

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

**EV TARIMI UYGULAMALARINA YÖNELİK TOPRAK NEMİ KESTİRİM
YÖNTEMLERİ**

YÜKSEK LİSANS TEZİ

Hilal TÜRK

Bilişim Uygulamaları Anabilim Dalı

Bilişim Uygulamaları Tezli Yüksek Lisans Programı

HAZİRAN 2018

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

**EV TARIMI UYGULAMALARINA YÖNELİK TOPRAK NEMİ KESTİRİM
YÖNTEMLERİ**

YÜKSEK LİSANS TEZİ

**Hilal TÜRK
(708141006)**

Bilişim Uygulamaları Anabilim Dalı

Bilişim Uygulamaları Tezli Yüksek Lisans Programı

Tez Danışmanı: Doç. Dr. Behçet Uğur TÖREYİN

HAZİRAN 2018

İTÜ, Bilişim Enstitüsü'nün 708141006 numaralı Yüksek Lisans Öğrencisi Hilal TÜRK, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “EV TARIMI UYGULAMALARINA YÖNELİK TOPRAK NEMİ KESTİRİM YÖNTEMLERİ” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç. Dr. Behçet Uğur TÖREYİN**

Istanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Lütfiye DURAK ATA**

Istanbul Teknik Üniversitesi

Dr. Öğr. Üyesi Süha TUNA

Fatih Sultan Mehmet Vakıf Üniversitesi

Teslim Tarihi : **30 Nisan 2018**

Savunma Tarihi : **4 Haziran 2018**





Aileme,



ÖNSÖZ

Bu çalışmada toprak nemini tahmine yönelik enterpolasyon yöntemleri ve kullanım amaçları araştırılmıştır. Enterpolasyon yöntemlerinden, enterpolasyonun uygulama alanlarından, toprak nemi tahmin işlemlerinden ve tarımda bilişimin öneminden bahsedilmiştir.

Bilişim yaşamın birçok alanında kolaylıklar sağlamaktadır. Bilişim uygulamalarının tarımda kullanımı, tarımda ürünlerden elde edilen verimi arttırmaya katkı sağlamaktadır. Bu katkıların arasında toprak neminin tahmini de önemli bir paya sahiptir. Bu tahminlerin başarısını arttırmak için kullanılan yöntemler arasından, uygun olan yöntemler araştırılmıştır, bazıları uygulanmıştır, katkı sağlayabilecek öneriler sunulmuş ve karşılaştırmalar yapılmıştır.

Tez çalışmamın planlanmasında, araştırılmasında, oluşturulmasında bana yol gösteren sayın hocam Doç. Dr. Behçet Uğur TÖREYİN'e, benden desteklerini esirgemeyen sayın hocalarım Prof. Dr. Lütfiye DURAK ATA'ya ve Dr. Öğr. Üyesi Süha TUNA'ya teşekkürlerimi sunarım.

Haziran 2018

Hilal TÜRK
(Bilgisayar Mühendisi, Öğretim Görevlisi)



İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET.....	xix
SUMMARY	xxi
1. GİRİŞ	1
1.1 Tezin Amacı	2
1.2 Literatür Araştırması	2
1.3 Hipotez	4
2. ENTERPOLASYON YÖNTEMLERİ.....	7
2.1 Ters-Mesafe Ağırlıklı Yöntem (Inverse Distance Weighted – IDW)	7
2.2 Polinom Yüzeyleri ile Enterpolasyon	8
2.3 En Küçük Kareler Yöntemine Göre Enterpolasyon	8
2.4 Kriging Yöntemleri ile Enterpolasyon.....	9
3. MATERYAL ve YÖNTEM.....	11
3.1 Arduino.....	14
3.2 Matlab.....	15
3.3 Visual Studio	16
3.4 Uygulanan Yöntem	16
3.4.1 Ters-mesafe ağırlıklı yöntem (inverse distance weighted - IDW).....	16
3.4.2 Ters-mesafe ağırlıklı yöntemin verilere uygulanması	19
3.4.3 Ters-mesafe ağırlıklı yöntemin yarıçap aralığı belirtilerek uygulanması.....	25
3.4.4 Ters-mesafe ağırlıklı yöntemin değeri ani değişenler çıkartılarak uygulanması.....	26
4. SAYISAL UYGULAMALAR.....	29
4.1 Kübik Enterpolasyon.....	29
4.2 Doğrusal Eterpolasyon	34
4.3 Spline Enterpolasyon	39
4.3.1 Doğrusal spline enterpolasyon	40
4.3.2 Kuadratik (ikinci dereceden) spline enterpolasyon.....	40
4.3.3 Kübik (üçüncü dereceden) spline enterpolasyon.....	40
5. SONUÇ ve ÖNERİLER.....	47
KAYNAKLAR	53
EKLER.....	57
ÖZGEÇMİŞ.....	77

KISALTMALAR

IDW	: Inverse Distance Weighted
MAPE	: Mean Absolute Percentage Error
BT	: Bilişim Teknolojileri
AMST-E	: Advanced Microwave Scanning Radiometer
OK	: Ordinary Kriging





ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 3.1 : Arduino Uno : Teknik Özellikler	15
Çizelge 3.2 : Toprak Nem Miktarı Aralıkları	19
Çizelge 3.3 : Sensörlerden Alınan Örnek Veri Kümesi.....	25
Çizelge 5.1 : Sensör_0 ve Sensör_2'nin gerçek değerleri ile diğer sensörlerden IDW hesaplamasıyla bulunan değerleri.....	47





ŞEKİL LİSTESİ

Sayfa

- Şekil 2.1** : IDW hesaplamasının şekil üzerinde gösterimi.....7
- Şekil 3.1** : 72x54 cm'lik saksıya 18 cm'lik aralıklarla yerleştirilen 6 adet sensörlerden verilerin alınması ve breadboard üzerinde kurulu olan arduino devresi ile seriporttan bilgisayara aktarımı..... 11
- Şekil 3.2** : Microsoft Visual Studio'da hazırlanan program arayüzü ile verilerin alındığı seriportların açılması, her sensörden alınan verilerin ilgili sensör düğmelerinden “.txt” dosyalara ayrı ayrı kaydedilmesi ve tüm sensörlerden gelen verilerin sensör numarası, değeri, tarihi okunarak ve “.txt” dosyalara aktarımı..... 12
- Şekil 3.3** : Microsoft Visual Studio'da hazırlanan programdan IDW algoritmalarının çalıştırılması. 1 numaralı gösterilen kısım sensör konumları, 2 numaralı kısım “.txt” dosyadan alınan veriler, 3 numaralı kısım alınacak verilerin zaman aralığının seçimi.....13
- Şekil 3.4** : IDW algoritmalarından alınan sonuçlara göre Matlab programında grafiklerin oluşturulması. Kırmızı noktalar ile Dayanak, mavi noktalar ile Referans noktalarının gösterimi. ... 13
- Şekil 3.5** : Arduino kartı, LM393 sensörü ve prob bağlantısının sağlanması..... 14
- Şekil 3.6** : IDW yönteminin hesaplanması için sensör konumları ve uzaklıkların gösterimi..... 17
- Şekil 3.7** : Toprağa yerleştirilen sensörlerin etki alanlarının belirlenmesi sonucunda sensörlerin birbirinden 18 cm uzaklıkta olacak şekilde saksının içinde konumlandırılması..... 18
- Şekil 3.8** : 72x54 cm'lik saksıya sensörlerin yerleşimini gösteren deney ortamının modellenmesi ve sensörlerin konumlandırılması..... 19
- Şekil 3.9** : IDW yönteminin uygulanacağı dayanak noktaları ve tahmin noktası. Sensör 4'e uzaklığı gösterilen noktaların değerleri ve uzaklıkları kullanılarak, Sensör 4'ün değerinin tahmin edilmesi..... 20
- Şekil 3.10** : IDW yönteminin uygulanacağı dayanak noktaları ve tahmin noktası. Sensör 2'ye uzaklığı gösterilen noktaların değerleri ve uzaklıkları kullanılarak, Sensör 2'nin değerinin tahmin edilmesi.....21
- Şekil 3.11** : 0. Sensörden deney süresinde alınan verilerin zamanla dağılımı. 19:40 zamanında, 2 numaralı sensör bölgesinde yapılan sulamanın, 0.sensörün okunan değerlerine etkisi. 22
- Şekil 3.12** : 1. Sensörden deney süresinde alınan verilerin zamanla dağılımı. 19:40 zamanında yapılan sulamanın 1.sensörden okunan değerlere etkisi.....22
- Şekil 3.13** : 2. Sensörden deney süresinde alınan verilerin zamanla dağılımı. 19:40 zamanında 2. sensör bölgesinde sulama yapıldığı için, okunan değerlerin sürekli olarak düştüğünün gösterimi. 23

Şekil 3.14 : 3. Sensörden deney süresinde alınan verilerin zamanla dağılımı. 19:40 zamanında yapılan sulama etkisi ile 3. Sensörden okunan değerlerin düşmesi.....	23
Şekil 3.15 : 4. Sensörden deney süresinde alınan verilerin zamanla dağılımı. 19:40 anında 2. Sensör bölgesinde yapılan sulamanın, 4. sensör değerlerine etkisinin daha az olduğunu görülmektedir. 4. sensör ve 2. Sensör arasındaki mesafenin etkisi.	24
Şekil 3.16 : 5. Sensörden deney süresinde alınan verilerin zamanla dağılımı. 5. sensör mesafe olarak 2. sensöre yakın olduğundan, özellikle 19:40 zamanından sonra 5. sensörden okunan değerler düşmüştür. 5. sensör değerlerinin, 2. sensör değerlerine yakın olduğunun gözlemlenmesi.	24
Şekil 3.17 : Farklı IDW algoritmalarının Microsoft Visual Studio’da çalıştırılması. 1 numara ile gösterilen program arayüzünde bulunan “Çap Aralığı” sekmesinin gösterimi.	26
Şekil 3.18 : IDW yönteminin değeri ani değişenlere yönelik algortiması. 1 numara ile hesaplama algoritmasının değiştirildiği onay kutusunun gösterimi. ...	27
Şekil 4.1 : Kübik Enterpolasyonun verilere uygulanması ile 12:03 anında oluşturulan kübik yüzey grafiği ve nem dağılım haritası.	30
Şekil 4.2 : Kübik Enterpolasyonun verilere uygulanması ile 12:33 anında oluşturulan kübik yüzey grafiği ve nem dağılım haritası.	31
Şekil 4.3 : Kübik Enterpolasyonun verilere uygulanması ile 13:03 anında oluşturulan kübik yüzey grafiği ve nem dağılım haritası. 12:33-13:03 aralığında yapılan sulamanın grafikte ve nem haritasında gösterilmesi.....	32
Şekil 4.4 : Kübik Enterpolasyonun verilere uygulanması ile 13:33 anında oluşturulan kübik yüzey grafiği ve nem dağılım haritası.	33
Şekil 4.5 : Kübik Enterpolasyonun verilere uygulanması ile 14:03 anında oluşturulan kübik yüzey grafiği ve nem dağılım haritası.	34
Şekil 4.6 : Doğrusal Enterpolasyonun verilere uygulanması ile 12:03 anında oluşturulan doğrusal yüzey grafiği ve nem dağılım haritası.	35
Şekil 4.7 : Doğrusal Enterpolasyonun verilere uygulanması ile 12:33 anında oluşturulan doğrusal yüzey grafiği ve nem dağılım haritası.	36
Şekil 4.8 : Doğrusal Enterpolasyonun verilere uygulanması ile 13:03 anında oluşturulan doğrusal yüzey grafiği ve nem dağılım haritası. 12:33-13:03 arasında yapılan sulamanın grafikte ve nem haritasında gösterilmesi.	37
Şekil 4.9 : Doğrusal Enterpolasyonun verilere uygulanması ile 13:03 anında oluşturulan doğrusal yüzey grafiği ve nem dağılım haritası.	38
Şekil 4.10 : Doğrusal Enterpolasyonun verilere uygulanması ile 14:03 anında oluşturulan doğrusal yüzey grafiği ve nem dağılım haritası.	39
Şekil 4.11 : Spline Enterpolasyonun verilere uygulanması ile 12:03 anında oluşturulan yüzey grafiği ve nem dağılım haritası.	41
Şekil 4.12 : Spline Enterpolasyonun verilere uygulanması ile 12:33 anında oluşturulan yüzey grafiği ve nem dağılım haritası.	42
Şekil 4.13 : Spline Enterpolasyonun verilere uygulanması ile 13:03 anında oluşturulan yüzey grafiği ve nem dağılım haritası. 12:33-13:03 arasında yapılan sulamanın grafikte ve nem dağılımında görülmesi.	43
Şekil 4.14 : Spline Enterpolasyonun verilere uygulanması ile 13:33 anında oluşturulan yüzey grafiği ve nem dağılım haritası.	44
Şekil 4.15 : Spline Enterpolasyonun verilere uygulanması ile 14:03 anında oluşturulan yüzey grafiği ve nem dağılım haritası.	45

Şekil 5.1 : IDW yönteminin her bir sensör için hesaplaması ve sonuçlara göre gerçek değerler ile MAPE’de hata oranlarının bulunması.....	48
Şekil 5.2 : Önerilen algoritmaya göre IDW yönteminin hesaplaması ve sonuçlara göre gerçek değerler ile MAPE’de hata oranlarının bulunması.....	48
Şekil 5.3 : Değeri ani değişen sensörlerin verilerini dahil etmeden IDW yönteminin hesaplanması. “Değeri ani değişenleri çıkar” onay kutusu işaretlendiğinde yapılan hesaplamamanın sonuçlarının gösterimi.....	49
Şekil 5.4 : Önerilen IDW algoritmasının sonuçlarına göre ortalama hatanın bulunması.....	50





EV TARIMI UYGULAMALARINA YÖNELİK TOPRAK NEMİ KESTİRİM YÖNTEMLERİ

ÖZET

Teknoloji hayatımızın her alanında kendini göstermektedir. Teknoloji insan hayatını daha kolay ve konforlu hale getirmektedir. Teknolojik gelişmeler ile birlikte iletişim, ulaşım, sağlık hizmetleri ve mühendislik gibi alanlarda da büyük ilerlemeler olmaktadır. Her alanda üretim hızlanmıştır, zaman kayıpları gittikçe azalmaktadır ve bunun sayesinde çeşitli yönlerden tasarruf sağlanmaktadır.

Bilişim Teknolojileri - BT çok yaygın olarak farklı sektörlerde kullanılmaktadır. Bunların içinde tarım faaliyetlerinin yürütüldüğü sektörler de bulunmaktadır. Tarım faaliyetlerinde teknoloji kullanımı ve BT verimliliği arttırmaktadır. Bundan dolayı, bu alanda yapılan teknolojik gelişmelere katkı yapacak bir çalışma alanı seçilmiştir.

Şehirlerde binalaşma arttıkça, yeşil alanlar azalmaktadır. Bunun yanında nüfus artışı ile birlikte gıda ihtiyacı artmaktadır. Organik gıda ihtiyacı ve yeşil alan ihtiyacı, insanları yeni çözümlere yöneltmiştir. Ev tarımı ve ev bahçeleri uygulamaları yaygınlaşmaktadır. Ev tarımı ve ev bahçeleri faaliyetlerini tasarruflu bir şekilde sürdürmek önemlidir. Ev içi tarım uygulamalarında verimi arttırmak için teknolojiden faydalanılmaktadır. Yaygınlaşan akıllı ev sistemlerine tarım faaliyetleri de dahil edilebilir.

Ev tarımı uygulamalarında kullanılabilecek olan bu çalışmada toprak neminin yüzeydeki dağılımının tahminine yönelik yapılmış olan bir çok çalışmayı inceleyerek yeni yöntemler üzerinde gözlemler yapılmıştır. Çalışmada kullanılan veriler, ev tarımı için kullanılabilecek basit, maliyeti ucuz olan, 1 Arduino ve 6 sensörden oluşan donanım devresinden anlık olarak alınmıştır ve bilgisayara kayıt edilmiştir.

Toprak üzerinden veri almak için alana 6 sensör noktası belirlenerek eşit mesafelerle sensörler yerleştirilmiştir. Yerleştirme konumları sensörlerin etki alanı hesaplanarak belirlenmiştir. Alınan veriler üzerinde, sensörlerin bulunduğu noktaların dışındaki noktalardaki değerlerin tahminine yönelik enterpolasyon çalışmaları yapılmıştır. Enterpolasyon çalışmalarına yaygın olarak kullanılan Ters-Mesafe Ağırlıklı Yöntem (Inverse Distance Weighted – IDW) yöntemi ile başlanmıştır. Sensörlerin bir tanesi tahmin noktası olarak belirlenerek, diğer sensör noktalarından alınan değerler ile tahmin noktasındaki sensörün değeri tahmin edilmeye çalışılmıştır. Sonuçlar gerçek veriler ile karşılaştırılarak hata ölçümü yapılmıştır.

Farklı enterpolasyon yöntemlerinden olan Doğrusal Enterpolasyon (Linear Interpolation), Spline Enterpolasyon (Spline Interpolation), Kübik Enterpolasyon (Cubic Interpolation) yöntemleri ayrı ayrı, verilere uygulanmıştır. Sonuçlara göre yüzeydeki nem dağılımı Matlab’da çizdirilerek görülmüştür. Farklı zaman aralıkları için yapılan ölçüm sonuçları karşılaştırılmıştır.

Ters Mesafe Ağırlıklandırma (Inverse Distance Weighting) IDW yöntemini geliştirmeye ve tahminlerin doğruluğunu arttırmaya yönelik iki algoritma önerilmiştir. İlk algoritmada tahmin noktaları için kapsama alanı oluşturularak, istenilen bölgede tahmin hesaplamasını yaparak uzakta bulunan noktaların, tahmine etkisinin en aza indirilmesi amaçlanmıştır. Tahmin edilecek noktalardan, kullanıcı tarafından belirlenen uzaklık aralığında olanların tahmini yapılmıştır. İkinci algoritmada ise değeri ani düşen sensörler tespit edilerek, değerlerdeki ani değişikliğin tahmin başarısını düşürmemesi için etkileri azaltılarak hesaplamaya yapılmıştır.

Sonuç olarak IDW algoritmalarından alınan sonuçlar değerlendirilerek hata oranları karşılaştırılmıştır. Hata oranları hesaplamasında, Ortalama Mutlak Yüzde Hata (Mean Absolute Percentage Error – MAPE) formülü kullanılmıştır. Bu karşılaştırmaların sonucunda, verilen aralığa bağlı olarak programın başarılı bir şekilde ölçüm yaptığı gözlemlenmiştir. Ayrıca değeri ani değişen sensörlerin tespit edilerek hesaplamaya dahil edilmemesinin, tahmin noktalarındaki doğruluğu arttırdığı gözlemlenmiştir.

Gelecek çalışmalarda, bu çalışmada önerilen algoritma geliştirilebilir. Toprak nemi ortam sıcaklığı, ortam nemi, toprak içindeki maddeler gibi başka birçok etkene bağlıdır. Bu sebeple ölçümlerin başarısını arttırmak için, bu etkenler de ölçülerek ve kontrol altına alınarak algoritmaya katılabilir. Oluşturulan elektronik devre ve yazılımlar entegre edilerek daha küçük hale getirilebilir. Oluşturulan kodlara ek olarak akıllı cihazlardan yönetilebilir bir sistem haline getirilerek, akıllı ev sistemine bu çalışma eklenebilir.

SOIL MOISTURE METHODS FOR HOME AGRICULTURAL APPLICATIONS

SUMMARY

Technology is manifested in all areas of our lives. Technology makes human life easier and more comfortable. Along with technological developments, there is also great progress in areas such as communication, transportation, health care and engineering. Every field production is accelerated, the time losses are getting smaller and thanks to this savings in various aspects are provided.

Information Technology is widely used in different sectors. These include sectors where agricultural activities are carried out. It increases the use of technology and IT productivity in agricultural activities. Therefore, a study area was selected to contribute to the technological developments in this area.

As urbanization increases in cities, green areas decrease. Besides this, with the increase of the population, the food need is increasing. The need for organic food and green space has led people to new solutions. Household agriculture and home garden applications are becoming widespread. It is important to carry out household agriculture and home gardens activities in a conservative way. Domestic agriculture practices benefit from technology to increase yields. Increasingly smart household systems can include agricultural activities.

In this study, which can be used in household agriculture applications, observations were made on new methods by examining many studies made on the estimation of the distribution of soil moisture on the surface. The data used in the study were taken instantaneously from the hardware circuit consisting of 1 Arduino and 6 sensors which can be used for home farming, and it is recorded on the computer.

In order to receive the data via the ground, 6 sensor points were set in the area and equally spaced sensors were installed. The placement locations were determined by calculating the domain of the sensors. Interpolation studies were performed on the received data to estimate the values at the points outside the points where the sensors are located. It started with the Inverse Distance Weighted (IDW) method, which is commonly used for interpolation studies. One of the sensors was determined as a prediction point and tried to estimate the value of the sensor at the predicted point by the values obtained from the other sensor points. Errors were measured by comparing the results with the actual data.

There are different interpolation techniques. These methods are divided into various sub-methods according to the region where the interpolation is to be performed and the purpose of the interpolation. There are three main approaches to solve interpolation problems. These approaches are interpolating with point interpolation, interpolation with a single function covering the entire region, and locally defined piecewise functions.

Inverse Distance Weighted Method is often used in interpolation operations. It is a local method for producing predictions based on neighboring spots. Estimation of unknown points is calculated by weighting according to the distance between points. Interpolation with polynomial surfaces is a method used when interpolation is required for a surface. Linear, quadratic, cubic surfaces are obtained in interpolation processes with orthogonal polynomials. Unknown binary bi-liners, bi-quadratic, bi-cubic surfaces are obtained in interpolation processes in non-nonspecific polynomials and in processes performed for Bi-linear, Bi-quadratic and Bi-cubic surfaces.

If interpolation according to least squares method, many areas such as prediction, spectral analysis, functional analysis are used. Interpolation processes with Kriging methods are commonly used interpolation methods in geographical statistics. Geology, environment, mining, meteorology, such as the work done in these areas are available. Simple Kriging, Ordinary Kriging (OK), Universal Kriging, Block Kriging, Indicator Kriging, Disjunctive Kriging and Cokriging are used in different areas.

In this work, the cubic interpolation applied to the data gives a unique set of cubic polynomials between the data points. Thus, smooth curves can be generated from the array points. The coefficient in cubic polynomials is the weight used for interpolation.

Linear Interpolation, Spline Interpolation and Cubic Interpolation methods, which are different interpolation methods, have been applied to the data separately. According to the results, the moisture distribution on the surface was seen in Matlab. The measurement results for different time intervals are compared.

Studies with the same data in different interpolation methods yielded similar moisture distribution results. These distributions are easily seen on the surfaces created in Matlab. The blue colored parts on the surface of the moisture show that the soil is damp and the yellow colored parts indicate that the soil is dry.

Inverse Distance Weighting two algorithms have been proposed to improve the IDW method and to increase the accuracy of estimates. In the first algorithm, it is aimed to create the coverage area for the prediction points and calculate the estimation of the desired region so that the distant points are deducted from the estimation effect. The predicted points are estimated by the user at the specified distance interval. In the second algorithm, the sensors are detected suddenly and the effects are reduced so as not to reduce the prediction success of the sudden change in the values.

As a result, the results obtained from the IDW algorithms are evaluated and the error rates are compared. In calculating error rates, the Mean Absolute Percentage Error MAPE formula is used. As a result of these comparisons, it was observed that the program performed a successful measurement based on the given range. Furthermore, it has been observed that the fact that suddenly changing sensors are detected and not included in the calculation, increases the accuracy at the predicted points.

In future studies, the proposed algorithm can be improved. Soil moisture depends on many other influences such as ambient temperature, ambient humidity, and soils in the soil. For this reason, to increase the success of the measurements, these factors can also be included in the algorithm by being measured and controlled. The generated electronic circuitry and software can be integrated to make it smaller. In addition to the generated codes, this work can be added to the intelligent home system by becoming a manageable system from smart devices.





1. GİRİŞ

Dünyada teknoloji her geçen gün gelişmektedir. Teknoloji hayatın birçok alanına girmekte ve kolaylıklar sağlamaktadır. Teknolojik gelişmelerin en öne çıktığı alanlar iletişim, bilişim ve ulaşım sektörleridir [1].

Bilişim teknolojileri sistemlerin ileriye yönelik çalışmalarını tahmin etmekte ve sistemleri akıllı sistemlere dönüştürmektedir. Bilişim teknolojilerinin yaygın kullanıldığı alanlardan biri de tarım faaliyetleridir. Tarım faaliyetlerinin başarılı bir şekilde yürütülmesi ülkelerin ekonomisine büyük bir katkı sağlamaktadır. Tarım faaliyetlerindeki başarıyı arttırmak için otomasyon sistemleri, makine öğrenmeleri, uzaktan algılama sistemleri gibi bilişim ve teknolojinin bir çok alanı ve bu alanlardaki yöntemler kullanılmaktadır [2].

Gelecek yıllarda Dünya üzerinde nüfusun artması ile birlikte gıda kıtlığına karşı, gıda üretimi artırımı ihtiyacı da doğmaktadır. Gıda kıtlığı ve gıdalardaki güvensizlik gibi sorunlara karşı ülkeler yerel gıda üretimi üzerinde durmaktadır. Ev tarımı, ev bahçeleri, ülkelerin organik ve güvenilir gıda ihtiyacına yönelik başvurduğu ve desteklediği yöntemlerden biridir [3].

Ayrıca ev bahçeleri insanlara psikolojik olarak çeşitli açılardan iyi gelmektedir. Özellikle metropoller gibi aşırı yapılanmış şehirlerde, yeşil alan eksikliğine karşı da ev bahçeleri alternatif çözüm olmaktadır [4].

Ev tarımının performansını da diğer tarım faaliyetlerindeki gibi arttırmaya yönelik çalışmalar yapılmaktadır. Ev tarımı faaliyetlerinde hava faktörü dışarıdaki tarım faaliyetlerine göre daha az etkili olabiliyorken sulama faktörü önemli etkenlerden biridir. Çünkü düzgün sulama verimi etkilemektedir. Diğer yandan ev içi uygulamalarda su tüketiminin minimuma indirilmesi ekonomik açıdan önemlidir [5]. Sensörler ve kontrol, veritabanı teknolojileri, dizayn ve yönetim yazılımları, uzaman sistemler, bulanık mantık ve bilgi iletim sistemleri su yönetimi stratejilerini etkileyen teknolojik gelişmelerdir [6]. Sulama sistemleri sayesinde, sulama işleminde harcanan iş gücü de büyük oranda azalmıştır. Tam otomatik sulama sistemleri sayesinde ise

sulama işlemleri iş gücüne gerek olmadan insanlardan bağımsız bir şekilde başlamakta ve sonlanmaktadır. Tarım faaliyetlerinde toprak nemi, tipi, bitkinin su ihtiyacı gibi etkenler incelenerek uygun sulama yöntemi seçilebilir [7].

NASA'da, Jet Propulsion Laboratuvarında, Huntington botanik bahçesinde toprağa yerleştirilen donanımlı sensörler ile sıcaklık, nem, oksijen ölçümleri yapılmıştır. Ayrıca bitki yetiştirme monitör sistemleri Phytalk Co. İsrail'de uygulanmıştır. Bu sistemler sensörleri kullanarak ölçümler yapmaktadır ve göndermektedir [8].

1.1 Tezin Amacı

Bu çalışmada ev tarımı faaliyetleri için maliyeti az, basit, az yer kaplayan bir sistem üzerinden toprağın nem miktarı tahmin edilmeye çalışılmıştır. Toprak nemini mümkün olduğu kadar az sensör ile tahmin etmek için enterpolasyon yöntemleri kullanarak genel yüzeye ait nem bilgisi oluşturulmaya çalışılmıştır. Yaygın kullanılan enterpolasyon yöntemlerinden IDW, Spline, Kübik, Doğrusal deneyerek ölçüm sonuçları karşılaştırılmıştır. Daha önce yapılan çalışmalara ek olarak IDW yöntemi üzerinde farklı algoritmalar önerilerek, tahmin başarısı arttırmaya çalışılmıştır.

1.2 Literatür Araştırması

İnsan kontrolü olmadan otomatik olarak bitki yetiştirilmesi için toprak nemi, hava sıcaklığı, ışık gibi faktörlerin otomasyonla yönetilmesi gerekmektedir. Eğer otomatik hale getirilmezse yetiştirme faaliyetleri için sürekli olarak insanların denetim yapması gerekecektir [9].

Bitki yetiştirmeye yönelik otomasyon sistemlerinde, kontrol altında tutulmak istenen veriler toplanır ve ilgili birimlere aktarılır. Yazılan algoritmaya göre bilgiler karşılaştırılarak uygun işlemler gerçekleştirilir [10].

Günlük sıcaklık, radyasyon, nem değerleri gözlemlenerek önceden tanımlanmış algoritmalar test edilmiştir. Aynı ayrı ayrı test edilen algoritmalar daha sonra birleştirilmiştir [11].

Toprak nemi de otomasyonlarda kontrol altında tutulması gereken önemli bir faktördür. Toprak neminin tahmin edilmesi ve otomasyonlara eklenmesi için, nem tahminine yönelik çok çeşitli araştırmalar yapılmıştır. Yapılan araştırmalarda farklı enterpolasyon teknikleri, farklı donanımlar kullanılmıştır.

Toprak nemi ve organik madde tahmini yapmak üzere NIR toprak sensörü kullanılmıştır. Bu sensör ile toprak yüzeyinde ve toprak altında ölçümler yapılmıştır. Toplanan veriler çoklu doğrusal regresyon kullanılarak analiz edilmiştir [12].

Başka bir yöntem olarak İleri Mikrodalga Tarama Radyometrisi (Advanced Microwave Scanning Radiometer - AMST-E), toprak neminin gözlemlenmesi için yeni bilgi kaynağı sağlamaktadır. Mikrodalga gözlemler, dielektrik sabiti üzerinde nemin etkisine ve toprağın yayılımına hassastır. Yüzeyin pürüzlü olması ve bitki örtüsü mikrodalga gözlemlerin toprak nemine hassasiyetini (duyarlılığını) azaltır. Frekans artarsa, bitki örtüsü ve yüzey etkilerini daha belirgin hale gelir bu yüzden düşük frekanslar tercih edilmektedir [13].

Mikrodenetleyici kontrollü damla sulama sistemi konusu üzerine yapılan çalışmada, 16F84 mikrodenetleyicisi kullanılmıştır. Yapılan bu sisteme nem sensörleri ile nem algılama devresi eklenmiştir ve toprağın nem bilgisine göre sistem çalışarak otomatik sulama faaliyetleri başlatılmıştır [14].

Uzaktan kontrollü otomatik sulama sistemi çalışmasında PIC 16F877 üzerinde çalışan yazılım, topraktan aldığı nem değerini bitkinin nem ihtiyacı ile karşılaştırarak sulama motorunu çalıştırmaktadır. Bu sistemin çalışmasında, modem kullanılmış ve uzaktan kontrolü sağlanmıştır. Yapılan önceki çalışmalar toprak nemini ölçerek otomatik sulamanın, sulama problemlerini çözülebileceğini ve nem algılayıcı otomatik sulama sistemlerinin %50 oranında su tasarrufu sağladığını ortaya koymuştur [15].

Bitkilerin köklerine yerleştirilmiş sensörler (DS1822-VH400) ile, toprağın sıcaklık ve nem bilgilerini kablosuz ağ üzerinden bir web uygulamasına ileten sistem oluşturulmuştur. Mikrodenetleyici üzerinde çalışan, toprak nemi ve sıcaklığı eşik değerlerine göre geliştirilen, algoritma programlanmıştır. 136 gün boyunca test edilen sistem, geleneksel sulama yöntemlerine göre %90'a varan su tasarrufu sağlamıştır [16].

Toprak nemi hidrolojik, biyolojik, biyokimyasal gibi bir çok alanda önemlidir. Toprak neminin mekansal dağılımı sürdürülebilir su kullanımı açısından çok önemlidir. Toprak neminin noktasal ölçümüne dair çok fazla çalışma yapılmıştır. Fakat yüzelsel olarak toprak neminin dağılımı önemlidir. Bunun için enterpolasyon yöntemleri gereklidir. Sıradan Kriging(Ordinary Kriging) OK, Ters Mesafe Ağırlıklandırma (Inverse Distance Weighting) IDW, Global Polinom İnterpolasyonu (Global Polynomial Interpolation) GPI, Yerel Polinom İnterpolasyonu (Local Polynomial Interpolation) LPI ve Radyal Temelli Fonksiyon (Radial Basis Function) RBF yöntemleri denenmiştir ve hatalar analiz edilerek en doğru yöntem için karşılaştırılmıştır. Henan bölgesinde en uygun enterpolasyon yöntemi olarak Sıradan Kriging öne çıkmıştır [17].

Çin’de, toprak nemini tahmin etmek için enterpolasyon yöntemlerinin karşılaştırıldığı bir çalışma yapılmıştır. Bu çalışmada toprak örnekleri toplanmıştır. 3 gün süren toprak örneği toplama çalışmaları esnasında yağmur yağmamıştır ve 153 noktadan toprak örneği alınmıştır. Toprak örnekleri 10 cm, 20 cm, 40 cm, 60 cm, 80 cm ve 100 cm derinliklerden alınmıştır. 153 noktadan alınan toprak örneği veri kümesi, 102 nokta data seti, 51 nokta test seti olarak iki alt kümeye ayrılmıştır. Bu verilere Sıradan Kriging (Ordinary Kriging) OK, Ters Mesafe Ağırlıklandırma (Inverse Distance Weighting) IDW, Doğrusal Regresyon Modeli (Linear Regression Model) LR, Regresyon Kriging (Regression Kriging) RK olmak üzere, 4 farklı enterpolasyon yöntemi ayrı ayrı uygulanmıştır. Sonuçlar, MAPE ve RMSE ile değerlendirilmiştir. Yapılan bu araştırmaya göre performans olarak, Regresyon Kriging (Regression Kriging) RK bu bölgede en başarılı sonuçları vermiştir [18].

1.3 Hipotez

Enterpolasyon yöntemleri kullanılarak yapılmış birçok çalışmada farklı sonuçlar elde edilmiş, yapılan çalışmalarda kullanılan enterpolasyon yöntemleri arasında performans karşılaştırmalarında başarı sıralaması değişmiştir. En çok kullanılan yöntemlerden biri olan IDW yönteminde performansı arttırarak sonucu daha doğru bir şekilde tespit etmek için, algoritmaya eklemeler yapılarak iki farklı algoritma önerilmiştir.

İlk algoritmada, merkez noktasına göre belirlenen iki farklı yarıçap değerine göre IDW hesaplaması yapılmaktadır. Uzaklık olarak girilen yarıçap değerleri aralığında olan noktaların tahmini yapılmaktadır.

Alınan sensör verileri ve yapılan deneylerde bazı sensör bölgelerinde sulama yapılmıştır. Bu sulama zamanlarından dolayı o bölgede bulunan sensörlerin değerlerinde düşüşlerin olduğu gözlemlenmiştir. Bu sensörler hesaplama dahil edildiğinde, ani olarak değerleri değiştiği için tahminin doğruluk oranını etkilemektedir. Bu sebeple ikinci algoritmada ise, değeri ani değişen veya değeri diğer sensörlerden okunan değerlerin çok altında olan sensörler tespit edilerek, bu tespit edilen sensörlerin IDW hesaplamasında olan etkisi azaltılmaktadır.





2. ENTERPOLASYON YÖNTEMLERİ

2.1 Ters Mesafe Ağırlıklı Yöntem (Inverse Distance Weighted - IDW)

Jeostatistiksel olmayan Ters Mesafe Ağırlıklı Yöntem sıklıkla enterpolasyon işlemlerinde kullanılmaktadır. Komşu noktalara göre tahmin ürettiği için lokal bir yöntemdir. Şekil 2.1’de yöntemin uygulanışı gösterilmektedir. Değeri bilinmeyen noktaların tahmini, noktaların birbirine olan uzaklığına göre ağırlıklandırılarak hesaplanmaktadır. Tahmin edilecek noktaya, mesafe olarak yakın olan noktaların ağırlığı fazla tutulmaktadır [19].

IDW yönteminde tahmin edilecek noktaya, en yakın noktalara fazla ağırlık verilir. Tahmin edilecek noktadan uzaklaştıkça ağırlığın etkisi azalmaktadır.

$$Z^*(x_0) = \sum_{i=1}^n w_i Z(x_i) \quad (2.1)$$

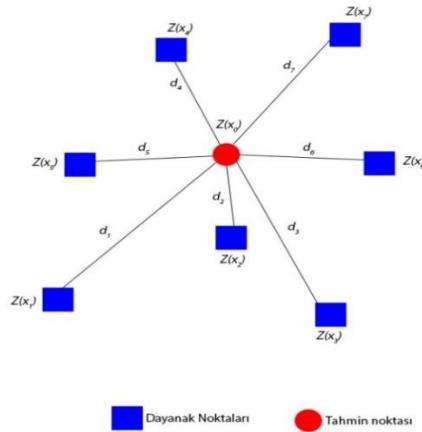
$Z^*(x_0)$: Değeri tahmin edilecek nokta

$Z(x_i)$: Değeri bilinen dayanak noktaları

w_i : Ağırlıkların karelerinin tersinin toplamı

d : Dayanak noktası ile tahmin noktası arasındaki uzaklık

$$w_i = \frac{1/d_i^2}{\sum_{i=1}^n 1/d_i^2} \quad (2.2)$$



Şekil 2.1 : IDW hesaplamasının şekil üzerinde gösterimi.

2.2 Polinom Yüzeyleri ile Enterpolasyon

Bir yüzey için enterpolasyon yapılmak istendiğinde kullanılan yöntemlerdir. Polinomların dereceleri yüzey hakkında bilgi vermektedir. Polinom yüzeyleri dikgen ve dikgen olmayan polinomlar şeklinde ikiye ayrılır.

Dikgen polinomlarla enterpolasyon işlemlerinde, yüksek dereceden polinomlar genellikle iki değişkenli olarak kullanılır. Doğrusal, kuadratik, kübik yüzeyler için yapılan işlemlerde her birinde farklı dayanak noktası kullanılır ve lineer, quadratik, kübik yüzeyler elde edilir.

Dikgen olmayan polinomlarda enterpolasyon işlemlerinde, Bi-lineer, Bi-quadratik, Bi-kübik yüzeyler için yapılan işlemlerde her birinde farklı bilinmeyenli bi-linner, bi-quadratik, bi-kübik yüzeyler elde edilir [20].

Dikgen polinomlarla enterpolasyonda,

$$N(x, y) = \sum_{k=0}^n \sum_{\substack{j=k-i \\ i=0}}^k a_{ij} x^i y^j \quad (2.3)$$

Dikgen olmayan polinomlarda ise,

$$N(x, y) = \sum_{i=0}^n \sum_{j=0}^n a_{ij} x^i y^j \quad (2.4)$$

denklemleri kullanılmaktadır [21].

2.3 En Küçük Kareler Yöntemine Göre Enterpolasyon, Ekstrapolasyon

En küçük kareler yöntemi, tahmin, spektral analiz, fonksiyonel analiz gibi bir çok alanda kullanılmaktadır. Yapılan gözlemlerin rastgele hatalar içerdikleri ve birbirinden bağımsız olduğu kabul edilmektedir. En küçük karelerle kollokasyon yönteminde sinyal, fazla sayıda sinyal değişkeninin fonksiyonu olarak ifade edilmektedir [20].

2.4 Kriging Yöntemleri ile Enterpolasyon

Coğrafi istatistikte yaygın kullanılan enterpolasyon yöntemleridir. Bu yöntemler bir çok alanda kullanılmaktadır. Jeoloji, çevre, maden, meteoroloji gibi alanlarda bu yöntemlerle yapılmış çalışmalar mevcuttur. Hesaplamalarda, IDW yöntemine benzer şekilde çevredeki noktaların değerlerinden yararlanılmaktadır.

Genel denklem,

$$Z_p = \sum_{i=1}^n W_i Z_i \quad (2.5)$$

Z_p : p noktasının aranan değeri

W_i : Her Z_i 'ye atanacak ağırlık değerleri

Z_i : Z_p 'nin hesaplanmasında kullanılan noktaların değeri

Bu denklemde Z değerleri, aralarındaki uzaklık veya konumları bilinen noktalardaki rastgele değişkenleri göstermektedir. W_i , ağırlıkların belirlenmesi bu yöntemde önemlidir çünkü enterpolasyon değerini etkilemektedir. Tahmin hatası minimum olacak şekilde ağırlıklar belirlenir. Bu Kriging yöntemine B.L.E.U (Best Linear Unbiased Estimator) denir.

Farklı alanlarda kullanılan, Simple Kriging, Ordinary Kriging (OK), Universal Kriging, Block Kriging, Indicator Kriging, Disjunctive Kriging, Cokriging gibi yöntemler vardır.

OK yaygın kullanılan yöntemlerdendir. Bu yöntemde bilinmeyen noktalar tahmin edilirken, değişkenlerin durağan ve ortalamalarının sabit olduğu kabul edilir. OK'de kullanılan denklem,

$$\hat{Z}(X_0) = \sum_{i=1}^N W_i Z(X_i) \quad (2.6)$$

$\hat{Z}(X_0)$: X_0 noktasına ait Kriging değerini,

$Z(X_i)$: Değişkenlerin X_i noktasında gözlenen değerleri,

W_i : $Z(X_i)$ 'ye karşılık gelen ağırlık değerlerini,

N : $\hat{Z}(X_0)$ 'nin Kriging tahmininde kullanılacak nokta sayısını

göstermektedir. Bu yöntemin IDW'den temel farkı, sadece mesafelerin ağırlıklarına bağlı olmayıp, model yarıvariogramda da bağlı olmasıdır [22].

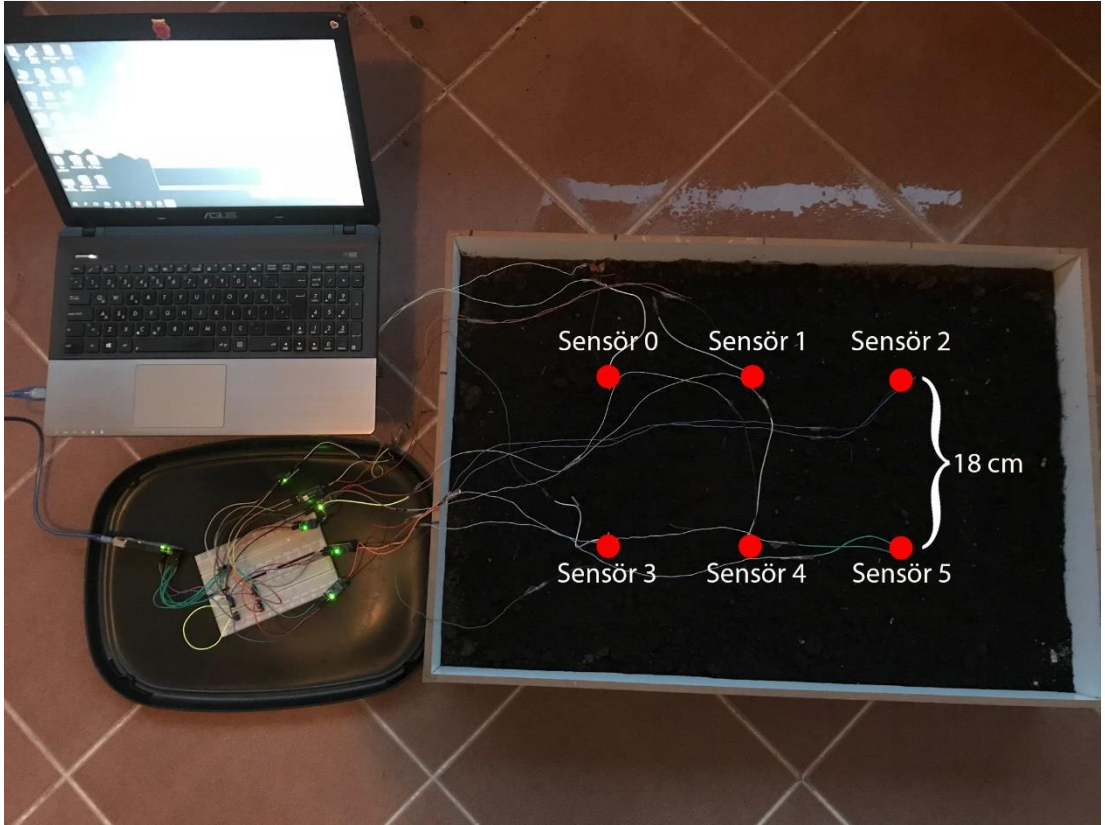
Evrensel kriging yöntemi ise bölgesel değişkenlerin durağanlık göstermediği durumlarda kullanılmaktadır. Bu yöntemde trendler hesaba katılmakta, 1. veya 2. dereceden polinomlar yada fourier trend olarak alınmaktadır. Enterpolasyon işlemi kriging yöntemi ile yapılırsa, karesel ortalama hata minimuma iner. Kriging yönteminin diğer yöntemlere göre avantajı esnek bir yöntem olması ve kestirim hatasının oranını değerlendirmeye olanak sağlamasıdır [20].



3. MATERYAL ve YÖNTEM

Bu çalışmanın gerçekleştirilebilmesi için farklı donanım ve yazılım platformlarında çalışmalar yapılmıştır.

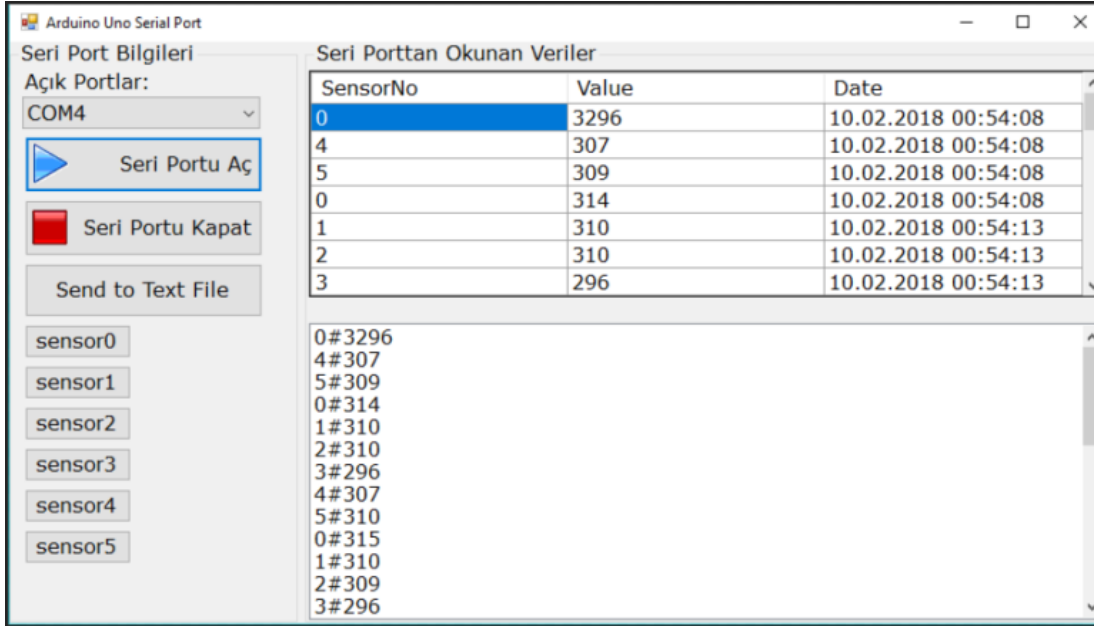
Çalışmanın donanım kısmında, Arduino ve bread board üzerine kurulan devreye 6 adet toprak nem sensörü Şekil 3.1’de görüldüğü gibi bağlanmıştır. Bu sensörler toprak yüzeyinde yaklaşık 5 cm derinliklere, birbirlerinden 18 cm eşit uzaklıklarda toprağa batırılmış ve sensörlerden veri alınmıştır.



Şekil 3.1 : 72x54 cm’lik saksıya 18 cm’lik aralıklarla yerleştirilen 6 adet sensörlerden verilerin alınması ve breadboard üzerinde kurulu olan arduino devresi ile seriporttan bilgisayara aktarımı.

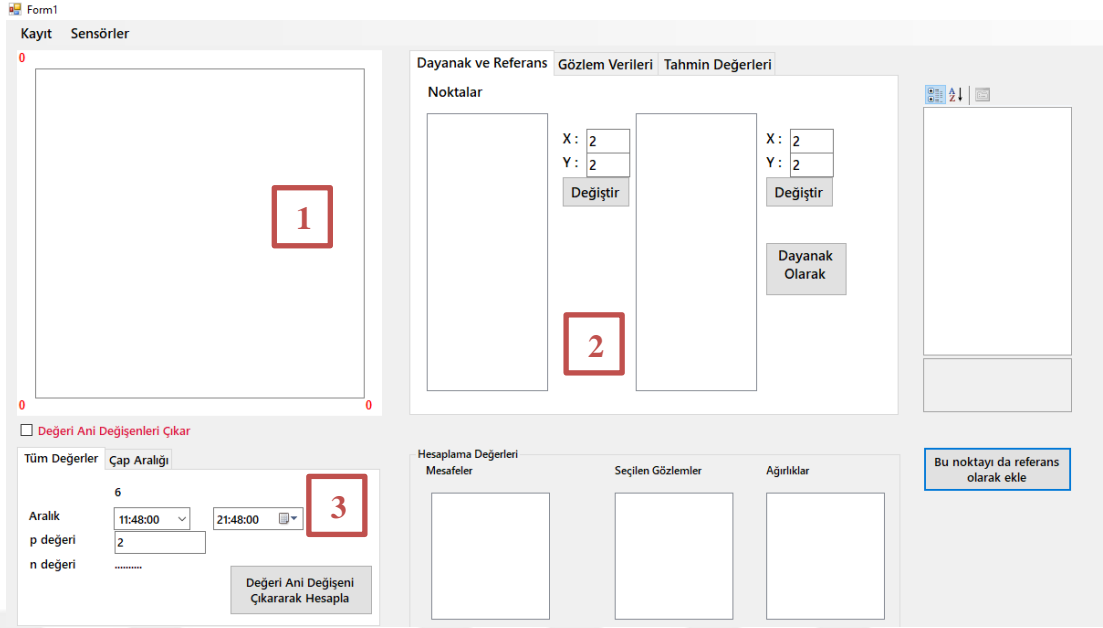
Ölçülen veriler öncelikle arduino arayüz programından okunmuştur. Arduino’dan okunan veriler, Şekil 3.2’de ara yüzü görüldüğü gibi Visual Studio’da hazırlanan program kullanılarak “.txt” dosyasına kaydedilmiştir. Hazırlanan bu programda bilgisayarın açık portu belirlendikten sonra, seri port açılır ve veriler listenlemeye

başlar. Arayüzde görülmekte olan butonlar ile alınan verilerin tümü birden bir “.txt” dosyasına kaydedilebilirken, ayrı ayrı sensörlerin değerlerini de farklı “.txt” dosyalara kaydetmek mümkündür.



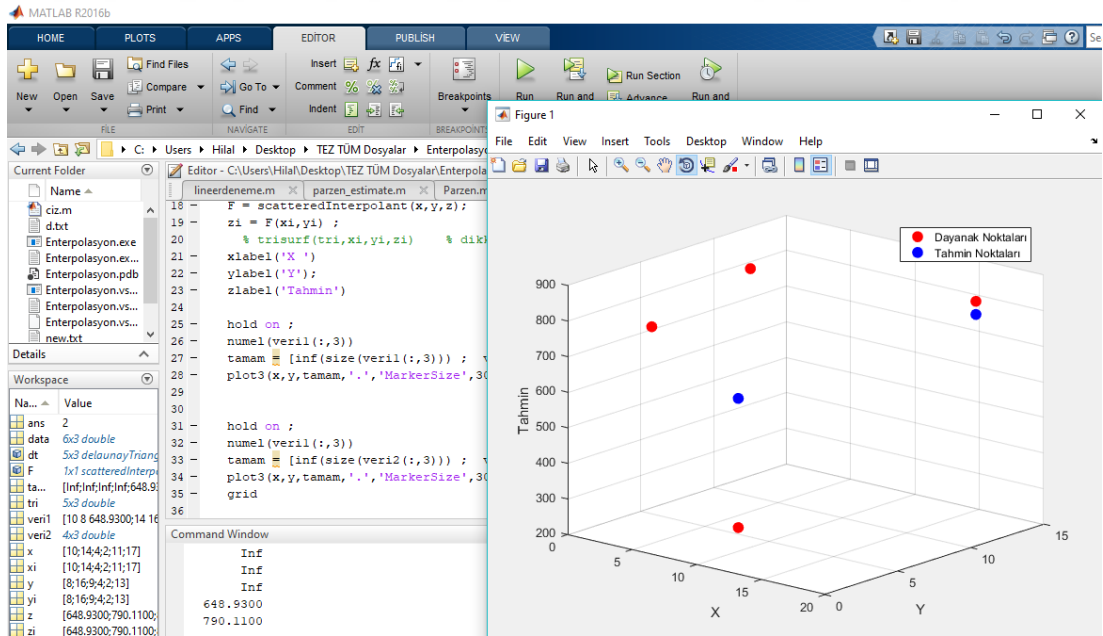
Şekil 3.2 : Microsoft Visual Studio’da hazırlanan program arayüzü ile verilerin alındığı seriportların açılması, her sensörden alınan verilerin ilgili sensör düğmelerinden “.txt” dosyalara ayrı ayrı kaydedilmesi ve tüm sensörlerden gelen verilerin sensör numarası, değeri, tarihi okunarak ve “.txt” dosyalara aktarımı.

Oluşturulan “.txt” dosyaları Microsoft Visual Studio’da yazılan başka bir programa yüklenerek burada farklı IDW algoritmaları verilere uygulanmıştır. Şekil 3.3’de bahsedilen programın arayüzü görülmektedir.



Şekil 3.3 : Microsoft Visual Studio’da hazırlanan programdan IDW algoritmalarının çalıştırılması. 1 numaralı gösterilen kısım sensör konumları, 2 numaralı kısım “.txt” dosyadan alınan veriler, 3 numaralı kısım alınacak verilerin zaman aralığının seçimi.

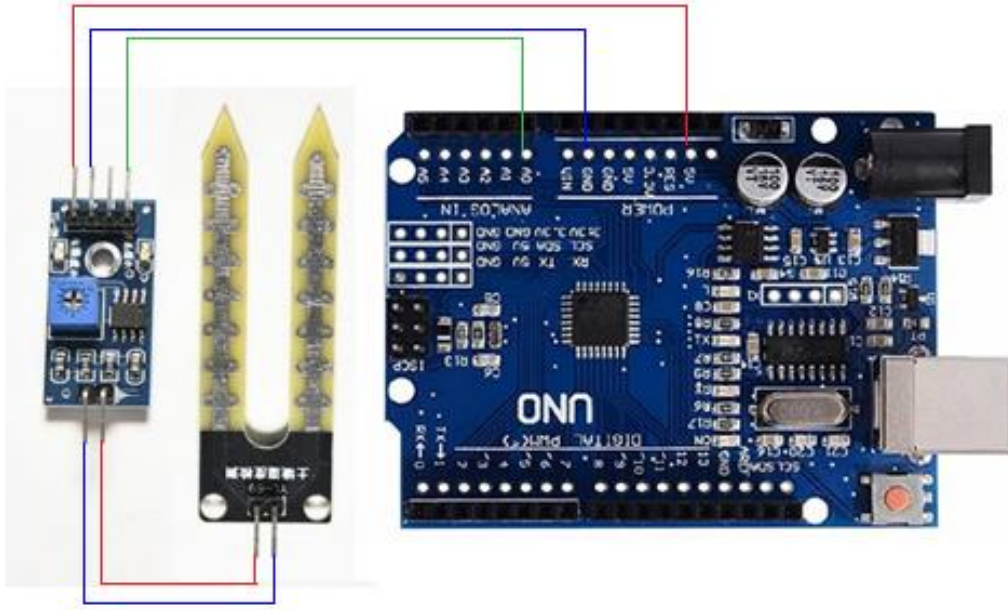
Programda yapılan hesaplamalardan sonra Şekil 3.4’de görüldüğü gibi Matlab programı ve Excel programı kullanılarak, yüzey, dağılım, hata gibi grafikler, sonuçları daha iyi gözlemleyebilmek için elde edilmiştir.



Şekil 3.4 : IDW algoritmalarından alınan sonuçlara göre Matlab programında grafiklerin oluşturulması. Kırmızı noktalar ile Dayanak, mavi noktalar ile Referans noktalarının gösterimi.

3.1 Arduino

Arduino, ATmega328 tabanlı bir mikrodenetleyici karttır. Üzerinde 14 adet dijital Giriş/Çıkış pini bulunmaktadır. Bunların 6'sı PWM (Pulse Width Modulation ya da Darbe Genişlik Modülasyonu) çıkışı olarak kullanılabilir. Ayrıca üzerinde 6 adet analog giriş vardır. Arduino, Arduino yazılımı (IDE) kullanılarak programlanabilir. C dili kütüphaneleri kullanılmaktadır. Şekil 3.5'te Arduino Uno kartı ve sensör bağlantısı görülmektedir.



Şekil 3.5 : Arduino kartı, LM393 sensörü ve prob bağlantısının sağlanması.

Arduino kartların, Uno, Leonardo, 101, Esplora, Mikro, Mega, Ethernet, Mini, Nano gibi çeşitleri vardır. Geliştiricilere ihtiyaçlarına göre uygun olan kartı seçerek çalışma imkânı sunmaktadır. Arduino Uno genellikle en çok tercih edilen kart tipidir [23]. Arduino Uno kartının Teknik özellikler Çizelge 3.1'de görülmektedir.

Çizelge 3.1 : Arduino Uno : Teknik Özellikler.

Mikrodenetleyici	ATmega328P
Çalışma gerilimi	5V
Giriş Voltajı (önerilen)	7-12V
Giriş Voltajı (limit)	6-20V
Dijital I / O Pimleri	14 (6 tanesi PWM çıkışı sağlar)
PWM Dijital I / O Pimleri	6
Analog Giriş Pimleri	6
I / O Pimi başına DC Akım	20 mA
3.3V Pin için DC Akım	50 mA
Flash Bellek	0,5 KB olan 32 KB (Atmega328P)
SRAM	2 KB (Atmega328P)
EEPROM	1 KB (Atmega328P)
Saat hızı	16 MHz
Uzunluk	68,6 mm
Genişlik	53,4 mm
Ağırlık	25 g

Bu kartlar sayesinde, bilgisayar-sensör arasında veri alışverişi için, veri işleme kartına gerek kalmamaktadır. Bilgisayar ile seri port veya USB üzerinden haberleşebilen bu kartlar sinyal gönderimini ve alımını sağlayabilmektedir [24].

Projede 6 adet sensör için giriş pinine, güç pinlerine ve seri port ile iletişim sağlayacak kanala ihtiyaç duyulmaktadır. Bu çalışmada Arduino uno kullanılmasının sebebi analog giriş-çıkış sayısının yeterli olması, çalışma geriliminin uygun olması, USB girişi ile bilgisayara direk bağlanabilmesi ve maliyetinin az olmasıdır.

3.2 Matlab

Dünya üzerinde milyonlarca mühendis ve bilim adamı çalışmalarında analiz ve hesaplama yapmak, grafik oluşturmak, veri elde etmek gibi amaçlar için Matlab kullanmaktadır. Paralel hesaplama, gömülü sistemler, hesaplamalı biyoloji, robotik, veri analizi, otomotiv, endüstriyel otomasyon, havacılık ve savunma gibi bir çok alanda kullanılan bir yazılımdır [25].

Projenin farklı adımlarında Matlab yazılımı kullanılmıştır. Çeşitli hesaplamalar yapmak ve grafikler elde edilmiştir. Kübik Enterpolasyon, Spline Enterpolasyon ve Doğrusal enterpolasyon tahminleri için Matlab üzerinden çalışılmıştır. Ayrıca geliştirilen yazılım üzerinden hesaplanan IDW yönteminin sonuçları da Matlab'a aktararak dağılım yüzeyleri gözlemlenmiştir.

3.3 Visual Studio

Microsoft tarafından geliştirilen Visual Studio, tümleşik bir geliştirme ortamıdır. Masaüstü uygulamaları, web uygulamaları, oyun ve mobil gibi bir çok platform için yazılım geliştirmeyi ve geliştirilen yazılımları test etme ortamı sunmaktadır. Visual studio, C#/VB, C++, Javascript, Python gibi kullanılan dilden bağımsız olarak, bilgi ipuçları ile kod yazmada kolaylık sağlamaktadır [26].

Bu projede ilk olarak verilerin alınarak düzenli bir şekilde bilgisayarda kayıt edilmesi aşaması için Visual Studio kullanılarak bir masaüstü uygulaması tasarlanmıştır. Alınan veriler txt dosyalara kayıt edilmiştir. Diğer aşamada ise alınan veriler yine Visual Studio'da hazırlanan ayrı bir program arayüzü ile okutulmuştur ve verilere IDW yöntemi uygulanmıştır.

Programın farklı algoritma seçeneklerine göre bölümleri bulunmaktadır. Önerilen algoritmalar da aynı program üzerinde çalıştırılarak, veriler kayıt edilmektedir.

3.4 Uygulanan Yöntem

Yapılan araştırmalarda, farklı araçlar ve enterpolasyon yöntemleri ile toprak nemi tahmini üzerine çeşitli çalışmalar yapıldığı gözlemlenmiştir. Coğrafi bilgi sistemlerinde en çok kullanılan yöntem, IDW yöntemidir.[27] İncelenecek enterpolasyon yöntemi olarak Ters-uzaklık ağırlıklandırma (IDW) yöntemi seçilmiştir.

Bu çalışmada yaygın kullanılan IDW yönteminin hesaplaması ile birlikte iki farklı algoritma önerilmiştir. Bu algortimaların tümü aynı yazılım arayüzünden, farklı bölümlerden çalıştırılabilecek şekilde programlanmıştır.

3.4.1 Ters mesafe ağırlıklı yöntem (Inverse distance weighted – IDW)

Ters-uzaklık ağırlıklandırma yöntemi (IDW), yaygın kullanılan bir yöntemdir. IDW yönteminde ölçüm yapılırken komşu noktalardan tahmin yapıldığından, lokal bir yöntemdir. Bu yöntemde noktaların birbirlerine olan uzaklıklarının ağırlıkları kullanılarak, değeri bilinmeyen noktaların tahmin edilmesi amaçlanır [19]. Bu yöntemde göre tahmin edilecek nokta için, yakınlarındaki noktaların, uzaktaki noktalara göre ağırlığı fazladır.

$$Z^*(x_0) = \sum_{i=1}^n w_i Z(x_i) \quad (3.1)$$

$Z^*(x_0)$: Deęeri tahmin edilecek nokta

$Z(x_i)$: Deęeri bilinen dayanak noktaları

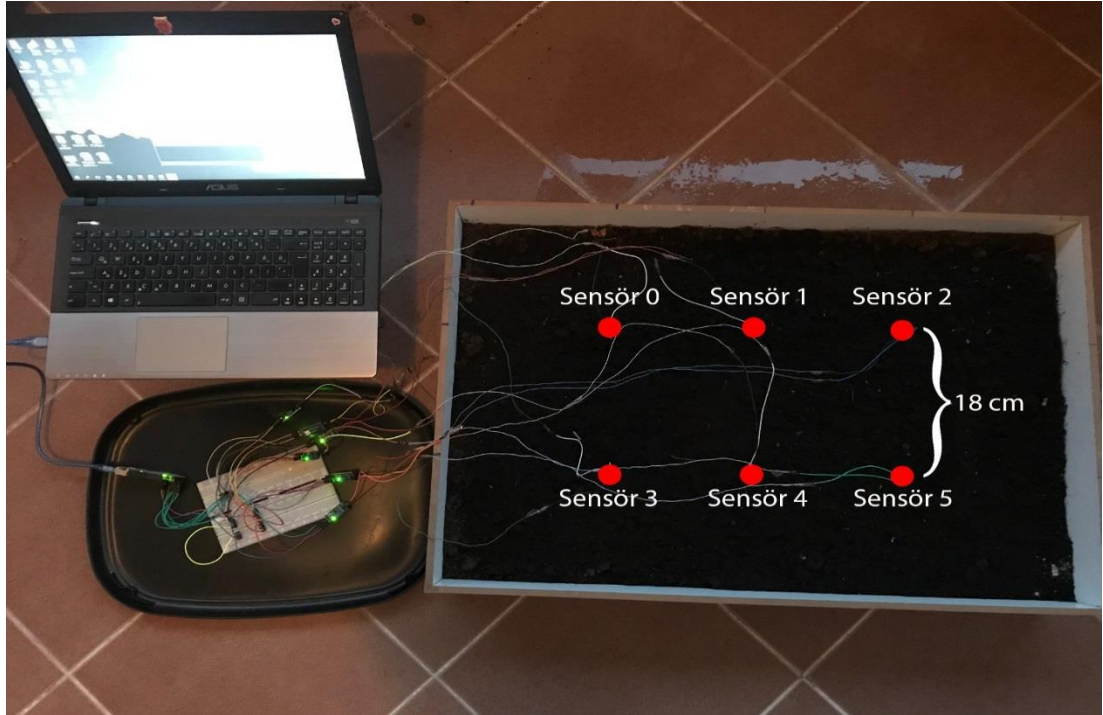
w_i : Aęırlıkların karelerinin tersinin toplamı

d : Dayanak noktası ile tahmin noktası arasındaki uzaklık

$$w_i = \frac{1/d_i^2}{\sum_{i=1}^n 1/d_i^2} \quad (3.2)$$

Şekil 3.6'da IDW yönteminin hesaplanması için toprak üzerinde sensörlerin konumları ve birbirlerine olan uzaklıkları görülmektedir. Herhangi bir sensörün bulunduğu nokta, tahmin noktası olarak belirlenebilmektedir.

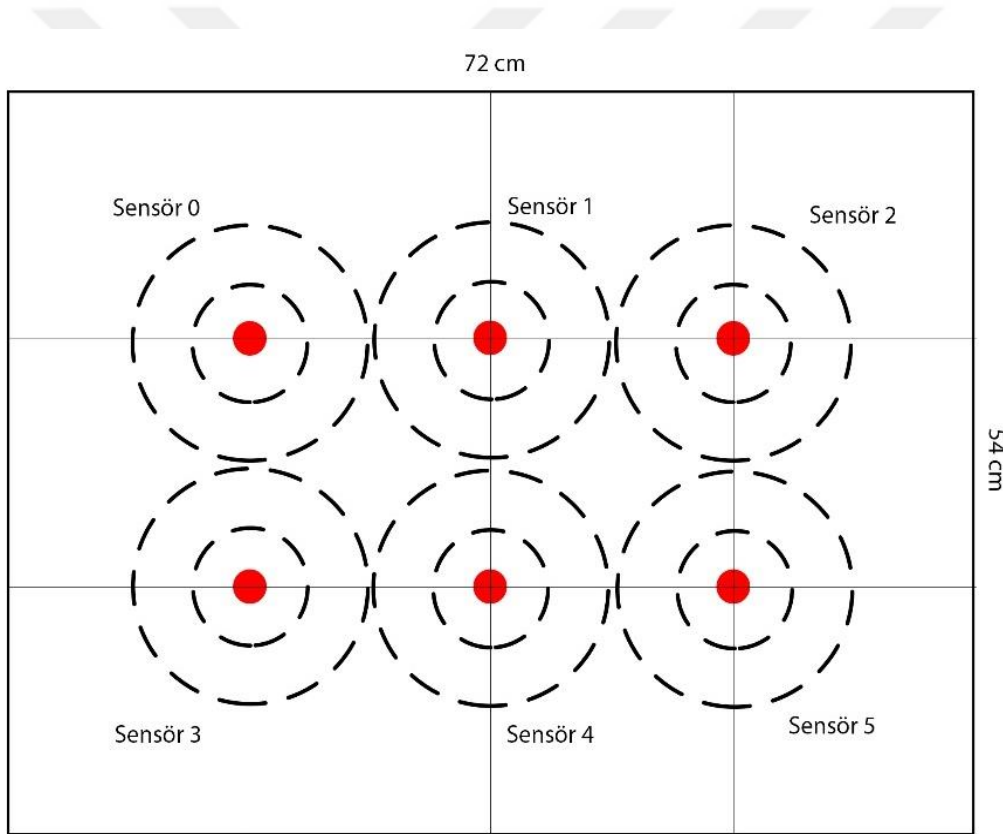
Tahmin noktası olarak seçilen noktanın, tüm noktalara uzaklığı ayrı ayrı hesaplanmalıdır. Sensörlerin yatay ve dikey aralıkları 18 cm olarak belirlenmiştir.



Şekil 3.6 : IDW yönteminin hesaplanması için sensör konumları ve uzaklıklarının gösterimi.

Şekil 3.7’de topraktan verileri almak için kurulan devre ve sensörlerin yerleşimi görülmektedir. Sensörler 72x54 cm boyutlarındaki özel yaptırılmış saksıya düzenli aralıklarla yerleştirilerek, sensörlerin etki alanı hesaplandı. Deneysel olarak ölçümler gerçekleştirilmiştir.

Ölçüm yapılırken sensörler aynı hizada ilk olarak 3 cm’lik uzaklıklarla yerleştirilmiş 1/8 litre su, 0. ve 1. Sensor arasına dökülmüş, tüm sensörlerden alınan veriler gözlemlenmiştir. Daha sonra sensörler arasındaki mesafe sırasıyla 6 cm, 9 cm, 12 cm, 15 cm, 18 cm’ye çıkarılmış aynı işlem tekrarlanmıştır. 15 cm’ye kadar olan tüm ölçümlerde sensörler sulamanın yapıldığı sensörlerle aynı değeri vermiştir. Bu ölçüm sensörlerin etki alanı mesafesinin 15 cm olarak bulunduğunu göstermiştir.



Şekil 3.7 : Toprağa yerleştirilen sensörlerin etki alanlarının belirlenmesi sonucunda sensörlerin birbirinden 18 cm uzaklıkta olacak şekilde saksının içinde konumlandırılması.

Kullanılan LM393 sensörünün ucuna bağlı olan problar toprağa saplanarak, problarda oluşan gerilime göre ölçüm yapılmaktadır. Alınan değerler 0-1023 arasındadır. Sensörden alınan değerlerin istatistiklerine göre toprağın nem durumu Çizelge 3.2’de gösterilmiştir.

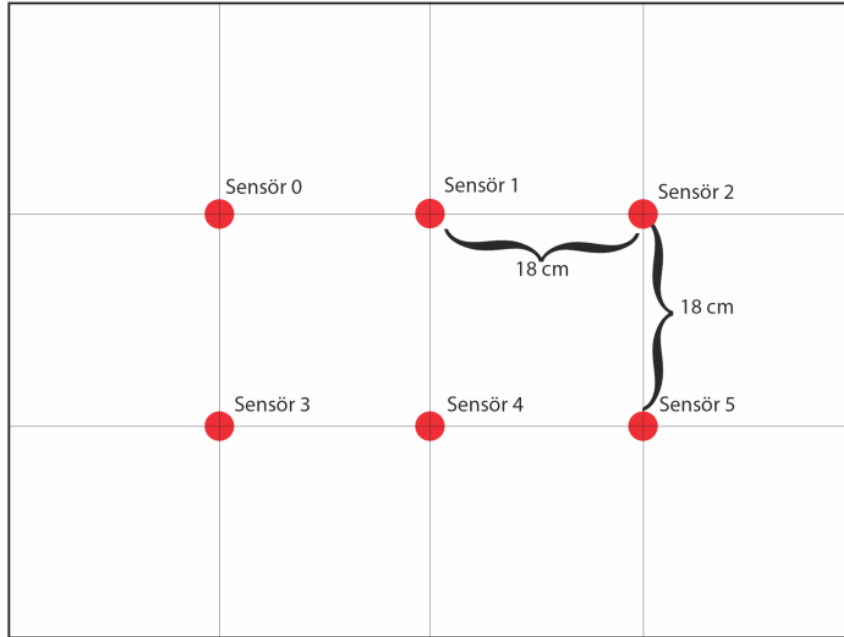
Çizelge 3.2 : Toprak Nem Miktarı Aralıkları.

Toprak Nem Miktarı	
Değer Aralığı	Nem Durumu
0-300	Nemli
300-900	Orta Nemli
900-1023	Kuru

Toprak nemi, sadece topraktaki su miktarına bağlı olmayıp, toprak içerisindeki bazı diğer maddelere de bağlıdır. Deneyimizde ev tarımında yaygın kullanılan torflu toprak kullanılmıştır. Çizelge 3.2’de verilen değerler, toprağın durumlarına göre, ölçümlerde alınan yaklaşık değerler ile oluşturulmuştur.

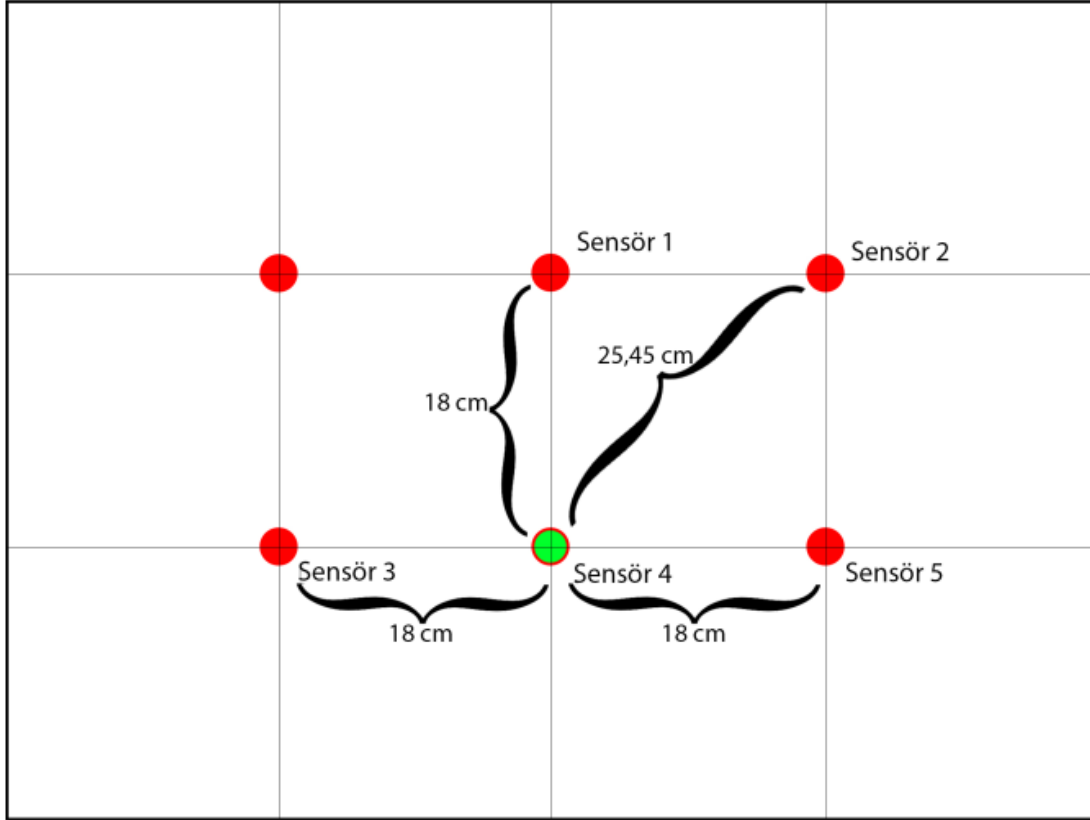
3.4.2 Ters-mesafe ağırlıklı yöntemin verilere uygulanması

6 adet LM393 sensöründen ölçülen değer, 1 adet Arduino Uno üzerinden bilgisayara aktarılmıştır. Aktarılan verilerin tümü analiz edilmiştir. Sensör konumları, Şekil 3.8’de görüldüğü gibi tüm deneyler için belirlenmiştir. IDW yönteminin uygulanacağı dayanak noktaları ve tahmin noktası seçilmiştir. Projenin bu kısmında elimizde değerleri olan sensör noktalarından bir tanesi tahmin noktası, diğerleri ise dayanak noktaları olacak şekilde hesaplama yapılmıştır.



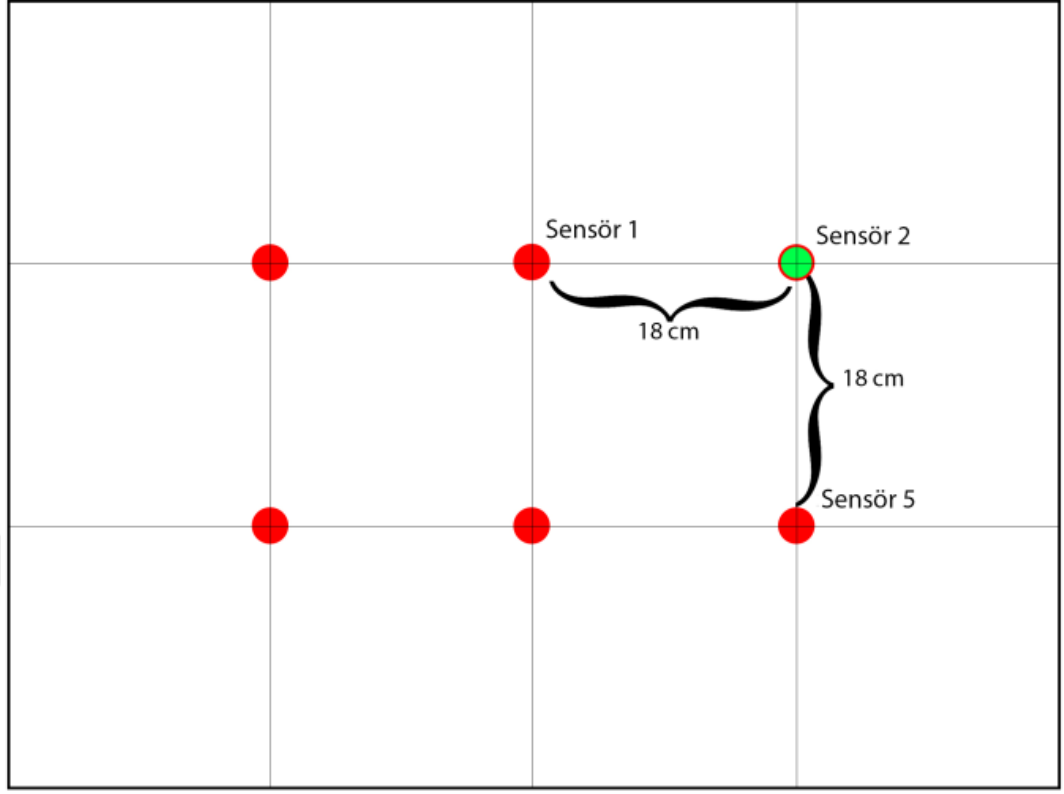
Şekil 3.8 : 72x54 cm’lik saksıya sensörlerin yerleşimini gösteren deney ortamının modellenmesi ve sensörlerin konumlandırılması.

Her biri birbirinden 18 cm uzaklıkta olacak şekilde 6 adet sensör alana yerleştirilmiştir. Deneylerde farklı tahmin noktaları, farklı sayıda dayanak noktasından alınan verilere göre ve tahmin edilecek noktaya olan uzaklıklarının ağırlıklarına göre belirlenerek, IDW yöntemi uygulanmıştır. Şekil 3.9’da ve Şekil 3.10’da farklı senaryolar görülmektedir. IDW yönteminin performansını ölçmek için değerlendirilen sensörler kullanılmıştır. Şekil 3.9’da tahmin edilecek 4 numaralı sensörün değeri, belirlenen 1, 2, 3, 5 numaralı sensörlerin değerlerine ve 4 numaralı sensörden buldukları uzaklığa göre hesaplanmıştır. Burada bütün dayanak noktalarında ağırlık, uzaklığın karesi olarak alınmıştır. Alınan sonuçlar daha önceden 4 numaralı sensörden alınan değeri doğrulamıştır.



Şekil 3.9 : IDW yönteminin uygulanacağı dayanak noktaları ve tahmin noktası. Sensör 4’e uzaklığı gösterilen noktaların değerleri ve uzaklıkları kullanılarak, Sensör 4’ün değerinin tahmin edilmesi.

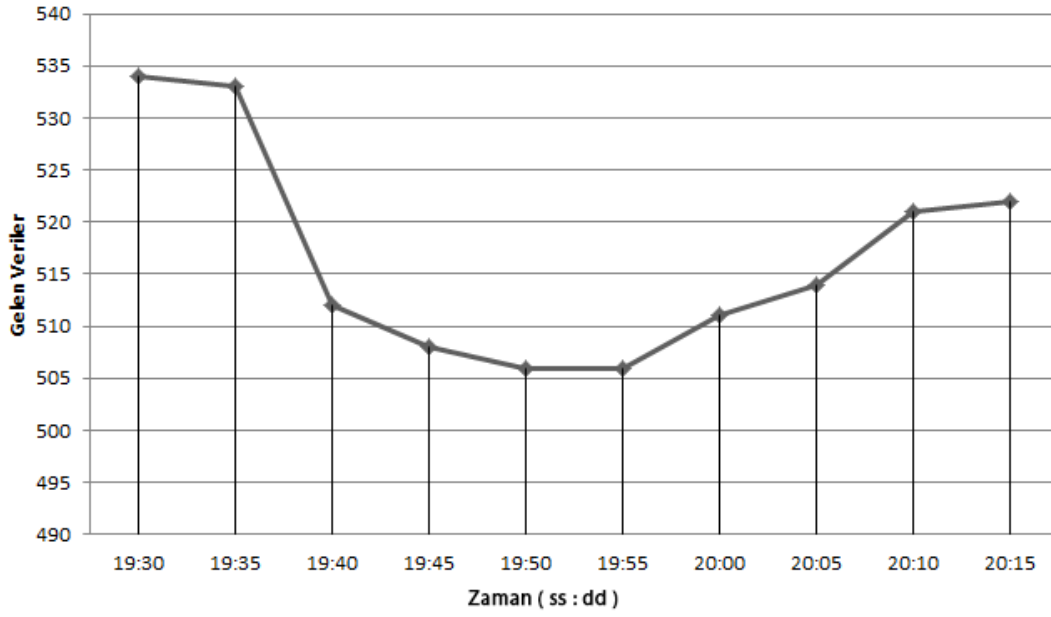
Şekil 3.10’da ise dayanak noktaları azaltılmış, tahmin edilecek nokta olarak 2 numaralı sensör seçilmiştir. Yapılan işlemler sonucunda, tahmin edilen verinin, daha önceden ölçülmüş 2 numaralı sensörün değerini tam olarak yansıtmadığı tespit edilmiştir.



Şekil 3.10 : IDW yönteminin uygulanacağı dayanak noktaları ve tahmin noktası. Sensör 2'ye uzaklığı gösterilen noktaların değerleri ve uzaklıkları kullanılarak, Sensör 2'nin değerinin tahmin edilmesi.

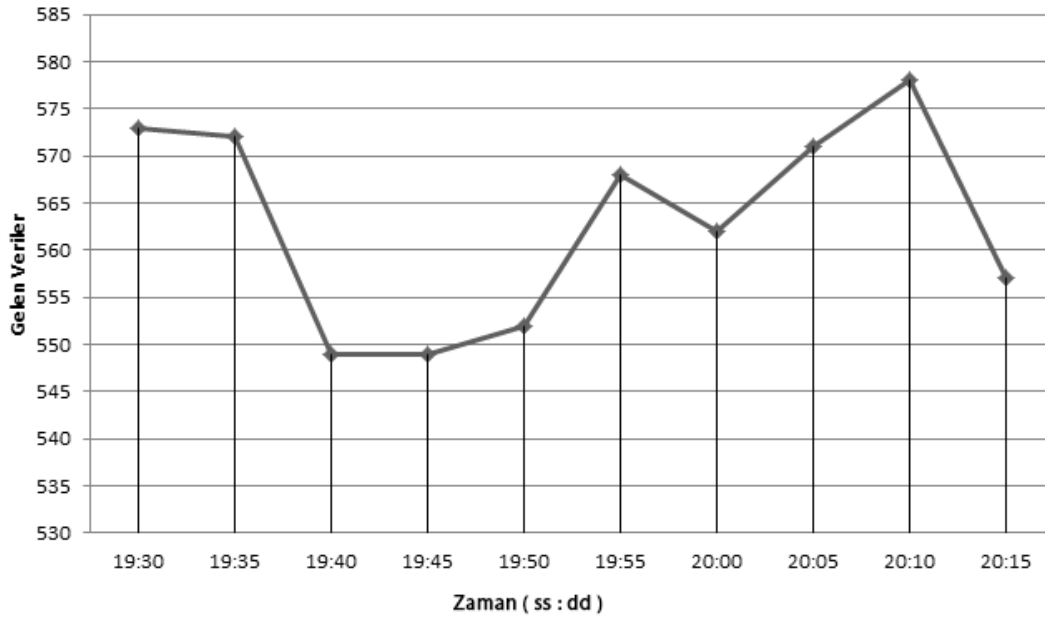
Şekil 3.11, 3.12, 3.13, 3.14, 3.15, 3.16'da sensörlerden alınan değerlerin değişimi gözlemlenmektedir. Özellikle 2 numaralı sensör bölgesinde sulama işlemi yapıldıktan sonra alınan ölçüm değerindeki değişim grafikte açıkça görülmektedir. 2 numaralı sensöre yakın olan sensör bölgelerinin de bu sulama işleminden etkilendiği yine sensörlerin ölçüm-zaman grafiklerinde görülmektedir.

0. Sensörün Veri-Zaman Grafiği



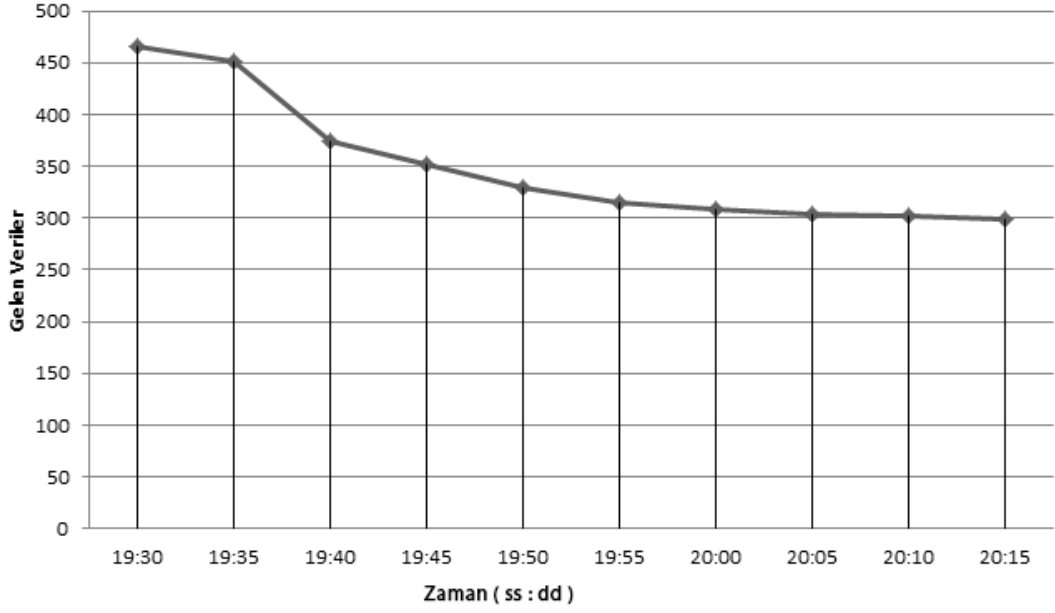
Şekil 3.11 : 0. Sensörden deney süresinde alınan verilerin zamanla dağılımı. 19:40 zamanında, 2 numaralı sensör bölgesinde yapılan sulamanın, 0.sensörün okunan değerlerine etkisi.

1. Sensörün Veri-Zaman Grafiği



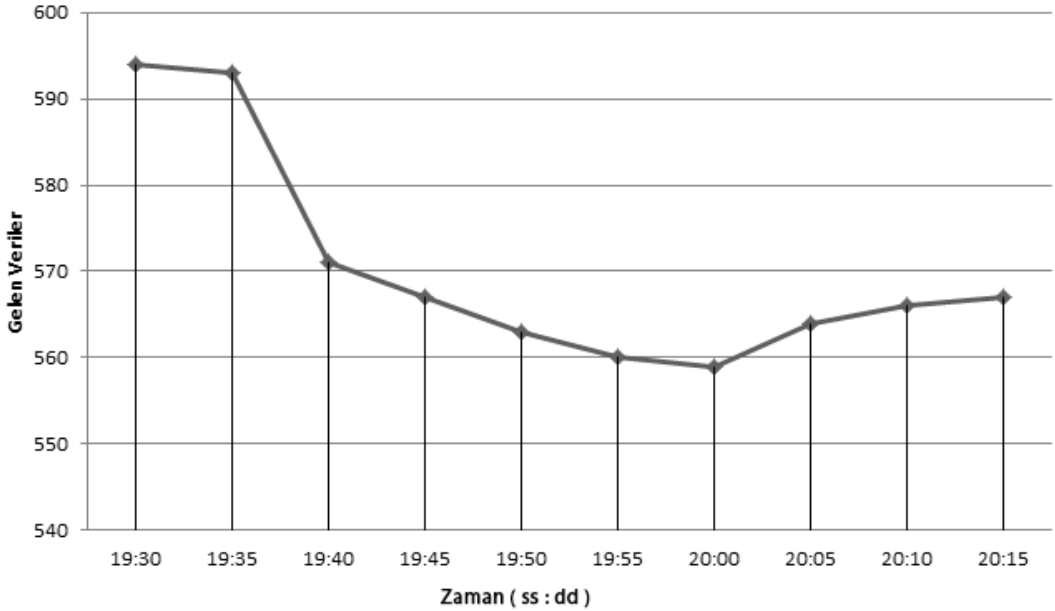
Şekil 3.12 : 1. Sensörden deney süresinde alınan verilerin zamanla dağılımı. 19:40 zamanında yapılan sulamanın 1.sensörden okunan değerlere etkisi.

2. Sensörün Veri-Zaman Grafiği



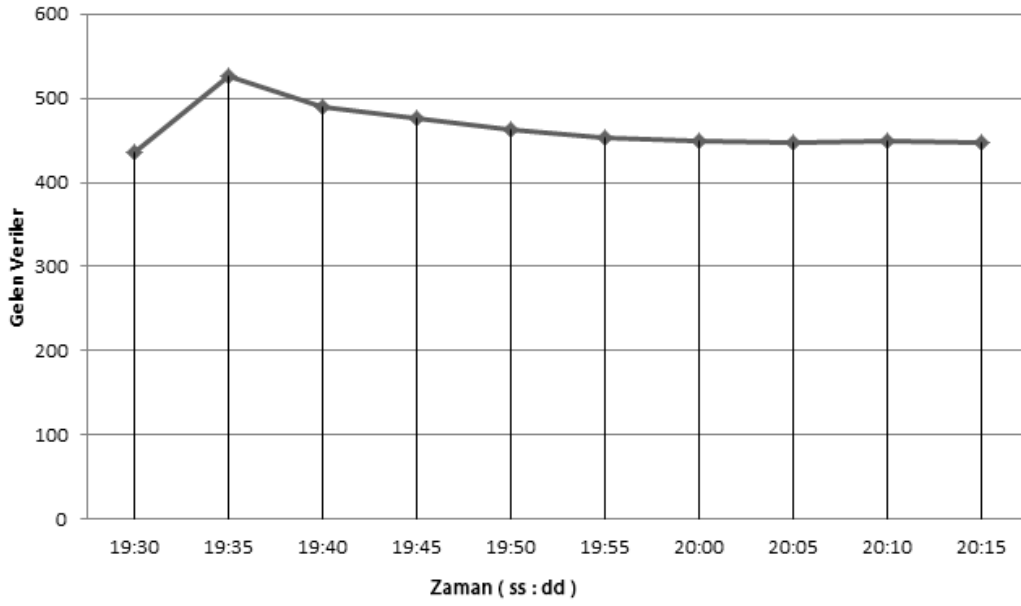
Şekil 3.13 : 2. Sensörden deney süresinde alınan verilerin zamanla dağılımı. 19:40 zamanında 2. sensör bölgesinde sulama yapıldığı için, okunan değerlerin sürekli olarak düştüğünün gösterimi.

3. Sensörün Veri-Zaman Grafiği



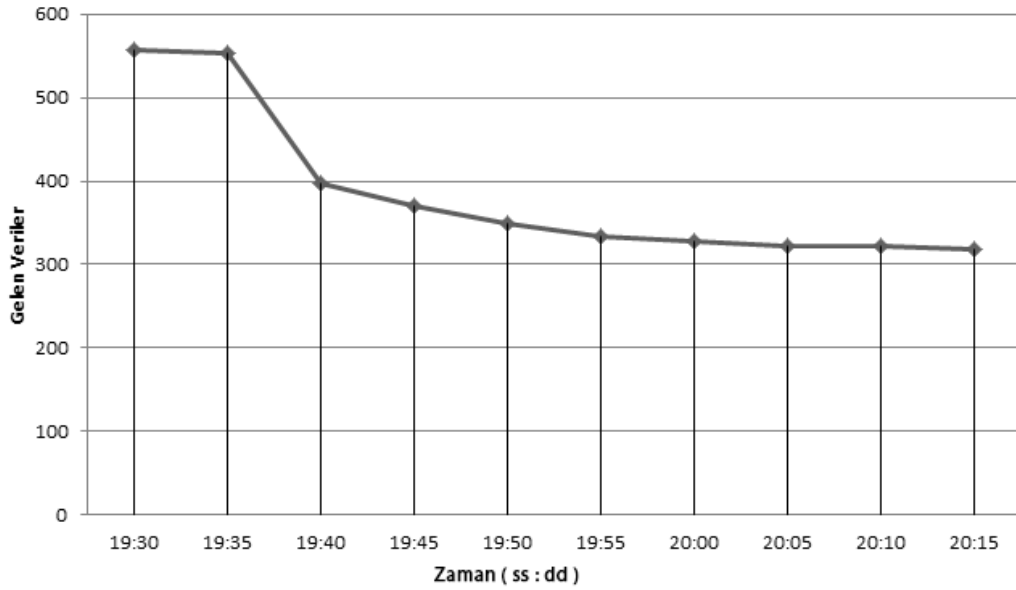
Şekil 3.14 : 3. Sensörden deney süresinde alınan verilerin zamanla dağılımı. 19:40 zamanında yapılan sulama etkisi ile 3. Sensörden okunan değerlerin düşmesi.

4. Sensörün Veri-Zaman Grafiği



Şekil 3.15 : 4. Sensörden deney süresinde alınan verilerin zamanla dağılımı. 19:40 anında 2. Sensör bölgesinde yapılan sulamanın, 4. Sensör değerlerine etkisinin daha az olduğunu görülmektedir. 4. Sensör ve 2. Sensör arasındaki mesafenin etkisi.

5. Sensörün Veri-Zaman Grafiği



Şekil 3.16 : 5. Sensörden deney süresinde alınan verilerin zamanla dağılımı. 5. sensör mesafe olarak 2. sensöre yakın olduğundan, özellikle 19:40 zamanından sonra 5. sensörden okunan değerler düşmüştür. 5. sensör değerlerinin, 2. sensör değerlerine yakın olduğunun gözlemlenmesi.

Çalışmada topraktan alınan veriler üzerinde yaygın enterpolasyon yöntemlerinden olan, IDW yöntemi denenmiştir. Şekil 3.9’da yapılan ölçümlerde, 4 numaralı sensörün değeri 19:40 anında diğer sensörlerden ve Çizelge 3.3’te görülmekte olan verilere göre 486,99 olarak hesaplanmıştır. Şekil 3.10’da görülen senaryoya göre yapılan ölçümlerde, 2 numaralı sensörün değeri, 20:15 anında Çizelge 3.3’te görülmekte olan diğer sensörlerin verilerine göre 438 olarak hesaplanmıştır.

Çizelge 3.3 : Sensörlerden Alınan Örnek Veri Kümesi.

Sensör No	Zaman									
	19:30	19:35	19:40	19:45	19:50	19:55	20:00	20:05	20:10	20:15
0	534	533	512	508	506	506	511	514	521	522
1	573	572	549	549	552	568	562	571	578	557
2	466	451	375	351	330	315	308	304	302	99
3	594	593	571	567	563	560	559	564	566	567
4	436	527	490	476	463	453	450	448	450	448
5	557	554	397	371	349	334	327	322	322	319

Farklı zaman noktalarında alınan veriler ve seçilen farklı dayanak noktaları ile yapılan incelemelerde ev tipi bitki yetiştirme ortamlarında basit ölçüm ve tahminler için IDW yönteminin başarı sağladığı gözlemlenmiştir. Fakat tahminlerin doğruluğu IDW yönteminde seçilen dayanak noktalarının sayısına ve uzaklıklara verilen ağırlık kuvvetine göre değişkenlik göstermektedir.

3.4.3 Ters-mesafe ağırlıklı yöntemin yarıçap aralığı belirtilerek uygulanması

Yapılan deneysel çalışmalarda uzaklık ve sensörlerin ölçüm mesafelerinin, sonuçlara önemli derecede etki ettiği açıkça gözlemlenmiştir. Bir sensör bölgesinde yapılan sulama yakınındaki sensörün değerlerini de etkilemektedir. Ayrıca uzakta olan sensörlerin tahmin değerine etkisinin az olması ile birlikte, sonuçlar açısından doğruluk oranını düşüren bir etkisi de vardır. Bu sebeple IDW yönteminin başarı oranını arttırmak için önerilen algoritmada, tahmin edilecek noktalar girilen iki farklı yarıçap değerine göre belirlenmektedir. Eğer bir veya daha çok tahmin noktasının merkez noktaya uzaklığı, kullanıcı tarafından girilen yarıçap aralığında değilse, o nokta veya noktalar hesaplamaya dahil edilmemektedir.

Programın arayüzünde Şekil 3.17’de görülmekte olan kısımdan, programın çalışma algoritması değiştirilmektedir. Burada eğer 1. çap aralığı değeri girilirse, bu uzaklıkta bulunan tahmin noktalarının değeri hesaplanmaktadır. 1. ve 2. çap aralığı değerleri girilirse, bu iki çap aralığında kalan tahmin noktaları hesaplamaya dahil edilmektedir.

Şekil 3.17 : Farklı IDW algoritmalarının Microsoft Visual Studio’da çalıştırılması. 1 numara ile gösterilen program arayüzünde bulunan “Çap Aralığı” sekmesinin gösterimi.

3.4.4 Ters-mesafe ağırlıklı yöntemin değeri ani değişenler çıkartılarak uygulanması

IDW yönteminde, bir noktanın değeri 15 cm yakınındaki noktaları etkilemektedir. Fakat bu etkinin oranı çok az olmaktadır. Yapılan deneysel ölçüm çalışmalarında, bazı sensör noktalarında sulama yapılmıştır. Sulama yapılan sensörün değerinde direk büyük oranda bir düşüş olurken, yakınında bulunan sensörün değerinde de sulama noktasında bulunan sensöre göre az da olsa bir düşüş olduğu alınan verilerde gözlemlenmiştir. Değeri aniden düşen sensörler tahmin performansını da düşürmektedir. Çalışmanın bu kısmında değeri ani değişen sensörleri tespit ederek, bu sensörlerin tahmin noktası değerinin hesaplanmasına olan etkisini azaltmak için, programın arka kısmında bir algoritma eklenmiştir.

Bu algoritmaya göre belirlenen azami değerin altında kalan veriler ayırılarak IDW hesaplamasına dahil edilmemektedir. Program arayüzünde önerilen algoritmayı çalıştıran kısım, Şekil 3.18’de görülmektedir. “Değeri Ani Değişenleri Çıkar” etiketli onay kutusu işaretlenirse program anlatılan algoritmaya göre çalışmaktadır.

The screenshot shows the software interface for the IDW algorithm. The main window is titled 'Form1' and contains several panels:

- Map Panel:** A grid of points labeled 0 through 6. Point 1 is highlighted with a red box and a red arrow pointing to it. The grid is bounded by coordinates 0, 18, and 36 on both axes.
- Dayanak ve Referans Panel:** Contains two tabs: 'Gözlem Verileri' and 'Tahmin Değerleri'. Under 'Gözlem Verileri', there is a list of points:
 - 1. Nokta (0 - 0)
 - 2. Nokta (18 - 0)
 - 3. Nokta (36 - 0)
 - 4. Nokta (0 - 18)
 - 5. Nokta (18 - 18)
 - 6. Nokta (36 - 18)
 The '1. Nokta (16 - 10)' is selected. Input fields for X (36) and Y (18) are visible, along with a 'Değiştir' button.
- Hesaplama Değerleri Panel:** Contains three tabs: 'Mesafeler', 'Seçilen Gözlemler', and 'Ağırlıklar'. The 'Seçilen Gözlemler' tab is active, showing a list of points with their coordinates:
 - 0 - 747,493827160494
 - 1 - 746,700579180003
 - 2 - 746,235767940373
 - 3 - 746,638457865931
 - 4 - 745,809065630536
 - 5 - 745,073319122082
- Control Panel:** Includes a checkbox labeled 'Değeri Ani Değişenleri Çıkar' which is checked. Below it are input fields for 'Tüm Değerler', 'Çap Aralığı' (6), 'Aralık' (11:48:00 to 21:48:00), 'p değeri' (2), and 'n değeri' (0). There are 'Hesapla' and 'Değeri Ani Değişeni Çıkararak Hesapla' buttons.
- Other Panels:** A 'kullanılacak Ölçüm' panel with a 'Bu noktayı da referans olarak ekle' button.

Şekil 3.18 : IDW yönteminin değeri ani değişenlere yönelik algoritması. 1 numara ile hesapama algoritmasının değiştirildiği onay kutusunun gösterimi.



4. SAYISAL UYGULAMALAR

Bu bölümde 6 sensörden, yapılan ölçümler sonucunda alınan ve “.txt” dosyasına kaydedilen veriler kullanılarak, kübik, doğrusal ve spline enterpolasyon yöntemleri uygulanmıştır. 30 dakika ara ile alınan verilere göre, farklı enterpolasyon yöntemlerinin nem haritaları oluşturulmuştur. Aynı zaman dilimi için, her bir enterpolasyon yöntemine ait iki adet harita bulunmaktadır. Bu haritalarda üç boyutlu ve iki boyutlu olarak deney ortamının nem dağılımı görülmüştür.

4.1 Kübik Enterpolasyon

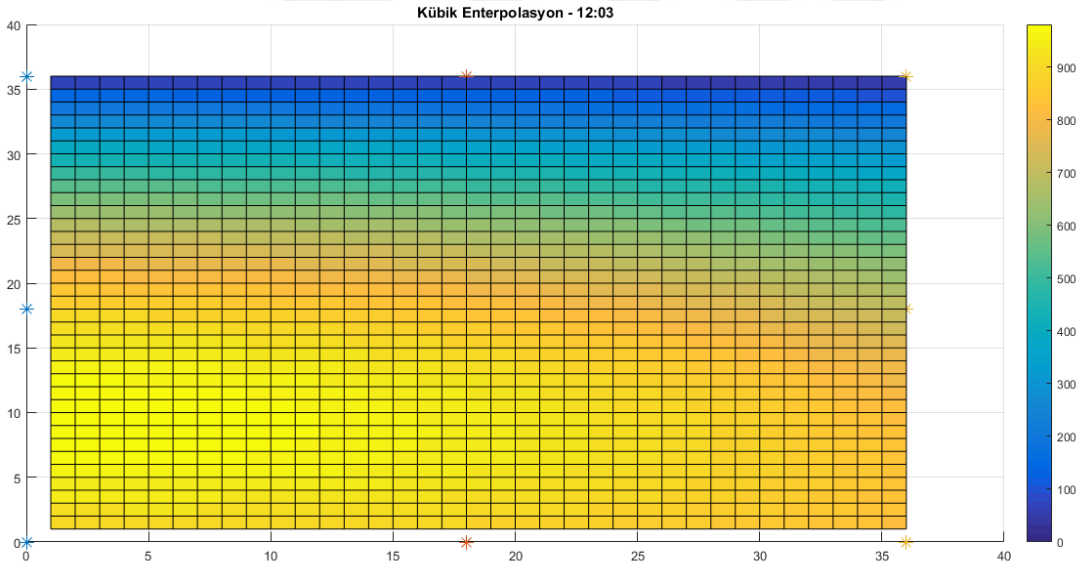
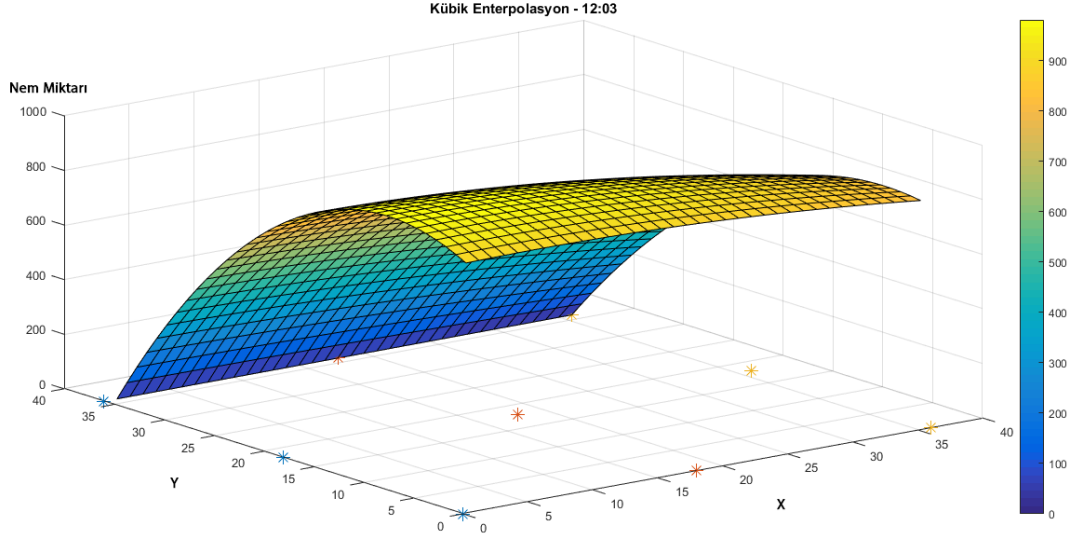
Veri noktaları arasında benzersiz bir dizi kübik polinom oluşturur. Böylelikle dizi noktalarından düzgün eğriler oluşturulabilir. Kübik polinomlardaki katsayı, enterpolasyon için kullanılan ağırlıklardır [28].

$$Z(x_0) = \begin{cases} z_1(x) & \text{if } x_1 \leq x < x_2 \\ z_2(x) & \text{if } x_2 \leq x < x_3 \\ \vdots \\ \vdots \\ z_{n-1}(x) & \text{if } x_{n-1} \leq x < n \end{cases} \quad (4.1)$$

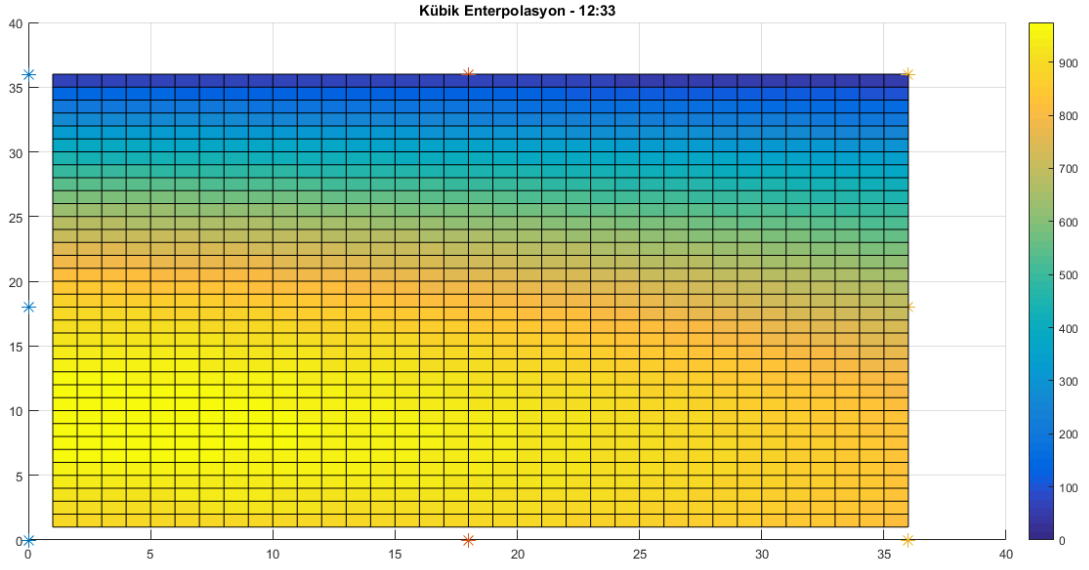
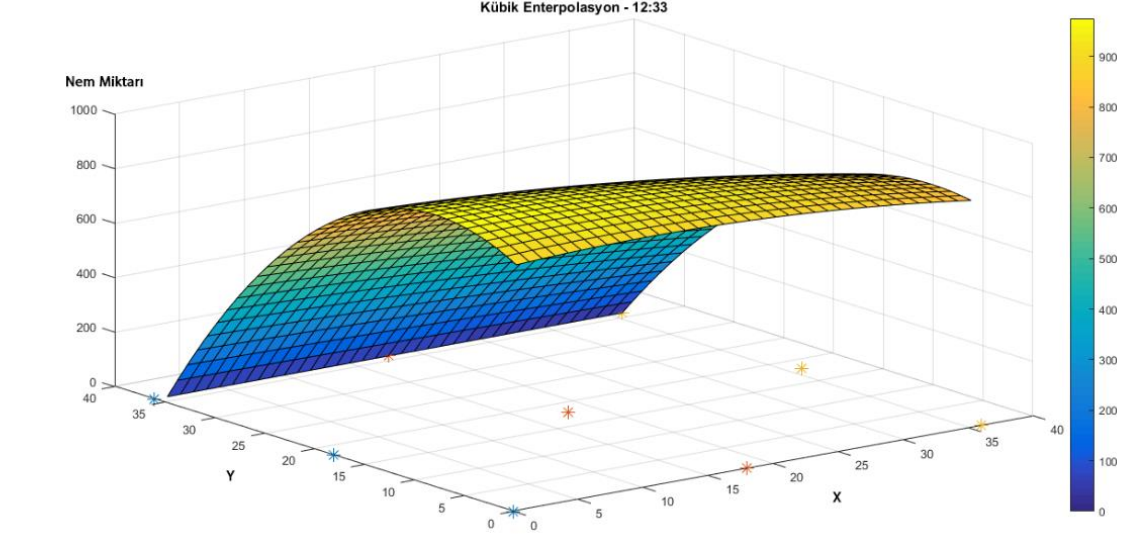
s_i , üçüncü dereceden tanımlanan bir polinomdur.

$$z_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \quad i = 1, 2, \dots, (n - 1). \quad (4.2)$$

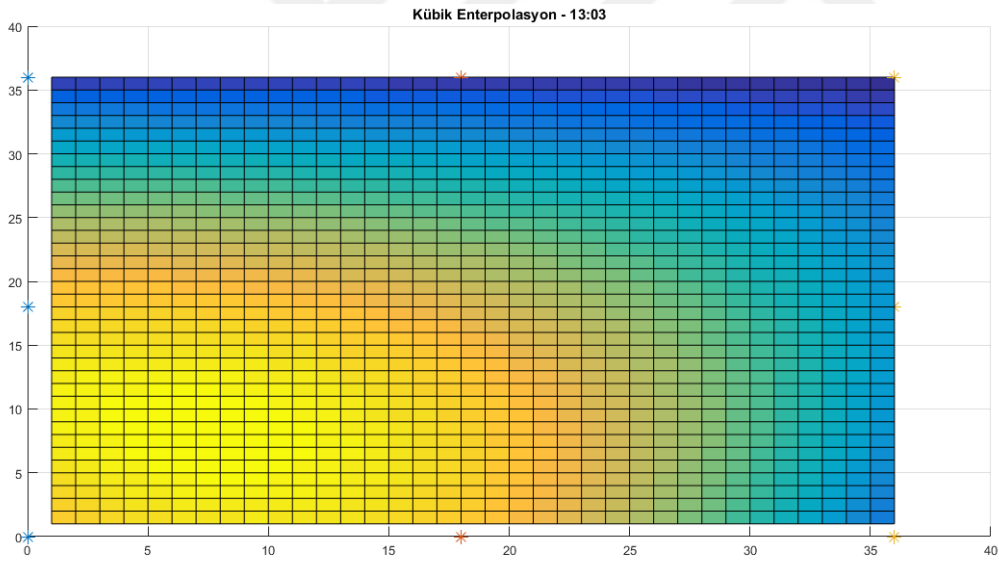
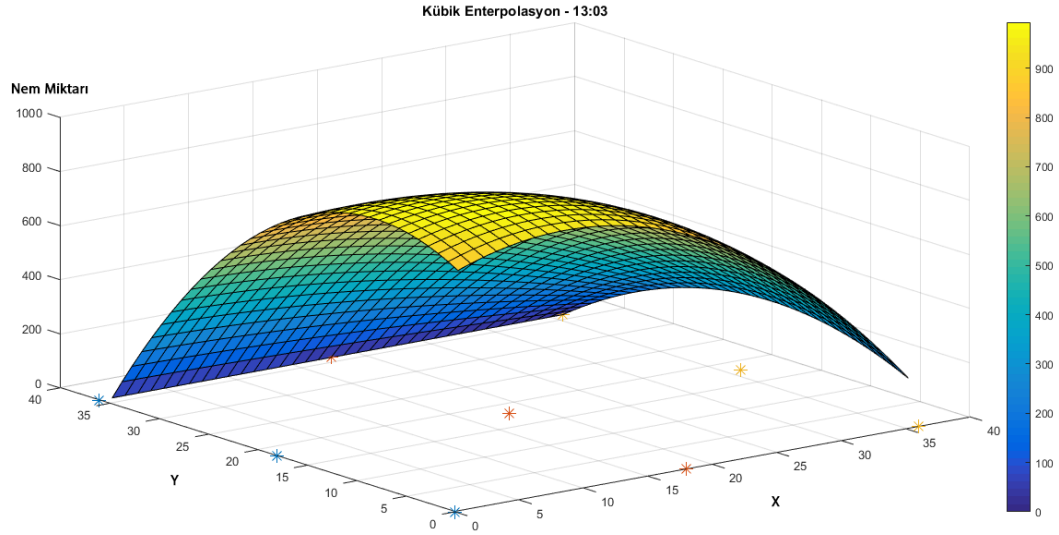
Sensörlerden alınan verilere, kübik enterpolasyon uygulandığında sonuçların dağılımı Şekil 4.1, Şekil 4.2, Şekil 4.3, Şekil 4.4 ve Şekil 4.5'te görülmektedir. Özellikle Kübik Enterpolasyon - 12:33 ve Kübik Enterpolasyon - 13:03 grafikleri karşılaştırıldığında, deney amaçlı sulama yapılan 2 numaralı ve 5 numaralı sensörlerdeki değerlerin düşüşü rahatlıkla gözlemlenmektedir.



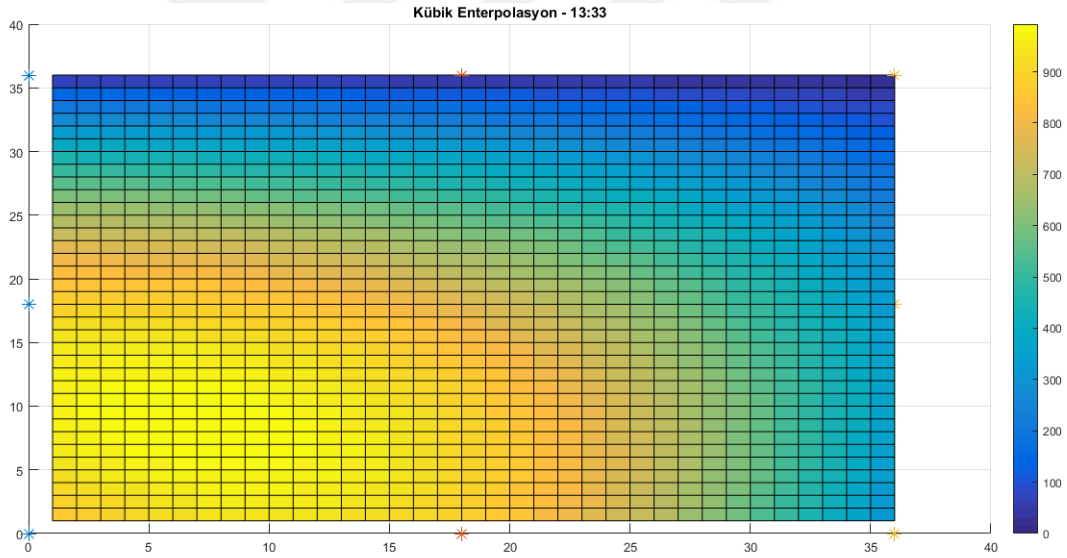
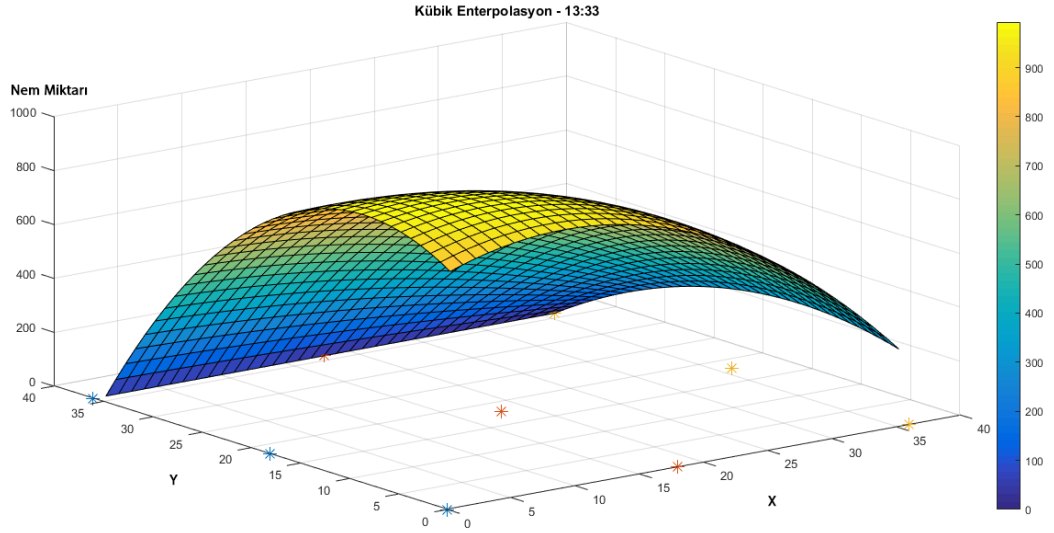
Şekil 4.1 : Küçük Enterpolasyonun verilere uygulanması ile 12:03 anında oluşturulan küçük yüzey grafiği ve nem dağılım haritası.



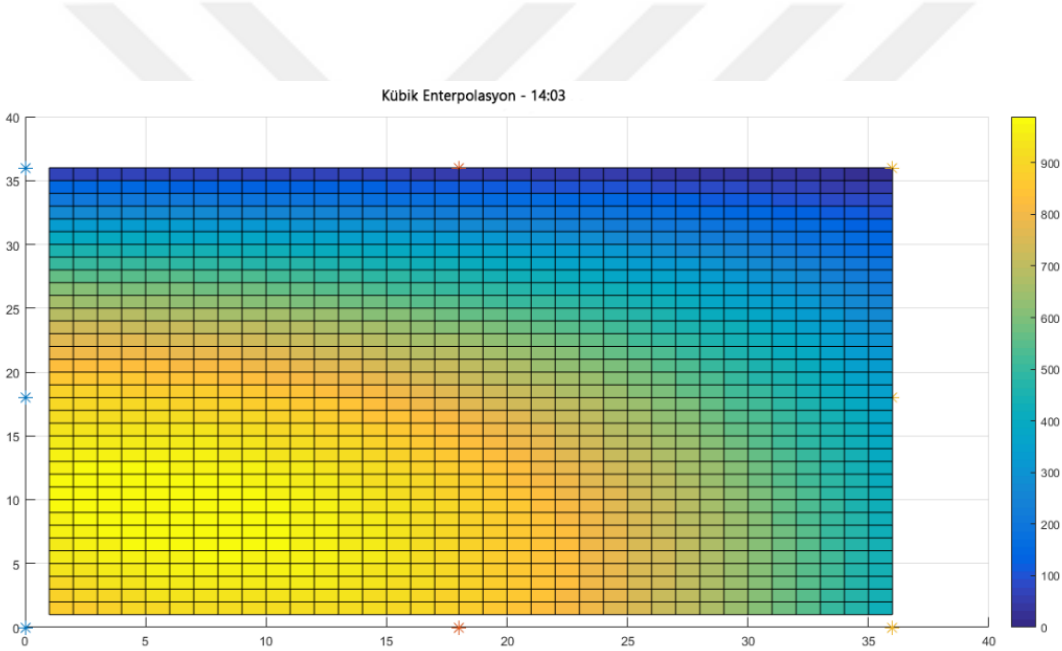
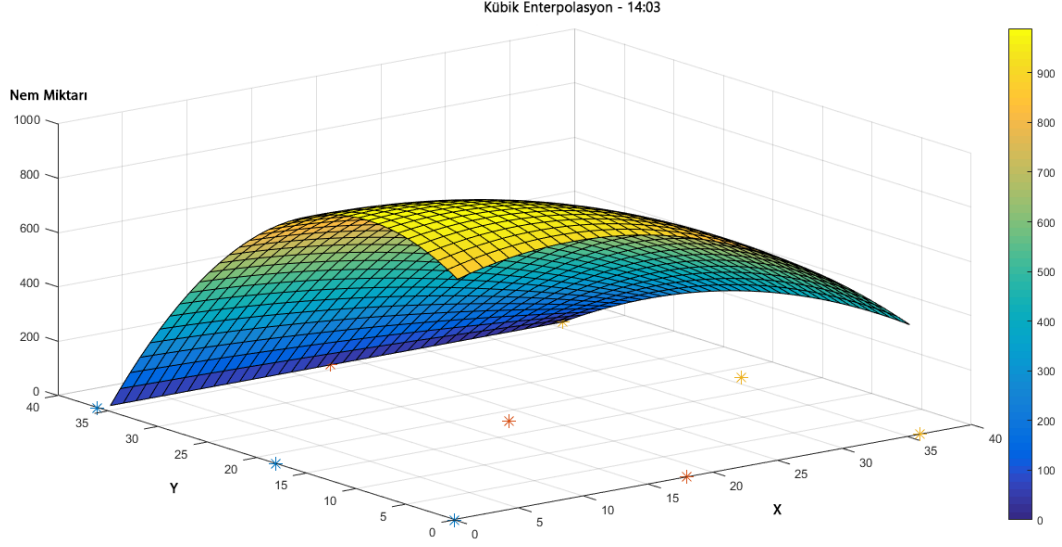
Şekil 4.2 : Kübik Enterpolasyonun verilere uygulanması ile 12:33 anında oluşturulan kübik yüzey grafiği ve nem dağılım haritası.



Şekil 4.3 : Küçük Enterpolasyonun verilere uygulanması ile 13:03 anında oluşturulan küçük yüzey grafiği ve nem dağılım haritası. 12:33-13:03 aralığında yapılan sulamanın grafikte ve nem haritasında gösterilmesi.



Şekil 4.4 : Küçük Enterpolasyonun verilere uygulanması ile 13:33 anında oluşturulan kübik yüzey grafiği ve nem dağılım haritası.



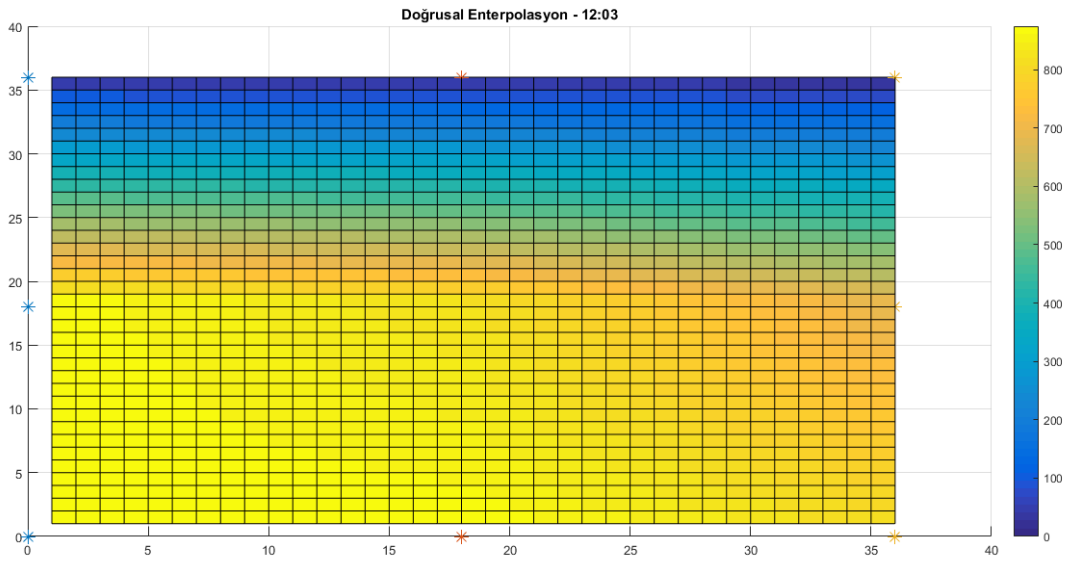
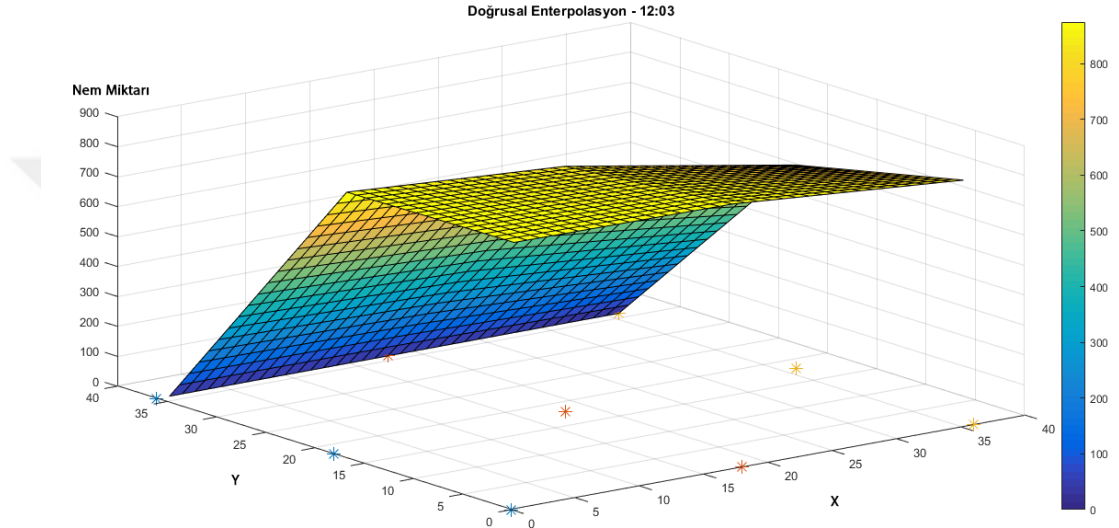
Şekil 4.5 : Kübik Enterpolasyonun verilere uygulanması ile 14:03 anında oluşturulan kübik yüzey grafiği ve nem dağılım haritası.

4.2 Doğrusal Enterpolasyon

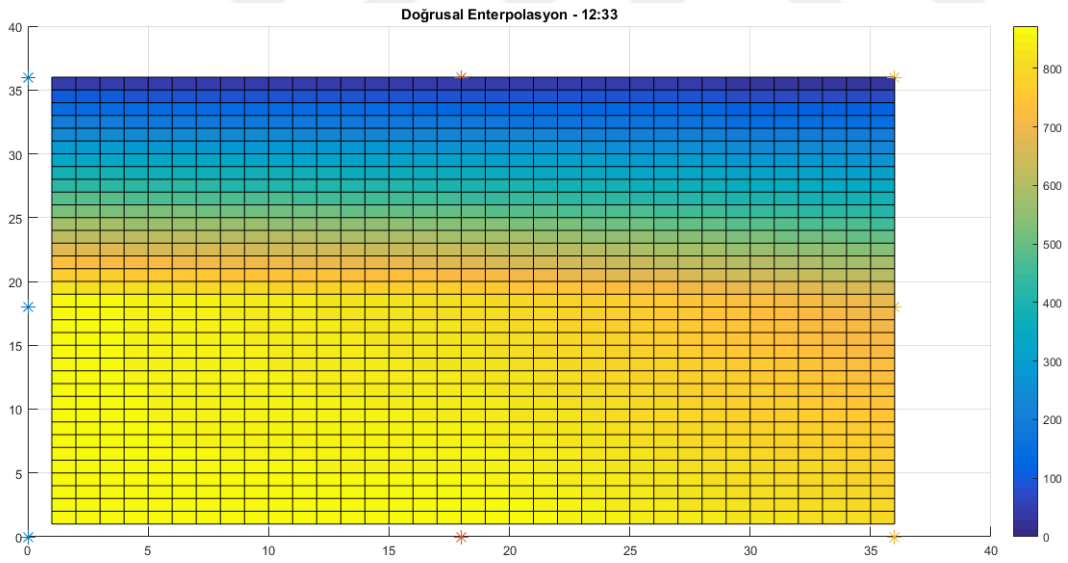
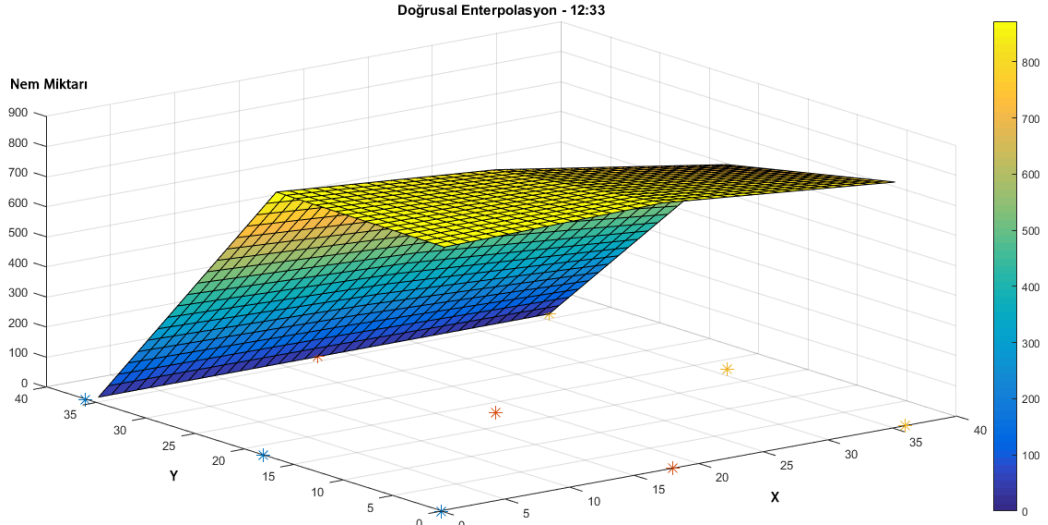
Doğrusal Enterpolasyon yönteminde, iki farklı değişkene karşılık gelen fonsiyon değerleri olan (x_1, y_1) , (x_2, y_2) , bir doğru ile birleştirilir. Bilinen iki nokta arasındaki mesafe az ise, bilinmeyen nokta için bulunacak enterpolasyon fonsiyonunun değeri de az olur. Bu yöntemin avantajı, hesaplamaların basitliği ve hızıdır. Değerleri bilinen noktalar arasında, birinci dereceden doğru denklemi elde edilerek ara değerleri bulmaktadır [29].

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1} \quad (4.3)$$

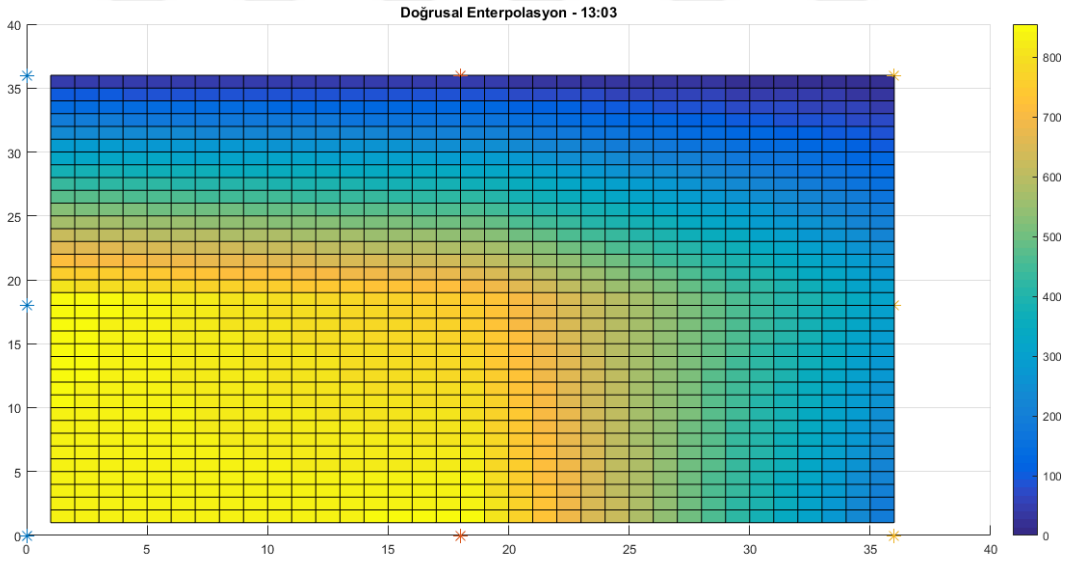
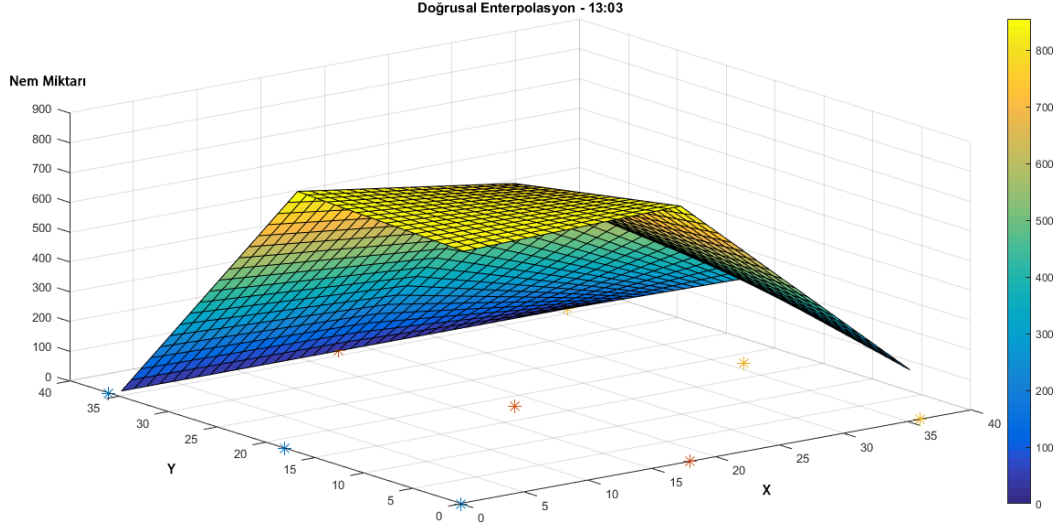
Verilere, doğrusal enterpolasyon uygulandığında sonuçların dağılımı Şekil 4.6, Şekil 4.7, Şekil 4.8, Şekil 4.9 ve Şekil 4.10'da görülmektedir. Özellikle Doğrusal Enterpolasyon - 12:33 nem haritasında ve Doğrusal Enterpolasyon - 13:03 haritası karşılaştırıldığında, deney amaçlı sulama yapılan 2 numaralı ve 5 numaralı sensörlerdeki değerlerin düşüşü rahatlıkla gözlemlenmektedir.



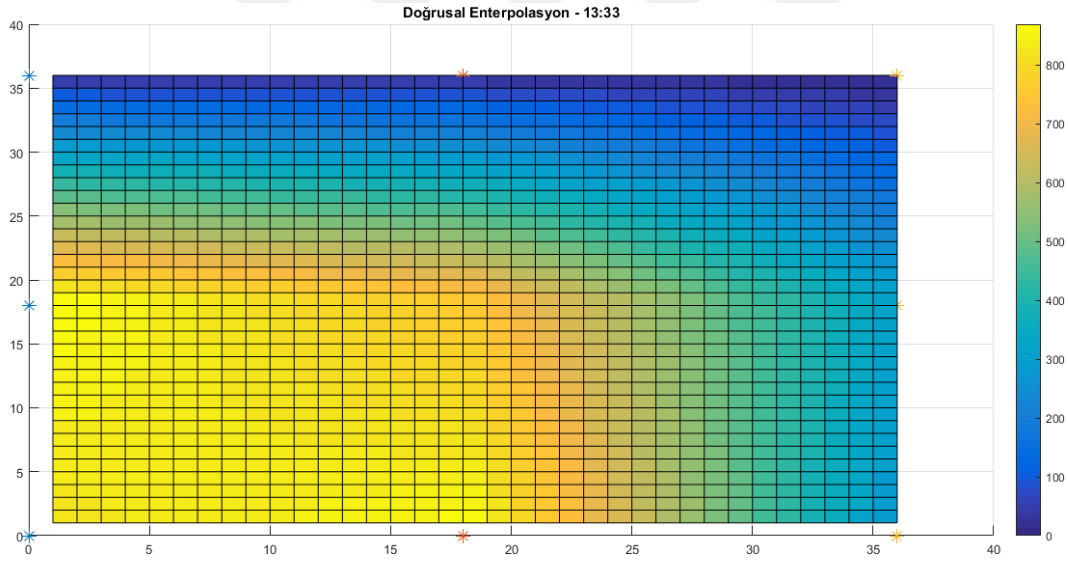
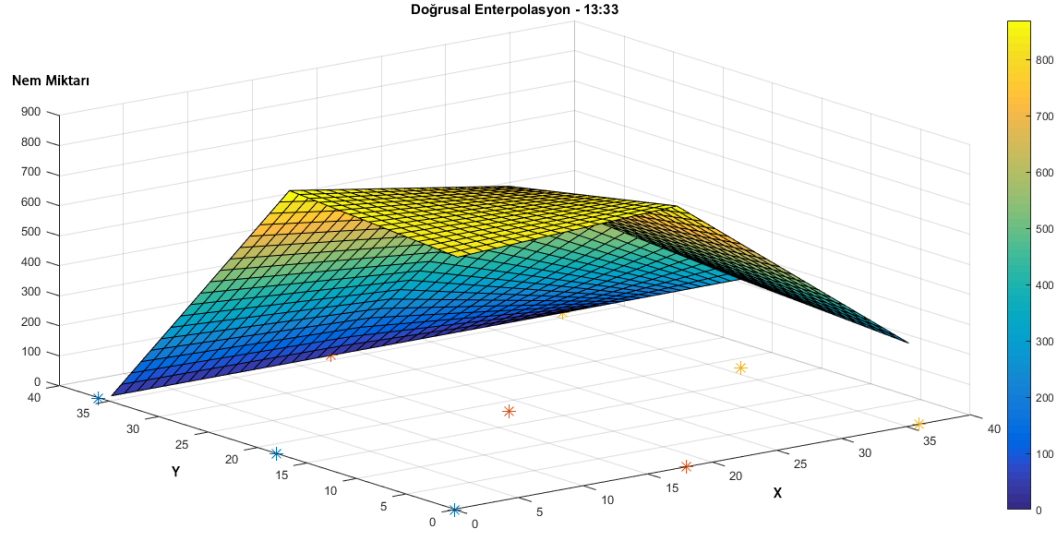
Şekil 4.6 : Doğrusal Enterpolasyonun verilere uygulanması ile 12:03 anında oluşturulan doğrusal yüzey grafiği ve nem dağılım haritası.



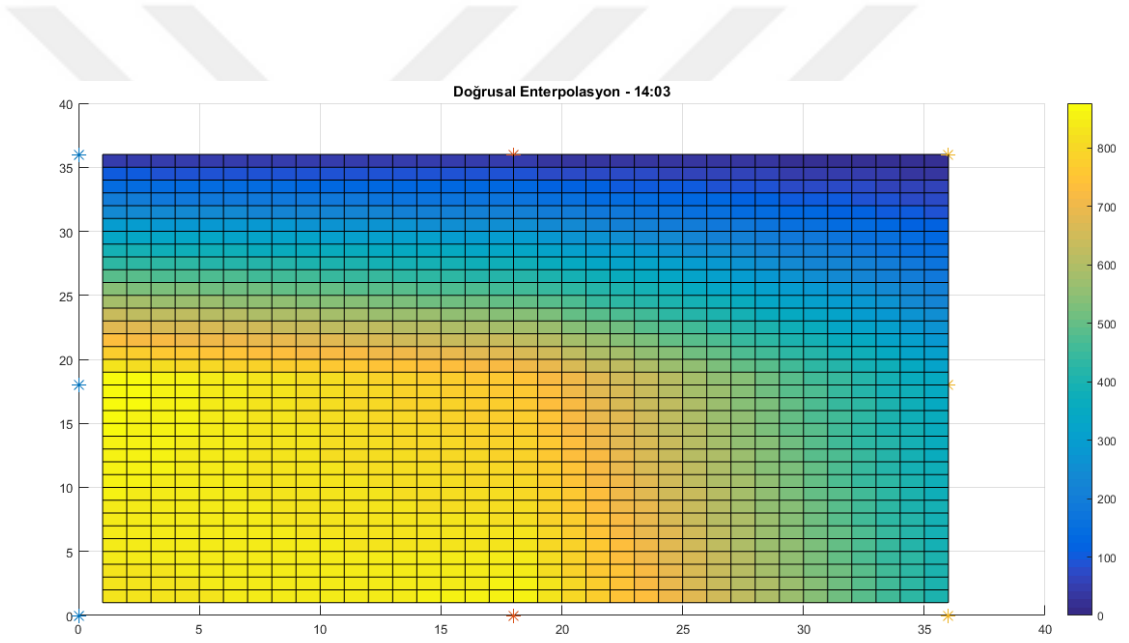
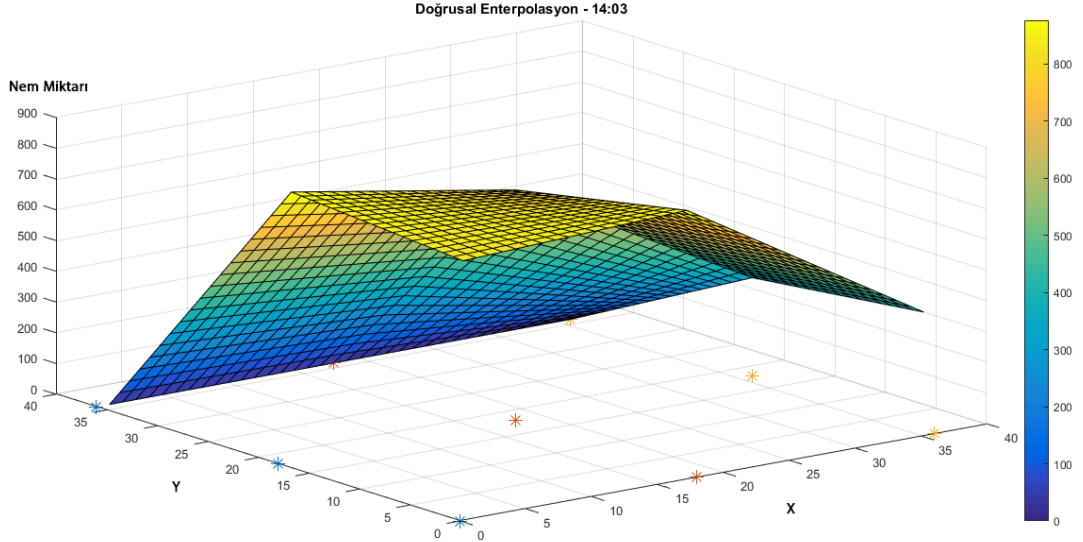
Şekil 4.7 : Doğrusal Enterpolasyonun verilere uygulanması ile 12:33 anında oluşturulan doğrusal yüzey grafiği ve nem dağılım haritası.



Şekil 4.8 : Doğrusal Enterpolasyonun verilere uygulanması ile 13:03 anında oluşturulan doğrusal yüzey grafiği ve nem dağılım haritası. 12:33-13:03 arasında yapılan sulamanın grafikte ve nem haritasında gösterilmesi.



Şekil 4.9 : Doğrusal Enterpolasyonun verilere uygulanması ile 13:03 anında oluşturulan doğrusal yüzey grafiği ve nem dağılım haritası.



Şekil 4.10 : Doğrusal Enterpolasyonun verilere uygulanması ile 14:03 anında oluşturulan doğrusal yüzey grafiği ve nem dağılım haritası.

4.3 Spline Enterpolasyon

Görüntü işlemede oldukça sık kullanılan bir enterpolasyon yöntemidir. Spline enterpolasyon sayesinde, ara değer hesaplanırken düşük dereceli polinomlar kullanılabilir. Ayrıca oluşturulan fonksiyon veri setindeki tüm noktalardan geçmektedir [30].

Fazla verinin bulunduğu, küçük dereceli polinomlarda işlem yaparken kullanılan enterpolasyon yöntemidir. Veriler çok küçük aralıklara ayrılır, küçük dereceli polinomlar uydurulur [29].

Spline Enterpolasyon, yatay olarak ölçüm yapılan ve kısa mesafelerde büyük değişimlerin olduğu alanlar için uygun bir yöntem değildir. Daha çok derinlik, yüksklik, hava kirliliği gibi ölçümlerin hesaplanmasında bu yöntem kullanılmaktadır [31].

4.3.1 Doğrusal Spline Enterpolasyon

$$Z^*(x_0) = x_i + \frac{x_{i+1} - x_i}{d_{i+1} - d_i} (d - d_i) \quad (4.4)$$

Elde bulunan doğru ikilileri kullanılarak, her bir aralık için her bir aralık için ayrı bir doğru denklemi belirlenir.

4.3.2 Kuadratik (İkinci Dereceden) Spline Enterpolasyon

$$Z^*(x_0) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 \quad (4.5)$$

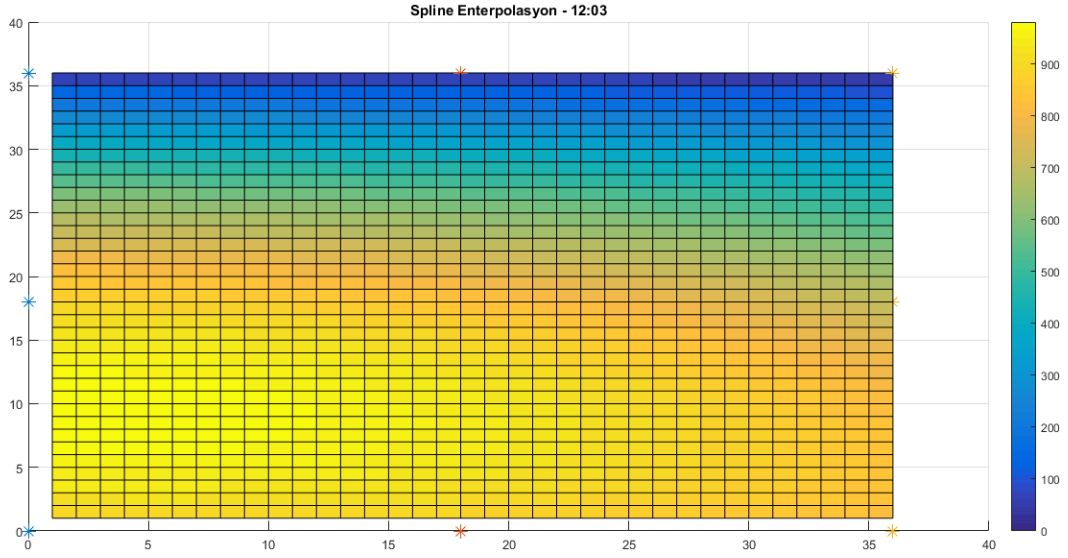
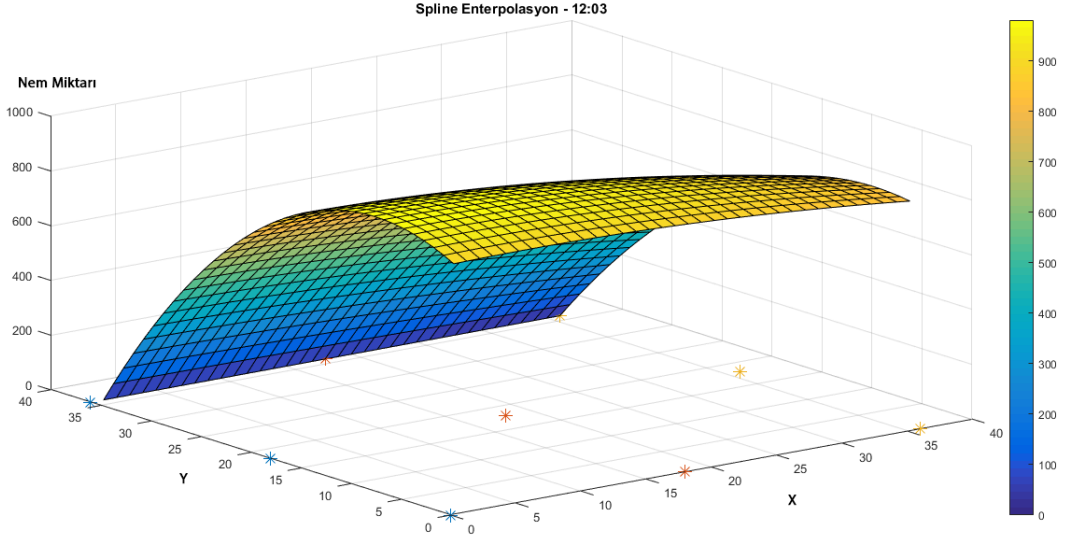
Doğrusal spline enterpolasyondan farklı olarak, değeri bilinen noktalar için her aralığa ikinci dereceden bir polinom uyumlanmaya çalışılır. Herhangi iki noktadan yalnız bir doğru geçmesine karşın, sonsuz sayıda ikinci dereceden polinom geçmektedir. Bu nedenle polinomlar tekil hale getirilmelidir.

4.3.3 Kübik (Üçüncü Dereceden) Spline Enterpolasyon

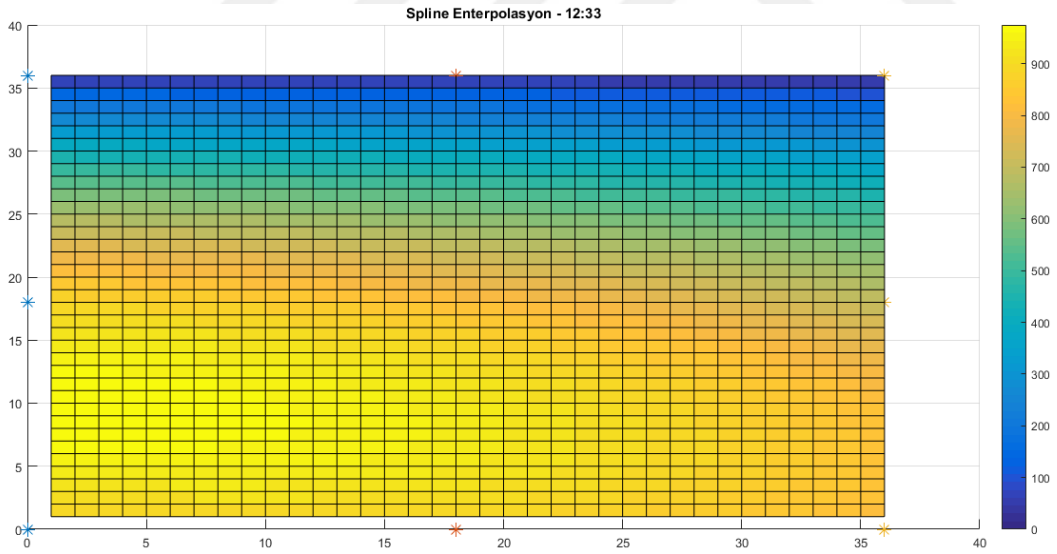
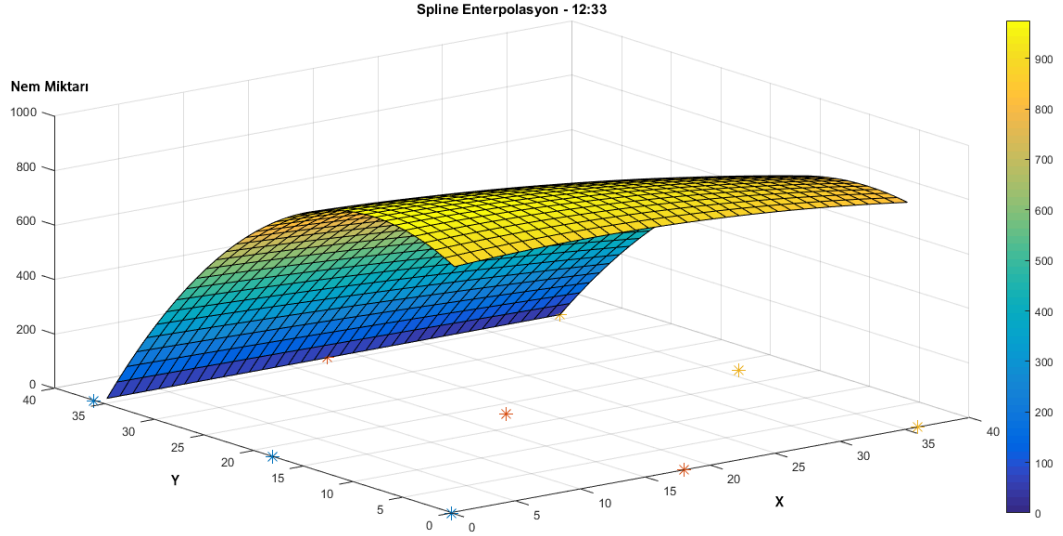
$$Z^*(x_0) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \quad (4.6)$$

Ölçüm noktası çoğaldıkça, her bir aralığın uzunluğu da azalmaktadır. Üçüncü dereceden polinomlar, bu aralıklardaki değişimleri başarılı bir şekilde temsil etmektedir. Kübik spline enterpolasyon, noktalar için her bir aralığa üçüncü dereceden bir polinom uyumlama esasına dayanmaktadır.

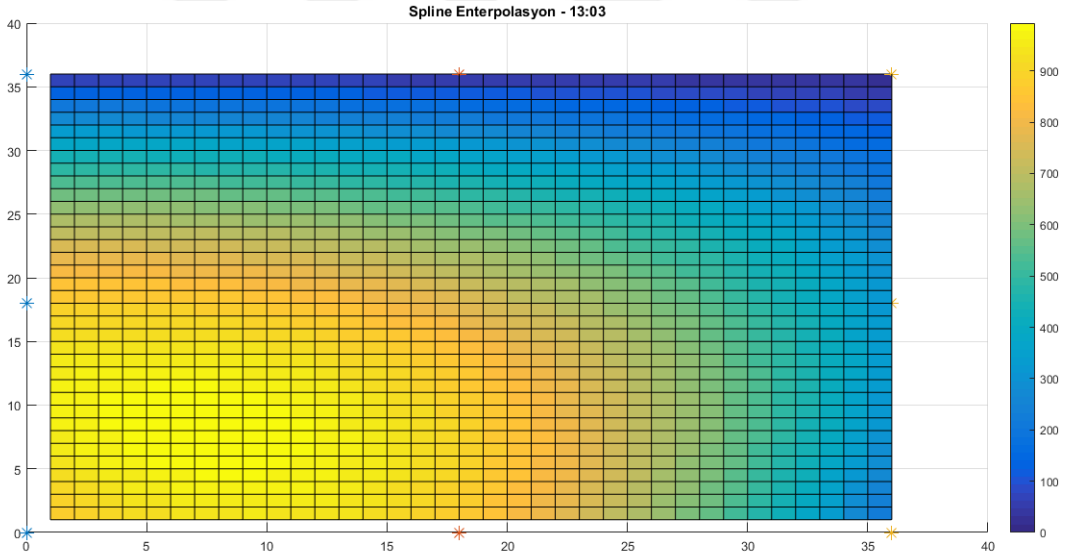
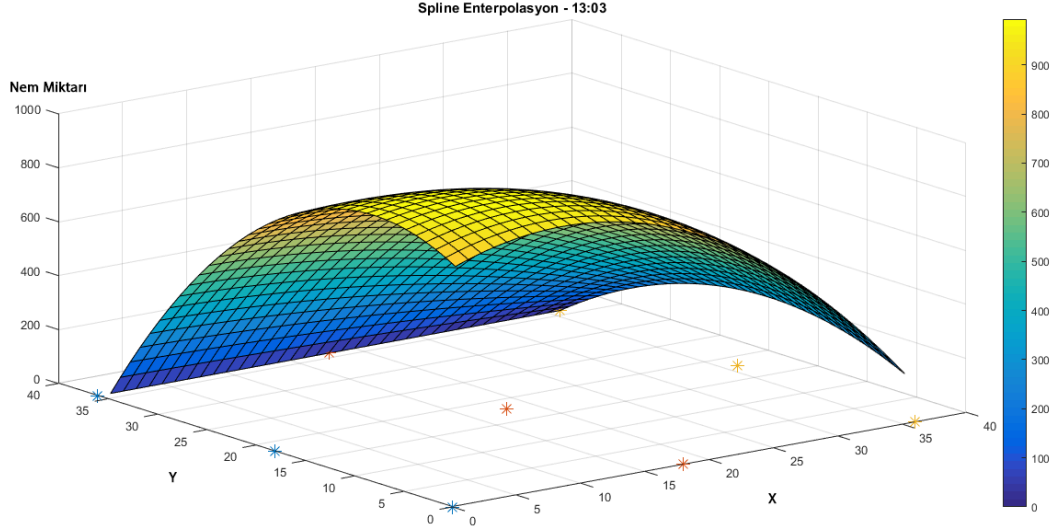
Verilere, spline enterpolasyon uygulandığında sonuçların dağılımı Şekil 4.11, Şekil 4.12, Şekil 4.13, Şekil 4.14 ve Şekil 4.15'te görülmektedir. Özellikle Spline Enterpolasyon - 12:33 ve Spline Enterpolasyon - 13:03 grafikleri karşılaştırıldığında, deney amaçlı sulama yapılan 2 numaralı ve 5 numaralı sensörlerdeki değerlerin düşüşü rahatlıkla gözlemlenmektedir.



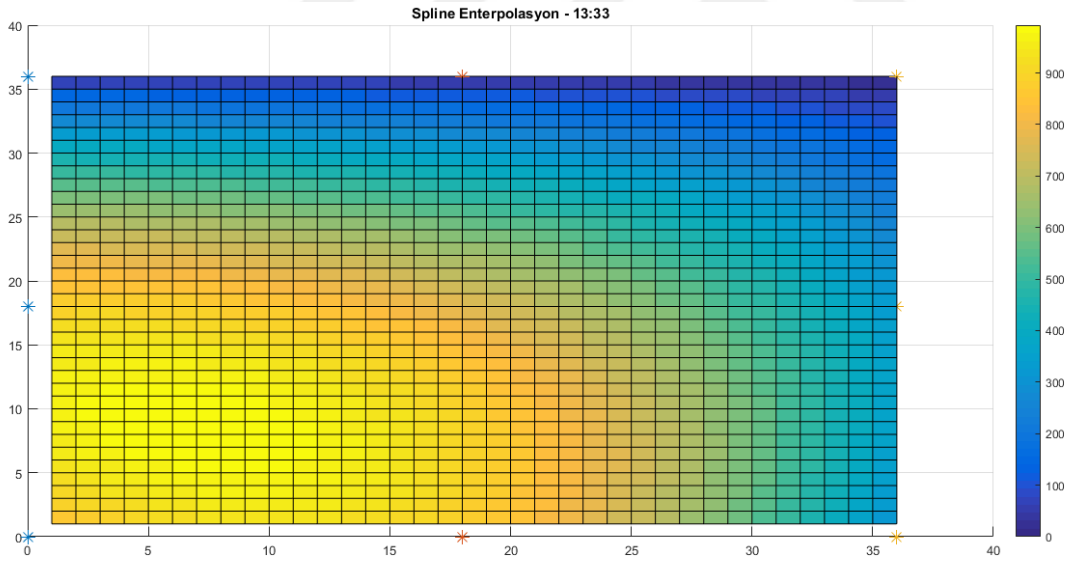
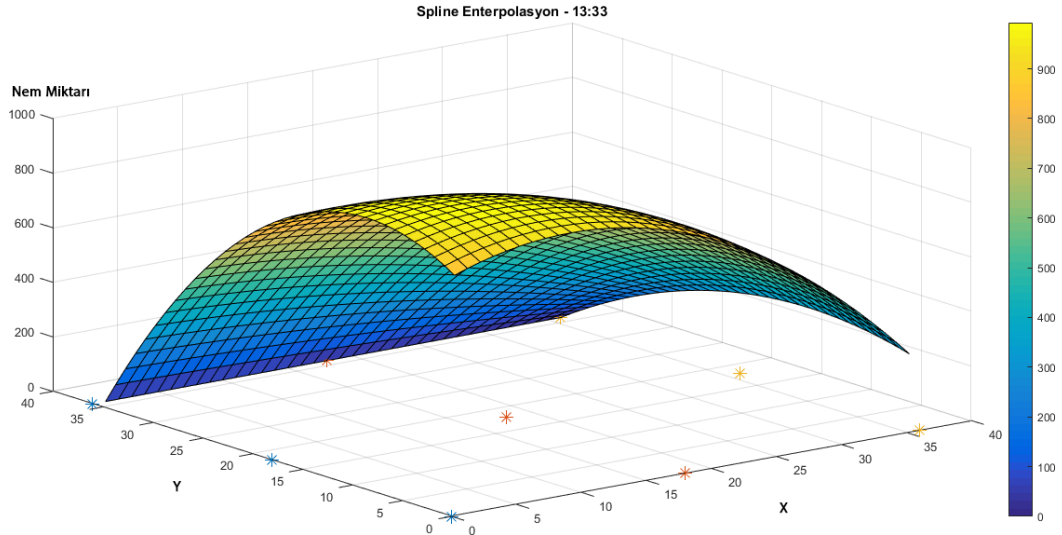
Şekil 4.11 : Spline Enterpolasyonun verilere uygulanması ile 12:03 anında oluşturulan yüzey grafiği ve nem dağılım haritası.



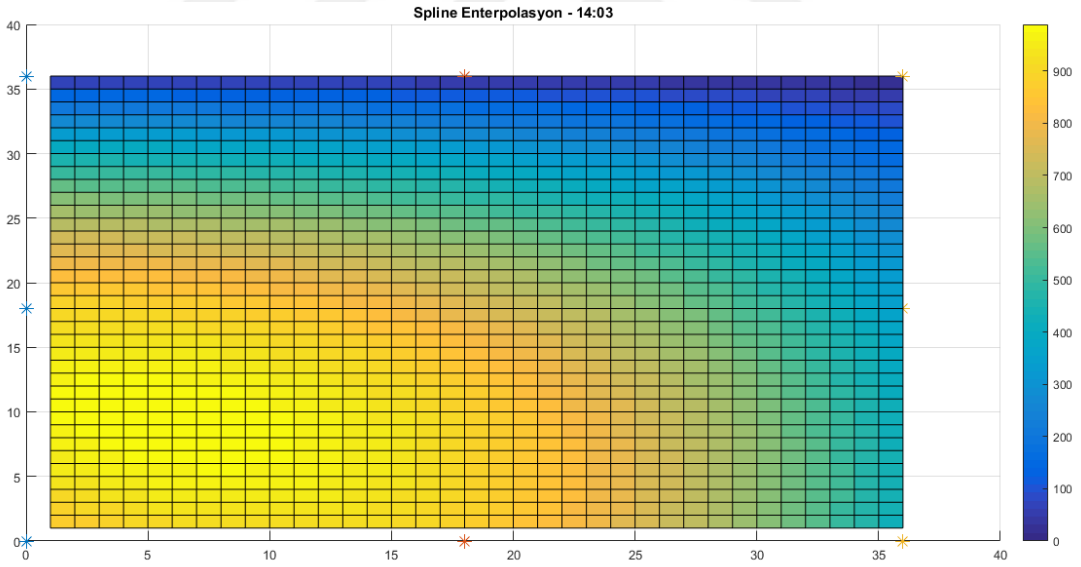
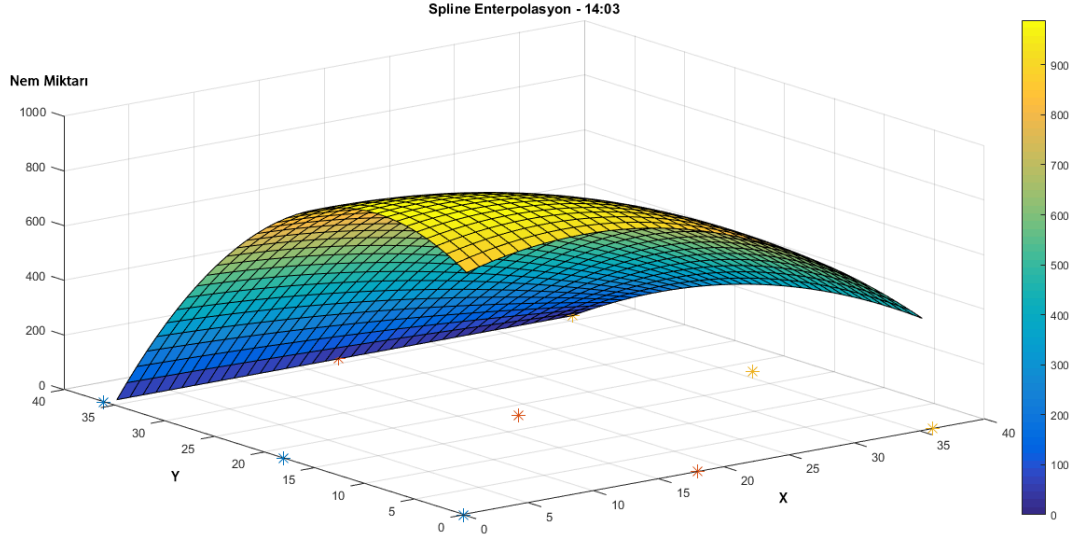
Şekil 4.12 : Spline Enterpolasyonun verilere uygulanması ile 12:33 anında oluşturulan yüzey grafiği ve nem dağılım haritası.



Şekil 4.13 : Spline Enterpolasyonun verilere uygulanması ile 13:03 anında oluşturulan yüzey grafiği ve nem dağılım haritası. 12:33-13:03 arasında yapılan sulamanın grafikte ve nem dağılımında görülmesi.



Şekil 4.14 : Spline Enterpolasyonun verilere uygulanması ile 13:33 anında oluşturulan yüzey grafiği ve nem dağılım haritası.



Şekil 4.15 : Spline Enterpolasyonun verilere uygulanması ile 14:03 anında oluşturulan yüzey grafiği ve nem dağılım haritası.



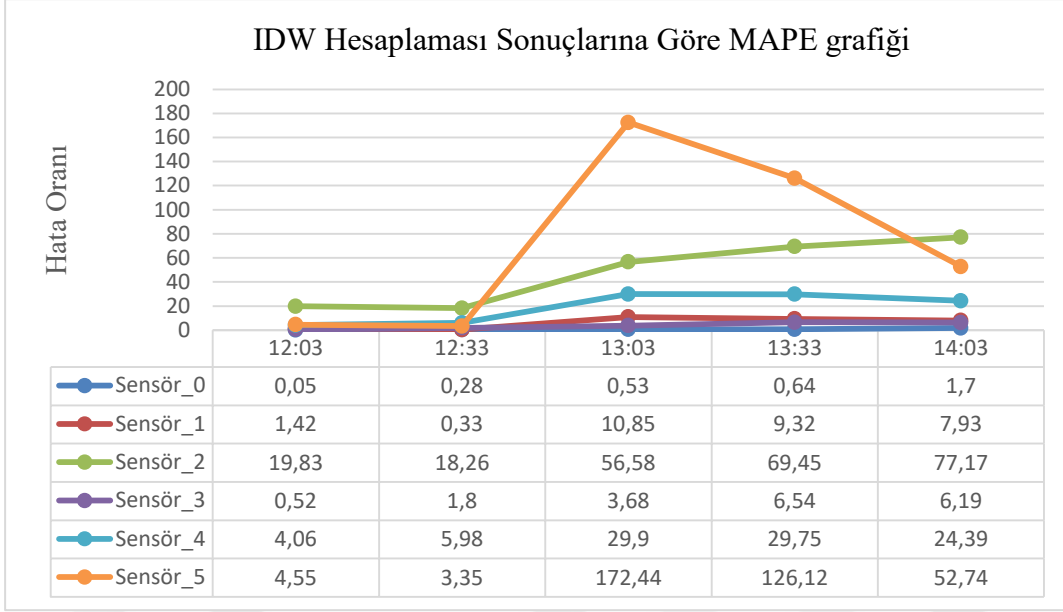
5. SONUÇ VE ÖNERİLER

Yaygın kullanılan IDW yöntemi hesaplaması yapıldığında, bazı sensör bölgelerinde tahmin başarılı olurken, bazı noktalarda sonuçlar gerçek değerlerden oldukça uzaktır. Çizelge 5.1’te özellikle 0 numaralı sensör ve 2 numaralı sensör seçilmiştir. 2 numaralı sensörün değerinde, 13:33’te büyük bir değişim görülmektedir. Şekil 5.1’de görülmekte olan grafik, tüm sensörler üzerinde yapılan hesaplamaları ve hataları göstermektedir. Alınan sonuçlar üzerinde MAPE hesaplaması yapılmıştır ve hata oranları bulunmuştur.

Çizelge 5.1 : Sensör_0 ve Sensör_2’nin gerçek değerleri ile diğer sensörlerden IDW hesaplamasıyla bulunan değerleri

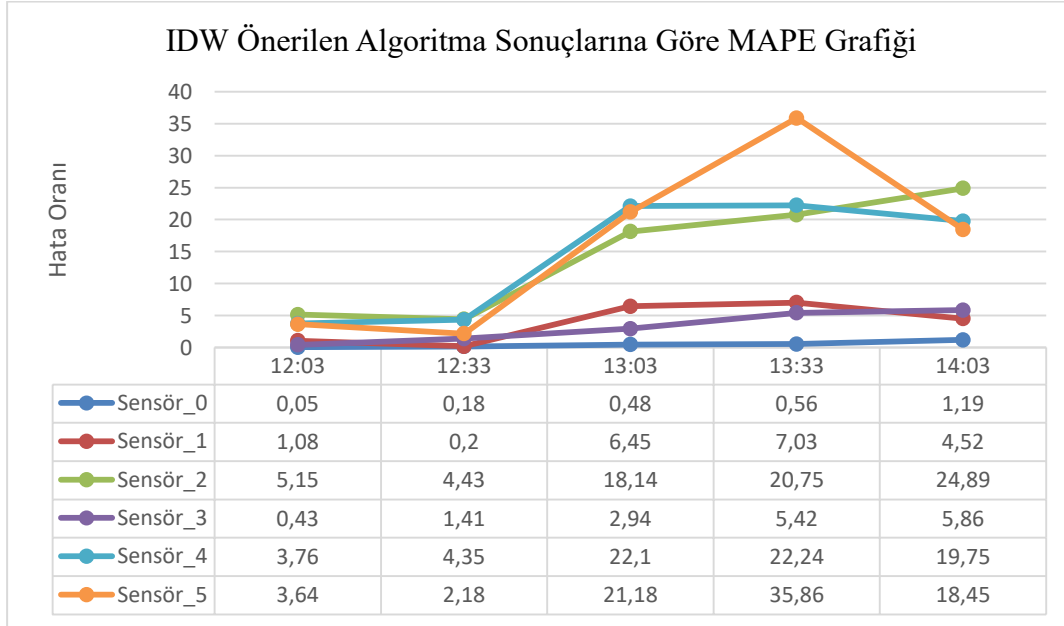
Sensör_0	Zaman	IDW ile bulunan Sensör_0 değeri
870,00	12:03:32	873,4487
871,00	12:33:36	857,4153
862,00	13:03:39	870,4061
876,00	13:33:42	873,9034
884,00	13:58:44	880,8925

Sensör_2	Zaman	IDW ile bulunan Sensör_2 değeri
681,00	12:03:33	816,0424
682,00	12:33:36	806,5536
286,00	13:03:39	447,8129
297,00	13:33:42	503,2793
321,00	14:03:45	568,728



Şekil 5.1 : IDW yönteminin her bir sensör için hesaplaması ve sonuçlara göre gerçek değerler ile MAPE’de hata oranlarının bulunması.

Değeri aniden değişen sensörlere ait verilerin, IDW hesaplamasında sonuçların doğruluğunu etkilediği ve tahmin sonucunun doğruluğunu düşürdüğü gözlemlenmiştir. Şekil 5.2’de önerilen algoritma ile IDW hesaplamalarının yapıldığı ve hata oranlarının MAPE uygulanarak bulunduğu görülmektedir. İki grafik karşılaştırıldığında hata oranlarında azalma olduğu görülmektedir.



Şekil 5.2 : Önerilen algoritmaya göre IDW yönteminin hesaplaması ve sonuçlara göre gerçek değerler ile MAPE’de hata oranlarının bulunması.

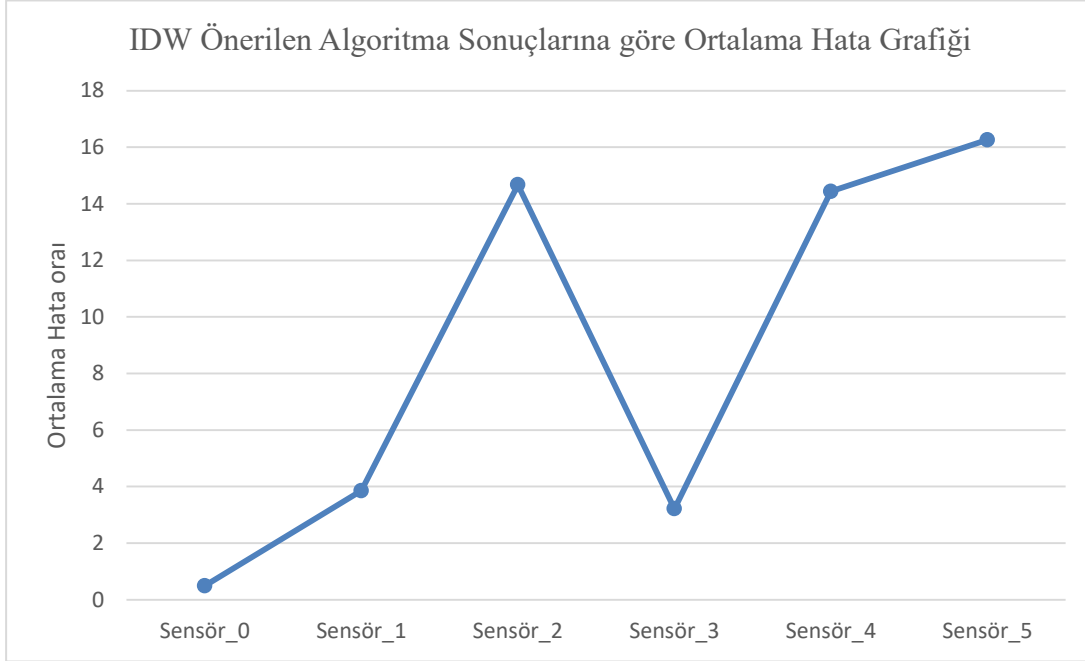
Bu çalışmada önerilen algoritmaya göre program çalıştırıldığında, IDW hesapma sonuçlarının gerçek verilere yakınlığının arttığı gözlemlenmiştir. Şekil 5.3’de aynı veriler üzerinde 2 farklı IDW hesaplaması yapıldığı görülmektedir. İlk yapılan işlemde değeri ani değişen sensörler hesaplamaya dahil edilmiştir ve tahmin noktasının değeri 658,22 olarak bulunmuştur. İkinci olarak Şekil 5.3’de de görüldüğü gibi, “Değeri Ani Değişenleri Çıkar” kutusu işaretlenmiş ve program önerilen yeni algoritmaya göre, aynı veriler üzerinde çalıştırılmıştır. Programın çalıştırılması sonucunda tahmin noktasının değeri, 668,39 olarak görülmüştür.

The figure consists of two screenshots of a software interface, labeled 'Form1'. Both screenshots show a map on the left with a red arrow pointing to a specific location. The right side of the interface is divided into several sections:

- Dayanak ve Referans Gözlem Verileri Tahmin Değerleri:** This section displays the calculated value. In the top screenshot, the value is 791. In the bottom screenshot, the value is 820,7. Both values are circled in red.
- Değeri Ani Değişenleri Çıkar:** A checkbox that is unchecked in the top screenshot and checked in the bottom screenshot.
- Tüm Değerler Çap Aralığı:** A section with input fields for 'Aralık' (13:00:00 to 13:30:00), 'p değeri' (2), and 'n değeri' (0). Buttons for 'Hesapla' and 'Değeri Ani Değişenleri Çıkararak Hesapla' are present.
- Hesaplama Değerleri Mesafeler:** A section with a list of distances and weights. In the top screenshot, the list is empty. In the bottom screenshot, it contains five entries: 0 - 820,531204900953; 1 - 820,167463760519; 2 - 820,176048052875; 3 - 821,199401347382; 4 - 820,80380871023; 5 - 820,434719068394.
- Seçilen Gözlemler Ağrılıklar:** A section with a list of selected observations and their weights. In the top screenshot, the list is empty. In the bottom screenshot, it contains five entries: 0 - 862; 1 - 748; 2 - 286; 3 - 827; 4 - 853; 5 - 219.
- Diğer:** A section with a dropdown menu showing 'Diğer' and a table with columns 'kullanilaci numara', 'omakleri', 'X', and 'Y'. The values are: kullanilaci numara: 5, omakleri: 36, X: 0, Y: 0.
- Kullanılacak Ölçüm:** A section with a button labeled 'Bu noktayı da referans olarak ekle'.

Şekil 5.3 : Değeri ani değişen sensörlerin verilerini dahil etmeden IDW yönteminin hesaplanması. “Değeri ani değişenleri çıkar” onay kutusu işaretlendiğinde yapılan hesaplamaların sonuçlarının gösterimi.

Çalışmaların sonuçlarına bakıldığında IDW yönteminin ara değer kesitiriminde başarılı bir yöntem olduğu görülmüştür. Fakat ani değişikliklerin olduğu durumlarda hata oranlarında artış görülmektedir. Bu sebeple önerilen algoritmanın, sonuçları incelendiğinde ise tahminlerdeki başarı oranını arttırdığı görülmüştür. Şekil 5.4'te algoritmadan alınan sonuçlara göre ortalama hatayı göstermektedir. Hata oranlarına bakıldığında en fazla % 18 oranında hata olduğu görülmektedir.



Şekil 5.4 : Önerilen IDW yönteminin sonuçlarına göre ortalama hatanın bulunması.

Sonuç olarak, toprak nemini etkileyen birden fazla faktör vardır. Yapılan ölçümlerde, alınan verilerin toprak türüne, toprağın içindeki maddelere, nem ölçümü yapılan ortamın nemine, ölçümün yapıldığı mevsime göre değişiklik gösterdiği gözlemlenmiştir. Ölçüm yapılırken verileri alan sensörlerin toprak altındaki derinliği de yine sonuçlara etki etmektedir. Yapılan araştırmalar, ölçüm yapılacak ortama göre uygun sensör seçildiğini göstermiştir.

Akıllı ev sistemleri ve akıllı nesnelerin yaygınlaşması, organik tarıma duyulan ihtiyaç, yeşil alanların şehirlerde azalması gibi faktörler ev tarımı uygulamalarını yaygınlaştırmaktadır. Yapılan bu çalışma akıllı ev sistemlerine dahil edilerek, ev tarımı faaliyetlerinin verimini arttırabilir ve insan kontrolü gerekmeden, zaman ve tarım faaliyetlerinde tasarruf sağlayabilir.

İlerleyen çalışmalarda, önerilen algoritmada deęişiklikler yapılarak, yeni yöntemler denenerek tahminlerin başarı oranı daha fazla arttırmak mümkün gözükmektedir. Bu çalışmada yer alan algoritmalara, toprak nemini etkileyen toprak türü, toprak içindeki maddeler, ortam nemi, ortam sıcaklığı gibi dięer faktörler de kontrol edilerek katkılar sağlanabilir. Bezier eğrileri ile bu çalışmaya katkılar sağlanabilir.

Tasarlanan devre ve oluşturulan kodlar akıllı cihazlarda çalışabilecek şekilde düzenlenebilir. Bu çalışma akıllı ev sistemlerine dahil edilerek tüm sistemin tek elden yönetilmesi, takibi sağlanabilir.





KAYNAKLAR

- [1] **Çiftçi, H.** (2004). Türkiye'nin Bilim Ve Teknoloji Stratejisi. *Çukurova Üniversitesi, Türkiye* : Adana
- [2] **Soydan, B.** (2015). Akıllı Sulama Sistemleri. <http://www.turkishtimedergi.com/tarim/tarima-yon-veren-10-yeni-teknoloji/>. Erişim tarihi: 26.08.2017.
- [3] **Galhena, D. H. & Freed, R. & Maredia M. K.** (2013). Home gardens: a promising approach to enhance household food security and wellbeing. *Galhena et al. Agriculture & Food Security* (2:8)
- [4] **Brewer, C.** (2016). The Benefits Of Growing Your Own Home Garden: Body And Mind. <http://www.finerminds.com/health-fitness/home-garden/>, Erişim Tarihi: 02.10.2017
- [5] **Syme, G. J. & Shao, Q. & Po, M. & Campbell, E.** (2004). Predicting and understanding home garden water use. *Landscape and Urban Planning* (121-128). Australia : WA
- [6] **Tarhan, S.** (2015). Bilgi Teknolojilerinin Tarımda Kullanımı. *Translation Preface. CIGR Handbook of Agricultural Engineering Volume VI Information Technology. Edited by CIGR-The International Commission of Agricultural Engineering; Volume Editor, Axel Munack. St. Joseph, Michigan. Copyright American Society of Agricultural Engineers, USA: ASABE*
- [7] **Altın, S.** (2006). Mikrodenetleyici Kullanarak Zirai Sulama Sistemlerinin Otomatik Hale Getirilmesi. *Gazi Üniversitesi Fen Bilimleri Enstitüsü, Türkiye* : Ankara
- [8] **Shin, C. & Lee, Y. & Lee, M. & Park, J. & Yoe, H.** (2009). Design of Ubiquitous Glass Green Houses. *Software Technologies for Future Dependable Distributed Systems, Korea*
- [9] **Bayar, E. & Genç, M.** (2013). Isparta Kent Merkezindeki Tarihi Mistik Mekânlardaki Ağaç Türleri. Mevcut Durumları ve Olası Çoğaltma Yöntemleri, *V. Süs Bitkileri Kongresi, Türkiye* : Yalova
- [10] **Demir, B. & Öztürk, İ.** (2007). Sera Otomasyon Sistemlerine İlişkin Bilgisayar Uygulamaları. *Tarımsal Mekanizasyon 24. Ulusal Kongresi, Türkiye* : Kahramanmaraş
- [11] **Thornton, P.E. & Hasenauer, H. & White, M.A.** (2000). Simultaneous estimation of daily solar radiation and humidity from observed temperature and precipitation: an application over complex terrain in Austria. *Agricultural and Forest Meteorology* 104 255–271 , Austria : Wien

- [12] **Hummel, J.W. & Sudduth, K.A. & Hollinger, S.E.** (2001). Soil moisture and organic matter prediction of surface and subsurface soils using an NIR soil sensor. *Computers and Electronics in Agriculture* 32 149–165, USA : Columbia
- [13] **Njoku E.G. & Jackson, T.J. & Lakshmi, V. & Chan, A.K. & Nghiem, S.V.** (2003). Soil Moisture Retrieval From AMSR-E. *IEEE Transactions On Geoscience And Remote Sensing, Vol. 41, No. 2*, USA : Columbia
- [14] **Ünüvar, A.K.** (2009). Mikrodenetleyici Kontrollü Damla Sulama Sistem Tasarımı. *Mustafa Kemal Üniversitesi Fen Bilimleri Enstitüsü, Türkiye : Antakya*
- [15] **Çakır, A. & Çalıř, H.** (2008). Uzaktan Kontrollü Otomatik Sulama Sistemi Tasarımı ve Uygulaması. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi 11-3 258-261*, Türkiye : Isparta
- [16] **Gutiérrez, J. & Villa-Medina, J.F. & Nieto-Garibay, A. & Porta-Gándara, M.A.** (2014). Automated Irrigation System Using a Wireless Sensor Network and GPRS Module. *IEEE Transactions On Instrumentation And Measurement, Vol. 63, No. 1*, Mexico
- [17] **Ding, Y. & Wang, Y. & Miao, Q.** (2011). Research on the Spatial Interpolation Methods of Soil Moisture Based on GIS. *International Conference on Information Science and Technology*, China : Jangsu
- [18] **Yao, X. & Fu, B. & Lü, Y. & Sun, F. & Wang, S. & Liu, M.** (2013). Comparison of Four Spatial Interpolation Methods for Estimating Soil Moisture in a Complex Terrain Catchment. *State Key Laboratory of Urban and Regional Ecology, Research Center for Eco-Environmental Sciences, USA*
- [19] **Kazancı, S.Z. & Kayıkçı, E.T.** (2012). Konumsal Enterpolasyon Yöntemleri Uygulamalarında Optimum Parametre Seçimi: Doğu Karadeniz Bölgesi Günlük Ortalama Sıcaklık Verileri Örneđi. *Karadeniz Teknik Üniversitesi Üniversitesi, Mühendislik Fakültesi, Harita Mühendisliđi Bölümü, Türkiye : Trabzon*
- [20] **Yiđit, C.Ö.** (2003). Elipsoidal Yüksekliklerin Ortometrik Yüksekliđe Dönüşümünde Kullanılan Enterpolasyon Yöntemlerinin Karşılaştırılması. *Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Türkiye : Konya*
- [21] **Şentürk, E. & İnce C.D.** (2015). Kocaeli İlinde Yerel Jeoidin Enterpolasyon Yaklaşımıyla Belirlenmesi. *TMMOB Harita ve Kadastro Mühendisleri Odası, 15. Türkiye Harita Bilimsel ve Teknik Kurultayı, Türkiye : Ankara*
- [22] **Aydın, O. & Çiçek, İ.** (2013). Ege Bölgesi'nde Yađışın Mekânsal Dađılımı. *COĞRAFİ BİLİMLER DERGİSİ CBD 11 (2), 101-120*, Türkiye : Ankara
- [23] **Arduino.** (t.y.), <https://store.arduino.cc>, Eriřim Tarihi: 18.07.2017
- [24] **Şakar, G.** (2013). Bildiriler Kitabı. 16. *Ulusal Makina Teorisi Sempozyumu*, Türkiye : Erzurum
- [25] **Mathworks.** (t.y.), <https://ch.mathworks.com/>, Eriřim Tarihi: 20.07.2017

- [26] **VisualStudio.** (t.y.), <https://www.visualstudio.com/tr>, Eriřim Tarihi: 25.07.2017
- [27] **Demircan, M. & Arabacı, H. & Bölük, E. & Akçakaya, A. & Şensoy, S. & Ekici, M.** (2013). İklim Normalleri Ve 1981-2010 Sıcaklık Normallerinin Coğrafi Bilgi Sistemleri İle Topografya Kullanarak Yüksek Çözünürlüklü Grid Veri Setinin Üretilmesi. *6th Atmospheric Science Symposium – ATMOS*, Türkiye : İstanbul
- [28] **McKinley, S. & Levine, M.** (t.y.). Cubic Spline Interpolation
- [29] **Vatansever, F. & Doğalı, G.** (2011). Klasik Enterpolasyon Yöntemleri ve Yapay Sinir Ağı Yaklaşımlarının Karşılaştırılması. *6th International Advanced Technologies Symposium (IATS'11)*, Türkiye : Elazığ
- [30] **Spline İnterpolasyonu** (t.y.),
https://acikders.ankara.edu.tr/pluginfile.php/3712/mod_resource/content/1/Ders02_Spline_Interpolasyonu.pdf, Eriřim Tarihi: 07.09.2017
- [31] **Tural, S.** (2011). Gerçek Zamanlı Meteoroloji Verilerinin Toplanması, Analizi ve Haritalanması. *Ankara Üniversitesi Fen Bilimleri Enstitüsü*, Türkiye : Ankara



EKLER

EK A : YAZILIM KODLARI

EK A.1 : Matlab Kodu

```
clear all
clc
close all
veri1 = importdata('sonuc.txt');
veri2 = importdata('veri.txt');
data = [veri1 ; veri2]

x = data(:,1) ; y = data(:,2) ; z = data(:,3) ;
dt = delaunayTriangulation(x,y) ;
tri = dt.ConnectivityList ;
xi = dt.Points(:,1) ;
yi = dt.Points(:,2) ;
%benim = veri1(3,:);

F = scatteredInterpolant(x,y,z);
zi = F(xi,yi) ;
% trisurf(tri,xi,yi,zi) % dikkat
xlabel('X ')
ylabel('Y');
zlabel('Tahmin')
hold on ;
numel(veri1(:,3))
tamam = [inf(size(veri1(:,3))) ; veri1(:,3)]
plot3(x,y,tamam,'.', 'MarkerSize',30, 'color', 'red')

hold on ;
numel(veri1(:,3))
tamam = [inf(size(veri2(:,3))) ; veri1(:,3)]
plot3(x,y,tamam,'.', 'MarkerSize',30, 'color', 'blue') ; legend('Dayanak Noktaları',
'Tahmin Noktaları')
```

EK A.2 : Arduino Kodu

```
int potpin,potpin1,potpin2,potpin3,potpin4,potpin5;
void setup()
{
  Serial.begin(9600);
}
```

```

void loop()
{
    potpin=analogRead(A0);
    potpin1=analogRead(A1);
    potpin2=analogRead(A2);
    potpin3=analogRead(A3);
    potpin4=analogRead(A4);
    potpin5=analogRead(A5);
    Serial.print("0#");
    Serial.println(potpin);
    Serial.print("1#");
    Serial.println(potpin1);
    Serial.print("2#");
    Serial.println(potpin2);
    Serial.print("3#");
    Serial.println(potpin3);
    Serial.print("4#");
    Serial.println(potpin4);
    Serial.print("5#");
    Serial.println(potpin5);
    delay(180000);
}

```

EK A.3 : Visual Studio Seri Port Kodu

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO.Ports;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Runtime;
using System.Runtime.InteropServices;
using System.IO;
using System.Text.RegularExpressions;

namespace ReadDataFromSerialPortWindowsApp
{
    public partial class Index : Form
    {
        [DllImport("wininet.dll")]
        private extern static bool InternetGetConnectedState(out int Description, int
ReservedValue);
        public Index()
        {
            InitializeComponent();

```

```

}
SerialPort mySerialPort;
List<WheatherForceData> arduinoDatas = new List<WheatherForceData>();
int i;
private void btnOpenSerialPort_Click(object sender, EventArgs e)
{
    try
    {
        mySerialPort = new SerialPort(cmbAvailablePorts.Text);
        mySerialPort.BaudRate = 9600;
        mySerialPort.Parity = Parity.None;
        mySerialPort.DataBits = 8;
        mySerialPort.StopBits = StopBits.One;
        mySerialPort.Handshake = Handshake.None;
        mySerialPort.DataReceived += new
SerialDataReceivedEventHandler(DataReceivedHandler);
        mySerialPort.Open();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Bir hata oluştu! Hata Mesajı:\n" + ex.Message,
"HATA", MessageBoxButtons.OKCancel, MessageBoxIcon.Error,
MessageBoxDefaultButton.Button1);
    }
}
private void DataReceivedHandler(object sender,
SerialDataReceivedEventArgs e)
{
    SerialPort sp = (SerialPort)sender;
    string indata = sp.ReadLine();
    lstData.Invoke((Action)(() => lstData.Items.Add(indata)));
    dataGridView1.Invoke((Action)(() => dataGridView1.Rows.Insert(i++,
AddRowToDataGrid(GetDataFromArduino(indata), dataGridView1)));
}
private void Index_Load(object sender, EventArgs e)
{
    cmbAvailablePorts.Items.AddRange(SerialPort.GetPortNames());
}
private SensorData GetDataFromArduino(string data)
{
    string[] getData = data.Split('#');
    SensorData sensorData = new SensorData();
    sensorData.SensorNo = getData[0];
    sensorData.SensorValue = getData[1];
    sensorData.SentDate = DateTime.Now;
    return sensorData;
}
}

```

```

private DataGridViewRow AddRowToDataGrid(SensorData data,
DataGridView dataGridView)
{
    DataGridViewRow row =
(DataGridViewRow)dataGridView.Rows[0].Clone();
    row.Cells[0].Value = data.SensorNo;
    row.Cells[1].Value = data.SensorValue;
    row.Cells[2].Value = data.SentDate;
    return row;
}
private void btnSendToTextFile_Click(object sender, EventArgs e)
{
    string path = @"C:\Users\Hilal\Desktop\NewData1204.txt";
    StreamWriter sw = new StreamWriter(path);
    foreach (DataGridViewRow row in dataGridView1.Rows)
    {
        sw.WriteLine(string.Format("{0} {1} {2}",
row.Cells["NAME"].Value, row.Cells["TEMPERATURE"].Value,
row.Cells["SENTDATE"].Value));
    }
    MessageBox.Show("Veriler Kaydedildi");
    sw.Close();
}

private void button1_Click(object sender, EventArgs e)
{
    string input = @"C:\Users\Hilal\Desktop\tezkodlar\NewData.txt";
    string pattern = @"^0";
    Regex g = new Regex(pattern);
    using (StreamReader r = new StreamReader(input))
    {
        string line;
        while ((line = r.ReadLine()) != null)
        {
            Match m = g.Match(line);

            if (m.Success)
            {
                string v = m.Groups[1].Value;
                label2.Text += " " + line + "\n";
            }
        }
    }
    string pathFile = @"C:\Users\Hilal\Desktop\tezkodlar\";
    string fileName = "sensor0.txt";
    System.IO.File.WriteAllText(pathFile + fileName, label2.Text);
    MessageBox.Show("Veriler Kaydedildi" + pathFile);
}
private void button2_Click(object sender, EventArgs e)

```

```

{
    string input = @"C:\Users\Hilal\Desktop\tezkodlar\NewData.txt";
    string pattern = @"^1";
    Regex g = new Regex(pattern);
    using (StreamReader r = new StreamReader(input))
    {
        string line;
        while ((line = r.ReadLine()) != null)
        {
            Match m = g.Match(line);

            if (m.Success)
            {
                string v = m.Groups[1].Value;
                label3.Text += " " + line + "\n";
            }
        }
    }
    string pathFile = @"C:\Users\Hilal\Desktop\tezkodlar\";
    string fileName = "sensor1.txt";
    System.IO.File.WriteAllText(pathFile + fileName, label3.Text);
    MessageBox.Show("Veriler Kaydedildi" + pathFile);
}

private void button3_Click(object sender, EventArgs e)
{
    string input = @"C:\Users\Hilal\Desktop\tezkodlar\NewData.txt";
    string pattern = @"^2";
    Regex g = new Regex(pattern);
    using (StreamReader r = new StreamReader(input))
    {
        string line;
        while ((line = r.ReadLine()) != null)
        {
            Match m = g.Match(line);
            if (m.Success)
            {
                string v = m.Groups[1].Value;
                label4.Text += " " + line + "\n";
            }
        }
    }
    string pathFile = @"C:\Users\Hilal\Desktop\tezkodlar\";
    string fileName = "sensor2.txt";
    System.IO.File.WriteAllText(pathFile + fileName, label4.Text);
    MessageBox.Show("Veriler Kaydedildi" + pathFile);
}

private void button4_Click(object sender, EventArgs e)
{

```

```

string input = @"C:\Users\Hilal\Desktop\tezkodlar\NewData.txt";
string pattern = @"^3";
Regex g = new Regex(pattern);
using (StreamReader r = new StreamReader(input))
{
    string line;
    while ((line = r.ReadLine()) != null)
    {
        Match m = g.Match(line);

        if (m.Success)
        {
            string v = m.Groups[1].Value;
            label5.Text += " " + line + "\n";
        }
    }
}
string pathFile = @"C:\Users\Hilal\Desktop\tezkodlar\";
string fileName = "sensor3.txt";
System.IO.File.WriteAllText(pathFile + fileName, label5.Text);
MessageBox.Show("Veriler Kaydedildi" + pathFile);
}
private void button5_Click(object sender, EventArgs e)
{
    string input = @"C:\Users\Hilal\Desktop\tezkodlar\NewData.txt";
    string pattern = @"^4";
    Regex g = new Regex(pattern);
    using (StreamReader r = new StreamReader(input))
    {
        string line;
        while ((line = r.ReadLine()) != null)
        {
            Match m = g.Match(line);
            if (m.Success)
            {
                string v = m.Groups[1].Value;
                label6.Text += " " + line + "\n";
            }
        }
    }
    string pathFile = @"C:\Users\Hilal\Desktop\tezkodlar\";
    string fileName = "sensor4.txt";
    System.IO.File.WriteAllText(pathFile + fileName, label6.Text);
    MessageBox.Show("Veriler Kaydedildi" + pathFile);
}
private void button6_Click(object sender, EventArgs e)
{
    string input = @"C:\Users\Hilal\Desktop\tezkodlar\NewData.txt";
    string pattern = @"^5";

```



```

Regex g = new Regex(pattern);
using (StreamReader r = new StreamReader(input))
{
    string line;
    while ((line = r.ReadLine()) != null)
    {
        Match m = g.Match(line);

        if (m.Success)
        {
            string v = m.Groups[1].Value;
            label7.Text += " " + line + "\n";
        }
    }
}
string pathFile = @"C:\Users\Hilal\Desktop\tezkodlar\";
string fileName = "sensor5.txt";
System.IO.File.WriteAllText(pathFile + fileName, label7.Text);
MessageBox.Show("Veriler Kaydedildi" + pathFile);
}

private void btnCloseSerialPort_Click(object sender, EventArgs e)
{
    mySerialPort.Close();
}
}
}

```

EK A.4 : Visual Studio Proje Kodu

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;

namespace Enterpolasyon
{
    public partial class Form1 : Form
    {
        List<Nokta> referansKonumlari;
        List<Nokta> dayanaKonumlari;
        List<Ornek> olcumDegerleri;
    }
}

```

```

public Form1()
{
    InitializeComponent();
    referansKonumlari = new List<Nokta>();
    dayanaKonumlari = new List<Nokta>();
    olcumDegerleri = new List<Ornek>();
    DateTime bugun = DateTime.Today;
    DateTime tarih = new DateTime(bugun.Year, bugun.Month, bugun.Day,
11, 48, 0);
    dateTimePicker1.Value = tarih;
    dateTimePicker2.Value = tarih.AddHours(10);
}

/// <summary>
/// listboxa aktarılmış olan ölçüm değerlerini okur
/// </summary>
private void olcumOku()
{
    olcumDegerleri.Clear();
    for (int i = 0; i < listBox2.Items.Count; i++)
    {
        olcumDegerleri.Add(Ornek.olustur(listBox2.Items[i].ToString()));
    }
    List<Ornek> liste = olcumDegerleri.GroupBy(p => p.tarih).Select(g =>
g.First()).ToList();
}

private void sorguolcumoku()
{
    olcumDegerleri.Clear();
    for (int i = 0; i < listBox2.Items.Count; i++)
    {
        olcumDegerleri.Add(Ornek.olustur(listBox2.Items[i].ToString()));
    }
    List<Ornek> liste = olcumDegerleri.GroupBy(p => p.tarih).Select(g =>
g.First()).ToList();

    // ortalama hesaplandı
    if (checkBox1.Checked)
    {
        for (int i = 0; i < liste.Count; i++)
        {
            List<Ornek> uyanlar = olcumDegerleri.FindAll(delegate(Ornek
aranan) { return aranan.deger >= 300; }).OrderBy(p => p.tarih).ToList();

            if (uyanlar.Count > 0)
            {
                #region ortalama alma işlemi

```

```

        double toplam = 0;
        for (int k = 0; k < uyanlar.Count; k++)
        {
            toplam += uyanlar[k].deger;
        }

        toplam = toplam / uyanlar.Count;
        #endregion
        olcumDegerleri[i].deger = toplam;
    }
}
}
}
private void listeGoster()
{
    listDayanak.Items.Clear();
    for (int i = 0; i < referansKonumlari.Count; i++)
    {
        referansKonumlari[i].numara = Convert.ToString(i + 1);
        listDayanak.Items.Add(referansKonumlari[i]);
    }

    listReferans.Items.Clear();
    for (int i = 0; i < dayanaKonumlari.Count; i++)
    {
        dayanaKonumlari[i].numara = Convert.ToString(i + 1);
        listReferans.Items.Add(dayanaKonumlari[i]);
    }

    btnHesapla.Visible = true;
    lblN.Text = olcumDegerleri.Count.ToString();
    // textBox1.Text = (konuular.Count - 1).ToString();
}

private void kutu_Paint(object sender, PaintEventArgs e)
{
    try
    {
        SolidBrush kalem = new SolidBrush(Color.White);
        SolidBrush firca = new SolidBrush(Color.Red);
        SolidBrush firca2 = new SolidBrush(Color.Blue);
        float ebX1, ebY1, ebX2, ebY2;
        ebX1 = 0;
        ebY1 = 0;
        ebX2 = 0;
        ebY2 = 0;
        if (referansKonumlari != null)
        {
            if (referansKonumlari.Count > 0)

```

```

        {
            ebX1 = referansKonumlari.OrderByDescending(p =>
p.X).ToList()[0].X;
            ebY1 = referansKonumlari.OrderByDescending(p =>
p.Y).ToList()[0].Y;
        }
    }
    if (dayanaKonumlari != null)
    {
        if (dayanaKonumlari.Count > 0)
        {
            ebX2 = dayanaKonumlari.OrderByDescending(p =>
p.X).ToList()[0].X;
            ebY2 = dayanaKonumlari.OrderByDescending(p =>
p.Y).ToList()[0].Y;
        }
    }

    int yari = 20;
    float ebX = Math.Max(ebX1, ebX2);
    float ebY = Math.Max(ebY1, ebY2);
    float oranX = (kutu.Width - yari * 2) / ebX;
    float oranY = (kutu.Height - yari * 2) / ebY;
    for (int i = 0; i < referansKonumlari.Count; i++)
    {
        var siradaki = referansKonumlari[i];

        float x = siradaki.X * oranX +( yari ) ;
        float y = kutu.Height - siradaki.Y * oranY - yari ;

        e.Graphics.FillEllipse(firca2, new RectangleF(x - yari / 2, y - yari /
2, yari, yari));
        e.Graphics.DrawString(siradaki.numara.ToString(), new
Font("Arial", 10, FontStyle.Bold), kalem, new PointF(x - 5, y - 5));
    }
    for (int i = 0; i < dayanaKonumlari.Count; i++)
    {
        var siradaki = dayanaKonumlari[i];
        float x = siradaki.X * oranX + (yari);
        float y = kutu.Height - siradaki.Y * oranY - yari;

        e.Graphics.FillEllipse(firca, new RectangleF(x - yari / 2, y - yari / 2,
yari, yari));
        e.Graphics.DrawString(siradaki.numara.ToString(), new
Font("Arial", 10, FontStyle.Bold), kalem, new PointF(x - 5, y - 5));
    }

    e.Graphics.DrawRectangle(new Pen(Color.Gray, 1), new
Rectangle(yari, yari, kutu.Width - 2 * yari, kutu.Height - 2 * yari));

```

```

        e.Graphics.DrawString(ebY.ToString(), new Font("Arial", 10,
        FontStyle.Bold), firca, new PointF(0, 0));
        e.Graphics.DrawString(ebX.ToString(), new Font("Arial", 10,
        FontStyle.Bold), firca, new PointF(kutu.Width - yari, kutu.Height - yari));
        e.Graphics.DrawString(0.ToString(), new Font("Arial", 10,
        FontStyle.Bold), firca, new PointF(0, kutu.Height - yari));
    }
    catch
    {
    }
}

```

```

private void kutu_MouseDoubleClick(object sender, MouseEventArgs e)
{

```

```

    try
    {
        int a = listDayanak.SelectedIndex;

        referansKonumlari[a].X = e.X;
        referansKonumlari[a].Y = e.Y;
        kutu.Invalidate();

        listDayanak.Items.Clear();

        for (int i = 0; i < referansKonumlari.Count; i++)
            listDayanak.Items.Add(referansKonumlari[i]);
    }
    catch
    {
    }
}

```

```

private List<Ornek> ornekleriSec(int sayi)
{

```

```

    DateTime tar1 = dateTimePicker1.Value;
    DateTime tar2 = dateTimePicker2.Value;
    // Hesaplamaya katılacak verilerin seçimi
    List<Ornek> hesaplanacaklar = olcumDegerleri.FindAll(delegate(Ornek
    aranan) { return aranan.tarih >= tar1 && aranan.tarih <= tar2; }).OrderBy(p =>
    p.tarih).ToList();
    List<Ornek> secilenler = hesaplanacaklar.GroupBy(p =>
    p.sensorNo).Select(g => g.First()).ToList().OrderBy(p => p.sensorNo).ToList();
    List<Ornek> sonuc = new List<Ornek>();
    for (int i = 0; i < sayi; i++)
        sonuc.Add(secilenler[i]);
    listSecilenler.Items.Clear();
    for (int i = 0; i < sonuc.Count; i++)
        listSecilenler.Items.Add(sonuc[i].sensorNo.ToString() + " - " +

```

```

sonuc[i].deger.ToString());
    return sonuc;
}

private List<Ornek> ortorneksec(int sayi)
{
    DateTime tar1 = dateTimePicker1.Value;
    DateTime tar2 = dateTimePicker2.Value;
    // Hesaplamaya katılacak verilerin seçimi
    List<Ornek> hesaplanacaklar = olcumDegerleri.FindAll(delegate(Ornek
aranan) { return aranan.tarih >= tar1 && aranan.tarih <= tar2; }).OrderBy(p =>
p.tarih).ToList();
    List<Ornek> uyanlar = hesaplanacaklar.FindAll(delegate(Ornek
aranan){return aranan.deger >= 300;}).GroupBy(p => p.sensorNo).Select(g =>
g.First()).ToList().OrderBy(p => p.sensorNo).ToList();

    List<Ornek> sonuc = new List<Ornek>();
    for (int i = 0; i < sayi; i++)
        sonuc.Add(uyanlar[i]);

    listSecilenler.Items.Clear();
    for (int i = 0; i < sonuc.Count; i++)
        listSecilenler.Items.Add(sonuc[i].sensorNo.ToString() + " - " +
sonuc[i].deger.ToString());

    return sonuc;
}

List<IDWEnterpolasyon> algoritmalar = new List<IDWEnterpolasyon>();
private void btnHesapla_Click(object sender, EventArgs e)
{
    try
    {
        algoritmalar.Clear();
        listTahminDegerleri.Items.Clear();

        olcumOku();
        // Seçilenler
        List<Ornek> secilenler =
ornekleriSec(Convert.ToInt16(lblSensorSayisi.Text));

        for (int i = 0; i < dayanaKonumlari.Count; i++)
        {
            dayanaKonumlari[i].ornekleri =
olcumDegerleri.FindAll(delegate(Ornek aranan) { return
Convert.ToInt32(aranan.sensorNo) ==
Convert.ToInt32(dayanaKonumlari[i].numara) - 1; });
        }
        for (int i = 0; i < dayanaKonumlari.Count; i++)

```

```

        {
            IDWEnterpolasyon algo = new
IDWEnterpolasyon(referansKonumlari, secilenler);
            algo.pDegeri = Convert.ToInt16(txtPDegeri.Text);
            algo.referans = dayanaKonumlari[i];
            double siradaki = algo.hesapla();
            algoritmalar.Add(algo);
            listTahminDegerleri.Items.Add(algo);
        }
        tabControl1.SelectedIndex = 2;

    }
    catch (Exception hata)
    {
        MessageBox.Show(hata.ToString());
    }
}
private void Form1_Load(object sender, EventArgs e)
{
    this.WindowState = FormWindowState.Maximized;
    btnHesapla.Visible = false;
}

private void btnOku_Click(object sender, EventArgs e)
{
    try
    {
        openFileDialog1.ShowDialog();
        System.IO.StreamReader okuyucu = new
System.IO.StreamReader(openFileDialog1.FileName);
        string[] parcalar = okuyucu.ReadToEnd().Split('\n');
        okuyucu.Close();

        listBox2.Items.Clear();
        for (int i = 0; i < parcalar.Length; i++)
        {
            if (parcalar[i].Length == 0)
                continue;
            listBox2.Items.Add(parcalar[i].Trim());
        }

    }
    catch
    {
    }
}

private void hToolStripMenuItem_Click(object sender, EventArgs e)

```

```

{
    try
    {
        Nokta_Ekrani ekran = new Nokta_Ekrani(true);
        ekran.ShowDialog();
        if (ekran.noktalar != null)
            this.referansKonumlari = ekran.noktalar;
        listeGoster();
        kutu.Invalidate();
    }
    catch
    {
    }
}

private void kutu_Click(object sender, EventArgs e)
{
}

private void button1_Click(object sender, EventArgs e)
{
    Nokta referans = referansKonumlari[referansKonumlari.Count - 1];
    Nokta yeni = new Nokta();
    yeni.X = referans.X;
    // Tahminle bulunan noktaları, dayanak noktası verileirne ekler.
    for (int i = 0; i < dayanaKonumlari.Count; i++)
    {
        string sonuc = String.Format("{0}\t{1}\t{2}", i+1,
listTahminDegerleri.Items[i].ToString(), DateTime.Now.ToLongTimeString());
        listBox2.Items.Add(sonuc);
    }
    MessageBox.Show("Yeni değerler eklendi!");
}

public void noktaEkle(Nokta yeni)
{
    if (referansKonumlari == null)
        referansKonumlari = new List<Nokta>();

    referansKonumlari.Add(yeni);

    listeGoster();
    kutu.Invalidate();
}

public void referansEkle(Nokta yeni)
{
    if (dayanaKonumlari == null)

```



```

        dayanaKonumlari = new List<Nokta>();
        dayanaKonumlari.Add(yeni);
        listeGoster();
        kutu.Invalidate();
    }

    private void referansKonumlarıBelirleToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        try
        {
            Nokta_Ekrani ekran = new Nokta_Ekrani(false);
            ekran.ShowDialog();
            if (ekran.noktalar != null)
                this.dayanaKonumlari = ekran.noktalar;
            kutu.Invalidate();
            listeGoster();
        }
        catch
        {
        }
    }

    private void yeniSensörEkleToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        frmYeniSensor sa = new frmYeniSensor(this, true);
        sa.ShowDialog();
    }

    private void referansNoktasıEkleToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        frmYeniSensor sa = new frmYeniSensor(this, false);
        sa.ShowDialog();
    }

    private void listDayanak_SelectedIndexChanged(object sender, EventArgs
e)
    {
        try
        {
            int a = listDayanak.SelectedIndex;
            txtX.Text = referansKonumlari[a].X.ToString();
            txtY.Text = referansKonumlari[a].Y.ToString();
            propertyGrid1.SelectedObject = referansKonumlari[a];
        }
    }

```

```

    }
    catch
    {
    }
}
}
private void listReferans_SelectedIndexChanged(object sender, EventArgs
e)
{
    try
    {
        int a = listReferans.SelectedIndex;
        textBox1.Text = dayanaKonumlari[a].X.ToString();
        textBox3.Text = dayanaKonumlari[a].Y.ToString();
    }
    catch
    {
    }
}
private void btnDayanakDegistir_Click(object sender, EventArgs e)
{
    try
    {
        int a = listDayanak.SelectedIndex;
        referansKonumlari[a].X = Convert.ToInt32(txtX.Text);
        referansKonumlari[a].Y = Convert.ToInt32(txtY.Text);
        listeGoster();
        kutu.Invalidate();
        listDayanak.SelectedIndex = a;
    }
    catch
    {
    }
}
private void btnReferansDegistir_Click(object sender, EventArgs e)
{
    try
    {
        int a = listReferans.SelectedIndex;

        dayanaKonumlari[a].X = Convert.ToInt32(textBox1.Text);
        dayanaKonumlari[a].Y = Convert.ToInt32(textBox3.Text);
        listeGoster();
        kutu.Invalidate();
        listReferans.SelectedIndex = a;
    }
    catch
    {
    }
}

```

```

    }
    catch
    {
    }
}

private void btnAktar_Click(object sender, EventArgs e)
{
    try
    {

        int a = listReferans.SelectedIndex;

        var yeni = dayanaKonumlari[a];
        referansKonumlari.Add(yeni);
        dayanaKonumlari.RemoveAt(a);

        listeGoster();
        kutu.Invalidate();
    }
    catch
    {
    }
}

private void btnKaydet_Click(object sender, EventArgs e)
{
    try
    {
        StreamWriter yazici1 = new StreamWriter(@"sonuc.txt");

        for (int i = 0; i < dayanaKonumlari.Count; i++)
        {
            string satir = String.Format("{0} {1} {2}", dayanaKonumlari[i].X,
            dayanaKonumlari[i].Y, listTahminDegerleri.Items[i].ToString().Replace(',', '.'));
            yazici1.WriteLine(satir);
        }
        yazici1.Close();
        StreamWriter yazici = new StreamWriter(@"veri.txt");
        List<Ornek> secilenler =
        ornekleriSec(Convert.ToInt16(lblSensorSayisi.Text));

        for (int i = 0; i < referansKonumlari.Count; i++)
        {
            string satir = String.Format("{0} {1} {2}", referansKonumlari[i].X,
            referansKonumlari[i].Y, secilenler[i].deger.ToString().Replace(',', '.'));

```

```

        yazici.WriteLine(satir);
    }
    yazici.Close();
    MessageBox.Show("İşlem tamamlandı");
}
catch(Exception hata)
{
    MessageBox.Show(hata.ToString());
}
}

private void listTahminDegerleri_SelectedIndexChanged(object sender,
EventArgs e)
{
    try
    {
        listSecilenler.Items.Clear();
        listAgirlik.Items.Clear();
        listMesafeler.Items.Clear();

        IDWEnterpolasyon secilen =
        algoritmalar[listTahminDegerleri.SelectedIndex];

        listMesafeler.Items.Clear();
        listAgirlik.Items.Clear();

        for (int i = 0; i < secilen.agirliklar.Count; i++)
            listAgirlik.Items.Add(secilen.agirliklar[i]);

        for (int i = 0; i < secilen.mesafeler.Count; i++)
            listMesafeler.Items.Add(secilen.mesafeler[i]);

        for (int i = 0; i < secilen.secilenGozlemDegerleri.Count; i++)

listSecilenler.Items.Add(secilen.secilenGozlemDegerleri[i].ToString());

    }
    catch
    {
    }
}

private void btnAralikHesapla_Click(object sender, EventArgs e)
{
    try
    {
        algoritmalar.Clear();
    }
}

```

```

listTahminDegerleri.Items.Clear();

olcumOku();
List<Ornek> secilenler =
ornekleriSec(Convert.ToInt16(lblSensorSayisi.Text));

double cap1 = Convert.ToDouble(txtCap1.Text);
double cap2 = Convert.ToDouble(txtCap2.Text);

for (int i = 0; i < dayanaKonumlari.Count; i++)
{
    IDWEnterpolasyon algo = new
IDWEnterpolasyon(referansKonumlari, secilenler);
    algo.capVarmi = true;

    algo.pDegeri = Convert.ToInt16(txtPDegeri.Text);
    algo.referans = dayanaKonumlari[i];
    algo.cap1 = cap1;
    algo.cap2 = cap2;
    double siradaki = algo.hesapla();
    algoritmalar.Add(algo);
    listTahminDegerleri.Items.Add(algo);
}

for (int i = 0; i < dayanaKonumlari.Count; i++)
{
    dayanaKonumlari[i].ornekleri =
olcumDegerleri.FindAll(delegate(Ornek aranan) { return
Convert.ToInt32(aranan.sensorNo) ==
Convert.ToInt32(dayanaKonumlari[i].numara) - 1; });
}

tabControl1.SelectedIndex = 2;

}
catch (Exception hata)
{
    MessageBox.Show(hata.ToString());
}
}

private void button2_Click(object sender, EventArgs e)
{
    try
    {
        algoritmalar.Clear();
        listTahminDegerleri.Items.Clear();

        sorguolcumoku();
    }
}

```

```

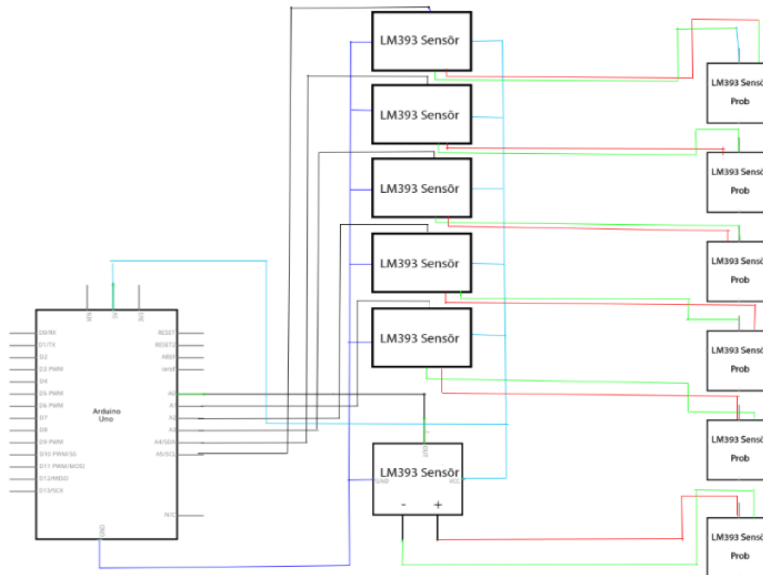
// Seçilenler
List<Ornek> secilenler =
ortorneksec(Convert.ToInt16(lblSensorSayisi.Text));

for (int i = 0; i < dayanaKonumlari.Count; i++)
{
    dayanaKonumlari[i].ornekleri =
olcumDegerleri.FindAll(delegate(Ornek aranan) { return
Convert.ToInt32(aranan.sensorNo) ==
Convert.ToInt32(dayanaKonumlari[i].numara) - 1; });
}

for (int i = 0; i < dayanaKonumlari.Count; i++)
{
    IDWEnterpolasyon algo = new
IDWEnterpolasyon(referansKonumlari, secilenler);
    algo.pDegeri = Convert.ToInt16(txtPDegeri.Text);
    algo.referans = dayanaKonumlari[i];
    double siradaki = algo.hesapla();
    algoritmalar.Add(algo);
    listTahminDegerleri.Items.Add(algo);
}
tabControl1.SelectedIndex = 2;
}
catch (Exception hata)
{
    MessageBox.Show(hata.ToString());
}
}
}
}
}
}
}

```

EK B : DEVRE ŞEMASI



ÖZGEÇMİŞ

Ad Soyad : Hilal TÜRK
Doğum Yeri ve Tarihi : Erzurum, 20.02.1987
E-Posta : hlturk25@gmail.com

ÖĞRENİM DURUMU

Lisans : 2009, Süleyman Demirel Üniversitesi, Teknik Eğitim Fakültesi, Bilgisayar ve Kontrol Öğretmenliği
Lisans : 2016, Trakya Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği

MESLEKİ DENEYİM VE ÖDÜLLER

- 2009 yılında öğretmenliğe başladı. 2012 yılına kadar farklı kurumlarda mesleğe devam etti.
- 2012 yılında Avrasya Üniversitesi'nde öğretim görevlisi olarak çalışmaya başladı. Aynı yıl Avrasya Üniversitesi'nde bilgi-işlem daire başkanlığı ve MYO müdür yardımcılığı görevleri yaptı.
- 2014 yılında İstanbul Ayvansaray Üniversitesi'nde öğretim görevlisi olarak çalışmaya başladı.
- 2015 yılında İstanbul Teknik Üniversitesi Bilişim Enstitüsü bünyesinde Bilişim Uygulamaları anabilim dalında yüksek lisans yapmaya başladı.
- İstanbul Ayvansaray Üniversitesi'nde program başkanı olarak görevine devam ediyor.

TEZDEN TÜRETİLEN YAYINLAR/SUNUMLAR

- **Türk, H., Töreyn B.U.** 2018. Ev içi Tarım Uygulamaları için Toprak Nemi Tahmininin Ters-Uzaklık Yöntemi ile Kestirimi Soil Moisture Estimation with Inverse Distance Weighting for Home Agriculture Applications. 26. *IEEE Sinyal İşleme Ve İletişim Uygulamaları Kurultayı, 2-5 Mayıs 2018, Türkiye, İzmir*