

# 西门子S7-400模块代理商6ES7403-1JA00-0AA0

产品名称	西门子S7-400模块代理商6ES7403-1JA00-0AA0
公司名称	湖南西控自动化设备有限公司
价格	.00/件
规格参数	西门子:授权代理商 PLC:一级代理商 德国:售后保障服务
公司地址	中国（湖南）自由贸易试验区长沙片区开元东路1306号开阳智能制造产业园（一期）4#栋301
联系电话	17838383235 17838383235

## 产品详情

### 突破噪声与异常值的困扰（3）中位值滤波法（含示例代码）

前言：

在前面的文章中，我们介绍了一阶滞后滤波法和算术平均滤波法。这篇文章，我们来介绍中位值滤波法。他们都是模拟量信号处理中，常用的滤波方法之一。这三种方法都可以用来平滑信号，去除噪声和波动，但它们的实现方式和效果略有不同，侧重点不同：

算术平均滤波法侧重于平滑信号，减小噪声和波动。

一阶滞后滤波法侧重于快速响应信号的变化，较快地跟踪信号的动态变化。

中位值滤波法侧重于去除脉冲噪声和突变，保留信号特征。

选择适当的滤波方式取决于应用场景、信号特性和滤波要求。根据需要，可以根据信号的特点和噪声类型，选择合适的滤波方法或者结合多种滤波方法，以达到佳的滤波效果。

2

引言：

在现实生活和工程领域中，我们经常会遇到信号受到干扰或含有异常值的情况。为了得到准确的信号信息，我们需要对这些干扰进行滤除。中位值滤波法是一种有效的滤波方法，能够在一定程度上克服干扰，提高信号的质量和可靠性。

3

中位值滤波法的原理：

中位值滤波法的原理非常简单，它通过对一组采样值排序并选择其中的中位数作为输出值。具体步骤如下：

初始化一个长度为  $n$  的数组，将输入的采样值存入数组中。对数组进行排序，可以使用冒泡排序等方法。输出排序后的数组中的中间值，即中位数。

4

中位值滤波法的优缺点：

中位值滤波法具有以下优点：

**良好的去噪效果：**中位值滤波法能够有效去除脉冲噪声和突变，特别是对于连续的噪声和异常值有较好的滤除效果。它基于选择中间值作为滤波后的数值，减少了异常值对滤波结果的影响。

**保留信号特征：**相比于其他滤波方法，中位值滤波法相对保持了信号的原始特征。它不会对信号的整体形态造成明显的变化，更适合需要保留信号细节或特征的应用场景。

**鲁棒性较强：**中位值滤波法对于极大或极小的异常值有较好的鲁棒性和抗干扰能力。由于选取中间值作为输出，而不是受异常值的影响较大的平均值，因此较不容易受到极端值的影响。

中位值滤波法也存在一些缺点：

**计算复杂度较高：**中位值滤波法需要对一段时间内的多个采样值进行排序，这在计算上相对较为复杂。特别是对于大型数据集或实时应用来说，计算时间可能较长，影响实时性能。

**抑制快速变化的信号：**由于中位值滤波法是基于排序的操作，它对于快速变化的信号的响应较慢，无法很好地保留信号的快速变化部分。如果应用中需要较快的动态响应，可能不适合使用中位值滤波法

5

中位值滤波法的应用场景：

中位值滤波法在信号处理中有广泛的应用场景，特别适合处理存在脉冲噪声和突变的信号。以下是一些适合应用中位值滤波法的场景：**传感器信号处理：**当传感器测量值存在脉冲噪声或突变时，中位值滤波法可用于去除这些异常值。例如，对于温度、湿度、光照等传感器的数据，中位值滤波法可以过滤掉异

常的测量值，得到较为稳定和准确的结果。数据采集和信号采样：在数据采集或信号采样过程中，噪声会对采集到的数据产生干扰。中位值滤波法可以应用于对采样数据进行平滑和去噪，提高数据质量和准确性。数字通信系统：在数字通信系统中，信号中常常存在时域或频域上的突变或异常波形。中位值滤波法可用于去除这些突变，使得信号更平滑，减小通信中的误码率和误差。

6

示例代码：

下面是一个基于博图平台的中位值滤波法的案例程序，如果想要应用在实际项目当中，可能需要根据实际情况进行修改：

```
FUNCTION_BLOCK "中位值滤波法" { S7_Optimized_Access := 'TRUE' } VERSION : 0.1 V
AR_INPUT   inValue : Real; // 输入需要滤波的值   timeMs : Int :
= 10; // 设定多久采集一次数, 单位ms END_VAR
VAR_OUTPUT   filter : Real; // 输出滤波后的结果 END_VAR
VAR_IN_OUT   n : Int := 5; // 采集数量 END_VAR VAR
    staP : Bool;   staTimer { InstructionName := 'TON_TIME'; LibVersion := '1.0
'} : TON_TIME;   staValue { S7_SetPoint := 'True' } : Array[0..999] of Real;
    staTempValue : Array[0..999] of Real;   staX : Int;   staJ : Int;
    stal : Int;   value_temp : Real; END_VAR
```

```

VAR_TEMP   temMod : Int;   temTime : Time; END_VAR BEGIN
//n 须为3<n<999的奇数 IF #n < 3 THEN   #n := 3; END_IF; IF #n
> 999 THEN   #n := 999; END_IF; #temMod := #n MOD 2; IF #temM
od = 0 THEN   #n := #n + 1; END_IF; //转换成时间格式 #temTime :
= INT_TO_TIME(#timeMs); //生成脉冲 , 对n个数堆栈 #staTimer(IN := NOT #staP,
    PT := #temTime,
    Q => #staP); IF #staP THEN   FOR #staX := 0 TO #n -
2 DO   #staValue[#staX] := #staValue[#staX + 1];   END_FOR;
    #staValue[#n - 1] := #inValue; END_IF; // 将value数组中的数复制到temp_value
数组中 MOVE_BLK(IN := #staValue[0],   COUNT := 1000,
    OUT => #staTempValue[0]); //对temp_value数组中数从小到大排序 FOR #staJ :
= 0 TO #n - 3 DO   FOR #staI := #staJ TO #n - 2 DO
    IF #staTempValue[#staJ] > #staTempValue[#staI + 1] THEN   #v
alue_temp := #staTempValue[#staI + 1];   #staTempValue[#staI + 1]
:= #staTempValue[#staJ];   #staTempValue[#staJ] := #value_temp;
    END_IF;   END_FOR; END_FOR; //在排好序的数组中,取中间数做
为输出 #filter := #staTempValue[(#n - 1) / 2];END_FUNCTION_BLOCK

```

该程序实现了一个中位值滤波法的函数块，通过传入采样值和采样次数，可以得到滤波后的输出值。在程序中，通过使用冒泡排序对采样值进行排序，并选取排序后的中间值作为滤波结果。

7

总结：

中位值滤波法是一种简单但有效的信号处理方法，具有广泛的应用场景。它通过选择一组采样值的中位数来平滑信号、去除噪声和异常值。然而，中位值滤波法对于快速变化的参数效果较差，且存在一定的延迟。在实际应用中，需要根据具体情况综合考虑选择合适的滤波方法。