

implementSA

自动根据输入的码元信号，码字长度，码字数量进行编码，输出码集，码集最小码字响应距离，所有码字响应距离。

码元由码元信号在输入信号矩阵中的下标决定，由 0 起始（下标 1 的码元信号对应码元 0）。

1. 输入参数：

```
% symbolSignals 码元信号（多维矩阵：通道数*信号点数*码元数量）。
% codeLength 码字长度，即目标需以几位码元编码。
% reqCodewordNum 所需码字数量，即需编码多少目标。
% reheat 算法重载次数（缺省时默认为 0）。重载可能得到更好结果，但会消耗更多时间。
% fig 是否绘制码字响应距离图（缺省时默认为'y'，取消'n'）
```

输出参数：

```
% reqCodeword 目标编码码集
% minCodewordEnergyDiff 最小码字响应距离
% reqCodewordEnergyDiffMatrix 码字响应距离
```

Example

示例数据 Example symbolSignals.mat 中有 9*500*4 矩阵数据，模拟 4 个 9 通道信号点数为 500 的 SSVEP 信号，以 symbolSignals 参数输入。

码字长度 codeLength 设置为 4。

编码 40 目标故设置 reqCodeword = 40。

运行

```
[reqCodeword, minCodewordEnergyDiff, reqCodewordEnergyDiffMatrix] =  
implementSA ( symbolSignals, codeLength, reqCodewordNum )
```

输出结果：

reqCodeword

reqCodeword ✕					
40x4 double					
	1	2	3	4	
1	0	0	2	2	
2	0	0	3	0	
3	0	1	0	2	
4	0	2	1	0	
5	0	2	3	3	
6	0	3	1	3	
7	0	3	2	0	
8	1	0	1	3	
9	1	0	2	1	
10	1	0	3	2	

minCodewordEnergyDiff

minCodewordEnergyDiff ✕			
1x1 double			
	1	2	3
1	93.4106		
2			

reqCodewordEnergyDiffMatrix: 码字编号与 reqCodeword 中对应。

reqCodewordEnergyDiffMatrix ✕							
40x40 double							
	1	2	3	4	5	6	7
1	0	111.4496	110.3082	174.6065	159.2469	141.8540	110.4068
2	111.4496	0	157.0628	128.8748	110.4068	158.7318	94.5519
3	110.3082	157.0628	0	157.6103	141.8540	159.6759	192.5270
4	174.6065	128.8748	157.6103	0	111.9772	94.5519	95.0994
5	159.2469	110.4068	141.8540	111.9772	0	113.0200	142.3493
6	141.8540	158.7318	159.6759	94.5519	113.0200	0	94.0567
7	110.4068	94.5519	192.5270	95.0994	142.3493	94.0567	0
8	141.7554	158.6332	187.7652	157.0628	175.5308	93.4106	187.4672
9	93.9581	141.1093	204.2662	204.2662	223.3281	205.9353	140.0665
10	94.4533	110.3082	140.0665	239.1830	158.1055	206.4305	204.8601
11	187.9161	205.1905	140.6140	140.6140	224.4033	177.1012	205.8366
12	221.2625	203.7187	141.2601	139.9679	158.6332	206.3319	204.7615
13	157.6103	239.1830	140.6140	110.3082	159.6759	142.2507	205.4076
14	173.9604	158.1055	221.2625	93.9581	141.2079	188.5100	94.4533

以 reqCodewordEnergyDiffMatrix 绘图:

