# Maximum Power Point Tracking (MPPT) algorithms

TN117 | Posted on March 25, 2021 | Updated on June 24, 2025

Nicolas CHERIX
Head of Engineering
imperix • in

Table of Contents

Maximum Power Point Tracking is a family of control algorithms that aims at optimizing the use of a power source that possesses a fluctuating power profile.

Indeed, some power sources, like solar panels, present power characteristics that strongly depend on the operating conditions. For instance, the cloud coverage significantly impacts the capability of a panel to deliver electricity. As such, maximizing the extracted power requires identifying – and tracking – the operating point that provides the highest power level as a function of the operating conditions.

Therefore, Maximum Power Point Tracking (MPPT) is often applied in renewable energy systems – e.g. photovoltaic plants or wind turbines – as their power delivery capability varies significantly and in an unpredictable manner. Other special operating points may be interesting to track, such as the maximum efficiency point tracking (MEPT), or other optimum, e.g. related to operating costs.

## Software resources

Maximum Power Point Tracking algorithm only (Simulink model)Download

TN117_PV_inverter_SimulinkDownload

## Principles of operation

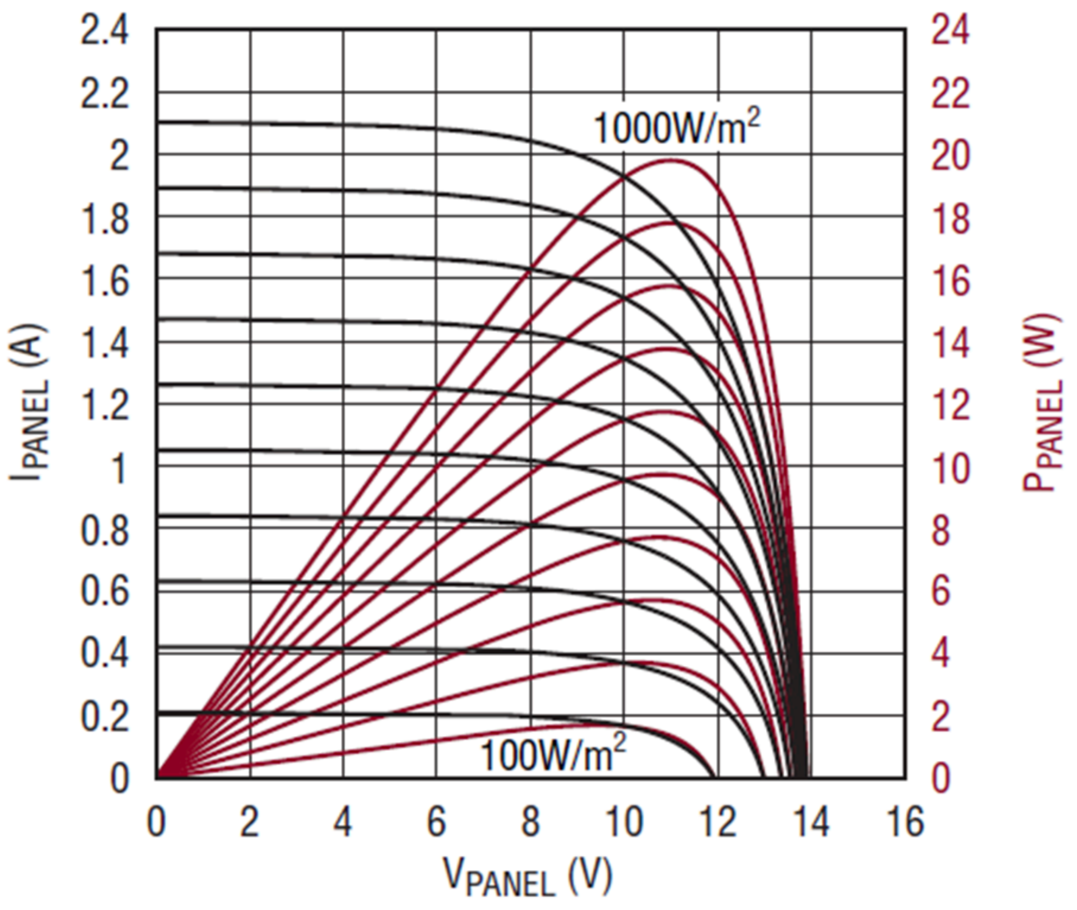
For practically all real power sources, the power that can be extracted varies with the operating point. While electrical sources are related to the voltage/current pair, the same principle also applies to force/speed, flux/surface, etc.

In all cases, the inevitable internal resistance (or equivalent quantity) limits the maximum possible output power. Non-linear or more complex characteristics also exist, but with the same result: the maximum power point is not located at the [max. voltage · max. current] point (or equivalent quantity). Therefore, the operating point that delivers the maximum power must be constantly tracked by searching for the best voltage · current combination.

## Photovoltaic solar panel example

For instance, photovoltaic panels (PV panels) possess a well-known output characteristic, featuring an internal resistance that quickly decreases close to the open-circuit voltage (assuming a current source model).

This results in a bump-shaped power-voltage characteristic, whose top is typically located between 60-80% of the open-circuit voltage. This point is however not fixed but varies with the output current, which depends itself on the temperature and irradiance, i.e. the operating conditions of the PV cells themselves.



Typical IV-PV curves of a solar panel

In some cases (such as, to some extent, photovoltaic systems), the output characteristics (here I-V) are relatively well-known and precise, such that they can be used to locate the maximum power point using look-up tables. This is however not the case for all systems, motivating the use of more empirical approaches.

## Maximum Power Point Tracking algorithms

To date, numerous maximum power point tracking algorithms have been proposed, with various trade-offs between performance (tracking speed, accuracy) and complexity (need for sensors, mathematical modeling, computation burden, etc.).

## MPPT example for solar inverter

## P&O MPPT algorithm basics

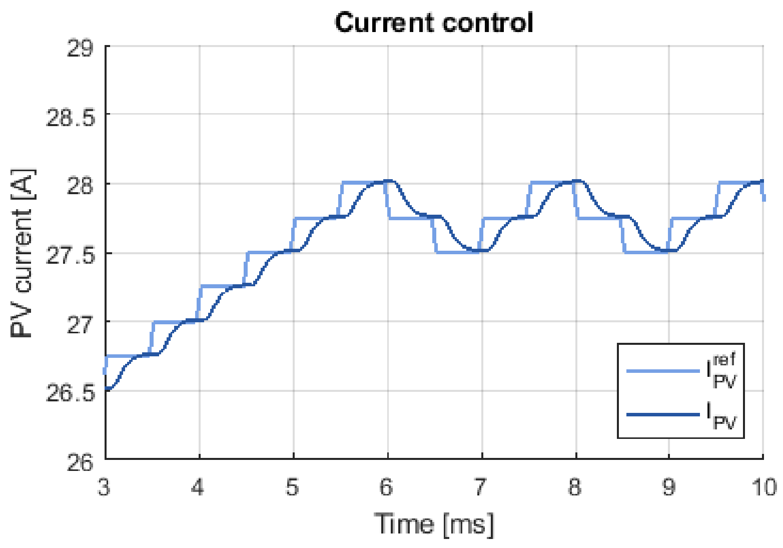
Among other possible algorithms, the **Perturb and Observe** (P&O) tracking algorithm actively varies the current set-point – i.e. adds a small perturbation – and observes the corresponding impact on the output power. Depending on whether that perturbation tends to increase or decrease the output power, the current setpoint is respectively increased or decreased accordingly.

In other words, if adding a small ∆I to the current setpoint increases the resulting output power, then the subsequent current setpoint is further increased. Reciprocally, if a positive ∆I tends to reduce the output power, then the subsequent current reference is reduced by ∆I.

The entirely empirical approach of the *Perturb and Observe* algorithm requires that:

- The requested setpoint is correctly followed, meaning that the current control (if any) is not saturated and can reach steady state before the result is evaluated.
- There is only one global maximum power point.

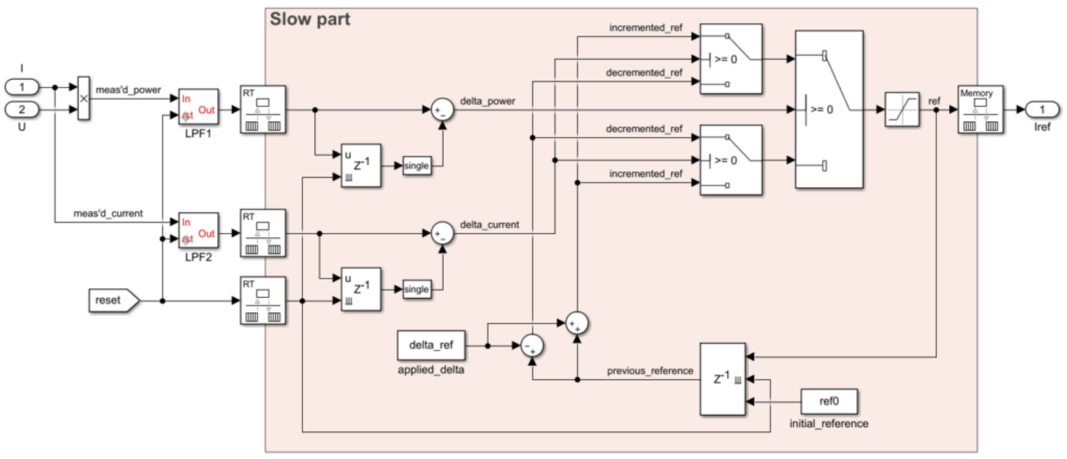
As such, this maximum power point tracking algorithm is designed for use as part of a discretized process that is slower than the current control dynamics. This can typically be implemented using a multi-rate technique, where the current control is executed within the main control interrupt (fast control loop) and the MPPT algorithm executed within a secondary control interrupt (slow control loop).



Typical execution of a Pertub and Observe MPPT

## Maximum Power Point Tracking with Simulink

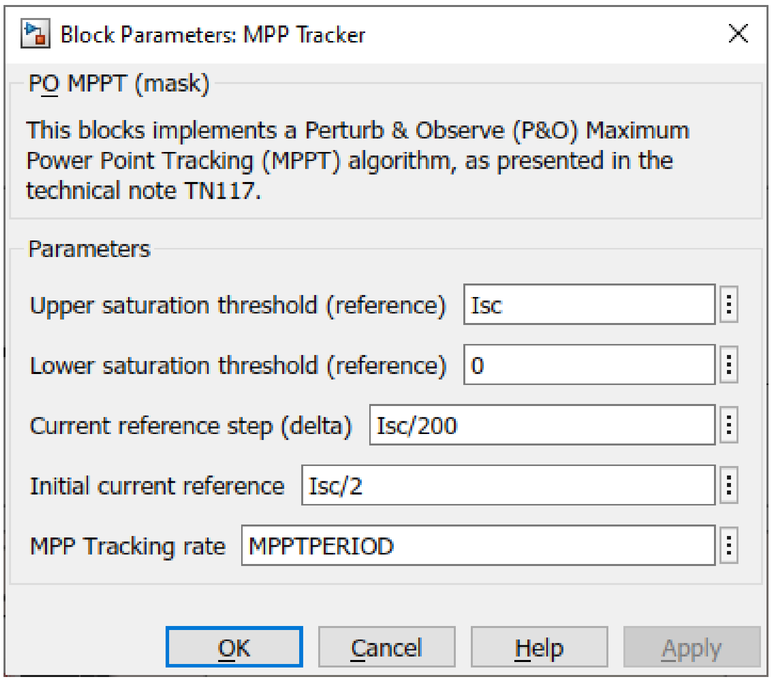
The proposed maximum power point tracking algorithm can be implemented as shown below. It requires the introduction of a slower control rate for the MPPT itself. The management of multiple control rates within Simulink is further explained in Multi-rate control with Simulink (PN145).



Maximum power point tracking implementation overview in Simulink

The corresponding mask requires the following parameters:

- `Upper saturation threshold`: Maximum output value.
- `Lower saturation threshold`: Minimum output value.
- `Current reference step (delta)`: Current increment added (or subtracted) for the previous setpoint.
- `Initial current reference`: Initial value at startup.
- `MPP Tracking rate`: The control period used for the execution of the MPPT algorithm (here 10x slower than the main control interrupt rate).


Block Parameters: MPP Tracker

PO MPPT (mask)

This blocks implements a Perturb & Observe (P&O) Maximum Power Point Tracking (MPPT) algorithm, as presented in the technical note TN117.

Parameters

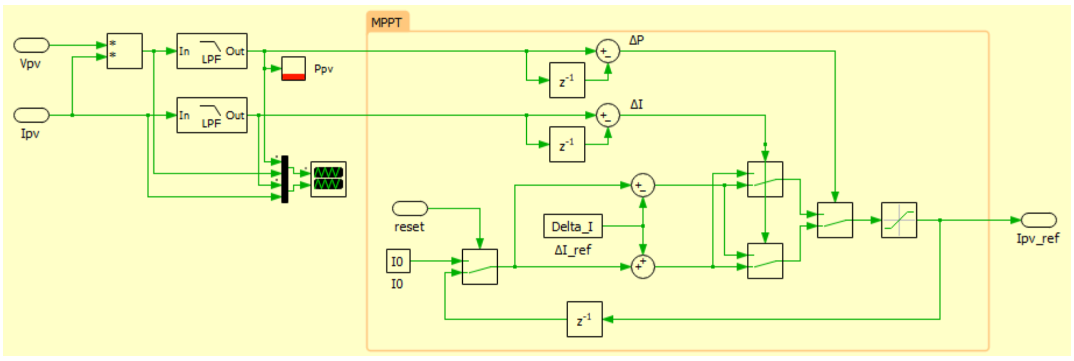
| | |
|---|---|
| Upper saturation threshold (reference) | Isc |
| Lower saturation threshold (reference) | 0 |
| Current reference step (delta) | Isc/200 |
| Initial current reference | Isc/2 |
| MPP Tracking rate | MPPTPERIOD |

imperix P&O MPPT configuration on Simulink

## MPPT with PLECS

In PLECS, the proposed maximum power point tracking algorithm can be implemented as shown below. As in Simulink, it requires the introduction of a slower control rate for the

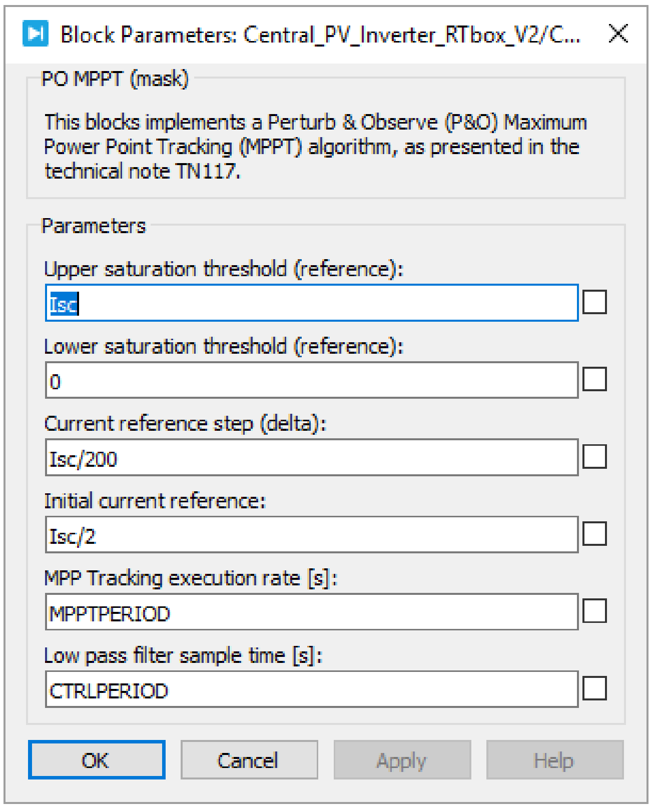
MPPT itself. The management of multiple control rate within PLECS is further explained in [Multi-rate control with PLECS (PN155)](#).



Maximum power point tracking implementation overview in PLECS

The corresponding mask requires the following parameters:

- `Upper saturation threshold`: Maximum output value.
- `Lower saturation threshold`: Minimum output value.
- `Current reference step (delta)`: Current increment added (or subtracted) for the previous setpoint.
- `Initial current reference`: Initial value at startup.
- `MPP Tracking rate`: The control period used for the execution of the MPPT algorithm (here 10x slower than the main control interrupt rate).
- `Low pass filter sample time`: The sample time used for the average of the PV voltage and current. It should be the interruption period.



Block Parameters: Central_PV_Inverter_RTbox_V2/C...

PO MPPT (mask)

This blocks implements a Perturb & Observe (P&O) Maximum Power Point Tracking (MPPT) algorithm, as presented in the technical note TN117.

Parameters

Upper saturation threshold (reference): Isc

Lower saturation threshold (reference): 0

Current reference step (delta): Isc/200

Initial current reference: Isc/2

MPP Tracking execution rate [s]: MPPTPERIOD

Low pass filter sample time [s]: CTRLPERIOD

imperix P&O MPPT configuration in PLECS

# Maximum Power Point Tracking with C/C++ code

The imperix CPP SDK provides pre-written routines for maximum power point tracking. As for control-related routines, the MPPT is based on:

- A pseudo-object `MPPTracker`, which contains pre-computed parameters as well as state variables.
- A configuration function, meant to be called during `UserInit()`, named `ConfigMPPTracker()`.
- A run-time function, meant to be called during a slow routine, such as `SlowSubTask()`, named `RunMPPTracker()`.

Suitable C code for the implementation of the *Perturb and Observe* MPPT algorithm is given below. The necessary parameters are documented within the controller.h header file.

```
#include "user.h"
#include "../API/controllers.h"         // Discrete-time controllers

MPPTracker string1_mppt, string2_mppt;  // Maximum Power Point Trackers
PIDController Ipv_reg;                  // Controller for the PV current

tUserSafe UserBackground();
void SlowSubTask();
float SubTaskTimer = 0.0;
bool SubTaskFlag = false;

USER_SAFE UserInit(void)
{
  // Configure the main timebases on CLOCK_0:
  Clock_SetPeriod(CLOCK_0, (int)(SWITCHING_FREQUENCY));

  // Configure the interrupts:
  ConfigureMainInterrupt(UserInterrupt, CLOCK_0, 0.5);
  RegisterBackgroundCallback(UserBackground);

  // Configure the PI controllers:
  ConfigPIDController(&Ipv1_reg, 12.0, 0.3, 0.0, 60, -60, SAMPLING_PERIOD, 10);

  // Configure and initialize the MPP-trackers:
  #define increment 0.01
  ConfigMPPTracker(&mymppt, increment, 7.0, 16.0, 1.5, 0.01);

  return SAFE;
}

tUserSafe UserInterrupt(void)
{
  // Measure all the necessary quantities:
  Upv = Adc_GetValue(8); // Voltage on the PV string
  Ipv = -Adc_GetValue(0); // PV current

  // Execute the current controllers on the MPPT string:
  Epv = Upv - RunPIController(&Ipv_reg, Ipv_ref - Ipv);
  CbPwm_SetDutyCycle(PWM_CHANNEL_3, Epv/Udc);
```

```cpp
  // Compute the power drawn from the PV panel:
  #define k_iir_lpf 0.05
  Ppv = k_iir_lpf* (Upv*Ipv) + (1.0-k_iir_lpf)* Ppv;

  SubTaskTimer += SWITCHING_PERIOD;
  if(SubTaskTimer >= MPPTPERIOD){
    SubTaskTimer = SubTaskTimer - MPPTPERIOD;
    SubTaskFlag  = true;
  }

  return SAFE;
}

tUserSafe UserBackground()
{
  if(SubTaskFlag)
  {
    SlowSubTask();
    SubTaskFlag = false;
  }
  return SAFE;
}

void SlowSubTask()
{
  // When appropriate, execute the MPPT algorithm:
  if (enable_MPPT==1){
    // Run the Maximum Power Point Tracking (MPPT) algorithm:
    Ipv_ref = RunMPPTracking(&string_mppt, Ipv, Ppv);
  }
  else{
    // Leave the setpoints unaltered
  }
}
```
Code language: C++ (cpp)

## Academic references

[1] T. Esram and P. L. Chapman, “Comparison of Photovoltaic Array Maximum Power Point Tracking Techniques,” in IEEE Transactions on Energy Conversion, June 2007.