Improving the Kalman Filter

Our current Kalman filter regularization for SAN is not converging as expected. One possible reason could lie in the fundamental setup of the basic SAN and our KFSAN. In basic SAN, we add a regularization constant to the denominator of p and q:

$$p = \frac{I_1^+ - I_1^-}{2\left(I_1^+ + I_1^- - 2I_0\right)} = \frac{N_p}{D_p},\ q = \frac{I_2^+ - I_2^-}{2\left(I_2^+ + I_2^- - 2I_0\right)} = \frac{N_q}{D_q}.$$

But in KFSAN, if the denominator D is small, we just set $p, q = 0$ there. That's because one thing we want to achieve in KFSAN is not having to inflate the denominator. But by doing this, we make the setup different and KFSAN will not properly return to basic SAN at even high SNR.

Additionally, I have a few considerations:

- Ideally, when signal-to-noise is very high, KFSAN will trust the previous state only and returns to basic SAN. But in Kalman filter we use R, not SNR=S/R. If SNR is high, R is small and it will trust the measurement, not the previous state.
- Do we really want to "return to basic SAN?" Because this means we'd have to regularize D at some point.

With these in mind, I propose a new Kalman filter setup that tries to avoid division by zero:

The setup is very similar to our current KF. I only changed 6 and 7 so that the measurement is the numerator and H is the denominator. State vector stays as p, q. This way, we avoid the division in the measurement process.

	Definition	Function
1	$x = \begin{pmatrix} p \\ q \end{pmatrix}$	state vector (unknown). p, q are the speckle EF parameters derived in SAN.
2	$\Phi = I$	state transition matrix: models how the state evolves. Here we assume the state is constant.
3	$\hat{x}^-, \hat{x}^+$	state estimate, before and after measurement.
4	$u = -\hat{x}^+$	control input. This is the anti-speckle we apply to null the speckle.
5	$\Gamma = I$	propagation matrix: convert control input into how it changes the state vector.
6	$z = Hx + \mathbf{N} = \begin{pmatrix} N_p \\ N_q \end{pmatrix}$	measurement. $\mathbf{N}$ is the measurement noise.
7	$H = \begin{pmatrix} D_p & 0 \\ 0 & D_q \end{pmatrix}$	linear observation matrix: convert measurement into how it changes the state vector.
8	K	The Kalman gain. Based on the state covariance and measurement noise, it tells the algorithm to trust the measurement more or the prediction more.
9	$P^-, P^+ = \begin{pmatrix} \sigma_p^2 & 0 \\ 0 & \sigma_q^2 \end{pmatrix}$	state covariance matrix, before and after measurement.
10	Q	process noise matrix: noise from DM actuators.
11	$R = \begin{pmatrix} \mathbf{N}_p^2 & 0 \\ 0 & \mathbf{N}_q^2 \end{pmatrix}$	measurement noise matrix

The iteration of Kalman filter can be summarized as follows:

$$\hat{x}_k^- = \hat{x}_{k-1}^+ + u_{k-1}, \;\; P_k^- = P_{k-1}^+ + Q_{k-1}$$
$$K_k = P_k^- \, H_k^\mathsf{T} \left(H_k \, P_k^- \, H_k^\mathsf{T} + R_k\right)^{-1}$$
$$\hat{x}_k^+ = \hat{x}_k^- + K_k\big(z_k - H_k \, \hat{x}_k^-\big), \;\; P_k^+ = \left[(P_k^-)^{-1} + H_k^\mathsf{T} R_k^{-1} H_k\right]^{-1}$$

For each pixel, this simplifies to:

$$K_{p,k} = \frac{P_{p,k}^- \, D_{p,k}}{D_{p,k}^2 \, P_{p,k}^- + R_{p,k}}$$
$$\hat{p}_k^+ = \hat{p}_k^- + K_{p,k}\left(N_{p,k} - D_{p,k} \, \hat{p}_k^-\right)$$
$$P_{p,k}^+ = \frac{P_{p,k}^- \, R_{p,k}}{D_{p,k}^2 \, P_{p,k}^- + R_{p,k}}$$

and similarly for q.

The measurement noise $\mathbf{N}$ is still:

$$\mathbf{N} = \sqrt{N_* + n_{pix}(N_S + N_D + N_R^2)}, \tag{1}$$

where N_* is the total number of photons collected from object of interest, n_{pix} is the number of pixels under consideration (in SAN this would usually be 1), N_s is the total number of photons/pixel from background/sky, N_D is the total number of dark current electrons/pixel, N_R is the total number of electrons/pixel from read noise.

1 Estimating P_0 and R

For P_0, since we didn't change x, we could use the same P_0 as before.

The measurement noise R is much simpler in this setup. Using $\mathrm{Var}(N_p) = \mathrm{Var}(I_{1+}) + \mathrm{Var}(I_{1-})$, $\mathrm{Var}(N_q) = \mathrm{Var}(I_{2+}) + \mathrm{Var}(I_{2-})$,

$$R = \mathrm{diag}(\mathrm{Var}(N_p), \mathrm{Var}(N_q)),$$

where

$$\mathrm{Var}(I) = \frac{I}{G} + \frac{N_S + N_D + N_R^2}{G^2},$$

same as in our old KF.

Under this new KF setup, we make the H matrix the denominator of SAN coefficients. This sacrifices/ignores the noise of the denominators (our R matrix has only the noise of numerators), but would hopefully make the model converge as expected.