```
1:  Algorithm Augmented_MCL(X_{t-1}, u_t, z_t, m):
2:      static w_slow, w_fast
3:      X̄_t = X_t = ∅
4:      w_avg = 0
5:      for m = 1 to M do
6:          x_t^[m] = sample_motion_model(u_t, x_{t-1}^[m])
7:          w_t^[m] = measurement_model(z_t, x_t^[m], m)
8:          X̄_t = X̄_t + ⟨x_t^[m], w_t^[m]⟩
9:          w_avg = w_avg + (1/M) w_t^[m]
10:     endfor
11:     w_slow = w_slow + α_slow(w_avg − w_slow)
12:     w_fast = w_fast + α_fast(w_avg − w_fast)
13:     for m = 1 to M do
14:         with probability max{0.0, 1.0 − w_fast/w_slow} do
15:             add random pose to X_t
16:         else
17:             draw i ∈ {1, ..., N} with probability ∝ w_t^[i]
18:             add x_t^[i] to X_t
19:         endwith
20:     endfor
21:     return X_t
```

Table 8.3 An adaptive variant of MCL that adds random samples. The number of random samples is determined by comparing the short-term with the long-term likelihood of sensor measurements.

in lines 11 and 12. The algorithm requires that $0 \leq \alpha_{\text{slow}} \ll \alpha_{\text{fast}}$. The parameters α_{slow}, and α_{fast}, are decay rates for the exponential filters that estimate the long-term, and short-term, averages, respectively. The crux of this algorithm can be found in line 14: During the resampling process, a random sample is added with probability

$$\max\{0.0,\ 1.0 - w_{\text{fast}}/w_{\text{slow}}\} \tag{8.5}$$

Otherwise, resampling proceeds in the familiar way. The probability of adding a random sample takes into consideration the divergence between