AUTO-REGRESSIVE MODEL PARAMETER ESTIMATION USING KALMAN FILTERING

Sameen Islam
University of Southampton
England, United Kingdom
si1u19@soton.ac.uk

ABSTRACT

In the real-world, it is often not possible to make a direct measurement of an underlying phenomena. In such cases, we observe some effect and measure its change over time. However, this kind of measurement is prone to drift which can cause fatal errors. Kalman filter can help us in this situation by learning to predict the next underlying state of a system given some external measurement, taking into account the uncertainty in the estimate of state prediction and measurement noise. We perform a series of experiments using this filter to determine its efficacy in learning some stochastic process.

1 INTRODUCTION

Kalman filtering makes use of state space models to represent dynamic systems which change over time to make a sequential optimal state estimation. The filter requires us to know the relationship between the internal state of a system and the instrument with which we measure it, and it takes into account both the noise in instrument measurement and internal state change. The Bayesian framework is used to represent noise processes which carry uncertainty. Applying the Kalman filter is popular in a wide range of state estimation problems - from robotic motion planning and object position forecasting in virtual reality, to flight trajectory optimisation. One of the first applications of this algorithm has been in the Apollo project which was used to estimate the flight path of the manned spacecraft to the moon and back to Earth.

Let us consider a situation where a spacecraft needs to burn fuel at a high enough temperature to create thrust required to travel to Mars. Fuel such as liquid hydrogen is used in rockets which can reach temperatures of $\sim 3000°C$. This kind of extreme temperature can damage mechanical parts of the engine and cause catastrophic failure. Real-time monitoring of internal temperature inside the combustion chamber is required to mitigate risk. However, a practical constraint does not allow sensors to operate in such high temperatures. An optimum position for the sensor is outside the combustion chamber where it is cooler, however, from this location it is not possible to take direct measurements of the internal temperature. In this scenario, we may use the Kalman filter to find the best estimate of the internal temperature from an indirect measurement - the external temperature. In this way, we can extract information about what we cannot measure by measuring what we can.

The Kalman filter is an on-line algorithm, meaning it sequentially receives observations based on some underlying state changes and uses this to estimate a joint probability distribution $P(\mathbf{x_n}|\mathbf{z_n})$ where $\mathbf{x_n}$ is the state and $\mathbf{z_n}$ are the observations seen up to time n. Under the Bayesian framework, we can describe the joint posterior as follows:

$$P(\mathbf{x_n}|\mathbf{z_n}) = \frac{P(\mathbf{z_n}|\mathbf{x_n})P(\mathbf{x_n}|\mathbf{z_{n-1}})}{P(\mathbf{z_n}|\mathbf{z_{n-1}})} \tag{1}$$

which is the distribution of states $\mathbf{x_n}$ at time n conditioned on all past observations consisting of $\mathbf{z_n} = \{\mathbf{z_1}, \ldots, \mathbf{z_n}\}$. This is formulated under Bayes theorem where $P(\mathbf{z_n}|\mathbf{x_n})$ is the likelihood and $P(\mathbf{x_n}|\mathbf{z_{n-1}})$ is the prior, normalised using $P(\mathbf{z_n}|\mathbf{z_{n-1}})$. The filter is a generative process as it makes stochastic transitions and generates stochastic observations as time passes, sampling transitions from density $\mathcal{N}(\mathbf{F}\mathbf{x_{n-1}}, \mathbf{Q})$ where $\mathbf{F}$ is the state transition matrix and $\mathbf{Q}$ is the process noise. Observations are sampled from density $\mathcal{N}(\mathbf{Hx_n}, \mathbf{R})$ where $\mathbf{H}$ is the observation matrix which defines the relationship between system's internal state and external instrument measurements, and $\mathbf{R}$ is the measurement noise.

At time $n-1$, the state given observation is

$$P(\mathbf{x_{n-1}}|\mathbf{z_{n-1}}) = \mathcal{N}(\mathbf{x_{n-1}}|\mathbf{\hat{x}_{n-1|n-1}}, \mathbf{P_{n-1|n-1}}) \tag{2}$$

where $\mathbf{x_{n-1}}$ is the probability distribution with mean $\mathbf{\hat{x}_{n-1|n-1}}$ and covariance $\mathbf{P_{n-1|n-1}}$ which models the uncertainty due to noise. Next, a predict step is taken when moving from time $n-1$ to n:

$$\begin{aligned} P(\mathbf{x_n}|\mathbf{z_{n-1}}) &= \mathcal{N}(\mathbf{x_n}|\mathbf{\hat{x}_{n|n-1}}, \mathbf{P_{n|n-1}}) \\ &= \mathcal{N}(\mathbf{F}|\mathbf{\hat{x}_{n-1|n-1}}, \mathbf{FP_{n|n-1}}) \end{aligned} \tag{3}$$

At time n, a joint probability distribution is estimated based on:

$$P(\mathbf{x_n}|\mathbf{z_n}) = \mathcal{N}(\mathbf{x_n}|\mathbf{\hat{x}_{n|n}}, \mathbf{P_{n|n}}) \tag{4}$$

where the understanding is updated based on the arrival of new data:

$$\begin{aligned} \mathbf{\hat{x}_{n|n}} &= \mathbf{\hat{x}_{n|n-1}} + \mathbf{P_{n|n-1}}\mathbf{H}^T(\mathbf{HP_{n|n-1}}\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{z_n} - \mathbf{H\hat{x}_{n|n-1}}) \\ &= \mathbf{\hat{x}_{n|n-1}} + \mathbf{k_n}(\mathbf{z_n} - \mathbf{H\hat{x}_{n|n-1}}) \end{aligned} \tag{5}$$

Equation (5) describes that the estimated state at time n given we have seen states up to time n is the result of our state estimate at time n given we have seen states up to time $n-1$ and a Kalman gain $\mathbf{k_n}$ which is also used with observations when updating uncertainty:

$$\begin{aligned} \mathbf{P_{n|n}} &= \mathbf{P_{n|n-1}} - \mathbf{P_{n|n-1}}\mathbf{H}^T(\mathbf{HP_{n|n-1}}\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{HP_{n|n-1}} \\ &= (\mathbf{I} - \mathbf{k_n H})\mathbf{P_{n|n-1}} \end{aligned} \tag{6}$$

$\boldsymbol{k_n}$ takes a value in range $[0, 1]$ and is found by minimising the expected error in posterior state estimate $\mathbb{E}[||\mathbf{x} - \mathbf{x_{n|n}}||^2]$ in the form of:

$$\nabla_{\boldsymbol{k_n}} Tr(\mathbf{P_{n|n}}) \tag{7}$$

minimising the trace of innovation covariance $\mathbf{P_{n|n}}$ with respect to $\boldsymbol{k}(n)$.

2 PRELIMINARIES

In this paper we consider a parameter estimation problem for an auto-regressive (AR) process where sample value measured at time n comes from a weighted sum and some white noise. This generative model for a time series is called auto-regressive as the weighted sum term induces a correlation.

$$s(n) = \sum_{k=1}^{p}(a_k \cdot s(n-k)) + v(n) \tag{8}$$

this AR model is said to be of order p indexed by n as time passes and a_k where $k \in 1, \ldots, p$ are the model parameters. In our initial experiment, we shall observe signals coming from AR process and estimate the underlying parameters a_k by using a Kalman filter. We will then evaluate the quality of this estimate by keeping a_k constant. Next, we shall generate an additional time series where these underlying parameters are slowing changed over time. In this experiment, we wish to observe how well does a Kalman filter estimate and track the changing parameters.

From the perspective of filtering, we let θ be the estimation of underlying a_k. Then, we assume the AR process is making a random walk as the underlying model with some noise w:

$$\theta(n) = \theta(n-1) + w(n) \tag{9}$$

The uni-variate observation we make is the scalar output which is a linear function of underlying state with the input from past values with some perturbation $v(n)$ representing $\mathbf{R}$:

$$y(n) = \theta^T \cdot \mathbf{x_n} + v(n) \tag{10}$$

by observing the outputs $y(n)$, we wish to make an estimate of θ which gets more accurate over time. For the random walk, our state predictions and its uncertainty will be:

$$\begin{aligned} \boldsymbol{\theta}(n|n-1) &= \boldsymbol{\theta}(n-1|n-1) \\ P(n|n-1) &= P(n-1|n) + Q \end{aligned} \tag{11}$$

The innovation $e(n)$ is defined as the difference between the true value and prediction:

$$e(n) = y(n) - \mathbf{x}(n)^T \cdot \boldsymbol{\theta}(n|n-1) \tag{12}$$

The posterior distribution of state estimates will be updated by:

$$\begin{aligned} \boldsymbol{\theta}(n|n) &= \boldsymbol{\theta}(n|n-1) + \mathbf{k}(n) \cdot e(n) \\ P(n|n) &= (\mathbf{I} - \mathbf{k}(n) \cdot \mathbf{x}(n)^T) \cdot P(n|n-1) \end{aligned} \tag{13}$$

where the gain is given by:

$$\boldsymbol{k}(n) = \frac{P(n \mid n-1) \cdot \boldsymbol{x}(n)}{\boldsymbol{R} + \boldsymbol{x}(n)^T \cdot P(n \mid n-1) \cdot \boldsymbol{x}(n)} \tag{14}$$

which tells us that increasing process noise $\mathbf{Q}$ will increase the value and increase the measurement noise $\mathbf{R}$ will decrease the value of the gain term respectively. It is important to note this relationship as we treat treat $\mathbf{Q}$ and $\mathbf{R}$ as hyper-parameters for tuning prediction accuracy.

3 Results

Resembling the simplest case, we monitor an AR process of order $p = 2$ with its underlying parameters kept stationary, which are arbitrarily set to $a_1 = 1.2$ and $a_2 = -0.4$. We treat the noise variances R and $\mathbf{Q}$ as hyper-parameters of the Kalman filter. In this case, the measurement noise R is set to be a fraction of the data variance of the first few samples, and $\mathbf{Q}$, is $\beta \cdot \mathbf{I}$. Note that we treat β as Q henceforth. Figures 2, 3 and 4 show how changing hyper-parameter R varies the convergence properties of the filter.

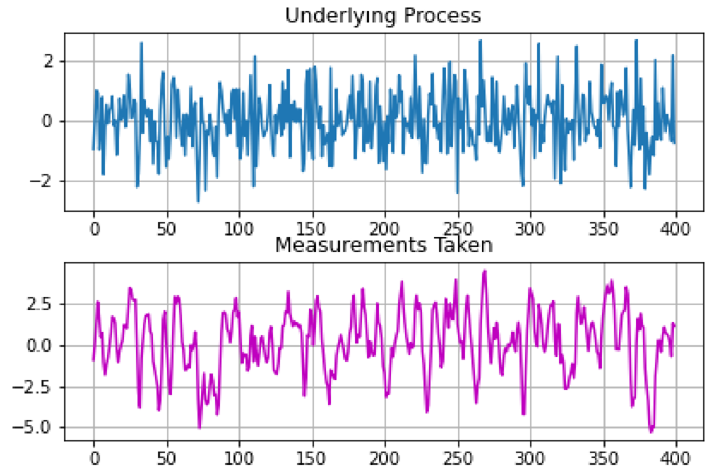

Figure 1: Measurement signals generated from an AR time series of order $p = 2$ which uses Gaussian noise as its underlying process. The underlying model parameters a_k are not changed.

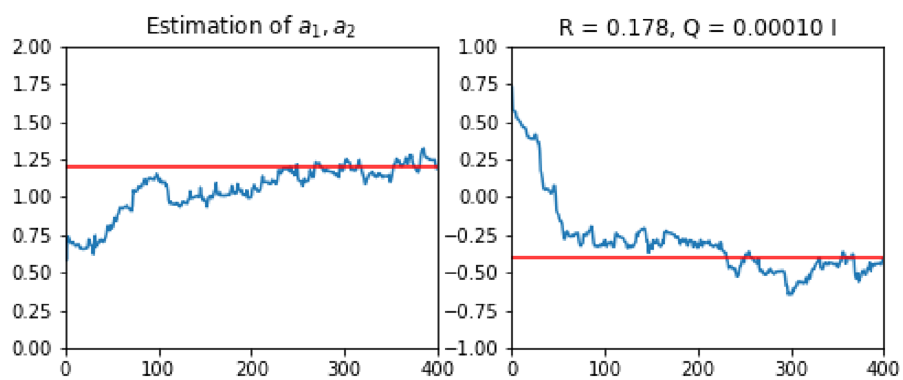

Figure 2: Estimation of true values $a_1 = 1.2$ and $a_2 = -0.4$. The hyper-parameters $R = 0.178$ and $Q = 0.00010$ are used as control and we see successful convergence.

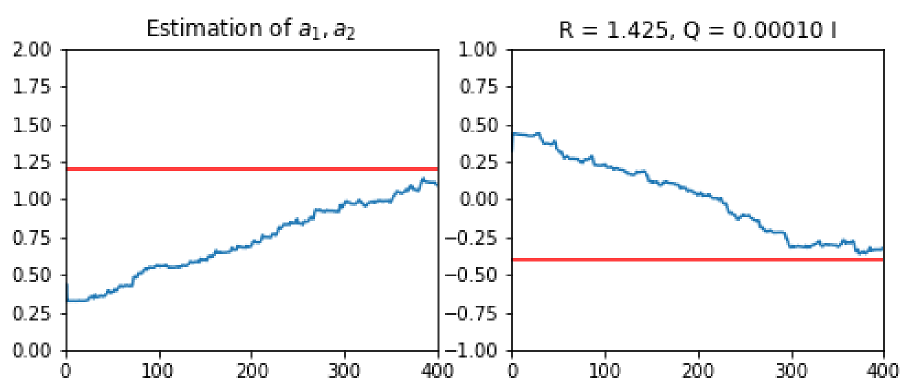

Figure 3: Higher value of R causes a smooth approach but fails to converge.

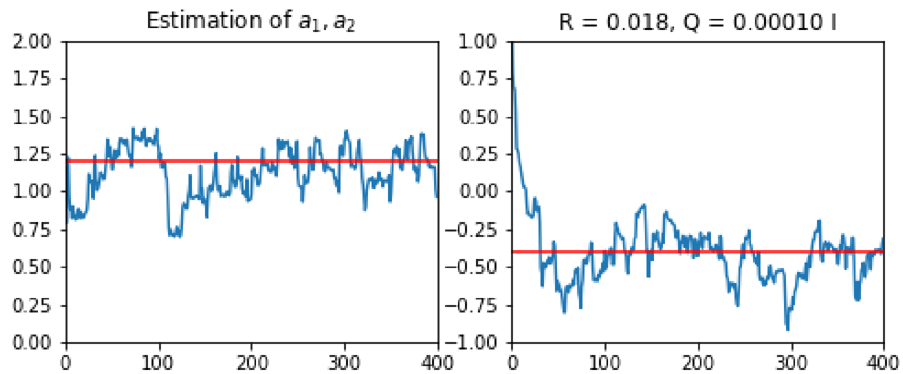

Figure 4: Lower value of R causes a noisy approach and fails to converge.

We investigated the convergence properties by varying hyper-parameter Q, which shows that for both low and high values, convergence fails to take place in different ways, as shown in figures 6 and 5. Then, we investigated the convergence properties of the filter on a time-varying AR process, where the parameter a_k was slowly changed. This process is shown by figure 7.

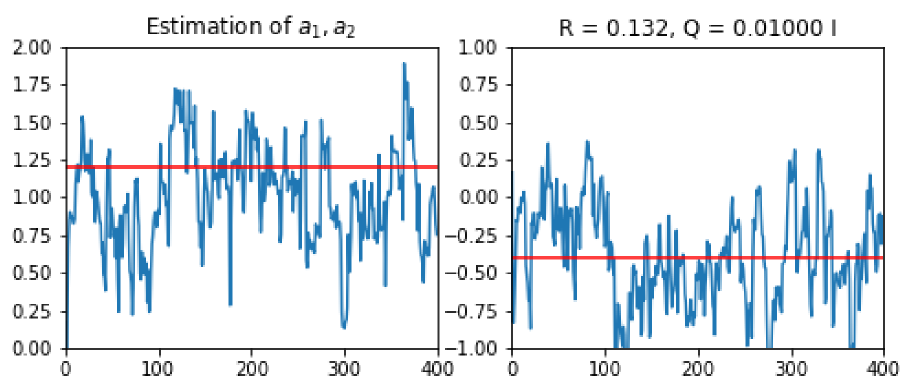

Figure 5: Higher value of Q approaches true parameter value too slowly fails to converge.

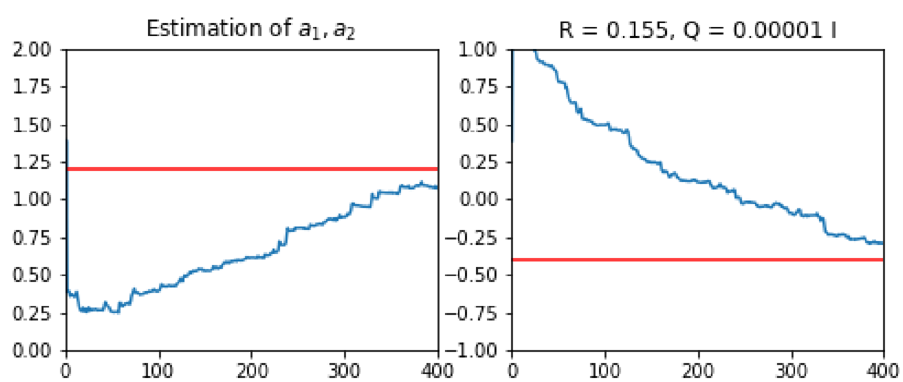

Figure 6: Lower value of Q approaches true parameter value too slowly and fails to converge.

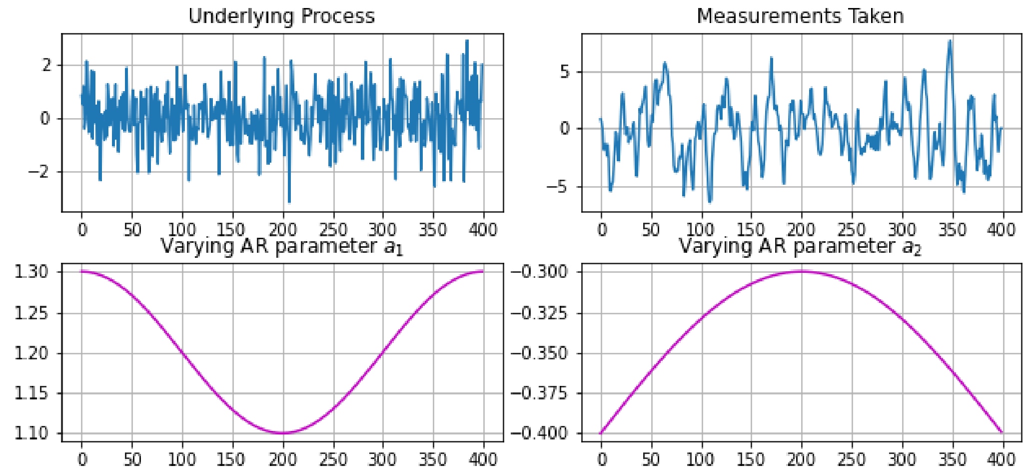

Figure 7: To test the tracking ability of Kalman filter, we slowly change both parameters a_1 and a_2 which causes a change in the way we receive measurement values.

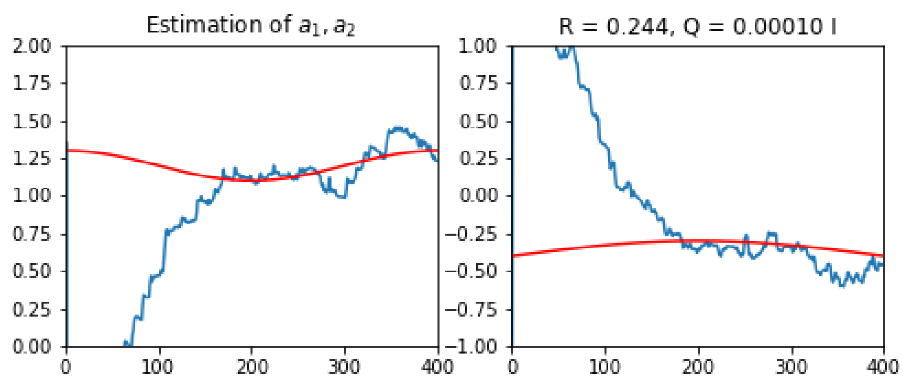

Figure 8: True values of changing parameters a_1 (left) and a_2 (right) are indicated in red. Kalman filter approximations are shown in blue. We find that approximately half the total number of observations need to be made before the filter approximations converge.

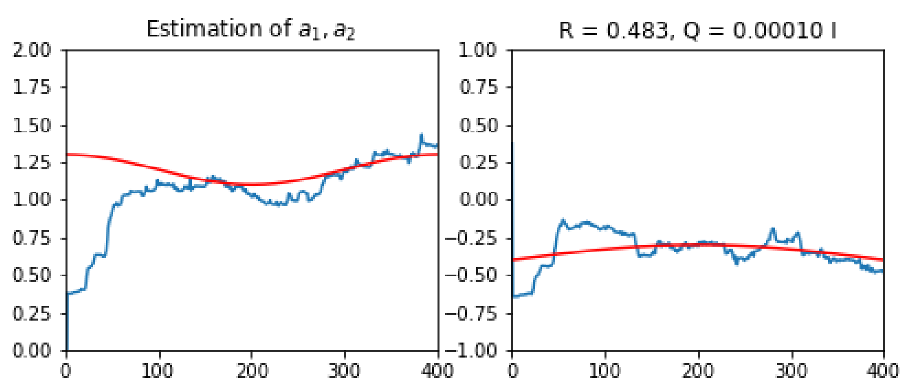

Figure 9: Higher R has resulted in faster approximation and convergence of changing latent parameters.

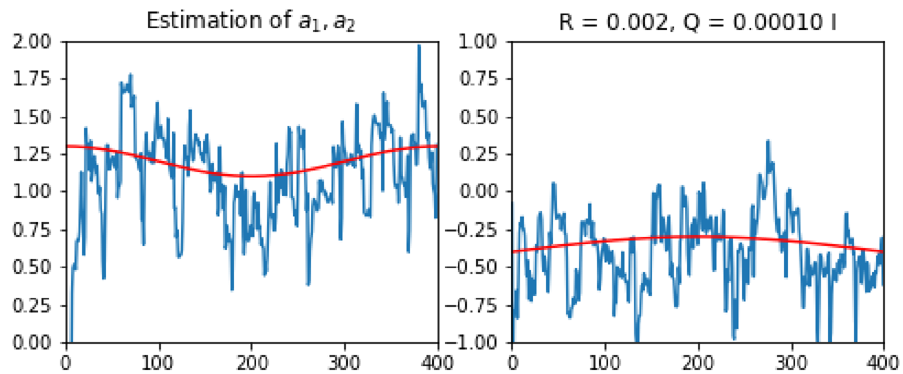

Figure 10: Lower R has resulted in very noisy approximation which would make for very unreliable predictions.

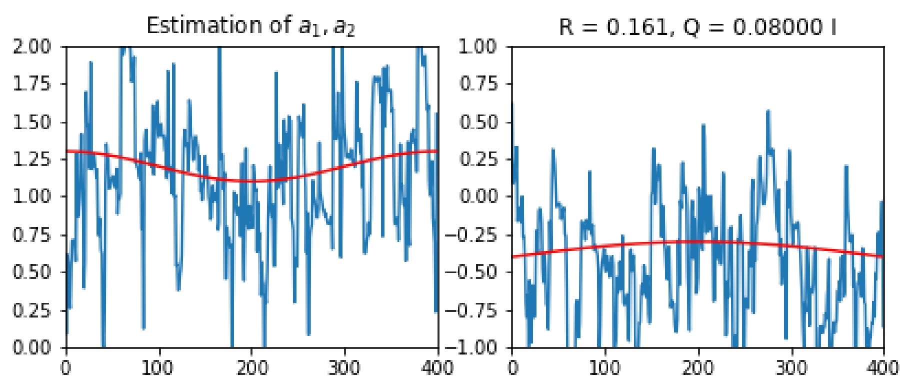

Figure 11: Higher Q has resulted in very noisy approximation which would make for very unreliable predictions.

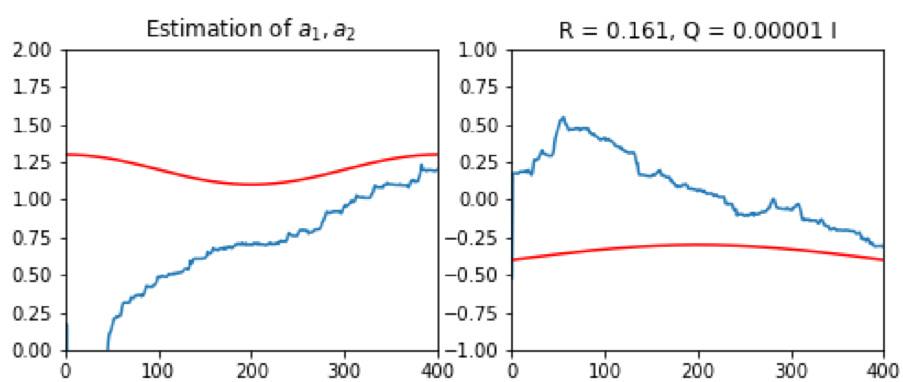

Figure 12: Lower Q has resulted in a smooth approximation but has failed to converge, meaning estimations will still be unreliable.

4 Conclusion

We have defined the Kalman filter, which has a two-step process consisting of a predict and update step. During prediction, it produces estimates of current state variables with uncertainties, which are assumed to be Gaussian. During update, these estimates are revised with higher importance given to states with more certainty. The algorithm is on-line, therefore it can operate in real-time. We have also shown it is an effective parameter estimation model as it has successfully estimated the parameters of a second-order auto-regressive (AR) model which simulates a real-world stochastic process, with the assumption its noise is normally distributed. This gives us the ability to successfully predict the next state of some stochastic system accurately through tuning two hyper-parameters: the process and measurement noise.

Furthermore, we performed Kalman filtering on a second-order AR model performing a random walk, while its parameters were gradually changed. This scenario could be understood, for example, as a simplified model of the stock markets, where two latent factors that are responsible for changing the economy were being tracked by the filter. In our experiments formed of synthetic data, the algorithm has shown that it can successfully predict the next time step into the future, given observations seen up to current point in time.