Harvard Applied Mathematics 205

The Kalman Filter

Presented By: Michael S. Emanuel

23-Sep-2021

Outline

- Introduction and Motivation
- Kalman Filter in One Dimension
- Kalman Filter in $\mathbb{R}^n$
- Group Activity: Kalman Filter Exercise

Introduction and Motivation

Controlling a Spacecraft

How do you control a spacecraft?

- You receive a stream of noisy sensor readings.
- You also know the equations of motion and thrust.
- Each approach would give you a different estimate.
- What is the best way to combine them?

GPS vs. "Integrated Odometer"

Have you ever been in a car with a GPS system?

What happens when the signal is lost, e.g. in a tunnel?

If it's a self driving car, the car can sense its speed and direction.

How should we estimate the car's position?

Drone Navigation

Have you ever flown a drone or seen a friend do it?[1]

Recent models include stability control and automatic landing.

The GPS signals have an error tolerance on the order of 5 meters.

How do they do it?

[1] Danyun is a really good drone pilot.

Robot Control System

Suppose you work at Boston Dynamics on the robot dog spot.

You have a detailed physics model of how spot moves.

You also have sporadic and noisy sensor readings.

How do you plan the robot's motion?

The Kalman Filter for Navigation and Control

The **Kalman Filter** provides an efficient procedure for combining noisy signals in a system with well understood dynamics.

- Historically used by NASA in the US space program
- State estimation and control in many vehicles and robots
- Rigorous probabilistic model can derive equations
- Ostensibly a linear model, but many control problems can be effectively linearized over the relevant time scale

Learning Goals

- Understand the theoretical underpinning of the Kalman Filter
- Learn the equations to update the estimated state $\hat{\mathbf{x}}$ and variance $\hat{P}$ after a sensor reading $\mathbf{z}$
- Be positioned to use the Kalman Filter intelligently in applications
- Give those new to control theory a useful introduction

Kalman Filter in One Dimension

Problem Specification

Setup: 1D dynamical system in discrete time.

$$x_{k+1} = Ax_k + w$$
$$z_{k+1} = x_k + v$$

x is the state variable (e.g. position).
z is a noisy measurement from a sensor.
A is a scalar controlling the dynamics.
w and v are noise on the input and sensor readings with distributions $w \sim \mathcal{N}(0, \tau^2)$ and $v \sim \mathcal{N}(0, \sigma^2)$.

Random Variables and Scalar Parameters

At the risk of being pedantic, let's carefully separate categories of random variables from scalar parameters.

- x and z are random variables (state and sensor readings)
- w and v are random variables (noise on x and z)
- A is a known scalar parameter (dynamics of x)
- τ^2 and σ^2 are scalar variances that we assume

Realizations of Random Variables and Parameter Estimates

- z_k is one realization of z at step k; it is observed
- x_k is one realization of x at step k, it is hidden
- $\hat{x}_k$ is our estimate of the mean of x at the start of step k
- $\hat{P}_k$ is our estimate of the variance of x at the start of step k
- Our belief starting step k is $x_k \sim \mathcal{N}(\hat{x}_k, \hat{P}_k)$

Prediction of Position x_1: Setup

Initial state: position x is normal with mean μ_0 and variance P_0:

$$x_0 \sim \mathcal{N}(\mu_0, P_0)$$

Calculate probability distribution of x_1 using L.O.T.P.:

$$p(x_1) = \int p(x_1|x_0)p(x_0)dx_0$$

The variable $x_1|x_0$ is distributed $\sim \mathcal{N}(Ax_0, \tau^2)$, so $p(x_1|x_0)$ is the normal PDF of this distribution, namely
$p(x_1|x_0) = (2\pi)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x_1 - Ax_0)^2/\tau^2\right\}$.
The variable x_0 is distributed $\sim \mathcal{N}(\mu_0, P)$, so $p(x_0)$ is the PDF
$p(x_0) = (2\pi)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}x_0^2/P_0^2\right\}$.

Prediction of Position x_1: Calculation

While it's possible to do a messy integral in Mathematica...
Here is the clean Stat 110 way to calculate the distribution of x_1:

- $x_1 = (Ax_0) + w$
- (Ax_0) and w are both normally distributed random variables
- w is just random noise, so it's independent of x_0
- Theorem (Stat 110): The sum of two independent normal random variables is also a normal random variable...
- and the means and variance just add up
- $\Rightarrow x_1$ is normal with mean $A\mu_0 + 0$ and variance $A^2 P_0 + \tau^2$.

Prediction of Position x_1: Result

It's customary to drop the notation μ_0 and just call the expected initial position x_0. Then

$$\boxed{x_1 \sim \mathcal{N}(Ax_0, A^2 P_0 + \tau^2)} \tag{1}$$

We write the predicted position $\hat{x}_1$ and updated variance as P_1:

$$\begin{aligned}
\hat{x}_1 &= Ax_0 \\
\hat{P}_1 &= A^2 P_0 + \tau^2 \\
x_1 &\sim \mathcal{N}(\hat{x}_1, \hat{P}_1)
\end{aligned}$$

Correction of Position x_1: Setup

After we see the sensor reading z_1, what is the updated probability distribution of x_1? Use Bayes' Rule!

$$p(x_1|z_1) \propto p(z_1|x_1)p(x_1)$$

We know the prior x_1 from Eq. 1. And the conditional distribution of z_1 given x_1 is a normal that just adds noise of variance σ^2,

$$z_1|x_1 \sim \mathcal{N}(x_1, \sigma^2)$$

Multiplying the two terms:

$$p(x_1|z_1) \propto \exp\left(-\frac{1}{2}\frac{(z_1 - x_1)^2}{\sigma^2}\right)\exp\left(-\frac{1}{2}\frac{(x_1 - \hat{x}_1)^2}{P_1}\right)$$

Correction of Position x_1: Calculation

Now choose $\hat{x}_1$ to maximize log of posterior $p(x_1|z_1)$.

$$\begin{aligned}
\frac{\partial \log p(x_1|z_1)}{\partial x_1} &= -\frac{(x_1 - z_1)}{\sigma^2} - \frac{(x_1 - \hat{x}_1)}{\hat{P}_1} = 0 \\
\Rightarrow x_1 &= \left(\frac{z_1}{\sigma^2} + \frac{\hat{x}_1}{\hat{P}_1}\right) \Big/ \left(\frac{1}{\sigma^2} + \frac{1}{\hat{P}_1}\right) \qquad (2) \\
&= \frac{\hat{P}_1 z_1 + \sigma^2 \hat{x}_1}{\hat{P}_1 + \sigma^2}
\end{aligned}$$

Kalman Gain Definition

There is a special way to write Eq. 2. Label the previous estimate of $\hat{x}_1^p$ (for the predictor step) to disambiguate it from this revised estimate, $\hat{x}_1$. Similarly, label the previous variance estimate $\hat{P}_1^p$. Define the **Kalman gain**, K_1 by

$$\boxed{K_1 \equiv \frac{\hat{P}_1^p}{\hat{P}_1^p + \sigma^2}} \tag{3}$$

Then the updated position mean $\hat{x}_1$ and variance $\hat{P}_1$ are

$$\boxed{\begin{aligned} \hat{x}_1 &= \hat{x}_1^p + K_1(z_1 - \hat{x}_1^p) \\ \hat{P}_1 &= (1 - K_1)\hat{P}_1^p \end{aligned}} \tag{4}$$

Kalman Filter 1D Summary

Here is one full update cycle from $(\hat{x}_{k-1}, \hat{P}_{k-1})$ to $(\hat{x}_k, \hat{P}_k)$:

- $\hat{x}_k^p = A\hat{x}_{k-1}$ (predictor step - position)
- $\hat{P}_k^p = A^2\hat{P}_{k-1} + \tau^2$ (predictor step - variance)
- $K_k = \frac{\hat{P}_k^p}{\hat{P}_k^p + \sigma^2}$ (Kalman gain)
- $\hat{x}_k = \hat{x}_k^p + K_k(z_k - \hat{x}_k^p)$ (corrector step - position)
- $\hat{x}_k = (1 - K_k)\hat{P}_k^p$ (corrector step - variance)

Key insight: the model always updates the probability distribution of x_k and P_k to be normal!

The above recipe was derived to calculate $\hat{x}_1$ and $\hat{P}_1$ from $\hat{x}_0$, $\hat{P}_0$ and z_1 but it works for any other k equally well.

Two Extreme Cases: $\sigma = 0$ or $\sigma = \infty$

When $\sigma^2 = 0$, our sensors have no noise.

- The Kalman gain K_1 goes to one
- The corrector step simplifies to $\hat{x}_1 = z_1$.
- Intuition: when the sensor is **perfect**, our estimate is to parrot back the sensor reading.

When $\sigma^2 = \infty$, our sensors are random number generators.

- The Kalman gain K_1 goes to zero
- The corrector step simplfies to $\hat{x}_1 = \hat{x}_1^p$.
- Intuition: when the sensor is **garbage**, ignore it and keep the prior.

Kalman Filter in $\mathbb{R}^n$

Dynamical System Specification

We model a linear dynamical system with update rule

$$\boxed{\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k} \tag{5}$$

- Vector $\mathbf{x} \in \mathbb{R}^n$ - the **state** of the system
- Matrix $\mathbf{A} \in \mathbb{R}^{nxn}$ - the **transition matrix**
- Vector $\mathbf{u} \in \mathbb{R}^r$ - the **control inputs** vector
- Matrix $\mathbf{B} \in \mathbb{R}^{nxr}$ - the **control output** matrix
- Vector $\mathbf{w} \in \mathbb{R}^n$ - Gaussian **input noise**, $\mathbf{w} \sim \mathcal{N}(0, Q)$

The terms $\mathbf{B}$ and $\mathbf{u}$ are optional for problems that have control inputs. They can also be abused to shoehorn locally linear problems into this framework.

Measurement Process

Measurements are linear in the inputs, with noise added

$$\boxed{\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k} \tag{6}$$

- Vector $\mathbf{z} \in \mathbb{R}^m$ - the **measurement outputs**
- Matrix $\mathbf{H} \in \mathbb{R}^{mxn}$ - the **connection matrix** ($\mathbf{x}$ to $\mathbf{z}$)
- Vector $\mathbf{v} \in \mathbb{R}^m$ - the **sensor noise** $\sim \mathcal{N}(0, \mathbf{R})$

Estimation Error and Noise Covariance

Define $\hat{\mathbf{x}}_k$ as the estimate of current state at step k.
The **estimation error** $\mathbf{e}_k$ is

$$\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k \tag{7}$$

Define the covariance matrix $\mathbf{P}_k$ of the estimation errors by

$$\mathbf{P}_k = \mathrm{E}[\mathbf{e}_k \mathbf{e}_k^T] = \mathrm{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] \tag{8}$$

Define matrices $\mathbf{Q}$ and $\mathbf{R}$ for the covariances of $\mathbf{w}$ and $\mathbf{v}$:

$$\mathbf{Q} = \mathrm{E}[\mathbf{w}\mathbf{w}^T]$$
$$\mathbf{R} = \mathrm{E}[\mathbf{v}\mathbf{v}^T]$$

These are assumed to be positive semi-definite.

Predictor Setup

In the predictor step we calculate an *a priori* estimate

$$\hat{\mathbf{x}}_k^p = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} \tag{9}$$

Calculate the covariance $\mathbf{P}_k^p$ of the measurement error $\mathbf{e}_k$

$$\mathbf{P}_k^p = \mathrm{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_k^p)(\mathbf{x}_k - \hat{\mathbf{x}}_k^p)^T] \tag{10}$$

Simplify the difference term:

$$\begin{aligned} \mathbf{x}_k - \hat{\mathbf{x}}_k^p &= \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} - \mathbf{A}\hat{\mathbf{x}}_{k-1} \\ &= \mathbf{A}\mathbf{e}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \end{aligned}$$

Predictor Covariance Calculation

Recall that constants don't affect covariance, i.e.
$\mathrm{Cov}[X + c, Y + c] = \mathrm{Cov}[X, Y]$.
$\mathbf{Bu}_{k-1}$ is assumed known so it's like a constant and

$$\mathbf{P}_k^p = \mathrm{Var}[\mathbf{Ae}_{k-1} + \mathbf{w}_{k-1}]$$

The error $\mathbf{e}_{k-1}$ accumulated prior to step $k - 1$, so it is independent of the signal noise $\mathbf{w}_{k-1}$, which occurs between steps $k - 1$ and k. Therefore the two terms are independent and the variances add.

Predictor Covariance Calculation

Completing the calculation of the covariance from last page:

$$\begin{aligned}
\mathrm{Var}[\mathbf{A}\mathbf{e}_{k-1}] &= \mathrm{E}[(\mathbf{A}\mathbf{e}_{k-1})(\mathbf{A}\mathbf{e}_{k-1})^T] \\
&= \mathrm{E}[\mathbf{A}(\mathbf{e}_{k-1}\mathbf{e}_{k-1}^T)\mathbf{A}^T] = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T \\
\mathrm{Var}[\mathbf{w}_{k-1}] &= \mathrm{E}[\mathbf{w}\mathbf{w}^T] = \mathbf{Q}
\end{aligned}$$

Combining the two terms we find

$$\boxed{\mathbf{P}_k^p = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}} \tag{11}$$

Predictor Covariance: Comparison to Scalar Case

Compare the matrix / vector covariance of the predictor in Eq. 11 with the scalar result.

- The term $A^2 P_{k-1}$ has been replaced by $A P_{k-1} A^T$.
- If we view the scalar A as a 1×1 matrix, we can see these are in fact consistent.
- The noise variance σ^2 has been replaced by the noise covariance matrix Q
- This is also consistent since the 1×1 "covariance matrix" of a scalar is just its variance.

A recurring theme in numerical linear algebra is that a matrix times its transpose is often analogous to a squared scalar number in a 1D problem.

Corrector Step: Setup

Now suppose a sensor measurement $\mathbf{z}_k$ becomes available.
We will update $\hat{\mathbf{x}}_k$ to $\mathbf{x}_k$ via the equation

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^p + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\mathbf{x}_k^p) \tag{12}$$

The term $\mathbf{z}_k - \mathbf{H}\mathbf{x}_k^p$ is called the **measurement residual**.

Why is that? If our prediction $\mathbf{x}_k^p$ had been correct, the measurement would have been $\mathbf{H}\mathbf{x}_k^p$.

The actual result was $\mathbf{z}_k$, so the measurement residual is the "surprise" (new information gleaned).

The 1D measurement residual was just $z_k - \hat{x}_k$ since we had no $\mathbf{H}$ matrix in that case.

Corrector Step: Measurement Error

Substitute using $\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}$ in Eq. 12 and

$$\begin{aligned}\hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^p + \mathbf{K}_k(\mathbf{H}\mathbf{x}_k + \mathbf{v} - \mathbf{H}\mathbf{x}_k^p) \\ &= (\mathbf{I}_n - \mathbf{K}_k\mathbf{H})\hat{\mathbf{x}}_k^p + \mathbf{K}_k\mathbf{H}\mathbf{x}_k + \mathbf{K}_k\mathbf{v}_k\end{aligned} \tag{13}$$

Now substitute (13) for $\hat{\mathbf{x}}_k$ in the error covariance in Eq. 8

$$\mathbf{P}_k = \mathrm{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T]$$

First simplify the measurement error term:

$$\begin{aligned}\mathbf{x}_k - \hat{\mathbf{x}}_k &= \mathbf{x}_k - \left\{(\mathbf{I}_n - \mathbf{K}_k\mathbf{H})\hat{\mathbf{x}}_k^p + \mathbf{K}_k\mathbf{H}\mathbf{x}_k + \mathbf{K}_k\mathbf{v}_k\right\} \\ &= (\mathbf{I}_n - \mathbf{K}_k\mathbf{H})\mathbf{x}_k - (\mathbf{I}_n - \mathbf{K}_k\mathbf{H})\hat{\mathbf{x}}_k^p - \mathbf{K}_k\mathbf{v}_k \\ &= (\mathbf{I}_n - \mathbf{K}_k\mathbf{H})(\mathbf{x}_k - \hat{\mathbf{x}}_k^p) - \mathbf{K}_k\mathbf{v}_k\end{aligned} \tag{14}$$

Corrector Step: Variance

Now calculate the variance $\mathbf{P}_k$ using Eq. 14 (measurement error). Notice the term $(\mathbf{I}_n - \mathbf{K}_k\mathbf{H})(\mathbf{x}_k - \hat{\mathbf{x}}_k^p)$ is a random variable that is determined *before* the noise vector $\mathbf{v}_k$ is drawn from $\mathcal{N}(0, \mathbf{R})$. So the cross terms vanish and $\mathbf{P}_k$ is the sum of two variances.

$$
\begin{aligned}
\mathbf{P}_k &= \mathrm{E}[\{(\mathbf{I}_n - \mathbf{K}_k\mathbf{H})(\mathbf{x}_k - \hat{\mathbf{x}}_k^p)\}\{(\mathbf{I}_n - \mathbf{K}_k\mathbf{H})(\mathbf{x}_k - \hat{\mathbf{x}}_k^p)\}^T] \\
&\quad + \mathrm{E}[(\mathbf{K}_k\mathbf{v}_k)(\mathbf{K}_k\mathbf{v}_k)^T] \\
&= \mathrm{E}\left[(\mathbf{I}_n - \mathbf{K}_k\mathbf{H})\left\{(\mathbf{x}_k - \hat{\mathbf{x}}_k^p)(\mathbf{x}_k - \hat{\mathbf{x}}_k^p)^T\right\}(\mathbf{I}_n - \mathbf{K}_k\mathbf{H})^T\right] \\
&\quad + \mathrm{E}\left[\mathbf{K}_k\left\{\mathbf{v}_k\mathbf{v}_k^T\right\}\mathbf{K}_k\right]
\end{aligned}
\tag{15}
$$

Now, by definition, the first term in blue is just $\mathbf{P}_k^p$ (the variance before we did the correction). And the second term in blue is just the noise covariance $\mathbf{R}$.

Corrector Step: Optimization

Putting the pieces of this epic calculation together,

$$\boxed{\mathbf{P}_k = (\mathbf{I}_n - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^p(\mathbf{I}_n - \mathbf{K}_k\mathbf{H})^T + \mathbf{K}_k\mathbf{R}\mathbf{K}_k^T} \tag{16}$$

It remains to choose $\mathbf{K}_k$ to minimize a suitable error.
A natural choice is the total variance of the estimates,
$TV = \sum P_{kk}$. This is just the trace $\mathrm{tr}(\mathbf{P}_k)$.
Select $\mathbf{K}_k$ to minimize $\mathrm{tr}(\mathbf{P}_k)$. Use scalar-matrix differentiation techniques[2] and we obtain

$$\frac{\partial \mathrm{tr}(\mathbf{P}_k)}{\partial \mathbf{K}_k} = -2(\mathbf{H}\mathbf{P}_k^p)^T + (\mathbf{H}\mathbf{P}_k^p\mathbf{H}^T + \mathbf{R})^{-1} \tag{17}$$

[2] A good reference is *The Matrix Cookbook*

Corrector Step: Kalman Gain

Set the derivative in Eq. 17 to zero for the optimal Kalman gain

$$\boxed{\mathbf{K}_k = \mathbf{P}_k^p \mathbf{H}^T \left(\mathbf{H}\mathbf{P}_k^p \mathbf{H}^T + \mathbf{R} \right)^{-1}} \tag{18}$$

Use this matrix for $\mathbf{K}_k$ in Eq. 12 to update $\hat{\mathbf{x}}_k^p$ to $\hat{\mathbf{x}}_k$.
We can substitute Eq. 18 for $\mathbf{K}_k$ in Eq. 16. The result after a somewhat messy calculation is

$$\boxed{\mathbf{P}_k = (\mathbf{I}_n - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^p} \tag{19}$$

Kalman Filter: Summary

Here is one full update cycle from $(\mathbf{x}_{k-1}, \mathbf{P}_{k-1})$ to $(\mathbf{x}_k, \mathbf{P}_k)$

- $\hat{\mathbf{x}}_k^p = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_k$ (Eq. 9, Predictor)
- $\mathbf{P}_k^p = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}$ (Eq. 11, Predictor Variance)
- $\mathbf{K}_k = \mathbf{P}_k^p\mathbf{H}^T\left(\mathbf{H}\mathbf{P}_k^p\mathbf{H}^T + \mathbf{R}\right)^{-1}$ (Eq. 18, Kalman Gain)
- $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^p + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\mathbf{x}_k^p)$ (Eq. 12, Corrector)
- $\mathbf{P}_k = (\mathbf{I}_n - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^p$ (Eq. 19, Corrector Variance)
- Invariant: $\mathbf{x}_k \sim \mathcal{N}(\hat{\mathbf{x}}_k, \mathbf{P}_k)$ after measurement z_k

Comparing Vector to Scalar: Perfect Correspondence!

We can build intuition by comparing the vector formulas to the scalar formulas.
Assume here that $\mathbf{H} = \mathbf{I}$, i.e. the measurement $\mathbf{z}_k = \mathbf{x}_k + \mathbf{v}_k$

Var	Shape	Vector	Scalar
$\hat{\mathbf{x}}_k^p$	nxn	$\mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_k$	$A\hat{x}_{k-1} + Bu_k$
$\mathbf{P}_k^p$	nx1	$\mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}$	$A^2 P_{k-1} + Q$
$\mathbf{K}_k$	nxm	$\mathbf{P}_k^p \left(\mathbf{P}_k^p + \mathbf{R}\right)^{-1}$	$(\hat{P}_k^p)(\hat{P}_k^p + R)^{-1}$
$\hat{\mathbf{x}}_k$	nx1	$\hat{\mathbf{x}}_k^p + \mathbf{K}_k(\mathbf{z}_k - \mathbf{x}_k^p)$	$\hat{x}_k^p + K_k(z_k - \hat{x}_k^p)$
$\mathbf{P}_k$	nxn	$(\mathbf{I}_n - \mathbf{K}_k)\mathbf{P}_k^p$	$(1 - K_k)\hat{P}_k^p$

In making the comparison, I renamed τ^2 to Q and σ^2 to R.

Group Activity: Kalman Filter Exercise

Group Activity: Kalman Filter Simulation

Problem: A projectile is launched from the ground at position $(x, y) = (0, 0)$ with initial velocity $(u, v) = (50, 100)$.

The equations of motion are assumed to be

$$\begin{aligned} \dot{x} &= u \qquad & \dot{u} &= 0 \\ \dot{y} &= v \qquad & \dot{v} &= g \end{aligned} \tag{20}$$

where $g = 9.80 \text{m/s}^2$ is Earth's gravitational field.

Projectile Problem: Equations of Motion

Discretize time using a constant time step dt.

Assume the input noise w is in velocity units.

Assume the initial conditions are known exactly, i.e. $P_0 = 0\mathbf{I}_4$.
The equations of motion are

$$\begin{aligned}
x_{k+1} &= x_k + udt + wdt \\
y_{k+1} &= y_k + vdt + wdt \\
u_{k+1} &= u_k dt + wdt \\
v_{k+1} &= v_k dt - gdt + wdt
\end{aligned} \tag{21}$$

Baseline Simulation and Synthetic Data

Analyze this problem with a Kalman Filter, synthetically simulating your own data.

- Use the state vector $\mathbf{x} = [x, y, u, v]^T$ to formulate the dynamics in matrix form.
- Simulate the evolution of the projectile until T=25 sec or it hits the ground. Use $dt = 0.005$s and $\tau = 0.2\frac{\text{m}}{\text{s}}$.
  Consider this to be the "ground truth."
- Code a function to create synthetic data with noisy sensor measurements of the position.
  The sensor readout is (x, y) with $\sigma = 10$m. What is $\mathbf{H}$?

Kalman Filter and Simulated Runs

Now experiment with a Kalman Filter on the simulated data

- Code a Kalman filter to estimate the projectile's trajectory. Feed it input data every N_{freq} steps; N_{freq} is a parameter.
- Set $N_{\text{freq}} = 1$ and plot three series: the true trajectory, the measured trajectory, and the filtered trajectory.
- Repeat this previous step, this time using $N_{\text{freq}} = 500$.
- Experiment with changing σ and τ.