Portfolio 2 - Kalman Filter

Lennard Rose 5122787

July 10, 2022

Contents

1 Evaluation

The graphs and tables of this report were generated using the provided Jupyter-Notebook. The following subsections are named after the respective code-cells. To observe the influence on the estimates, a combination of all parameter pairs was performed. The noise was randomly chosen from a normal distribution around the original coordinate. In the following report, this is also referred to as "error range" (although it is not really a range) to match the implementation. This error range is described with σ as the scale and not with the variance σ^2 to match the numpy implementations annotations. The process noise covariance matrix Q as well as the observation noise covariance matrix R consisted only of ones and were scaled by a value, here referred to as Q and R value.

1.1 Comparing different timesteps(dt), error ranges, Q matrices and R matrices

The following graphs were the results of varying timestep width (dt) of either 0.1 or 0.5.
The error ranges, that determined the parameters of the normal distribution that the added noise was randomly drawn from, had a μ of 0 and a σ of either 0.5 or 1.
The ball was thrown from the coordinates x = and y = 10 with a velocity of 10 and an angle of 35°.
The Q as well as the R matrix had values set to either 0.01, 0.1 or 0.5.
There was no dropout.

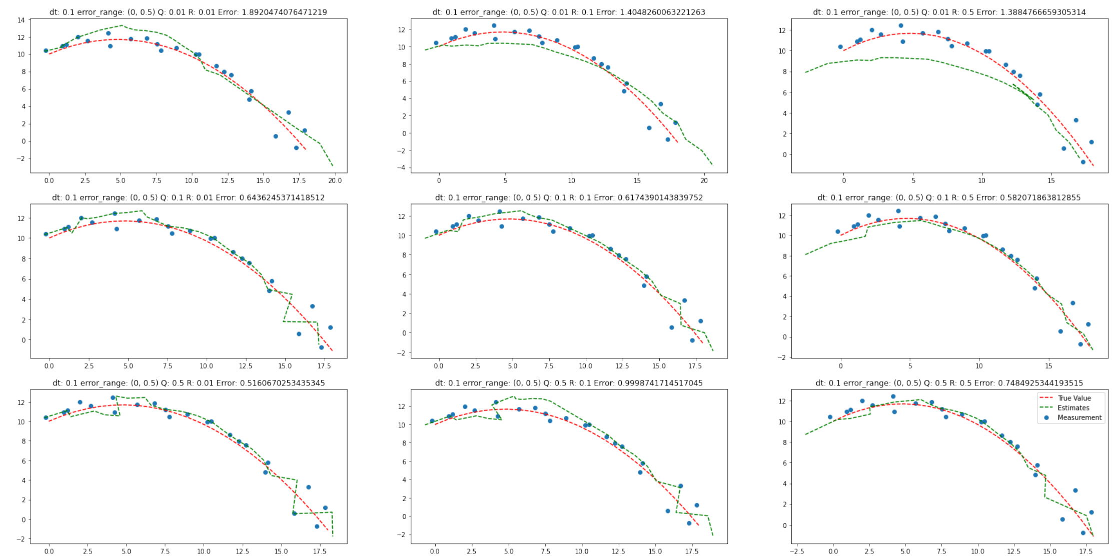

Figure 1: Result-graphs of the Kalman Filter with a dt of 0.1, an error range of $\mu = 0$ $\sigma = 0.5$ and varying values in the Q and R matrices.

dt	Error-range	Factor Q	Factor R	Error
0.1	(0, 0.5)	0.50	0.01	0.516067
0.1	(0, 0.5)	0.10	0.50	0.582072
0.1	(0, 0.5)	0.10	0.10	0.617439
0.1	(0, 0.5)	0.10	0.01	0.643625
0.1	(0, 0.5)	0.50	0.50	0.748493
0.1	(0, 0.5)	0.50	0.10	0.999874
0.1	(0, 0.5)	0.01	0.50	1.388477
0.1	(0, 0.5)	0.01	0.10	1.404826
0.1	(0, 0.5)	0.01	0.01	1.892047

Table 1: Result-table of the Kalman Filter with a dt of 0.1, an error range of $\mu = 0$ $\sigma = 0.5$ and varying values in the Q and R matrices.

The setup, presented in figure 1 and table 1.1 with a dt of 0.1 and an error range of $\mathcal{N}(\mu = 0, \sigma = 0,5)$. Smaller values in Q (0.01) did not seem to work very well. Higher values lead to a ”zigzag” pattern at the end of the trajectory curve, following the measurements that are far of the true path. This seems reasonable, because with the higher Q values the Kalman Filter ”assumes” less precision of its own estimation-process, giving more attention to the measurements.

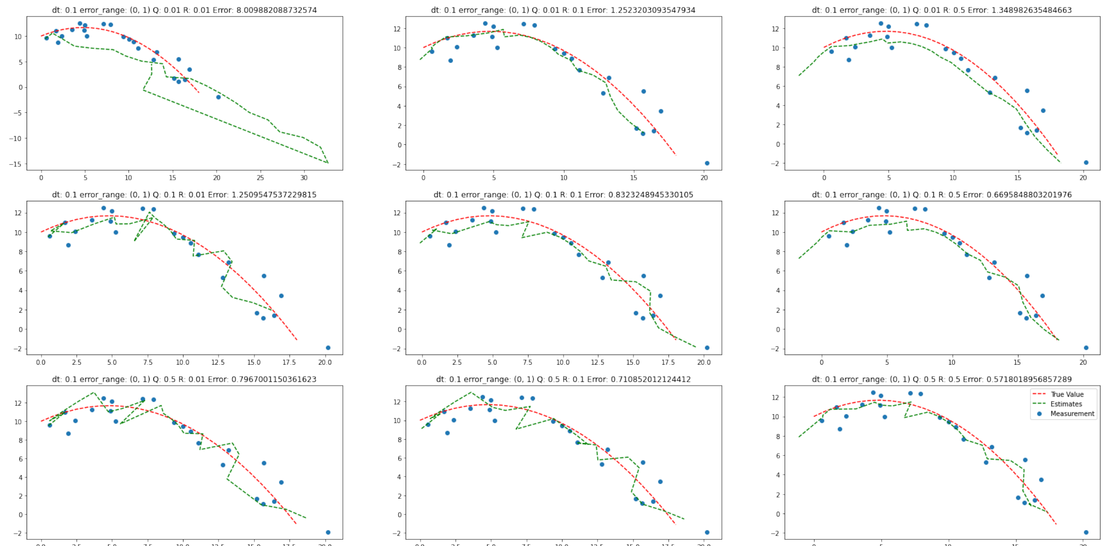

Figure 2: Result-graphs of the Kalman Filter with a dt of 0.1, an error range of $\mu = 0$ $\sigma = 1$ and varying values in the Q and R matrices.

dt	Error-range	Factor Q	Factor R	Error
0.1	(0, 1)	0.50	0.50	0.571802
0.1	(0, 1)	0.10	0.50	0.669585
0.1	(0, 1)	0.50	0.10	0.710852
0.1	(0, 1)	0.50	0.01	0.796700
0.1	(0, 1)	0.10	0.10	0.832325
0.1	(0, 1)	0.10	0.01	1.250955
0.1	(0, 1)	0.01	0.10	1.252320
0.1	(0, 1)	0.01	0.50	1.348983
0.1	(0, 1)	0.01	0.01	8.009882

Table 2: Result-table of the Kalman Filter with a dt of 0.1, an error range of $\mu = 0$ $\sigma = 1$ and varying values in the Q and R matrices.

In figure 2 and table 2, the error range has a σ is evaluated to 1. It can be seen, that the measurements are spreaded far more across the graph. The Kalman Filter has a lot more struggle to filter out the noise than in the previous graphs of figure 1. Higher values in R and Q resulted in lower errors.

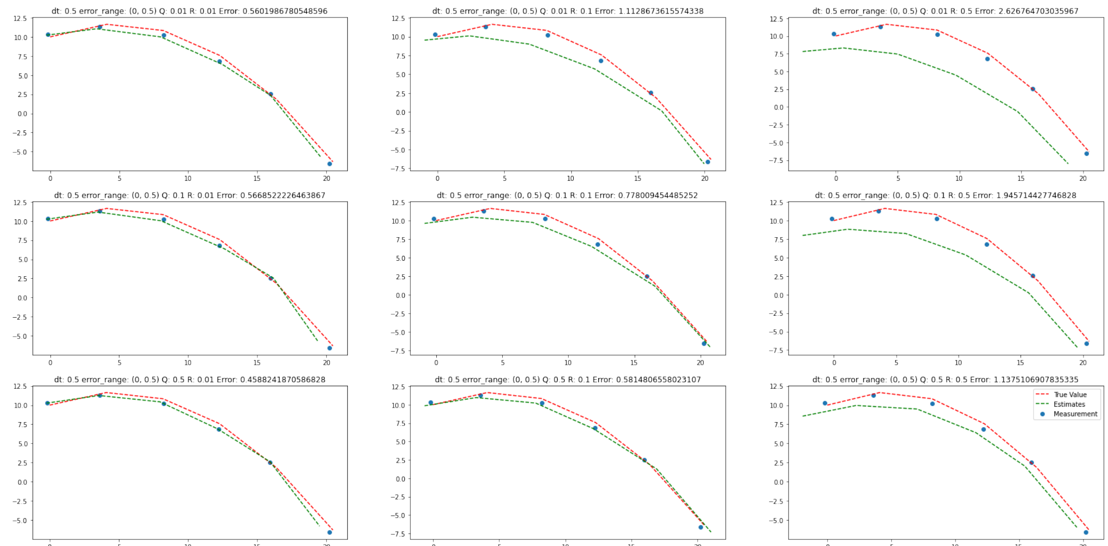

Figure 3: Result-graphs of the Kalman Filter with a dt of 0.5, an error range of $\mu = 0$ $\sigma = 0.5$ and varying values in the Q and R matrices.

dt	Error-range	Factor Q	Factor R	Error
0.5	(0, 0.5)	0.50	0.01	0.458824
0.5	(0, 0.5)	0.01	0.01	0.560199
0.5	(0, 0.5)	0.10	0.01	0.566852
0.5	(0, 0.5)	0.50	0.10	0.581481
0.5	(0, 0.5)	0.10	0.10	0.778009
0.5	(0, 0.5)	0.01	0.10	1.112867
0.5	(0, 0.5)	0.50	0.50	1.137511
0.5	(0, 0.5)	0.10	0.50	1.945714
0.5	(0, 0.5)	0.01	0.50	2.626765

Table 3: Result-table of the Kalman Filter with a dt of 0.5, an error range of $\mu = 0$ $\sigma = 0.5$ and varying values in the Q and R matrices.

In figure 3 and table 1.1, it can be seen that row-wise (every row has the same Q-value) the error increases along with the increasing R-values. Column-wise (every column has the same R-value), the error is decreasing with the increasing Q-values. With an error-range of a scale of only 0.5 around the mean 0, the measurements are reasonably accurate, resulting in good performance with low values in the observation noise matrix R.

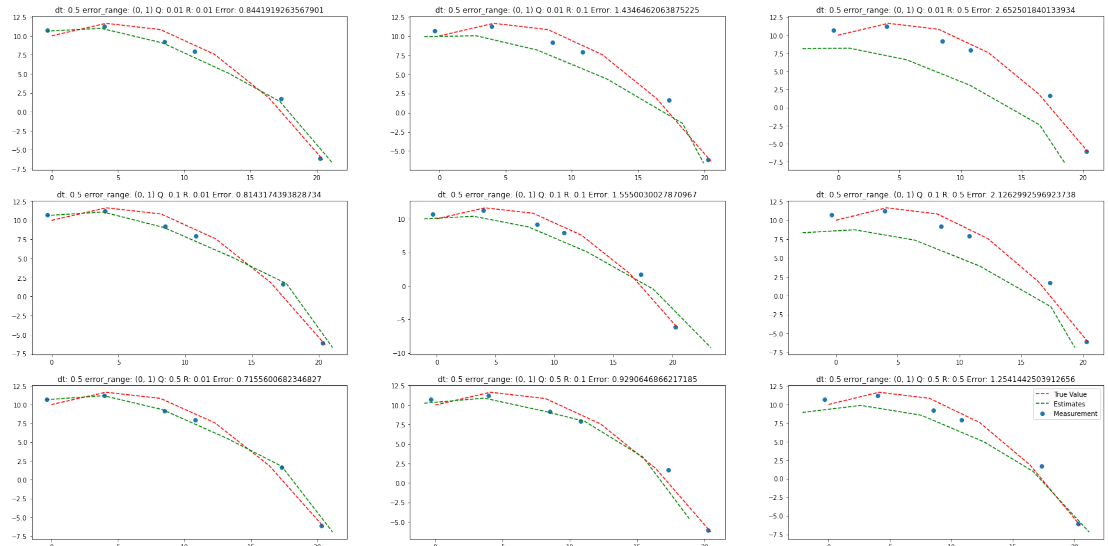

Figure 4: Result-graphs of the Kalman Filter with a dt of 0.5, an error range of $\mu = 0$ $\sigma = 1$ and varying values in the Q and R matrices.

dt	Error-range	Factor Q	Factor R	Error
0.5	(0, 1)	0.50	0.01	0.715560
0.5	(0, 1)	0.10	0.01	0.814317
0.5	(0, 1)	0.01	0.01	0.844192
0.5	(0, 1)	0.50	0.10	0.929065
0.5	(0, 1)	0.50	0.50	1.254144
0.5	(0, 1)	0.01	0.10	1.434646
0.5	(0, 1)	0.10	0.10	1.555003
0.5	(0, 1)	0.10	0.50	2.126299
0.5	(0, 1)	0.01	0.50	2.652502

Table 4: Result-table of the Kalman Filter with a dt of 0.5, an error range of $\mu = 0$ $\sigma = 1$ and varying values in the Q and R matrices.

Results as visualized in figure 4 and table 4, show a much worse performance compared to the previous setup with the lower error-range. This was to expect, even though that doubling the scale of the error, did not result in an twice as high error in the estimates. The best performing Q and R value combinations are quite similar to those in table 1.1.

1.2 Comparing different Q and R matrices on dropout over 6 timesteps (10-15)

The following graphs and tables are the results of the Kalman Filter with variable Q and R matrices processing the trajectory curve with measurement-dropout over 6 timesteps (10 to 15). The ball was thrown from the coordinates x = 0 and y = 10 with a velocity of 10 and an angle of 35°. Timestep-value dt was 0.1. The normal distribution that the added noise was randomly drawn from, had a μ of 0 and a σ of 0.2. The Q and R matrices had all their entries set to either 0.1, 0.5 or 1.

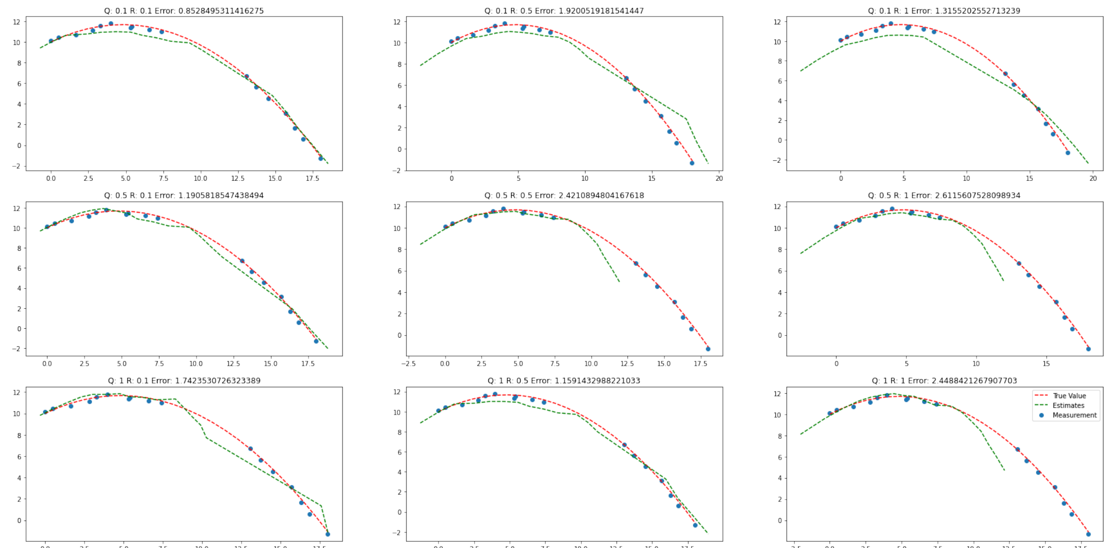

Figure 5: Result-graphs of the Kalman Filter with dropout and varying values in the Q and R matrices.

Factor Q	Factor R	Error	Ratio Q/R
0.1	0.1	0.852850	1.0
1.0	0.5	1.159143	2.0
0.5	0.1	1.190582	5.0
0.1	1.0	1.315520	0.1
1.0	0.1	1.742353	10.0
0.1	0.5	1.920052	0.2
0.5	0.5	2.421089	1.0
1.0	1.0	2.448842	1.0
0.5	1.0	2.611561	0.5

Table 5: Result-table of the Kalman Filter with dropout and varying values in the Q and R matrices.

The only setting of the R matrix in figure 5 and table 5 were no "losing" of the trajectory-path occurs is with values of 0.1 (the smallest). It is counterintuitive that most of the time the smaller the Q and R values the better the performance. There is no pattern in the Ratio of Q to R values to explain this behaviour. I assumed that because of the dropout in the measurement, higher R values would lead to better performance.

1.3 Comparing different throwing parameters (startposition, angle, velocity)

The following graphs are the results of varying throwing parameters. The ball was thrown from the coordinates x being 0 or 0.5 and y with values of either 1 or 5. Velocity-value was set to 10 or 20. The throwing angle was set to 10 or 50 degrees. Timestep-value dt was 0.1. The normal distribution that the added noise was randomly drawn from, had a μ of 0 and a σ of 0.2. The Q and R matrices had all values set to 0.1.

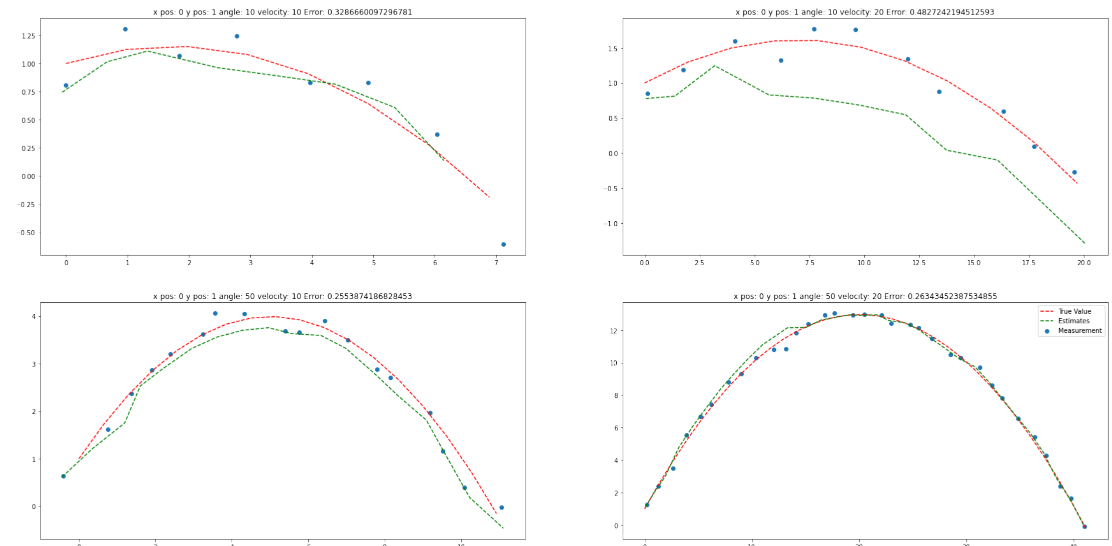

Figure 6: Results of the Kalman Filter with startpositon x = 0, y = 1, a throwing angle of 10 or 50 degrees and velocity of 10 or 20.

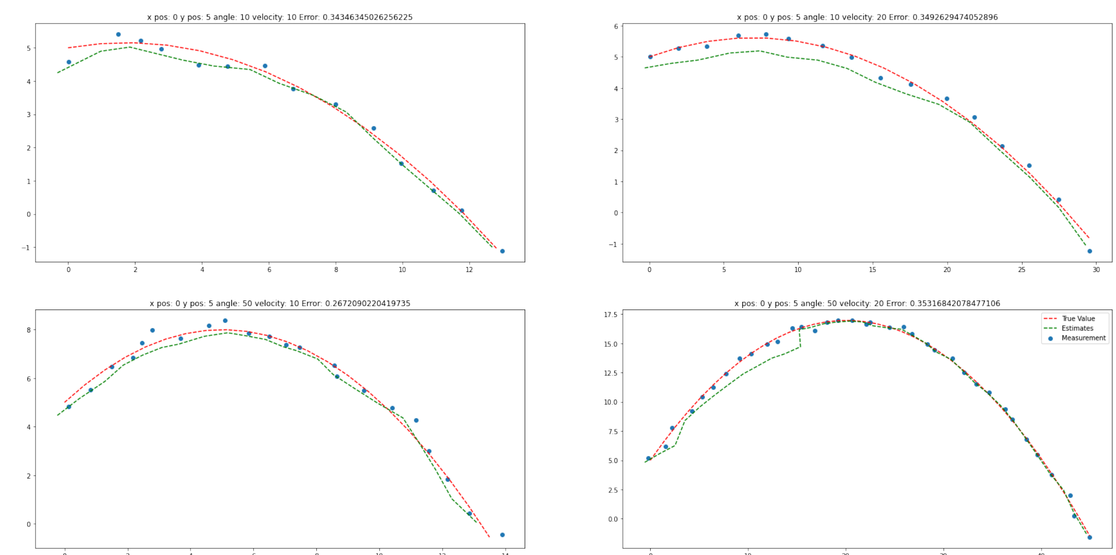

Figure 7: Results of the Kalman Filter with startpositon x = 0, y = 5, a throwing angle of 10 or 50 degrees and velocity of 10 or 20.

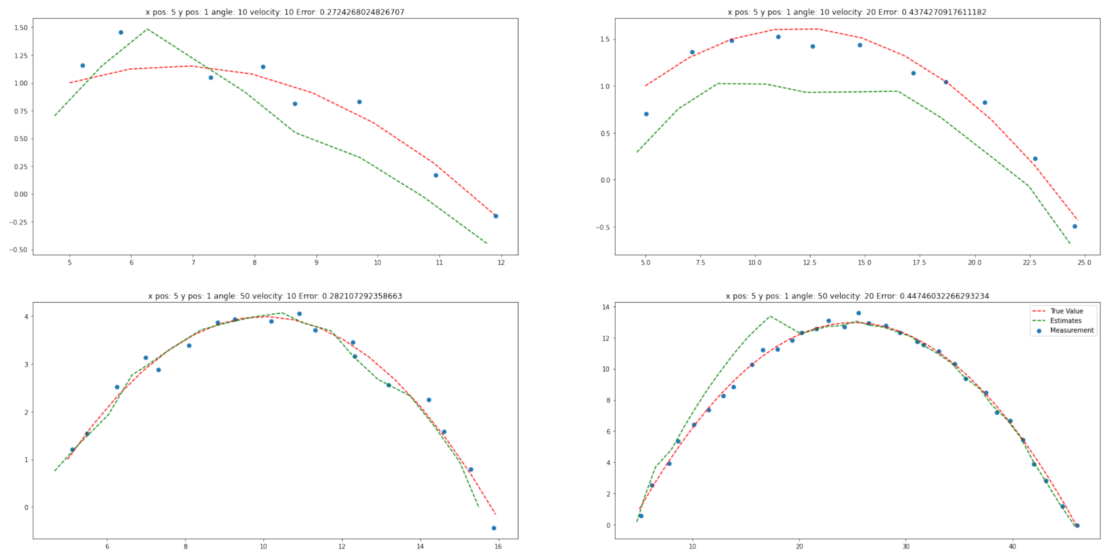

Figure 8: Results of the Kalman Filter with startpositon x = 5, y = 1, a throwing angle of 10 or 50 degrees and velocity of 10 or 20.

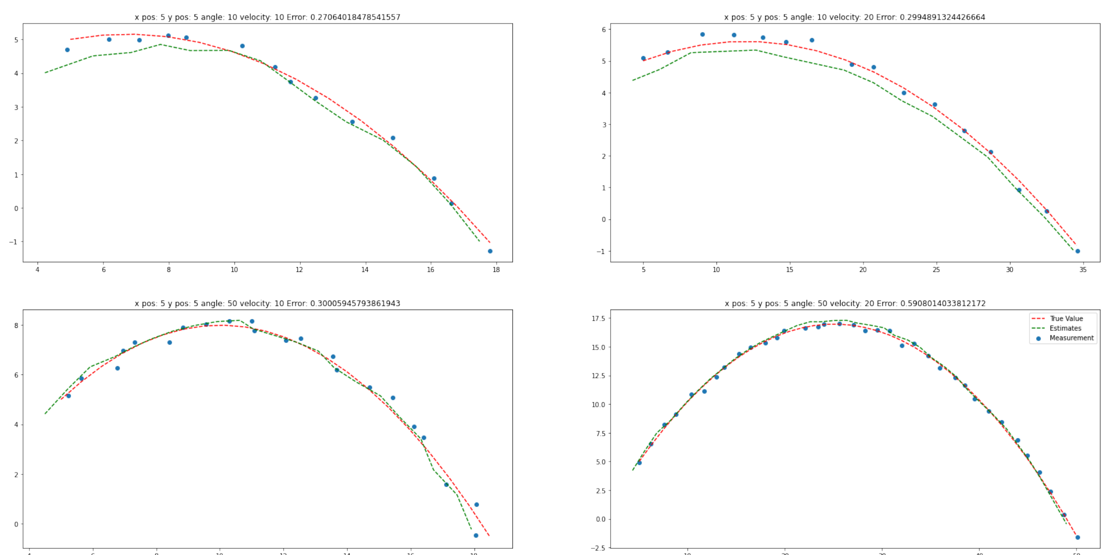

Figure 9: Results of the Kalman Filter with startpositon x = 5, y = 5, a throwing angle of 10 or 50 degrees and velocity of 10 or 20.

pos x	pos y	angle	velocity	Error
0.0	1.0	50.0	10.0	0.255387
0.0	1.0	50.0	20.0	0.263435
0.0	5.0	50.0	10.0	0.267209
5.0	5.0	10.0	10.0	0.270640
5.0	1.0	10.0	10.0	0.272427
5.0	1.0	50.0	10.0	0.282107
5.0	5.0	10.0	20.0	0.299489
5.0	5.0	50.0	10.0	0.300059
0.0	1.0	10.0	10.0	0.328666
0.0	5.0	10.0	10.0	0.343463
0.0	5.0	10.0	20.0	0.349263
0.0	5.0	50.0	20.0	0.353168
5.0	1.0	10.0	20.0	0.437427
5.0	1.0	50.0	20.0	0.447460
0.0	1.0	10.0	20.0	0.482724
0.0	1.0	10.0	20.0	0.482724
5.0	5.0	50.0	20.0	0.590801

Table 6: Table comparing the different throwing parameters of the above graphs

All graphs in this section show different trajectory lines. Different x-values only shift the trajectory-line on the horizontal axis. Some combinations (e.g. y=5, angle=50, velocity=20) resulted in longer "air-time" and more measurement points. Although expected, this did not elevate the performance of the Kalman Filter significantly.

1.4 Comparing different state value initializations

The following graphs are the results of varying initial states. The ball was thrown from the coordinates x = 0 and y = 10. Velocity was set to 10. The throwing angle was 35 degrees. Timestep-value dt was 0.1. The normal distribution that the added noise was randomly drawn from, had a μ of 0 and a σ of 0.3 for the horizontal-axis and 0.5 for the vertical-axis. The Q and R matrices had all values set to 0.1. The vector ($\in \mathbb{R}^{4x1}$) representing the initial state was set to either [0 0 0 0], [1 1 1 1] or [1 2 3 4].

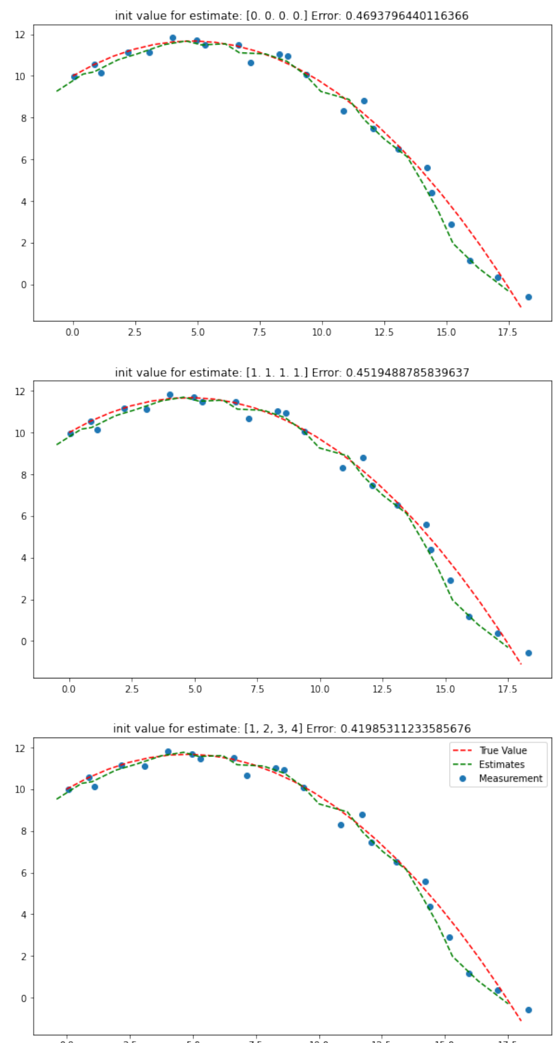

Figure 10: Result-graphs of the Kalman Filter with varying state variable initializations.

Init Values	Error
[0 0 0 0]	0.469380
[1 1 1 1]	0.451949
[1 2 3 4]	0.419853

Table 7: Result-table of the Kalman Filter with varying state variable initializations.

The above graphs 10 of the Kalman Filter with varying state vector initializations as well as table 7 show no significant improvements with alternative values. Even if one could argue that values that are higher than 0, are probably closer to the true values, the best and the worst error in the table just differ by 0.05.

2 Kalman Gain

The Kalman Gain is represented by:

$$K_t = \frac{\Sigma_t^{'} C_t^T}{C_t \Sigma_t^{'} C_t^T + R_t} \tag{1}$$

$\Sigma_t^{'}$ is the estimate covariance matrix, showing the error in the estimate.
C_t is the observation model, a matrix to map the values of our state to those of the measurement/observation o_t (In our example the x and y coordinates, velocity of x and y are left out).
R is the Measurement Noise Covariance Matrix depending on the expected noise of the measurement. If the measurements are done by e.g a sensor with a high tolerance-range, values in R should be big.

The $\Sigma_t^{'} C_t^T$ in the nominator as well as the $C_t \Sigma_t^{'} C_t^T$ in the the denominator of formula 2 represent the error in the estimates. R_t in the denominator is the assumed error in the Measurement.
With this in mind and for further explanation we simplify the equation 1 to:

$$K_t = \frac{Error_{Estimation}}{Error_{Estimation} + Error_{Measurement}} \tag{2}$$

With big error in the estimates and small error in the measurement, K approaches 1. With this, estimations are made with bigger influence of the measurements.
With small values in the estimation error and big values in the measurement error, K approaches 0. This is a direct consequence of the estimations becoming stable. When making further estimations, they are considered more.
With the values of K being in the interval [0, 1] the Kalman Filter scales the influence of the measurement o_t and the predicted state value $\mu_t^{'}$ on the state value μ_t as seen in equation 3.

$$\mu_t = \mu_t^{'} + (K_t o_t - K_t C_t \mu_t^{'}) \tag{3}$$

Further, the Kalman Gain also adjust the estimate covariance matrix Σ_t to the errors, as seen in 4.

$$\Sigma_t = (I - K_t C_t)\Sigma_t^{'} \tag{4}$$

The Kalman Gain should become smaller and smaller over the iterations. This means that the estimates are getting closer to the true values (becoming stable) without being dependent on the measurements, which will continue to vary.
In summary, the Kalman Gain is a weight whose value is determined by considering the error of the measurement and of the estimation. It is used to control the Kalman filter with respect to the measurement and its own state.