

4. UYGULAMALAR

4.1 Ara Değer Bulma ve Veri Gridleme

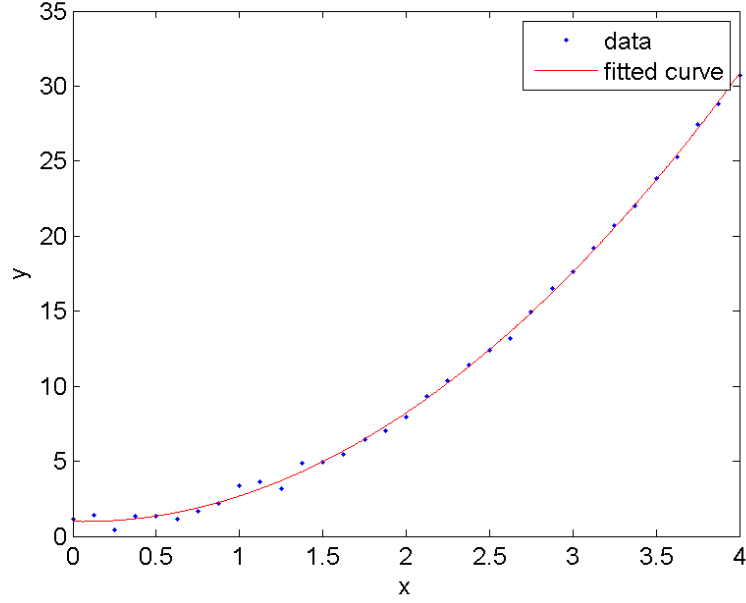
Günümüz mühendislik ve bilimsel ölçme işlemleri sürekli olayların frekans, uzaklık, zaman gibi bağımsız bir değişkene bağlı olarak ayrı noktalarda örneklenmesi ile yürütülür. Ölçme sıklığının artırılması ile sürekli olaya en yakın bir kayıt elde edilebilir. Bunun tersi olarak yeteri sıklıkta örneklenmeyen bir kayıt, ölçülen süreçten tamamen farklı bir davranış sergileyebilir. Ölçme sıklığı (örnekleme) çok sayıda etmene bağlıdır. Ancak birçok durumda örnek sayısını artırmanın, verileri saklama, ölçme maliyeti, veri işlem gereksinimleri gibi konularda zorluklar çıkaracağı açıktır. Bu nedenlerden, süreci yeterince yakınlıkta tanımlayacak örnekleme ve ölçme aralıklarının bulunması kendi başına bir mühendislik problemi olmaktadır.

Ölçme aralığı ve buna bağlı başka parametreleri saptanarak ayrı noktalarda ölçülmüş veriler, ilgili süreci tanımlama, bir mühendislik ya da bilim problemini çözme, bir durumu ortaya koyma gibi amaçlarla çeşitli süreçlerden geçirilebilmektedir. Bu süreçler sırasında gerek gösterim gerekse hesaplama amaçları ile ölçüm yapılmamış bir zaman, uzaklık ya da frekans değerindeki veriye gereksinim duyulabilmektedir. Ölçülmüş noktalardaki veri davranışından yola çıkılarak ara bir noktadaki değerler çeşitli yollarla hesaplanabilmektedir. Bunlardan ilki veriye davranışına uygun bir sürekli fonksiyon çakıştırmak ve bu sayede istenilen herhangi bir bağımsız değişken değerine karşılık gelen verinin hesaplanmasını olanaklı kılmaktır. MATLAB araç kutuları içerisinde yalnızca bu amaçla geliştirilmiş bir araç kutusu bulunmaktadır (Curve Fitting Toolbox). Bu araç kutusu arayüzüne, komut satırından `cftool` yazılarak erişilebilir. Arayüz açılmadan da araç kutusundaki fonksiyonlar ayrı ayrı çağırılarak kullanılabilir. Aşağıdaki örnekte ikinci dereceden bir fonksiyon oluşturularak bir miktar gürültü eklenmiştir. Oluşturulan bu veri, deneme amaçlı kullanılarak ikinci dereceden bir polinomun çakıştırılması ile ilgili MATLAB programı aşağıda verilmiştir. Polinoma çakıştırma işleminin uygulamadaki karşılığı veri ile uyumlu polinomun katsayılarının hesaplanmasıdır. Verinin davranışına bakılarak değişik derecelerden polinomlar çakıştırma fonksiyonu olarak seçilebilir. Polinoma çakıştırma işlemi için Bölüm 3.2'de alıştırmalar bölümünde verilen denklem sisteminin oluşturularak çözülmesi de kullanılabilir.

```
x=0:.125:4;
y=2*x.^2-0.5*x+1;
y1=y+0.25*randn(size(y));
f=fit(x',y1','poly2');%'poly1': doğru 'poly3': üçüncü derece ...
f
plot(f,x,y1)
```

f =

```
Linear model Poly2:
f(x) = p1*x^2 + p2*x + p3
Coefficients (with 95% confidence bounds):
p1 = 1.978 (1.895, 2.061)
p2 = -0.4339 (-0.7795, -0.08833)
p3 = 0.9811 (0.6824, 1.28)
```



Şekil 4. 1 Bir veri kümesine ikinci dereceden bir polinom çakıştırılması. Veriler mavi noktalar ile çakıştırılan polinom ise düz kırmızı eğri ile gösterilmiştir.

Sayısal veriye $f(x) = p_1 \cdot x^2 + p_2 \cdot x + p_3$ gibi bir polinom çakıştırıldığında x için herhangi bir değer verilerek veriye o noktada bir yaklaşım sağlanabilir. Bu işlemin ne anlama geldiği Şekil 4.1'de verilen veri noktaları ve ona çakıştırılan eğrinin davranışına bakılarak anlaşılabilir. Veriye davranışına bağlı olarak, üstel bir fonksiyon, sinüzoidallerin toplamı, Fourier serisi gibi farklı sürekli fonksiyonlar da çakıştırılabilir.

MATLAB kütüphanesinde aradeğer bulma ile ilgili kullanılan bir diğer fonksiyon `interp1` adını taşımaktadır. Fonksiyon doğrusal ara değer bulma yöntemini kullanmaktadır. Fonksiyonun genel kullanımını aşağıdaki gibidir:

```
di = interp(x,y,xi);
```

Yukarıdaki yazılışta x 'e bağlı y değerlerinden yola çıkılarak xi gibi bir ara noktadaki veri kestirilmeye çalışılmaktadır. Hesaplanan değer di değişkenine aktarılmaktadır. Aşağıdaki örnek kullanımın daha iyi anlaşılmasına yardımcı olacaktır.

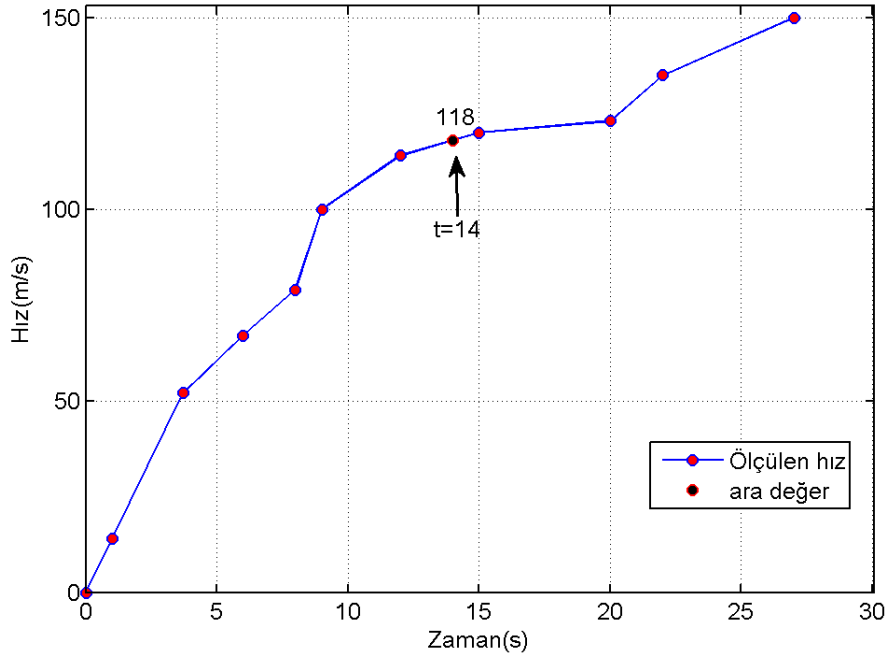
```
clear all;close all;clc
% Bir aracın zamana karşı ölçülmüş hız değerleri
t=[0 1 3.7 6 8 9 12 15 20 22 27];
v=[0 14 52 67 79 100 114 120 123 135 150];

%Aracın t=14 saniyedeki hızını bulunuz.
ti=14;
vi=interp1(t,v,ti);%t=14 deki aradeğerin hesaplanması
```

```

plot(t,v,'b-o','MarkerFaceColor','r')
hold on
plot(14,vi,'r o','MarkerFaceColor','k')
%grafiğin düzenlenmesi
grid on;box on
xlabel('Zaman(s)')
ylabel('Hız(m/s)')
legend('Ölçülen hız','ara değer')

```



Şekil 4. 2 Bir aracın hızının ölçülmeyen bir ara zaman değerinde hesaplanması

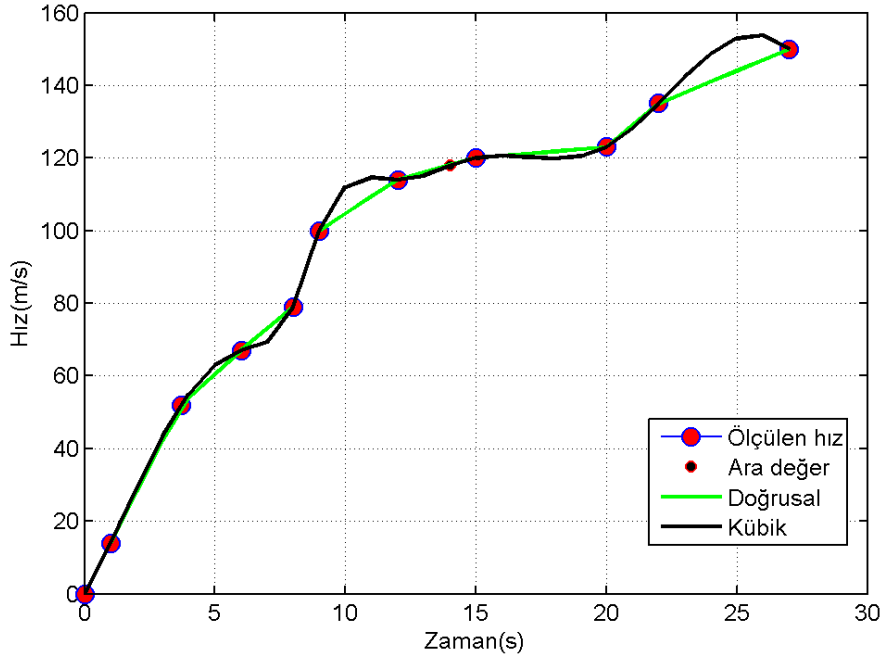
Bu amaçla kullanılacak bir diğer MATLAB fonksiyonu `spline` ismi ile kütüphanede yer almaktadır. Fonksiyonun, birbirine komşu iki veri noktası arasında doğrusal değil üçüncü derece bir polinom davranışı gösterdiği varsayımı ile çalışır. Kübik ara değer bulma fonksiyonu önceki örnek için

```

vii=spline(t,v,ti)
vii=
    117.7473

```

şeklinde çağrılarak kullanılabilir.



Şekil 4. 3 Doğrusal ve kübik aradeğer bulma fonksiyonları ile yeniden örneklenen veri

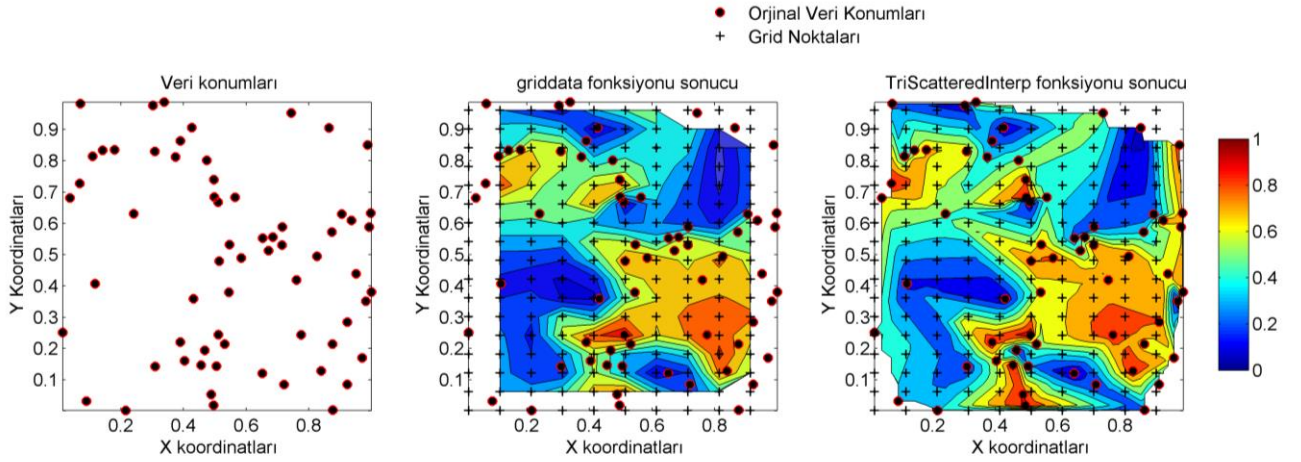
Birden çok parametre, konum ya da değişkene bağlı verilerin belirli bir sistematik ile ara değerlerinin bulunması işlemi yukarıda verilen örneklerden biraz daha karışık hale gelebilmektedir. Örnek olarak bir çalışma alanında x ve y konumlarına bağlı ölçülen bir jeofizik veriyi ele alabiliriz. Bu tür bir veri kümesinden beklenen çalışma alanını x ve y yönlerinde eşit aralıklarla taramasıdır. Ancak bu çok zaman mümkün olmaz. Ölçülen verilerin görüntülenmesi ya da işlenmesi sırasında bu verilerin bir grid (ızgara) üzerine düşen noktadaki değerlerine gerek duyulur. Bu amaçla veri gridleme şeklinde adlandırdığımız yöntem grubuna başvururuz. Burada işlemin ayrıntılarına girilmeyip yalnızca MATLAB dilindeki uygulamasına örnek verilecek ve gerekli açıklamalar örnek üzerinde yapılacaktır.

```
% Veri Gridleme ve Konturlama Uygulaması
clear all;close all;clc
% Sınama verisi
% Rastgele x,y konumlarında ölçüldüğü varsayılan z değerleri üretiliyor
x=rand(64,1);y=rand(64,1);z=rand(64,1);
dx=.1; % x yönünde grid aralığı
dy=.06;% y yönünde grid aralığı
% Izgaranın düğüm noktaları oluşturuluyor
xi=min(x):dx:max(x);
yi=min(y):dy:max(y);
%grid ızgarası oluşturuluyor
[X,Y]=meshgrid(xi,yi);
%Veri griddata fonksiyonu ile gridleniyor x ve y sırası ile verilerin konumları, z
bu konularda tanımlı veri, X ve Y ise verilerin yeniden gridleneceği ızgara
noktalarıdır.
Z=griddata(x,y,z,X,Y,'cubic'); Tırnak içinde gridleme işlemi ile ilgili bir seçenek
yazılmıştır. Diğer seçenekler 'linear' ve 'nearest' şeklindedir.
```

```

%verilerin konumları çizdiriliyor
subplot(131)
plot(x,y,'r o','MarkerFaceColor','k');box on;axis image
xlabel('X koordinatları');ylabel('Y Koordinatları')
%grid ızgarası çizdiriliyor
subplot(132)
plot(x,y,'r o','MarkerFaceColor','k');box on;axis image
hold on
plot(X,Y,'k +')
plot(X,Y,'k .')
%gridlenmiş değerler çizdiriliyor
contourf(xi,yi,Z)
alpha(.75);
legend('Orjinal Veri Konumları','Grid Noktaları')
xlabel('X koordinatları');ylabel('Y Koordinatları')

```



Şekil 4. 4 Bir düzlemde dağınık olarak bulunan veri konumları kullanılarak yapılan gridleme işleminin sonucu

```

%griddata fonksiyonu MATLAB'ın sonraki sürümlerinde kaldırılacaktır. Onun
%yerine TriScatteredInterp fonksiyonu aşağıdaki gibi kullanılabilir.
F = TriScatteredInterp(x,y,z);
[qx,qy] = meshgrid(sort(x),sort(y));
qz = F(qx,qy);
%Sonuçların çizdirilmesi
subplot(133)
contourf(sort(x),sort(y),qz);hold on
plot(x,y,'r o','MarkerFaceColor','k');box on;axis image
plot(X,Y,'k +')
plot(X,Y,'k .')

```

Yukarıdaki örnekte, verilerin ölçüldüğü noktalar dışında kalan ve bir ızgaranın köşe noktalarında yer alan yeni noktalarda veriye bir yaklaşım yapılmaktadır. Bu örnekte birbirine komşu noktalar üçgenler oluşturacak şekilde birleştirilmekte ve üçgenin kenarları üzerinde kübik yaklaşımla aradegerler hesaplanmaktadır. Programda kullanılan **meshgrid** fonksiyonu verilen x ve y değerlerinin olası tüm birleşimlerinden oluşan bir grid ağı ya da ızgarası oluşturmaktadır. **contourf** fonksiyonu ise x_i ve y_i konumlarına bağlı olarak hesaplanan yeni değerleri renklendirilmiş bir kontur çizgileri grubu ile görselleştirmek amacıyla kullanılmıştır. Bu fonksiyonlar hakkında geniş bilgi için komut satırında **doc contourf** ve **doc meshgrid** yazılabilir. Örnekte 2-boyutta veri gridlemek için kullanılan **griddata** fonksiyonuna benzer olarak 3-boyutlu veri gridleme işlemi için **griddata3** fonksiyonu aşağıdaki yazımla kullanılabilir.

```
w = griddata3(x,y,z,v,xi,yi,zi)
```

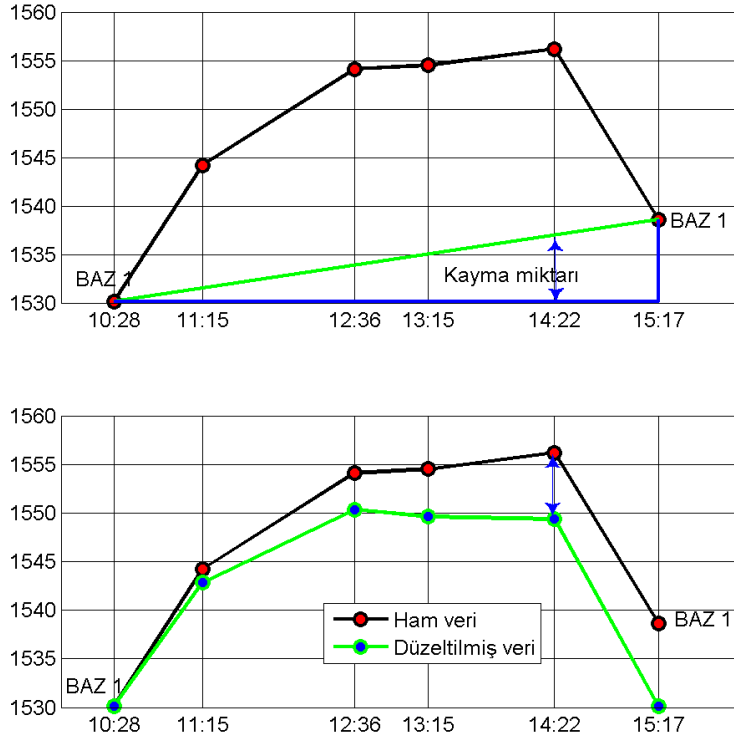
Burada x,y ve z verinin ait olduğu konum koordinatları, v verinin kendisi, x_i,y_i ve z_i ise verinin yeniden gridleneceği koordinatlarıdır. **griddata** ve **griddata3** MATLAB'ın yeni sürümlerinde kaldırılacağından **TriScatteredInterp** fonksiyonunun kullanımı da ayrıca verilmiştir.

Veri gridleme işleminde var olan konum ve bilgilerden yola çıkılarak veri bulunmayan bir konumdaki veriye yaklaşım yapıldığından hesaplama yapılan nokta yakınında veri bulunması gereklidir. Başka bir deyişle gridleme yapılmak istenen nokta ya da noktalar veri noktaları arasında kalmıyor ve çalışma alanı dışına taşıyorsa bu noktalarda gridleme yapılamayacağı gözönüne alınmalıdır.

Alıştırmalar

1. Gravite ölçümlerinde genellikle ölçüm cihazından kaynaklanan bazı hatalar ortaya çıkar. Bu hata belirli aralıklarla baz olarak adlandırılan sabit bir noktaya dönerek ölçümün tekrarlanması yoluyla düzeltilir. Beklenti, aynı konumda aynı okumanın tekrarlanabilmesidir. Ancak bu gerçekleşmez. Baz noktasına tekrarlı gelişlerde okunan değerlere bir doğru çakıştırarak yapılacak düzeltme hesaplanabilir. Geliş zamanları arasındaki okumalara bu düzeltme eklenir ya da çıkarılır. Aşağıdaki okuma değerleri için bu düzeltmeleri yapan bir program yazınız.

ISTASYON	Saat	Okuma
BAZ1	10:28	1530.15
STA1	11:15	1544.20
STA2	12:36	1554.10
STA3	13:15	1554.50
STA4	14:22	1556.20
BAZ1	15:17	1538.60



Şekil 4. 5 Gravite ölçümlerine yapılan kayma (drift) düzeltmesi

2. <http://goo.gl/e3B1Uy> adresinden indireceğiniz sıkıştırılmış dizin içinde sistematik olarak isimlendirilmiş veri dosyaları bulunmaktadır. Dosyalardan birinin içeriği aşağıdaki gibidir:

```

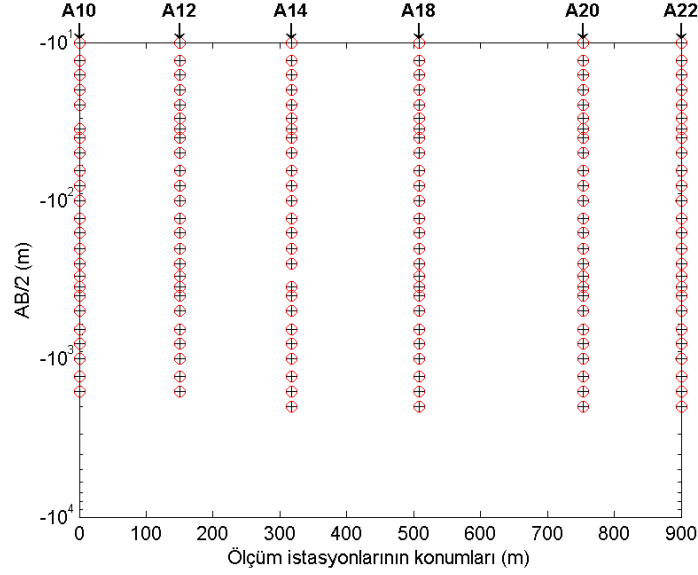
0 0 51
24
10      2      46
13      2      45
16      2      42.5
20      2      42
25      2      41.5
35      2      41
40      2      44
50      2      44
65      10     38
80      10     35
100     10     30
130     10     25.5
160     10     23
200     10     20
250     40     18.5
300     40     17
350     40     17.5
400     40     17
500     80     18.5
650     80     20.5
800     80     18
1000    200    15
1300    200    15.5
1600    200    16.5

```

Dosya bir istasyonda ölçülmüş düşey elektrik sondajı verilerini içermektedir. Birinci satırda ölçüm istasyonunun x,y ve z konumları verilmektedir. İkinci satırda kaç adet veri olduğu bilgisi

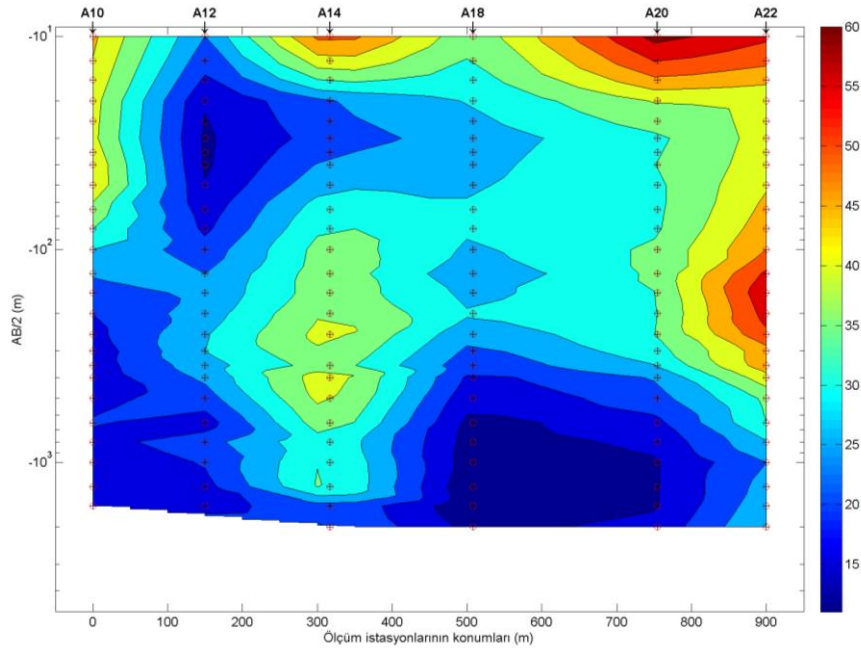
yer almaktadır. Sonraki satırların sütunlarında sırasıyla AB/2 (düşey eksen değerleri), MN/2 problemin çözümü ile ilgisiz bir ölçme parametresi ve ölçülen görünür öz direnç yer almaktadır.

Ölçüm noktalarının yatay ve düşey konumlarının görünümü aşağıdaki şekilde verilmiştir.



Şekil 4. 6 Ölçümlerin yatay ve düşey konumları

Verilen bilgiler ışığında 6 istasyonda ölçülmüş olan görünür öz direnç değerlerini yatay eksen istasyon konumları, düşey eksen AB/2 olacak şekilde gridleyiniz. Yatay grid aralığı 50 m, düşey grid aralığı ise 10m alınız. Sonuçları contour fonksiyonu ile görselleştiriniz. Yükseklik değerlerini dikkate almayınız. Tüm işlemleri doğru yaptığınızda aşağıdaki gibi bir grafik elde etmeniz beklenmektedir.



Şekil 4. 7 Verilerin gridlenerek renklendirilmiş kontur alıkları ile görselleştirilmesi