

```
1  var a = 1;
```

```
1  a = 1
```

```
1  def fus():  
2      dwawd = 10  
3      return "ds"
```

## 附件

附件清单：

- xxx 代码

**sobel 边缘检测代码**

```
function GAdsa
```

1. dwda
2. dwafwfe
3. fawfe
4. erf
5. dwadwa
6. dwad

**附录 i** dwadf

**附录 ii** dwfgh

**附录 iii** grhyh

**附录 iv** grncsq

**附录 v** ddwfafa

dwawdaf

dwawdaf

dwawdaf

dwawdaf

dwawdaf

```

1      %
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

2  %滑窗线性检测
3  %处理空速
4  %
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

5  function type_mat1 = TAS(data)
6  %
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

7      %关键参数
8      w = 10;
9
10 %
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

11     velocity = data(:,11); %获得空速序列
12     dvelocity = abs(diff(velocity));
13     k_s = mean(dvelocity);
14     delta_s = mean(dvelocity) * w;
15     t = data(:,1); %获得时间序列
16     big_win_start = 1; %大小滑窗初始化
17     big_win_end = 1+w;
18     small_win_start = 1;
19     small_win_end = 1+w;
20     MAX_time_index = length(t); %序列长度
21     index = 1;
22     %1:上升 0:平稳 -1:下降
23     while small_win_end <= MAX_time_index %不断滑动检测小窗
24         [p,~,~] = polyfit(t(small_win_start:small_win_end),velocity(
                small_win_start:small_win_end),1); %拟合小窗数据
25         if small_win_start == big_win_start %初始化滑动窗
26             if p(1) >= k_s
27                 Type = 1;
28             elseif p(1) <= -k_s
29                 Type = -1;
30             else

```

```

31         Type = 0;
32     end
33     small_win_start = small_win_start + 1;
34     small_win_end = small_win_end + 1;
35     big_win_end = small_win_end;
36 else
37     if Type == 1
38         if p(1) > 0 %前一刻和此刻都是上升
39             small_win_start = small_win_start + 1;
40             small_win_end = small_win_end + 1;
41             big_win_end = small_win_end;
42         else %变号了, 固定更新
43             type_mat(index) = 1;
44             index_mat(index) = big_win_end-1;
45             index = index + 1;
46             small_win_start = small_win_end;
47             small_win_end = small_win_start + w;
48             big_win_start = small_win_start;
49             big_win_end = small_win_end;
50             delta = 0;
51         end
52     elseif Type == -1
53         if p(1) < 0 %前一刻和此刻都是下降
54             small_win_start = small_win_start + 1;
55             small_win_end = small_win_end + 1;
56             big_win_end = small_win_end;
57         else %变号了, 固定更新
58             type_mat(index) = -1;
59             index_mat(index) = big_win_end-1;
60             index = index + 1;
61             small_win_start = small_win_end;
62             small_win_end = small_win_start + w;
63             big_win_start = small_win_start;
64             big_win_end = small_win_end;
65             delta = 0;
66         end
67     else
68         if abs(p(1)) >= k_s
69             type_mat(index) = 0;

```

```

70         index_mat(index) = big_win_end-1;
71         index = index + 1;
72         small_win_start = small_win_end;
73         small_win_end = small_win_start + w;
74         big_win_start = small_win_start;
75         big_win_end = small_win_end;
76         delta = 0;
77     else
78         delta = max(velocity(big_win_start:big_win_end))-
              min(velocity(big_win_start:big_win_end));
79         if abs(delta) >= delta_s
80             if delta > 0
81                 type_mat(index) = 1;
82                 index_mat(index) = big_win_end-1;
83                 index = index + 1;
84                 small_win_start = small_win_end;
85                 small_win_end = small_win_start + w;
86                 big_win_start = small_win_start;
87                 big_win_end = small_win_end;
88                 delta = 0;
89             else
90                 type_mat(index) = -1;
91                 index_mat(index) = big_win_end-1;
92                 index = index + 1;
93                 small_win_start = small_win_end;
94                 small_win_end = small_win_start + w;
95                 big_win_start = small_win_start;
96                 big_win_end = small_win_end;
97                 delta = 0;
98             end
99         else
100             small_win_start = small_win_start + 1;
101             small_win_end = small_win_end + 1;
102             big_win_end = small_win_end;
103         end
104     end
105 end
106 end
107 end

```

```

108
109     type_mat(index) = Type; %保存最后一个状态
110     index_mat(index) = MAX_time_index-2;
111     type_mat1 = zeros('like',type_mat);
112     type = type_mat(1);
113     if type == 1
114         type_mat1(1:index_mat(1)) = 1;
115     elseif type == 0
116         type_mat1(1:index_mat(1)) = 0;
117     else
118         type_mat1(1:index_mat(1)) = -1;
119     end
120     for l = 2:length(type_mat)
121         type = type_mat(l);
122         if type == 1
123             type_mat1(index_mat(l-1):index_mat(l)) = 1;
124         elseif type == 0
125             type_mat1(index_mat(l-1):index_mat(l)) = 0;
126         else
127             type_mat1(index_mat(l-1):index_mat(l)) = -1;
128         end
129     end
130     type_mat1 = type_mat1(1:MAX_time_index-2);
131 end

```