PART I - Preparatory Questions-Answers

1. **State**: It is a collection of variables that describe the system under consideration. Eg. System under consideration – Robot and its environment. One can describe this system using the robot's position and velocity relative to the environment. **Control** (u_t): A signal that changes the state of the system. Eg. U_t = MOVE; this command causes the robot to move thereby changing its position and velocity, i.e. state of the system. **Measurement** (z_t): The process of gathering information about the state of the system. Eg. A GPS gathers information about the robot's location.
2. Belief uncertainty during update decreases, because the uncertainty in the predict phase is weighted by Kalman gain and subtracted from itself to generate the updated uncertainity. Irrespective of large uncertainty (small Kalman gain) or small uncertainty (Large kalman gain) in measurement (Q matrix), there is some Kalman gain to weight the predict phase uncertainty. The above statements are true provided the Kalman gain is a positive value.
3. The Kalman gain weighs between measurement and belief. It species the degree to which the measurement (z_t) is incorporated into the new belief.
4. As the measurement uncertainty (Q matrix) becomes large, the Kalman gain reduces thereby resulting very little update in the update step.
5. A low uncertainty in the measurement model (low Q matrix) and large uncertainty in system model (R matrix) will give the measurement an increased effect on the new estimate.
6. Belief uncertainty at t increases during prediction because the uncertainty of the state space system model gets added to the uncertainty of belief at t-1.
7. The kalman gain aims at minimising the mean-squared error. This gain is used in the calculation of new estimate. Thus, the new mean obtained is minimized mean square error estimate. So to prove that Kalman filter isn't optimal we need to find a better μ. $\int_{-\infty}^{\infty}(x-\mu_{better})^2 G(x,\mu,\Sigma)dx$
   By minimising the above equ-ation, it can be found that $\mu_{better} = \mu$
8. MLE is a special case of MAP. If the prior is a constant (uniform distribution) then MAP is named as MLE. In the Kalman filter derivation, the prior is not a constant but a Gaussian distribution. Hence, Kalman filter is a MAP estimator.
9. EKF calculates an approximation to true belief. It inherits the belief representation from LKF, this belief is a Gaussian (like LKF) but approximate, not exact as in LKF. The EKF applies KF to the linearized nonlinear system.
10. EKF can converge to a wrong estimate, can diverge or become inconsistent due to:
    - Non-linear transforms may distort the Gaussian
    - The noise might not be a Gaussian as assumed
    - As the non-linearity of the system dynamics increase, the chance of divergence increases.
11. In case of divergence, the covariance can be adjusted.
12. Uniform distribution with $-\pi < \theta < \pi$. The position will be like a ring. Gaussian on radius and uniform distribution on angle.$x = r\ cos\varphi; y = r\ sin\varphi$
13. Knowing the bearing along with the range will have a Gaussian on heading with a completely correlated covariance.
14. It will look like a cresent or arc or c shape. Heading is correlated with position along the arc.

15. The linearization will produce a straight line which cannot move the estimate along the curved crescent. The Gaussian will not be able to represent the crescent shape and hence will diverge.

PART II

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

   εk will have the dimensions of the state vector x and δk will have the same dimensions are measurement vector. White Gaussians are defined by two moments- μ (mean), which is equal to zero, and some 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.**

   Hint: Some of these variables are part of our estimation model and some are for simulating the car motion.

Variable	Roles/usage
x	State vector containing position and velocity of the car
xhat	Estimated state vector (after applying Kalman filter)
P	Covariance of the state vector
G	An identity matrix, which gets multiplied with simulation noise matrix R. This is done in order to enable the addition of simulation noise during the prediction step.
D	Constant that gets multiplied with the measurement noise vector Q. This is done in order to enable the addition of measurement noise in the Kalman gain calculation step.
Q	Measurement noise matrix
R	Simulation noise matrix
wStdP	Standard deviation of the simulated position state variable
wStdV	Standard deviation of the simulated velocity state variable
vStd	Standard deviation of the measured position
u	Control variable, in this case acceleration/deceleration value.
PP	Matrix to store all P(covariance of the state vector) values during each simulation run.

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. Choose 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 actually simulated) process noise/measurement noise by a factor of 100 (only one change per run)? 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.

Answer:

Quantity changed	Expectation
R increased	Increases the uncertainty in state variable velocity, as velocity is not measured and incorporated into the estimate. In case of the state variable position, increasing R results in an increased uncertainty in the prediction stage but countered in the update step by the Kalman gain because the measured position is incorporated into estimated position.
R decreased	Decreasing the R value, decreases the uncertainty in the estimate, especially the velocity state variable.
Q increased	Kalman gain decreases. This affects the extent to which the measurement affects the estimate. The uncertainty of the estimate increases.
Q decreased	Kalman gain increases. The uncertainty of the estimate decreases. Especially for position state variable, as the extent to which the less uncertain (low Q) measurement gets incorporated into the estimate increases, making the estimate less uncertain.
R,Q increased	Increasing the R,Q, results in increased uncertainty in the estimate. This is because the measurement is noisy as well as the model.
R,Q decreased	Decreasing R and Q implies that we have a less noisy model and measurement, which will eventually result in less uncertain estimate.

Original:
Standard deviation of error in position (second half): 0.050247m
Standard deviation of error in velocity (second half): 0.206917m/s
R increased:
Standard deviation of error in position (second half): 0.074306m
Standard deviation of error in velocity (second half): 0.461836m/s
R decreased:
Standard deviation of error in position (second half): 0.135364m
Standard deviation of error in velocity (second half): 0.295539m/s
Q increased:
Standard deviation of error in position (second half): 0.081738m
Standard deviation of error in velocity (second half): 0.217796m/s
Q decreased:
Standard deviation of error in position (second half): 0.075858m
Standard deviation of error in velocity (second half): 0.457890m/s
R& Q decreased:
Standard deviation of error in position (second half): 0.052486m
Standard deviation of error in velocity (second half): 0.186982m/s
R&Q increased:
Standard deviation of error in position (second half): 0.068819m
Standard deviation of error in velocity (second half): 0.262475m/s

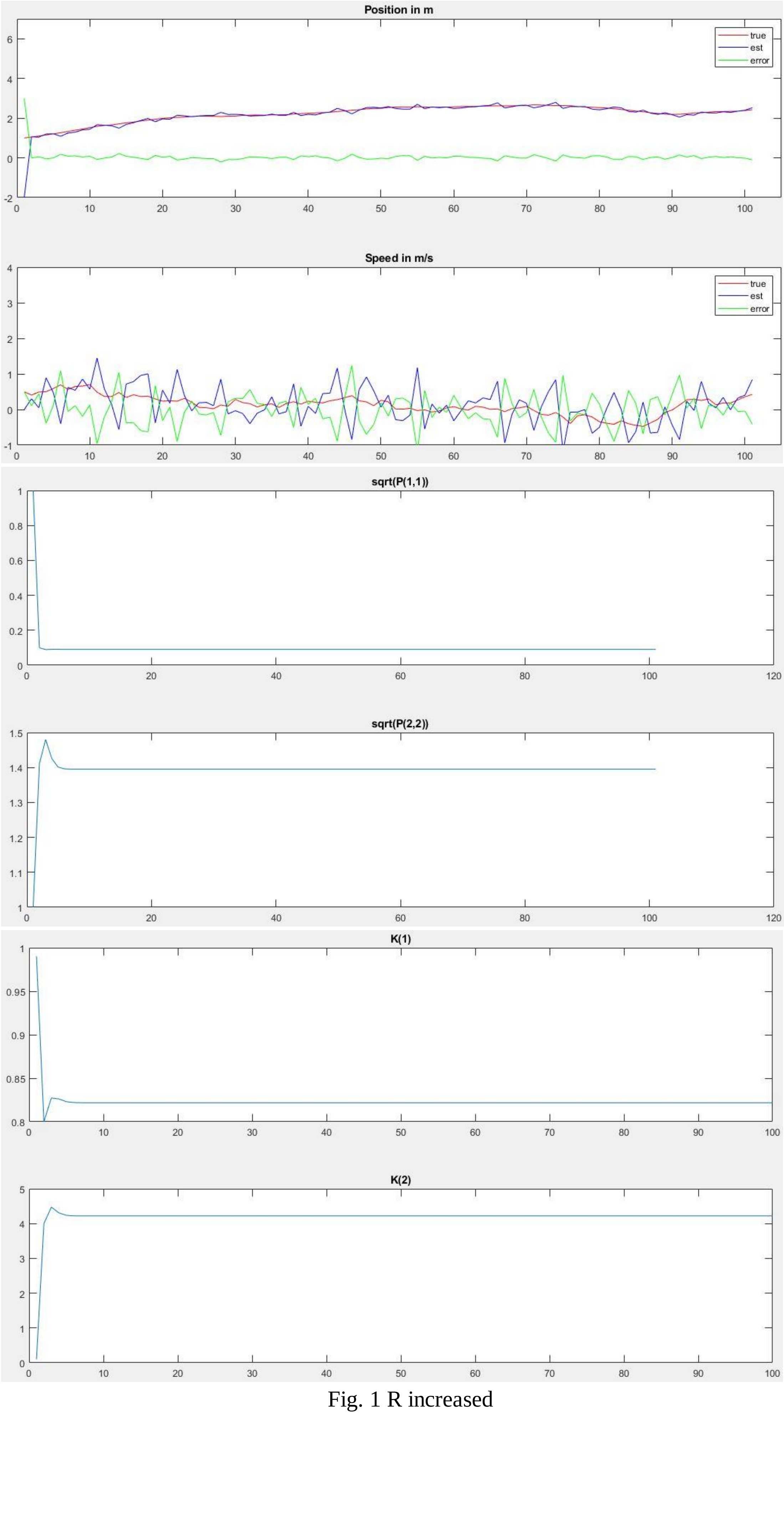


Fig. 1 R increased

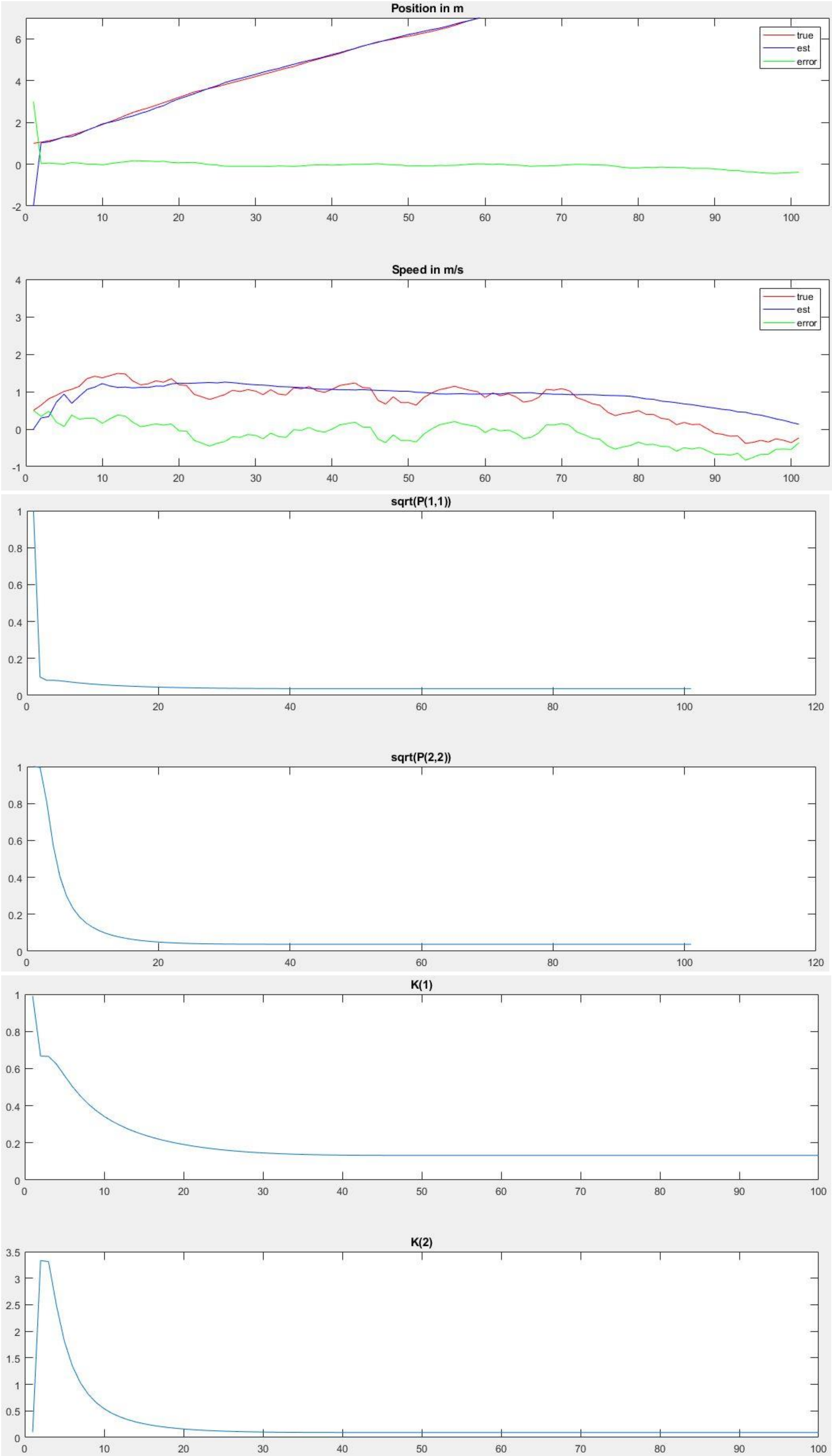


Fig. 2 R decreased

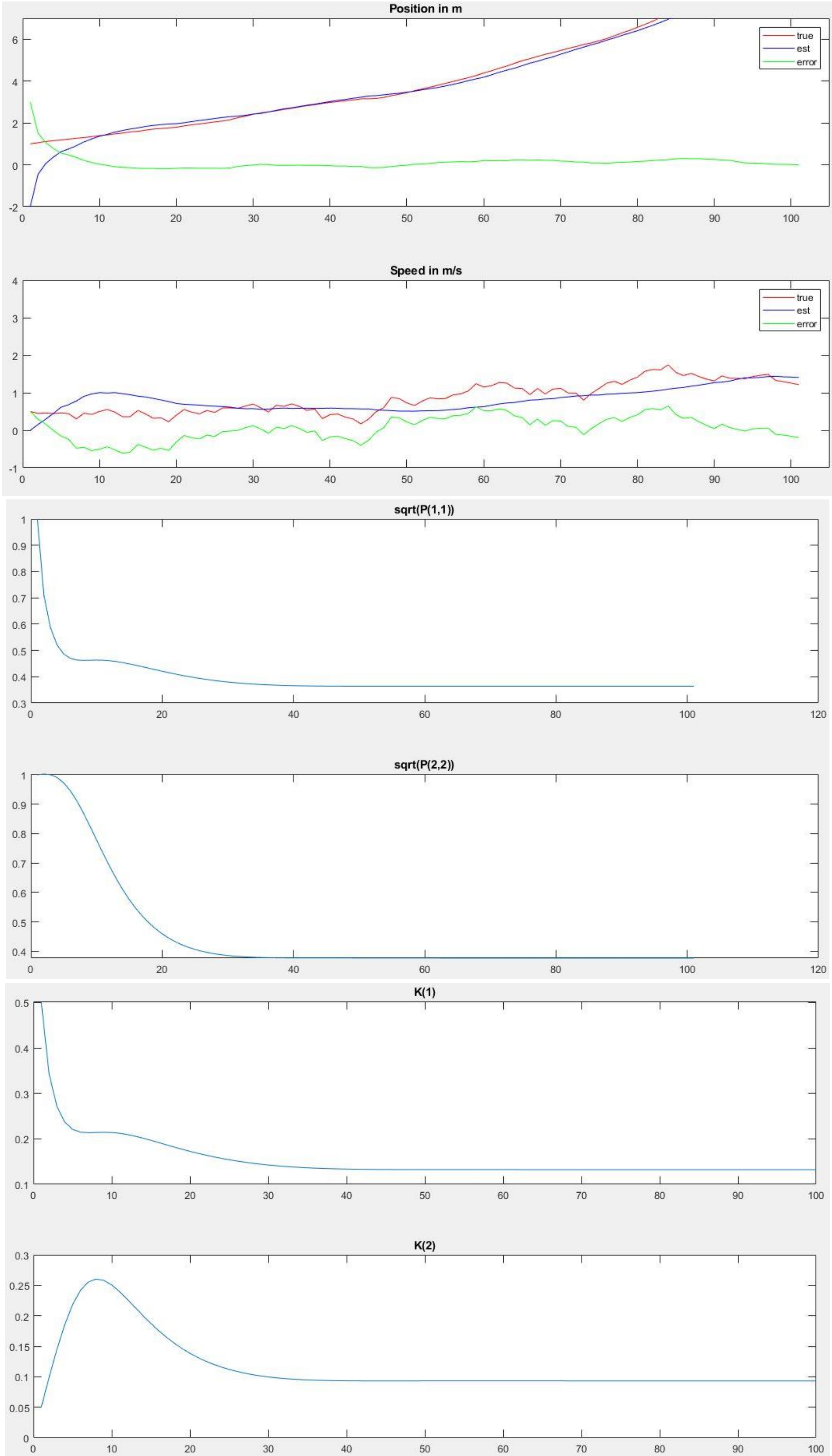


Fig. 3 Q increased

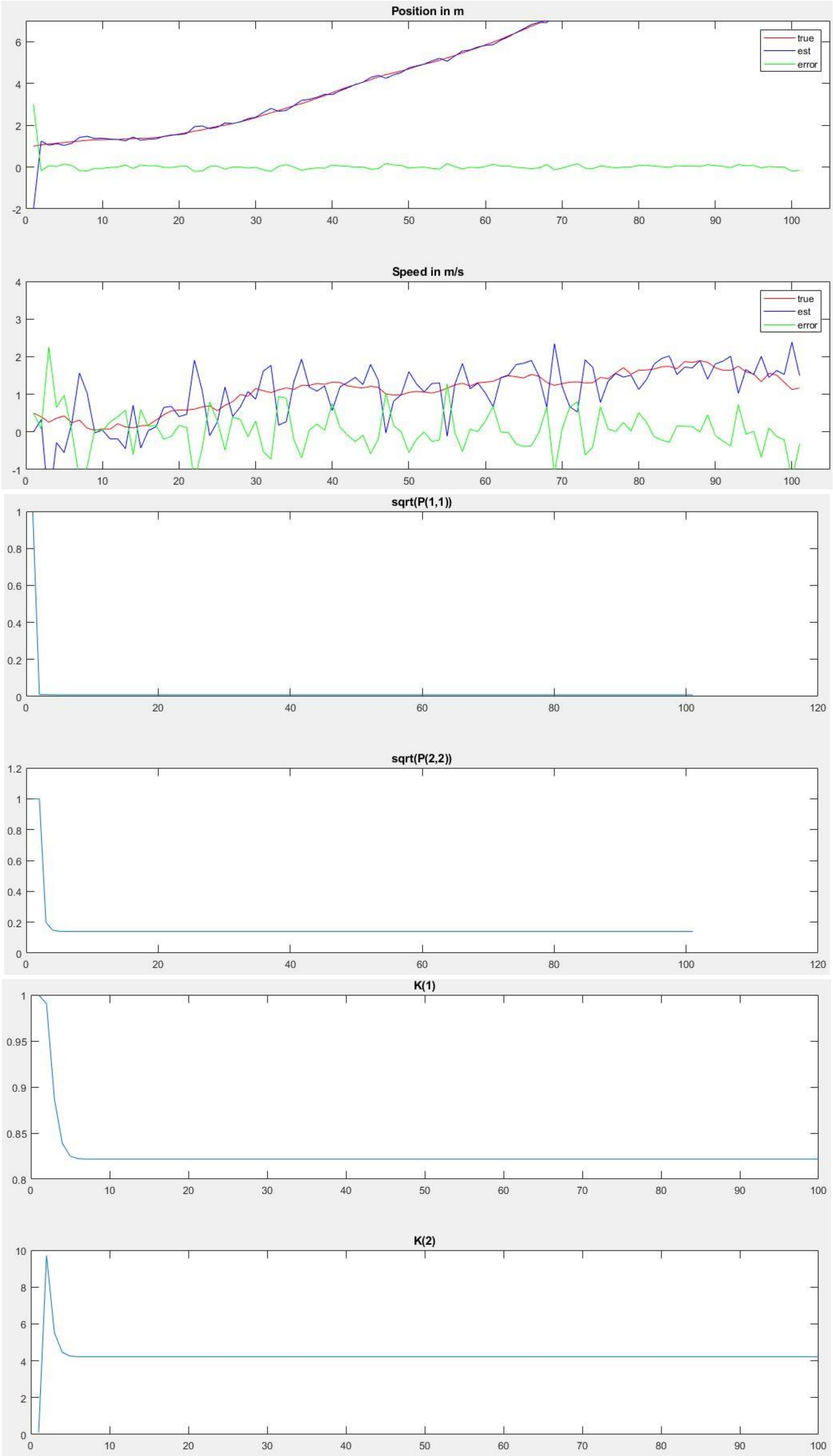


Fig. 4 Q decreased

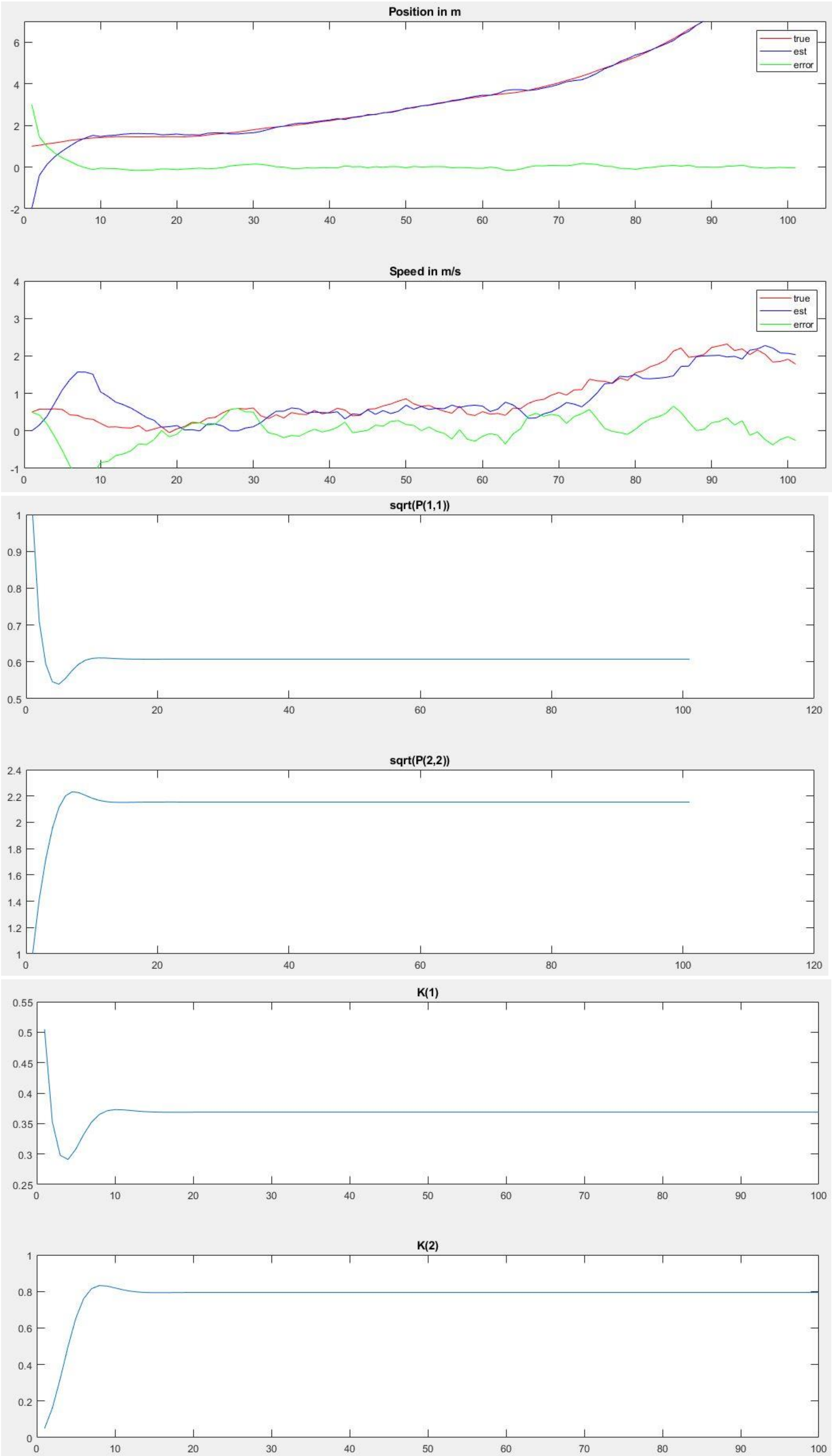


Fig. 5 R, Q increased

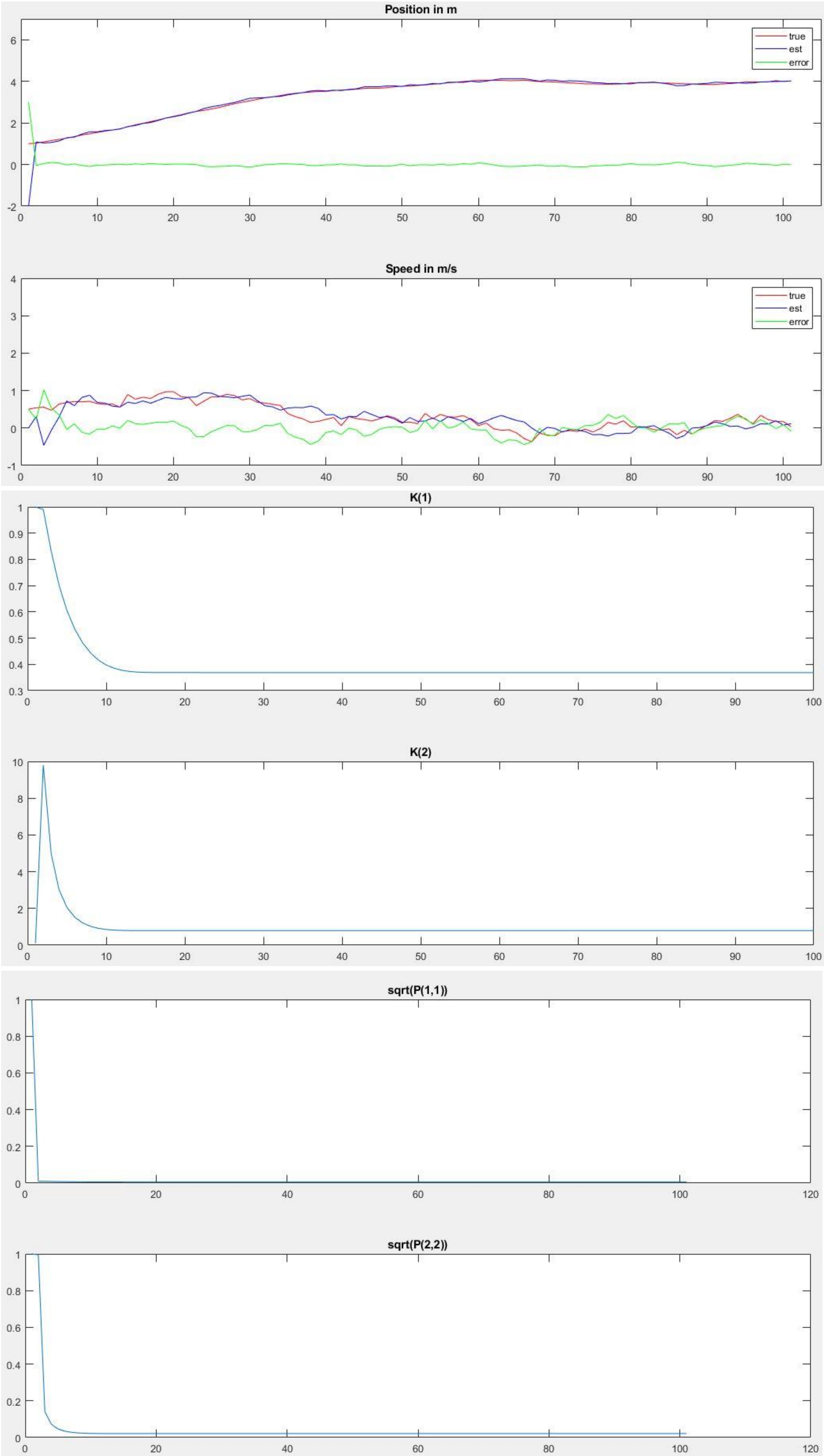


Fig. 6 R, Q decreased

4. **How do the initial values for P and ˆx affect the rate of convergence and the error of the estimates (try both much bigger and much smaller)?**
    - If initial state is relatively very far from true state (xhat = 100, true state = 1) and uncertainty (P =0), then Kalman gain is very low which implies update step or state measurement has very little effect on the estimate. So the time taken for convergence is high.
    - If initial state is relatively less far from true state (xhat = -2) and uncertainty (P =0), the time taken for convergence is relatively less than what was obtained for xhat=100
    - If the P = 100, then Kalman gain is high and the degree to which measurement model is incorporated in to the estimate increases, the update step impacts the estimate, thereby decreasing the convergence time.

    Refer figures below.

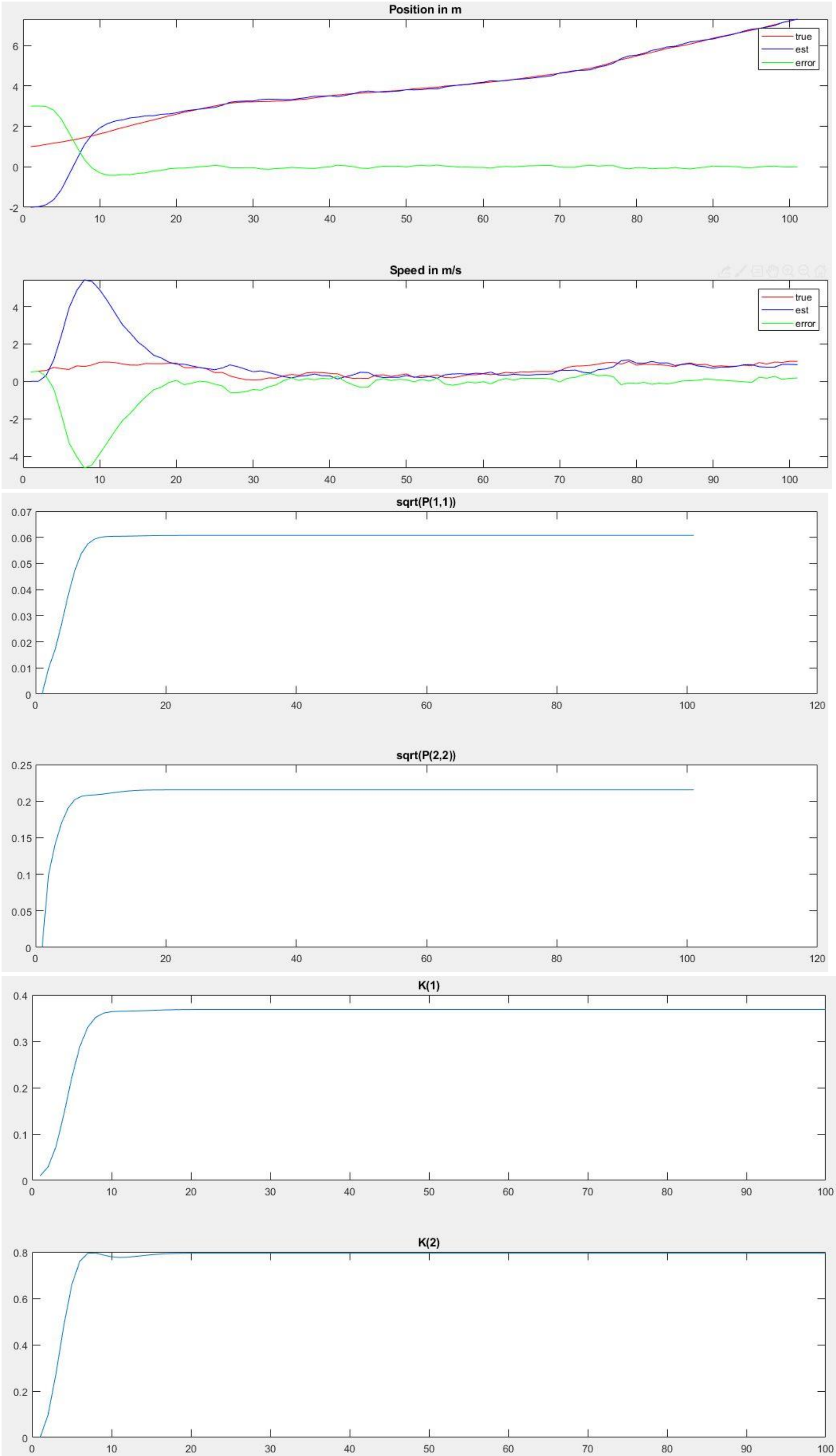

Position in m
true
est
error
Speed in m/s
true
est
error
sqrt(P(1,1))
sqrt(P(2,2))
K(1)
K(2)

Fig. 7 P_0_X_-2

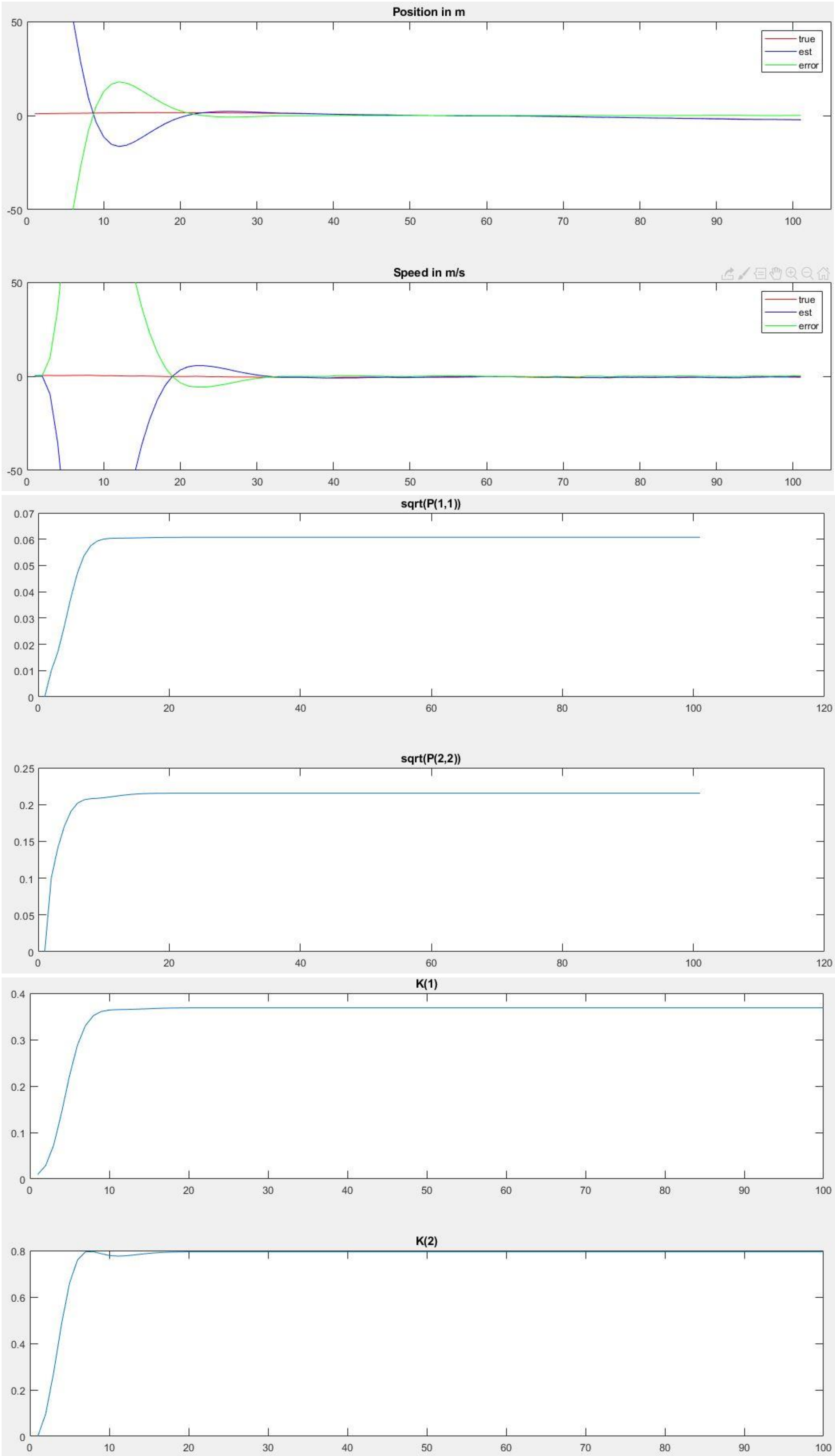

Position in m
true
est
error
Speed in m/s
true
est
error
sqrt(P(1,1))
sqrt(P(2,2))
K(1)
K(2)

Fig. 8 P_0_X_100

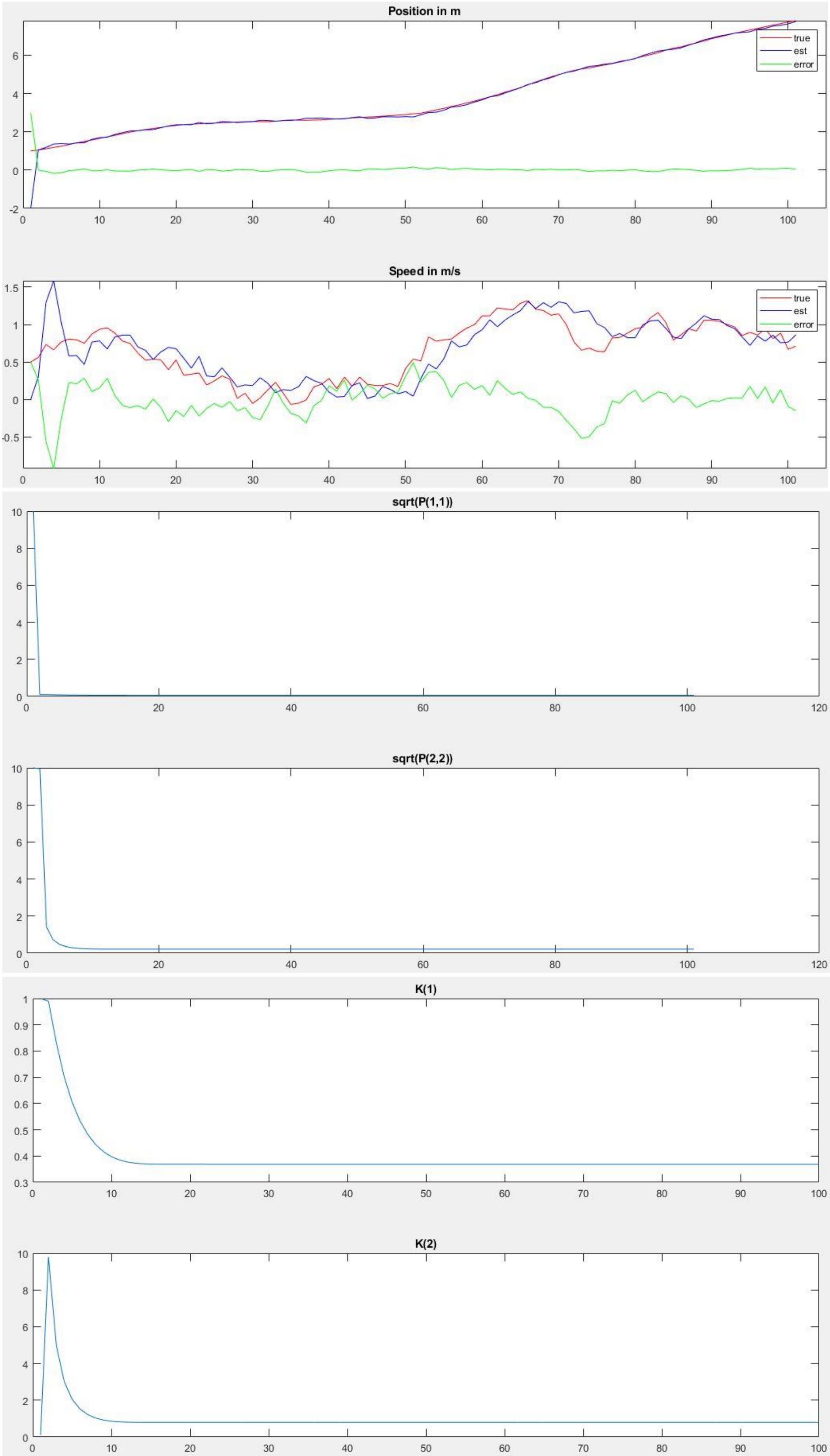

Position in m
true
est
error
Speed in m/s
true
est
error
sqrt(P(1,1))
sqrt(P(2,2))
K(1)
K(2)

Fig. 9 P_100_X_-2

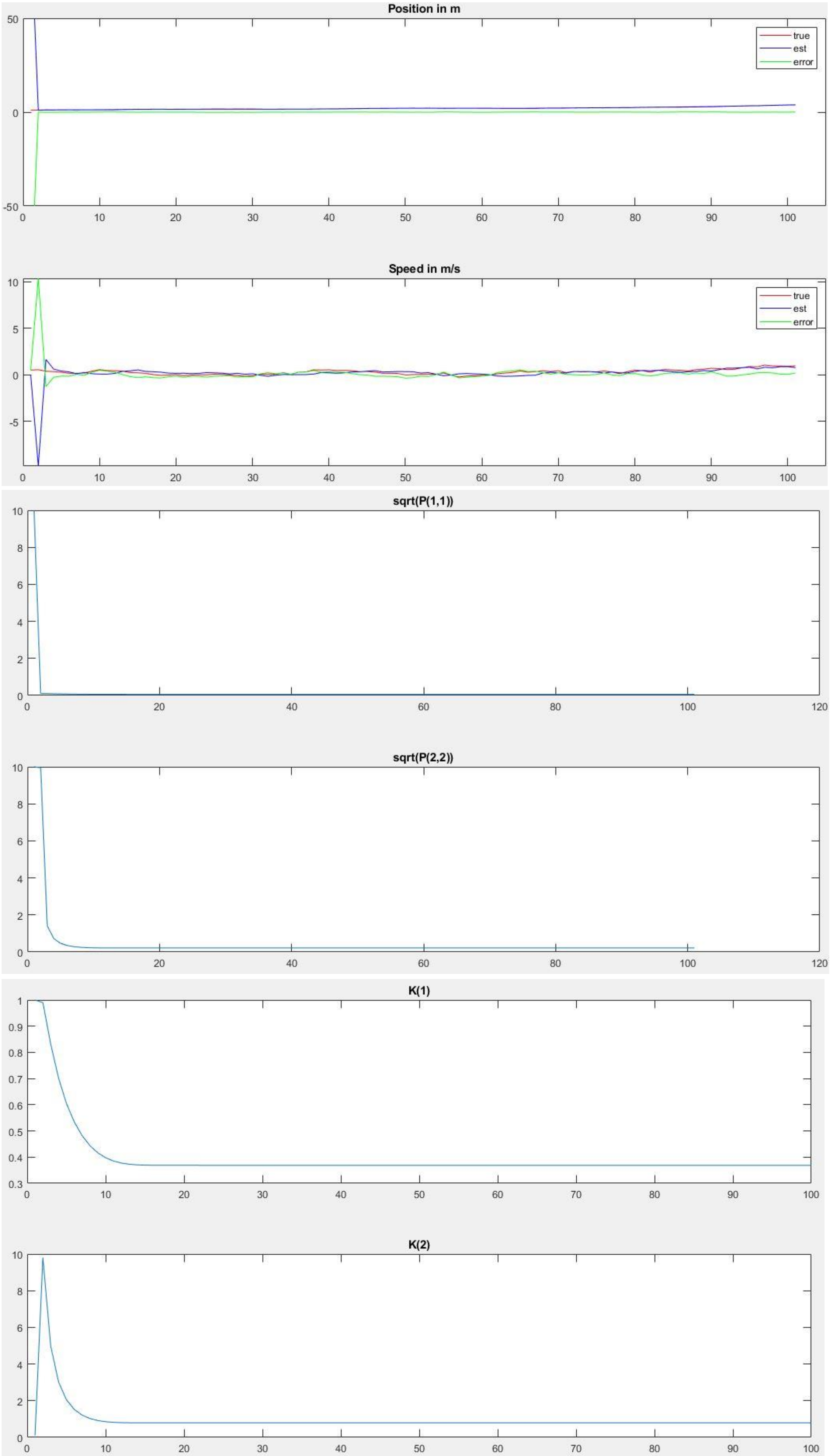


Fig. 10 P_100_X_100

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

$$p(x_t|u_{1:t}, z_{1:t}, \overline{x_o}, M) = \underbrace{\eta p(z_t|x_t, M)}_{\text{Update step}} \underbrace{\int p(x_t|u_t, x_{t-1}) p(x_{t-1}|z_{1:t-1}, u_{1:t-1}, \overline{x_o}, M)\, dx_{t-1}}_{\text{Predict step}}$$

$\overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}$ - Predict step

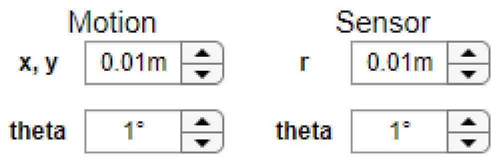
$bel(x_t) = \eta p(z_t|x_t, M)\, \overline{bel}(x_t)$ - Update step

6. **In the Maximum Likelihood data association, we assumed that the measurements are independent of each other. Is this a valid assumption? Explain why.** In real world, measurements are not generally independent of each other. But for modelling purpose, we have assumed data independence to reduce computational complexity. An example of data independence violation is when a robot is moving along a wall, for the first measurement it measures a wall, it moves for some time but is still along the wall so its next measurement is also a wall. These two measurements are not independent of each other.
7. **What are the bounds for δM in (8)? How does the choice of δM affect the outlier rejection process? What value do you suggest for λM 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?** $\delta_M = \int_0^{\lambda_M} X_2^2(u)\, du$
   - The bounds of δ_M are given by [0,1].
   - δ_M is directly proportional to λ_M, hence as we increase δ_M, the mahalanobis distance λ_M increases, thereby increasing the acceptance rate of measurements (which can also be noise)
   - For the case of "reliable measurements all arising from features in our map, that is all our measurements come from features on our map," λ_M can be high as we need to accept these reliable measurements without losing information.
   - For this case – "unreliable measurements with many arising from so called clutter or spurious measurements," it is advisable to keep the λ_M low so as to filter out as many noisy measurements as possible, thus preventing divergence.
8. **Can you think of some down-sides of the sequential update approach (Algorithm 3)? Hint: How does the first (noisy) measurement affect the intermediate results?** During sequential update, if the first measurement is noisy, the updated mean and variance using that particular measurement will be wrong. The robot will assume that it's in a position where it actually isn't. Even though we get a better measurement during the second time, the prediction step in the second time uses this noisy mean. This may lead to a diverging filter.

9. **How can you modify Algorithm 4 to avoid redundant recomputations?** The computation time can be reduced by avoiding the two for loops. The j-for loop (landmark loop), especially the computation of $\widehat{z_{t,j}}, H_{t,j}, S_{t,j}$ can be run once instead of i (number of measurements) times and the required values from the loop can be saved as vectors.
10. **What are the dimensions of ¯vt and ¯Ht in Algorithm 4? What were the corresponding dimensions in the sequential update algorithm? What does this tell you?**
    Batch update:
    H_bar = RN x C
    µ_bar = RN x C
    Sequential update:
    H_bar = RN x C
    µ_bar = RN x C
    where, N is the total number of inliers
    R is row dimension of each H_bar and µ_bar
    C is the column dimension of each H_bar and µ_bar
    Conclusion: Sequential update is memory friendly.

Simulation results:

Data set 1: Accurate sensor readings, low co-variances.
Simulation conditions: With batch associate, batch update and outlier detection (1%).
Noise model:



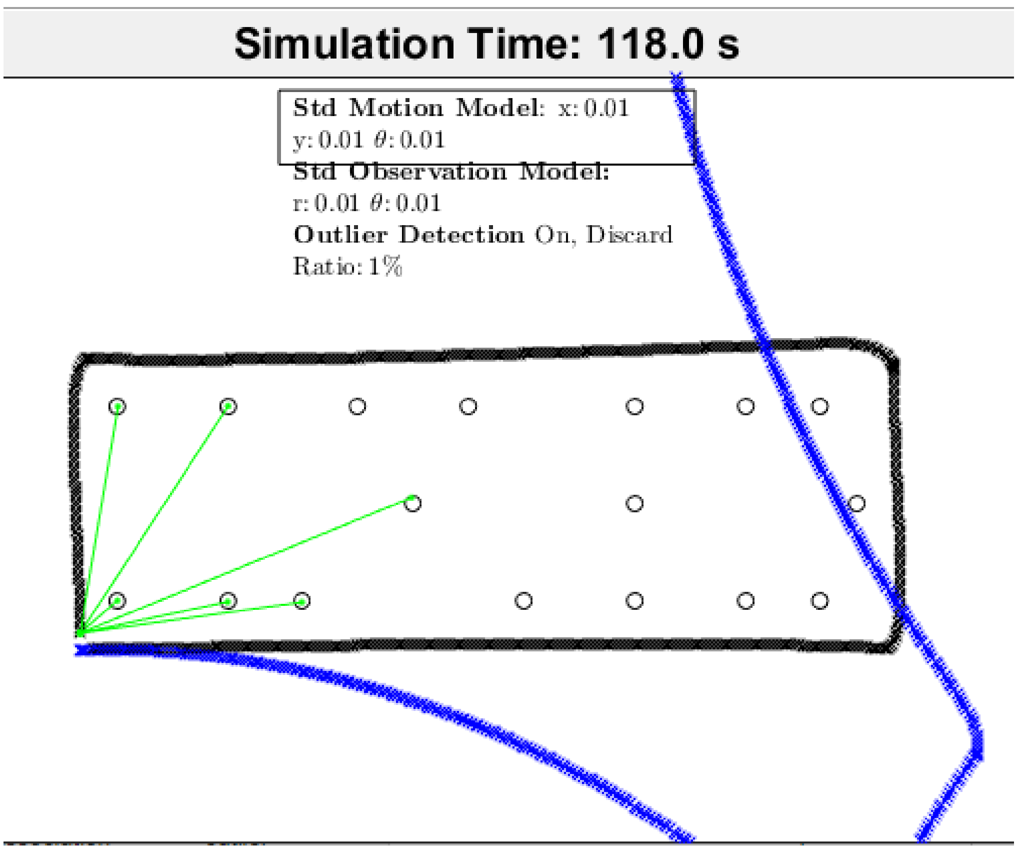


Fig 11. Data set 1 position output

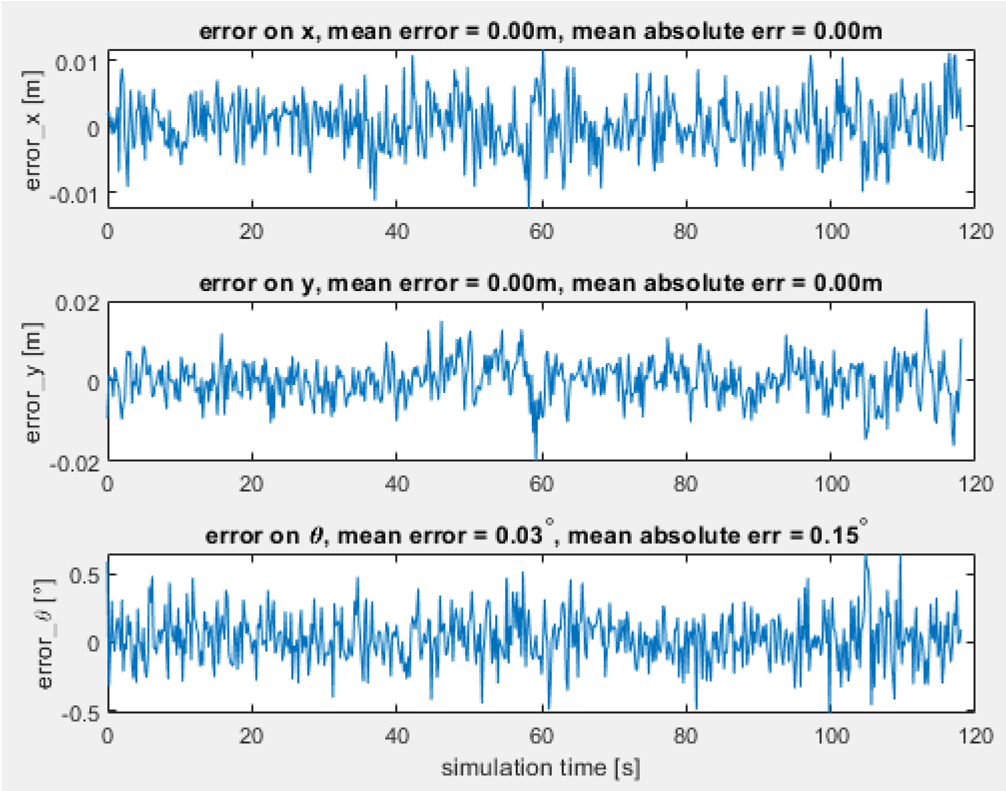


Fig 12. State estimation errors

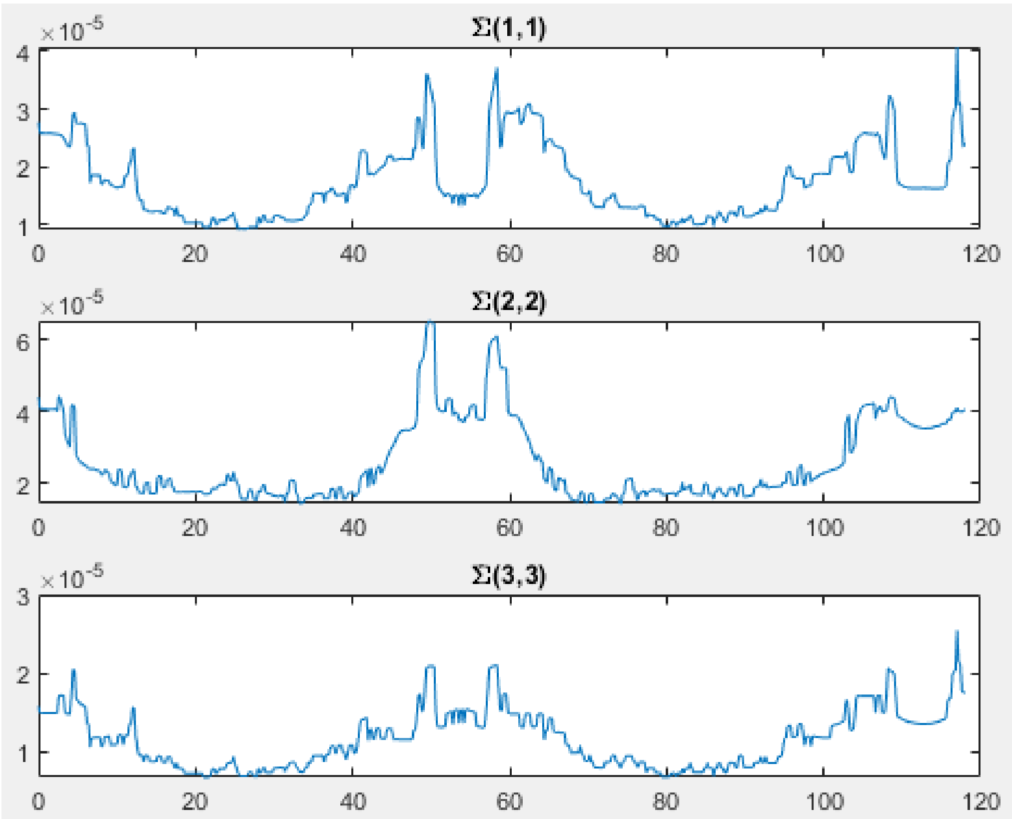


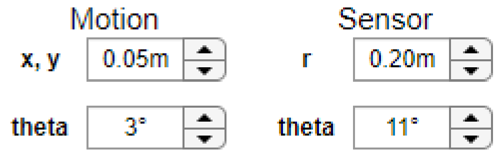
Fig 13. Evolution of state co-variance matrix

EKF performance for this set up is really good, the uncertainty in the motion model is well adjusted by the accuracy of the sensor. This balance is brought about by the EKF algorithm thereby providing a good tracking.

Data set 2: Measurement noise modelled as a co-variance

Simulation conditions: With batch associate, batch update and outlier detection (20%).

Noise model:



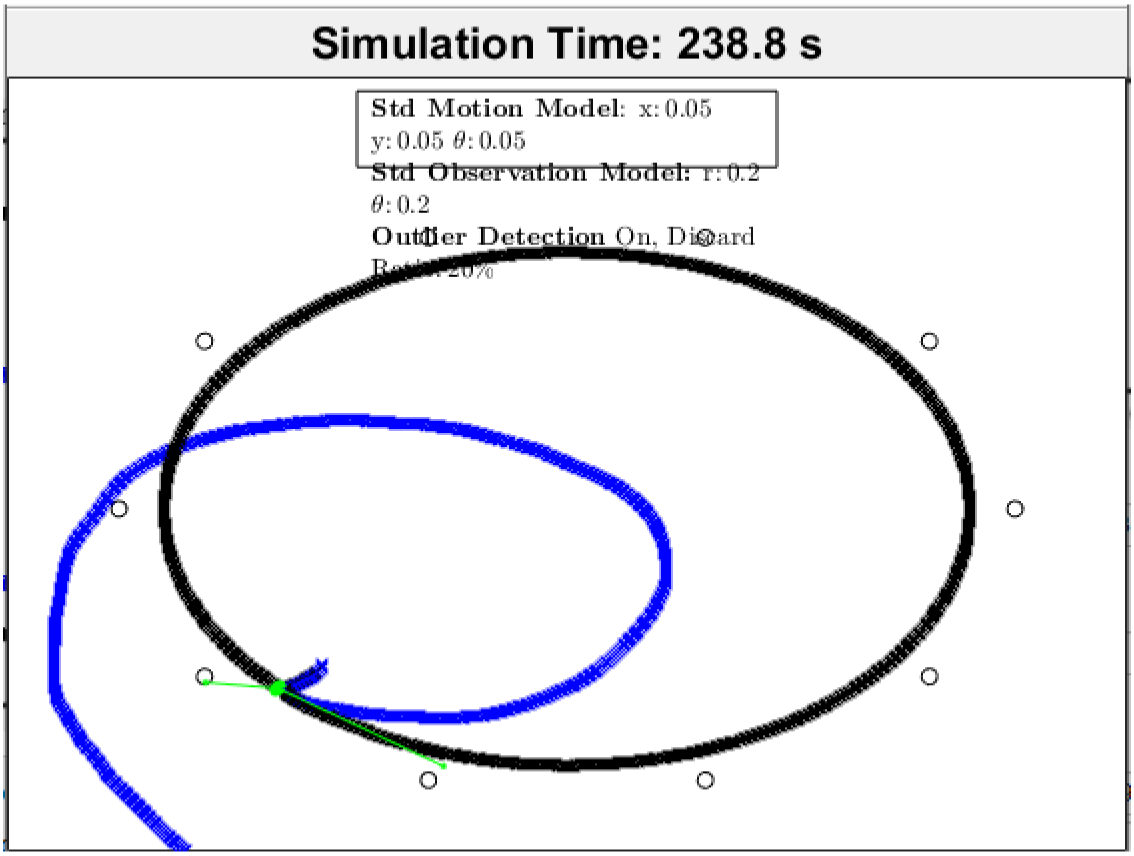


Fig 14. Data set 2 position output

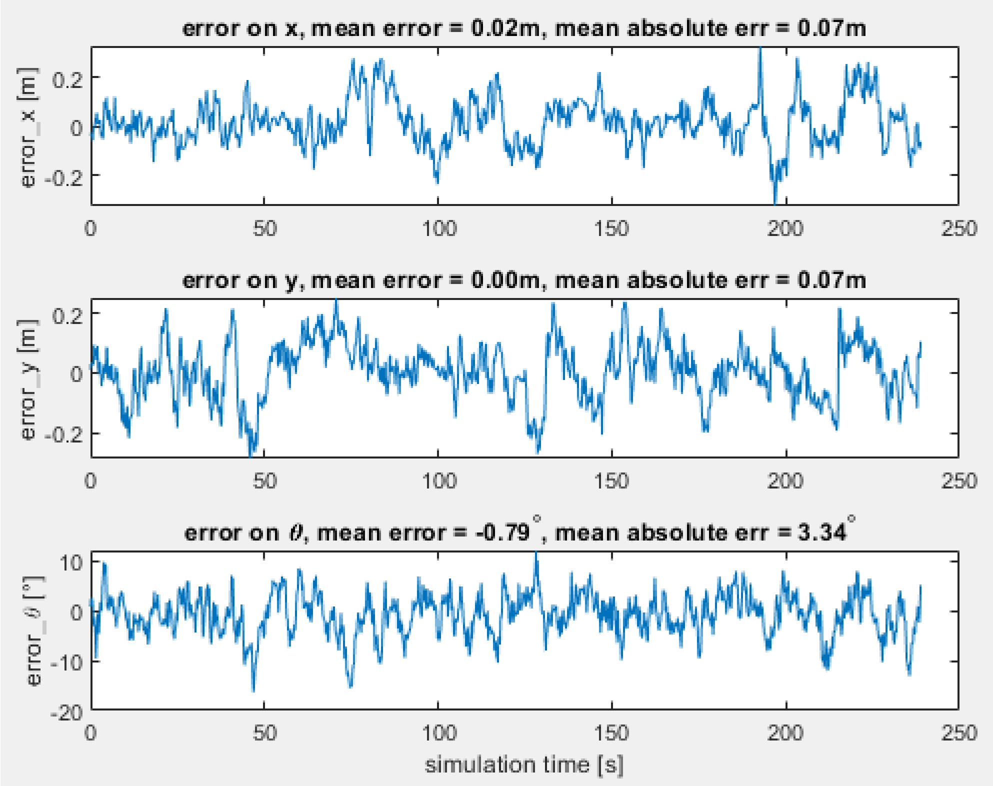


Fig 15. State estimation errors

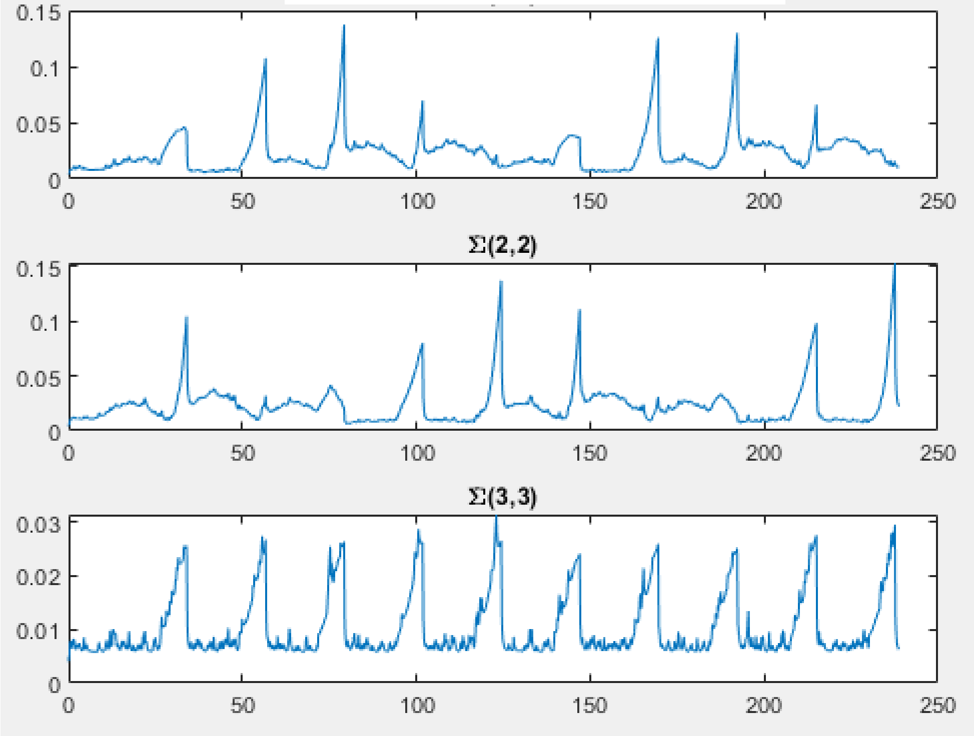


Fig 16. Evolution of state co-variance matrix

Data set 3: No odometry information available, hence large standard deviation for the process.

Simulation conditions: With batch associate, sequential update

Noise model:

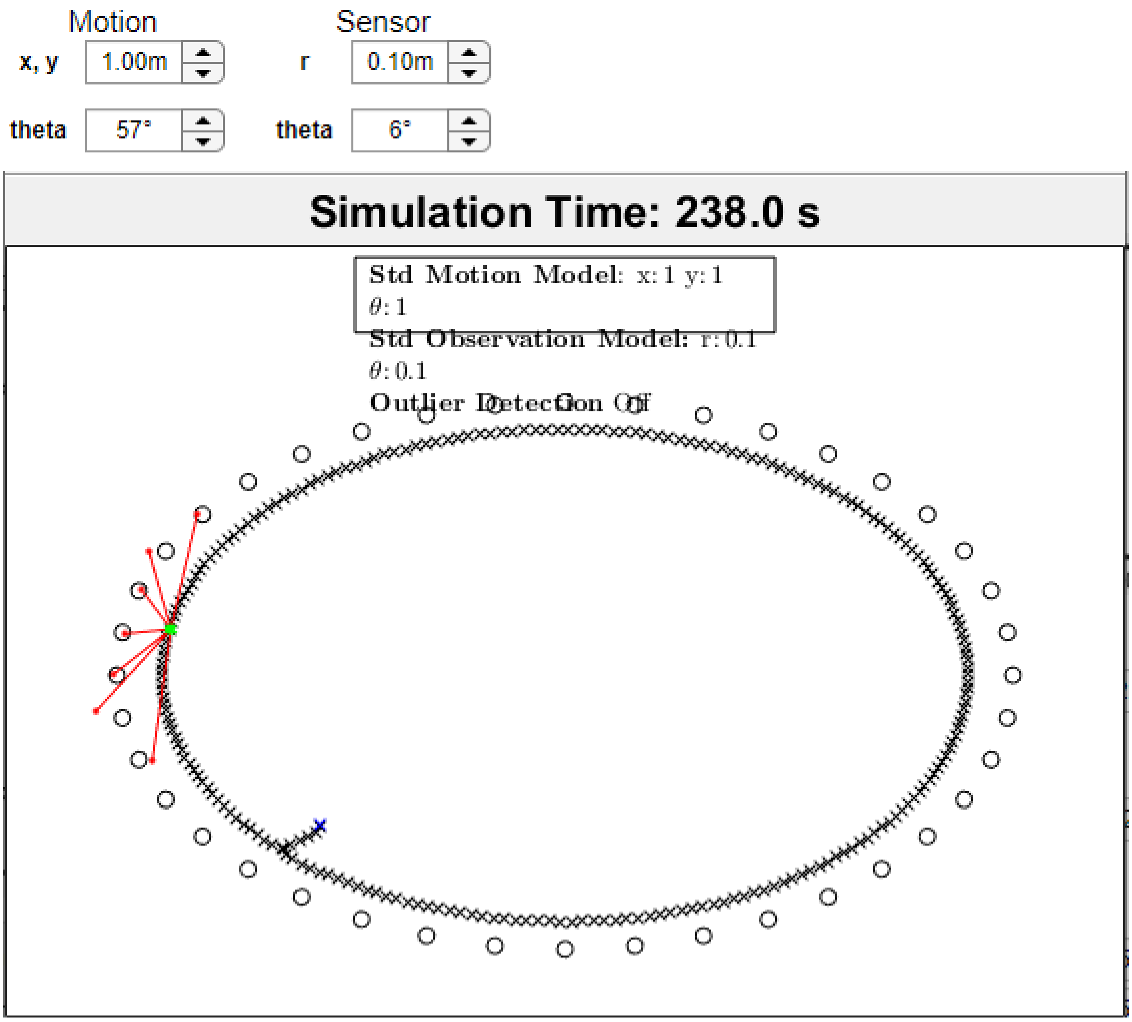


Fig 17. Data set 3 position output with sequential update

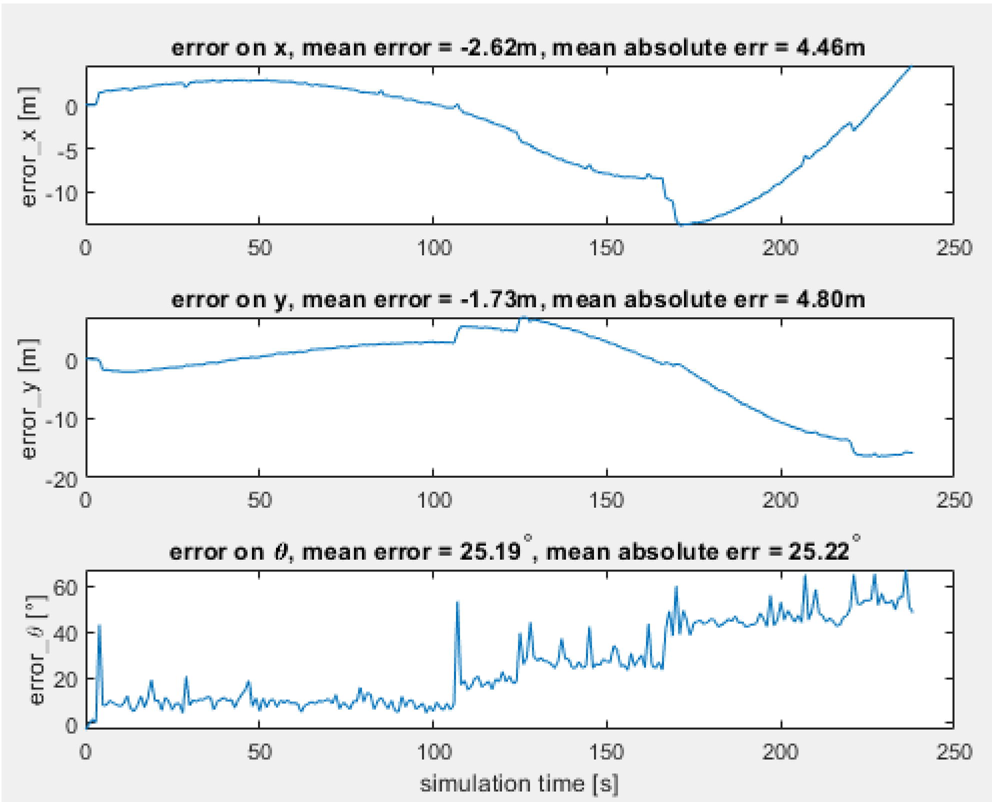


Fig 18. State estimation errors

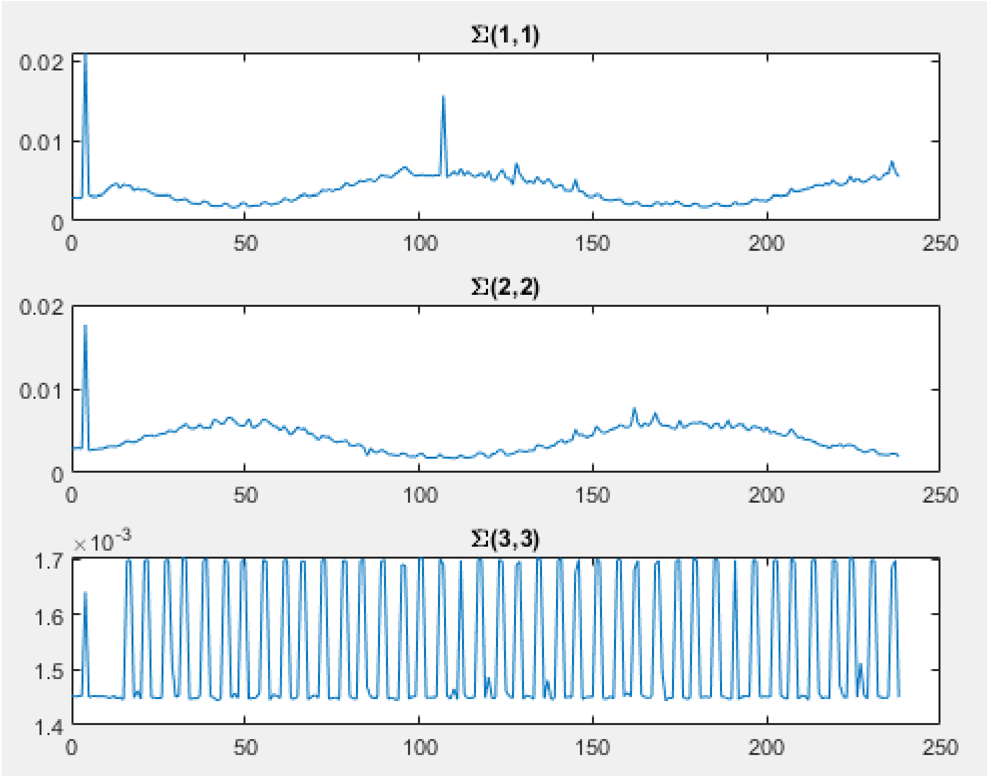
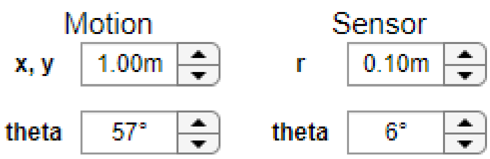


Fig 19. Evolution of state co-variance matrix

Data set 3: No odometry information available, hence large standard deviation for the process.

Simulation conditions: With batch associate, batch update

Noise model:



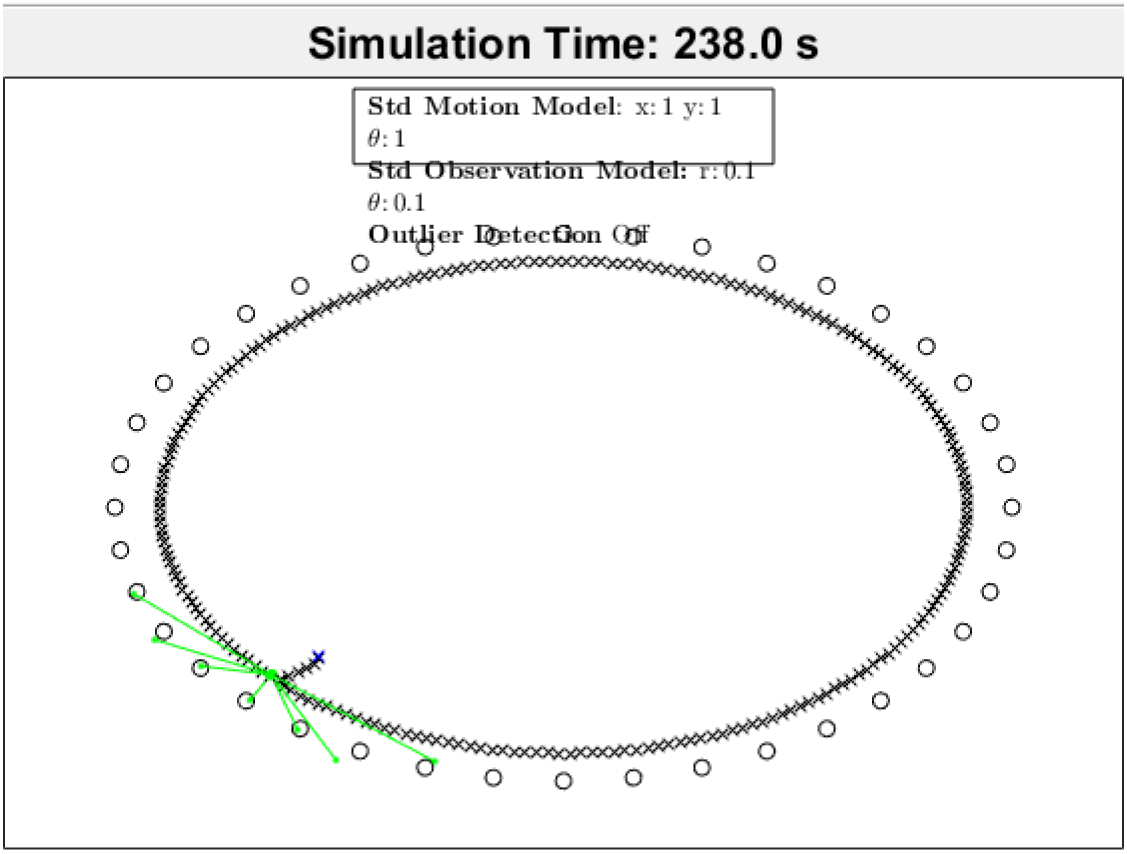


Fig 20. Data set 3 position output with batch update

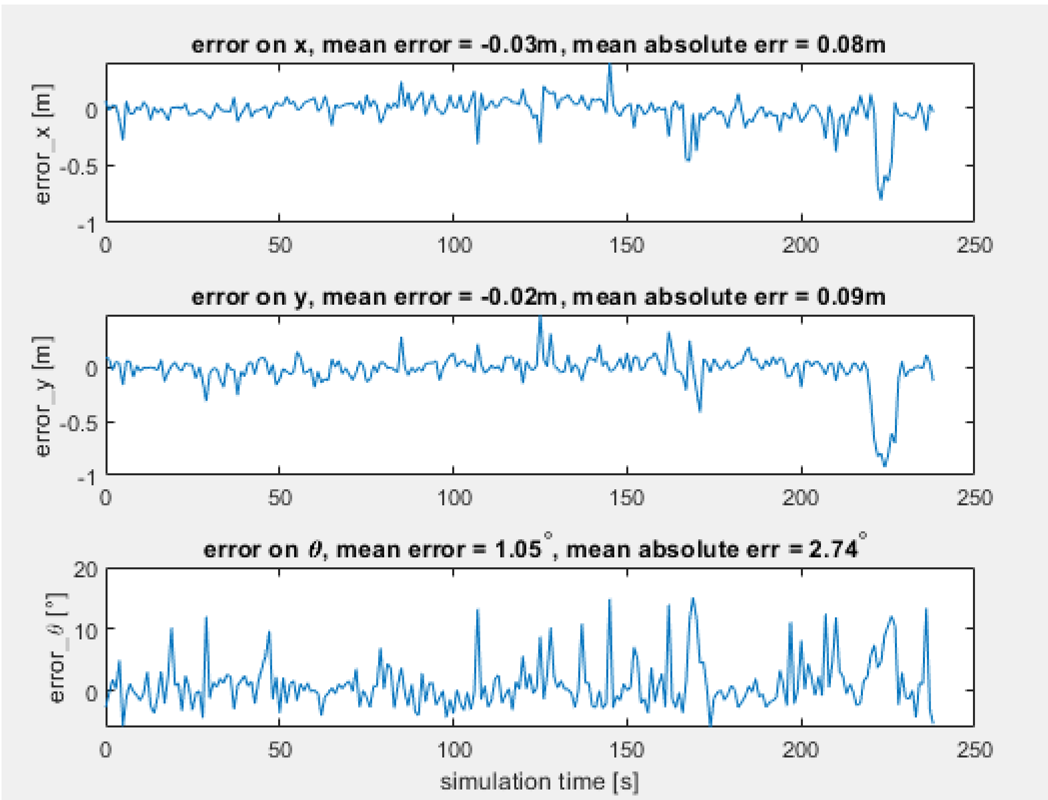


Fig 21. State estimation errors

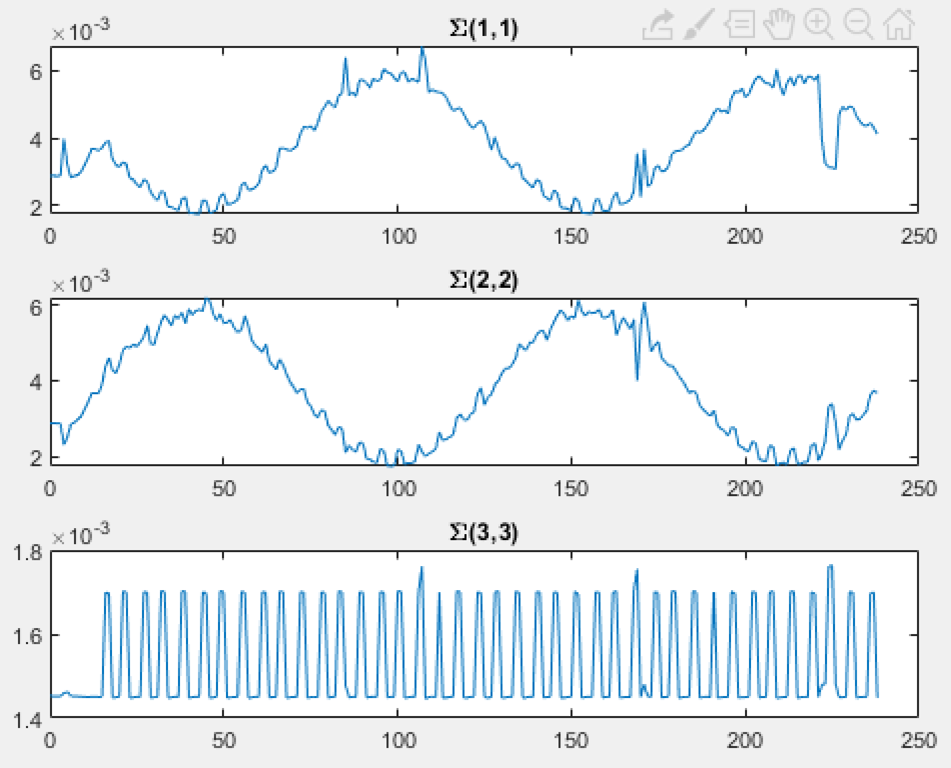


Fig 22. Evolution of state co-variance matrix

Conclusion:

From testing out different data sets