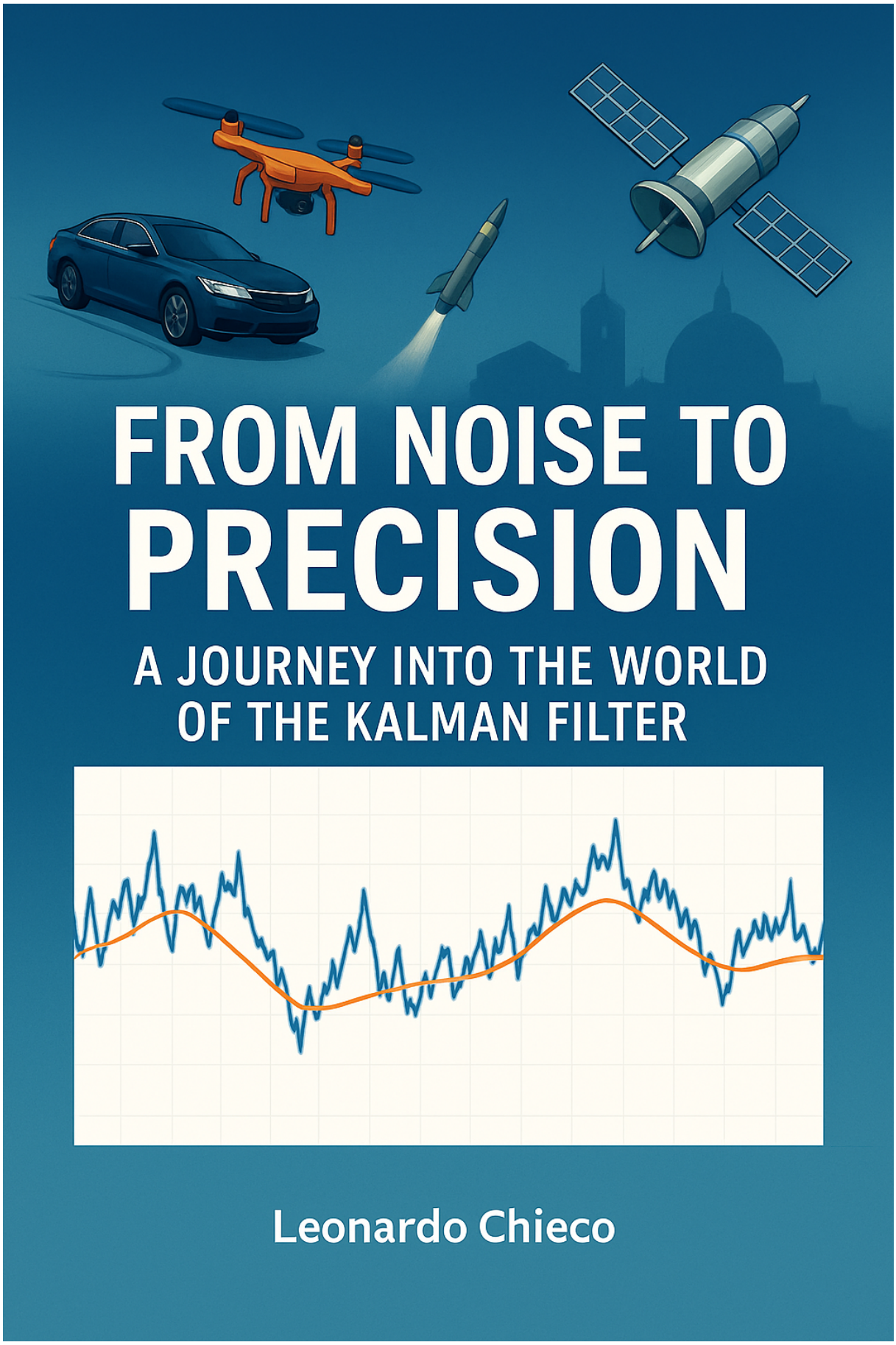
FROM NOISE TO
PRECISION
A JOURNEY INTO THE WORLD
OF THE KALMAN FILTER
Leonardo Chieco

From Noise to Precision

A Journey into the World of the Kalman Filter

Leonardo Chieco
leochieco72@gmail.com
linkedin.com/in/leonardo-chieco-53550b129

Abstract

In modern technology, the Kalman filter plays a central role in many applications, ranging from space engineering to robotics, finance, and artificial intelligence. Despite its wide use and importance, the Kalman filter is often seen as a difficult topic, reserved for experts. The goal of this tutorial is to provide an introductory and accessible overview, combining rigor with clarity. It is intended for readers who have a basic understanding of statistics and linear algebra. This article brings together notes collected from books and online resources, reviewed and organized to make the subject clearer and easier to understand.

The author

Leonardo Chieco is an electronic engineer with more than 20 years of experience in designing and developing software for control and automation systems (PC/PLC), electronic boards for industrial applications, firmware, robotics, and mechatronics.

Contact: leochieco72@gmail.com | LinkedIn Profile

Contents

1 Introduction

The Kalman filter is named after Rudolf Emil Kalman, a Hungarian-American mathematician and engineer (1930–2016), who introduced it in 1960 as a tool for solving the problem of state estimation in dynamic systems. His work, initially underestimated, later became crucial in NASA's space program. The filter was widely used during the development of the Apollo missions, contributing to the success of the Moon landing through its ability to provide precise estimates of trajectories and positions in real time, even in the presence of noise and measurement uncertainties.
Many real dynamic systems (moving cars, flying drones, market indices) are described by time-dependent quantities that cannot be observed with perfect precision. Sensors used to measure these quantities (GPS, radar, inertial sensors) provide noisy data,provide noisy data, that is, data affected by random errors. Moreover, the physics of the system itself, introduces additional uncertainty caused by factors that cannot be modeled perfectly, such as wind, friction, or electronic noise.
The Kalman filter addresses the problem of estimating, as accurately as possible, the true state of a dynamic system using partial and noisy measurements together with a model of the system. In other words, it combines the model and the measurements to infer what is really happening, since sensors provide only imperfect information.

The Kalman filter is at the core of a surprising number of technologies used every day or employed in advanced fields:

- **Robotics:** estimation of position and orientation of autonomous robots.
- **Autonomous vehicles:** sensor fusion of radar, LiDAR, cameras, and IMU data.
- **Finance:** prediction of market trends and volatility.
- **Flight control:** estimation of speed and position of airplanes and drones.
- **Biomedical engineering:** continuous monitoring of physiological signals like the heartbeat.

The Kalman filter is an algorithm that estimates the state of a linear dynamic system affected by noise. It optimally combines two sources of information:

- the model of the system
- the measurements, which are also noisy and linearly related to the system state

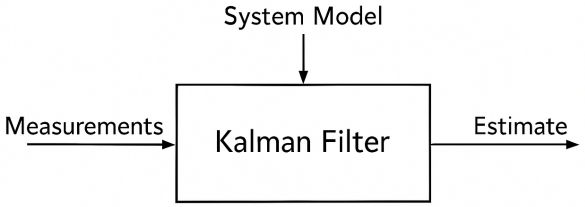

Figure 1: Operation of the Kalman filter (schematic view).

One refers to an optimal estimator when a mathematical method, given a set of observations, provides an estimate of a quantity by minimizing the error according to a well-defined criterion The Kalman filter belongs to this category, as it minimizes any quadratic error function of the estimate by using all available information. Starting from a model and a sequence of imperfect measurements, the filter produces predictions and successive corrections, moving progressively closer to the true value of the quantity of interest.
These predictions, initially coarse, are refined over time by the recursive prediction–correction algorithm, which gradually 'learns' the characteristics of the system.
Without exploring the mathematical details for now, it is important to understand that the Kalman filter operates in two distinct phases:

1. **Prediction:** a mathematical model of the system is used to estimate the system state at the next time step, based on the current information.

2. **Update (or correction):** when a new measurement becomes available from the sensor, the filter compares it with the prediction and corrects the estimate according to how much the measurement is trusted relative to the prediction itself.

This cycle repeats continuously, providing an updated estimate of the system state at each new time step.
In the context of the Kalman filter, the distinction between prediction and estimation has a precise technical meaning and is fundamental for understanding how the algorithm works.
The *prediction* is the first step of the recursive cycle. It is a calculation based solely on:

- the estimated state at the previous time step,

- the dynamic model of the system,

- and any control inputs (if present).

After the prediction, the filter receives a new observation (e.g., from a sensor) and performs the update or correction step. At this stage, the prediction is adjusted according to the error between the actual measurement and the expected one, yielding an optimal estimate of the state. The measurement is typically affected by noise, so the algorithm balances the information coming from the model (prediction) and from the sensor (measurement). This weighted combination is the so-called 'estimate', that is, a predicted state refined using the measured data.
In summary:

- **Prediction:** based only on the model, before the new measurement is taken into account.

- **Updated estimate:** obtained after incorporating the measurement, representing the corrected version of the prediction.

Although the goal is to keep a simple and accessible, some basic concepts are still necessary to understand the principles of the Kalman filter. Among these:

- **Basic statistics:** in particular, what mean, variance, and covariance are, and how to interpret them.

- **Concept of noise:** white noise, Gaussianity, and measurement errors.

- **Elementary linear algebra:** basic notions about matrices, matrix multiplication, and vectors.

Some statistical notions will be reviewed before being used, but a basic understanding is still required to follow the material comfortably. This tutorial deliberately avoids going into the formal derivation and the more complex mathematical proofs of the Kalman filter equations. Instead, it adopts a practical, application oriented approach that helps the reader intuitively understand how the filter works and why it is useful. The main goal is to give a conceptual and practical understanding: why the filter works, what it means to 'merge' measurements and models, and how this process relates to real-world situations. Throughout the tutorial, a practical example is used as a reference: a car moving in a straight line at constant speed. It is shown, step by step, how the Kalman filter can correct noisy position measurements and provide an increasingly accurate estimate not only of the position but also of the speed, even though the speed is not measured directly. This example helps to intuitively grasp the ideas of prediction, update, noise, and Kalman gain. In the following chapters, the operation of the filter will be explored by combining theory with practical examples and step-by-step explanations, and concluding with two simple Python programs that allow the reader to simulate and practice what has been learned.

2 Statistics review

This chapter introduces some basic statistical ideas that will be useful for better understanding certain aspects of the Kalman filter.

- A **random variable** is a function that assigns a numerical value to each possible outcome of a random experiment. For example, if a voltage is being measured, the random variable X can take any expected voltage value. If a die is being rolled, X takes the discrete values from 1 to 6.

- **Variance and standard deviation**

  The **variance** of a random variable X, denoted by σ_X^2 or $\text{Var}(X)$, is a measure of the dispersion of a data set relative to its mean. If the **mean** is denoted by μ_X, the variance is defined as follows:

  $$\sigma_X^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_X)^2$$

  As a unit of measurement, the variance has the square of the units of X. If X is a length measured in meters, the variance is expressed in m^2. The standard deviation, denoted by the Greek letter σ, is the square root of the variance.

- **Normal distribution**

  The normal (or Gaussian) distribution is one of the most fundamental concepts in statistics and probability. It is a mathematical function that describes how data are distributed around a mean value. The normal distribution has the characteristic shape of a symmetric bell: higher in the middle and gradually decreasing toward the sides. This means that values close to the mean are the most likely to occur, while values far from the mean are increasingly rare. For this reason, it is often referred to as the Gaussian bell curve.

  A normal distribution is defined by two parameters: mean (μ) and standard deviation (σ). For the sake of completeness, the equation of the normal distribution is shown below:

  $$f(x, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

  In the following image, an example of a gaussian curve is shown:

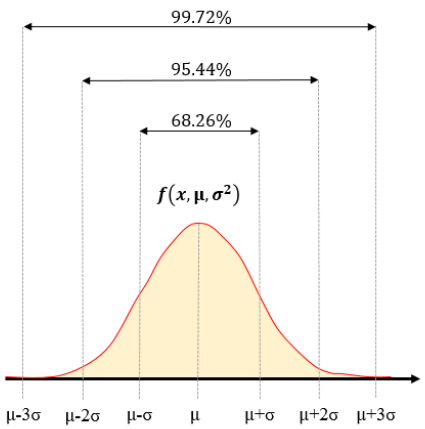

Figure 2: Bell-shaped gaussian curve.

This curve expresses an idea that fits many real processes: the highest probabilities of a phenomenon are concentrated around the mean. The probabilities decrease as one moves away from the mean value, either to the right or to the left. In particular, about 68% of the observations fall within the interval $[\mu-\sigma, \mu+\sigma]$. The higher the value of σ, the flatter the bell becomes along the horizontal axis. The lower the value of σ, the narrower and taller the bell becomes around the mean. It is important to note that the total area under the curve is always equal to 1, because it represents all the probabilities of the event.

It is worth noting that if a random variable X is normally distributed with variance σ^2, the random variable kX is also normally distributed and its variance is $k^2\sigma^2$.

- **Covariance**

  Covariance is a statistical measure that describes how two random variables move together. In other words, it indicates whether, and to what extent, two quantities tend to vary in a correlated way. Intuitively, it can be stated that:

  - If one variable tends to increase when the other increases, the covariance is positive.
  - If one variable tends to decrease when the other increases, the covariance is negative.
  - If there is no clear linear relationship between the two, the covariance is close to zero.

  In mathematical terms, for two random variables X and Y, the covariance is computed as follows:

$$\mathrm{Cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_X)(y_i - \mu_Y)$$

  where:

  - x_i and y_i are the observed values,
  - μ_X and μ_Y are the corresponding means of the random variables X and Y,
  - n is the number of observations.

- **Variance of sum of two random variables**

  The variance of the sum of two correlated variables, X and Y, is given by the following formula:

$$\mathrm{Var}(X+Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\ \mathrm{Cov}(X,Y)$$

  where $\mathrm{Cov}(X,Y)$ represents the covariance between X and Y.

- **Additive property of the covariance**

  The covariance of the sum of two variables with a third one is the sum of the individual covariances:

$$\mathrm{Cov}(X,\ Y+Z) = \mathrm{Cov}(X,Y) + \mathrm{Cov}(X,Z)$$

- **Accuracy and precision**

  Accuracy measures how close a measurement is to the true value. Precision measures how close repeated measurements are to each other, regardless of the true value.

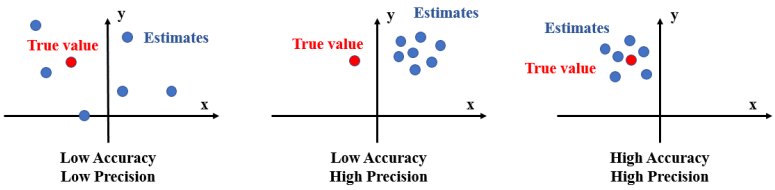

Figure 3: Differece between accuracy and precision.

A "good" measurement should be both precise and accurate.

Some concepts are now introduced to help better understand the mathematics of the Kalman filter.
The measurements obtained through sensors or instruments on a hypothetical fixed quantity z are modeled as random variables. Each measurement random variable Z_i is expressed as

$$Z_i = z + R_i,$$

where R_i denotes the noise random variable.
It is assumed that the noise has zero mean value and variance $\sigma^2_{R_i}$. Consequently, $Z_i = z + R_i$ has mean equal to z and variance equal to $\sigma^2_{R_i}$.
The probability density function of a measurement random variable describes how the measured value may vary due to noise. In the Gaussian case, this density has the familiar bell-shaped form and quantifies the uncertainty associated with the measurement process.
Consider now a single noisy measurement z_1 of an unknown fixed quantity z. The value z_1 is a realization of the random variable $Z_1 = z + R_1$, where R_1 denotes Gaussian noise with known variance. When the goal is to infer which values of z are most plausible, the same Gaussian expression can be interpreted differently: instead of being viewed as a density of Z_1, it is regarded as a function of the unknown z. This function expresses how compatible each candidate value of z is with the observed measurement and is referred to as the *likelihood* of z given the observation z_1. For a Gaussian model, the likelihood is maximized at the value of z that best explains the measurement, which corresponds to the peak of the curve.
If two independent measurements, z_1 and z_2, are available, each leads to a separate likelihood function, reflecting the compatibility of each possible value of z with the corresponding observation. Since the two measurements provide independent pieces of information, the overall likelihood is obtained by multiplying the two individual likelihoods. The resulting function is again Gaussian but narrower, indicating a reduced uncertainty about the possible values of z.
The figure below qualitatively illustrates how the two Gaussian likelihood functions, associated with the two measurements, combine to yield a sharper and more informative assessment of the unknown quantity z.

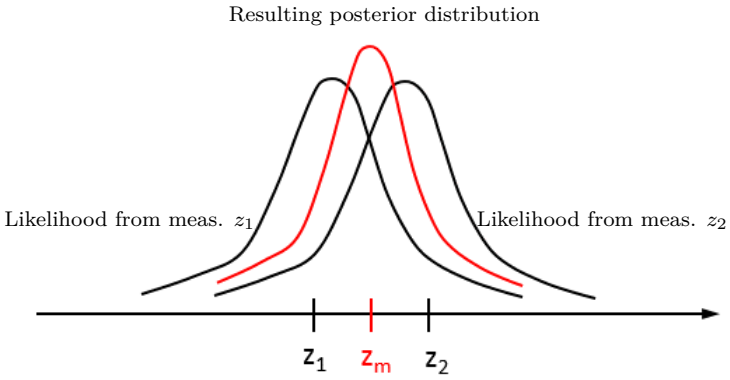

Figure 4: Qualitative illustration of the likelihood functions from two independent measurements and their resulting posterior distribution.

The resulting likelihood reaches its maximum at a point halfway between z_1 and z_2 (assuming equal variances). By definition, with a single measurement the estimation error depends on σ_R^2. When combining N independent measurements, each with the same variance σ_R^2, the variance of the resulting estimate is reduced to σ_R^2/N. Therefore, as the number of available measurements increases, the quality of the estimate improves.
Intuitively, when averaging several independent measurements (sample mean) on a true value z, the noise (i.e., the variance) becomes progressively less influential. Mathematically:

$$\mathrm{Var}(\hat{Z}) = \mathrm{Var}\left(\frac{1}{N}\sum_{i=1}^{N} Z_i\right) = \frac{1}{N^2}\sum_{i=1}^{N}\mathrm{Var}(Z_i) = \frac{1}{N^2}\,N\sigma_R^2 = \frac{\sigma_R^2}{N}.$$

What happens if the measurements have different accuracies? In this case, is it still convenient to take their average? Obviously not.
According to the maximum-likelihood criterion, the best estimate of the true value is given by

$$\hat{z} = \frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2}\,z_1 + \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}\,z_2.$$

In other words, more weight is assigned to the more accurate measurement, that is, the one with the smaller variance σ^2.

3 Introduction to the Kalman filter

Suppose one wants to track, at every instant, the motion of a vehicle moving at constant speed along a straight line.

Figure 5: Introductory example

Moreover, assume that the only available sensor provides measurements of the vehicle's current position. At time $t = 0$, the initial position is $x_0 = 0$, and the vehicle is observed to be moving at a constant speed of 60 km/h (this choice is made to simplify the calculations).
It is assumed that the position and velocity of the car are recorded at regular intervals Δt, for example $\Delta t = 1$ min. Since the system is observed at discrete sampling instants, the continuous-time notation $t = 0$ simply corresponds to the discrete index $i = 0$. Each iteration i therefore represents the system evaluated at time $t = i\,\Delta t$.
In a perfect world, the speed would remain constant at 60 km/h, and the position at a non-negative integer iteration step i would be described by the physical model:

$$\begin{cases} x_i = x_{i-1} + \Delta t \cdot v_{i-1} \\ v_i = v_{i-1} \end{cases}, \qquad i \geq 1$$

For convenience, these relations are often written in matrix form:

$$\begin{bmatrix} x_i \\ v_i \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{i-1} \\ v_{i-1} \end{bmatrix},$$

in a more compact form:

$$\mathbf{x}_i = A\,\mathbf{x}_{i-1}$$

The state vector is denoted in bold as $\mathbf{x}_i$. It contains all the information that characterizes the state of the system at each instant. In this case, it includes position and velocity.
However, the world is not perfect. At least two aspects can be identified that may complicate these assumptions:

i. The position measurements are noisy, and this noise may even vary with the velocity. In the general case, when both position and velocity are measured using two different sensors, the measurement noise is written as the vector

$$\boldsymbol{\eta} \sim \mathcal{N}(0, R),$$

where the notation "$\sim \mathcal{N}(\mu, \Sigma)$" means that the random variable follows a Gaussian distribution with mean μ and covariance Σ. In this general setting, the measurement noise covariance matrix is

$$R = \begin{bmatrix} r_{xx} & r_{xv} \\ r_{vx} & r_{vv} \end{bmatrix}.$$

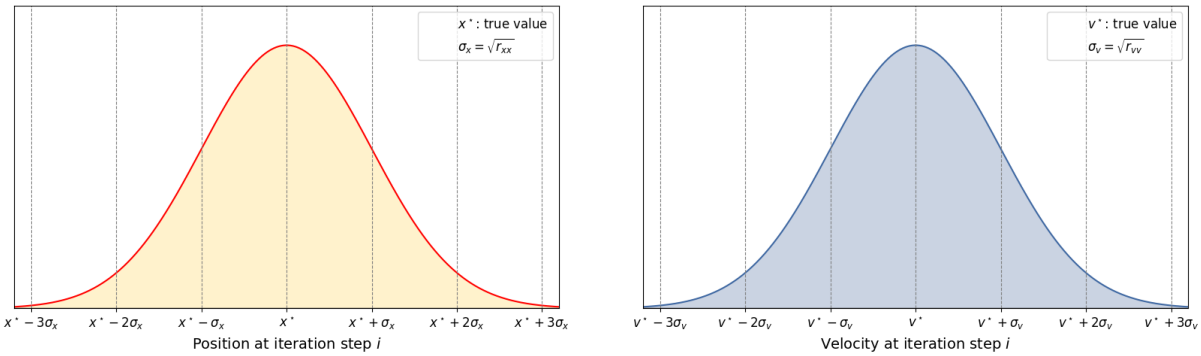

Figure 6: Gaussian noise on the measurements.

The variance term r_{xx} is related to the uncertainty in the position measurement, the variance term r_{vv} is related to the uncertainty of the velocity measurement, and the covariance terms ($r_{xv} = r_{vx}$) reflect the correlation between the position and velocity measurement errors.

However, since this tutorial considers only a position sensor, the measurement noise reduces to a scalar random variable

$$\eta \sim \mathcal{N}(0, \sigma_r^2),$$

meaning that the measurement noise is Gaussian with zero mean and variance σ_r^2 (the covariance matrix R collapses to the single variance $\sigma_r^2 = r_{xx}$).

ii. The process is not perfectly modeled. For example, real road conditions (such as potholes, traffic, wind, etc.) are not taken into account. This means that the "ideal" model of the vehicle's motion is subject to perturbations. These perturbations are described by the process noise vector

$$\mathbf{w} \sim \mathcal{N}(0, Q),$$

again meaning that $\mathbf{w}$ is a Gaussian random vector with zero mean and covariance matrix Q. The same considerations made for the general matrix R apply here as well.

In particular, q_{11} represents the variance of the noise acting on the position dynamics, and q_{22} represents the variance of the noise acting on the velocity dynamics, while the off diagonal terms describe their correlation.

Under the assumption of ignoring real-world imperfections (for now), it is possible to estimate where the car is located and what its velocity is after one minute. The velocity would still be 60 km/h, and the position would be 1 km from the starting point (60 km/h = 1 km/min). It is assumed, as mentioned above, that a sensor is available which provides the position of the car every minute (but not the velocity). Suppose that the sensor reports a distance of 0.9 km from the starting point. Thus, on one hand there is the noisy sensor measurement (0.9 km), and on the other hand the mathematical model (1.0 km). Which value should be trusted? There are two estimates, but it is not known which one is closer to the true state. Can the two estimates be combined to obtain a position that is more accurate than either of them individually?
This is where the Kalman filter comes into play.
Recall that the Kalman filter is an algorithm that combines past observations (even noisy ones) and current measurements (also noisy) to estimate the state of a system (for example, the position and velocity of the car).

It is now possible to combine the information collected so far. On one hand, there is the state equation of the system introduced earlier. On the other hand, the measurements y_i are linearly related to the state and are also affected by Gaussian noise with zero mean and variance σ_r^2. Therefore, the two equations are:

$$\mathbf{x}_i = A\,\mathbf{x}_{i-1} + \mathbf{w}_{i-1}, \qquad \mathbf{w}_{i-1} \sim \mathcal{N}(0, Q) \qquad i \geq 1 \tag{1}$$

$$y_i = H\,\mathbf{x}_i + \eta_i, \qquad H = \begin{bmatrix} 1 & 0 \end{bmatrix}, \ \eta_i \sim \mathcal{N}(0, \sigma_r^2) \qquad i \geq 1 \tag{2}$$

The equations can be written explicitly for the case under consideration:

$$\begin{bmatrix} x_i \\ v_i \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{i-1} \\ v_{i-1} \end{bmatrix} + \begin{bmatrix} w_{x,i-1} \\ w_{v,i-1} \end{bmatrix}, \qquad \begin{bmatrix} w_{x,i-1} \\ w_{v,i-1} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, Q \right),$$

$$Q = \begin{bmatrix} q_{xx} & q_{xv} \\ q_{vx} & q_{vv} \end{bmatrix}.$$

and

$$y_i = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{H} \begin{bmatrix} x_i \\ v_i \end{bmatrix} + \eta_i, \qquad \eta_i \sim \mathcal{N}\left(0, \sigma_r^2\right),$$

Note, again, that $H = [\,1\ 0\,]$ because only the position is measured, whereas no information is available about the velocity.
Recall that $\mathbf{w}$ (the process noise) represents the uncertainty or the unexpected variability in the real behaviour of the system. Even if the mathematical model describes the system as, for example, a uniform linear motion, in reality there may be perturbations such as wind, variable friction, or any other unmodeled external influence.
The noise η (the measurement noise) represents the error or inaccuracy of the measurement instruments. Each sensor (such as a GPS, an encoder, or a radar) has a certain uncertainty, due to technological, environmental, or electronic limitations.
As described earlier, the Kalman filter algorithm consists of two distinct steps:

- **Prediction**: in this phase, the previous estimates of the system state are passed through the idealized version of the dynamic system (the matrix A), producing the prediction for the state at the next iteration step. At this stage, no information from the measurement instruments has been incorporated yet.

- **Update**: Now the new information from the measurement instruments must be incorporated to update and improve the prediction. In this phase, the prediction of the system state and its covariance is combined with the new (noisy) measurements. The result will be the final state estimate.

  The first step, therefore, is to compute the residual or innovation term, that is, the difference between what was predicted and what was measured. The larger this term is, the less the prediction can be trusted.

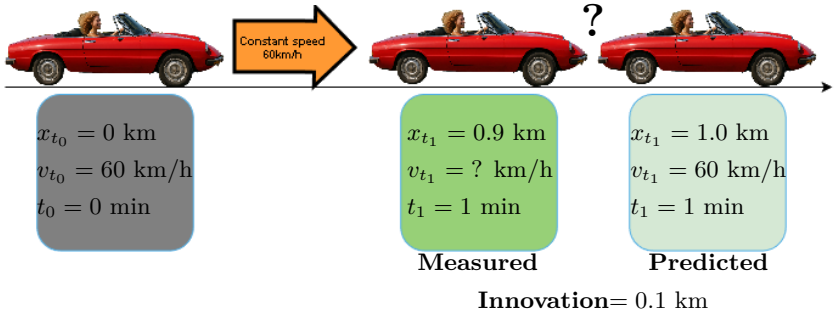

Figure 7: Concept of "innovation"

In this case, the residual is $1.0 - 0.9 = 0.1\,\text{km}$. It is necessary to decide how much to correct the prediction based on:

  - how much the prediction can be trusted,
  - how much the measurement instrument can be trusted.

An important aspect is now introduced: the covariance matrix of the state estimation error, denoted by P.
If the state has n components, P is an $n \times n$ matrix that contains:

- on the diagonal: the variances of each state error, that is, the measure of how uncertain each variable is.
- off the diagonal: the covariances between state errors, that is, the measure of how much the errors of two variables are correlated.

P indicates "how much the estimate can be trusted": if the variances are small, the estimate is reliable; if they are large, there is significant uncertainty. In other words, if P is large, it means that the estimation is highly uncertain, and therefore more weight will be given to the new measurements. If P is small, the estimate is considered reliable, and the filter trusts the predicted model more.
For the uniform linear motion with state $[x_i\ v_i]$, P is a 2×2 matrix:

$$P = \begin{bmatrix} \text{Var(Pos)} & \text{Cov(Pos, Vel)} \\ \text{Cov(Vel, Pos)} & \text{Var(Vel)} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$$

The top-left element, Var(Pos), indicates how certain the estimated position is. The bottom-right element, Var(Vel), indicates how certain the estimated velocity is. The off diagonal elements (covariances) show how the position and velocity errors are correlated. It is important not to confuse P with Q. The differences are shown in the following table.

	State Variance P	**Process Noise Q**
What it represents	The current uncertainty of the estimate.	The additional uncertainty at each step, caused by imperfections in the model.
Who computes it	The Kalman filter updates it at every step (prediction and correction).	It is chosen by the user, based on the physical knowledge of the system.
How it changes over time	It may increase or decrease (it grows during prediction and decreases with accurate measurements).	It is constant or defined a priori (a fixed parameter or a function of Δt).

Before proceeding, the following definitions are introduced:

Definitions:

$$x = \text{true position}, \qquad v = \text{true velocity}.$$

$$\hat{x}^-,\ \hat{v}^- = \text{predicted estimates (before using the measurement)}.$$

$$\hat{x}^+,\ \hat{v}^+ = \text{corrected estimates (after the measurement)}.$$

$$y = \text{position measurement: } y = x + \eta,$$

where η is the measurement noise, with zero mean and variance σ_r^2.

$$e_x^- = x - \hat{x}^-, \qquad e_v^- = v - \hat{v}^- \quad \text{(predicted errors)}.$$

$$\mathrm{Var}(e_x^-) = p_{11}^-, \qquad \mathrm{Var}(e_v^-) = p_{22}^-, \qquad \mathrm{Cov}(e_x^-, e_v^-) = p_{12}^-.$$

$$P^- = \begin{bmatrix} p_{11}^- & p_{12}^- \\ p_{21}^- & p_{22}^- \end{bmatrix}.$$

A detailed description of each symbol is provided in the following table.

Symbol	Name	When it is computed	Meaning
$\begin{bmatrix} \hat{x}^- \\ \hat{v}^- \end{bmatrix}$	Predicted state (a priori)	After the prediction step, before using the new measurement	Position and velocity predicted by the model, before the measurement
$\begin{bmatrix} \hat{x}^+ \\ \hat{v}^+ \end{bmatrix}$	Updated state (a posteriori)	After the update step (correction with the measurement)	Position and velocity corrected using the measurement
e_x^-	Predicted position error	After prediction	Difference between estimated and true position before correction
e_x^+	Updated position error	After update	Difference between estimated and true position after correction
e_v^-	Predicted velocity error	After prediction	Difference between estimated and true velocity before correction
e_v^+	Updated velocity error	After update	Difference between estimated and true velocity after correction
$P^- = \begin{bmatrix} p_{11}^- & p_{12}^- \\ p_{21}^- & p_{22}^- \end{bmatrix}$	Predicted covariance matrix	After prediction	Variances and covariances of the estimation errors before the measurement
$P^+ = \begin{bmatrix} p_{11}^+ & p_{12}^+ \\ p_{21}^+ & p_{22}^+ \end{bmatrix}$	Updated covariance matrix	After update	Variances and covariances of the estimation errors after the measurement

It has been shown that, in this example, there is a discrepancy between the measurement and the predicted position. A multiplicative factor k_x must be introduced to correct the prediction by "weighting" the detected error, that is, $(y - \hat{x}^-)$.

The state update, that is, the update of the position estimate, as shown at the beginning of the tutorial, can be written as follows:

$$\hat{x}^+ = \hat{x}^- + k_x\big(y - \hat{x}^-\big)$$

Note: the iteration index is omitted for readability. The analysis refers to the snapshot at the i-th iteration step.
The position error is:

$$e_x^+ = x - \hat{x}^+ = x - \hat{x}^- - k_x\big(y - \hat{x}^-\big) = e_x^- - k_x\big(x + \eta - \hat{x}^-\big).$$

$$e_x^+ = e_x^- - k_x\big(x + \eta - \hat{x}^-\big) = e_x^- - k_x\big(e_x^- + \eta\big) = (1 - k_x)\, e_x^- - k_x\, \eta.$$

The analysis now turns to the variances, assuming that the prediction noise and the measurement noise are independent:

$$\mathrm{Var}(e_x^+) = \mathrm{Var}\big((1 - k_x)e_x^- - k_x\eta\big) = (1 - k_x)^2\, \mathrm{Var}(e_x^-) + k_x^2\, \mathrm{Var}(\eta).$$

$$\mathrm{Var}(e_x^+) = (1 - k_x)^2\, p_{11}^- + k_x^2\, \sigma_r^2.$$

The derivative with respect to k_x is now computed and set equal to zero in order to determine the minimum point:

$$\frac{d\ \mathrm{Var}(e_x^+)}{dk_x} = 2(1 - k_x)p_{11}^- + 2k_x\sigma_r^2 = 0$$

$$(1 - k_x)p_{11}^- + k_x\sigma_r^2 = 0$$

$$k_x = \frac{p_{11}^-}{p_{11}^- + \sigma_r^2}.$$

The ratio obtained is called the Kalman gain.
If k_x equals 0, it means that $p_{11}^- = 0$, and therefore there is complete confidence in the prediction or in the ideal model ("the vehicle is certainly within the green rectangle in Figure 7"). If it equals 1, it means that $\sigma_r^2 = 0$, and therefore full confidence is placed in the measurement provided by the instrument ("the vehicle is certainly within the blue rectangle in Figure 7"). In practice, this value lies between 0 and 1.
The position correction is therefore obtained by taking the prediction and adding the term $k_x \times$ (surprise term). The larger the gain k_x is, the greater the correction will be. This difference is called the innovation or "surprise term".
The previous discussion applies to the position. What happens in the case of velocity? In fact, the velocity is not measured in any way (recall that the instrument provides only the position), yet the Kalman filter is still able to produce an estimate based on the available data. This is an important result.
Consider how many quantities cannot be measured directly in practice, while the Kalman filter is able to estimate them starting from other measurements that are easier and more economical to obtain.
For the velocity, the last relation derived for the position can be used, considering that the correction to the velocity estimate (which is assumed to be constant) can be performed starting from the measurable position variation Δx over the time interval, multiplied by a "weight", that is the Kalman gain associated with the velocity.
Defining k_v as the Kalman gain for the velocity, the filter update (velocity component) is:

$$\hat{v}^+ = \hat{v}^- + k_v\big(y - \hat{x}^-\big)$$

The velocity error after the update is:

$$e_v^+ = v - \hat{v}^+ = v - \hat{v}^- - k_v\big(y - \hat{x}^-\big) = e_v^- - k_v\big(y - \hat{x}^-\big).$$

Considering that:

$$(y - \hat{x}^-) = \big((x + \eta) - \hat{x}^-\big) = (e_x^- + \eta)$$

It follows that:

$$e_v^+ = e_v^- - k_v(y - \hat{x}^-) = e_v^- - k_v(\, e_x^- + \eta\,).$$

Proceeding to the variances of the left- and right-hand sides:

$$\mathrm{Var}(e_v^+) = \mathrm{Var}\big(e_v^- - k_v(\, e_x^- + \eta\,)\big)$$

$$\mathrm{Var}(e_v^+) = \mathrm{Var}(e_v^-) + k_v^2\ \mathrm{Var}(e_x^- + \eta) - 2k_v\ \mathrm{Cov}\big(e_v^-,\, e_x^- + \eta\big)$$

$$\mathrm{Var}(e_v^+) = p_{22}^- + k_v^2(p_{11}^- + \sigma_r^2) - 2k_v\, p_{12}^-. \tag{3}$$

Note:

$$\mathrm{Cov}(e_v^-,\, e_x^- + \eta) = \mathrm{Cov}(e_v^-,\, e_x^-) + \mathrm{Cov}(e_v^-,\, \eta) = p_{12}^- + 0 = p_{12}^-.$$

$$\mathrm{Cov}(e_v^-,\, \eta) = 0 \quad \text{because } e_v^- \text{ and } \eta \text{ are uncorrelated.}$$

The derivative of (3) with respect to k_v is now computed and set equal to zero in order to determine the minimum point.

$$\frac{d\ \mathrm{Var}(e_v^+)}{dk_v} = 2k_v(p_{11}^- + \sigma_r^2) - 2p_{12}^- = 0$$

$$k_v = \frac{p_{12}^-}{p_{11}^- + \sigma_r^2}.$$

The gain associated with the velocity is the predicted position–velocity covariance p_{12}^- divided by the effective variance of the quantity being measured (p_{11}^-, the predicted position uncertainty, plus σ_r^2, the measurement noise). If the position and velocity are strongly correlated (large p_{12}^-), the position measurement significantly updates the velocity. If the measurement is very noisy (large σ_r^2), the update is smaller.

Therefore, it can be stated that:

- k_x indicates how much confidence is placed in the position measurement compared with the predicted position.
- k_v indicates how much an error in the predicted position implies an error in the velocity.

Returning to the expression for $\mathrm{Var}(e_x^+)$:

$$\mathrm{Var}(e_x^+) = (1 - k_x)^2\, p_{11}^- + k_x^2\, \sigma_r^2.$$

This becomes:

$$p_{11}^+ = (1 - k_x)^2\, p_{11}^- + k_x^2\, \sigma_r^2.$$

It is observed that:

$$1 - k_x = 1 - \frac{p_{11}^-}{\sigma_r^2 + p_{11}^-} = \frac{\sigma_r^2 + p_{11}^- - p_{11}^-}{\sigma_r^2 + p_{11}^-} = \frac{\sigma_r^2}{\sigma_r^2 + p_{11}^-}.$$

Therefore, the previous equation is reconsidered:

$$p_{11}^+ = (1 - k_x)^2 \, p_{11}^- + k_x^2 \, \sigma_r^2,$$

which becomes:

$$p_{11}^+ = \left(\frac{\sigma_r^2}{\sigma_r^2 + p_{11}^-} \right)^2 p_{11}^- + \left(\frac{p_{11}^-}{\sigma_r^2 + p_{11}^-} \right)^2 \sigma_r^2.$$

By collecting terms:

$$p_{11}^+ = \frac{(\sigma_r^2)^2}{(\sigma_r^2 + p_{11}^-)^2} \, p_{11}^- + \frac{(p_{11}^-)^2}{(\sigma_r^2 + p_{11}^-)^2} \, \sigma_r^2 = \frac{\sigma_r^2 p_{11}^-}{(\sigma_r^2 + p_{11}^-)^2} \left(\sigma_r^2 + p_{11}^- \right) = \frac{\sigma_r^2 p_{11}^-}{\sigma_r^2 + p_{11}^-}.$$

Since

$$k_x = \frac{p_{11}^-}{\sigma_r^2 + p_{11}^-},$$

it follows that

$$p_{11}^+ = p_{11}^- \, (1 - k_x).$$

The update equation for the covariance has been obtained.
Important! The equation shows that the uncertainty of the estimate decreases at every iteration of the filter, since $(1 - k_x) \leq 1$.
If the measurement uncertainty is high, the Kalman gain becomes small and the estimate requires more time to converge. Conversely, when the measurement uncertainty is low, the Kalman gain increases, allowing the estimate to reach a near-zero uncertainty more quickly.
It is therefore up to the user to decide how many measurements to perform. If a voltage is being measured and a precision of 10 mV (σ) is required, measurements should be taken until the variance of the state estimate (σ^2) becomes lower than $100\,\mathrm{mV}^2$.
The individual elements of the covariance matrix P^- are now computed. The analysis begins by recalling the relationship between the errors:

$$e_{x,i}^- = e_{x,i-1}^+ + e_{v,i-1}^+ \, \Delta t + w_{x,i-1} \ ,$$

where t represents the iteration step, $w_{x,i-1}$ represents the effect of the process noise on the position (with variance q_{xx}).
The prediction of the velocity error is:

$$e_{v,i}^- = e_{v,i-1}^+ + w_{v,i-1} \ ,$$

where $w_{v,i-1}$ represents the effect of the process noise on the velocity (with variance q_{vv}).
It is assumed that the process noise terms w are independent of the a posteriori errors e_x^+ and e_v^+, which is a standard modeling assumption.
At the step t, the update of each element of the covariance matrix P_i^- is derived as follows.

1. **Element** $p_{11,i}^-$

$$p_{11,i}^- = \mathrm{Var}(e_{x,i}^-) = \mathrm{Var}(e_{x,i-1}^+ + e_{v,i-1}^+ \, \Delta t + w_{x,i-1}).$$

$$p_{11,i}^{-} = \mathrm{Var}(e_{x,i-1}^{+}) + \mathrm{Var}(e_{v,i-1}^{+}\,\Delta t) + \mathrm{Var}(w_{x,i-1}) + 2\ \mathrm{Cov}(e_{x,i-1}^{+},\, e_{v,i-1}^{+}\,\Delta t) + \\ +2\ \mathrm{Cov}(e_{x,i-1}^{+},\, w_{x,i-1}) + 2\ \mathrm{Cov}(w_{x,i-1},\, e_{v,i-1}^{+}\,\Delta t).$$

Using the assumptions of independence between w and the a posteriori errors:

$$p_{11,i}^{-} = p_{11,i-1}^{+} + \Delta t^2\, p_{22,i-1}^{+} + q_{xx} + 2\Delta t\, p_{12,i-1}^{+} + 0 + 0,$$

$$p_{11,i}^{-} = p_{11,i-1}^{+} + \Delta t^2 p_{22,i-1}^{+} + 2\Delta t\, p_{12,i-1}^{+} + q_{xx}.$$

2. Element $p_{12,i}^{-}$

$$p_{12,i}^{-} = \mathrm{Cov}(e_{x,i-1}^{-},\, e_{v,i-1}^{-}) = \mathrm{Cov}(e_{x,i-1}^{+} + e_{v,i-1}^{+}\,\Delta t + w_{x,i-1},\, e_{v,i-1}^{+} + w_{v,i-1}).$$

Expanding:

$$\begin{aligned} p_{12,i}^{-} = {} & \mathrm{Cov}(e_{x,i-1}^{+}, e_{v,i-1}^{+}) + \Delta t\ \mathrm{Var}(e_{v,i-1}^{+}) + \mathrm{Cov}(w_{x,i-1}, e_{v,i-1}^{+}) \\ & + \mathrm{Cov}(e_{x,i-1}^{+}, w_{v,i-1}) + \Delta t\ \mathrm{Cov}(e_{v,i-1}^{+}, w_{v,i-1}) + \mathrm{Cov}(w_{x,i-1}, w_{v,i-1}). \end{aligned}$$

All cross terms involving $w_{x,i-1}$ or $w_{v,i-1}$ vanish except for the process–noise covariance term:

$$\mathrm{Cov}(w_{x,i-1}, w_{v,i-1}) = q_{xv}.$$

Thus:

$$p_{12,i}^{-} = p_{12,i-1}^{+} + \Delta t\, p_{22,i-1}^{+} + 0 + 0 + \Delta t\, 0 + q_{xv},$$

$$p_{12,i}^{-} = p_{12,i-1}^{+} + \Delta t\, p_{22,i-1}^{+} + q_{xv}.$$

3. Element $p_{22,i}^{-}$

$$p_{22,i}^{-} = \mathrm{Var}(e_{v,i-1}^{-}) = \mathrm{Var}(e_{v,i-1}^{+} + w_{v,i-1}).$$

$$p_{22,i}^{-} = \mathrm{Var}(e_{v,i-1}^{+}) + \mathrm{Var}(w_{v,i-1}) + 2\ \mathrm{Cov}(e_{v,i-1}^{+}, w_{v,i-1}).$$

Since $e_{v,i-1}^{+}$ and $w_{v,i-1}$ are independent:

$$p_{22,i}^{-} = p_{22,i-1}^{+} + q_{vv}.$$

4. Element $p_{21,i}^{-}$

Because the covariance matrix is symmetric by definition:

$$p_{21,i}^{-} = p_{12,i}^{-}, \qquad (q_{xv} = q_{vx})$$

It is easy to check that the matrix form of P_i^{-} can be expressed as follows:

$$
\begin{aligned}
P_i^- &= \begin{bmatrix} p_{11,i}^- & p_{12,i}^- \\ p_{21,i}^- & p_{22,i}^- \end{bmatrix} \\
&= \begin{bmatrix} p_{11,i-1}^+ + 2\Delta t\, p_{12,i-1}^+ + \Delta t^2 p_{22,i-1}^+ + q_{xx} & p_{12,i-1}^+ + \Delta t\, p_{22,i-1}^+ + q_{xv} \\ p_{12,i-1}^+ + \Delta t\, p_{22,i-1}^+ + q_{xv} & p_{22,i-1}^+ + q_{vv} \end{bmatrix} = \\
&= \begin{bmatrix} p_{11,i-1}^+ + 2\Delta t\, p_{12,i-1}^+ + \Delta t^2 p_{22,i-1}^+ & p_{12,i-1}^+ + \Delta t\, p_{22,i-1}^+ \\ p_{12,i-1}^+ + \Delta t\, p_{22,i-1}^+ & p_{22,i-1}^+ \end{bmatrix} + \begin{bmatrix} q_{xx} & q_{xv} \\ q_{xv} & q_{vv} \end{bmatrix} = \\
&= AP_{i-1}^+ A^\top + Q,
\end{aligned}
$$

where $A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$ is the matrix defined in the state equation (1).

The following step consists in grouping all the relations obtained so far into a table. The basic definitions are recalled below:

x = true position, $\quad v$ = true velocity.

$\hat{x}^-$, $\hat{v}^-$ = predicted estimates (before using the measurement).
$\hat{x}^+$, $\hat{v}^+$ = corrected estimates (after the measurement).

y = position measurement: $\quad y = x + \eta$, with η measurement noise, zero mean and variance σ_r^2.

$e_x^- = x - \hat{x}^-$, $\quad e_v^- = v - \hat{v}^-$: predicted errors.

$\mathrm{Var}(e_x^-) = p_{11}^-$, $\quad \mathrm{Var}(e_v^-) = p_{22}^-$, $\quad \mathrm{Cov}(e_x^-, e_v^-) = p_{12}^-$.

$P^- = \begin{bmatrix} p_{11}^- & p_{12}^- \\ p_{21}^- & p_{22}^- \end{bmatrix}$

	Equation	Name
Prediction	$\hat{x}_i^- = \hat{x}_{i-1}^+ + \hat{v}_{i-1}^+ \Delta t$	Predicted position
	$\hat{v}_i^- = \hat{v}_{i-1}^+$	Predicted velocity (constant)
	$p_{11,i}^- = p_{11,i-1}^+ + 2\Delta t\, p_{12,i-1}^+ + \Delta t^2 p_{22,i-1}^+ + q_{xx}$	Predicted position variance
	$p_{12,i}^- = p_{12,i-1}^+ + \Delta t\, p_{22,i-1}^+ + q_{xv}$	Predicted pos.-velocity cov.
	$p_{22,i}^- = p_{22,i-1}^+ + q_{vv}$	Predicted velocity variance
Correction	$k_x = \dfrac{p_{11,i}^-}{\sigma_r^2 + p_{11,i}^-} \qquad k_v = \dfrac{p_{12,i}^-}{\sigma_r^2 + p_{11,i}^-}$	Kalman gain calculation
	$\tilde{y}_i = y_i - \hat{x}_i^-$	Innovation (meas. residual)
	$\hat{x}_i^+ = \hat{x}_i^- + k_x \tilde{y}_i$	Position correction
	$\hat{v}_i^+ = \hat{v}_i^- + k_v \tilde{y}_i$	Velocity correction
	$p_{11,i}^+ = (1 - k_x)\, p_{11,i}^-$	Updated position variance
	$p_{12,i}^+ = (1 - k_v)\, p_{12,i}^-$	Updated covariance
	$p_{22,i}^+ = p_{22,i}^- - k_v p_{12,i}^-$	Updated velocity variance

These equations, together with their iterative application, constitute the Kalman filter. In the case of more complex systems, the computations become heavier, and a matrix formulation is usually preferred (which is valid also for the case just considered), as reported in the following table. In this case, a multi-sensor scenario is considered, thus the variance σ_r^2 is replaced by the covariance matrix R, and the measurement $\mathbf{y}_i$ is a vector.

	$\mathbf{x}_0^+,\ P_0^+$	Initial estimation
	$\mathbf{x}_i = A\,\mathbf{x}_{i-1} + \mathbf{w}_{i-1}, \quad \mathbf{w}_{i-1} \sim \mathcal{N}(0, Q)$, $\mathbf{y}_i = H\,\mathbf{x}_i + \eta_i, \qquad \eta_i \sim \mathcal{N}(0, R)$	Basic equations (for $i \geq 1$)
Predict	$\mathbf{x}_i^- = A\,\mathbf{x}_{i-1}^+$	Prediction: state
	$P_i^- = A P_{i-1}^+ A^\top + Q$	Prediction: covariance
Update	$\tilde{\mathbf{y}}_i = \mathbf{y}_i - H\,\mathbf{x}_i^-$	Innovation
	$K_i = P_i^- H^\top \left(H P_i^- H^\top + R\right)^{-1}$	Kalman gain
	$\mathbf{x}_i^+ = \mathbf{x}_i^- + K_i \tilde{\mathbf{y}}_i$	Update: state
	$P_i^+ = (I - K_i H)\, P_i^-$	Update: covariance

The matrix expression of the Kalman gain is

$$K_i = P_i^-\, H^\top \left(H\, P_i^-\, H^\top + R\right)^{-1}.$$

Since in this example the measurement is scalar, R collaspes to σ_r^2 and the matrix H reduces to

$$H = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad H^\top = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Then

$$H\,P_i^-\,H^\top = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}p_{11,i}^- & p_{12,i}^- \\ p_{12,i}^- & p_{22,i}^-\end{bmatrix}\begin{bmatrix}1 \\ 0\end{bmatrix} = p_{11,i}^-.$$

Similarly,

$$P_i^- H^\top = \begin{bmatrix}p_{11,i}^- \\ p_{12,i}^-\end{bmatrix}.$$

Substituting these two results into the matrix formula gives

$$K_i = \begin{bmatrix}p_{11,i}^- \\ p_{12,i}^-\end{bmatrix}\frac{1}{p_{11,i}^- + \sigma_r^2},$$

that is,

$$K_i = \begin{bmatrix}\dfrac{p_{11,i}^-}{p_{11,i}^- + \sigma_r^2} \\ \dfrac{p_{12,i}^-}{p_{11,i}^- + \sigma_r^2}\end{bmatrix}.$$

This corresponds exactly to the scalar expressions previously derived for k_x and k_v.

4 Measurement and Process Noise Considerations

The variance of the measurement noise σ_r^2 depends on the type of sensor being used and does not change across iterations. Below are reported, as an example, some approximate values based on the type of sensor.

Sensor type	**Approximate value of σ_r**	**Estimated variance σ_r^2**
Temperature (NTC, Pt100)	0.01–0.05 °C	1×10^{-4} – 2.5×10^{-3} °C^2
MEMS accelerometer	tens of μg	$\sim (10\,\mu g)^2 = 1 \times 10^{-10}$ g^2
Radar / LiDAR	0.025 m	6.25×10^{-4} m^2
Relative humidity	0.3 %RH	9×10^{-2} (%RH)^2
Atmospheric pressure	0.18 Pa	3.24×10^{-2} Pa^2
Optical encoder (1024 counts/rev)	1 count (≈ 0.35 deg)	0.1225 deg^2

The process noise Q represents the uncertainty of the mathematical model being used. In general, the more the system is affected by unmodelled external disturbances, the larger Q should be. The typical form of Q for uniform linear motion (as commonly found in the literature) is:

$$Q = \sigma_a^2 \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix}$$

where σ_a^2 represents the variance of the unmodeled acceleration (such as wind, friction, uphill or downhill motion, and other effects). As an order of magnitude, it can be assumed that:

- Slow and stable system: $\sigma_a \approx 0.01\,\mathrm{m/s^2}$
- System with moderate variations: $\sigma_a \approx 0.1\,\mathrm{m/s^2}$
- Highly dynamic system: $\sigma_a \geq 1\,\mathrm{m/s^2}$

As a practical rule of thumb, one may start by estimating σ_r^2 from the sensor datasheets and by assigning intermediate initial values to Q, noting that:

- if Q is too small, the filter reacts slowly to real changes;
- if Q is too large, the filter becomes noisy.

5 Numerical example

It is now time to put into practice what has been learned by summarizing what has been presented in this tutorial regarding the application of the Kalman filter to the uniform linear motion of a vehicle. The procedure follows the methodology previously introduced. The state vector is defined as follows:

$$\mathbf{x}(t) = \begin{bmatrix} x(t) \\ v(t) \end{bmatrix}$$

Where $x(t)$ is the position a time t and $v(t)$ is the velocity at the same time. According to physics, the dynamic model of the process is:

$$\frac{d}{dt}\left(\begin{bmatrix} x(t) \\ v(t) \end{bmatrix}\right) = \begin{bmatrix} v(t) \\ 0 \end{bmatrix}$$

$\frac{dv(t)}{dt} = 0$ because the motion is assumed to occur at constant velocity.
By discretizing with a time step Δt and introducing a process noise term:

$$\mathbf{x}_i = A\,\mathbf{x}_{i-1} + \mathbf{w}_{i-1}$$

Note: the notation $\mathbf{x}_i$ (in bold) indicates the state vector, not the position. The matrix A is given by:

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{w} \sim \mathcal{N}(0, Q)$$

Only the position is measured at each iteration step i. The measurement, as usual, is affected by noise:

$$y_i = H\,\mathbf{x}_i + \eta_i, \qquad H = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad \eta_i \sim \mathcal{N}(0, \sigma_r^2)$$

The Kalman relations are therefore:

Prediction:

$$\mathbf{x}_i^- = A\,\mathbf{x}_{i-1}^+$$
$$P_i^- = A\,P_{i-1}^+ A^\top + Q$$

Correction:

$$K_i = P_i^- H^\top \left(H\,P_i^- H^\top + \sigma_r^2\right)^{-1}$$
$$\mathbf{x}_i^+ = \mathbf{x}_i^- + K_i\left(y_i - H\,\mathbf{x}_i^-\right)$$

The methodology is now available to begin performing calculations. The initial assumptions are defined as follows.

Initial state:

$$\mathbf{x}_0^+ = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Initial covariance matrix:

$$P_0^+ = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$$

- Initial position variance: $p_{11,0}^+ = 1.0\ \text{km}^2$
- Initial velocity variance: $p_{22,0}^+ = 4.0\ (\text{km/min})^2$

Process noise Q:

$$Q = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$$

- $q_{xx} = q_{11} = 0.01\ \text{km}^2$
- $q_{vv} = q_{22} = 0.01\ (\text{km/min})^2$

Measurement noise variance:

$$\sigma_r^2 = 0.25\ \text{km}^2 \quad \text{(standard deviation: } 0.5\ \text{km)}$$

The time step is set to $\Delta t = 1$ minute.

First iteration ($i = 1$)

Prediction

First step

$$\mathbf{x}_1^- = A\,\mathbf{x}_0^+ = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Second step

$$P_1^- = A\,P_0^+ A^\top + Q$$

$$= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix} + Q$$

$$= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 4 \\ 0 & 4 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 4 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix} = \begin{bmatrix} 5 & 4 \\ 4 & 4 \end{bmatrix}$$

$$P_1^- = \begin{bmatrix} 5 & 4 \\ 4 & 4 \end{bmatrix} + \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} = \begin{bmatrix} 5.01 & 4 \\ 4 & 4.01 \end{bmatrix}$$

Measurement

Assume that the measured position is $y_1 = 1.1\,\text{km}$ (with noise).

Correction

Third step

$$K_1 = P_1^- H^\top \left(H\,P_1^- H^\top + \sigma_r^2 \right)^{-1}$$

$$= \begin{bmatrix} 5.01 & 4 \\ 4 & 4.01 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 5.01 & 4 \\ 4 & 4.01 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \sigma_r^2 \right)^{-1}$$

$$= \begin{bmatrix} 5.01 \\ 4 \end{bmatrix} (5.01 + 0.25)^{-1} = \begin{bmatrix} 5.01 \\ 4 \end{bmatrix} \frac{1}{5.26} = \begin{bmatrix} 0.9529 \\ 0.7601 \end{bmatrix}$$

Fourth step

$$\mathbf{x}_1^+ = \mathbf{x}_1^- + K_1 \left(y_1 - H\,\mathbf{x}_1^-\right)$$

$$= \begin{bmatrix}0\\0\end{bmatrix} + \begin{bmatrix}0.9529\\0.7601\end{bmatrix} \left(1.1 - \begin{bmatrix}1 & 0\end{bmatrix} \begin{bmatrix}0\\0\end{bmatrix}\right) = \begin{bmatrix}0.9529\\0.7601\end{bmatrix} (1.1) = \begin{bmatrix}1.0482\\0.8361\end{bmatrix}$$

Fifth step

$$P_1^+ = (I - K_1 H)\, P_1^-$$

$$= \left(\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} - \begin{bmatrix}0.9529\\0.7601\end{bmatrix} \begin{bmatrix}1 & 0\end{bmatrix}\right) \begin{bmatrix}5.01 & 4\\4 & 4.01\end{bmatrix}$$

$$= \left(\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} - \begin{bmatrix}0.9529 & 0\\0.7601 & 0\end{bmatrix}\right) \begin{bmatrix}5.01 & 4\\4 & 4.01\end{bmatrix}$$

$$= \begin{bmatrix}1 - 0.9529 & 0\\-0.7601 & 1\end{bmatrix} \begin{bmatrix}5.01 & 4\\4 & 4.01\end{bmatrix} = \begin{bmatrix}0.253 & 0.191\\0.191 & 0.950\end{bmatrix}$$

To summarize:

Iteration 1		
Quantity	**Value**	
Predicted state	$\mathbf{x}_1^- = \begin{bmatrix}0\\0\end{bmatrix}$	
Predicted covariance	$P_1^- = \begin{bmatrix}5.01 & 4\\4 & 4.01\end{bmatrix}$	
Kalman gain	$K_1 = \begin{bmatrix}0.9529\\0.7601\end{bmatrix}$	
Measurement	$y_1 = 1.1$ km	
Updated state	$\mathbf{x}_1^+ = \begin{bmatrix}1.0482\\0.8361\end{bmatrix}$	Input for the next iteration
Updated covariance	$P_1^+ = \begin{bmatrix}0.253 & 0.191\\0.191 & 0.95\end{bmatrix}$	

Second iteration ($i = 2$)

Prediction

First step

$$\mathbf{x}_2^- = A\,\mathbf{x}_1^+ = \begin{bmatrix}1 & \Delta t\\0 & 1\end{bmatrix} \begin{bmatrix}1.0482\\0.8361\end{bmatrix} = \begin{bmatrix}1 & 1\\0 & 1\end{bmatrix} \begin{bmatrix}1.0482\\0.8361\end{bmatrix} = \begin{bmatrix}1.8843\\0.8361\end{bmatrix}$$

Second step

$$P_2^- = A\,P_1^+ A^\top + Q$$

$$= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.253 & 0.191 \\ 0.191 & 0.95 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix} + Q$$

$$= \begin{bmatrix} 1.595 & 1.141 \\ 1.141 & 0.96 \end{bmatrix}$$

Measurement

Assume that the measured position is $y_2 = 2.2\,\text{km}$ (with noise).

Correction

Third step

$$K_2 = P_2^- H^\top \left(H\,P_2^- H^\top + \sigma_r^2\right)^{-1}$$

$$= \begin{bmatrix} 1.595 & 1.141 \\ 1.141 & 0.96 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1.595 & 1.141 \\ 1.141 & 0.96 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \sigma_r^2 \right)^{-1}$$

$$= \begin{bmatrix} 1.595 \\ 1.141 \end{bmatrix} (1.595 + 0.25)^{-1} = \begin{bmatrix} 1.595 \\ 1.141 \end{bmatrix} \frac{1}{1.845} = \begin{bmatrix} 0.8645 \\ 0.6184 \end{bmatrix}$$

Fourth step

$$\mathbf{x}_2^+ = \mathbf{x}_2^- + K_2 \left(y_2 - H\,\mathbf{x}_2^-\right)$$

$$= \begin{bmatrix} 1.8843 \\ 0.8361 \end{bmatrix} + \begin{bmatrix} 0.8645 \\ 0.6184 \end{bmatrix} \left(2.2 - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1.8843 \\ 0.8361 \end{bmatrix} \right)$$

$$= \begin{bmatrix} 1.8843 \\ 0.8361 \end{bmatrix} + \begin{bmatrix} 0.8645 \\ 0.6184 \end{bmatrix} (0.3157) = \begin{bmatrix} 2.1570 \\ 1.0316 \end{bmatrix}$$

Fifth step

$$P_2^+ = (I - K_2 H)\,P_2^-$$

$$= \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.8645 \\ 0.6184 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \right) \begin{bmatrix} 1.595 & 1.141 \\ 1.141 & 0.96 \end{bmatrix} =$$

$$= \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.8645 & 0 \\ 0.6184 & 0 \end{bmatrix} \right) \begin{bmatrix} 1.595 & 1.141 \\ 1.141 & 0.96 \end{bmatrix} = \begin{bmatrix} 0.1355 & 0 \\ -0.6184 & 1 \end{bmatrix} \begin{bmatrix} 1.595 & 1.141 \\ 1.141 & 0.96 \end{bmatrix} = \begin{bmatrix} 0.216 & 0.155 \\ 0.155 & 0.254 \end{bmatrix}$$

To summarize:

Iteration 2		
Quantity	**Value**	
Predicted state	$\mathbf{x}_2^- = \begin{bmatrix} 1.8843 \\ 0.8361 \end{bmatrix}$	
Predicted covariance	$P_2^- = \begin{bmatrix} 1.595 & 1.141 \\ 1.141 & 0.960 \end{bmatrix}$	
Kalman gain	$K_2 = \begin{bmatrix} 0.8645 \\ 0.6184 \end{bmatrix}$	
Measurement	$y_2 = 2.2$ km	
Updated state	$\mathbf{x}_2^+ = \begin{bmatrix} 2.1570 \\ 1.0316 \end{bmatrix}$	Input for the next iteration
Updated covariance	$P_2^+ = \begin{bmatrix} 0.216 & 0.155 \\ 0.155 & 0.254 \end{bmatrix}$	

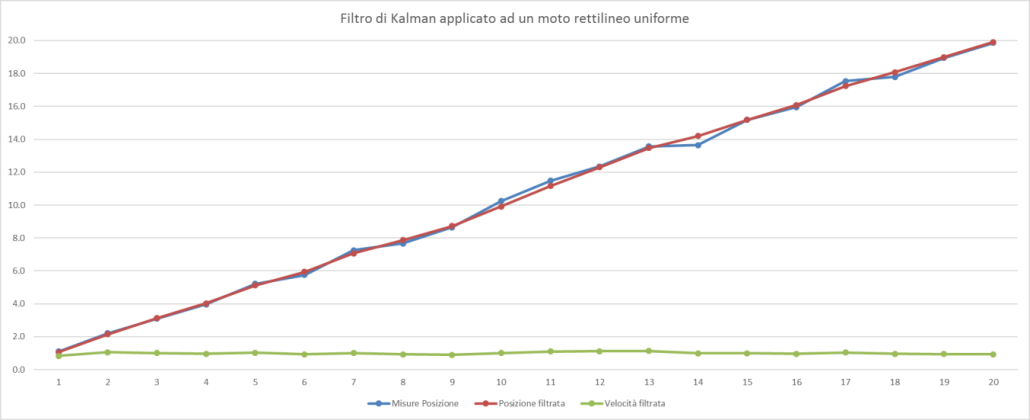

Figure 8: Chart of the filtered signals

The figure above shows a plot of a simulation of uniform linear motion. The red line represents the filtered signal, while the blue line corresponds to the measurements provided by the sensor. The green line indicates the estimated velocity (1 km/min = 60 km/h). A Python source code for simulating this specific application of the Kalman filter is provided at the end of the tutorial.

6 Kalman filter for complex systems

In complex systems, the state may change not only due to its natural evolution but also because of external actions. This contribution has not been considered so far. To include this information in the model, the control matrix B is introduced; it describes how an external action, represented by a control input u, affects the estimated state of the system in the Kalman filter. The state evolution equation must therefore be modified as follows:

$$\mathbf{x}_i = A\,\mathbf{x}_{i-1} + B\,\mathbf{u}_{i-1} + \mathbf{w}_{i-1}.$$

Where:

- A represents the natural dynamics of the system, that is, how the state evolves on its own in the absence of control actions.
- $B\,\mathbf{u}_{i-1}$ is the contribution due to external actions applied to the system: an engine providing thrust, an actuator moving a robotic arm, or a flight controller modifying the trajectory.
- $\mathbf{x}_i$ is the state of the system at iteration step i.

The control matrix B is used, for example, to:

- correctly scale the input: an acceleration expressed in $\mathrm{m/s}^2$ must be converted into an increment of velocity (m/s) or position (meters), taking into account the elapsed time;
- distribute the effect: a single control command may influence multiple state variables (for example, an acceleration affects both velocity and position);
- incorporate the time step Δt: many models include factors such as Δt or $\frac{\Delta t^2}{2}$ to respect physical laws.

Consider the case of an object falling under gravity. The gravitational acceleration g can be modeled as a constant control input: the matrix B converts this acceleration into changes in velocity and position at each step. In this way, even without measurements, the filter assumes that the velocity increases linearly and the position grows quadratically with time.
Without the matrix B, that external contribution would simply not exist in the model, and the filter would incorrectly attribute those changes to noise or measurement errors. This example is examined in detail below.

Figure 9: A more complex example

The procedure follows the methodology previously introduced. The state vector is defined as:

$$\mathbf{x}(t) = \begin{bmatrix} h(t) \\ \\ v(t) \end{bmatrix}$$

Where $h(t)$ is the (vertical) height at time t and $v(t)$ is the falling velocity at the same time. Physics describes the dynamic model of the process as:

$$\frac{d}{dt}\left(\begin{bmatrix} h(t) \\ \\ v(t) \end{bmatrix}\right) = \begin{bmatrix} v(t) \\ \\ -g \end{bmatrix}$$

The term g denotes the gravitational acceleration, equal to 9.81 $\mathrm{m/s^2}$. By discretizing with a time step Δt, the state equation becomes:

$$\mathbf{x}_i = A\,\mathbf{x}_{i-1} + B\,\mathbf{u}_{i-1} + \mathbf{w}_{i-1}.$$

Note: the notation $\mathbf{x}_i$ refers to the state vector, not to a position. The matrices A, B, and the control input u are given by:

$$A = \begin{bmatrix} 1 & \Delta t \\ \\ 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \\ \Delta t \end{bmatrix}, \qquad \mathbf{u}_{i-1} = u = -g. \text{ ($\mathbf{u}_{i-1}$ is a scalar and constant value)}$$

Recall that $\mathbf{w}$ represents the process noise. The term involving the matrix B has been introduced here because it is needed to formalize the causes that produce changes in the state. Only the height is measured at iteration step i. As usual, the measurement is affected by noise:

$$y_i = H\,\mathbf{x}_i + \eta_i, \qquad H = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

The sampling time is now set to $\Delta t = 0.1\,\mathrm{s}$. The process noise $\mathbf{w}$, with covariance Q, affects only the acceleration g with a small uncertainty, for example $\sigma_a^2 = 0.1\ (\mathrm{m/s^2})^2$. The measurement noise on the height is $\sigma_r^2 = 0.5\ \mathrm{m}^2$.
The Kalman relations are therefore:

Prediction:

$$\mathbf{x}_i^- = A\,\mathbf{x}_{i-1}^+ - B\,g$$
$$P_i^- = A\,P_{i-1}^+ A^\top + Q$$

Correction:

$$K_i = P_i^- H^\top \left(H\,P_i^- H^\top + \sigma_r^2\right)^{-1}$$
$$\mathbf{x}_i^+ = \mathbf{x}_i^- + K_i\left(y_i - H\,\mathbf{x}_i^-\right)$$
$$P_i^+ = (I - K_i H)\,P_i^-$$

Matrix Q and variance σ_r^2:

$$Q = \sigma_a^2 \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix} = \begin{bmatrix} 2.5 \times 10^{-6} & 5 \times 10^{-5} \\ \\ 5 \times 10^{-5} & 1 \times 10^{-3} \end{bmatrix}$$

$$\sigma_r^2 = 0.5$$

All the elements are now available to start the iterations of the Kalman filter.
The initial state is defined as:

$$\mathbf{x}_0^+ = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$$

The initial covariance matrix is:

$$P_0^+ = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

First iteration ($i = 1$)

Prediction

First step

$$\mathbf{x}_1^- = A\,\mathbf{x}_0^+ - B\,g = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \end{bmatrix} - \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} \quad (9.81)$$

$$= \begin{bmatrix} 10+0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.005 \\ 0.1 \end{bmatrix} (9.81) = \begin{bmatrix} 10 \\ 0 \end{bmatrix} + \begin{bmatrix} -0.04905 \\ -0.981 \end{bmatrix} = \begin{bmatrix} 9.95095 \\ -0.981 \end{bmatrix}$$

Second step

$$P_1^- = AP_0^+A^\top + Q = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0.1 & 1 \end{bmatrix} + Q = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0.1 & 1 \end{bmatrix} + Q$$

$$= \begin{bmatrix} 1.0100025 & 0.10005 \\ 0.10005 & 1.001 \end{bmatrix}$$

Measurement

Assume that the measured height is $y_1 = 10.2\,\mathrm{m}$ (with noise).

Correction

Third step

$$K_1 = P_1^- H^\top \left(H\,P_1^- H^\top + \sigma_r^2 \right)^{-1}$$

$$= \begin{bmatrix} 1.0100025 & 0.10005 \\ 0.10005 & 1.001 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1.0100025 & 0.10005 \\ 0.10005 & 1.001 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \sigma_r^2 \right)^{-1}$$

$$= \begin{bmatrix} 1.0100025 \\ 0.10005 \end{bmatrix} (1.0100025 + 0.5)^{-1} = \begin{bmatrix} 1.0100025 \\ 0.10005 \end{bmatrix} \frac{1}{1.5100025} = \begin{bmatrix} 0.669 \\ 0.0663 \end{bmatrix}$$

Fourth step

$$\mathbf{x}_1^+ = \mathbf{x}_1^- + K_1\left(y_1 - H\,\mathbf{x}_1^-\right)$$

$$= \begin{bmatrix} 9.95095 \\ -0.981 \end{bmatrix} + \begin{bmatrix} 0.669 \\ 0.0663 \end{bmatrix} \left(10.2 - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 9.95095 \\ -0.981 \end{bmatrix} \right)$$

$$= \begin{bmatrix} 9.95095 \\ -0.981 \end{bmatrix} + \begin{bmatrix} 0.669 \\ 0.0663 \end{bmatrix} (10.2 - 9.95095) = \begin{bmatrix} 9.95095 \\ -0.981 \end{bmatrix} + \begin{bmatrix} 0.669 \\ 0.0663 \end{bmatrix} (0.24905)$$

$$\approx \begin{bmatrix} 9.95095 \\ -0.981 \end{bmatrix} + \begin{bmatrix} 0.1666 \\ 0.0165 \end{bmatrix} \approx \begin{bmatrix} 10.118 \\ -0.965 \end{bmatrix}$$

Fifth step

$$P_1^+ = (I - K_1 H)\,P_1^-$$

$$= \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.669 \\ 0.0663 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \right) \begin{bmatrix} 1.0100025 & 0.10005 \\ 0.10005 & 1.001 \end{bmatrix}$$

$$= \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.669 & 0 \\ 0.0663 & 0 \end{bmatrix} \right) \begin{bmatrix} 1.0100025 & 0.10005 \\ 0.10005 & 1.001 \end{bmatrix}$$

$$= \begin{bmatrix} 1 - 0.669 & 0 \\ -0.0663 & 1 \end{bmatrix} \begin{bmatrix} 1.0100025 & 0.10005 \\ 0.10005 & 1.001 \end{bmatrix} = \begin{bmatrix} 0.331 & 0 \\ -0.0663 & 1 \end{bmatrix} \begin{bmatrix} 1.0100025 & 0.10005 \\ 0.10005 & 1.001 \end{bmatrix} \approx \begin{bmatrix} 0.334 & 0.0331 \\ 0.0331 & 0.994 \end{bmatrix}$$

To summarize:

Iteration 1		
Quantity	**Value**	
Predicted state	$\mathbf{x}_1^- = \begin{bmatrix} 9.95095 \\ -0.981 \end{bmatrix}$	
Predicted covariance	$P_1^- = \begin{bmatrix} 1.0100025 & 0.10005 \\ 0.10005 & 1.001 \end{bmatrix}$	
Kalman gain	$K_1 = \begin{bmatrix} 0.669 \\ 0.0663 \end{bmatrix}$	
Measurement	$y_1 = 10.2$ m	
Updated state	$\mathbf{x}_1^+ = \begin{bmatrix} 10.118 \\ -0.965 \end{bmatrix}$	Input for the next iteration
Updated covariance	$P_1^+ = \begin{bmatrix} 0.334 & 0.0331 \\ 0.0331 & 0.994 \end{bmatrix}$	

Second iteration ($i = 2$)

The procedure is now ready to continue with the next iteration.
Assuming that the chosen measurement is $y_2 = 9.8$ m, the determination of the intermediate quantities is left to the reader as an exercise. The resulting values are summarized in the table below:

Iteration 2		
Quantity	**Value**	
Predicted state	$\mathbf{x}_2^- = \begin{bmatrix} 9.972 \\ -1.945 \end{bmatrix}$	
Predicted covariance	$P_2^- = \begin{bmatrix} 0.351 & 0.133 \\ 0.133 & 0.995 \end{bmatrix}$	
Kalman gain	$K_2 = \begin{bmatrix} 0.412 \\ 0.156 \end{bmatrix}$	
Measurement	$y_2 = 9.8$ m	
Updated state	$\mathbf{x}_2^+ = \begin{bmatrix} 9.901 \\ -1.972 \end{bmatrix}$	Input for the next iteration
Updated covariance	$P_2^+ = \begin{bmatrix} 0.206 & 0.078 \\ 0.078 & 0.975 \end{bmatrix}$	

7 Conclusions

The path followed in this tutorial, starting from a simple uniform linear motion with position-only measurements, has made it possible to understand how the Kalman filter can transform noisy data into reliable estimates. Its strength does not lie only in "cleaning" the measurements, but primarily in optimally combining two sources of information: the state model (prediction) and the experimental observations (correction).

A central point is the importance of correctly modeling the uncertainties, both those of the measurements (η) and those of the process ($\mathbf{w}$). In the case of uniform linear motion, even without a direct measurement of the velocity, the filter was able to estimate it with increasing accuracy by exploiting the relationship between position and velocity.

This idea was then extended to another scenario: the fall of an object under the effect of gravity. In this case, the Kalman filter can accurately estimate both the falling velocity and the gravitational acceleration, even when starting from noisy position measurements.

If these examples helped clarify the logic behind the Kalman filter, the next step is to address more complex scenarios: estimating the trajectory of a drone subject to wind, tracking a satellite in orbit, or integrating GPS and IMU sensors in a vehicle. Each new context requires a careful choice of the model, the matrices Q and R, and the strategies to handle states that are not directly observable. As complexity increases, it becomes evident that the Kalman filter is not merely an algorithm, but a true method for intelligently fusing information in the presence of uncertainty.

A Python programs

Uniform linear motion

```python
import numpy as np

# Problem parameters
dt = 1  # minuti

# Model matrices
A = np.array([[1, dt],
              [0, 1]])

H = np.array([[1, 0]])

Q = np.array([[0.01, 0],
              [0, 0.01]])

R = np.array([[0.25]])

# Estimated initial state (position, velocity)
x_plus = np.array([[0.0],
                   [0.0]])

# Estimated initial covariance matrix
P_plus = np.array([[1, 0],
                   [0, 4]])

measurements = [1.1, 2.2, 3.1, 4.0, 5.2, 5.9, 6.8, 7.9, 8.7, 10.4]

print(f"{'k':^3} | {'y':^6} | {'x-pos-':^8} | {'x-vel-':^8} | "
      f"{'x+pos':^8} | {'x+vel':^8} | "
      f"{'Kx':^8} | {'Kv':^8} | "
      f"{'p11+':^8} | {'p12+':^8} | {'p21+':^8} | {'p22+':^8}")
print("-"*120)

for k, y in enumerate(measurements, start=1):
    # Predict
    x_minus = A @ x_plus
    P_minus = A @ P_plus @ A.T + Q

    # Update
    t = y - (H @ x_minus)
    S = H @ P_minus @ H.T + R
    K = P_minus @ H.T @ np.linalg.inv(S)

    x_plus = x_minus + K @ t
    P_plus = (np.eye(2) - K @ H) @ P_minus

    Kx = K[0,0]; Kv = K[1,0]

    print(f"{k:^3} | {y:6.2f} | {x_minus[0,0]:8.3f} | {x_minus[1,0]:8.3f} |
        "
          f"{x_plus[0,0]:8.3f} | {x_plus[1,0]:8.3f} | "
          f"{Kx:8.4f} | {Kv:8.4f} | "
          f"{P_plus[0,0]:8.4f} | {P_plus[0,1]:8.4f} | "
          f"{P_plus[1,0]:8.4f} | {P_plus[1,1]:8.4f}")
```

Accelerated motion (gravity)

```python
import numpy as np

dt = 0.1
g = -9.81

A = np.array([[1, dt],
              [0, 1]])

B = np.array([[0.5*dt*dt],
              [dt]])

H = np.array([[1, 0]])

Q = np.array([[2.5e-6, 5e-5],
              [5e-5,   1e-3]])

R = np.array([[0.5]])

x_plus = np.array([[10.0],
                   [0.0]])

P_plus = np.array([[1.0, 0.0],
                   [0.0, 1.0]])

measurements = [10.2, 9.8, 9.5]

print(f"{'k':^3} | {'y':^6} | {'x-pos-':^8} | {'x-vel-':^8} | "
      f"{'x+pos':^8} | {'x+vel':^8} | "
      f"{'Kx':^8} | {'Kv':^8} | "
      f"{'p11+':^8} | {'p12+':^8} | {'p21+':^8} | {'p22+':^8}")
print("-"*120)

for k, y in enumerate(measurements, start=1):
    # Predict
    x_minus = A @ x_plus + B * g
    P_minus = A @ P_plus @ A.T + Q

    # Update
    t = y - (H @ x_minus)
    S = H @ P_minus @ H.T + R
    K = P_minus @ H.T @ np.linalg.inv(S)

    x_plus = x_minus + K @ t
    P_plus = (np.eye(2) - K @ H) @ P_minus

    Kx = K[0,0]; Kv = K[1,0]

    print(f"{k:^3} | {y:6.2f} | {x_minus[0,0]:8.3f} | {x_minus[1,0]:8.3f} | "
          f"{x_plus[0,0]:8.3f} | {x_plus[1,0]:8.3f} | "
          f"{Kx:8.4f} | {Kv:8.4f} | "
          f"{P_plus[0,0]:8.4f} | {P_plus[0,1]:8.4f} | "
          f"{P_plus[1,0]:8.4f} | {P_plus[1,1]:8.4f}")
```