

2023 年 3 月 23 日

まつやま Re・再来館設置システムの実測データの時刻修正に関する研究の中間報告

祖父江匠真

1 概要

松山市教育委員会によって小学校などに設置されている太陽発電計測システムの PC は、まつやま Re・再来館に設置されているものを除くと、インターネットに接続されていないオフライン環境で計測が行われている。このため、PC の内部時計が経過時間とともにずれており、実測データのタイムスタンプが不正な値になっている。

そこで、タイムスタンプの修正を行うために、実測データと、太陽光発電シミュレーターにより生成された理論データとの、相互相関関数を用いた時刻合わせプログラムの実装を行っている。

その時刻合わせプログラムを実行した際に、結果として得られるずれ時間の推定結果が正しいことを確認するため、タイムスタンプが正確な実測データを用いてずれ時間の推定結果が 0 秒となることを確認する作業を現在行っている。

まつやま Re・再来館に設置されている計測システムはインターネットに接続されており、さらにシステムを実行しているマイコンを 24 時間ごとに再起動しているため、マイコン内部の時計は常にインターネット時刻に同期されており、計測データのタイムスタンプは正確である。

本報告では、まつやま Re・再来館の実測データを入力として時刻合わせプログラムを実行した際の、ずれ時間の推定結果について述べる。

2 相互相関とは

相互相関は、2 つの信号の類似性を測定するために用いられる手法の一つである。相互相関を用いることで、2 つの信号の類似度を示す数値を算出し、信号間の関係を調査することができる。相互相関値が高ければ高いほど、2 つの信号は互いに類似していると言える。

相互相関を用いた信号間のずれの検出方法は以下の手順で行われる [1]。

1. 2 つの異なる雑音信号（信号 A および信号 B とする）を用意する。ただしこれらの信号の雑音が大きい時間は同じ長さである。

2. 信号 A と信号 B をずらしながら重ね合わせる. このずらしの大きさを「ラグ時間」と称する. ラグ時間が 0 である場合, 信号 A と信号 B は完全に一致する状態である. ラグ時間が正である場合, 信号 B は信号 A より遅れていることを示し, ラグ時間が負である場合はその逆である.
3. それぞれのラグ時間において, 信号 A と信号 B が重なる部分の積の和 (内積) を算出する. これを相互相関と呼ぶ.
4. それぞれのラグ時間ごとに相互相関を計算し, 相互相関が最大となる際のラグ時間を特定する. この際に得られたラグ時間は, 信号 A と信号 B のずれ時間を示している.

例として, 図 1 に示す 2 つの信号 sig1, sig2 に対して相互相関を計算した場合, sig2 を右に 800 秒スライドさせた際に相互相関が最大となるため, これらのラグ時間は 800 秒であると計算される. 信号 fit は sig2 を 800 秒分右へスライドさせたものと sig1 を重ねてプロットしたものであり, 相互相関によってずれ時間が正しく推定出来ていることが分かる.

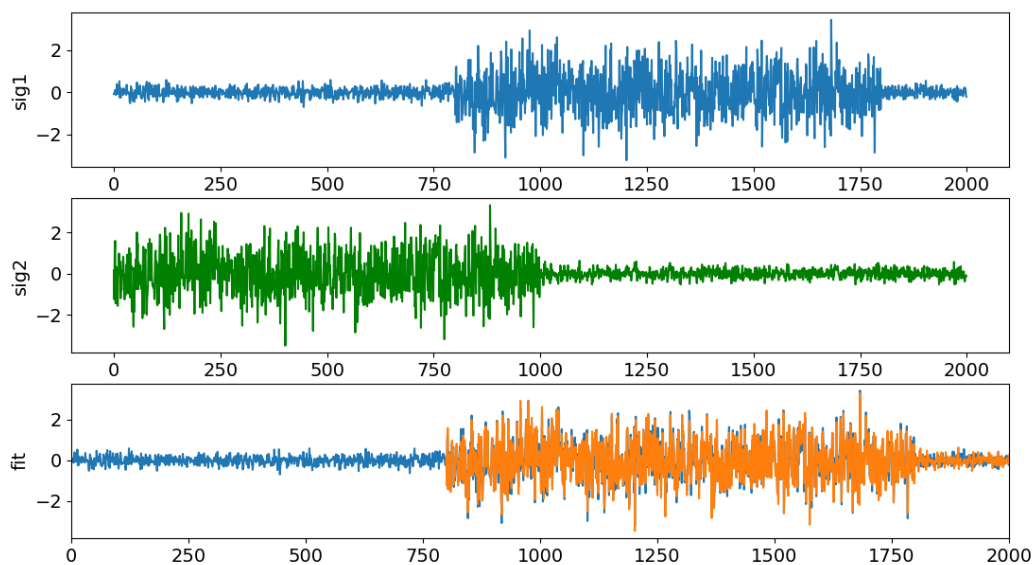


図 1: 相互相関に使用した信号とずれ時間分スライドさせて重ねたもの

3 理論データのシミュレーション方法について

理論データのシミュレーションは, 太陽光発電システムのシミュレーションを行うための Python ライブラリである Pylib を用いて行う [2].

Pylib を用いることで緯度, 経度, 標高, タイムゾーン, 日時などを入力として, 快晴時の地表に到達する日射量を推定することができる.

例として, 2022 年 4 月 8 日に計測された実測データを取り上げ, それに対応する理論データを Pylib で計算して合わせてプロットすることで, Pylib の予測精度がどの程度のものか視覚的に説明する.

図 2 に示すのは 2022 年 4 月 8 日に松山 Re・再来館で計測された実測データである.

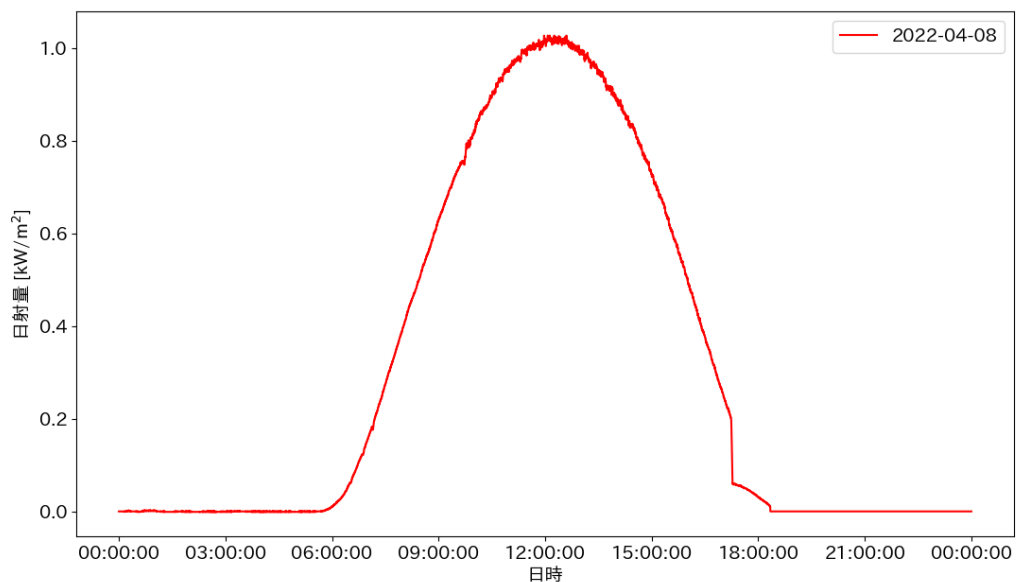


図 2: 2022 年 4 月 8 日の実測データ

図 2 の実測データのタイムスタンプ列を Pylib に入力し, シミュレーションした結果を図 3 に示す.

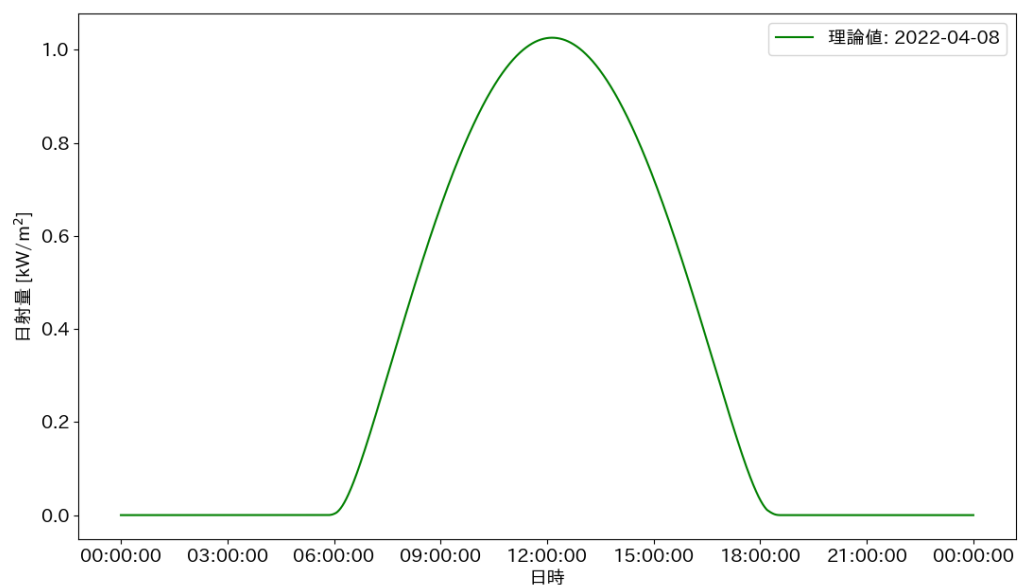


図 3: 2022 年 4 月 8 日の実測データに対応する理論データ

実測データと理論データを合わせてプロットしたものを, 図 4に示す.

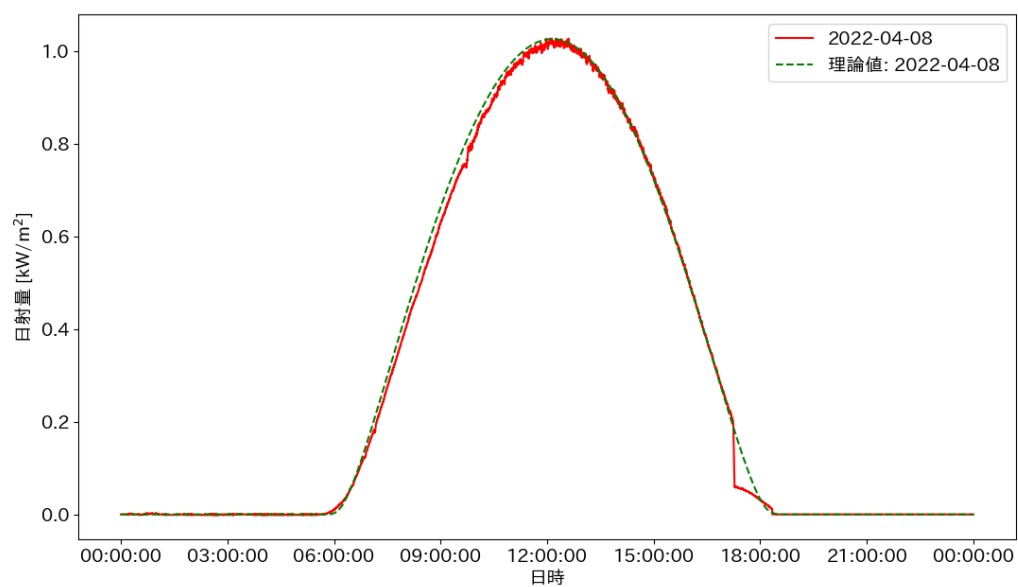


図 4: 2022 年 4 月 8 日の実測データと対応する理論データ

このように快晴の日の実測データに対する理論データであれば、午後の概形は概ね一致する。午前の概形も一致させるには pvlib の太陽光パネルが水平面から傾いている角度と、太陽光パネルが真北から時計回りにどの方向に向いているかを表す角度を調整すればよいが、月日ごとに再調整する必要がある。

4 時刻合わせプログラムのアルゴリズムに関する検討

現在実装されている時刻合わせプログラムでは、以下のアルゴリズムで実測データと理論データのずれ時間を推定している。

1. まず実測データを読み込み、データのサンプリング間隔を 1 秒ごとの等間隔になるよう numpy の interp 関数を使用して線形補間を行うなど、前処理を実施する。
2. 次に実測データのタイムスタンプ情報を Pvlib に入力し、晴天の日における松山 Re・再来館の日射量の理論値を計算する。
3. 最後に、これらの実測データと理論データを用いて相互相関関数を計算し、最大相関値を持つインデックスを特定する。このインデックスが二つのデータのずれ時間を示し、インデックスに対応する値がラグ時間となる。

例えば、2022 年 4 月 8 日の実測データに対してずれ時間を計算した場合、理論データに対して実測データを 165 秒進めたときに相互相関が最大となり、ずれ時間の推定結果は-165 秒となった。したがって本手法の計算精度は ± 165 秒程度と言える。

時刻合わせプログラムで行う前処理では、実測データのリサンプリングの他に、実測データのフィルタリングや正規化などが行われる。この前処理の組み合わせ方によって相互相関によるずれ時間の計算結果が改善するため、様々な前処理の組み合わせを試みており、それぞれの組み合わせにおける相互相関の計算結果について以下に述べる。

4.1 指定時間帯外の日射量を 0 に置き換える実測データの前処理

2022 年 4 月 8 日の実測データを観察すると、日没前に日射量が急激に低下する箇所が存在する。

これは、おそらく松山 Re・再来館の周辺にあるマンション等の建物によって太陽光が遮られることが原因と考えられる。このような実測データと理論データで日射量が大きく異なる箇所が相互相関の計算結果に影響を与えていると推測し、そのような箇所を実測データから除去する前処理を実施した。

図 5 は、2022 年 4 月 8 日の実測データと理論データの日射量が大きく異なる時間帯に限定してフィルタリングを行い、相互相関を計算したものである。

このフィルタリングでは, 図 5のように指定時間帯外の実測データと理論データの
の日射量を 0 に置き換えている.

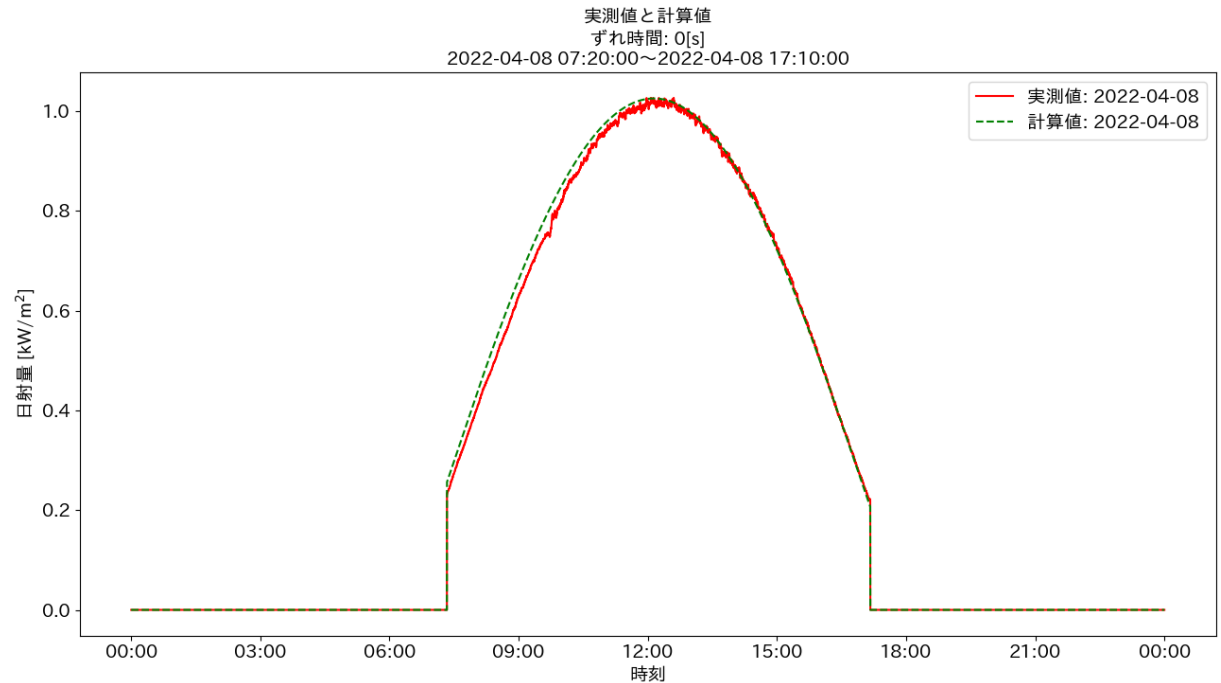


図 5: フィルタリング後の 2022 年 4 月 8 日の実測データと対応する理論データ

相互相関によるずれ時間の推定結果は 0 秒となり, 一見ずれ時間予測を改善できているように見えるが, 図 6に示すような理論データと概形が大きく異なる実測データを取る日を入力としてずれ時間を推定した際も, 推定結果が 0 秒となってしまう。

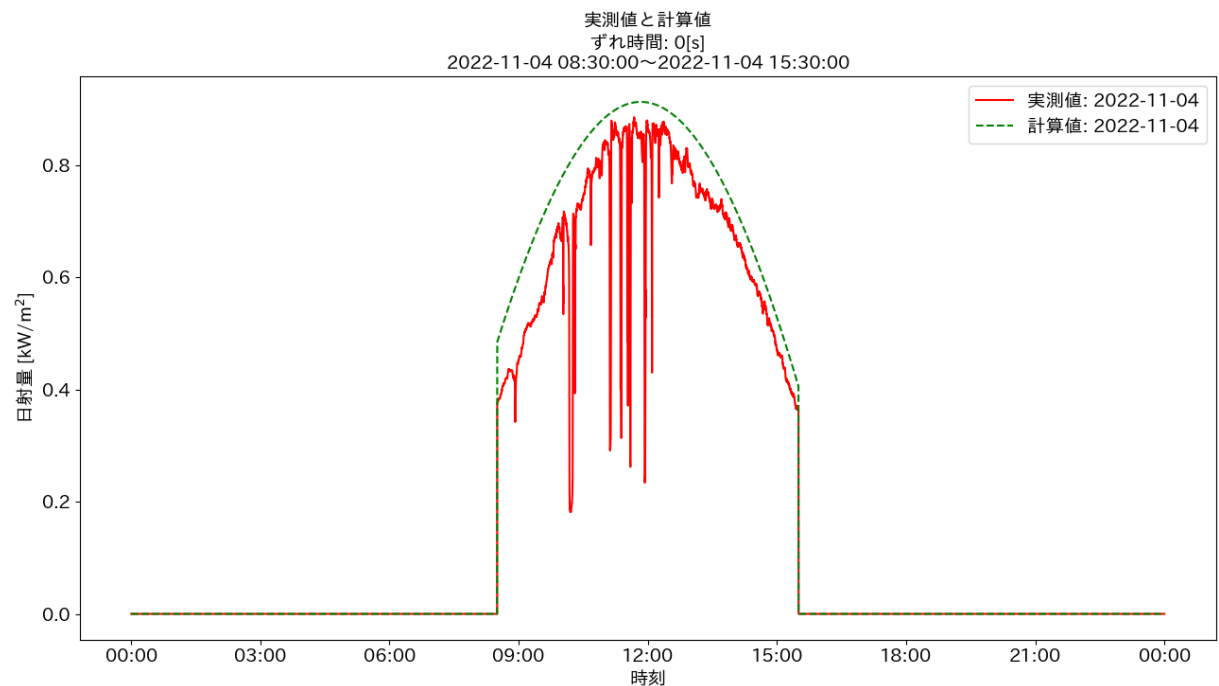


図 6: フィルタリング後の 2022 年 11 月 4 日の実測データと対応する理論データ

このように実測データの概形に関係なく, 推定結果が常に 0 秒になってしまう原因としては, フィルタリング時に指定時間帯外の日射量を全て 0 に置き換えたことで, 実測データと理論データの両方において指定時間帯の両端で日射量が 0 に急激に低下する箇所が現れており, この箇所が相互相関の計算時に強く作用することで, 指定時間帯内の概形に関わらず, ずれ時間の推定結果が常に 0 秒になってしまうと考えられる.

4.2 実測データに対する特定の日射量によるフィルタリングおよび日射量の減算

図 6 のように「実測データを, 指定した時間帯外の日射量が 0 になるように置き換える」という方法では, 指定時間帯の両端において日射量が 0 へと急激に低下する箇所が相互相関計算に強く影響し, その結果, 相互相関による推定結果が常に 0 秒になると推測された.

従って, 日射量が 0 へと急激に低下する箇所を作らずに実測データをフィルタリングする方法として, 図 7 に示すように実測データの日射量をしきい値の時にそのしきい値だけ実測データから減算する方法を考案した. ただし減算した時に負になる場合はゼロにしている.

2022/04/08 の実測データである図 7では図 5と同様のしきい値である日射量が 0.2kW/m^2 以上のデータのみをフィルタリングした結果である。

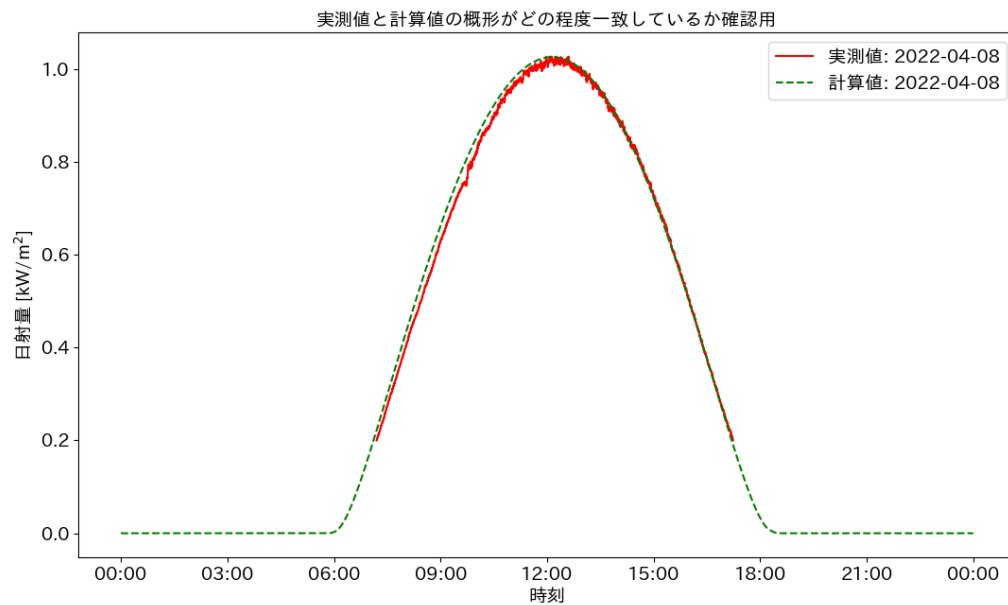


図 7: 日射量が 0.2kW/m^2 以上のみになるようフィルタリングした実測データ

そして、図 7の実測データからフィルタリングの際に指定した日射量の値である 0.2 だけ減算し、図 8に示すように下にスライドさせる。

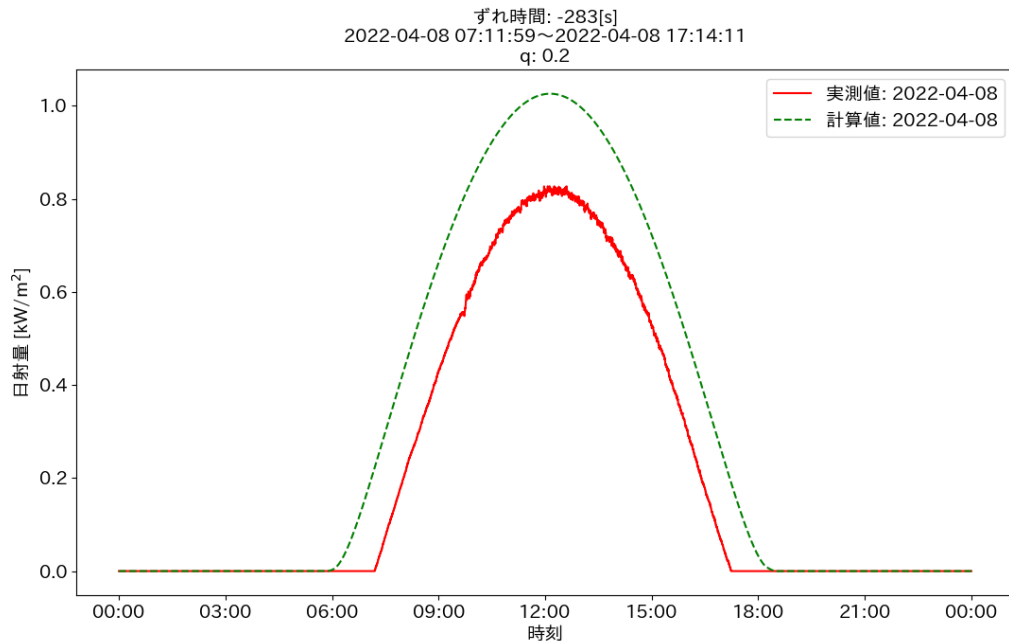


図 8: フィルタリングに使用した日射量分, 減算した後の実測データ

この方法により, フィルタリングの両端で相互相関計算に強く影響することによって, 推定結果のずれ時間が常に 0 秒になる問題を回避できることが期待される.

しかしながら, 2022 年 4 月 8 日のデータにおいては, 相互相関によるずれ時間の推定結果が -283 秒となり, 本前処理手法によってずれ時間の推定結果が改善できなかった.

5 まとめ

比較的正確に推定できる晴天時における 1 日分の実測データを入力として時刻合わせプログラムを実行した場合, 推定されたずれ時間は, 期待値よりも 165 秒程度ずれてることを述べた. 推定ずれ時間を 1 秒以内にするためには, 今後データの前処理方法あるいは日射量シミュレーションアルゴリズムをさらに改良することが必要がある.

参考文献

- [1] inoory, "2 つの信号間の遅延を推定する", <https://qiita.com/inoory/items/3ea2d447f6f1e8c40ffa>, 参照 Mar 20, 2023.

- [2] Sandia National Laboratories and pvlib python Development Team. , "pvlib python
pvlib python 0.9.5 documentation",
<https://pvlib-python.readthedocs.io/en/stable/> , 参照 Mar 20, 2023.