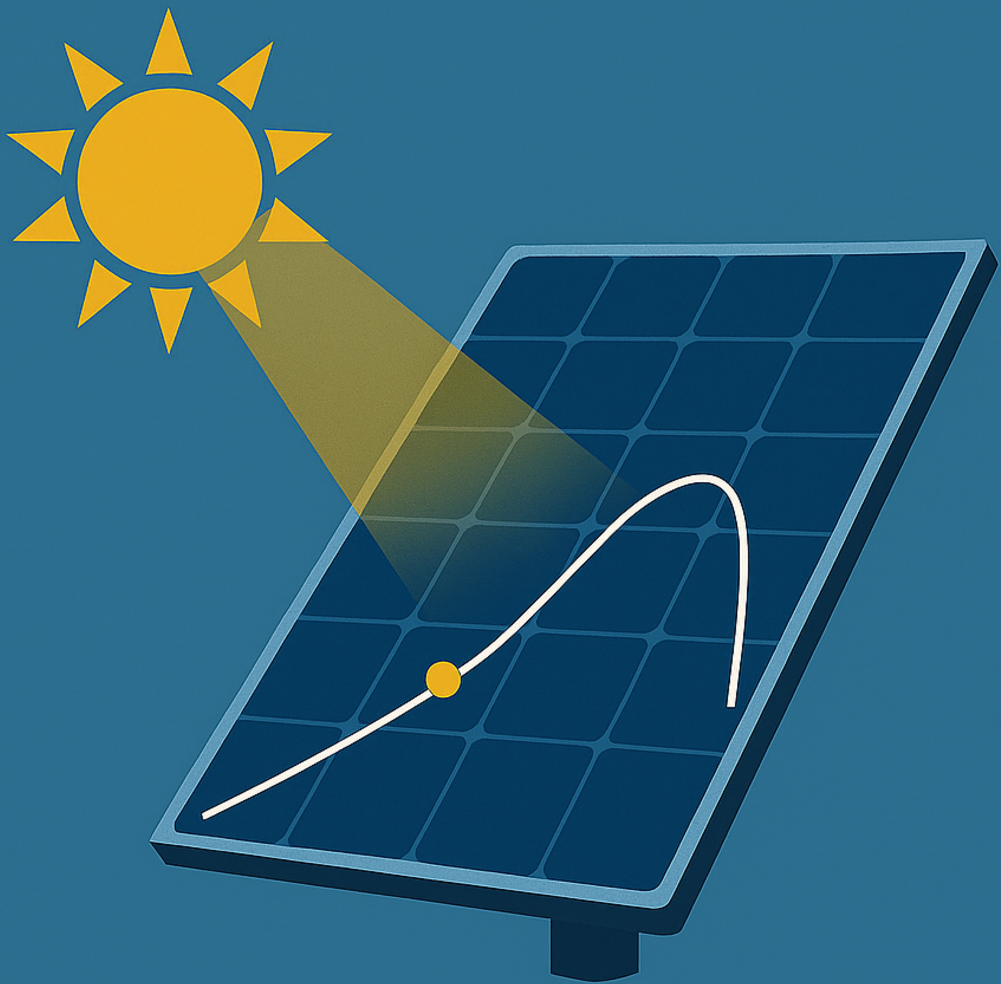
Analysis and implementation of MPPT algorithms: from the photovoltaic model to the Kalman filter



Leonardo Chieco

Analysis and Implementation of MPPT Algorithms
From the Photovoltaic Model to the Kalman Filter

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

Abstract

The efficiency of a photovoltaic system critically depends on the operating point at which the solar panel is made to operate. Photovoltaic cells do not provide a constant power output: it varies with irradiance, temperature, and the voltage across the module. To fully exploit the available energy, modern DC-DC converters adopt a control system known as MPPT (Maximum Power Point Tracking), capable of dynamically regulating voltage and current in order to maintain operation at the Maximum Power Point (MPP).
In this article, the physical and control principles underlying MPPT are analyzed, the main algorithms used are examined, and finally an advanced approach based on a Kalman filter is presented, as proposed in the publication *"A Kalman Filter Based MPPT Algorithm"* (IEEE, 2011), with an example of implementation in Python accompanied by a simulation of the dynamic behavior of the system.

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 Operating principle of solar photovoltaic panels

A *photovoltaic solar panel* (also referred to as a *PV module*) is an electronic device capable of directly converting light energy into electrical energy by means of the so-called photovoltaic effect. This effect, discovered in the 19th century by Alexandre-Edmond Becquerel, manifests itself in certain semiconductor materials in which the absorption of photons causes the release of electric charge carriers (electrons and holes), which can be separated and collected to generate a direct electric current.

The basic element of a solar panel is the *photovoltaic cell*.

Each cell consists of a thin layer of semiconductor material, generally silicon (Si), properly doped to create a p-n junction.

The p-layer (positive) contains an excess of holes (positive charge carriers), obtained by doping silicon with trivalent elements such as boron (B).

The n-layer (negative) contains an excess of free electrons, obtained by doping silicon with pentavalent elements such as arsenic (As).

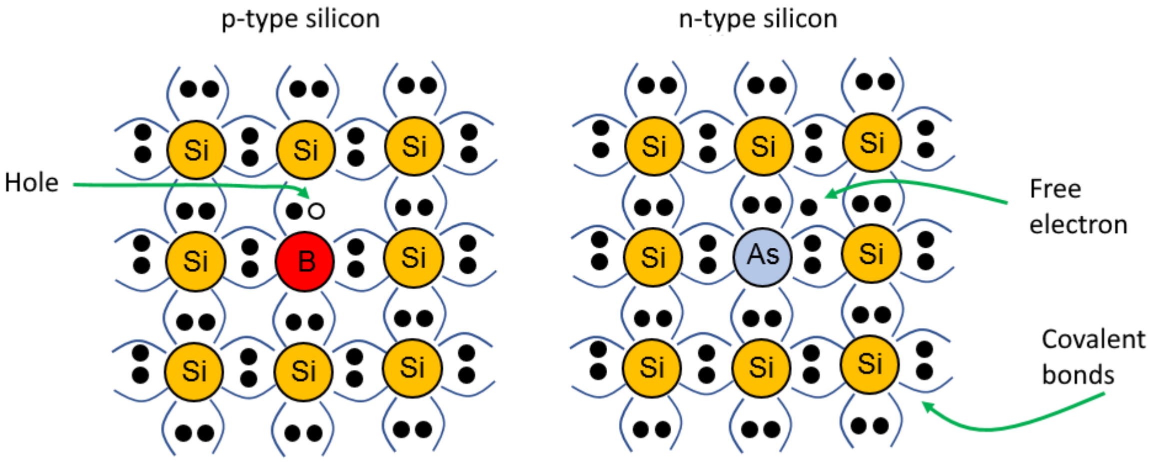
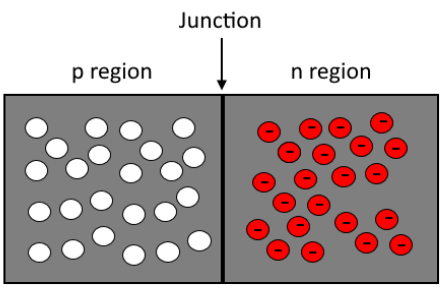
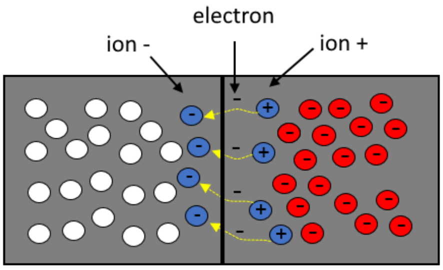


Figure 1: p-doped and n-doped silicon lattice

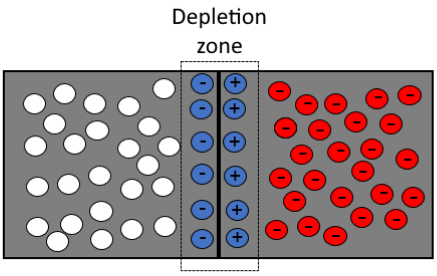
When two differently doped regions of silicon (one of p-type and one of n-type) are brought together, a p-n junction is formed.



At the instant of contact, the two materials are not in equilibrium. In the n-type region there is a high concentration of free electrons, while in the p-type region there are many holes.



This difference in concentration causes electrons at the contact point to spontaneously diffuse from the n-region (where they are in excess) toward the p-region (where they are lacking), while holes diffuse in the opposite direction, from the p-region to the n-region. When an electron moves from the n-type region to the p-type region, it combines with a hole, therefore causing both mobile carriers to disappear. Only the negative ions (which have captured an electron) and the positive ions (which have given up an electron) remain.



This process takes place in a thin region around the junction, where the concentration of mobile carriers is therefore strongly reduced (depletion region). The separation of fixed charges creates an internal electric field directed from the n-type region (positive) toward the p-type region (negative), which generates a force capable of repelling both additional electrons attempting to diffuse from n to p, and the holes trying to diffuse from p to n.

As diffusion continues, the accumulated charge increases, and therefore the electric field opposing further diffusion also grows.

At this point, a potential barrier v_b is formed, representing the energy difference required for an electron to cross the junction. For silicon, this barrier is typically on the order of 0.6-0.7 V.
When the p-n junction is exposed to sunlight, photons with energy greater than the silicon band gap (approximately 1.12 eV) can break one of the covalent bonds that bind the atoms in the crystal lattice of the semiconductor. In this way, electron-hole pairs are generated, which constitute the charge carriers responsible for the photovoltaic current.

If the pair is generated within or near the depletion region, the internal electric field immediately separates the two charges:

- the electron is accelerated toward the n-type region,
- while the hole is driven toward the p-type region.

This separation produces a potential difference across the junction, which can be used to power an external circuit.

It is precisely this action of the internal field, spontaneously generated within the depletion region, that makes the direct conversion of light into electricity possible.

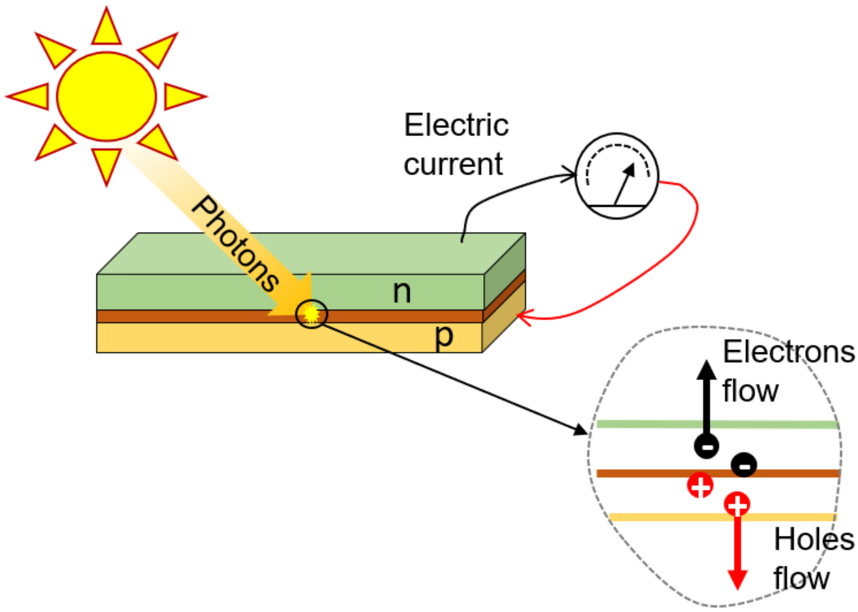


Figure 2: PV cell operation

A photovoltaic cell can therefore be regarded as an ideal current generator in which:

- the voltage v across the cell quickly settles to values approximately equal to its nominal value as soon as sunlight is present, and is a parameter strongly influenced by temperature;
- the amount of current i delivered is instead directly proportional to the incident solar irradiance and varies only slightly with temperature;

In addition to these well-known effects, there are losses that make the photovoltaic cell a non-ideal current generator. In particular, it is necessary to take into account two resistances (series and shunt), which are defined as follows:

- R_{SH} (shunt resistance): this resistance models all internal losses of the photovoltaic cell, such as manufacturing defects, design imperfections, and non-ideal purity of the materials used.
- R_S (series resistance): this resistance models all loss mechanisms encountered by the current along its path, such as the silicon-conductor contact resistance and the resistance of the top and bottom contacts of the entire cell.

The simplified model of a photovoltaic cell is shown in the following figure (single-diode model).

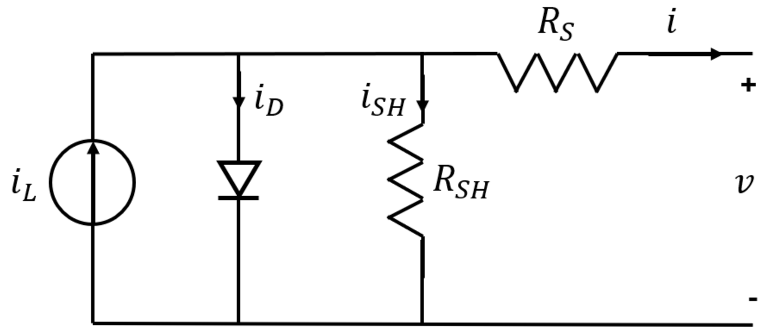


Figure 3: Electric model of a PV cell

The expression for i can be obtained by applying Kirchhoff's law:

$$i = i_L - i_D - i_{SH} = i_L - i_S\Big(e^{\frac{qv_D}{nkT}} - 1\Big) - \frac{v_D}{R_{SH}}.$$

It is observed that:

$$v_D = v_{R_{SH}} = v - iR_S.$$

Thus:

$$i = i_L - i_S\left(e^{\frac{q(v-iR_S)}{nkT}} - 1\right) - \frac{v - iR_S}{R_{SH}}.$$

where:

- i_L is the current generated by the photovoltaic effect,
- i_S is the diode reverse saturation current,
- R_S is the series resistance,
- R_{SH} is the shunt resistance,
- q is the electron charge,
- k is Boltzmann's constant,
- T is the absolute temperature,
- n is the ideality factor (or emission coefficient), a number between 1 and 2 that describes how much the real diode deviates from an ideal one. For an ideal diode, this factor is equal to 1.

A photovoltaic module is the assembly of multiple cells connected in series and/or in parallel:

- Series connection increases the output voltage.
- Parallel connection increases the output current.

Multiple PV modules connected in series form a *string*, while several strings connected together form a *photovoltaic array* capable of generating power on the order of several hundred watts.

The electrical characteristic of a PV module is not fundamentally different from that of a single photovoltaic cell. In fact, the same equivalent circuit and the corresponding Kirchhoff current

equation introduced for the cell can be used to describe a module, provided that the parameters are interpreted as module quantities.
For a module composed of N_s cells in series (and possibly N_p parallel strings), the model parameters are obtained by properly scaling the cell parameters: the voltages scale approximately with N_s, the currents with N_p, and the series and shunt resistances are adjusted accordingly. Once this scaling has been applied, the current–voltage characteristic of the PV module is computed using the same diode model equation previously derived for the single cell.

For example, a typical module consisting of 36 silicon cells connected in series can provide approximately 18–20 V at open circuit and 5–6 A at maximum power. Each individual cell is capable of delivering a similar amount of current under operating conditions (on the order of 5–6 A at its own maximum power point), but only at about 0.55-0.6 V. When the cells are connected in series, their voltages add up while the current remains the same through all of them. As a result, the module exhibits a short-circuit and maximum-power current comparable to that of a single cell, whereas its voltage increases in proportion to the number of series-connected cells.

Note: from this point onward, the single-diode model and all related equations shall be considered as referring to a photovoltaic module, rather than to an individual cell. All parameters and characteristics are therefore interpreted as module-level quantities obtained through the appropriate scaling of the underlying cell model.

Given the above considerations, the electrical behavior of a photovoltaic panel is represented by the current-voltage (i-v) curve shown in the following image.

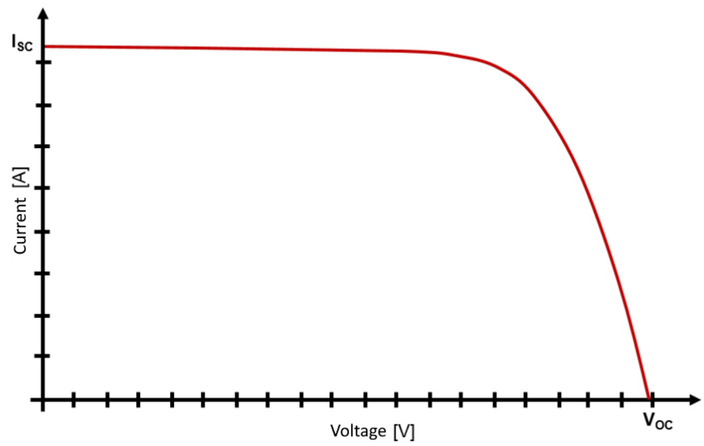


Figure 4: the i-v characteristic of a PV panel

Under open-circuit conditions (with no load connected), the output current is zero and the voltage reaches its maximum value, referred to as the open-circuit voltage v_{OC}.

Under short-circuit conditions (with a zero-resistance load), the voltage is zero and the current reaches its maximum value, referred to as the short-circuit current i_{SC}.

Between these two extremes, the current decreases as the voltage increases, forming a characteristic convex curve. The electrical power delivered by the panel is given by $p(v) = v \cdot i(v)$ and exhibits a maximum at a specific operating voltage v_{MPP} and current i_{MPP}.

The point $(v_{\mathrm{MPP}}, i_{\mathrm{MPP}})$ is referred to as the "maximum power point" (MPP).

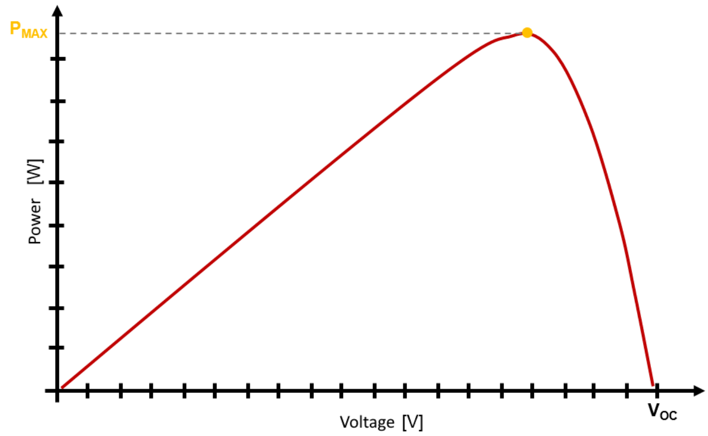


Figure 5: the p-v characteristic of a PV panel

The panel should always operate near the MPP in order to maximize the generated power and the overall efficiency of the system.

Environmental conditions significantly affect the position of the MPP:

- Solar irradiance determines the current that the module can generate: as irradiance increases, the i-v curve shifts upward.
- Cells temperature affects the voltage: when temperature rises, both v_{MPP} and v_{OC} tend to decrease.

If the panel is connected to a load that does not force it to operate at the MPP, the extracted power will be lower than the available one. For example, a panel with $v_{\mathrm{MPP}} = 30\,\mathrm{V}$ and $i_{\mathrm{MPP}} = 8\,\mathrm{A}$ ideally produces about 240 W. If it is connected directly to a 12 V battery without any adaptation stage, the generated current will drop significantly, and the transferred power will decrease accordingly.

The MPPT regulation forces the photovoltaic module to operate continuously at its maximum power point (MPP), keeping its output voltage close to v_{MPP}. A subsequent regulation stage converts this power so that it can be used to charge the battery.

An MPPT regulator is essentially a DC-DC converter (buck, boost, or buck-boost depending on the voltage levels) placed between the photovoltaic array and the battery, with the task of:

- Operating the PV module near its MPP (or reasonably close to it), thereby maximizing the extracted power.
- Adapting this power to the battery's charging voltage, with the appropriate current, while respecting the proper charging stages.

In technical terms, the MPPT regulator operates by adjusting the impedance seen by the photovoltaic panel, so that the operating point $(v_{\mathrm{op}}, i_{\mathrm{op}})$ lies close to the maximum power point.

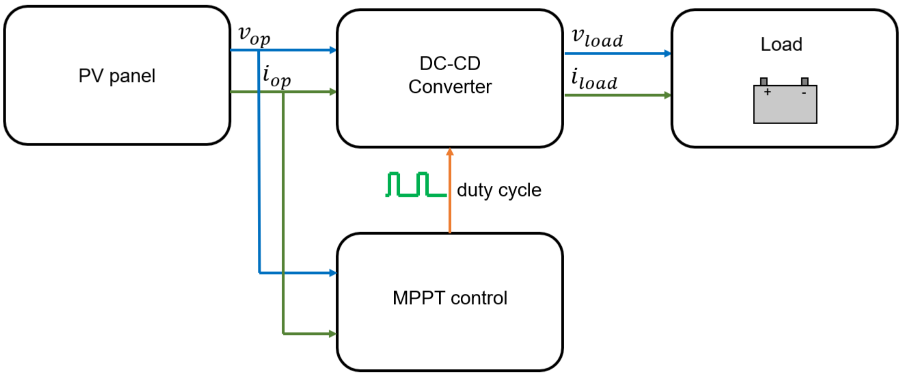


Figure 6: Block diagram of an MPPT system

By adjusting the duty cycle of the DC-DC converter, the MPPT controller effectively modifies the operating current i_{op}. Since the panel behaves as a nonlinear source, changing i_{op} forces a corresponding change in the operating voltage v_{op}, thus allowing the system to move along the i-v curve in search of the MPP.

Note: hereafter, reference will be made to adjusting the operating voltage v_{op}, although this action is in practice carried out by modifying the duty cycle.

It is important to note that the maximum power point (MPP) of the panel cannot always be exploited. The MPPT system can extract the maximum available power only if the load (such as a battery) is able to absorb all the power provided by the module.

In the context of charging a lithium battery, the MPPT is fully operative and effective during the Constant Current (CC) phase, since in this phase the battery demands and absorbs the maximum possible current, thus using all the power that the regulator can provide.
Conversely, during the Constant Voltage (CV) phase (or float phase), the regulator maintains the voltage at the specified maximum value. As the battery approaches full charge, the charging current decreases drastically. As a consequence, the regulator can no longer extract the maximum power from the panel (p_{MPP}), because the battery is simply no longer able to absorb it.

It is also essential that the operating voltage v_{op} of the panel is sufficiently higher than the battery voltage, plus the margin required by the converter. If the panel voltage is too low compared to the battery voltage, the MPPT will not be able to raise the operating point toward the MPP and will therefore be unable to extract optimal power.
For example, if the battery is at 12 V and the panel v_{MPP} under full sunlight is only 15 V, a buck converter would have very little margin. It is generally preferable to use a panel or a string (i.e., a series connection of multiple PV modules) with v_{MPP} around 30–40 V.
Moreover, the maximum output current of the regulator (toward the battery) must be consistent with the panel's power, since $p_{\text{max}} \approx v_{\text{MPP}} \cdot i_{\text{MPP}}$. The regulator must therefore be rated for an adequate current.

In the design of a photovoltaic system, it is important to remember that its operation is strongly influenced by two main environmental parameters:

- Solar irradiance

- As irradiance increases, the short-circuit current i_{SC} increases almost linearly.
- The open-circuit voltage v_{OC}, on the other hand, increases logarithmically and at a much slower rate.
- The maximum available power increases proportionally to the incident radiation.

- Cell temperature
  - As temperature increases, the open-circuit voltage v_{OC} decreases significantly (approximately $-2\,\mathrm{mV/°C}$ per silicon cell).
  - The short-circuit current i_{SC} increases slightly.
  - The overall efficiency decreases, since the thermal increase reduces the semiconductor bandgap.

A fundamental parameter for assessing the efficiency and quality of a photovoltaic module is the so called fill factor (FF), a dimensionless number that measures the degree of purity and the effective utilization of the silicon wafer that constitutes the module. It ranges between 0 and 1, and the closer it is to unity, the higher the panel quality, the better the materials used, and consequently the lower the losses and associated aging effects. The fill factor (FF) is therefore defined as:

$$FF = \frac{v_{\mathrm{MPP}}\, i_{\mathrm{MPP}}}{v_{\mathrm{OC}}\, i_{\mathrm{SC}}}$$

Simplified Mathematical Model

Like the photovoltaic cell, as mentioned above, the output current of a photovoltaic panel can be modeled by the following relation:

$$i = i_L - i_S\left(e^{\frac{q(v-R_S i)}{nkT}} - 1\right) - \frac{v - R_S i}{R_{SH}}$$

The same single-diode model used for the cell is employed here, with all parameters interpreted as module-level quantities obtained through the appropriate scaling.

The current i appears on both sides of the equation, therefore it must be solved to obtain the explicit function $i(v)$. This procedure is rather complex and lies beyond the scope of this tutorial.

It is important to note, however, that for many practical applications (fast simulations, control algorithms, experimental fitting) it is convenient to use a simplified analytical form that satisfies the main physical constraints:

- $i(0) = i_{\mathrm{SC}}$ (short-circuit current),
- $i(v_{\mathrm{OC}}) = 0$ (zero current at open circuit),
- it exhibits the correct qualitative shape: nearly constant current for small v, followed by a rapid drop to zero around v_{OC}.

A commonly used form is:

$$i(v) = i_{\mathrm{SC}}\left(1 - \left(\frac{v}{v_{\mathrm{OC}}}\right)^n\right) \tag{1}$$

It satisfies the above constraints exactly and, by varying n, it allows the "curvature" of the i-v curve to be modeled. If n is small, the curve is smoother, whereas if n is large, it decreases rapidly with a more pronounced "knee".

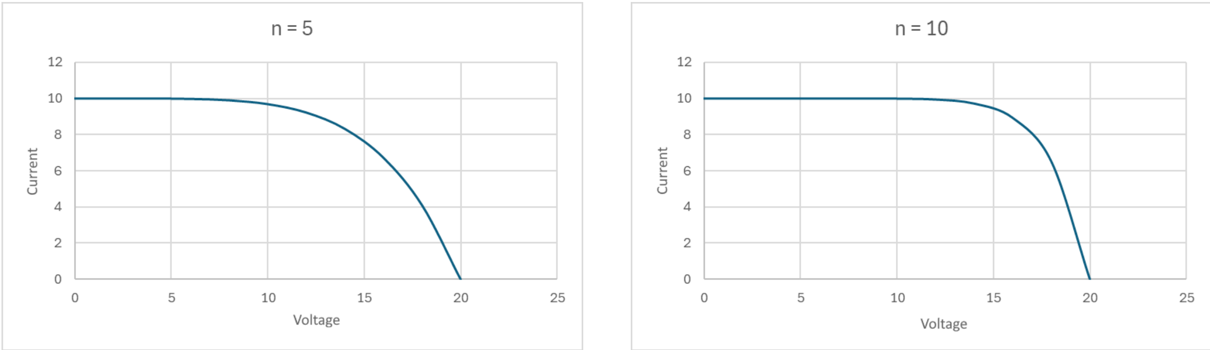


Figure 7: i-v curves obtained from eq.(1)
with $n = 5$ and $n = 10$

It should be noted that this relation is a simplified empirical approximation, not an exact solution.
There are several ways to determine the value of n. The simplest approach consists of fitting experimental data, or alternatively the following method may be used. If the model is required to reproduce the experimental current at the maximum power point, then:

$$i_{\mathrm{MPP}} = i_{\mathrm{SC}}\left(1 - \left(\frac{v_{\mathrm{MPP}}}{v_{\mathrm{OC}}}\right)^{n}\right)$$

from which, isolating n, it follows that

$$\left(\frac{v_{\mathrm{MPP}}}{v_{\mathrm{OC}}}\right)^{n} = 1 - \frac{i_{\mathrm{MPP}}}{i_{\mathrm{SC}}}.$$

and therefore

$$n = \frac{\ln\left(1 - \frac{i_{\mathrm{MPP}}}{i_{\mathrm{SC}}}\right)}{\ln\left(\frac{v_{\mathrm{MPP}}}{v_{\mathrm{OC}}}\right)}.$$

Note: the value of n is determined for specific environmental conditions and does not account for variations in temperature or irradiance. Therefore, this model is valid only for the particular operating condition for which it has been calibrated.

2 MPPT algorithms

Over the years, several algorithms have been developed and are now commonly implemented in MPPT regulators for the purpose of locating and tracking the MPP. Among the most widely used, the following may be mentioned:

- Perturb and Observe (P&O)
- Incremental Conductance (IncCond)

Perturb and observe method

The P&O method is among the most widely used due to its simplicity. A small perturbation Δv is applied to the panel voltage v_{op}, and the corresponding variation in power is measured. If

$$\frac{\Delta p}{\Delta v} > 0 \quad \rightarrow \quad \text{the operating point moves toward the MPP} \quad \rightarrow \quad v_{\text{op}} \text{ is increased}$$

If

$$\frac{\Delta p}{\Delta v} < 0 \quad \rightarrow \quad \text{the operating point moves away from the MPP} \quad \rightarrow \quad v_{\text{op}} \text{ is decreased}$$

The method is simple to implement but is sensitive to noise and introduces small oscillations around the optimal operating point.

Incremental conductance method

It is known that at the MPP the derivative of P with respect to v is zero, and that

$$\frac{d\big(x\,f(x)\big)}{dx} = f(x) + x\,\frac{df(x)}{dx}.$$

Therefore,

$$\frac{dp}{dv} = \frac{d\big(v \cdot i(v)\big)}{dv} = i + v\,\frac{di}{dv} = 0 \quad \Rightarrow \quad \frac{di}{dv} = -\,\frac{i}{v}.$$

This simple equation is the core of the Incremental Conductance method, which ultimately compares the instantaneous conductance $\frac{i_{\text{op}}}{v_{\text{op}}}$ with the incremental conductance $\frac{\Delta i}{\Delta v}$, the latter being a discrete approximation of $\frac{di}{dv}$.

The decision criterion becomes:

$$\frac{\Delta i}{\Delta v} = -\,\frac{i_{\text{op}}}{v_{\text{op}}} \quad \Rightarrow \quad \text{the operating point is at the MPP}$$

$$\frac{\Delta i}{\Delta v} > -\,\frac{i_{\text{op}}}{v_{\text{op}}} \quad \Rightarrow \quad \text{the operating point is to the left of the MPP}$$

$$\frac{\Delta i}{\Delta v} < -\,\frac{i_{\text{op}}}{v_{\text{op}}} \quad \Rightarrow \quad \text{the operating point is to the right of the MPP}$$

This allows the DC-DC controller to determine in which direction to move without having to oscillate around the optimal point, as occurs in the P&O method.

It is noted that the optimization is not fixed: the MPP varies with environmental conditions such as temperature and irradiance. Moreover, there exists a trade-off between tracking speed

and stability. If the algorithm adjusts too aggressively, oscillations may be induced; conversely, if it responds too slowly, power is lost during rapid changes.
In the presence of partially shaded panel strings, multiple local power peaks may appear. The algorithm must therefore avoid becoming trapped in a local maximum instead of reaching the global one.
Advanced methods such as Fuzzy Logic, Neural Networks, and Kalman Filtering have been proposed to improve dynamic response and stability under high noise levels or rapid irradiance variations.

3 MPPT algorithm with a Kalman filter

Maximum Power Point Tracking (MPPT) is a fundamental component in modern photovoltaic systems. While traditional algorithms introduce oscillations around the optimal operating point and may become slow or unstable under changing weather conditions, a new family of strategies has emerged in recent years: state–filter-based MPPT methods, such as the Kalman filter.
The paper "A Kalman Filter Based MPPT Algorithm" (IEEE, 2011) proposes precisely this idea: using the measured power and panel voltage to reconstruct the internal state of the system, in particular the slope of the p–v curve and the voltage v_{MPP}.

The fundamental idea behind the approach is that the maximum power point (MPP) of a photovoltaic panel is not directly observable: there is no sensor capable of indicating at any given time "the MPP is exactly at X volts". However, when the operating point moves closer to or farther from the MPP, this change produces measurable effects, particularly variations in the delivered power as the voltage changes.

The proposed method relies on the fact that the MPP voltage cannot be measured directly, although its value influences how the power changes when the operating voltage is slightly perturbed. For this reason, the state vector includes both the power gradient $\frac{dp}{dv}$ and the optimal voltage v_{MPP} as hidden states, meaning that they are not measured directly but must be estimated from the available data. Among them, only v_{MPP} is a non-observable physical quantity, since no measurement provides direct information about its value, while the gradient is introduced mainly to obtain a smoother and more reliable estimation.
The key idea behind the algorithm is that the power gradient is zero at the maximum power point and changes sign when moving to either side of it. By estimating $\frac{dp}{dv}$ in real time, the filter can determine whether the operating point lies above or below the MPP voltage and adjust its estimate accordingly.

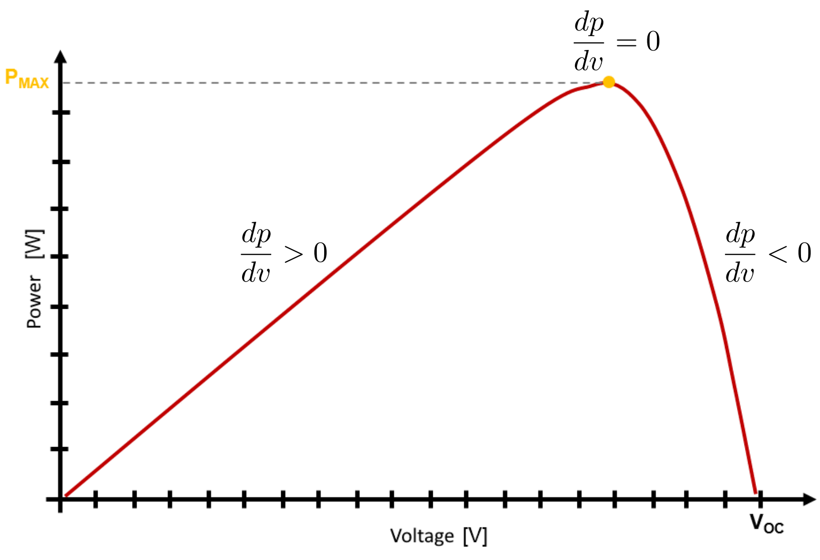


Figure 8: Sign change of $\dfrac{dp}{dv}$ in the p-v curve

The Kalman filter is used precisely to estimate and smooth this derivative, avoiding oscillations and improving robustness compared with traditional methods (such as P&O).

The identification of the system model is now considered.

The power delivered by the panel is assumed to be a function of the discrete-time index k and of the voltage:

$$p_{\mathrm{op},k} = f(v_{\mathrm{op},k}) + \eta_k \tag{2}$$

where η_k represents the Gaussian measurement error

$$\eta_k \sim \mathcal{N}(0, \sigma_R^2).$$

A small and known perturbation Δv is applied to the current operating voltage $v_{\mathrm{op},k}$ (i.e. $\Delta v =$ 0.2 V)
and the corresponding variation in power is observed:

$$\Delta p_k = p(v_{\mathrm{op},k} + \Delta v) - p(v_{\mathrm{op},k}).$$

From these quantities, an approximate derivative can be estimated:

$$\frac{dp}{dv} \approx \frac{\Delta p_k}{\Delta v} = \left(\frac{\Delta p}{\Delta v}\right)_k.$$

However, this estimate is noisy and unstable.

The Kalman filter is applied to provide an optimal estimate of this derivative, thus yielding information for the subsequent voltage v_{MPP} estimation. The state is therefore defined as:

$$\mathbf{x}_k = \begin{bmatrix} \left(\frac{\Delta p}{\Delta v}\right)_k \\ v_{\mathrm{MPP},k} \end{bmatrix} = \begin{bmatrix} \text{Estimated power gradient} \\ \text{Estimated maximum power point voltage} \end{bmatrix}.$$

It is assumed that the system follows a slow and quasi-stationary dynamics, meaning that the parameters do not change abruptly from one sample to the next. This makes it possible to use a very simple state transition model:

$$\mathbf{x}_k = A\,\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \qquad k \geq 1,$$

where

- $\mathbf{w}_{k-1}$ is the process noise, modeled as Gaussian noise with covariance matrix Q
- A is the state transition matrix

Under the assumption that the system evolves slowly over time, meaning that no significant state variations are expected between two consecutive sampling instants, it can be stated that:

$$\mathbf{x}_k \approx \mathbf{x}_{k-1}.$$

thus:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

This means that:

$$\mathbf{x}_k = A\,\mathbf{x}_{k-1} + \mathbf{w}_{k-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \left(\frac{\Delta p}{\Delta v}\right)_{k-1} \\ v_{\mathrm{MPP},\,k-1} \end{bmatrix} + \begin{bmatrix} w_{1,\,k-1} \\ w_{2,\,k-1} \end{bmatrix} \implies \begin{cases} \left(\frac{\Delta p}{\Delta v}\right)_k = \left(\frac{\Delta p}{\Delta v}\right)_{k-1} + w_{1,\,k-1} \\ v_{\mathrm{MPP},\,k} = v_{\mathrm{MPP},\,k-1} + w_{2,\,k-1} \end{cases}$$

This assumption is entirely reasonable, since the i-v curve of the panel does not change rapidly unless the irradiance varies abruptly. Consequently, both the maximum power point and the gradient do not change significantly from one step to the next.
It is noted that, if all the elements of A were different from zero:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix},$$

the following relations would hold:

$$\begin{cases} \left(\dfrac{\Delta p}{\Delta v}\right)_k = a_{11} \left(\dfrac{\Delta p}{\Delta v}\right)_{k-1} + a_{12}\, v_{\mathrm{MPP},\, k-1} + w_{1,\, k-1} \\ v_{\mathrm{MPP},\, k} = a_{21} \left(\dfrac{\Delta p}{\Delta v}\right)_{k-1} + a_{22}\, v_{\mathrm{MPP},\, k-1} + w_{2,\, k-1} \end{cases}$$

There is no known direct physical relationship between the gradient $\dfrac{dp}{dv}$ and the value of v_{MPP}: the former is a local derivative quantity, whereas the latter is a global equilibrium point of the p-v curve. Therefore, the introduction of the coefficients a_{12} and a_{21} (i.e., cross couplings) would impose an unrealistic link and could destabilize the estimation. Thus, $a_{21} = a_{12} = 0$.

The observation equation (that is, the quantity that is actually measured) is:

$$y_k = H\, \mathbf{x}_k + \mu_k$$

where μ_k is the measurement noise affecting the computed power gradient, which, in turn, is correlated with the noise over the power measurements defined in eq.(2)

$$\eta_k \sim \mathcal{N}(0, \sigma_R^2) \implies \mu_k \sim \mathcal{N}\left(0, \frac{2\sigma_R^2}{(\Delta v)^2}\right).$$

The matrix H is

$$H = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

since the only directly available measurement (derived from power and voltage data) is the approximate derivative $\dfrac{\Delta p}{\Delta v}$.

An initial covariance matrix is assumed to be:

$$P_0^+ = \begin{bmatrix} \sigma^2_{\frac{dp}{dv}} & 0 \\ 0 & \sigma^2_{v_{\mathrm{MPP}}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

- Gradient variance: $\sigma^2_{\frac{dp}{dv}} = 1\,(\mathrm{W/V})^2$, representing the uncertainty on the gradient.
- Voltage variance: $\sigma^2_{v_{\mathrm{MPP}}} = 1\,\mathrm{V}^2$, representing the uncertainty on the voltage.

The elements p_{12}^+ and p_{21}^+ are equal to zero because no dependency exists between uncertainties of $\dfrac{\Delta p}{\Delta v}$ and of v_{MPP}.

Further considerations on $\sigma^2_{v_{\mathrm{MPP}}}$ and on the process variance q_{22} associated with the second state element $v_{\mathrm{MPP},k}$ will be provided in the numerical example below (see Section 4).

It is also recalled that larger initial values of P indicate less confidence in the prediction model, which in this case is conservative since $A = I$ and therefore $\mathbf{x}_k = \mathbf{x}_{k-1}$. As a result, higher values of P make the filter trust the measurements more than the model, allowing it to adjust more quickly during the first iterations.

As already described in this Kalman filter tutorial, the Kalman filter consists of two stages:

1. Prediction:
$$\mathbf{x}_k^- = A\,\mathbf{x}_{k-1}^+$$
$$P_k^- = A\,P_{k-1}^+\,A^\top + Q$$
2. Update:
$$K_k = P_k^- H^\top \big(H P_k^- H^\top + \sigma_R^2\big)^{-1}$$
$$\mathbf{x}_k^+ = \mathbf{x}_k^- + K_k\big(y_k - H\,\mathbf{x}_k^-\big)$$
$$P_k^+ = (I - K_k H)\,P_k^-,$$

where

- $\mathbf{x}_k^+$ is the estimated state at time step index k,
- P_k^+ is the error covariance matrix,
- K_k is the Kalman gain,
- Q and σ_R^2 represent the process noise covariance matrix and the measurement noise variance, respectively.

Under these assumptions, and since $H = \begin{bmatrix}1 & 0\end{bmatrix}$, it is easy to see that the updated estimate $v_{\text{MPP},k}^+$ is not directly corrected by the Kalman filter: the measurement only acts on the first state component. A coupling between the two states could be indirectly introduced by suitable off-diagonal terms in the process noise covariance Q. In the present formulation, however, such coupling is not considered. Thus, the estimation of the MPP voltage $v_{\text{MPP},k}^+$ needs to be explicitly updated now that an estimate of the gradient is available.
The strategy used updates the MPP voltage estimate according to the following rules:

- if the gradient is positive, $v_{\text{MPP},k}^+$ has to be increased;
- if the gradient is negative, $v_{\text{MPP},k}^+$ has to be decreased;
- if the gradient is zero, $v_{\text{MPP},k}^+$ is the true MPP voltage.

Therefore, once the updated estimate of the gradient $\left(\dfrac{\Delta p}{\Delta v}\right)_k^+$ has been obtained, the MPPT algorithm updates the $v_{\text{MPP},k}^+$ estimate according to an adaptive rule:

$$v_{\text{MPP},k}^c = v_{\text{MPP},k}^+ + \alpha\left(\frac{\Delta p}{\Delta v}\right)_k^+ \tag{3}$$

where α is a small step factor that determines the convergence speed of the estimation process of the MPP voltage.

When the estimated derivative approaches zero, the voltage $v^c_{\mathrm{MPP,k}}$ converges to the optimal true value v_{MPP}.

By introducing the matrix C for the MPP voltage estimation stage

$$C = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix},$$

the updated state $\mathbf{x}^c_k$ can be expressed as:

$$\mathbf{x}^c_k = C\mathbf{x}^+_k.$$

The matrix C is designed so that the MPP voltage estimation stage, in the scope of the same iteration, does not interfere with the Kalman filtering stage, which operates exclusively on the first component of the state vector. Conversely, C modifies only the second (hidden) state, whose purpose is to track the evolution of MPP voltage estimate and drive it toward the true MPP voltage value.

Finally, the control stage is performed. Its purpose is to move the panel operating voltage v_{op} toward the MPP, based on the most recent estimation of the MPP voltage. Since the DC-DC converter cannot instantaneously apply a new voltage setpoint to the PV module, an inertial behaviour is introduced through the following update law:

$$v_{\mathrm{op},k} = v_{\mathrm{op},k-1} + \beta\big(v^c_{\mathrm{MPP},k} - v_{\mathrm{op},k-1}\big), \tag{4}$$

where the coefficient β models the dynamic response of the power converter.

A summary of one complete MPPT iteration (omitting the matrix A, since $A = I$) is reported in Table 1.

Step	**Operation**
Kalman predict	$\mathbf{x}^-_k = \mathbf{x}^c_{k-1}$ $P^-_k = P^+_{k-1} + Q$
Kalman update	$K_k = P^-_k H^\top \big(H\, P^-_k\, H^\top + \sigma^2_R\big)^{-1}$ $\mathbf{x}^+_k = \mathbf{x}^-_k + K_k\big(y_k - H\mathbf{x}^-_k\big)$ $P^+_k = (I - K_k H)\, P^-_k$
MPP estimate update	$\mathbf{x}^c_k = C\,\mathbf{x}^+_k$
Control (operating voltage)	$v_{\mathrm{op},k} = v_{\mathrm{op},k-1} + \beta\big(v^c_{\mathrm{MPP},k} - v_{\mathrm{op},k-1}\big)$

Table 1: Summary of one MPPT iteration, including Kalman prediction and update, MPP voltage estimation, and operating voltage control.

4 Numerical Example

An example is now provided to illustrate the algorithm more clearly. In this numerical example, the photovoltaic generator is modelled through the nonlinear i-v characteristic

$$i(v) = i_{\mathrm{SC}}\left(1 - \left(\frac{v}{v_{\mathrm{OC}}}\right)^n\right), \qquad p(v) = v \cdot i(v),$$

with the parameters $i_{\mathrm{SC}} = 8\,\mathrm{A}$, $v_{\mathrm{OC}} = 38\,\mathrm{V}$, and $n = 9$. This model yields a true maximum power point located at approximately

$$v_{\mathrm{MPP,true}} \simeq 29.4 \text{ V}.$$

Initial parametrization

The initial system state used as input for the Kalman filter is chosen as

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

meaning that the initial gradient estimate is set to $\left(\frac{\Delta p}{\Delta v}\right)_0^c = 0$, while the initial v_{MPP} voltage estimate is set to $v_{\mathrm{MPP},0}^c = 10$ V. An initial panel operating voltage $v_{\mathrm{op},0}$ is assumed to be

$$v_{\mathrm{op},0} = 9\ V.$$

The initial error covariance, the process noise covariance and the measurement noise variance are defined as:

$$P_0^+ = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad Q = \begin{bmatrix} 10^{-3} & 0 \\ 0 & 10^{-3} \end{bmatrix}, \qquad \sigma_R^2 = 0.05.$$

The state transition and observation matrices are specified as:

$$A = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad H = \begin{bmatrix} 1 & 0 \end{bmatrix},$$

while the parameter α described in eq.(3) is

$$\alpha = 0.05\,,$$

meaning that C is

$$C = \begin{bmatrix} 1 & 0 \\ 0.05 & 1 \end{bmatrix}.$$

Finally, the operating voltage update coefficient β used in eq.(4) is set to

$$\beta = 0.5\,.$$

First iteration ($k = 1$)

This iteration (as well as all subsequent ones) consists of three stages:

1. *Kalman estimation*: computation of the updated state estimate $\mathbf{x}_1^+$. The main goal is to calculate the quantity $\left(\frac{\Delta p}{\Delta v}\right)_1^+$ (first component of the state vector) to update the estimate of the MPP voltage.
2. *MPP estimate update*: computation of the estimate of the MPP voltage $\mathbf{x}_1^c = C\,\mathbf{x}_1^+$, from which the quantity $v^c_{\text{MPP},1}$ (second component of the state vector) is extracted and used as reference for the PV panel operating voltage.
3. *Control update*: computation of the operating voltage that the DC-DC converter should impose to the panel
$$v_{\text{op},1} = v_{\text{op},0} + \beta\big(v^c_{\text{MPP},1} - v_{\text{op},0}\big).$$

Kalman estimation

Prediction

The predicted state and covariance are obtained as:

$$\mathbf{x}_1^- = \mathbf{x}_0^c = \begin{bmatrix} 0 \\ 10 \end{bmatrix}.$$

The predicted error covariance is:

$$P_1^- = P_0^+ + Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 10^{-3} & 0 \\ 0 & 10^{-3} \end{bmatrix} = \begin{bmatrix} 1.001 & 0 \\ 0 & 1.001 \end{bmatrix}.$$

Measurement

To simulate the first noisy measurement of the power gradient, two close operating points around the initial operating voltage $v_{\text{op},0}$ are considered. This small perturbation reflects what real MPPT controllers do when estimating the local power slope by slightly adjusting the panel operating voltage. $\Delta v = 0.2\,\text{V}$ is assumed to be applied to the current operating voltage in order to obtain the two evaluations points a and b:

$$v_a = v_{\text{op},0} = 9 \text{ V}, \qquad v_b = v_{\text{op},0} + \Delta v = 9.2 \text{ V}.$$

In a real implementation, the power measurements would be obtained by combining voltage and current sensors. In this numerical example, the photovoltaic model is used to generate the corresponding power values, to which artificial Gaussian noise is added to emulate measurement uncertainty.
The power at the two operating points is therefore:

$$p(v) = v\,i(v), \qquad i(v) = i_{\text{SC}}\Big(1 - (v/v_{\text{OC}})^n\Big),$$

with $i_{\text{SC}} = 8\,\text{A}$, $v_{\text{OC}} = 38\,\text{V}$ and $n = 9$. Evaluating the model gives:

$$p_a = p(v_a) \approx 72 \text{ W}, \qquad p_b = p(v_b) \approx 73.6 \text{ W}.$$

Gaussian measurement noise with variance $\sigma_R^2 = 0.05$ is added to both samples:

$$p_a^{\text{meas}} = p_a + \eta_a, \qquad p_b^{\text{meas}} = p_b + \eta_b, \qquad \eta_a, \eta_b \sim \mathcal{N}(0, \sigma_R^2).$$

The resulting noisy measurement of the power gradient is:

$$y_1 = \frac{p_b^{\text{meas}} - p_a^{\text{meas}}}{\Delta v} = \frac{p_b - p_a}{0.2} + \frac{\eta_b - \eta_a}{0.2}.$$

Since $(p_b - p_a)/0.2 \approx 8.0\,\text{W/V}$, and assuming a noise realization $\eta_b - \eta_a \approx 0.2$, the resulting measurement is:

$$y_1 \approx 8.0 + \frac{0.2}{0.2} = 9.0 \text{ W/V}.$$

Update

The measurement residual (innovation) is given by:

$$\tilde{y}_1 = y_1 - H\,\mathbf{x}_1^- = 9 - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 10 \end{bmatrix} = 9.$$

The innovation covariance is:

$$S_1 = H\,P_1^- H^\top + \sigma_R^2 = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1.001 & 0 \\ 0 & 1.001 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0.05 = 1.001 + 0.05 = 1.051.$$

The Kalman gain is then computed as:

$$K_1 = P_1^- H^\top S_1^{-1} = \begin{bmatrix} 1.001 & 0 \\ 0 & 1.001 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \frac{1}{1.051} = \frac{1}{1.051} \begin{bmatrix} 1.001 \\ 0 \end{bmatrix} \approx \begin{bmatrix} 0.9524 \\ 0 \end{bmatrix}.$$

The updated state estimate is:

$$\mathbf{x}_1^+ = \mathbf{x}_1^- + K_1 \tilde{y}_1 \approx \begin{bmatrix} 0 \\ 10 \end{bmatrix} + \begin{bmatrix} 0.9524 \\ 0 \end{bmatrix} 9.0 = \begin{bmatrix} 0.9524 \cdot 9.0 \\ 10 \end{bmatrix} \approx \begin{bmatrix} 8.57 \\ 10 \end{bmatrix}.$$

Finally, the updated error covariance is:

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

where

$$I - K_1 H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.9524 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 - 0.9524 & 0 \\ 0 & 1 \end{bmatrix} \approx \begin{bmatrix} 0.0476 & 0 \\ 0 & 1 \end{bmatrix}.$$

Thus,

$$P_1^+ \approx \begin{bmatrix} 0.0476 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1.001 & 0 \\ 0 & 1.001 \end{bmatrix} \approx \begin{bmatrix} 0.0477 & 0 \\ 0 & 1.001 \end{bmatrix}.$$

It is worth noting that $p_{11,\,1}^+ \approx 0.0477$ is far smaller than the initial value $p_{11,\,0}^+ = 1$, indicating that the estimation has become significantly more confident. Further considerations needs to be done for the value of $p_{22,\,1}^+$ and this will indeed be discussed in the second iteration.

Moreover, it can be observed that, given the choice of A, H, P_0^+, and Q, the Kalman filter does not update the second component of the state vector ($v_{\text{MPP},1}^+ = v_{\text{MPP},0}^+ = 10$). This is consistent with what is implemented in the MPP estimate update stage.

MPP estimate update

Here the following application takes place

$$\mathbf{x}_1^c = C\mathbf{x}_1^+,$$

Thus:

$$\begin{aligned}
\mathbf{x}_1^c = \begin{bmatrix} \left(\dfrac{\Delta p}{\Delta v}\right)_1^c \\ v_{\text{MPP},1}^c \end{bmatrix} = C\mathbf{x}_1^+ = \begin{bmatrix} 1 & 0 \\ 0.05 & 1 \end{bmatrix} \begin{bmatrix} \left(\dfrac{\Delta p}{\Delta v}\right)_1^+ \\ v_{\text{MPP},1}^+ \end{bmatrix} \\
= \begin{bmatrix} \left(\dfrac{\Delta p}{\Delta v}\right)_1^+ \\ 0.05 \left(\dfrac{\Delta p}{\Delta v}\right)_1^+ + v_{\text{MPP},1}^+ \end{bmatrix} = \begin{bmatrix} 8.57 \\ 0.05 \cdot 8.57 + 10 \end{bmatrix} = \begin{bmatrix} 8.57 \\ 10.429 \end{bmatrix}
\end{aligned}$$

As expected, the application of the matrix C does not affect the Kalman filter computation within the same iteration, since the first element of the state vector remains unchanged between $\mathbf{x}_1^c$ and $\mathbf{x}_1^+$. Conversely, the second element (the MPP voltage estimation) is updated and begins increasing in order to converge toward the true v_{MPP}.

Control stage

The final step consists of updating the operating voltage of the panel. The rule in eq. (4) is applied as follows:

$$v_{\text{op},1} = v_{\text{op},0} + 0.5\left(v_{\text{MPP},1}^c - v_{\text{op},0}\right) = 9 + 0.5(10.429 - 9) = 9.714 \text{ V}.$$

This value represents the actual voltage that the DC-DC converter will impose on the PV panel during the next sampling interval.

Second iteration ($k = 2$)

Like the previous iteration, here are three steps to be performed:

1. *Kalman estimation*: computation of the updated state estimate $\mathbf{x}_2^+$. The main goal is to calculate the quantity $\left(\frac{\Delta p}{\Delta v}\right)_2^+$ (first component of the state vector) to update the estimate of the MPP voltage.

2. *MPP estimate update*: computation of the estimate of the MPP voltage $\mathbf{x}_2^c = C\,\mathbf{x}_2^+$, from which the quantity $v_{\mathrm{MPP},2}^c$ (second component of the state vector) is extracted and used as reference for the PV panel operating voltage.

3. *Control update*: computation of the operating voltage that the DC-DC converter should impose to the panel
$$v_{\mathrm{op},2} = v_{\mathrm{op},1} + \beta\big(v_{\mathrm{MPP},2}^c - v_{\mathrm{op},1}\big).$$

Kalman estimation

Prediction

The predicted state and covariance are obtained as:

$$\mathbf{x}_2^- = \mathbf{x}_1^c = \begin{bmatrix} 8.57 \\ 10.429 \end{bmatrix}.$$

The predicted error covariance is:

$$P_2^- = P_1^+ + Q = \begin{bmatrix} 0.0477 & 0 \\ 0 & 1.001 \end{bmatrix} + \begin{bmatrix} 10^{-3} & 0 \\ 0 & 10^{-3} \end{bmatrix} = \begin{bmatrix} 0.0487 & 0 \\ 0 & 1.002 \end{bmatrix}.$$

Measurement

Like the previous iteration, the noisy measurement of the power gradient is obtained by considering two close operating points around the current operating voltage $v_{\mathrm{op},1}$ ($\Delta v = 0.2\,\mathrm{V}$ as in the previous iteration):

$$v_a = v_{\mathrm{op},1} = 9.714\ \mathrm{V}, \qquad v_b = v_{\mathrm{op},1} + \Delta v = 9.914\ \mathrm{V}.$$

The power at the two operating points is computed using the same photovoltaic model:

$$p(v) = v\,i(v), \qquad i(v) = i_{\mathrm{SC}}\Big(1 - (v/v_{\mathrm{OC}})^n\Big),$$

with $i_{\mathrm{SC}} = 8\,\mathrm{A}$, $v_{\mathrm{OC}} = 38\,\mathrm{V}$ and $n = 9$. Evaluating the model yields:

$$p_a = p(v_a) \approx 77.71\ \mathrm{W}, \qquad p_b = p(v_b) \approx 79.31\ \mathrm{W}.$$

As before, Gaussian measurement noise with variance $\sigma_R^2 = 0.05$ is added to both power samples:

$$p_a^{\mathrm{meas}} = p_a + \eta_a, \qquad p_b^{\mathrm{meas}} = p_b + \eta_b, \qquad \eta_a, \eta_b \sim \mathcal{N}(0, \sigma_R^2).$$

The resulting noisy measurement of the power gradient at iteration $k = 2$ is therefore:

$$y_2 = \frac{p_b^{\mathrm{meas}} - p_a^{\mathrm{meas}}}{\Delta v} = \frac{p_b - p_a}{0.2} + \frac{\eta_b - \eta_a}{0.2}.$$

Since $(p_b - p_a)/0.2 \approx 8.0\,\mathrm{W/V}$, and assuming a noise realization $\eta_b - \eta_a \approx 0.1$, the resulting measurement is:

$$y_2 \approx 8.0 + \frac{0.1}{0.2} = 8.5 \text{ W/V}.$$

Update

The measurement residual (innovation) is given by:

$$\tilde{y}_2 = y_2 - H\,\mathbf{x}_2^- = 8.5 - \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}8.57\\10.429\end{bmatrix} = -0.07.$$

The innovation covariance is:

$$S_2 = H\,P_2^- H^\top + \sigma_R^2 = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}0.0487 & 0\\0 & 1.002\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} + 0.05 = 0.0487 + 0.05 = 0.0987.$$

The Kalman gain is then computed as:

$$K_2 = P_2^- H^\top S_2^{-1} = \begin{bmatrix}0.0487 & 0\\0 & 1.002\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix}\frac{1}{0.0987} = \frac{1}{0.0987}\begin{bmatrix}0.0487\\0\end{bmatrix} \approx \begin{bmatrix}0.4934\\0\end{bmatrix}.$$

The updated state estimate is:

$$\mathbf{x}_2^+ = \mathbf{x}_2^- + K_2\tilde{y}_2 \approx \begin{bmatrix}8.57\\10.429\end{bmatrix} + \begin{bmatrix}0.4934\\0\end{bmatrix}(-0.07) = \begin{bmatrix}8.57 - 0.0345\\10.429\end{bmatrix} \approx \begin{bmatrix}8.5355\\10.429\end{bmatrix}.$$

Finally, the updated error covariance is:

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

where

$$I - K_2 H = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} - \begin{bmatrix}0.4934\\0\end{bmatrix}\begin{bmatrix}1 & 0\end{bmatrix} = \begin{bmatrix}1 - 0.4934 & 0\\0 & 1\end{bmatrix} \approx \begin{bmatrix}0.5066 & 0\\0 & 1\end{bmatrix}.$$

Thus,

$$P_2^+ \approx \begin{bmatrix}0.5066 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}0.0487 & 0\\0 & 1.002\end{bmatrix} \approx \begin{bmatrix}0.0247 & 0\\0 & 1.002\end{bmatrix}.$$

Like in the previous iteration, it is worth noting that $p_{11,\,2}^+ \approx 0.0247$ is far smaller than the initial value $p_{11,\,1}^+ = 0.0477$, indicating that the estimation has become significantly more confident.

Moreover, it can be observed that the second diagonal element of the covariance matrix, $p_{22,\,2}^+$, increases with respect to its initial value. This behaviour is expected: since v_{MPP} is not observable through the measurement equation, the Kalman filter is unable to reduce the uncertainty associated with this state. As a consequence, the corresponding covariance grows over time, reflecting the increasing uncertainty of an unobservable variable.

It is important to note that this effect does not compromise the stability or the performance of the algorithm, because the estimation of v_{MPP} is not directly performed by the Kalman filter but by the subsequent MPP estimate update stage. For this reason, in the context of this numerical example, the values of $p_{22,0}^+$ and q_{22} could even be set to zero without altering the behaviour of the MPPT algorithm.

Again, it can be observed that the Kalman filter does not update the second component of the state vector, since $v_{\mathrm{MPP},2}^+ = v_{\mathrm{MPP},1}^c = 10.429$. This is consistent with the structure of the MPP estimate update stage.

MPP estimate update

Here the following application takes place

$$\mathbf{x}_2^c = C\mathbf{x}_2^+,$$

Thus:

$$\mathbf{x}_2^c = \begin{bmatrix} \left(\dfrac{\Delta p}{\Delta v}\right)_2^c \\ v_{\text{MPP},2}^c \end{bmatrix} = C\mathbf{x}_2^+ = \begin{bmatrix} 1 & 0 \\ 0.05 & 1 \end{bmatrix} \begin{bmatrix} \left(\dfrac{\Delta p}{\Delta v}\right)_2^+ \\ v_{\text{MPP},2}^+ \end{bmatrix}$$
$$= \begin{bmatrix} \left(\dfrac{\Delta p}{\Delta v}\right)_2^+ \\ 0.05\left(\dfrac{\Delta p}{\Delta v}\right)_2^+ + v_{\text{MPP},2}^+ \end{bmatrix} = \begin{bmatrix} 8.5355 \\ 0.05 \cdot 8.5355 + 10.429 \end{bmatrix} = \begin{bmatrix} 8.5355 \\ 10.856 \end{bmatrix}$$

As expected, the application of the matrix C does not affect the Kalman filter computation within the same iteration, since the first element of the state vector remains unchanged between $\mathbf{x}_2^c$ and $\mathbf{x}_2^+$. Conversely, the second element (the MPP voltage estimate) is updated and begins increasing in order to converge toward the true v_{MPP}.

Control stage

The final step consists of updating the operating voltage of the panel. The rule in eq. (4) is applied as follows:

$$v_{\text{op},2} = v_{\text{op},1} + 0.5\left(v_{\text{MPP},2}^c - v_{\text{op},1}\right) = 9.714 + 0.5(10.856 - 9.714) = 10.285 \text{ V}.$$

This value represents the actual voltage that the DC-DC converter will impose on the PV panel during the next sampling interval.

Subsequent iterations ($k \geq 3$)

The subsequent iterations proceed in exactly the same way as the first two. As k increases, the operating voltage steadily moves toward the true maximum power point voltage. The combination of the Kalman-based gradient estimation and the MPP–estimate update ensures convergence even in the presence of measurement noise.

A complete numerical simulation, implementing a larger number of iterations and showing the full convergence behaviour of the algorithm, is reported in Appendix A, where a Python script is provided to reproduce and extend this example.

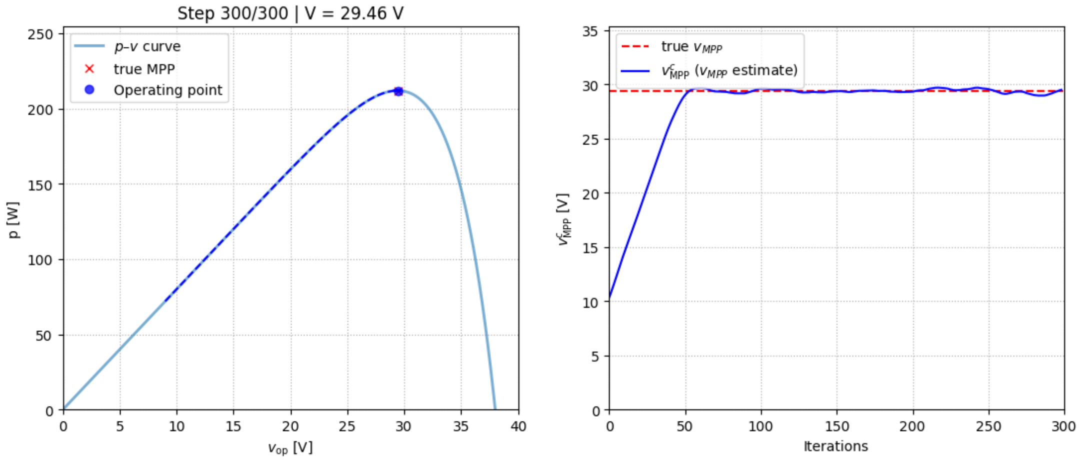


Figure 9: Operation of the Kalman-based MPPT algorithm

5 Conclusions

In this tutorial, the complete path has been explored from understanding the electrical behaviour of a photovoltaic panel to implementing an advanced maximum power point tracking (MPPT) algorithm. The focus was not only on reviewing traditional techniques such as Perturb & Observe and Incremental Conductance, but also on introducing a more modern perspective: the use of a Kalman filter to dynamically estimate both the power gradient and the location of the maximum power point.

This method relies on a simple yet powerful principle: although the MPP cannot be measured directly, one can indirectly observe the effect that small variations of the voltage produce on the instantaneous power. Treating both the power gradient $\frac{dp}{dv}$ and the optimal voltage v_{MPP} as hidden states to be estimated allows the filter to adapt in the presence of noise, irradiance fluctuations, measurement ripple, and the inherently slow dynamics of the system. The Kalman MPPT does not merely react to local variations of the power; instead, it builds a coherent, filtered, and predictive estimate of the MPP trajectory over time.

The Python code provided, together with the numerical examples, has illustrated step by step how this approach can be implemented in a transparent manner. It has been shown how the filter iteratively updates its estimates based on the incoming measurements, how the covariance progressively decreases as the system gains confidence in the parameters, and how the operating voltage is smoothly regulated towards the MPP with a stable behaviour and without excessive oscillations.

In conclusion, the use of the Kalman filter for MPPT represents a significant step toward more intelligent and reliable control systems, especially in real-world scenarios where environmental conditions are far from ideal. With a proper implementation, this method makes it possible to systematically extract the maximum available power from the photovoltaic panel, making it a promising choice for advanced applications in modern solar energy systems.

A Python program

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import animation
from IPython.display import HTML
# ============================================================
# PV MODEL
# ============================================================
def pv_model(V, Isc=8, Voc=38.0, n=9):
    V = np.atleast_1d(V).astype(float)
    V = np.clip(V, 0, Voc)
    I = Isc * (1 - (V / Voc) ** n)
    I[I < 0] = 0
    P = V * I
    return I, P

# ============================================================
# KALMAN FILTER MPPT
# ============================================================
class KalmanMPPT:

    def __init__(self):
        self.x = np.zeros((2, 1))
        self.P = np.eye(2)
        self.A = np.eye(2)
        self.H = np.array([[1.0, 0.0]])
        self.Q = np.eye(2) * 1e-3
        self.R = np.array([[5e-2]])

    def predict(self):
        self.x = self.A @ self.x
        self.P = self.A @ self.P @ self.A.T + self.Q

    def update(self, z):
        z = np.array([[z]], dtype=float)
        y = z - self.H @ self.x
        S = self.H @ self.P @ self.H.T + self.R
        K = self.P @ self.H.T @ np.linalg.inv(S)
        self.x = self.x + K @ y
        self.P = (np.eye(2) - K @ self.H) @ self.P


# ============================================================
# SIMULATION
# ============================================================
def simulate(Isc=8.0, Voc=38.0, n=9):
    Vplot = np.linspace(0, Voc, 600)
    _, Pplot = pv_model(Vplot, Isc, Voc, n)
    idx_mpp = np.argmax(Pplot)
    Vtrue_mpp, Ptrue_mpp = Vplot[idx_mpp], Pplot[idx_mpp]

    #Initialization and parametrization
    Voper = 9
    V_MPP_est_0 = 10

    #Kalman filter initialization
    kf = KalmanMPPT()
```

```
    power_meas_noise_variance = 0.05
    kf.x[1, 0] = V_MPP_est_0
    alpha = 0.05 #"learning rate" in the v_MPP estimation
    #matrix to apply for the v_MPP estimation
    C = np.array([[1.0, 0.0],
              [alpha, 1.0]], dtype=float)

    beta = 0.5
    #Voltage variation for gradient estimation
    delta_v = 0.2

    #Initialization of the arrays of operating voltage, power at the operating voltage and MPP voltage estimation respectilvely
    Vhist, Phist, Vest_hist = [], [], []

    #Simulation run for 300 iterations
    for _ in range(300):

        _, P = pv_model(Voper, Isc, Voc, n)
        _, Pp = pv_model(Voper + delta_v, Isc, Voc, n)

        #power measurement with noise
        P_noise = P + np.random.normal(0, np.sqrt(power_meas_noise_variance))
        Pp_noise = Pp + np.random.normal(0, np.sqrt(power_meas_noise_variance))

        #Gradient computation
        dPdV = (Pp_noise - P_noise) / (delta_v)

        #store the voper before new update (for chart population)
        Voper_prev = Voper

        #Kalman predict
        kf.predict()

        #Kalman update
        kf.update(float(dPdV))

        # MPP estimate update (system state by applying the matrix C)
        kf.x = C @ kf.x

        Voper = Voper + beta * (kf.x[1, 0] - Voper)

        Voper = float(np.clip(Voper, 0, Voc))

        Vhist.append(Voper_prev)

        Phist.append(float(P[0]))

        Vest_hist.append(float(kf.x[1, 0]))

    return Vplot, Pplot, Vhist, Phist, Vest_hist, Vtrue_mpp, Ptrue_mpp


# ==========================================================
# ANIMATION (P-V curve) + FIGURE 9 (V_est vs V_mpp)
# ==========================================================
```

```
Vplot, Pplot, Vhist, Phist, Vest_hist, Vtrue_mpp, Ptrue_mpp = simulate()

# --- Figure 1: P-V with operation point ---
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(13, 5))
ax1.plot(Vplot, Pplot, label="-pv curve", lw=2,alpha=0.6)
ax1.plot([Vtrue_mpp], [Ptrue_mpp], "rx", label="true MPP")
pt, = ax1.plot([], [], "bo", label="Operating point",alpha=0.75)
trace, = ax1.plot([], [], "b--")
ax1.set_xlim(0, 40)
ax1.set_ylim(0, np.max(Pplot) * 1.2)
ax1.set_xlabel(r"$v_{\mathrm{op}}$ [V]")
ax1.set_ylabel("p [W]")
ax1.legend()
ax1.grid(True, ls=':')

# --- Figure 2: V_est vs V_mpp (similar to figure 9 in the tutorial) ---
ax2.plot([Vtrue_mpp]*len(Vhist), 'r--', label=r"true $v_{MPP}$")
line_vest, = ax2.plot([], [], 'b', label=r"$v_{\mathrm{MPP}}^{c}$ ($v_{MPP}
    $ estimate)")
ax2.set_xlim(0, len(Vhist))
ax2.set_ylim(0, 1.2*Vtrue_mpp)
ax2.set_xlabel("Iterations")
ax2.set_ylabel(r"$v_{\mathrm{MPP}}^{c}$ [V]")
ax2.legend()
ax2.grid(True, ls=':')

def init():
    pt.set_data([], [])
    trace.set_data([], [])
    line_vest.set_data([], [])
    return pt, trace, line_vest

def update(i):
    v, p = Vhist[i], Phist[i]
    pt.set_data([v], [p])
    trace.set_data(Vhist[:i], Phist[:i])
    line_vest.set_data(np.arange(i), Vest_hist[:i])
    ax1.set_title(f"Step {i+1}/{len(Vhist)} | V = {v:.2f} V")
    return pt, trace, line_vest

ani = animation.FuncAnimation(fig, update, init_func=init,
frames=len(Vhist), interval=80, blit=True)
HTML(ani.to_jshtml())
```