```
var a = 1;
```

```
a = 1
```

```
def fus():
    dwawd = 10
    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 grrncsq

附录 v ddwfafa

dwawdaf

dwawdaf

dwawdaf

dwawdaf

dwawdaf

```
    % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%滑窗线性检测
%处理空速
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function type_mat1 = TAS(data)
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %关键参数
    w = 10;

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    velocity = data(:,11); %获得空速序列
    dvelocity = abs(diff(velocity));
    k_s = mean(dvelocity);
    delta_s = mean(dvelocity) * w;
    t = data(:,1); %获得时间序列
    big_win_start = 1; %大小滑窗初始化
    big_win_end = 1+w;
    small_win_start = 1;
    small_win_end = 1+w;
    MAX_time_index = length(t);%序列长度
    index = 1;
    %1:上升 0:平稳 -1:下降
    while small_win_end <= MAX_time_index %不断滑动检测小窗
        [p,~,~] = polyfit(t(small_win_start:small_win_end),velocity(small_win_start:small_win_end),1); %拟合小窗数据
        if small_win_start == big_win_start %初始化滑动窗
            if p(1) >= k_s
                Type = 1;
            elseif p(1) <= -k_s
                Type = -1;
            else
```

```
            Type = 0;
        end
        small_win_start = small_win_start + 1;
        small_win_end = small_win_end + 1;
        big_win_end = small_win_end;
    else
        if Type == 1
            if p(1) > 0 %前一刻和此刻都是上升
                small_win_start = small_win_start + 1;
                small_win_end = small_win_end + 1;
                big_win_end = small_win_end;
            else  %变号了，固定更新
                type_mat(index) = 1;
                index_mat(index) = big_win_end-1;
                index = index + 1;
                small_win_start = small_win_end;
                small_win_end = small_win_start + w;
                big_win_start = small_win_start;
                big_win_end = small_win_end;
                delta = 0;
            end
        elseif Type == -1
            if p(1) < 0 %前一刻和此刻都是下降
                small_win_start = small_win_start + 1;
                small_win_end = small_win_end + 1;
                big_win_end = small_win_end;
            else  %变号了，固定更新
                type_mat(index) = -1;
                index_mat(index) = big_win_end-1;
                index = index + 1;
                small_win_start = small_win_end;
                small_win_end = small_win_start + w;
                big_win_start = small_win_start;
                big_win_end = small_win_end;
                delta = 0;
            end
        else
            if abs(p(1)) >= k_s
                type_mat(index) = 0;
```

```
                    index_mat(index) = big_win_end-1;
                    index = index + 1;
                    small_win_start = small_win_end;
                    small_win_end = small_win_start + w;
                    big_win_start = small_win_start;
                    big_win_end = small_win_end;
                    delta = 0;
                else
                    delta = max(velocity(big_win_start:big_win_end))-
                       min(velocity(big_win_start:big_win_end));
                    if abs(delta) >= delta_s
                        if delta > 0
                            type_mat(index) = 1;
                            index_mat(index) = big_win_end-1;
                            index = index + 1;
                            small_win_start = small_win_end;
                            small_win_end = small_win_start + w;
                            big_win_start = small_win_start;
                            big_win_end = small_win_end;
                            delta = 0;
                        else
                            type_mat(index) = -1;
                            index_mat(index) = big_win_end-1;
                            index = index + 1;
                            small_win_start = small_win_end;
                            small_win_end = small_win_start + w;
                            big_win_start = small_win_start;
                            big_win_end = small_win_end;
                            delta = 0;
                        end
                    else
                        small_win_start = small_win_start + 1;
                        small_win_end = small_win_end + 1;
                        big_win_end = small_win_end;
                    end
                end
            end
        end
    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
```