Antenna Array Processing

****

****

HW9

****

****

********************************** Mohammadreza Arani ::::::::::::: 810100511

****

** 1401/09/20

****

```
clear; clc; close all;
```

در این تمرین می خواهیم آرایه ای عمودی از آنتن ها را در گیرنده به گونه ای بچینیم که بردارهای steering vector کمترین correlation را باهم داشته باشند تا عملکرد زاویه یابی به بهترین نحو ممکن باشد. فرکانس سیگنال حامل را $f_c = 300\ MHz$ و $d_{max} = 10\ m$ و $d_{min} = 0\ m$ در نظر بگیرید. زوایای ارتفاعی را نیز از ۹۰- درجه تا ۹۰+ درجه در نظر بگیرید.

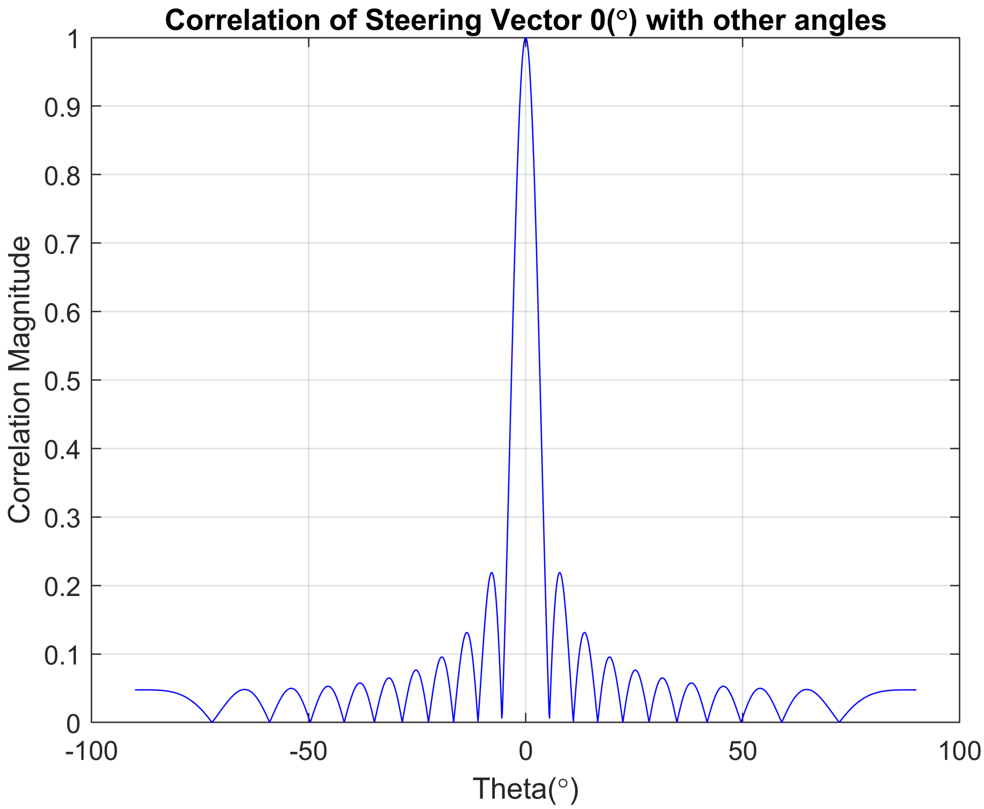
Initialization:

```
theta = -90:0.1:90;

d_min = 0;
d_max = 10;
fc =    300e+06; % 300MHz
c = 3e+08;%Speed in Air

Lambda =  c/fc;% Lambda*fc = c => Lambda = c/fc
K = 2*pi/Lambda;
```

Part 1)

الف) اگر خاطرتان باشد در درس آنتن همیشه فواصل آنتن ها را برابر $\frac{\lambda}{2}$ در نظر می گرفتند. شما هم آنتن ها
با همین فواصل به صورت یکنواخت بچینید و سپس اندازه ی correlation بردارهای steering زوایا
مختلف را با بردار steering متناظر با صفر درجه رسم کنید. به این شکل پترن آرایه هم گفته می شود. تو
داشته باشید در این حالت شما ۲۱ آنتن در چیدمان آرایه استفاده کرده اید.

```
D = (d_min:Lambda/2:d_max)'; % Antenna Positions <Uniform Linear Array> - d = Lambda/2
M = length(D);

% Using Given Code of steering.m

MAP = exp(1j*K*D*sind(theta)) ;  % Matrix of Steering Vectors

zav=0;

a=exp(1j*K*D*sind(zav)); % Steering Vector corresponding to 0 degrees

g=abs(a'*MAP); % Inner Product of  steering Vector for 0 degrees and MAP
g=g/max(g); % Normalization of g
plot(theta,(g),'b');
grid on
title("Correlation of Steering Vector 0(\circ) with other angles")
xlabel("Theta(\circ)")
ylabel("Correlation Magnitude")
```




```
% We have Used 21 elements of Antenna! --> Above Figure is called:
% Antenna
% Pattern
```

Part 2)

ب) حال می خواهیم تا حد ممکن اندازه ی correlation ها در حد همان قسمت الف بماند ولی تعداد آنتن کمتری را در چیدمان آرایه استفاده کنیم. طبیعتا ناگزیر خواهیم بود تا آنتن ها را به صورت غیر یکنواخت بچینیم. مطابق آنچه در جلسه ی ۲۲م درس شرح داده شد، سعی کنید با تعداد آنتن کمتر تقریبا به همان correlation های قسمت الف دست یابید.

برای حل قسمت ب به نکات زیر توجه داشته باشید:

- ✓ پکیج cvx را دانلود کنید و ابتدا cvx_setup را یک بار run کنید.
- ✓ کد متلب تان را در همان فولدر cvx قرار دهید. ممکن است اگر پکیج را add کنید با error هایی مواجه شوید.
- ✓ گریدبندی زاویه ها را تا حدی ریز کنید که کامپیوترتان قادر به حل مساله باشد. طبیعتا اول با گریدهای بزرگتر شروع کنید و وقتی مطمئن شدید کدتان درست است، گریدبندی را کم کم ریزتر کنید. برای مکان آنتن ها هم همین مورد را در نظر داشته باشید.

```
% cvx_setup  --- >  DOne Before
% CVX Path has been Added!

% Starting the algorithm:

A  = exp(1j*K*D*sind(theta));

 cvx_begin
```

```
Warning: A non-empty cvx problem already exists in this scope.
   It is being overwritten.
```

```

        variable W(M);
        variable t;
        %variable Z(n,n) symmetric;

        minimize( t ) % Minimize a Convex Function

        subject to % Our Constraints:
            ones(1,M)*W == 1;
            abs(W'*A)        <= t;

  cvx_end
```

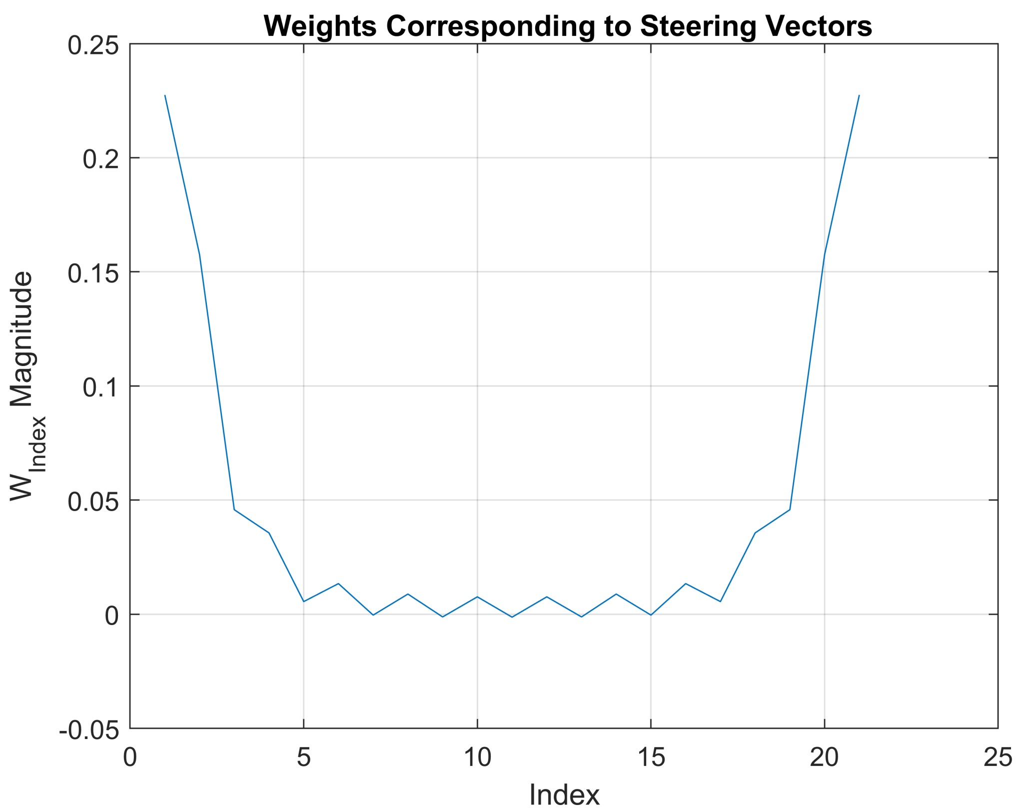
```
Calling SDPT3 4.0: 7204 variables, 1823 equality constraints
   For improved efficiency, SDPT3 is solving the dual problem.
------------------------------------------------------------

 num. of constraints = 1823
 dim. of socp   var  = 5402,   num. of socp blk  = 1801
 dim. of linear var  = 1801
 dim. of free   var  =  1 *** convert ublk to lblk
*******************************************************************
   SDPT3: Infeasible path-following algorithms
```

```
*******************************************************************
 version  predcorr  gam  expon  scale_data
    NT      1      0.000   1        0
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj    cputime
-------------------------------------------------------------------
 0|0.000|0.000|2.4e+03|1.1e+03|6.6e+06| 3.642686e-11  0.000000e+00| 0:0:00| spchol  1  1
 1|0.968|0.978|7.6e+01|2.4e+01|2.1e+05|-1.055986e+00 -8.399412e+01| 0:0:01| spchol  1  1
 2|0.957|0.885|3.3e+00|2.8e+00|9.3e+03|-8.978524e-03 -8.664693e+01| 0:0:01| spchol  1  1
 3|0.993|0.998|2.3e-02|9.6e-03|1.3e+02|-2.543817e-03 -7.144787e+01| 0:0:01| spchol  1  1
 4|1.000|0.210|2.2e-05|1.2e-02|6.0e+01|-4.488817e-03 -5.972472e+01| 0:0:01| spchol  1  1
 5|1.000|0.834|1.1e-05|2.1e-03|1.0e+01|-7.067311e-03 -1.005781e+01| 0:0:01| spchol  1  1
 6|1.000|0.123|3.4e-06|1.8e-03|8.8e+00|-2.172815e-02 -8.856664e+00| 0:0:01| spchol  1  1
 7|1.000|0.373|1.1e-06|1.1e-03|5.7e+00|-3.570528e-02 -5.696144e+00| 0:0:01| spchol  1  1
 8|1.000|0.308|1.8e-06|7.9e-04|4.0e+00|-5.542682e-02 -4.039854e+00| 0:0:01| spchol  1  1
 9|1.000|0.205|5.6e-07|6.3e-04|3.2e+00|-8.556018e-02 -3.297794e+00| 0:0:01| spchol  1  1
10|1.000|0.259|3.7e-07|4.6e-04|2.4e+00|-1.347718e-01 -2.579511e+00| 0:0:01| spchol  1  1
11|1.000|0.233|1.3e-07|3.6e-04|1.9e+00|-2.300499e-01 -2.130102e+00| 0:0:01| spchol  1  1
12|1.000|0.316|6.5e-08|2.4e-04|1.3e+00|-4.014790e-01 -1.703309e+00| 0:0:01| spchol  1  1
13|0.918|0.326|2.2e-08|1.6e-04|8.1e-01|-6.338652e-01 -1.442553e+00| 0:0:01| spchol  2  2
14|0.933|0.484|7.7e-09|8.4e-05|3.8e-01|-8.228417e-01 -1.204758e+00| 0:0:01| spchol  2  2
15|0.976|0.382|1.2e-09|5.3e-05|1.9e-01|-9.231295e-01 -1.116512e+00| 0:0:01| spchol  2  2
16|0.989|0.842|4.5e-10|1.1e-05|3.7e-02|-9.674924e-01 -1.004458e+00| 0:0:01| spchol  2  1
17|0.878|0.919|1.0e-10|1.6e-06|8.7e-03|-9.915881e-01 -1.000276e+00| 0:0:01| spchol  2  2
18|0.816|0.940|1.9e-11|3.5e-07|1.9e-03|-9.982898e-01 -1.000169e+00| 0:0:01| spchol  1  2
19|0.980|0.983|3.8e-12|7.6e-08|7.1e-05|-9.999324e-01 -1.000003e+00| 0:0:01| spchol  2  2
20|0.989|0.989|8.8e-11|2.9e-09|1.2e-06|-9.999988e-01 -1.000000e+00| 0:0:01| spchol 23 12
  stop: primal infeas has deteriorated too much, 3.9e-01
21|0.568|0.944|8.8e-11|2.9e-09|1.2e-06|-9.999988e-01 -1.000000e+00| 0:0:01|
-------------------------------------------------------------------
 number of iterations   = 21
 primal objective value = -9.99998836e-01
 dual   objective value = -1.00000003e+00
 gap := trace(XZ)       = 1.20e-06
 relative gap           = 4.00e-07
 actual relative gap    = 3.99e-07
 rel. primal infeas (scaled problem)   = 8.77e-11
 rel. dual     "        "       "      = 2.86e-09
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "        "       "      = 0.00e+00
 norm(X), norm(y), norm(Z) = 2.0e+00, 1.9e+01, 3.4e+01
 norm(A), norm(b), norm(C) = 2.1e+02, 2.0e+00, 2.4e+00
 Total CPU time (secs)  = 1.37
 CPU time per iteration = 0.07
 termination code       = -7
 DIMACS: 8.8e-11  0.0e+00  3.5e-09  0.0e+00  4.0e-07  4.0e-07
-------------------------------------------------------------------

------------------------------------------------------------
Status: Inaccurate/Solved
Optimal value (cvx_optval): +1
```

```
Cost = cvx_optval;
W_answers = W;
```

See the CVX Results:

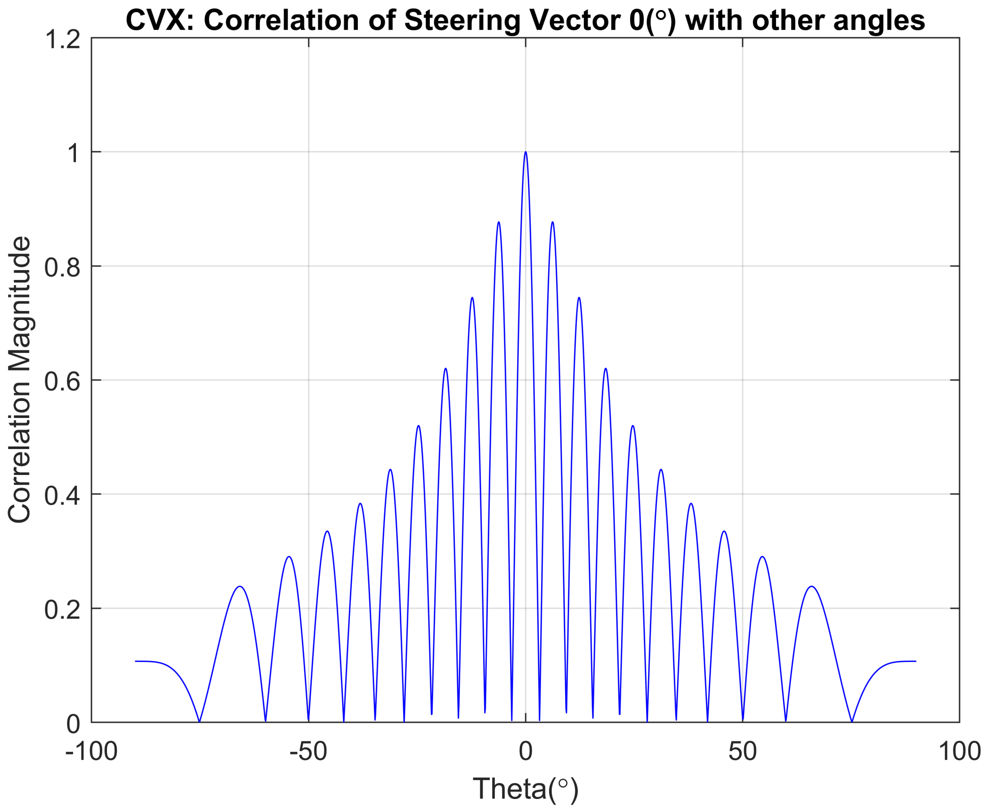
```
figure(2)

plot((1:length(W_answers)),W_answers);
grid on
title(" Weights Corresponding to Steering Vectors")
xlabel("Index")
ylabel("W_{Index} Magnitude")
```




```
```

See the Corresponding Pattern:

```
MAP = exp(1j*K*D*sind(theta)) ;  % Matrix of Steering Vectors

zav=0;

plot(theta,abs(W'*A),'b');
grid on
title("CVX: Correlation of Steering Vector 0(\circ) with other angles")
xlabel("Theta(\circ)")
ylabel("Correlation Magnitude")
```



New CVX Problem: <Over place of elements>

First distribute many points over y-axis and then choose those giving us the best pattern! --- >> Best Pattern is a pattern in which we see a delta-shaped beam.

```
D_init = linspace(d_min, d_max,1e2 )';
P = length(D_init); % a random large number -->> Number of elements in a dense array!


Max_poss_deg = asind(Lambda/d_max) ;
Theta_important = theta((theta > Max_poss_deg) | (theta < -Max_poss_deg) ); % Choose those Thet
A_new = exp(1j*K*D_init*sind(Theta_important)); % New MAP Matrix




 cvx_begin
```

```
Warning: A non-empty cvx problem already exists in this scope.
   It is being overwritten.
```

```
        variable W(P);
        variable t;
```

```
        minimize( t ) % Minimize a Convex Function

        subject to % Our Constraints:
            ones(1,P)*W           == 1;
            abs(W'*A_new)         <= t;
            W                     >= 0;
            W                     <= 1;


  cvx_end
```

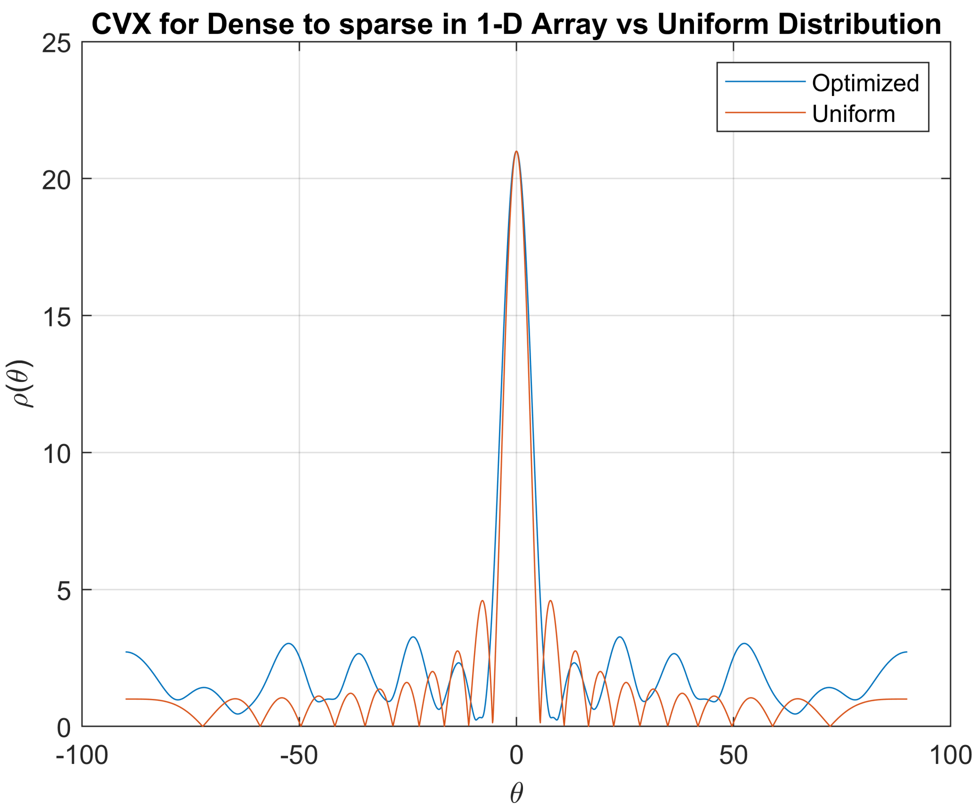
```
Calling SDPT3 4.0: 6945 variables, 1787 equality constraints
   For improved efficiency, SDPT3 is solving the dual problem.
------------------------------------------------------------

 num. of constraints = 1787
 dim. of socp   var  = 5058,   num. of socp blk  = 1686
 dim. of linear var  = 1886
 dim. of free   var  =  1 *** convert ublk to lblk
*******************************************************************
   SDPT3: Infeasible path-following algorithms
*******************************************************************
 version  predcorr  gam  expon  scale_data
    NT      1      0.000   1        0
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj    cputime
-------------------------------------------------------------------
 0|0.000|0.000|2.3e+03|5.9e+01|7.2e+06| 8.685620e+03  0.000000e+00| 0:0:00| spchol  1  1
 1|0.976|0.810|5.4e+01|1.1e+01|3.0e+05| 8.229289e+03 -6.923182e+01| 0:0:01| spchol  1  1
 2|0.795|0.290|1.1e+01|8.1e+00|1.3e+05| 6.931789e+03 -1.773792e+02| 0:0:01| spchol  1  1
 3|0.298|0.288|7.8e+00|5.8e+00|1.0e+05| 6.416177e+03 -4.250503e+02| 0:0:01| spchol  1  1
 4|0.602|0.850|3.1e+00|8.8e-01|3.0e+04| 5.127671e+03 -8.373416e+02| 0:0:01| spchol  1  1
 5|0.754|0.435|7.7e-01|5.0e-01|1.1e+04| 2.017221e+03 -1.083761e+03| 0:0:01| spchol  1  1
 6|0.993|0.979|5.1e-03|1.1e-02|7.1e+02| 2.340290e+02 -3.793389e+02| 0:0:01| spchol  1  1
 7|0.985|0.937|6.2e-05|1.9e-03|5.1e+01| 1.963464e+01 -2.869824e+01| 0:0:02| spchol  1  1
 8|0.966|0.492|2.0e-06|1.0e-03|2.0e+01| 1.386462e+00 -1.898146e+01| 0:0:02| spchol  1  1
 9|1.000|0.466|1.4e-07|5.4e-04|1.2e+01| 2.470327e-01 -1.217060e+01| 0:0:02| spchol  1  1
10|0.843|0.962|6.1e-08|2.0e-05|5.3e-01| 3.648872e-02 -4.912558e-01| 0:0:02| spchol  1  1
11|0.917|0.250|5.8e-09|1.5e-05|4.1e-01| 5.094894e-03 -4.007611e-01| 0:0:02| spchol  1  1
12|1.000|0.740|1.8e-10|4.1e-06|1.7e-01|-4.114644e-03 -1.777443e-01| 0:0:03| spchol  1  1
13|0.516|0.514|9.9e-11|2.0e-06|1.2e-01|-1.916598e-02 -1.392856e-01| 0:0:03| spchol  1  1
14|0.724|0.825|3.2e-11|4.4e-07|5.7e-02|-4.468000e-02 -1.012560e-01| 0:0:03| spchol  1  1
15|0.472|0.927|2.4e-11|1.4e-07|3.0e-02|-5.715033e-02 -8.759641e-02| 0:0:03| spchol  1  2
16|0.805|0.710|6.0e-12|7.2e-08|1.4e-02|-7.048592e-02 -8.417225e-02| 0:0:04| spchol  2  2
17|0.175|0.729|5.3e-12|3.2e-08|1.1e-02|-7.155487e-02 -8.206272e-02| 0:0:04| spchol  2  2
18|0.888|0.570|1.3e-11|2.5e-08|4.9e-03|-7.626086e-02 -8.113118e-02| 0:0:04| spchol  2  2
19|0.857|0.679|5.3e-11|1.1e-08|2.2e-03|-7.817156e-02 -8.034945e-02| 0:0:04| spchol  2  2
20|0.975|0.700|5.0e-11|5.1e-09|8.0e-04|-7.920112e-02 -8.000095e-02| 0:0:04| spchol  2  2
21|0.984|0.823|7.3e-11|1.9e-09|1.8e-04|-7.966490e-02 -7.984308e-02| 0:0:05| spchol  2  2
22|0.632|0.848|3.5e-11|4.2e-10|7.9e-05|-7.974617e-02 -7.982497e-02| 0:0:05| spchol  3  3
23|0.824|0.932|4.2e-10|1.9e-10|3.0e-05|-7.979035e-02 -7.982039e-02| 0:0:05| spchol  3  3
24|0.939|0.924|6.1e-10|7.2e-11|5.8e-06|-7.981339e-02 -7.981915e-02| 0:0:05| spchol  3  3
25|0.923|0.953|4.8e-09|2.2e-11|1.3e-06|-7.981762e-02 -7.981896e-02| 0:0:06| spchol  4  4
26|0.632|0.944|2.1e-09|2.5e-11|7.0e-07|-7.981823e-02 -7.981893e-02| 0:0:06| spchol  4  4
27|0.637|0.943|1.2e-09|3.7e-11|3.7e-07|-7.981856e-02 -7.981892e-02| 0:0:06| spchol  5  4
28|0.634|0.943|1.0e-09|5.6e-11|2.0e-07|-7.981873e-02 -7.981892e-02| 0:0:06| spchol  5  5
29|0.633|0.943|8.2e-10|8.4e-11|1.0e-07|-7.981882e-02 -7.981892e-02| 0:0:06| spchol  5  5
30|0.633|0.943|5.1e-10|1.3e-10|5.6e-08|-7.981886e-02 -7.981892e-02| 0:0:07| spchol  4  4
31|0.634|0.943|2.9e-10|1.1e-10|3.0e-08|-7.981889e-02 -7.981892e-02| 0:0:07| spchol  5  4
32|0.635|0.943|1.8e-10|6.4e-11|1.6e-08|-7.981890e-02 -7.981892e-02| 0:0:07|
```

```
  stop: max(relative gap, infeasibilities) < 1.49e-08
-------------------------------------------------------------------
 number of iterations   = 32
 primal objective value = -7.98189029e-02
 dual   objective value = -7.98189185e-02
 gap := trace(XZ)       = 1.57e-08
 relative gap           = 1.35e-08
 actual relative gap    = 1.34e-08
 rel. primal infeas (scaled problem)   = 1.84e-10
 rel. dual     "        "       "      = 6.37e-11
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "        "       "      = 0.00e+00
 norm(X), norm(y), norm(Z) = 6.1e-01, 1.3e+00, 1.1e+01
 norm(A), norm(b), norm(C) = 4.2e+02, 2.0e+00, 4.3e+01
 Total CPU time (secs)  = 7.01
 CPU time per iteration = 0.22
 termination code       =  0
 DIMACS: 1.8e-10  0.0e+00  1.4e-09  0.0e+00  1.3e-08  1.4e-08
-------------------------------------------------------------------

------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +0.0798189
```

Choose (k) Best Weights:

```
% Find k largest elements of array:
% [B,I] = maxk(___) finds the indices of the largest k values of A and returns them in I.
k = M;
[W_answers_dense , W_Indexes ] = maxk( W , k )  ;


D_dense_Chosen = D_init(W_Indexes);

Theta_test = 0;


Map_new = exp(1j*K*D_dense_Chosen*sind(theta));
% Map_new = Map_new./repmat(sqrt(sum(abs(Map_new).^2)),K,1);


a_new = (exp(1j*K*D_dense_Chosen*sind(Theta_test)));

Corr_Final = abs(a_new'*Map_new); % Correlation of New MAP with the Chosen Positions

figure(3)

plot(theta, Corr_Final);

grid on
xlabel('\theta')
ylabel('\rho(\theta)')
title("CVX for Dense to sparse in 1-D Array vs Uniform Distribution")
```

```
hold on

plot(theta, g*max(Corr_Final));

legend("Optimized","Uniform")
```




```

```

Try with more number of elements:

```
D_init = linspace(d_min, d_max,1e3 )';
P = length(D_init); % a random large number -->> Number of elements in a dense array!


Max_poss_deg = asind(Lambda/d_max) ;
Theta_important = theta((theta > Max_poss_deg) | (theta < -Max_poss_deg) ); % Choose those Thet
A_new = exp(1j*K*D_init*sind(Theta_important)); % New MAP Matrix




 cvx_begin
```

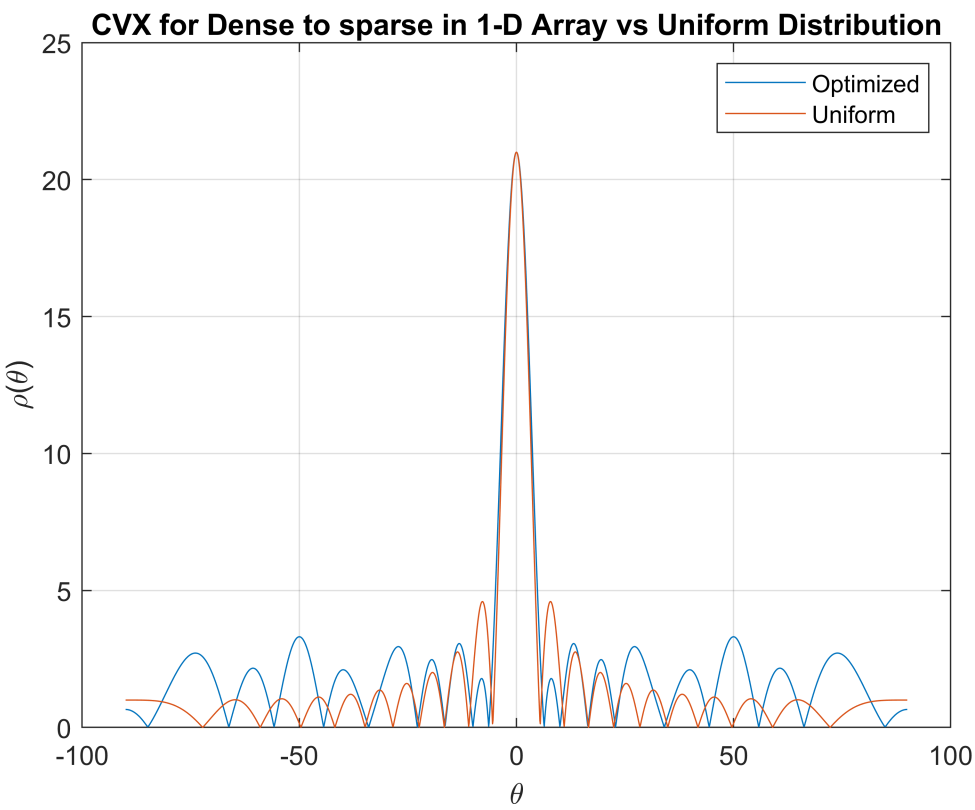
```
        variable W(P);
        variable t;


        minimize( t ) % Minimize a Convex Function

        subject to % Our Constraints:
            ones(1,P)*W          == 1;
            abs(W'*A_new)        <= t;
            W                    >= 0;
            W                    <= 1;


  cvx_end
```

```
Calling SDPT3 4.0: 8745 variables, 2687 equality constraints
   For improved efficiency, SDPT3 is solving the dual problem.
------------------------------------------------------------

 num. of constraints = 2687
 dim. of socp   var  = 5058,   num. of socp blk  = 1686
 dim. of linear var  = 3686
 dim. of free   var  =  1 *** convert ublk to lblk
 number of dense column in A = 2
*******************************************************************
   SDPT3: Infeasible path-following algorithms
*******************************************************************
 version  predcorr  gam  expon  scale_data
    NT      1      0.000   1        0
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj    cputime
-------------------------------------------------------------------
 0|0.000|0.000|2.9e+03|7.7e+01|2.7e+07| 1.214249e+05  0.000000e+00| 0:0:03| spchol  1  1
 1|0.978|0.820|6.4e+01|1.4e+01|3.0e+06| 1.163489e+05 -8.880441e+01| 0:0:06| chol  1  1
 2|0.124|0.038|5.6e+01|1.4e+01|2.9e+06| 1.157226e+05 -2.908504e+02| 0:0:11| chol  1  1
 3|0.131|0.098|4.9e+01|1.2e+01|2.7e+06| 1.143483e+05 -6.269007e+02| 0:0:15| chol  1  1
 4|0.191|0.211|3.9e+01|9.8e+00|2.2e+06| 1.127354e+05 -1.609169e+03| 0:0:20| chol  1  1
 5|0.496|0.103|2.0e+01|8.8e+00|1.9e+06| 1.048144e+05 -1.933406e+03| 0:0:24| chol  1  1
 6|0.574|0.763|8.5e+00|2.1e+00|5.7e+05| 9.298859e+04 -4.090741e+03| 0:0:29| chol  1  1
 7|0.624|0.158|3.2e+00|1.8e+00|4.0e+05| 6.451711e+04 -4.840209e+03| 0:0:33| chol  1  1
 8|0.971|0.952|9.1e-02|8.7e-02|4.3e+04| 2.497136e+04 -5.036101e+03| 0:0:38| chol  1  1
 9|0.983|0.994|1.6e-03|2.3e-03|1.2e+03| 6.973303e+02 -2.450478e+02| 0:0:43| chol  1  1
10|0.979|0.785|3.4e-05|9.8e-04|1.7e+02| 6.654421e+01 -6.533520e+01| 0:0:49| chol  1  1
11|1.000|0.870|1.9e-08|1.9e-04|5.6e+01| 2.094402e+01 -3.314889e+01| 0:0:54| chol  1  1
12|0.985|0.910|3.4e-09|2.3e-05|3.8e+00| 6.226269e-01 -3.209182e+00| 0:1:00| chol  1  1
13|0.953|0.874|9.4e-08|3.4e-06|4.5e-01| 3.461339e-02 -4.187012e-01| 0:1:06| chol  1  1
14|0.578|0.248|4.8e-07|2.6e-06|3.8e-01| 4.492601e-02 -3.399039e-01| 0:1:12| chol  1  1
15|0.913|0.608|3.8e-08|1.2e-06|2.2e-01| 3.549689e-02 -1.816813e-01| 0:1:18| chol  1  1
16|0.329|0.225|2.6e-08|8.6e-07|1.9e-01| 2.199607e-02 -1.670477e-01| 0:1:24| chol  1  1
17|0.739|0.829|6.8e-09|3.2e-07|9.5e-02|-1.011557e-02 -1.048926e-01| 0:1:30| chol  1  1
18|0.513|0.922|3.3e-09|1.5e-07|5.6e-02|-3.759459e-02 -9.317669e-02| 0:1:36| chol  1  1
19|0.799|0.924|4.8e-10|8.5e-08|2.0e-02|-6.403529e-02 -8.378114e-02| 0:1:42| chol  1  1
20|0.886|0.877|9.1e-11|3.0e-08|7.3e-03|-7.394467e-02 -8.123348e-02| 0:1:48| chol  1  1
21|0.874|0.714|6.5e-09|1.1e-08|3.1e-03|-7.742272e-02 -8.051786e-02| 0:1:55| chol  1  1
22|0.938|0.706|1.1e-07|4.7e-09|1.2e-03|-7.892981e-02 -8.013572e-02| 0:2:01| chol  1  1
23|0.990|0.773|2.8e-07|1.8e-09|4.6e-04|-7.947119e-02 -7.992747e-02| 0:2:07| chol  2  2
24|0.948|0.874|1.5e-08|7.0e-10|1.5e-04|-7.968546e-02 -7.983643e-02| 0:2:13| chol  2  2
25|0.601|0.576|5.8e-09|2.6e-10|9.1e-05|-7.974281e-02 -7.983347e-02| 0:2:19| chol  2  2
26|0.699|0.937|4.0e-09|2.0e-10|4.1e-05|-7.977824e-02 -7.981932e-02| 0:2:26| chol  3  3
```

```
27|0.924|0.921|1.8e-08|2.3e-10|1.0e-05|-7.980603e-02 -7.981628e-02| 0:2:32| chol  4  4
28|0.760|0.878|7.6e-08|3.4e-10|4.9e-06|-7.981062e-02 -7.981551e-02| 0:2:38| chol  5  5
29|0.898|0.955|1.2e-07|4.8e-10|1.5e-06|-7.981359e-02 -7.981508e-02| 0:2:44| chol
  linsysolve: Schur complement matrix not positive definite
  switch to LU factor. lu 22   4
30|0.658|0.648|2.6e-07|8.7e-10|8.0e-07|-7.981418e-02 -7.981498e-02| 0:2:51| lu 20   4
31|0.689|0.942|1.5e-07|1.1e-09|4.1e-07|-7.981450e-02 -7.981490e-02| 0:2:58| lu 18   4
32|0.639|0.921|7.5e-08|1.7e-09|2.2e-07|-7.981467e-02 -7.981487e-02| 0:3:04| lu 12   3
33|0.640|0.943|7.1e-08|2.5e-09|1.2e-07|-7.981475e-02 -7.981486e-02| 0:3:10| lu 29  18
34|0.687|0.560|3.6e-08|2.8e-09|8.3e-08|-7.981479e-02 -7.981486e-02| 0:3:18| lu 16  14
35|0.736|0.863|6.2e-08|1.3e-09|4.4e-08|-7.981482e-02 -7.981486e-02| 0:3:25| lu 30   5
36|0.495|0.528|5.9e-08|1.3e-09|3.3e-08|-7.981483e-02 -7.981485e-02| 0:3:33| lu 30  10
37|0.024|0.032|7.0e-08|2.0e-09|3.5e-08|-7.981483e-02 -7.981485e-02| 0:3:40|
  stop: progress is bad
-------------------------------------------------------------------
 number of iterations   = 37
 primal objective value = -7.98148291e-02
 dual   objective value = -7.98148536e-02
 gap := trace(XZ)       = 3.27e-08
 relative gap           = 2.82e-08
 actual relative gap    = 2.11e-08
 rel. primal infeas (scaled problem)   = 5.87e-08
 rel. dual     "        "       "      = 1.30e-09
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "        "       "      = 0.00e+00
 norm(X), norm(y), norm(Z) = 6.9e-01, 1.3e+00, 3.2e+01
 norm(A), norm(b), norm(C) = 1.3e+03, 2.0e+00, 5.3e+01
 Total CPU time (secs)  = 220.19
 CPU time per iteration = 5.95
 termination code       = -5
 DIMACS: 5.9e-08  0.0e+00  3.4e-08  0.0e+00  2.1e-08  2.8e-08
-------------------------------------------------------------------

------------------------------------------------------------
Status: Inaccurate/Solved
Optimal value (cvx_optval): +0.0798149
```

```
% Find k largest elements of array:
% [B,I] = maxk(___) finds the indices of the largest k values of A and returns them in I.
k = M;
[W_answers_dense , W_Indexes ] = maxk( W , k )  ;


D_dense_Chosen = D_init(W_Indexes);

Theta_test = 0;



Map_new = exp(1j*K*D_dense_Chosen*sind(theta));
% Map_new = Map_new./repmat(sqrt(sum(abs(Map_new).^2)),K,1);


a_new = (exp(1j*K*D_dense_Chosen*sind(Theta_test)));

Corr_Final = abs(a_new'*Map_new); % Correlation of New MAP with the Chosen Positions
```

```
close all
figure(4)

plot(theta, Corr_Final);

grid on
xlabel('\theta')
ylabel('\rho(\theta)')
title("CVX for Dense to sparse in 1-D Array vs Uniform Distribution")

hold on

plot(theta, g*max(Corr_Final));

legend("Optimized","Uniform")
```



Antenna Positions:

```
figure(6)
subplot(1,2,1)
stem(sort(D_init(W_Indexes)))
grid on
xlabel("Antenna Index")
ylabel("Antenna Position")
title("Antenna Position over Z-Axis")
```

```
subplot(1,2,2)
plot(zeros(1,length(W_Indexes)) , D_init(W_Indexes)  , 'r*')
grid on
title("Antenna Position on the Tower")
xlabel("Tower Base")
ylabel("Antenna Height")
```