2023 年 3 月 17 日

実測値と理論値の時刻合わせに関する報告

祖父江匠真

1　概要

本研究室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中で、まつやま Re・再来館に設置されているものを除くと、ほとんどが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、パーソナルコンピュータの内部時計が経過時間とともにずれる現象が生じ、実測データのタイムスタンプが不正確になっている。

一方、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たオンライン環境で運用されており、さらにシステムを実行しているラズベリーパイが 24 時間ごとに再起動されているため、内部時計は常に同期され、計測データのタイムスタンプも正確である。

そこで、まつやま Re・再来館で得られた実測データを対象に、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のペアに対して、相互相関関数を用いた時刻合わせプログラムの実装が行われている。

まつやま Re・再来館の実測データはタイムスタンプのずれがないため、時刻合わせプログラムにこのデータを入力し、推定結果が 0 秒となることで、正しく動作していると判断できる。そのため、推定結果 0 秒を目標として実装が進められている。

現時点の進捗では、晴天時の 1 日分の実測データを用いて時刻合わせプログラムを実行した結果、アルゴリズムによっては 50〜200 秒程度のタイムスタンプのずれが生じており、さらなる改善が必要である。

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統計的手法である。具体的には、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。この数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。相互相関は、信号処理や時系列データ解析において、信号の遅延やズレを検出する目的で幅広く活用されている。

相互相関を用いた信号間のズレの検出方法は以下の手順で行われる。

1. 2 つの信号（信号 A および信号 B とする）を用意する。これらの信号は同じ長さであることが望ましいが、異なる場合でも適用が可能である。

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」と称する。ラグが 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、ラグが負である場合はその逆である。

3. それぞれのラグにおいて、信号 A と信号 B の重ね合わせた部分の積の和（内積）を算出する。これを相互相関関数と呼ぶ。

4. それぞれのラグごとに相互相関関数を計算し、相互相関関数が最大となる際のラグを特定する。この際に得られたラグは、信号 A と信号 B のズレ時間を示している。

例示すると、以下に示す 2 つの信号に対して相互相関を計算した場合、緑色の波形を右に 800 秒スライドさせた際に相互相関が最大となるため、これらの信号間のズレ時間は相互相関により 800 秒であると推定される。

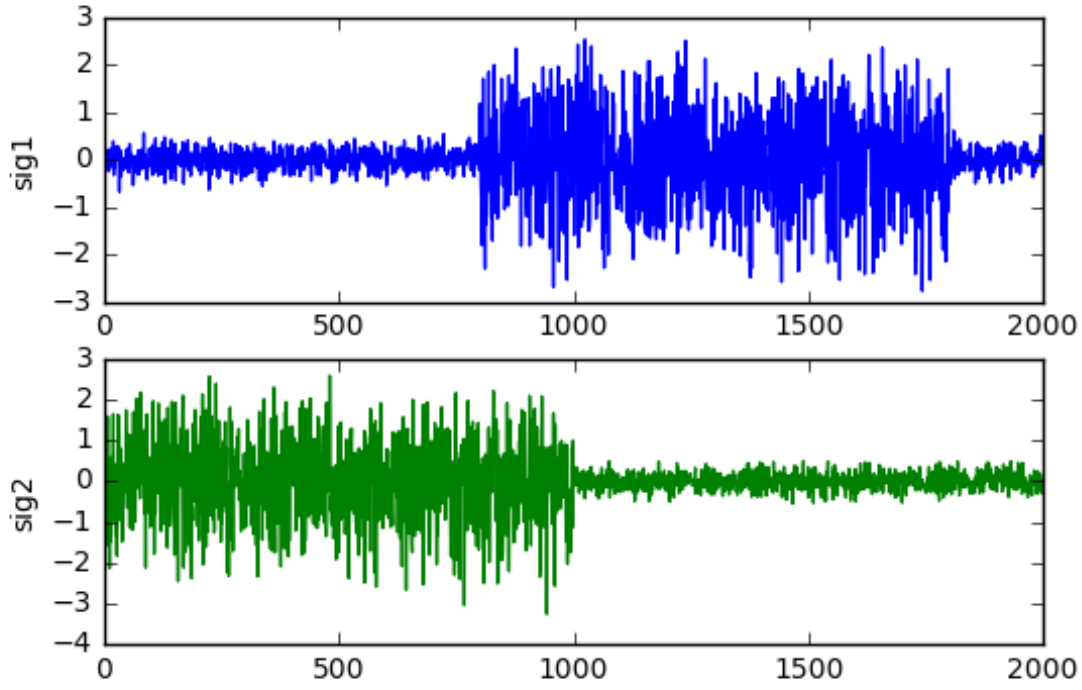

図 1: 相互相関によるズレ時間の推定例

相互相関を用いたズレの検出は、ノイズの影響を受けにくく、信号のパターンが不明な場合や信号の形状が複雑な場合でも、信号間の関係を正確に評価することが可能である。ただし、計算コストが高いため、大規模なデータやリアルタイ

ムの信号処理においては、効率的なアルゴリズムや近似手法の検討が必要となる。相互相関は、信号解析や時系列データの処理において、多くのアプリケーションで有益な手法であり、その活用によって信号間の関係性の把握や信号の特性解析が行われている。今後も、さらなる研究開発によって、相互相関を用いた信号処理技術の向上が期待される。

3 Pvlibを使用した理論データのシミュレーションについて

Pvlibは、Pythonで実装された太陽光発電システムのシミュレーションを行うためのライブラリである。晴天の日における日射量の推定は、以下の手順で進める。

1. **位置情報の設定**: まず、緯度、経度、標高、タイムゾーンなどを指定し、pvlib.location.Locati オブジェクトを生成する。このオブジェクトは、太陽位置や日射量の計算に使用される。

2. **日時データの準備**: 推定対象の日時範囲を指定し、対応する表データを作成する。

3. **太陽位置の計算**: pvlib.solarposition.get_solarposition 関数を用いて、指定された位置情報および日時データを基に、太陽の高度角と方位角を計算する。

4. **大気透過率の計算**: pvlib.clearsky.ineichen 関数を使用し、大気透過率を計算する。この関数は、指定された位置情報と太陽位置に基づいて、大気の透過率を推定し、全天日射量、直達日射量、拡散日射量を計算する。

5. **地表での日射量の推定**: 最後に、pvlib.irradiance.get_total_irradiance 関数を用いて、地表での日射量を推定する。この関数は、太陽位置、大気透過率、入射角などの情報に基づいて、地表での直達日射量、拡散日射量、反射日射量を計算する。

これらの手順に従って、晴天の日における地表での日射量を推定することができる。得られた日射量データは、実測データとの比較に利用される。

例示すると、以下に示されているのは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。

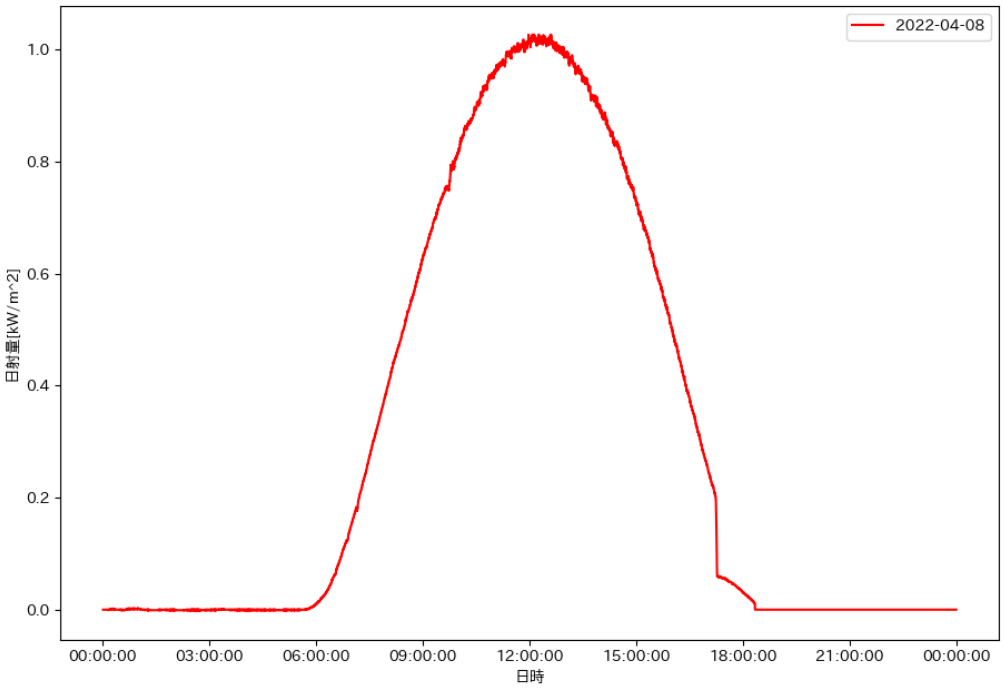

図 2: 実測データ

この実測データのタイムスタンプ列を Pvlib に入力し、晴天の日におけるシミュレーション結果を以下に示す。

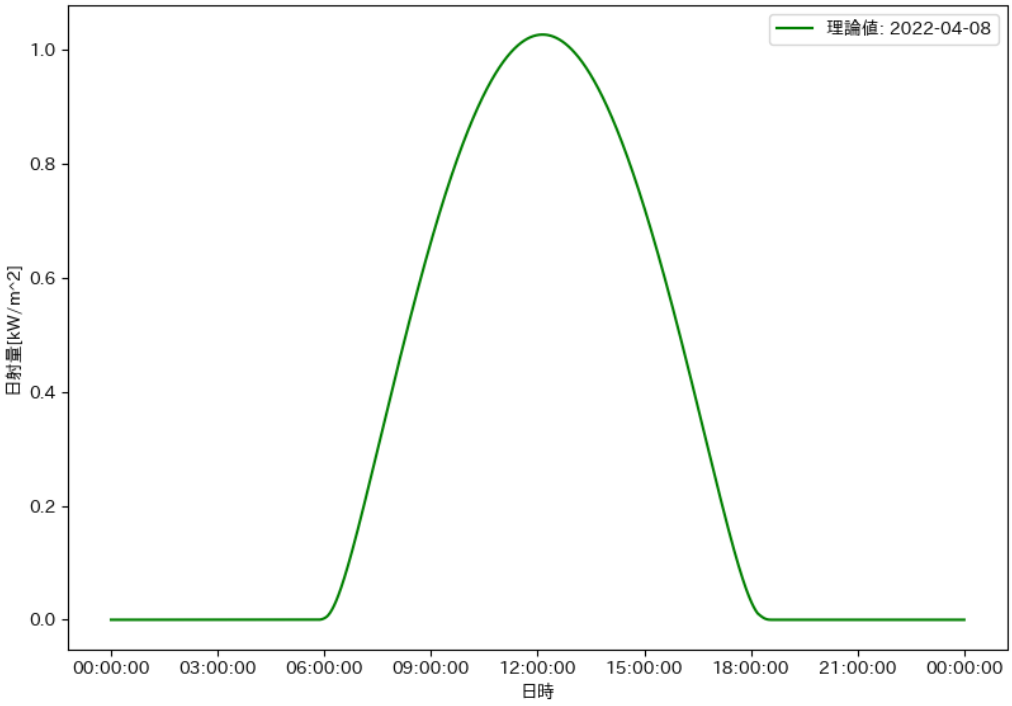

図 3: シミュレーション結果

実測データと理論データを合わせてプロットする。

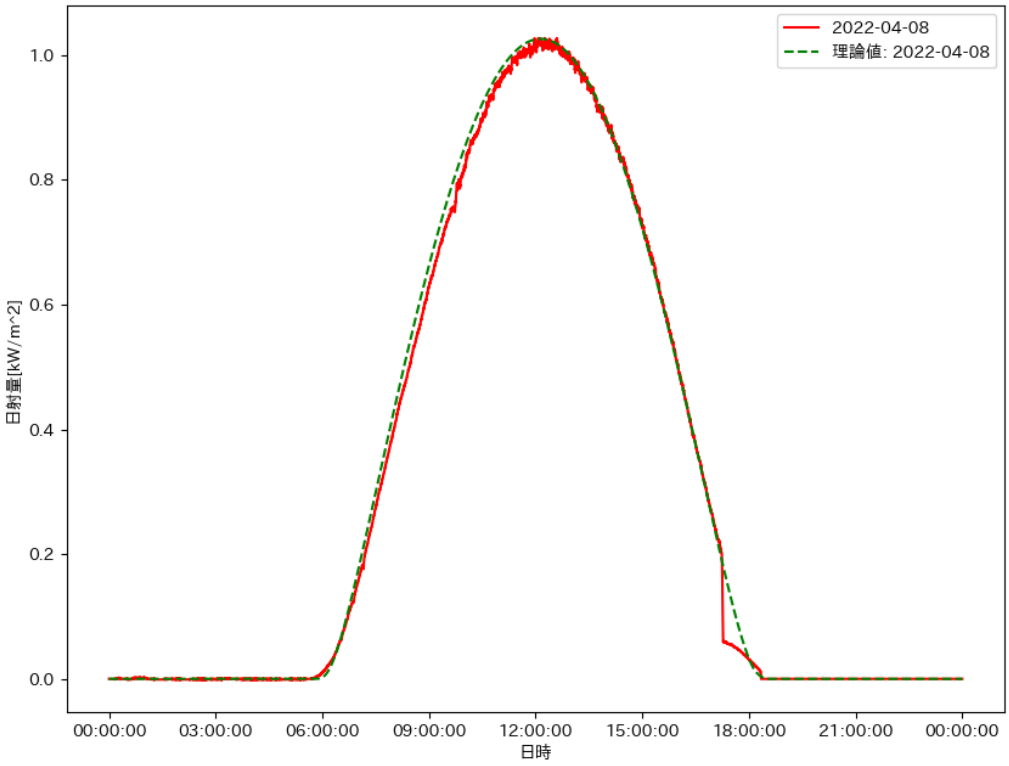

図 4: 実測データと理論データのプロット

このように晴天の日の実測データであれば、理論データは概ね一致する。

4　時刻合わせプログラムの実装に関する検討

現在実装されている時刻合わせプログラムでは、以下の手順で実測データと理論データのズレ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リサンプリングするなど、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太陽光発電シミュレーターである Pvlib に入力し、晴天の日における松山 Re・再来館の日射量を予測する。
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ズレ時間を示し、インデックスに対応するラグがズレ時間となる。

例えば、2022 年 4 月 8 日の実測データに対してズレ時間を計算した結果、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、ズレ時間の推定結果は 165 秒となった。

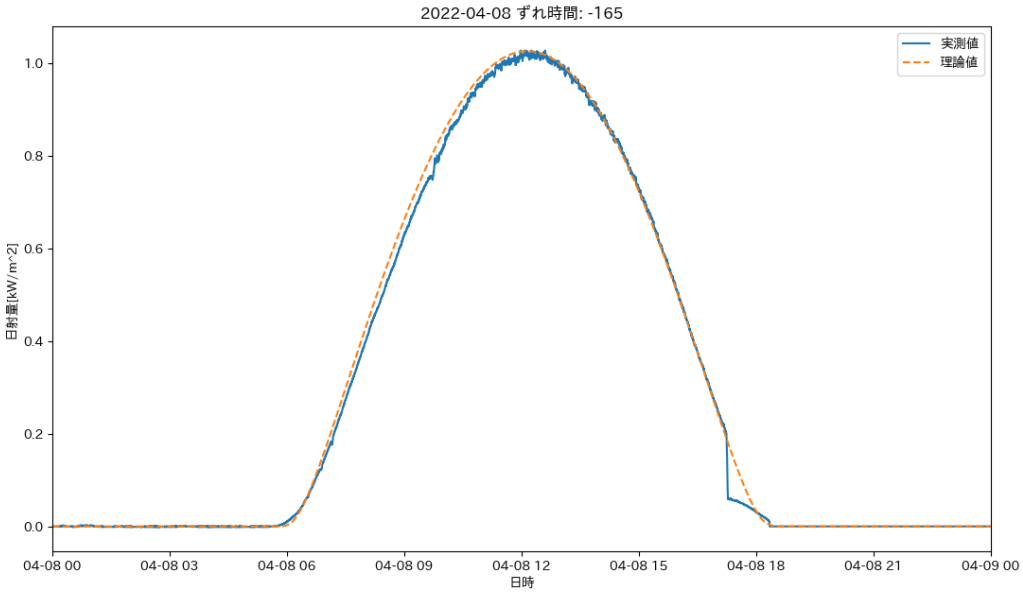

図 5: 2022 年 4 月 8 日の実測データと理論データの相互相関

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ズレ時間の推定結果が変化するため、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推定結果について検討する。

4.1 指定期間外の日射量を 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推定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。

以下に示す図は、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らない期間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、指定期間外の実測データと理論データの日射量を 0 に置き換えている。

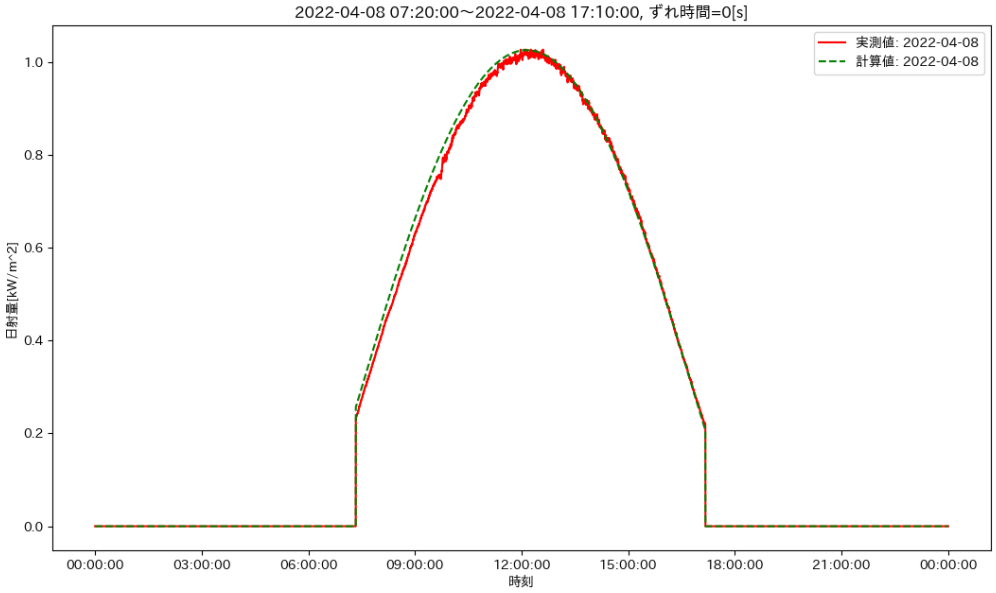

図 6: フィルタリング後の 2022 年 4 月 8 日の実測データと理論データの相互相関

相互相関によるズレ時間の推定結果は 0 秒となり、一見うまくズレ時間を予測できているように見えるが、次の図に示すような理論データと大きく概形が異なる実測データとの相互相関でも同様にズレ時間の推定結果が 0 秒となってしまった。

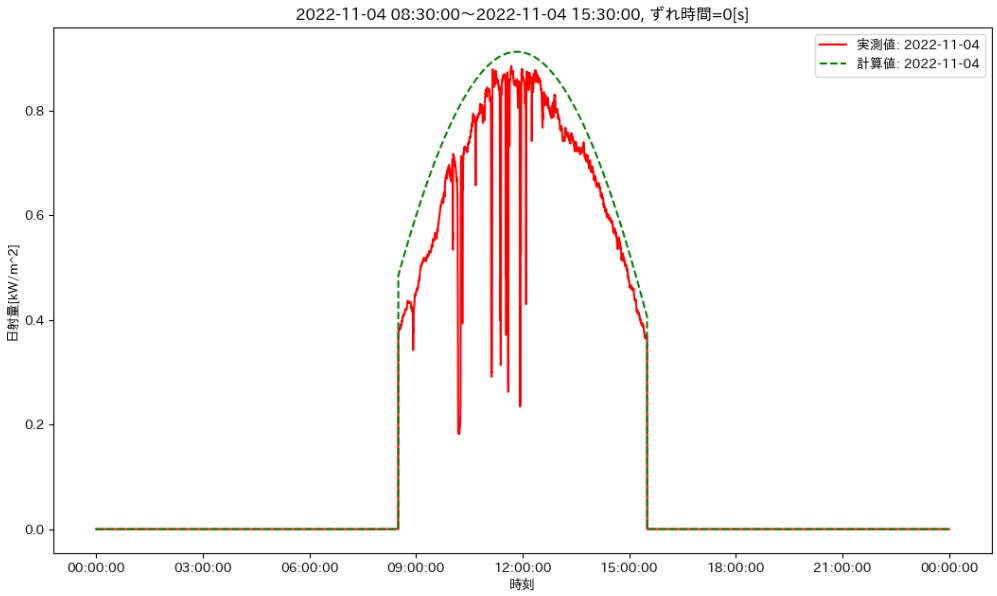

図 7: 概形が異なる実測データと理論データの相互相関

実測データの概形に関係なく、推定結果が常に 0 秒となってしまう原因としては、フィルタリング時に指定期間外の日射量を全て 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期間の両端で日射量が 0 に急激に低下する箇所が現れ、この箇所が相互相関の計算時に強く作用することで、指定期間内の概形に関わらず、ズレ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

「実測データを、指定した期間外の日射量が 0 になるように置き換える」という方法では、指定期間の両端において日射量が 0 に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 へ急激に低下する箇所を作らない手法として、以下の図に示すように実測データの日射量を特定の値以下に切り落とすフィルタリングを行い、さらに指定した値だけ実測データを下方向にスライドさせる方法を採用した。

以下に示すのは 2022/04/08 の実測データと理論データであるが、この実測データを次の図に示すように日射量が 0.2kW/m^2 以上のみになるようフィルタリングする

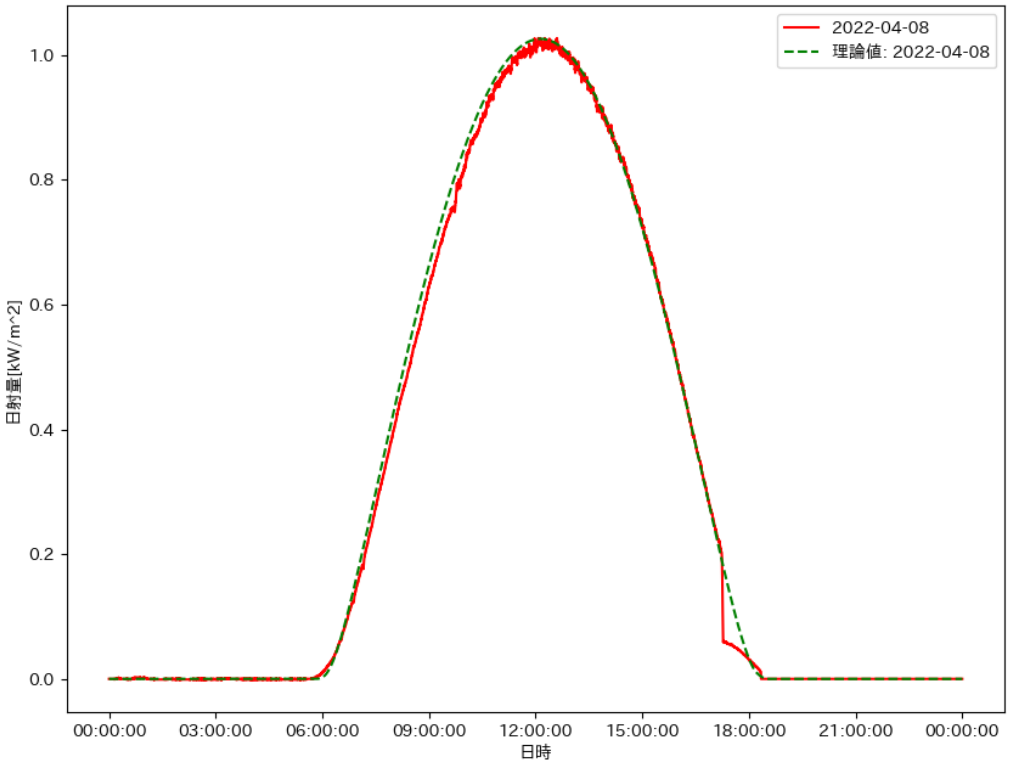

図 8: フィルタリング前の 2022 年 4 月 8 日の実測データ

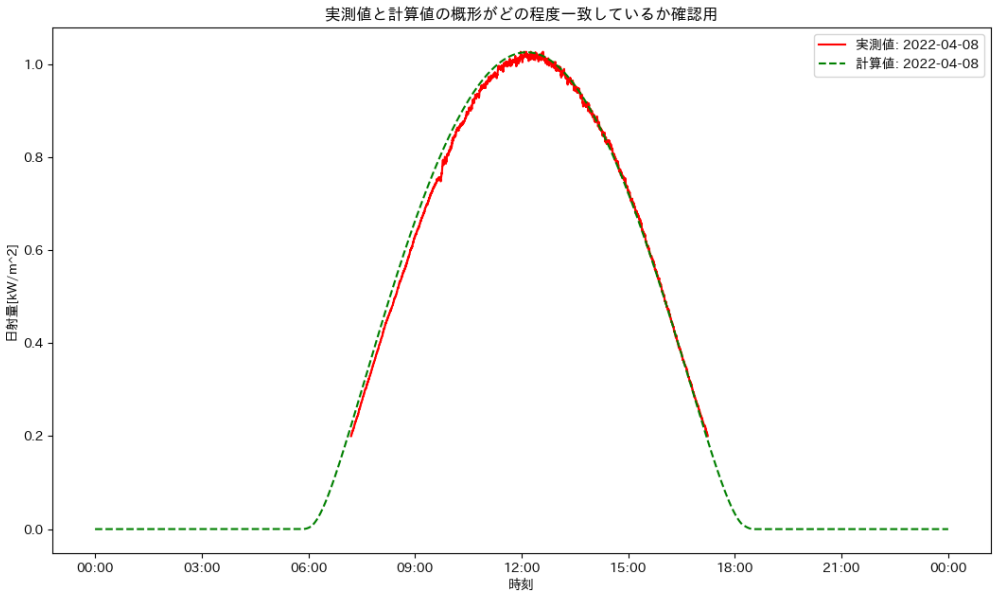

図 9: 日射量が 0.2kW/m^2 以上のみになるようフィルタリングした実測データ

そして、上の図の実測データから 0.2 を減算し、以下の図のようにする

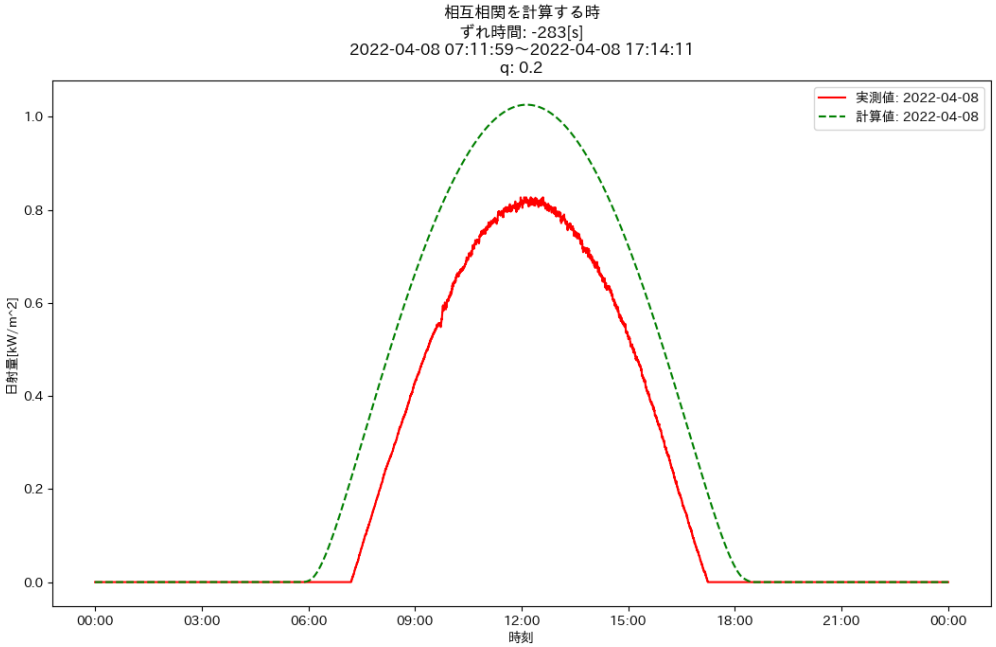

図 10: 減算後の実測データ

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、推定結果のズレ時間が常に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、相互相関によるズレ時間の推定結果が-283秒となり、本前処理手法によってズレ時間の推定結果が改善されることは確認できなかった。

5　まとめ

晴天時における1日分の実測データを用いた場合、推定されたズレ時間は、真の値である0秒から平均して50〜200秒程度ずれていることが明らかとなった。この結果から、データの前処理方法や日射量シミュレーションアルゴリズムの改良が必要であることが示唆されている。

具体的には、高精度な相関計算手法の導入や、データ前処理手法の改善が検討されるべきである。さらに、異なる日や天候条件下における実測データを用いて、プログラムの汎用性を評価することも重要であると考えられる。