Pseudo-Algorithms

Nodari Alessandro & Proserpio Lorenzo

November 3, 2022

Algorithm 1: Predict

for $j = 0, ..., N-1$ **do**
    check=False;
    if *new_measurement* $= 0$ *and first_particle*$[j]$ *is alive* **then**
        draw $U \sim U[0,1]$;
        if $U <$*probability of death* **then**
            kill the j-component of all the particles;
        end
    else
        if *first_particle*$[j]$ *is dead* **then**
            draw $U \sim U[0,1]$;
            if $U <$*probability of birth* **then**
                give birth to the j-component of all the particles with distribution $\mathcal{N}$(new_measurement, std_particles);
            end
        else
            if *new_measurement is voted as a clutter* **then**
                use the Kalman filters to update the mean of the states and the covariance, using the last estimation as observation;
                update the weight of the particle with the log-likelihood;
                add 1 to the clutter count;
                if *clutter count* $>$ *NC* **then**
                    check=True;
                end
            else
                use the Kalman filters to update the mean of the states and the covariance, using the new_measurement as observation;
                update the weight of the particle with the log-likelihood;
                reset the clutter count to 0;
            end
            if *check* $=$ *True* **then**
                kill the j-component of all the particles;
            end
        end
    end
end

Algorithm 2: Main

for $i = 0, 1, ..., NP$ **do**
  create particle[i];
end
for $i = 0, 1, ..., T$ **do**
  Predict(particles, new_measurement);
  normalize the weights of the particle; stratified resample the particles if needed;
  collect the particles result;
end
for $t = 0, 1, ..., T$ **do**
  combine the predictions of the particles with the corresponding weights;
end