SSV COMPUTATION [FOR A FIXED FREQ]

$$N(s) = \begin{bmatrix} \dfrac{1}{s+1} & \dfrac{1}{s+2} \\ \dfrac{1}{s+3} & \dfrac{1}{s+5} \end{bmatrix} \rightarrow N(j\omega) = \begin{bmatrix} \dfrac{1}{j\omega+1} & \dfrac{1}{j\omega+2} \\ \dfrac{1}{j\omega+3} & \dfrac{1}{j\omega+5} \end{bmatrix} \rightarrow N(j\omega_x) = \begin{bmatrix} \dfrac{1}{j\omega_x+1} & \dfrac{1}{j\omega_x+2} \\ \dfrac{1}{j\omega_x+3} & \dfrac{1}{j\omega_x+5} \end{bmatrix}$$

$$N(j1) = \begin{bmatrix} \dfrac{1}{j1+1} & \dfrac{1}{j1+2} \\ \dfrac{1}{j1+3} & \dfrac{1}{j1+5} \end{bmatrix}$$

<table>
<tr><th>MATLAB code</th><th>Code OUTPUT</th></tr>
<tr><td>

```
clear all;close all;clc;
%%
N_s=[tf([1],[1,1]),tf([1],[1,2]);
     tf([1],[1,3]),tf([1],[1,5])];
%% compute the matrix for a given freq
Njw1=evalfr(N_s,1*i)
```

</td><td>

```
>> Njw1=evalfr(N_s,1*i)

Njw1 =

   0.5000 - 0.5000i   0.4000 - 0.2000i
   0.3000 - 0.1000i   0.1923 - 0.0385i
```

</td></tr>
</table>

CASE 1: FULL DELTA [UNSTRUCTURED UNCERTAINTY]

$$\Delta = \{\Delta \in \mathbb{C}^{n\times n}\}$$

$$\mathcal{D} = \{D: D\Delta = \Delta D, \forall \Delta \in \mathbf{\Delta}\}$$

$$\mathcal{D} = \{dI_n: d \in \mathbb{R}, d > 0\}$$

$$\mu_{\mathbf{\Delta}}(M) = \mu_{\mathbf{\Delta}}(DMD^{-1})$$

$$\mu_{\mathbf{\Delta}}(M) \leq \inf_{D\in\mathcal{D}} \bar{\sigma}(DMD^{-1})$$

Since, there is a suitable parameterization for $\boxed{D \in \mathcal{D}}$, the problem can be expressed more specifically as,

$$\inf_{D\in\mathcal{D}} \bar{\sigma}(DMD^{-1}) = \inf_{D\in\mathcal{D}} \bar{\sigma}(dI_2Md^{-1}I_2)$$

Matlab code

```
%% CASE 1: Full Delta Set ---> single d --> d*eye(n), d in REAL

% COST FUNCTION
% For full delta   --> d*eye(n)
% For delta*eye(n) --> FULL D
% For diag[delta1*eye(s1),delta2*eye(s2),delta3*eye(s3)] --> diag[D1,D2,D3]
obj_fun = @(d,Njwx) max(svd([d*eye(2)]*Njwx*inv([d*eye(2)])));                                % d*I, d in REAL

% test the cost function
% obj_fun(d,Njwx)
obj_fun(2,Njw1)
%
% MINIMIZE THE COST WRT "D"
% x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
d_vec = fmincon(@(d)obj_fun(d,Njw1),[1],[],[],[],[],1e-5*ones(1,1),10*ones(1,1),[],[]);
d_mat=d_vec*eye(2);
% COMPUTE THE COST FOR THE MINIMIZER-D , "THAT IS THE ANSWER"
FMINCON_SSV=max(svd(d_mat*Njw1*inv(d_mat)));

rho_N=max(abs(eig(Njw1)));       % spectral radius of Njw1
sigma_max_N=max(svd(Njw1));                 % simga-max of Njw1

% COMPUTE THE SSV USING MATLAB-MUSSV FCN
% bounds = mussv(N,BlockStructure)
[bounds,muinfo] = mussv(Njw1,[2,2]);
MATLAB_MUSSV=bounds(2);
clc;
fprintf("FMINCON_SSV \t rho_N \t sigma_max_N \t MATLAB_MUSSV\n");
fprintf("%6.4f       \t %6.4f \t %6.4f       \t %6.4f\n",FMINCON_SSV,rho_N,sigma_max_N,MATLAB_MUSSV);
```

Code OUTPUT

FMINCON_SSV	rho_N	sigma_max_N	MATLAB_MUSSV
0.9152	0.8680	0.9152	0.9152

CASE 2: repeating-scalar-delta [STRUCTURED UNCERTAINTY]

$$\Delta = \{\delta I_n : \delta \in \mathbb{C}\}$$

$$\mathcal{D} = \{D : D\Delta = \Delta D, \forall \Delta \in \boldsymbol{\Delta}\}$$

$$\mathcal{D} = \{D : D \in \mathbb{C}^{n\times n}, D = D^H > 0\}$$

$$\mu_{\boldsymbol{\Delta}}(M) = \mu_{\boldsymbol{\Delta}}(DMD^{-1})$$

$$\mu_{\boldsymbol{\Delta}}(M) \leq \inf_{D\in\mathcal{D}} \bar{\sigma}(DMD^{-1})$$

Since, there is a suitable parameterization for $\boxed{D \in \mathcal{D}}$, the problem can be expressed more specifically as,

$$\mathcal{D} = \left\{ D = \begin{bmatrix} d_1 & d_3 + jd_4 \\ d_3 - jd_4 & d_2 \end{bmatrix} : \boxed{d_1, \dots, d_4 \in \mathbb{R}}, \boxed{d_1, \dots, d_4 > 0} \right\}$$

$$\inf_{D\in\mathcal{D}} \bar{\sigma}(DMD^{-1}) = \inf_{d_1,\dots,d_4>0} \bar{\sigma}\left(\begin{bmatrix} d_1 & d_3 + jd_4 \\ d_3 - jd_4 & d_2 \end{bmatrix} M \begin{bmatrix} d_1 & d_3 + jd_4 \\ d_3 - jd_4 & d_2 \end{bmatrix}^{-1} \right)$$

Matlab code

```
%% CASE 2: Structured Delta Set [delta*eye(n)] ---> FULL D,hermitian(like symmetric),complex
% for 2x2 D matrix, it can be parameterized as:
% D=[d(1)+i*d(2),d(3)+i*d(4);
%    d(5)+i*d(6),d(7)+i*d(8)]
% so, 2x2=4 entries and since they are complex, 4x2=8 terms are needed
% but since D=D*>0, hermitian-pos-def,
% it can be parameterized as:
% D=[d(1),d(3)+i*d(4);d(3)-i*d(4),d(2)]

% COST FUNCTION
% obj_fun = @(d,Njwx) max(svd([D]*Njwx*inv([D])));        % FULL D
obj_fun = @(d,Njwx) max(svd([d(1),d(3)+i*d(4);d(3)-i*d(4),d(2)]*Njwx*inv([d(1),d(3)+i*d(4);d(3)-i*d(4),d(2)])));
% HERMITIAN D


% test the cost function
% obj_fun(d,Njwx)
obj_fun(randn(4).^2,Njw1)
%
% MINIMIZE THE COST WRT "D"
% x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
d_vec = fmincon(@(d)obj_fun(d,Njw1),[randn(4).^2],[],[],[],[],1e-5*ones(4,1),10*ones(4,1),[],[]);
d_mat=[d_vec(1),d_vec(3)+i*d_vec(4);d_vec(3)-i*d_vec(4),d_vec(2)];
% COMPUTE THE COST FOR THE MINIMIZER-D , "THAT IS THE ANSWER"
FMINCON_SSV=max(svd(d_mat*Njw1*inv(d_mat)));
%
rho_N=max(abs(eig(Njw1)));                    % spectral radius of Njw1
sigma_max_N=max(svd(Njw1));                   % simga-max of Njw1

% COMPUTE THE SSV USING MATLAB-MUSSV FCN
% bounds = mussv(N,BlockStructure)
[bounds,muinfo] = mussv(Njw1,[2,0])
MATLAB_MUSSV=bounds(2);
clc;
fprintf("FMINCON_SSV \t rho_N \t sigma_max_N \t MATLAB_MUSSV\n");
fprintf("%6.4f       \t %6.4f \t %6.4f       \t %6.4f\n",FMINCON_SSV,rho_N,sigma_max_N,MATLAB_MUSSV);
```

Code OUTPUT

```
FMINCON_SSV        rho_N    sigma_max_N      MATLAB_MUSSV
0.8680             0.8680       0.9152           0.8680
```

CASE 3: diagonal-delta [STRUCTURED UNCERTAINTY]

$$\boldsymbol{\Delta} = \{\mathrm{diag}[\delta_1, \delta_2]: \delta_1, \delta_2 \in \mathbb{C}\}$$

$$\mathcal{D} = \{D: D\Delta = \Delta D, \forall \Delta \in \boldsymbol{\Delta}\}$$

$$\mathcal{D} = \left\{\mathrm{diag}[\mathrm{d}_1, \mathrm{d}_2]: \boxed{\mathrm{d}_1, \mathrm{d}_2 \in \mathbb{R}}, \boxed{\mathrm{d}_1, \mathrm{d}_2 > 0}\right\}$$

$$\mu_{\boldsymbol{\Delta}}(M) = \mu_{\boldsymbol{\Delta}}(DMD^{-1})$$

$$\mu_{\boldsymbol{\Delta}}(M) \leq \inf_{D \in \mathcal{D}} \bar{\sigma}(DMD^{-1})$$

Since, there is a suitable parameterization for $\boxed{D \in \mathcal{D}}$, the problem can be expressed more specifically as,

$$\mathcal{D} = \left\{D = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}: \boxed{d_1, d_2 \in \mathbb{R}}, \boxed{d_1, d_2 > 0}\right\}$$

$$\inf_{D \in \mathcal{D}} \bar{\sigma}(DMD^{-1}) = \inf_{d_1, d_2 > 0} \bar{\sigma}\left(\begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix} M \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}^{-1}\right)$$

Matlab code

```
%% CASE 3: Structured Delta Set diag[delta1,delta2] ---> diag[d1,d2], d1,d2 REAL
% for 2x2 D matrix, it can be parameterized as:
% D=diag[d(1),d(2)]
% so, 2 terms are needed

% COST FUNCTION
obj_fun = @(d,Njwx) max(svd(diag([d(1),d(2)])*Njwx*inv(diag([d(1),d(2)]))));                % diag D

% test the cost function
% obj_fun(d,Njwx)
obj_fun(randn(2).^2,Njw1)
%
% MINIMIZE THE COST WRT "D"
% x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
d_vec = fmincon(@(d)obj_fun(d,Njw1),[randn(2).^2],[],[],[],[],1e-5*ones(2,1),10*ones(2,1),[],[]);
d_mat=diag([d_vec(1),d_vec(2)])
% COMPUTE THE COST FOR THE MINIMIZER-D , "THAT IS THE ANSWER"
FMINCON_SSV=max(svd(d_mat*Njw1*inv(d_mat)))
%
rho_N=max(abs(eig(Njw1)));                    % spectral radius of Njw1
sigma_max_N=max(svd(Njw1));                   % simga-max of Njw1

% COMPUTE THE SSV USING MATLAB-MUSSV FCN
% bounds = mussv(N,BlockStructure)
[bounds,muinfo] = mussv(Njw1,[1,1;1,1])
MATLAB_MUSSV=bounds(2);
clc;
fprintf("FMINCON_SSV \t rho_N \t sigma_max_N \t MATLAB_MUSSV\n");
fprintf("%6.4f       \t %6.4f \t %6.4f       \t %6.4f\n",FMINCON_SSV,rho_N,sigma_max_N,MATLAB_MUSSV);
```

Code OUTPUT

```
FMINCON_SSV         rho_N    sigma_max_N       MATLAB_MUSSV
0.9057              0.8680       0.9152             0.9057
```

SSV COMPUTATION [FOR A FIXED FREQ]

$$N(s)=\begin{bmatrix}\frac{1}{s+1} & \frac{1}{s+2} & \frac{1}{s+3} & \frac{1}{s+3} & \frac{1}{s+4}\\ \frac{1}{s+5} & \frac{1}{s+1} & \frac{1}{s+2} & \frac{1}{s+3} & \frac{1}{s+4}\\ \frac{1}{s+2} & \frac{1}{s+4} & \frac{1}{s+7} & \frac{1}{s+3} & \frac{1}{s+4}\\ \frac{1}{s+2} & \frac{1}{s+4} & \frac{1}{s+7} & \frac{1}{s+3} & \frac{1}{s+4}\\ \frac{1}{s+2} & \frac{1}{s+4} & \frac{1}{s+7} & \frac{1}{s+3} & \frac{1}{s+4}\end{bmatrix}\to N(j\omega)=\begin{bmatrix}\frac{1}{j\omega+1} & \frac{1}{j\omega+2} & \frac{1}{j\omega+3} & \frac{1}{j\omega+3} & \frac{1}{j\omega+4}\\ \frac{1}{j\omega+5} & \frac{1}{j\omega+1} & \frac{1}{j\omega+2} & \frac{1}{j\omega+3} & \frac{1}{j\omega+4}\\ \frac{1}{j\omega+2} & \frac{1}{j\omega+4} & \frac{1}{j\omega+7} & \frac{1}{j\omega+3} & \frac{1}{j\omega+4}\\ \frac{1}{j\omega+2} & \frac{1}{j\omega+4} & \frac{1}{j\omega+7} & \frac{1}{j\omega+3} & \frac{1}{j\omega+4}\\ \frac{1}{j\omega+2} & \frac{1}{j\omega+4} & \frac{1}{j\omega+7} & \frac{1}{j\omega+3} & \frac{1}{j\omega+4}\end{bmatrix}$$

$$N(j1)=\begin{bmatrix}\frac{1}{j1+1} & \frac{1}{j1+2} & \frac{1}{j1+3} & \frac{1}{j1+3} & \frac{1}{j1+4}\\ \frac{1}{j1+5} & \frac{1}{j1+1} & \frac{1}{j1+2} & \frac{1}{j1+3} & \frac{1}{j1+4}\\ \frac{1}{j1+2} & \frac{1}{j1+4} & \frac{1}{j1+7} & \frac{1}{j1+3} & \frac{1}{j1+4}\\ \frac{1}{j1+2} & \frac{1}{j1+4} & \frac{1}{j1+7} & \frac{1}{j1+3} & \frac{1}{j1+4}\\ \frac{1}{j1+2} & \frac{1}{j1+4} & \frac{1}{j1+7} & \frac{1}{j1+3} & \frac{1}{j1+4}\end{bmatrix}$$

CASE 4: MIXED CASE [SQUARE CASE]

$$\mathbf{\Delta} = \left\{ \begin{bmatrix} \delta_1 I_2 & & \\ & \delta_2 & \\ & & \Delta_3 \end{bmatrix} : \delta_1, \delta_2 \in \mathbb{C}, \Delta_3 \in \mathbb{C}^{2\times 2} \right\}, \mathcal{D} = \{D : D\Delta = \Delta D, \forall \Delta \in \mathbf{\Delta}\}$$

$$\mathcal{D} = \left\{ \begin{bmatrix} D_1 & & \\ & D_2 & \\ & & D_3 I_2 \end{bmatrix} : \boxed{D_1 \in \mathbb{C}^{2\times 2}, D_1 = D_1^H > 0}, \boxed{D_2 \in \mathbb{R}, D_2 > 0}, \boxed{D_3 \in \mathbb{R}, D_3 > 0} \right\}$$

$$\mu_{\mathbf{\Delta}}(M) = \mu_{\mathbf{\Delta}}(DMD^{-1}) \rightarrow \mu_{\mathbf{\Delta}}(M) \leq \inf_{D \in \mathcal{D}} \bar{\sigma}(DMD^{-1})$$

Since, there is a suitable parameterization for $\boxed{D \in \mathcal{D}}$, the problem can be expressed more specifically as,

$$\mathcal{D} = \left\{ \begin{bmatrix} \begin{bmatrix} d_1 & d_3 + jd_4 \\ d_3 + jd_4 & d_2 \end{bmatrix} & & \\ & d_5 & \\ & & d_6 I_2 \end{bmatrix} : \boxed{d_1, \ldots, d_6 \in \mathbb{R}}, \boxed{d_1, \ldots, d_6 > 0} \right\}$$

$$\inf_{D \in \mathcal{D}} \bar{\sigma}(DMD^{-1}) = \inf_{d_1, \ldots, d_6 > 0} \bar{\sigma}(DMD^{-1})$$

MATLAB code

```
clear all;close all;clc;
%%
N_s=[tf([1],[1,1]),tf([1],[1,2]),tf([1],[1,3]),tf([1],[1,3]),tf([1],[1,4]);
     tf([1],[1,5]),tf([1],[1,1]),tf([1],[1,2]),tf([1],[1,3]),tf([1],[1,4]);
     tf([1],[1,2]),tf([1],[1,4]),tf([1],[1,7]),tf([1],[1,3]),tf([1],[1,4]);
     tf([1],[1,2]),tf([1],[1,4]),tf([1],[1,7]),tf([1],[1,3]),tf([1],[1,4]);
     tf([1],[1,2]),tf([1],[1,4]),tf([1],[1,7]),tf([1],[1,3]),tf([1],[1,4])];
%% compute the matrix for a given freq
Njw1=evalfr(N_s,1*i)
%% CASE 4:  DELTA STRUCTURE = blkdiag([delta1*eye(2),delta2,DELTA_3]):delta1,delta2 in COMPLEX, DELTA_3 in 2x2 COMPLEX MATRIX

% COST FUNCTION
% D=blkdiag([d(1),d(3)+i*d(4);d(3)-i*d(4),d(2)],[d(5)],[d(6)*eye(2)])
% obj_fun = @(d,Njwx) max(svd([D]*Njwx*inv([D])));
obj_fun = @(d,Njwx) max(svd([blkdiag([d(1),d(3)+i*d(4);d(3)-i*d(4),d(2)],[d(5)],[d(6)*eye(2)])]...
    *Njwx*inv([blkdiag([d(1),d(3)+i*d(4);d(3)-i*d(4),d(2)],[d(5)],[d(6)*eye(2)])])));
% test the cost function
% obj_fun(d,Njwx)
obj_fun(randn(6,1).^2,Njw1)
%
% MINIMIZE THE COST WRT "D"
% x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
d_vec = fmincon(@(d)obj_fun(d,Njw1),[randn(6,1).^2],[],[],[],[],1e-5*ones(6,1),10*ones(6,1),[],[]);
d_mat=blkdiag([d_vec(1),d_vec(3)+i*d_vec(4);d_vec(3)-i*d_vec(4),d_vec(2)],[d_vec(5)],[d_vec(6)*eye(2)])
% COMPUTE THE COST FOR THE MINIMIZER-D , "THAT IS THE ANSWER"
FMINCON_SSV=max(svd(d_mat*Njw1*inv(d_mat)));
%
rho_N=max(abs(eig(Njw1)));                    % spectral radius of Njw1
sigma_max_N=max(svd(Njw1));                   % simga-max of Njw1

% COMPUTE THE SSV USING MATLAB-MUSSV FCN
% bounds = mussv(N,BlockStructure)
[bounds,muinfo] = mussv(Njw1,[2,0;1,1;2,2])
MATLAB_MUSSV=bounds(2);
clc;
fprintf("FMINCON_SSV \t rho_N \t sigma_max_N \t MATLAB_MUSSV\n");
fprintf("%6.4f       \t %6.4f \t %6.4f       \t %6.4f\n",FMINCON_SSV,rho_N,sigma_max_N,MATLAB_MUSSV);
```

Code OUTPUT

```
FMINCON_SSV        rho_N    sigma_max_N       MATLAB_MUSSV
1.6757             1.6313         1.6797             1.6757
```

CASE 4: MIXED CASE [RECTANGULAR CASE]

SSV COMPUTATION [FOR A FIXED FREQ]

$$N(s)=\begin{bmatrix}\frac{1}{s+1} & \frac{1}{s+2} & \frac{1}{s+3} & \frac{1}{s+3} & \frac{1}{s+4} & \frac{1}{s+3} & \frac{1}{s+4}\\ \frac{1}{s+5} & \frac{1}{s+1} & \frac{1}{s+2} & \frac{1}{s+3} & \frac{1}{s+4} & \frac{1}{s+3} & \frac{1}{s+4}\\ \frac{1}{s+2} & \frac{1}{s+4} & \frac{1}{s+7} & \frac{1}{s+3} & \frac{1}{s+4} & \frac{1}{s+3} & \frac{1}{s+2}\\ \frac{1}{s+2} & \frac{1}{s+4} & \frac{1}{s+7} & \frac{1}{s+3} & \frac{1}{s+4} & \frac{1}{s+3} & \frac{1}{s+4}\\ \frac{1}{s+2} & \frac{1}{s+4} & \frac{1}{s+7} & \frac{1}{s+3} & \frac{1}{s+4} & \frac{1}{s+3} & \frac{1}{s+3}\\ \frac{1}{s+5} & \frac{1}{s+1} & \frac{1}{s+2} & \frac{1}{s+3} & \frac{1}{s+4} & \frac{1}{s+3} & \frac{1}{s+4}\\ \frac{1}{s+2} & \frac{1}{s+4} & \frac{1}{s+7} & \frac{1}{s+3} & \frac{1}{s+4} & \frac{1}{s+3} & \frac{1}{s+4}\end{bmatrix}$$

$$N(j1)=\begin{bmatrix}\frac{1}{s+1} & \frac{1}{s+2} & \frac{1}{s+3} & \frac{1}{s+3} & \frac{1}{s+4} & \frac{1}{s+3} & \frac{1}{s+4}\\ \frac{1}{s+5} & \frac{1}{s+1} & \frac{1}{s+2} & \frac{1}{s+3} & \frac{1}{s+4} & \frac{1}{s+3} & \frac{1}{s+4}\\ \frac{1}{s+2} & \frac{1}{s+4} & \frac{1}{s+7} & \frac{1}{s+3} & \frac{1}{s+4} & \frac{1}{s+3} & \frac{1}{s+2}\\ \frac{1}{s+2} & \frac{1}{s+4} & \frac{1}{s+7} & \frac{1}{s+3} & \frac{1}{s+4} & \frac{1}{s+3} & \frac{1}{s+4}\\ \frac{1}{s+2} & \frac{1}{s+4} & \frac{1}{s+7} & \frac{1}{s+3} & \frac{1}{s+4} & \frac{1}{s+3} & \frac{1}{s+3}\\ \frac{1}{s+5} & \frac{1}{s+1} & \frac{1}{s+2} & \frac{1}{s+3} & \frac{1}{s+4} & \frac{1}{s+3} & \frac{1}{s+4}\\ \frac{1}{s+2} & \frac{1}{s+4} & \frac{1}{s+7} & \frac{1}{s+3} & \frac{1}{s+4} & \frac{1}{s+3} & \frac{1}{s+4}\end{bmatrix}_{s\leftarrow j1}$$

CASE 5: MIXED CASE [RECTANGULAR CASE]

$$\mathbf{\Delta} = \left\{ \begin{bmatrix} \delta_1 I_2 & & & \\ & \delta_2 & & \\ & & \Delta_3 & \\ & & & \Delta_4 \end{bmatrix} : \begin{array}{c} \boxed{\delta_1 \in \mathbb{C}} \\ \boxed{\delta_2 \in \mathbb{C}} \\ \boxed{\Delta_3 \in \mathbb{C}^{2\times 3}} \\ \boxed{\Delta_4 \in \mathbb{C}^{2\times 3}} \end{array} \right\}, \mathcal{D} = \{D : D\Delta = \Delta D, \forall \Delta \in \mathbf{\Delta}\}$$

$$\mathcal{D} = \{\mathcal{D}_L, \mathcal{D}_R\} = \left\{ D_L = \begin{bmatrix} D_1 & & & \\ & D_2 & & \\ & & D_3 I_3 & \\ & & & D_4 \end{bmatrix}, D_R = \begin{bmatrix} D_1 & & & \\ & D_2 & & \\ & & D_3 I_2 & \\ & & & D_4 I_2 \end{bmatrix} : \begin{array}{c} \boxed{D_1 \in \mathbb{C}^{2\times 2}, D_1 = D_1^H > 0} \\ \boxed{D_2 \in \mathbb{R}, D_2 > 0} \\ \boxed{D_3 \in \mathbb{R}, D_3 > 0} \\ \boxed{D_4 \in \mathbb{R}, D_4 > 0} \end{array} \right\}$$

$$\mathcal{D} = \{\mathcal{D}_L, \mathcal{D}_R\} = \left\{ D_L = \begin{bmatrix} \boxed{\begin{matrix} d_1 & d_3 + jd_4 \\ d_3 - jd_4 & d_2 \end{matrix}} & & & \\ & \boxed{d_5} & & \\ & & \boxed{d_6 I_3} & \\ & & & \boxed{d_7} \end{bmatrix}, D_R = \begin{bmatrix} \boxed{\begin{matrix} d_1 & d_3 + jd_4 \\ d_3 - jd_4 & d_2 \end{matrix}} & & & \\ & \boxed{d_5} & & \\ & & \boxed{d_6 I_2} & \\ & & & \boxed{d_7 I_2} \end{bmatrix} : \begin{array}{c} \boxed{d_1, \ldots, d_7 \in \mathbb{R}} \\ \boxed{d_1, \ldots, d_7 \in \mathbb{R}} \end{array} \right\}$$

$$\mu_{\mathbf{\Delta}}(M) = \mu_{\mathbf{\Delta}}(DMD^{-1}) \rightarrow \mu_{\mathbf{\Delta}}(M) \leq \inf_{D \in \mathcal{D}} \bar{\sigma}(DMD^{-1}) = \inf_{d_1, \ldots, d_7 > 0} \bar{\sigma}(DMD^{-1})$$

Since, there is a suitable parameterization for $\boxed{D \in \mathcal{D}}$, the problem can be expressed more specifically as,

MATLAB code

```
clear all;close all;clc;
N_s=[tf([1],[1,1]),tf([1],[1,2]),tf([1],[1,3]),tf([1],[1,3]),tf([1],[1,4]),tf([1],[1,3]),tf([1],[1,4]);
     tf([1],[1,5]),tf([1],[1,1]),tf([1],[1,2]),tf([1],[1,3]),tf([1],[1,4]),tf([1],[1,3]),tf([1],[1,4]);
     tf([1],[1,2]),tf([1],[1,4]),tf([1],[1,7]),tf([1],[1,3]),tf([1],[1,4]),tf([1],[1,3]),tf([1],[1,2]);
     tf([1],[1,2]),tf([1],[1,4]),tf([1],[1,7]),tf([1],[1,3]),tf([1],[1,4]),tf([1],[1,3]),tf([1],[1,4]);
     tf([1],[1,2]),tf([1],[1,4]),tf([1],[1,7]),tf([1],[1,3]),tf([1],[1,4]),tf([1],[1,3]),tf([1],[1,3]);
     tf([1],[1,5]),tf([1],[1,1]),tf([1],[1,2]),tf([1],[1,3]),tf([1],[1,4]),tf([1],[1,3]),tf([1],[1,4]);
     tf([1],[1,2]),tf([1],[1,4]),tf([1],[1,7]),tf([1],[1,3]),tf([1],[1,4]),tf([1],[1,3]),tf([1],[1,4])];
%% compute the matrix for a given freq
Njw1=evalfr(N_s,1*i)
%% CASE 5:  DELTA STRUCTURE = blkdiag([delta1*eye(2),delta2,DELTA_3,DELTA_4]):
% delta1,delta2 in COMPLEX, DELTA_3 in 2x3 COMPLEX MATRIX, DELTA_4 in 2x3 COMPLEX MATRIX
% COST FUNCTION
% DL=blkdiag([d(1),d(3)+i*d(4);d(3)-i*d(4),d(2)],[d(5)],[d(6)*eye(3)],[d(7)])
% DR=blkdiag([d(1),d(3)+i*d(4);d(3)-i*d(4),d(2)],[d(5)],[d(6)*eye(2)],[d(7)*eye(2)])
% obj_fun = @(d,Njwx) max(svd([DL]*Njwx*inv([DR])));
DLgen=@(d) blkdiag([d(1),d(3)+i*d(4);d(3)-i*d(4),d(2)],[d(5)],[d(6)*eye(3)],[d(7)])
DRgen=@(d) blkdiag([d(1),d(3)+i*d(4);d(3)-i*d(4),d(2)],[d(5)],[d(6)*eye(2)],[d(7)*eye(2)])
obj_fun = @(d,Njwx) max(svd([DLgen(d)]*Njwx*inv([DRgen(d)])));
% MINIMIZE THE COST WRT "D"
% x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
d_vec = fmincon(@(d)obj_fun(d,Njw1),[randn(7,1).^2],[],[],[],[],1e-5*ones(7,1),10*ones(7,1),[],[]);
DL_mat=DLgen(d_vec)
DR_mat=DRgen(d_vec)
% COMPUTE THE COST FOR THE MINIMIZER-D , "THAT IS THE ANSWER"
FMINCON_SSV=max(svd(DL_mat*Njw1*inv(DR_mat)));
rho_N=max(abs(eig(Njw1)));                            % spectral radius of Njw1
sigma_max_N=max(svd(Njw1));                           % simga-max of Njw1
% COMPUTE THE SSV USING MATLAB-MUSSV FCN
% bounds = mussv(N,BlockStructure)
[bounds,muinfo] = mussv(Njw1,[2,0;1,1;2,3;2,1])
MATLAB_MUSSV=bounds(2);
clc;
fprintf("FMINCON_SSV \t rho_N \t sigma_max_N \t MATLAB_MUSSV\n");
fprintf("%6.4f       \t %6.4f \t %6.4f       \t %6.4f\n",FMINCON_SSV,rho_N,sigma_max_N,MATLAB_MUSSV);
```

Code OUTPUT

FMINCON_SSV	rho_N	sigma_max_N	MATLAB_MUSSV
2.1974	2.2120	2.2646	2.1974

Kemin ZHOU,John C Doyle "Essentials of Robust Control" EXAMPLE 10.2

INFO ABOUT MATLAB MUSSV

$\gg$ **[bounds,rowd] = mu(M,blk)**

where the structure of the Δ is specified by a two-column matrix **blk**. for example, a

$$\Delta = \begin{bmatrix} \delta_1 I_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \delta_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \Delta_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta_5 I_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Delta_6 \end{bmatrix}$$

$$\delta_1, \delta_2, \delta_5, \in \mathbb{C}, \ \Delta_3 \in \mathbb{C}^{2\times 3}, \Delta_4 \in \mathbb{C}^{3\times 3}, \Delta_6 \in \mathbb{C}^{2\times 1}$$

can be specified by

$$\mathbf{blk} = \begin{bmatrix} 2 & 0 \\ 1 & 1 \\ 2 & 3 \\ 3 & 3 \\ 3 & 0 \\ 2 & 1 \end{bmatrix}.$$

Note that Δ_j is not required to be square. The outputs of the program include a 2×1 vector **bounds** containing the upper and lower bounds of $\mu_{\mathbf{\Delta}}(M)$ and the row vector **rowd** containing the scaling D. The D matrix can be recovered by

$\gg [\mathbf{D}_\ell, \mathbf{D_r}] = \mathbf{unwrapd}(\mathbf{rowd}, \mathbf{blk})$

where D_ℓ and D_r denote the left and right scaling matrices used in computing the upper-bound $\inf \overline{\sigma}\left(D_\ell M D_r^{-1}\right)$ when some full blocks are not necessarily square and they are equal if all full blocks are square.

Kemin ZHOU,John C Doyle “Essentials of Robust Control” EXAMPLE 10.2

INFO ABOUT MATLAB MUSSV

INFO ABOUT D_L, D_R

$$\mu_{\mathbf{\Delta}}(M) \le \inf_{D \in \mathcal{D}} \bar{\sigma}(DMD^{-1})$$

$$\mu_{\mathbf{\Delta}}(M) \le \inf_{D \in \mathcal{D}} \bar{\sigma}(D_L M D_R^{-1})$$

and

$$\mathbf{\Delta} = \left\{ \Delta = \begin{bmatrix} \delta_1 I_2 & & & \\ & \delta_2 & & \\ & & \Delta_3 & \\ & & & \Delta_4 \end{bmatrix} : \ \delta_1, \delta_2 \in \mathbb{C}, \Delta_3 \in \mathbb{C}^{2\times 3}, \Delta_4 \in \mathbb{C}^{2\times 1} \right\}.$$

Then blk $= \begin{bmatrix} 2 & 0 \\ 1 & 1 \\ 2 & 3 \\ 2 & 1 \end{bmatrix}$ and the MATLAB program gives bounds $= \begin{bmatrix} 10.5955 & 10.5518 \end{bmatrix}$

and

$$D_\ell = \begin{bmatrix} D_1 & & & \\ & 0.7638 & & \\ & & 0.8809 I_3 & \\ & & & 1.0293 \end{bmatrix}$$

$$D_r = \begin{bmatrix} D_1 & & & \\ & 0.7638 & & \\ & & 0.8809 I_2 & \\ & & & 1.0293 I_2 \end{bmatrix}$$