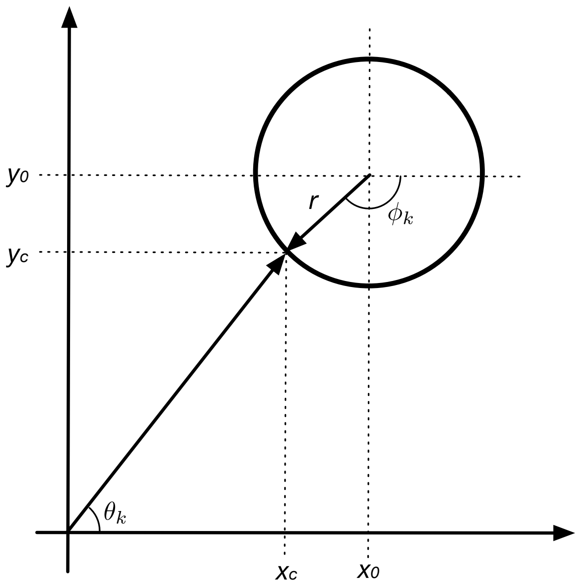
CR 09 Hidden Markov models for time series analysis
Lab Session – **Tracking a car driving around a circular track using Particle Filtering**

S. Derrode

The idea of this Lab is to implement and test particle filtering with a simple but realistic situation. We will take the opportunity to study the influence of the number of particles, the re-sampling strategy, the noise power...
Remark: the time index is denoted by '*k*'.

Context

- Suppose that we have a car driving around a circle with radius *200m*. Its velocity is nearly constant at *2°* per sample, but is perturbed by additive Gaussian noise modeling bumps, maneuvers…
- We have a sensor located at $500\sqrt{2}m$ from the center of the track that measures angle to the car, θ_k with noise $\pm 5°$ typical.
- We wish to estimate ϕ_k (*i.e.* state $x_k = \phi_k$).



Equations and constants

- The state equation is
  $$\phi_{k+1} = \phi_k + \omega + w_k$$
  where ω is set to *2°* per time sample, and
  $$w_k \rightsquigarrow \mathcal{N}\left(0, \sigma_w^2\right)$$
- The position of the car is given by
  $$x_{c,k} = r\cos\phi_k + x_0$$
  $$y_{c,k} = r\sin\phi_k + y_0$$
- The observations are given by θ_k according to
  $$\theta_k = \tan^{-1}\left(\frac{y_{c,k}}{x_{c,k}}\right) + v_k$$
  with
  $$v_k \rightsquigarrow \mathcal{N}\left(0, \sigma_v^2\right)$$
- Let $x_0 = y_0 = 500, r = 200$ and let
  $$\sigma_v = \frac{5\pi}{180}, \quad \sigma_w = \frac{10\pi}{180}, \quad \phi_0 \rightsquigarrow \mathcal{N}\left(\frac{50\pi}{180}, \left(\frac{10\pi}{180}\right)^2\right)$$

For the estimator of state, we will use the mean:

$$\phi_k^{Mean} = \sum_{i=1}^{N_p} w_k^i \; \phi_k^i$$

For the importance density *q*, we will use the simplest:

$$q\left(\phi_k|\phi_{k-1}, \theta_k\right) = q\left(\phi_k|\phi_{k-1}\right) \rightsquigarrow \mathcal{N}\left(\phi_{k-1} + \frac{2\pi}{180}, \sigma_w^2\right)$$

So that we have $\tilde{w}_n^i = w_{n-1}^i \; p\left(\theta_k|\phi_k\right)$ (don't forget to normalize the weights!)

For the data-driven density, we have:

$$p\left(\theta_k|\phi_k\right) \rightsquigarrow \mathcal{N}\left(\tan^{-1}\left(\frac{r\sin\phi_k + y_0}{r\cos\phi_k + x_0}\right), \sigma_v^2\right)$$

For the number of particles N_p, let's test!

Work to be done during the session

The main goal of this lab is to illustrate the particle degeneracy phenomenon of the particle filter (SIS algorithm) on the simple model described above, and to implement the resampling strategy (SIR algorithm).
Please follow the steps below and reports the figures and your own comments on a Microsoft Word-like report.
At the end of the session, don't forget to send a pdf version of your report to stephane.derrode@ec-lyon.fr.

1. Download the skeleton of the program (two python files and an empty repository called "*figures*"). Examine the code of "*TrackCar_PF.py*" that implements the model above and the suited particle filter (SIS strategy). Run the code and have a look to the generated figures in the "figures" repository.
   Remark: Please note the command "*np.random.seed(0)*" at the beginning of the main program and read the comment around. Also note that the resampling threshold *Nthr* is set to 0.0 (which means "no resampling").
   The MSE (Mean Square Error) between the true (simulated) states and the states estimated using the particle filter is reported on the Terminal. It should be around 0.722 if you don't change the parameters: seed set to 0, threshold set to 0.0, and a number of 200 particles.

2. The generated figure "*particlePath_200_0.0.png*" shows that the particle filter performs well at the beginning and then starts behaving less well (note that perfect recovering of the states is not possible). Does increasing the number of particles systematically improves the result? Try a few values and reports the obtained MSE. Comment on the results.

3. In file "*PlotPF.py*", write a function to plot the evolution of the of the particle weights and of the logarithm of the weights. For the formatting of the figure, you can use the other functions in the same file as an example. Comment on the figure in your report.

4. Write the code of the "resample" function which implements the SIR strategy for the PF. Make as many experiments as you want to test the effectiveness of the resampling strategy wrt the SIS one. Copy the code of the function in your report and, illustrate and comment your findings.

Resampling algorithm:

$$\left\{(x_n)^i, w_n^i\right\}_{i=1}^{N_p} = RESAMPLE\left(\left\{(x_n)^i, w_n^i\right\}_{i=1}^{N_p}\right)$$

- Cumulative summation of weights (*n* is given) $c_n^i = \sum_{j=1}^{i} w_n^j$
- Make a draw from $u_1 \rightsquigarrow \mathcal{U}\left(0, \left(N_p^{-1}\right)^2\right), i = 1$
- For j=1 to N_p:
  - $u_j = u_1 + (j-1)/N_p$
  - While $u_j > c_n^i, \quad i = i+1$
  - Assign new $\text{new } x_n^j = \text{ old } x_n^i, \quad \text{new } w_n^j = 1/N_p$