V. Streaming Solutions for Least-Squares Problems

In our discussion of least-squares so far, we have focussed on STATIC problems. A set of measurements

$$y = Ax + \text{noise}$$

comes in all at once, and we use them all to estimate x.

In this section, we will shift our focus to streaming problems. We observe

$$\begin{aligned} y_0 &= A_0 x + \text{noise} \\ y_1 &= A_1 x + \text{noise} \\ &\vdots \\ y_k &= A_k x + \text{noise} \\ &\vdots \end{aligned}$$

At each time k, we want to form the best estimate of x from the observations $y_0, y_1, \ldots, y_k$ seen up to that point.

Moreover, we would like to do this in an efficient manner. The size of the problem is growing with k — rather than resolving the problem from scratch, we would like a principled ($ fast) way to update the solution when a new observation is made.

We will consider two basic frameworks:

① Recursive Least Squares (RLS):
The vector x does not vary

② The Kalman Filter:
The vector x_k moves at every time step and we have a (linear) model for how it moves.

Note: The measurement matrices $A_0, A_1, \ldots$ can be different, and can even have a different number of rows. We will work under the assumption that the total number of measurements we have seen at any point exceeds the number of unknowns, and if we form

$$\underline{A}_K = \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_K \end{bmatrix}$$

then $(\underline{A}_K^T \underline{A}_K)^{-1}$ exists.

This assumption is not really necessary — it just makes the discussion easier — and generalizing is not that hard.

The foundation of our discussion is the matrix inversion lemma, also known as the Sherman-Morrison-Woodbury equation.

The Matrix Inversion Lemma

Let A, B, C be matrices as follows:

A: $N \times N$, invertible

B: $p \times N$

C: $p \times p$, invertible

Then

$$\boxed{(A + B^T C B)^{-1} = A^{-1} - A^{-1} B^T (C^{-1} + B A^{-1} B^T)^{-1} B A^{-1}}$$

The basic idea is that the $N \times N$ inverse $(A + B^T C B)^{-1}$ can be written as an update to A^{-1} by computing a $p \times p$ inverse $(C^{-1} + B A^{-1} B^T)^{-1}$.

A special (and often used case) is when $C = I$:

$$\boxed{(A + B^T B)^{-1} = A^{-1} - A^{-1} B^T (I + B A^{-1} B^T)^{-1} B A^{-1}}$$

proof of M.I.L.

We want to solve

$$(A + B^T C B)\, w = v$$

for any right-hand side v.

Set $z = CBw \Rightarrow C^{-1} z = Bw$

Now we have two sets of equations

$$\begin{aligned} Aw + B^T z &= v \\ Bw - C^{-1} z &= 0 \end{aligned}$$

Manipulating the first equation yields

$$w = A^{-1}(v - B^T z)$$

Plugging this into the second equation gives us

$$BA^{-1}v - BA^{-1}B^T z - C^{-1} z = 0$$

$$\Rightarrow \quad z = (C^{-1} + BA^{-1}B^T)^{-1} BA^{-1} v$$

$$\Rightarrow w = A^{-1}v - A^{-1}B^T (C^{-1} + BA^{-1}B^T)^{-1} BA^{-1} v$$

$$= (A^{-1} - A^{-1}B^T (C^{-1} + BA^{-1}B^T)^{-1} BA^{-1}) v$$

Q: Suppose A^{-1} has been pre-calculated. How many operations are needed to solve

$$(A + B^T B)\, w = v \quad ?$$

Updating Least-Squares Solutions

Suppose we have observed

$$y_0 = A_0 x + \text{noise}$$

and have formed the estimate

$$\hat{x}_0 = (A_0^T A_0)^{-1} A_0^T y_0$$

Now we observe

$$y_1 = A_1 x + \text{noise}$$

Given y_0 & y_1, the least-squares estimate formed from

$$\begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} A_0 \\ A_1 \end{bmatrix} x + \text{noise}$$

is

$$\hat{x}_1 = (A_0^T A_0 + A_1^T A_1)^{-1} (A_0^T y_0 + A_1^T y_1)$$

Now let

$$P_0 = (A_0^T A_0)^{-1}$$

$$P_1 = (A_0^T A_0 + A_1^T A_1)^{-1}$$

(so $P_1^{-1} = A_0^T A_0 + A_1^T A_1$

$= P_0^{-1} + A_1^T A_1$)

Then using the matrix inversion lemma

$$P_1 = P_0 - P_0 A_1^T (I + A_1 P_0 A_1^T)^{-1} A_1 P_0$$

For example, suppose we see just one new measurement

$$y_1 = a_1^T x + \text{noise} \qquad a_1 \in \mathbb{R}^N$$

then

$$\hat{x}_1 = \left[P_0 - P_0 a_1 (1 + a_1^T P_0 a_1)^{-1} a_1^T P_0 \right] \left(A_0^T y + y_1 \cdot a_1 \right)$$

set $u = P_0 a_1$. Then

$$\hat{x}_1 = \hat{x}_0 + y_1 u - \frac{(a_1^T \hat{x}_0)}{1 + a_1^T u} \cdot u - \frac{y_1 (a_1^T u)}{1 + a_1^T u} \cdot u$$

$$= \hat{x}_0 + \left(\frac{1}{1 + a_1^T u} \right) (y_1 - a_1^T \hat{x}_0) \cdot u$$

$\Rightarrow$ we can update the solution with one vector-matrix multiply and two inner products.

In general, we have

$$\hat{x}_1 = P_1 (A_0^T y_0 + A_1^T y_1)$$

$$= P_1 (P_0^{-1} \hat{x}_0 + A_1^T y_1)$$

and since

$$P_0^{-1} = P_1^{-1} - A_1^T A_1$$

$$\Rightarrow \hat{x}_1 = P_1 (P_1^{-1} \hat{x}_0 - A_1^T A_1 \hat{x}_0 + A_1^T y_1)$$

$$= \hat{x}_0 + K_1 (y_1 - A_1 \hat{x}_0)$$

where

$$K_1 = P_1 A_1^T$$

It is pretty clear that we can generalize this to moving from step k-1 to step k, keep track of $\hat{x}_k$ and

$$P_k = (\underline{A}_k^T \underline{A}_k)^{-1} = (A_0^T A_0 + A_1^T A_1 + A_2^T A_2 + \cdots + A_k^T A_k)^{-1}$$

Recursive Least Squares Algorithm

Given
$$y_0 = A_0 x + \text{noise}$$
$$y_1 = A_1 x + \text{noise}$$
$$\vdots$$
$$y_K = A_K x + \text{noise}$$
$$\vdots$$

RLS is an online algorithm for computing the best estimate for x — call it $\hat{x}_K$ — from $y_1, y_2, \ldots, y_K$.

Intialize:

(y_0 appears)

$$P_0 = (A_0^T A_0)^{-1}$$
$$\hat{x}_0 = P_0 A_0^T y_0$$

Loop over $K = 1, 2, \ldots$

(y_K appears)

$$P_K = P_{K-1} - P_{K-1} A_K^T (I + A_K P_{K-1} A_K^T)^{-1} A_K P_{K-1}$$
$$K_K = P_K A_K^T$$
$$\hat{x}_K = \hat{x}_{K-1} + K_K (y_K - A_K \hat{x}_{K-1})$$

The Kalman Filter

Our recursive least-squares algorithm for updating our estimate given a series of observation vectors looked something like a filter: new data comes in, and we use it (along with collected knowledge of the old data) to produce a new output.

In our recursive LS discussion above, x was fixed for the entire series of observations.

The Kalman filter incorporates dynamics for x into our estimation framework. It addresses the case where x changes (in a somewhat predictable way) from observation to observation.

The classic example is trying to estimate the position and velocity of an airplane. If the plane is in motion, these will of course change over time. But the path of the plane is somewhat predictable (it can't turn or change speeds too quickly).

How do we incorporate these dynamics into our least-squares framework?

Here is the model:
As before, we make noisy observations of an unknown vector

$$y_k = A_k x_k + e_k \qquad (e_k = \text{"measurement error"})$$

But now, x_k changes. This change is modeled by the action of a known $N \times N$ matrix F_k:

$$x_{k+1} = F_k x_k + \varepsilon_k \qquad (\varepsilon_k = \text{"state error"})$$

We will assume that:

1. e_k has covariance matrix R_k
2. e_k and e_j are uncorrelated for $j \neq k$
3. ε_k has covariance matrix Q_k
4. ε_k and ε_j are uncorrelated for $j \neq k$
5. ε_k and e_j are uncorrelated $\forall\, j, k$

Say we make 3 observations. Our equations are then

$$A_0 x_0 = y_0 \quad \text{(w/ error } e_0\text{)}$$
$$F_0 x_0 - x_1 = 0 \quad \text{(w/ error } \varepsilon_0\text{)}$$
$$A_1 x_1 = y_1 \quad \text{(w/ error } e_1\text{)}$$
$$F_1 x_1 - x_2 = 0 \quad \text{(w/ error } \varepsilon_1\text{)}$$
$$A_2 x_2 = y_2 \quad \text{(w/ error } e_2\text{)}$$

So the system $Ax = y$ we are interested in is

$$\begin{bmatrix} A_0 & 0 & 0 \\ F_0 & -I & 0 \\ 0 & A_1 & 0 \\ 0 & F_1 & -I \\ 0 & 0 & A_2 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ 0 \\ y_1 \\ 0 \\ y_2 \end{bmatrix}$$

The covariance matrix is

$$\begin{bmatrix} R_0 & 0 & \cdots & & \\ 0 & Q_0 & & & \\ \vdots & & R_1 & & \\ \vdots & & & Q_1 & \\ & & & & R_2 \end{bmatrix}$$

Note that now we are estimating x_0, x_1, x_2 at each step. Every time we receive a new observation, we update the current $\hat{x}_K$ ("filtering") as well as our estimates for $\hat{x}_0, \hat{x}_1, \dots, \hat{x}_{K-1}$ ("smoothing").

Notation:

$\hat{x}_{K|i}$ = estimate of x_K after the i^{th} observation

$P_{K|i}$ = information matrix used to update x_K after the i^{th} observation

The naive way to solve the above is to construct

$$\begin{bmatrix} A_0 & & & & \\ F_0 & -I & & & \\ & A_1 & & & \\ & F_1 & \ddots & & \\ & & & F_{K-1} & -I \\ & & & & A_K \end{bmatrix}$$

and solve for $\hat{x}_{0|K}, \hat{x}_{1|K}, \hat{x}_{2|K}, \dots, \hat{x}_{K|K}$ explicitly (matrix gets huge).

Instead, we can adapt the ideas from R.L.S. to solve for the most current estimate sequentially.

The derivation is a little trickier, since now there is a "link" between X_0 and X_1 (and X_k and X_{k+1} in general) — not only are we adding more rows to the fundamental system, we are adding more columns as well.

We will do the derivation for the update for the fundamental least-squares problem

$$\begin{bmatrix} y_0 \\ 0 \\ y_1 \\ 0 \\ \vdots \\ y_k \end{bmatrix} = \begin{bmatrix} A_0 & 0 & 0 & \cdots & \\ F_0 & -I & 0 & \cdots & \\ 0 & A_1 & 0 & \cdots & \\ 0 & F_1 & -I & & \\ \vdots & \vdots & A_2 & \ddots & \\ & & & F_{k-1} & -I \\ & & & & A_k \end{bmatrix} \begin{bmatrix} X_{0|k} \\ X_{1|k} \\ \vdots \\ X_{k|k} \end{bmatrix} + \text{noise}$$

$$\underline{Y}_k = \underline{\underline{M}}_k \underline{X}_k + \text{noise}$$

Here, the least-squares estimate is

$$\underline{\hat{X}}_K = \begin{bmatrix} \hat{X}_{0|K} \\ \hat{X}_{1|K} \\ \vdots \\ \hat{X}_{K|K} \end{bmatrix} = \left(\underline{M}_K^T \underline{M}_K\right)^{-1} \underline{M}_K^T \underline{y}_K$$

This is the same as the BLUE when we take all the covariance matrices to be the identity

$$R_K = I, \quad Q_K = I \qquad \forall K$$

Extending this discussion to general R_K, Q_K is straightforward – we will give the general BLUE equations at the end.

We will only be concerned with computing the <u>most recent</u> segment of $\underline{\hat{X}}_K$, we call this $\hat{X}_{K|K}$. As we will see below, there is an efficient way to do this without having to compute the rest of the segments in $\underline{\hat{X}}_K$.

We can, however, work backwards to compute $\hat{x}_{k-1|k}$, $\hat{x}_{k-2|k}$, etc once $\hat{x}_{k|k}$ is computed. We will not discuss this further here — there are plenty of great references on this if you want the details.

We start by observing the first set of measurements

$$y_0 = A_0 x_0 + \text{noise}$$

and forming the estimate

$$\hat{x}_{0|0} = P_0 A_0^T y \qquad P_0 = (A_0^T A_0)^{-1}$$

Next, we predict the motion of x, incorporating

$$0 = F_0 x_0 - x_1 + \underline{\text{noise}}$$

↳ since the motion model may not be perfect

This adds rows and columns to our system

$$\underbrace{\begin{bmatrix} A_0 & 0 \\ F_0 & -I \end{bmatrix}}_{G_0} \begin{bmatrix} x_{010} \\ x_{110} \end{bmatrix} + \text{noise} = \begin{bmatrix} y_0 \\ 0 \end{bmatrix}$$

The least-squares solution to this system obeys

$$G_0^T G_0 \begin{bmatrix} \hat{x}_{010} \\ \hat{x}_{110} \end{bmatrix} = G_0^T \begin{bmatrix} y_0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} A_0^T A_0 + F_0^T F_0 & -F_0^T \\ -F_0 & I \end{bmatrix} \begin{bmatrix} \hat{x}_{010} \\ \hat{x}_{110} \end{bmatrix} = \begin{bmatrix} A_0^T & F_0^T \\ 0 & -I \end{bmatrix} \begin{bmatrix} y_0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} A_0^T y_0 \\ 0 \end{bmatrix}$$

To compute the inverse of $G_0^T G_0$, we need to be able to solve

$$\begin{bmatrix} A_0^T A_0 + F_0^T F_0 & -F_0^T \\ -F_0 & I \end{bmatrix} \begin{bmatrix} w \\ z \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}$$

for any u, v. We have

$$\begin{aligned} (A_0^T A_0 + F_0^T F_0) w - F_0^T z &= u \\ -F_0 w + z &= v \end{aligned}$$

so

$$z = v + F_0 w$$

and

$$A_0^T A_0 w + F_0^T F_0 w - F_0^T v - F_0^T F_0 w = u$$

$$\Rightarrow \quad A_0^T A_0 w = u + F_0^T v$$

$$\Rightarrow \quad w = P_0 u + P_0 F_0^T v \qquad P_0 = (A_0^T A_0)^{-1}$$

$$z = F_0 P_0 u + F_0 P_0 F_0^T v + v$$

$$\Rightarrow (G_0^T G_0)^{-1} = \begin{bmatrix} P_0 & P_0 F_0^T \\ F_0 P_0 & P_{1|0} \end{bmatrix}$$

where

$$P_{1|0} = I + F_0 P_0 F_0^T$$

So $$\begin{bmatrix} \hat{x}_{0|0} \\ \hat{x}_{1|0} \end{bmatrix} = \begin{bmatrix} P_0 A_0^T y_0 \\ F_0 P_0 A_0^T y_0 \end{bmatrix} = \begin{bmatrix} \hat{x}_{0|0} \\ F_0 \hat{x}_{0|0} \end{bmatrix}$$

As we would hope, we haven't made any new measurements yet, so our estimate of x_0 hasn't changed. Our best estimate (right now) of x_1 is simply $\hat{x}_{1|0} = F_0 \hat{x}_{0|0}$ — the motion matrix applied to the last estimate.

Now we incorporate the new measurements, making the system

$$\underbrace{\begin{bmatrix} A_0 & 0 \\ F_0 & -I \\ 0 & A_1 \end{bmatrix}}_{M_1} \begin{bmatrix} x_{0|1} \\ x_{1|1} \end{bmatrix} + \text{noise} = \begin{bmatrix} y_0 \\ 0 \\ y_1 \end{bmatrix}$$

Since we have just added rows, we can use RLS to update the solution.

$$\begin{bmatrix} \hat{x}_{011} \\ \hat{x}_{111} \end{bmatrix} = \begin{bmatrix} \hat{x}_{010} \\ \hat{x}_{110} \end{bmatrix} + K_1 \left(y_1 - \begin{bmatrix} 0 & A_1 \end{bmatrix} \begin{bmatrix} \hat{x}_{010} \\ \hat{x}_{110} \end{bmatrix} \right)$$

$$= \begin{bmatrix} \hat{x}_{010} \\ \hat{x}_{110} \end{bmatrix} + K_1 (y_1 - A_1 \hat{x}_{110})$$

where

$$K_1 = (\underline{M}_1^T \underline{M}_1)^{-1} \begin{bmatrix} 0 \\ A_1^T \end{bmatrix}$$

We can use the matrix inversion lemma to compute K_1:

$$(\underline{M}_1^T \underline{M}_1)^{-1} \begin{bmatrix} 0 \\ A_1^T \end{bmatrix} = \left(G_0^T G_0 + \begin{bmatrix} 0 \\ A_1^T \end{bmatrix} \begin{bmatrix} 0 & A_1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ A_1^T \end{bmatrix}$$

$$= J_0 \begin{bmatrix} 0 \\ A_1^T \end{bmatrix} - J_0 \begin{bmatrix} 0 \\ A_1^T \end{bmatrix} \left(I + \begin{bmatrix} 0 & A_1 \end{bmatrix} J_0 \begin{bmatrix} 0 \\ A_1^T \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 & A_1 \end{bmatrix} J_0 \begin{bmatrix} 0 \\ A_1^T \end{bmatrix}$$

where

$$J_0 = (G_0^T G_0)^{-1} = \begin{bmatrix} P_0 & P_0 F_0^T \\ F_0 P_0 & P_{110} \end{bmatrix}$$

Note that

$$J_0 \begin{bmatrix} 0 \\ A_1^T \end{bmatrix} = \begin{bmatrix} P_0 F_0^T A_1^T \\ P_{1|0} A_1^T \end{bmatrix}$$

and

$$\begin{bmatrix} 0 & A_1 \end{bmatrix} J_0 \begin{bmatrix} 0 \\ A_1^T \end{bmatrix} = A_1 P_{1|0} A_1^T$$

Thus

$$K_1 = \begin{bmatrix} P_0 F_0^T A_1^T \\ P_{1|0} A_1^T \end{bmatrix} - \begin{bmatrix} P_0 F_0^T A_1^T \\ P_{1|0} A_1^T \end{bmatrix} (I + A_1 P_{1|0} A_1^T)^{-1} A_1 P_{1|0} A_1^T$$

$$= \begin{bmatrix} P_0 F_0^T A_1^T \\ P_{1|0} A_1^T \end{bmatrix} \left(I - (I + A_1 P_{1|0} A_1^T)^{-1} A_1 P_{1|0} A_1^T \right)$$

For any matrix M with $I+M$ invertible, it is a fact that

$$I - (I+M)^{-1} M = (I+M)^{-1}$$

(simply multiply both sides by $I+M$)

Thus

$$K_1 = \begin{bmatrix} P_0 F_0^T A_1^T \\ P_{1|0} A_1^T \end{bmatrix} (I + A_1 P_{1|0} A_1^T)^{-1}$$

If we are only interested in the estimate of the most current x, we can take the bottom segment of these equations to get

$$\hat{x}_{1|1} = \hat{x}_{1|0} + \underline{K}_{1,1}\left(y_1 - A_1 \hat{x}_{1|0}\right)$$

$$K_{1,1} = P_{1|0} A_1^T \left(I + A_1 P_{1|0} A_1^T\right)^{-1}$$

Finally we can update the "information matrix" for the next iteration (i.e. $P_{1|1}$) efficiently in terms of $\underline{K}_{1,1}$.

Notice that once $\hat{x}_{0|0}$ has been calculated and extrapolated into $\hat{x}_{1|0} = F_0 \hat{x}_{0|0}$, we no longer need y_0 or A_0 (or even $\hat{x}_{0|0}$...) — all of the information about what has happened in the past has been folded into $\hat{x}_{1|0}$.

Now, what was the equivalent "information matrix" used to form $\hat{x}_{1|0}$?

Recall the "predict" normal system

$$\begin{bmatrix} A_0^T A_0 + F_0^T F_0 & -F_0^T \\ -F_0^T & I \end{bmatrix} \begin{bmatrix} \hat{x}_{0|0} \\ \hat{x}_{1|0} \end{bmatrix} = \begin{bmatrix} A_0^T y_0 \\ 0 \end{bmatrix}$$

Since $\hat{x}_{1|0}$ was completely determined by $\hat{x}_{0|0}$, we could have solved for $\hat{x}_{1|0}$ directly by taking

$$\hat{x}_{0|0} = (A_0^T A_0 + F_0^T F_0)^{-1} (A_0^T y + F_0^T x_1)$$

and then

$$\hat{x}_{1|0} = \underbrace{\left(I - F_0 (A_0^T A_0 + F_0^T F_0)^{-1} F_0^T\right)^{-1}} F_0 (A_0^T A_0 + F_0^T F_0)^{-1} A_0^T y$$

$$= F_0 P_0 F_0^T + I \qquad \text{(using matrix inversion lemma)}$$

$$= P_{1|0}$$

So $\hat{x}_{1|0}$ is a least-squares estimate formed with information matrix $P_{1|0}$.

Then, given the new measurements of x_1, we can update the information matrix just as in RLS:

$$P_{1|1}^{-1} = P_{1|0}^{-1} + A_1^T A_1$$

Again applying the matrix inversion lemma, we can write this as

$$P_{1|1} = (I - K_{1,1} A_1) P_{1|0}$$

(check this at home)

Kalman Filter

Given the optimal estimate $\hat{x}_{K|K}$ and info matrix $P_{K|K}$, here is how we update given new measurements y_K:

State update extrapolation

$$\hat{x}_{K+1|K} = F_K \hat{x}_{K|K}$$

Error covariance extrapolation

$$P_{K+1|K} = F_K P_{K|K} F_K^T + I$$

Kalman gain

$$K_{K+1,K+1} = P_{K+1|K} A_{K+1}^T (A_{K+1} P_{K+1|K} A_{K+1}^T + I)^{-1}$$

State update:

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1,k+1}\left(y_{k+1} - A_{k+1}\hat{x}_{k+1|k}\right)$$

Covariance update:

$$P_{k+1|k+1} = \left(I - K_{k+1,k+1}A_{k+1}\right)P_{k+1|k}$$

To incorporate covariance matrices for the measurement error, $R_k = E[e_k e_k^T]$, and the state error, $Q_k = E[\Sigma_k \Sigma_k^T]$, we have the following modifications:

Error covariance extrapolation

$$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_k$$

Kalman gain

$$K_{k+1,k+1} = P_{k+1|k} A_{k+1}^T \left(A_{k+1} P_{k+1|k} A_{k+1}^T + R_k\right)^{-1}$$

(You can derive these at home!)

RLS vs. Kalman

This simple example illustrates the essential difference between modeling x as static and using RLS, or incorporating dynamics and using the Kalman filter.

Suppose we are taking the pulse of a patient. We will make a series of measurements, each one with a little bit of error.

Static (RLS):

We model the measurement at time k with the scalar equation

$$y_k = 1 \cdot x + \text{noise}$$

where x = the pulse is fixed. We know that

$$\hat{x}_{k-1} = \left(\begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_0 \\ \vdots \\ y_{k-1} \end{bmatrix}$$

$$= \frac{y_0 + y_1 + \cdots + y_{k-1}}{k}$$

and $P_K = \frac{1}{K}$

Given

$$y_K = x + e_K$$

we can compute

$$P_K^{-1} = P_{K-1}^{-1} + A_K^T A_K$$

$$= K + 1$$

$$\Rightarrow P_K = \frac{1}{K+1} \qquad \text{(as expected)}$$

and

$$K_K = \frac{1}{K+1},$$

$$\hat{x}_K = \hat{x}_{K-1} + \frac{1}{K+1}\left(y_K - \hat{x}_{K-1}\right)$$

$$= \frac{\sum_{j=0}^{K-1} y_j}{K} + \frac{y_K}{K+1} - \frac{\sum_{j=0}^{K-1} y_j}{K\cdot(K+1)}$$

$$= \left(1 - \frac{1}{K+1}\right)\frac{\sum_{j=0}^{K-1} y_j}{K} + \frac{y_K}{K+1}$$

$$= \frac{\sum_{j=0}^{K} y_j}{K+1}$$

So the static least-squares estimate is the average of all previous observations.

Dynamic (Kalman):

Now along with the measurements

$$y_k = x_k + \text{noise}$$

We will assume the pulse drifts in between measurements

$$x_k = x_{k-1} + \text{error}$$

(i.e. $F_{k-1} = 1$ for all k). This model says we expect the pulse to be the same at time k as it was at time $k-1$, but recognizes that it might not be.

① Write down the full Kalman system for $k = 0, 1, 2$

(2) Calculate $\hat{x}_{111}$ and $\hat{x}_{010}$

(3) Calculate $\hat{x}_{2|2}$, $\hat{x}_{1|2}$, $\hat{x}_{0|2}$

(4) What is the difference between the result for $\hat{x}_{212}$ and the corresponding RLS estimate?
Why is this happening?