**Project: Wireless Direction Finding System based on Kerberos SDR**

|  |  |
| --- | --- |
| **Author** | Name： 全组 Student ID: |
| **Introduction:**  In this project, we built an experimental platform using Kerberos SDR and Raspberry Pi to realize wireless direction finding and radar sensing system. The core of our experimental theory is Music algorithm and Ambiguity function. By Music algorithm, we can measure the angle of arrival between signal source and receiving antenna. By using Ambiguity function, we can get doppler frequency deviation of moving object by cross correlation operation. As a lot of work was done in the theoretical analysis of the project in the early stage, we did not encounter too many difficulties in the algorithm design and the presentation of experimental results. However, in the part of building the experimental platform, we spent a lot of time in configuring the driver of Kerberos SDR. Meanwhile, how to design the relative position of antenna array element and USRP is a problem we should think carefully. With the efforts of all team members, we overcame these problems and realized the requirements of this project.  **Task Allocation:**  **11913019 仇琨元: 编写MUSIC与Ambiguity Function核心程序 +提出异构计算方案 + 频谱仪分析实验现象 (合作)**  **11911303 吉辰卿：实验平台搭建 + Music算法基础部分设计分析 +频谱仪分析实验现象(合作)**  **11510473 赵青宇：Music算法提升部分 + 多普勒感知部分实验设计分析**  **11911528 邓煜：多普勒感知的理论 + 实验系统设计分析**  **11911118 吴沭豪：实验数据接收处理，生成结果图像**  **11910921 刘远卓：Music算法理论部分+DOA概念解读**  **Part 1: Basic Introduction**   * 1. **Music algorithm**   MUSIC algorithm is an important foundation of direction finding theory of spatial spectrum estimation. The specific algorithm principle is as follows:   1. Suppose there are N antennas and D signals, all of which can be represented by a matrix , and the signals received by the antenna are represented by a matrix .      1. The matrix X can be represented by the matrix S and the Angle matrix A:   Where is noise,   1. The theoretically received signal autocorrelation matrix can be written as:   H  H   1. We also obtain the autocorrelation matrix of the actual received signal:   The eigenvalues of 𝐑\_𝑋 can be expressed as:  This formula is very similar to the one we got in step 3. At this point, if we arrange the eigenvalues in the diagonal matrix from large to small, all the values will be roughly divided into two groups. And the one with the larger value has D remainder (n-d) and the one with the smaller value. The former represents the group D signal received, while the latter is noise. We take the eigenvectors corresponding to the eigenvalues of noise as the basis to form the noise space . If there is a signal at , then will be A very small value because they are orthogonal.   1. Let's define a function:   There is a spike when the signal is detected, then we get the angle we want.  **Doppler Sensing**  The proactive sensing is that the BS has to generate the detection signal, and analyze the influence of the measured object on the detection signal to achieve the purpose of perception. So the BS has to both transmit and receive data Signals, which means it is full duplex mode. This mode is difficult to realize. So in our project we choose the other mode-passive sensing.    In this project we choose the passive sensing mode. In this system, the BS send data signal and the passive receiver is to receive the signal from surveillance channel and reference channel. Analyze the signals from two channel we can obtain the doppler frequency and velocity of moving object. There are 3 elements in our system-USRP, KerberosSDR and moving people. The USRP serves as signal transmitter, and the KerberosSRD serves as the passive receiver.    Through the surveillance channel and reference channel we obtain 2 signals:  Here we use different antenna to get signals from the 2 channels. we divide the receiving antenna into two parts: Reference and Surveillance. There is one antenna in Reference and three antennas in Surveillance. They receive the data read by Reference Channel and Surveillance Channel respectively, and the data of these two channels are irrelevant.  Define:  Then we use Ambiguity Function to do the estimation:      θc：estimated via the phased array of the passive receiver, e.g., MUSIC algorithm.  s: estimated with and .  : estimated with , θ and s, .  : with the direction of car’s velocity and f.  To calculate the : , where fc is carrier frequency, f is the Doppler frequency offset.  In the surveillance we can obtain the f1 and f2 ,v1 and v2.Then use the composition of vectors we can finally estimate the velocity of moving object.    **Part 2: Program Design**  **2.1 Music algorithm**  Since the mathematical principle of the MUSIC estimation is given above as  Where the are defined as  The MATLAB code for the MUSIC algorithm is simply the direct translation of the matrix expressions above. To simplify the coding and push the efficiency to the maximum, the MUSIC function in this project only adapts to the uniform linear array(ULA).    The argument angs is an array containing all the angles to be scanned, and the argument antarg is a double number that represents the minimal phase difference between the two adjacent antennas. In the guarantee of the positive definiteness of the covariance matrix, the matrix adds to its transpose before the EVD.  The loops are very inefficient in MATLAB compared to the frightening efficiency of its vectorized computation. Therefore, when composing MUSIC routines involving large dense matrix decompositions and products, any loops lead to unacceptable execution lag and excessive CPU heat load.    These two segments from two different implementations of MUSIC algorithm are typical wrong example. The for loop shown in the segments causes high CPU load and time cost.  To make the maximum use of the computation power of the i7-9750HQ CPU on my laptop, the generation of the steering matrix is embedded into the code, rather than passing a pre-built matrix through a pointer. After the MATLAB Coder trans-compilation, the efficiency of the embedded steering matrix is significantly higher than passing a huge pre-built matrix through a pointer or memcpy.    Testing result of the MUSIC subroutine using two different DOA sets is shown above. It’s easy to recognize in the picture that the spatial spectrum produced by the MUSIC subroutine is very sharp. However, the sharpness of the spectrum is reduced when the difference between the DOAs is lower than 20°.  **2.2 Ambiguity function**  **Ambiguity function is the correlation of two sequences**  So the Ambiguity Function can be calculated much faster in the frequency domain with multiple FFTs and one IFFT operation, rather than the stupid and inefficient for loops.    There has already been high performance implementation of FFT and IFFT such as FFTW and OpenBLAS package, thus it is significantly faster to evaluate the ambiguity function in the frequency domain, as the procedures shown below.  Firstly, the frequency domain expression of the two inputs is obtained using FFT.  For one single delay, the Doppler frequency shift is scanned through FFT.  Then use the vector to correlate with each column of the frequency scanning matrix :  The column-by-column correlation scans through all possible delay.  Finally, after rearranging the Ambiguity Function obtained by the butterfly rule of FFT and IFFT, the positive frequency part is taken by using the fftshift function.  A=ifftshift(ifft(F\_A(theta,f\_d)))  Moreover, the Phased Array System Toolbox of MATLAB provides a built-in function ambgfun() to calculate the Ambiguity Function.    After analyzing the code, it is discovered that this toolbox function computes the Ambiguity Function by the frequency domain correlation method.  **Part 3: Experimental Platform**  Our experimental platform is mainly composed of USRP, Kerberos SDR, Raspberry Pi and PC in hardware. Firstly, LTE signals is transmitted by USRP and then received by quad-channel antenna array of Kerberos SDR after transmission through wireless channel. After that, the data received on the four antenna array elements are read on the raspberry Pi, which is pre-configured with Kerberos SDR driver, and then converted into .csv file. Then, the data received by Raspberry Pi is transmitted to PC through VNC, and it is merged into a .mat file and handed to MATLAB for algorithm processing. MATLAB processes the received data of four-channel antenna array elements and calculates DOA by running Music algorithm. Finally, we draws relevant radiation graphs to show the results of our measurement. The specific system block diagram is as follows:  C:/Users/Administrator/AppData/Local/Temp/wpp.CNuMSfwpp  Figure. Flow chart of building experimental platform  Next, we set up the platform for experiments according to each step of the above process, as shown below:  NH8U~10]GTDB%O[0CRF_M(W  **接收天线阵元**  Figure. The relative position of the USRP transmitter and the KerberosSDR receiving antenna array element  4]`~ME3CHUH)FJ_JC%L38VV  **发射机程序我们选用了Lab6 Part3的LTETx.m**  Figure. USRP transmitter placement  (We put the computer under the desk to eliminate the interference)  J(TDUD@5PMXH_4W[9S53C2J 20220530222215  Figure. KerberosSDR Driver configuration on Raspberry Pi  (The picture on the right shows successful reading of antenna data on raspberry Pi)  IMG_256  Figure.The Raspberry Pi captures the signals emitted by the USRP and generates waveforms corresponding to the four antennas  Next, we show some pictures from our experiment:  XP7EX~KZGC`(`BG6)9XGUS7 LS(CH~{MKV~5{JO8(F~9BR2  Figure. The optimization scheme we designed for DOA measurement  (Left: Add a baffle Right:Place the antenna array element is placed in the center of the table) | |
| 20220531153136  Figure. The deep optimization scheme we designed for DOA measurement  ( Place the antenna array element in the center of the table and a baffle is placed behind it)  993B9602241ED1BBCCBDB068981B4CBE  **受试者朝着Surveillance Channel Antenna运动**  The arrangement of four antenna array elements in doppler radar perception experiment  (Left: Reference Channel Antenna Right: Surveillance Channel Antenna)  **Part 4: Experiment Result**  **4.1 Data processing**  First, We use the Raspberry Pi to capture the antenna signal transmitted by the USRP. Then we save the data to the csv file.  IMG_256  Second, we use VNC to transfer the experimental data to the computer.    find the transfer files module    select the file we want to transfer    save the file to your computer  Third, we use matlab program to merge the received csv data      Fourth, we turn csv data to mat data      Finally, we load the mat data to the matlab and then we can anslysis the data further.  **4.2 First try for DOA Estimation**  In our first attempt at DOA estimation, we placed the receiving antenna array on 506 next to my desk and ensured that the USRP and the receiving antenna array had an angle of about 30 degrees. According to the theory of MUSIC algorithm, we can estimate the arrival Angle of DOA to be about 30 degrees under ideal conditions. The experimental results are as follows:  **(To make our results less intrusive, we filter out angles greater than or less than -90 degrees)**  W_F){4}EVQ@ZI`P6`RDJ`RM ~$A$J81CR(T)TVRT`NK@WEA  Figure.DOA estimation of first experiment results  From the above results, we can see that our DOA estimation results are messy and produce peak values at multiple angles. After our discussion, we found that since the signal transmitted by USRP is oriented in all directions, if the reflector is not added, the received signal is oriented in all directions. Maybe some signals in the opposite direction are cancelled, but more signals still enter the receiver. This is why our DOA estimation result is messy.  **4.3 Second try for DOA Estimation**  To solve the problem in the above experiment, we add a reflector and make the included angle between the signal reflected by the reflector and the antenna array element approximately 30 degrees. As follows:  20220531165251  Figure. Second-try experiment by adding a reflector  Let's take a look at the results of this improved experiment:  **(In order to remove more interference from our results, we filter out angles greater than 90 degrees or less than -90 degrees)**  ]GI%GQ1V6)R4LFVOK04XNYN 20220531165558  Figure.DOA estimation of second experiment results  From the estimation results of this DOA, we can see that the number of peak angle detected is less than the first experiment and there is a strong peak near 30 degrees, which shows that we can indeed reflect the angle to 30 degrees in certain range and be detected by the KerberosSDR antenna array element by adding a baffle. The range of the peak angle detected this time is larger than the first experiment, it reflects that our baffle does have a certain reflection effect, which can greatly increase the angle of some signals reflected by the baffle to reach the receiving antenna array element. While this improvement is a bit of an improvement for less number of peak angle detected than the first experiment, there is still some interference from certain angles. Therefore, we will try another improvement next.  **4.4 Third improvement for DOA Estimation**  In the previous experiments, we have obtained the reflected echoes in multiple directions in the results, and there are often clutter interference. In practical applications, the radar direction finding system needs to accurately measure the reflected wave of a single target. So we imagine how to simulate this result in a laboratory environment. After the observation of the previous experiment, we found that the laboratory wall and the high seats around would have an impact on the results. According to the basic principle of MUSIC, when a group of waves enters the receiving antenna, and there is another group of waves entering in the opposite direction, the original beam received by the antenna will be offset, so that the phase perception of the received data is 0. So we put a board in the center of the laboratory as an obstacle, and put the receiver on the edge of the board to ensure that the distance between the receiver and the laboratory is symmetrical. At this time, the obstacles will cause spatial asymmetry, so that the receiver can feel the reflected wave in one direction.    Figure. Transmitter location    Figure. Location of receiver and obstacles  After the experiment, we found that the interference of the surrounding echo was obviously removed, and only the angle of the target obstacle was measured.    Figure.DOA estimation of third experiment results  **4.5 Speed Test**  In the Doppler velocity test, we changed the experimental environment. For the Doppler frequency offset test, there are two channels, reference and surveillance, are required. However, we cannot specify which in the receiving antenna is responsible for receiving the reference signal of USRP and which is responsible for receiving the surveillance signal reflected by the human body. So we must separate them physically, and we designed the following experimental environment.    Figure. Readjusted experimental environment  The environmental distribution is shown in the figure below. In the experimental, our sampling rate is 20MHz, and 20M data is sampled within 1 second. The data received by the three antennas act as the data of the surveillance channel, and the data received by the last antenna is the data of the reference channel.    Figure. Specific position distribution in the experiment  The test results are shown in the figure below. In the DOA test, we only felt the position of the subject. In the Doppler frequency offset test, we tested the results when there were a large number of laboratories and when there was no external interference. Because the data obtained from the actual test is large, in order to obtain this result, we only analyzed the first 1/20 of the data, with a total of 4M data. When there is no interference in the laboratory environment, it is obvious that the subjects have a frequency offset about 18Hz.    Figure. DOA results during personnel movement    Figure. The laboratory is densely staffed compared with a single laboratory environment  In order to calculate the specific moving speed of the subjects, we calculated it through the formula. In the formula, C is the speed of light , F is the Doppler frequency offset 18Hz, and Fc is the carrier frequency. At the beginning, we set 2.2GHz in the program in the laboratory. However, after the teacher's correction, we found that the frequency range of Kerberos SDR is form 24 MHz to 1766 Mhz. We carried out relevant verification. Through the detection of the spectrometer, we set the transmitter frequency as 2.2GHz in the program, but the transmitter program part we modified is actually the center frequency of the simulation part, and the actual transmission frequency is 947.1Mhz. Because negligence we did not modify it, the experiment happened to be successful. After calculation, the moving speed of the subject is 2.85m/s, which is consistent with our actual observation.    Figure. Relationship between velocity and Doppler frequency offset  **Experience**  **11911303 吉辰卿：**  **这次实验在平台搭建上不算太复杂。然而遗憾的地方是我们使用树莓派读取SDR四路天线阵元的数据时采用了for循环依次读取，而更严谨的做法是使用python的并行计算来读取数据，这样在测试的时候由于没有时延会更加准确。我们虽然在这块查阅了资料并进行尝试，但由于时间关系，我们暂时还没能很好地这个问题，这是我们在未来可以突破改进的一个点。**  **同时，在我们小组pre的最后，吴老师问了我们一个很有意思的问题：既然你们USRP发射LTE信号的载波频率为2.2GHz，但Kerberos SDR能接收到的信号最大频率只有1.8GHz。那么，你们的实验怎么能够正常完成的呢？对于这个问题，我私下联系王锐老师课题组并向其借用了频谱仪进行测量。测量结果如下所示：**  **我们首先将配置仍然和之前一样设成2.2GHz，如下所示：**  **20220608213141**  **结果如下所示：**  20220608213425  **频谱仪显示发射的中心频率仅有947.1MHz，那么为什么会这样呢？我们仔细检查了发射机的程序，果然出现了问题：**  **20220608214056**  **原来我们刚刚只修改了txsim的仿真参数，并将其作为构建txWaveform的依据。而我们发射的txWaveform其实是加载到中心频率为900MHz的载波上的。因此，出现这个问题的原因本质是我们错误理解了代码，txsim设置发射频率作用不到USRP配置上，实际发射出去的LTE信号中心频率应在900MHz附近。当我们将这段代码的中心频率参数再次修改到2.2GHz的时候，频谱仪显示如下：**  **20220608214642**  **很明显，在2.2GHz处频谱仪出现了一个小峰值，然而在2.4GHz左右出现了一个更大的峰值，这很显然就是WIFI的信号。至此，我们就能够完全回答吴老师在pre时提出的问题了。最后感谢吴老师从大二上学期一直陪伴我们走到现在，****每一次实验课在吴老师的指导下我们总能从小组项目中收获颇多。对此，我想说：吴老师，您辛苦了！**  **11510473 赵青宇：**  **在进行多普勒测试时，因为测试的数据量较大，直接运行多普勒测试代码消耗时间较长，并且得到的多普勒结果是各个时间段的叠加，于是我们只选取了其中一部分的数据做多普勒测试。没能得到运动物体的实时数据信息。**  **在整个测试的过程中，因为我负责实验工作的实行，在得到实验结果后没有仔细检查实验中会出现的细节问题。导致我们组出现了理论上完全站不住脚的实验结果，虽然经过后期检查发现是因为程序运用的问题，实验上没有太多问题。但是这也给我们敲响了警钟，在进行具体的科学实验时，并不能因为得到了看似可靠的实验结果，就不在去检查实验中的细节问题。这次实验犯下的错误给我们上了深刻的一课，最后在这里很感谢吴光老师的栽培，陪伴了通信专业几乎所有的实验课程，感谢吴光老师的付出！**  **11911528 邓煜：**  **这个实验系统我们根据王锐老师上课讲到的感知系统进行搭建，使用了MUSIC算法和模糊函数进行数据处理和分析。同时我们也使用到了之前实验课学到的一些知识，例如对USRP发射的控制。实际的硬件实验中遇到了不少困难，同时软件程序的设计也有改进的空间（例如并行读取数据），一开始时实际上我们还对这个系统中的大部分环节比较陌生，在仇同学的讲解以及大家的互相帮助探索中，逐渐完成了整个系统的搭建和运行。在这个克服困难的过程中我们对于USRP，树莓派，KerberosSDR，以及多普勒感知的方法都有了更加深刻的认识，而这个从理论到实践的过程也非常令人欣喜。感谢实验同组同学在实验中的耐心与坚持，感谢吴老师的教导。** 11913019仇琨元: **本次实验的代码编写相对来说比较简单,比较花时间的地方主要是运用MATLAB向量化编程的技巧优化MUSIC算法和Ambiguity Function函数的运行性能.另外由于Windows端难以正常安装Kerberos SDR所需的开源驱动,因此我提出了树莓派Linux系统采集Kerberos SDR数据进行简单处理后,通过网络传输给Windows端计算的异构计算方案.由于我直到报告时都在隔离,因此无法进行实地调试,最终树莓派不仅无法计算协方差,在通过局域网回传数据时也需要手动传输CSV文件,而效率更高实时性更强的做法是使用python的threading库给每个SDR并发一个线程读取数据，在树莓派上计算协方差矩阵、从原始矩阵的4096个双精度数压缩为16个双精度数后,通过UDP协议或者WebSocket协议直接传输到Windows电脑.**    **鉴于MATLAB能直接生成DLL文件,然而缺乏WebSocket通信的相关支持包,python的NumPy库速度上不如MATLAB,然而python网络编程非常容易,因此接收树莓派的WebSocket数据、并进行高速计算有两种可行的方法.第一是用C++语言手工编写一个接收WebSocket数据的程序源码,然后在MATLAB中编译生成相应的MEX文件,实时接收来自树莓派的数据;第二则是用MATLAB Coder依据已经做好的M代码生成高速计算MUSIC算法和Ambiguity Function的程序源码,在C++ IDE中进行修改调试后编译得到相应的DLL文件,然后用python的cython库调用DLL实现高速计算.**  **最后,感谢小组同学们共同的坚持和对我隔离期间无法回校参与工作的体谅,没有同学们在校内辛苦的调试开发,我提出的方案也无法成为现实.在此更要感谢吴老师从大二开始陪伴我走过通信工程专业的绝大多数重要课程.我的自学能力、读帮助文档的习惯和独立自主解决问题的决心基本都来自于完成吴老师的实验课和项目. 对此，我想说：吴老师，您辛苦了！**  **11911118 吴沭豪：**  **在本次实验中，我负责的是数据处理。一开始数据的转换、数据的格式不匹配、数据的行列与代码不符等一系列问题让我遇到了些许麻烦。经过不断的尝试，我才得以解决这些问题。同时，因为实验中采集的数据量较为的庞大，在处理数据这一方面需要花费大量的时间，这也让我们无法第一时间去检测实验结果，偶尔因为实验人员的干扰或者实验环境搭建偏差的导致实验误差，我们也只能在繁琐的数据处理之后才能发现，因此有时我们不得不进行多次实验来得到一个存在较小实验误差的结果。**  **最后，感谢小组同学们的耐心与坚持，让我们能够在短短一周的时间内完成这次的实验。也感谢吴光老师的一直以来的栽培，从大二到现在，每一次实验课程在吴光老师的指导下我们总能有颇多的收获，很感谢吴光老师两年来的陪伴和付出。**  **11910291 刘远卓：**  **在本次实验中我负责Music算法的理论分析和初步实现进一步巩固了对于理论知识的掌握，同时也辅助小组其它同学完成了环境的搭建和硬件调式。在完成实验汇报之后，对于老师提到问题我们也是多方寻求资料寻找答案。在这个过程中包括我在内的小组成员对常见的无线电设备（手机、电脑）等的工作频率有了一个更加全面的了解。同时也通过频谱仪的分析更加清晰地认识到了USRP的工作原理，在原有的实验之外又有了新的收获。感谢小组成员的付出，更要感谢吴老师教导能够让我们对于实验的仪器的细节有了更加深刻地了解。** | |
| **Score** | 自评得分：100 |