

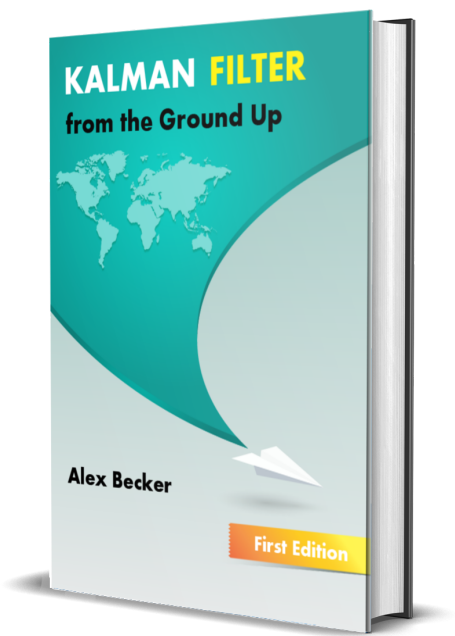
# THE $\alpha - \beta - \gamma$ FILTER

This chapter is introductory, and it describes the  $\alpha - \beta$  and  $\alpha - \beta - \gamma$  filters. These filters are frequently used for time series data smoothing. The principles of the  $\alpha - \beta(-\gamma)$  filter are closely related to the Kalman Filter principles.

## EXAMPLE 1 – WEIGHTING THE GOLD

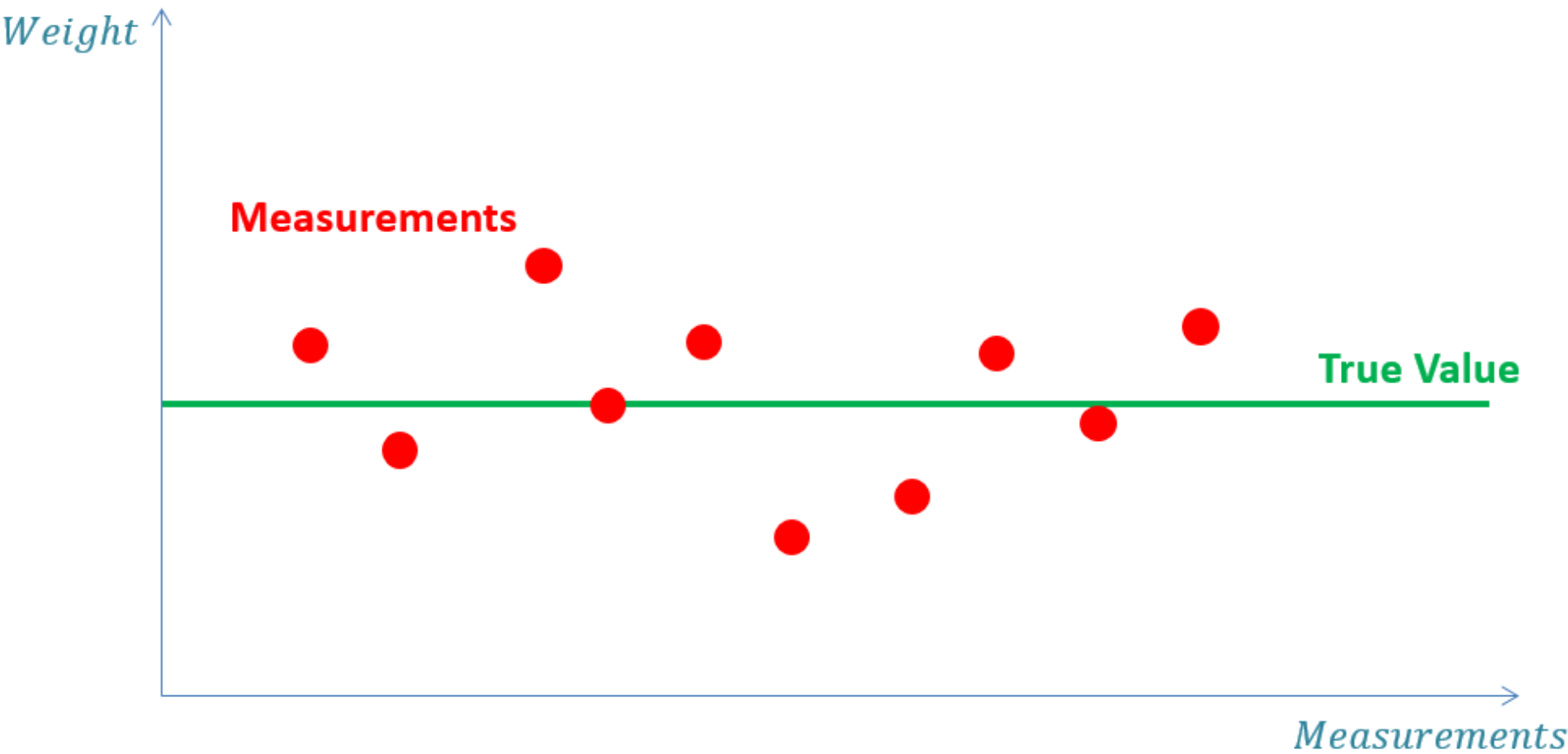
Now we are ready for the first simple example. In this example, we estimate the state of the static system. A static system is a system that doesn't change its state over a reasonable period. For instance, the static system could be a tower, and the state would be its height.

In this example, we estimate the weight of the gold bar. We have unbiased scales, i.e., the measurements don't have a systematic error, but the measurements do include random noise.



 [Get the book](#)

The system is the gold bar, and the system state is the weight of the gold bar. The dynamic model of the system is constant since we assume that the weight doesn't change over short periods.



At the time  $n$ , the estimate  $\hat{x}_{n,n}$  would be the average of all previous measurements:

$$\hat{x}_{n,n} = \frac{1}{n}(z_1 + z_2 + \dots + z_{n-1} + z_n) = \frac{1}{n} \sum_{i=1}^n (z_i)$$

Example Notation:

- $x$  is the true value of the weight
- $z_n$  is the measured value of the weight at time  $n$
- $\hat{x}_{n,n}$  is the estimate of  $x$  at time  $n$  (the estimate is made after taking the measurement  $z_n$  )
- $\hat{x}_{n+1,n}$  is the estimate of the future state ( $n + 1$  ) of  $x$ . The estimate is made at the time  $n$ . In other words,  $\hat{x}_{n+1,n}$  is a predicted state or extrapolated state
- $\hat{x}_{n-1,n-1}$  is the estimate of  $x$  at time  $n - 1$  (the estimate is made after taking the measurement  $z_{n-1}$  )
- $\hat{x}_{n,n-1}$  is a prior prediction - the estimate of the state at time  $n$ . The prediction is made at the time  $n - 1$

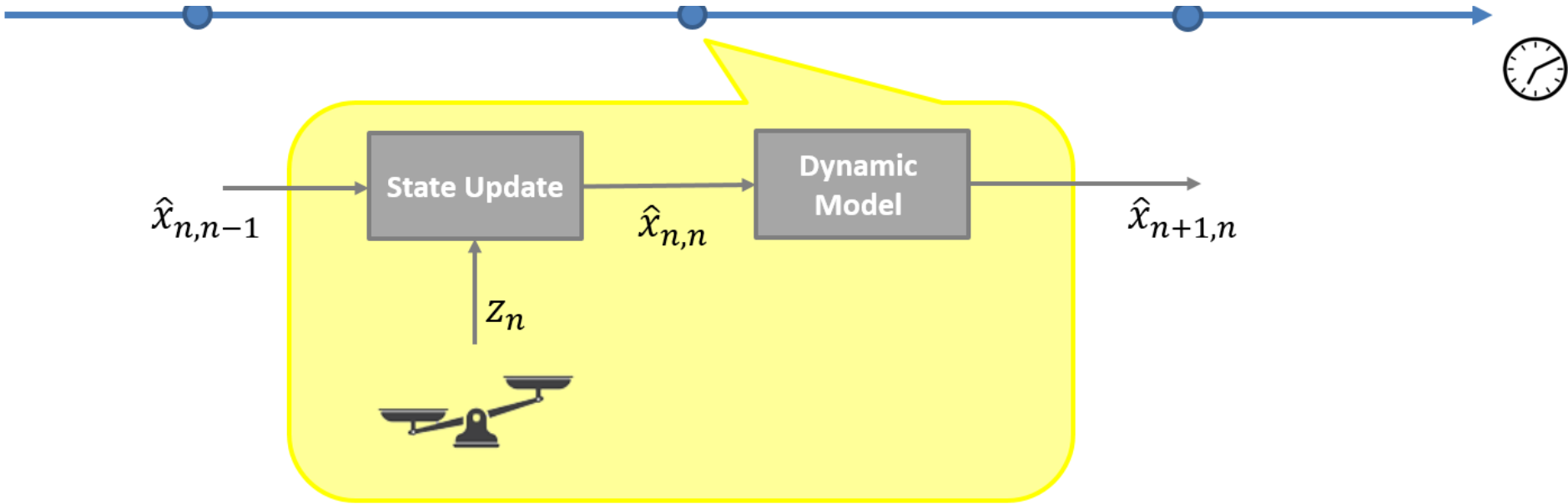
Note: In the literature, a caret (or hat) over a variable indicates an estimated value.

The dynamic model in this example is static (or constant) since the weight of gold doesn't change over time, therefore  $\hat{x}_{n+1,n} = \hat{x}_{n,n}$ .

Although the above equation is mathematically correct, it is not practical for implementation. In order to estimate  $\hat{x}_{n,n}$  we need to remember all historical measurements; therefore, we need a large memory. We also need to recalculate the average repeatedly if we want to update the estimated value after every new measurement. Thus, we need a more powerful CPU.

It would be more practical to keep the last estimate only ( $\hat{x}_{n-1,n-1}$ ) and update it after every new measurement. The following figure exemplifies the required algorithm:

- Estimate the current state based on the measurement and prior prediction.
- Predict the next state based on the current state estimate using the Dynamic Model.



We can modify the averaging equation for our needs using a small mathematical trick:

Notes	
$\hat{x}_{n,n} = \frac{1}{n} \sum_{i=1}^n (z_i)$	Average formula: sum of $n$ measurements divided by $n$
$= \frac{1}{n} \left( \sum_{i=1}^{n-1} (z_i) + z_n \right)$	Sum of the $n - 1$ measurements plus the last measurement divided by $n$
$= \frac{1}{n} \sum_{i=1}^{n-1} (z_i) + \frac{1}{n} z_n$	Expand
$= \frac{1}{n} \frac{n-1}{n-1} \sum_{i=1}^{n-1} (z_i) + \frac{1}{n} z_n$	Multiply and divide by term $n - 1$
$= \frac{n-1}{n} \frac{1}{n-1} \sum_{i=1}^{n-1} (z_i) + \frac{1}{n} z_n$	Reorder The 'orange' term is the prior estimate
$= \frac{n-1}{n} \hat{x}_{n-1,n-1} + \frac{1}{n} z_n$	Rewriting the sum
$= \hat{x}_{n-1,n-1} - \frac{1}{n} \hat{x}_{n-1,n-1} + \frac{1}{n} z_n$	Distribute the term $\frac{n-1}{n}$
$= \hat{x}_{n-1,n-1} + \frac{1}{n} (z_n - \hat{x}_{n-1,n-1})$	Reorder

$\hat{x}_{n-1,n-1}$  is the estimated state of  $x$  at the time  $n - 1$ , based on the measurement at the time  $n - 1$ .

Let's find  $\hat{x}_{n,n-1}$  (the predicted state of  $x$  at the time  $n$ ), based on  $\hat{x}_{n-1,n-1}$  (the estimation at the time  $n - 1$ ). In other words, we would like to extrapolate  $\hat{x}_{n-1,n-1}$  to the time  $n$ .

Since the dynamic model in this example is static, the predicted state of  $x$  equals the estimated state of  $x$ :  $\hat{x}_{n,n-1} = \hat{x}_{n-1,n-1}$ .

Based on the above, the estimate of the current state  $\hat{x}_{n,n}$ , can be written as follows:

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + \frac{1}{n} (z_n - \hat{x}_{n,n-1})$$

The above equation is one of the five Kalman filter equations. It is called the **State Update Equation**. It means the following:

value of the current state

=

value of the current state

+

Factor

X

Measurement

-

value of the current state

The factor  $\frac{1}{n}$  is specific to our example. We will discuss the vital role of this factor later, but right now, I would like to note that in "Kalman Filter language," this factor is called the **Kalman Gain**. It is denoted by  $K_n$ . The subscript  $n$  indicates that the Kalman Gain can change with every iteration.

The discovery of  $K_n$  was one of Rudolf Kalman's significant contributions.

Before we get into the guts of the Kalman Filter, we use the Greek letter  $\alpha_n$  instead of  $K_n$ .

So, the State Update Equation looks as follows:

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + \alpha_n (z_n - \hat{x}_{n,n-1})$$

The term  $(z_n - \hat{x}_{n,n-1})$  is the "measurement residual," also called **innovation**. The innovation contains new information.

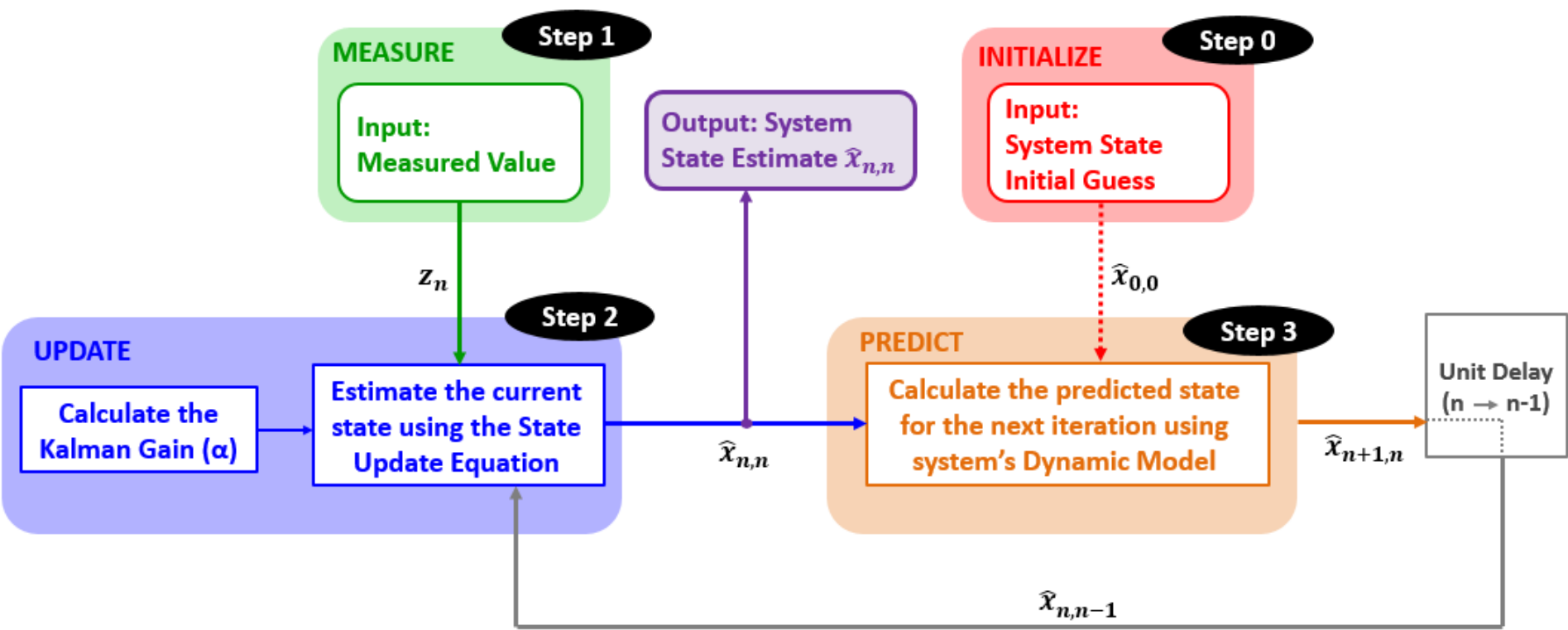
In this example,  $\frac{1}{n}$  decreases as  $n$  increases. In the beginning, we don't have enough information about the current state; thus, the first estimation is based on the first measurement  $\frac{1}{n}|_{n=1} = 1$ . As we continue, each successive measurement has less weight in the estimation process, since  $\frac{1}{n}$  decreases. At some point, the contribution of the new measurements will become negligible.

Let's continue with the example. Before we make the first measurement, we can guess (or rough estimate) the gold bar weight simply by reading the stamp on the gold bar. It is called the **Initial Guess**, and it is our first estimate.

The Kalman Filter requires the initial guess as a preset, which can be very rough.

ESTIMATION ALGORITHM

The following chart depicts the estimation algorithm that is used in this example.



Now we are ready to start the measurement and estimation process.

THE NUMERICAL EXAMPLE

ITERATION ZERO

INITIALIZATION

Our initial guess of the gold bar weight is 1000 grams. The initial guess is used only once for the filter initiation. Thus, it won't be required for successive iterations.

PREDICTION

The weight of the gold bar is not supposed to change. Therefore, the dynamic model of the system is static. Our next state estimate (prediction) equals the initialization:

$$\hat{x}_{1,0} = \hat{x}_{0,0} = 1000g$$

FIRST ITERATION

STEP 1

Making the weight measurement with the scales:

$$z_1 = 996g$$

STEP 2

Calculating the gain. In our example  $\alpha_n = \frac{1}{n}$  , thus:

$$\alpha_1 = \frac{1}{1} = 1$$

Calculating the current estimate using the State Update Equation:

$$\hat{x}_{1,1} = \hat{x}_{1,0} + \alpha_1 \left( z_1 - \hat{x}_{1,0} \right) = 1000 + 1 \left( 996 - 1000 \right) = 996g$$

Note: The initial guess could be any number in this specific example. Since  $\alpha_1 = 1$  , the initial guess is eliminated in the first iteration.

STEP 3

The dynamic model of the system is static; thus, the weight of the gold bar is not supposed to change. Our next state estimate (prediction) equals to current state estimate:

$$\hat{x}_{2,1} = \hat{x}_{1,1} = 996g$$

SECOND ITERATION

After a unit time delay, the **predicted estimate** from the previous iteration becomes the **prior estimate** in the current iteration:

$$\hat{x}_{2,1} = 996g$$

STEP 1

Making the second measurement of the weight:

$$z_2 = 994g$$

STEP 2

Calculating the gain:

$$\alpha_2 = \frac{1}{2}$$

$$\hat{x}_{2,2} = \hat{x}_{2,1} + \alpha_2 \left( z_2 - \hat{x}_{2,1} \right) = 996 + \frac{1}{2}(994 - 996) = 995g$$

STEP 3

$$\hat{x}_{3,2} = \hat{x}_{2,2} = 995g$$

THIRD ITERATION

$$z_3 = 1021g \qquad \alpha_3 = \frac{1}{3}$$

$$\hat{x}_{3,3} = 995 + \frac{1}{3}(1021 - 995) = 1003.67g$$

$$\hat{x}_{4,3} = 1003.67g$$

FOURTH ITERATION

$$z_4 = 1000g \qquad \alpha_4 = \frac{1}{4}$$

$$\hat{x}_{4,4} = 1003.67 + \frac{1}{4}(1000 - 1003.67) = 1002.75g$$

$$\hat{x}_{5,4} = 1002.75g$$

FIFTH ITERATION

$$z_5 = 1002g \qquad \alpha_5 = \frac{1}{5}$$

$$\hat{x}_{5,5} = 1002.75 + \frac{1}{5}(1002 - 1002.75) = 1002.6g$$

$$\hat{x}_{6,5} = 1002.6g$$

SIXTH ITERATION

$$z_6 = 1010g \qquad \alpha_6 = \frac{1}{6}$$

$$\hat{x}_{6,6} = 1002.6 + \frac{1}{6}(1010 - 1002.6) = 1003.83$$

$$\hat{x}_{7,6} = 1003.83g$$

SEVENTH ITERATION

$$z_7 = 983g \qquad \alpha_7 = \frac{1}{7}$$

$$\hat{x}_{7,7} = 1003.83 + \frac{1}{7}(983 - 1003.83) = 1000.86g$$

$$\hat{x}_{8,7} = 1000.86g$$

EIGHTH ITERATION

$$\hat{x}_{8,8} = 1000.86 + \frac{1}{8}(971 - 1000.86) = 997.125g$$
$$\hat{x}_{9,8} = 997.125g$$

NINTH ITERATION

$$z_9 = 993g \qquad \alpha_9 = \frac{1}{9}$$
$$\hat{x}_{9,9} = 997.125 + \frac{1}{9}(993 - 997.125) = 996.67g$$
$$\hat{x}_{10,9} = 996.67g$$

TENTH ITERATION

$$z_{10} = 1023g \qquad \alpha_{10} = \frac{1}{10}$$
$$\hat{x}_{10,10} = 996.67 + \frac{1}{10}(1023 - 996.67) = 999.3g$$
$$\hat{x}_{11,10} = 999.3g$$

We can stop here. The gain decreases with each measurement. Therefore, the contribution of each successive measurement is lower than the contribution of the previous measurement. We get pretty close to the true weight, which is 1000g. If we were making more measurements, we would get closer to the true value.

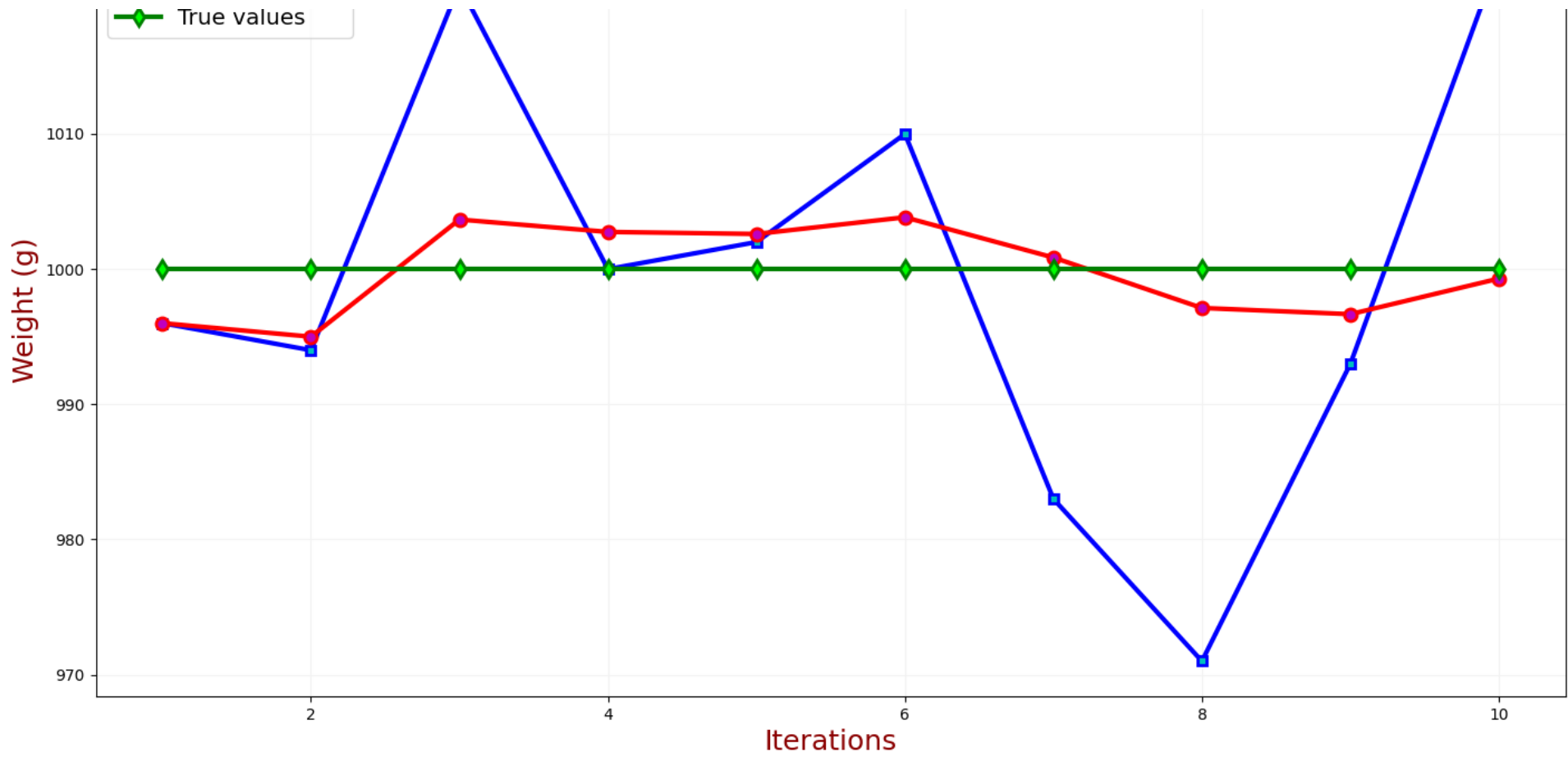
The following table summarizes our measurements and estimates, and the chart compares the measured values, the estimates, and the true value.

<i>n</i>	1	2	3	4	5	6	7	8	9	10
<i>α<sub>n</sub></i>	1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{7}$	$\frac{1}{8}$	$\frac{1}{9}$	$\frac{1}{10}$
<i>z<sub>n</sub></i>	996	994	1021	1000	1002	1010	983	971	993	1023
$\hat{x}_{n,n}$	996	995	1003.67	1002.75	1002.6	1003.83	1000.86	997.125	996.67	999.3
$\hat{x}_{n+1,n}$	996	995	1003.67	1002.75	1002.6	1003.83	1000.86	997.125	996.67	999.3

RESULTS ANALYSIS

The following chart compares the true, measured, and estimated values.





The estimation algorithm has a smoothing effect on the measurements and converges toward the true value.

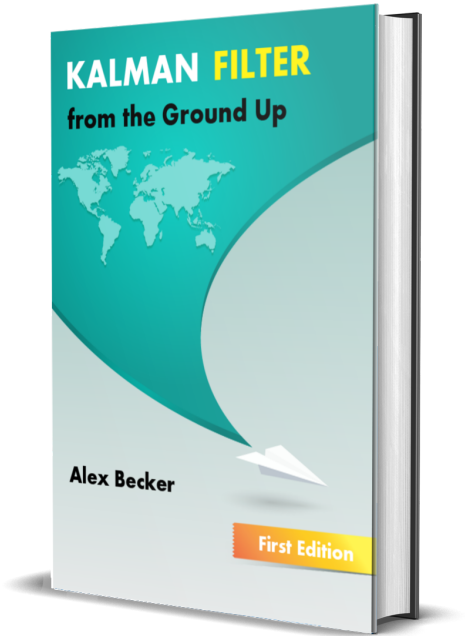
EXAMPLE SUMMARY

In this example, we've developed a simple estimation algorithm for a static system. We have also derived the state update equation, one of the five Kalman Filter equations. We will revise the state update equation [in the next chapter](#).

EXAMPLE 2 – TRACKING THE CONSTANT VELOCITY AIRCRAFT IN ONE DIMENSION

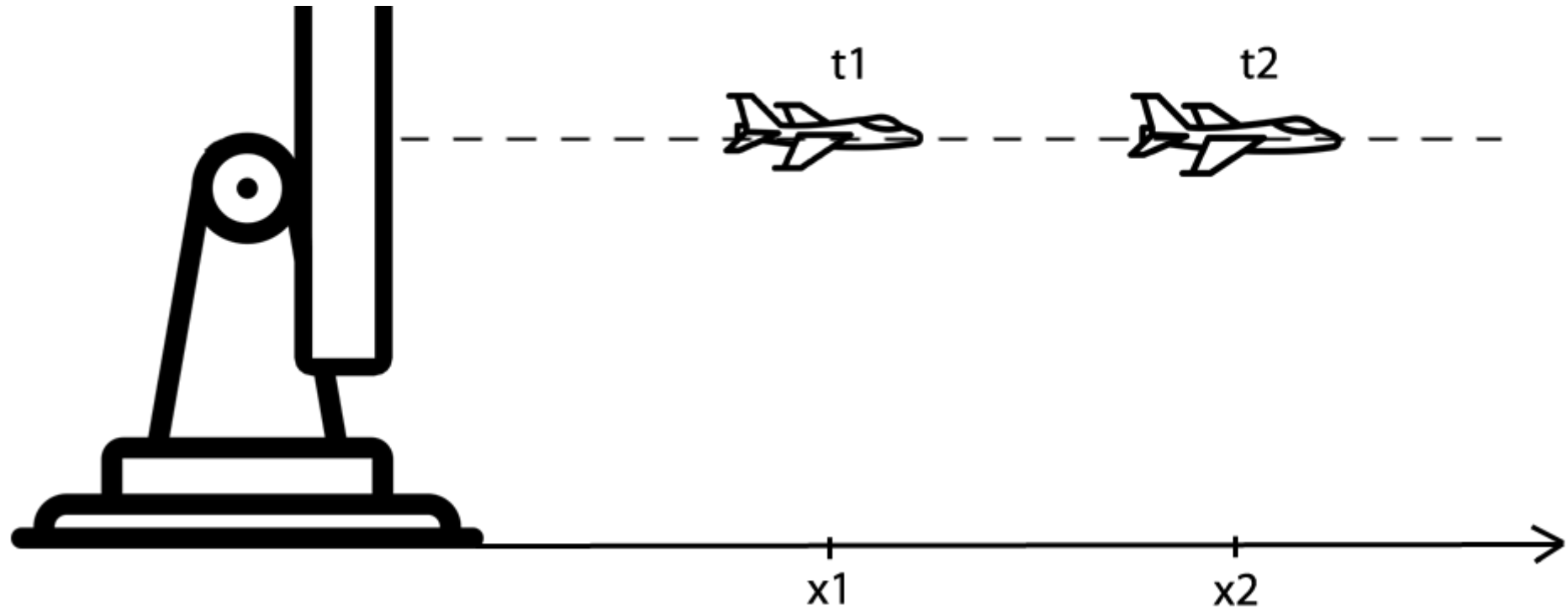
It is time to examine a dynamic system that changes its state over time. In this example, we track a constant-velocity aircraft in one dimension using the  $\alpha - \beta$  filter.

Let us assume a one-dimensional world. We assume an aircraft that is moving radially away from the radar (or towards the radar). In the one-dimensional world, the angle to the radar is constant, and the aircraft altitude is constant, as shown in the following figure.



Get the book





$x_n$  represents the range to the aircraft at time  $n$ . The aircraft velocity can be approximated using the range differentiation method - the change in the measured range with time.

Thus, the velocity is a derivative of the range:

$$\dot{x} = v = \frac{dx}{dt}$$

The radar sends a track beam in the direction of the target at a constant rate. The track-to-track interval is  $\Delta t$ .

Two motion equations describe the system dynamic model for constant velocity motion:

$$x_{n+1} = x_n + \Delta t \dot{x}_n$$
$$\dot{x}_{n+1} = \dot{x}_n$$

According to these equations, the aircraft range at the next track cycle equals the range at the current track cycle plus the target velocity multiplied by the track-to-track interval. Since we assume constant velocity in this example, the velocity at the next cycle equals the velocity at the current cycle.

The above system of equations is called a **State Extrapolation Equation** (also called a **Transition Equation** or a **Prediction Equation**) and is also one of the five Kalman filter equations. This system of equations extrapolates the current state to the next state (prediction).

We have already used the State Extrapolation Equation in the previous example, where we assumed that the weight at the next state equals the weight at the current state.

The State Extrapolation Equations depend on the system dynamics and differ from example to example.

There is a general form of the State Extrapolation Equation in matrix notation. We learn it later.

In this example, we use the above equations specific to our case.

Note: we have already learned two of the five Kalman Filter equations:

- State Update Equation
- State Extrapolation Equation

Now we are going to modify the State Update Equation for our example.

Let the radar track-to-track (  $\Delta t$  ) period be 5 seconds. Assume that at time  $n - 1$ , the estimated range of the UAV (Unmanned Aerial Vehicle) is 30,000m, and the estimated UAV velocity is 40m/s.

Using the State Extrapolation Equations, we can predict the target position at time  $n$ :

$$\hat{x}_{n,n-1} = \hat{x}_{n-1,n-1} + \Delta t \hat{\dot{x}}_{n-1,n-1} = 30000 + 5 \times 40 = 30200m$$

The target velocity prediction for time  $n$ :

$$\hat{\dot{x}}_{n,n-1} = \hat{\dot{x}}_{n-1,n-1} = 40m/s$$

However, at time  $n$ , the radar measures range (  $z_n$  ) of 30,110m and not 30,200m as expected. There is a 90m gap between the expected (predicted) range and the measured range. There are two possible reasons for this gap:

- The radar measurements are not precise
- The aircraft velocity has changed. The new aircraft velocity is:  $\frac{30,110-30,000}{5} = 22m/s$

Which of the two statements is true?

Let us write down the State Update Equation for the velocity:

$$\hat{\dot{x}}_{n,n} = \hat{\dot{x}}_{n,n-1} + \beta \left( \frac{z_n - \hat{x}_{n,n-1}}{\Delta t} \right)$$

The value of the factor  $\beta$  depends on the precision level of the radar. Suppose that the  $1\sigma$  precision of the radar is 20m. The 90 meters gap between the predicted and measured ranges most likely results from a change in the aircraft velocity. We should set the  $\beta$  factor to a high value in this case. If we set  $\beta = 0.9$ , then the estimated velocity would be:

$$\hat{\dot{x}}_{n,n} = \hat{\dot{x}}_{n,n-1} + \beta \left( \frac{z_n - \hat{x}_{n,n-1}}{\Delta t} \right) = 40 + 0.9 \left( \frac{30110 - 30200}{5} \right) = 23.8m/s$$

On the other hand, suppose that the  $1\sigma$  precision of the radar is 150m. Then the 90 meters gap probably results from the radar measurement error. We should set the  $\beta$  factor to a low value in this case. If we set  $\beta = 0.1$ , then the estimated velocity would be:

$$\hat{\dot{x}}_{n,n} = \hat{\dot{x}}_{n,n-1} + \beta \left( \frac{z_n - \hat{x}_{n,n-1}}{\Delta t} \right) = 40 + 0.1 \left( \frac{30110 - 30200}{5} \right) = 38.2m/s$$

If the aircraft velocity has changed from 40m/s to 22m/s, we see this after 10 track cycles (running the above equation 10 times with  $\beta = 0.1$ ). If the gap has been caused by measurement error, then the successive measurements would be in front or behind the predicted positions. Thus on average, the target velocity would not change.

The State Update Equation for the aircraft position is similar to the equation that was derived in the previous example:

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + \alpha (z_n - \hat{x}_{n,n-1})$$

Unlike the previous example, where the  $\alpha$  factor is recalculated in each iteration (  $\alpha_n = \frac{1}{n}$  ), the  $\alpha$  factor is constant in this example.

The magnitude of the  $\alpha$  factor depends on the radar measurement precision. For high precision-radar, we should choose high  $\alpha$ , giving high weight to the measurements. If  $\alpha = 1$ , then the estimated range equals the measured range:

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + 1 (z_n - \hat{x}_{n,n-1}) = z_n$$

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + 0 \left( z_n - \hat{x}_{n,n-1} \right) = \hat{x}_{n,n-1}$$

So, we have derived a system of equations that composes the State Update Equation for the radar tracker. They are also called  $\alpha - \beta$  track update equations or  $\alpha - \beta$  track filtering equations.

The State Update Equation for position:

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + \alpha \left( z_n - \hat{x}_{n,n-1} \right)$$

The State Update Equation for velocity:

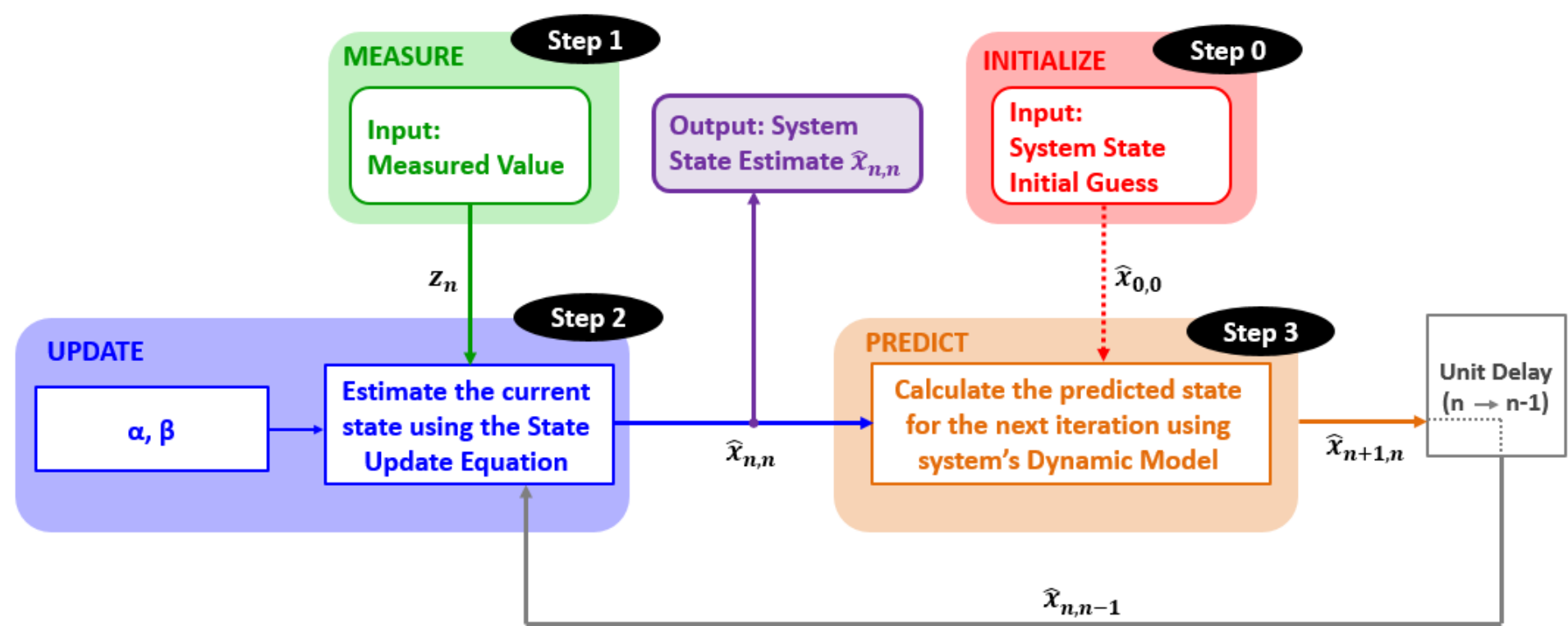
$$\hat{\dot{x}}_{n,n} = \hat{\dot{x}}_{n,n-1} + \beta \left( \frac{z_n - \hat{x}_{n,n-1}}{\Delta t} \right)$$

Note: In some books, the  $\alpha - \beta$  filter is called the **g-h filter**, where the Greek letter  $\alpha$  is replaced by the English letter g, and the English letter h replaces the Greek letter  $\beta$ .

Note: In this example, we are deriving the aircraft velocity from the range measurements ( $\dot{x} = \frac{\Delta x}{\Delta t}$ ). Modern radars can measure radial velocity directly using the Doppler Effect. However, my goal is to explain the Kalman Filter, not the radar operation. So, for the sake of generality, I will keep deriving the velocity from the range measurements in our examples.

ESTIMATION ALGORITHM

The following chart depicts the estimation algorithm that is used in this example.



Unlike the previous example, the Gain values  $\alpha$  and  $\beta$  are given for this example. For the Kalman Filter, the  $\alpha$  and  $\beta$  are replaced by Kalman Gain, which is calculated at each iteration, but we learn it later.

Now we are ready to start a numerical example.

THE NUMERICAL EXAMPLE

Consider an aircraft moving radially toward (or away from) a radar in a one-dimensional world.

The  $\alpha - \beta$  filter parameters are:

The track-to-track interval is 5 seconds.

Note: In this example, we use an imprecise radar and a low-speed target (UAV) for better graphical representation. The radars are usually more precise in real life, and the targets can be much faster.

ITERATION ZERO

INITIALIZATION

The initial conditions for the time  $n = 0$  are given:

$$\begin{aligned}\hat{x}_{0,0} &= 30000m \\ \hat{\dot{x}}_{0,0} &= 40m/s\end{aligned}$$

Note: The **Track Initiation** (or how we get the initial conditions) is an important topic that will be discussed later. Right now, our goal is to understand the basic  $\alpha - \beta$  filter operation, so let's assume that the initial conditions are given by somebody else.

PREDICTION

The initial guess should be extrapolated to the first cycle using the State Extrapolation Equations:

$$\begin{aligned}\hat{x}_{n+1,n} &= \hat{x}_{n,n} + \Delta t \hat{\dot{x}}_{n,n} \rightarrow \hat{x}_{1,0} = \hat{x}_{0,0} + \Delta t \hat{\dot{x}}_{0,0} = 30000 + 5 \times 40 = 30200m \\ \hat{\dot{x}}_{n+1,n} &= \hat{\dot{x}}_{n,n} \rightarrow \hat{\dot{x}}_{1,0} = \hat{\dot{x}}_{0,0} = 40m/s\end{aligned}$$

FIRST ITERATION

In the first cycle (  $n = 1$  ), the initial guess is the prior estimate:

$$\begin{aligned}\hat{x}_{n,n-1} &= \hat{x}_{1,0} = 30200m \\ \hat{\dot{x}}_{n,n-1} &= \hat{\dot{x}}_{1,0} = 40m/s\end{aligned}$$

STEP 1

The radar measures the aircraft range:

$$z_1 = 30171m$$

STEP 2

Calculating the current estimate using the State Update Equation:

$$\begin{aligned}\hat{x}_{1,1} &= \hat{x}_{1,0} + \alpha \left( z_1 - \hat{x}_{1,0} \right) = 30200 + 0.2 \left( 30171 - 30200 \right) = 30194.2m \\ \hat{\dot{x}}_{1,1} &= \hat{\dot{x}}_{1,0} + \beta \left( \frac{z_1 - \hat{x}_{1,0}}{\Delta t} \right) = 40 + 0.1 \left( \frac{30171 - 30200}{5} \right) = 39.42m/s\end{aligned}$$

STEP 3

Calculating the next state estimate using the State Extrapolation Equations:

$$\hat{\dot{x}}_{2,1} = \hat{\dot{x}}_{1,1} = 39.42m/s$$

SECOND ITERATION

After a unit time delay, the predicted estimate from the previous iteration becomes the prior estimate in the current iteration:

$$\begin{aligned}\hat{x}_{2,1} &= 30391.3m \\ \hat{\dot{x}}_{2,1} &= 39.42m/s\end{aligned}$$

STEP 1

The radar measures the aircraft range:

$$z_2 = 30353m$$

STEP 2

Calculating the current estimate using the State Update Equation:

$$\begin{aligned}\hat{x}_{2,2} &= \hat{x}_{2,1} + \alpha \left( z_2 - \hat{x}_{2,1} \right) = 30391.3 + 0.2 \left( 30353 - 30391.3 \right) = 30383.64m \\ \hat{\dot{x}}_{2,2} &= \hat{\dot{x}}_{2,1} + \beta \left( \frac{z_2 - \hat{x}_{2,1}}{\Delta t} \right) = 39.42 + 0.1 \left( \frac{30353 - 30391.3}{5} \right) = 38.65m/s\end{aligned}$$

STEP 3

Calculating the next state estimate using the State Extrapolation Equations:

$$\begin{aligned}\hat{x}_{3,2} &= \hat{x}_{2,2} + \Delta t \hat{\dot{x}}_{2,2} = 30383.64 + 5 \times 38.65 = 30576.9m \\ \hat{\dot{x}}_{3,2} &= \hat{\dot{x}}_{2,2} = 38.65m/s\end{aligned}$$

THIRD ITERATION

$$\begin{aligned}z_3 &= 30756m \\ \hat{x}_{3,3} &= 30576.9 + 0.2 \left( 30756 - 30576.9 \right) = 30612.73m \\ \hat{\dot{x}}_{3,3} &= 38.65 + 0.1 \left( \frac{30756 - 30576.9}{5} \right) = 42.2m/s \\ \hat{x}_{4,3} &= 30612.73 + 5 \times 42.2 = 30823.9m \\ \hat{\dot{x}}_{4,3} &= 42.2m/s\end{aligned}$$

FOURTH ITERATION

$$\begin{aligned}z_4 &= 30799m \\ \hat{x}_{4,4} &= 30823.9 + 0.2 \left( 30799 - 30823.9 \right) = 30818.93m \\ \hat{\dot{x}}_{4,4} &= 42.2 + 0.1 \left( \frac{30799 - 30823.9}{5} \right) = 41.7m/s \\ \hat{x}_{5,4} &= 30818.93 + 5 \times 41.7 = 31027.6m \\ \hat{\dot{x}}_{5,4} &= 41.7m/s\end{aligned}$$

FIFTH ITERATION

$$\begin{aligned}\hat{x}_{5,5} &= 31027.6 + 0.2 (31018 - 31027.6) = 31025.7m \\ \hat{\dot{x}}_{5,5} &= 41.7 + 0.1 \left( \frac{31018 - 31027.6}{5} \right) = 41.55m/s \\ \hat{x}_{6,5} &= 31025.7 + 5 \times 41.55 = 31233.4m \\ \hat{\dot{x}}_{6,5} &= 41.55m/s\end{aligned}$$

SIXTH ITERATION

$$\begin{aligned}z_6 &= 31278m \\ \hat{x}_{6,6} &= 31233.4 + 0.2 (31278 - 31233.4) = 31242.3m \\ \hat{\dot{x}}_{6,6} &= 41.55 + 0.1 \left( \frac{31278 - 31233.4}{5} \right) = 42.44m/s \\ \hat{x}_{7,6} &= 31242.3 + 5 \times 42.44 = 31454.5m \\ \hat{\dot{x}}_{7,6} &= 42.44m/s\end{aligned}$$

SEVENTH ITERATION

$$\begin{aligned}z_7 &= 31276m \\ \hat{x}_{7,7} &= 31454.5 + 0.2 (31276 - 31454.5) = 31418.8m \\ \hat{\dot{x}}_{7,7} &= 42.44 + 0.1 \left( \frac{31276 - 31454.5}{5} \right) = 38.9m/s \\ \hat{x}_{8,7} &= 31418.8 + 5 \times 38.9 = 31613.15m \\ \hat{\dot{x}}_{8,7} &= 38.9m/s\end{aligned}$$

EIGHTH ITERATION

$$\begin{aligned}z_8 &= 31379m \\ \hat{x}_{8,8} &= 31613.15 + 0.2 (31379 - 31613.15) = 31566.3m \\ \hat{\dot{x}}_{8,8} &= 38.9 + 0.1 \left( \frac{31379 - 31613.15}{5} \right) = 34.2m/s \\ \hat{x}_{9,8} &= 31566.3 + 5 \times 34.2 = 31737.24m \\ \hat{\dot{x}}_{9,8} &= 34.2m/s\end{aligned}$$

NINTH ITERATION

$$\begin{aligned}z_9 &= 31748m \\ \hat{x}_{9,9} &= 31737.24 + 0.2 (31748 - 31737.24) = 31739.4m \\ \hat{\dot{x}}_{9,9} &= 34.2 + 0.1 \left( \frac{31748 - 31737.24}{5} \right) = 34.4m/s \\ \hat{x}_{10,9} &= 31739.4 + 5 \times 34.4 = 31911.4m \\ \hat{\dot{x}}_{10,9} &= 34.4m/s\end{aligned}$$

TENTH ITERATION

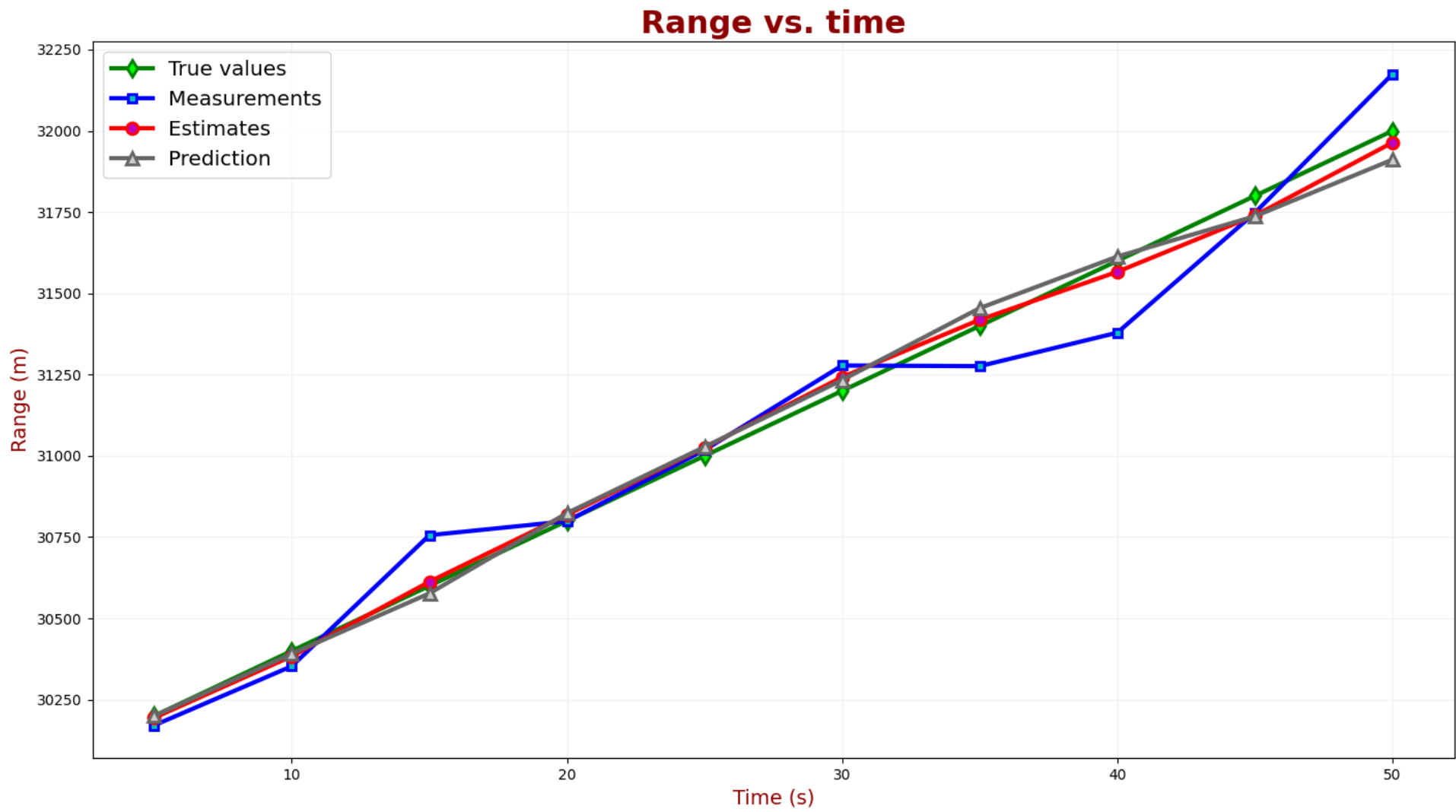
$$\hat{x}_{10,10} = 31911.4 + 0.2 (32175 - 31911.4) = 31964.1m$$
$$\hat{\dot{x}}_{10,10} = 34.4 + 0.1 \left( \frac{32175 - 31911.4}{5} \right) = 39.67m/s$$
$$\hat{x}_{11,10} = 31964.1 + 5 \times 39.67 = 32162.45m$$
$$\hat{\dot{x}}_{11,10} = 39.67m/s$$

The following table summarizes our measurements and estimates.

<i>n</i>	1	2	3	4	5	6	7	8	9	10
<i>z<sub>n</sub></i>	30171	30353	30756	30799	31018	31278	31276	31379	31748	32175
<i><math>\hat{x}_{n,n}</math></i>	30194.2	30383.64	30612.73	30818.93	31025.7	31242.3	31418.8	31566.3	31739.4	31964.1
<i><math>\dot{\hat{x}}_{n,n}</math></i>	39.42	38.65	42.2	41.7	41.55	42.44	38.9	34.2	34.4	39.67
<i><math>\hat{x}_{n+1,n}</math></i>	30391.3	30576.9	30823.9	31027.6	31233.4	31454.5	31613.15	31737.24	31911.4	32162.45
<i><math>\dot{\hat{x}}_{n+1,n}</math></i>	39.42	38.65	42.2	41.7	41.55	42.44	38.9	34.2	34.4	39.67

RESULTS ANALYSIS

The following chart compares the true values, measured values, and estimates.

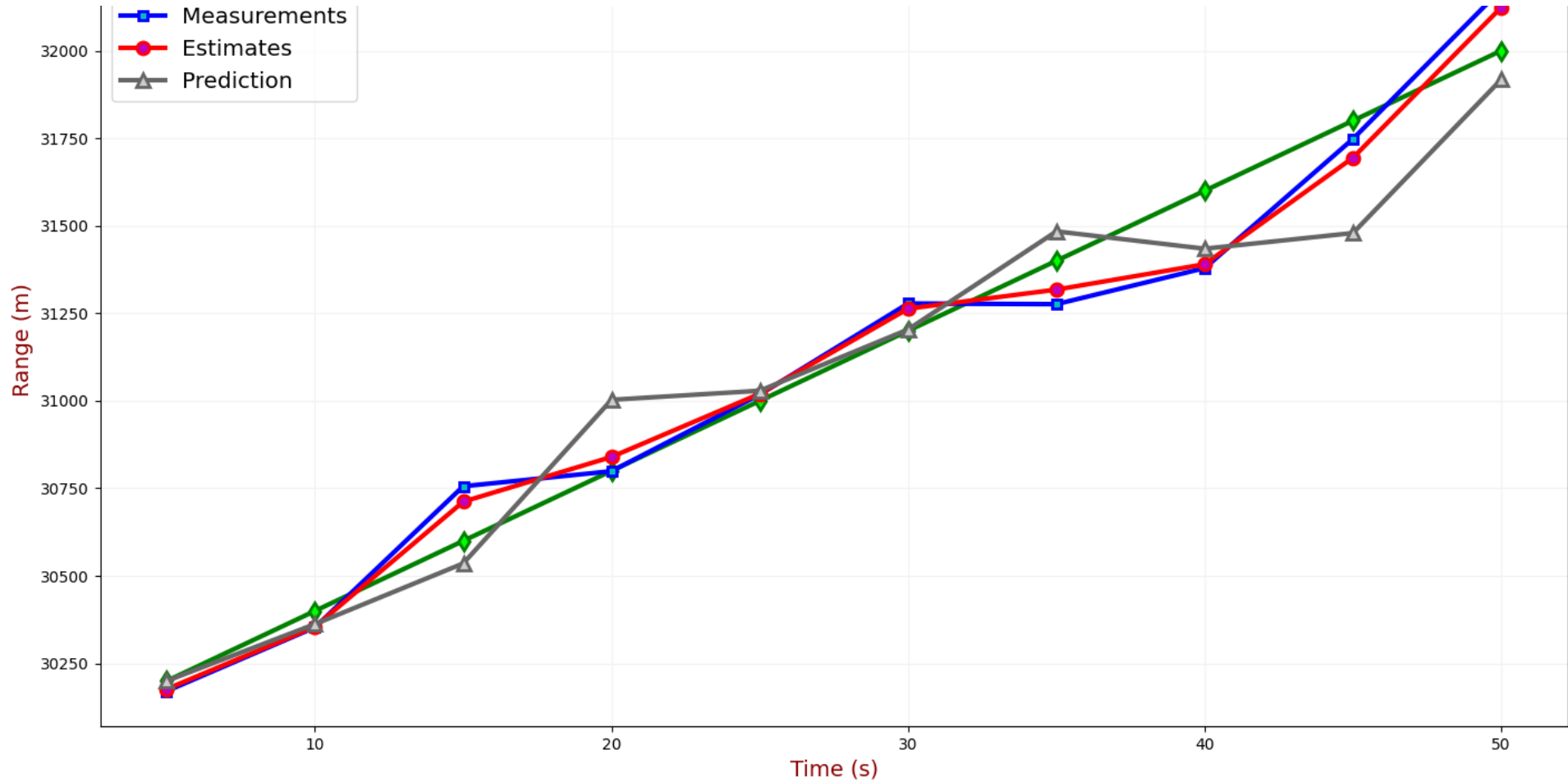


Our estimation algorithm has a smoothing effect on the measurements and converges toward the true value.

USING HIGH  $\alpha$  AND  $\beta$

The following chart depicts the true, measured, and estimated values for  $\alpha = 0.8$  and  $\beta = 0.5$ .





The "smoothing" degree of this filter is much lower. The "current estimate" is very close to the measured values, and predicted estimate errors are relatively high.

So, shall we always choose low values for  $\alpha$  and  $\beta$  ?

The answer is NO. The value of  $\alpha$  and  $\beta$  should depend on the measurement precision. If we use high-precision equipment, like laser radar, we would prefer a high  $\alpha$  and  $\beta$  that follow measurements. In this case, the filter would quickly respond to a velocity change of the target. On the other hand, if measurement precision is low, we prefer low  $\alpha$  and  $\beta$ . In this case, the filter smoothes the uncertainty (errors) in the measurements. However, the filter reaction to target velocity changes would be much slower.

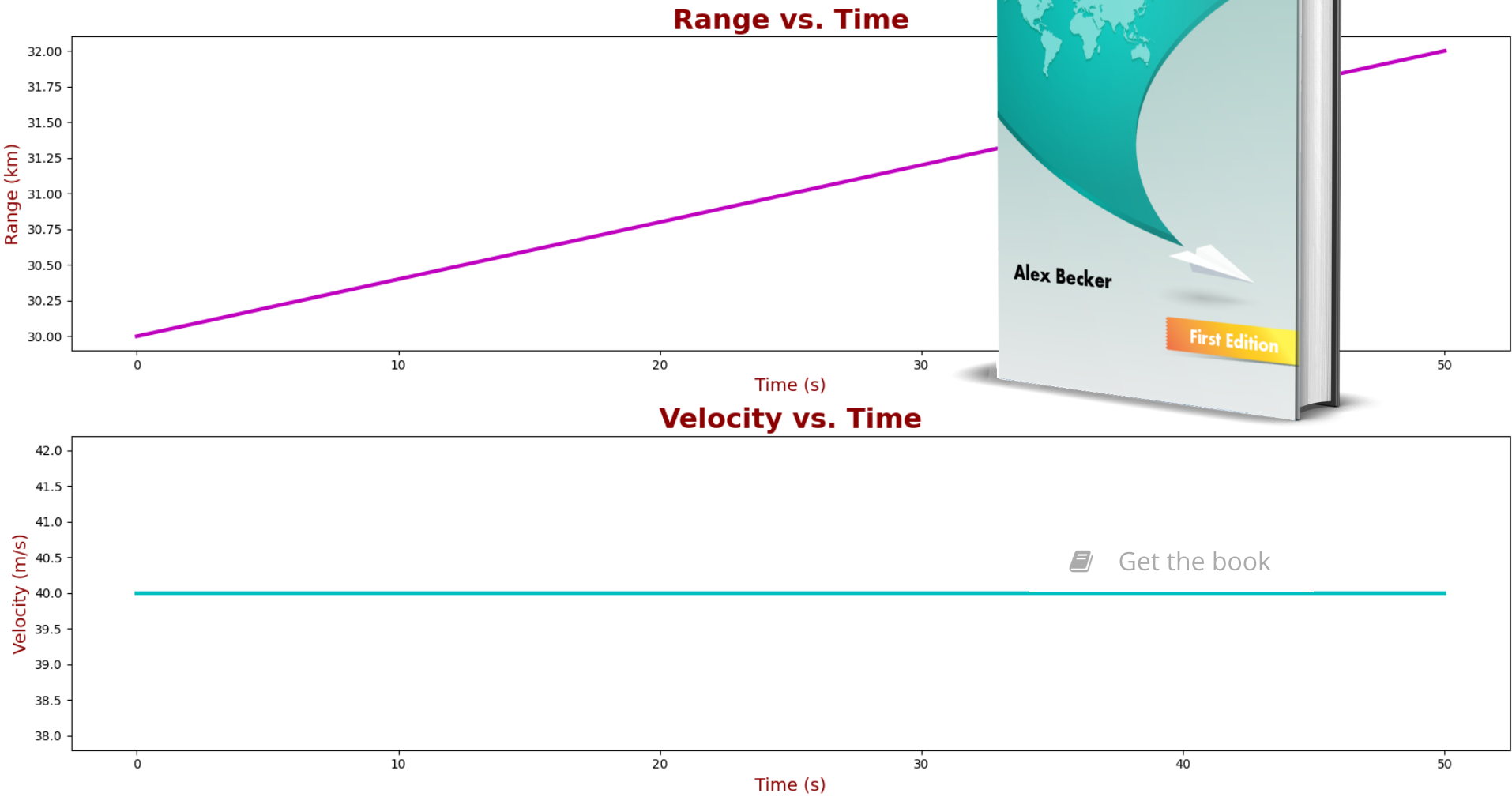
EXAMPLE SUMMARY

We've derived the  $\alpha - \beta$  filter state update equation. We've also learned the State Extrapolation Equation. We've developed an estimation algorithm for a one-dimensional dynamic system based on the  $\alpha - \beta$  filter and solved a numerical example for a constant velocity target.

EXAMPLE 3 – TRACKING ACCELERATING AIRCRAFT IN ONE

In this example, we track an aircraft moving with constant acceleration with the  $\alpha - \beta$  filter.

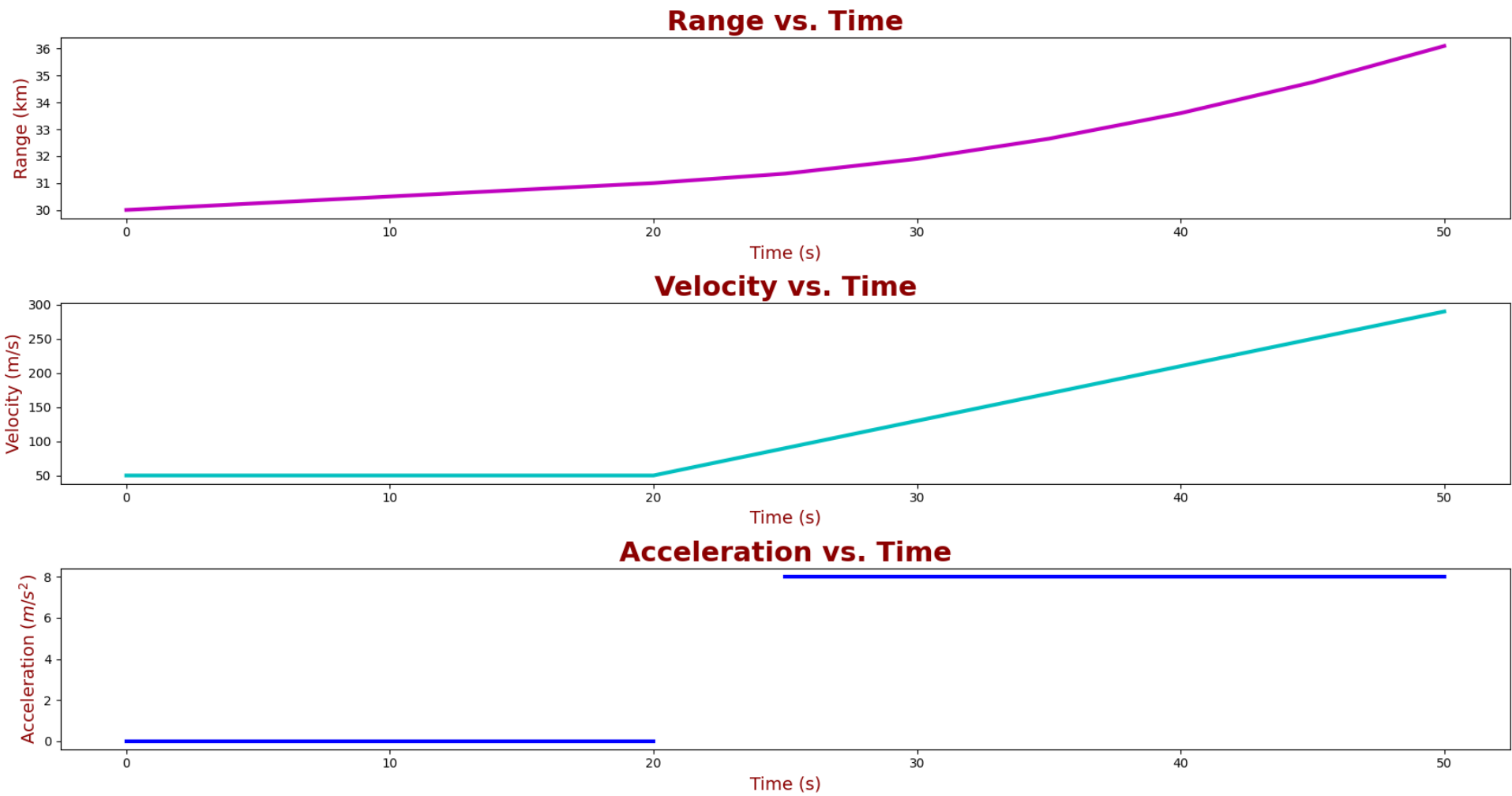
In the previous example, we tracked a UAV moving at a constant velocity of 40m/s. The following chart depicts the target range and velocity vs. time.



As you can see, the range function is linear.

Now let's examine a fighter aircraft. This aircraft moves at a constant velocity of 50m/s for 20 seconds. Then the aircraft accelerates with a constant acceleration of 8m/s<sup>2</sup> for another 35 seconds.

The following chart depicts the target range, velocity and acceleration vs. time.



As you can see from the chart, the aircraft velocity is constant for the first 20 seconds and then grows linearly. The range grows linearly for the first 20 seconds and then grows quadratically.

THE NUMERICAL EXAMPLE

Consider an aircraft moving radially toward (or away from) a radar in a one-dimensional world.

The  $\alpha - \beta$  filter parameters are:

- $\alpha = 0.2$
- $\beta = 0.1$

The track-to-track interval is 5 seconds.

ITERATION ZERO

INITIALIZATION

The initial conditions for the time  $n = 0$  are given:

$$\hat{x}_{0,0} = 30000m$$
$$\hat{\dot{x}}_{0,0} = 50m/s$$

Note: The Track Initiation (or how we get the initial conditions) is an important topic that will be discussed later. Right now, our goal is to understand the basic  $\alpha - \beta$  filter operation, so let's assume that the initial conditions are given by somebody else.

PREDICTION

The initial guess shall be extrapolated to the first cycle using the State Extrapolation Equations:

$$\hat{x}_{n+1,n} = \hat{x}_{n,n} + \Delta t \hat{\dot{x}}_{n,n} \rightarrow \hat{x}_{1,0} = \hat{x}_{0,0} + \Delta t \hat{\dot{x}}_{0,0} = 30000 + 5 \times 50 = 30250m$$
$$\hat{\dot{x}}_{n+1,n} = \hat{\dot{x}}_{n,n} \rightarrow \hat{\dot{x}}_{1,0} = \hat{\dot{x}}_{0,0} = 50m/s$$

ITERATIONS 1-10

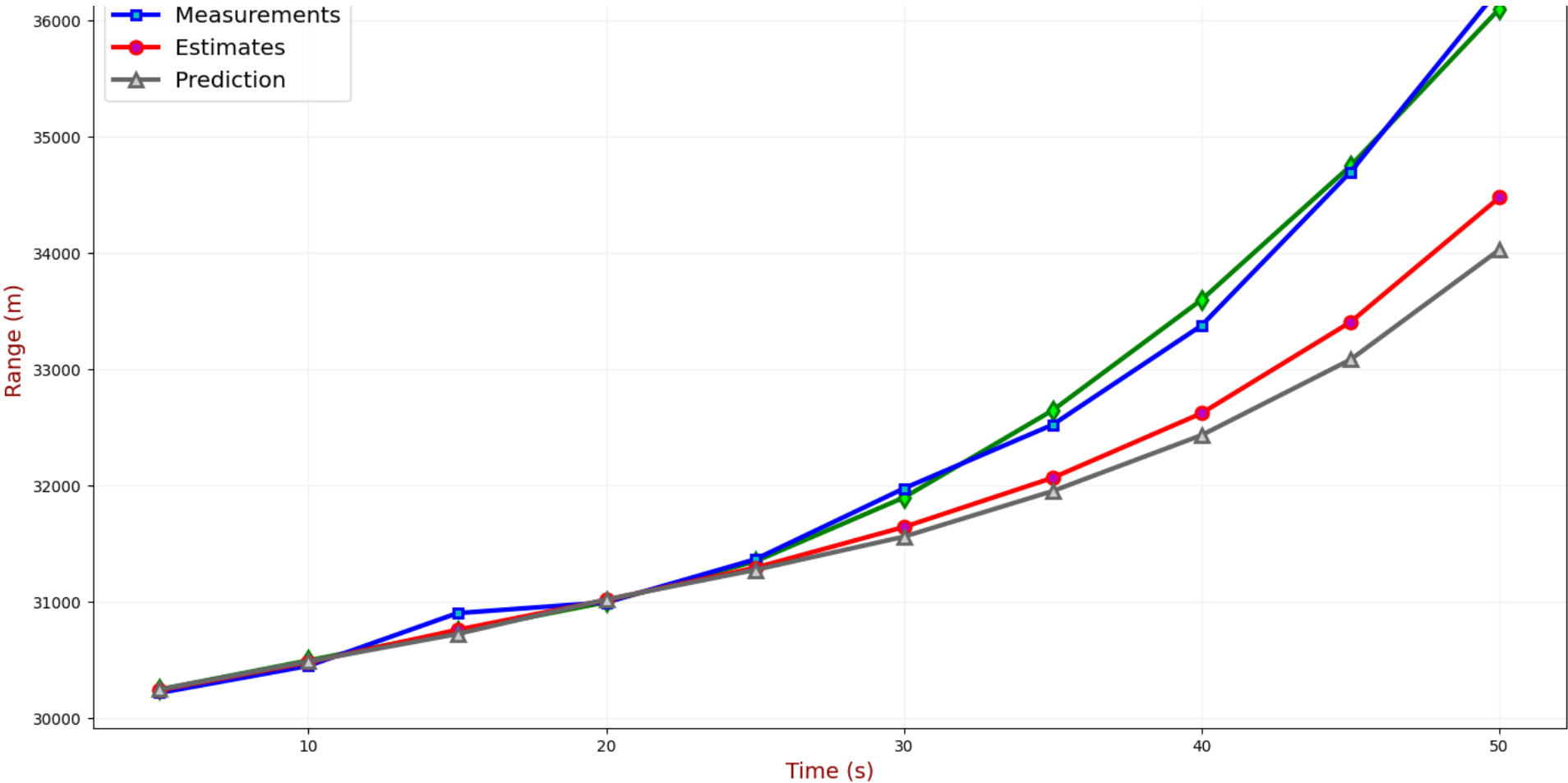
All filter iterations are summarized in the following table:

$n$	$z_n$	Current state estimates ( $\hat{x}_{n,n}, \hat{\dot{x}}_{n,n}$ )	Prediction ( $\hat{x}_{n+1,n}, \hat{\dot{x}}_{n+1,n}$ )
1	30221m	<div><math display="block">\hat{x}_{1,1} = 30250 + 0.2 (30221 - 30250) = 30244.2m</math><math display="block">\hat{\dot{x}}_{1,1} = 50 + 0.1 \left( \frac{30221 - 30250}{5} \right) = 49.42m/s</math></div>	<div><math display="block">\hat{x}_{2,1} = 30244.2 + 5 \times 49.42 = 30491.3m</math><math display="block">\hat{\dot{x}}_{2,1} = 49.42m/s</math></div>
2	30453m	<div><math display="block">\hat{x}_{2,2} = 30491.3 + 0.2 (30453 - 30491.3) = 30483.64m</math><math display="block">\hat{\dot{x}}_{2,2} = 49.42 + 0.1 \left( \frac{30453 - 30491.3}{5} \right) = 48.65m/s</math></div>	<div><math display="block">\hat{x}_{3,2} = 30483.64 + 5 \times 48.65 = 30726.9m</math><math display="block">\hat{\dot{x}}_{3,2} = 48.65m/s</math></div>
3	30906m	<div><math display="block">\hat{x}_{3,3} = 30726.9 + 0.2 (30906 - 30726.9) = 30762.7m</math><math display="block">\hat{\dot{x}}_{3,3} = 48.65 + 0.1 \left( \frac{30906 - 30726.9}{5} \right) = 52.24m/s</math></div>	<div><math display="block">\hat{x}_{4,3} = 30762.7 + 5 \times 52.24 = 31023.9m</math><math display="block">\hat{\dot{x}}_{4,3} = 52.24m/s</math></div>

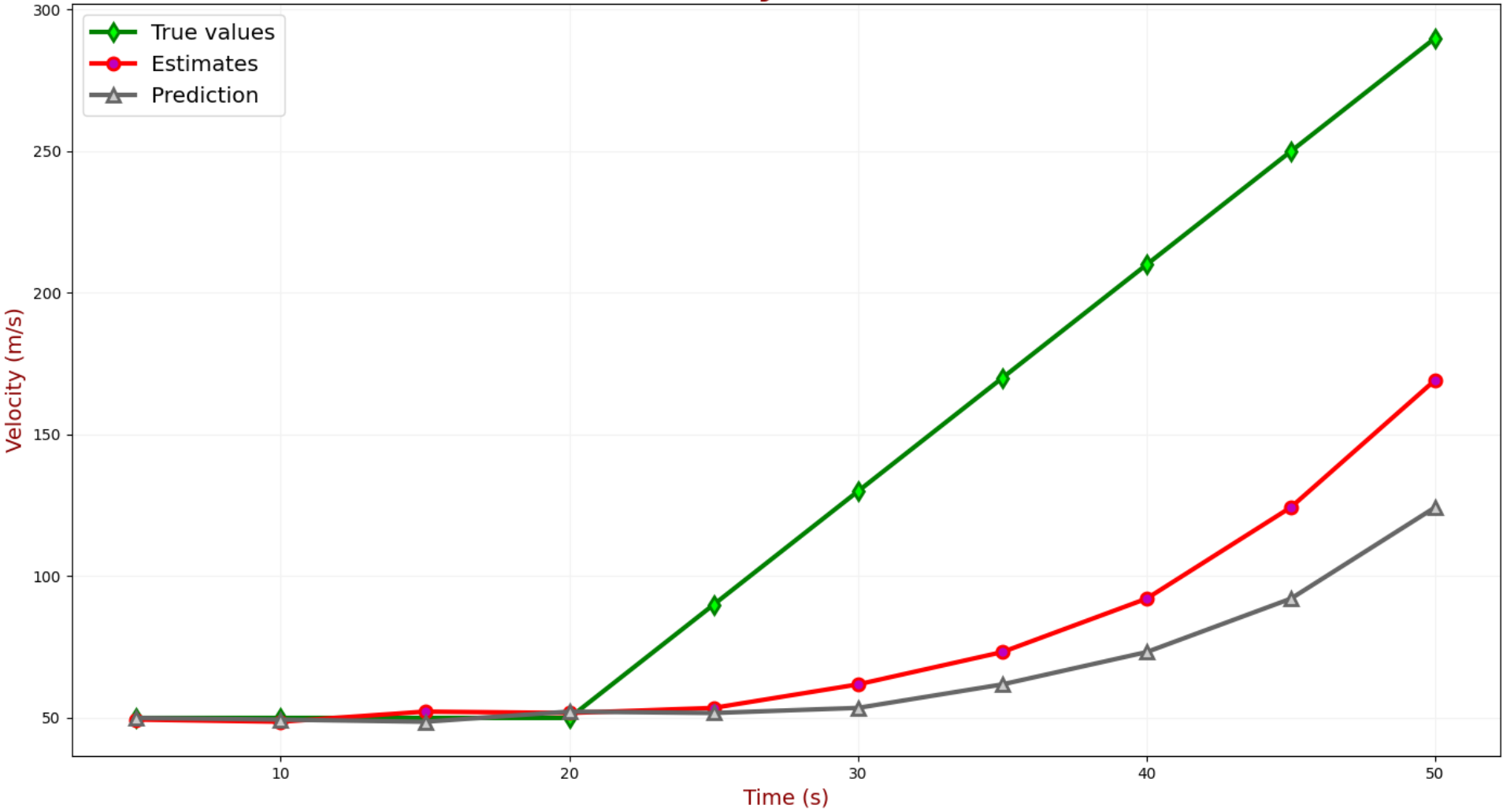
4	30999m	$\hat{x}_{4,4} = 31023.9 + 0.2 (30999 - 31023.9) = 31018.93m$ $\hat{\dot{x}}_{4,4} = 52.24 + 0.1 \left( \frac{30999 - 31023.9}{5} \right) = 51.74m/s$	$\hat{x}_{5,4} = 31018.93 + 5 \times 51.74 = 31277.6m$ $\hat{\dot{x}}_{5,4} = 51.74m/s$
5	31368m	$\hat{x}_{5,5} = 31277.6 + 0.2 (31368 - 31277.6) = 31295.7m$ $\hat{\dot{x}}_{5,5} = 51.74 + 0.1 \left( \frac{31368 - 31277.6}{5} \right) = 53.55m/s$	$\hat{x}_{6,5} = 31295.7 + 5 \times 53.55 = 31563.4m$ $\hat{\dot{x}}_{6,5} = 53.55m/s$
6	31978m	$\hat{x}_{6,6} = 31563.4 + 0.2 (31978 - 31563.4) = 31646.3m$ $\hat{\dot{x}}_{6,6} = 53.55 + 0.1 \left( \frac{31978 - 31563.4}{5} \right) = 61.84m/s$	$\hat{x}_{7,6} = 31646.3 + 5 \times 61.84 = 31955.5m$ $\hat{\dot{x}}_{7,6} = 61.84m/s$
7	32526m	$\hat{x}_{7,7} = 31955.5 + 0.2 (32526 - 31955.5) = 32069.6m$ $\hat{\dot{x}}_{7,7} = 61.84 + 0.1 \left( \frac{32526 - 31955.5}{5} \right) = 73.25m/s$	$\hat{x}_{8,7} = 32069.6 + 5 \times 73.25 = 32435.85m$ $\hat{\dot{x}}_{8,7} = 73.25m/s$
8	33379m	$\hat{x}_{8,8} = 32435.85 + 0.2 (33379 - 32435.85) = 32624.5m$ $\hat{\dot{x}}_{8,8} = 73.25 + 0.1 \left( \frac{33379 - 32435.85}{5} \right) = 92.1m/s$	$\hat{x}_{9,8} = 32624.5 + 5 \times 92.1 = 33085m$ $\hat{\dot{x}}_{9,8} = 92.1m/s$
9	34698m	$\hat{x}_{9,9} = 33085 + 0.2 (34698 - 33085) = 33407.6m$ $\hat{\dot{x}}_{9,9} = 92.1 + 0.1 \left( \frac{34698 - 33085}{5} \right) = 124.37m/s$	$\hat{x}_{10,9} = 33407.6 + 5 \times 124.37 = 34029.5m$ $\hat{\dot{x}}_{10,9} = 124.37m/s$
10	36275m	$\hat{x}_{10,10} = 34029.5 + 0.2 (36275 - 34029.5) = 34478.6m$ $\hat{\dot{x}}_{10,10} = 124.37 + 0.1 \left( \frac{36275 - 34029.5}{5} \right) = 169.28m/s$	$\hat{x}_{11,10} = 34478.6 + 5 \times 169.28 = 35325m$ $\hat{\dot{x}}_{11,10} = 169.28m/s$

RESULTS ANALYSIS

The following charts compare the true, measured, and estimated values for the range and velocity for the first 75 seconds.



Velocity vs. time



You can see a constant gap between true or measured values and estimates. The gap is called a **lag error**. Other common names for the lag error are:

- Dynamic error
- Systematic error
- Bias error
- Truncation error

The lag error appears during the acceleration period. After the acceleration period, the filter closes the gap and converges toward the true value. However, a significant lag error can result in the target loss. The lag error is unacceptable in certain applications, such as missile guidance or air defense.

EXAMPLE SUMMARY

We've examined the lag error caused by target acceleration.

# EXAMPLE 4 – TRACKING ACCELERATING AIRCRAFT WITH THE $\alpha - \beta - \gamma$ FILTER

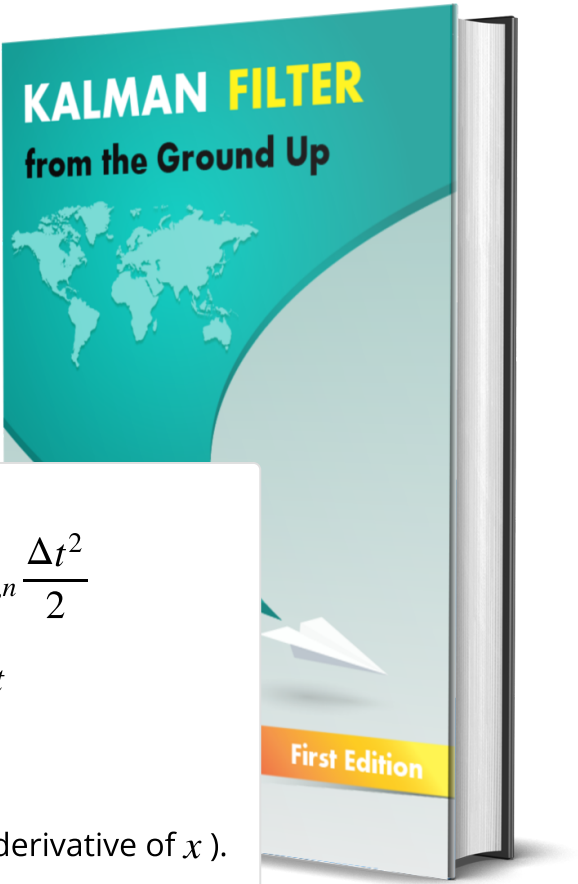
In this example, we track an aircraft using the  $\alpha - \beta - \gamma$  filter. The aircraft is moving with a constant acceleration.

## THE $\alpha - \beta - \gamma$ FILTER

The  $\alpha - \beta - \gamma$  filter (sometimes called **g-h-k filter**) considers a target acceleration. Thus, the State Extrapolation Equations become:

$$\hat{x}_{n+1,n} = \hat{x}_{n,n} + \hat{\dot{x}}_{n,n} \Delta t + \hat{\ddot{x}}_{n,n} \frac{\Delta t^2}{2}$$
$$\hat{\dot{x}}_{n+1,n} = \hat{\dot{x}}_{n,n} + \hat{\ddot{x}}_{n,n} \Delta t$$
$$\hat{\ddot{x}}_{n+1,n} = \hat{\ddot{x}}_{n,n}$$

Where  $\ddot{x}_n$  is acceleration (the second derivative of  $x$  ).



The State Update Equations become:

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + \alpha \left( z_n - \hat{x}_{n,n-1} \right)$$
$$\hat{\dot{x}}_{n,n} = \hat{\dot{x}}_{n,n-1} + \beta \left( \frac{z_n - \hat{x}_{n,n-1}}{\Delta t} \right)$$
$$\hat{\ddot{x}}_{n,n} = \hat{\ddot{x}}_{n,n-1} + \gamma \left( \frac{z_n - \hat{x}_{n,n-1}}{0.5 \Delta t^2} \right)$$

 [Get the book](#)

## THE NUMERICAL EXAMPLE

Let's take the scenario from the previous example: an aircraft that moves with a constant velocity of 50m/s for 20 seconds and then accelerates with a constant acceleration of 8m/s<sup>2</sup> for another 35 seconds.

The  $\alpha - \beta - \gamma$  filter parameters are:

- $\alpha = 0.5$ .
- $\beta = 0.4$ .
- $\gamma = 0.1$ .

The track-to-track interval is 5 seconds.

precise in real life, and the targets can be much faster.

ITERATION ZERO

INITIALIZATION

The initial conditions for the time  $n = 0$  are given:

$$\begin{aligned}\hat{x}_{0,0} &= 30000m \\ \hat{\dot{x}}_{0,0} &= 50m/s \\ \hat{\ddot{x}}_{0,0} &= 0m/s^2\end{aligned}$$

PREDICTION

The initial guess should be extrapolated to the first cycle using the State Extrapolation Equations:

$$\begin{aligned}\hat{x}_{n+1,n} &= \hat{x}_{n,n} + \hat{\dot{x}}_{n,n}\Delta t + \hat{\ddot{x}}_{n,n}\frac{\Delta t^2}{2} \rightarrow \hat{x}_{1,0} = \hat{x}_{0,0} + \hat{\dot{x}}_{0,0}\Delta t + \hat{\ddot{x}}_{0,0}\frac{\Delta t^2}{2} = 30000 + 50 \times 5 + 0 \times \frac{5^2}{2} = 30250m \\ \hat{\dot{x}}_{n+1,n} &= \hat{\dot{x}}_{n,n} + \hat{\ddot{x}}_{n,n}\Delta t \rightarrow \hat{\dot{x}}_{1,0} = \hat{\dot{x}}_{0,0} + \hat{\ddot{x}}_{0,0}\Delta t = 50 + 0 \times 5 = 50m/s \\ \hat{\ddot{x}}_{n+1,n} &= \hat{\ddot{x}}_{n,n} \rightarrow \hat{\ddot{x}}_{1,0} = \hat{\ddot{x}}_{0,0} = 0m/s^2\end{aligned}$$

FIRST ITERATION

In the first cycle (  $n = 1$  ), the initial guess is the prior estimate:

$$\begin{aligned}\hat{x}_{n,n-1} &= \hat{x}_{1,0} = 30250m \\ \hat{\dot{x}}_{n,n-1} &= \hat{\dot{x}}_{1,0} = 50m/s \\ \hat{\ddot{x}}_{n,n-1} &= \hat{\ddot{x}}_{1,0} = 0m/s^2\end{aligned}$$

STEP 1

The radar measures the aircraft range:

$$z_1 = 30221m$$

STEP 2

Calculating the current estimate using the State Update Equation:

$$\begin{aligned}\hat{x}_{1,1} &= \hat{x}_{1,0} + \alpha \left( z_1 - \hat{x}_{1,0} \right) = 30250 + 0.5 \left( 30221 - 30250 \right) = 30235.5m \\ \hat{\dot{x}}_{1,1} &= \hat{\dot{x}}_{1,0} + \beta \left( \frac{z_1 - \hat{x}_{1,0}}{\Delta t} \right) = 50 + 0.4 \left( \frac{30221 - 30250}{5} \right) = 47.68m/s \\ \hat{\ddot{x}}_{1,1} &= \hat{\ddot{x}}_{1,0} + \gamma \left( \frac{z_1 - \hat{x}_{1,0}}{0.5\Delta t^2} \right) = 0 + 0.1 \left( \frac{30221 - 30250}{0.5 \times 5^2} \right) = -0.23m/s^2\end{aligned}$$

STEP 3

Calculating the next state estimate using the State Extrapolation Equations:

$$\hat{x}_{2,1} = \hat{x}_{1,1} + \hat{\dot{x}}_{1,1}\Delta t + \hat{\ddot{x}}_{1,1}\frac{\Delta t^2}{2} = 30235.5 + 47.68 \times 5 + (-0.23) \times \frac{5^2}{2} = 30471m$$



$$\hat{x}_{2,1} = \hat{x}_{1,1} = -0.23m/s$$

SECOND ITERATION

After a unit time delay, the predicted estimate from the foregoing iteration becomes the prior estimate in the current iteration.

$$\hat{x}_{2,1} = 30471m$$

$$\hat{\dot{x}}_{2,1} = 46.52m/s$$

$$\hat{\ddot{x}}_{2,1} = -0.23m/s^2$$

STEP 1

The radar measures the aircraft range:

$$z_2 = 30453m$$

STEP 2

Calculating the current estimate using the State Update Equation:

$$\hat{x}_{2,2} = \hat{x}_{2,1} + \alpha \left( z_2 - \hat{x}_{2,1} \right) = 30471 + 0.5 \left( 30453 - 30471 \right) = 30462m$$

$$\hat{\dot{x}}_{2,2} = \hat{\dot{x}}_{2,1} + \beta \left( \frac{z_2 - \hat{x}_{2,1}}{\Delta t} \right) = 46.52 + 0.4 \left( \frac{30453 - 30471}{5} \right) = 45.08m/s$$

$$\hat{\ddot{x}}_{2,2} = \hat{\ddot{x}}_{2,1} + \gamma \left( \frac{z_2 - \hat{x}_{2,1}}{0.5\Delta t^2} \right) = -0.23 + 0.1 \left( \frac{30453 - 30471}{0.5 \times 5^2} \right) = -0.38m/s^2$$

STEP 3

Calculating the next state estimate using the State Extrapolation Equations:

$$\hat{x}_{3,2} = \hat{x}_{2,2} + \hat{\dot{x}}_{2,2}\Delta t + \hat{\ddot{x}}_{2,2}\frac{\Delta t^2}{2} = 30462 + 45.08 \times 5 + (-0.38) \times \frac{5^2}{2} = 30682.7m$$

$$\hat{\dot{x}}_{3,2} = \hat{\dot{x}}_{2,2} + \hat{\ddot{x}}_{2,2}\Delta t = 45.08 + (-0.38) \times 5 = 43.2m/s$$

$$\hat{\ddot{x}}_{3,2} = \hat{\ddot{x}}_{2,2} = -0.38m/s^2$$

ITERATIONS 3-10

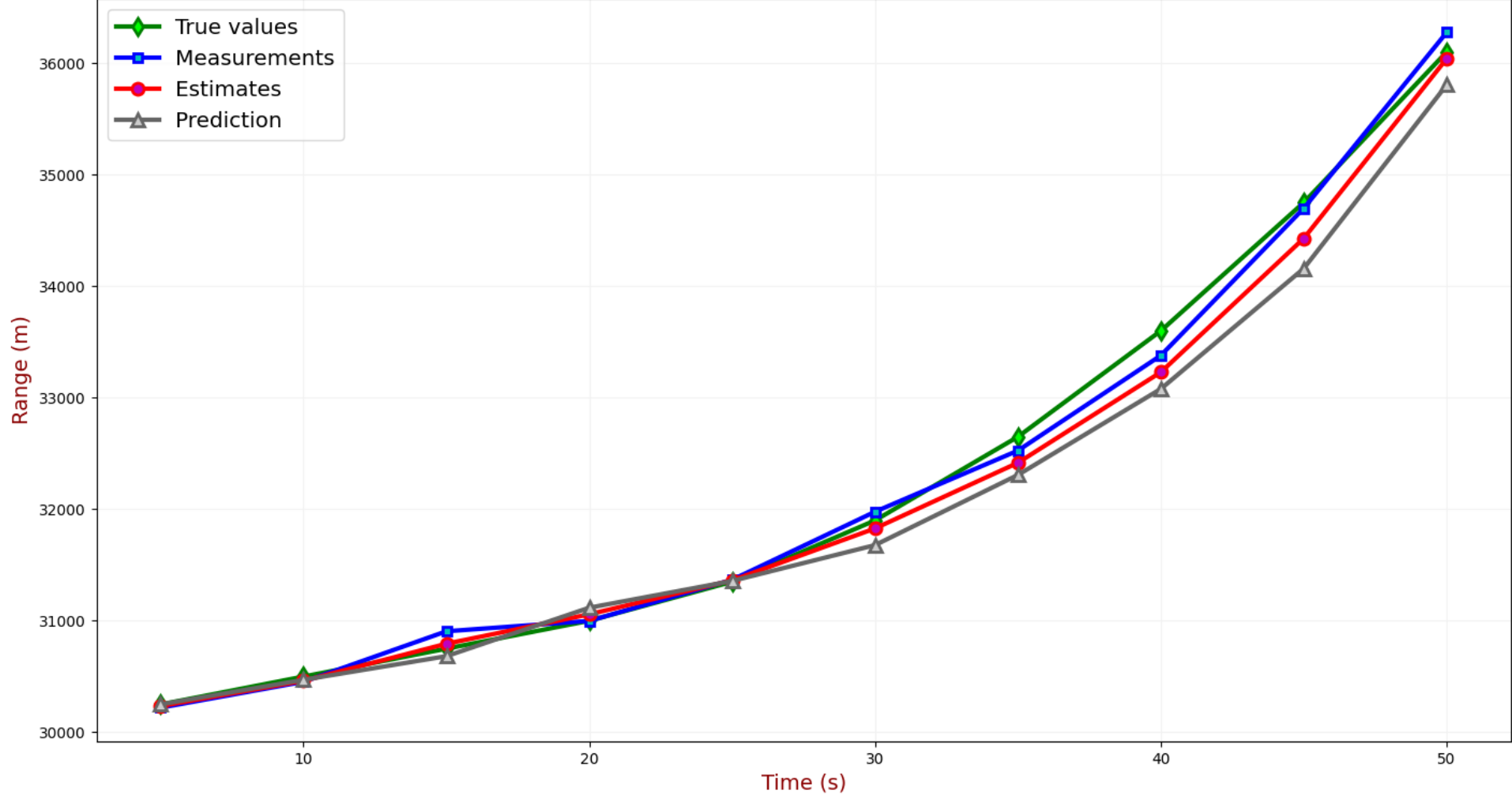
The calculations for the successive iterations are summarized in the following table:

<i>n</i>	<i>z<sub>n</sub></i>	Current state estimates ( <i>ĥ<sub>n,n</sub></i> , <i>ĥ̇<sub>n,n</sub></i> , <i>ĥ̈<sub>n,n</sub></i> )	Prediction ( <i>ĥ<sub>n+1,n</sub></i> , <i>ĥ̇<sub>n+1,n</sub></i> , <i>ĥ̈<sub>n+1,n</sub></i> )
3	30906 <i>m</i>	<div><math display="block">\hat{x}_{3,3} = 30682.7 + 0.5 \left( 30906 - 30682.7 \right) = 30794.35m</math><math display="block">\hat{\dot{x}}_{3,3} = 43.2 + 0.4 \left( \frac{30906 - 30682.7}{5} \right) = 61.06m/s</math><math display="block">\hat{\ddot{x}}_{3,3} = -0.38 + 0.1 \left( \frac{30906 - 30682.7}{0.5 \times 5^2} \right) = 1.41m/s^2</math></div>	<div><math display="block">\hat{x}_{4,3} = 30794.35 + 5 \times 61.06 + 1.41 \times \frac{5^2}{2} =</math><math display="block">\hat{\dot{x}}_{4,3} = 61.06 + 1.41 \times 5 = 68.1m/s</math><math display="block">\hat{\ddot{x}}_{4,3} = 1.41m/s^2</math></div>

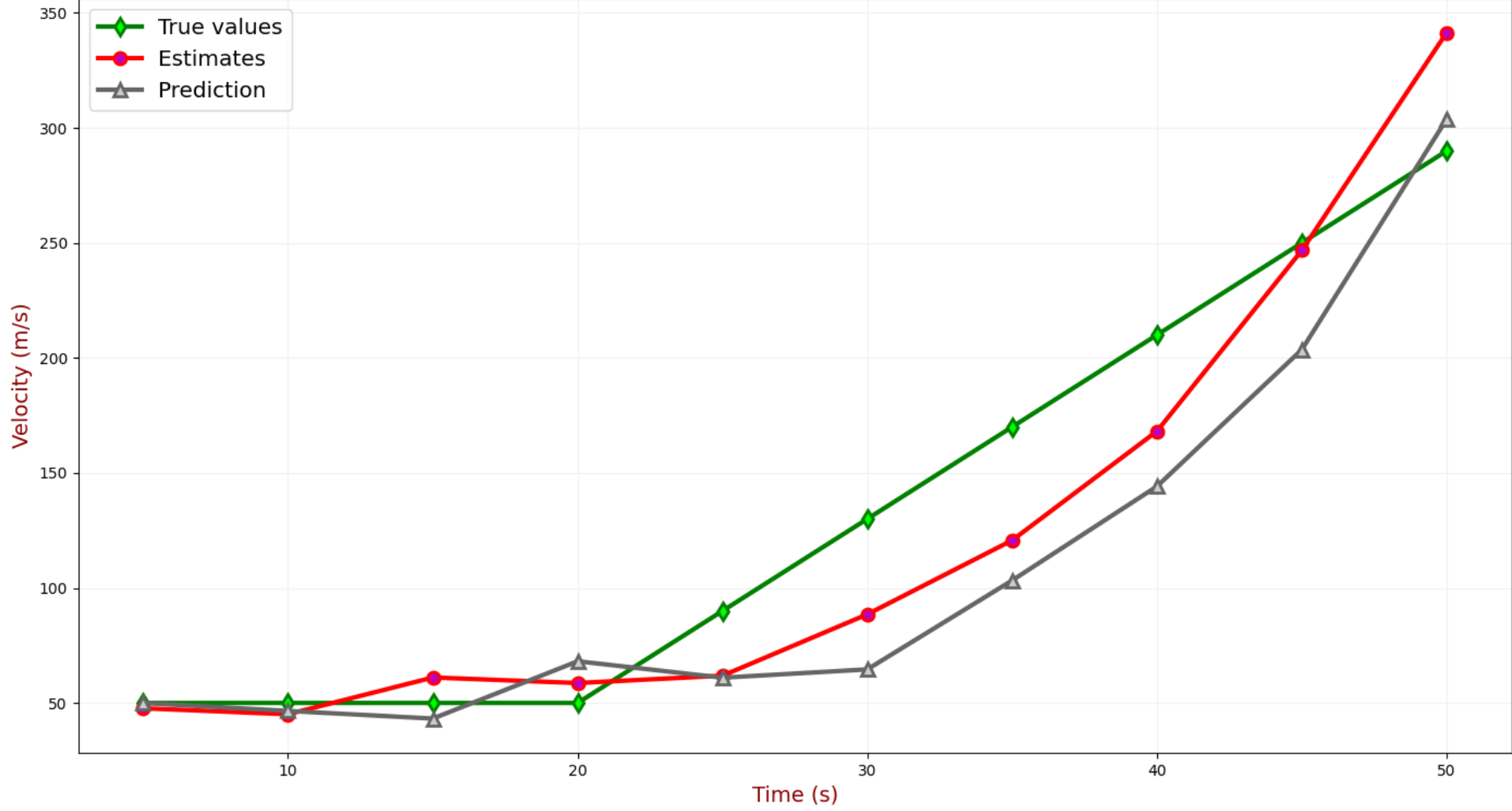
4	30999m	$\hat{x}_{4,4} = 31117.3 + 0.5 (30999 - 31117.3) = 31058.15m$ $\hat{\dot{x}}_{4,4} = 68.1 + 0.4 \left( \frac{30999 - 31117.3}{5} \right) = 58.65m/s$ $\hat{\ddot{x}}_{4,4} = 1.41 + 0.1 \left( \frac{30999 - 31117.3}{0.5 \times 5^2} \right) = 0.46m/s^2$	$\hat{x}_{5,4} = 31058.15 + 5 \times 58.65 + 0.46 \times \frac{5^2}{2} =$ $\hat{\dot{x}}_{5,4} = 58.65 + 0.46 \times 5 = 61m/s$ $\hat{\ddot{x}}_{5,4} = 0.46m/s^2$
5	31368m	$\hat{x}_{5,5} = 31357.2 + 0.5 (31368 - 31357.2) = 31362.6m$ $\hat{\dot{x}}_{5,5} = 61 + 0.4 \left( \frac{31368 - 31357.2}{5} \right) = 61.8m/s$ $\hat{\ddot{x}}_{5,5} = 0.46 + 0.1 \left( \frac{31368 - 31357.2}{0.5 \times 5^2} \right) = 0.55m/s^2$	$\hat{x}_{6,5} = 31362.6 + 5 \times 61.8 + 0.55 \times \frac{5^2}{2} =$ $\hat{\dot{x}}_{6,5} = 61.8 + 0.55 \times 5 = 64.6m/s$ $\hat{\ddot{x}}_{6,5} = 0.55m/s^2$
6	31978m	$\hat{x}_{6,6} = 31678.7 + 0.5 (31978 - 31678.7) = 31828.3m$ $\hat{\dot{x}}_{6,6} = 64.6 + 0.4 \left( \frac{31978 - 31678.7}{5} \right) = 88.5m/s$ $\hat{\ddot{x}}_{6,6} = 0.55 + 0.1 \left( \frac{31978 - 31678.7}{0.5 \times 5^2} \right) = 2.95m/s^2$	$\hat{x}_{7,6} = 31828.3 + 5 \times 88.5 + 2.95 \times \frac{5^2}{2} =$ $\hat{\dot{x}}_{7,6} = 88.5 + 2.95 \times 5 = 103.26m$ $\hat{\ddot{x}}_{7,6} = 2.95m/s^2$
7	32526m	$\hat{x}_{7,7} = 32307.8 + 0.5 (32526 - 32307.8) = 32416.9m$ $\hat{\dot{x}}_{7,7} = 103.26 + 0.4 \left( \frac{32526 - 32307.8}{5} \right) = 120.7m/s$ $\hat{\ddot{x}}_{7,7} = 2.95 + 0.1 \left( \frac{32526 - 32307.8}{0.5 \times 5^2} \right) = 4.7m/s^2$	$\hat{x}_{8,7} = 32416.9 + 5 \times 120.7 + 4.7 \times \frac{5^2}{2} =$ $\hat{\dot{x}}_{8,7} = 120.7 + 4.7 \times 5 = 144.17m$ $\hat{\ddot{x}}_{8,7} = 4.7m/s^2$
8	33379m	$\hat{x}_{8,8} = 33079.1 + 0.5 (33379 - 33079.1) = 33229.05m$ $\hat{\dot{x}}_{8,8} = 144.17 + 0.4 \left( \frac{33379 - 33079.1}{5} \right) = 168.2m/s$ $\hat{\ddot{x}}_{8,8} = 4.7 + 0.1 \left( \frac{33379 - 33079.1}{0.5 \times 5^2} \right) = 7.1m/s^2$	$\hat{x}_{9,8} = 33229.05 + 5 \times 168.2 + 7.1 \times \frac{5^2}{2} =$ $\hat{\dot{x}}_{9,8} = 168.2 + 7.1 \times 5 = 203.6m/s$ $\hat{\ddot{x}}_{9,8} = 7.1m/s^2$
9	34698m	$\hat{x}_{9,9} = 34158.5 + 0.5 (34698 - 34158.5) = 34428.2m$ $\hat{\dot{x}}_{9,9} = 203.6 + 0.4 \left( \frac{34698 - 34158.5}{5} \right) = 246.8m/s$ $\hat{\ddot{x}}_{9,9} = 7.1 + 0.1 \left( \frac{34698 - 34158.5}{0.5 \times 5^2} \right) = 11.4m/s^2$	$\hat{x}_{10,9} = 34428.2 + 5 \times 246.8 + 11.4 \times \frac{5^2}{2} =$ $\hat{\dot{x}}_{10,9} = 246.8 + 11.4 \times 5 = 303.8m/s$ $\hat{\ddot{x}}_{10,9} = 11.4m/s^2$
10	36275m	$\hat{x}_{10,10} = 35804.7 + 0.5 (36275 - 35804.7) = 36039.8m$ $\hat{\dot{x}}_{10,10} = 303.8 + 0.4 \left( \frac{36275 - 35804.7}{5} \right) = 341.4m/s$ $\hat{\ddot{x}}_{10,10} = 11.4 + 0.1 \left( \frac{36275 - 35804.7}{0.5 \times 5^2} \right) = 15.2m/s^2$	$\hat{x}_{11,10} = 36039.8 + 5 \times 341.4 + 15.2 \times \frac{5^2}{2} =$ $\hat{\dot{x}}_{11,10} = 341.4 + 15.2 \times 5 = 417.3m/s$ $\hat{\ddot{x}}_{11,10} = 15.2m/s^2$

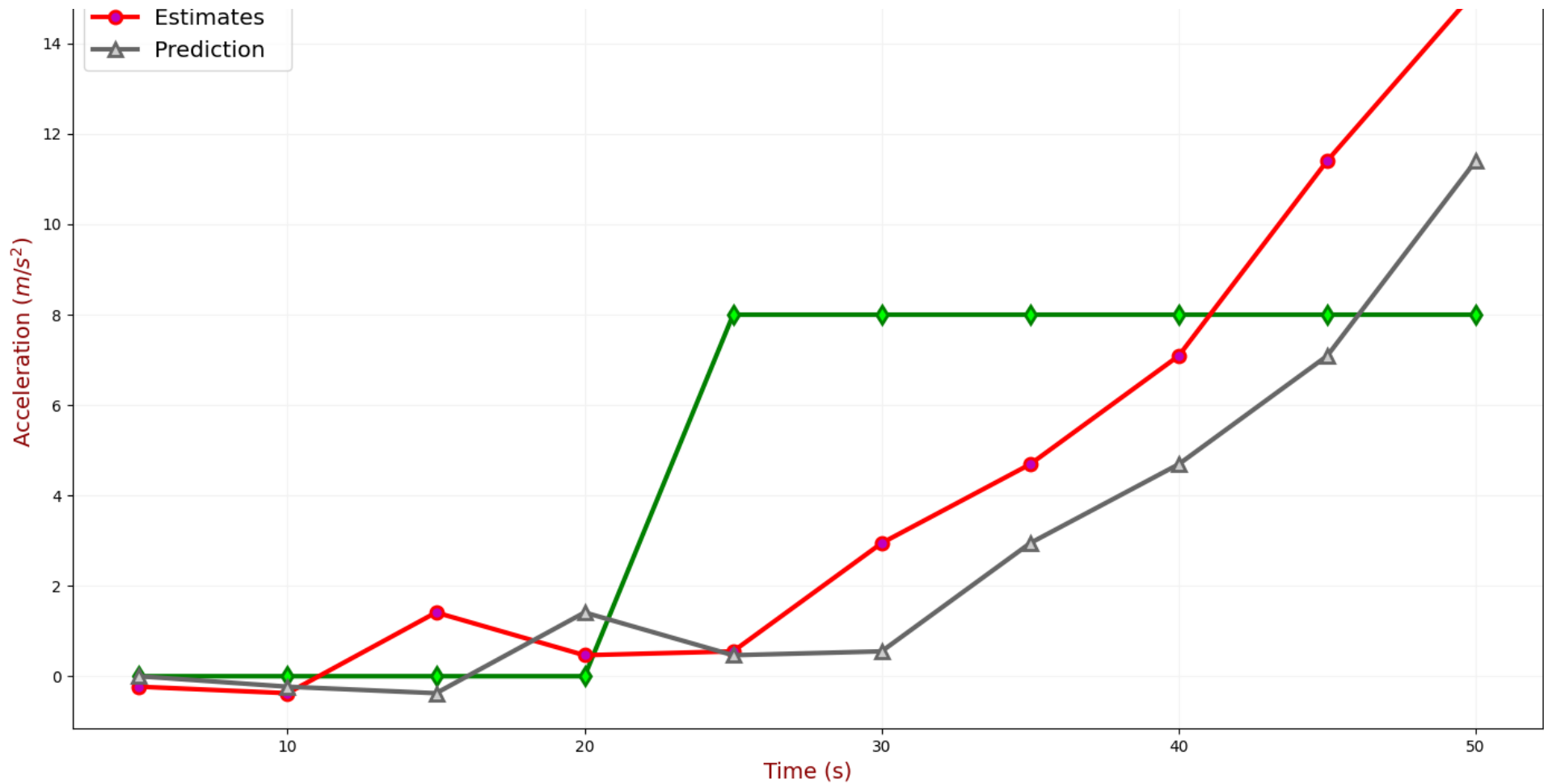
The following charts compare the true, measured, and estimated values for the range, velocity, and acceleration for the first 50 seconds.

Range vs. time



Velocity vs. time





EXAMPLE SUMMARY

As you can see, the  $\alpha - \beta - \gamma$  filter with dynamic model equations that include acceleration can track the target with constant acceleration and eliminate the lag error.

But what happens in the case of a maneuvering target? The target can suddenly change the flight direction by making a maneuver. The dynamic model of the target can also include a jerk (changing acceleration). In such cases, the  $\alpha - \beta - \gamma$  filter with constant  $\alpha - \beta - \gamma$  coefficients produces estimation errors and, in some cases, loses the target track.

The Kalman filter can handle uncertainty in the dynamic model. It is our next topic, right after the summary.

# SUMMARY OF THE $\alpha - \beta - (\gamma)$ FILTER

There are many types of  $\alpha - \beta - (\gamma)$  filters, and they are based on the same principle:

- The current state estimation is based on the state update equations.
- The following state estimation (prediction) is based on the dynamic model equations.

The main difference between these filters is the selection of weighting coefficients  $\alpha - \beta - (\gamma)$ . Some filter types use constant weighting coefficients; others compute weighting coefficients for every filter iteration (cycle).

The choice of the  $\alpha$ ,  $\beta$  and  $\gamma$  is crucial for proper functionality of the estimation algorithm.

What should be the  $\alpha - \beta - (\gamma)$  parameters?

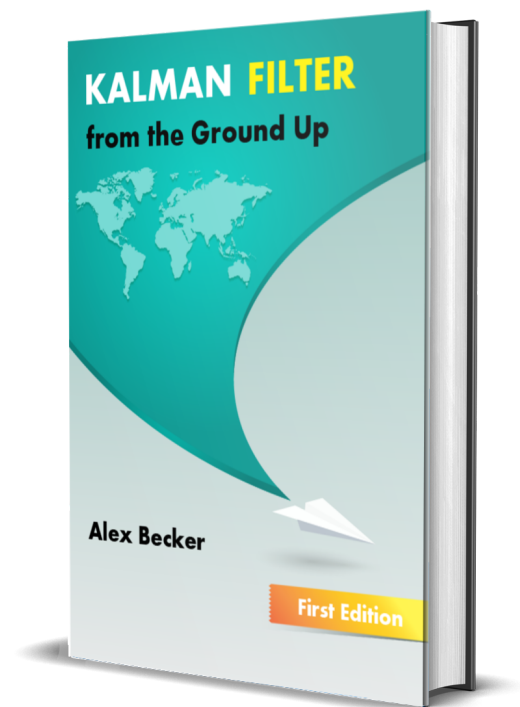
I described the  $\alpha - \beta - (\gamma)$  filter as an introductory to the Kalman Filter, thus, I won't cover this topic. The curious reader can find many books and papers on this topic. As a reference, I would recommend the following:

*Dirk Tenne, Tarunraj Singh. "Optimal Design of  $\alpha - \beta - (\gamma)$  Filters". State University of New York at Buffalo.*

Another important issue is the initiation of the filter, i.e., providing the initial value for the first filter iteration.

The following list includes the most popular  $\alpha - \beta - (\gamma)$  filters:

- Wiener Filter
- Bayes Filter
- Fading-memory polynomial Filter
- Expanding-memory (or growing-memory) polynomial Filter
- Least-squares Filter
- Benedict-Bordner Filter
- Lumped Filter
- Discounted least-squares  $\alpha - \beta$  Filter
- Critically damped  $\alpha - \beta$  Filter
- Growing-memory Filter
- Kalman Filter
- Extended Kalman Filter
- Unscented Kalman Filter
- Extended Complex Kalman Filter
- Gauss-Hermite Kalman Filter
- Cubature Kalman Filter
- Particle Filter



 [Get the book](#)

[◀ Previous](#) [Next ▶](#)

© Copyright 2023, Alex Becker