

# 媒体信号处理基础-实验报告 2

学号： 3150105267

姓名：卢雨洁

## 1、 实验内容及要求

实验工具：MATLAB 2018a

实验内容：

1. 用 matlab 语言编码实现长度为 8 的一维离散余弦变换 ( **不能使用 matlab 内嵌的 dct()函数** );
2. 应用 matlab 内嵌的 dct()函数验证自己所实现代码的正确性；
3. 对 1D 随机信号利用**上面实现的离散余弦变换**对信号进行变换，然后做反变换，对比恢复后的信号与原信号之间的差异；
4. 随机生成包含一定高斯噪声的正弦随机信号，然后利用 DCT 变换对信号进行去噪处理。

## 2、 关键代码及注释

STEP1 关键代码与注释

解决问题如下

- 1、 用 matlab 语言编码实现长度为 8 的一维离散余弦变换 ( **不能使用 matlab 内嵌的 dct()函数** );
- 2、 应用 matlab 内嵌的 dct()函数验证自己所实现代码的正确性；

function result = my\_idct(N)%一维离散余弦反变换

length = N%长度为 N

my\_dct = DLab2\_dct(length)%通过已实现的变换函数得到 DCT 变换序列

reverse\_dct = idct(my\_dct)%利用 matlab 自带反变换函数验证

for i = 0:length-1%遍历计算每一项

if i == 0%根据公示计算每一项的 cu 系数

cu = sqrt(1/length);

else

cu = sqrt(2/length);

end

sum = 0;%初始各项和为 0

for j = 0:length-1

sum = sum + my\_dct(j + 1) \* cos(pi \* (j + 0.5) \* i / length);%

各项和按照公式计算

end

my\_rdct(i + 1) = cu \* sum;%记录每一项的值

end

## STEP2 关键代码与注释

解决问题如下

- 3、对 1D 随机信号利用上面实现的离散余弦变换对信号进行变换，  
然后做反变换，对比恢复后的信号与原信号之间的差异；

```
function result = my_dct(N)%一维离散余弦变换，长度为 N
seq = rand(1, N);%长度为 N 的序列 in
length = N;%长度为 N
seq%显示 seq 的内容
for i = 0:length-1 %遍历计算 cu 系数，使得 DCT 变换矩阵成为正交矩阵
    if i == 0 %根据公示设置 cu 系数的值
        cu = sqrt(1/length);
    else
        cu = sqrt(2/length);
    end
    sum = 0;%初始为 0
    for j = 0:length-1%进行累加
        sum = sum + seq(j + 1) * cos(pi * (j + 0.5) * i / length);%按照公式计算各项和
    end
    result(i + 1) = cu * sum;%计算变换后序列的每一项的值
end
result%输出 my_dct 的 DCT 变换结果与 verify_result 作比较
verify_result = dct(seq)%利用 matlab 自带的 dct 函数对 seq 序列作变换
```

### STEP3

解决问题如下

- 4、随机生成包含一定高斯噪声的正弦随机信号，然后利用 DCT 变换对信号进行去噪处理。

```
x = (1:100) + 50 * cos((1:100) * 2 * pi / 40);%长度为 100 的正弦信号
```

```
n = 5 * rand(1, 100);%高斯伪随机信号噪声
```

```
plot(x, 'r')%红颜色线为原信号
```

```
hold on
```

```
x = x + n;%随机生成包含一定高斯噪声的正弦随机信号
```

```
plot(x, 'g')%绿颜色线为带噪声的信号
```

```
plot(x)
```

```
x_dct = dct(x);%DCT 变换
```

```
x_dct(21:100) = 0;%DCT 去噪
```

```
xx = idct(x_dct);%DCT 反变换
```

```
plot(xx, 'b')%蓝颜色线为利用 DCT 去噪的信号
```

```
hold off
```

## 5、 实验结果及分析

结果一：

一维 DCT 变换与 dct()函数验证长度为 8 的序列

```
>> DLab2_dct(8)

seq =

     1     2     3     4     5     6     7     8

result =

    12.7279    -6.4423    -0.0000    -0.6735         0    -0.2009    -0.0000    -0.0507

verify_result =

    12.7279    -6.4423         0    -0.6735         0    -0.2009         0    -0.0507

ans =

    12.7279    -6.4423    -0.0000    -0.6735         0    -0.2009    -0.0000    -0.0507

|
```

seq：长度为 8 的序列

result：实现的算法 DCT 变换的结果

verify\_result：利用 matlab 自带函数验证 seq 变换结果与实现的算法结果相同

结果二:

长度为 8 时的随机序列反变换与原信号的比较

```
seq =  
    0.8499    0.6835    0.0045    0.9196    0.3823    0.7013    0.2880    0.2193  
  
result =  
    1.4313    0.3325   -0.0566    0.4161    0.2453    0.1366   -0.1673   -0.6016  
  
verify_result =  
    1.4313    0.3325   -0.0566    0.4161    0.2453    0.1366   -0.1673   -0.6016  
  
my_dct =  
    1.4313    0.3325   -0.0566    0.4161    0.2453    0.1366   -0.1673   -0.6016  
  
reverse_dct =  
    0.8499    0.6835    0.0045    0.9196    0.3823    0.7013    0.2880    0.2193  
  
my_rdct =  
    0.6139    1.1677    0.0941    0.8437    0.4405    0.3718   -0.0072   -0.1046  
  
error =  
   -0.2360    0.4842    0.0896   -0.0759    0.0582   -0.3296   -0.2952   -0.3239
```

seq: 随机长度 1D 随机信号

result/my\_dct: 利用实现的离散余弦变换对信号进行变换的结果

verify\_result: dct()函数验证信号结果

reverse\_dct: 原信号

my\_rdct: 反变换

error: 恢复后的信号与原信号之间的差异

## 长度为 5 时的随机序列反变换与原信号的比较

seq =

0.5483	0.8343	0.0898	0.0054	0.7825
--------	--------	--------	--------	--------

result =

1.0109	0.1673	0.4600	-0.5857	-0.1128
--------	--------	--------	---------	---------

verify\_result =

1.0109	0.1673	0.4600	-0.5857	-0.1128
--------	--------	--------	---------	---------

my\_dct =

1.0109	0.1673	0.4600	-0.5857	-0.1128
--------	--------	--------	---------	---------

reverse\_dct =

0.5483	0.8343	0.0898	0.0054	0.7825
--------	--------	--------	--------	--------

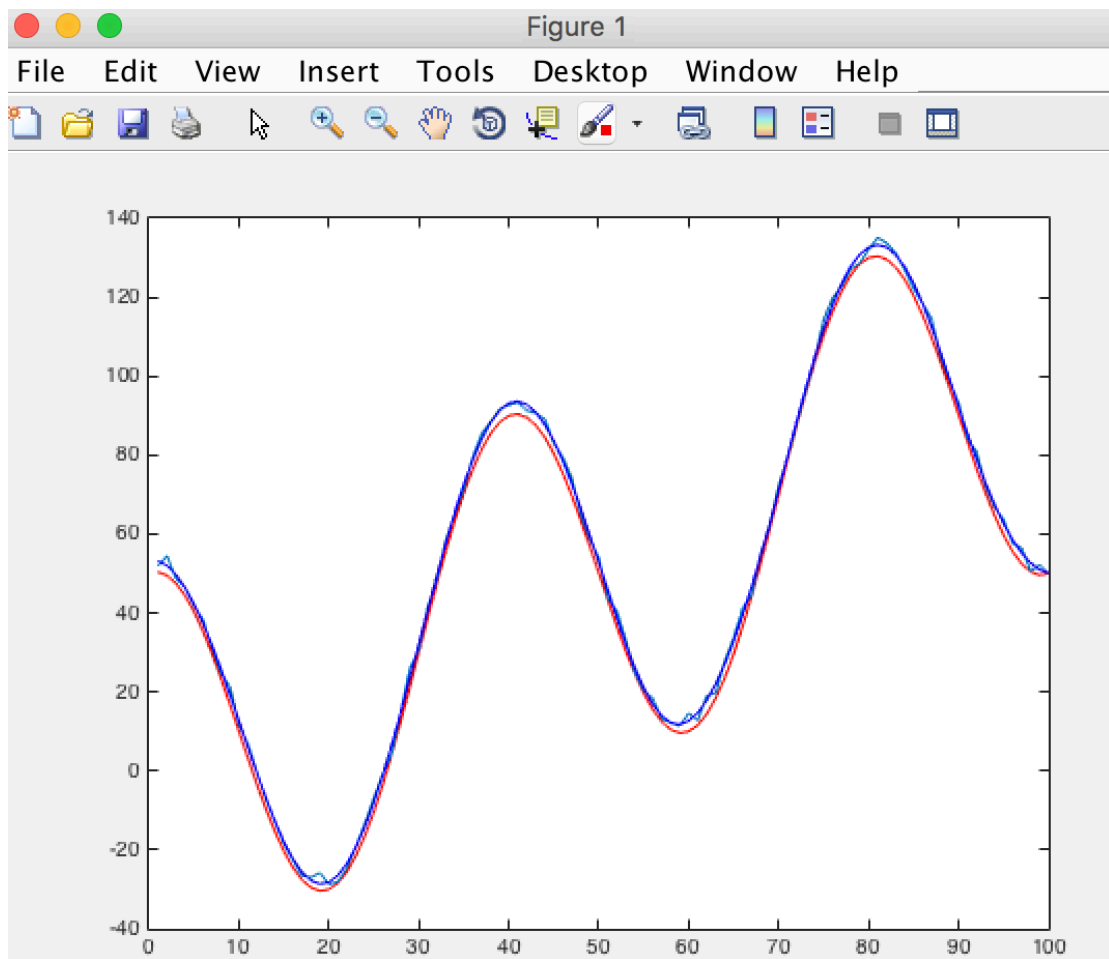
my\_rdct =

0.4203	0.9558	0.2503	-0.0352	0.6805
--------	--------	--------	---------	--------

error =

-0.1281	0.1214	0.1605	-0.0406	-0.1020
---------	--------	--------	---------	---------

结果三：



`plot(x, 'r')`%红颜色线为原信号

`plot(x + n, 'g')`%绿颜色线为带噪声的信号

`plot(xx, 'b')`%蓝颜色线为利用 DCT 去噪的信号