Advanced Tracking

Stefanela Stevanović, 63220492

I. INTRODUCTION

The goal of this assignment was to implement three motion models: Random Walk (RW), Nearly-Constant Velocity (NCV), and Nearly-Constant Acceleration (NCA). In the first part, Kalman filtering method using these three models was tested with different p and q parameters on artificially generated trajectories. In the second part of the assignment, particle filter tracker that uses the NCV motion model and a color histogram as a visual model was implemented. Tracker's accuracy and robustness were analyzed by running it on different sequences from the VOT2014 dataset. Furthermore, it was examined how different number of particles and different motion models affect the tracking performance.

II. KALMAN FILTERING

We conducted an experiment to evaluate the performance of three different motion models, Random Walk (RW), Nearly Constant Velocity (NCV), and Nearly Constant Acceleration (NCA), using the Kalman filter on three different curves. The motion models are parameterized by two variables, q and r, where q represents the uncertainty in the motion of the object and r represents the uncertainty in the measurements. Increasing either q or r results in a larger amount of uncertainty for the corresponding source of noise. We varied the values of q and r for each motion model and analyzed their impact on the tracking accuracy of the Kalman filter. Our results, shown in Figures 1, 2, and 3, demonstrate that increasing the value of r generally leads to worse tracking accuracy across all three motion models. Conversely, increasing the value of q improves performance particularly for the second and the third curve. On a Fig. 3 we see than on more complex curve Kalman filtering method underperforms in formarison to the Fig. 1 and Fig 2.

In addition, we also observed that the Nearly Constant Acceleration (NCA) model consistently outperformed the Nearly Constant Velocity (NCV) model, which in turn performed better than the Random Walk (RW) model. This suggests that incorporating information about acceleration in the motion model can lead to more accurate tracking than assuming a constant velocity or purely random motion. In summary, our findings emphasize the importance of selecting appropriate values for the parameters of the motion models in order to achieve optimal performance with the Kalman filter.

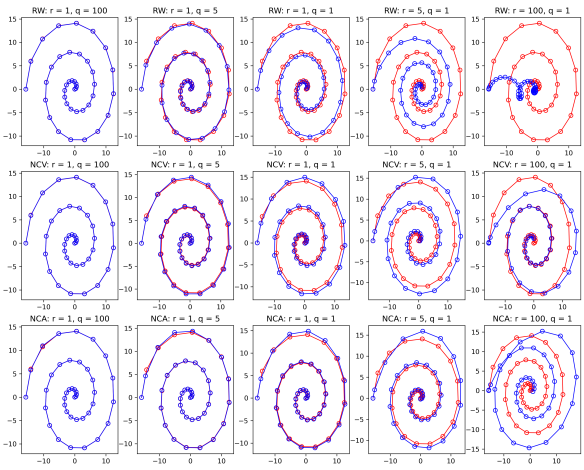

Fig. 1: Kalman filtering method on a spiral curve using RW, NCV and NCA motion models and different values of parameters r and q.

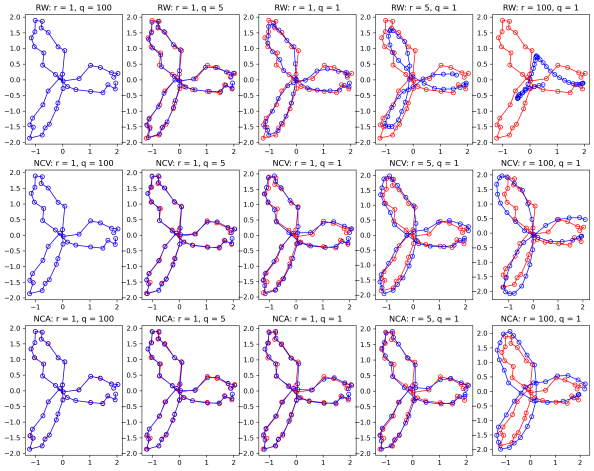

Fig. 2: Kalman filtering method on a jagged curve using RW, NCV and NCA motion models and different values of parameters r and q.

III. PARTICLE FILTER TRACKER

A. EXPERIMENT 1: Particle Filter Tracker on different sequences.

In the first experiment we implemented particle filter tracker using NCV motion model and tested it on the different sequences of the VOT2014 data set using following parameters: update factor α = 0.2, kernel σ = 0.1, enlarge_factor = 2, q = 100, n_particles = 20 and 16 histogram bins. Obtained results are shown in Table 1.

The first reported metric in the table I is the average overlap, which measures how much the predicted bounding box overlaps with the ground truth bounding box. An average overlap of 1 indicates perfect tracking, while a value of 0 indicates complete failure to track the object. Overall, the results suggest

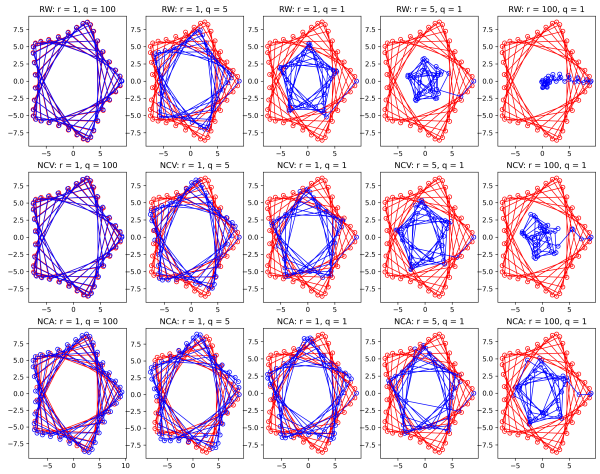

Fig. 3: Kalman filtering method on a complex curve using RW, NCV and NCA motion models and different values of parameters r and q.

TABLE I: Tracking performance of the particle filter tracked on different sequences

Sequence	Average Overlap	Failures	Tracking Speed (FPS)
bolt	0.42	8	90
david	0.54	2	340
diving	0.34	0	284
drunk	0.42	2	75
polarbear	0.43	2	135
sphere	0.35	5	111
skating	0.48	6	79

that the tracker performs better on some sequences than others, with the ”david” sequence showing the highest average overlap score and the ”diving” sequence having zero failures. The tracking speed also varies significantly across sequences.

B. EXPERIMENT 2: Particle Filter Tracker with different number of particles.

In this experiment, we analyzed the impact of the number of particles on the performance of the particle filter tracker that uses NCV motion model. To conduct this experiment, we used the ’bolt’ sequence with the following number of particles: 5, 10, 25, 50, 75 and 100. We kept all other parameters consistent with the first experiment. The ’bolt’ sequence was chosen because it had the highest number of failures in the first experiment, and therefore we expected to see a more significant change in the number of failures with varying numbers of particles. The result are presented in table 2.

TABLE II: Tracking performance of the particle filter tracked on ’bolt’ sequence with different number of particles.

No. of Particles	Average Overlap	Failures	Tracking Speed (FPS)
5	0.39	13	145
10	0.40	11	93
25	0.47	8	85
50	0.48	7	48
75	0.52	7	36
100	0.51	7	27

As the number of particles increases, the average overlap between the predicted and ground-truth bounding boxes also increases, while the number of failures decreases. However, there is a trade-off between tracking accuracy and speed, as the tracking speed decreases with an increasing number of particles. Overall, the results suggest that increasing the number of particles can improve tracking performance, but it also requires careful consideration of the computational resources available.

C. EXPERIMENT 3: Particle Filter Tracker with different motion models and parameters.

In this experiment we tested the performance of the particle filter tracker on ’sphere’ sequence with 50 particles, using different motion models (RW, NCV and NCA) and different values of parameter q (1, 5 and 100). All other parameters were kept consisted with the first experiment.

TABLE III: Tracking performance on ’sphere’ sequence with different motion models and parameter q

Parameter q	Model	Average Overlap	Failures	Tracking Speed (FPS)
1	RW	0.22	6	51
	NCV	0.40	6	48
	NCA	0.42	6	41
5	RW	0.41	6	49
	NCV	0.43	5	40
	NCA	0.48	5	36
100	RW	0.30	3	49
	NVC	0.31	4	46
	NCA	0.29	3	45

In terms of tracking performance, it seems that the NCA model generally performed better than the other models, especially with larger values of q. It’s also interesting to note that q generally leads to a decrease in the number of failures, but the increase in average overlap is not consistent across all models. Second thing we can note is that RW is the fastest and NCA is the slowest model of the tree models for all three q values. As for robustness, we can see that all models have a similar number of failures for each value of q, indicating that they are all fairly robust.

APPENDIX

Random Walk model

$$state = [x\ y]$$

$$F = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\Phi = e^{F\Delta T}$$

$$L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$R = r \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$Q = \int (\Phi(x) \cdot L) \cdot q \cdot (\Phi(x) \cdot L)^T dx$$

Nearly-Constant Velocity

$$state = [x\ y\ vx\ vy]$$

$$F = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\Phi = e^{F\Delta T}$$

$$L = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$R = r \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$Q = \int (\Phi(x) \cdot L) \cdot q \cdot (\Phi(x) \cdot L)^T dx$$

Nearly-Constant Acceleration

$$state = [x\ y\ vx\ vy\ ax\ ay]$$

$$F = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\Phi = e^{F\Delta T}$$

$$L = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$R = r \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$Q = \int (\Phi(x) \cdot L) \cdot q \cdot (\Phi(x) \cdot L)^T dx$$