
Algorithm 1 Insertion Sort

```
1: for  $i = 1$  to  $length \dots$  do  
2:    $Current = x[j]$   
3:    $Temp = i - 1$   
4:   while  $Temp > 0$  and  $arr[Temp] > current$  do  
5:      $arr[Temp + 1] = arr[Temp]$   
6:      $Temp --$   
7:   end while  
8:    $arr[Temp + 1] = Current$   
9: end for  
10:
```

Algorithm 2 Game Theory Controller

```
1: for Every time step do
2:   Calculate target seeking command  $\mathbf{x}_{tsCmd}$  (Eq.: 3.12)
3:   for All map measurements from  $\mathbf{x}_{Map}$  do
4:     Denormalize measurement (Eq.: 3.14)
5:     Add margin of safety (Eq.: 3.15)
6:     Calculate altitude difference  $\Delta h_{ObsSafe_j}$  to aircraft (Eq.: 3.16)
7:     if  $\Delta h_{ObsSafe_j} > 0$  then
8:       Add measurement to set of critical measurements  $\mathcal{M}_{crit}$  (Eq.:
3.17)
9:     end if
10:  end for
11:  for All measurements in  $\mathcal{M}_{crit}$  do
12:    Calculate local obstacle avoidance vector (Eq.: 3.20)
13:  end for
14:  Sum over all local avoidance vectors (Eq.: 3.22)
15:  Transform to global coordinate frame to receive  $\mathbf{x}_{oaCmd}$  (Eq.: 3.23)
16:  Calculate obstacle avoidance weight  $w_{oa}$  based on critical zone weight
(Eq.: 3.24)
17:  Calculate target seeking weight  $w_{ts}$  as  $1 - w_{oa}$  (Eq.: 3.13)
18:  Calculate command vector  $\mathbf{x}_{HsaCmd} = w_{oa}\mathbf{x}_{oaCmd} + w_{ts}\mathbf{x}_{tsCmd}$  (Eq.:
3.11)
19: end for
```
