Student Name: Riccardo Periotto
Student ID: 19980915-T633

EL2320 - Applied Estimation
Lab 1

Part I

Linear Kalman Filter

1. What is the difference between a 'control' u_t, a 'measurement' z_t and the state x_t? Give examples of each

   All the three concepts relate to the robot environment and its interaction with it. The state is what directly characterizes the environment in which the robot acts and, depending on how it is defined and the type of application, can contain different information (e.g. the internal status of the robot, the position of external objects within the environment, etc). The control and the measurement are related to how the robot interacts with the environment (i.e. the state). A control signal u_t is something that modifies the state of the robot, while a measurement z_t is a signal that provides information about the robot's state. For example, a possible control signal in industrial robotics is the torque the motor have to apply to a revolut joint to turn an arm. On the other hand, an example of a measurement in the same field could be the flow of data from the distance sensors (e.g. those that identifies a person and stop the robot for security reasons).

2. Can uncertainty in the belief increase during an update? Why (or not)?

   To answer this question it is important to underline that in the filters we saw there are two types of update: the control update (also called prediction) and the measurement update (also called correction). After the control update, the algorithm obtains a posterior $\overline{bel}(x_t)$ that incorporates the uncertainty of the state transition probability $p(x_t|x_{t-1}, u_t)$. After this first step, the uncertainty usually increases. The second update instead exploits the information of the measurement z_t to correct the posterior belief just calculated, obtaining $bel(x_t)$. This last step usually decreases the uncertainty, using the information bought by the measurement (which is, most of the time, a proiection of the state). Mathematically, this can be seen considering the instruction for calculating the uncertainty: $\Sigma_t = (I - K_t C_t)\overline{\Sigma}_t$. Given how we defined the initial uncertainty and the Kalman gain, the new uncertainty Σ_t is always lower than the intermediate $\overline{\Sigma}_t$.

   After an entire iteration of the algorithm, how the uncertainty changes with respect to the previous time step depends on the combination of the two steps: there is no way to predict if it will increase or decrease beforehead.

3. During update what is it that decides the weighing between measurements and belief?

   The trade-off between measurements and belief derives from the relative size of Q and Σ, respectively the prediction and measurement models uncertainty.

4. What would be the result of using a too large covariance (Q matrix) for the measurement model?

The larger the covariance matrix Q, the lower the trust of the filter in the measurement. In this case, the final value returned by the algorithm will be more related to the prediction step and the convergence to a low uncertainty will be slower.

5. What would give the measurements an increased effect on the updated state estimate?

   A lower uncertainty matrix Q, which will increase the Kalman gain K_t

6. What happens to the belief uncertainty during prediction? How can you show that?

   The uncertainty during prediction is calculated as:

   $$\overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$$

   In general, this increases the uncertainty from time $t-1$ to time t, but it depends on the value of A_t and the noise R_t.

7. How can we say that the Kalman filter is the optimal and minimum least square error estimator in the case of independent Gaussian noise and Gaussian priori distribution? (Just describe the reasoning not a formal proof.)

   The filter is optimal in the sense that it uses all the information at its disposal in the best way possible. More formally, it minimizes the expected variance in the estimated state (which corresponds to saying that it is the minimum least square error estimator). Given the conditions in which it is applied, both the prediction and the update steps of the filter result in Gaussian distributions. Computing the same Gaussian moltiplication performed by the filter and zeroing their derivatives, it is possible to prove that the Kalman filter modes used by the filter on the resulting distributions are those optimal in both the steps.

   In other words, the Kalman Filter does not make any approximation, it uses the real distributions and relies on linear algebra.

8. In the case of Gaussian white noise and Gaussian priori distribution, is the Kalman Filter a MLE and/or a MAP estimator?

   The Kalman Filter calculates its belief at time t starting from the one at time $t-1$. This last belief plays the role of a prior distribution over the state, with its parameter. For this reason, starting with a belief distribution, the filter is a MAP estimator. The only case in which the filter can be considered an MLE estimator is when there is no prior belief, for example when initializing the algorithm.

Extended Kalman Filter

9. How does the extended Kalman filter relate to the Kalman filter?

   The Extended Kalman filter overrides the assumption that the transition and the measurement probabilities follow a linear distribution, allowing them to be governed by nonlinear functions, usually identified with the letters g and h respectively. Referencing to the math formulations:

   $$x_t = g(u_t, x_{t-1}) + \epsilon_t$$

   $$z_t = h(x_t) + \delta_t$$

   The practical difference for handling this new condition is that the EKF has to linearise the functions before using them. The rest of the mechanism behind the filter remains the same, but the obtained result is just an approximation of the true belief.

10. Is the EKF guaranteed to converge to a consistent solution?

    No; in general there are no guarantees that the EKF converges to a consistent solution. The quality of the result mainly depends on the local non-linearity of the functions involved.

11. If our filter seems to diverge often can we change any parameter to try and reduce this?

    The parameters we can change to reduce the divergence are the two model uncertainties Q and R. We can't know beforehand whether the problem originates in the system model or the measurement model, but it is possible to figure this out by changing the relative size of the two covariance matrices and making tests.

Localization

12. If a robot is completely unsure of its location and measures the range r to a know landmark with Gaussian noise, what does its posterior belief of its location $p(x, y, \theta|r)$ look like? So a formula is not needed but describe it at least

    Assuming that the position of the landmark is known, the shape of the posterior belief is a donut around the position of the landmark itself, with a peak at a distance r and spread following the Gaussian noise of the measure. The distribution of the belief over θ instead, is a uniform distribution over its domain (as we don't know anything about it).

13. If the above measurement also included a bearing how would the posterior look?

    The uncertainty on its Cartesian position x and y would be the same as before. What would change is the uncertainty on the third dimension θ, which would be governed by the uncertainty of the corresponding measurement, instead of being uniform.

    To notice that now the Cartesian coordinates and the steer angle are now correlated.

14. If the robot moves with relatively good motion estimation (prediction error is small) but a large initial uncertainty in heading how will the posterior look after traveling a long distance without seeing any features?

    It will look like a circular sector, which will be the larger the initial uncertainty (in term of degrees). The belief will be concentrated around a ring whose radius corresponds to the predicted distance travelled from the starting point, which is assumed to be accurate.

15. If the above robot then sees a point feature and measures range and bearing to it how might the EKF update go wrong?

    The main problem is that in this scenario, with large uncertainty, the shape of the prior is not a Gaussian distribution. The linearization will produce an update along the direction of the approximated Gaussian, which is much different from the real distribution along arc shape.

Part II - Matlab Excercises

Warm up problem with Standard Kalman Filter

1. What are the dimensions of ϵ_k and δ_k? What parameters do you need to define in order to uniquely characterize a white Gaussian?

   For this example, ϵ_k is a 2×1 vector, while δ_k is a scalar. In general, ϵ_k has dimension $N \times 1$, with N being the dimension of the state; δ_k has dimension $M \times 1$, with M the number of measurements of the system.

   A white Gaussian is characterized only by its covariance Σ, as its mean is zero by definition. For scalar cases, the covariance collapses in the simple variance.

2. Make a table showing the roles/usages of the variables(x, xhat, P, G, D, Q, R, wStdP, wStdV, vStd, u, PP). To do this one must go beyond simply reading the comments in the code to seeing how the variable is used. (hint some of these are our estimation model and some are for simulating the car motion).

Variable	Usage/role
x	The real state of the system (in the simulation)
xhat	The estimated state of the system given by the Kalman Filter
P	The estimated covariance matrix for the system state belief
G	Matrix that maps the process noise on the state
D	Matrix that maps the measurement noise on the measure itself; for this system it is a single scalar
Q	Covariance matrix on the measurement
R	Covariance matrix on the state
wStdP	Standard deviation of the Gaussian noise on the simulated position; it is used both for simulating the system and for defining the covariance matrix
wStdV	Standard deviation of the Gaussian noise on the simulated velocity; as before, it has a double usage in the code
vStd	Standard deviation of the Gaussian noise on the measurement of the. Also this one has a double application in the code
u	Control input signal (i.e. the acceleration affecting the system velocity)
PP	Vector containing the uncertainties calculated by the Kalman algorithm at each step

Table 1: system variables description

   Note that Q and R are defined oppositely to how we usually use them.

3. Please answer this question with one paragraph of text that summarizes broadly what you learn/deduce from changing the parameters in the code as described below. Chose two illustrative sets of plots to include as demonstration. What do you expect if you increase/decrease the covariance matrix of the modeled (not the actual simulated) process noise/measurement noise 100 times (one change in the default parameters each time) for the same underlying

system? Characterize your expectations. Confirm your expectations using the code (save the corresponding figures so you can analyze them in your report). Do the same analysis for the case of increasing/decreasing both parameters by the same factor at the same time. (Hint: It is the mean and covariance behavior over time that we are asking about.)

When dealing with these parameters, what matters is the relative size of the two covariance matrices. Increasing the covariance matrix of the process noise R results in greater Kalman gain, which gives more importance to the measurements with respect to the process model when calculating the estimated state. On the other hand, increasing the covariance matrix of the measurement noise Q results in a smaller Kalman gain, thus reducing how much the filter trusts the measurement and making it to rely more on the process model.

When decreasing the covariances, the reasoning above is inverted.

An additional consideration is that when the Kalman gain is small, the system is subject to a slower convergence rate (assuming to start with a large uncertainty).

The sets of figures in the following confirm what anticipated with the theory above. In particular, Figure 1, Figure 2 and Figure 3 report the results obtained after multiplying matrix R by 100. The first notable difference with respect to the result obtained with the default parameters is the oscillation affecting the expected velocity. The filter trusts the measurements model more, but it does not have any direct velocity measurement. Consequently, the uncertainty over the velocity is relevant, while the one affecting the position remains low. Whatsmore, as anticipated, the filter gives more importance to the measurements and so the Kalman gains are greater.

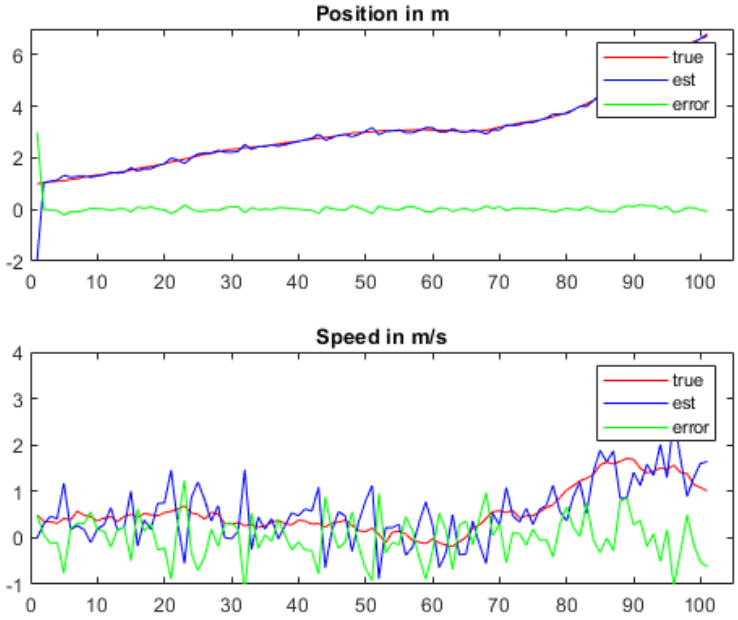

Figure 1: states result when increasing the process covariance matrix R

Figure 4 and Figure 5 report the state variables and the covariances of the system when leaving R to its given value and multiplying matrix Q by 100. In this case, the position prediction remains fine, while the velocity is smoother than before. This behaviour is usually

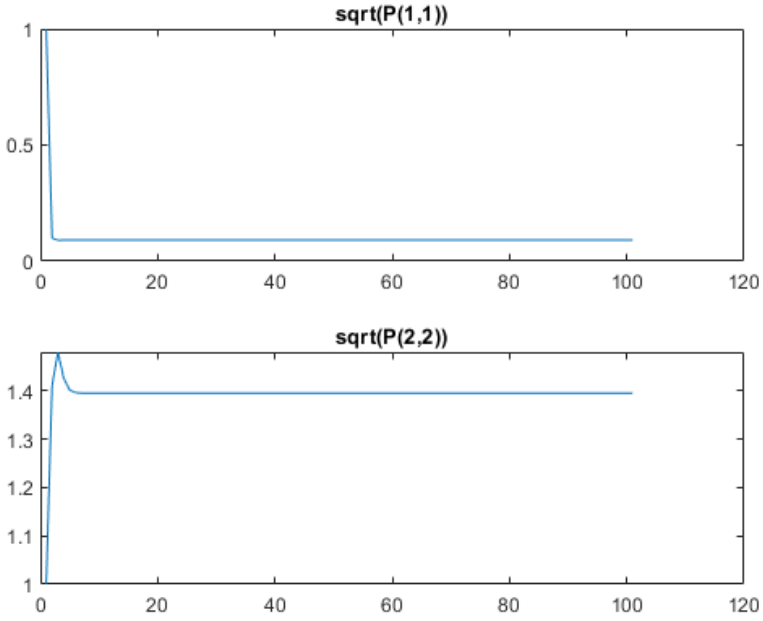

Figure 2: covariance result when increasing the process covariance matrix R

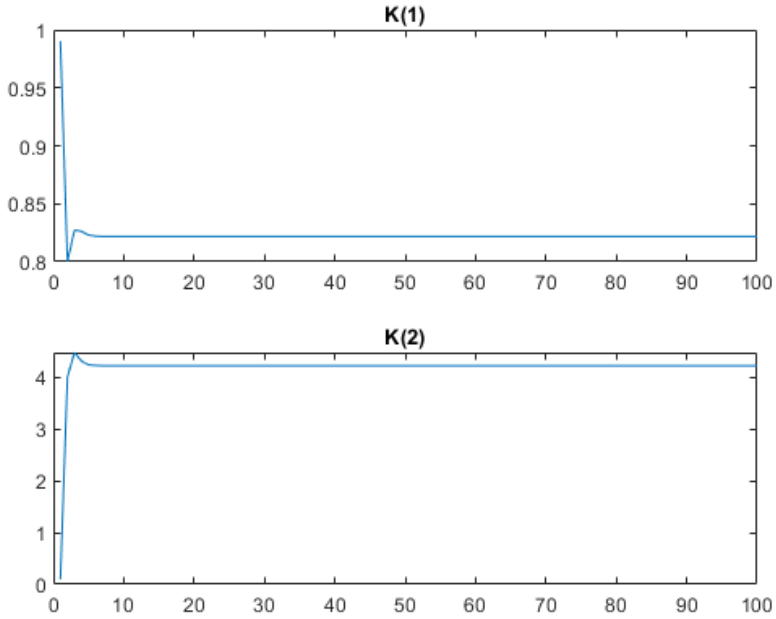

Figure 3: gain result when increasing the process covariance matrix R

described saying that the convergence of the system is slower, which in practice means that our model has less trust in the measurements than for the prediction. The uncertainty over the velocity remains the same as in the default case because we don't have any measure of the velocity variable. Things are different for the position, which is affected by the model and the measurement uncertainty, even if this last one is weighted using a lower Kalman gain, as shown in Figure 6.

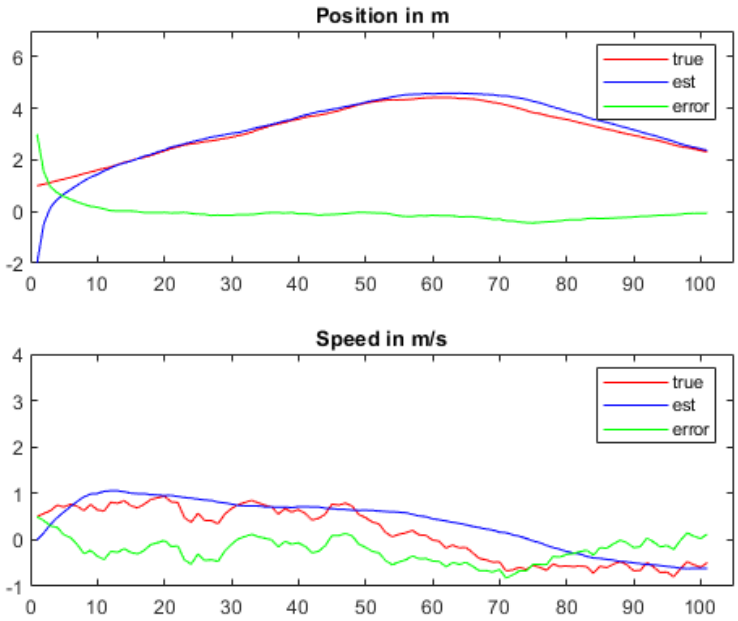

Figure 4: states result when increasing the measurement covariance matrix Q

When increasing both the covariance matrices R and Q, the estimated state and the Kalman gain of the filter do not change. This is again because what matters when calculating the gain is the ratio between the two parameters, which is not affected when acting in the requested way. Figure 7 and Figure 9 confirm the theoretical expectation. What changes with respect to the default case is the uncertainty of the estimation P: as shown in Figure 8, P is now definitely greater. This is a consequence of having more uncertainties in the system models.

4. How do the initial values for P and xhat affect the rate of convergence and the error of the estimates (try both much bigger and much smaller)?

   Starting with a large P corresponds to being uncertain about the initial system state. The direct implication is to have a more uncertain prior and a greater Kalman gain (i.e. the filter trusts the measurement more than the system model, because it is aware of the uncertainty in the state). However, with a large value of K, the system converges rapidly in the initial phase, as reported in Figure

   On the other hand, starting with a small P corresponds to being sure about the initial position. In this case, measurements are considered less by the filter, and indeed the applied gain is smaller. As shown Figure , the model has a lower convergence rate.

   Finally, a smaller initial guess on the state xhat would change neither the convergence rate nor the error significantly. Contrarily, a large initial value for the state makes the system to

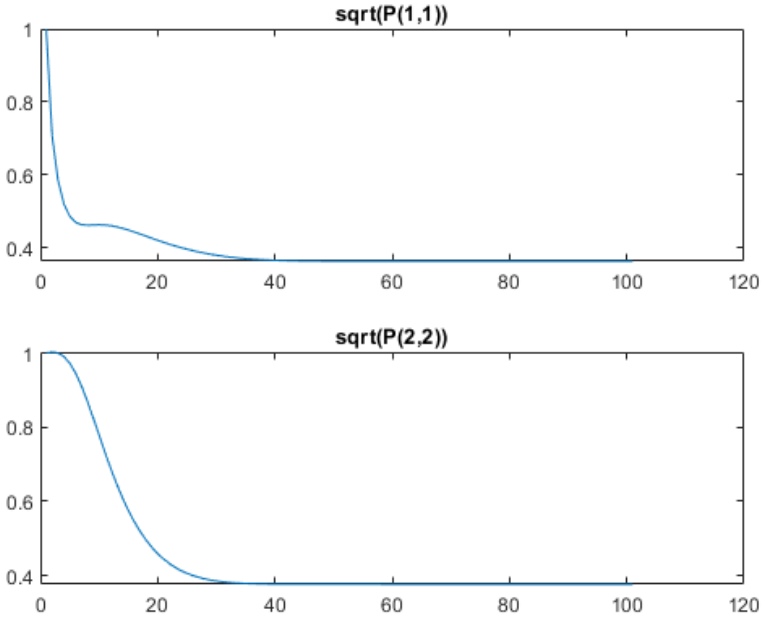

Figure 5: covariance result when increasing the measurement covariance matrix Q

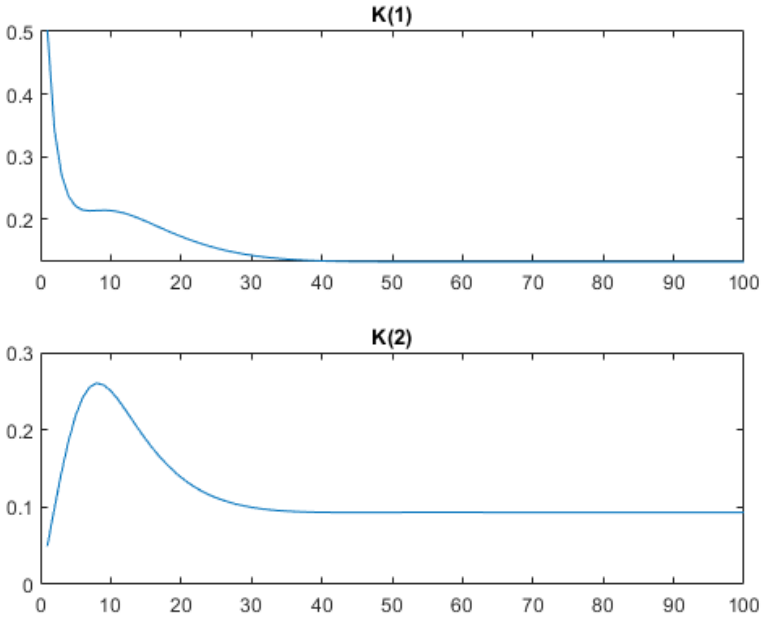

Figure 6: gain result when increasing the measurement covariance matrix Q

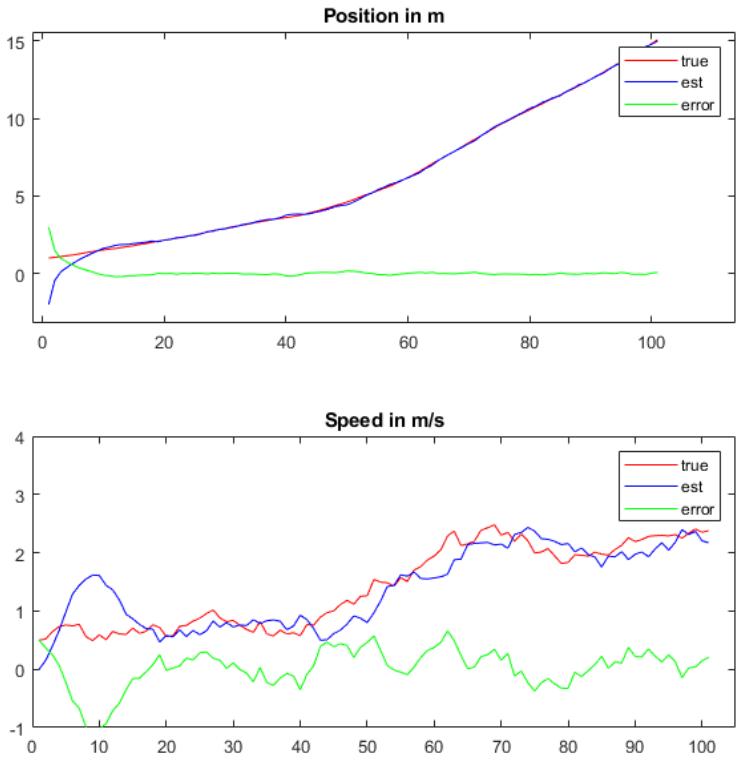

Figure 7: states result when increasing both the process and measurement covariance matrices R and Q

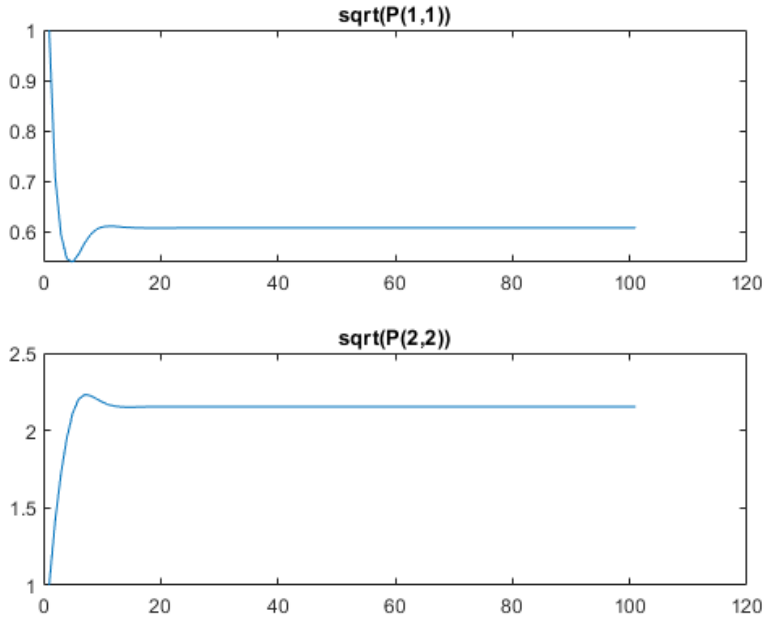

Figure 8: covariance result when increasing both the process and measurement covariance matrices R and Q

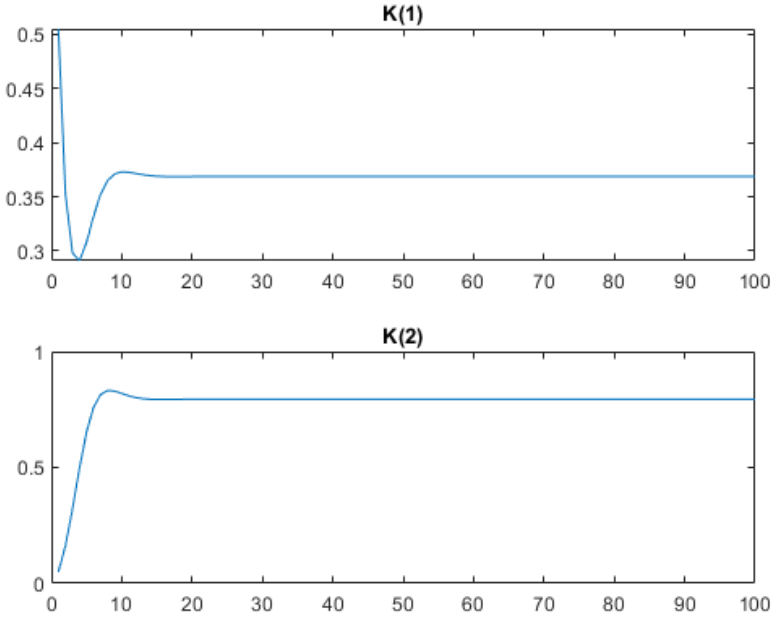

Figure 9: gain result when increasing both the process and measurement covariance matrices R and Q

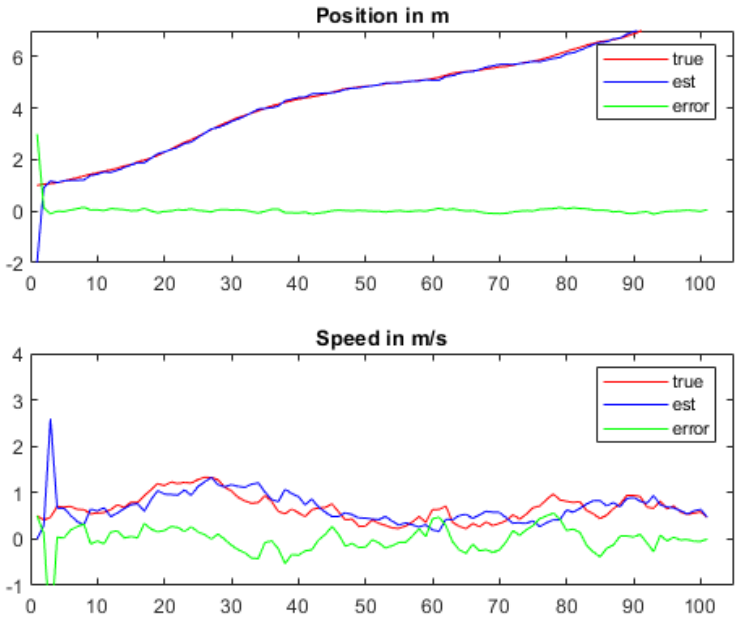

Figure 10: large initial uncertainty P (multiplied by 100)

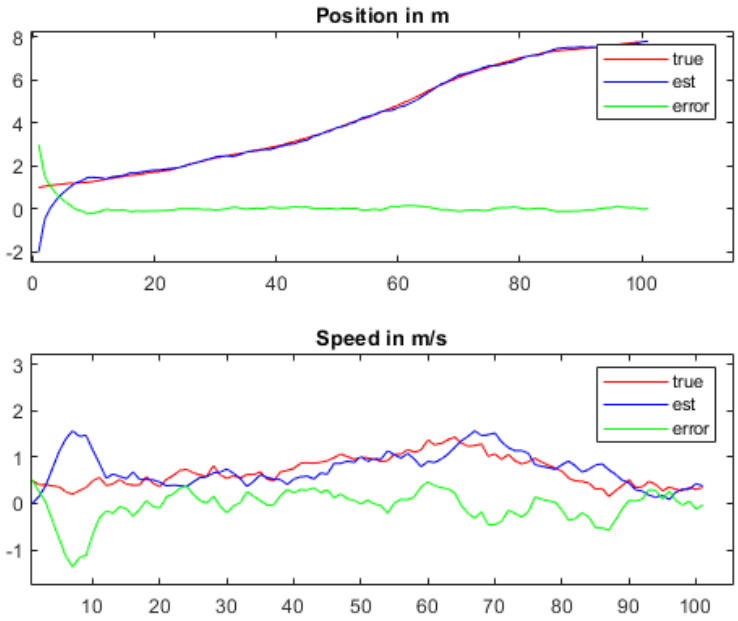

Figure 11: small initial uncertainty P (multiplied by 0.01)

converge slower and with a higher error, as shown in the figure Figure 13

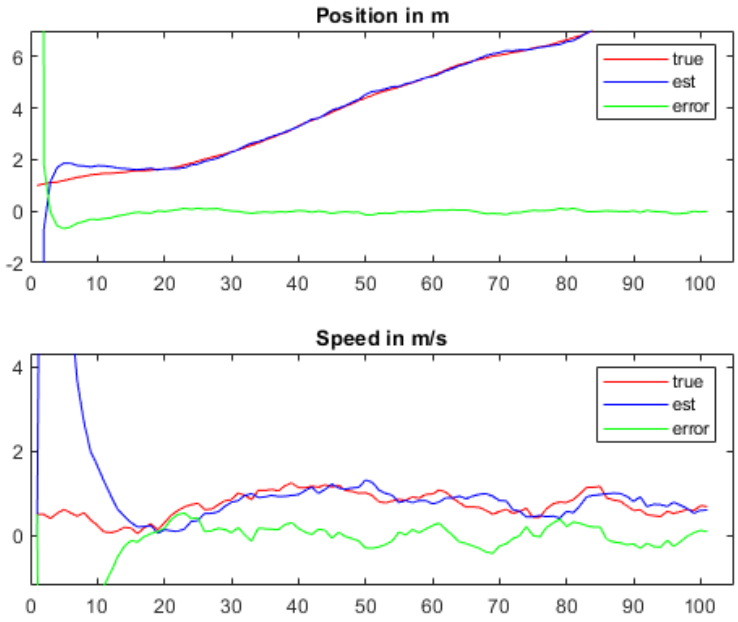

Figure 12: large (wrong) initial guess on the state

Main problem: EKF Localization

Considering a first order assumption and Bayesian update as follows:

$$\begin{cases} p(\mathbf{x_t}|\mathbf{u_{1:t}}, \mathbf{z_{1:t}}, \bar{\mathbf{x}}_\mathbf{0}, M) & = \eta p(\mathbf{z_t}|\mathbf{x_t}, M) \int p(\mathbf{x_t}|\mathbf{u_t}, \mathbf{x_{t-1}}) p(\mathbf{x_{t-1}}|\mathbf{z_{1:t-1}}, \mathbf{u_{1:t-1}}, \bar{\mathbf{x}}_\mathbf{0}, M) \mathbf{dx_{t-1}} \\ p(\mathbf{x_0}|\bar{\mathbf{x}}_\mathbf{0}) & = \delta(\mathbf{x_0} - \bar{\mathbf{x}}_\mathbf{0}) \end{cases} \tag{2}$$

$$\begin{cases} bel(\mathbf{x_t}) & = p(\mathbf{x_t}|\mathbf{u_{1:t}}, \mathbf{z_{1:t}}, \bar{\mathbf{x}}_\mathbf{0}, M) = \eta p(\mathbf{z_t}|\mathbf{x_t}, M) \overline{bel}(\mathbf{x_t}) \\ \overline{bel}(\mathbf{x_t}) & = p(\mathbf{x_t}|\mathbf{u_{1:t}}, \mathbf{z_{1:t-1}}, \bar{\mathbf{x}}_\mathbf{0}, M) = \int p(\mathbf{x_t}|\mathbf{u_t}, \mathbf{x_{t-1}}) bel(\mathbf{x_{t-1}}) \mathbf{dx_{t-1}} \\ bel(\mathbf{x_0}) & = p(\mathbf{x_0}|\bar{\mathbf{x}}_\mathbf{0}) = \delta(\mathbf{x_0} - \bar{\mathbf{x}}_\mathbf{0}) \end{cases} \tag{3}$$

5. Which parts of (2) and (3) are responsible for prediction and update steps?

   In the set of equations (2), both the prediction and the update steps are defined in the first line. In set (3) instead, the line responsible for the prediction is the first, while the one responsible for the update is the second.

6. In the maximum likelihood data association, we assumed that the measurements are independent of each other. Is this a valid assumption? Explain why

   The independence assumption is valid. As shown by the model, each measurement only depends on the landmark position, the robot position, and the time. In each step, the last

two variables are known, making the set of measurements conditionally independent. This same assumption is adopted in chapter 6.6 and equation 7.16 in the reference book.

7. Given the threshold $\lambda_M = X_2^{-2}(\delta_M)$, what are the bounds for δ_M? How does the choice of δ_MM affect the outlier rejection process? What value do you suggest for λ_M when we have reliable measurements all arising from features in our map, that is all our measurements come from features on our map? What about a scenario with unreliable measurements with many arising from so called clutter or spurious measurements

   The final threshold has to be a probability value, so the bounds of δ_M are 0 and 1. When k is 2, the inverse of the chi-square function is monotonically increasing. Given this property, the larger the input value δ_M, the larger the resulting threshold. As anticipated, the aim of using a threshold is to identify overconfident measurements. With a reliable measurement model, the threshold can be large because the probability of having outliers is small. With unreliable model instead, spurious measurements are more common, and so a lower threshold enables the algorithm to reject them.

8. Can you think of some down-sides of the sequential update approach? Hint: How does the first [noisy] measurements affect the intermediate results?

   The down-side is making the update considering the noise of every single measurement. A better solution would be the one proposed next, in which the noise decreases by averaging between different measurements. A better explanation is given in the Batch Update section.

9. How can you modify the algorithm for EKF Localization with Batch update shown in Figure 13 to avoid redundant re-computations?

   In the algorithm, $\hat{z}_{t,j}$, $H_{t,j}$, and $S_{t,j}$ are computed at every iteration for each observation, even if they depend only on the landmark. An improvement is to calculate these values before running the nested cycles.

10. What are the dimension of $\overline{\nu}_t$ and $\overline{H}_t$ in the algorithm? What were the corresponding dimensions in the sequential update algorithm? What does this tell you?

    The dimension of $\overline{\nu}_t$ is $2n \times 1$, while $\overline{H}_t$ has dimension $2n \times 3$, where n is the number of inliers. In the sequential update algorithm, the dimensions were 2×1 and 2×3 respectively. This is a direct consequence of being able to consider multiple measurements in a single update, increasing the robustness. However, this also increases the computation, as the matrices involved in the calculation are larger.

Data sets

This last section describes and contains the experiments performed on the three datasets provided. For each experiment, there is a brief description, the set of parameters used, and the plots showing the results.

1. Dataset 1

   For this dataset, the laser scanner has an accuracy of (1 cm, 1 degree), and the odometry information has an un-modeled noise of approximately 1 cm and 1 degree per time step. Running the simulation leaving the default parameters respects the requested constraints over the errors, as shown in the plot of Figure 14

Algorithm 4 EKF Localization with Batch update for the i^{th} time step

for all Observations i in z_t **do**
  for all Landmarks j in M **do**

$$
\begin{aligned}
\hat{\mathbf{z}}_{t,j} &= \mathbf{h}(\bar{\boldsymbol{\mu}}_t, M, j) \\
H_{t,j} &= H(\bar{\boldsymbol{\mu}}_t, M, j, \hat{\mathbf{z}}_{t,j}) \\
S_{t,j} &= H_{t,j}\bar{\Sigma}_t(H_{t,j})^T + Q \\
\boldsymbol{\nu}_t^{i,j} &= \mathbf{z_{t,i}} - \hat{\mathbf{z}}_{\mathbf{t,j}} \\
D_t^{i,j} &= (\boldsymbol{\nu}_t^{i,j})^T(S_{t,j})^{-1}\boldsymbol{\nu}_t^{i,j} \\
\psi_t^{i,j} &= \det(2\pi S_{t,j})^{-\frac{1}{2}} exp[-\tfrac{1}{2}D_t^{i,j}]
\end{aligned}
$$

  end for

$$
\begin{aligned}
\hat{c}_t^i &= \arg\max_j \ \psi_t^{i,j} \\
\hat{o}_t^i &= D^{i,c_t^i} \geq \lambda_M \\
\bar{\boldsymbol{\nu}}_t^i &= \boldsymbol{\nu}_t^{i,\hat{c}_t^i} \\
\bar{H}_{t,i} &= H_{t,\hat{c}_t^i}
\end{aligned}
$$

end for

{For inlier indices $1, ..., n$}

$$
\begin{aligned}
\bar{\boldsymbol{\nu}}_t &= \begin{bmatrix} (\bar{\boldsymbol{\nu}}_t^1)^T & (\bar{\boldsymbol{\nu}}_t^2)^T & \dots & (\bar{\boldsymbol{\nu}}_t^n)^T \end{bmatrix}^T \\
\bar{H}_t &= \begin{bmatrix} (\bar{H}_{t,1})^T & (\bar{H}_{t,2})^T & \dots & (\bar{H}_{t,n})^T \end{bmatrix}^T \\
\bar{Q}_t &= \begin{bmatrix} Q & 0 & 0 & 0 \\ 0 & Q & 0 & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & Q \end{bmatrix}_{(2n \times 2n)} \\
K_t &= \bar{\Sigma}_t(\bar{H}_t)^T(\bar{H}_t\bar{\Sigma}_t(\bar{H}_t)^T + \bar{Q}_t)^{-1} \\
\boldsymbol{\mu}_t &= \bar{\boldsymbol{\mu}}_t + K_t\bar{\boldsymbol{\nu}}_t \\
\Sigma_t &= (I - K_t\bar{H}_t)\bar{\Sigma}_t
\end{aligned}
$$

Figure 13: EKF Localization with Batch update

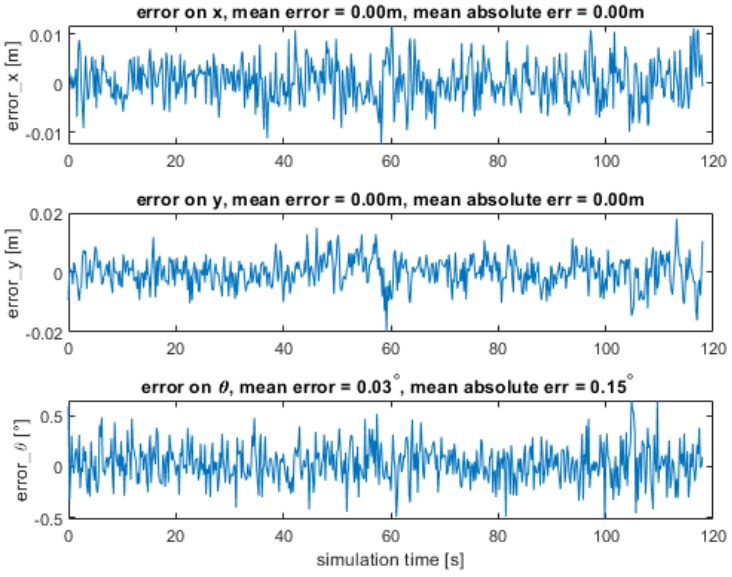

Figure 14: state errors over simulation time, experiment 1

2. Dataset 2

In this second scenario, the measurements have undergone a white Gaussian with a standard deviation of 0.2 (m, rad) and contain many outliers. The covariance matrix of the process and measurement noise adopted to detect and reject these outliers are the following:

$$R = \begin{pmatrix} 0.01^2 & 0 & 0 \\ 0 & 0.01^2 & 0 \\ 0 & 0 & (\frac{2\pi}{360})^2 \end{pmatrix}, Q = \begin{pmatrix} 0.2^2 & 0 \\ 0 & 0.2^2 \end{pmatrix}$$

With these matrices and by setting the outlier detection at 10%, the simulation results in a mean absolute error lower than 0.06 (m, rad) on all dimensions, as shown in Figure 15.

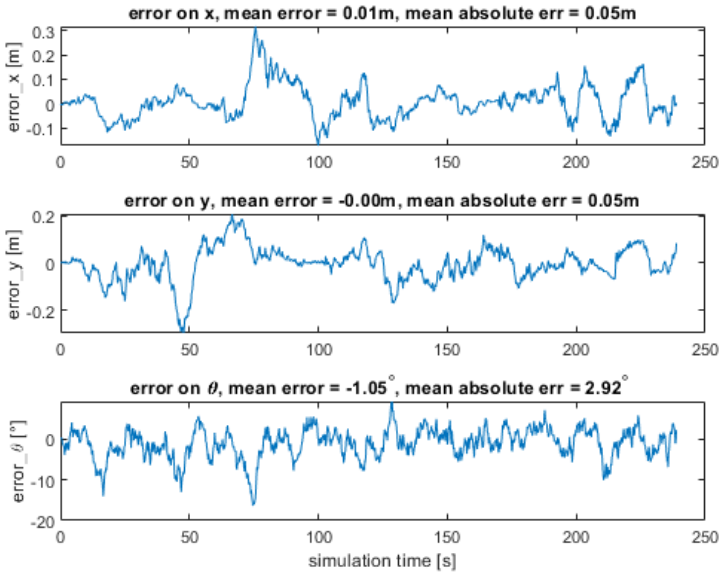

Figure 15: state errors over simulation time, experiment 2

Running the same simulation by setting the detection of outliers to the default value of 20%, the results are worse. The reason for this behavior may be that by setting a higher threshold, the filter not only excludes outliers but also measurements containing the correct information.

3. Dataset 3

The last experiment does not include any odometry information. As suggested, to compensate for it, the solution is to consider a big standard deviation for the process noise. The covariance matrices R and Q in this last example are the following:

$$R = \begin{pmatrix} 1^2 & 0 & 0 \\ 0 & 1^2 & 0 \\ 0 & 0 & 1^2 \end{pmatrix}, Q = \begin{pmatrix} 0.1^2 & 0 \\ 0 & 0.1^2 \end{pmatrix}$$

Figure 16 and Figure 17 respectively show the errors and covariances plots when running the algorithm with the sequential update algorithm, while Figure 18 and Figure 19 when using the batch update algorithm. As reported in the plots, the results of the batch-updated algorithm are far better, with a mean absolute error lower than 0.1 (m, rad) on all dimensions. The uncertainty of the estimate is two orders of magnitude higher than that of the simulation performed on the first dataset, as one might expect given the absence of odometry. On

the other hand, it is comparable with the uncertainty of the second dataset, in which the measurements were sporadic.

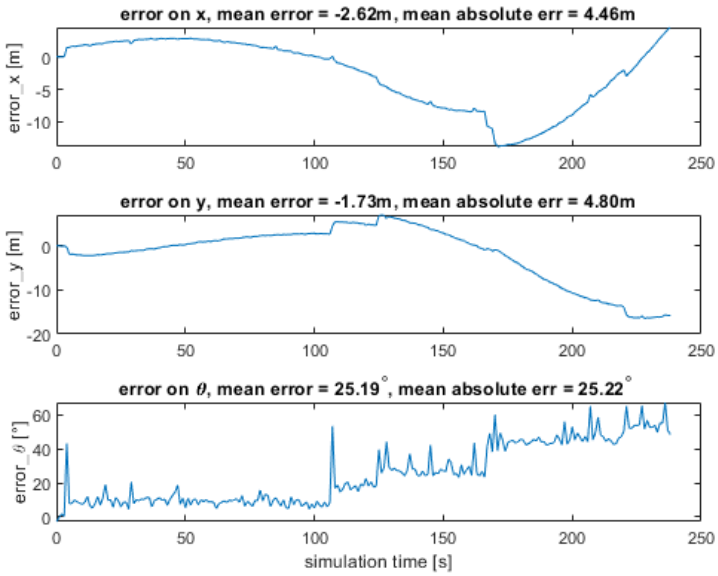

Figure 16: state errors with sequential update algorithm, experiment 3

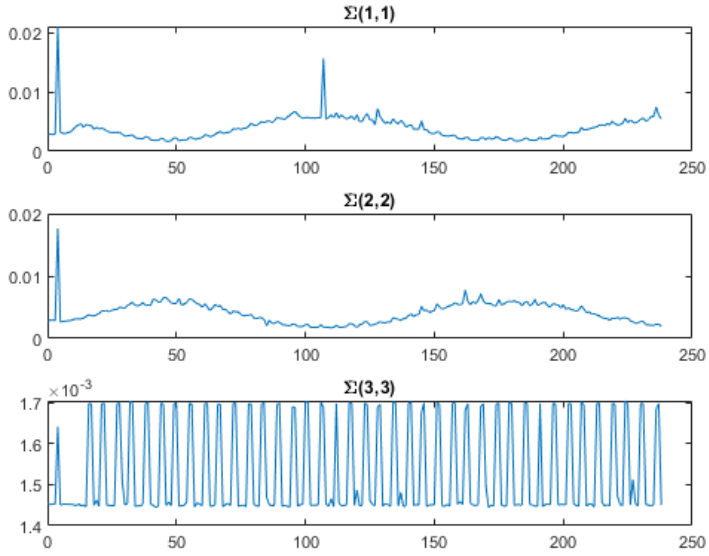

Figure 17: covariances with the sequential-update algorithm, experiment 3

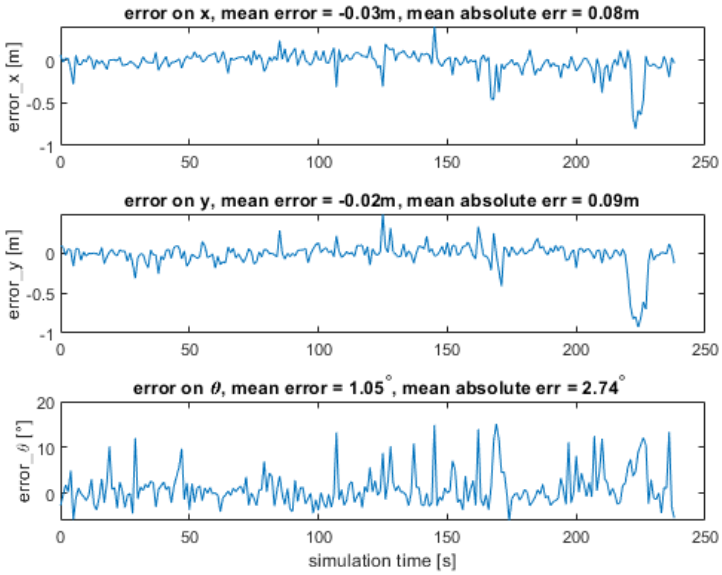

Figure 18: state errors with the batch-update algorithm, experiment 3

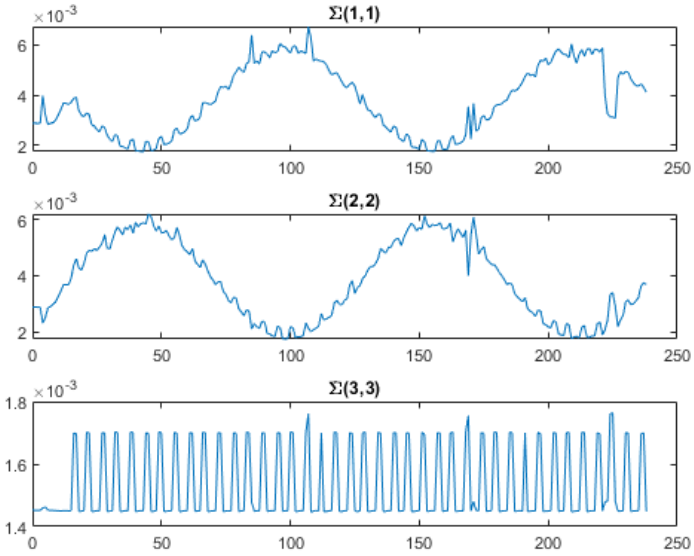

Figure 19: covariances with the batch-update algorithm, experiment 3