelleme denklemleri:

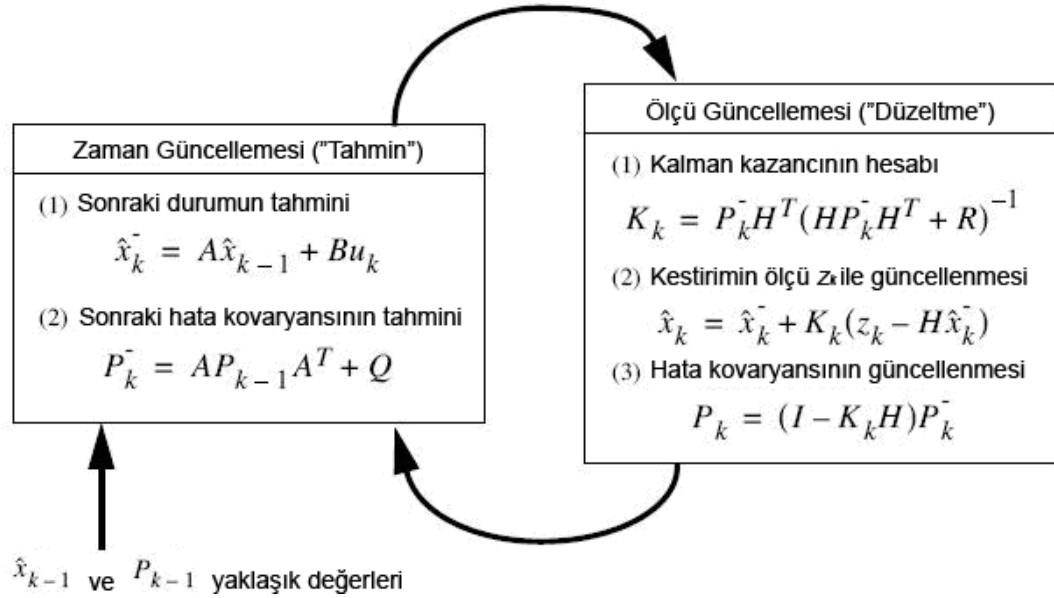
$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (3.11)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-) \quad (3.12)$$

$$P_k = (I - K_k H)P_k^- \quad (3.13)$$

Ölçü güncellemede hesaplanacak ilk değer Kalman kazancı K dir. Daha sonra gerçek ölçü z_k yı kullanarak *a posteriori* durum kestirimi hesaplanır. Son adım ise *a posteriori* hata kovaryansını hesaplamaktır.

Zaman ve ölçü güncellemeleri yapıldıktan sonra, işlem tekrarlı bir biçimde önceki *a posteriori* kestirimlerini kullanarak yeni *a priori* kestirimlerini hesaplar. Aşağıdaki şekil bu işlemleri resmetmektedir.



Şekil-3.2. Tekrarlamalı Kalman filtresinin çalışma prensibi (Welch and Bishop, 2001).

4. İKİ BOYUTLU KALMAN FİLTRESİ

Wang (1998)'in da atıfta bulunduğu üzere, 1970'lere kadar Kalman filtresinin iki boyuta genişletilmesi için ciddi çalışmalar yapılmamıştır. Ancak mevcut 2-B Kalman işlemcilerinin çoğu görüntü işleme için, özellikle görüntü

restorasyonu için geliştirilmiştir. Bu işlemciler, iki sebepten dolayı sayısal arazi modelleme uygulamalarına tamamen uygun değildir. İlk olarak, Kalman filtrelemesi model-tabanlı bir filtredir. Bu yüzden, eğer tekrarlamalı bir 2-B Kalman filtreleme algoritması görüntü restorasyonu için uygunsa, kurulmuş olan modelin, diğer 2-B uygulamalara uygun olacağı, mesela SAM için, garanti değildir. İkinci olarak, görüntü restorasyonu uygulamalarının aksine, gridli SAM kullanan arazi modellemesi kendine has özelliklere sahiptir. Bu nedenlerden dolayı mevcut 2-B Kalman filtreleme teknikleri benimsenmemiş, arazi modelleme uygulamaları için yeni bir 2-B Kalman filtreleme tekniği geliştirilmiştir. Wang (1998)'de Sayısal Arazi Modelleri için tanıtılan bu iki boyutlu Kalman filtresi bu tezin ana dayanağıdır. Çünkü SAM ve lazer tarama verileri benzer karakteristik özelliklere sahiptirler. Her iki veri seti de X,Y,Z kartezyen koordinatlarına sahip nokta kümeleridir.

4.1. 2-B Kalman İşlemcisi

Wang (1998)'de belirtildiği üzere, Kalman filtrelemesinin önemli bir kavramı 'durum' dur. Bir sistemin durumu, sistemin gelecek tepkisinin belirlenmesi için gerekli, geçmiş ve şimdi hakkındaki minimum bilgi olarak tarif edilebilir. Kesin rastlantısal bir durumda, gelecek gürültülü gözlemleri verilmiş, gelecek yanıtların en uygun, gelişigüzel kestirimlerinin belirlenmesi için gerekli, geçmiş ve şimdiki kestirimler hakkındaki minimum miktardaki bilgi olarak düşünülebilir. Durum kavramı aslında Kalman filtresini yönetmek için stokastik fark denklemlerini oluşturur.

$H(i, j)$ yükseklik, $H_x(i, j)$ X eksenı boyunca yüksekliđin birinci kısmi türevi ve $H_y(i, j)$ Y eksenı boyunca yüksekliđin birinci türevi, durum vektörünü oluşturmak için seçilirler. 'Yüksekliđin birinci kısmi türevi' terimini X ve Y eksenleri boyunca eğimi göstermek için kullanılır;

$$S(i, j) = \begin{bmatrix} H(i, j) \\ H_x(i, j) \\ H_y(i, j) \end{bmatrix} \quad (4.1)$$

İşlemcinin, raster taramada olduğu gibi, nokta bulutu verisini belirli bir sırada, örneğin sağdan sola, yukarıdan aşağıya, hesaplama yaptığını farz ederiz. Dolayısıyla, her işleme noktasında bazı nokta bulutu noktaları ‘geçmiş’ (işlenmiş noktalar) olarak, bazıları ‘gelecek’ (işlenmemiş noktalar) olarak ve o anda işlenmekte olan nokta ise ‘şimdi’ olarak düşünülür.

Arazi geometrisine göre, durum vektör noktası (i, j) ’nin bir dereceye kadar komşuları olan $(i-1, j)$ ve $(i, j-1)$ noktaları ile modellenebileceği varsayılır (Wang, 1998).

$$H(i, j) = b(i, j) \left[H(i-1, j) + H_x(i-1, j)dx \right] + c(i, j) \left[H(i, j-1) + H_y(i, j-1)dy \right] + v_H(i, j)$$

$$H_x(i, j) = H_x(i-1, j) + v_{H_x}(i, j)$$

$$H_y(i, j) = H_y(i, j-1) + v_{H_y}(i, j) \quad (4.2)$$

Burada v_H , v_{H_x} , v_{H_y} beyaz dizi, dx ve dy nokta bulutu’nun örnekleme aralığı, $b(i, j)$ ve $c(i, j)$ ise karışım faktörleridir.

(4.2) numaralı denklem tekrar yazılacak olursa, bu rastlantısal modelin dinamik modeli

$$S(i, j) = B(i, j) \cdot S(i-1, j) + C(i, j) \cdot S(i, j-1) + v_s(i, j) \quad (4.3)$$

olur.

$$B(i, j) = \begin{bmatrix} b(i, j) & b(i, j)dx & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$C(i, j) = \begin{bmatrix} c(i, j) & 0 & c(i, j)dy \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$v_s(i, j) = \begin{bmatrix} v_H(i, j) \\ v_{H_x}(i, j) \\ v_{H_y}(i, j) \end{bmatrix}$$

İse, $Q(i, j)$ bilinen kovaryansı ile dinamik modele ilişkin normal dağılımlı düzeltmelerdir.

Yükseklik gözlemesinin, ilgili pozisyonda durum vektörü ile lineer bir ilişkiye sahip olduğu varsayılır:

$$Z(i, j) = D \cdot S(i, j) + v_z(i, j) \quad (4.4)$$

Burada $D = [1 \ 0 \ 0]$, $v_z(i, j)$ ise bilinen $R(i, j)$ kovaryansı ile, $v_s(i, j)$ ile bağımsız olmak üzere, normal dağılımlı ölçülere ilişkin düzeltmelerdir.

(4.4) numaralı eşitlik, 2-B Kalman işlemcisinin fonksiyonel modeli olarak da bilinir.

Bu varsayımlar üzerinden, Wang (1998)'e göre, tekrarlı Kalman filtreleme işlemini denetlemek için aşağıdaki eşitlikler türetilir:

$$S^-(i, j) = B(i, j) \cdot S^+(i-1, j) + C(i, j) \cdot S^+(i, j-1) \quad (4.5)$$

$$P^-(i, j) = B(i, j) \cdot P^+(i-1, j) \cdot B(i, j)^T + C(i, j) \cdot P^+(i, j-1) \cdot C(i, j)^T + Q(i, j) \quad (4.6)$$

$$K(i, j) = P^-(i, j) D^T (D P^-(i, j) D^T + R(i, j))^{-1} \quad (4.7)$$

$$S^+(i, j) = S^-(i, j) + K(i, j)(Z(i, j) - D S^-(i, j)) \quad (4.8)$$

$$P^+(i, j) = [I - K(i, j) \cdot D] \cdot P^-(i, j) \quad (4.9)$$

Burada:

$S^-(i, j)$ durum vektör kestiriminin tahmin matrisi,

$P^-(i, j)$ $S^-(i, j)$ ile ilişkilendirilmiş kovaryans matrisi,

$K(i, j)$ Kalman kazancı,

$S^+(i, j)$ durum vektör kestiriminin güncellenmiş hali,

$P^+(i, j)$ $S^+(i, j)$ ile ilişkilendirilmiş kovaryans matrisidir.

$B(i, j)$ ve $C(i, j)$ matrisleri, $P^+(i-1, j)$ ve $P^+(i, j-1)$ 'in kapsadığı varyans ve kovaryans faktörlerine göre belirlenmelidir. Uygulamada, yaklaşık olarak 0.5 gibi sabit bir değer alınabilir ki bunun anlamı, X ve Y yönlerinden o anki işlenen noktaya olan katkı eşit ağırlıklıdır (Wang, 1998).

4.2. İki Adımlı Filtreleme Yöntemi

Kalman filtreleme teorisine göre, durum vektörünün kalitesi, nokta bulutundaki pozisyonuna göre farklılık gösterecektir. Bu yüzden, durum vektörünün kestirimi, işlenmiş noktalardaki tüm bilgilerin kullanılmasıyla en

uygun biçimde türetilir. Filtreleme yapılırken işlenmiş nokta sayısı artacak, dolayısıyla işleme kaydının son noktalarına yakın olan durum vektörü, başlangıç noktalarına yakın olan durum vektöründen genellikle daha iyi olacaktır.

Bu problemin üstesinden gelmek için, önerilen 2-B Kalman işlemcisini, aynı nokta bulutu verisinin üzerinden, farklı yöneltmeyle, iki defa uygulayan, iki adımlı filtreleme yöntemi tanıtılmıştır. İlk olarak, işlemci sağdan sola ve yukarıdan aşağıya doğru uygulanır. Sonrasında aynı işlem tam ters yönlerde, soldan sağa ve aşağıdan yukarıya doğru, uygulanır. (i, j) pozisyonlu herhangi bir nokta bulutu elemanı için, ilk işlem, nokta bulutunun yarı düzlemlili, $S_F^+(i, j)$ ile gösterilen, en uygun durum vektörü kestirimini ve onun kovaryans matrisi $P_F^+(i, j)$ 'yi oluşturur. İkinci işlem, o nokta için ilk işlemde kullanılmamış, nokta bulutunun diğer yarı düzlemini kullanan, diğer bir en uygun durum vektörü kestirimi $S_B^+(N-i, M-j)$ 'yi ve onun kovaryansı $P_B^+(N-i, M-j)$ 'yi üretir. Bu durumda, (i, j) noktasının durum vektörü kestirimi ve onun kovaryansı aşağıdaki gibi türetilir:

$$S_{final}(i, j) = \left[\left(P_B^+(N-i, M-j) \right)^{-1} + \left(P_F^+(i, j) \right)^{-1} \right]^{-1} \cdot \left[\left(P_F^+(i, j) \right)^{-1} S_F^+(i, j) + \left(P_B^+(N-i, M-j) \right)^{-1} S_B^+(N-i, M-j) \right]$$

$$P_{final}(i, j) = \left[\left(P_B^+(N-i, M-j) \right)^{-1} + \left(P_F^+(i, j) \right)^{-1} \right]^{-1} \quad (4.10)$$

Burada N, M nokta bulutunun satır ve sütun sayılarıdır (Wang, 1998).

5. YAPAY VERİ ÜZERİNDE DENEMELER

Ek-2'de, 4. bölümde anlatılan esaslara göre kodlanan filtreyi, gerçek veri üzerinde çalıştırmadan önce, filtrenin ürettiği sonuçları daha iyi anlamak ve karşılaştırma yapabilmek amacıyla yapay veriler üzerinde yapılan çalışma

sunulmuştur. Bu amaçla, Matlab'da, girilen küre parametrelerine ve X,Y düzlemindeki örnekleme aralığına göre veri üreten kod yazıldı (Ek-3). Sonrasında, bu veri üzerinden, yatay, düşey ve eğik olacak şekilde bazı noktalar silinerek filtre çalıştırıldı. Böylelikle, filtrenin ürettiği değerler ile orijinal değerler arasında karşılaştırma yapmak mümkün oldu.

Amaç, boşluk doldurmak olunca, bu tezde uygulanan yöntemi başka bir yöntemle karşılaştırmadan yorum yapmak doğru olmayacaktır. Bu yüzden aynı veriler bir de lineer enterpolasyon yöntemiyle işlenip, Kalman filtresi ve lineer enterpolasyon arasında da karşılaştırma yapılarak yorumlanmıştır. Bu amaç doğrultusunda, Ek-4'te sunulan kod yazıldı.

Tarayıcı ile yapılan tarama verileri, sistemdeki ölçme cihazlarının doğruluk kriterlerine göre bir takım gürültü içermektedir. Yapay verilerden alınacak sonuçların daha gerçekçi olması için, tüm verilere bir de normal dağılımlı gürültü eklendikten sonra Kalman filtresi ve lineer enterpolasyon işlemi uygulandı. Bu amaç için, Matlab'da Ek-5'teki kod yazıldı.

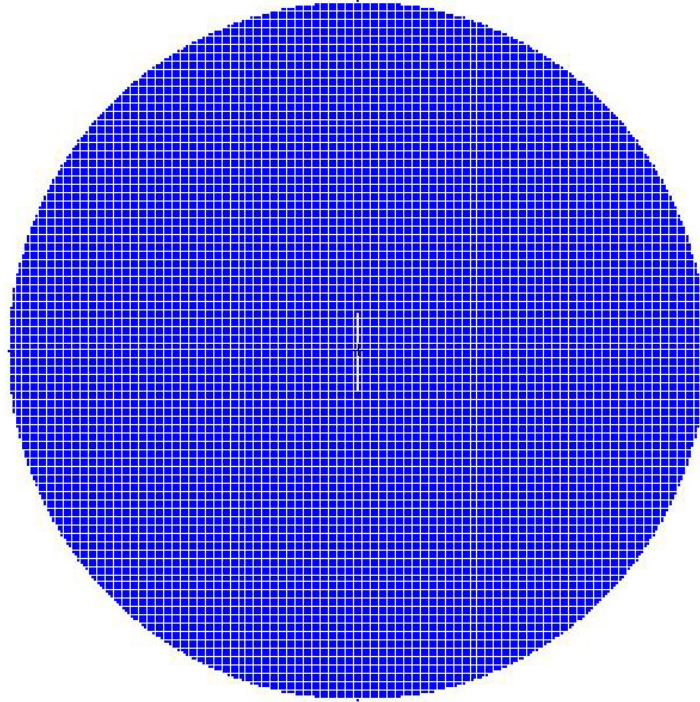
İlk olarak, Kalman filtresinin ve lineer enterpolasyon yönteminin gürültüsüz yapay veri üzerindeki sonuçları araştırıldı. Bu amaç doğrultusunda, Ek-3'teki kod kullanılarak, çapı 10m ve XY düzlemindeki yatay ve düşey nokta aralığı 0.04m olacak şekilde küre yüzeyi örneklendi. Sonrasında bu yüzey üzerinden, yatay, düşey ve eğik olacak şekilde nokta dilimleri silinerek, veriler Kalman filtresi ve lineer enterpolasyon yöntemlerine tabii tutuldu. İki yöntemden elde edilen sonuçlar birbirleri ile karşılaştırıldı. Bu adımda Kalman filtresi için:

$$R = \begin{bmatrix} 0.001^2 \end{bmatrix}$$

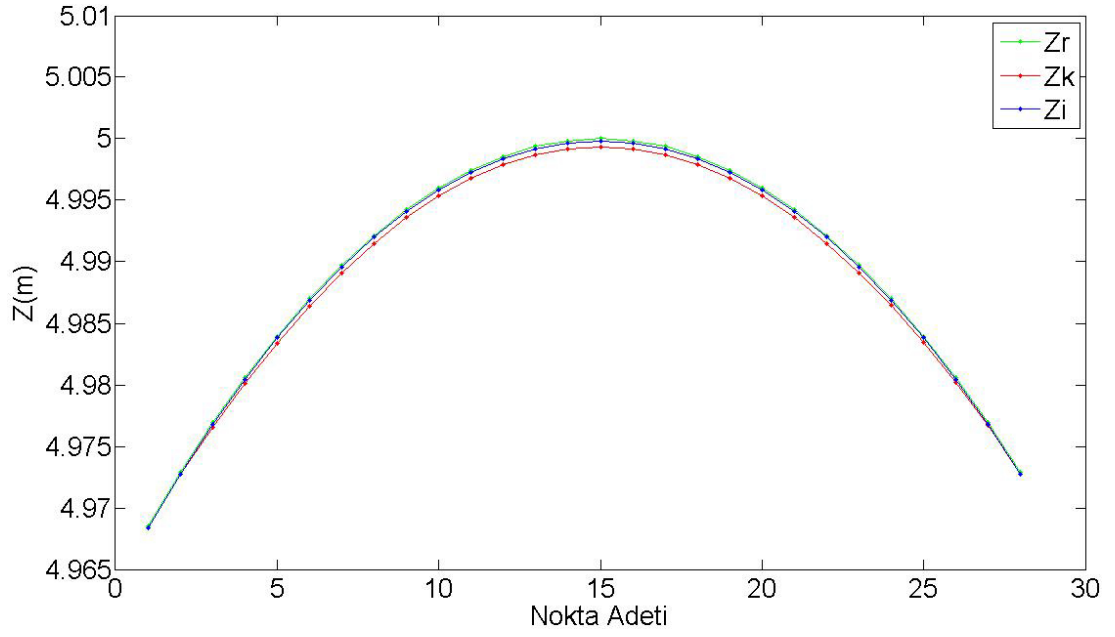
$$Q = \begin{bmatrix} 0.006^2 & 0 & 0 \\ 0 & 0.001^2 & 0 \\ 0 & 0 & 0.001^2 \end{bmatrix}$$

değerleri kullanıldı.

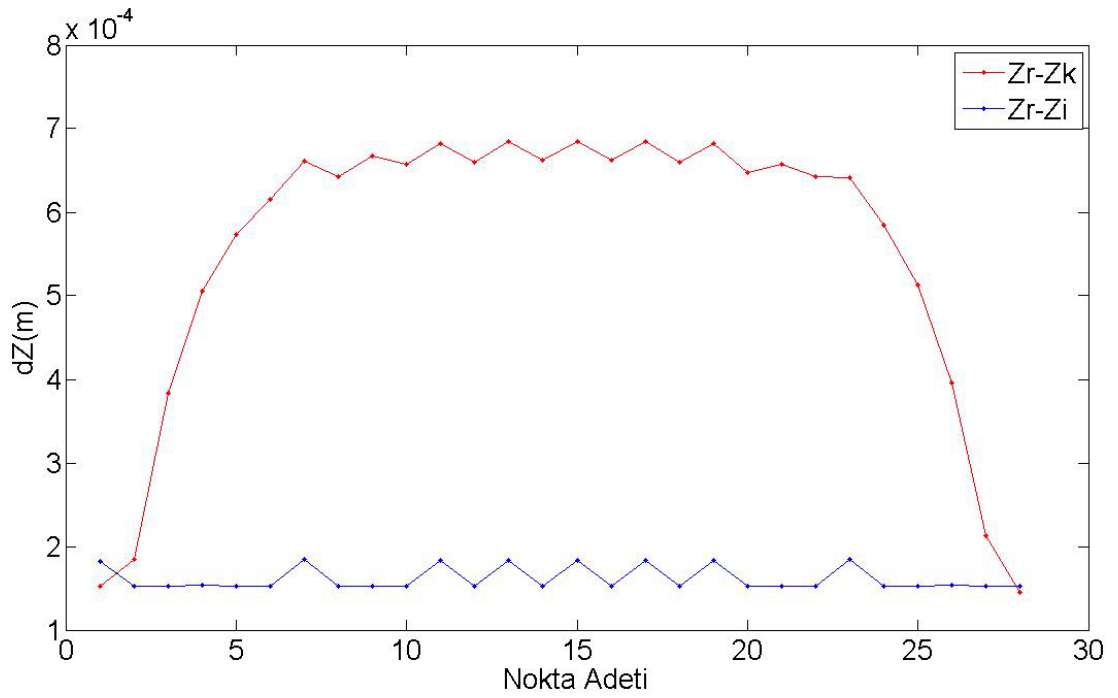
İlk olarak, örneklenen gürültüsüz yapay veride düşey 1 dilimlik boşluk oluşturuldu (Şekil-5.1). Sonrasında veriye iki boyutlu Kalman filtresi ve lineer enterpolasyon yöntemleri uygulandı.



Şekil-5.1. Yapay veriden silinen 1 dilimlik düşey noktalar



Şekil-5.2. Yapay veriden silinen 1 dilimlik düşey noktaların Z_r , Z_k , Z_i grafiği

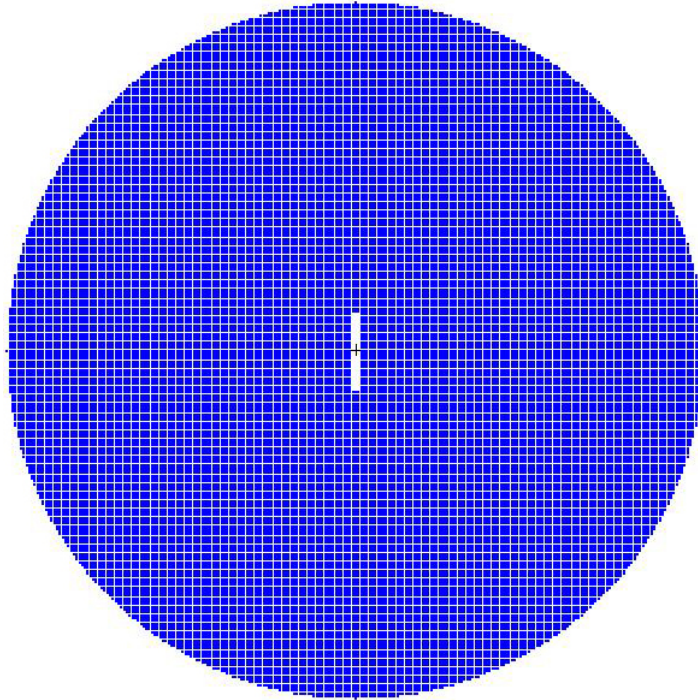


Şekil-5.3. Yapay veriden silinen 1 dilimlik düşey noktaların $Z_r - Z_k$, $Z_r - Z_i$ grafiği

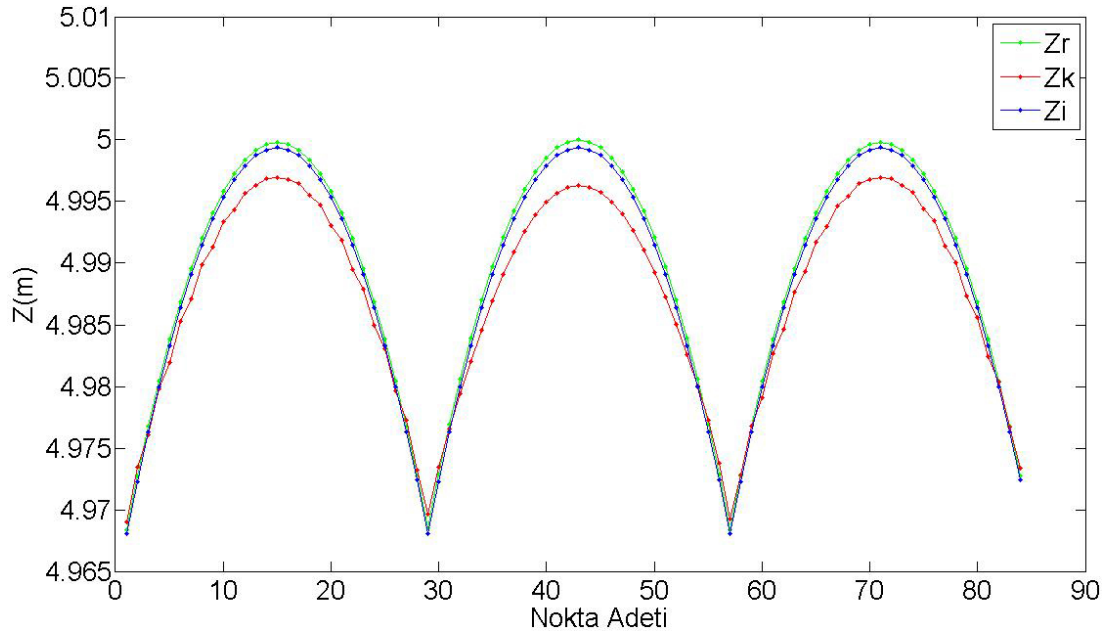
Yukarıda Şekil-5.2, gürültüsüz yapay veriden silinen noktaların yüksekliklerini (Z_r), iki boyutlu Kalman filtresinin kestirdiği yükseklik değerini

(Z_k) ve lineer enterpolasyon yönteminin ürettiği yükseklik değerini (Z_i) göstermektedir. Şekil-5.2 incelendiğinde, her iki yöntemin ürettiği Z değerlerinin, gerçek değerden ne kadar saptıkları görülebilmektedir. Bu sapmaların sayısal büyüklükleri ise Şekil-5.3'de görülebilir. Şekil-5.3, yapay veriden silinen 1 dilimlik düşey boşluk için, orijinal değer olan Z_r 'nin, iki boyutlu Kalman filtresi ve lineer enterpolasyon yönteminin ürettiği Z_k ve Z_i yükseklik değerlerinden olan farklarını (dZ) göstermektedir. 2-B Kalman filtresi için minimum ve maksimum farklar sırasıyla 0.145mm ve 0.685mm'dir. Lineer enterpolasyon yönteminin sonuçları incelendiğinde, minimum dZ 0.152mm, maksimum dZ 0.184mm'dir. Her nokta için, gerçek değerden, hesaplanan değerlerin farkı alınarak elde edilen dağılım incelendiğinde, iki boyutlu Kalman filtresinin standart sapması 0.178mm ve lineer enterpolasyon yönteminin standart sapması 0.014mm'dir. Bu aşamada lineer enterpolasyon yöntemi daha iyi sonuç vermiştir.

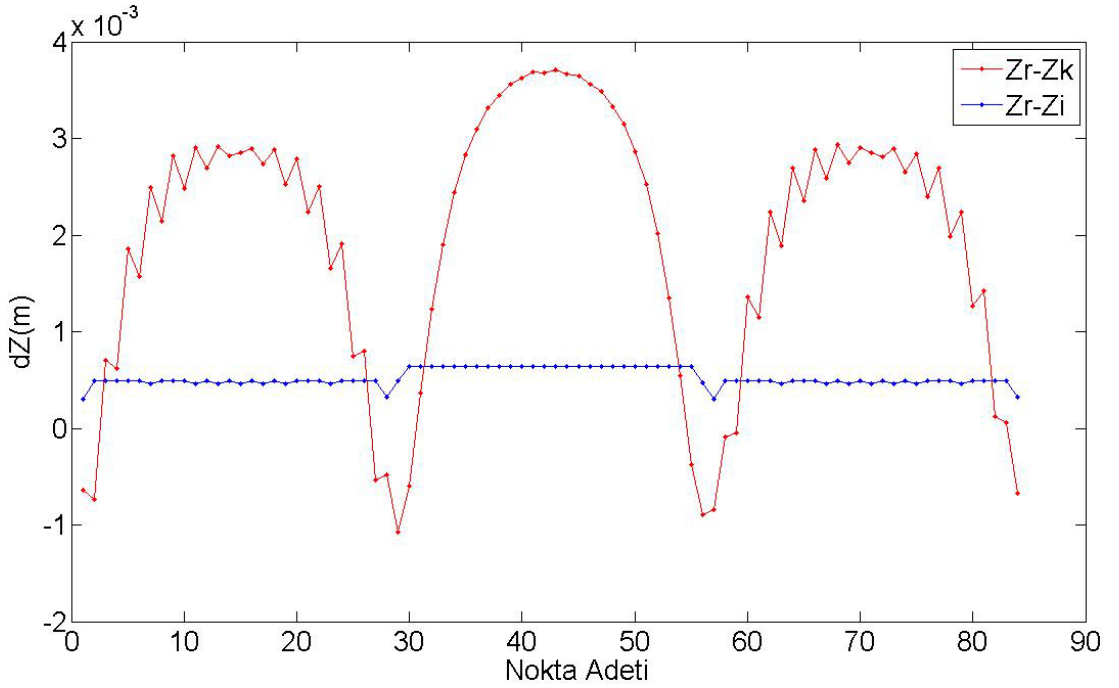
Örneklenen veriden silinen 3 dilimlik düşey boşluk için (Şekil-5.4) aynı işlemler tekrarlandığında, sonuçlar Şekil-5.5 ve Şekil-5.6'daki gibidir.



Şekil-5.4. Yapay veriden silinen 3 dilimlik düşey noktalar



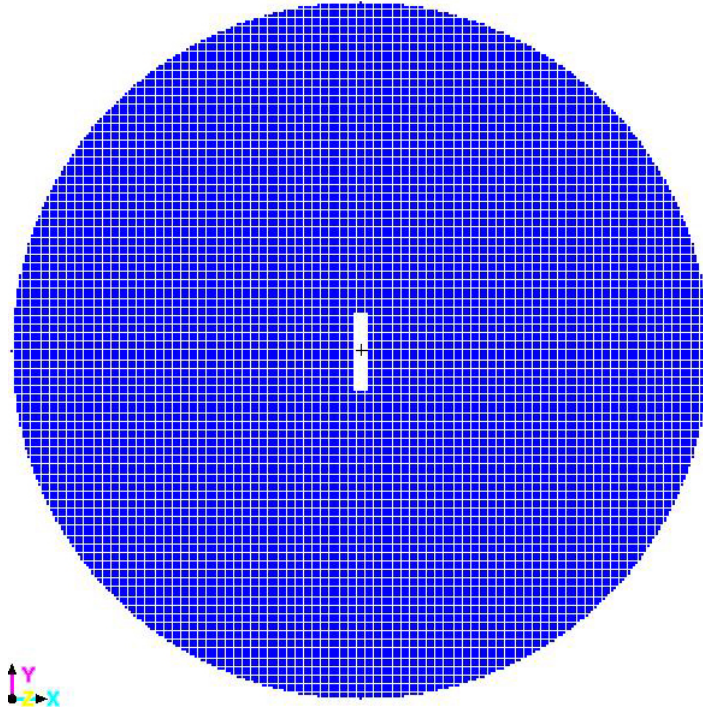
Şekil-5.5. Yapay veriden silinen 3 dilimlik düşey noktaların Z_r , Z_k , Z_i grafiği



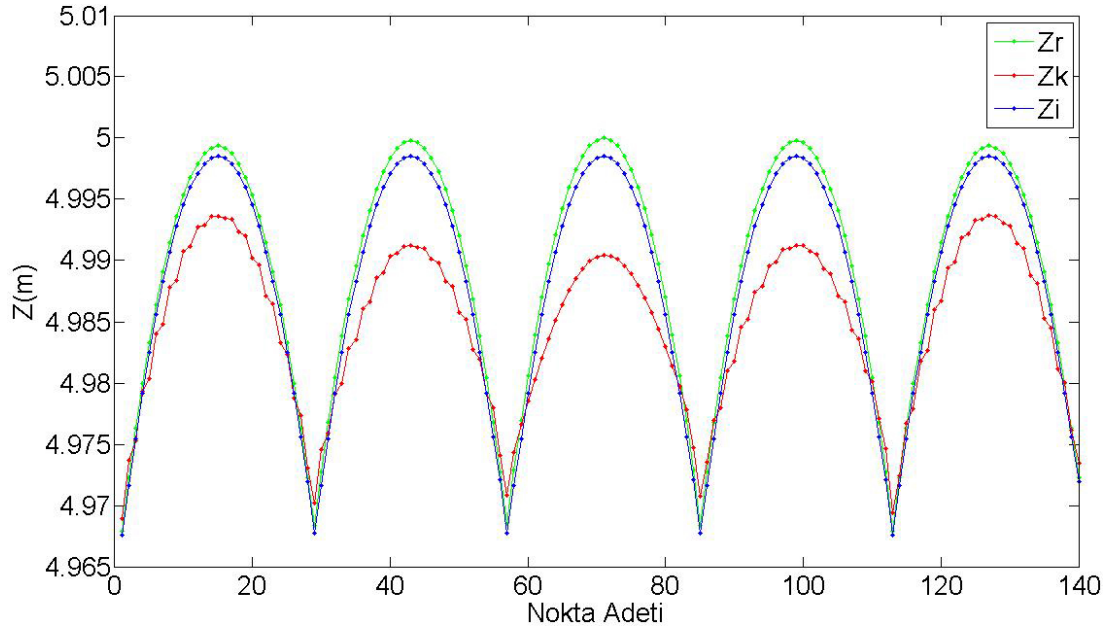
Şekil-5.6. Yapay veriden silinen 3 dilimlik düşey noktaların $Z_r - Z_k$, $Z_r - Z_i$ grafiği

Şekil-5.6 incelendiğinde, iki boyutlu Kalman filtresi için minimum ve maksimum dZ değeri ve standart sapma sırasıyla, -1.078mm, 3.705mm ve 1.356mm'dir. Lineer enterpolasyon yöntemi için aynı değerlere bakılacak olunursa sırasıyla, 0.306mm, 0.642mm ve 0.088mm'dir. Bu deneme, lineer enterpolasyon yöntemi daha iyi sonuç vermiştir.

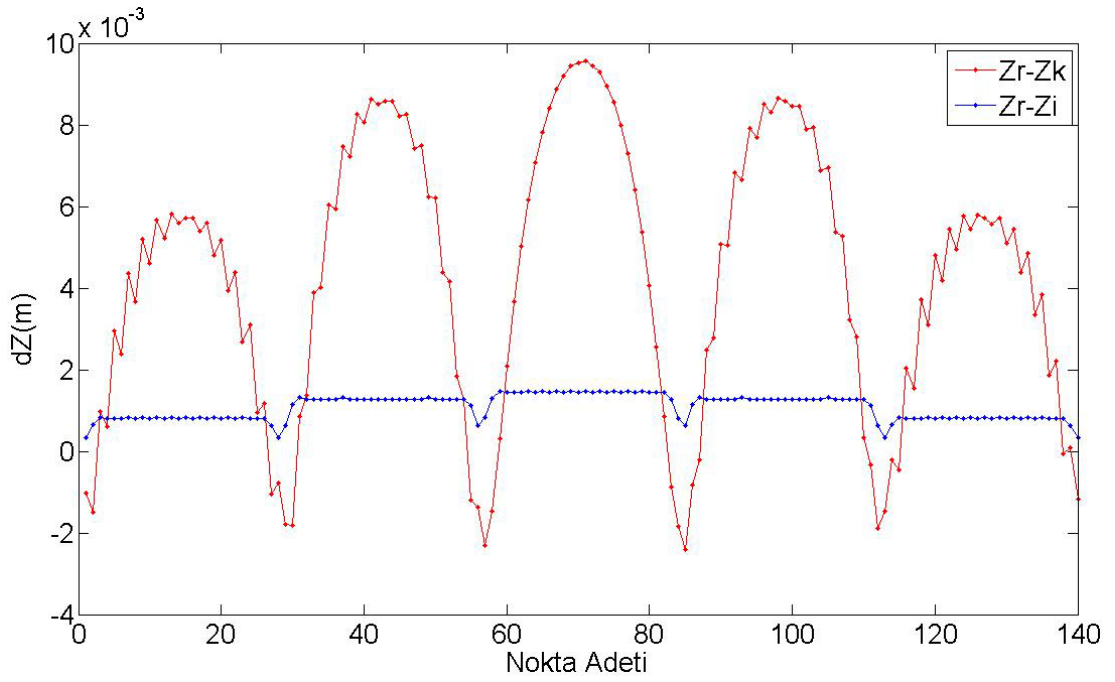
Örneklenen gürültüsüz yapay verideki düşey boşluklar için yapılan son denemede, veriden 5 dilimlik bir parça silindi (Şekil-5.7). Her iki yöntem uygulandıktan sonra elde edilen sonuçlar Şekil-5.8 ve Şekil-5.9'da görülebilir.



Şekil-5.7. Yapay veriden silinen 5 dilimlik düşey noktalar



Şekil-5.8. Yapay veriden silinen 5 dilimlik düşey noktaların Z_r , Z_k , Z_i grafiği



Şekil-5.9. Yapay veriden silinen 5 dilimlik düşey noktaların Z_r-Z_k , Z_r-Z_i grafiği

Şekil-5.9 incelendiğinde, iki boyutlu Kalman fitresi için, minimum dZ değeri -2.405mm , maksimum dZ 9.565mm ve standart sapma ise 3.345mm 'dir. Lineer enterpolasyon yöntemi için aynı değerler incelendiğinde,

minimum maksimum dZ ve standart sapma değerleri sırasıyla 0.335mm, 1.465mm ve 0.304mm'dir. Bu denemede de, lineer enterpolasyon yöntemi daha iyi sonuç vermiştir. Yukarıda gürültüsüz yapay veriden silinen düşey noktaların grafikleri bir tabloda özetlenecek olursa:

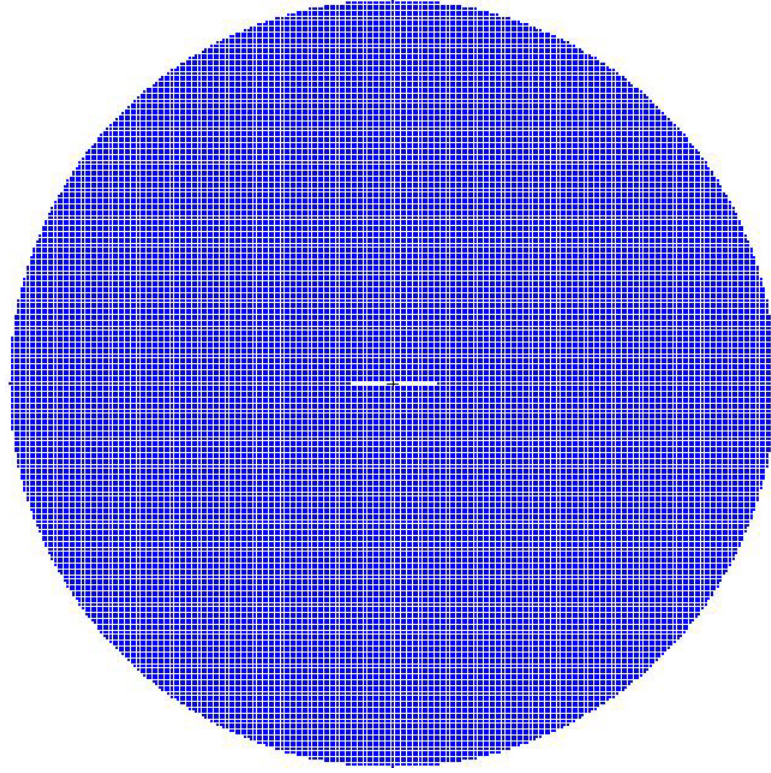
Tablo-5.1. Gürültüsüz verilerdeki düşey boşluklar için hesaplanan standart sapmalar

	2-B Kalman Filtresi			Lineer Enterpolasyon		
	Min dZ (mm)	Max dZ (mm)	Std sap (mm)	Min dZ (mm)	Max dZ (mm)	Std sap (mm)
Düşey 1 dilim	0.145	0.685	0.178	0.152	0.184	0.014
Düşey 3 dilim	-1.078	3.705	1.356	0.306	0.642	0.088
Düşey 5 dilim	-2.401	9.565	3.345	0.335	1.465	0.304

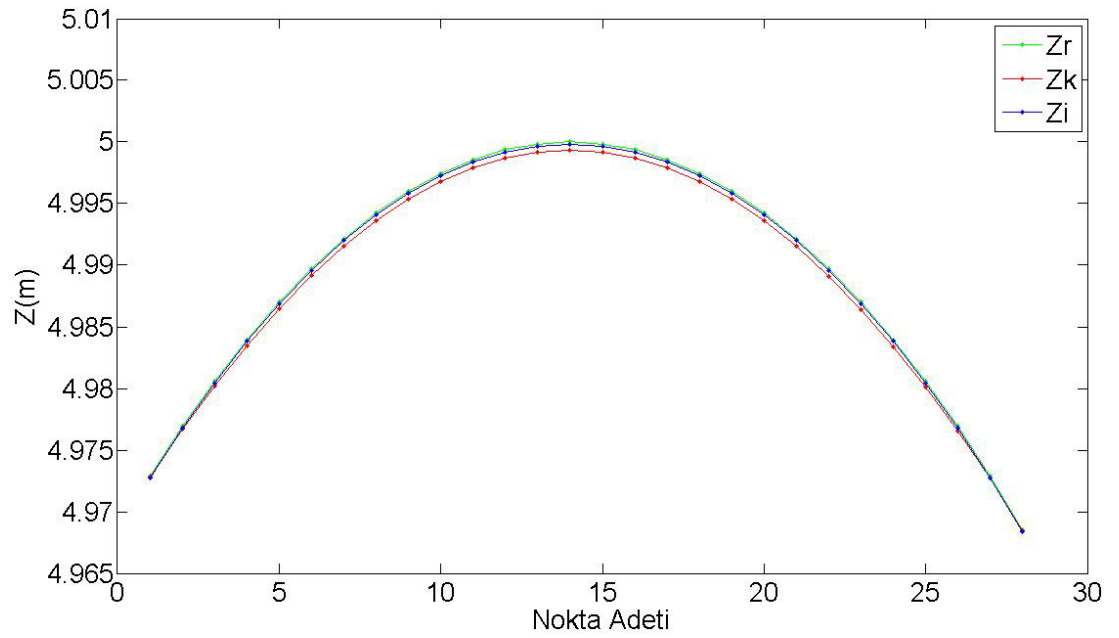
Tablo-5.1 göstermektedir ki, her iki yöntemde düşey yönlü boşluklar arttıkça, kestirilen değerlerin hatası da artmaktadır. Bu eğilim, 2-B Kalman filtresinin sonuçları için daha belirgindir. Elde edilen sonuçlara dayanılarak, lineer enterpolasyon yöntemi, bu tip verilerde daha doğru sonuç üretmektedir.

Benzer denemeler gürültüsüz veriden silinerek elde edilen yatay boşluklar için de tekrarlandı. Ek-3'teki kodun ürettiği veride yatay yönlü 1, 3 ve 5 dilimlik boşluklar oluşturuldu ve bu verilere her iki yöntem uygulandı.

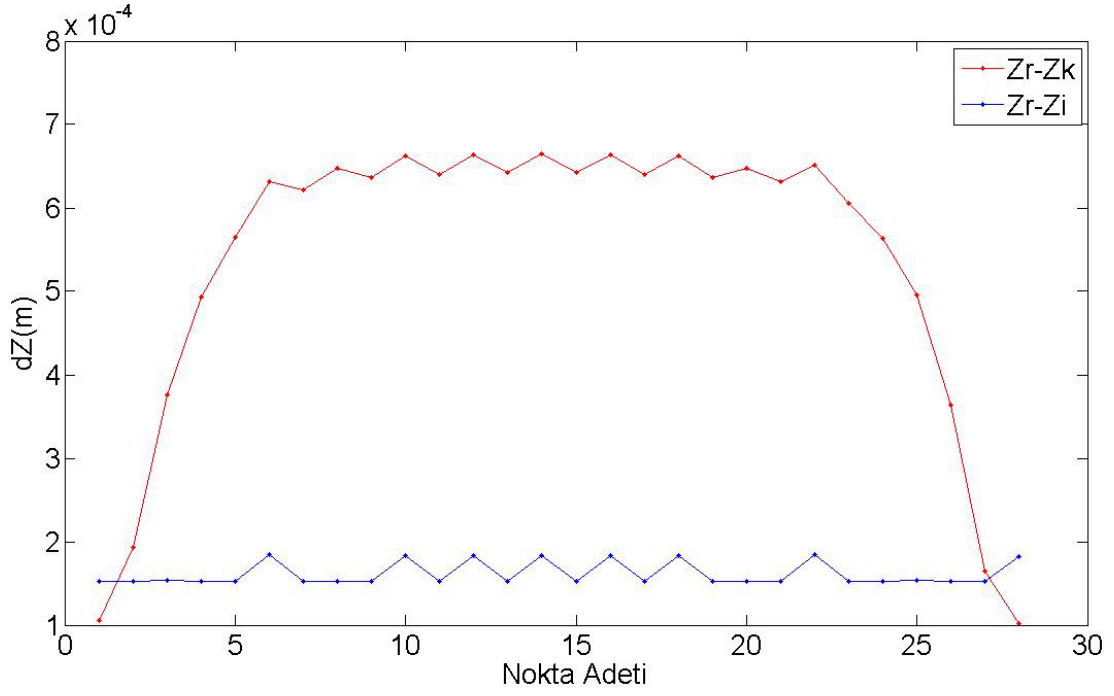
Denemelere, yatay yönlü 1 dilimlik boşluk ile başlandı. Şekil-5.10 silinen bu boşluk görünmektedir. Elde edilen sonuçlar Şekil-5.11 ve Şekil-5.12'de sunulmuştur.



Şekil-5.10. Yapay veriden silinen 1 dilimlik yatay noktalar



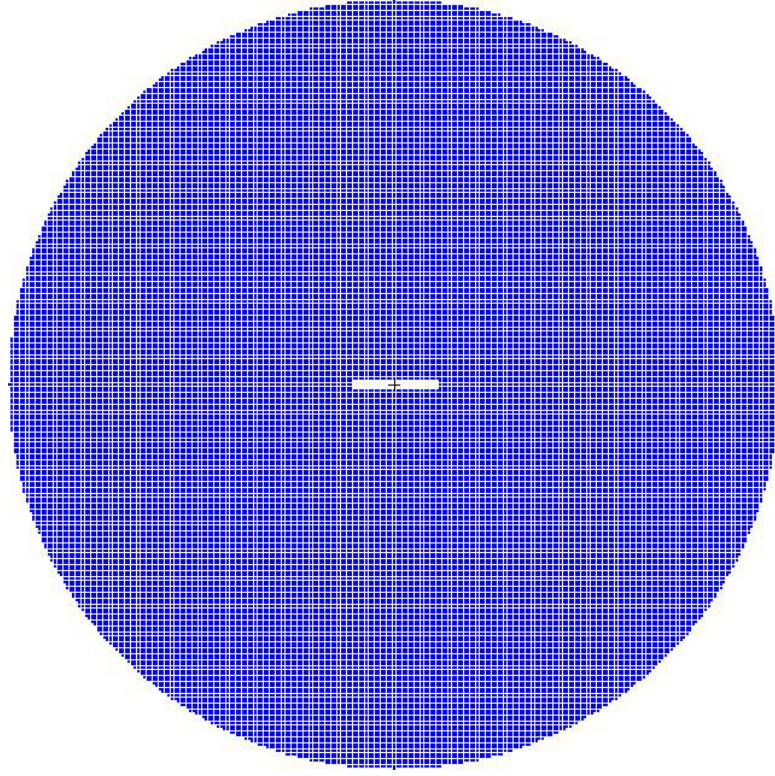
Şekil-5.11. Yapay veriden silinen 1 dilimlik yatay noktaların Z_r , Z_k , Z_i grafiği



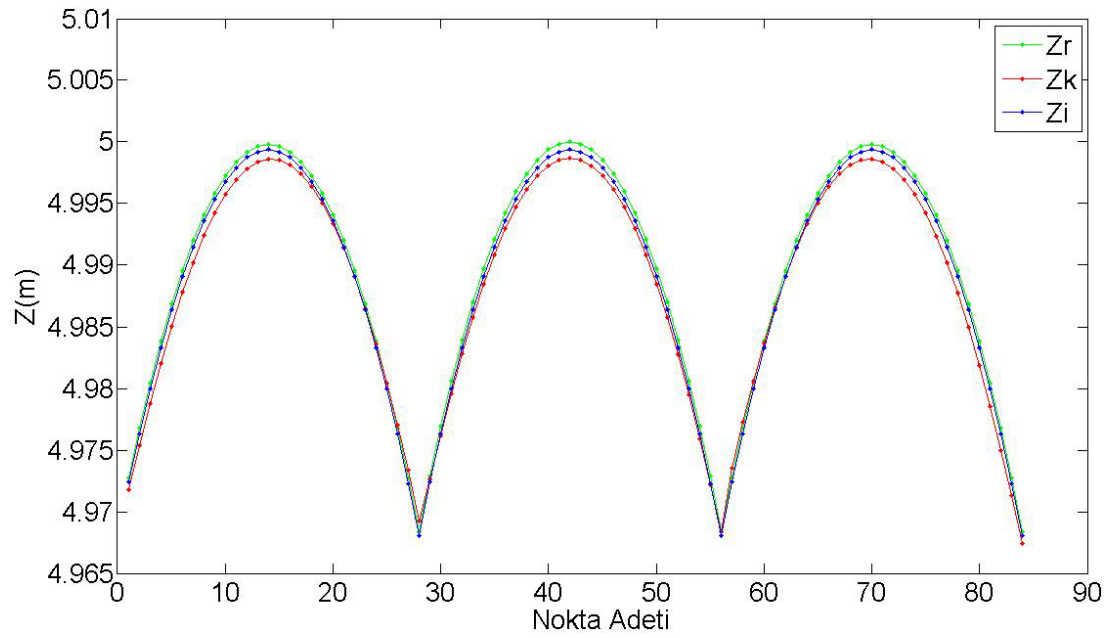
Şekil-5.12. Yapay veriden silinen 1 dilimlik yatay noktaların Z_r-Z_k , Z_r-Z_i grafiği

Yapay veriden silinen 1 dilimlik yatay boşluklar için iki yöntemin ürettiği sonuçlar incelendiğinde, Kalman filtresi için minimum, maksimum dZ değerleri ve standart sapma sırasıyla 0.102mm, 0.665mm ve 0.183mm'dir. Aynı değerler lineer enterpolasyon yöntemi için sırasıyla 0.152mm, 0.184mm ve 0.014mm'dir (Şekil-5.12). Sonuç olarak, bu aşama da lineer enterpolasyon yöntemi daha doğru sonuç üretmiştir.

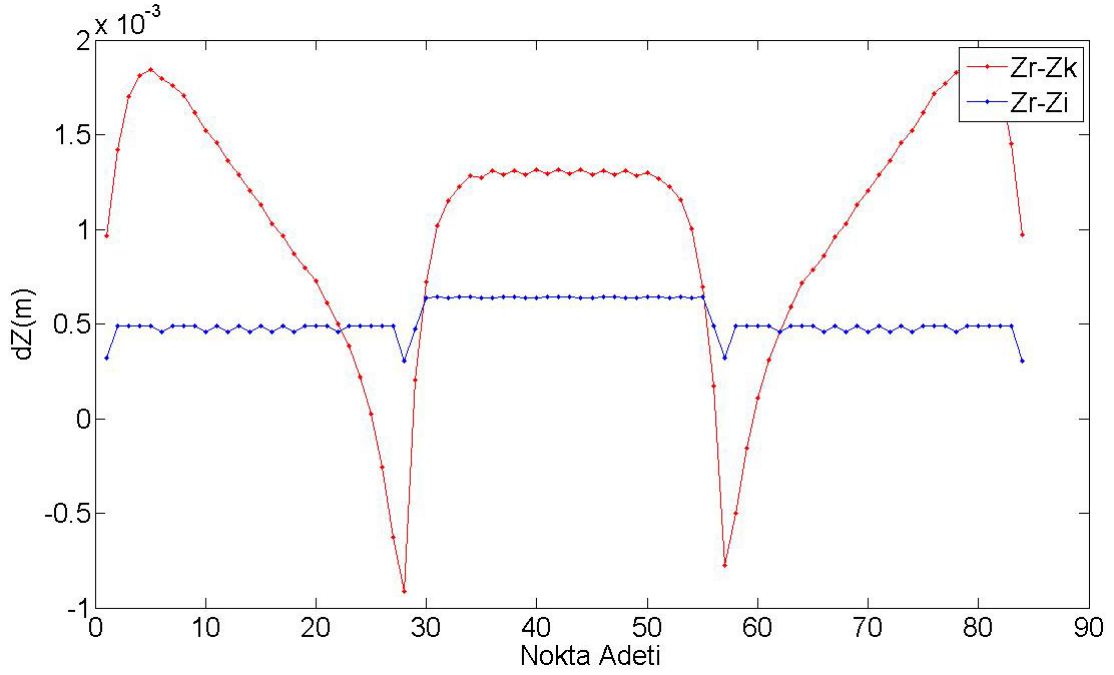
Sonrasında, gürültüsüz yapay veride düşey yönlü 3 dilimlik boşluk oluşturularak (Şekil-5.13), veriye her iki yöntem uygulandı. Sonuçlar Şekil-5.14 ve Şekil-5.15'de görülebilir.



Şekil-5.13. Yapay veriden silinen 3 dilimlik yatay noktalar



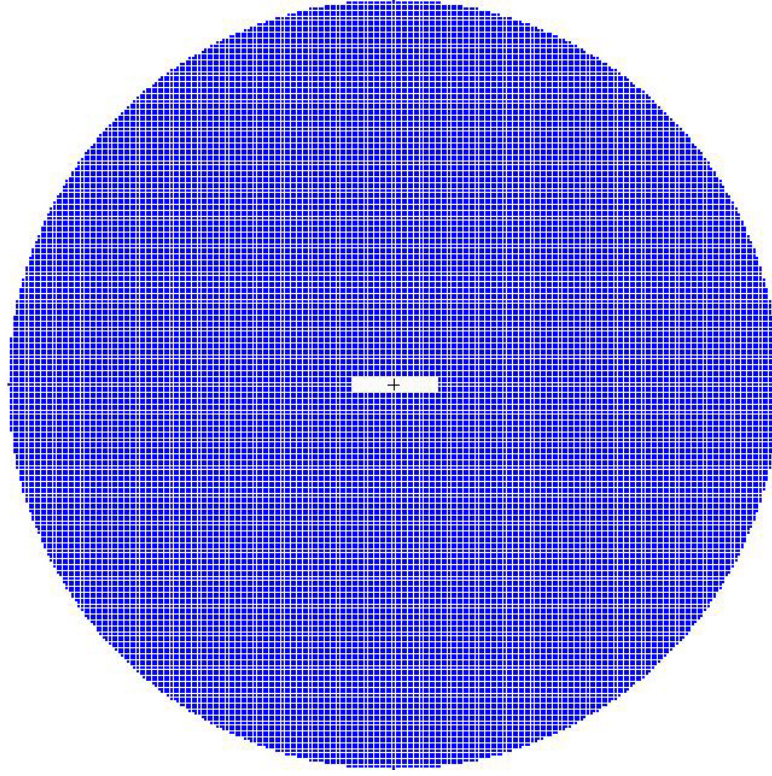
Şekil-5.14. Yapay veriden silinen 3 dilimlik yatay noktaların Z_r , Z_k , Z_i grafiği



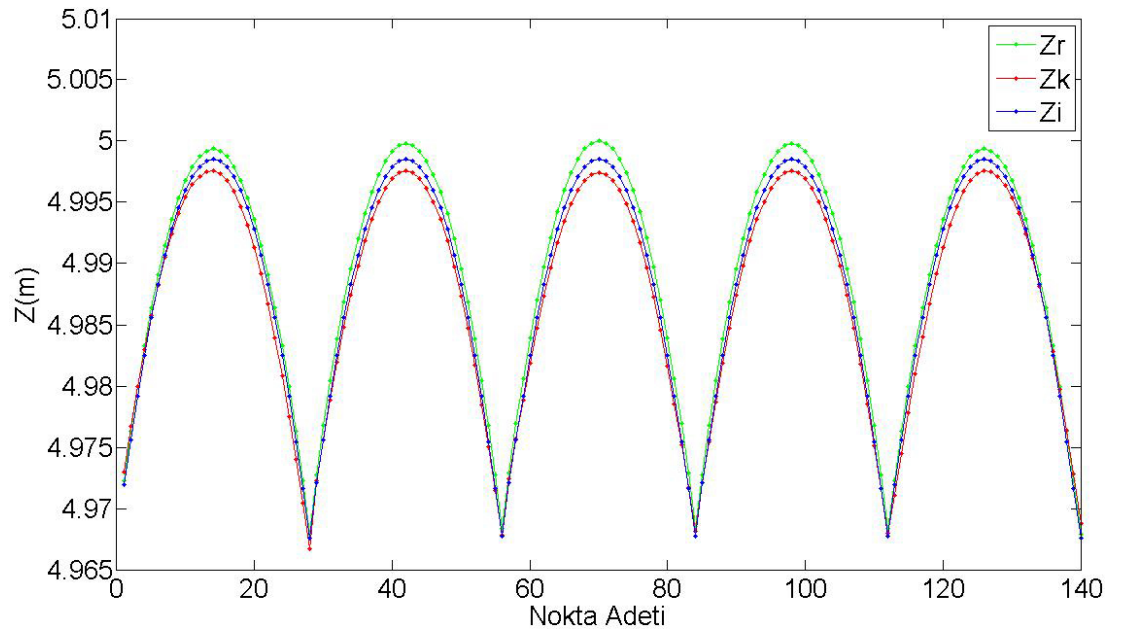
Şekil-5.15. Yapay veriden silinen 3 dilimlik yatay noktaların Z_r-Z_k , Z_r-Z_i grafiği

Yapay veriden silinen 3 dilimlik yatay noktaların oluşturduğu boşluğa her iki yöntem uygulandığında, iki boyutlu Kalman filtresi için minimum dZ değeri -0.911mm , maksimum dZ değeri 1.920mm ve standart sapma 0.631mm 'dir. Lineer enterpolasyon yönteminin verdiği sonuçlar ise, minimum dZ değeri 0.306mm , maksimum dZ değeri 0.642mm ve standart sapma 0.088mm 'dir. Bu denemede, lineer enterpolasyon yöntemi daha başarılıdır.

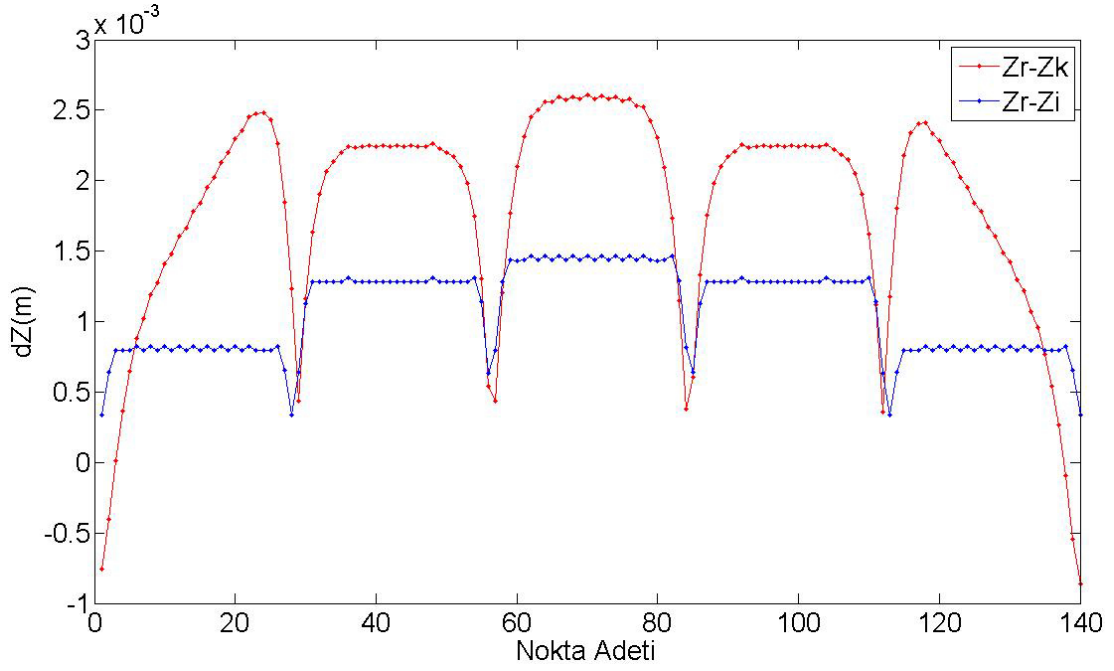
Benzer şekilde, gürültüsüz yapay veriden 5 dilimlik nokta kümesi silindi (Şekil-5.16). Elde edilen sonuçlar Şekil-5.17 ve Şekil-5.18'de sunulmuştur.



Şekil-5.16. Yapay veriden silinen 5 dilimlik yatay noktalar



Şekil-5.17. Yapay veriden silinen 5 dilimlik yatay noktaların Z_r , Z_k , Z_i grafiği



Şekil-5.18. Yapay veriden silinen 5 dilimlik yatay noktaların Z_r-Z_k , Z_r-Z_i grafiği

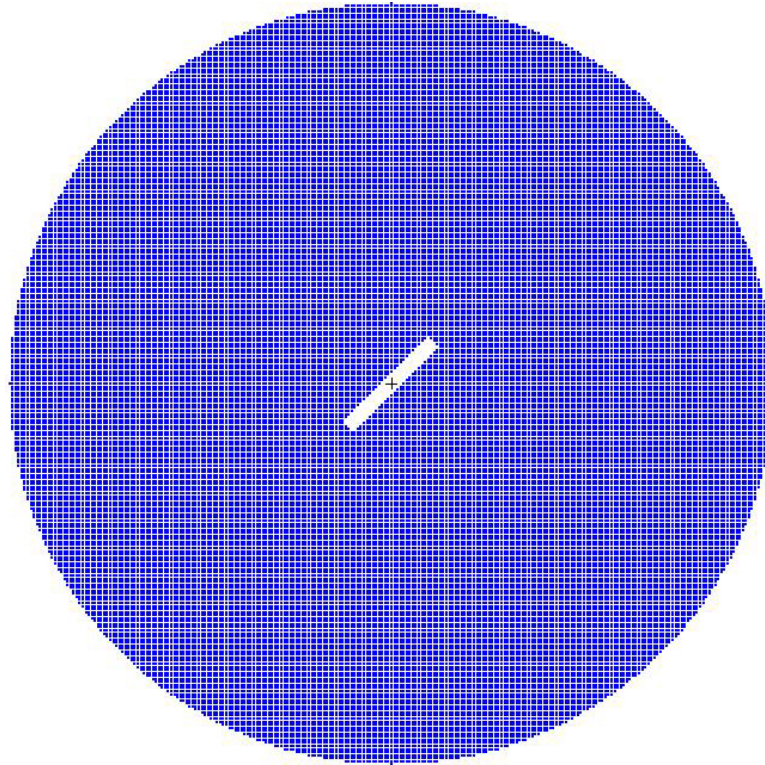
Yapay veriden silinen 5 dilimlik yatay noktalar için her iki yöntemin sonuçları incelendiğinde, Kalman filtresi için minimum, maksimum dZ ve standart sapma değerleri sırasıyla, -0.859mm , 2.605mm ve 0.761mm 'dir. Lineer enterpolasyon yöntemi için ise 0.335mm , 1.465mm ve 0.304mm 'dir. Her ne kadar standart sapmalar yakın olsa da, özellikle Şekil-5.18'e bakılarak, lineer enterpolasyon yönteminin ürettiği sonuçların biraz daha iyi olduğu söylenilebilir.

Tablo-5.2. Gürültüsüz verilerdeki yatay boşluklar için hesaplanan standart sapmalar

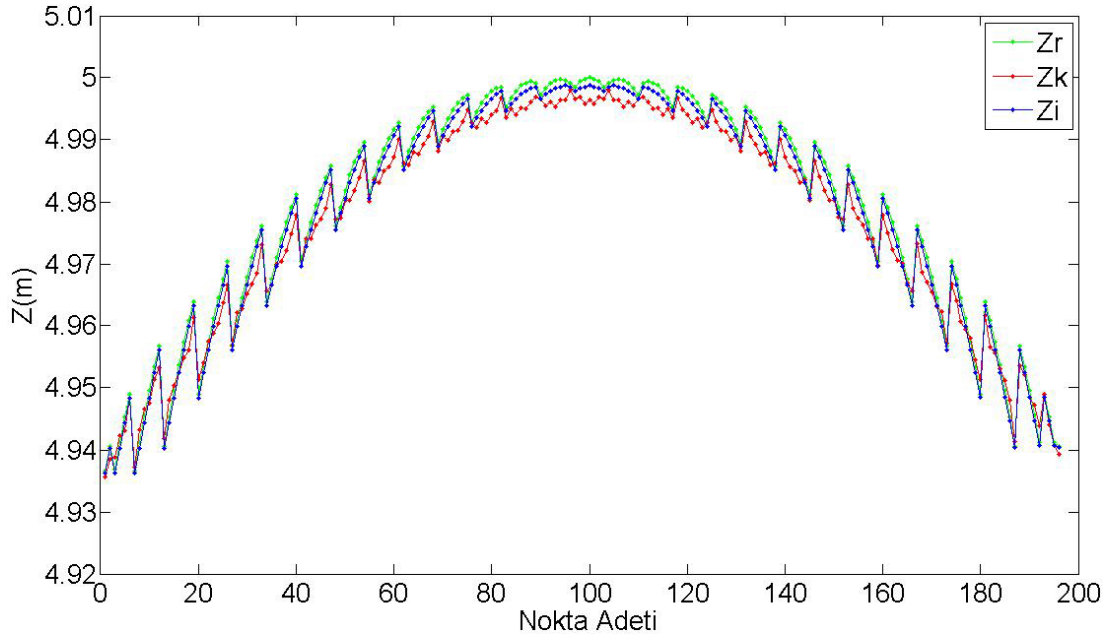
	2-B Kalman Filtresi			Lineer Enterpolasyon		
	Min dZ (mm)	Max dZ (mm)	Std sap (mm)	Min dZ (mm)	Max dZ (mm)	Std sap (mm)
Yatay 1 dilim	0.102	0.665	0.183	0.152	0.184	0.014
Yatay 3 dilim	-0.911	1.920	0.631	0.306	0.642	0.088
Yatay 5 dilim	-0.859	2.605	0.761	0.335	1.465	0.304

Tablo-5.2 incelendiğinde, lineer enterpolasyon yönteminin daha doğru değerler ürettiği görülmektedir. Fakat ilginç olan Kalman filtresinin yatayda, düşeye göre daha iyi sonuç vermesidir.

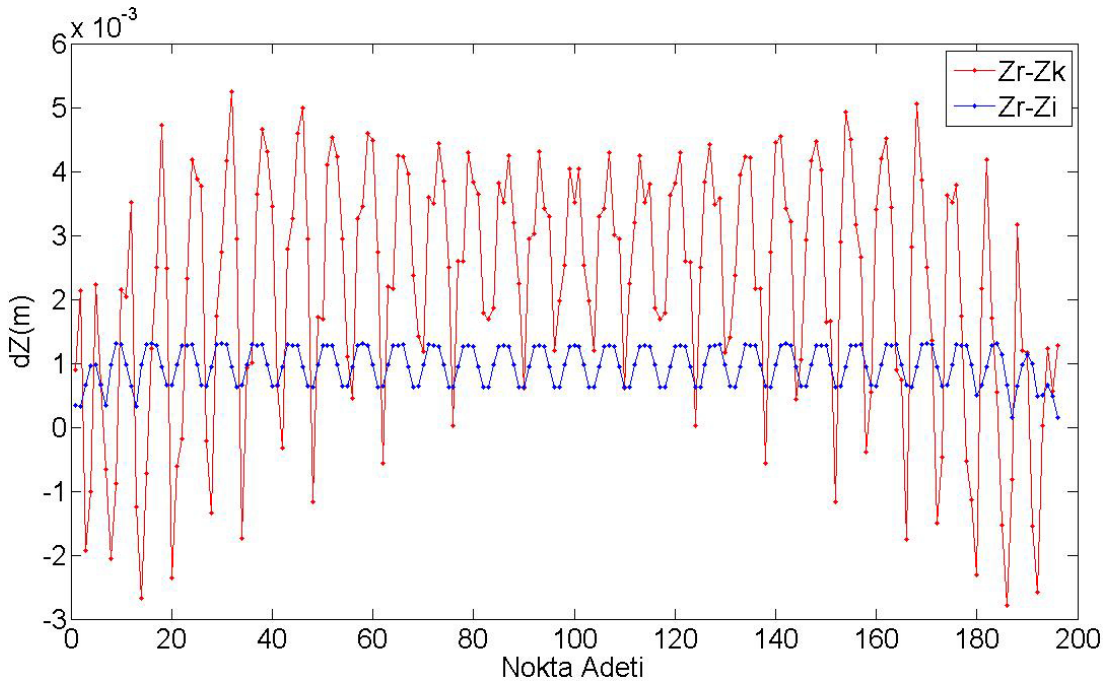
Gürültüsüz veri üzerinde yapılan son deney ise, örneklenen küre yüzeyi üzerinden eğik doğrultulu bir dilimin silinerek (Şekil-5.19), silinen bu boşluktaki Z değerlerinin Kalman filtresi ve lineer enterpolasyon yöntemiyle kestirilmeye çalışılmasıdır.



Şekil-5.19. Yapay veriden silinen eğik noktalar



Şekil-5.20. Yapay veriden silinen eğik noktaların Z_r , Z_k , Z_i grafiği



Şekil-5.21. Yapay veriden silinen eğik noktaların $Z_r - Z_k$, $Z_r - Z_i$ grafiği

Her iki yöntemin eğik boşluk için sonuçları şöyledir: İki boyutlu Kalman filtresi için minimum dZ değeri -2.793mm, maksimum dZ değeri 5.239mm ve

standart sapma 1.925mm'dir. Lineer enterpolasyon yönteminin sonuçları ise, minimum dZ değeri 0.155mm, maksimum dZ değeri 1.316mm ve standart sapma ise 0.298mm'dir. Bu deneme için de, lineer enterpolasyon yönteminin daha doğru sonuç verdiğini söylemek yanlış olmaz. Gürültüsüz veride yapılan tüm denemelerin sonuçları tek bir tabloda toplanırsa Tablo-5.3 elde edilir.

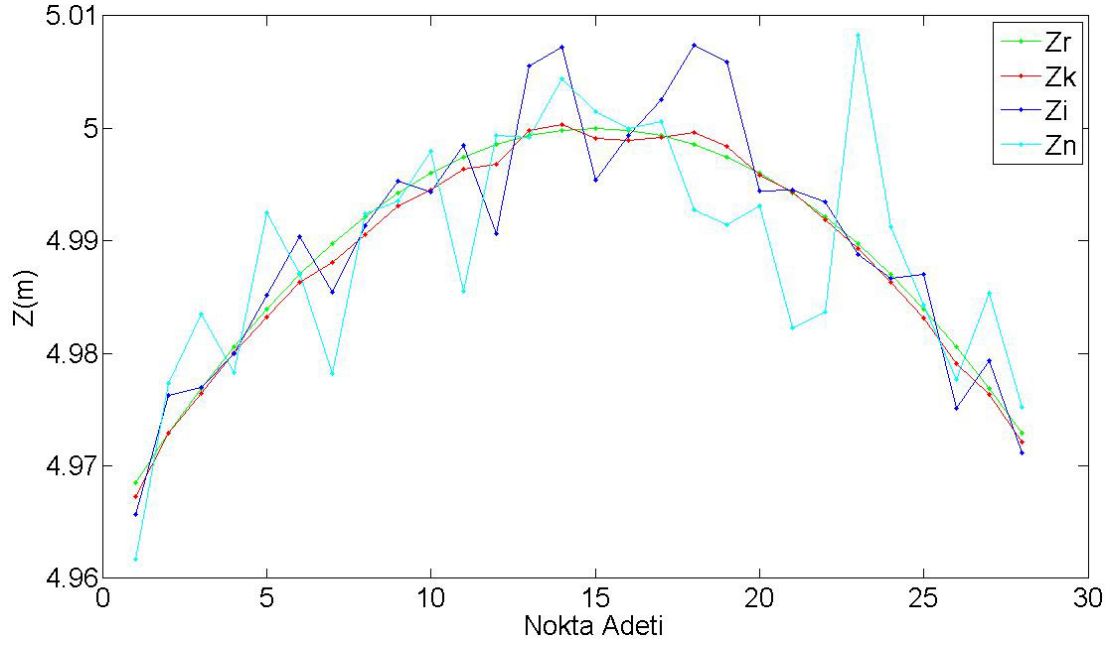
Tablo-5.3. Gürültüsüz verilerde yapılan tüm denemelerin özeti

	2-B Kalman Filtresi			Lineer Enterpolasyon		
	Min dZ (mm)	Max dZ (mm)	Std sap (mm)	Min dZ (mm)	Max dZ (mm)	Std sap (mm)
Düşey 1 dilim	0.145	0.685	0.178	0.152	0.184	0.014
Düşey 3 dilim	-1.078	3.705	1.356	0.306	0.642	0.088
Düşey 5 dilim	-2.401	9.565	3.345	0.335	1.465	0.304
Yatay 1 dilim	0.102	0.665	0.183	0.152	0.184	0.014
Yatay 3 dilim	-0.911	1.920	0.631	0.306	0.642	0.088
Yatay 5 dilim	-0.859	2.605	0.761	0.335	1.465	0.304
Eğik	-2.793	5.239	1.925	0.155	1.316	0.298

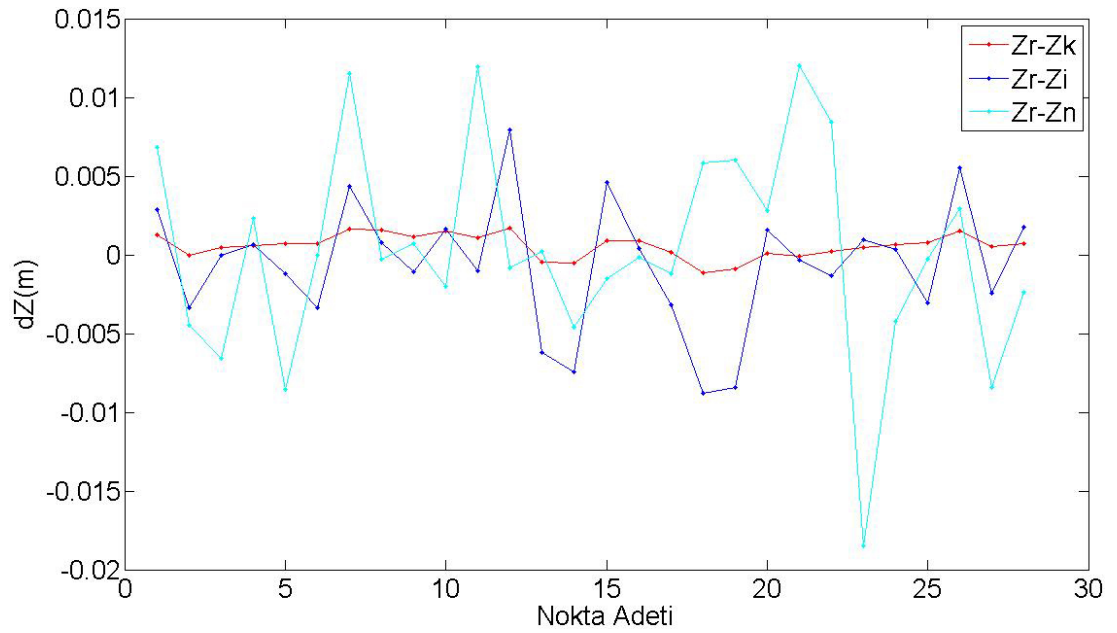
Tablo-5.3 incelendiğinde, lineer enterpolasyon yönteminin düşey ve yatay boşluklar için ürettiği sonuçlara bakılırsa, aynı dilimler için benzer sonuçlar çıkmıştır. Öte yandan 2-B Kalman filtresi, yatay boşluklarda düşeye nazaran daha doğru sonuç vermiştir.

Yersel lazer taramalarından elde edilen veri, her ölçmede olduğu gibi, gürültüye sahiptir. Dolayısıyla deneylerin daha gerçekçi sonuç vermesi için, yapay veriye normal dağılımlı gürültü eklenmiş ve sonrasında veri, Kalman filtresi ve lineer enterpolasyon yöntemleriyle işlenmiştir. Ek-5'te sunulan kod ile veriye normal dağılımlı, $\mu=0$ (ortalama) ve $\sigma=0.006m$ (standart sapma) olacak şekilde gürültü eklenmiştir. Gürültülü veride silinen noktalar,

gürültüsüz veridekiler ile aynı olacak şekilde seçilmiştir. Gürültülü veriler işlenirken Kalman filtresinde $R = \begin{bmatrix} 0.006^2 \end{bmatrix}$ alınmıştır.



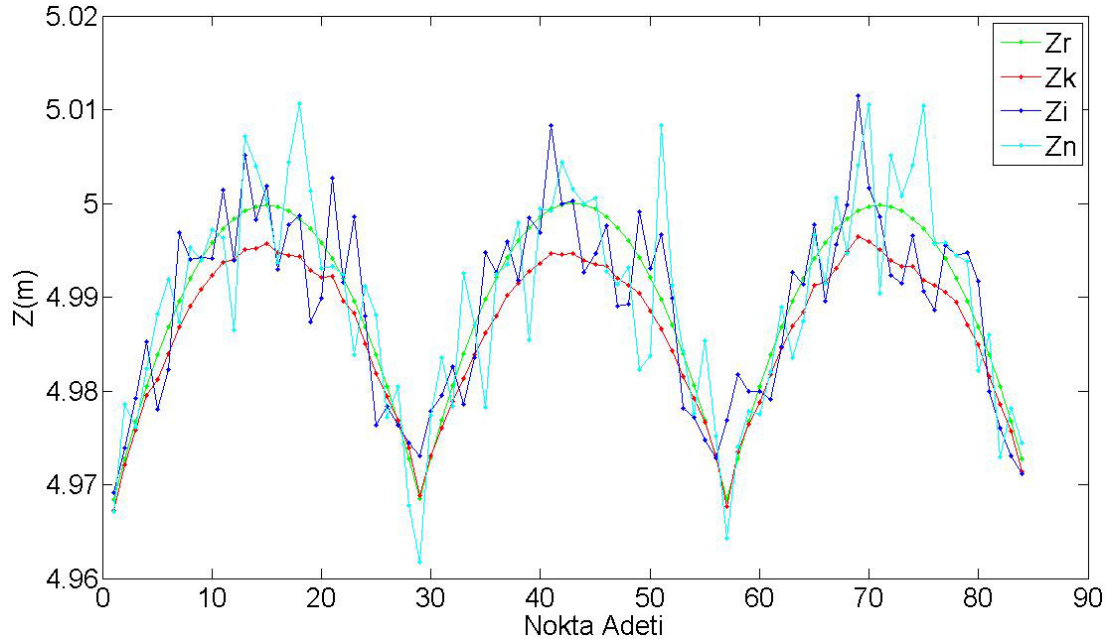
Şekil-5.22. Gürültülü yapay veriden silinen 1 dilimlik düşey noktaların Z_r , Z_k , Z_i , Z_n grafiği



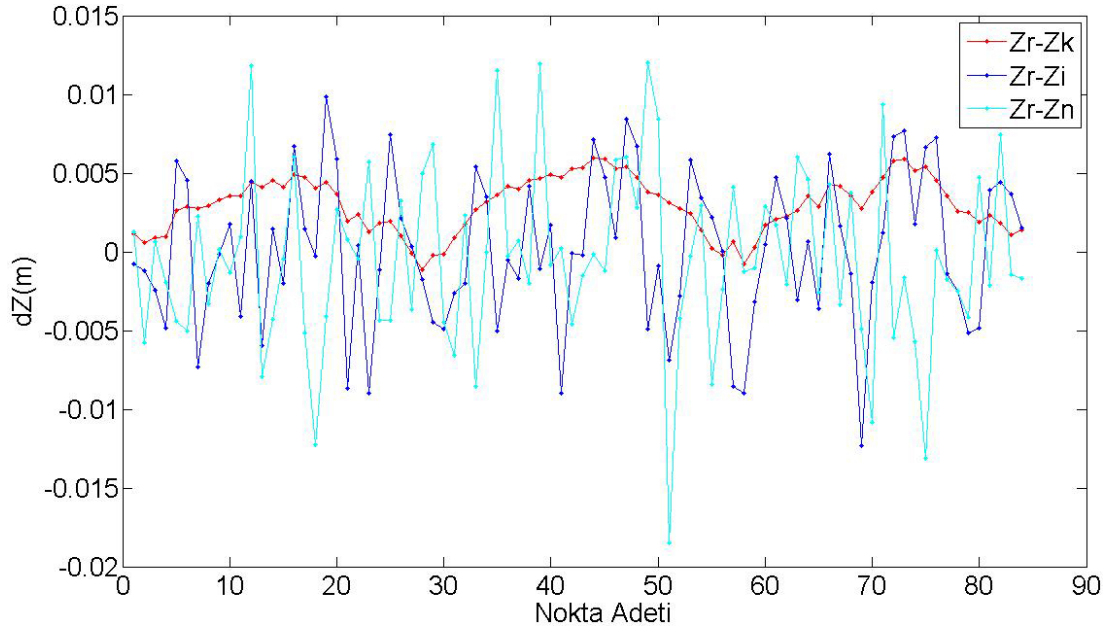
Şekil-5.23. Gürültülü yapay veriden silinen 1 dilimlik düşey noktaların Z_r-Z_k , Z_r-Z_i , Z_r-Z_n grafiği

Yukarıda Şekil-5.22 ve Şekil-5.23'de gürültülü yapay veri üzerinde yapılan ilk denemenin sonuçları görülmektedir. Gürültülü yapay veriden silinen 1 dilimlik düşey noktalar için iki boyutlu Kalman filtresinin ürettiği minimum, maksimum dZ ve standart sapma sırasıyla -1.130mm, 1.710mm ve 0.749mm'dir. Lineer enterpolasyon yöntemi için ise -8.818mm, 7.944mm ve 4.023mm'dir. Gürültülü veri üzerinde yapılan bu ilk denemede, iki boyutlu Kalman filtresi, lineer enterpolasyon yönteminden daha başarılı sonuç vermiştir.

Gürültülü yapay veri üzerinde yapılan ikinci denemede, her iki yöntemin 3 dilimlik boşluk için sonuçları elde edilmiştir (Şekil-5.24, Şekil-5.25).



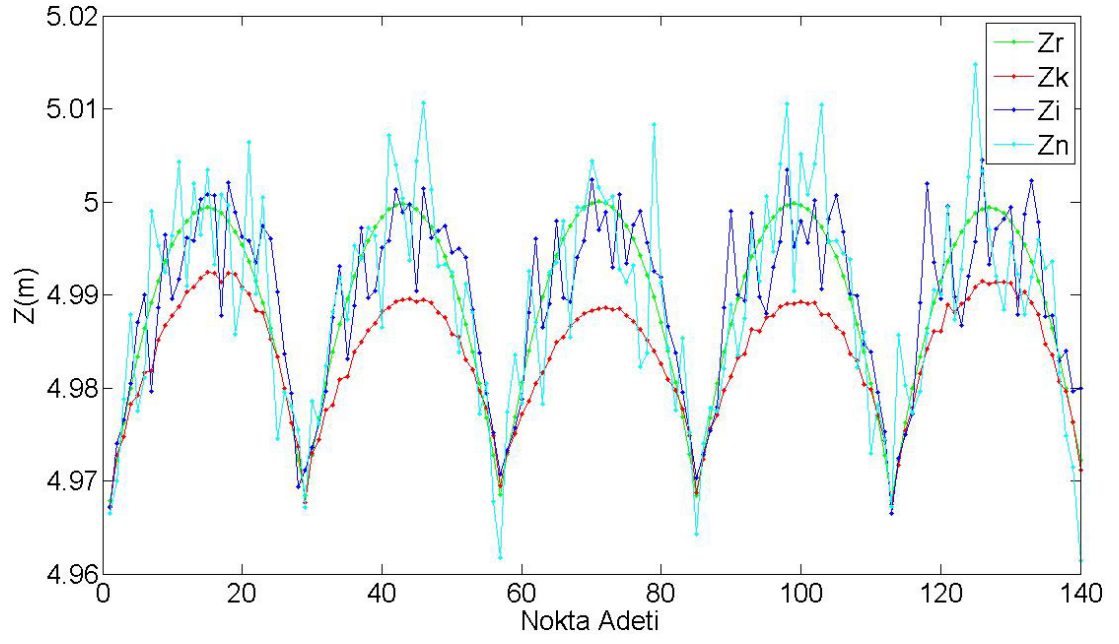
Şekil-5.24. Gürültülü yapay veriden silinen 3 dilimlik düşey noktaların Z_r , Z_k , Z_i , Z_n grafiği



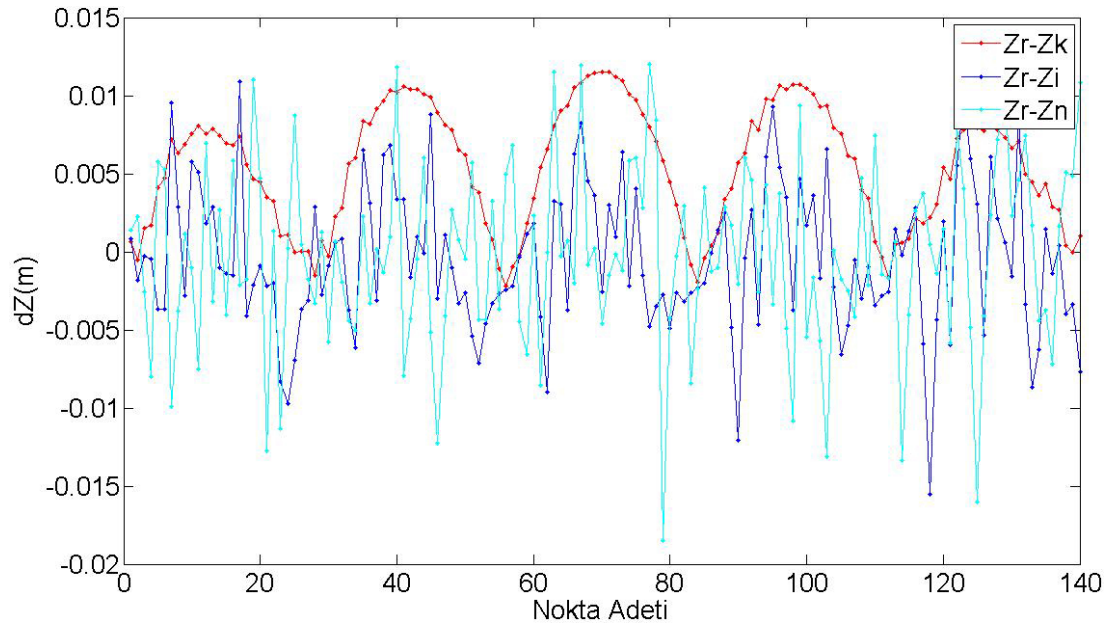
Şekil-5.25. Gürültülü yapay veriden silinen 3 dilimlik düşey noktaların Z_r-Z_k , Z_r-Z_i , Z_r-Z_n grafiği

Gürültülü veri üzerinde yapılan ikinci denemede, Kalman filtresi için minimum dZ değeri -1.167mm, maksimum dZ değeri 5.942mm ve standart sapma 1.749mm'dir. Lineer enterpolasyon yöntemine baktığımızda, minimum dZ değeri -12.337mm, maksimum dZ değeri 9.867mm ve standart sapma 4.792mm'dir. Bu denemede gösteriyor ki, iki boyutlu Kalman filtresi, gürültülü veri üzerinde daha başarılıdır.

Sonraki denemede, gürültülü yapay veriden silinen 5 dilimlik boşluk için, Şekil-5.26 ve Şekil-5.27'deki sonuçlar elde edilmiştir.



Şekil-5.26. Gürültülü yapay veriden silinen 5 dilimlik düşey noktaların Z_r , Z_k , Z_i , Z_n grafiği



Şekil-5.27. Gürültülü yapay veriden silinen 5 dilimlik düşey noktaların $Z_r - Z_k$, $Z_r - Z_i$, $Z_r - Z_n$ grafiği

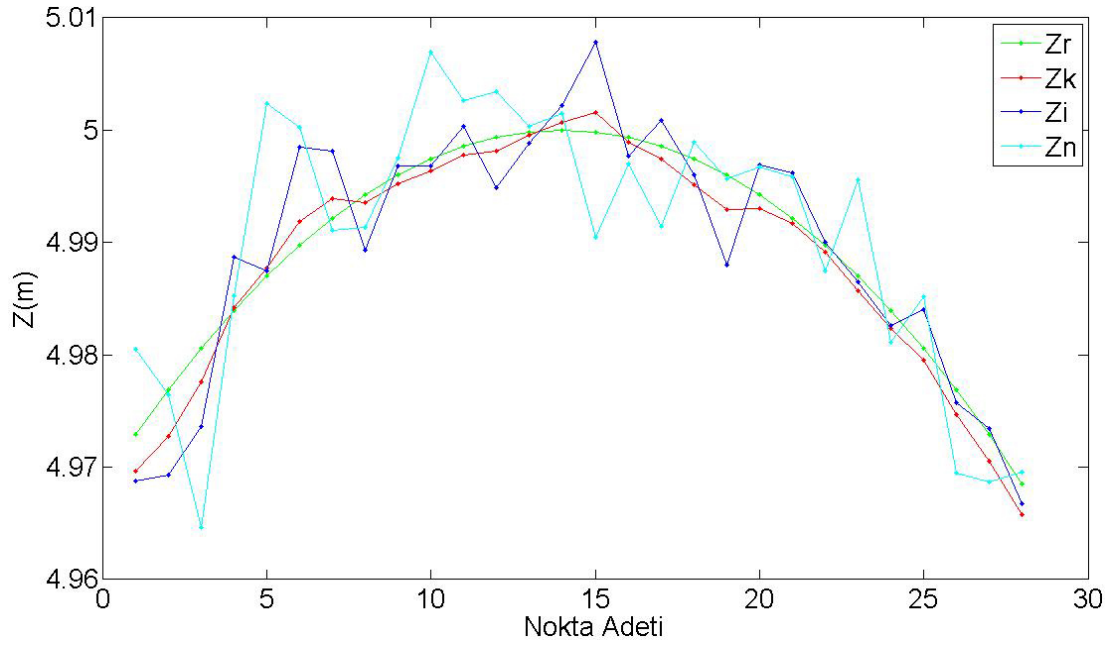
Gürültülü yapay veriden silinen 5 dilimlik düşey noktalar için her iki yöntemin sonuçları incelendiğinde, Kalman filtresi için minimum, maksimum dZ değerleri ve standart sapma sırasıyla -2.167mm, 11.535mm ve 3.826mm'dir. Lineer enterpolasyon yöntemi için ise, -15.553mm, 10.909mm ve 4.622mm'dir.

Tablo-5.4. Gürültülü verilerdeki düşey boşluklar için hesaplanan standart sapmalar

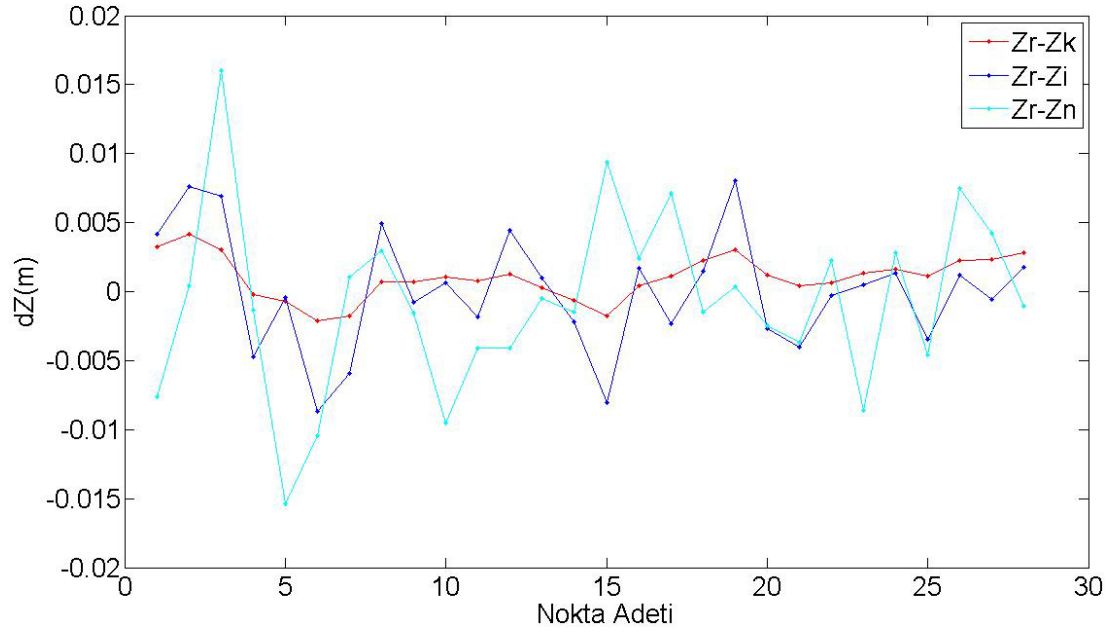
	2-B Kalman Filtresi			Lineer Enterpolasyon		
	Min dZ (mm)	Max dZ (mm)	Std sap (mm)	Min dZ (mm)	Max dZ (mm)	Std sap (mm)
Düşey 1 dilim	-1.130	1.710	0.749	-8.818	7.944	4.023
Düşey 3 dilim	-1.167	5.942	1.749	-12.337	9.867	4.792
Düşey 5 dilim	-2.167	11.535	3.826	-15.553	10.909	4.622

Şekil-5.22 den Şekil-5.27 kadar olan grafikler incelendiğinde, gürültülü verideki düşey boşluklarda Kalman filtresinin, lineer enterpolasyon yöntemine göre daha iyi sonuç verdiği görülmektedir. Verideki boşluk arttıkça 2-B Kalman filtresinin ürettiği sonuçların doğruluğu da düşmektedir.

Benzer şekilde bu iki yöntemin, gürültülü verideki yatay boşluklar için verdiği sonuçlar da araştırılmıştır. İlk olarak, gürültülü yapay veriden 1 dilimlik boşluk silinerek, her iki yöntem veriye uygulanmıştır. Yapılan bu ilk denemenin sonuçları Şekil-5.28 ve Şekil-5.29'da görülebilir.



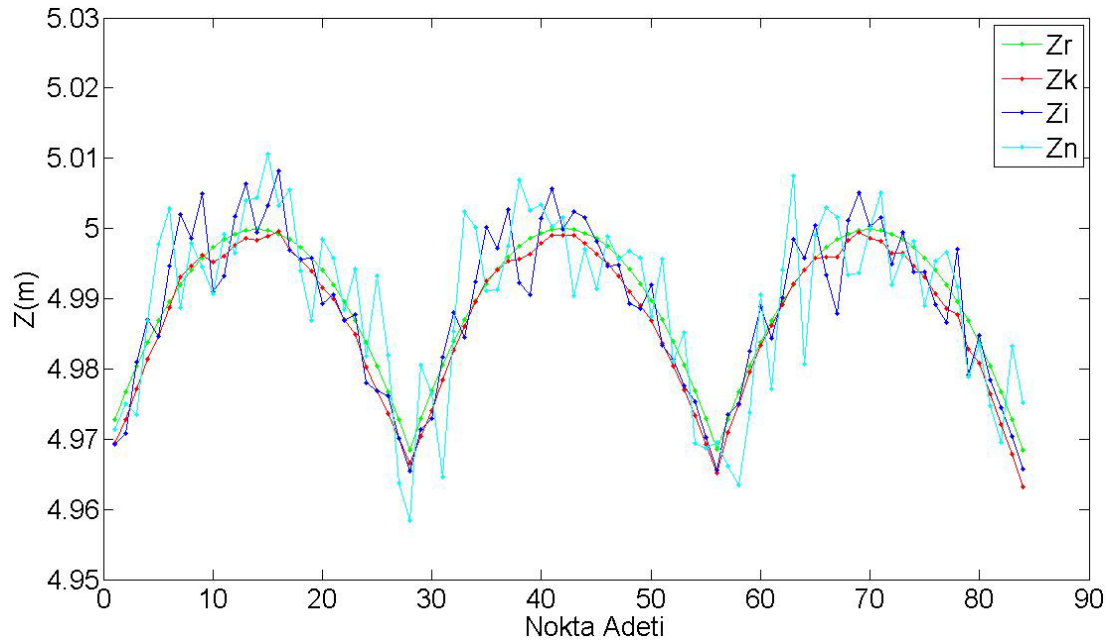
Şekil-5.28. Gürültülü yapay veriden silinen 1 dilimlik yatay noktaların Z_r , Z_k , Z_i , Z_n grafiği



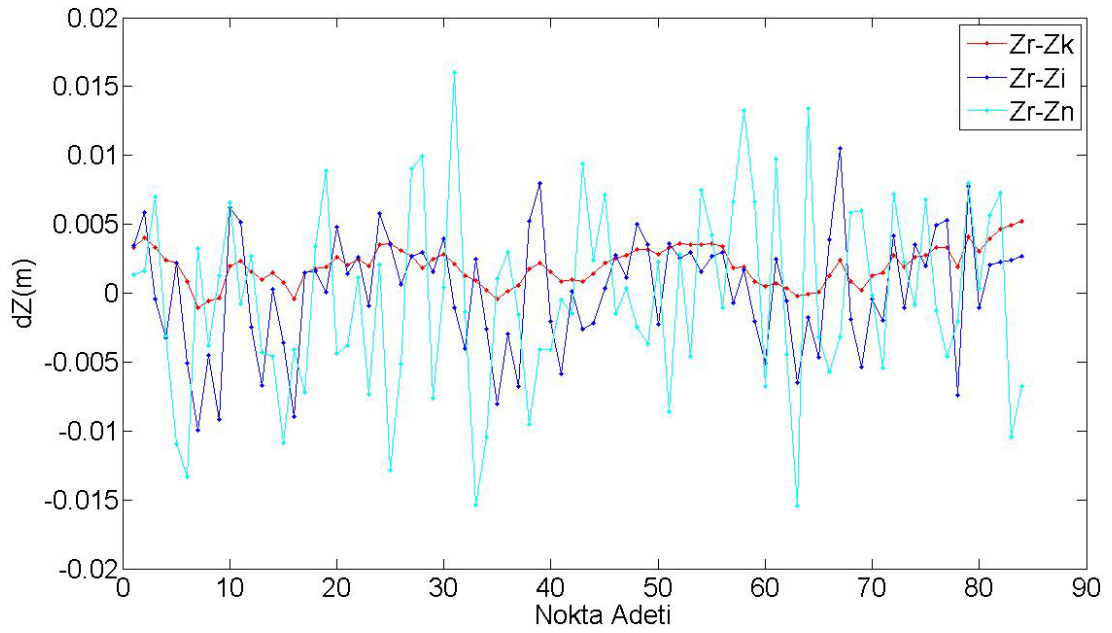
Şekil-5.29. Gürültülü yapay veriden silinen 1 dilimlik yatay noktaların Z_r-Z_k , Z_r-Z_i , Z_r-Z_n grafiği

Gürültülü yapay veriden silinen 1 dilimlik yatay noktalar için iki boyutlu Kalman filtresinin hesapladığı minimum dZ değeri -2.089mm , maksimum dZ değeri 4.193mm ve standart sapma 1.565mm 'dir. Lineer enterpolasyon yöntemi için ise -8.695mm , 8.010mm ve 4.272mm 'dir. Gürültülü yapay veride yapılan bu denemede de 2-B Kalman filtresi daha başarılıdır.

Benzer şekilde, gürültülü yapay verideki 3 dilimlik boşluklar için yapılan denemeden elde edilen sonuçlar, Şekil-5.30 ve Şekil-5.31'de gösterilmiştir.



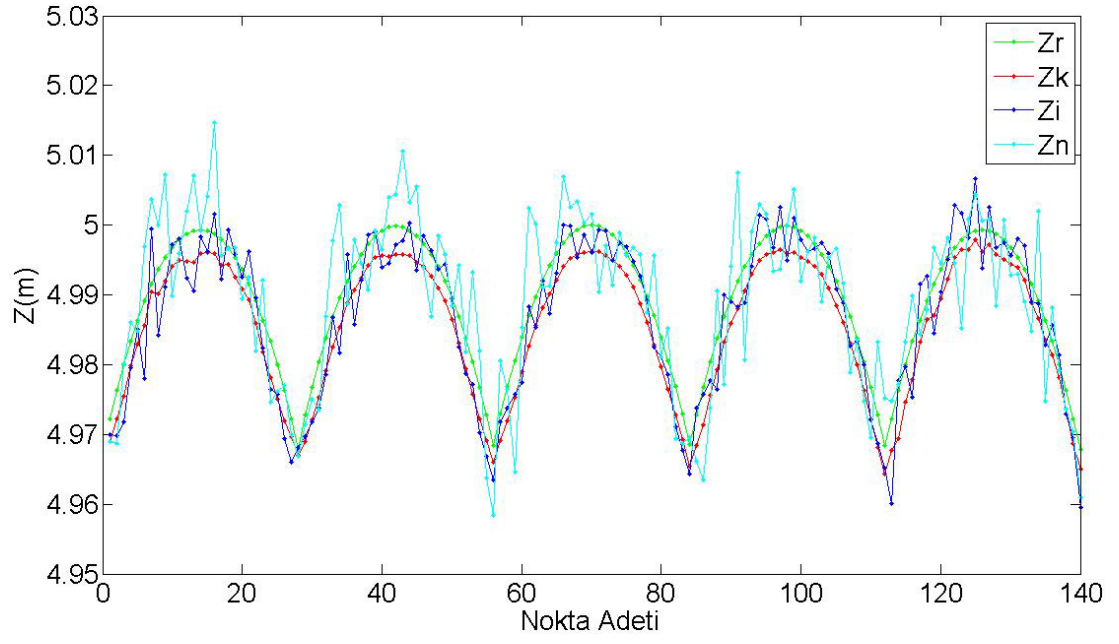
Şekil-5.30. Gürültülü yapay veriden silinen 3 dilimlik yatay noktaların Z_r , Z_k , Z_i , Z_n grafiği



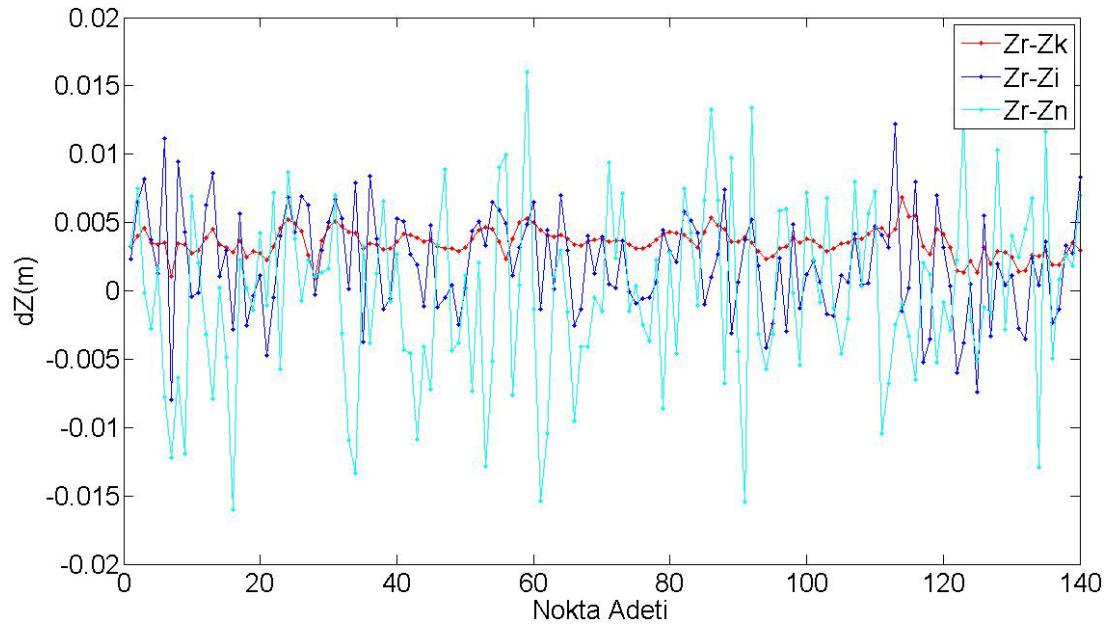
Şekil-5.31. Gürültülü yapay veriden silinen 3 dilimlik yatay noktaların Z_r-Z_k , Z_r-Z_i , Z_r-Z_n grafiği

Gürültülü yapay veriden silinen 3 dilimlik yatay noktalar için, iki boyutlu Kalman filtresinin ürettiği sonuçlar şöyledir: Minimum dZ değeri -1.061mm, maksimum dZ değeri 5.209mm ve standart sapma 1.357mm'dir. Lineer enterpolasyon yöntemi ise, minimum dZ için -9.948mm değerini, maksimum dZ için 10.487mm değerini ve standart sapma için 4.245mm değerini üretmiştir. Bu denemede de, iki boyutlu Kalman filtresi, lineer enterpolasyon yönteminden daha doğru sonuç sağlamaktadır.

Gürültülü yapay veriden silinen 5 dilimlik nokta kümesi sonucu elde edilen veri üzerinde yapılan denemelerin sonucu, aşağıda Şekil-5.32 ve Şekil-5.33'de gösterilmektedir.



Şekil-5.32. Gürültülü yapay veriden silinen 5 dilimlik yatay noktaların Z_r , Z_k , Z_i , Z_n grafiği



Şekil-5.33. Gürültülü yapay veriden silinen 5 dilimlik yatay noktaların Z_r-Z_k , Z_r-Z_i , Z_r-Z_n grafiği

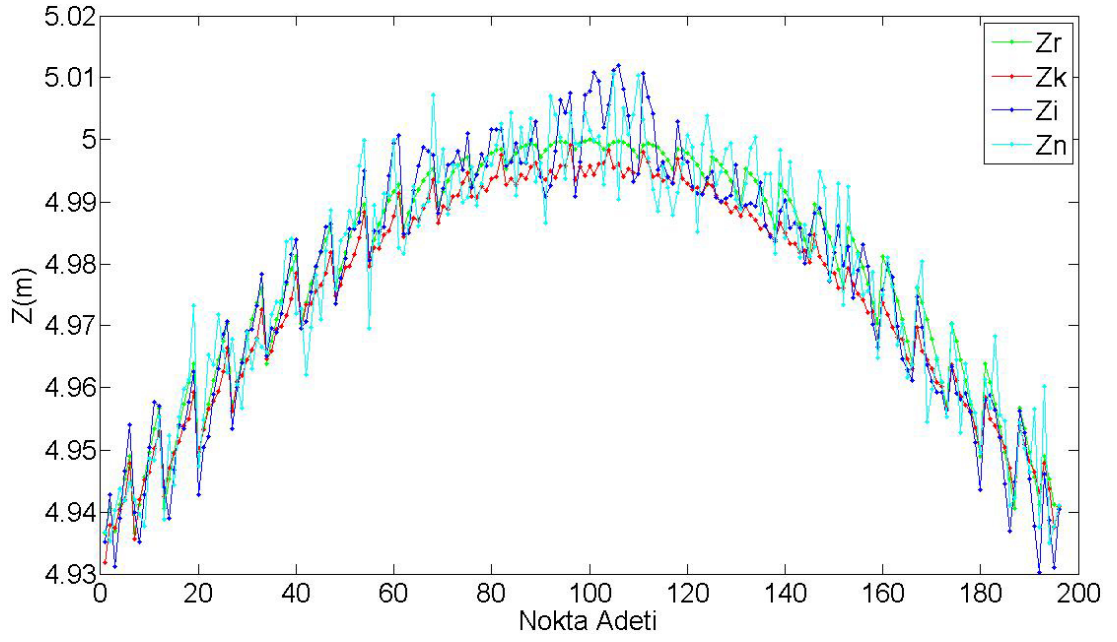
Gürültülü yapay veriden silinen 5 dilimlik yatay noktalar için Şekil-5.33 incelendiğinde, Kalman filtresi için minimum dZ, maksimum dZ ve standart sapma değerleri sırasıyla 0.931mm, 6.863mm ve 0.955mm'dir. Aynı değerler lineer interpolasyon yöntemi için sırasıyla, -7.984mm, 12.191mm ve 3.747mm'dir. Gürültülü veriler üzerinde yapılan tüm denemelerde olduğu gibi, iki boyutlu Kalman filtresi, lineer interpolasyon yöntemine göre daha doğru sonuç üretmiştir.

Tablo-5.5. Gürültülü verilerdeki yatay boşluklar için hesaplanan standart sapmalar

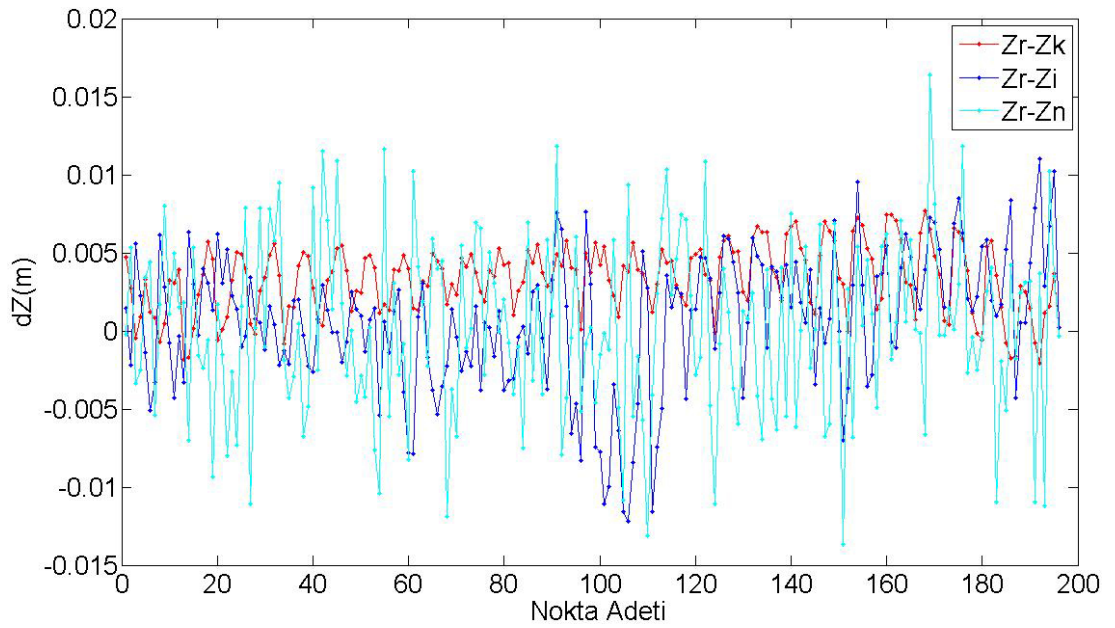
	2-B Kalman Filtresi			Lineer Enterpolasyon		
	Min dZ (mm)	Max dZ (mm)	Std sap (mm)	Min dZ (mm)	Max dZ (mm)	Std sap (mm)
Yatay 1 dilim	-2.089	4.193	1.565	-8.695	8.010	4.272
Yatay 3 dilim	-1.061	5.209	1.357	-9.948	10.487	4.245
Yatay 5 dilim	0.931	6.863	0.955	-7.984	12.191	3.747

Tablo-5.5 incelendiğinde, Kalman filtresinin gürültülü verideki yatay boşluklar için, lineer enterpolasyon yöntemine göre daha iyi sonuç verdiği görülmektedir.

Söz konusu karşılaştırma, gürültülü verideki eğik boşluklar için yapıldığında, sonuç aşağıdaki gibidir (Şekil-5.34, Şekil-5.35).



Şekil-5.34. Gürültülü yapay veriden silinen eğik noktaların Z_r , Z_k , Z_i , Z_n grafiği



Şekil-5.35. Gürültülü yapay veriden silinen eğik noktaların $Z_r - Z_k$, $Z_r - Z_i$, $Z_r - Z_n$ grafiği

Yapılan son denemede, gürültülü yapay veriden silinen eğik noktalar için Kalman filtresi, minimum dZ , maksimum dZ ve standart sapma değerleri

sırasıyla -2.052mm, 7.669mm ve 2.182mm'dir. Lineer enterpolasyon yöntemi ise sırasıyla -12.216mm, 11.019mm ve 4.328mm'dir. Sonuç, önceki denemeleri desteklemektedir. Gürültülü veride, iki boyutlu Kalman filtresi daha başarılıdır. Gürültülü veri üzerinde yapılan tüm denemeler tek bir tablo altında toplanırsa, Tablo-5.6 elde edilir.

Tablo-5.6. Gürültülü verilerde yapılan tüm denemelerin özeti

	2-B Kalman Filtresi			Lineer Enterpolasyon		
	Min dZ (mm)	Max dZ (mm)	Std sap (mm)	Min dZ (mm)	Max dZ (mm)	Std sap (mm)
Düşey 1 dilim	-1.130	1.710	0.749	-8.818	7.944	4.023
Düşey 3 dilim	-1.167	5.942	1.749	-12.337	9.867	4.792
Düşey 5 dilim	-2.167	11.535	3.826	-15.553	10.909	4.622
Yatay 1 dilim	-2.089	4.193	1.565	-8.695	8.010	4.272
Yatay 3 dilim	-1.061	5.209	1.357	-9.948	10.487	4.245
Yatay 5 dilim	0.931	6.863	0.955	-7.984	12.191	3.747
Eğik	-2.052	7.669	2.182	-12.216	11.019	4.328

Gürültülü veri üzerinde yapılan son deney de gösteriyor ki, Kalman filtresi, lineer enterpolasyon yönteminden daha başarılıdır. Gürültüsüz veri üzerinde yapılan denemelerde, 2-B Kalman filtresinin yatay boşluklarda, düşey boşluklara nazaran daha doğru sonuç verdiği gözlenmişti. Gürültülü veri üzerinde yapılan denemelerde de bu sonuç değişmemiştir. Filtreyi daha dengeli hale getirmek için, soldan sağa-yukarıdan aşağıya ve sağdan sola-aşağıdan yukarıya iki adımlı uygulanan filtreye, aşağıdan yukarıya-sağdan sola ve yukarıdan aşağıya-soldan sağa işleme adımları eklenerek filtre dört adımlı hale getirilebilir.

Yukarıda sunulan tüm deney sonuçları incelendiğinde, gürültüsüz veride lineer enterpolasyon yönteminin daha başarılı olduğu, gürültülü veride ise iki boyutlu Kalman filtresinin daha başarılı olduğu açık olarak

görülmektedir. İki boyutlu Kalman filtresi, gürültünün olumsuz etkisini bastırarak daha doğru sonuç üretirken, lineer enterpolasyon yöntemi gürültüden olumsuz etkilenmektedir.

6. UYGULAMA

Yersel lazer tarayıcılar sundukları avantajlardan ötürü, günümüzde birçok mühendislik ve mimari projelerde uygulama alanları kazanmışlardır. Üç boyutlu veriyi hızlı ve doğru bir şekilde sağlamaları bakımından, yersel lazer tarayıcılar gelecekte daha da yaygın olarak kullanılacaklardır. Yersel lazer tarayıcılar kullanılırken karşılaşılan en büyük problemlerden birisi de, tarama yaparken, taranan cismin önünden geçen insanlar, araçlar veya taranan cismin önünde yer alan diğer cisimlerin, asıl taranan cismin üzerinde gölgeler bırakarak boşluklara neden olmasıdır.

Bu tezle amaçlanan, bu boşlukların, iki boyutlu Kalman filtresi ile doğru bir şekilde doldurulup doldurulamayacağını araştırmasını yapmaktır. Araştırmalarımız sonucunda Wang (1998)'de tanıtılan algoritma esas alınarak bir uygulamaya gidilmiştir.

Bu amaçla, 4. bölümde anlatılan iki boyutlu Kalman filtresi, 5. bölümde sunulan yapay verilerdeki denemeler esas alınarak, gerçek yersel lazer tarama verisine uygulandı.

Örnek veri olarak, Bimtaş A.Ş.'nin, Tarihi Yarımada projesi kapsamındaki Süleymaniye Camii'nde yapılan iç kubbe taramaları kullanıldı. Veri olarak bu taramaların seçilmesinin nedenleri şunlardır: İlk olarak, taramada kullanılan alet (HDS3000 Leica), üzerinde bulunan küresel düzeç

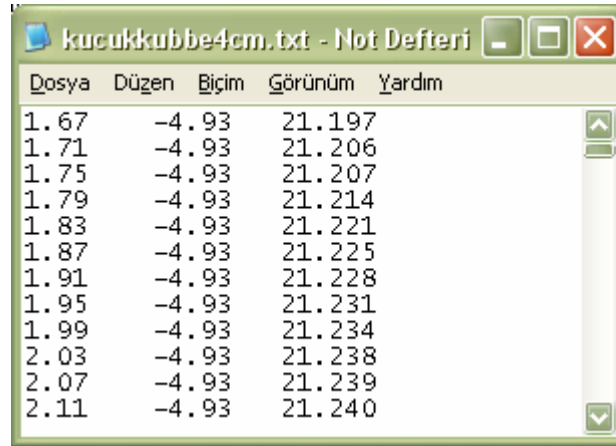
sayesinde, tarama koordinat sistemindeki Z koordinatı, yükseklikle çakışık durumdadır. Böylelikle hiçbir koordinat dönüşümü yapmaksızın 4. bölümde anlatılan denklemler kullanılabilir. İkinci neden ise veride, avize ve avize kablolarından kaynaklanan, istenmeyen boşlukların olmasıdır.

Filtreyi veri üzerinde çalıştırabilmek için, verinin belirli aralıklarda gridlenmiş olması gerekir. Arazide yapılan taramalar bu nitelikte değildir. Bu yüzden, yaklaşık 2.5cm çözünürlüğe sahip ham taramalar önce, Polyworks yazılımına import edilerek, X,Y düzleminde 4cm nokta aralığına sahip olacak şekilde gridlenmiştir. Böylelikle veri daha da seyreltilerek, çalışması daha kolay bir hal almıştır. Çünkü lazer nokta bulutları üzerinde çalışırken karşılaşılan en büyük sorunlardan birisi de, veri miktarının çok büyük olmasıdır.

Nokta bulutunu, Matlab programında okuyabilmek için her noktanın X, Y, Z koordinatları bir satırda olacak şekilde ascii formatta hazırlandı. Bu dosyayı Matlab'ta

```
M_in = dlmread('kucukkubbe4cm.txt')
```

komutuyla direk olarak bir diziye atamak mümkündür. Bu dizinin boyutu (dosyadaki satır sayısı x sütun sayısı) dır.



Dosya	Düzen	Biçim	Görünüm	Yardım
1.67	-4.93	21.197		
1.71	-4.93	21.206		
1.75	-4.93	21.207		
1.79	-4.93	21.214		
1.83	-4.93	21.221		
1.87	-4.93	21.225		
1.91	-4.93	21.228		
1.95	-4.93	21.231		
1.99	-4.93	21.234		
2.03	-4.93	21.238		
2.07	-4.93	21.239		
2.11	-4.93	21.240		

Şekil-6.1. Nokta Bulutu ascii dosya formatı (X,Y,Z) koordinatları

Filtrenin veri üzerinde çalıştırılması için verinin bir matris gibi olması gerekir. Veri daha önce belirli bir araklıkla gridlendiği için, X ve Y koordinatlarına göre matris formuna getirilebilir. Dosyadaki minimum ve maksimum X ve Y değerleri ve örnekleme aralığına göre veri, matris formuna getirilir.

```
sorted = sort(M_in,1)
```

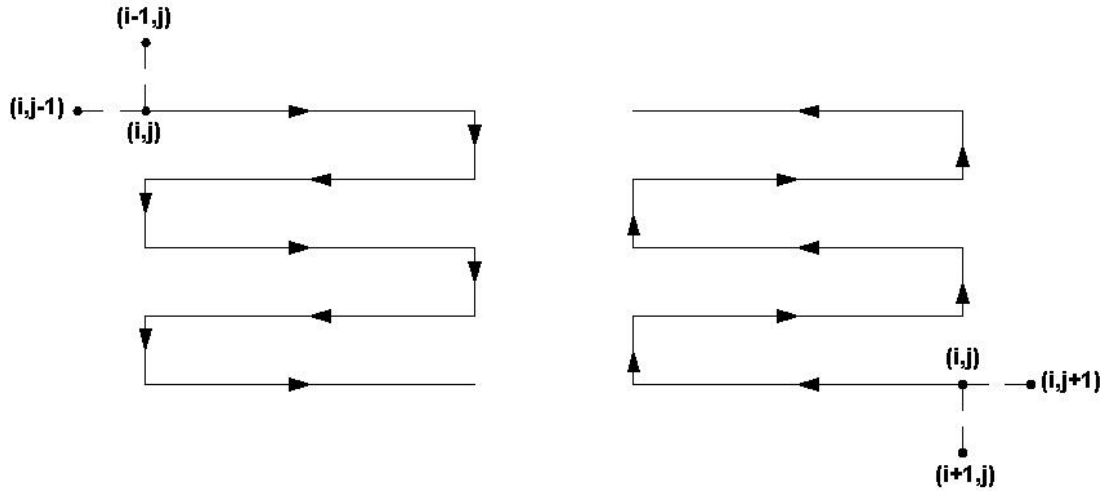
komutuyla veri sıralandıktan sonra,

```
x_min = sorted(1,1)
y_min = sorted(1,2)
x_max = sorted(row_M_in,1)
y_max = sorted(row_M_in,2)
```

komutlarıyla minimum ve maksimum değerler bulunabilir. Sonraki işlem, bu matris üzerindeki her hücre için ölçme alanı dışındaki noktalara *NaN* (not a number yani boş değer), nokta varsa 1, tarama alanı içerisinde boşluk varsa 0 değerlerini atamaktır. Bu atamadan sonra veri, üzerinde filtreyi çalıştırmaya hazır hale gelmektedir.

Filtre, Şekil-6.2’de görüldüğü üzere, iki adımda, gridlenmiş veri üzerinde gezerek, boşluklarda en uygun Z koordinat değerlerini

hesaplanmaktadır. Her bir adım ayrı ayrı hesaplanarak, daha sonra (4.10)'daki eşitlikler kullanılarak ortalama değeri hesaplanmaktadır.



Şekil-6.2. Kullanılan Kalman filtresinin işlem adımları

Algoritma her işlem adımında, gidiş yönünün solundaki ve gerisindeki noktaları kullanarak, o an işlenmekte olan nokta için gerekli bilgileri toplar. Filtre, bu iki noktadan herhangi birinin ölçülmemiş olması durumunda, gerekli ağırlık değişikliklerini yapıp işleme devam etmektedir. Bu ağırlıklar 4. bölümde anlatılan C ve B matrisleri ile ayarlanmaktadır. İterasyonun ilk noktası için

$$S^- = \begin{bmatrix} z \\ 0 \\ 0 \end{bmatrix}$$

$$P = Q$$

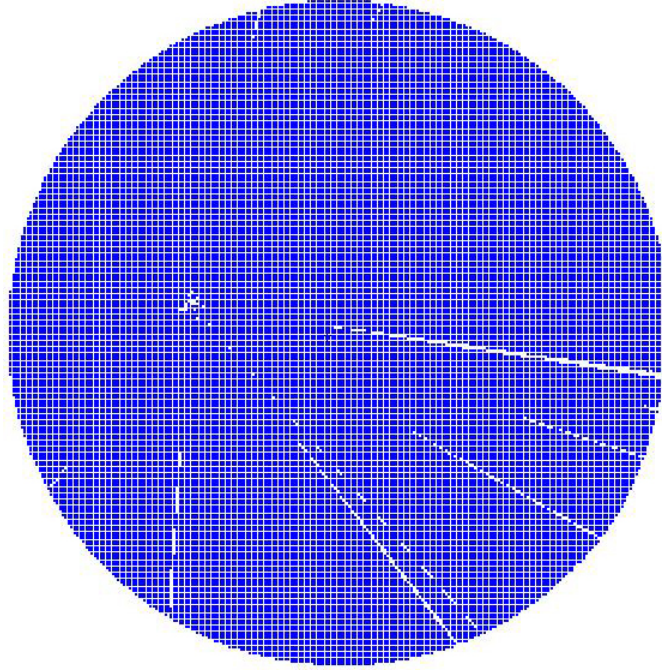
kabul edilir.

Son olarak her iki adımda hesaplanan ve ortalaması alınan değerler, matriste 0 olarak atanan yani tarama alanı içerisinde olup da, boş olan

noktalar, aşağıdaki komutla bir ascii dosyaya X,Y,Z koordinatları olarak yazılır.

```
dlmwrite('bosluk_output.txt', Bosluk, 'delimiter', '\t',  
'precision', '%.3f')
```

Uygulamanın son bölümünde, bu tezde sunulan iki boyutlu Kalman filtresi, Süleymaniye Camii'nin taranan kubbelerinden birine uygulandı. Tarama sırasında kubbe üzerinde oluşan boşlukların doldurulması amaçlandı.



Şekil-6.3. Orijinal veri üstten görünüm

Tablo-6.2. İlk adımda hesaplanan değerler

	205	206	207	208	209	210	211	212	213	214	215	216	217
88	[22 367,5...	[22 323,5...	[22 275,2...	[22 227,2...	[22 175,5...	[22 118,-8...	[22 065,-1...	[22 011,4...	[21 958,2...	[21 903,2...	[21 843,2...	[21 782,1...	[21 707,1...
89	[22 39,5.8...	[22 345,5...	[22 295,2...	[22 247,2...	[22 195,5...	[22 141,-8...	[22 083,-1...	[22 029,3...	[21 978,2...	[21 92,2.6...	[21 866,2...	[21 798,1...	[21 372,-1...
90	[22 396,6...	[22 349,5...	[22 303,2...	[22 252,2...	[22 199,5...	[22 145,-8...	[22 089,-1...	[22 036,4...	[21 981,2...	[21 928,2...	[21 869,2...	[21 811,1...	[21 726,5...
91	[22 41,5.9...	[22 369,5...	[22 32,2.2...	[22 268,2...	[22 219,5...	[22 164,-8...	[22 11,-1...	[22 057,4...	[22 004,2...	[21 944,2...	[21 89,2.5...	[21 831,1...	[21 775,4...
92	[22 415,6...	[22 371,5...	[22 323,2...	[22 272,2...	[22 224,5...	[22 172,-8...	[22 116,-1...	[22 059,4...	[22 004,2...	[21 952,2...	[21 895,2...	[21 839,1...	[21 779,7...
93	[22 432,6...	[22 383,5...	[22 339,2...	[22 292,2...	[22 244,5...	[22 192,-8...	[22 138,-1...	[22 08,4.1...	[22 025,2...	[21 967,2...	[21 913,2...	[21 86,1.4...	[21 798,6...
94	[22 438,6...	[22 39,5.8...	[22 342,2...	[22 292,2...	[22 246,5...	[22 197,-8...	[22 144,-1...	[22 086,4...	[22 027,2...	[21 971,2...	[21 915,2...	[21 859,1...	[21 801,8...
95	[22 455,6...	[22 404,5...	[22 357,2...	[22 31,2.3...	[22 261,5...	[22 21,-8...	[22 163,-1...	[22 102,4...	[22 046,2...	[21 987,2...	[21 935,2...	[21 877,1...	[21 821,7...
96	[22 453,6...	[22 408,5...	[22 356,2...	[22 31,2.3...	[22 256,5...	[22 211,-8...	[22 163,-1...	[22 104,4...	[22 045,2...	[21 987,2...	[21 935,3...	[21 876,1...	[21 82,9.8...
97	[22 469,6...	[22 417,5...	[22 371,2...	[22 318,2...	[22 273,5...	[22 226,-8...	[22 173,-1...	[22 123,4...	[22 063,2...	[22 009,2...	[21 947,2...	[21 894,1...	[21 835,7...
98	[22 465,6...	[22 416,5...	[22 367,2...	[22 321,2...	[22 273,5...	[22 226,-8...	[22 173,-1...	[22 121,4...	[22 064,2...	[22 008,2...	[21 951,3...	[21 896,1...	[21 84,1.0...
99	[22 473,6...	[22 429,5...	[22 381,2...	[22 332,2...	[22 287,5...	[22 236,-8...	[22 185,-1...	[22 134,4...	[22 079,2...	[22 024,2...	[21 968,2...	[21 911,1...	[21 853,8...
100	[22 476,6...	[22 43,6.0...	[22 381,2...	[22 338,2...	[22 289,5...	[22 234,-8...	[22 184,-1...	[22 131,4...	[22 079,2...	[22 028,2...	[21 973,3...	[]	
101	[22 493,6...	[22 446,5...	[22 401,2...	[22 352,2...	[22 32,5.7...	[22 277,-8...	[22 231,-1...	[22 181,4...	[22 127,2...	[22 079,2...	[22 026,3...	[22 026,0...	[22 026,0...
102	[22 447,6...	[22 401,5...	[22 357,2...	[22 314,2...	[22 276,5...	[22 231,-8...	[22 185,-1...	[22 139,4...	[22 097,3...	[22 044,3...	[21 986,3...	[21 931,-6...	[21 87,-1...
103	[22 504,2...	[22 457,2...	[22 409,2...	[22 361,2...	[22 314,-3...	[22 261,-1...	[22 208,-2...	[22 16,-1...	[22 108,3...	[22 051,3...	[21 997,1...	[21 942,7...	[21 882,9...
104	[22 501,-6...	[22 459,-5...	[22 408,1...	[22 362,1...	[22 311,-2...	[22 262,-4...	[22 208,-5...	[22 157,-3...	[22 102,4...	[22 047,4...	[21 997,1...	[21 94,4.2...	[21 881,1...
105	[22 508,-8...	[22 468,-7...	[22 421,1...	[22 371,1...	[22 324,-2...	[22 273,-4...	[22 222,-5...	[22 173,-4...	[22 118,4...	[22 062,4...	[22 008,1...	[21 951,5...	[21 894,1...
106	[22 509,-9...	[22 466,-8...	[22 421,1...	[22 37,1.1...	[22 319,-3...	[22 27,-5...	[22 219,-5...	[22 17,-4...	[22 117,4...	[22 061,4...	[22 006,1...	[21 947,6...	[21 889,1...
107	[22 521,-9...	[22 478,-8...	[22 428,1...	[22 381,1...	[22 332,-3...	[22 282,-5...	[22 232,-6...	[22 179,-4...	[22 124,4...	[22 067,4...	[22 017,1...	[21 96,6.3...	[21 904,1...
108	[22 516,-1...	[22 472,-8...	[22 423,1...	[22 375,1...	[22 331,-3...	[22 278,-5...	[22 23,-6...	[22 175,-4...	[22 121,4...	[22 066,4...	[22 013,1...	[21 958,6...	[21 9,1.52...
109	[22 528,-1...	[22 482,-9...	[22 434,1...	[22 385,1...	[22 341,-3...	[22 288,-5...	[22 241,-6...	[22 188,-4...	[22 134,4...	[22 077,4...	[22 024,1...	[21 969,6...	[21 907,1...
110	[22 525,-1...	[22 478,-9...	[22 428,1...	[22 381,1...	[22 336,-3...	[22 287,-5...	[22 236,-6...	[22 185,-4...	[22 131,4...	[22 075,4...	[22 021,1...	[21 962,6...	[21 903,1...
111	[22 532,-1...	[22 486,-9...	[22 437,1...	[22 388,1...	[22 343,-3...	[22 294,-5...	[22 244,-6...	[22 193,-4...	[22 14,4.4...	[22 083,4...	[22 028,1...	[21 972,6...	[21 915,1...
112	[22 528,-1...	[22 481,-9...	[22 434,1...	[22 386,1...	[22 339,-3...	[22 29,-5...	[22 239,-6...	[22 186,-4...	[22 13,4.4...	[22 076,4...	[22 023,1...	[21 967,6...	[21 911,1...
113	[22 534,-1...	[22 492,-9...	[22 445,1...	[22 394,1...	[22 344,-3...	[22 295,-5...	[22 248,-6...	[22 197,-4...	[22 146,4...	[22 092,4...	[22 035,1...	[21 977,6...	[21 917,1...
114	[22 528,-1...	[22 482,-9...	[22 438,1...	[22 391,1...	[22 337,-3...	[22 291,-5...	[22 242,-6...	[22 189,-4...	[22 138,4...	[22 087,4...	[22 028,1...	[21 973,6...	[21 912,1...
115	[22 533,-1...	[22 489,-9...	[22 444,1...	[22 395,1...	[22 34,-3...	[22 296,-5...	[22 25,-6...	[22 199,-4...	[22 146,4...	[22 089,4...	[22 035,1...	[21 982,3...	[21 923,1...
116	[22 524,-1...	[22 48,-9...	[22 437,1...	[22 385,1...	[22 337,-3...	[22 29,-5...	[22 242,-6...	[22 187,-4...	[22 134,4...	[22 079,4...	[22 028,1...	[21 97,6.5...	[21 916,1...
117	[22 532,-1...	[22 487,-9...	[22 437,1...	[22 385,1...	[22 337,-3...	[22 29,-5...	[22 242,-6...	[22 187,-4...	[22 134,4...	[22 079,4...	[22 028,1...	[21 97,6.5...	[21 916,1...

Tablo-6.3. İkinci adımda hesaplanan değerler

	205	206	207	208	209	210	211	212	213	214	215	216	217
88	[22 375,1...	[22 331,1...	[22 284,1...	[22 236,1...	[22 185,-5...	[22 128,-5...	[22 074,-5...	[22 02,-5...	[21 968,-1...	[21 913,-1...	[21 853,-1...	[21 792,1...	[21 69,-1...
89	[22 386,1...	[22 34,1.1...	[22 291,1...	[22 242,1...	[22 19,-5...	[22 135,-5...	[22 077,-5...	[22 024,-5...	[21 972,-1...	[21 914,-1...	[21 858,-1...	[21 793,1...	[21 351,-2...
90	[22 404,1...	[22 357,1...	[22 312,1...	[22 261,1...	[22 209,-5...	[22 155,-5...	[22 099,-5...	[22 046,-5...	[21 99,-1...	[21 937,-1...	[21 879,-1...	[21 824,-5...	[21 767,-6...
91	[22 406,1...	[22 364,1...	[22 315,1...	[22 263,1...	[22 214,-5...	[22 159,-5...	[22 105,-5...	[22 052,-5...	[21 998,-1...	[21 938,-1...	[21 884,-1...	[21 826,-3...	[21 771,-3...
92	[22 423,1...	[22 379,1...	[22 331,1...	[22 281,1...	[22 233,-5...	[22 182,-5...	[22 126,-5...	[22 069,-5...	[22 013,-1...	[21 961,-1...	[21 905,-1...	[21 849,-4...	[21 789,-5...
93	[22 428,1...	[22 378,1...	[22 334,1...	[22 287,1...	[22 239,-5...	[22 187,-5...	[22 133,-5...	[22 074,-5...	[22 019,-1...	[21 961,-1...	[21 907,-1...	[21 853,-2...	[21 791,-3...
94	[22 446,1...	[22 398,1...	[22 35,1.2...	[22 3,1.22...	[22 254,-5...	[22 206,-5...	[22 153,-5...	[22 096,-5...	[22 037,-1...	[21 981,-1...	[21 924,-1...	[21 869,-4...	[21 811,-5...
95	[22 45,1.3...	[22 399,1...	[22 351,1...	[22 305,1...	[22 255,-5...	[22 204,-5...	[22 157,-5...	[22 096,-5...	[22 04,-1...	[21 981,-1...	[21 929,-1...	[21 871,-1...	[21 814,-2...
96	[22 461,1...	[22 415,1...	[22 364,1...	[22 317,1...	[22 264,-5...	[22 219,-5...	[22 171,-5...	[22 114,-5...	[22 054,-1...	[21 996,-1...	[21 944,-1...	[21 885,-3...	[21 828,-4...
97	[22 463,1...	[22 411,1...	[22 365,1...	[22 313,1...	[22 268,-5...	[22 221,-5...	[22 167,-5...	[22 117,-5...	[22 057,-1...	[22 002,-1...	[21 94,-1...	[21 887,-3...	[21 828,-1...
98	[22 471,1...	[22 424,1...	[22 374,1...	[22 329,1...	[22 281,-4...	[22 233,-5...	[22 181,-4...	[22 129,-4...	[22 073,-1...	[22 016,-1...	[21 961,-1...	[21 904,-2...	[21 849,-3...
99	[22 468,1...	[22 424,9...	[22 375,1...	[22 327,1...	[22 281,-4...	[22 229,-4...	[22 178,-4...	[22 127,-3...	[22 072,-1...	[22 017,-9...	[21 96,-1...	[21 884,0...	[21 839,0...
100	[22 485,9...	[22 439,7...	[22 39,8.2...	[22 347,9...	[22 295,-2...	[22 241,-3...	[22 19,-2...	[22 137,-2...	[22 084,-3...	[22 033,-2...	[21 975,-3...	[]	
101	[22 491,1...	[22 443,-1...	[22 399,-2...	[22 349,-1...	[22 303,4...	[22 249,4...	[22 191,4...	[22 126,4...	[22 048,4...	[21 993,3...	[21 936,2...	[21 88,2.7...	[21 827,2...
102	[22 533,5...	[22 494,5...	[22 45,5.2...	[22 404,5...	[22 357,4...	[22 307,4...	[22 256,4...	[22 205,4...	[22 103,4...	[22 049,3...	[21 991,2...	[21 934,2...	[21 875,2...
103	[22 502,5...	[22 455,5...	[22 406,5...	[22 358,5...	[22 31,4.9...	[22 257,4...	[22 204,4...	[22 155,4...	[22 1,4.81...	[22 044,3...	[21 99,3.2...	[21 934,3...	[21 874,2...
104	[22 509,5...	[22 466,5...	[22 416,5...	[22 37,5.2...	[22 319,4...	[22 271,4...	[22 217,3...	[22 166,3...	[22 11,4.6...	[22 055,3...	[22 005,3...	[21 949,2...	[21 89,2.5...
105	[22 503,5...	[22 462,5...	[22 415,5...	[22 368,5...	[22 318,4...	[22 267,4...	[22 216,4...	[22 167,4...	[22 112,4...	[22 056,3...	[22 001,3...	[21 943,3...	[21 887,2...
106	[22 516,5...	[22 473,5...	[22 428,5...	[22 378,5...	[22 326,4...	[22 278,4...	[22 227,3...	[22 178,3...	[22 125,4...	[22 069,3...	[22 014,3...	[21 955,2...	[21 898,2...
107	[22 516,5...	[22 472,5...	[22 422,5...	[22 375,5...	[22 326,4...	[22 276,4...	[22 226,4...	[22 172,4...	[22 117,4...	[22 06,3.9...	[22 01,3.2...	[21 953,3...	[21 897,2...
108	[22 523,5...	[22 479,5...	[22 43,5.0...	[22 382,5...	[22 338,4...	[22 286,4...	[22 238,4...	[22 183,3...	[22 129,4...	[22 074,3...	[22 021,3...	[21 966,2...	[21 908,2...
109	[22 522,5...	[22 476,5...	[22 428,5...	[22 379,5...	[22 335,5...	[22 282,4...	[22 235,4...	[22 182,4...	[22 127,4...	[22 07,3.9...	[22 018,3...	[21 962,3...	[21 899,2...
110	[22 531,5...	[22 485,5...	[22 435,5...	[22 387,5...	[22 343,4...	[22 294,4...	[22 244,4...	[22 193,3...	[22 139,4...	[22 083,3...	[22 03,3.0...	[21 97,2.8...	[21 912,2...
111	[22 526,5...	[22 48,5.5...	[22 431,5...	[22 382,5...	[22 337,4...	[22 288,4...	[22 238,4...	[22 186,4...	[22 133,4...	[22 075,4...	[22 02,3.2...	[21 964,3...	[21 907,2...
112	[22 534,5...	[22 488,5...	[22 441,5...	[22 393,5...	[22 345,4...	[22 297,4...	[22 247,4...	[22 194,3...	[22 138,4...	[22 084,3...	[22 031,3...	[21 975,2...	[21 919,2...
113	[22 528,5...	[22 485,5...	[22 438,5...	[22 387,5...	[22 338,4...	[22 289,4...	[22 242,4...	[22 19,4.1...	[22 139,4...	[22 085,4...	[22 036,3...	[21 969,3...	[21 909,2...
114	[22 534,5...	[22 488,5...	[22 444,5...	[22 398,5...	[22 344,4...	[22 297,4...	[22 249,4...	[22 197,3...	[22 145,4...	[22 094,3...	[22 036,3...	[21 981,2...	[21 92,2.6...
115	[22 527,5...	[22 483,5...	[22 437,5...	[22 387,5...	[22 333,4...	[22 29,4.5...	[22 243,4...	[22 192,4...	[22 138,4...	[22 081,4...	[22 027,3...	[21 974,3...	[21 915,2...
116	[22 53,5.6...	[22 486,5...	[22 442,5...	[22 392,5...	[22 343,4...	[22 296,4...	[22 249,4...	[22 194,3...	[22 14,4.6...	[22 086,3...	[22 036,3...	[21 977,2...	[21 924,2...
117	[22 537,5...	[22 49,5.5...	[22 43,5.2...	[22 385,5...	[22 336,5...	[22 288,4...	[22 237,4...	[22 184,4...	[22 13,4.8...	[22 078,4...	[22 02,3...	[21 965,3...	[21 913,2...

Tablo-6.4. İki adımın ortalaması

	205	206	207	208	209	210	211	212	213	214	215	216	217
88	[22.371,6...	[22.327,5...	[22.281,7...	[22.232,1...	[22.18,-2...	[22.123,-3...	[22.069,-3...	[22.016,-2...	[21.963,5...	[21.908,5...	[21.848,-8...	[21.787,1...	[21.698,9...
89	[22.388,6...	[22.343,5...	[22.293,1...	[22.245,1...	[22.193,-2...	[22.138,-3...	[22.08,-3...	[22.026,-2...	[21.975,5...	[21.918,5...	[21.862,-8...	[21.781,1...	[21.362,-2...
90	[22.4,6.68...	[22.353,5...	[22.308,1...	[22.257,1...	[22.204,-2...	[22.15,-3...	[22.094,-3...	[22.041,-2...	[21.985,5...	[21.933,5...	[21.874,-8...	[21.818,4...	[21.745,-7...
91	[22.408,6...	[22.366,5...	[22.318,1...	[22.265,1...	[22.217,-2...	[22.161,-3...	[22.107,-3...	[22.055,-2...	[22.001,5...	[21.941,5...	[21.887,-8...	[21.828,4...	[21.773,5...
92	[22.419,6...	[22.375,5...	[22.327,1...	[22.276,1...	[22.229,-2...	[22.177,-3...	[22.121,-3...	[22.064,-2...	[22.008,5...	[21.957,5...	[21.9,-8.8...	[21.844,5...	[21.784,9...
93	[22.43,6.6...	[22.381,5...	[22.337,1...	[22.29,1.7...	[22.242,-2...	[22.19,-3...	[22.136,-3...	[22.077,-2...	[22.022,5...	[21.964,5...	[21.91,-8...	[21.856,6...	[21.795,1...
94	[22.442,6...	[22.394,5...	[22.346,1...	[22.296,1...	[22.25,-2...	[22.202,-3...	[22.149,-3...	[22.091,-2...	[22.032,5...	[21.976,5...	[21.919,-8...	[21.864,6...	[21.806,1...
95	[22.452,6...	[22.401,5...	[22.354,1...	[22.308,1...	[22.258,-2...	[22.207,-3...	[22.16,-3...	[22.099,-2...	[22.043,5...	[21.984,5...	[21.932,-8...	[21.874,6...	[21.818,2...
96	[22.457,6...	[22.411,5...	[22.36,1.7...	[22.314,1...	[22.26,-2...	[22.215,-3...	[22.167,-3...	[22.109,-2...	[22.049,6...	[21.991,5...	[21.94,-8...	[21.88,7.0...	[21.824,2...
97	[22.466,6...	[22.414,5...	[22.368,1...	[22.316,1...	[22.271,-2...	[22.224,-3...	[22.17,-3...	[22.12,-2...	[22.06,6.3...	[22.005,6...	[21.944,-8...	[21.89,7.7...	[21.831,2...
98	[22.468,6...	[22.42,5.3...	[22.37,1.6...	[22.325,1...	[22.276,-2...	[22.229,-3...	[22.176,-3...	[22.124,-2...	[22.069,6...	[22.012,6...	[21.956,-8...	[21.9,8.22...	[21.845,3...
99	[22.471,5...	[22.427,4...	[22.378,1...	[22.33,1.6...	[22.285,-2...	[22.234,-2...	[22.182,-2...	[22.131,-2...	[22.076,8...	[22.021,7...	[21.964,-6...	[21.888,7...	[21.847,4...
100	[22.481,5...	[22.435,4...	[22.386,1...	[22.344,1...	[22.29,-7...	[22.236,-1...	[22.186,-1...	[22.133,-6...	[22.08,1.6...	[22.029,1...	[21.973,-1...	[]	
101	[22.492,6...	[22.444,-2...	[22.402,1...	[22.351,1...	[22.315,1...	[22.267,-4...	[22.215,-7...	[22.168,3...	[22.087,1...	[22.036,1...	[21.979,3...	[21.929,1...	[21.875,1...
102	[22.488,6...	[22.447,5...	[22.404,1...	[22.361,1...	[22.323,4...	[22.276,1...	[22.226,-1...	[22.171,4...	[22.042,2...	[21.971,-1...	[21.951,-3...	[21.945,-3...	[21.867,-2...
103	[22.504,9...	[22.446,-1...	[22.409,7...	[22.359,7...	[22.312,-1...	[22.258,-5...	[22.205,-8...	[22.156,-4...	[22.104,2...	[22.047,1...	[21.993,5...	[21.938,9...	[21.879,5...
104	[22.504,-3...	[22.463,-3...	[22.411,7...	[22.366,6...	[22.315,-1...	[22.267,-2...	[22.214,-2...	[22.162,-2...	[22.107,2...	[22.052,2...	[22.001,7...	[21.945,2...	[21.885,6...
105	[22.505,-4...	[22.465,-3...	[22.418,6...	[22.368,6...	[22.32,-1...	[22.269,-2...	[22.218,-2...	[22.169,-2...	[22.115,2...	[22.059,2...	[22.005,7...	[21.947,2...	[21.891,7...
106	[22.512,-4...	[22.47,-4...	[22.424,5...	[22.374,5...	[22.322,-1...	[22.274,-2...	[22.223,-2...	[22.174,-2...	[22.121,2...	[22.065,2...	[22.01,7.8...	[21.951,3...	[21.893,7...
107	[22.519,-4...	[22.475,-4...	[22.425,5...	[22.378,5...	[22.329,-1...	[22.279,-2...	[22.229,-3...	[22.176,-2...	[22.12,2.2...	[22.064,2...	[22.014,7...	[21.956,3...	[21.901,7...
108	[22.519,-5...	[22.476,-4...	[22.426,5...	[22.378,5...	[22.335,-1...	[22.282,-2...	[22.234,-3...	[22.179,-2...	[22.125,2...	[22.07,2.1...	[22.017,7...	[21.962,3...	[21.904,7...
109	[22.525,-5...	[22.479,-4...	[22.431,5...	[22.382,5...	[22.338,-1...	[22.285,-2...	[22.238,-3...	[22.185,-2...	[22.131,2...	[22.073,2...	[22.021,7...	[21.966,3...	[21.903,7...
110	[22.528,-5...	[22.482,-4...	[22.431,5...	[22.384,5...	[22.34,-1...	[22.291,-2...	[22.24,-3...	[22.189,-2...	[22.135,2...	[22.079,2...	[22.026,7...	[21.966,3...	[21.908,7...
111	[22.529,-5...	[22.483,-4...	[22.434,5...	[22.385,5...	[22.34,-1...	[22.291,-2...	[22.241,-3...	[22.19,-2...	[22.137,2...	[22.079,2...	[22.024,7...	[21.968,3...	[21.911,7...
112	[22.531,-5...	[22.484,-4...	[22.437,5...	[22.39,5.1...	[22.342,-1...	[22.294,-2...	[22.243,-3...	[22.19,-2...	[22.134,2...	[22.08,2.1...	[22.027,7...	[21.971,3...	[21.915,7...
113	[22.531,-5...	[22.488,-4...	[22.442,5...	[22.39,5.1...	[22.341,-1...	[22.292,-2...	[22.245,-3...	[22.194,-2...	[22.142,2...	[22.089,2...	[22.032,7...	[21.973,3...	[21.913,7...
114	[22.531,-5...	[22.485,-4...	[22.441,5...	[22.394,5...	[22.34,-1...	[22.294,-2...	[22.246,-3...	[22.193,-2...	[22.141,2...	[22.09,2.1...	[22.032,7...	[21.977,3...	[21.916,7...
115	[22.53,-5...	[22.486,-4...	[22.441,5...	[22.391,5...	[22.337,-1...	[22.293,-2...	[22.247,-3...	[22.196,-2...	[22.142,2...	[22.085,2...	[22.031,7...	[21.978,3...	[21.919,7...
116	[22.527,-5...	[22.483,-4...	[22.439,5...	[22.388,5...	[22.34,-1...	[22.293,-2...	[22.246,-3...	[22.19,-2...	[22.137,2...	[22.082,2...	[22.031,7...	[21.973,3...	[21.92,7.6...
117	[22.53,-5...	[22.484,-4...	[22.439,5...	[22.39,5.1...	[22.34,-1...	[22.293,-2...	[22.246,-3...	[22.19,-2...	[22.137,2...	[22.082,2...	[22.031,7...	[21.973,3...	[21.92,7.6...

Tablo-6.5. Gerçek ölçüler

	205	206	207	208	209	210	211	212	213	214	215	216	217
88	22.371	22.327	22.28	22.232	22.18	22.123	22.069	22.016	21.963	21.908	21.848	21.789	21.7
89	22.388	22.343	22.293	22.245	22.193	22.138	22.08	22.026	21.975	21.917	21.864	21.794	21.3
90	22.4	22.353	22.308	22.257	22.204	22.15	22.094	22.041	21.985	21.933	21.874	21.819	21.7
91	22.408	22.367	22.318	22.265	22.217	22.161	22.107	22.055	22.002	21.941	21.887	21.828	21.7
92	22.419	22.375	22.327	22.276	22.229	22.177	22.121	22.064	22.008	21.957	21.9	21.844	21.7
93	22.43	22.38	22.337	22.29	22.242	22.19	22.136	22.077	22.022	21.964	21.91	21.857	21.7
94	22.442	22.394	22.346	22.296	22.25	22.202	22.149	22.091	22.032	21.976	21.919	21.864	21.8
95	22.453	22.401	22.354	22.308	22.258	22.207	22.161	22.099	22.043	21.984	21.932	21.874	21.8
96	22.457	22.412	22.36	22.314	22.259	22.215	22.168	22.109	22.049	21.991	21.94	21.88	21.8
97	22.467	22.414	22.368	22.315	22.271	22.224	22.17	22.121	22.06	22.006	21.943	21.891	21.8
98	22.468	22.42	22.37	22.325	22.277	22.23	22.177	22.125	22.069	22.012	21.956	21.901	21.8
99	22.47	22.427	22.378	22.329	22.285	22.233	22.182	22.131	22.076	22.021	21.965	21.908	21.8
100	22.48	22.434	22.385	22.342	22.293	22.238	22.188	22.135	22.083	22.032	21.974	NaN	Na
101	22.491	22.443	22.399	22.349	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
102	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	22.098	22.045	21.987	21.929	21.8
103	22.505	22.458	22.409	22.361	22.314	22.26	22.207	22.158	22.104	22.047	21.993	21.938	21.8
104	22.506	22.463	22.412	22.367	22.315	22.267	22.213	22.162	22.106	22.051	22.001	21.945	21.8
105	22.505	22.465	22.418	22.368	22.321	22.27	22.219	22.17	22.115	22.059	22.005	21.947	21.8
106	22.512	22.47	22.425	22.374	22.322	22.274	22.223	22.174	22.121	22.065	22.01	21.951	21.8
107	22.519	22.475	22.425	22.378	22.329	22.279	22.229	22.176	22.12	22.063	22.014	21.956	21.9
108	22.519	22.476	22.426	22.378	22.335	22.282	22.234	22.179	22.125	22.07	22.017	21.962	21.9
109	22.525	22.479	22.431	22.382	22.338	22.285	22.238	22.185	22.131	22.073	22.021	21.966	21.9
110	22.528	22.482	22.431	22.384	22.34	22.291	22.24	22.189	22.135	22.079	22.026	21.966	21.9
111	22.529	22.483	22.434	22.385	22.34	22.291	22.241	22.19	22.137	22.079	22.024	21.968	21.9
112	22.531	22.484	22.437	22.39	22.342	22.294	22.243	22.19	22.133	22.08	22.027	21.971	21.9
113	22.531	22.489	22.442	22.39	22.341	22.292	22.245	22.194	22.143	22.089	22.032	21.973	21.9
114	22.531	22.485	22.441	22.395	22.34	22.294	22.246	22.193	22.141	22.091	22.032	21.977	21.9
115	22.53	22.486	22.441	22.391	22.336	22.293	22.247	22.196	22.142	22.085	22.031	21.978	21.9
116	22.527	22.483	22.44	22.388	22.34	22.293	22.246	22.19	22.137	22.082	22.032	21.973	21.9
117	22.53	22.484	22.439	22.39	22.341	22.293	22.246	22.19	22.137	22.082	22.032	21.973	21.9

Tablo-6.1'den Tablo-6.5'e kadar, bir takım girdi ve çıktı değerleri görülmektedir. Bu değerlerin anlamı şudur: Mesela (102,205) indisli değer için Tablo-6.1'e baktığımızda 0, yani taranmış bölgedeki boşluk olduğu anlaşılır. Tablo-6.2'de bu boşluk için hesaplanan ilk değer görülür, o da 22.447'dir. Tablo-6.3'de aynı boşluk için, ikinci adımda hesaplanan değeri görülür (22.533). Tablo-6.4'de ise bu iki değer in hesaplanmış ortalaması görülmektedir (22.488). Filtre, sadece boşluklar için değil, ölçülen noktalar için de kestirim değeri hesaplamaktadır. Mesela (101,205) indisli hücre için, sırasıyla, 1 yani nokta var, 22.493 ilk adım kestirim değeri, 22.491 ikinci adım kestirim değeri, 22.492 ortalama değer ve 22.491 ölçülen değerdir. Filtrenin bu özelliği kullanılarak ileride uyuşumsuz ölçü testi yapmak da mümkündür.

7. SONUÇ ve ÖNERİLER

Bu tezde, günümüzde birçok uygulamada kullanılmaya başlanan yersel lazer tarama verisinin, iki boyutlu Kalman filtresi kullanılarak iyileştirilmesi amaçlanmıştır. Tarama anında, asıl taranan cismin önünden geçen taşıt ve insanların veya asıl taranan cismin önünde duran diğer cisimlerin oluşturduğu boşlukların iki boyutlu Kalman filtresi ile nasıl doldurulabileceği araştırılmıştır.

Bu amaç doğrultusunda Matlab ortamında kodlanan filtre, gürültüsüz ve gürültülü farklı türdeki boşluklara uygulanmıştır. İki boyutlu Kalman filtresinin ürettiği sonuçları daha iyi anlayabilmek için aynı verilere bir de lineer enterpolasyon yöntemi uygulanarak, sonuçlar karşılaştırılmıştır.

Sonuç olarak, bu tezde kullanılan iki boyutlu Kalman filtresi, küre yüzeyi üzerinde bulunan boşlukların doldurulmasında başarılıdır. Elde edilen sonuçlar incelendiğinde, filtre gerçek gücünü gürültü veriler üzerinde göstermektedir. Lineer enterpolasyon yöntemi gürültüden olumsuz yönde etkilenirken, iki boyutlu Kalman filtresi bu testlerden başarıyla geçmiştir. Dolayısıyla özellikle gürültülü veri setleri için, lineer enterpolasyon yöntemine alternatif olabilecek niteliktedir.

İlerideki çalışmalarda, soldan sağa-yukarıdan aşağıya ve sağdan sola-aşağıdan yukarıya iki adımlı uygulanan filtreye, aşağıdan yukarıya-sağdan sola ve yukarıdan aşağıya-soldan sağa işleme adımları eklenerek filtre dört adımlı hale getirilip, sonuçlar karşılaştırılabilir. Filtreye uyumsuz ölçü testi eklenerek, verinin kaba hatalardan arındırılması sağlanabilir. Filtre, farklı özellikteki yüzeyler üzerinde denenerek, ürettiği sonuçlar araştırılabilir.

KAYNAKLAR

Lichti D. D. and Gordon S. J., Error Propagation in Directly Georeferenced Terrestrial Laser Scanner Point Clouds for Cultural Heritage Recording, FIG Working Week 2004, Athens, Greece, May 22-27, 2004

Fröhlich C. and Mettenleiter M., 2004. Terrestrial Laser Scanning – New Perspectives in 3D Surveying, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVI-8/W2, Freiburg, Germany, pp. 7-13.

Fang G. et. al., 2001. Fast and Accurate Close Range 3d Modelling By Laser Scanning System, <http://cipa.icomos.org/fileadmin/papers/potsdam/2001-21-gf01.pdf?PHPSESSID=b46638220d3b64083f011b1e8dbfc514> adresinden 05.03.2007 tarihinde alınmıştır.

Wehr A., Laser Scanning and its Potential to Support 3D Panoramic Recording, http://www2.informatik.hu-berlin.de/sv/pr/PanoramicPhotogrammetryWorkshop2005/Paper/PanoWS_Berlin2005_Wehr.pdf adresinden 05.03.2007 tarihinde alınmıştır.

Ek-1, Leica HDS 3000 Teknik özellikleri, http://www.leica-geosystems.com/hds/en/Leica_HDS3000.pdf adresinden 05.03.2007 tarihinde alınmıştır.

Kalman R. E., 1960. A New Approach to Linear Filtering and Prediction Problems. Transaction of the ASME. Journal of Basic Engineering, 82 (D series): pp. 35-45.

Welch G. and Bishop G., 2001. An Introduction to the Kalman Filter, http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf adresinden 05.03.2007 tarihinde alınmıştır.

[URL1] Kalman Filter, http://en.wikipedia.org/wiki/Kalman_filter adresinden 05.03.2007 tarihinde alınmıştır.

Maybeck P. S., 1979. Stochastic Models, Estimation and Control, http://www.cs.unc.edu/~welch/media/pdf/maybeck_ch1.pdf adresinden 05.03.2007 tarihinde alınmıştır.

Wang P., 1998. Applying two Dimensional Kalman Filtering for Digital Terrain Modeling. <http://www.ifp.uni-stuttgart.de/publications/commIV/wang71.pdf> adresinden 05.03.2007 tarihinde alınmıştır.

ÖZGEÇMİŞ

Kişisel Bilgiler:

Doğum Tarihi : 29.01.1981

Doğum Yeri : Tavşanlı

Öğrenim:

2004- Y. Lisans : GYTE Mühendislik ve Fen Bilimleri Enstitüsü
Jeodezi ve Fotogrametri Müh.

1999-2004 Lisans : İTÜ İnşaat Fak. Jeodezi ve Fotogrametri Müh.

1996-1999 Lise : Eskişehir Yunusemre Anadolu Öğretmen Lisesi

Çalıştığı Kurumlar:

2005- :BİMTAŞ İMP Lazer Tarama Grubu

EKLER

EK-1: Leica HDS3000 Teknik Özellikleri

EK-2: Matlab Uygulama Kodları

EK-3: Matlab Küre örnekleme kodu

EK-4: Matlab Lineer Enterpolasyon Kodu

EK-5: Matlab Gürültü Ekleme Kodu

Ek-1: Leica HDS3000 Teknik Özellikleri

Leica HDS3000 Product Specifications

GENERAL	
INSTRUMENT TYPE	High-speed, high-accuracy laser radar scanner with 360° x 270° field-of-view
USER INTERFACE	Notebook PC
SCANNER DRIVE	Servo motor
OPTICAL VIEWER	Integrated video camera
SYSTEM PERFORMANCE	
SINGLE POINT ACCURACY*	
Position	6mm
Distance	4mm
Angle (horizontal)	60 micro-radians
Angle (vertical)	60 micro-radians
MODELED SURFACE PRECISION**	2mm
TARGET ACQUISITION ACCURACY†	1.5mm
DATA INTEGRITY MONITORING	Periodic accuracy self-checking during operation and at startup
* All specifications shown @ 1m - 50m range	
** Subject to modeling methodology	
† Algorithmic fit to planar HDS targets	
LASER SCANNING SYSTEM	
TYPE	Pulsed; proprietary microchip
COLOR	Green
LASER CLASS	Class 3R (IEC 60825-1)
RANGE	
Optimal effective range	1m-100m
To 10% reflectivity targets	Up to 100m (typical)
SCAN RATE	Up to 1800 points/second*
* Maximum scan rate dependent on scan resolution and selected field-of-view	
SCAN DENSITY (RESOLUTION)	
Spot size	≤ 6mm from 0-50 meters
Selectability	Independently selectable vertical and horizontal point-to-point measurement spacing †
Point Spacing	Select by total number of points per unit area or minimum distance between points (at specified range) †
Maximum Sample Density	1.2mm †
Scan row (horizontal)	20,000 points/row, maximum †
Scan column (vertical)	5,000 points/column, maximum †
FIELD-OF-VIEW (PER SCAN)	
Horizontal	360° (maximum) †
Vertical	270° (maximum) †
Aiming/Sighting	Optical sighting using QuickScan™ button
SCANNING OPTICS	Single mirror, panoramic, dual-window design † Environmentally protected by housing and two glass shields
SCAN MOTORS	Direct drive, brushless
DATA & POWER TRANSFER TO/ FROM TURRET	Contact-free: optical data link and inductive power transfer
COMMUNICATIONS	Static Internet Protocol (IP) Address
COLOR DIGITAL IMAGING	User defined pixel resolution: Low, Medium, High† Single 24" x 24" image: 1024 x 1024 pixels (1 megapixel) @ "High" setting Full 360° x 270° dome: 111 images, approx. 64 megapixels, spatially rectified
STATUS INDICATORS	3 LED's (on base) indicate system ready, laser on, and comm. status

ELECTRICAL	
POWER SUPPLY	12V input, QTY (2) Power Supply units provided with system
POWER CONSUMPTION	< 80W average
BATTERY TYPE	Sealed lead acid
POWER PORTS	2, simultaneous use, "hot swap" capable
TYPICAL DURATION	Up to 6 hours continuous use (nominal temp.)
POWER STATUS INDICATORS	5 LEDs indicate charging status and power levels (low/medium/high)
ENVIRONMENTAL	
OPERATING TEMP.	0°C to 40°C
STORAGE TEMP.	-25°C to 65°C
LIGHTING	Fully operational between bright sunlight and complete darkness
HUMIDITY	Non-condensing atmosphere
SHOCK	40G's (max. to scanner transport case)
DUST/HUMIDITY	IEC Specification IP52

PHYSICAL		
	DIMENSIONS	WEIGHT
SCANNER	10.5" D" x 14.5" W x 20" H 265mm D x 370mm W x 510mm H w/o handles, w/o table stand	16 kg (35lbs), nominal
POWER SUPPLY UNIT	8.5" D x 9.25" W x 8.5" H 165mm D x 236mm W x 215mm H w/o handles	12 kg (26lbs), nominal

STANDARD ACCESSORIES	
Scanner transport case	
Tribach (Leica Professional Series)	
Surveying tripod	
Ethernet cable for connection of scanner to notebook PC	
Two Power Supply cases. Each includes:	
Power Supply	
Cable for battery connection to scanner	
Power Supply charger	
Cyclone™-SCAN software	
HARDWARE OPTIONS	
Notebook PC (Standard or Enhanced)	
HDS scan targets and target accessories	
Service agreement for HDS3000	

NOTEBOOK PC FOR SCANNING Δ	
COMPONENT	REQUIRED (minimum)
Processor	1.4 GHz Pentium M or similar
RAM	512MB SDRAM
Network card	Ethernet
Display	SXGA +
Operating system	Windows XP (Professional or Home Edition, SP1 or higher) Windows 2000 (SP2 or higher)

Δ Minimum requirements for modeling operations are different.
Please refer to Cyclone datasheet for specifications.

CYCLONE – SCAN	
"Fly-around," pan & zoom, and freely rotate point clouds, true-color intensity mapped clouds, wire meshes, "shrinkwrap" surfaces, and models in 3D	
Real-time 3D visualization while scanning †	
Point cloud and 3D model Level of Detail (LOD) for fast visualization	
Fast "shrinkwrap" rendering of point clouds to meshes	
Decimation of point clouds (nth point)	
View point clouds with intensity or true-color mapping	
Limit box for efficient viewing and interaction of selected regions	
Targeted, single-shot pre-scan ranging †	

CYCLONE – SCAN (CONT.)	
Automatic creation of panoramic digital image mosaic †	
Panoramic digital image viewer †	
Georeferencing over known or assumed survey point †	
Instrument height (H.I.) input during data capture †	
Target height input during data capture †	
Point-and-scan QuickScan™ feature to interactively set horizontal field-of-view †	
Scan filtering to optionally exclude data based on:	
Area of interest via rectangular or free form polygonal areas †	
Range †	
Return intensity †	
Pre-set drop-down list or custom settings †	
User-defined quality-of-fit checks	
Atmospheric correction	
Measure & dimension point clouds and models	
Slope distances	
ΔX, ΔY, ΔZ distances	
Create and manage annotations	
Create and manage layers	
Assign colors & materials to objects	
View scanner locations and field-of-view	
Environmental lighting	
Save/restore views	
Save screen image as image file	
Undo/redo support	
Automated acquisition of HDS targets †	
Scanner command scripting †	

DIRECT IMPORT FORMATS	
ASCII point data (XYZ, SVY, PTS, PTX, TXT)	
RIEGL 3DD	
Cyclone Object Exchange (COE) format (COE Data Transfer Products)	
CGP	
BMP, JPEG, TIFF	
DIRECT EXPORT FORMATS	
ASCII point data (XYZ, SVY, PTS, PTX, TXT)	
BMP, JPEG, TIFF	
Cyclone Object Exchange (COE) format (COE Data Transfer Products)	

INDIRECT EXPORT FORMATS	
AutoCAD (via COE for AutoCAD plug-in)	
MicroStation (via COE for MicroStation plug-in)	
PDS (via MicroStation, COE for MicroStation plug-in)	
AutoPLANT (via AutoCAD, COE for AutoCAD plug-in)	
ORDERING INFORMATION	
Contact Leica Geosystems HDS Division (Cyra Technologies, Inc.) or authorized manufacturer's representatives	

All specifications are subject to change without notice.

All ± accuracy specifications at 1 sigma unless indicated otherwise

† SmartScan™ Technology feature



Leica
Geosystems

Ek-2: Matlab Uygulama Kodları

```
% Emre Pınarcı İKİ BOYUTLU KALMAN FİLTRESİNİN LAZER TARAMASINA
UYGUNLAMASI
% 2007
disp('Script Başladı...');
M_in = dlmread('kucukkubbe4cm.txt'); %noktabulututu dosyası
M(satırsayısı x 3)ye atanıyor
[row_M_in, column_M_in] = size(M_in); % M_in matrisinin satır ve
sütun sayısını al
sorted = sort(M_in,1); % M_in matrisinin kolonlarını artan
sıralamada sırala
disp('Dosya Okundu... Nokta sayısı:'); disp(row_M_in);

x_min = sorted(1,1); % minimum x değeri
y_min = sorted(1,2); % minimum y değeri
x_max = sorted(row_M_in,1); % maximum x değeri
y_max = sorted(row_M_in,2); % maximum y değeri

step_along_x = 0.04; % x eksenini boyunca nokta aralığı
step_along_y = 0.04; % y eksenini boyunca nokta aralığı

row_M_out = int32((y_max - y_min)/step_along_y) + 1; %oluşacak
matrisin satır sayısı
column_M_out = int32((x_max - x_min)/step_along_x) + 1; %oluşacak
matrisin sütun sayısı
disp('Noktaların Oluşturduğu Matrisin'); disp('Satır Sayısı:');
disp(row_M_out); disp('Sütun Sayısı:');disp(column_M_out);

X = NaN(row_M_out, column_M_out); % X değerlerinin atanacağı matris
Y = NaN(row_M_out, column_M_out); % Y değerlerinin atanacağı matris
Z = NaN(row_M_out, column_M_out); % Z değerlerinin atanacağı matris
Point = zeros(row_M_out, column_M_out); % Point değerlerinin
atanacağı logical matris 0-1

for i=1:row_M_in % sırasıyla x,y,z değerleri X,Y,Z matrislerine atar
    strarraycolumn = int32((M_in(i,1) - x_min) / step_along_x) +
1);
    strarrayrow = int32((M_in(i,2) - y_min) / step_along_y) + 1);
    X(strarrayrow,strarraycolumn) = M_in(i,1);
    Y(strarrayrow,strarraycolumn) = M_in(i,2);
    Z(strarrayrow,strarraycolumn) = M_in(i,3);
    Point(strarrayrow,strarraycolumn) = 1;
end

for i=1:row_M_out % Point matrisi içerisinde veri sınırının dışında
kalan elemanlara nan atar
    for j=1:column_M_out
        if (Point(i,j)==1)
            break;
        else
            Point(i,j)=NaN;
        end
    end
    for j=column_M_out:-1:1
        if (Point(i,j)==1)
            break;
        else
```

```

        Point(i,j)=NaN;
    end
end
end

S_minus = cell(row_M_out, column_M_out); % S_minus değerlerinin
atanacağı cell array 3x1
P_minus = cell(row_M_out, column_M_out); % P_minus değerlerinin
atanacağı cell array 3x3
K = cell(row_M_out, column_M_out); % K değerlerinin atanacağı cell
array 3x1
S_plus = cell(row_M_out, column_M_out); % S_plus değerlerinin
atanacağı cell array 3x1
P_plus = cell(row_M_out, column_M_out); % P_plus değerlerinin
atanacağı cell array 3x3
R=0.001;
Q=eye(3); Q(1,1)=0.009; Q(2,2)=0.0002; Q(3,3)=0.0002;
D=[1 0 0];

disp('Sol Üst Köşeden işlem başladı...');
for i=1:row_M_out %half plane de sol üst köşeden başlayarak işlem
yapar
    last_processed_i=0;
    last_processed_j=0;
    if(rem(i,2)==0) %i çift ise
        for j=column_M_out:-1:1
            if(isnan(Point(i,j))) % nokta yoksa döngüye devam eder
                continue;
            elseif(Point(i,j)==0) % boşluğa geldiğinde S-=S+, P-=P+
olacak (boşluk mutlaka arada olacağından sağ=nokta yada boşluk)
                if(isnan(Point(i-1,j))) % üstte nokta yoksa nan ise
(üst=nan, sağ=nokta yada boşluk)
                    c=1; %yukarıda işlenecek nokta olmadığı için tüm
ağırlık Cye binecek c=1 b=0
                    C=zeros(3,3);
                    C(1,1)=c;
                    C(1,3)=c*step_along_x;
                    C(3,3)=1;
                    S_minus{i,j}=C*S_plus{i,j+1};
                    P_minus{i,j}=C*P_plus{i,j+1}*C'+Q;
                    K{i,j}=P_minus{i,j}*D'*inv(D*P_minus{i,j}*D'+R);
                    S_plus{i,j}=S_minus{i,j};
                    P_plus{i,j}=P_minus{i,j};
                    last_processed_i=i;
                    last_processed_j=j;
                else % üst=nokta yada boşluk, sağ nokta yada boşluk
                    b=0.5; %C ve B eşit ağırlıklı olacak
                    c=0.5;
                    B=zeros(3,3);
                    C=zeros(3,3);
                    B(1,1)=b;
                    B(1,2)=b*step_along_y;
                    B(2,2)=1;
                    C(1,1)=c;
                    C(1,3)=c*step_along_x;
                    C(3,3)=1;
                    S_minus{i,j}=B*S_plus{i-1,j}+C*S_plus{i,j+1};
                    P_minus{i,j}=B*P_plus{i-
1,j}*B'+C*P_plus{i,j+1}*C'+Q;

                    K{i,j}=(P_minus{i,j}*D'*inv(D*P_minus{i,j}*D'+R));

```

```

        S_plus{i,j}=S_minus{i,j};
        P_plus{i,j}=P_minus{i,j};
        last_processed_i=i;
        last_processed_j=j;
    end
    else % nokta varsa
        if(last_processed_i==0) % işlenen nokta satırın ilk
noktasıysa (sağı nan olacağı için)
            if(isnan(Point(i-1,j))) % üstte nokta yoksa nan
ise (üst=nan, sağ=nan)
                S_minus{i,j}=[Z(i,j);0;0];
                P_minus{i,j}=Q;

K{i,j}=(P_minus{i,j}*D'*inv(D*P_minus{i,j}*D'+R));
                S_plus{i,j}=S_minus{i,j}+K{i,j}*(Z(i,j)-
D*S_minus{i,j});

                P_plus{i,j}=(Q-K{i,j}*D)*P_minus{i,j};
                last_processed_i=i;
                last_processed_j=j;
            else % üstte boşluk veya nokta varsa (üst=nokta
yada boşluk, sağ=nan)
                b=1;
                B=zeros(3,3);
                B(1,1)=b;
                B(1,2)=b*step_along_y;
                B(2,2)=1;
                S_minus{i,j}=B*S_plus{i-1,j};
                P_minus{i,j}=B*P_plus{i-1,j}*B'+Q;

K{i,j}=P_minus{i,j}*D'*inv(D*P_minus{i,j}*D'+R);
                S_plus{i,j}=S_minus{i,j}+K{i,j}*(Z(i,j)-
D*S_minus{i,j});

                P_plus{i,j}=(Q-K{i,j}*D)*P_minus{i,j};
                last_processed_i=i;
                last_processed_j=j;
            end
            else % işlenen nokta satırın ilk noktası değilse
yani sağı nan değilse
                if(isnan(Point(i-1,j))) % üstte nokta yoksa nan
ise (üst=nan, sağ=nokta yada boşluk)
                    c=1; %yukarıda işlenecek nokta olmadığı için
tüm ağırlık Cye binecek c=1 b=0
                    C=zeros(3,3);
                    C(1,1)=c;
                    C(1,3)=c*step_along_x;
                    C(3,3)=1;
                    S_minus{i,j}=C*S_plus{i,j+1};
                    P_minus{i,j}=C*P_plus{i,j+1}*C'+Q;

K{i,j}=P_minus{i,j}*D'*inv(D*P_minus{i,j}*D'+R);
                    S_plus{i,j}=S_minus{i,j}+K{i,j}*(Z(i,j)-
D*S_minus{i,j});

                    P_plus{i,j}=(Q-K{i,j}*D)*P_minus{i,j};
                    last_processed_i=i;
                    last_processed_j=j;
                else % üstte boşluk veya nokta varsa (üst=nokta
yada boşluk, sol=nokta yada boşluk)
                    b=0.5; %C ve B eşit ağırlıklı olacak
                    c=0.5;
                    B=zeros(3,3);
                    C=zeros(3,3);

```

```

        B(1,1)=b;
        B(1,2)=b*step_along_y;
        B(2,2)=1;
        C(1,1)=c;
        C(1,3)=c*step_along_x;
        C(3,3)=1;
        S_minus{i,j}=B*S_plus{i-
1,j}+C*S_plus{i,j+1};
        P_minus{i,j}=B*P_plus{i-
1,j}*B'+C*P_plus{i,j+1}*C'+Q;
        K{i,j}=P_minus{i,j}*D'*inv(D*P_minus{i,j}*D'+R);
        S_plus{i,j}=S_minus{i,j}+K{i,j}*(Z(i,j)-
D*S_minus{i,j});
        P_plus{i,j}=(Q-K{i,j}*D)*P_minus{i,j};
        last_processed_i=i;
        last_processed_j=j;
    end
end
end
else % i çift değilse (tek ise)
    for j=1:column_M_out
        if(isnan(Point(i,j))) % nokta yoksa döngüye devam eder
            continue;
        elseif(Point(i,j)==0) % boşluğa geldiğinde S-=S+, P-=P+
olacak (boşluk mutlaka arada olacağından sol=nokta yada boşluk)
            if(isnan(Point(i-1,j))) % üstte nokta yoksa nan ise
(üst=nan, sağ=nokta yada boşluk)
                c=1; %yukarıda işlenecek nokta olmadığı için tüm
ağırlık Cye binecek c=1 b=0
                C=zeros(3,3);
                C(1,1)=c;
                C(1,3)=c*step_along_x;
                C(3,3)=1;
                S_minus{i,j}=C*S_plus{i,j-1};
                P_minus{i,j}=C*P_plus{i,j-1}*C'+Q;
                K{i,j}=P_minus{i,j}*D'*inv(D*P_minus{i,j}*D'+R);
                S_plus{i,j}=S_minus{i,j};
                P_plus{i,j}=P_minus{i,j};
                last_processed_i=i;
                last_processed_j=j;
            else % üst=nokta yada boşlu, sağ nokta yada boşluk
b=0.5; %C ve B eşit ağırlıklı olacak
c=0.5;
                B=zeros(3,3);
                C=zeros(3,3);
                B(1,1)=b;
                B(1,2)=b*step_along_y;
                B(2,2)=1;
                C(1,1)=c;
                C(1,3)=c*step_along_x;
                C(3,3)=1;
                S_minus{i,j}=B*S_plus{i-1,j}+C*S_plus{i,j-1};
                P_minus{i,j}=B*P_plus{i-1,j}*B'+C*P_plus{i,j-
1}*C'+Q;
                K{i,j}=(P_minus{i,j}*D'*inv(D*P_minus{i,j}*D'+R));
                S_plus{i,j}=S_minus{i,j};
                P_plus{i,j}=P_minus{i,j};
                last_processed_i=i;

```

```

        last_processed_j=j;
    end
    else % nokta varsa
        if(i==1)%ilk satır işleniyorsa yukarda nokta
        olmadığında sadece soldaki noktayı kullan
            if(last_processed_i==0)%iterasyonun ilk
başlangıç noktası (üst=nan, sol=nan)
                %temp_Z=Z(i,j);
                S_minus{i,j}=[Z(i,j);0;0];
                P_minus{i,j}=Q;

K{i,j}=(P_minus{i,j}*D'*inv(D*P_minus{i,j}*D'+R));
                S_plus{i,j}=S_minus{i,j}+K{i,j}*(Z(i,j)-
D*S_minus{i,j});

                P_plus{i,j}=(Q-K{i,j}*D)*P_minus{i,j};
                last_processed_i=i;
                last_processed_j=j;
            else %iterasyonun başlangıç noktası geçtikten
sonra (üst=nan, sol=nokta yada boşluk)
                c=1; %yukarıda işlenecek nokta olmadığı için
tüm ağırlık Cye binecek c=1 b=0
                C=zeros(3,3);
                C(1,1)=c;
                C(1,3)=c*step_along_x;
                C(3,3)=1;
                S_minus{i,j}=C*S_plus{i,j-1};
                P_minus{i,j}=C*P_plus{i,j-1}*C'+Q;

K{i,j}=P_minus{i,j}*D'*inv(D*P_minus{i,j}*D'+R);
                S_plus{i,j}=S_minus{i,j}+K{i,j}*(Z(i,j)-
D*S_minus{i,j});

                P_plus{i,j}=(Q-K{i,j}*D)*P_minus{i,j};
                last_processed_i=i;
                last_processed_j=j;
            end
        else % işlenen satır ilk satır değilse yani i>1 ise
            %disp('i');
            if(last_processed_i==0) % işlenen nokta satırın
ilk noktasıysa (solu nan olacağı için)
                if(isnan(Point(i-1,j))) % üstte nokta yoksa
nan ise (üst=nan, sol=nan)
                    S_minus{i,j}=[Z(i,j);0;0];
                    P_minus{i,j}=Q;

K{i,j}=(P_minus{i,j}*D'*inv(D*P_minus{i,j}*D'+R));
                    S_plus{i,j}=S_minus{i,j}+K{i,j}*(Z(i,j)-
D*S_minus{i,j});

                    P_plus{i,j}=(Q-K{i,j}*D)*P_minus{i,j};
                    last_processed_i=i;
                    last_processed_j=j;
                else % üstte boşluk veya nokta varsa
(üst=nokta yada boşluk, sol=nan)
                    b=1;
                    B=zeros(3,3);
                    B(1,1)=b;
                    B(1,2)=b*step_along_y;
                    B(2,2)=1;
                    S_minus{i,j}=B*S_plus{i-1,j};
                    P_minus{i,j}=B*P_plus{i-1,j}*B'+Q;

K{i,j}=P_minus{i,j}*D'*inv(D*P_minus{i,j}*D'+R);

```

```

        S_plus{i,j}=S_minus{i,j}+K{i,j}*(Z(i,j)-
D*S_minus{i,j});
        P_plus{i,j}=(Q-K{i,j}*D)*P_minus{i,j};
        last_processed_i=i;
        last_processed_j=j;
    end
    else % işlenen nokta satırın ilk noktası değilse
        yani solu nan değilse
            if(isnan(Point(i-1,j))) % üstte nokta yoksa
                nan ise (üst=nan, sol=nokta yada boşluk)
                c=1; %yukarıda işlenecek nokta olmadığı
                için tüm ağırlık Cye binecek c=1 b=0
                C=zeros(3,3);
                C(1,1)=c;
                C(1,3)=c*step_along_x;
                C(3,3)=1;
                S_minus{i,j}=C*S_plus{i,j-1};
                P_minus{i,j}=C*P_plus{i,j-1}*C'+Q;
            K{i,j}=P_minus{i,j}*D'*inv(D*P_minus{i,j}*D'+R);
            S_plus{i,j}=S_minus{i,j}+K{i,j}*(Z(i,j)-
D*S_minus{i,j});
            P_plus{i,j}=(Q-K{i,j}*D)*P_minus{i,j};
            last_processed_i=i;
            last_processed_j=j;
        else % üstte boşluk veya nokta varsa
            (üst=nokta yada boşluk, sol=nokta yada boşluk)
            b=0.5; %C ve B eşit ağırlıklı olacak
            c=0.5;
            B=zeros(3,3);
            C=zeros(3,3);
            B(1,1)=b;
            B(1,2)=b*step_along_y;
            B(2,2)=1;
            C(1,1)=c;
            C(1,3)=c*step_along_x;
            C(3,3)=1;
            S_minus{i,j}=B*S_plus{i-
1,j}+C*S_plus{i,j-1};
            P_minus{i,j}=B*P_plus{i-
1,j}*B'+C*P_plus{i,j-1}*C'+Q;
            K{i,j}=P_minus{i,j}*D'*inv(D*P_minus{i,j}*D'+R);
            S_plus{i,j}=S_minus{i,j}+K{i,j}*(Z(i,j)-
D*S_minus{i,j});
            P_plus{i,j}=(Q-K{i,j}*D)*P_minus{i,j};
            last_processed_i=i;
            last_processed_j=j;
        end
    end
end
end
end
end
end
disp('Sol_Üst Köşeden başlayan işlem sonlandı...');

soladogru=0;
S_minus2 = cell(row_M_out, column_M_out); % S_minus değerlerinin
atanacağı cell array 3x1

```

```

P_minus2 = cell(row_M_out, column_M_out); % P_minus değerlerinin
atanacağı cell array 3x3
K2 = cell(row_M_out, column_M_out); % K değerlerinin atanacağı cell
array 3x1
S_plus2 = cell(row_M_out, column_M_out); % S_plus değerlerinin
atanacağı cell array 3x1
P_plus2 = cell(row_M_out, column_M_out); % P_plus değerlerinin
atanacağı cell array 3x3

disp('Sağ_Alt Köşeden işlem başladı...');
for i=row_M_out:-1:1
    last_processed_i=0;
    if(i==row_M_out) % işlenen nokta ilk satırdaysa
        for j=column_M_out:-1:1
            if(isnan(Point(i,j))) % nokta yoksa döngüye devam eder
                continue;
            elseif(Point(i,j)==0) % boşluğa geldiğinde S-=S+, P-=P+
olacak (boşluk mutlaka arada olacağından sağ=nokta yada boşluk)
                c=1; %aşağıda işlenecek nokta olmadığı için tüm
ağırlık Cye binecek c=1 b=0
                C=zeros(3,3);
                C(1,1)=c;
                C(1,3)=c*step_along_x;
                C(3,3)=1;
                S_minus2{i,j}=C*S_plus2{i,j+1};
                P_minus2{i,j}=C*P_plus2{i,j+1}*C'+Q;
                K2{i,j}=P_minus2{i,j}*D'*inv(D*P_minus2{i,j}*D'+R);
                S_plus2{i,j}=S_minus2{i,j};
                P_plus2{i,j}=P_minus2{i,j};
            else % nokta varsa
                if(last_processed_i==0) % ilk satırın ilk noktası
işleniyorsa (alt=nan, sağ=nan)
                    S_minus2{i,j}=[Z(i,j);0;0];
                    P_minus2{i,j}=Q;
                K2{i,j}=(P_minus2{i,j}*D'*inv(D*P_minus2{i,j}*D'+R));
                    S_plus2{i,j}=S_minus2{i,j}+K2{i,j}*(Z(i,j)-
D*S_minus2{i,j});
                    P_plus2{i,j}=(Q-K2{i,j}*D)*P_minus2{i,j};
                    last_processed_i=i;
                    last_processed_j=j;
                else %iterasyonun başlangıç noktası geçtikten sonra
(al=nan, sağ=nokta yada boşluk)
                    c=1; %aşağıda işlenecek nokta olmadığı için tüm
ağırlık Cye binecek c=1 b=0
                    C=zeros(3,3);
                    C(1,1)=c;
                    C(1,3)=c*step_along_x;
                    C(3,3)=1;
                    S_minus2{i,j}=C*S_plus2{i,j+1};
                    P_minus2{i,j}=C*P_plus2{i,j+1}*C'+Q;
                K2{i,j}=P_minus2{i,j}*D'*inv(D*P_minus2{i,j}*D'+R);
                    S_plus2{i,j}=S_minus2{i,j}+K2{i,j}*(Z(i,j)-
D*S_minus2{i,j});
                    P_plus2{i,j}=(Q-K2{i,j}*D)*P_minus2{i,j};
                    last_processed_i=i;
                    last_processed_j=j;
            end
        end
    end
end
end

```

```

else % işlenen nokta ilk satırda değilse
    if(soladogru==0) %filtre sağa doğru hareket eder
        for j=1:column_M_out
            if(isnan(Point(i,j))) % nokta yoksa döngüye devam
eder
                continue;
            elseif(Point(i,j)==0)% boşluğa geldiğinde S-=S+, P-
=P+ olacak (boşluk mutlaka arada olacağından sağ=nokta yada boşluk)
                if(isnan(Point(i+1,j))) % altta nokta yoksa nan
ise (alt=nan, sol=nokta yada boşluk)
                    c=1; %yukarıda işlenecek nokta olmadığı için
tüm ağırlık Cye binecek c=1 b=0
                    C=zeros(3,3);
                    C(1,1)=c;
                    C(1,3)=c*step_along_x;
                    C(3,3)=1;
                    S_minus2{i,j}=C*S_plus2{i,j-1};
                    P_minus2{i,j}=C*P_plus2{i,j-1}*C'+Q;

K2{i,j}=P_minus2{i,j}*D'*inv(D*P_minus2{i,j}*D'+R);
                    S_plus2{i,j}=S_minus2{i,j};
                    P_plus2{i,j}=P_minus2{i,j};
                    last_processed_i=i;
                    last_processed_j=j;
                else % alt=nokta yada boşlu, sol nokta yada
boşluk
                    b=0.5; %C ve B eşit ağırlıklı olacak
                    c=0.5;
                    B=zeros(3,3);
                    C=zeros(3,3);
                    B(1,1)=b;
                    B(1,2)=b*step_along_y;
                    B(2,2)=1;
                    C(1,1)=c;
                    C(1,3)=c*step_along_x;
                    C(3,3)=1;

S_minus2{i,j}=B*S_plus2{i+1,j}+C*S_plus2{i,j-1};

P_minus2{i,j}=B*P_plus2{i+1,j}*B'+C*P_plus2{i,j-1}*C'+Q;

K2{i,j}=(P_minus2{i,j}*D'*inv(D*P_minus2{i,j}*D'+R));
                    S_plus2{i,j}=S_minus2{i,j};
                    P_plus2{i,j}=P_minus2{i,j};
                    last_processed_i=i;
                    last_processed_j=j;
                end
            else % nokta varsa
                if(last_processed_i==0)%iterasyonun ilk
başlangıç noktası (alt=nan, sol=nan)
                    S_minus2{i,j}=[Z(i,j);0;0];
                    P_minus2{i,j}=Q;

K2{i,j}=(P_minus2{i,j}*D'*inv(D*P_minus2{i,j}*D'+R));
                    S_plus2{i,j}=S_minus2{i,j}+K2{i,j}*(Z(i,j)-
D*S_minus2{i,j});
                    P_plus2{i,j}=(Q-K2{i,j}*D)*P_minus2{i,j};
                    last_processed_i=i;
                    last_processed_j=j;
                else % işlenen nokta satırın ilk noktası değilse
yani solu nan değilse

```

```

        if(isnan(Point(i+1,j))) % altta nokta yoksa
nan ise (alt=nan, sol=nokta yada boşluk)
            c=1; %aşağıda işlenecek nokta olmadığı
için tüm ağırlık Cye binecek c=1 b=0
            C=zeros(3,3);
            C(1,1)=c;
            C(1,3)=c*step_along_x;
            C(3,3)=1;
            S_minus2{i,j}=C*S_plus2{i,j-1};
            P_minus2{i,j}=C*P_plus2{i,j-1}*C'+Q;

K2{i,j}=P_minus2{i,j}*D'*inv(D*P_minus2{i,j}*D'+R);

S_plus2{i,j}=S_minus2{i,j}+K2{i,j}*(Z(i,j)-D*S_minus2{i,j});
P_plus2{i,j}=(Q-
K2{i,j}*D)*P_minus2{i,j};
            last_processed_i=i;
            last_processed_j=j;
        else % altta boşluk veya nokta varsa
(alt=nokta yada boşluk, sol=nokta yada boşluk)
            b=0.5; %C ve B eşit ağırlıklı olacak
            c=0.5;
            B=zeros(3,3);
            C=zeros(3,3);
            B(1,1)=b;
            B(1,2)=b*step_along_y;
            B(2,2)=1;
            C(1,1)=c;
            C(1,3)=c*step_along_x;
            C(3,3)=1;

S_minus2{i,j}=B*S_plus2{i+1,j}+C*S_plus2{i,j-1};

P_minus2{i,j}=B*P_plus2{i+1,j}*B'+C*P_plus2{i,j-1}*C'+Q;

K2{i,j}=P_minus2{i,j}*D'*inv(D*P_minus2{i,j}*D'+R);

S_plus2{i,j}=S_minus2{i,j}+K2{i,j}*(Z(i,j)-D*S_minus2{i,j});
P_plus2{i,j}=(Q-
K2{i,j}*D)*P_minus2{i,j};
            last_processed_i=i;
            last_processed_j=j;
        end
    end
end
    end
    soladogru=1;
    else %filtre sola doğru hareket eder
        for j=column_M_out:-1:1
            if(isnan(Point(i,j))) % nokta yoksa döngüye devam
eder
                continue;
            elseif(Point(i,j)==0)% boşluğa geldiğinde S==S+, P-
=P+ olacak (boşluk mutlaka arada olacağından sağ=nokta yada boşluk)
                if(isnan(Point(i+1,j))) % altta nokta yoksa nan
ise (alt=nan, sağ=nokta yada boşluk)
                    c=1; %yukarıda işlenecek nokta olmadığı için
tüm ağırlık Cye binecek c=1 b=0
                    C=zeros(3,3);
                    C(1,1)=c;
                    C(1,3)=c*step_along_x;

```

```

C(3,3)=1;
S_minus2{i,j}=C*S_plus2{i,j+1};
P_minus2{i,j}=C*P_plus2{i,j+1}*C'+Q;

K2{i,j}=P_minus2{i,j}*D'*inv(D*P_minus2{i,j}*D'+R);
S_plus2{i,j}=S_minus2{i,j};
P_plus2{i,j}=P_minus2{i,j};
last_processed_i=i;
last_processed_j=j;
else % alt=nokta yada boşluk, sağ nokta yada
boşluk
    b=0.5; %C ve B eşit ağırlıklı olacak
    c=0.5;
    B=zeros(3,3);
    C=zeros(3,3);
    B(1,1)=b;
    B(1,2)=b*step_along_y;
    B(2,2)=1;
    C(1,1)=c;
    C(1,3)=c*step_along_x;
    C(3,3)=1;

S_minus2{i,j}=B*S_plus2{i+1,j}+C*S_plus2{i,j+1};
P_minus2{i,j}=B*P_plus2{i+1,j}*B'+C*P_plus2{i,j+1}*C'+Q;

K2{i,j}=(P_minus2{i,j}*D'*inv(D*P_minus2{i,j}*D'+R));
S_plus2{i,j}=S_minus2{i,j};
P_plus2{i,j}=P_minus2{i,j};
last_processed_i=i;
last_processed_j=j;
end
else % nokta varsa
    if(last_processed_i==0)%iterasyonun ilk
başlangıç noktası (alt=nan, sağ=nan)
        S_minus2{i,j}=[Z(i,j);0;0];
        P_minus2{i,j}=Q;

K2{i,j}=(P_minus2{i,j}*D'*inv(D*P_minus2{i,j}*D'+R));
S_plus2{i,j}=S_minus2{i,j}+K2{i,j}*(Z(i,j)-
D*S_minus2{i,j});
P_plus2{i,j}=(Q-K2{i,j}*D)*P_minus2{i,j};
last_processed_i=i;
last_processed_j=j;
else % işlenen nokta satırın ilk noktası değilse
yani sağı nan değilse
    if(isnan(Point(i+1,j))) % altta nokta yoksa
nan ise (alt=nan, sağ=nokta yada boşluk)
        c=1; %aşağıda işlenecek nokta olmadığı
için tüm ağırlık Cye binecek c=1 b=0
        C=zeros(3,3);
        C(1,1)=c;
        C(1,3)=c*step_along_x;
        C(3,3)=1;
        S_minus2{i,j}=C*S_plus2{i,j+1};
        P_minus2{i,j}=C*P_plus2{i,j+1}*C'+Q;

K2{i,j}=P_minus2{i,j}*D'*inv(D*P_minus2{i,j}*D'+R);

S_plus2{i,j}=S_minus2{i,j}+K2{i,j}*(Z(i,j)-D*S_minus2{i,j});

```

```

P_plus2{i,j}=(Q-
K2{i,j}*D)*P_minus2{i,j};
    last_processed_i=i;
    last_processed_j=j;
    else % altta boşluk veya nokta varsa
    (alt=nokta yada boşluk, sol=nokta yada boşluk)
        b=0.5; %C ve B eşit ağırlıklı olacak
        c=0.5;
        B=zeros(3,3);
        C=zeros(3,3);
        B(1,1)=b;
        B(1,2)=b*step_along_y;
        B(2,2)=1;
        C(1,1)=c;
        C(1,3)=c*step_along_x;
        C(3,3)=1;

S_minus2{i,j}=B*S_plus2{i+1,j}+C*S_plus2{i,j+1};

P_minus2{i,j}=B*P_plus2{i+1,j}*B'+C*P_plus2{i,j+1}*C'+Q;

K2{i,j}=P_minus2{i,j}*D'*inv(D*P_minus2{i,j}*D'+R);

S_plus2{i,j}=S_minus2{i,j}+K2{i,j}*(Z(i,j)-D*S_minus2{i,j});
    P_plus2{i,j}=(Q-
K2{i,j}*D)*P_minus2{i,j};
        last_processed_i=i;
        last_processed_j=j;
    end
end
end
end
soladogru=0;
end
end
end
disp('Sağ _Alt Köşeden başlayan işlem sonlandı...');

disp('S_final hesaplanıyor...');
S_final = cell(row_M_out, column_M_out); % S_plus ve S_plus2
değerlerinin ortaamalarının atanacağı cell array 3x1
P_final = cell(row_M_out, column_M_out); % S_plus ve S_plus2
değerlerinin ortaamalarının atanacağı cell array 3x3
for i=1:row_M_out
    for j=1:column_M_out
        if(isnan(Point(i,j))) % nokta yoksa döngüye devam eder
            continue;
        else

S_final{i,j}=inv(inv(P_plus2{i,j})+inv(P_plus{i,j}))*(inv(P_plus{i,j}
))*S_plus{i,j}+inv(inv(P_plus2{i,j})+inv(P_plus{i,j}))*S_plus2{i,j};
        P_final{i,j}=inv(inv(P_plus2{i,j})+inv(P_plus{i,j}));
            end
        end
    end
end
disp('S_final hesaplandı...');

disp('hesaplanan boşlukları dosyaya yazdırıyor...');
Bosluk = zeros(1,3);
bosluk_counter=1;
%x_coordinate=0;

```

```

%y_coordinate=0;
for i=1:row_M_out
    for j=1:column_M_out
        if(Point(i,j)==0) %boşluksa
            %disp('i');disp(i);disp('j');disp(j);
            x_coordinate=x_min+double(j-1)*step_along_x;
            y_coordinate=y_min+double(i-1)*step_along_y;
            Bosluk(bosluk_counter,1)=x_coordinate;
            Bosluk(bosluk_counter,2)=y_coordinate;
            tmp=S_final{i,j};
            Bosluk(bosluk_counter,3)=tmp(1);
            bosluk_counter=bosluk_counter+1;
            %disp('X'); disp(x_coordinate); disp('Y');
disp(y_coordinate); disp('Z'); disp(tmp(1));
        else
            continue;
        end
    end
end
disp('yazılacak nokta sayısı=');disp(bosluk_counter);
dlmwrite('bosluk_output.txt', Bosluk, 'delimiter', '\t',
'precision', '%.3f');
disp('"bosluk_output.txt" adlı dosyaya yazıldı...');

disp('script sonlandı...')

```

Ek-3: Matlab Küre örnekleme kodu

```
% Parametreleri girilen kürenin yüzey koordinatlarını belirli bir
gride
% göre hesaplayan script Emre Pınarcı 2007
disp('script başladı...');
grid_interval=0.04; % x,y düzlemindeki grid aralığı
d=10; % kürenin çapı
r=d/2;
Coordinates=zeros(1,3);
counter=1;
disp('Küre parametreleri:'); disp('çap='); disp(d); disp('grid
aralığı='); disp(grid_interval);

disp('küre yüzey koordinatları hesaplanıyor...');
for y=0:grid_interval:d
    for x=0:grid_interval:d
        z=sqrt(r^2-(x-r)^2-(y-r)^2);
        if(isreal(z))
            Coordinates(counter,1)=x;
            Coordinates(counter,2)=y;
            Coordinates(counter,3)=z;
            counter=counter+1;
        else
            continue;
        end
    end
end
disp('küre yüzey koordinatları hesaplandı. '); disp('Hesaplanan nokta
sayısı='); disp(counter);

disp('Koordinatlar dosyaya yazılıyor...');
dlmwrite('kure_koordinat.txt', Coordinates, 'delimiter', '\t',
'precision', '%.3f');
disp('Koordinatlar "kure_koordinat.txt" adlı dosyaya yazıldı. ');
disp('script sonlandı.');
```

Ek-4: Matlab lineer enterpolasyon kodu

```

% z=F(x,y) yüzeyi geçirir Emre Pınarcı 2007

disp('Script başladı...');
disp('Üçgenleme başladı...');
UX=zeros(row_M_out,column_M_out);
UY=zeros(row_M_out,column_M_out);
UZ=zeros(row_M_out,column_M_out);
ti=0:0.04:10;
for i=1:row_M_out
    for j=1:column_M_out
        if(isnan(X(i,j)))
            continue;
        else
            UX(i,j)=X(i,j);
            UY(i,j)=Y(i,j);
            UZ(i,j)=Z(i,j);
        end
    end
end
[XI,YI] = meshgrid(ti,ti);
ZI=griddata(UX,UY,UZ,XI,YI);
mesh(XI,YI,ZI), hold
disp('Üçgenleme tamamlandı. ');disp('doldurulan boşluklar
ayıklanıyor... ');
%plot3(UX,UY,UZ,'o'), hold off
[bosluk_row, bosluk_col]=size(Bosluk);
UBosluk=zeros(bosluk_row,bosluk_col);
counter=1;
for i=1:row_M_out
    for j=1:column_M_out
        if(Point(i,j)==0)
            UBosluk(counter,1)=XI(i,j);
            UBosluk(counter,2)=YI(i,j);
            UBosluk(counter,3)=ZI(i,j);
            counter=counter+1;
        end
    end
end
disp('doldurulan boşluklar ayıklandı. ');
disp('yazılacak nokta sayısı=');disp(bosluk_row);
dlmwrite('bosluk_ucgen_output.txt', UBosluk, 'delimiter', ' ',
'precision', '%.3f', 'newline', 'pc');
disp('"bosluk_ucgen_output.txt" adlı dosyaya yazıldı...');

disp('Script sonlandı. ');

```

Ek-5: Matlab gürültü ekleme kodu

```
% Gürültü Ekleme Emre Pınarcı 2007

disp('Script başladı...');
disp('kure_koordinat.txt dosyası okunuyor...');
Zin=dlmread('kure_koordinat.txt');
[row,col]=size(Zin);
disp('okunan dosya sayısı=');disp(row);
mu=0; %eklenecek gürültünün ortalaması
sd=0.006; %standart sapma
%q=49072; %sayı adeti
disp('normal dağılımlı gürültü oluşturuluyor...');
N=random('normal',mu,sd,row,1);
disp('gürültü oluşturuldu.');
```