Boundaries in Kalman and Particle Filter

Marius Oechslein

Faculty of Computer Science and Business Information Systems
University of Applied Sciences Würzburg-Schweinfurt
Würzburg, Germany
marius.oechslein@study.thws.de

Abstract—**The Kalman and the Particle filter are often used for estimating the real position of objects based on noisy observations. In this paper we investigate how non-linearity, like the the impact of a thrown ball with a wall, can be modelled with the Kalman and Particle filter.**
While the Kalman filter by definition is only applicable for linear systems, the Particle filter can also be used for non-linear systems. Therefore it is interesting to explore how robust the Particle filter is for these kinds of non-linearities.
To investigate the robustness of the Particle filter to non-linearities, the Particle filter is first implemented with low variance and many observations. This performance is then compared to an instance with a low amount of observations and an instance with high variance in the observations. Since a low amount of observations and high variance is known to have a negative effect on the performance of the particle filter, it will be interesting to see whether the particle model can still model the non-linearity.

Index Terms—**Particle Filter, Kalman Filter, Ball Throw, Boundary**

I. INTRODUCTION

The throw of a ball with noisy observations by a 2D camera can easily be modelled by a Kalman and Particle filter. Both the Kalman and the Particle filter are applicable for this problem due to the uncertainty in the noisy observations. The Kalman filter by definition is able to model linear systems with high accuracy and the advantage of computationally cheap calculations. The Particle filter is also able to model such a problem by recursively estimating the most likely position of the ball over time.
In a non-linear system like the ball hitting a wall, which changes the direction of the ball, the Kalman filter can no longer model the systems dynamics. Although the Kalman filter tries to model the system dynamics as best as it can, it will not be able to model this non-linearity due to its basic assumptions. The Particle filter on the other hand is able to model such non-linearities. But it is interesting to investigate how robust the Particle filter is to such non-linearities like the ball changing its direction when hitting a wall.
To investigate this robustness the amount of observations is decreased and the variance of the observations are increased. The point of interest is if the Particle filter will still be able to model the non-linear direction change after the ball hits the wall.
It was decided to only study the variance and the number of observations for the robustness, since the main point of interest was the direction change when the ball hit the wall. Next to the variance and the number of observations, also the

noise distribution and the starting assumed starting position has an effect on the robustness of the Particle filter. Since the main interest of this paper is the impact with the wall, it was decided to only test the Particle filter on different values for the variance and the number of observations. The noise was always assumed normally distributed and the starting positions were set to the real starting positions of the ball.

II. PHYSICS OF A BALL THROW

The physics of throwing a ball have to be implemented in the state transition to be able to estimate the position of the ball in the next time step. The x-position can be modelled like this:

$$x_{t+1} = x_t + v_x * \Delta t,$$
$$x, v_x \in \mathbb{R} \tag{1}$$

where the v_x is the ball's constant velocity in x-direction and x_t is the x-position of the ball at time step t.
The position and velocity in y-direction on the other hand are a bit more complex to calculate since they are affected by gravity.

$$y_{t+1} = y_t + v_y \Delta t - 0.5g\Delta t^2,$$
$$y, v_y \in \mathbb{R} \tag{2}$$

with g being the gravity constant at $g = 9.81$.
When hitting a wall, the only change in the physical motions is that the velocity in x-direction has to be negated. So at a after a certain time step where the wall is, the velocity in x-direction has to be updated to this $x^{vel} = -x^{vel}$. The velocity in y-direction is unchanged with the assumption that there is no fraction when the ball hits the wall and that the wall is exactly vertical. The equation for the x-position also remains unchanged.

III. KALMAN AND PARTICLE FILTER

In this section the Kalman and Particle filter are presented. For both the Kalman and Particle filter we define the hidden state as follows:

$$\mathbf{q} = (x, y, v_x, v_y) \tag{3}$$

Only the x and y-positions are incorporated in the observations, given by

$$\mathbf{o} = (x, y) \tag{4}$$

Both Kalman and Particle filters implement the following equation for the recursive density estimation:

$$p(\mathbf{q}_t|\langle\mathbf{o}\rangle_t) = p(\mathbf{o}_t|\mathbf{q}_t) \int p(\mathbf{q}_t|\mathbf{q}_{t-1})p(\mathbf{q}_{t-1}|\langle\mathbf{q}\rangle_{t-1})d\mathbf{q}_{t-1} \tag{5}$$

The Kalman filter does so by the Kalman gain algorithm [2]. One important property of the Kalman filter is its linearity assumption. Due to these assumptions non-linear systems cannot be modelled by the Kalman filter [2]. The Particle filter on the other hand can handle non-linearities and can be implemented by the Condensation algorithm [3].

IV. RESULTS

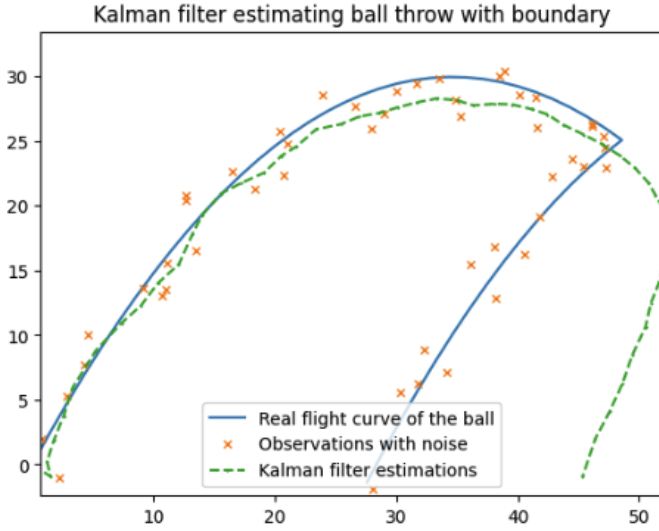First we look into the estimation of the Kalman filter.

Fig. 1. Kalman filter estimating ball throw with boundary. From own calculations.

In fig. 1 we can observe the Kalman filter trying to estimate the non-linear ball trajectory with a linear function, which results in bad estimations after the ball hit the wall, as expected.

TABLE I
KALMAN FILTER ESTIMATION ERRORS. BOUNDARY 50 M AWAY.

	Before hitting the wall	After hitting the wall
Error	1.2 m	17.73 m

Next we explore the estimations of the Particle filter with low variance and a reasonable amount of observations.

In fig. ?? we can see that the Particle filter models the non-linear ball throw very well, as expected.

To test the robustness of the Particle filter, we next drastically decrease the number of time steps. From fig. 3 it is clear that the Particle filter has problems estimating the positions with these circumstances. Although it is interesting to note that the general direction change can be modelled even though the observations are so few.
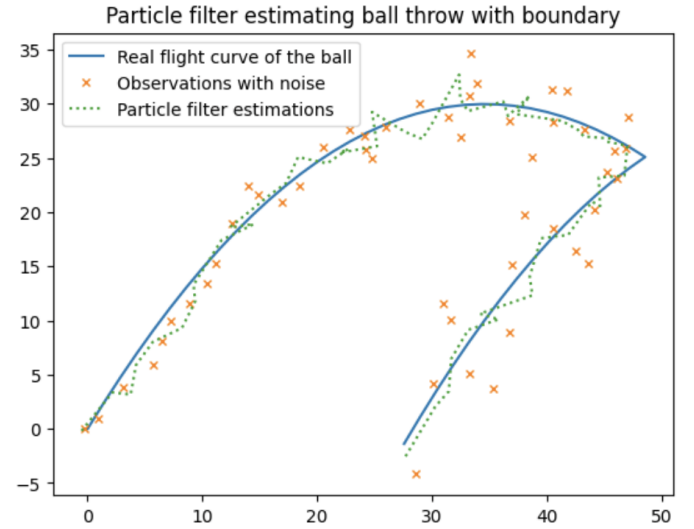

Fig. 2. Particle filter estimating ball throw with boundary. From own calculations.

TABLE II
PARTICLE FILTER ESTIMATION ERRORS. BOUNDARY 50 M AWAY. BOUNDARY KNOWN IN STATE TRANSITION.

	Before hitting the wall	After hitting the wall
Error	1.87 m	3.01 m

The second variable the Particle filter should be robust to is the variance of the observations to the real the trajectory. Although Particle filter also has a problem estimating the positions when the observations have high variance, as seen in fig. 4, the direction change after the ball hitting the wall is also generally modelled in this case.

V. DISCUSSION

1. We see that the kalman filter is not able to apply for this "non-linear" problem. Because Kalman filter expects linear relations. Maybe an advanced Kalman filter would be applicable.
2. We see that Particle filter with knowledge of boundary can handle it quite nicely.
3. What we see for Particle filter without knowledge of boundary.

VI. CONCLUSION

It would be interesting to see whether an advanced form of the kalman filter would be able to handle this non-linearity. - Naming which kind of kalman filter would be able to do it probably. Other than that we could look at the particle filter, additionally not knowing the system dynamics in the state transition at all (now it just doesn't know of the boundary).

REFERENCES

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply
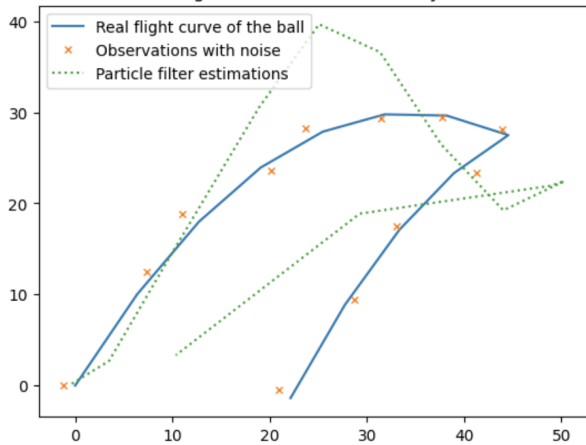
Fig. 3. Particle filter estimating ball throw with boundary and few observations. From own calculations.
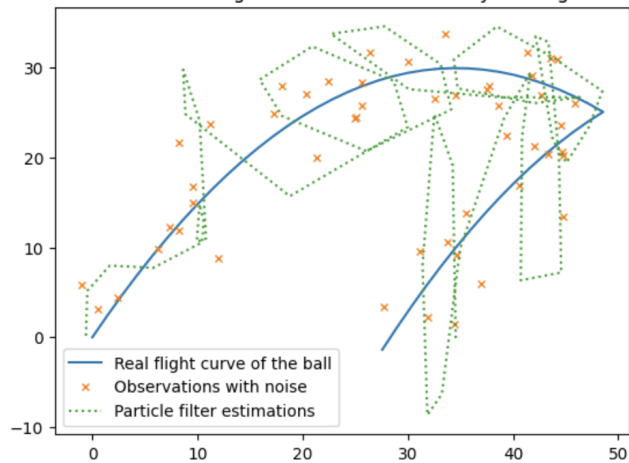

Fig. 4. Particle filter estimating ball throw with boundary and high variance. From own calculations.

to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at

REFERENCES

[1] Fetzer, T., Bullmann, M., Ebner, M., Kastner, S., Deinzer, F., & Grzegorzek, M. Interacting Multiple Model Particle Filter for Indoor Positioning Applications.

[2] Thrun, S. (2002). Probabilistic robotics. Communications of the ACM, 45(3), 52-57.

[3] Isard, M., & Blake, A. (1998). CONDENSATION–conditional density propagation for visual tracking. International journal of computer vision, 29(1), 5.