AIMS CDT Signal Processing. Lab session 2

State space models (Lectures 5 and 6)

In this practical you will derive and implement a Kalman filter for various estimation tasks. There are two parts, the second with various extension. You expected to complete all of them in the time available, but we hope they give some ideas if you intend to use extensions of Kalman processes.

Part 1 - Adaptive Linear Regression

The lecture notes give an example of an adaptive linear regressor (or filter), which consists of a vector of weights $\mathbf{w}$, defining a linear model to predict a noisy output $y$. The noisy timeseries is depicted in Figure 1 below. The data can be found at http://www.robots.ox.ac.uk/~sjrob/AIMS/Lab2/kf1d_signal.mat (note that Matlab files are easy to import into Python too!). Here, $y$ is a function of time, so your prediction equation for the estimate on $y$ is:

$$\hat{y} = \mathbf{w}^T \mathbf{T}_{t,t-L}$$

where $\mathbf{T}$ is the vector of all times, $L$ is the length of the weight vector and $\mathbf{T}_{t,t-L}$ is the vector of the $L$ most recent values of $\mathbf{T}$.

As usual, before you start coding you should write down the equations you are attempting to implement - these are in the lecture notes. The first exercise is to code it up in MATLAB, Python or your favourite development language.

Play around with the signal noise ($\mathbf{R}$), the process noise ($\mathbf{Q}$) and the length of your filter. A reasonable starting value of the process noise is around $10^{-7}$, and the signal noise is 10. Try and find the best combination of values to filter out the noise and match the true signal. It’s worth thinking about how you would, armed with data alone, define an optimisation process for this.

Part 2 - Tracking

There is a boat moving around in a major shipping lane. Unfortunately, the radar used to track boats is inaccurate and regularly drops out. Your task is to find a way to make up for the deficiencies in the radar. The real track and the observed data are shown in Figure 2 below.
You are given a set of data at http://www.robots.ox.ac.uk/~sjrob/AIMS/Lab2/kf2d_signal.mat gathered over the last 15 minutes or so (1000 seconds). There are six vectors:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_n )</td>
<td>Corrupted time vector. Missing some values.</td>
</tr>
<tr>
<td>( x_n )</td>
<td>Missing the same values as ( t_n ), also additive gaussian noise with variance 30.</td>
</tr>
<tr>
<td>( y_n )</td>
<td>Missing the same values as ( t_n ), also additive gaussian noise with variance 30.</td>
</tr>
<tr>
<td>( t )</td>
<td>Full time vector.</td>
</tr>
<tr>
<td>( x )</td>
<td>Full x vector for testing.</td>
</tr>
<tr>
<td>( y )</td>
<td>Full y vector for testing.</td>
</tr>
</tbody>
</table>

You are expected to return, for each time step between 0 and 1000, the mean and variance prediction of your KF. You are given the full \( x \), \( y \) and \( t \) vectors for testing purposes.

**Design of KF**

A strength of the KF is that you can explicitly model the dynamics of your system. Think about the dynamics of the boat - the two controls are the throttle and the rudder. The
throttle controls the boat’s velocity, and the rudder angle and the boat’s velocity combined control the rate of rotation. Then, \( \dot{x} = v \cos(\theta) \) and \( \dot{y} = v \sin(\theta) \).

While it is possible to make a very rich model of the boat, we will start with a **constant velocity model**. This assumes that the boat’s state can be adequately represented by \([x, y, \dot{x}, \dot{y}]\), and acceleration is absorbed into the process noise.

Before coding, start by writing down the plant model (\(F\)), the observation model (\(G\)) along with the process noise model (\(J\)) and the observation noise model (\(V\)) for this simple KF. What values will you use for the observation and process noise?

Once you have done this, try a richer model such as the constant acceleration (state is \([x, y, \dot{x}, \dot{y}, \ddot{x}, \ddot{y}]\)) or an EKF\(^1\) based model taking into account the non-linear dynamics of the boat.

For anyone feeling very adventurous, there is an elegant link between the KF and **Gaussian Processes**, which you have already been working with. Some details of these relationships, along with the form of *kernel function* for a GP associated with e.g. constant velocity and constant acceleration KF models can be found in recent papers.\(^2\)\(^3\)

\(^1\)http://en.wikipedia.org/wiki/Extended_Kalman_filter
\(^2\)http://www.robots.ox.ac.uk/~sjrob/Pubs/cam_gp.pdf
\(^3\)http://www.robots.ox.ac.uk/~sjrob/Pubs/Fusion2010_0132.pdf.