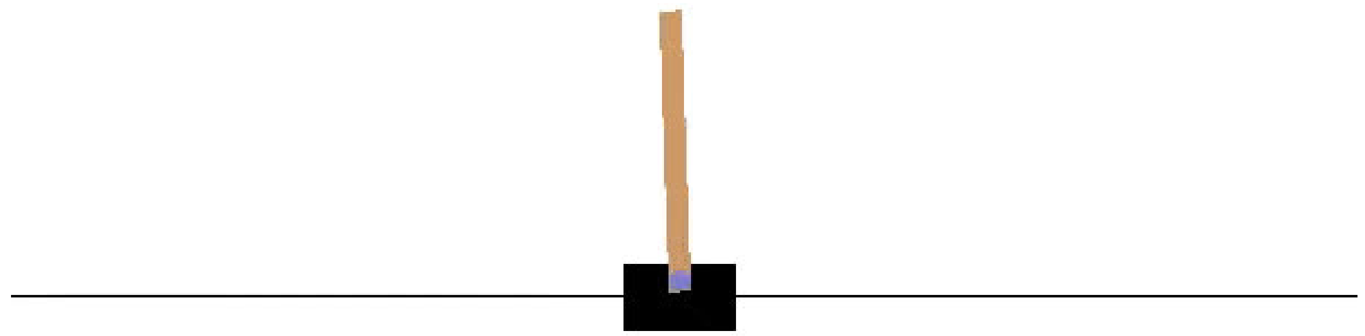
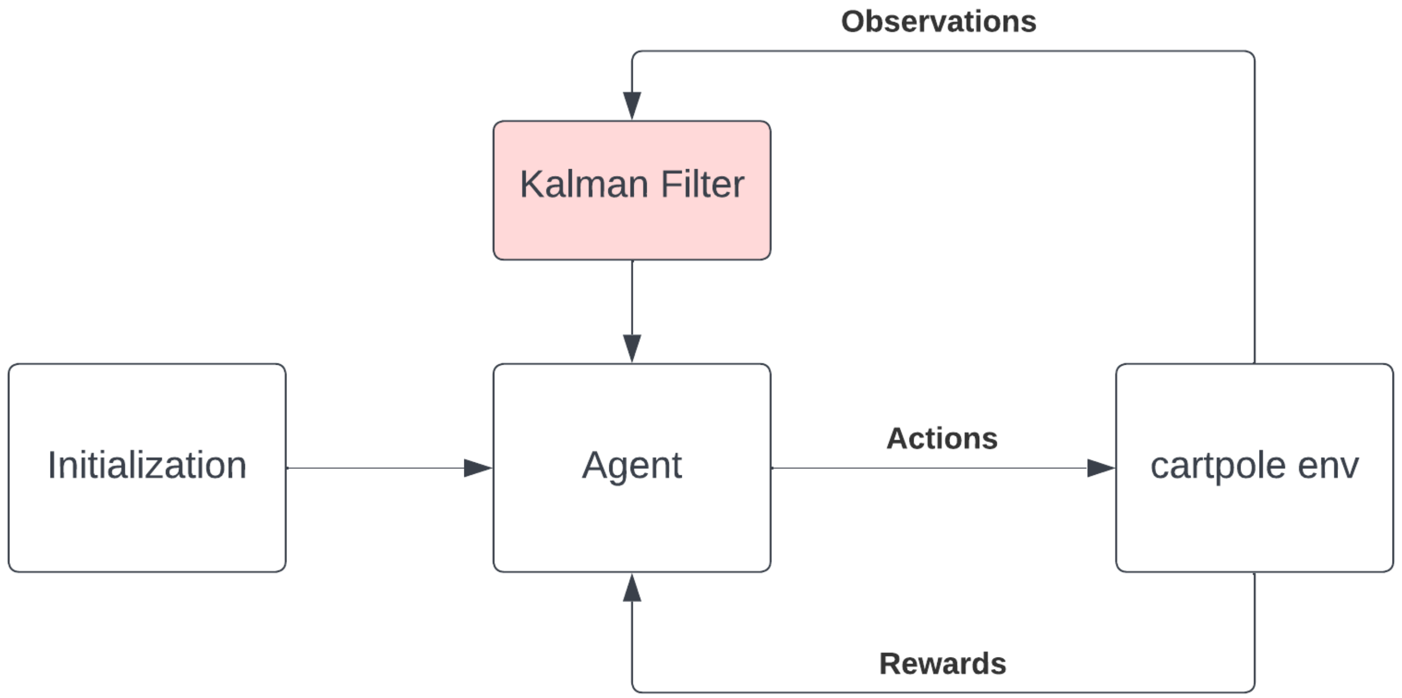
Applying Kalman Filter to Cart and Pole Control Problem



OpenAI Gym's Classic Control Simulation of Cart-pole



Filtering noise in measurements from cartpole env

Kalman Filter Derivation

Mean-squared-error Filter

Process: $x_{k+1} = \Phi x_k + w_k$

Measurements: $z_k = H x_k + v_k$

x_k	State vector at time *k*
Φ	State transition matrix
z_k	Measurement vector at time *k*
H	Factor linking *x* and *z*
v_k, w_k	Process noise

Here, covariances of process noises are assumed to be constant. Represented as below

$$Q = E\left[w_k w_k^T\right]$$

$$R = E\left[v_k v_k^T\right]$$

https://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman%20filter.pdf

Error Covariance Matrix (at time *k*) P_k can be represented as:

$$P_k = E\left[e_k e_k^T\right] = E\left[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T\right]$$

State Update Equation:

$$\hat{x}_k = \hat{x}'_k + K_k(z_k - H\hat{x}'_k)$$

$$\hat{x}_k = \hat{x}'_k + K_k(Hx_k + v_k - H\hat{x}'_k)$$

$$P_k = E\left[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T\right]$$

$$\begin{aligned} P_k = E\ [&[(I - K_kH)(x_k - \hat{x}'_k) - K_kv_k] \\ &[(I - K_kH)(x_k - \hat{x}'_k) - K_kv_k]^T\Big] \end{aligned}$$

$$P_k = (I - K_kH)E\left[(x_k - \hat{x}'_k)(x_k - \hat{x}'_k)^T\right](I - K_kH) + K_kE\left[v_kv_k^T\right]K_k^T$$

$$P_k = (I - K_kH)P'_k(I - K_kH)^T + K_kRK_k^T$$

$\hat{x}'_k$	Known Estimate at time *k*
$\hat{x}_k$	Predicted Estimate at time *k*
K_k	Kalman Gain
$z_k - H\hat{x}'_k$	Measurement Residual
P'_k	Prior Estimate of P_k

https://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman%20filter.pdf

Covariance Matrix P_{kk} :

$$P_{kk} = \begin{bmatrix} E\left[e_{k-1}e_{k-1}^T\right] & E\left[e_k e_{k-1}^T\right] & E\left[e_{k+1}e_{k-1}^T\right] \\ E\left[e_{k-1}e_k^T\right] & E\left[e_k e_k^T\right] & E\left[e_{k+1}e_k^T\right] \\ E\left[e_{k-1}e_{k+1}^T\right] & E\left[e_k e_{k+1}^T\right] & E\left[e_{k+1}e_{k+1}^T\right] \end{bmatrix}$$

Differentiating $T\left[P_k\right]$ w.r.t K_k to find minimum:

$$P_k = P_k' - K_k H P_k' - P_k' H^T K_k^T + K_k \left(H P_k' H^T + R\right) K_k^T$$

$$T\left[P_k\right] = T\left[P_k'\right] - 2T\left[K_k H P_k'\right] + T\left[K_k \left(H P_k' H^T + R\right) K_k^T\right]$$

$$\frac{dT\left[P_k\right]}{dK_k} = -2(HP_k')^T + 2K_k\left(HP_k'H^T + R\right)$$

$$\left(HP_k'\right)^T = K_k\left(HP_k'H^T + R\right)$$

Kalman Gain Equation:

$$K_k = P_k' H^T \left(H P_k' H^T + R\right)^{-1}$$

https://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman%20filter.pdf

Measurement Prediction Covariance of *measurement residual* ($z_k - H\hat{x}'_k$):

$$S_k = HP'_kH^T + R$$

$$\begin{aligned} P_k &= P'_k - K_kHP'_k - P'_kH^TK_k^T + K_k\left(HP'_kH^T + R\right)K_k^T \\ &= P'_k - P'_kH^T\left(HP'_kH^T + R\right)^{-1}HP'_k \\ &= P'_k - K_kHP'_k \\ &= (I - K_kH)\,P'_k \end{aligned}$$

$$P_k = (I - K_kH)\,P'_k \qquad \hat{x}_k = \hat{x}'_k + K_k\left(z_k - H\hat{x}'_k\right) \qquad K_k = P'_kH^T\left(HP'_kH^T + R\right)^{-1}$$

Updating Covariance | Updating Estimate | Kalman Gain

Using the above three equations, we can project $\hat{x}_k$ to *k + 1*

$$\hat{x}'_{k+1} = \Phi\hat{x}_k$$

$$\begin{aligned} e'_{k+1} &= x_{k+1} - \hat{x}'_{k+1} \\ &= \left(\Phi x_k + w_k\right) - \Phi\hat{x}_k \\ &= \Phi e_k + w_k \end{aligned}$$

https://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman%20filter.pdf

$$P_k \;=\; E\left[e_k e_k^T\right] \;=\; E\left[\left(x_k \;-\; \hat{x}_k\right)\left(x_k \;-\; \hat{x}_k\right)^T\right]$$

Extending error covariance matrix to time *k+1*

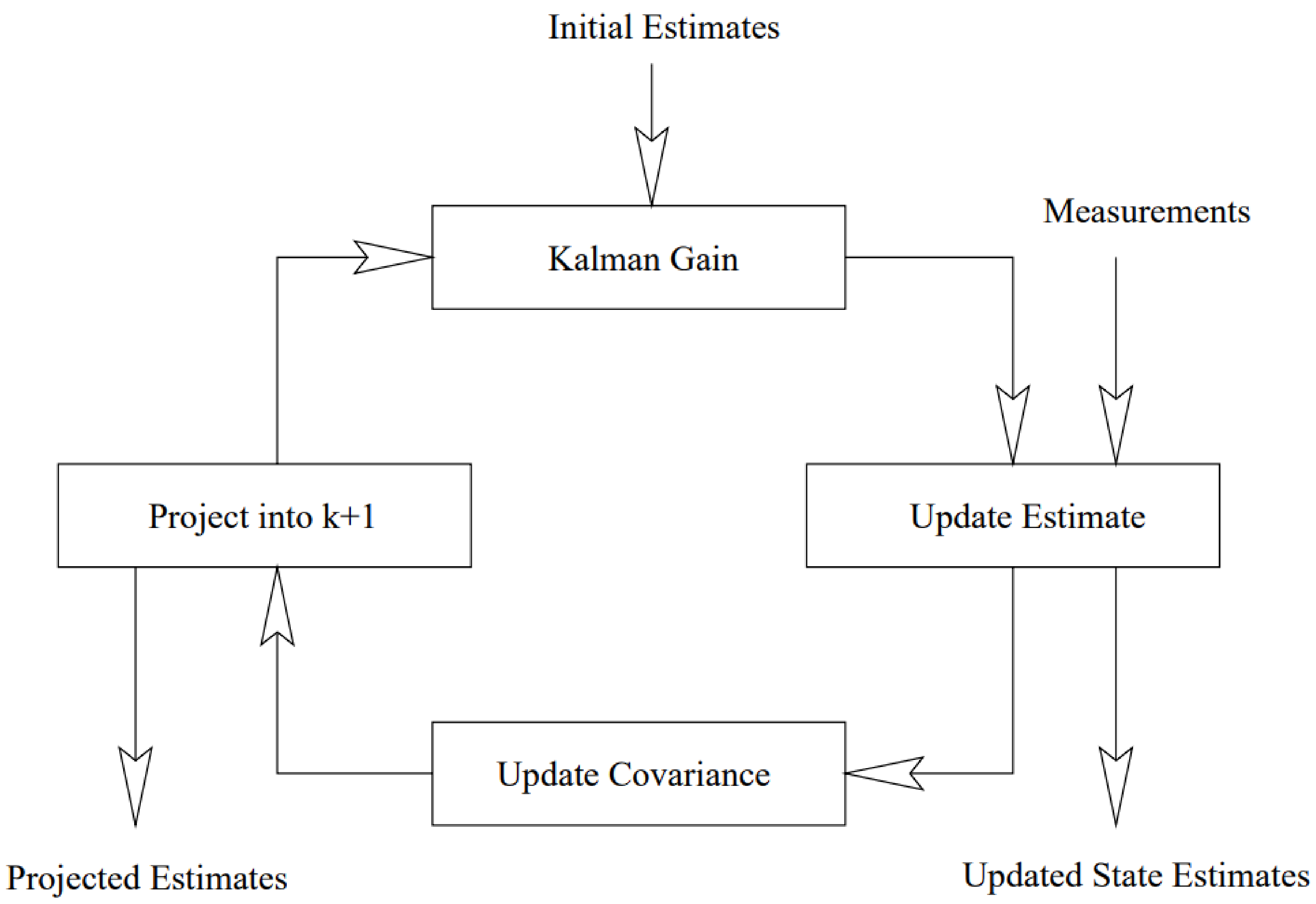
$$P'_{k+1} \;=\; E\left[e'_{k+1} e^{T\prime}_{k+1}\right] \;=\; E\left[\left(\Phi e_k \;+\; w_k\right)\left(\Phi e_k \;+\; w_k\right)^T\right]$$

Projection into time *k+1*

$$\begin{aligned} P'_{k+1} &= E\left[e'_{k+1} e^{T\prime}_{k+1}\right] \\ P'_{k+1} &= E\left[\Phi e_k \left(\Phi e_k\right)^T\right] \;+\; E\left[w_k w_k^T\right] \\ P'_{k+1} &= \Phi P_k \Phi^T \;+\; Q \end{aligned}$$

$\square$

https://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman%20filter.pdf



Description	Equation
Kalman Gain	$K_k = P_k' H^T \left(H P_k' H^T + R \right)^{-1}$
Update Estimate	$\hat{x}_k = \hat{x}_k' + K_k \left(z_k - H \hat{x}_k' \right)$
Update Covariance	$P_k = \left(I - K_k H \right) P_k'$
Project into $k+1$	$\hat{x}_{k+1}' = \Phi \hat{x}_k$ $P_{k+1} = \Phi P_k \Phi^T + Q$

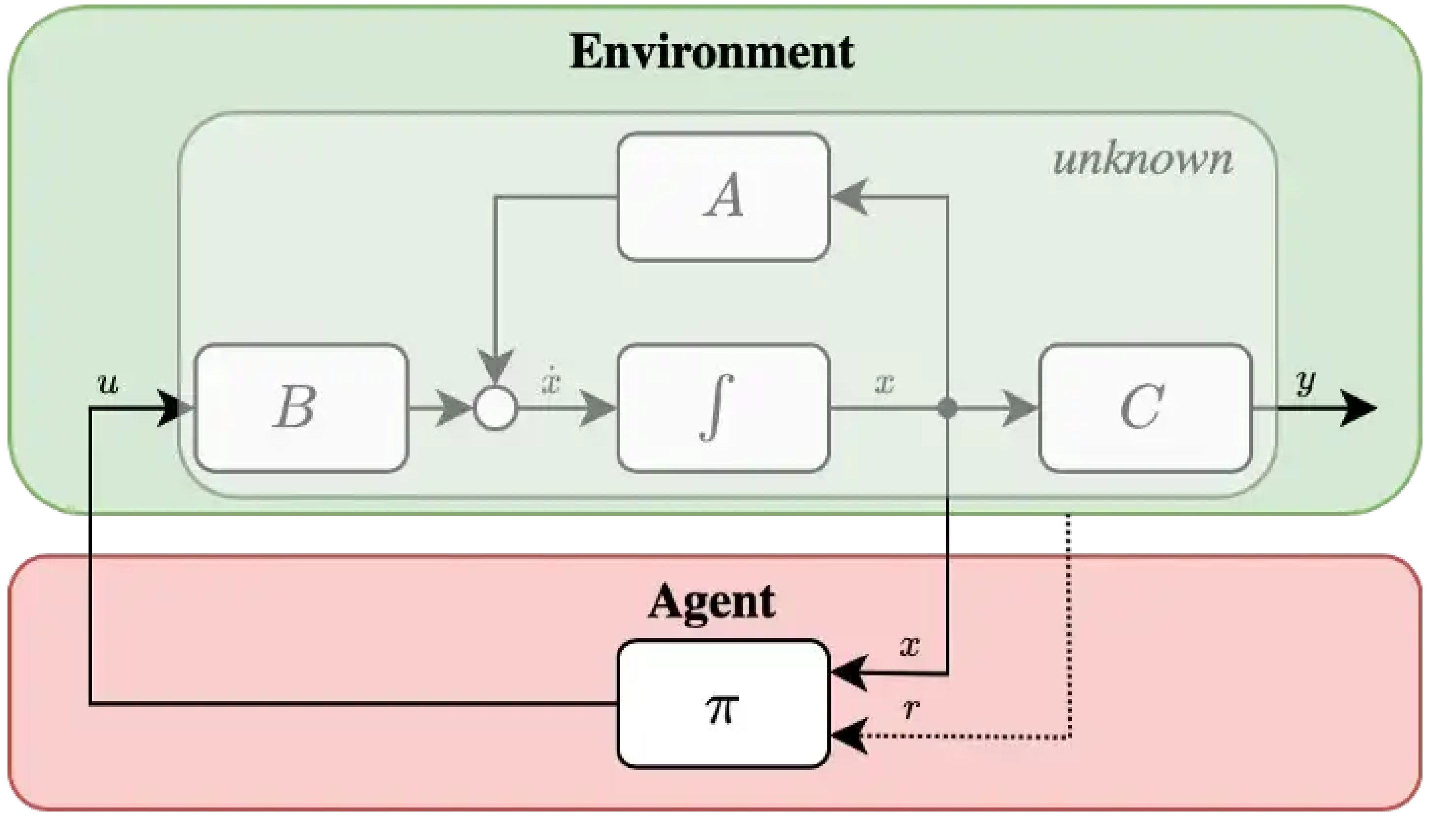
Kalman Filter Recursive Algorithm

https://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman%20filter.pdf

Cart-pole solution

Modifying OpenAI's cart-pole environment by introducing sensor inaccuracies and added randomness in measurements.

Existing Deep Q-Learning based solution would be used to compare performance in modified environment with Kalman Filter and in a noise-free environment.



https://towardsdatascience.com/comparing-optimal-control-and-reinforcement-learning-using-the-cart-pole-swing-up-openai-gym-772636bc48f4