Time Series-4

Mehmet Eyyupoglu - s174448

Github Repo for the structured code: github.com/eyyupoglu/time-series-4

1 Question 4.1

First we replace the comlicated constants with simple signs,

$$\frac{dT_i}{dt} = \alpha_1(T_m - T_i) + \alpha_2(T_a - T_i) + \alpha_3\phi_h + \alpha_4 G_v \tag{1}$$

$$\frac{dT_m}{dt} = \beta_1(T_i - T_m) \tag{2}$$

These two equations can be written as system as following. Besides, since it will hold at any t, we choose t-1 in the system,

$$\frac{d}{dt}\begin{bmatrix} T_i^{t-1} \\ T_m^{t-1} \end{bmatrix} = \begin{bmatrix} \alpha_1 & -\alpha_1 \\ \beta_1 & -\beta_1 \end{bmatrix} \begin{bmatrix} T_i^{t-1} \\ T_m^{-1} \end{bmatrix} + \begin{bmatrix} \alpha_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} T_i^{t-1} \\ T_m^{t-1} \end{bmatrix} + \begin{bmatrix} \alpha_2 & \alpha_3 & \alpha_4 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} T_a \\ \phi_h \\ G_v \end{bmatrix} \tag{3}$$

Here, we use the euler's approximation for discretesation. Euler says that a continous slope at a point can be approximated by the slope of hypothenus of a triangle. Non perpendicular points of this triangle are any arbitrary two points on the function. In other words,

$$\frac{dT}{dt} = T_t - T_{t-1} \tag{4}$$

Then our equation becomes,

$$\begin{bmatrix} T_i^t - T_i^{t-1} \\ T_m^t - T_m^{t-1} \end{bmatrix} = \begin{bmatrix} \alpha_1 & -\alpha_1 \\ \beta_1 & -\beta_1 \end{bmatrix} \begin{bmatrix} T_i^{t-1} \\ T_m^{t-1} \end{bmatrix} + \begin{bmatrix} \alpha_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} T_i^{t-1} \\ T_m^{t-1} \end{bmatrix} + \begin{bmatrix} \alpha_2 & \alpha_3 & \alpha_4 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} T_a^{t-1} \\ \phi_h^{t-1} \\ G_v^{t-1} \end{bmatrix} \tag{5}$$

We leave the next time points isolated on the left hand side,

$$\begin{bmatrix} T_i^t \\ T_m^t \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} T_i^{t-1} \\ T_m^{t-1} \end{bmatrix} + \begin{bmatrix} \alpha_1 & -\alpha_1 \\ \beta_1 & -\beta_1 \end{bmatrix} \begin{bmatrix} T_i^{t-1} \\ T_m^{t-1} \end{bmatrix} + \begin{bmatrix} \alpha_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} T_i^{t-1} \\ T_m^{t-1} \end{bmatrix} + \begin{bmatrix} \alpha_2 & \alpha_3 & \alpha_4 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} T_a^{t-1} \\ \phi_h^{t-1} \\ G_v^{t-1} \end{bmatrix} \tag{6}$$

Finally,

$$\begin{bmatrix} T_i^t \\ T_m^t \end{bmatrix} = \begin{bmatrix} 2\alpha_1 + 1 & -\alpha_1 \\ \beta_1 & -\beta_1 + 1 \end{bmatrix} \begin{bmatrix} T_i^{t-1} \\ T_m^{t-1} \end{bmatrix} + \begin{bmatrix} \alpha_2 & \alpha_3 & \alpha_4 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} T_a^{t-1} \\ \phi_h^{t-1} \\ G_v^{t-1} \end{bmatrix} \tag{7}$$

When we add the measurement errors, we get the desired form of the state equation,

$$\begin{bmatrix} T_i^t \\ T_m^t \end{bmatrix} = A \begin{bmatrix} T_i^{t-1} \\ T_m^{t-1} \end{bmatrix} + B \begin{bmatrix} T_a^{t-1} \\ \phi_h^{t-1} \\ G_v^{t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \end{bmatrix} \tag{8}$$

2 Question 4.2

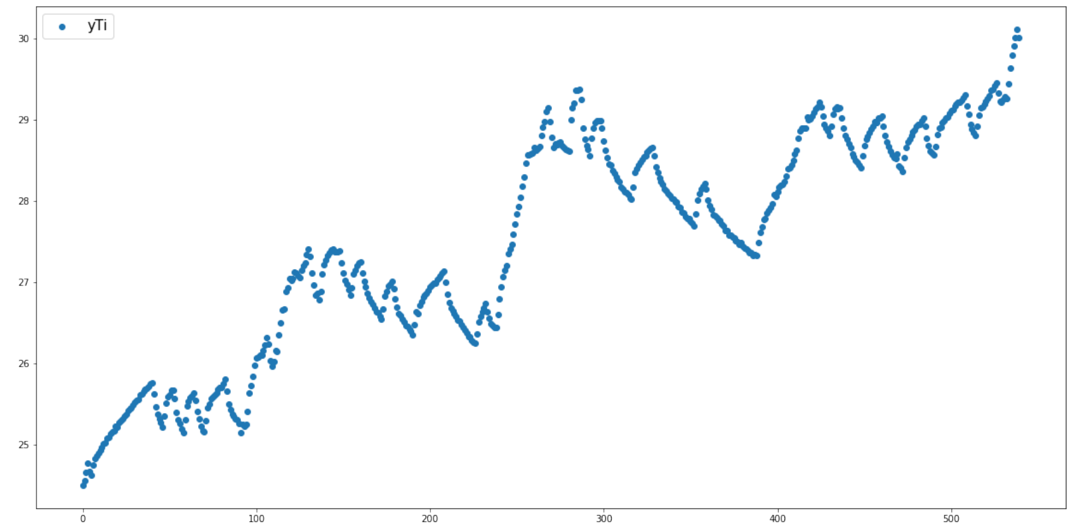

Figure 1: room air temperature, measured

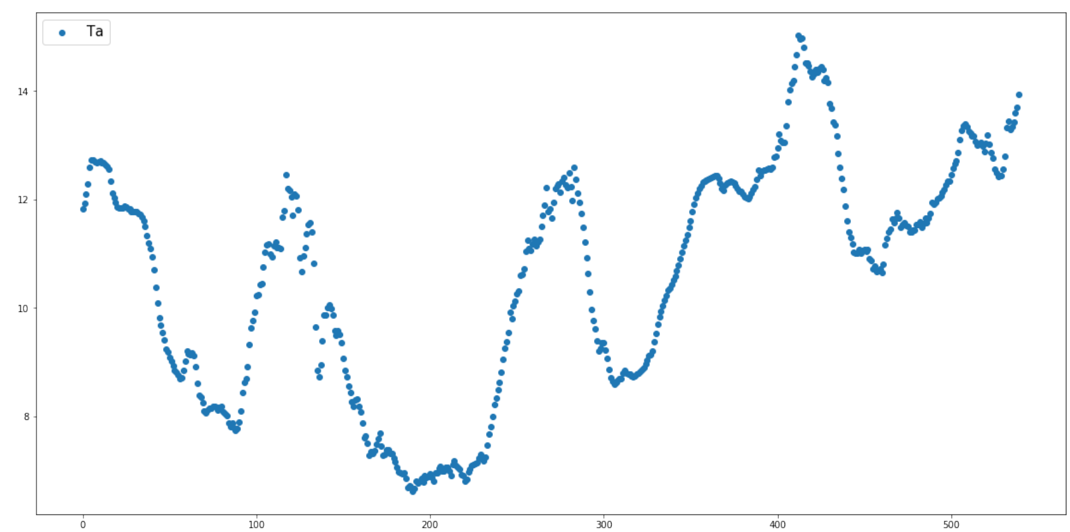

Figure 2: The outdoor temperature

We see that the room air temperature has sharp increases probably because the system runs the heater in those intervals. It has sharp increases together with sharp decreases. We can also predict the day and night times just by looking at the outdoor temperature. Even though the data has only 3 days of data, we can see the daily cycles in the outdoor temperature.

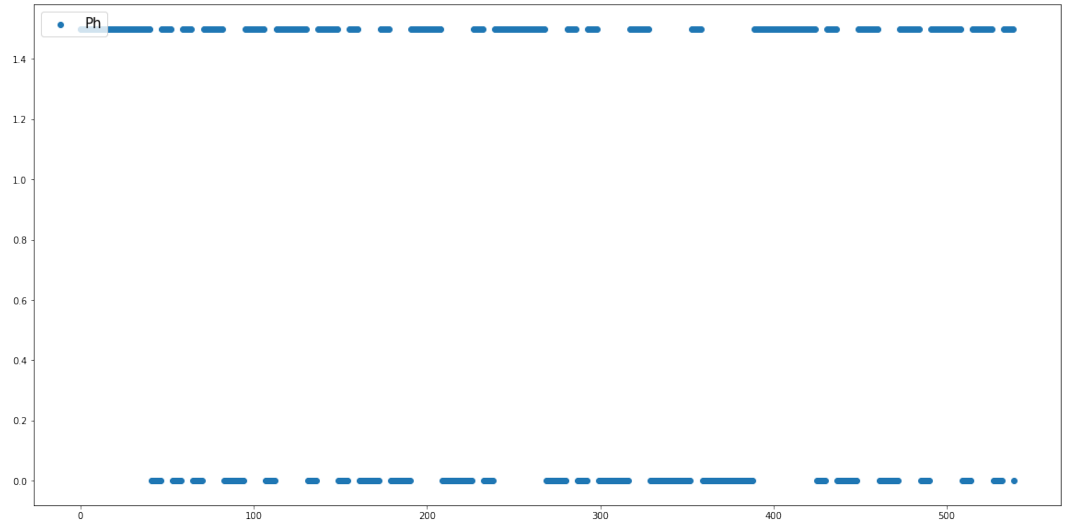

Figure 3: Heating Power

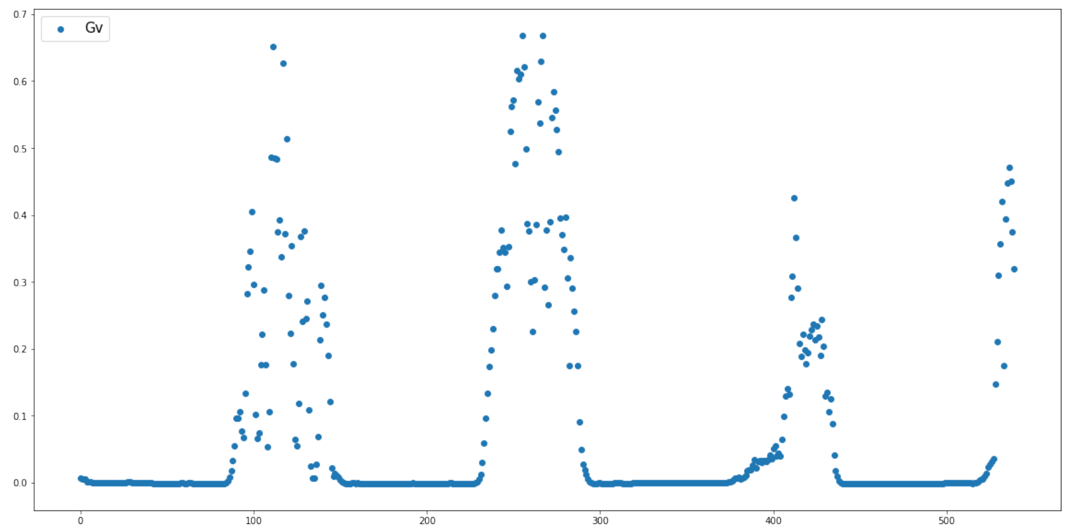

Figure 4: Solar radiation

Solar radiation seems to have a daily seaonalities as well which makes sense because there is solar radiation only when there is sun. Therefore, it gives us a more clear picture when the sun is out.

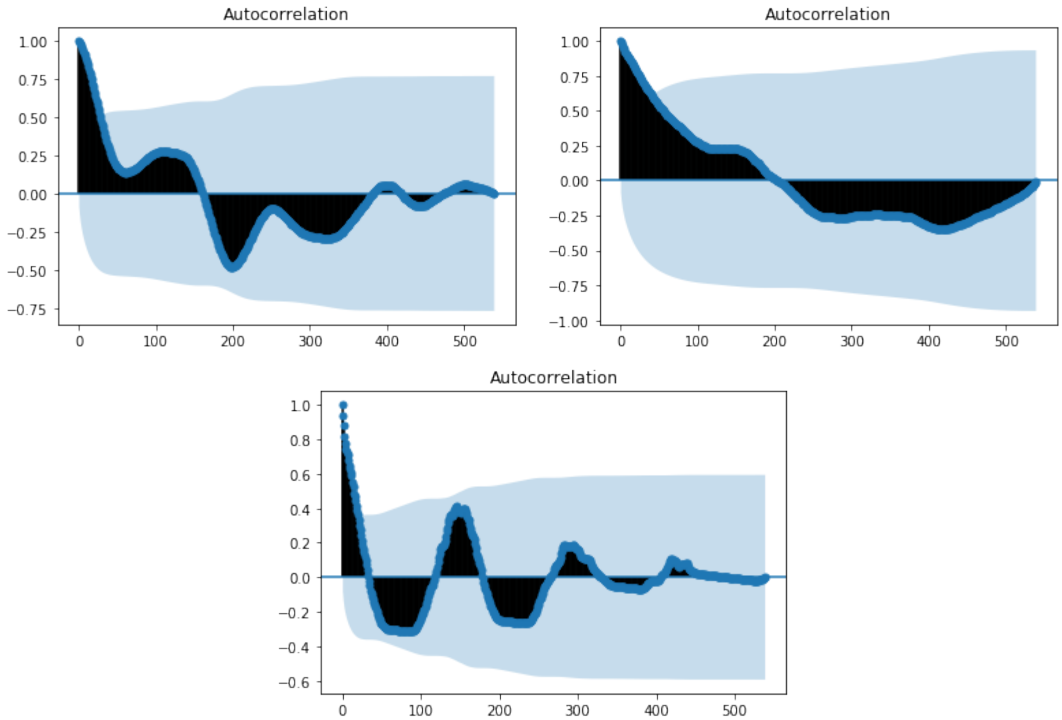

Figure 5: Ta-ACF, yTi-ACF and Gv-ACF respectively

We see that the only one that seems to have a chance to be stationary is the solar radiation which is the one at the bottom.

3 Question 4.3

Visualize the 1-step predictions of the states.

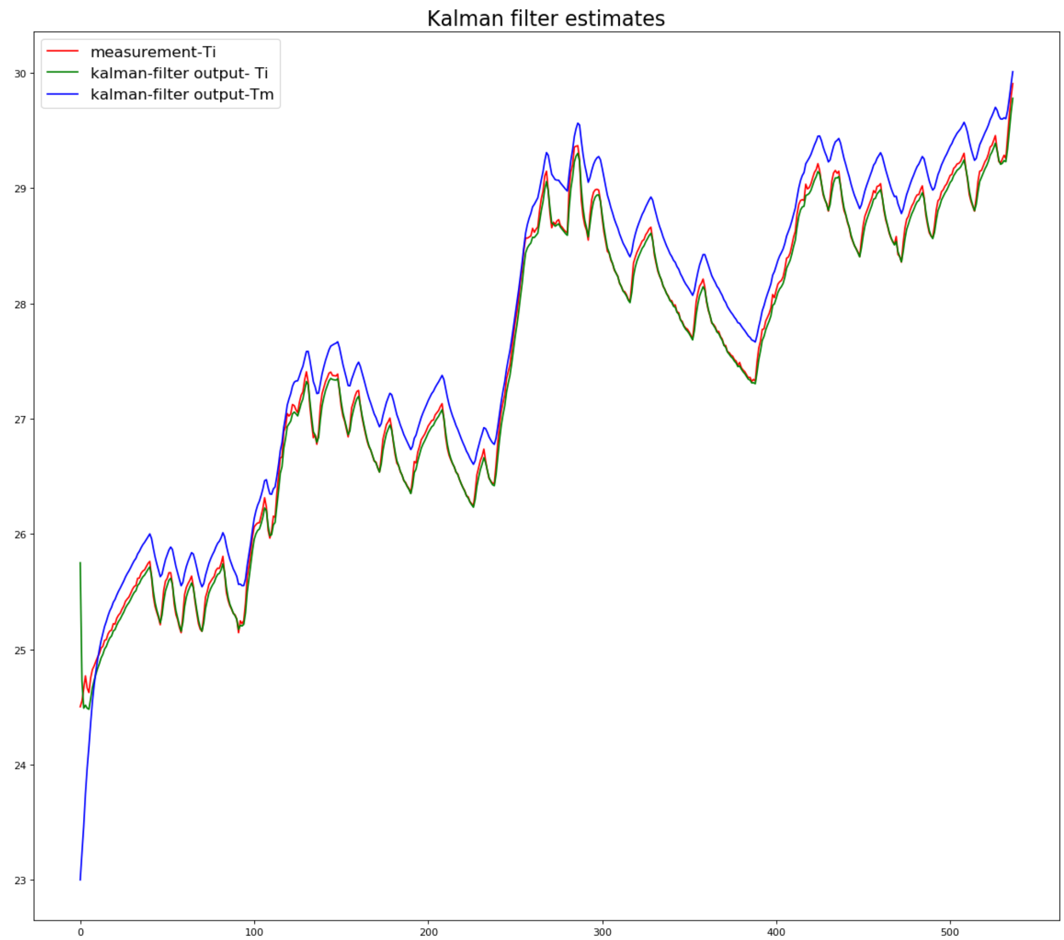

Figure 6: Kalman filter estimates gets closer to the real data very quickly

Predict the indoor temperature of the last third of the data.

We find the standard deviation of the predictions as the square root of the estimated variances of the states. Then we use them to get the 95 percentiles of student-t distribution so that we have the upper and lower interval boundaries. $\begin{bmatrix} \mu - z\sigma \\ \mu + z\sigma \end{bmatrix}$ is used to get these values.

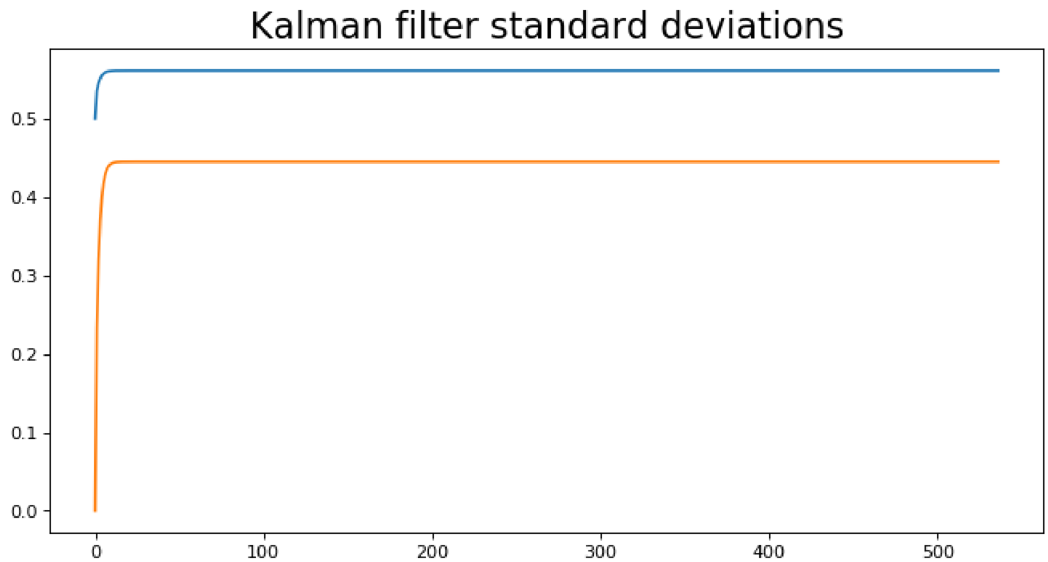

Figure 7: Standard deviations that are calculated from the each step covariances are shown. They converge very quickly.

3.0.1 Predict the indoor temperature of the last third of the data

	1	2	3
Original Data	*30.015*	*30.1075*	*30.0125*
Prediction Upper 95	30.765845338415403	30.86509900593946	30.84192080266257
Predictions	**29.893649439128733**	**29.99290310665279**	**29.96972490337591**
Prediction Lower 95	29.021453539842064	29.12070720736612	29.09752900408924

Table 1: Prediction of the last observation

4 Question 4.4

Expectation maximization algorithm

New Σ_2 is **0.00024351**. New Σ_1 is $\begin{bmatrix} \mathbf{0.00186735} & \mathbf{0.01007461} \\ \mathbf{0.01007461} & \mathbf{0.08056345} \end{bmatrix}$ We use expectation maximization(EM) algorithm for this. The idea is to maximize the sum of each measurement's probability.

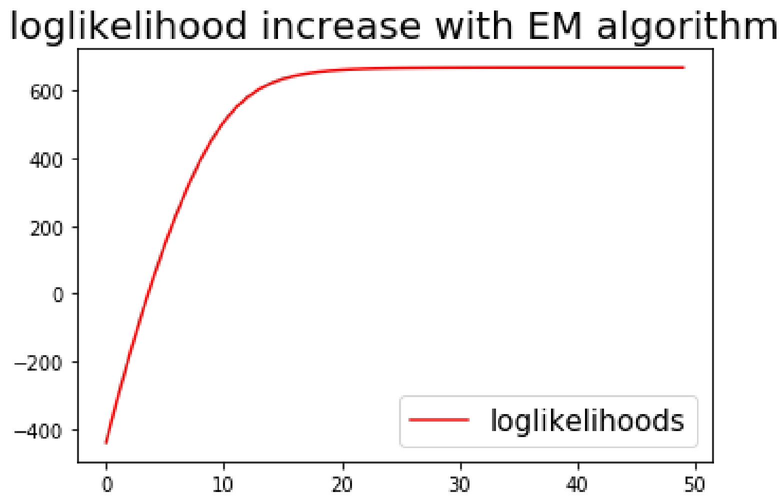

Figure 8: This is how the loglikelihood of the fitting increases with every step of EM algorithm.

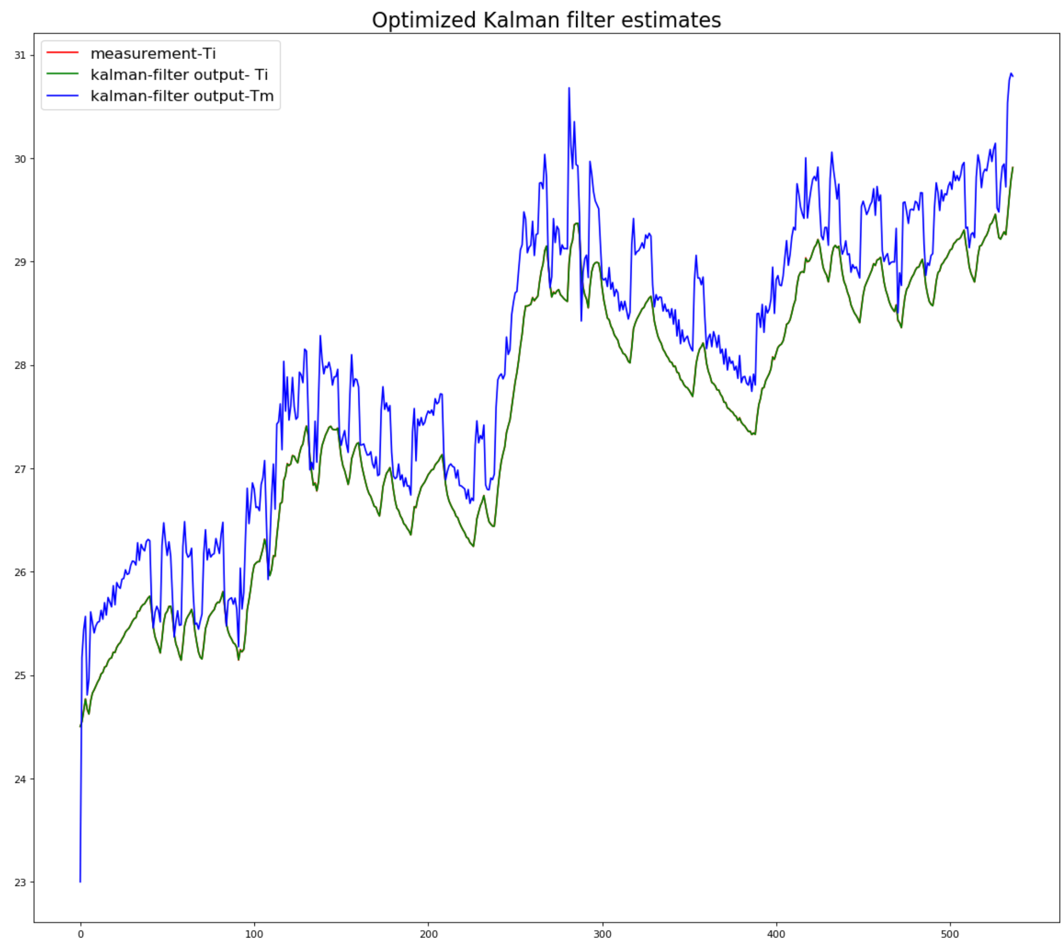

Figure 9: One step predictons of the EM optimized kalman filter is shown. Note that the reason we cannot see the difference between original and predicted Ti, is that the predictions are extremely close to the original data.

	1	2	3
Original Data	*30.015*	*30.1075*	*30.0125*
Prediction Upper 95	30.05280143477772,	30.166242569724687,	30.080826783722575
Predictions	**30.014773581289916,**	**30.10674277212054,**	**30.0198434526241**
Prediction Lower 95	29.97674572780211,	30.047242974516394,	29.95886012152563

Table 2: Predictions after the Expectation Maximization optimization

Optimization with loglikelihood maximization

We use the package called scipy.minimize and send our objective function(loglikelihood calculator). New Σ_2 is **0.0001**. New Σ_1 is $\begin{bmatrix} \mathbf{0.1069} & \mathbf{0.3986} \\ \mathbf{0.3986} & \mathbf{0.001} \end{bmatrix}$

	1	2	3
Original Data	*30.015*	*30.1075*	*30.0125*
Prediction Upper 95	30.02385419379815	30.119447392304664	30.024387174952803
Predictions	**30.01499895483748**	**30.10749287128901**	**30.01256564427253**
Prediction Lower 95	30.00614371587681,	30.095538350273358,	30.00074411359226

Table 3: Predictions after the maximum likelyhood optimization

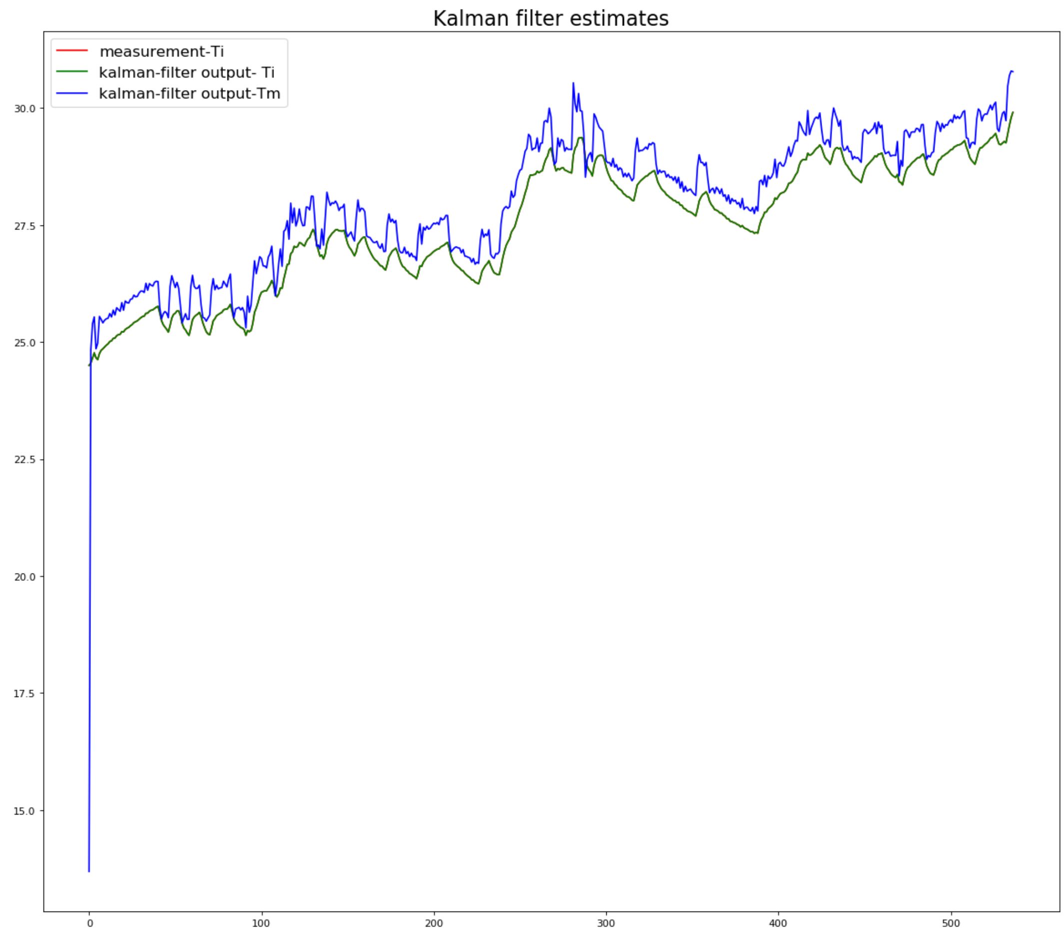

Figure 10: One step predictons of the MLE optimized kalman filter is shown. Note that the reason we cannot see the difference between original and predicted Ti, is that the predictions are extremely close to the original data.

Note that the maximum likelihood optimization with 50 iterations gave a better result than EM optimization with 50 iterations. The difference seems very small though.