**Interim report: Application Research on DOA Estimation Based on Software-Defined Radio Receiver**

|  |  |
| --- | --- |
| **Author** | Name：张艺洋 韦新元 Student ID: 11811523 11810421 |
| **Introduction**  Direction of arrival estimation is an active field in array signal processing. It has broad application value in the fields of communication, radar, exploration and navigation. However, most of the researchers innovated the DOA estimation algorithms, and most of these algorithms were verified on the simulation platform. As we all know, the experimental results of the simulation platform deviate from the results in engineering applications. This article focuses on the engineering application of DOA estimation, using a KerberosSDR device and four omnidirectional antennas as a signal receiver, and using a Raspberry Pi as a data processor to implement a system with a simple structure and reliable DOA estimation performance.  KerberosSDR is a new 4-input Coherent RTL-SDR. RTL-SDR is a very cheap software-defined radio receiver. Each RTL-SDR is composed of an RTL2832U chip and an R820T tuner. It can receive radio frequency signals from 25MHz to 1.75GHz in space and convert it to baseband. Finally, the digital 8-bit sampling signal is output from the USB port. There is a noise source module inside KerberosSDR, which can realize the sampling time synchronization and phase synchronization of the four signal receiving channels. The four signal receiving channels share a clock source, and the four digital signals communicate with the Raspberry Pi through a USB HUB. Run the signal processing algorithm on the Raspberry Pi and display the DOA estimation result and signal strength in real time through the web page.  The communication frequency between the UAV and the remote control in this experiment is 2.400- 2.4835 GHz, which is not in the RTL-SDR receiving frequency range, Therefore, a small FM transceiver is fixed on the UAV as a signal source, The transmission frequency of the FM transceiver is 446.0063MHz. The UAV equipped with a FM transceiver hovers in the air, so as to ensure that the signal sent by the FM transceiver is not blocked by obstacles, Use this signal source to verify the DOA estimation accuracy of the system.  **Theoretical knowledge:**  **DOA(Direction Of Arrival)**  **Introduction**  Suppose the system have M antennas, N signal packages, K targets.  Begin with time difference, if the signal arrives at ULA with angle , from the figure we can notice that there are different s, which causes phase difference , where c is the propagation speed of light, m is the number of arrays.  图表, 雷达图  描述已自动生成  We can induce the formula of arrived signal , assume there is only one signal package  We can simplify  When it comes to N packages, they come from N different directions:  **The simplest DOA estimation: spatial Fourier transform**  The form of the received signal    Although we don't know the angle of the signal, for a given array, the mathematical form of its steering vector is known. For example, for ULA, it must be of Vandermonde structure. Based on this, we have a method of DOA estimation.  Specifically, we can construct a steering vector, the angle of which may be given as α, then we can construct a steering vector with the incoming wave direction α as    Use our assumed steering vector a(α) and the received signal to do the vector inner product, that is    The result should be a scalar. A simple calculation can get    The equal sign is taken at α=θ.  From this inequality, we can see that if we are right, that is, α=θ, then the result obtained is a maximum value.  Therefore, we can guess all the angles again and find the one with the largest result. The corresponding angle is the result of our DOA estimation.  Here can lead to a method of DOA estimation, the pseudo code is presented as follows:    **Algorithm simulation example**  Example 1: Assuming that there is only one target at θ1=5°, the result is  https://img-blog.csdnimg.cn/20200511222035229.jpg?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3B3YW5nOTU=,size_16,color_FFFFFF,t_70#pic_center  Example 2: Assuming that the two targets are respectively located at θ 1 = 5 °, θ 2 = 10 °, the result obtained by the above method is  https://img-blog.csdnimg.cn/20200511222153895.jpg?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3B3YW5nOTU=,size_16,color_FFFFFF,t_70#pic_center  Example 3: Assuming that the two targets are located at θ 1 = 5 °, θ 2 = 30 °, the result obtained by the above method is  https://img-blog.csdnimg.cn/20200511222236394.jpg?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3B3YW5nOTU=,size_16,color_FFFFFF,t_70#pic_center  It can be seen from the three simulation examples that there is no problem with a single target, but when the two targets are too close, the DOA algorithm cannot distinguish between the two targets. This brings certain problems to our experiment:  1. The first is the actual effect of this algorithm. We can see that as the target approaches in the experiment, the effect of the DOA algorithm is relatively poor, and we cannot effectively distinguish the target. This requires us to introduce an effective distance threshold for the algorithm. When the distance is less than this threshold, we cannot use this algorithm to distinguish.  2. Combining the conclusions we got in the previous experiments and the knowledge that Mr. Wu told us in class, we can know that for an algorithm, there is always an extra cost. The additional cost of the algorithm is an important constraint that Mr. Wu repeatedly emphasizes throughout the communication principles and the entire content of the wireless communication course. This brings us to the question that needs to be considered in our experiments: Is there a higher resolution algorithm? And is there any additional overhead proposed by Mr. Wu for this algorithm?  **Traditional: MVDR(Minimum Variance Distortionless Response) Method etc**  First introducing weight vector , this vector helps us coordinate a specific direction to receive signals. It also make a great contribution in constraining the variance.  图示  描述已自动生成  The beam formed signal can be written as:  From the formula above we can calculate the beam formed signal power  If we take out the original signal :  Obviously, we want to minimize the noise and makes the signal go through the gateway completely, so we have our mathematic expression:  **MVDR beamforming calculation steps**  Step1: Estimate the autocorrelation matrix R from the received snapshot signal x (n );  Step2: Calculate the inverse matrix R^-1 of the autocorrelation matrix R;  Step3: According to the geometry of the array, construct the corresponding steering vector a(θ);  Step4: Make θ follow a certain step, scan at the angle you want to observe, and calculate Pθ successively;  Step5: Perform spectral peak search on Pθ to find the θ corresponding to the peak point;  **Conclusions and reflections**  1. The MVDR beamforming method can only process incoherent signals.  In solving the equation (8), the inverse operation of the autocorrelation matrix R is carried out. This requires R to be full rank, that is, the signals are irrelevant. If there is a coherent signal, then the above derivation cannot continue until equation (8). So, what if the signals are coherent?  2. MVDR beamforming is versatile, not limited to linear arrays.  It can be seen from the derivation throughout the text that there is no specific structure applied to a (θ ). For other forms of arrays, just modify the form of a (θ );  Use the MVDR beamforming method for DOA estimation without knowing the number of sources. MUSIC, ESPRIT algorithms, etc. all need to estimate the number of sources;  Using the MVDR beamforming method for DOA estimation, the resolution is much higher than that of the spatial FFT, which can be seen from the following simulation.  **Simulation results**  Suppose a uniform linear array has 16 elements, λ / 2 array; take 1024 snapshots to estimate the autocorrelation matrix R, two signals enter the large array from 10° and 20° directions respectively, and the signal-to-noise ratio is 10dB. Taking the signal coherent and incoherent conditions, using the MVDR beamforming method described in this article and spatial FFT and DOA estimation, the results are as follows.  5.1 DOA estimation with MVDR beamforming method  https://img-blog.csdnimg.cn/20200527130251494.jpg?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3B3YW5nOTU=,size_16,color_FFFFFF,t_70#pic_centerhttps://img-blog.csdnimg.cn/20200527130334772.jpg?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3B3YW5nOTU=,size_16,color_FFFFFF,t_70#pic_center  It can be seen from the simulation results that when the signal is incoherent, this method has a higher resolution; but when the signal is coherent, although there are still two peaks in the 10° and 20° directions, the corresponding ordinate is smaller. , And there are peaks in other places, which brings difficulty to the subsequent detection algorithm.  As a comparison, the results of the spatial FFT are also placed here. It can be seen that the resolution of the MVDR beamforming method is much higher.  https://img-blog.csdnimg.cn/20200527130752526.jpg?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3B3YW5nOTU=,size_16,color_FFFFFF,t_70#pic_center  **Conventional Subspace-Based: MUSIC, ESPRIT**  **MUSIC(Multiple Signal Classification)**  The MUSIC algorithm is also called the decomposition subspace algorithm. The MUSIC algorithm has good angle measurement performance when performing DOA estimation on non-coherent signal sources. Since the MUSIC algorithm breaks through the performance bottleneck of the linear prediction algorithm, it can distinguish multiple target signal sources existing in a beam.  The mathematical model of the target signal source is:  Assuming that the noise is spatially ideal white noise and the noise power is , the received data covariance matrix of the antenna array can be obtained from above:  Eigenvalue decomposition of **:**  Where is a subspace formed by eigenvector corresponding to large eigenvalues, which also becomes a signal subspace, and is a subspace formed by eigenvector corresponding to small eigenvalues, and also becomes a noise subspace. Under ideal conditions, the steering vector in the signal subspace is orthogonal to the noise subspace:  Considering that the actual received data matrix is limited, the maximum likelihood estimate of the covariance matrix is:  The MUSIC algorithm is implemented with minimum optimized search:  The spatial spectral of MUSIC algorithm is:  This is the matlab simulation process carried out in our root data  clc; clear all; close all;  %% -------------------------initialization-------------------------  f = 500; % frequency  c = 1500; % speed sound  lambda = c/f; % wavelength  d = lambda/2; % array element spacing  M = 10; % number of array elements  N = 100; % number of snapshot  K = 6; % number of sources  doa\_phi = [-30, 0, 20, 40, 60, 75]; % direction of arrivals  %% generate signal  dd = (0:M-1)'\*d; % distance between array elements and reference element  A = exp(-1i\*2\*pi\*dd\*sind(doa\_phi)/lambda); % manifold array, M\*K  S = sqrt(2)\(randn(K,N)+1i\*randn(K,N)); % array of random signal, K\*N  X = A\*S; % received data without noise, M\*N  X = awgn(X,10,'measured'); % received data with SNR 10dB  %% calculate the covariance matrix of received data and do eigenvalue decomposition  Rxx = X\*X'/N; % covariance matrix  [U,V] = eig(Rxx); % eigenvalue decomposition  V = diag(V); % vectorize eigenvalue matrix  [V,idx] = sort(V,'descend'); % sort the eigenvalues in descending order  U = U(:,idx); % reset the eigenvector  P = sum(V); % power of received data  P\_cum = cumsum(V); % cumsum of V  %% define the noise space  J = find(P\_cum/P>=0.95); % or the coefficient is 0.9  J = J(1); % number of principal component  Un = U(:,J+1:end);  %% music for doa; seek the peek  theta = -90:0.1:90; % steer theta  doa\_a = exp(-1i\*2\*pi\*dd\*sind(theta)/lambda); % manifold array for seeking peak  music = abs(diag(1./(doa\_a'\*(Un\*Un')\*doa\_a))); % the result of each theta  music = 10\*log10(music/max(music)); % normalize the result and convert it to dB  %% plot  figure;  plot(theta, music, 'linewidth', 2);  title('Music Algorithm For Doa', 'fontsize', 16);  xlabel('Theta(°)', 'fontsize', 16);  ylabel('Spatial Spectrum(dB)', 'fontsize', 16);  grid on;    It can be seen that when the incident signals are not correlated with each other, the traditional MUSIC algorithm can detect the approximate direction of arrival of six sources with high resolution, which are -29.7°, 0°, 19.8°, 39.8°, 60.4°, 74.7° , But there is still the problem of estimation accuracy, and there are many improved MUSIC algorithms that can be improved.  It should be noted that the degree of freedom of a half-wavelength uniform linear array with the number of elements M is M-1, which means that the maximum number of sources that can be resolved by the linear array is M-1. At the same time, if there is a coherent source, the effect of the MUSIC algorithm will be unsatisfactory  **Spatial smoothing MUSIC algorithm**  According to the information we consulted, we found that when multiple incident signals are coherent, the traditional MUSIC algorithm is not ideal. This is because when the multiple incident signals we use are coherent, part of the energy will be dissipated into the noise subspace, making the MUSIC algorithm unable to effectively estimate it.  In order to solve this situation, we found out the relevant methods through research and investigation. We have mainly learned by looking up information  Decoherence through dimensionality reduction processing is called dimensionality reduction processing because this method splits the original array into many sub-arrays, and reconstructs the received data covariance matrix through the covariance matrix of the sub-arrays. The DOF of the array will vary with If it is reduced, the number of coherent signals that can be resolved is reduced.  Let's first look at the effect of traditional MUSIC algorithm for DOA estimation of coherent signals.  This is the matlab simulation process carried out in our root data  clc; clear all; close all;  %% -------------------------initialization-------------------------  f = 500; % frequency  c = 1500; % speed sound  lambda = c/f; % wavelength  d = lambda/2; % array element spacing  M = 20; % number of array elements  N = 100; % number of snapshot  K = 6; % number of sources  coef = [1; exp(1i\*pi/6);...  exp(1i\*pi/3); exp(1i\*pi/2);...  exp(2i\*pi/3); exp(1i\*2\*pi)]; % coherence coefficient, K\*1  doa\_phi = [-30, 0, 20, 40, 60, 75]; % direction of arrivals  %% generate signal  dd = (0:M-1)'\*d; % distance between array elements and reference element  A = exp(-1i\*2\*pi\*dd\*sind(doa\_phi)/lambda); % manifold array, M\*K  S = sqrt(2)\(randn(1,N)+1i\*randn(1,N)); % vector of random signal, 1\*N  X = A\*(coef\*S); % received data without noise, M\*N  X = awgn(X,10,'measured'); % received data with SNR 10dB  %% calculate the covariance matrix of received data and do eigenvalue decomposition  Rxx = X\*X'/N; % covariance matrix  [U,V] = eig(Rxx); % eigenvalue decomposition  V = diag(V); % vectorize eigenvalue matrix  [V,idx] = sort(V,'descend'); % sort the eigenvalues in descending order  U = U(:,idx); % reset the eigenvector  P = sum(V); % power of received data  P\_cum = cumsum(V); % cumsum of V  %% define the noise space  J = find(P\_cum/P>=0.95); % or the coefficient is 0.9  J = J(1); % number of principal component  Un = U(:,J+1:end);  %% music for doa; seek the peek  theta = -90:0.1:90; % steer theta  doa\_a = exp(-1i\*2\*pi\*dd\*sind(theta)/lambda); % manifold array for seeking peak  music = abs(diag(1./(doa\_a'\*(Un\*Un')\*doa\_a))); % the result of each theta  music = 10\*log10(music/max(music)); % normalize the result and convert it to dB  %% plot  figure;  plot(theta, music, 'linewidth', 2);  title('Music Algorithm For Doa', 'fontsize', 16);  xlabel('Theta(°)', 'fontsize', 16);  ylabel('Spatial Spectrum(dB)', 'fontsize', 16);  grid on;   This is the result of our algorithm simulation. It can be seen that for coherent signals, the traditional MUSIC algorithm DOA estimation effect is very poor. **Spatial smoothing algorithm**  The dimensionality reduction processing and decoherence methods mainly include spatial smoothing processing algorithms, and the spatial smoothing processing algorithms can be divided into forward spatial smoothing algorithm (FSS), backward smoothing algorithm (BSS), forward and backward smoothing algorithm (FBSS), as described above Said that the estimation effect of these algorithms is very good, but the aperture of the array is lost, resulting in a decrease in the number of resolvable coherent signals.  **Linear array signal model**  https://img-blog.csdnimg.cn/20201028152823249.png?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3FxXzM2NTgzMzcz,size_16,color_FFFFFF,t_70#pic_center  **Forward spatial smoothing algorithm**  The forward spatial smoothing algorithm divides the array into multiple overlapping sub-arrays, and then averages the covariance matrix of the data received by the sub-arrays. When the number of sub-array elements is greater than or equal to the number of coherent signals, the coherence can be effectively decohered.https://img-blog.csdnimg.cn/20201030120335719.png?#pic_center  As shown in the figure above, we evenly divide the M-element array into L sub-arrays, and each sub-array has N=M-L+1 array elements. Taking the leftmost sub-array as the reference array, define the received data of the J-th sub-array as:    Then the covariance matrix (also called the spatial smoothing matrix) of the received data of the J-th subarray can be expressed as    among them,    A1 is the flow matrix of the first sub-array, that is, the reference array.  Therefore, the covariance matrix after forward space smoothing can be obtained by averaging the covariance matrix of each sub-matrix.    Using forward spatial smoothing covariance matrix and MUSIC algorithm, the orientation of multiple coherent signals can be distinguished. It can be proved that this method can detect up to M/2 coherent signals.  This is the matlab simulation process carried out in our root data  clc; clear all; close all;  %% -------------------------initialization-------------------------  f = 500; % frequency  c = 1500; % speed sound  lambda = c/f; % wavelength  d = lambda/2; % array element spacing  M = 20; % number of array elements  N = 100; % number of snapshot  K = 6; % number of sources  L = 10; % number of subarray  L\_N = M-L+1; % number of array elements in each subarray  coef = [1; exp(1i\*pi/6);...  exp(1i\*pi/3); exp(1i\*pi/2);...  exp(2i\*pi/3); exp(1i\*2\*pi)]; % coherence coefficient, K\*1  doa\_phi = [-30, 0, 20, 40, 60, 75]; % direction of arrivals  %% generate signal  dd = (0:M-1)'\*d; % distance between array elements and reference element  A = exp(-1i\*2\*pi\*dd\*sind(doa\_phi)/lambda); % manifold array, M\*K  S = sqrt(2)\(randn(1,N)+1i\*randn(1,N)); % vector of random signal, 1\*N  X = A\*(coef\*S); % received data without noise, M\*N  X = awgn(X,10,'measured'); % received data with SNR 10dB  %% reconstruct convariance matrix  %% calculate the covariance matrix of received data and do eigenvalue decomposition  Rxx = X\*X'/N; % origin covariance matrix  Rf = zeros(L\_N, L\_N); % reconstructed covariance matrix  for i = 1:L  Rf = Rf+Rxx(i:i+L\_N-1,i:i+L\_N-1);  end  Rf = Rf/L;  [U,V] = eig(Rf); % eigenvalue decomposition  V = diag(V); % vectorize eigenvalue matrix  [V,idx] = sort(V,'descend'); % sort the eigenvalues in descending order  U = U(:,idx); % reset the eigenvector  P = sum(V); % power of received data  P\_cum = cumsum(V); % cumsum of V  %% define the noise space  J = find(P\_cum/P>=0.95); % or the coefficient is 0.9  J = J(1); % number of principal component  Un = U(:,J+1:end);  %% music for doa; seek the peek  dd1 = (0:L\_N-1)'\*d;  theta = -90:0.1:90; % steer theta  doa\_a = exp(-1i\*2\*pi\*dd1\*sind(theta)/lambda); % manifold array for seeking peak  music = abs(diag(1./(doa\_a'\*(Un\*Un')\*doa\_a))); % the result of each theta  music = 10\*log10(music/max(music)); % normalize the result and convert it to dB  %% plot  figure;  plot(theta, music, 'linewidth', 2);  title('Music Algorithm For Doa', 'fontsize', 16);  xlabel('Theta(°)', 'fontsize', 16);  ylabel('Spatial Spectrum(dB)', 'fontsize', 16);  grid on;   It can be seen that when the 6 incident signals are uniformly coherent, the MUSIC algorithm based on forward smoothing can better estimate the DOA, but there are still estimation accuracy problems, such as the signal with a true incident angle of 75° The bearing is estimated to be 74.2°. **Backward spatial smoothing algorithm**  https://img-blog.csdnimg.cn/20201030120657822.png?#pic_center  Backward spatial smoothing is more accurately conjugate backward spatial smoothing, which is to smooth the covariance matrix of the conjugate received data of the backward sub-array. Define the first conjugate backward subarray {M,M−1,...,M−p+1} to be composed, and the second subarray to be composed of {M−1,M−2,...,M−p}, in turn The number of sub-arrays is L=M−p+1.  It is easy to know the relationship between the conjugate backward spatial smoothing covariance matrix and the forward spatial smoothing covariance matrix :    Using backward spatial smoothing covariance matrix and MUSIC algorithm can also distinguish the orientation of multiple coherent signals. It can be proved that the method can detect M/2 coherent signals at most.  This is the matlab simulation process carried out in our root data  clc; clear all; close all;  %% -------------------------initialization-------------------------  f = 500; % frequency  c = 1500; % speed sound  lambda = c/f; % wavelength  d = lambda/2; % array element spacing  M = 20; % number of array elements  N = 100; % number of snapshot  K = 6; % number of sources  L = 10; % number of subarray  L\_N = M-L+1; % number of array elements in each subarray  coef = [1; exp(1i\*pi/6);...  exp(1i\*pi/3); exp(1i\*pi/2);...  exp(2i\*pi/3); exp(1i\*2\*pi)]; % coherence coefficient, K\*1  doa\_phi = [-30, 0, 20, 40, 60, 75]; % direction of arrivals  %% generate signal  dd = (0:M-1)'\*d; % distance between array elements and reference element  A = exp(-1i\*2\*pi\*dd\*sind(doa\_phi)/lambda); % manifold array, M\*K  S = sqrt(2)\(randn(1,N)+1i\*randn(1,N)); % vector of random signal, 1\*N  X = A\*(coef\*S); % received data without noise, M\*N  X = awgn(X,10,'measured'); % received data with SNR 10dB  %% reconstruct convariance matrix  %% calculate the covariance matrix of received data and do eigenvalue decomposition  Rxx = X\*X'/N; % origin covariance matrix  H = fliplr(eye(M)); % transpose matrix  Rxxb = H\*(conj(Rxx))\*H;  Rf = zeros(L\_N, L\_N); % reconstructed covariance matrix  for i = 1:L  Rf = Rf+Rxxb(i:i+L\_N-1,i:i+L\_N-1);  end  Rf = Rf/L;  [U,V] = eig(Rf); % eigenvalue decomposition  V = diag(V); % vectorize eigenvalue matrix  [V,idx] = sort(V,'descend'); % sort the eigenvalues in descending order  U = U(:,idx); % reset the eigenvector  P = sum(V); % power of received data  P\_cum = cumsum(V); % cumsum of V  %% define the noise space  J = find(P\_cum/P>=0.95); % or the coefficient is 0.9  J = J(1); % number of principal component  Un = U(:,J+1:end);  %% music for doa; seek the peek  dd1 = (0:L\_N-1)'\*d;  theta = -90:0.1:90; % steer theta  doa\_a = exp(-1i\*2\*pi\*dd1\*sind(theta)/lambda); % manifold array for seeking peak  music = abs(diag(1./(doa\_a'\*(Un\*Un')\*doa\_a))); % the result of each theta  music = 10\*log10(music/max(music)); % normalize the result and convert it to dB  %% plot  figure;  plot(theta, music, 'linewidth', 2);  title('Music Algorithm For Doa', 'fontsize', 16);  xlabel('Theta(°)', 'fontsize', 16);  ylabel('Spatial Spectrum(dB)', 'fontsize', 16);  grid on;    It can be seen that when the six incident signals are uniformly coherent, the MUSIC algorithm based on backward spatial smoothing can better estimate its DOA, and the estimation accuracy is higher.  **Forward/backward spatial smoothing algorithm**  The forward and conjugate backward spatial smoothing covariance matrix are defined as the average of the forward spatial smoothing covariance matrix and the conjugate backward spatial smoothing covariance matrix, namely:    So as long as the number of spatial smoothing is greater than or equal to the number of coherent signal sources, the forward and conjugate backward spatial smoothing covariance matrices are generally full-rank. The maximum number of coherent signal sources that can be detected using the forward/backward spatial smoothing method is 2M/3. You may be curious how this maximum number of coherent signal source detections is obtained?  Assuming: The number of array elements of the array antenna is M, and the number of forward/backward spatial smoothing is L times respectively. Then the number of elements of each subarray is N=M−L+1. At the same time, it can be known that the maximum resolution is The number of signals is M−L, that is, the number of elements of the subarray minus 1; the number of signals that can be resolved by smoothing N times in the forward and backward directions is 2L. In the maximum case, the two are equal, so M−L= 2L, that is, L=M/3; Therefore, 2L=2M/3, so the maximum number of signals that can be resolved in the forward/backward spatial smoothing is 2M/3. Therefore, the forward/backward spatial smoothing improvement technology can greatly increase the array aperture.  This is the matlab simulation process carried out in our root data  clc; clear all; close all;  %% -------------------------initialization-------------------------  f = 500; % frequency  c = 1500; % speed sound  lambda = c/f; % wavelength  d = lambda/2; % array element spacing  M = 20; % number of array elements  N = 100; % number of snapshot  K = 6; % number of sources  L = 10; % number of subarray  L\_N = M-L+1; % number of array elements in each subarray  coef = [1; exp(1i\*pi/6);...  exp(1i\*pi/3); exp(1i\*pi/2);...  exp(2i\*pi/3); exp(1i\*2\*pi)]; % coherence coefficient, K\*1  doa\_phi = [-30, 0, 20, 40, 60, 75]; % direction of arrivals  %% generate signal  dd = (0:M-1)'\*d; % distance between array elements and reference element  A = exp(-1i\*2\*pi\*dd\*sind(doa\_phi)/lambda); % manifold array, M\*K  S = sqrt(2)\(randn(1,N)+1i\*randn(1,N)); % vector of random signal, 1\*N  X = A\*(coef\*S); % received data without noise, M\*N  X = awgn(X,10,'measured'); % received data with SNR 10dB  %% reconstruct convariance matrix  %% calculate the covariance matrix of received data and do eigenvalue decomposition  Rxx = X\*X'/N; % origin covariance matrix  H = fliplr(eye(M)); % transpose matrix  Rxxb = H\*(conj(Rxx))\*H;  Rxxfb = (Rxx+Rxxb)/2;  Rf = zeros(L\_N, L\_N); % reconstructed covariance matrix  for i = 1:L  Rf = Rf+Rxxfb(i:i+L\_N-1,i:i+L\_N-1);  end  Rf = Rf/L;  [U,V] = eig(Rf); % eigenvalue decomposition  V = diag(V); % vectorize eigenvalue matrix  [V,idx] = sort(V,'descend'); % sort the eigenvalues in descending order  U = U(:,idx); % reset the eigenvector  P = sum(V); % power of received data  P\_cum = cumsum(V); % cumsum of V  %% define the noise space  J = find(P\_cum/P>=0.95); % or the coefficient is 0.9  J = J(1); % number of principal component  Un = U(:,J+1:end);  %% music for doa; seek the peek  dd1 = (0:L\_N-1)'\*d;  theta = -90:0.1:90; % steer theta  doa\_a = exp(-1i\*2\*pi\*dd1\*sind(theta)/lambda); % manifold array for seeking peak  music = abs(diag(1./(doa\_a'\*(Un\*Un')\*doa\_a))); % the result of each theta  music = 10\*log10(music/max(music)); % normalize the result and convert it to dB  %% plot  figure;  plot(theta, music, 'linewidth', 2);  title('Music Algorithm For Doa', 'fontsize', 16);  xlabel('Theta(°)', 'fontsize', 16);  ylabel('Spatial Spectrum(dB)', 'fontsize', 16);  grid on;    Because the improved technology of forward/backward spatial smoothing greatly increases the array aperture, it can be seen from the above DOA results that the resolution has been improved.  **ESPRIT(Estimating Signal Parameters Via Rotational Invariance Techniques)**  The received signal is subjected to spatial Fourier transform (the difference between spatial Fourier transform and discrete-time Fourier transform is that the sum of the spatial Fourier transform is the space position m of the array element, while the time-domain Fourier transform is calculated The sum variable is discrete time n), and then the square of the modulus is taken to obtain the spatial spectrum, and the arrival direction of the signal is estimated (the phase φ corresponding to the maximum value of the spatial spectrum, and then according to the definition φ=2πdsinθ/λ, calculate θ).  Step 1  Calculate autocorrelation , apply eigenvalues decomposition to obtain eigenvectors  ***[V,D] = eig(A) produces a diagonal matrix D of eigenvalues and a full matrix V whose columns are the corresponding eigenvectors so that A\*V = V\*D.***  Step 2  Construct matrix and , they are the first M-1 columns and last M-1 columns of respectively.  Step 3  Calculate the eigenvalues of  Step 4  Calculate the  ***angle(H) returns the phase angles, in radians, of a matrix with complex elements.***  **Three algorithms compare the simulation process**  This is the matlab simulation process carried out in our root data  clc,clear all,close all  %% 产生信号样本  N=100;M=10;%信号样本数目和阵元个数  K=2;%信源个数  theta=[-10;40]\*pi/180;  SNR=[10;20];sigma=1;  Am=sqrt(2\*sigma^2\*10.^(SNR/10));  % Am=[sqrt(10.^(SNR/10))];  S=Am\*ones(1,N);  S(2,:)=S(2,:).\*exp(1i\*2\*pi\*rand(1,N));  for a=1:M  for b=1:K  A(a,b)=exp(-1i\*(a-1)\*pi\*sin(theta(b)));%第 b 列对应的都是 theta(b)  end  end  V=zeros(M,N);  for m=1:M  v=wgn(1,N,0,'complex');  v=v-mean(v);  v=v/std(v);  V(m,:)=v;  end  X=A\*S+V;  %% 利用接受数据估计信号的空间相关矩阵 R  R=zeros(M,M);  for i=1:N  R=R+X(:,i)\*X(:,i)';  end  R=R/N;%是一个统计平均  %MUSIC 算法  [VR,D]=eig(R);  D=real(D);  [B,IX]=sort(diag(D));  G=VR(:,IX(M-K:-1:1));  MUSICP=[];  for n=-pi/2:pi/180:pi/2  a=exp(-1i\*[0:M-1]'\*pi\*sin(n));  MUSICP=[MUSICP,1/(a'\*G\*G'\*a)];  MUSICP=real(MUSICP);end  n=length(MUSICP);  maxx=max(MUSICP);  figure,plot(-90:1:90,10\*log10((MUSICP+eps)/maxx)+3.5),axis([-90,90,-  60,inf]),title('MUSIC 算法')  %RootMUSIC 算法  syms z  pz=z.^([0:M-1]');  pz1=(z^(-1)).^([0:M-1]);  fz=z^(M-1)\*pz1\*G\*G'\*pz;  a=sym2poly(fz);  r=roots(a);  r1=abs(r);  for i=1:2\*K %每个信号源有 K 个  [Y,I(i)]=min(abs(r1-1));  r1(I(i))=inf;  end  for i=1:2\*K  theta\_esti(i)=asin(-angle(r(I(i)))/pi)\*180/pi;  end  %ESPRIT 算法  S=VR(:,IX(M:-1:M-K+1));  S1=S(1:M-1,:);  S2=S(2:M,:);  fai=S1\S2;  [U\_fai,V\_fai]=eig(fai);  for i=1:K  ESPRITtheta\_esti(i)=asin(-angle(V\_fai(i,i))/pi)\*180/pi;  end  %MVDR 算法  MVDRP=[];  for n=-pi/2:pi/180:pi/2  a=exp(-1i\*[0:M-1]'\*pi\*sin(n));  MVDRP=[MVDRP,1/(a'\*inv(R)\*a)];  end  n=length(MVDRP);  maxx=max(MVDRP);  figure,plot(-90:1:90,10\*log10((MVDRP+eps)/maxx)+3.5),axis([-90,90,-  35,inf]),title('MVDR')  %F-SAPES 算法  P=6;%子阵数目L=M+1-P;%子阵阵元数目，书上是 M-1  Rf=zeros(L,L);  for i=1:P  Rf=Rf+X(i:i+L-1)\*X(i:i+L-1)'/N;  end  Rf=Rf/P; %子阵平滑后的空间相关矩阵  n1=0:P-1;  n2=0:L-1;  cc=[1 zeros(1,L-1)];  for n3=-90:.5:90  fy=exp(1i\*pi\*sin(n3/180\*pi));  tt=[(fy.^(n1')).' zeros(1,M-P)];  Tfy=toeplitz(cc,tt);  GfTheta=1./(P^2)\*Tfy\*R\*Tfy';  Qf=Rf-GfTheta;  aTheta=fy.^(-n2');  Wof=(Qf\aTheta)./(aTheta'\*(Qf\aTheta));  sigma2sTheta(((n3+90)/.5+1))=Wof'\*GfTheta\*Wof;  end  maxx=max(sigma2sTheta);  figure,plot(-90:.5:90,10\*log10((sigma2sTheta+eps)/maxx)+3.5),axis([-90,90,-  35,inf]),title('F-SAPES')      The three pictures from top to bottom are simulation images of the MUSIC algorithm, MVDR algorithm, and F-SAPES algorithm. Because it is a preliminary exploration of the algorithm, we have a certain understanding of the principles and operation process of the three algorithms, but there is no complete system for the analysis process of the effect of the three algorithms. We have simulated the results of the three algorithms. With a certain understanding, a certain analysis was carried out. However, our overall grasp of the three algorithms is not yet in place, there are still certain deficiencies in the construction of the knowledge system, and there may still be certain imperfections in the principle analysis. Therefore, our analysis of the three algorithms will not be presented in the report. We will focus on this aspect and comprehensively improve it in subsequent experiments and reports. | |
|  | |
| **Experience**  The above is all the contents of our interim report. In the process of completing the interim report, we mainly carried out the preliminary preparation and research work of the project. In the process of completing these tasks, we mainly conduct all the pre-preparation work by consulting data.  In order to make our interim report more organized and to better guide our subsequent project work, we divided the preliminary preparations into two parts. The first part is the process of reviewing papers. We have reviewed more than five related papers that are both general and informative, and first have a comprehensive understanding of the entire project process. To guide the follow-up study by systematically learning the introduction of experimental knowledge is what Mr. Wu repeatedly emphasized in the experiment, and it is also what we need to follow in each experiment.  The second part is the process of systematic learning. In this process, we systematically compared the content of the paper, and learned the coding process of the algorithm with the help of CSDN tools, and carried out a systematic comparison and comprehensive analysis of the three algorithms. In this process, we have a certain understanding of the operation process of the algorithm through the simulation algorithm in the data. A certain amount of learning has been carried out on the simulation process of the algorithm and the optimization of the algorithm. Analyze the algorithm through the results.  The third part is the process of our internalizing knowledge and re-learning. We internalize the knowledge through systematic learning, and then transform it into our own learning results. Most of the intellectual content in our interim report is edited and written by ourselves through this process. | |
| **Score** |  |

**references**

[1] Zhou Wei, Liu Yaoyu, Hu Qinghong, Deng Guanghai, Hu Li, Wu Junbiao, Lin Hua. Evaluation of the difference between Fuzitang decoction and formula granule by HPLC fingerprint spectrum combined with chemical pattern recognition[J/OL]. Chinese Journal of Hospital Pharmacy: 1- 8[2021-04-14].http://kns.cnki.net/kcms/detail/42.1204.R.20210413.0949.002.html.

[2]Yin Yuanwei, Liu Yuehang, Xie Hui, Liu Xing. Research on SystemVue Modeling and Simulation Method of Radar Signal Processing in Clutter Environment[J]. Computer Measurement and Control, 2020, 28(01): 141-146.

[3] Liu Zhiyun. Research on LFMCW radar signal processing and realization based on multi-channel [D]. University of Electronic Science and Technology of China, 2018.

[4]Liu Longfei. Design of multi-model radar signal processing system based on FPGA[D]. Dalian Maritime University, 2014.

[5] Li Shengkui, Liu Mingming, Pan Guangshan, Bi Zhengfeng. Application of ground penetrating radar method in leakage detection of special oily sewage pipeline[J]. Shandong Land and Resources, 2021, 37(04): 67-72.

[6] Li Shengkui, Liu Mingming, Pan Guangshan, Bi Zhengfeng. Application of ground penetrating radar method in leakage detection of special oily sewage pipeline[J]. Shandong Land and Resources, 2021, 37(04): 67-72.

[7] Fan Dianliang, Xing Gengli, Liao Hongjian, Mao Jin. On the construction of UAV public security prevention and control system[J]. China Security Technology and Application, 2021(01): 45-49.

[8] Zhou Yuang, Chen Yijun, Xie Wenxuan, Zhu Yongzhong. Overview of radar detection technology based on vortex electromagnetic waves [J]. Modern Radar, 2021, 43(02): 57-66.

[9]Ningbo Liu, Hao Ding, Yong Huang, Yunlong Dong, Wang Guoqing, Kai Dong. The annual progress of X-band radar sea detection experiment and data acquisition[J]. Journal of Radars, 2021, 10(01): 173-182.

[10]Chen X Gregory,Graužinytė Miglė,van der Vaart Aad W,Boll Björn. Applying Pattern Recognition as a Robust Approach for Silicone Oil Droplet Identification in Flow-Microscopy Images of Protein Formulations.[J]. Journal of pharmaceutical sciences,2021,110(4).

[11]Kordzadeh Ali,Hanif Mohammad A,Ramirez Manfred J,Railton Nicholas,Prionidis Ioannis,Browne Thomas,Lederle FA,Freischlag JA,Kyriakides TC,Nordon IM,Karthikesalingam A,Hinchliffe RJ,Greenhalgh RM,Brown LC,Powell JT,De Bruin JL,Baas AF,Buth J,Karthikesalingam A,Al Jundi W,Jackson D,Kret MR,Azarbal AF,Mitchell EL,Chisci E,Setacci F,Iacoponi F,Black SA,Carrell TW,Bell RE,Weerakkody RA,Walsh SR,Cousins C,Michaels JA,Drury D,Thomas SM.,Schanzer A,Greenberg RK,Hevelone N,Karthikesalingam A,Holt PJ,Vidal Diez A,Lo BWY,Macdonald RL,Baker A,Anagnostou T,Remzi M,Lykourinas M,Herman Saffar O,Boger Z,Kuo PJ,Wu SC,Chien PC,Wise ES,Hocking KM,Brophy CM.,Luebke T,Majd P,Mylonas SN,Karthikesalingam A,Attallah O,Ma X,Mcfarland G,Lee JT.,Zettervall SL,Schermerhorn ML,Soden PA,Riambau V,Blanco C,Jabalera EG.,Brady AR,Thompson SG,Fowkes FGR,Sidloff D,Stather P,Dattani N,Kordzadeh A,Esfahlani SS.,Hashimoto DA,Rosman G,Rus D. Prediction, pattern recognition and modelling of complications post-endovascular infra renal aneurysm repair by artificial intelligence[J]. Vascular,2021,29(2).

**CSDN**

<https://blog.csdn.net/qq_36583373/article/details/109364878>

<https://blog.csdn.net/pwang95/article/details/104999880?utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7Edefault-5.control&dist_request_id=1328767.60819.16176271490634367&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7Edefault-5.control>

<https://blog.csdn.net/ai136172022/article/details/80967796>

<https://blog.csdn.net/pwang95/article/details/92800159?spm=1001.2014.3001.5501>

<https://download.csdn.net/download/pwang95/14828091?spm=1001.2014.3001.5501>