component_annihilation

April 27, 2017

1 Component Annihilation

The following note is a description of a method for component annihilation for stereo-panning estimation. In the following description, this method is described as a post-processing procedure. However, it is desirable to implement the functionality of this method as part of the likelihood-model in the segmentation algorithm and this still remains unsolved.

1.0.1 Fitting a Gaussian mixture model to panning parameter space

The challenge of overfitting, a Gaussian mixture to the distribution space is amiguous. It is the case that both the MMDL, BIC and Varational Bayes will estimate a model order k that is larger than the true order. However, the GMM-model is designed to describe every single data point as being part of a Gaussian distribution. The ambiguouity is that the overfitting of the Gaussian mixture model can be exploited as a parameter to utilize for initialization of the EM-algorithm, which is also the case for the MMDL [?].

By starting with k, where k is much larger than the true/optimal number of mixture components, the adopted algorithm is robust with respect to initialization of the EM-algorithm. The MMDL algorithm applies component annihilation, by adopting a Dirichlet prior on the mixing probabilities [?], and selects the number of components by annihilating the weakest component in the M-step of an iterative component-wise EM (CEM^2) [?]. This procedure leads to a smaller model order and still describes every data point as being part of a Gaussian distribution. It is important to notice that every true parameter is then described by at least one or more of the clusters, often more than one.

1.0.2 Clustering and component metrics

After model selection by the MMDL algorithm, each cluster is described by one or more components. Therefore, we have applied a post-processing step to select the correct number of clusters from prior spatial knowledge of the the conditional distributions. This post-processing step selects clusters from an analysis on each cluster covariance compared to the number of estimated points embedded in each cluster. We know from [?] that due to the non-disjoint spectral overlap of sources, the variance increases in the amplitude direction. Therefore, we propose to select clusters with largest amount estimated points, relative to the size of the respective covariance and its rotational angle in the parameter space. We describe this for the kth covariance $\mathbf{C}_k$ in the following.

We define $\mathbf{C}_k$ geometrically as an ellipsoid by applying the singular value decomposition as,

$$\mathbf{C}_k = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{svd1}$$

Where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal rotation matrices and the diagonal of $\mathbf{\Sigma}$ contains the principal axes a^2 and b^2.

We compute the angle θ of the principal axes a to the x-axis,

$$\theta = \tan\left(\mathbf{u_2}/\mathbf{u_1}\right)^{-1} \tag{angle1}$$

we center $\mathbf{C}_k$ by subtracting the mean μ_k as $\mathbf{d} = \mathbf{C}_k - \mu_k$

The x and y coordinates of each estimate is given as,

$$(x, y) = (\mathbf{d}_1 \cos\theta + \mathbf{d}_2 \sin\theta, -\mathbf{d}_1 \sin\theta + \mathbf{d}_2 \cos\theta) \tag{xy}$$

We count the number of points inside cluster k. The specific point (x, y) is inside the ellipse k if,

$$\frac{x^2}{a_k^2} + \frac{y^2}{b_k^2} < 1 \tag{elip}$$

Lastly we compute the size of $\mathbf{C}_k$ as the determinant of $\mathbf{C}_k$ and we compute the "shadow" of the covariance on the x-axis as $s_k = a_k \cos\theta_k + b_k \sin\theta_k$.

1.0.3 Annihilation steps

The following component annihilation uses the cluster parameters from the MMDL and by comparing these to the data, the true clusters are selected and the rest of the data points will be removed. In the first implementation (post processing) we have two rules applied which is:

- If a cluster shares an estimated point with a smaller cluster, all points that is only part of the bigger cluster is removed. The overlapping bigger clusters are refered to as sticky clusters.
- A geometric threshhold is applied based on $\det(\mathbf{C}_k)$, the number of points embedded in $\mathbf{C}_k$ and the shadow s on the x-axis.

1.0.4 Thresholding on the cluster angle and size

We know from [?] that spectral overlap causes larger variance in the amplitude direction, where optimally the covariance would be very small and diagonal or have greatest variance in the delay-direction. Therefore we can apply a threshhold from the rotation angle θ and the size of the region relative to the given number of estimates in the region.

We define a variable $0 \leq p_k \leq 1$ which is the percentage of points that is inside the kth cluster. We notice that the mixing parameter α is proportional to $\frac{p}{\det(\mathbf{C})}$. We define a metric of peakiness m for the kth source as,

$$m_k = \frac{p_k}{\det(\mathbf{C}_k) s_k} \tag{metric}$$

where s_k is the amplitude shadow, $s_k = a_k \cos\theta_k + b_k \sin\theta_k$. The metric m_k carries implicit information of both the size and angle of the kth cluster region, that includes a percentage of all the estimated points (after sticky clusters have been removed). From all metrics $\mathbf{m}$ we define a threshold where m_k is relative to the smallest $m = m_1$. The metric ratio is,

$$ratio_k = \frac{m_k}{m_1} \tag{thr}$$

Through experiments, we have found that this method of component annihilation has good performance for precisely estimating the number of sources in the mixture and the panning parameters. This is shown in the following simulation on some mixture of 7 sources from the SQAM-database.

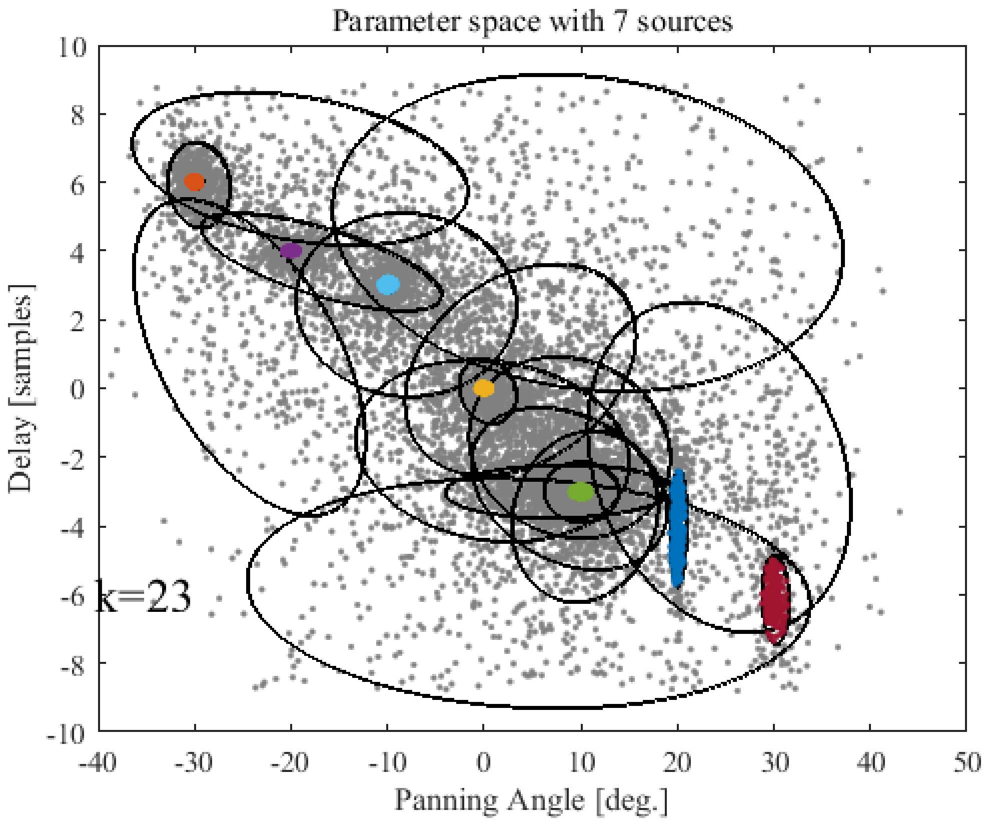
Simulation in MATLAB In the figure showing the parameter space it is clear that the covariance assumption desribed above holds and all 7 correct clusters have been estimated. The selected clusters are shown in color and the unselected clusters are shown in grey. The correct cluster covariances are either very small and close to diagonal or they are larger and have dominant variance mainly in the delay direction (upwards). We notice that the MMDL algorithm in this case chose a $k = 18$. Notice that the 11 "wrong clusters" are large with random covariance structure and rotation angle.

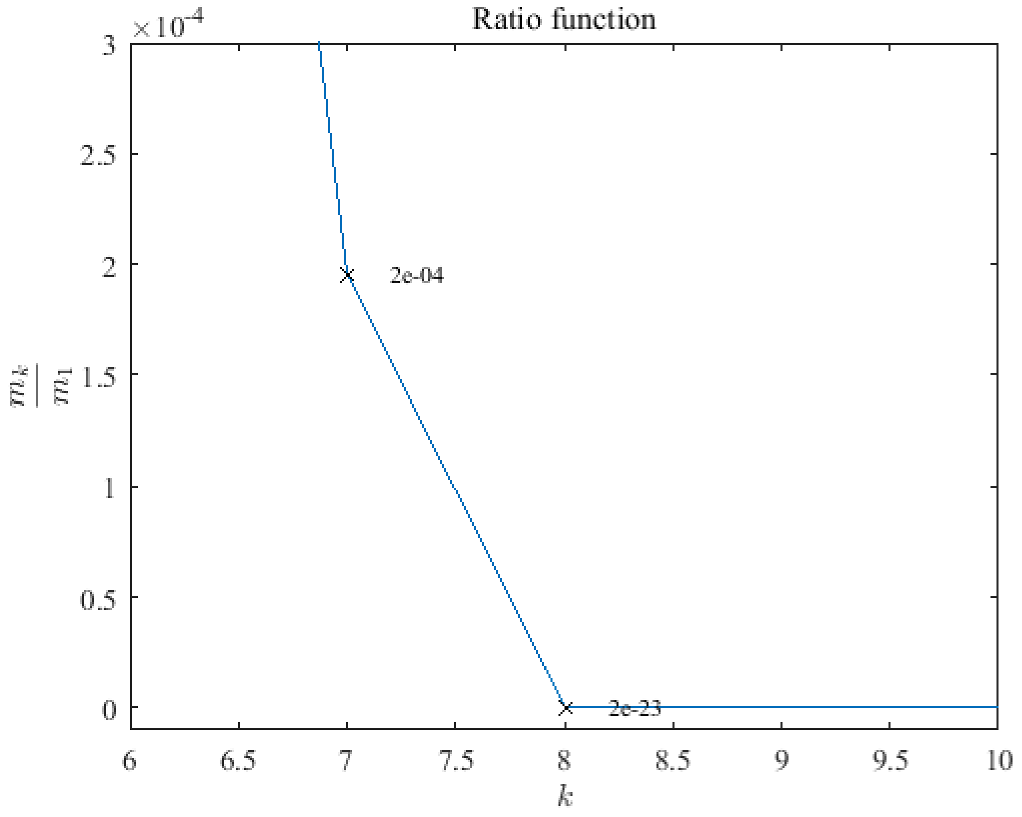
The next figure shows the ratio function in (thr). We note that between $k = 7$ and $k = 8$ there is a ratio difference on the order of 10^{18}, which often is sufficient for a fixed threshold.

```
In [1]: cd /home/jacob/panning_project/test_programs/;
        load panning_data2.mat;
        X = panning_data;
        fs=44100;
        trueNumClusters=7;
        figure;
        [~,~,~,~,~,~,~,~,estimatedParam,~,X2]  = ...
        smc_GMM_MMDL_normdetC(X.*[1/45 1/(150e-6*fs)], fs,trueNumClusters);
        xlabel('Panning Angle [deg.]'); ylabel('Delay [samples]');
        title('Parameter space with 7 sources');

        estimatedParam
```

```
Warning: Failed to converge in 100 iterations during replicate 4.
> In kmeans2/loopBody (line 183)
  In internal.stats.parallel.smartForReduce (line 136)
  In kmeans2 (line 53)
  In kmeans (line 314)
  In mixtures4_remove_outliers (line 113)
  In smc_GMM_MMDL_normdetC (line 16)
estimatedParam =
  -29.9942    6.0101
  -19.9993    4.0048
   -9.9693    3.0208
   -0.0075    0.0014
    9.9882   -3.0066
   20.0239   -4.0553
   30.0699   -6.1334
```

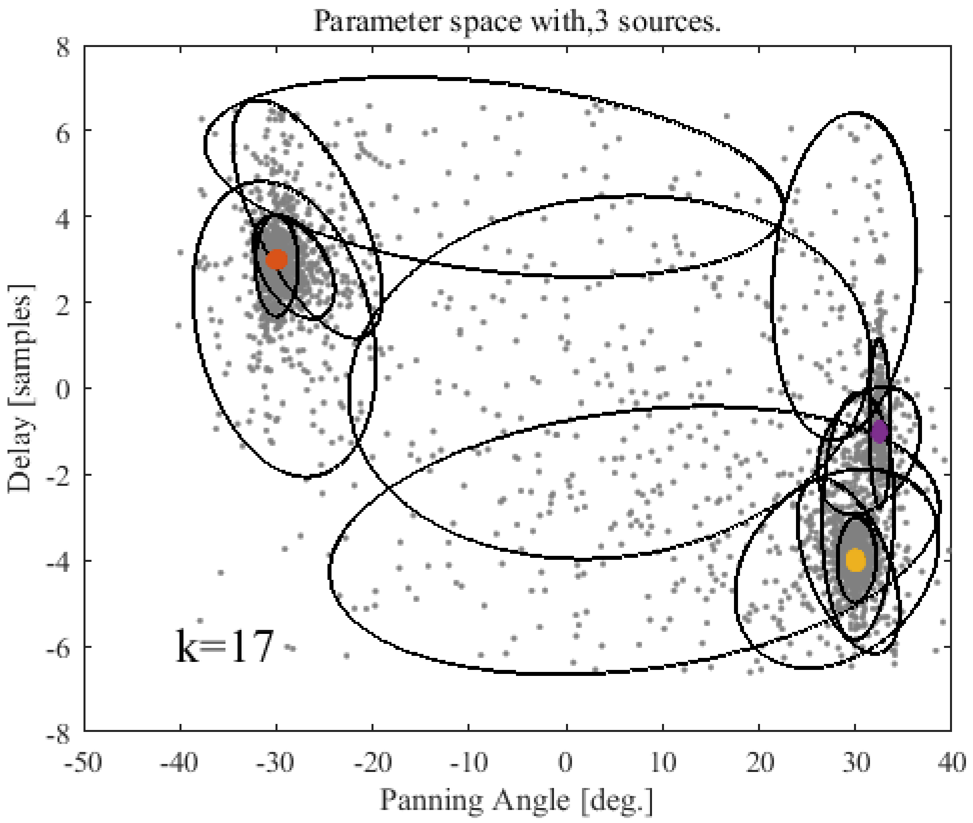

Parameter space with 7 sources
Delay [samples]
Panning Angle [deg.]
k=23
10
8
6
4
2
0
-2
-4
-6
-8
-10
-40
-30
-20
-10
0
10
20
30
40
50

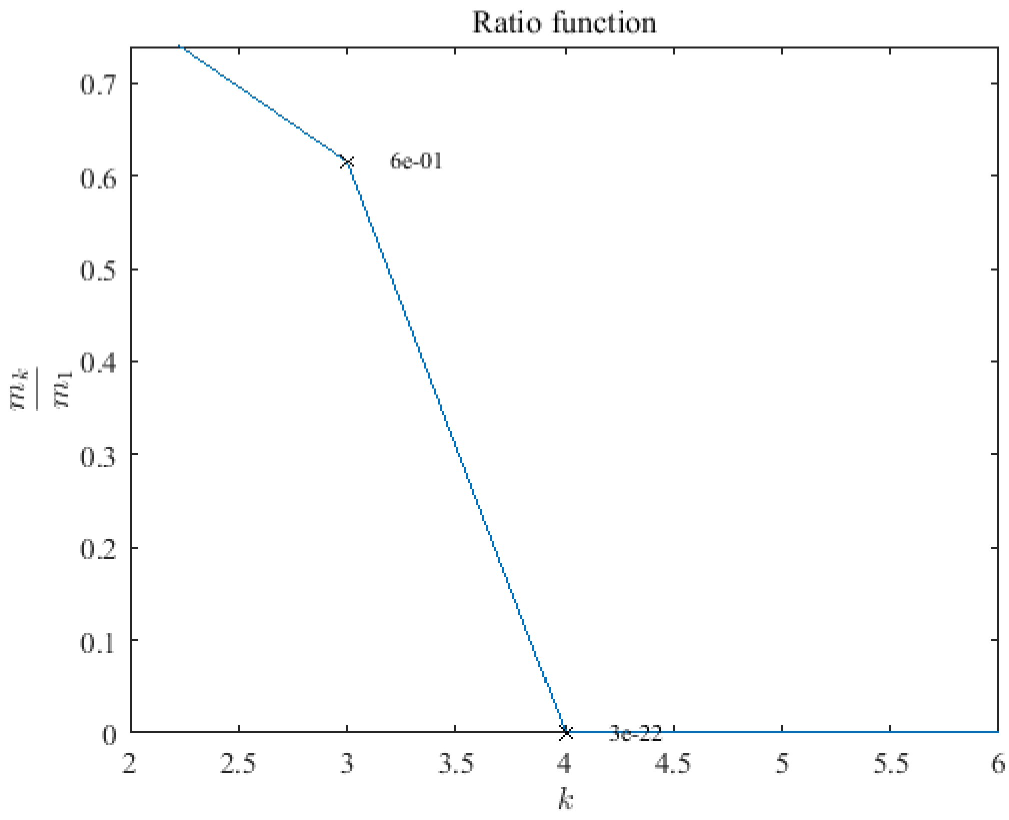

Ratio function
$\times 10^{-4}$
3
2.5
2
1.5
1
0.5
0
$\frac{m_k}{m_1}$
2e-04
2e-23
6
6.5
7
7.5
8
8.5
9
9.5
10
k

We run the same algorithm on 3 parameter spaces with mixtures made from the SQAM audio.

```
In [2]: clear all;
        cd /home/jacob/panning_project/test_programs/;
        load test_covration_100iter_notoptimal16april.mat
        index=randi(100);
        X=savedX{index};
        [~,~,~,~,~,~,~,~,estimatedParam,~,X2]  = ...
        smc_GMM_MMDL_normdetC(X, fs,size(true(index).Points,1));
        xlabel('Panning Angle [deg.]'); ylabel('Delay [samples]');
        title(strcat('Parameter space with, ', sprintf('%1.0f',size(true(index).Poi
        trueParam = true(index).Points
        estimatedParam
```

```
trueParam =
   32.5000   -1.0000
   30.0000   -4.0000
  -30.0000    3.0000
estimatedParam =
  -29.9591    2.9981
   29.9972   -4.0038
   32.4922   -0.9976
```

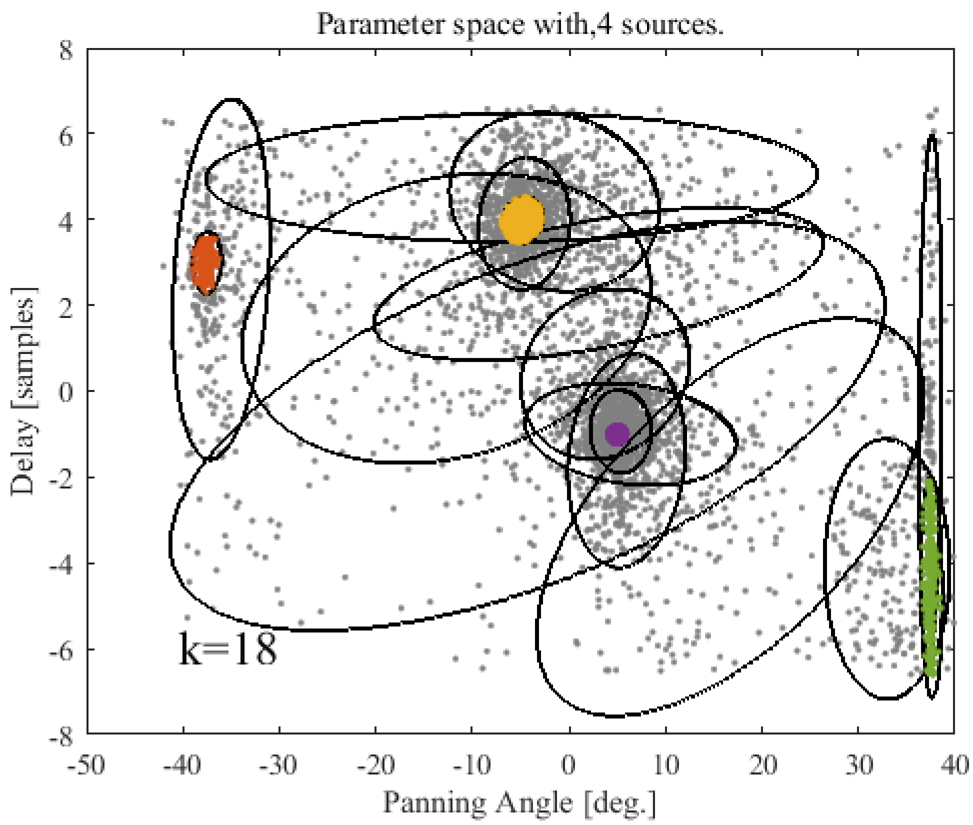

Parameter space with,3 sources.
Delay [samples]
Panning Angle [deg.]
k=17
8
6
4
2
0
-2
-4
-6
-8
-50
-40
-30
-20
-10
0
10
20
30
40

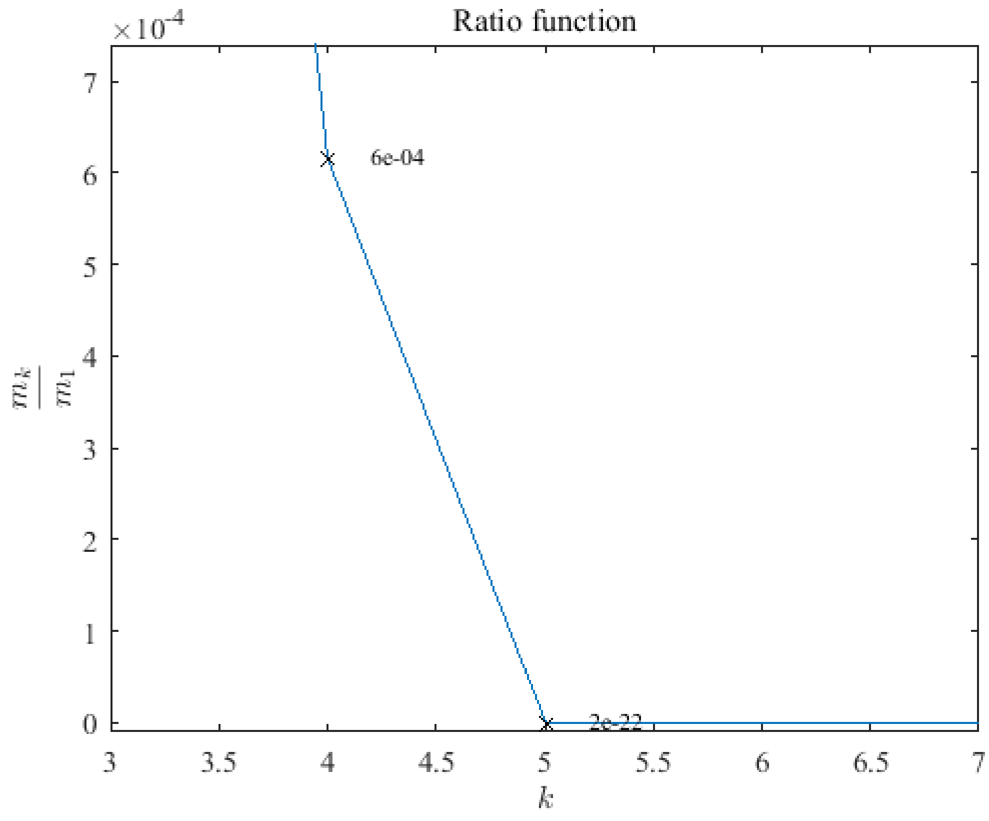


```
In [3]: clear all;
        cd /home/jacob/panning_project/test_programs/;
        load test_covration_100iter_notoptimal16april.mat
        index=randi(100);
        X=savedX{index};
        [~,~,~,~,~,~,~,~,estimatedParam,~,X2]  = ...
        smc_GMM_MMDL_normdetC(X, fs,size(true(index).Points,1));
        xlabel('Panning Angle [deg.]'); ylabel('Delay [samples]');
        title(strcat('Parameter space with, ', sprintf('%1.0f',size(true(index).Poi
        trueParam = true(index).Points
        estimatedParam

trueParam =
   37.5000   -5.0000
    5.0000   -1.0000
  -37.5000    3.0000
   -5.0000    4.0000
estimatedParam =
  -37.5622    3.0013
   -4.8934    3.9880
    4.9929   -1.0039
```

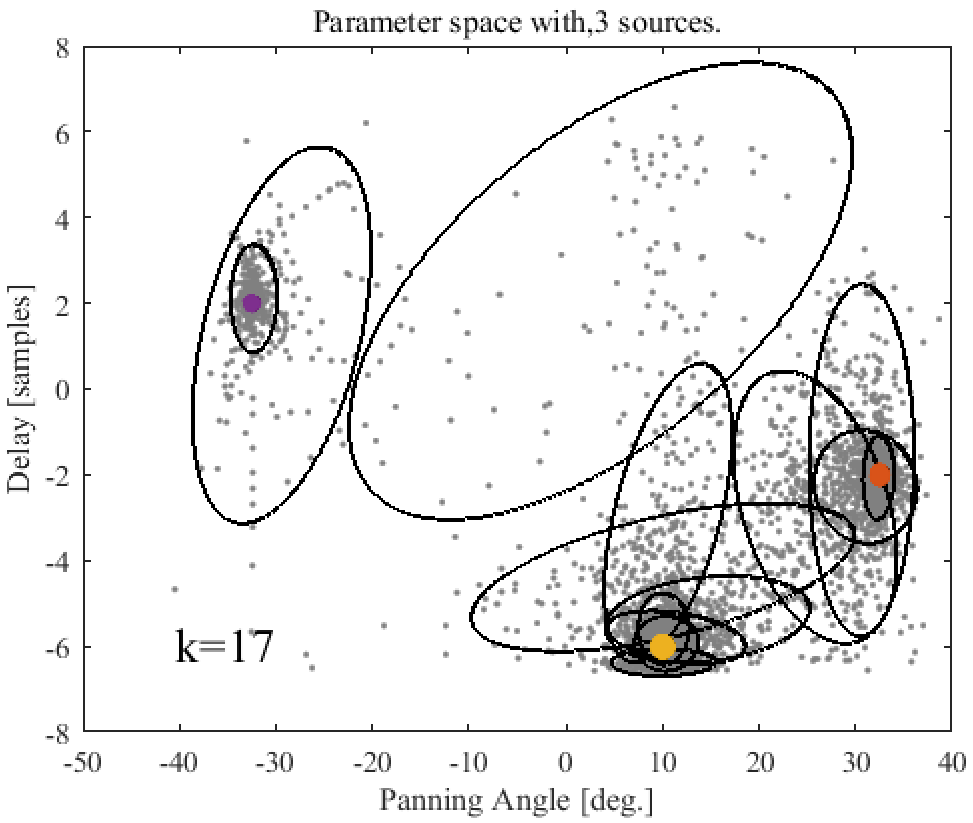
```
37.4738    -4.5625
```

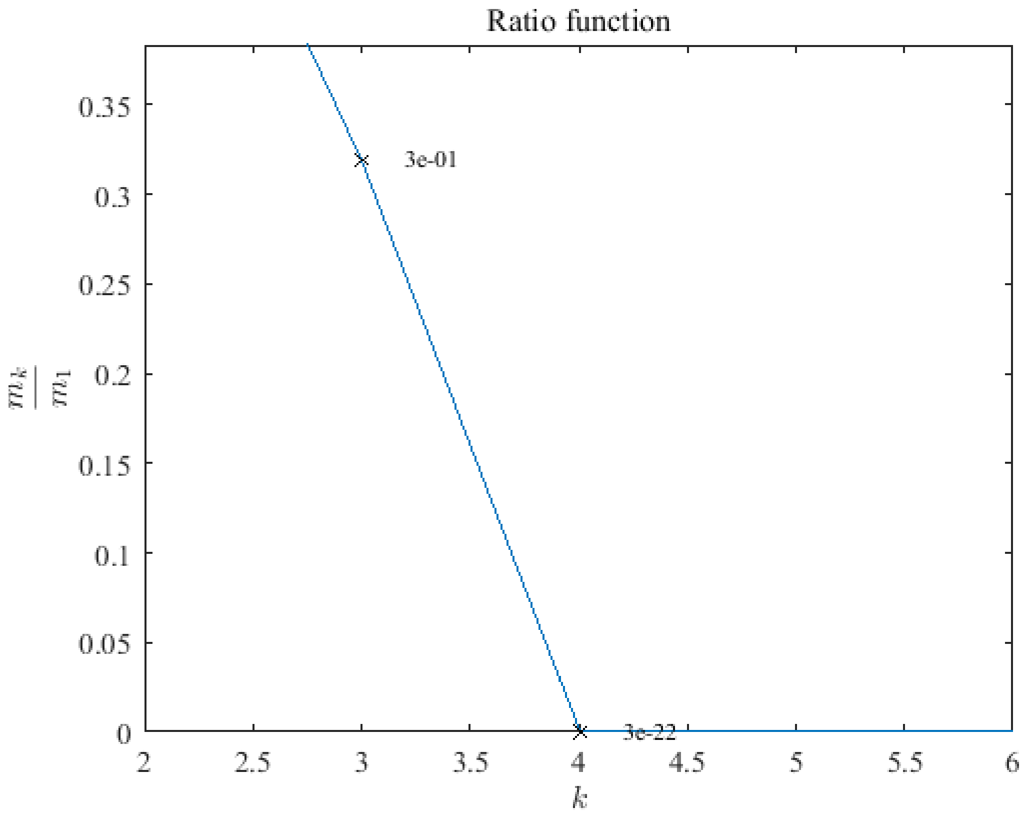




```
In [4]: clear all;
        cd /home/jacob/panning_project/test_programs/;
        load test_covration_100iter_notoptimal16april.mat
        index=randi(100);
        X=savedX{index};
        [~,~,~,~,~,~,~,~,estimatedParam,~,X2]  = ...
        smc_GMM_MMDL_normdetC(X, fs,size(true(index).Points,1));
        xlabel('Panning Angle [deg.]'); ylabel('Delay [samples]');
        title(strcat('Parameter space with, ', sprintf('%1.0f',size(true(index).Poi
        trueParam = true(index).Points
        estimatedParam

Warning: Failed to converge in 100 iterations during replicate 4.
> In kmeans2/loopBody (line 183)
  In internal.stats.parallel.smartForReduce (line 136)
  In kmeans2 (line 53)
  In kmeans (line 314)
  In mixtures4_remove_outliers (line 113)
  In smc_GMM_MMDL_normdetC (line 16)
trueParam =
   10.0000   -6.0000
   32.5000   -2.0000
  -32.5000    2.0000
estimatedParam =
  -32.5053    1.9984
   10.0131   -6.0106
   32.5140   -2.0014
```




Ratio function
$\frac{m_k}{m_1}$
k
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
2
2.5
3
3.5
4
4.5
5
5.5
6
3e-01
3e-22