
Algorithm 1 BubbleSort

Require: *Integer Array A***Ensure:** *SortedArray A*

```
1: f = TRUE
2: c = 1
3: while f AND c < A.length do
4:   f = FALSE
5:   for i = 1 to arr.length - 1 do
6:     if A[i] > A[i + 1] then
7:       f = TRUE
8:       v = A[i]
9:       A[i] = A[i + 1]
10:      A[i + 1] = v
11:    end if
12:  end for
13: end while
```

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
```
