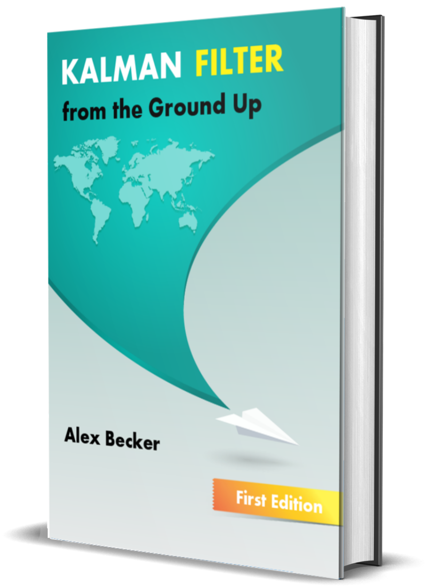
STATE EXTRAPOLATION EQUATION

The first Kalman Filter equation that I would like to describe is the **state extrapolation equation**.



Using the state extrapolation equation, we can predict the next system state based on the knowledge of the current state. It extrapolates the state vector from the present (time step n) to the future (time step $n+1$).

The state extrapolation equation describes the model of the dynamic system. In the literature, it is also called:

- Predictor Equation
- Transition Equation
- Prediction Equation
- Dynamic Model
- State Space Model

The general form of the state extrapolation equation in a matrix notation is:

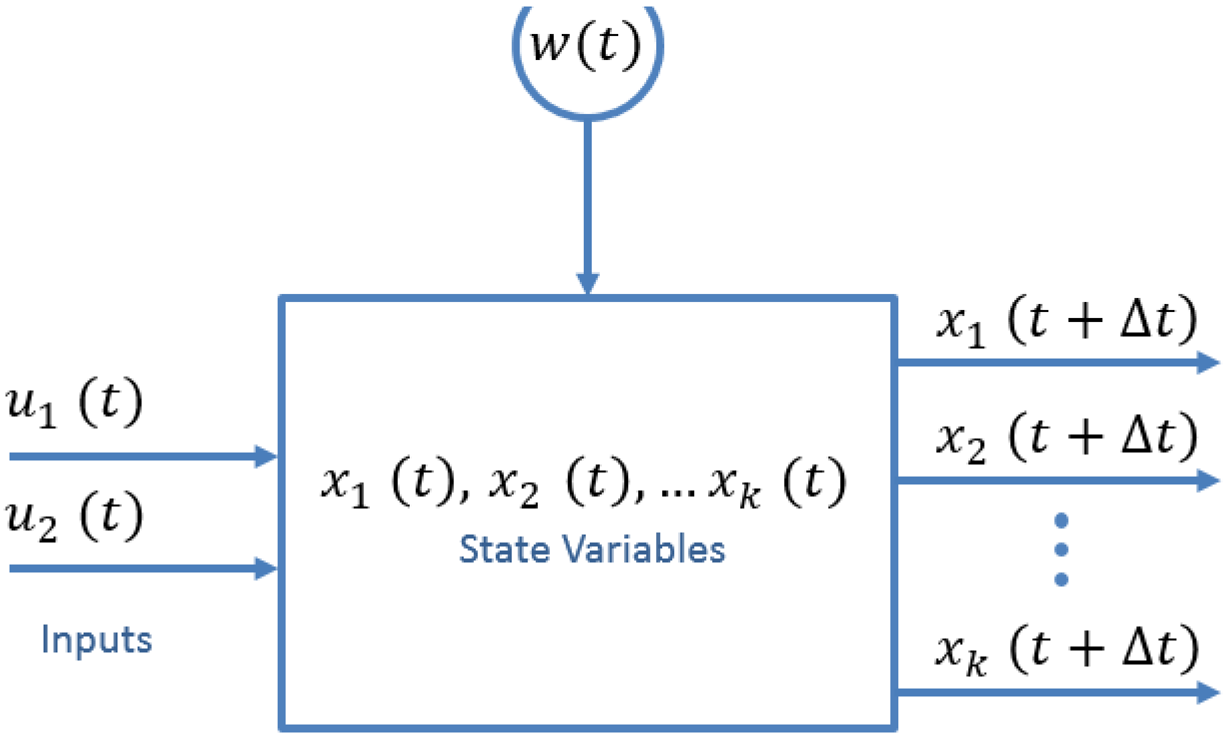
$$\boldsymbol{\hat{x}}_{n+1,n} = \boldsymbol{F}\boldsymbol{\hat{x}}_{n,n} + \boldsymbol{G}\boldsymbol{u}_{n} + \boldsymbol{w}_{n}$$

Where:

$\boldsymbol{\hat{x}}_{n+1,n}$	is a predicted system state vector at time step $n+1$
$\boldsymbol{\hat{x}}_{n,n}$	is an estimated system state vector at time step n
$\boldsymbol{u}_{n}$	is a **control variable** or **input variable** - a measurable (deterministic) input to the system
$\boldsymbol{w}_{n}$	is a **process noise** or disturbance - an unmeasurable input that affects the state
$\boldsymbol{F}$	is a **state transition matrix**
$\boldsymbol{G}$	is a **control matrix** or **input transition matrix** (mapping control to state variables)

> Note: In the literature, the state transition matrix F is sometimes denoted by a Greek letter Φ.

The following figure provides a schematic description of the state extrapolation equation.



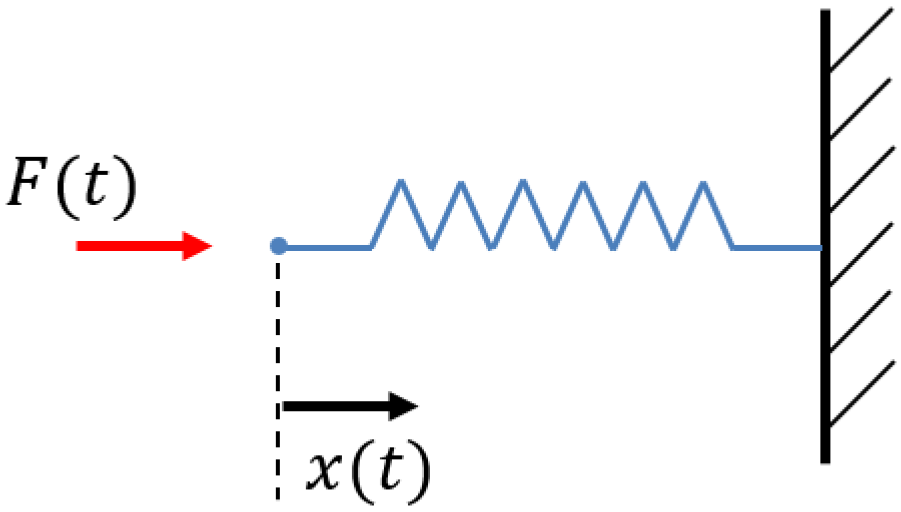
The state variables may represent attributes of the system that we wish to know.

For example, a moving vehicle has three attributes: position, velocity, and acceleration.

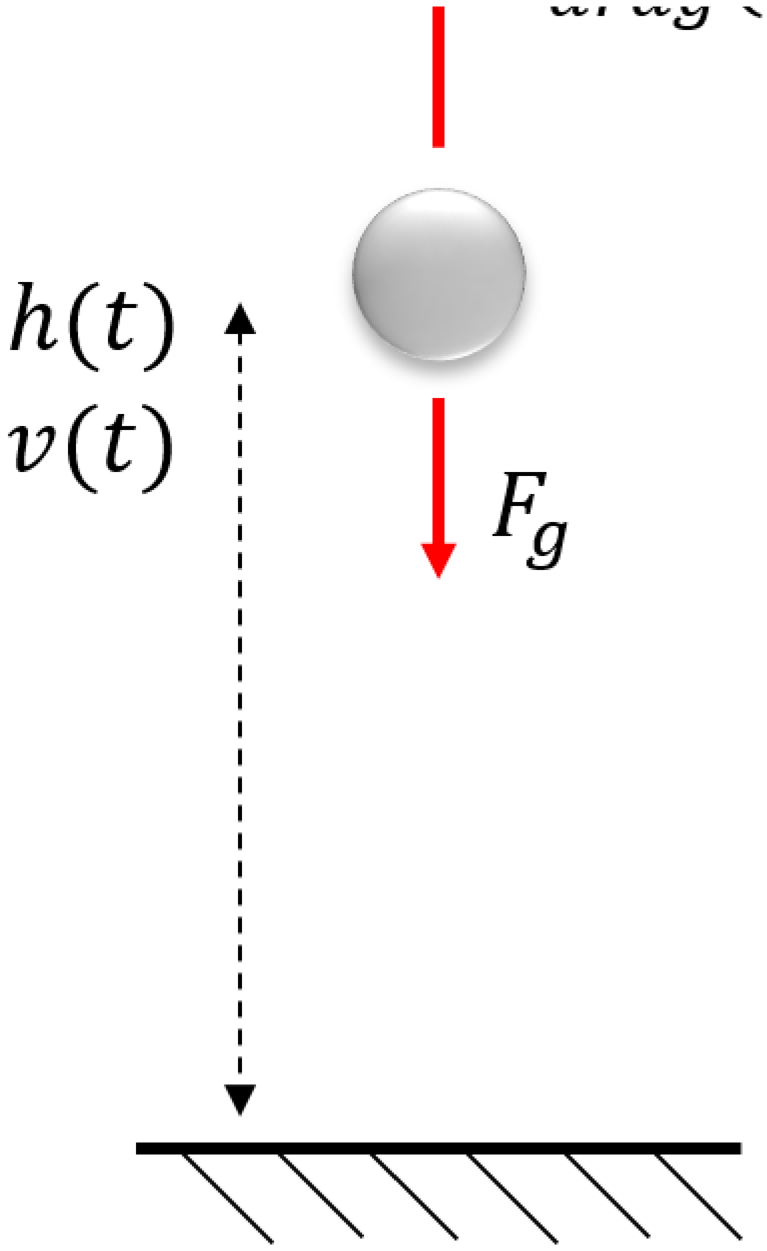
You might ask yourself, which attributes are the state variables, and which attributes are the input to the system?

- Moving mechanical systems have attributes such as position, velocity, acceleration, and drag.
- A force that acts on a system should be considered an external forcing function, i.e., an input to the system that controls the state vector (position and velocity in the constant acceleration case).
- Newton's second law tells us that $F = ma$. Thus we can consider acceleration as an external input to the system.
- The position and the velocity are the primary state variables of interest.

For instance, in a spring system, the force applied to the spring $F(t)$ is an input $u(t)$, while the spring displacement $x(t)$ is the system state.



For a falling object, the inputs are the gravitational force F_g and the drag force $F_{drag}(t)$, while the object height $h(t)$ and velocity $v(t)$ are the system states.



> Note: The process noise w_n does not typically appear directly in the equations of interest. Instead, this term is used to model the uncertainty in the Covariance Extrapolation Equation.

Let's take a look at several examples of the state extrapolation equation.

EXAMPLE - AIRPLANE - NO CONTROL INPUT

In this example, we define the State Extrapolation Equation for an airplane, assuming a constant acceleration model.

In this example, there is no control input.

We will assume no control input. We will see the control input in the next example.

$$\boldsymbol{u}_n = 0$$

Consider an airplane moving in three-dimensional space with constant acceleration. The state vector $\boldsymbol{\hat{x}_n}$ that describes the estimated airplane position, velocity, and acceleration in a cartesian coordinate system (x, y, z) is:

$$\boldsymbol{\hat{x}}_{n} = \left[\begin{matrix} \hat{y}_{n} \\ \hat{z}_{n} \\ \hat{\dot{x}}_{n} \\ \hat{\dot{y}}_{n} \\ \hat{\dot{z}}_{n} \\ \hat{\ddot{x}}_{n} \\ \hat{\ddot{y}}_{n} \\ \hat{\ddot{z}}_{n} \end{matrix} \right]$$

> Note: Don't confuse the estimated state vector $\boldsymbol{\hat{x}_{n}}$ (bold-face font) and the estimated airplane position at x - axis denoted by $\hat{x}_{n}$ (normal-face font). Both variables are denoted by the same letter in the literature, and I am trying to be consistent.

The state transition matrix $\boldsymbol{F}$ is:

$$\boldsymbol{F} = \left[\begin{matrix} 1 & 0 & 0 & \Delta t & 0 & 0 & 0.5\Delta t^{2} & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0.5\Delta t^{2} & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0.5\Delta t^{2} \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{matrix} \right]$$

The state extrapolation equation is:

$$\boldsymbol{\hat{x}}_{n+1,n} = \boldsymbol{F}\boldsymbol{\hat{x}}_{n}$$

$$\left[\begin{matrix} \hat{x}_{n+1,n} \\ \hat{y}_{n+1,n} \\ \hat{z}_{n+1,n} \\ \hat{\dot{x}}_{n+1,n} \\ \hat{\dot{y}}_{n+1,n} \\ \hat{\dot{z}}_{n+1,n} \\ \hat{\ddot{x}}_{n+1,n} \\ \hat{\ddot{y}}_{n+1,n} \\ \hat{\ddot{z}}_{n+1,n} \end{matrix} \right] = \left[\begin{matrix} 1 & 0 & 0 & \Delta t & 0 & 0 & 0.5\Delta t^{2} & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0.5\Delta t^{2} & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0.5\Delta t^{2} \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{matrix} \right] \left[\begin{matrix} \hat{x}_{n,n} \\ \hat{y}_{n,n} \\ \hat{z}_{n,n} \\ \hat{\dot{x}}_{n,n} \\ \hat{\dot{y}}_{n,n} \\ \hat{\dot{z}}_{n,n} \\ \hat{\ddot{x}}_{n,n} \\ \hat{\ddot{y}}_{n,n} \\ \hat{\ddot{z}}_{n,n} \end{matrix} \right]$$

The matrix multiplication results:

$$\begin{cases} \hat{y}_{n+1,n} = \hat{y}_{n,n} + \hat{\dot{y}}_{n,n}\Delta t + \frac{1}{2}\hat{\ddot{y}}_{n,n}\Delta t^2 \\ \hat{z}_{n+1,n} = \hat{z}_{n,n} + \hat{\dot{z}}_{n,n}\Delta t + \frac{1}{2}\hat{\ddot{z}}_{n,n}\Delta t^2 \\ \hat{\dot{x}}_{n+1,n} = \hat{\dot{x}}_{n,n} + \hat{\ddot{x}}_{n,n}\Delta t \\ \hat{\dot{y}}_{n+1,n} = \hat{\dot{y}}_{n,n} + \hat{\ddot{y}}_{n,n}\Delta t \\ \hat{\dot{z}}_{n+1,n} = \hat{\dot{z}}_{n,n} + \hat{\ddot{z}}_{n,n}\Delta t \\ \hat{\ddot{x}}_{n+1,n} = \hat{\ddot{x}}_{n,n} \\ \hat{\ddot{y}}_{n+1,n} = \hat{\ddot{y}}_{n,n} \\ \hat{\ddot{z}}_{n+1,n} = \hat{\ddot{z}}_{n,n} \end{cases}$$

EXAMPLE - AIRPLANE WITH CONTROL INPUT

This example is similar to the previous example, but now we have a sensor connected to the pilot's controls, so we have additional information about the airplane acceleration based on the pilot's commands.

The state vector $\boldsymbol{\hat{x}_n}$ that describes the estimated airplane position and velocity in a cartesian coordinate system (x, y, z) is:

$$\boldsymbol{\hat{x}}_n = \begin{bmatrix} \hat{x}_n \\ \hat{y}_n \\ \hat{z}_n \\ \hat{\dot{x}}_n \\ \hat{\dot{y}}_n \\ \hat{\dot{z}}_n \end{bmatrix}$$

The control vector $\boldsymbol{u_n}$ that describes the measured airplane acceleration in a cartesian coordinate system (x, y, z) is:

$$\boldsymbol{u}_n = \begin{bmatrix} \ddot{x}_n \\ \ddot{y}_n \\ \ddot{z}_n \end{bmatrix}$$

The state transition matrix $\boldsymbol{F}$ is:

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

The control matrix $\boldsymbol{G}$ is:

$$G = \begin{bmatrix} 0 & 0.5\Delta t^2 & 0 \\ 0 & 0 & 0.5\Delta t^2 \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix}$$

The state extrapolation equation is:

$$\boldsymbol{\hat{x}}_{n+1,n} = \boldsymbol{F\hat{x}}_{n,n} + \boldsymbol{Gu}_{n,n}$$

$$\begin{bmatrix} \hat{x}_{n+1,n} \\ \hat{y}_{n+1,n} \\ \hat{z}_{n+1,n} \\ \hat{\dot{x}}_{n+1,n} \\ \hat{\dot{y}}_{n+1,n} \\ \hat{\dot{z}}_{n+1,n} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{n,n} \\ \hat{y}_{n,n} \\ \hat{z}_{n,n} \\ \hat{\dot{x}}_{n,n} \\ \hat{\dot{y}}_{n,n} \\ \hat{\dot{z}}_{n,n} \end{bmatrix} + \begin{bmatrix} 0.5\Delta t^2 & 0 & 0 \\ 0 & 0.5\Delta t^2 & 0 \\ 0 & 0 & 0.5\Delta t^2 \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix} \begin{bmatrix} \ddot{x}_n \\ \ddot{y}_n \\ \ddot{z}_n \end{bmatrix}$$

EXAMPLE – FALLING OBJECT

Consider a free-falling object. The state vector includes the altitude h and the object's velocity $\dot{h}$:

$$\boldsymbol{\hat{x}}_n = \begin{bmatrix} \hat{h}_n \\ \hat{\dot{h}}_n \end{bmatrix}$$

The state transition matrix $\boldsymbol{F}$ is:

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

The control matrix $\boldsymbol{G}$ is:

$$\boldsymbol{G} = \begin{bmatrix} 0.5\Delta t^2 \\ \Delta t \end{bmatrix}$$

The input variable $\boldsymbol{u}_n$ is:

$$\boldsymbol{u}_n = \begin{bmatrix} g \end{bmatrix}$$

Where g is the gravitational acceleration.

We don't have a sensor that measures acceleration, but we know that for a falling object, acceleration equals g.

The state extrapolation equation looks as follows:

$$\left\lfloor \hat{\dot{h}}_{n+1,n} \right\rfloor \quad \left\lfloor 0 \quad 1 \right\rfloor \left\lfloor \hat{\dot{h}}_{n,n} \right\rfloor \quad \left\lfloor \Delta t \right\rfloor$$

The matrix multiplication results in the following:

$$\begin{cases} \hat{h}_{n+1,n} = \hat{h}_{n,n} + \hat{\dot{h}}_{n,n}\Delta t + 0.5\Delta t^2 g \\ \hat{\dot{h}}_{n+1,n} = \hat{\dot{h}}_{n,n} + \Delta t g \end{cases}$$

STATE EXTRAPOLATION EQUATION DIMENSIONS

The following table specifies the matrix dimensions of the state extrapolation equation variables:

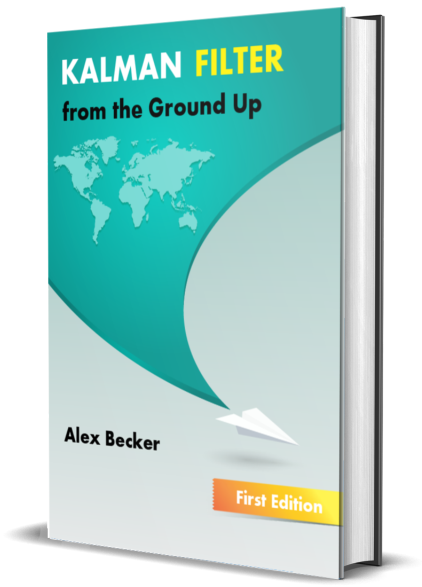
Variable	Description	Dimension
$\boldsymbol{x}$	state vector	$n_x \times 1$
$\boldsymbol{F}$	state transition matrix	$n_x \times n_x$
$\boldsymbol{u}$	input variable	$n_u \times 1$
$\boldsymbol{G}$	control matrix	$n_x \times n_u$
$\boldsymbol{w}$	process noise vector	$n_x \times 1$

LINEAR TIME-INVARIANT SYSTEMS

The Linear Kalman Filter assumes the **LTI (Linear Time-Invariant)** system model.

So, what is "linear," and what is "time-invariant"?

Linear systems are described by systems of equations in which the variables are never multiplied with each other but only with constants and then summed up. Linear systems are used to describe both static and dynamic relationships between variables.



A linear system is a system whose output function $y(t)$ satisfies the following equation:

$$y(t) = \mathcal{F}\left(a \times g\left(t\right) + b \times h\left(t\right)\right) = a \times \mathcal{F}\left(g\left(t\right)\right) + b \times \mathcal{F}\left(h\left(t\right)\right)$$

Where:

a and b are constant real numbers

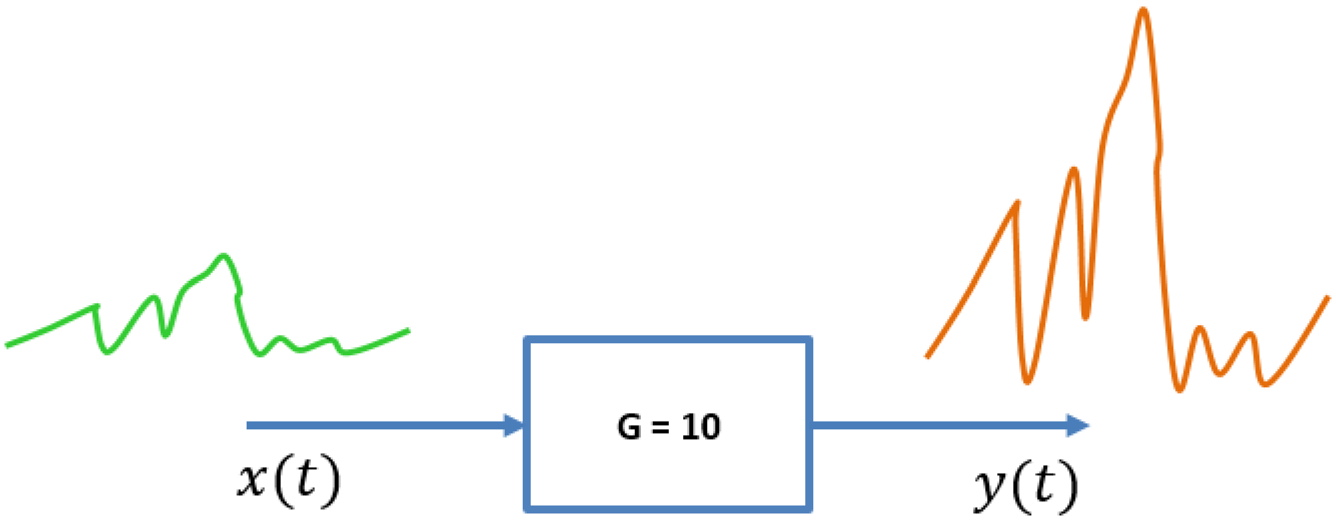
g and h are any arbitrary functions of an independent variable t

A linear system follows two basic rules:

1. You can "factor out" constant multiplicative scale factors (the a and b above).
2. The system's response to a sum of inputs is the sum of the responses to each input separately.

A **time-invariant** system has a **system function** that is not a direct function of time.

Let's take an amplifier with gain $G = 10$ as an example.



This system is time-invariant. Although the system's output changes with time, the system function is not time-dependent.

A time-invariant system is one where a time delay (or shift) in the input sequence causes an equivalent time delay in the system's output sequence.