Applied Estimation Lab 2—Particle Filter

Michal Staniaszek

December 5, 2013

1 Part I—Preparatory Questions

1. Particles are used to represent a hypothesis of a single possible state of the system, which are drawn from some probability distribution. A single particle alone is not particularly useful, as it has no explanatory power about the possbile distribution of the true state. Instead, a large number of particles are used to represent belief in the form of a particle cloud.

2. The *importance weight* is a weight assigned to each particle to indicate its expressive power as a part of the particle cloud. Particle weights depend on the measurement update. If the measurements that we find by simulating a set of measurements from the particle's state closely match the measurements given by our true system, the particle is given a proportionally higher weight. If the measurements are a poor match, then it receives a low weight. As such, a higher weight indicates a higher probability of the system being in that state. The *target distribution* is the true state of the system. We do not know what the actual distribution of the true state is, but if it were possible to have an infinite number of particles, then we could represent the target distribution exactly. The *proposal distribution* is the distribution that is represented by the particles. This distribution is the target distribution, approximated by a finite number of particles. In the particle filter, we try to estimate the target distribution by the proposal distribution, the shape of which is defined by the positions and importance weights of a particle cloud.

3. Particle deprivation occurs as a result of the resampling step of the filter. While it is more likely when there are not enough particles to cover all of the relevant regions of the target distribution, it happens in particle filters with any number of particles due to the nature of random sampling. The danger of deprivation is that it can result in there being no particles near the true state of the system.

4. If instead of resampling we simply updated weights for the particles, we would end up with particles in areas of the space which simply did not need representation because the probability of the true state being in that area is very low. These particles would be wasted representing this space. If we instead resample and represent the areas in which the system has a high probability of being, then we cover more positions that might be the true state. In essence, we would like to have the number of particles in a region proportional to the probability of the state of being in that region.

5. The average of the particle set is usually not a good representation of the set, particularly in cases where the proposal distribution is multimodal, that is, when it has multiple

peaks, or there are multiple groups of particles. In the case of a two-peaked distribution, the average will end up in the centre of the two groups, and is not representative of any state in which the system is likely to be.

6. To make inferences about the probability of states between particles we can use histograms or Gaussian kernels. In the case of histograms, we can find the probability of the system being in an intermediate bin by interpolating the values of two adjacent bins. With Gaussian kernels, we can place a Gaussian on each particle, with a height proportional to the weight of the particle and with the same variance for each particle, depending on some uncertainty measure. Having summed all of these Gaussians, we could get information about intermediate states by looking at the resulting distribution.

7. Sample variance is the variability introduced due to random sampling from a distribution. As we sample only a certain number of times, the statistics of the new distribution will vary slightly from that of the original. This can cause problems as if the sample variance is too large the representation of the true belief will not be good. There are a number of techniques available to perform variance reduction. One technique is to increase the time between subsequent resamplings. Another is to use low variance sampling, which uses a stochastic process to select samples instead of sampling them independently.

8. If the pose uncertainty is large, the target distribution has a large spread, and therefore may have many peaks in different places which must all be represented to have an accurate proposal distribution. Do do so, we need a large number of particles. Thus, for a higher pose uncertainty, a larger number of particles are required.

2 Part II—Warm-up Problem

1. In the 2D state space model, the initial value of θ_0 is never modified after it is set. This means that although we introduce noise in to x and y, the angle of motion always remains the same, so we will end up with a jagged line. In contrast, the 3D state space model allows the modification of θ_0 from its original value based on the noise which is added. This results in small variations in its value, so it is possible to get slightly curved motion due to the use of the angle from the previous timestep, $x_{t-1,\theta}$.

2. With this model, the initial translational velocity v_0 and angular velocity ω_0 must be known or fixed. With this model it is only possible to model perfectly circular motions, in a specific direction. This is because once the initial values have been set, it is not possible to modify them other than by the additive noise.

3. The constant part is a normalisation constant. Values are divided by a number proportional to Σ_Q, which represents the spread of the Gaussian.

4. For multinomial resampling, M random numbers must be generated, corresponding to the number of particles. With systematic sampling only a single one has to be generated.

5. With multinomial sampling, the probability of not sampling a given particle p_i with weight w_i is $1 - w_i$. The remainder of the probability is the probability that any of the other $M - 1$ particles are selected. For a specific particle *not* to be selected after a whole round of M independent sampling attempts is $(1 - w_i)^M$. Thus, the probability

Particles	Error (2D case)	Error (3D case)
10	10.4±43.7	2.2±2.9
30	11.7±45.0	1.2±1.1
50	1.3±3.1	1.0±1.2
100	1.1±2.7	1.0±2.5
200	0.7±1.3	0.7±1.2
400	0.5 ±0.5	0.9±2.5
600	0.5 ±0.6	0.5±0.4
1000	0.4±0.2	0.5±0.7

Table 1: Error in estimate for single trials with different numbers of particles using either a 2D or 3D state space representation of motion. Below 100 particles, convergence depends heavily on the random initialisation placing particles close to the true state. The 3D representation appears to result in lower errors for fewer particles, and above 50 particles there is little difference in the results.

that a particles *does* survive the resampling step is $1 - (1 - w_i)^M$, regardless of its weight. In the case of systematic sampling, the random number $r \sim U[0, \frac{1}{M}]$ and the step size $\frac{1}{M}$ play an important role in deciding whether a particle survives. There are three interesting cases to consider: $w_i = \frac{1}{M} + \epsilon$, where ϵ is a small positive number, $w_i = \frac{1}{M}$, and $0 < w_i < \frac{1}{M}$. In the first case, the particle is guaranteed to survive—because the step size is smaller than the width of the bin, we will always choose it. The other two cases depend on the random number r chosen as the starting point and the ordering of the weights.

6. The measurement noise is modelled by `Sigma_Q`, and the process noise by `Sigma_R`.

7. Without the diffusion step, the filter very quickly ends up using only a single particle, or all the particles in exactly the same location. This is because all the particles which survive to the next step make exactly the same motion as the particle that they were sampled from. As a result of the random sampling, the single point which has more particles on it will be more likely to be selected again, and so eventually all particles will be identical to this one. This particles is the one which has the highest weight in the initial step.

8. If no resampling is done, we retain all of the particles in the original random set that we chose. The particles will all move according to the controls applied to the system with added noise, but there will be no convergence to the true state. Each particle acts as a simulation of the system from a different starting state. Some of the particles will represent something close to the true motion because of the random initialisation.

9. When the measurement noise is low, the estimate is very inaccurate as it is assumed that if predicted measurements do not correspond very closely to the true measurement they are seen as outliers, and as a result particles do not converge to the true state. For measurement noise close to 1, the particles stay dispersed until a valid measurement is received, at which point they instantly converge to a tight cloud. After this point, increasing the noise simply increases the spread of the particle cloud, as more particles will have similar weights due to the modelled noise. Some experimental results can be seen in Table 2.

σ^2	Error
0.0001	194.9±0.3
1	4.7±26.8
10	0.9±7.5
100	0.4±0.2
1000	0.4±0.4
10000	0.5±0.9

Table 2: Modifying measurement noise with process noise constant at $R = \texttt{diag}(2, 2, 0.01)$ for the fixed target with 1000 particles. For $\sigma^2 < 1$, error is high due to the overoptimistic measurement model.

10. When the process noise R is low, the diffusion of the particle cloud in the prediction step is small, which results in a tight particle cloud. The convergence of the cloud, and its error are highly dependent on the initialisation. Since the process has almost no noise, particles do not move around apart from by being resampled. The cloud converges gradually to the point which is closest to the true state, the particle with the highest weight. As the noise increases, the spread becomes greater and particles make larger motions in the space, which means that convergence is reached more quickly as particles are more likely to move into the vicinity of the true state.

11. When choosing the process noise model, it is important to consider which motion model is being used, depending on the simulated motion. If the motion model does not match the simulated motion, then the process noise should be increased to compensate for the inability of the motion model to capture the true motion of the system. If the correct motion model is being used, then in general a lower process noise is acceptable.

12. Choosing the motion model which corresponds with the actual motion of the object results in better estimates, and means that lower noise parameters can be used, as well as fewer particles. If the motion model does not match, then more particles are required to ensure that the true state lies within the particle cloud.

13. In the third type of measurements, 50% are outliers which are very far from the true state of the system. It is possible to determine that these are outliers by looking at their likelihood given our particle cloud—if the average likelihood of the measurement is very low, then we can say with relative confidence that it is an outlier and reject it based on some threshold. In the code, the `thresh_avg_likelihood` parameter allows the adjustment of this average likelihood.

14. Table 3 shows experimental results for several settings of the noise parameters Q and R with 1000 particles. From the limited number of experiments performed, we obtain a best precision of 7.9 ± 4.1 for the linear model and 7.9 ± 4.2 for the circular model, both from $Q = \texttt{diag}(300, 300)$, $R = \texttt{diag}(4, 4, 0.02)$. From the same setting we get an error of 20.0 ± 4.7 for the fixed model. The fixed model requires an increased process noise to come close to the error values for the other models. The best precision that could be obtained was 11.3 ± 5.5, using $Q = \texttt{diag}(500, 500)$, $R = \texttt{diag}(30, 30, 0.15)$. These results indicate that the linear and circular models give very similar results, while the fixed model is much more sensitive to the choice of noise models. In general, we could say that the better the model corresponds to the true motion, the lower the noise required to get reasonable results.

Noise Models	Motion model	Error
$Q = \texttt{diag}(100, 100)$, $R = \texttt{diag}(2, 2, 0.01)$	f	220.5±134.7
	l	9.0±4.7
	c	8.5±4.3
$Q = \texttt{diag}(300, 300)$, $R = \texttt{diag}(4, 4, 0.02)$	f	20.0±4.7
	l	7.9±4.1
	c	7.9±4.2
$Q = \texttt{diag}(500, 500)$, $R = \texttt{diag}(10, 10, 0.05)$	f	14.7±5.1
	l	8.4±4.3
	c	8.5±4.3
$Q = \texttt{diag}(1000, 1000)$, $R = \texttt{diag}(10, 10, 0.05)$	f	19.7±4.9
	l	8.6±4.0
	c	8.5±3.9
$Q = \texttt{diag}(1000, 1000)$, $R = \texttt{diag}(30, 30, 0.15)$	f	12.8±5.2
	l	12.7±5.2
	c	12.8±5.2
$Q = \texttt{diag}(500, 500)$, $R = \texttt{diag}(30, 30, 0.15)$	f	11.3±5.5
	l	10.2±5.4
	c	10.2±5.4

Table 3: Precision of different measurement models for object moving in a circle using various motion models and 1000 particles. Motion model f is fixed, l is linear, c is circular. Q is the measurement noise, and R the process noise. While the linear and circular model errors are generally close, the fixed model requires much higher uncertainty to come close to the lowest errors of the others.

3 Part III—Monte Carlo Localisation

15. The outlier detection approach is mostly affected by the measurement noise model Q, and also by the value of λ_Ψ, which determines the threshold for outliers. As Q tends towards zero, more and more outliers are detected as the filter requires increasingly precise measurements in order for the measurement to have a reasonable likelihood. When Q is zero, all measurements which are not identical to the true measurement are rejected, with the likelihood function forming a Dirac delta at the point of the measurement.

16. When outliers are not detected, the particle weights are corrupted, resulting in a worse representation of the posterior. This is because the filter considers the outlier measurements as valid, modifying the likelihood of the particle accordingly.

3.1 Results